

Article

CESA-MCFormer: An Efficient Transformer Network for Hyperspectral Image Classification by Eliminating Redundant Information

Shukai Liu , Changqing Yin * and Huijuan Zhang

School of Software, Tongji University, Shanghai 201800, China; 2131505@tongji.edu.cn (S.L.); mszhj@tongji.edu.cn (H.Z.)

* Correspondence: yinchangqing@tongji.edu.cn

Abstract: Hyperspectral image (HSI) classification is a highly challenging task, particularly in fields like crop yield prediction and agricultural infrastructure detection. These applications often involve complex image types, such as soil, vegetation, water bodies, and urban structures, encompassing a variety of surface features. In HSI, the strong correlation between adjacent bands leads to redundancy in spectral information, while using image patches as the basic unit of classification causes redundancy in spatial information. To more effectively extract key information from this massive redundancy for classification, we innovatively proposed the CESA-MCFormer model, building upon the transformer architecture with the introduction of the Center Enhanced Spatial Attention (CESA) module and Morphological Convolution (MC). The CESA module combines hard coding and soft coding to provide the model with prior spatial information before the mixing of spatial features, introducing comprehensive spatial information. MC employs a series of learnable pooling operations, not only extracting key details in both spatial and spectral dimensions but also effectively merging this information. By integrating the CESA module and MC, the CESA-MCFormer model employs a “Selection–Extraction” feature processing strategy, enabling it to achieve precise classification with minimal samples, without relying on dimension reduction techniques such as PCA. To thoroughly evaluate our method, we conducted extensive experiments on the IP, UP, and Chikusei datasets, comparing our method with the latest advanced approaches. The experimental results demonstrate that the CESA-MCFormer achieved outstanding performance on all three test datasets, with Kappa coefficients of 96.38%, 98.24%, and 99.53%, respectively.

Keywords: hyperspectral image classification; transformer; spatial attention; morphological convolution



Citation: Liu, S.; Yin, C.; Zhang, H. CESA-MCFormer: An Efficient Transformer Network for Hyperspectral Image Classification by Eliminating Redundant Information. *Sensors* **2024**, *24*, 1187. <https://doi.org/10.3390/s24041187>

Academic Editor: Robert Knuteson

Received: 3 December 2023

Revised: 5 February 2024

Accepted: 7 February 2024

Published: 11 February 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the continuous advancement of spectral imaging technology, hyperspectral data have achieved significant improvements in both spatial and spectral resolution. Compared to multispectral and RGB images, hyperspectral images (HSI) possess narrower bandwidths and a greater number of bands, allowing them to provide more detailed and continuous spectral information [1–3]. As a result, HSI have demonstrated tremendous potential in various earth observation fields [4] such as precision agriculture [5,6], urban planning [7,8], environmental management [9–11], and target detection [12–15]. Consequently, research on HSI classification has rapidly progressed.

While traditional HSI classification methods, such as nearest neighbor [16], Bayesian estimation [17], multinomial logistic regression [18,19], and Support Vector Machine (SVM) [20–23], have their merits in certain scenarios, these methods often have limitations in data representation and fitting capability, struggling to produce satisfactory classification results on more complex datasets. In contrast, in recent years, methods underpinned by

deep learning, thanks to their outstanding feature extraction capabilities, have gradually become the focus of research in HSI classification.

Convolutional Neural Networks (CNNs) dominate the field of deep learning and are capable of accumulating in-depth spatial features through layered convolution. As such, CNNs have been extensively applied to and researched in the classification of HSI [24,25]. Notably, Roy and colleagues [26] introduced a model named HybridSN. This model initially employs a 3D-CNN to extract spatial–spectral features from spectral bands that have undergone PCA dimensionality reduction. Subsequently, it uses a 2D-CNN to delve deeper into more abstract spatial feature hierarchies. Compared to a 3D-CNN, this hybrid approach simplifies the model architecture while effectively merging spatial and spectral information. Building on this, subsequent researchers have incorporated one-dimensional convolution based on central pixels to compensate for spectral information that might be lost after PCA reduction. Examples of this approach include the Cubic-CNN model proposed by J. Wang et al. [27] and the JigsawHSI model introduced by Moraga and others [28].

The Vision Transformer (ViT) model [29], which evolved from the natural language processing (NLP) domain, has also increasingly become a focal point in the field of deep learning. The ViT model segments images into fixed-size patches and leverages embedding techniques to obtain a broader receptive field. Furthermore, with the help of multi-head attention mechanisms, it adeptly captures the dependencies between different patches, thereby achieving higher processing efficiency and remarkable image recognition performance. Consequently, numerous studies have been dedicated to exploring the application of this model in HSI classification. For instance, a research team proposed the Spatial–Spectral Transformer (SST) model in [30]. They utilized VGGNet [31], from which several convolutional layers were removed, as a feature extractor to capture spatial characteristics from hyperspectral images. Subsequently, they employed the DenseTransformer to discern relationships between spectral sequences and used a multi-layer perceptron for the final classification task. Qing et al. introduced SATNet in [32], which effectively captures spectral continuity by adding position encoding vectors and learnable embedding vectors. Meanwhile, Hong and colleagues presented the SpectralFormer (SF) model in [33]. This model adopts the Group-wise Spectral Embedding (GSE) module to encode adjacent spectra, ensuring spectral information continuity, and utilizes the Cross-layer Adaptive Fusion (CAF) technique to minimize information loss during hierarchical transmission. X. He and their team introduced the SSFTT network in [34]. This model significantly simplifies the SST structure and incorporates Gaussian-weighted feature tagging for feature transformation, thus reducing computational complexity while enhancing classification performance. In recent studies, researchers have continued to explore more lightweight and effective methods for feature fusion and extraction based on the transformer architecture. For instance, Xuming Zhang and others proposed the CLMSA and PLMSA modules [35], while Shichao Zhang and colleagues introduced the ELS2T [36].

Due to the high correlation between adjacent bands in HSI, there is a significant amount of redundant information within HSI. Commonly, to mitigate the impact of this redundancy, the methods mentioned above [26,30,31] preprocess HSI using Principal Component Analysis (PCA). However, not using PCA leads to a significant decrease in model prediction accuracy, highlighting the model's deficiency in extracting key spectral information. As an unlearnable dimensionality reduction technique, PCA's process is often irreversible and can lead to information loss, such as the loss of spectral continuity [37]. Models reliant on PCA may thus produce suboptimal results. In transfer learning or few-shot image classification tasks, HSI are required to feed a large number of channels into the model to preserve as much original information as possible. This input of extensive channel data elevates the demands on feature extractors, necessitating their capability to efficiently process and extract key information from these numerous channels. Moreover, different datasets might require dimensionality reduction to different extents, making the selection of appropriate dimensions for each dataset a time-consuming operation. Therefore, we propose the CESA-MCFormer, which effectively extracts key information from HSI under

conditions of limited samples and numerous channels, achieving higher classification accuracy in downstream tasks without relying on PCA for dimension reduction. To achieve this, we have incorporated attention mechanisms and mathematical morphology.

Attention mechanisms have been extensively applied in various domains of machine learning and artificial intelligence. Hu et al. [38] introduced a “channel attention module” in their SE network structure to capture inter-channel dependencies. Woo et al. [39] proposed CBAM, which combines channel and spatial attention, adaptively learning weights in both dimensions to enhance the network’s expressive power and robustness. Meanwhile, Zhong et al. [40] presented a deep convolutional neural network model, integrating both a “global attention mechanism” and a “local attention mechanism” in sequence to capture both global and local contextual information. Inspired by these advancements, researchers began incorporating spatial attention into HSI classification. Several studies [41–44] combine spectral and spatial attention mechanisms, enabling adaptive selection of key features within HSI. However, in HSI classification, a common practice is to segment the HSI into small patches and classify each patch based on its center pixel. Yet, these methods do not sufficiently consider the importance of the center pixel. This approach makes the information provided by the center pixel crucial. Recent studies have recognized this, such as those cited in [45,46], which employed the Central Attention Module (CAM). This module determines feature weights by analyzing the correlation of each pixel with the center pixel. However, considering the phenomena of same material, different spectra and different materials, same spectra in HSI, relying solely on similarity to the center pixel for weight allocation might overlook important spatial information provided by other pixels. Therefore, effectively weighting the center pixel while taking global spatial information into account remains a challenge.

Mathematical Morphology (MM) primarily focuses on studying the characteristics of object morphology, processing and describing object shapes and structures using mathematical tools such as set theory, topology, and functional analysis [47]. In previous HSI classification tasks, researchers often utilized attribute profiles (APs) and extended morphological profiles (EPs) to extract spatial features more effectively [48–51]. However, this approach typically requires many structuring elements (SEs), which are non-trainable and thus unable to effectively capture dynamic feature changes. To overcome these limitations, Roy et al. proposed the Morphological Transformer (morphFormer) in [52], combining trainable MM operations with transformers, thereby enhancing the interaction between HSI features and the CLS token through learnable pooling operations. However, this method involves simultaneous dilation and erosion of spatio-spectral features, where each SE introduces a significant number of parameters. This not only risks losing fine-grained feature information during feature selection but also leads to model overfitting and reduced robustness, especially in scenarios with limited data. Hence, there is substantial room for improvement in the application of MM in HSI classification.

The core contributions of this study are as follows:

- We designed a flexible and efficient Center Enhanced Spatial Attention (CESA) module specifically for hyperspectral image feature extraction. This module can be easily integrated into various models, enhancing focus on areas around the center pixel while considering global spatial information;
- We introduced Morphological Convolution (MC) to replace the traditional linear layer feature extraction mechanism in the transformer encoder. MC selects fine-grained features through a strategy of separating and then integrating spatial and spectral features, significantly reducing the number of parameters and enhancing the model’s robustness;
- Utilizing these modules, we developed the CESA-MCFormer feature extractor, capable of effectively extracting key features from a multitude of channels, supporting various downstream classification tasks. We conducted in-depth ablation experiments to provide practical and theoretical insights for researchers exploring and applying similar modules.

The rest of the paper is organized as follows: In Section 2, we provide an overview of the CESA-MCFormer’s overall framework and detail our proposed CESA and MC modules. Section 3 describes the experimental datasets, results under various parameter settings, and an analysis of the model parameters. Finally, Section 4 concludes with our research findings.

2. Methodology

The architecture of CESA-MCFormer is illustrated in Figure 1. For an HSI patch of size $c \times h \times w$, the spectral continuity information is initially extracted through a 3D–2D Conv Block [52], and the dimensionality is transformed to 64. Subsequently, the HSI feature of size $64 \times h \times w$ is fed into the Emb Block for mixing spatial and spectral features, generating a 64×64 feature matrix. Then, a learnable CLS token, initialized to zero, is introduced for feature aggregation, along with a learnable matrix of size 65×64 , also initialized to zero, for spatial–spectral position encoding. After combining the feature map with the position encoding, it is passed through multiple iterations of the Transformer Encoder for deep feature extraction, and the extracted features are then input into the Classifier Head for downstream classification tasks. Next, we will provide a detailed introduction to the Emb Block and Transformer Encoder.

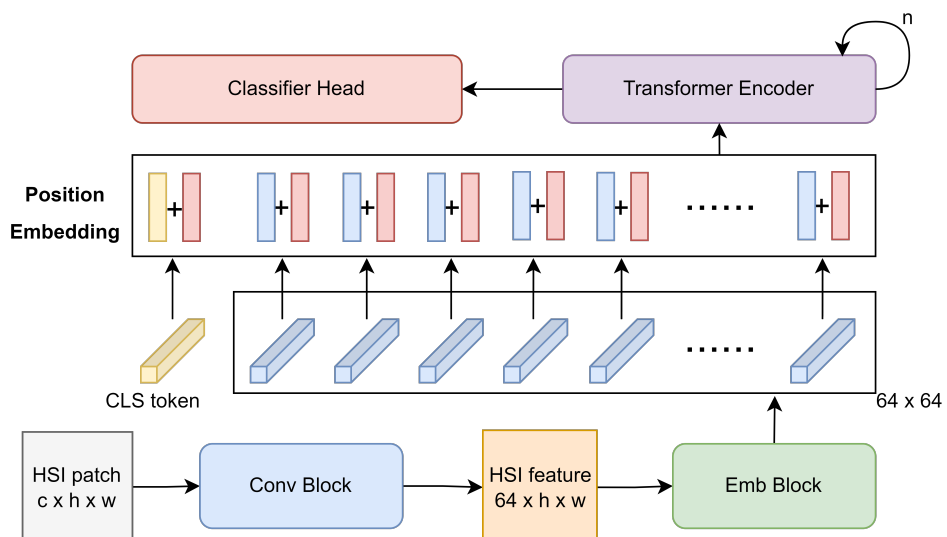


Figure 1. Overall architecture of CESA-MCFormer.

2.1. Emb Block

Given that the HSI patches input into the model are generally small (with a spatial size of 11×11 adopted in this study), we introduce the Emb Block to directly mix and encode global spatial features. This approach equips the model with a global receptive field before deep feature extraction, as illustrated in Figure 2. Since the information provided by the central pixel of the HSI patch is crucial, CESA is first used to weight information at different positions, aiding the model in actively eliminating redundant information. Then, we introduce a learnable weight matrix $W_a \in \mathbb{R}^{64 \times 64}$ initialized using Xavier normal initialization, composed of 64 scoring vectors. By calculating the dot product between HSI features and each scoring vector, we score the features of each pixel. The scores are then transformed into mixing weights using the softmax function. Another learnable weight matrix $W_b \in \mathbb{R}^{64 \times 64}$, initialized in the same manner, is introduced to remap the HSI features of each pixel point through matrix multiplication. Finally, by multiplying the two matrices, we mix the spatial features based on the mixing weights to obtain the final feature encoding matrix.

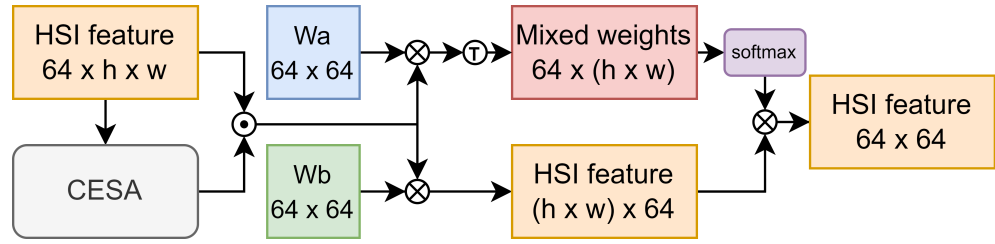


Figure 2. Overall architecture of EMB Block. The symbol “T” represents the transpose of a matrix. The symbol “.” represents element-wise multiplication of matrices, and the symbol “x” denotes matrix multiplication.

The overall architecture of CESA is illustrated in Figure 3. To comprehensively consider both global information and the importance of the central pixel, we meticulously designed two modules: Soft CESA and Hard CESA. Hard CESA, a non-learnable module, statically assigns higher weights to pixels closer to the center. Soft CESA, conversely, is a learnable module that uses global information as a reference, enabling the model to adaptively select more important spatial information. This design aims to effectively integrate both global and local information, enhancing the overall performance of the model.

Specifically, CESA takes an HSI or its feature map (F_{in}) as input. Both Hard CESA and Soft CESA calculate and output the hard probabilistic diversity map (M_h) and the soft probabilistic diversity map (M_s), respectively. The M_h and M_s maps are added together and then expanded along the channel dimension to match the size of F_{in} before being element-wise multiplied with F_{in} . Finally, an optional simple convolutional module is used to adjust the dimensions of the output feature (F_{out}). The implementation details of both Hard CESA and Soft CESA are presented in the following sections.

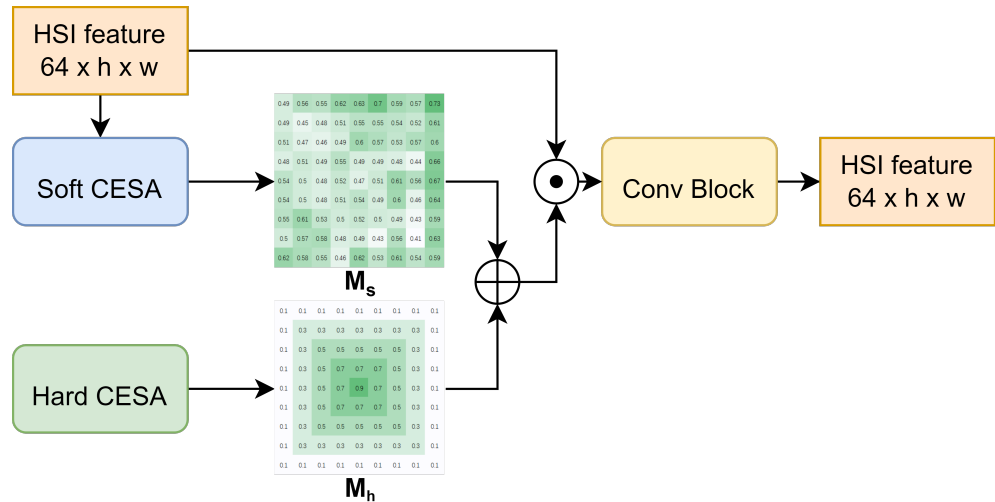


Figure 3. Overall architecture of CESA.

2.1.1. Hard CESA

The output M_h of Hard CESA depends only on the size of F_{in} and the hyperparameter K . For a pixel q in F_{in} , its position coordinates are defined as (x, y) , and its spectral features are denoted by $p = [p_1, p_2, \dots, p_c] \in \mathbb{R}^{1 \times c}$. We define q_c as the center pixel of the patch, and its coordinates are defined as (x_c, y_c) . The distance d between q_c and q is defined by the following Equation (1):

$$d = \max(|x - x_c|, |y - y_c|) \quad (1)$$

q_w for pixel q is defined as follows in Equation (2):

$$q_w = K - \frac{d}{h} \times (2K - 1) \quad (2)$$

where h is the length of the F_{in} . The hyperparameter K ($K \in [0.5, 1)$) controls the importance gap between the center and edge pixels. As K becomes larger, the weight of the center pixels becomes larger and the weight of the edge pixels becomes smaller. When $K = 0.5$, all pixels in the patch have equal weights, and therefore Hard CESA will not have any effect.

2.1.2. Soft CESA

As shown in Figure 4, Soft CESA processes F_{in} into three feature maps, F_1 , F_2 , and F_3 . F_1 and F_2 are used to represent the overall features of F_{in} , while F_3 is used to introduce the feature of the center pixel.

Specifically, for a pixel q in F_{in} , its position coordinates are defined as (x, y) , and its spectral features are denoted by $p = [p_1, p_2, \dots, p_c] \in \mathbb{R}^{1 \times c}$. The value of F_1 at position (x, y) , denoted as $m_1(x, y)$, can be calculated as follows:

$$m_1(x, y) = \max(p_1, p_2, \dots, p_c) \quad (3)$$

The value of F_2 at position (x, y) , denoted as $m_2(x, y)$, can be calculated as follows:

$$m_2(x, y) = \frac{1}{c} \sum_{i=1}^c p_i \quad (4)$$

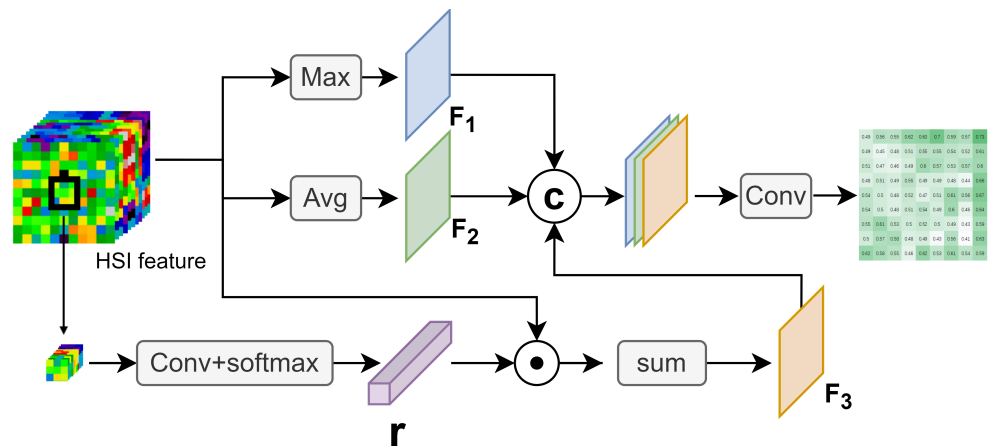


Figure 4. Overall architecture of Soft CESA.

To effectively extract the center overall feature in Soft CESA, we introduce a central weight vector $r = [r_1, r_2, \dots, r_c] \in \mathbb{R}^c$ to weight F_{in} . Therefore, the value of F_3 at position (x, y) can be represented as follows:

$$m_3(x, y) = \frac{1}{c} \sum_{i=1}^c p_i r_i \quad (5)$$

We extract the spectral features of the central pixel and its eight neighboring pixels, flatten them into a one-dimensional vector, and use this as the central feature vector $v_c \in \mathbb{R}^{9c}$. We introduce a matrix $A_b \in \mathbb{R}^{c \times (9c)}$ composed of c learnable spectral feature encoding vectors and a vector $l_b \in \mathbb{R}^c$ comprised of c bias terms to weight and sum the spectral bands at each position. The specific formula for calculating the corresponding r is as follows:

$$r = \text{softmax}(A_b v_c + l_b) \quad (6)$$

Finally, we concatenate $F1$, $F2$, and $F3$ along the channel dimension to form the final feature matrix F . After passing through a convolutional layer with a kernel size of 3×3 , a softmax activation function is applied to produce the final soft probabilistic diversity map M_s :

$$M_s = \text{sigmoid}(\text{conv}(F)) \quad (7)$$

It can be observed that the entire CESA model uses only $c \times (9c + 1) + (9 \times 3 + 1)$ learnable parameters, and the parameter c can be flexibly adjusted through the preceding conv Block. This means that the computational cost of CESA is very low, allowing it to be easily embedded into other models without significantly increasing the complexity of the original model.

2.2. Transformer Encoder

The primary function of the Transformer Encoder module is to extract deep spatial–spectral features through multiple iterations. As shown in Figure 5, in each iteration, HSI features are first processed through Spectral Morph and Spatial Morph for feature selection and extraction, followed by an interaction with the CLS token through Cross Attention, aggregating the spatial–spectral features into the CLS token.

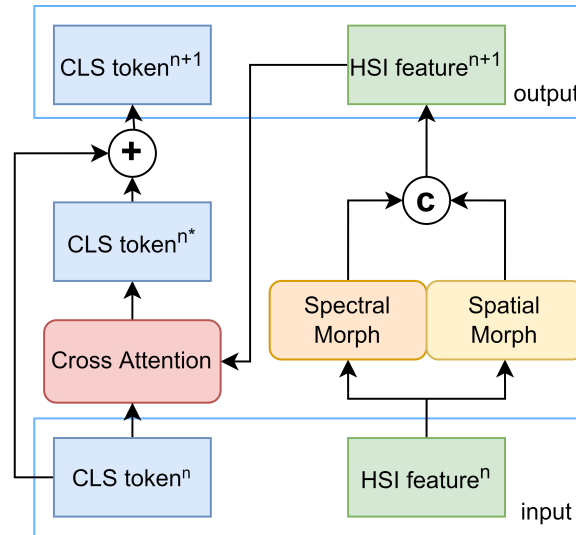


Figure 5. Overall architecture of Transformer Encoder.

To capture multi-dimensional features, we employ a multi-head attention mechanism in Cross Attention [29]. The input CLS token and HSI features are uniformly divided into eight parts along the spectral feature dimension, each with a feature length of eight. For each segmented feature, the CLS token serves as the query $q \in \mathbb{R}^{1 \times 8}$, and the matrix formed by concatenating the CLS token and HSI features is used as the key and value $k, v \in \mathbb{R}^{65 \times 8}$. The calculation method for Cross Attention is as follows:

$$X_{attn} = \text{dropout} \left(\text{softmax} \left(\frac{(q \times w_q) \times (k \times w_k)^T}{\sqrt{l}} \right) \right) \times (v \times w_v) \quad (8)$$

In this process, w_q , w_k , and $w_v \in \mathbb{R}^{8 \times 8}$ are all learnable parameters, while l is the feature length, set to eight in this study. After obtaining all eight groups of X_{attn} , they are reassembled along the spectral feature dimension. Then, they are processed through a linear layer followed by a dropout layer, resulting in the updated $CLS \ token^{n*} \in \mathbb{R}^{1 \times 64}$. This is then added to the input $CLS \ token^n$ to produce the final $CLS \ token^{n+1}$.

Inspired by morphFormer [52], we have also incorporated a Spectral Morph Block and a Spatial Morph Block into our model. The overall architecture of these two modules is

identical, as shown in Figure 6. Both modules process HSI features through erosion and dilation modules. After processing, the Spectral Morph block utilizes a 1×1 convolution layer (corresponding to the blue Conv block in Figure 6) to extract deeper channel information, while the Spatial Morph block uses a 3×3 convolution layer to aggregate more channel information. The Morphological Convolution (MC) we propose is represented by the erosion and dilation modules in Figure 6. Next, we will elaborate on how MC is implemented.

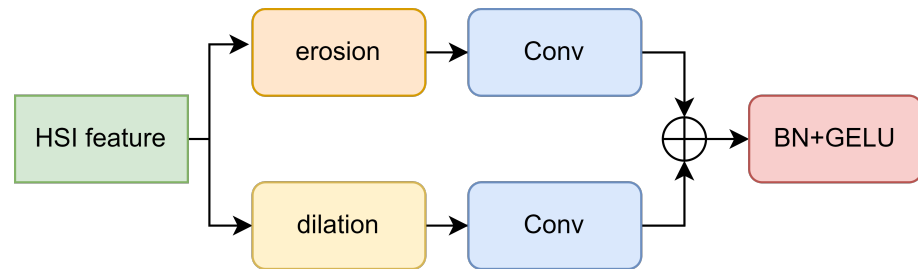


Figure 6. Overall architecture of the Spectral Morph Block and Spatial Morph Block.

MC's primary function is to eliminate redundant data during the feature extraction process, ensuring that as the depth of the encoder increases, the HSI feature retains only pivotal information. To accomplish this, we apply multiple learnable Structuring Elements (SEs) to the HS feature for morphological convolution. Through dilation, we select maximum values from adjacent features, emphasizing boundary details. In contrast, erosion allows us to identify the minimum values, effectively attenuating minor details. Additionally, directly employing SEs might inflate the parameter count, posing overfitting risks. To mitigate this, we separate the spectral and spatial SEs, significantly reducing parameters and thereby boosting the model's resilience.

Specifically, when using SEs with a spatial size of $k \times k$, to maintain the consistency of input and output dimensions of the module, we first reshape the spatial dimension of the HS feature into two dimensions and then pad its boundaries, resulting in the feature matrix $H \in \mathbb{R}^{(8+(k-1)) \times (8+(k-1)) \times 64}$. Next, by adopting a sliding window with a stride of 1, we segment H into 64 sub-blocks of size $k \times k \times 64$, referred to as X_{patch} . Subsequently, we further decompose X_{patch} in both spatial and spectral directions. Spatially, X_{patch} is divided into $k \times k$ vectors of dimension 64, denoted as $\{X_{a1}, X_{a2}, \dots, X_{ak \times k}\}$. Spectrally, X_{patch} is parsed into 64 vectors of dimension $k \times k$, represented as $\{X_{b1}, X_{b2}, \dots, X_{b64}\}$. We then introduce multiple SEs groups, where each group consists of a spatial vector of length $k \times k$ and a spectral vector of length 64. For simplicity, we name one group of SEs W , with its spectral vector labeled W_a and the spatial vector W_b . For any given X_{patch} and W , the dilation operation of the morphological convolution is shown in Figure 7.

First, we add each segmented feature vector to the corresponding W_a and W_b at their respective positions, then take the maximum value to obtain $h_{dil} \in \mathbb{R}^1$:

$$h_{dil}(X_{ai}, W_a) = \max_{j \in \{1, 2, \dots, 64\}} (X_{ai}(j) + W_a(j)) \quad (9)$$

$$h_{dil}(X_{bj}, W_b) = \max_{i \in \{1, 2, \dots, k \times k\}} (X_{bj}(i) + W_b(i)) \quad (10)$$

Then, we introduce two learnable vectors $h_a \in \mathbb{R}^{(k \times k)}$ and $h_b \in \mathbb{R}^{64}$, along with two learnable bias terms β_a and β_b . We concatenate the results from the previous step into two one-dimensional vectors, which are then dot-multiplied with h_a and h_b , respectively, and added to β_a and β_b , resulting in $g_{dil} \in \mathbb{R}^1$:

$$g_{dil}(X_a, W_a) = \text{concat}(h_{dil}(X_{a1}, W_a), h_{dil}(X_{a2}, W_a), \dots, h_{dil}(X_{ak \times k}, W_a)) \times h_a + \beta_a \quad (11)$$

$$g_{\text{dil}}(X_b, W_b) = \text{concat}(h_{\text{dil}}(X_{b1}, W_b), h_{\text{dil}}(X_{b2}, W_b), \dots, h_{\text{dil}}(X_{b64}, W_b)) \times h_b + \beta_b \quad (12)$$

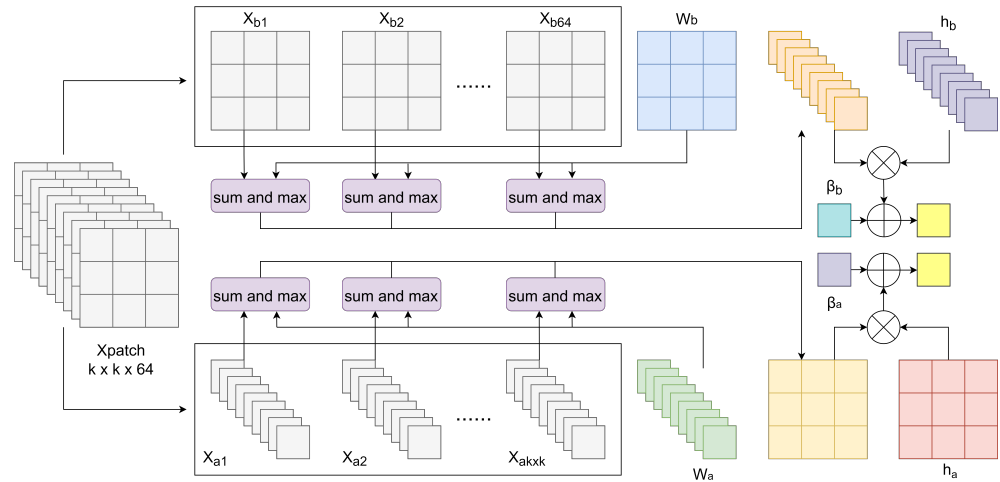


Figure 7. Overall architecture of dilation block.

Finally, we concatenate the two obtained feature values to form the convolution result of that X_{patch} under the specified W_a and W_b , referred to as $f_{\text{dil}} \in \mathbb{R}^2$:

$$f_{\text{dil}}(X_{\text{patch}}, W) = \text{concat}(g_{\text{dil}}(X_a, W_a), g_{\text{dil}}(X_b, W_b)) \quad (13)$$

In actual experiments, 16 groups of W were used in the erosion block. Therefore, after computing all the W with X_{patch} , the final HSI feature size obtained through the dilation module is 32×64 . Similarly, for any X_{patch} and W , the following formula describes the erosion operation $f_{\text{ero}}(X_{\text{patch}}, W)$ in the morphological convolution:

$$f_{\text{ero}}(X_{\text{patch}}, W) = \text{concat}(g_{\text{ero}}(X_a, W_a), g_{\text{ero}}(X_b, W_b)) \quad (14)$$

$$g_{\text{ero}}(X_a, W_a) = \text{concat}(h_{\text{ero}}(X_{a1}, W_a), h_{\text{ero}}(X_{a2}, W_a), \dots, h_{\text{ero}}(X_{ak \times k}, W_a)) \times h_a + \beta_a \quad (15)$$

$$g_{\text{ero}}(X_b, W_b) = \text{concat}(h_{\text{ero}}(X_{b1}, W_b), h_{\text{ero}}(X_{b2}, W_b), \dots, h_{\text{ero}}(X_{b64}, W_b)) \times h_b + \beta_b \quad (16)$$

$$h_{\text{ero}}(X_{ai}, W_a) = \min_{j \in \{1, 2, \dots, 64\}} (X_{ai}(j) - W_a(j)) \quad (17)$$

$$h_{\text{ero}}(X_{bj}, W_b) = \min_{i \in \{1, 2, \dots, k \times k\}} (X_{bj}(i) - W_b(i)) \quad (18)$$

Overall, we process the 64 X_{patch} using 32 sets of SEs. Specifically, 16 sets are responsible for the dilation operation, while the other 16 sets handle the erosion operation. This results in two 32×64 feature matrices. After spatial-spectral separation, the required parameter count for the SEs is reduced from $2 \times 32 \times k \times k \times 64$ to $2 \times 2 \times 16 \times (k \times k + 64 + 1)$. Additionally, MC operates similarly to traditional convolutional layers, allowing it to directly replace convolutional layers in models. This attribute endows MC with significant versatility and adaptability.

3. Results and Discussion

3.1. Dataset Description

To validate the effectiveness of the CESA-MCFormer feature extractor, we tested its performance in two types of classification tasks. Specifically, for the semantic segmentation

task, we used the Indian Pines dataset (IP), Pavia University dataset (UP), and Chikusei dataset; while, for the few-shot learning (FSL) task, the datasets included the IP, UP, Chikusei dataset, Botswana dataset, KSC dataset, and Salinas Valley dataset. The detailed information about these datasets is presented in Table 1.

Table 1. Dataset Information.

Dataset	Image Size	Number of Classes	Number of Bands
IP	145 × 145	16	220
UP	610 × 340	9	103
Chikusei	2517 × 2335	19	128
Botswana	1476 × 256	14	145
KSC	512 × 614	13	176
Salinas Valley	512 × 217	16	224

3.1.1. Semantic Segmentation Task

In the semantic segmentation task, we randomly selected 1% of the pixels from the UP and Chikusei datasets as the training set, with the remaining pixels as the test set. Given that the Oats class in the IP dataset has only 20 pixels, we randomly extracted 5% of the pixels from the IP dataset as the training set and the rest as the test set. In constructing the training set, we did not employ any augmentation methods nor use any dimensionality reduction techniques on the datasets. The specific number of samples for each category in each dataset is shown in Tables 2–4.

Table 2. Detailed information on the training and testing data samples for each class in the IP dataset.

Class No.	Class Name	Training	Testing
1	Alfalfa	3	43
2	Corn-notill	72	1356
3	Corn-mintill	42	788
4	Corn	12	225
5	Grass-pasture	25	458
6	Grass-trees	37	693
7	Grass-pasture-mowed	2	26
8	Hay-windrowed	24	454
9	Oats	1	19
10	Soybean-notill	49	923
11	Soybean-mintill	123	2332
12	Soybean-clean	30	563
13	Wheat	11	194
14	Woods	64	1201
15	Buildings-Grass-Trees-Drives	20	366
16	Stone-Steel-Towers	5	88
Total		520	9729

Table 3. Detailed information on the training and testing data samples for each class in the UP dataset.

Class No.	Class Name	Training	Testing
1	Asphalt	67	6564
2	Meadows	187	18,462
3	Gravel	21	2078
4	Trees	31	3033
5	Painted metal sheets	14	1331
6	Bare Soil	51	4978
7	Bitumen	14	1316
8	Self-Blocking Bricks	37	3645
9	Shadows	10	937
Total		432	42,344

Table 4. Detailed information on the training and testing data samples for each class in the Chikusei dataset.

Class No.	Class Name	Training	Testing
1	Water	29	2816
2	Bare soil (school)	29	2830
3	Bare soil (park)	3	283
4	Bare soil (farmland)	49	4830
5	Natural plants	43	4254
6	Weeds in farmland	12	1096
7	Forest	206	20,310
8	Grass	66	6449
9	Rice field (grown)	134	13,235
10	Rice field (first stage)	13	1255
11	Row crops	60	5901
12	Plastic house	22	2171
13	Manmade (non-dark)	13	1207
14	Manmade (dark)	77	7587
15	Manmade (blue)	5	426
16	Manmade (red)	3	219
17	Manmade grass	11	1029
18	Asphalt	9	792
19	Paved ground	2	143
Total		786	76,806

3.1.2. Few-Shot Learning Task

Given the extensive category requirements for few-shot learning (FSL) training, our study utilized six datasets, with the Chikusei, Botswana, KSC, and Salinas Valley used for model pretraining, and the IP and UP datasets for testing. To ensure consistency in input data sizes across all datasets in the FSL experiments, we standardized the dimensions of each dataset to 100 using BS-Nets [53].

For the datasets involved in pretraining, we selected classes with over 250 samples, randomly allocating 50 samples to the support set and 200 to the query set for each class. Specifically, Chikusei contributed 17 classes, Botswana 8, KSC 9, and Salinas Valley 16, totaling 50 distinct training classes. After pretraining, we randomly selected 10 samples from each class in the IP and UP datasets for model fine-tuning and testing.

In our study, during the pretraining on the IP dataset, each task randomly selected 16 out of 50 available classes, following a 16-way, 10-shot method. For the UP dataset, each training iteration randomly chose nine classes, also using a 10-shot approach. For each class, the support set consisted of 10 randomly selected samples out of 50, while the query set used all 200 samples. Moreover, no form of data augmentation was used to expand the datasets, neither in the pretraining nor in the fine-tuning stages.

3.2. Training Details and Evaluation Indicators

3.2.1. Configuration

All experiments were designed and conducted using PyTorch on a Ubuntu 18.04 x64 machine with 13th Gen Intel(R) Core(TM) i5-13600KF CPU, 32GB RAM, and an NVIDIA Geforce RTX 4080 16GB GPU.

3.2.2. Training Details

In our semantic segmentation task, we directly classify using the cls token connected to a fully connected layer, as depicted in the Classifier Head block in Figure 1. The Adam optimizer is used with a learning rate of 0.001, and CrossEntropy Loss functions as the loss criterion. For models such as HybridSN [26], Vit [29], SF [33], and SSFTT [34], we maintain a batch size of 64. For morphFormer [52] and our developed CESA-MCFormer, the batch

size is set at 32. Table 5 displays the floating point operations (FLOPS) and the number of parameters for various models.

For the FSL task, we incorporate two sets of trainable weights, summing HSI features weighted along spatial and spectral directions to create two feature vectors of length 64 each. These are concatenated with the cls token, forming a final vector of length 192. We average features of 10 samples from each class in the support set to represent class prototypes, with classification based on distances between query set features and these prototypes. For the convolutional network (HybridSN), we alter the final linear layer's output to a 192-length feature vector for uniformity in feature output length across models. The SGD optimizer is employed, setting the learning rate at 0.00001 and weight decay at 0.0005.

3.2.3. Evaluation Indicators

We used four quantitative evaluation metrics including overall accuracy (OA), average accuracy (AA), kappa coefficient (κ), and class-specific accuracy to quantitatively analyze the effectiveness of CESA. The higher the values of these metrics, the better the classification performance.

3.3. Semantic Segmentation Task Experimental Results

3.3.1. Classification Results

To verify the advanced nature of our CESA-MCFormer model, we conducted comparative experiments with several recently proposed models. These include HybridSN [26], Vit [29], SF [33], and SSFTT [34], which originally required PCA for dimensionality reduction in their respective papers. Conversely, morphFormer [52] does not require such reduction. In our experiments, we used a patch size of 11×11 as the input for all models, with K set to 0.8 in Hard CESA.

Tables 6–8 display the classification results of various models without using PCA dimensionality reduction, while Figures 8 and 9 present the visualization results on the IP and UP datasets. The experimental data demonstrate that CESA-MCFormer exhibits superior performance across all datasets, highlighting its exceptional feature extraction capability in the presence of abundant redundant information. Moreover, as observed from Figures 8 and 9, the combination of EMB Block and MC to eliminate redundancy has notably enhanced the model's accuracy in classifying complex pixels, especially in edge areas and categories with limited samples. This further underscores the outstanding performance of CESA-MCFormer.

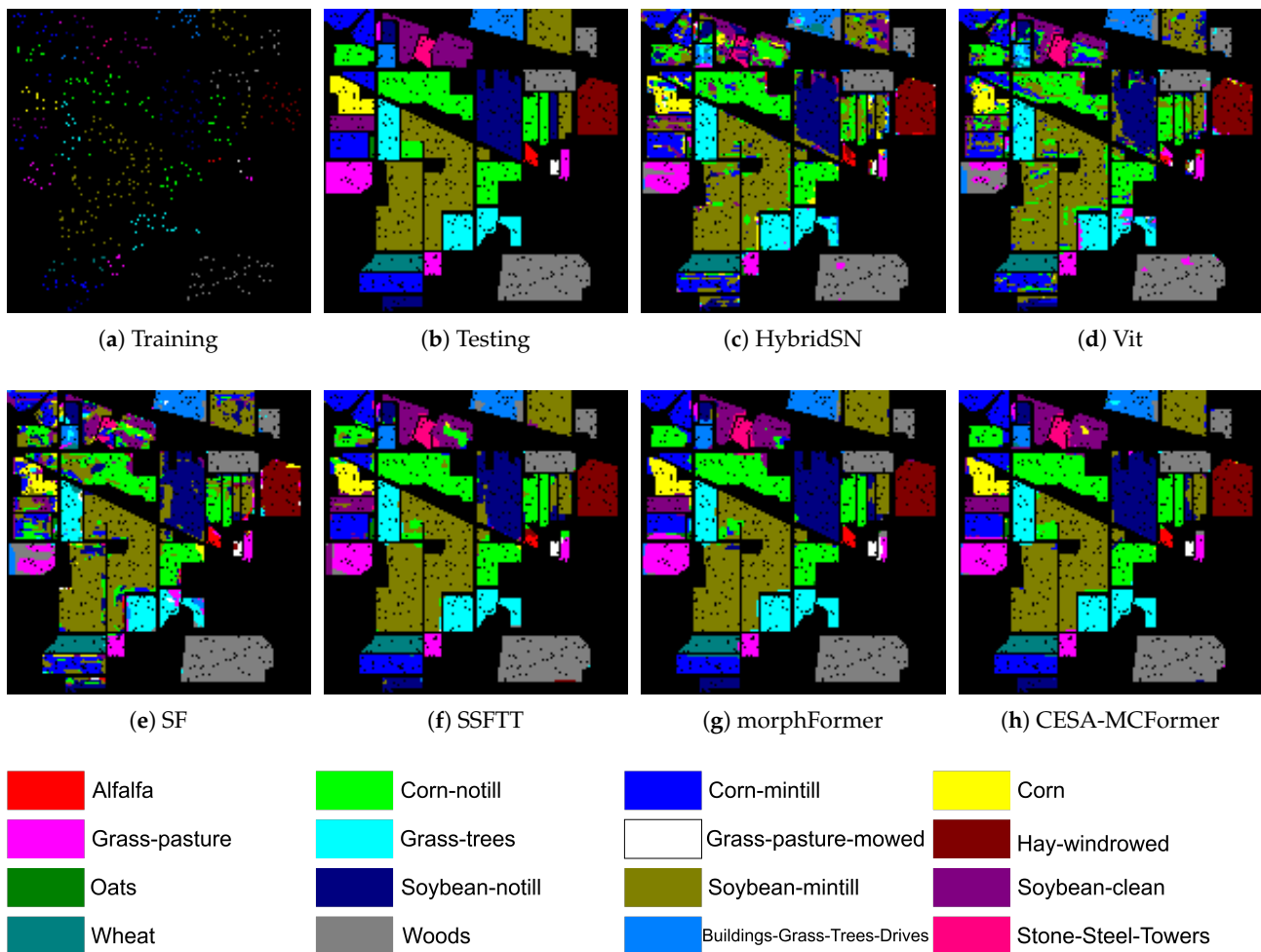
Tables 9 and 10 present the classification results of HybridSN, SpectralFormer, and SSFTT on the IP and UP datasets after applying PCA dimensionality reduction. Specifically, the dimensionality of the IP dataset was reduced to 30, while that of the UP dataset was reduced to 15. The results indicate that the performance of HybridSN, SpectralFormer, and SSFTT improved significantly after PCA reduction. However, their accuracy still falls short of our CESA-MCFormer model.

Table 5. The floating point operations (FLOPS) and the number of parameters for various models. “CESA-MCFormer*” refers to the CESA-MCFormer model without the inclusion of the CESA.

Class No	HybridSN	Vit	SF	SSFTT	morphFormer	CESA-MCFormer	CESA-MCFormer*
FLOPs (M)	44.80	75.34	24.91	29.88	41.78	32.24	27.67
Params (M)	1.12	0.61	0.11	0.51	0.25	0.36	0.25

Table 6. Classification accuracy of various models on the IP dataset (without PCA).

Class No	HybridSN	Vit	SF	SSFTT	morphFormer	CESA-MCFormer
1	62.79	39.53	83.72	58.14	97.67	93.02
2	73.45	74.78	64.16	91.81	92.92	96.61
3	70.05	69.29	50.25	94.54	88.07	95.81
4	67.56	57.33	53.78	77.78	91.11	90.22
5	76.86	46.51	57.42	79.91	89.74	95.85
6	94.95	86.29	81.53	98.70	99.42	100.00
7	46.15	42.31	76.92	100.00	100.00	100.00
8	96.70	95.15	92.51	100.00	99.34	99.34
9	63.16	26.32	73.68	89.47	31.58	84.21
10	76.60	73.67	76.71	87.22	95.56	95.12
11	81.86	84.95	83.40	97.51	97.13	97.64
12	44.76	51.33	45.83	82.77	88.28	95.38
13	100.00	100.00	98.97	100.00	97.94	100.00
14	98.17	96.92	99.25	98.83	99.75	99.17
15	59.84	73.50	84.43	85.79	91.53	87.98
16	55.68	95.45	51.14	97.73	100.00	100.00
OA (%)	79.24	78.38	75.59	93.15	94.96	96.82
AA (%)	73.04	69.58	73.36	90.01	91.25	95.65
κ (%)	76.28	75.17	71.96	92.17	94.26	96.38

**Figure 8.** Visualization results on the IP dataset.

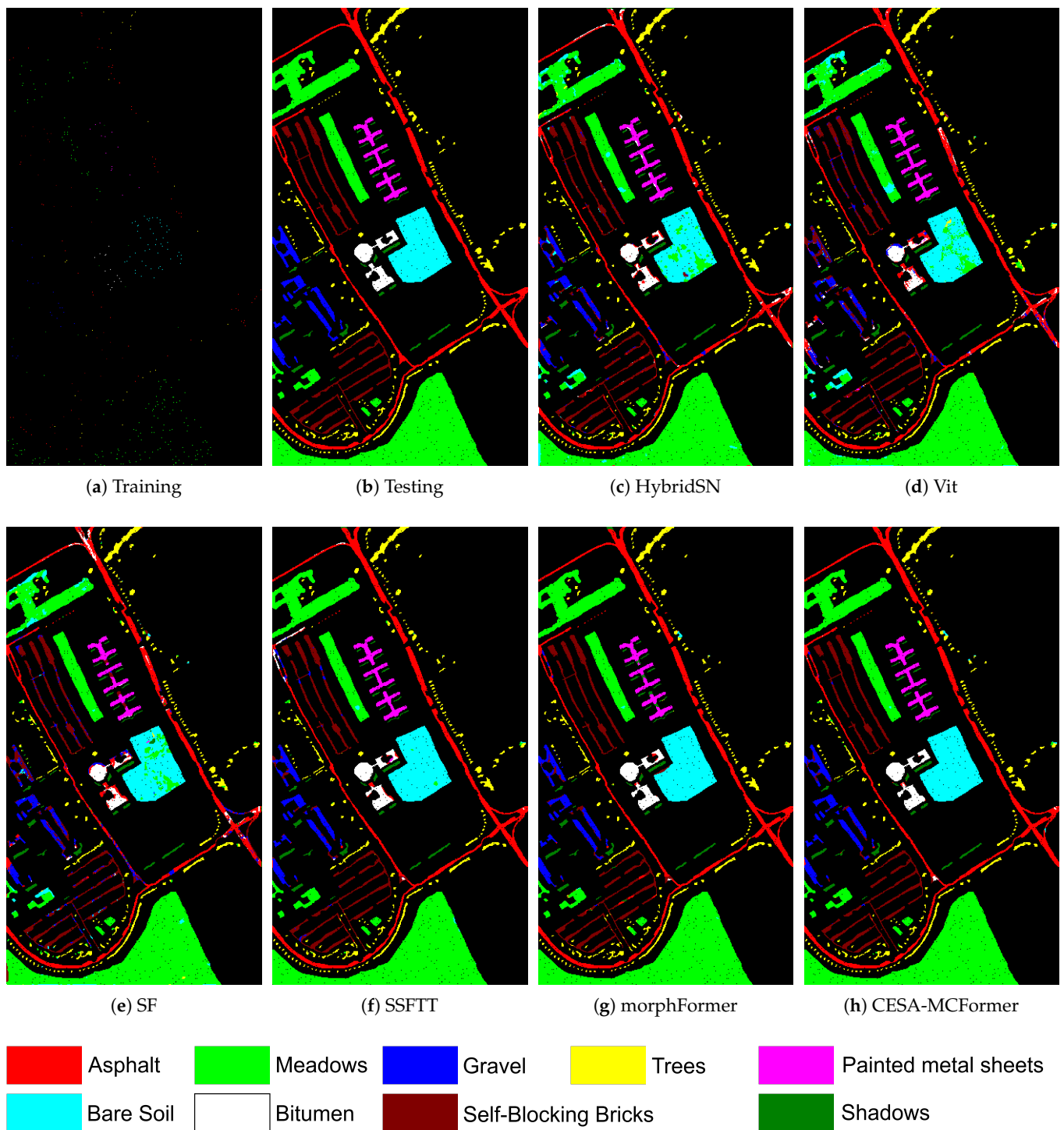


Figure 9. Visualization results on the UP dataset.

Table 7. Classification accuracy of various models on the UP dataset (without PCA).

Class No	HybridSN	Vit	SF	SSFTT	morphFormer	CESA-MCFormer
1	89.98	91.97	83.32	94.70	98.00	98.35
2	95.54	93.90	95.38	99.70	99.69	99.92
3	75.07	47.50	59.58	87.73	87.05	89.56
4	94.16	92.81	89.48	96.24	96.74	95.12
5	96.39	99.85	89.41	99.92	93.09	100.00
6	84.39	84.23	85.01	99.38	99.08	100.00
7	82.14	62.54	63.91	90.20	97.04	98.33
8	92.76	91.39	87.46	95.80	95.69	99.18
9	98.40	99.79	99.89	98.51	97.87	97.55
OA (%)	91.70	89.23	88.36	97.40	97.85	98.67
AA (%)	89.87	84.89	83.72	95.80	96.27	97.56
κ (%)	89.00	85.75	84.59	96.56	97.15	98.24

Table 8. Classification accuracy of various models on the Chikusei dataset (without PCA).

Class No	HybridSN	Vit	SF	SSFTT	morphFormer	CESA-MCFormer
1	97.41	94.74	91.58	99.61	99.96	99.15
2	95.23	94.03	99.01	95.30	99.82	99.72
3	0.00	18.37	23.67	46.64	22.61	85.16
4	99.79	97.67	98.44	97.48	98.69	99.88
5	99.79	96.94	97.32	99.98	100.00	99.98
6	97.90	97.26	93.89	91.88	98.54	91.79
7	100.00	100.00	100.00	100.00	100.00	100.00
8	99.74	98.71	99.18	100.00	100.00	99.94
9	99.81	99.89	99.99	100.00	99.77	100.00
10	97.37	99.20	95.30	100.00	100.00	100.00
11	100.00	99.75	100.00	99.83	99.76	100.00
12	96.96	93.74	98.11	96.32	97.14	96.78
13	95.53	95.53	94.86	95.53	95.53	95.53
14	99.79	98.87	99.00	99.97	99.91	100.00
15	99.06	93.19	95.54	100.00	99.30	99.53
16	100.00	98.17	99.54	93.61	100.00	100.00
17	99.03	86.10	97.57	100.00	100.00	100.00
18	79.80	91.41	88.76	90.03	97.85	99.37
19	16.78	31.47	42.66	88.11	88.11	95.10
OA (%)	98.65	97.97	98.38	99.01	99.34	99.59
AA (%)	88.10	88.69	90.23	94.44	94.58	98.00
κ (%)	98.44	97.65	98.13	98.85	99.24	99.53

Table 9. Classification accuracy of various models on the IP dataset (with PCA). “CESA-MCFormer*” refers to the CESA-MCFormer model without PCA.

Class No	HybridSN	SF	SSFTT	CESA-MCFormer	CESA-MCFormer*
1	72.09	51.16	100.00	76.74	93.02
2	92.99	78.17	94.91	96.02	96.61
3	98.98	73.86	98.35	95.18	95.81
4	94.67	61.78	100.00	88.89	90.22
5	90.83	84.50	98.69	90.17	95.85
6	99.86	93.94	99.13	99.42	100.00
7	57.69	11.54	73.08	100.00	100.00
8	94.49	100.00	99.56	100.00	99.34
9	68.42	10.53	52.63	100.00	84.21

Table 9. Cont.

Class No	HybridSN	SF	SSFTT	CESA-MCFormer	CESA-MCFormer *
10	91.12	85.59	97.29	94.58	95.12
11	96.57	87.78	97.60	98.37	97.64
12	86.50	69.45	84.37	88.28	95.38
13	97.94	98.45	100.00	100.00	100.00
14	98.92	90.67	99.67	99.92	99.17
15	96.45	62.02	93.44	90.16	87.98
16	48.86	81.82	95.45	100.00	100.00
OA (%)	94.60	83.33	96.78	96.23	96.82
AA (%)	86.65	71.33	92.76	94.86	95.65
κ (%)	93.84	80.97	96.33	95.70	96.38

Table 10. Classification accuracy of various models on the UP dataset (with PCA). “CESA-MCFormer **” refers to the CESA-MCFormer model without PCA.

Class No	HybridSN	SF	SSFTT	CESA-MCFormer	CESA-MCFormer *
1	98.86	88.82	98.28	98.49	98.35
2	99.96	96.59	99.97	99.87	99.92
3	85.51	71.80	90.38	91.77	89.56
4	92.91	91.16	97.36	95.19	95.12
5	100.00	95.87	100.00	100.00	100.00
6	99.76	79.23	98.49	99.16	100.00
7	97.04	75.08	98.94	98.02	98.33
8	92.76	84.94	91.44	98.30	99.18
9	93.17	92.32	93.06	95.73	97.55
OA (%)	97.69	89.95	97.96	98.56	98.67
AA (%)	95.55	86.20	96.44	97.39	97.56
κ (%)	96.93	86.58	97.29	98.09	98.24

3.3.2. Ablation Experiment

To better understand the roles of CESA and MC within the model, we conducted several ablation studies on the IP dataset. In these experiments, we continued to adopt the hyperparameter settings from the previous section. Given the similar overall architecture of CESA-MCFormer and morphFormer, and the outstanding performance of morphFormer when compared to other models, we chose morphFormer as our baseline and built upon it by adding modules.

Table 11 demonstrates the influence of MC and CESA on the final classification performance of the model. It is evident from the table that both modules have significantly enhanced the OA. Furthermore, when both modules are used in conjunction, there is an additional improvement in accuracy. However, in contrast to the pronounced improvement in AA brought about by CESA, the contribution of MC is relatively limited. This can be primarily attributed to the lower classification accuracy for classes with fewer samples. In scenarios with limited sample sizes, prior information becomes particularly crucial. Without the spatial prior information provided by CESA, relying solely on MC to process hyperspectral features that encapsulate comprehensive spatial information proves to be more challenging.

To further validate the superiority of CESA, we replaced it with the traditional Spatial Attention block (SA) and CAM in the CESA-MCFormer model and conducted comparative experiments. The results, presented in Table 12, demonstrate that CESA achieved the highest classification accuracy. This is attributed to CESA’s combination of Hard and Soft components, which not only incorporate prior information but also ensure the learnability of the entire module. Thus, CESA effectively amalgamates the advantages of SA and CAM, leading to improved performance.

Table 11. Ablation study results for CESA and MC. “CESA” stands for replacing the Emb Block, and “MC” stands for replacing the Transformer Encoder.

CESA	MC+	OA (%)	AA (%)	κ (%)
		94.96	91.25	94.26
✓		95.86	93.06	95.27
	✓	95.80	91.50	95.21
✓	✓	96.82	95.65	96.38

Table 12. Comparative experiments of the traditional Spatial Attention block (SA), CAM, and CESA.

	OA (%)	AA (%)	κ (%)
SA	95.79	91.76	95.19
CAM	95.94	93.17	95.37
CESA	96.82	95.65	96.38

3.4. Few-Shot Learning Task Experimental Results

To assess the generality and effectiveness of CESA-MCFormer with extremely limited samples, we conducted FSL experiments on the IP and UP datasets, using only 10 samples per class. The experimental results, as shown in Tables 13 and 14, indicate that CESA-MCFormer achieved optimal performance on both datasets.

Table 13. Classification accuracy of various models on the IP dataset.

	HybridSN	SSFTT	morphForm	CESA-MCFormer
OA (%)	68.90	69.56	71.26	73.29
AA (%)	81.10	81.87	82.97	84.61
κ (%)	65.17	65.85	67.82	70.05

Table 14. Classification accuracy of various models on the UP dataset.

	HybridSN	SSFTT	morphForm	CESA-MCFormer
OA (%)	70.81	74.43	74.03	77.54
AA (%)	77.90	79.39	78.90	83.55
κ (%)	63.44	67.17	66.71	71.46

4. Conclusions

This paper presents the CESA-MCFormer feature extractor, which boosts the model’s feature extraction capabilities with a “selection–extraction” strategy, enabling effective image feature extraction without reliance on PCA. CESA enables the model to incorporate spatial prior knowledge guided by an attention mechanism while maintaining the learnability of the module. The MC module introduces learnable pooling operations that effectively filter key information during deep feature extraction. Additionally, CESA-MCFormer adapts to various classification tasks by modifying the classifier head, and both CESA and MC modules can be flexibly integrated into other models to improve their feature extraction performance. Comparisons with other models in semantic segmentation and FSL tasks confirm the versatility and effectiveness of CESA-MCFormer, and ablation studies of CESA and MC attest to the efficacy of these two components.

The CESA-MCFormer has demonstrated exceptional versatility in surface object classification tasks. In future research, we intend to further explore the model’s application to subsurface exploration tasks (such as soil composition analysis) and are committed to further optimizing its performance.

Author Contributions: Methodology, S.L.; Validation, H.Z.; Writing—original draft, S.L.; Supervision, C.Y. and H.Z.; Funding acquisition, C.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Tongji University grant number 09-H30G02-9001-20/22.

Institutional Review Board Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Roy, S.K.; Kar, P.; Hong, D.; Wu, X.; Plaza, A.; Chanussot, J. Revisiting deep hyperspectral feature extraction networks via gradient centralized convolution. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5516619. [[CrossRef](#)]
2. Roy, S.K.; Mondal, R.; Paoletti, M.E.; Haut, J.M.; Plaza, A. Morphological convolutional neural networks for hyperspectral image classification. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2021**, *14*, 8689–8702. [[CrossRef](#)]
3. Ahmad, M.; Shabbir, S.; Roy, S.K.; Hong, D.; Wu, X.; Yao, J.; Khan, A.M.; Mazzara, M.; Distefano, S.; Chanussot, J. Hyperspectral image classification—traditional to deep models: A survey for future prospects. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2022**, *15*, 968–999. [[CrossRef](#)]
4. Signoroni, A.; Savardi, M.; Baronio, A.; Benini, S. Deep Learning Meets Hyperspectral Image Analysis: A Multidisciplinary Review. *J. Imaging* **2019**, *5*, 52. [[CrossRef](#)]
5. Li, K.; Huang, J.; Liu, X. Mapping wheat plant height using a crop surface model from unmanned aerial vehicle imagery and 3D feature points. *Comput. Electron. Agric.* **2019**, *164*, 104881. [[CrossRef](#)]
6. Atherton, D.; Choudhary, R.; Watson, D. Hyperspectral Remote Sensing for Advanced Detection of Early Blight (*Alternaria solani*) Disease in Potato (*Solanum tuberosum*) Plants Prior to Visual Disease Symptoms. In Proceedings of the 2017 ASABE Annual International Meeting, American Society of Agricultural and Biological Engineers, Spokane, WA, USA, 16–19 July 2017; p. 1.
7. Shafri, H.Z.M.; Taherzadeh, E.; Mansor, S.; Ashurov, R. Hyperspectral Remote Sensing of Urban Areas: An Overview of Techniques and Applications. *Res. J. Appl. Sci. Eng. Technol.* **2012**, *4*, 1557–1565.
8. Navin, M.S.; Agilandeswari, L. Multispectral and Hyperspectral Images Based Land Use/Land Cover Change Prediction Analysis: An Extensive Review. *Multimedia Tools Appl.* **2020**, *79*, 29751–29774. [[CrossRef](#)]
9. Andrew, M.E.; Ustin, S.L. The role of environmental context in mapping invasive plants with hyperspectral image data. *Remote Sens. Environ.* **2008**, *112*, 4301–4317. [[CrossRef](#)]
10. Stuart, M.B.; McGonigle, A.J.S.; Willmott, J.R. Hyperspectral imaging in environmental monitoring: A review of recent developments and technological advances in compact field deployable systems. *Sensors* **2019**, *19*, 3071. [[CrossRef](#)]
11. Rajabi, R.; Zehtabian, A.; Singh, K.D. Hyperspectral imaging in environmental monitoring and analysis. *Front. Environ. Sci.* **2024**. [[CrossRef](#)]
12. Nasrabadi, N.M. Hyperspectral target detection: An overview of current and future challenges. *IEEE Signal Process. Mag.* **2013**, *31*, 34–44. [[CrossRef](#)]
13. Chang, C.-I. Hyperspectral anomaly detection: A dual theory of hyperspectral target detection. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–20. [[CrossRef](#)]
14. Lee, G.; Lee, J.; Baek, J.; Kim, H.; Cho, D. Channel sampler in hyperspectral images for vehicle detection. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 2022. [[CrossRef](#)]
15. Shi, P.; Jiang, Q.; Li, Z. Hyperspectral Characteristic Band Selection and Estimation Content of Soil Petroleum Hydrocarbon Based on GARF-PLSR. *J. Imaging* **2023**, *9*, 87. [[CrossRef](#)]
16. Fauvel, M.; Tarabalka, Y.; Benediktsson, J.A.; Chanussot, J.A.; Tilton, J.C. Advances in spectral-spatial classification of hyperspectral images. *Proc. IEEE* **2013**, *101*, 652–675. [[CrossRef](#)]
17. SahIn, Y.E.; Arisoy, S.; Kayabol, K. Anomaly detection with Bayesian Gauss background model in hyperspectral images. In Proceedings of the 26th Signal Processing and Communications Applications Conference, SIU, Izmir, Turkey, 2–5 May 2018; pp. 1–4.
18. Haut, J.; Paoletti, M.; Paz-Gallardo, A.; Plaza, J.; Plaza, A. Cloud implementation of logistic regression for hyperspectral image classification. In Proceedings of the 17th International Conference on Computational and Mathematical Methods in Science and Engineering (CMMSE), Cadiz, Spain, 4–8 July 2017; Costa Ballena: Cádiz, Spain, 2017; Volume 3, pp. 1063–1073.
19. Li, J.; Bioucas-Dias, J.; Plaza, A. Spectral-spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 809–823. [[CrossRef](#)]
20. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [[CrossRef](#)]
21. Ye, Q.; Huang, P.; Zhang, Z.; Zheng, Y.; Fu, L.; Yang, W. Multiview learning with robust double-sided twin SVM. *IEEE Trans. Cybern.* **2021**, *52*, 12745–12758. [[CrossRef](#)]

22. Ye, Q.; Zhao, H.; Li, Z.; Yang, X.; Gao, S.; Yin, T.; Ye, N. L1-norm distance minimization-based fast robust twin support vector k-plane clustering. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 4494–4503. [[CrossRef](#)]
23. Chen, Y.-N.; Thaipisutikul, T.; Han, C.-C.; Liu, T.-J.; Fan, K.-C. Feature line embedding based on support vector machine for hyperspectral image classification. *Remote Sens.* **2021**, *13*, 130. [[CrossRef](#)]
24. Lee, H.; Kwon, H. Going deeper with contextual CNN for hyperspectral image classification. *IEEE Trans. Image Process.* **2017**, *26*, 4843–4855. [[CrossRef](#)]
25. Luo, Y.; Zou, J.; Yao, C.; Zhao, X.; Li, T.; Bai, G. HSI-CNN: A novel convolution neural network for hyperspectral image. In Proceedings of the International Conference on Audio, Language and Image Processing (ICALIP), Shanghai China, 16–17 July 2018.
26. Roy, S.K.; Krishna, G.; Dubey, S.R.; Chaudhuri, B.B. HybridSN: Exploring 3-D–2D CNN feature hierarchy for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 277–281. [[CrossRef](#)]
27. Wang, J.; Song, X.; Sun, L.; Huang, W.; Wang, J. A novel cubic convolutional neural network for hyperspectral image classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 4133–4148. [[CrossRef](#)]
28. Moraga, J.; Duzgun, H.S. JigsawHSI: A network for hyperspectral image classification. *arXiv* **2022**, arXiv:2206.02327.
29. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
30. He, X.; Chen, Y.; Lin, Z. Spatial-Spectral Transformer for Hyperspectral Image Classification. *Remote Sens.* **2021**, *13*, 498. [[CrossRef](#)]
31. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
32. Qing, Y.; Liu, W.; Feng, L.; Gao, W. Improved Transformer Net for Hyperspectral Image Classification. *Remote Sens.* **2021**, *13*, 2216. [[CrossRef](#)]
33. Hong, D.; Han, Z.; Yao, J.; Gao, L.; Zhang, B.; Plaza, A.; Chanussot, J. SpectralFormer: Rethinking Hyperspectral Image Classification with Transformers. *arXiv* **2021**, arXiv:2107.02988.
34. He, X.; Chen, Y.; Lin, Z. Spectral-Spatial Feature Tokenization Transformer for Hyperspectral Image Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 6824–6836.
35. Zhang, X.; Su, Y.; Gao, L.; Bruzzone, L.; Gu, X.; Tian, Q. A Lightweight Transformer Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5517617. [[CrossRef](#)]
36. Zhang, S.; Zhang, J.; Wang, X.; Wang, J.; Zhu, Z. ELS2T: Efficient Lightweight Spectral-Spatial Transformer for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5518416. [[CrossRef](#)]
37. Geiger, B.C.; Kubin, G. Relative information loss in the PCA. In Proceedings of the 2012 IEEE Information Theory Workshop, San Diego, CA, USA, 5 February 2012; pp. 562–566.
38. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
39. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
40. Zhong, Y.; Liu, L.; Yang, Y.; Loy, C.C. Dual Attention Network for Scene Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2018; pp. 3146–3154.
41. Zhang, Y.; Wang, X.; Liu, Y.; Zhang, X. Discriminative spectral-spatial attention-aware residual network for hyperspectral image classification. *IEEE Access* **2020**, *8*, 226169–226184.
42. Li, J.; Li, G.; Sun, Q.; Li, L. Residual Spectral–Spatial Attention Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 449–462.
43. Liu, L.; Zhang, P.; Zhang, W.; Lu, J.; Wang, J. Spectral-Spatial Attention Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 3232–3245.
44. Gevaert, A.S.; De Backer, S.S.; Schiavon, A.C.; Philips, W. SpectralSpatial Fused Attention Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*. [[CrossRef](#)]
45. Wang, C.; Ma, X.; Chen, Y.; Ren, Y.; Han, Z. Center Attention Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*. [[CrossRef](#)]
46. Wang, Y.; Zhang, L.; Zhang, L. Spatial Attention Guided Residual Attention Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 5966–5980. [[CrossRef](#)]
47. Serra, J.; Soille, P. *Morphological Image Analysis: Principles and Applications*; Springer: Berlin, Germany, 1994.
48. Pesaresi, M.; Benediktsson, J.A. A new approach for the morphological segmentation of high-resolution satellite imagery. *IEEE Trans. Geosci. Remote Sens.* **2001**, *39*, 309–320. [[CrossRef](#)]
49. Pedernana, M.; Marpu, P.R.; Mura, M.D.; Benediktsson, J.A.; Bruzzone, L. Classification of remote sensing optical and LiDAR data using extended attribute profiles. *IEEE J. Sel. Top. Signal Process.* **2012**, *6*, 856–865. [[CrossRef](#)]
50. Roy, S.K.; Chanda, B.; Chaudhuri, B.B.; Ghosh, D.K.; Dubey, S.R. Local morphological pattern: A scale space shape descriptor for texture classification. *Digit. Signal Process.* **2018**, *82*, 152–165. [[CrossRef](#)]
51. Hong, D.; Wu, X.; Ghamisi, P.; Chanussot, J.; Yokoya, N.; Zhu, X.X. Invariant attribute profiles: A spatial-frequency joint feature extractor for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 3791–3808. [[CrossRef](#)]

-
52. Roy, S.K.; Deria, A.; Shah, C.; Haut, J.M.; Du, Q.; Plaza, A. Spectral–Spatial Morphological Attention Transformer for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5503615. [[CrossRef](#)]
 53. Cai, Y.; Liu, X.; Cai, Z. BS-Nets: An End-to-End Framework for Band Selection of Hyperspectral Image. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 1969–1984. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.