remote sensing

Article

MSSFF: Advancing Hyperspectral Classification through Higher-Accuracy Multistage Spectral–Spatial Feature Fusion

Yuhan Chen [1,2], Qingyun Yan [1,*] and Weimin Huang [3]

[1] School of Remote Sensing and Geomatics Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China; yhchen@hrbeu.edu.cn
[2] Qingdao Innovation and Development Center (Base), Harbin Engineering University, Qingdao 266000, China
[3] Faculty of Engineering and Applied Science, Memorial University, St. John's, NL A1B 3X5, Canada; weimin@mun.ca
* Correspondence: 003257@nuist.edu.cn

Abstract: This paper presents the MSSFF (multistage spectral–spatial feature fusion) framework, which introduces a novel approach for semantic segmentation from hyperspectral imagery (HSI). The framework aims to simplify the modeling of spectral relationships in HSI sequences and unify the architecture for semantic segmentation of HSIs. It incorporates a spectral–spatial feature fusion module and a multi-attention mechanism to efficiently extract hyperspectral features. The MSSFF framework reevaluates the potential impact of spectral and spatial features on segmentation models and leverages the spectral–spatial fusion module (SSFM) in the encoder component to effectively extract and enhance these features. Additionally, an efficient Transformer (ET) is introduced in the skip connection part of deep features to capture long-term dependent features and extract global spectral–spatial information from the entire feature map. This highlights the significant potential of Transformers in modeling spectral–spatial feature maps within the context of hyperspectral remote sensing. Moreover, a spatial attention mechanism is adopted in the shallow skip connection part to extract local features. The framework demonstrates promising capabilities in hyperspectral remote sensing applications. The conducted experiments provide valuable insights for optimizing the model depth and the order of feature fusion, thereby contributing to the advancement of hyperspectral semantic segmentation research.

Keywords: convolutional neural networks (CNNs); hyperspectral image (HSI); image-based classification; vision transformer

check for updates

Citation: Chen, Y.; Yan, Q.; Huang, W. MSSFF: Advancing Hyperspectral Classification through Higher-Accuracy Multistage Spectral–Spatial Feature Fusion. *Remote Sens.* **2023**, *15*, 5717. https://doi.org/10.3390/rs15245717

Academic Editor: Salah Bourennane

Received: 16 October 2023
Revised: 8 December 2023
Accepted: 12 December 2023
Published: 13 December 2023

1. Introduction

Hyperspectral imagery (HSI) contains a wealth of spectral information and comprises multiple, and in some cases, hundreds of bands. This spectral information can be leveraged to classify important ground objects based on the characteristics exhibited across different bands. Feature extraction plays a pivotal role in HSI classification and has garnered growing interest among researchers. Hyperspectral remote sensing has made significant contributions in various domains. Such as military applications [1], medical research [2], water quality monitoring [3], and agricultural research [4].

However, the presence of numerous frequency bands in the hyperspectral data results in strong correlations between adjacent bands [5]. This correlation leads to a significant amount of redundant information for classification tasks [6]. Consequently, early approaches in hyperspectral classification primarily focused on data reduction techniques and feature engineering [7,8].

In recent years, with the advancements in deep learning, this technology has been increasingly adopted in various domains [9–11], including hyperspectral remote sensing, and has achieved remarkable success [6]. Deep learning models have the capability to extract meaningful knowledge from vast amounts of redundant data [12]. The multi-layer structure of these models enables the acquisition of higher-level semantic information from the samples [13].

Various deep learning models have been developed for hyperspectral data analysis, with convolutional neural network (CNN)-based models standing out due to their remarkable performance. Yu et al. [14] introduced a CNN architecture that takes a single pixel as input, enabling the network to directly learn the relationships between different spectral bands. Chen et al. [15] propose a 3D-CNN model with sparse constraints that can directly extract spectral–spatial features from HSI. Ghaderizadeh et al. [16] presented a hybrid 3D-2D CNN architecture. This hybrid CNN approach offers advantages over standalone 3D-CNN by reducing the model's complexity and mitigating the impacts of noise and limited training samples.

In addition to CNNs, several other network architectures have demonstrated strong performance in HSI classification. Recurrent neural networks (RNNs) are capable of capturing both long-term and short-term spectral dependencies and have found widespread application in HSI classification [17]. Fully convolutional networks (FCNs), a popular model in image segmentation, have been extensively employed in hyperspectral remote sensing tasks [18]. Transformers, which have shown significant advancements in recent years, have also been successfully applied to HSI classification [19–23]. Furthermore, graph convolutional networks (GCNs) have gained attention in HSI classification and have achieved notable performance [24,25].

However, the majority of these models for HSI analysis are primarily patch-based, necessitating laborious preprocessing steps and resulting in substantial storage requirements. Consequently, several studies [20,22,26,27] have attempted to address these challenges by directly performing semantic segmentation on HSI. In these approaches, HSIs are treated as multi-channel images, akin to conventional RGB images, and external ground object labels are employed for annotation. This process can be seen as manually marking and selecting regions of interest within the ROItools [28]. During the loss calculation, only the known ground object types are considered for gradient computation using masks. Experimental verification has demonstrated the simplicity and effectiveness of this approach. Nevertheless, the spectral–spatial characteristics of hyperspectral images are often not fully taken into account by most existing methods. Yu et al. [26] integrated Transformer features directly within the decoder part, overlooking the intrinsic global relationship between distinct patches [25]. In a similar vein, Chen et al. [20] employed a combination of convolution and Transformer in the encoder part to extract hyperspectral image (HSI) features. However, their approach models the spectral sequence in the upper layer of the model, while the spatial characteristics are modeled in the lower layer, thereby neglecting the consideration of consistent spectral–spatial characteristics.

Spatial–spectral fusion methods have been extensively employed in hyperspectral classification tasks for over a decade. Early research focused on analyzing the size, orientation, and contrast characteristics of spatial structures in images, followed by the utilization of support vector machines (SVMs) for classification purposes [29]. Subsequent studies explored supervised classification of hyperspectral images through segmentation and spectral features extracted from partition clustering [30]. Li et al. [31] investigated the use of 3D convolutional neural networks (3DCNN) for direct spatial–spectral fusion in classification tasks. More recently, a two-stage method inspired by image denoising and segmentation was proposed in [32] to merge spatial and spectral information. Moreover, Qiao et al. [33] introduced a novel approach that captures information by concurrently considering the interactions between channels, spectral bands, spatial depth and width. However, it should be noted that these methods primarily operate at the patch level and may not be directly applicable to semantic segmentation tasks.

Some recent works [34,35] have focused on enhancing convolutional modules to better capture spatial and channel details, yielding impressive performance across various tasks. However, when applied to HSIs, extracting both spatial and spectral features comprehensively becomes crucial. Conventional 2D convolutions are insufficient for effective hyperspectral feature extraction, while 3D convolutions exhibit high complexity and parameter redundancy. Thus, to address these limitations holistically, there is a need to employ modules that can extract both spectral and spatial features in hyperspectral tasks, thereby replacing traditional 2D and 3D convolutions. Several studies in the field of HSI [36,37], use new modules with attention mechanisms and multi-scale features to replace traditional convolutions, and have achieved good results in HSI patch-based classification tasks. However, these modules need to be used in conjunction with various different modules, and at the same time, the online module has a high number of parameters and complexity, making it difficult to apply to semantic segmentation tasks.

To simplify the modeling of spectral–spatial relationships in hyperspectral imaging sequences and establish a unified hyperspectral image semantic segmentation architecture. This paper proposes a novel image-based global spectral–spatial feature learning framework called MSSFF. In contrast to conventional classification methods, MSSFF utilizes the MMFF module to hierarchically model features in spectral–spatial sequences, resulting in outstanding classification performance even with a limited number of labeled samples (refer to Figure 1). Firstly, in the encoder component, effective extraction of hyperspectral features is achieved by incorporating a spectral feature fusion module and a spatial feature fusion module. Secondly, an efficient Transformer is introduced between the encoder and decoder to capture global dependencies among deep feature nodes. Lastly, a spatial attention mechanism is employed in the upper layer of the model to model region-level features. The contributions of our proposed MSSFF framework can be summarized as follows.

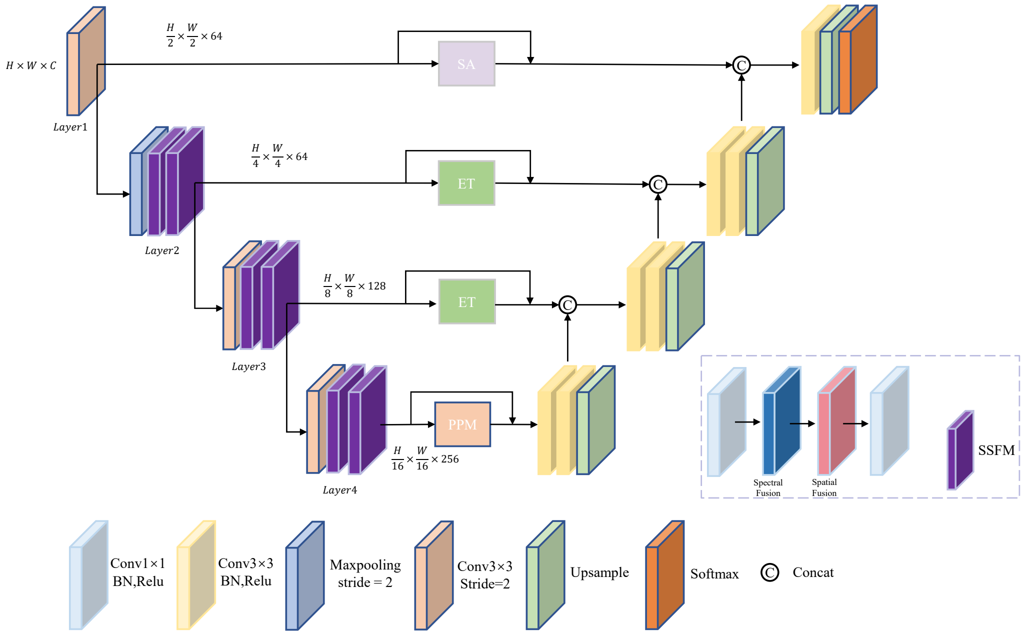


Figure 1. The overall framework of MSSFF. In the encoder, the first convolutional layer is modified to align with the spectral channel of the HSI. To enhance multi-scale feature extraction, a PPM is added at the encoder's end. Skip connections aid in gradient backpropagation, while the ET module captures global information, and the SA module focuses on local features in the upper layer. The decoder comprises three groups of upsampling and convolutional layers. During model training, only known samples are used to compute the loss gradient, excluding unknown samples.

The contributions of this paper can be summarized as follows:

(1) The paper introduces the MSSFF framework, a new method for hyperspectral semantic segmentation. It reevaluates the importance of spectral and spatial features and incorporates them effectively into the encoder. The framework also includes a Transformer in the skip connection section to capture global spectral–spatial information

from the feature map. This demonstrates the potential of Transformers in modeling spectral–spatial feature maps for hyperspectral remote sensing.

(2) We conducted a series of ablation experiments and module selection experiments to investigate the optimal depth of the hyperspectral semantic segmentation model. The results of these experiments confirmed that increasing the depth of the model beyond a certain point does not necessarily yield improved performance. Additionally, we explored the order of feature fusion and found that performing spectral feature fusion before spatial feature fusion yields better results. These findings suggest that considering spectral information before spatial information enhances the performance of the hyperspectral semantic segmentation model.

(3) We performed comparative experiments involving the patch-based method and the semantic segmentation method to assess the feasibility of our proposed approach in the field of hyperspectral semantic segmentation. The results of these experiments confirmed the effectiveness and viability of our method for hyperspectral semantic segmentation.

2. Method

As shown in Figure 1, we find that shallow models can effectively classify HSIs, so we propose an end-to-end shallow semantic segmentation model. HSIs are rich in spatial and spectral information, and spectral correlation and spatial correlation should be fully utilized for modeling. Therefore, in this work, we first propose a Backbone that simultaneously extracts spatial and spectral features, we use SSFM to replace the traditional convolution module, and at the end of the model, we use a pyramid pooling strategy to capture multiple scale contexts. In the decoder part, we followed the standard Unet architecture. However, we introduce the efficient Transformer in the skip connection part to model the deep feature map globally, and for the shallow (topmost) feature map, we use the spatial attention module for shallow feature extraction. Through the above modules, the accuracy of HSI classification is significantly improved. The following sections describe the core components of the framework.

The framework adopts an encoder–decoder architecture, and the encoder part is similar to ResNet18 [38], but we use SSFM to replace the standard Conv module in ResNet. In general, we need to pad the boundaries of the input HSI. We choose to fill the length and width of the HSI to a multiple of 16, assuming the input is an Indian image $I \in \mathbb{R}^{145\times145\times200}$, we fill it with $I \in \mathbb{R}^{160\times160\times200}$. The HSI is directly input for forward calculation. In the encoder part, we replace the input parameter of Backbone's first convolutional layer with the number of HSI spectral channels. A pyramid pooling module (PPM) is introduced at the end of the encoder. The multi-scale features extracted by the multi-scale aggregation module are very effective for the modeling of the framework. Residual connections between PPM and underlying feature maps can better facilitate gradient backpropagation. In the decoder part, one upsampling layer and two convolutional layers are set as a group, and there are three groups of upsampling modules in total. Before the upper and lower layer features are fused, the features of the encoder are enhanced by the ET or SA module, and then concat with the upsampled output of the lower layer features. Perform the same operation as above for each layer feature map of the encoder, and finally sample the feature map to the input size. To compute the loss, a small number of samples from the region are used to construct the mask. For the output of each batch, we only calculate the gradient of the known samples after the mask, and do not calculate the unknown samples.

2.1. Spectral–Spatial Fusion Module (SSFM)

To enhance the feature extraction capabilities of traditional 2D convolutions in both spectral and spatial domains, we introduce the concept of SSFM. Our approach involves the extraction and fusion of features from both the spectral and spatial dimensions. Specifically, we propose SSFM that applies the spectral feature fusion module first, followed by the connection of spatial feature fusion modules. The order of these modules will be discussed in the experimental results section.

2.1.1. Spectral Fusion Module

In order to fully leverage the potential of spectral features, we propose the integration of a spectral feature fusion module, as depicted in Figure 2. This module employs a split-extract-fusion strategy, which aims to address the challenges associated with extracting effective feature maps along the spectral dimension. In computer vision [38–40], particularly in the context of HSIs, the use of repeated convolutions for feature extraction can pose difficulties in capturing informative spectral-specific features, which has been identified as a critical issue [20–22]. Therefore, our proposed spectral feature fusion module provides a solution to overcome this flaw and improve the ability to extract meaningful spectral features in HSI analysis.

Given an input feature map $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$, firstly, we divide the features into two parts: $\mathbf{X}_1 \in \mathbb{R}^{H\times W\times (C/2)}$ and $\mathbf{X}_2 \in \mathbb{R}^{H\times W\times (C/2)}$, based on the spectral dimension C. Simultaneously, both feature sets undergo a 1×1 convolution operation, which compresses their dimensions by half, resulting in $\mathbf{X'}_1 \in \mathbb{R}^{H\times W\times (C/4)}$ and $\mathbf{X'}_2 \in \mathbb{R}^{H\times W\times (C/4)}$. Next, the features from the upper layer undergo extraction using both 1×1 and 3×3 convolution modules. Concatenation is then performed to obtain $\mathbf{X''}_1 \in \mathbb{R}^{H\times W\times C}$. Similarly, the features from the lower layer pass through a 1×1 convolution module while preserving their original features. Concatenation is performed again to obtain $\mathbf{X''}_2 \in \mathbb{R}^{H\times W\times C}$.

To obtain the combined feature representation, $\mathbf{X''}_1$ and $\mathbf{X''}_2$ are concatenated, resulting in the total feature representation $\mathbf{X''} \in \mathbb{R}^{H\times W\times 2C}$. Subsequently, an average pooling (Avg-Pooling) operation is applied to $\mathbf{X''}$, and the resulting weights are divided into two parts, corresponding to $\mathbf{X''}_1$ and $\mathbf{X''}_2$. These weights are used to perform feature weighting on the respective feature sets. Finally, the two weighted features are superimposed at the end of the module.

The following formula can be used to summarize:

$$\mathbf{X}_1, \mathbf{X}_2 = \mathrm{Split}(\mathbf{X}), \tag{1}$$

$$\mathbf{X'}_1 = \mathbf{W}^{C_1}\mathbf{X}_1, \mathbf{X'}_2 = \mathbf{W}^{C_2}\mathbf{X}_2, \tag{2}$$

where the operation denoted by *split* signifies the splitting of the input along the spectral dimension. Specifically, $\mathbf{W}^{C_1} \in \mathbb{R}^{(C/2)\times 1\times 1\times (C/4)}$ and $\mathbf{W}^{C_2} \in \mathbb{R}^{(C/2)\times 1\times 1\times (C/4)}$ are learnable weight matrices. These matrices are employed to facilitate the spectral-wise splitting and manipulation of the input features.

$$\mathbf{X''}_1 = \mathrm{Concat}\Big(\mathbf{W}^{C_{11}}\mathbf{X'}_1, \mathbf{W}^{C_{12}}\mathbf{X'}_1\Big), \tag{3}$$

$$\mathbf{X''}_2 = \mathrm{Concat}\Big(\mathbf{W}^{C_{13}}\mathbf{X'}_2, \mathbf{X'}_2\Big), \tag{4}$$

where we define the learnable weight matrices associated with different components as follows: $\mathbf{W}^{C_{11}} \in \mathbb{R}^{(C/2)\times 1\times 1\times (C/4)}$ represents the weight matrix for C_{11}, $\mathbf{W}^{C_{12}} \in \mathbb{R}^{(C/2)\times 1\times 1\times (C/4)}$ denotes the weight matrix for C_{12}, and $\mathbf{W}^{C_{13}} \in \mathbb{R}^{(C/2)\times 1\times 1\times (C/4)}$ corresponds to the weight matrix for C_{13}. These weight matrices are learnable parameters that are utilized within the given formulation for various processing steps and transformations. The function Concat refers to dimension concatenation.

$$\mathbf{X''} = \mathrm{Concat}(\mathbf{X''}_1, \mathbf{X''}_2), \tag{5}$$

After performing feature extraction, instead of directly concatenating or adding the two types of features, we adopt the approach proposed in [41,42] to selectively merge the output features from the feature extraction stage, denoted as $\mathbf{X''}_1$ and $\mathbf{X''}_2$. Subsequently, we apply global Avg-Pooling to aggregate the global spatial information and obtain $\mathbf{X}_{avg}$, which includes spectral statistics. Next, we normalize the global spatial information and multiply it element-wise with the feature map $\mathbf{X}''$, resulting in the generation of the feature importance vector $\mathbf{Y}$. To further refine the feature representation, we split the feature vector $\mathbf{Y}$ into two equal parts, yielding $\mathbf{Y}_1$ and $\mathbf{Y}_2$. Finally, we superimpose $\mathbf{Y}_1$ and $\mathbf{Y}_2$ to obtain the spectral refinement feature $\hat{\mathbf{Y}}$.

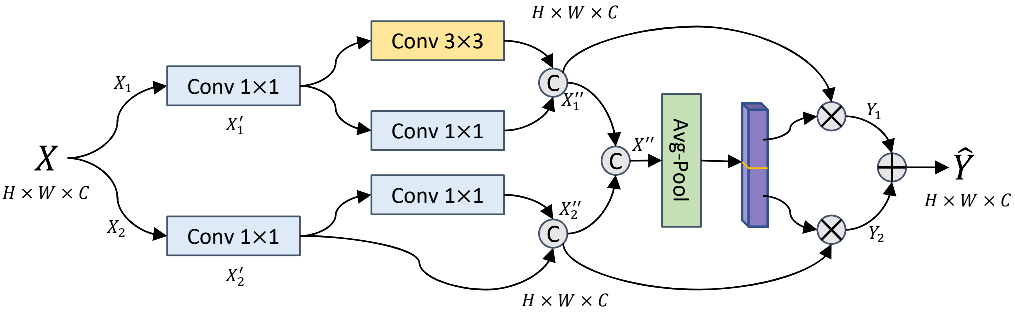


Figure 2. Spectral fusion module. This module employs a split-extraction-fusion strategy to enhance spectral features.

2.1.2. Spatial Fusion Module

To ensure the encoder effectively captures spatial features, we propose the integration of a spatial feature fusion module, as illustrated in Figure 3. This module employs separation and fusion operations to enhance its functionality. The primary objective of the separation operation is to distinguish informative feature maps from those containing comparatively less relevant spatial content. By subsequently fusing feature maps that possess rich information with those exhibiting lesser information, we can extract more comprehensive feature information than what can be achieved through convolution operations alone.

Specifically, we propose a method that utilizes group normalization (GN) for a given feature $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$. GN partitions the input spectral dimension into 16 groups, enabling independent calculations of the mean μ and variance σ for each group. The mean is computed by averaging the values within a group, while the variance is determined by calculating the squared differences between each value and the mean, followed by averaging the squared differences. Subsequently, the activations within each group are normalized by subtracting the group mean and dividing by the square root of the group variance. This normalization process ensures consistent and efficient feature scaling within each group. GN introduces learnable parameters, which include scaling and shifting factors for each group. These parameters enable the network to learn optimal scaling and shifting of the normalized activations. The scaling factor γ adjusts the normalized value, allowing for fine-grained control of the feature representation, while the shift factor β introduces a bias to the normalized value, aiding in capturing higher-order feature interactions.

$$\mathrm{GN}(\mathbf{X}) = \gamma \frac{\mathbf{X} - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \beta, \tag{6}$$

Simultaneously, the scaling factor γ within the GN layer serves as an indicator to quantify the variance of spatial pixels within each spectral dimension. The value of γ reflects the extent of spatial pixel variation, with richer spatial information resulting in a larger γ value. To obtain the weights for different feature maps, the following formula is employed: the features are multiplied with the weights within the GN layer. Subsequently, a sigmoid function is utilized to map the feature values to the interval [0, 1]. This process enables effective modulation and normalization of the feature representations.

$$\boldsymbol{W} = \frac{\gamma_i}{\sum_{n=1}^{C} \gamma_n}, i, n = 1, 2, \cdots, C, \tag{7}$$

$$\mathbf{X}_{mid} = \text{Sigmoid}(\text{GN}(\mathbf{X}) \otimes W), \tag{8}$$

Subsequently, a mask is constructed for the feature $\mathbf{X}_{mid}$ based on a threshold of 0.5. Values greater than or equal to 0.5 are assigned to $\mathbf{x}_1$, while values less than 0.5 are assigned to $\mathbf{x}_2$. These divisions result in two weighted features: $\mathbf{X}_1$, representing the information-rich feature, and $\mathbf{X}_2$, representing the less informative feature. To enhance the spatial feature fusion capability of the module and reduce spatial redundancy, the feature with rich information is added to the feature with less information. This is followed by a cross-reconstruction operation that facilitates comprehensive integration of the two weighted features, allowing for effective information exchange and generating more informative features. The resulting cross-reconstructed features are then concatenated to obtain spatial detail features, capturing fine-grained spatial information.

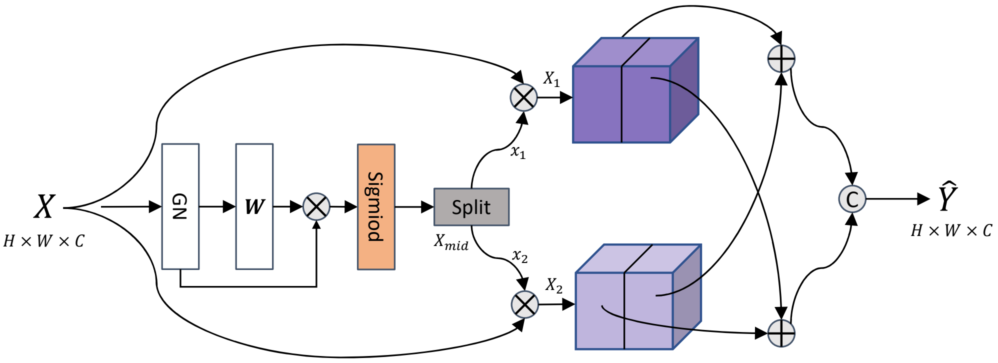


Figure 3. Spatial feature fusion module. This module employs separation and fusion operations to enhance spatial features.

2.2. Efficient Transformer (ET)

The standard Transformer model exhibits limitations in terms of high computational complexity and a lack of explicit spatial structure modeling. To address these shortcomings, researchers have proposed various enhanced Transformer models aimed at improving their performance in computer vision tasks. For instance, attention mechanism improvements [43], locality-based attention [44], and hybrid models [45] have been developed. Consequently, it is valuable to explore the integration of Transformer with convolutional models.

Recent research endeavors [46,47] have focused on replacing positional embedding in the Transformer model with convolution operations. By incorporating convolution operations into the Transformer, it becomes possible to effectively combine local and global features. Building upon the aforementioned concept, we present the ET that utilizes convolutional operations to effectively reduce the dimensionality of the feature space while capturing positional information. The architecture of ET is depicted in Figure 4. Furthermore, we introduce convolutional layers at both the input and output of the module to enhance the extraction of spatial features.

Space-reduced Efficient Multi-head Self-Attention (SEMSA) operates in a similar manner to Multi-head Self-Attention (MSA), as it takes **Q** (query), **K** (key), and **V** (value) as input and produces features of the original size as output. However, a key distinction lies in that SEMSA reduces the spatial scale of K and V before the attention operation. This reduction significantly diminishes the computational and memory overhead.

Specifically, in our study, we employ SEMSA as a replacement for the traditional MSA in the encoder module. Each instance of the ET comprises an attention layer and a feed-forward layer (FFN). Considering the high-resolution feature maps involved in hyperspectral semantic segmentation, we utilize convolution (SR) to reduce the spatial dimension of these feature maps while simultaneously learning spatial information. SEMSA operates in a similar manner to MSA, as it takes **Q**, **K** and **V** as input and produces features of the original size as output. However, a key distinction lies in that SEMSA reduces the spatial scale of **K** and **V** before the attention operation. This reduction significantly diminishes the computational and memory overhead. The SEMSA of stage i can be expressed as follows.

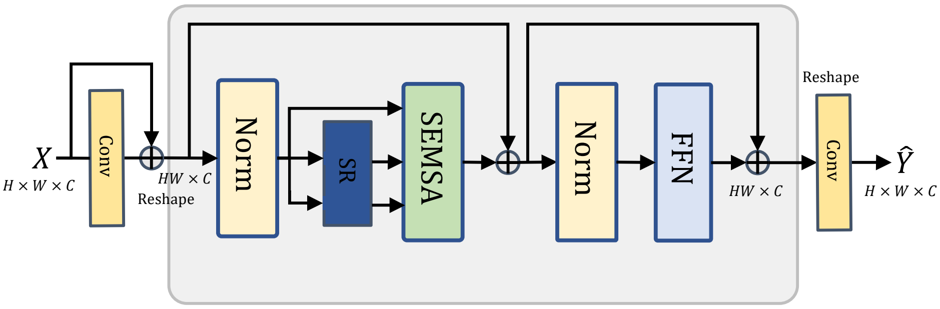


Figure 4. Efficient Transformer (ET), which utilizes convolution operations to efficiently reduce the dimensionality of the feature space while effectively capturing global information.

$$\text{SEMSA}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{Concat}(\mathbf{head}_0,\ldots,\mathbf{head}_N)\mathbf{W}^o, \tag{9}$$

Then, for the i-th *head*, it can be expressed by the following formula:

$$\mathbf{head}_i = \text{Attention}\Big(\mathbf{Q}\mathbf{W}_i^Q, \text{SR}(\mathbf{K})\mathbf{W}_i^K, \text{SR}(\mathbf{V})\mathbf{W}_i^V\Big), \tag{10}$$

where $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$, and $\mathbf{W}_i^V \in \mathbb{R}^{C\times C'}$ represent linear projection matrices, and the size C' of each head is equal to C/N. Here, N represents the number of attention heads. The function $SR(\cdot)$ denotes the utilization of convolution to reduce the dimensionality of the input feature space based on the reduction rate r^*.

$$\text{SR}(\mathbf{x}) = \text{Norm}\Big(\text{Reshape}(\mathbf{x}, r^*)\mathbf{W}^S\Big), \tag{11}$$

where $x \in \mathbb{R}^{HW\times C}$, where HW represents the spatial dimensions of the input and C denotes the number of spectrals. The reduction rate is denoted as R. The operation $\text{Reshape}(\mathbf{x}, r^*)$ refers to transforming x into a new shape of $\frac{HW}{R^2} \times R^2C$. Here, $\mathbf{W}^S \in \mathbb{R}^{R^2C\times C}$ corresponds to a linear projection matrix.

The attention calculation is defined as follows:

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V}, \tag{12}$$

where **Q**, **K**, and **V** represent the query, key, and value matrices, respectively. The variable d represents the dimension of the sequence.

2.3. Pyramid Pooling Module (PPM)

The PPM is shown in Figure 5. For the hyperspectral semantic segmentation task, it is crucial to consider spatial features at different scales. Utilizing pooling modules with varying sizes allows for the extraction of spatial feature information at different scales, thereby enhancing the model's robustness. To further address the loss of context information between different subregions, approaches such as [48,49] have introduced a hierarchical global prior structure. By incorporating language information from various scales and subregions, a global scene prior can be constructed based on the final layer feature map of the deep neural network, leading to significant improvements in region segmentation accuracy.

To implement this, the input feature map $\mathbf{X} \in \mathbb{R}^{H\times H\times C}$ is transformed into four feature maps with different spatial sizes. Subsequently, 1x1 convolutions are applied to reduce the dimensionality of the four feature maps. Next, the four different feature maps are resized to match the size of the input feature map using linear interpolation. Finally, the input feature map is concatenated with the four interpolated feature maps.

The above process can be expressed by the formula

$$\mathrm{PPM}(\mathbf{X}) = \mathrm{Concat}(\mathrm{Pool}_1(\mathbf{X}), \mathrm{Pool}_2(\mathbf{X}), \ldots, \mathrm{Pool}_n(\mathbf{X})), \tag{13}$$

$$\hat{\mathbf{Y}} = \mathrm{ConvModule}(\mathrm{PPM}(\mathbf{X})), \tag{14}$$

where $\mathbf{X}$ denotes the input feature map. $\mathrm{Pool}_i(\mathbf{X})$ represents the outcome of the ith pooling operation applied to the input feature map. The variable n signifies the number of pooling operations employed within the PPM. The function Concat refers to the concatenation of all the pooling results along the spectral dimension. Lastly, ConvModule represents a module encompassing convolution, batch normalization, and ReLU activation.

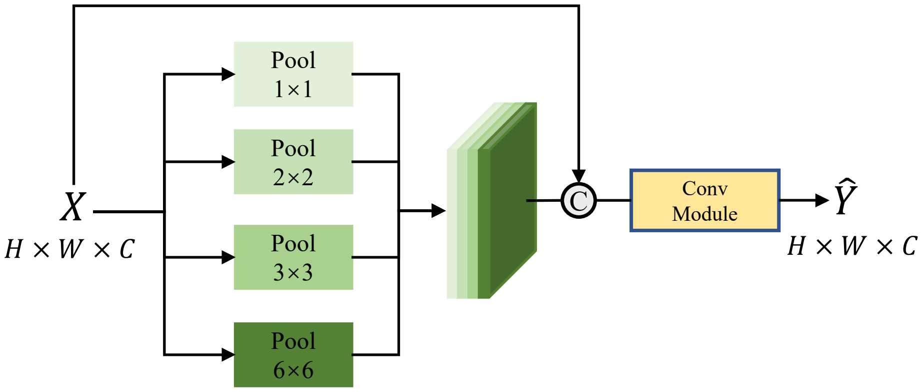


Figure 5. Pyramid pooling module (PPM), which helps enhance the model's understanding of complex visual scenes by aggregating features from different spatial scales.

2.4. Spatial Attention (SA)

The spatial attention in our work is modified from that in [39]. To apply SA, we first reduce the dimensionality of the channel features. Then, we perform average pooling and maximum pooling operations on the features to obtain corresponding results using the "avg" and "max" operations, respectively. These pooled features are concatenated together to form a single feature map.

Next, we utilize a two-dimensional convolutional layer with a kernel size of $(7, 7)$ to process the concatenated feature map. This convolutional operation can be represented by the following formula:

$$\hat{\mathbf{Y}} = \mathbf{X} \cdot \mathrm{Sigmoid}\left(\mathbf{W}^{SA}\left[\mathbf{X}_{avg}, \mathbf{X}_{max}\right]\right), \tag{15}$$

where $\mathbf{W}^{SA} \in \mathbb{R}^{1\times7\times7\times2}$ represents a learnable weight matrix. $\mathbf{X}_{avg}$ and $\mathbf{X}_{max}$ represent avg-pooling and maxpooling operations respectively, $\mathrm{Sigmoid}(\cdot)$ represents sigmoid activation function, and $\hat{\mathbf{Y}}$ represents module output features.

3. Experiments

3.1. Experimental Platform Parameter Settings

All experiments were conducted on a Windows 11 system equipped with an Intel (R) Core (TM) i5 10400 CPU @ 2.90 GHz processor and Nvidia GeForce RTX 3060 graphics card. To minimize experimental variability, the model adopts a controlled sampling approach by selecting a limited number of samples from the dataset for training. The experiment is conducted over 150 epochs, and all reported results are averaged over 5 independent experiments to ensure statistical significance. The model employs the AdamW optimizer with default parameters and initializes the learning rate to 5×10^{-4} . The loss function uses the standard cross-entropy, and the training process is the same as that in the literature [20,26]. We employ the hierarchical mask sampling method for calculating the loss function in our model. Specifically, we utilize masks to isolate relevant regions and compute the cross-entropy loss between the masked vectors and the corresponding ground truth objects. However, the presence of imbalanced class distributions and significant inter-class variations pose challenges. To address this, we adopt a strategy of random pixel sampling for known ground object categories. In this approach, we randomly select five pixels from each ground object category during multiple sampling iterations. This ensures comprehensive coverage of all known feature categories.

To verify the validity of the proposed method in this paper, a comparison is made between the segmentation effect of our proposed method (MSSFF) and several alternative methods, encompassing both patch-based approaches and semantic segmentation methods. The experiments are conducted on three publicly available datasets, namely Indian Pines (IA), Pavia Universitylia (PU), and Salinas (SA). In order to evaluate the performance of various models for HSI classification, the overall accuracy (OA), average accuracy (AA), and Kappa coefficient (K) are utilized as evaluation metrics.

3.2. Datasets

3.2.1. Indian Pines (IA)

The Indian Pines dataset was captured at a farm test site in northwest Indiana and collected using AVIRIS, an onboard sensor. In this paper, the data of 200 bands are classified after water absorption and low signal-to-noise ratio bands are eliminated. During the experiment, 10% of each type of ground object was selected for training, and the remaining samples were used for testing. When the number of selected samples of each type of ground object was less than five, we set it to 5. The specific training samples and test samples are shown in Table 1.

Table 1. The number of training and testing pixels per category in the IA dataset.

No.	Color.	Class.	Train.	Test.	Total.
1	🟩	Alfalfa	5	41	46
2	🟥	Corn-notill	143	1285	1428
3	🟨	Corn-mintill	83	747	830
4	🟦	Corn	24	213	237
5	🟪	Grass-pasture	49	434	483
6	⬜	Grass-trees	73	657	730
7	⬛	Grass-pasture-mowed	5	23	28
8	🟫	Hay-windrowed	48	430	478
9	🟫	Oats	5	15	20
10	🟩	Soybean-notill	98	874	972
11	🟪	Soybean-mintill	246	2209	2455
12	🟦	Soybean-clean	60	533	593
13	🟦	Wheat	21	184	205
14	🟧	Woods	127	1138	1265
15	🟨	Buildings-Grass-Trees	39	347	386
16	🟨	Stone-Steel-Towers	10	83	93
		Total	1036	9213	10,249

3.2.2. Pavia University (PU)

The dataset of Pavia University was shot in the University of Pavia, northern Italy, and was collected by airborne sensor ROSIS. In this paper, the data of 103 bands were classified by eliminating the bands affected by noise. During the experiment, 1% of each type of ground object was selected for training, and the remaining samples were used for testing. The specific training samples and test samples are shown in Table 2.

3.2.3. Salinas (SA)

The Salinas dataset was taken in the Salinas Valley, California, and the USA, and like the India dataset, it was collected using the airborne sensor AVIRIS. But unlike Indian Pines, it has a spatial resolution of 3.7 m. During the experiment, 1% of each type of ground object was selected for training, and the remaining samples were used for testing. The specific training samples and test samples are shown in Table 3.

Table 2. The number of training and testing pixels per category in the PU dataset.

No.	Color.	Class.	Train.	Test.	Total.
1	🟩	Asphalt	67	6564	6631
2	🟥	Meadows	187	18,462	18,649
3	🟨	Gravel	21	2078	2099
4	🟦	Trees	31	3033	3064
5	🟪	Metal sheets	14	1331	1345
6	⬜	Bare Soil	51	4978	5029
7	⬛	Bitumen	14	1316	1330
8	🟫	Bricks	37	3645	3682
9	🟫	Shadows	10	937	947
		Total	432	42,344	42,776

Table 3. The number of training and testing pixels per category in the SA dataset.

No.	Color.	Class.	Train.	Test.	Total.
1	🟩	Brocoli-green-weeds-1	21	1988	2009
2	🟥	Brocoli-green-weeds-2	38	3688	3726
3	🟨	Fallow	20	1956	1976
4	🟦	Fallow-rough-plow	14	1380	1394
5	🟪	Fallow-smooth	27	2651	2678
6	⬜	Stubble	40	3919	3959
7	⬛	Celery	36	3543	3579
8	🟫	Grapes-untrained	113	11,158	11,271
9	🟫	Soil-vinyard-develop	63	6140	6203
10	🟩	Corn-senesced-green-weeds	33	3245	3278
11	🟪	Lettuce-romaine-4wk	11	1057	1068
12	🟦	Lettuce-romaine-5wk	20	1907	1927
13	🟦	Lettuce-romaine-6wk	10	906	916
14	🟧	Lettuce-romaine-7wk	11	1059	1070
15	🟨	Vinyard-untrained	73	7195	7268
16	🟨	Vinyard-vertical-trellis	19	1788	1807
		Total	549	53,580	54,129

3.2.4. Houston (HU)

The Houston dataset was acquired using the ITRES CASI-1500 sensor in the vicinity of the University of Houston, Texas, USA, including nearby rural areas. This dataset serves as a benchmark and is commonly utilized to evaluate the performance of land cover classification models. The hyperspectral dataset consists of 349 × 1905 pixels with 144 wavelength bands spanning from 364 to 1046 nm at 10 nm intervals. During the experiment, 5% of each type of ground object was selected for training, and the remaining samples were used for testing. The specific training samples and test samples are shown in Table 4.

Table 4. The number of training and testing pixels per category in the Houston dataset.

No.	Color.	Class.	Train.	Test.	Total.
1	🟩	Healthy Grass	63	1188	1251
2	🟥	Stressed Grass	63	1191	1254
3	🟨	Synthetic Grass	35	662	697
4	🟦	Tree	63	1181	1244
5	🟪	Soil	63	1179	1242
6	⬜	Water	17	308	325
7	⬛	Residential	64	1204	1268
8	🟫	Commercial	63	1181	1244
9	🟫	Road	63	1189	1252
10	🟩	Highway	62	1165	1227
11	🟪	Railway	62	1173	1235
12	🟦	Parking Lot1	62	1171	1233
13	🟦	Parking Lot2	24	445	469
14	🟧	Tennis Court	22	406	428
15	🟨	Running Track	33	627	660
	Total		759	14,270	15,029

3.3. Comparative Experiment

Tables 5–8 present a comparative analysis of our proposed model alongside several patch-based frameworks, such as M3DCNN [50], HyBridSN [51], A2S2K [52], ViT [53], and SSFTT [54]. Additionally, the experimental results of Unet [55], PSPnet [48], Swin [44], and SegFormer [47], which are based on semantic segmentation frameworks, are also included for comparison. It is worth noting that semantic segmentation-based methods demonstrate superior performance in capturing global spatial information and exhibit significant advantages, particularly in scenarios with imbalanced training samples.

The experimental findings demonstrate the significant advantages of MSSFF when compared to both patch-based models and various semantic segmentation models. Specifically, M3DCNN, as a conventional 3DCNN model, suffers from parameter redundancy and inadequate extraction of spectral and spatial features, resulting in the poorest performance. ViT overlooks the unique characteristics of hyperspectral data by solely modeling the spectral sequence without considering the spectral similarity of ground objects, leading to subpar results. In contrast, HyBridSN leverages the strengths of both 3DCNN and 2DCNN, yielding certain improvements and highlighting the importance of feature redundancy in hyperspectral analysis. A2S2K adopts a residual-based 3DCNN approach where residual blocks are introduced into the hyperspectral domain. This design choice enables the model to effectively capture and exploit residual information, enhancing its ability to learn complex spatial and spectral features from hyperspectral data. Consequently, better results are achieved, although the computational complexity and parameter count of 3DCNN remain high. SSFTT employs a combination of 3DCNN and 2DCNN for feature extraction and incorporates Transformer to globally model the feature map. Notably, SSFTT outperforms other patch-based methods, underscoring the effectiveness of Transformers in modeling underlying feature maps.

However, the encoder component of Unet fails to fully consider the spatial and spectral characteristics of HSIs, resulting in poor correlation, particularly observed in the AA index, indicating significant misclassification issues with the Unet model. Similarly, PspNet shares the same encoder as Unet but introduces the PPM in the decoder to effectively capture semantic information at multiple scales, leading to improved performance. Swin Transformer incorporates Transformer in the encoder to globally model spectral and spatial features. Additionally, Swin Transformer includes UperNet in the decoder, enabling the capture of semantic information at various scales. Consequently, Swin Transformer demonstrates favorable results; however, Transformers still exhibit feature redundancy compared to convolutional methods.

In contrast, SegFormer leverages an efficient Transformer as the encoder while designing a simple and lightweight MLP decoder to reduce feature redundancy, resulting in outstanding performance across multiple tasks. Nevertheless, using a pure Transformer as the encoder for hyperspectral tasks may introduce invalid modeling, leading to poor model stability. To address this concern, MSSFF introduces SSFM, which considers both spectral and spatial features, as a replacement for the standard 2DCNN. The modification enhances stability and reduces model complexity. Additionally, MSSFF incorporates an efficient Transformer in the deep feature map, aligning with the findings of previous literature [54]. By considering feature extraction ability and model complexity, MSSFF achieves the best performance across the three datasets.

Table 5. Classification accuracy (%) of the IA image with different methods.

No.	M3DCNN	HyBridSN	A2S2K	ViT	SSFTT	Unet	PspNet	Swin	SegFormer	MSSFF
1	73.913	97.826	**100.000**	95.652	95.652	82.609	95.652	97.826	93.478	97.826
2	89.636	97.619	97.689	94.398	98.880	98.529	97.899	98.529	99.020	**99.510**
3	91.566	97.952	97.711	94.578	98.313	97.108	97.470	94.699	98.313	**99.036**
4	59.916	92.405	96.624	87.764	97.046	94.093	95.359	94.093	97.890	**100.000**
5	94.617	98.137	99.172	98.137	99.379	97.308	**100.000**	98.758	99.172	98.758
6	98.767	99.863	**100.000**	99.041	100.000	99.178	97.808	98.630	99.315	98.630
7	39.286	**100.000**	**100.000**	53.571	**100.000**	92.857	**100.000**	**100.000**	**100.000**	**100.000**
8	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	99.791	99.582	**100.000**	99.791	99.791
9	15.000	**100.000**	**100.000**	70.000	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**
10	90.123	99.280	98.251	98.457	99.486	96.914	**99.691**	98.868	**99.691**	99.486
11	92.872	99.430	99.674	95.764	99.104	**99.552**	97.882	95.642	99.430	99.511
12	83.305	94.772	97.639	88.702	95.110	97.133	94.435	**99.325**	97.133	98.988
13	99.512	99.024	98.049	98.049	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**
14	95.336	99.605	99.684	99.289	**100.000**	**100.000**	**100.000**	99.684	**100.000**	**100.000**
15	83.420	89.119	97.409	95.078	98.187	99.741	99.741	**100.000**	99.741	**100.000**
16	88.172	**100.000**	**100.000**	**100.000**	**100.000**	86.022	97.849	95.699	93.548	96.774
OA	91.228	98.234	98.819	96.009	98.976	98.429	98.312	97.795	99.151	**99.424**
AA	80.965	97.814	98.869	91.780	98.822	96.302	98.336	98.235	98.533	**99.269**
K	89.993	97.986	98.654	95.451	98.832	98.208	98.077	97.489	99.032	**99.344**

Table 6. Classification accuracy (%) of the PU image with different methods.

No.	M3DCNN	HyBridSN	A2S2K	ViT	SSFTT	Unet	PspNet	Swin	SegFormer	MSSFF
1	92.610	97.587	99.020	96.230	96.954	98.341	96.230	99.502	97.768	**99.955**
2	99.163	99.995	**100.000**	99.844	99.844	99.914	99.930	99.571	99.920	**100.000**
3	73.273	92.139	94.378	88.899	88.852	80.515	98.190	97.904	**99.285**	98.285
4	83.322	90.601	96.377	94.191	96.932	92.004	85.901	90.078	96.377	**98.792**
5	98.290	**100.000**	**100.000**	**100.000**	**100.000**	99.926	92.416	99.331	99.405	**100.000**
6	83.058	**100.000**	98.867	96.083	99.165	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**
7	48.496	98.947	91.955	96.617	99.925	91.053	96.842	98.722	99.248	**99.850**
8	73.574	94.324	87.344	78.599	94.758	91.798	99.620	97.882	99.321	**100.000**
9	23.337	92.819	96.410	97.888	93.031	89.229	88.807	72.122	82.049	**99.472**
OA	88.365	97.884	97.760	95.932	97.987	96.952	97.669	98.062	98.826	**99.806**
AA	75.014	96.268	96.039	94.261	96.607	93.642	95.326	95.012	97.041	**99.595**
K	84.316	97.190	97.027	94.592	97.329	95.950	96.908	97.429	98.445	**99.743**

Table 7. Classification accuracy (%) of the SA image with different methods.

No.	M3DCNN	HyBridSN	A2S2K	ViT	SSFTT	Unet	PspNet	Swin	SegFormer	MSSFF
1	**100.000**	**98.457**	**100.000**	**100.000**	**100.000**	88.552	**100.000**	99.004	99.851	**100.000**
2	**100.000**	**100.000**	**100.000**	**100.000**	99.866	99.544	**100.000**	98.658	**100.000**	**100.000**
3	99.899	**100.000**	**100.000**	99.949	**100.000**	93.421	**100.000**	**100.000**	**100.000**	**100.000**
4	92.539	98.852	99.785	98.278	**100.000**	96.700	**100.000**	99.713	**100.000**	**100.000**
5	98.208	98.842	98.096	97.461	98.581	95.108	99.627	99.627	99.328	**99.813**
6	99.949	99.848	**100.000**	**100.000**	**100.000**	98.257	99.065	99.495	**100.000**	**100.000**
7	99.944	**100.000**	99.972	**100.000**	99.609	97.262	**100.000**	99.860	**100.000**	**100.000**
8	92.175	97.711	97.809	95.582	99.139	99.814	99.548	99.938	99.867	**99.991**
9	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	97.550	**100.000**	**100.000**	**100.000**	**100.000**
10	97.956	98.383	99.420	96.522	99.237	90.818	**100.000**	**100.000**	99.878	**100.000**
11	97.472	98.502	99.438	98.408	99.345	87.921	**100.000**	97.097	**100.000**	**100.000**
12	98.080	**100.000**	**100.000**	99.429	99.948	90.867	95.745	98.443	98.755	98.651
13	44.323	97.817	**100.000**	80.677	96.834	66.376	**100.000**	**100.000**	**100.000**	**100.000**
14	97.009	99.346	99.159	97.383	98.224	71.963	99.907	99.533	99.907	**100.000**
15	84.741	93.960	93.010	89.461	95.570	99.009	**100.000**	99.876	98.638	**100.000**
16	98.783	98.284	99.225	99.170	99.723	73.326	**100.000**	**100.000**	**100.000**	**100.000**
OA	94.746	98.323	98.415	96.824	98.962	95.082	99.666	99.647	99.697	**99.941**
AA	93.817	98.750	99.120	97.020	99.130	90.405	99.618	99.453	99.764	**99.903**
K	94.146	98.131	98.234	96.462	98.843	94.507	99.628	99.607	99.663	**99.934**

Table 8. Classification accuracy (%) of the Houston2013 image with different methods.

No.	M3DCNN	HyBridSN	A2S2K	ViT	SSFTT	Unet	PspNet	Swin	SegFormer	MSSFF
1	97.475	99.832	**100.000**	97.727	99.579	98.321	97.202	98.002	97.442	99.041
2	98.573	98.405	98.908	98.489	98.657	96.890	93.620	91.148	92.105	99.841
3	98.640	99.547	99.698	**100.000**	99.396	100.000	99.857	99.570	99.570	99.857
4	98.393	98.985	99.831	**100.000**	99.323	90.595	87.862	90.354	93.810	99.598
5	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	98.712	**100.000**	99.436	98.551	**100.000**
6	**100.000**	99.029	**100.000**	85.113	94.822	94.769	99.692	99.077	**100.000**	**100.000**
7	97.261	**98.506**	98.091	91.784	98.921	96.609	95.426	95.978	95.347	98.423
8	88.917	88.917	85.025	85.787	88.156	78.135	86.656	84.244	85.611	**91.399**
9	86.375	91.505	**93.272**	87.721	89.823	85.543	83.387	83.387	82.348	93.131
10	98.113	99.485	**100.000**	99.657	97.684	95.355	99.511	**100.000**	**100.000**	99.837
11	98.806	98.039	98.721	98.039	99.488	94.980	99.109	**100.000**	**100.000**	**100.000**
12	92.314	**99.402**	99.488	98.036	98.804	92.944	98.135	97.242	97.242	98.378
13	68.610	97.758	97.982	80.717	**98.655**	95.309	98.081	98.294	98.294	98.294
14	99.509	99.754	**100.000**	98.280	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**
15	**100.000**	**100.000**	**100.000**	99.841	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**
OA	95.314	97.647	97.710	95.462	97.367	93.785	95.016	94.903	95.143	**98.250**
AA	94.866	97.944	98.068	94.746	97.554	94.544	95.903	95.782	96.021	**98.520**
K	94.933	97.456	97.525	95.092	97.153	93.281	94.613	94.491	94.750	**98.108**

The classification results of different methods are presented in Figures 6–9. It can be observed from the figures that there is a significant number of misclassifications between M3DCNN and ViT, particularly when dealing with ground objects that exhibit similar spectral characteristics. However, HyBridSN, A2S2K, and SSFTT show some improvements, although there are still instances of misclassifications. Unet and PspNet, which take into account spatial characteristics, notably reduce the misclassification phenomenon in the central areas of ground objects. However, misclassification still occurs in the edge connection areas of different ground objects. Swin and SegFormer employ a hierarchical Transformer as the encoder, providing a global receptive field. Nevertheless, there are still misclassifications for ground objects with similar spectral and spatial characteristics. MSSFF shows significant improvements in mitigating misclassifications for ground objects with similar spectral and spatial characteristics, with only very few misclassifications occurring in the edge areas of different ground objects. Overall, MSSFF exhibits excellent classification performance for diverse ground objects, fully considering their spectral and spatial characteristics.

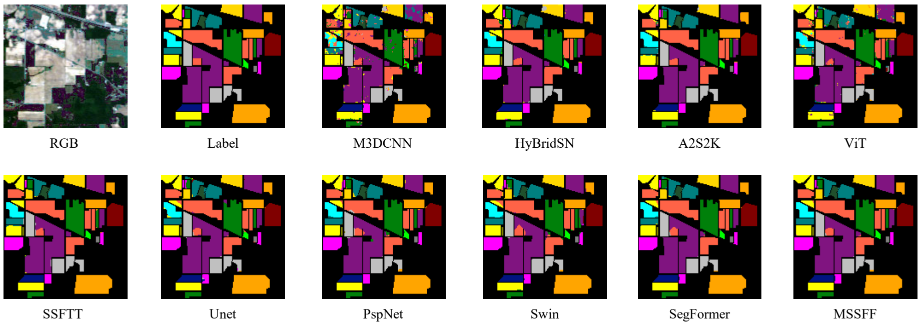


Figure 6. IA dataset ground feature classification result map.

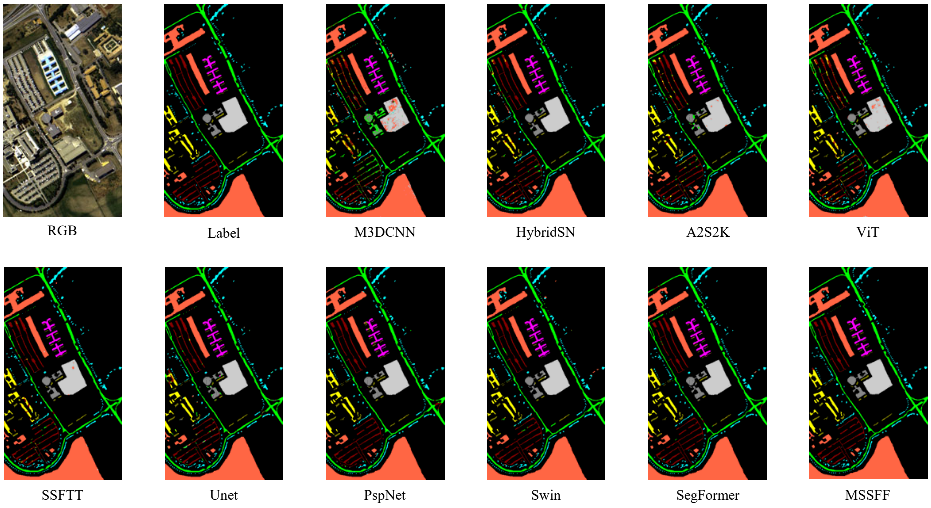


Figure 7. PU dataset ground feature classification result map.

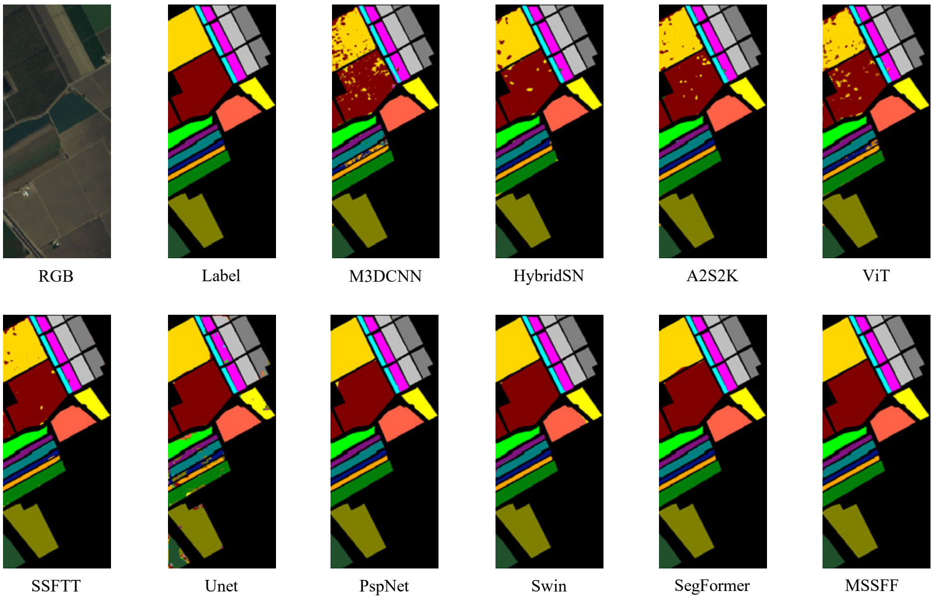
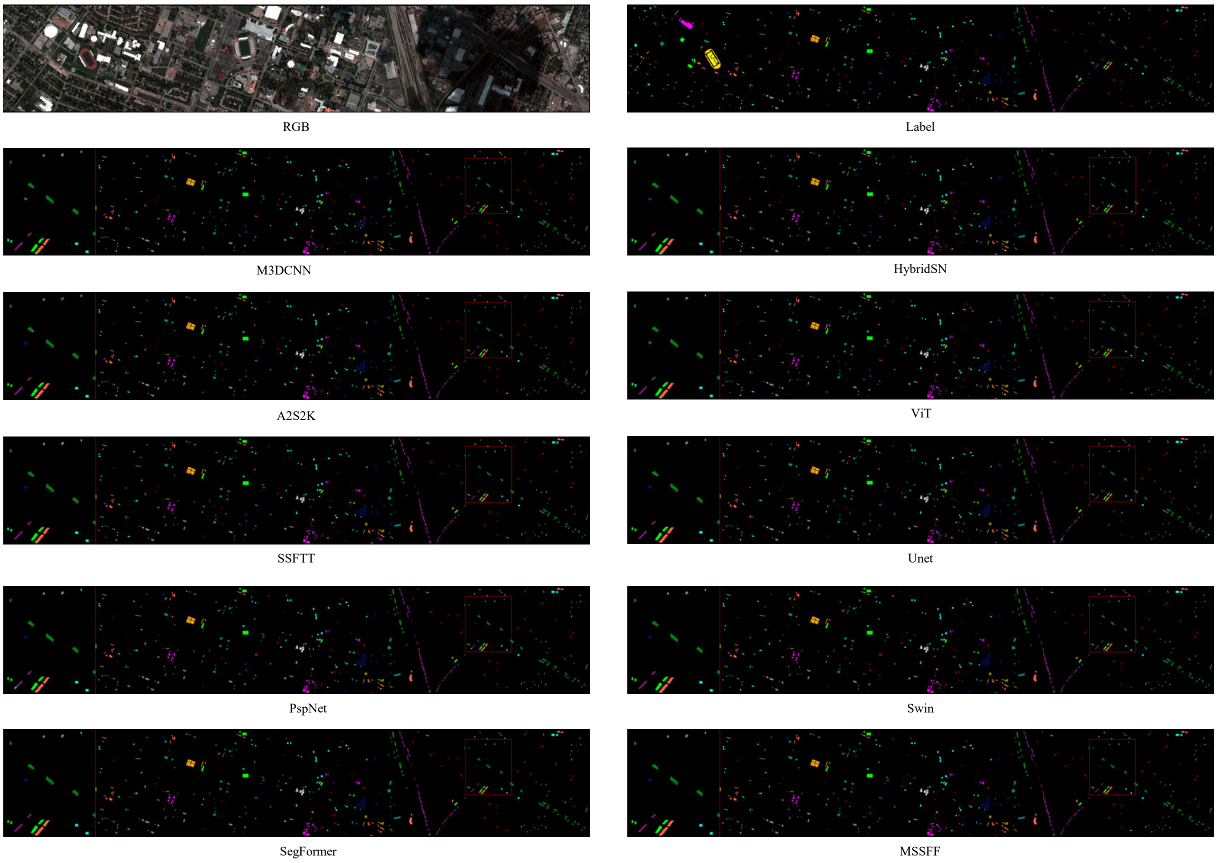


Figure 8. SA dataset ground feature classification result map.


RGB

Label

M3DCNN

HybridSN

A2S2K

ViT

SSFTT

Unet

PspNet

Swin

SegFormer

MSSFF

Figure 9. Houston2013 dataset ground feature classification result map.

3.4. Model Analysis

To verify the effectiveness of each component in the proposed MSSFF framework, this section focuses on conducting ablation experiments. Additionally, we also explore the selection of the number of layers in the encoder and the sequencing of the spectral feature fusion module and the spatial feature fusion module in SSFM.

3.4.1. Ablation Experiments

We conducted a series of ablation experiments to assess the individual contributions of the modules in the MSSFF method. The results of the ablation experiments are shown in Table 9. The MSSFF method comprises four modules: SSFM, PPM, ET, and SA. During the ablation experiments, we systematically removed these modules and evaluated the resulting changes in the classification metrics, namely OA, AA, and K.

When all modules were removed, the classification metric scores were relatively low, indicating the significant role of these modules in improving the classification performance. Specifically, when only the PPM was used, there was a significant improvement in the classification index, demonstrating its favorable impact on enhancing classification performance. Building upon the PPM, the addition of the ET module further improved the classification index, highlighting its positive influence on classification performance. The inclusion of the SA module resulted in slight improvements in the classification metrics. Although the observed improvements were small, they still indicated the contribution of the SA module to the enhancement of classification performance. Finally, when all modules (SSFM, PPM, ET, and SA) were utilized, the classification metrics (OA, AA, and K) achieved their highest levels. This observation underscores the effectiveness of combining these modules in improving the hyperspectral classification performance of the MSSFF method.

Table 9. Different module ablation experiments. The symbols "✓" and "✗" are used to indicate the act of selecting and not selecting a module, respectively.

SSFM	PPM	ET	SA	IA			PU			SA		
				OA	AA	K	OA	AA	K	OA	AA	K
✗	✗	✗	✗	98.556	98.367	98.353	98.745	98.193	98.338	99.507	99.461	99.451
✗	✓	✗	✗	99.054	98.027	98.921	99.140	98.642	98.860	99.666	99.645	99.628
✗	✓	✓	✗	99.180	98.468	99.065	99.439	98.934	99.256	99.797	99.720	99.774
✗	✓	✗	✓	99.093	98.486	98.965	99.275	98.835	99.040	99.782	99.803	99.757
✗	✓	✓	✓	99.219	98.546	99.110	99.444	98.986	99.263	99.869	99.829	99.854
✓	✗	✗	✗	99.083	98.739	98.954	99.435	99.076	99.183	99.758	99.599	99.768
✓	✓	✗	✗	99.132	98.801	99.010	99.584	99.302	99.449	99.871	99.781	99.856
✓	✓	✓	✗	99.317	98.949	99.221	99.640	99.324	99.523	99.887	99.799	99.875
✓	✓	✗	✓	99.268	98.995	99.166	99.619	99.477	99.495	99.882	99.748	99.868
✓	✓	✓	✓	99.424	99.269	99.344	99.806	99.595	99.743	99.941	99.903	99.934

Figure 10 illustrates the visualization of feature maps obtained from the MSSFF framework using SSFM and ET modules. A careful selection of representative feature maps was made for visual comparison, revealing that the visualization results obtained with the SSFM module exhibit enhanced refinement, capturing finer details such as object edges, contours, and texture structures. On the other hand, the visualization results obtained with the ET module demonstrate a wider receptive field and a greater emphasis on the overall context compared to those without ET. This visual analysis provides compelling evidence for the effectiveness and superiority of the designed SSFM and ET modules in the MSSFF framework.

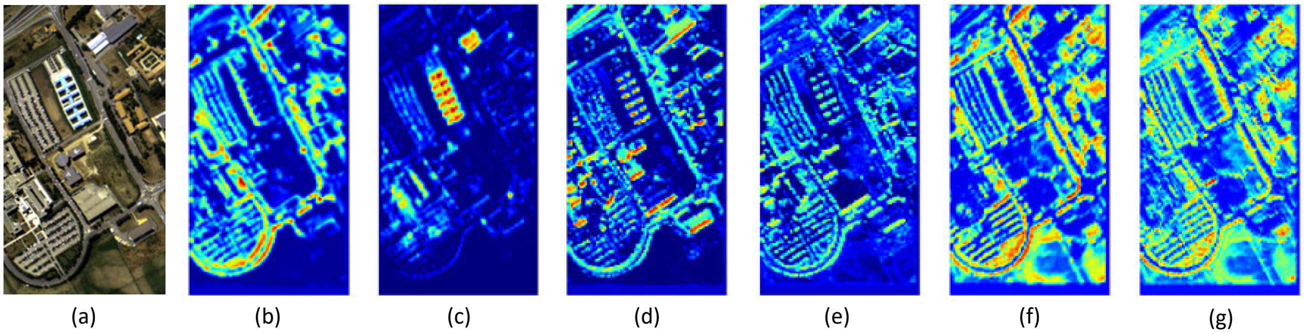


Figure 10. Visualization of selected encoder output features using three different methods. The different labels in the figure above refer to (**a**) RGB image, (**b**) and (**c**) base model, (**d**) and (**e**) using SSFM, and (**f**) and (**g**) using SSFM and ET.

3.4.2. Comparative Analysis of Attention Modules in MSSFF

We consider the impact of various types of attention modules on MSSFF. Specifically, we study and compare multiple existing attention mechanisms, including self-attention, channel attention, and spatial attention. Each attention module provides unique capabilities to capture different types of dependencies and enhances feature representation. Through comprehensive experiments, we identify the most effective attention module based on the characteristics of the dataset and the task goals. This systematic approach improves the performance of our deep learning models and enhances model interpretability. As shown in Table 10, the ET module achieved the best results on all three datasets.

Table 10. Attention module replacement experiment.

Method	**IA**			**PU**			**SA**		
	OA	**AA**	**K**	**OA**	**AA**	**K**	**OA**	**AA**	**K**
CBAM [39]	99.229	98.909	99.121	99.682	99.319	99.579	99.891	99.835	99.879
Triplet [56]	99.229	99.043	99.121	99.701	99.571	99.604	99.933	99.893	99.926
WMSA [44]	99.219	99.056	99.110	99.710	99.399	99.616	99.852	99.793	99.835
MSA [53]	99.346	99.248	99.255	99.659	99.401	99.548	99.906	99.868	99.895
ET	**99.424**	**99.269**	**99.344**	**99.806**	**99.595**	**99.743**	**99.941**	**99.903**	**99.934**

3.4.3. Fusion Module Order Selection

The results of the sequential selection experiments conducted on the spectral feature fusion module and spatial feature fusion module in SSFM are presented in Table 11. The feature fusion module employed in SSFM shares similarities with CBAM [39], as both require careful consideration of the order in which spectral and spatial dimensions are modeled. To comprehensively evaluate the impact of feature fusion, we divided the experiments into two parts: Space-Spectral and Spectral-Space.

Interestingly, our findings indicate that fusing the spectral dimension features of hyperspectral data prior to the fusion of spatial dimensions yields better results. We speculate that this is due to the fusion of spatial dimensions potentially causing a disruption to the spectral features, leading to a decline in the effectiveness of spectral feature fusion.

Table 11. Sequential selection experiments for feature fusion in SSFM.

No.	**IA**		**PU**		**SA**	
	Space-Spectral	**Spectral-Space**	**Space-Spectral**	**Spectral-Space**	**Space-Spectral**	**Spectral-Space**
1	93.478	**97.826**	98.975	**99.955**	**100.000**	**100.000**
2	**99.580**	99.510	99.887	**100.000**	**100.000**	**100.000**
3	97.349	**99.036**	**99.619**	98.285	**100.000**	**100.000**
4	**100.000**	**100.000**	97.324	**98.792**	**100.000**	**100.000**
5	98.344	**98.758**	99.851	**100.000**	99.701	**99.813**
6	98.493	**98.630**	**100.000**	**100.000**	**100.000**	**100.000**
7	**100.000**	**100.000**	**100.000**	99.850	99.860	**100.000**
8	99.791	99.791	**100.000**	**100.000**	**100.000**	99.991
9	**100.000**	**100.000**	98.944	**99.472**	**100.000**	**100.000**
10	99.074	**99.486**	–	–	**100.000**	**100.000**
11	99.389	**99.511**	–	–	**100.000**	**100.000**
12	98.314	**98.988**	–	–	96.679	**98.651**
13	**100.000**	**100.000**	–	–	**100.000**	**100.000**
14	**100.000**	**100.000**	–	–	**100.000**	**100.000**
15	**100.000**	**100.000**	–	–	99.725	**100.000**
16	96.774	96.774	–	–	99.225	**100.000**
OA	99.141	**99.424**	99.553	**99.806**	99.795	**99.941**
AA	98.787	**99.269**	99.400	**99.595**	99.699	**99.903**
K	99.021	**99.344**	99.409	**99.743**	99.772	**99.934**

3.4.4. Explore the Layers of Encoder

Regarding the impact of different layers in the encoder on the model, the corresponding results are presented in Table 12. Recent literature [20,54,57,58] has demonstrated the effectiveness of shallower models in hyperspectral object classification tasks. Therefore, we conducted an exploration by varying the number of layers in the encoder to assess their influence on model performance.

Table 12 clearly indicates that the number of layers in the encoder does not necessarily follow a "deeper is better" trend. Specifically, the model's performance does not consistently improve as the number of layers increases. On the contrary, there is a downward trend in model performance with an increasing number of layers. This phenomenon can be attributed to the introduction of excessive redundant information by overly deep encoders when processing hyperspectral data, which subsequently hampers model performance.

Based on these observations, we can conclude that for hyperspectral object classification tasks, a shallower encoder may be more suitable, and an excessively deep encoder does not necessarily lead to performance improvements. Thus, when designing the model, the number of layers in the encoder should be considered in a comprehensive manner, and an appropriate number of layers should be selected to achieve the optimal performance.

Table 12. Encoder layer exploration experiment.

No.	**IA**			**PU**			**SA**		
	3	**4**	**5**	**3**	**4**	**5**	**3**	**4**	**5**
1	93.478	**97.826**	95.652	98.854	**99.955**	99.005	**100.000**	**100.000**	99.900
2	99.300	**99.510**	99.300	**100.000**	**100.000**	99.995	**100.000**	**100.000**	**100.000**
3	98.072	**99.036**	97.229	95.141	**98.285**	98.142	**100.000**	**100.000**	**100.000**
4	**100.000**	**100.000**	99.578	98.172	**98.792**	98.597	**100.000**	**100.000**	**100.000**
5	**99.172**	98.758	**99.172**	**100.000**	**100.000**	**100.000**	99.552	**99.813**	99.776
6	**98.630**	**98.630**	98.356	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**
7	**100.000**	**100.000**	**100.000**	99.925	99.850	**100.000**	99.441	**100.000**	99.693
8	**99.791**	**99.791**	**99.791**	**100.000**	**100.000**	99.267	99.991	99.991	**100.000**
9	**100.000**	**100.000**	**100.000**	99.472	99.472	**99.683**	**100.000**	**100.000**	**100.000**
10	98.868	**99.486**	**99.486**	–	–	–	99.969	**100.000**	**100.000**
11	**99.511**	**99.511**	99.470	–	–	–	**100.000**	**100.000**	**100.000**
12	98.145	**98.988**	98.482	–	–	–	**98.651**	**98.651**	99.637
13	**100.000**	**100.000**	**100.000**	–	–	–	**100.000**	**100.000**	**100.000**
14	99.921	**100.000**	**100.000**	–	–	–	99.159	**100.000**	99.907
15	**100.000**	**100.000**	**100.000**	–	–	–	**100.000**	**100.000**	99.972
16	**96.774**	**96.774**	**96.774**	–	–	–	99.502	**100.000**	99.336
OA	99.200	**99.424**	99.190	99.439	**99.806**	99.582	99.856	**99.941**	99.924
AA	98.854	**99.269**	98.956	99.063	**99.595**	99.410	99.767	**99.903**	99.889
K	99.088	**99.344**	99.077	99.257	**99.743**	99.445	99.840	**99.934**	99.916

3.4.5. Mean Squared Error (MSE) Discussion on Different Methods

Although the confusion matrix accounts for the significant differences between different categories, we have observed that the patch-based methods (HyBridSN, A2S2K, and SSFTT) exhibit similar Kappa coefficients, OA, and AA. However, merely comparing the significance differences is insufficient to fully explain the relative merits of these methods. Therefore, we conducted further testing using the MSE metric on different datasets. The experimental results are shown in Table 13.

Through the analysis of the MSE metric, we have found that the SSFTT method demonstrated a distinct advantage over A2S2K and HyBridSN across all datasets. Particularly, on the lower-resolution IA and SA datasets, A2S2K showed relatively better performance compared to HyBridSN. However, on the higher-resolution PU dataset, A2S2K exhibited relatively poorer performance.

Table 13. MSE indicator values by different methods.

Dataset	**M3DCNN**	**HyBridSN**	**A2S2K**	**ViT**	**SSFTT**	**Unet**	**PspNet**	**Swin**	**SegFormer**	**MSSFF**
IA	3.4604	0.9210	0.5827	1.6187	0.4194	0.4436	0.5156	0.7707	0.2993	**0.2247**
PU	3.7061	0.5185	0.6401	1.1174	0.5167	0.6283	0.4062	0.4065	0.3093	**0.1153**
SA	1.9024	0.7576	0.6947	1.1847	0.4321	1.5610	0.0521	0.0604	0.1095	**0.0372**

4. Conclusions

In this paper, we propose an architecture called MSSFF that effectively combines spectral and spatial features for accurate hyperspectral semantic segmentation. MSSFF incorporates spectral and spatial feature aggregation modules within the encoder, allowing for the fusion of features and the generation of hierarchical representations. Additionally, in the deep layers of the encoder, we introduce a PPM for aggregating multi-scale semantic information. In the skip connection part, we employ an efficient Transformer to perform global modeling on deep feature maps, while utilizing a spatial attention mechanism for local feature extraction on shallow feature maps. Consequently, MSSFF exhibits strong capabilities in feature extraction as well as local–global modeling.

The performance of MSSFF was evaluated on three benchmark datasets, and it consistently outperformed other methods in terms of key evaluation metrics, including OA, AA, and Kappa. These results highlight the remarkable potential of MSSFF for hyperspectral semantic segmentation tasks, confirming its superiority over existing approaches.

Furthermore, we conducted an investigation into the impact of the number of layers in the encoder on the model's performance. Our analysis revealed that deeper models tend to yield better results, with the optimal performance achieved when the number of layers is set to four. In future research, we plan to explore the feasibility of shallow models for hyperspectral semantic segmentation and investigate the deployment of lightweight hyperspectral semantic segmentation models on resource-constrained devices.

Author Contributions: Conceptualization, methodology, software, Y.C., Q.Y. and W.H.; validation, Y.C. and Q.Y.; writing—original draft preparation, Y.C.; writing—review and editing, Q.Y. and W.H.; visualization, Y.C. and Q.Y.; supervision, Q.Y.; project administration, Q.Y.; funding acquisition, Q.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (42001362) to Q.Y.

Data Availability Statement: The datasets presented in this paper is available through https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes, accessed on 1 June 2023.

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

References

1. Shimoni, M.; Haelterman, R.; Perneel, C. Hypersectral Imaging for Military and Security Applications: Combining Myriad Processing and Sensing Techniques. *IEEE Geosci. Remote Sens. Mag.* **2019**, *7*, 101–117. [CrossRef]
2. Fei, B. Hyperspectral imaging in medical applications. In *Data Handling in Science and Technology*; Elsevier: Amsterdam, The Netherlands, 2019; Volume 32, pp. 523–565.
3. Liu, H.; Yu, T.; Hu, B.; Hou, X.; Zhang, Z.; Liu, X.; Liu, J.; Wang, X.; Zhong, J.; Tan, Z.; et al. Uav-borne hyperspectral imaging remote sensing system based on acousto-optic tunable filter for water quality monitoring. *Remote Sens.* **2021**, *13*, 4069. [CrossRef]
4. Feng, L.; Zhang, Z.; Ma, Y.; Sun, Y.; Du, Q.; Williams, P.; Drewry, J.; Luck, B. Multitask Learning of Alfalfa Nutritive Value From UAV-Based Hyperspectral Images. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 5506305. [CrossRef]
5. Li, Q.; Wang, Q.; Li, X. Exploring the relationship between 2D/3D convolution for hyperspectral image super-resolution. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 8693–8703. [CrossRef]
6. Paoletti, M.; Haut, J.; Plaza, J.; Plaza, A. Deep learning classifiers for hyperspectral imaging: A review. *ISPRS J. Photogramm. Remote Sens.* **2019**, *158*, 279–317. [CrossRef]
7. Zhao, W.; Du, S. Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 4544–4554. [CrossRef]
8. Jiang, J.; Ma, J.; Chen, C.; Wang, Z.; Cai, Z.; Wang, L. SuperPCA: A superpixelwise PCA approach for unsupervised feature extraction of hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4581–4593. [CrossRef]
9. Yan, Q.; Huang, W. Sea ice sensing from GNSS-R data using convolutional neural networks. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 1510–1514. [CrossRef]
10. Chen, Y.; Yan, Q.; Huang, W. MFTSC: A Semantically Constrained Method for Urban Building Height Estimation Using Multiple Source Images. *Remote Sens.* **2023**, *15*, 5552. [CrossRef]
11. Yan, Q.; Chen, Y.; Jin, S.; Liu, S.; Jia, Y.; Zhen, Y.; Chen, T.; Huang, W. Inland Water Mapping Based on GA-LinkNet from CyGNSS Data. *IEEE Geosci. Remote Sens. Lett.* **2022**, *20*, 1500305. [CrossRef]
12. Bharadiya, J.P. Leveraging Machine Learning for Enhanced Business Intelligence. *Int. J. Comput. Sci. Technol.* **2023**, *7*, 1–19.
13. Dhamo, H.; Navab, N.; Tombari, F. Object-driven multi-layer scene decomposition from a single image. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Repbulic of Korea, 27 October–2 November 2019; pp. 5369–5378.
14. Yu, S.; Jia, S.; Xu, C. Convolutional neural networks for hyperspectral image classification. *Neurocomputing* **2017**, *219*, 88–98. [CrossRef]

15. Chen, Y.; Jiang, H.; Li, C.; Jia, X.; Ghamisi, P. Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6232–6251. [CrossRef]
16. Ghaderizadeh, S.; Abbasi-Moghadam, D.; Sharifi, A.; Zhao, N.; Tariq, A. Hyperspectral Image Classification Using a Hybrid 3D-2D Convolutional Neural Networks. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 7570–7588. [CrossRef]
17. Hao, S.; Wang, W.; Salzmann, M. Geometry-Aware Deep Recurrent Neural Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 2448–2460. [CrossRef]
18. Li, J.; Zhao, X.; Li, Y.; Du, Q.; Xi, B.; Hu, J. Classification of Hyperspectral Imagery Using a New Fully Convolutional Neural Network. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 292–296. [CrossRef]
19. Hong, D.; Han, Z.; Yao, J.; Gao, L.; Zhang, B.; Plaza, A.; Chanussot, J. SpectralFormer: Rethinking Hyperspectral Image Classification With Transformers. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5518615. [CrossRef]
20. Chen, Y.; Liu, P.; Zhao, J.; Huang, K.; Yan, Q. Shallow-Guided Transformer for Semantic Segmentation of Hyperspectral Remote Sensing Imagery. *Remote Sens.* **2023**, *15*, 3366. [CrossRef]
21. Chen, Y.; Wang, B.; Yan, Q.; Huang, B.; Jia, T.; Xue, B. Hyperspectral Remote-Sensing Classification Combining Transformer and Multiscale Residual Mechanisms. *Laser Optoelectron. Prog.* **2023**, *60*, 1228002. [CrossRef]
22. Chen, Y.; Yan, Q. Vision Transformer is Required for Hyperspectral Semantic Segmentation. In Proceedings of the 2022 5th International Conference on Pattern Recognition and Artificial Intelligence (PRAI), Chengdu, China, 19–21 August 2022; pp. 36–40.
23. Qiao, X.; Roy, S.K.; Huang, W. Multiscale Neighborhood Attention Transformer With Optimized Spatial Pattern for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5523815. [CrossRef]
24. Yu, C.; Zhou, S.; Song, M.; Gong, B.; Zhao, E.; Chang, C.I. Unsupervised Hyperspectral Band Selection via Hybrid Graph Convolutional Network. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5530515. [CrossRef]
25. Shi, C.; Liao, Q.; Li, X.; Zhao, L.; Li, W. Graph Guided Transformer: An Image-Based Global Learning Framework for Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 5512505. [CrossRef]
26. Yu, H.; Xu, Z.; Zheng, K.; Hong, D.; Yang, H.; Song, M. MSTNet: A multilevel spectral–spatial transformer network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5532513. [CrossRef]
27. Zhu, Q.; Deng, W.; Zheng, Z.; Zhong, Y.; Guan, Q.; Lin, W.; Zhang, L.; Li, D. A spectral-spatial-dependent global learning framework for insufficient and imbalanced hyperspectral image classification. *IEEE Trans. Cybern.* **2021**, *52*, 11709–11723. [CrossRef] [PubMed]
28. Jia, K.; Liang, S.; Zhang, N.; Wei, X.; Gu, X.; Zhao, X.; Yao, Y.; Xie, X. Land cover classification of finer resolution remote sensing data integrating temporal features from time series coarser resolution data. *ISPRS J. Photogramm. Remote Sens.* **2014**, *93*, 49–55. [CrossRef]
29. Fauvel, M.; Tarabalka, Y.; Benediktsson, J.A.; Chanussot, J.; Tilton, J.C. Advances in spectral-spatial classification of hyperspectral images. *Proc. IEEE* **2012**, *101*, 652–675. [CrossRef]
30. Mehta, A.; Ashapure, A.; Dikshit, O. Segmentation-based classification of hyperspectral imagery using projected and correlation clustering techniques. *Geocarto Int.* **2016**, *31*, 1045–1057. [CrossRef]
31. Li, Y.; Zhang, H.; Shen, Q. Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network. *Remote Sens.* **2017**, *9*, 67. [CrossRef]
32. Chan, R.H.; Kan, K.K.; Nikolova, M.; Plemmons, R.J. A two-stage method for spectral–spatial classification of hyperspectral images. *J. Math. Imaging Vis.* **2020**, *62*, 790–807. [CrossRef]
33. Qiao, X.; Roy, S.K.; Huang, W. Rotation is All You Need: Cross Dimensional Residual Interaction for Hyperspectral Image Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *16*, 5387–5404. [CrossRef]
34. Liu, J.J.; Hou, Q.; Cheng, M.M.; Wang, C.; Feng, J. Improving convolutional networks with self-calibrated convolutions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10096–10105.
35. Li, J.; Wen, Y.; He, L. SCConv: Spatial and Channel Reconstruction Convolution for Feature Redundancy. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 6153–6162.
36. Ren, Q.; Tu, B.; Li, Q.; He, W.; Peng, Y. Multiscale adaptive convolution for hyperspectral image classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 5115–5130. [CrossRef]
37. Cai, W.; Ning, X.; Zhou, G.; Bai, X.; Jiang, Y.; Li, W.; Qian, P. A novel hyperspectral image classification model using bole convolution with three-direction attention mechanism: small sample and unbalanced learning. *IEEE Trans. Geosci. Remote Sens.* **2022**, *61*, 5500917. [CrossRef]
38. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016. pp. 770–778.
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. 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, 18–22 June 2018; pp. 7132–7141.
41. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual attention network for scene segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seoul, Repbulic of Korea, 27 October–2 November 2019; pp. 3146–3154.

42. Li, X.; Wang, W.; Hu, X.; Yang, J. Selective kernel networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seoul, Repbulic of Korea, 27 October–2 November 2019; pp. 510–519.
43. Wang, S.; Li, B.Z.; Khabsa, M.; Fang, H.; Ma, H. Linformer: Self-attention with linear complexity. *arXiv* **2020**, arXiv:2006.04768.
44. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 2021; pp. 10012–10022.
45. Jiang, Y.; Chang, S.; Wang, Z. Transgan: Two transformers can make one strong gan. *arXiv* **2021**, arXiv:2102.07074.
46. Li, K.; Wang, Y.; Zhang, J.; Gao, P.; Song, G.; Liu, Y.; Li, H.; Qiao, Y. Uniformer: Unifying convolution and self-attention for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 12581–12600. [CrossRef] [PubMed]
47. Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J.M.; Luo, P. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 12077–12090.
48. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 June 2017; pp. 2881–2890.
49. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587.
50. He, M.; Li, B.; Chen, H. Multi-scale 3D deep convolutional neural network for hyperspectral image classification. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3904–3908.
51. Roy, S.K.; Krishna, G.; Dubey, S.R.; Chaudhuri, B.B. HybridSN: Exploring 3-D–2-D CNN feature hierarchy for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2019**, *17*, 277–281. [CrossRef]
52. Roy, S.K.; Manna, S.; Song, T.; Bruzzone, L. Attention-based adaptive spectral–spatial kernel ResNet for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 7831–7843. [CrossRef]
53. 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 16 × 16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
54. Sun, L.; Zhao, G.; Zheng, Y.; Wu, Z. Spectral–spatial feature tokenization transformer for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–14. [CrossRef]
55. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the 18th International Conference, Munich, Germany, 5–9 October 2015; pp. 234–241.
56. Misra, D.; Nalamada, T.; Arasanipalai, A.U.; Hou, Q. Rotate to attend: Convolutional triplet attention module. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Virtual Conference, 5–9 January 2021; pp. 3139–3148.
57. Yan, H.; Zhang, E.; Wang, J.; Leng, C.; Basu, A.; Peng, J. Hybrid Conv-ViT Network for Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 5506105. [CrossRef]
58. Song, D.; Yang, C.; Wang, B.; Zhang, J.; Gao, H.; Tang, Y. SSRNet: A Lightweight Successive Spatial Rectified Network with Non-Central Positional Sampling Strategy for Hyperspectral Images Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5519115. [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.