remote sensing

MDPI

Article

AMHFN: Aggregation Multi-Hierarchical Feature Network for Hyperspectral Image Classification

Xiaofei Yang, Yuxiong Luo, Zhen Zhang *, Dong Tang, Zheng Zhou and Haojin Tang

School of Electronics and Communication Engineering, Guangzhou University, Guangzhou 510006, China; xiaofeiyang@gzhu.edu.cn (X.Y.); tangdong@gzhu.edu.cn (D.T.); zhouzheng@gzhu.edu.cn (Z.Z.); tanghaojin@gzhu.edu.cn (H.T.)
* Correspondence: zhangzhen@gzhu.edu.cn

Abstract: Deep learning methods like convolution neural networks (CNNs) and transformers are successfully applied in hyperspectral image (HSI) classification due to their ability to extract local contextual features and explore global dependencies, respectively. However, CNNs struggle in modeling long-term dependencies, and transformers may miss subtle spatial-spectral features. To address these challenges, this paper proposes an innovative hybrid HSI classification method aggregating hierarchical spatial-spectral features from a CNN and long pixel dependencies from a transformer. The proposed aggregation multi-hierarchical feature network (AMHFN) is designed to capture various hierarchical features and long dependencies from HSI, improving classification accuracy and efficiency. The proposed AMHFN consists of three key modules: (a) a Local-Pixel Embedding module (LPEM) for capturing prominent spatial-spectral features; (b) a Multi-Scale Convolutional Extraction (MSCE) module to capture multi-scale local spatial-spectral features and aggregate hierarchical local features; (c) a Multi-Scale Global Extraction (MSGE) module to explore multi-scale global dependencies and integrate multi-scale hierarchical global dependencies. Rigorous experiments on three public hyperspectral image (HSI) datasets demonstrated the superior performance of the proposed AMHFN method.

Keywords: deep learning; hyperspectral image classification; transformers; convolution neural network; feature fusion

Citation: Yang, X.; Luo, Y.; Zhang, Z.; Tang, D.; Zhou, Z.; Tang, H. AMHFN: Aggregation Multi-Hierarchical Feature Network for Hyperspectral Image Classification. *Remote Sens.* **2024**, *16*, 3412. https://doi.org/10.3390/rs16183412

Academic Editor: Akira Iwasaki

Received: 22 July 2024
Revised: 1 September 2024
Accepted: 2 September 2024
Published: 13 September 2024

1. Introduction

With the improvement of sensors, much more hyperspectral images are becoming available. HSI could offer abundant information to identify materials, as it records hundreds of bands on the electromagnetic spectrum of each pixel. Particularly, materials differ in their emission, reflection, and absorption of electromagnetic waves, making the identification and detection of different materials at a fine-grained level. The rich spectral information makes them indispensable in various fields, such as ecosystem measurement [1], mineral analysis [2,3], biomedical imaging [4], and precision agriculture [5].

In fact, HSI classification task aims to divide each pixel into the class labels. HSI classification methods could be divided into two categories: traditional HSI classification methods and deep learning-based HSI classification methods. In traditional HSI classification methods, researchers attempt to solve the HSI classification task by applying machine learning methods, such as K-Nearest Neighbors (KNNs) ([6,7]), Random Forests (RFs) ([8]), and Support Vector Machines (SVMs) ([9,10]). For instance, Li et al. [11] proposed a spectral band selection method to determine the optimal bands for subsequent feature learning by combining Markov Random Field (MRF) and spectral selection. However, it is important to highlight that these traditional HSI classification methods, which have been used for a long time in the field, are often faced with challenges in the process of manual feature extraction. This means that there is quite a bit of room for human error and subjectivity in the process. Additionally, they may also experience a failure in fully exploiting the rich and

complex spatial-spectral characteristics of different materials. This shortcoming can lead to inaccurate and unreliable results.

Deep learning-based HSI classification methods have been developed, showing strong feature extraction capability. Chen et al. [12] introduced an autoencoder method for pixel identification, while Hu et al. [13] designed a CNN-based method to capture local spatial features. Ran et al. [14] combined spectral band analysis with CNN. RNN has also been utilized due to its sequential data modeling capability, but RNN-based methods may not explore spatial information as well as CNNs, leading to poor classification. Mei et al. [15] established a five-layer CNN-based method, integrating spatial and spectral information; however, it explores them separately, resulting in insufficient use of spatial-spectral fusion features.

Three-dimensional CNN architectures beneficially extract fusion information from 4D tensors, enabling building models that exploit spatial-spectral fusion. Yang et al. [16] used two CNN branches capturing this and then combined them for fed to a fully connected layer extracting jointly spatial-spectral fusion features. Other methods like FCN Three-Stream and novel 3D-CNN were introduced to address HSI classification, comprising multiple 3D convolutional, pooling, and regularization layers, effectively capturing spatial-spectral fusion. However, the CNN-based types may struggle with capturing global HSI information.

Recently, transformer networks have been applied to computer vision tasks and have performed well [17,18], which is due to their ability to capture long-range dependence. For instance, Dosovitskiy et al. [19] first used transformers for image classification, introducing the vision image transformer (ViT) network. In the ViT model, input images are divided into nine patches and treated as a sequence of tokens with positional embeddings. These tokens are then fed into a series of transformer blocks to extract parameterized vectors. The transformer's key components are the self-attention mechanism and Multilayer Perception (MLP), which can gather spatial transformations and long-range dependencies. Unfortunately, the ViT model fails to utilize the 2D structure of images, which can decrease performance. To improve performance, local features from CNNs are used as input tokens to capture local spatial information. For example, Graham et al. [20] used convolution layers to extract local features, which are then fed into transformer blocks. However, these improved transformers do not fully integrate local features and global representations.

Inspired by the transformer model's sequential modeling capability, Dosovitskiy et al. [19] first introduced it into computer vision tasks as the Vision Transformer (ViT) for image classification. In ViT, input images are divided into blocks, positional information is added, and relationships between blocks are established. Inspired by this, He et al. [21] introduced ViT into hyperspectral image (HSI) classification as the SSF model, utilizing a CNN for local spatial feature capture and a transformer module for sequential spectral relationship capture. Mei et al. [22] proposed GAHT, combining a CNN and transformer to explore local relationships within spectral channels and construct a hierarchical transformer. However, these methods still have issues.

1. Most transformer-based methods explore global spatial dependencies, ignoring those in the spectral dimension. Existing transformer-based HSI classification methods struggle to capture long spectral dependencies, hindering performance improvements.
2. Now, most transformer-based methods may not be able to further refine the local feature during the training stage. This is mainly because transformers directly process the local spatial features through a multi-head self-attention mechanism, resulting in limiting the further exploitation of local features.

We present a new method known as Aggregation Multi-Hierarchical Feature Network (AMHFN) to tackle complex challenges in hyperspectral image classification. The AMHFN centers on two key modules: a Local-Pixel Embedding module (LPEM) and a Multi-Scale Convolutional Extraction (MSCE) module. The LPEM captures refined local features using a grouped convolution layer and a Batch Norm layer, while the MSCE utilizes multi-scale convolutional layers, an Efficient Channel Attention (ECA) layer [23], and an

Efficient Spatial Attention (ESA) layer to extract and re-weight local spatial-spectral features. The input HSI cube is projected into features that simultaneously possess global spectral information and refined local spatial information. These features then feed into a Multi-Scale Global Extraction (MSGE) module to capture and integrate global dependencies across both spatial and spectral dimensions. With this unique design, the proposed AMHFN excels at capturing global dependencies and exploring refined local features, significantly enhancing hyperspectral image classification performance. In this paper, our contributions could be summarized as follows:

1. We propose a novel hybrid hyperspectral image classification method, called Aggregation Multi-Hierarchical Feature Network (AMHFN), that captures and aggregates local hierarchical features and explores global dependencies of spectral information and prominent local spatial features.
2. We propose Local-Pixel Embedding module (LPEM) to exploit the refined local contextual spatial-spectral features. Specifically, the proposed LPEM consists of one grouped convolution layer to capture the hierarchical spatial-spectral features.
3. We further propose two modules to capture and aggregate the multi-scale hierarchical features. A Multi-Scale Convolutional Extraction (MSCE) module captures local spectral-spatial fusion information, while a Multi-Scale Global Extraction (MSGE) module captures and integrates global dependencies.
4. Finally, evaluated on three public HSI benchmarks, the proposed AMHFN outperforms other HSI classification methods.

The paper begins with a section delving into related work on HSI classification methods, followed by a section elaborating on the proposed AMHFN model. An experimental validation against three HSI datasets is presented in the next section. Concluding remarks and potential future work are offered in the final section.

The remainder of this paper is structured as follows: Section 2 delves into related work on HSI classification methods, Section 3 elaborates on the proposed AMHFN model, Section 4 presents thorough experimental validation against three HSI datasets, and Section 5 offers concluding remarks.

2. Related Works

2.1. HSI Classification Methods Based on CNNs

The superior local context modeling capability of convolutional neural networks (CNNs) has been a driving force behind the exploration of CNN-based methods for hyperspectral image (HSI) analysis. Slavkov et al. [24] introduced a CNN method for HSI classification, extracting spatial-spectral features from small neighborhoods. Xu et al. [25] developed a dual-channel CNN framework, capturing spectral-spatial features using 1D and 2D convolution. The two channels acquire spectral and spatial information, respectively, and then merge them through fully connected layers. Mei et al. [15] built a new CNN-based method named C-CNN for HSI classification, which could integrate the spatial background and spectral features using a five-layer CNN structure. Li et al. [26] treated the HSI input as a cube without any preprocessing or post-processing and utilized 3D convolution to simultaneously capture the local fusion features along in the spectral and spatial dimensions. However, the computational complexity of 3D convolution limits its application. To enhance this, Roy et al. [9] combined 2D and 3D convolutions in HybridSN, where 3D convolution focuses on spatial-spectral features and 2D convolution emphasizes more abstract spatial features.

Unlike CNNs, Recurrent Neural Networks (RNNs) are designed for sequential data and are utilized in the HSI domain to construct sequence models for processing adjacent spectra. Hang et al. [27] proposed a cascaded RNN model to eliminate redundant information between adjacent spectral bands. Mei et al. [28] proposed an HSI classification model combining CNNs and RNNs, where RNNs can learn spectral correlations within continuous spectra, while CNNs focus on salient features between adjacent pixels and spatial correlations. Additionally, several other backbone networks have been introduced to HSI,

including fully convolutional networks (FCNs) ([29,30]), generative adversarial networks (GANs) ([31,32]), CapsNet ([33]), and graph convolutional networks (GCNs) ([34]).

Current deep learning-based methods have some limitations in their architecture, although they have recorded success in HSI classification tasks. For example, RNNs-based methods may fail in exploring the global dependencies and extracting local contextual information, and CNN-based methods may fail in exploiting the global dependencies. Therefore, these HSI classification methods face a challenge in further improving the accuracy.

2.2. HSI Classification Methods Based on Transformers

Vaswani et al. [35] proposed the transformer architecture, using Multi-Head Self-Attention (MHSA) to model sequence relationships for NLP tasks. Inspired by this, Dosovitskiy et al. [19] introduced the transformer to computer vision, proposing the Vision Transformer (ViT) network. ViT divides the input image into blocks, transforms them into tokens, feeds them into MHSA to capture global dependencies, and then classifies the tokens through a fully connected layer. The ViT network excels in natural image classification tasks. Transformer-based methods are seeing growing use for hyperspectral image (HSI) classification [22,36], as evidenced by the development of models like SpectralFormer [37] and the Spectral–Spatial Feature Tokenization Transformer (SSFTT) [38]). These models incorporate novel components like the Groupwise Spectral Embedding (GSE) and Cross-layer Adaptive Fusion (CAF) modules, as well as a Gaussian distribution-weighted tokenization module, which work together to enhance the model's ability to learn localized spectral representations, facilitate efficient skip connections, and align deep semantic features with the sample's distribution. The use of these techniques allows for transformer-based models to outperform classical transformers, demonstrating their potential in the field of HSI classification.

2.3. HSI Classification Methods Based on Combining CNN and Transformer

To fully harness the distinct strengths of CNNs for spatial feature extraction and transformers for handling sequential features of any length, researchers have proposed various methods to combine these networks, aiming to enhance feature extraction capabilities for HSI classification tasks [39]. For instance, Tu et al. [40] proposed a hierarchical transformer architecture, termed local semantic feature aggregation-based transformer (LSFAT), for HSI classification, which consists of neighborhood aggregation-based attention (NAA) and neighborhood aggregation-based embedding (NAE) modules. Yang et al. [16] integrated CNN into a transformer to enhance performance and presented a novel transformer network, named hyperspectral image transformer (HiT) network, for HSI classification.

Although several networks have demonstrated promising classification performance, they often simply concatenate CNNs and transformers without fully leveraging their respective advantages. To address this, Ouyang et al. [41] incorporated convolution into the attention mechanism to capture global dependencies among tokens. They proposed HybridFormer, a transformer model that integrates spatial-spectral attention to emphasize the capture of both spectral and spatial dependencies. Similarly, Yang et al. [36] integrated convolution within the transformer framework and devised an adaptive 3D convolution projection module for shallow feature extraction.

Fixed receptive fields in the convolution projecting layer could limit the ability of transformer-based hyperspectral image (HSI) classification methods to explore and refine local spatial information. Most methods focus on global dependencies in the spatial dimension rather than the spatial-spectral dimension, potentially limiting their ability to capture hierarchical representations. This paper introduces a transformer-based HSI method, AMHFN, that aims to exploit multi-scale, hierarchical local and global features from HSI data, enhancing its capabilities.

3. Proposed Methodology

In this section, we give a brief introduction of the proposed AMHFN (as shown in Algorithm 1). As shown in Figure 1, it is a novel hybrid HSI classification method integrating the CNN and transformer. It consists of a "Stem" layer to extract the shallow features and three stages to capture the local and global multi-scale features. Specially, each stage comprises three key modules: a Local-Pixel Embedding module (LPEM) to retain the local spatial features, a Multi-Scale Convolutional Extraction (MSCE) module to capture the multi-scale hierarchical local spatial-spectral features, and a Multi-Scale Global Extraction (MSGE) module to explore the multi-scale hierarchical global dependencies. Because of these three key modules, the proposed AMHFN could model the spectral information and capture more refined multi-scale hierarchical local features.

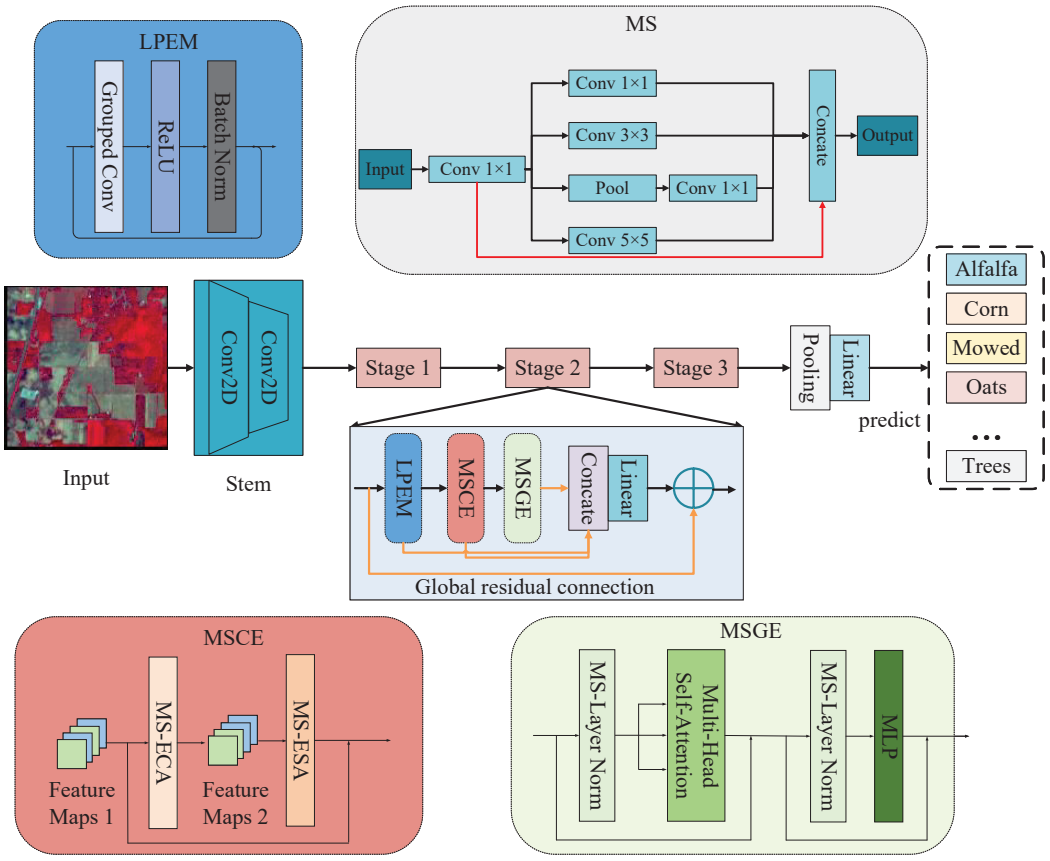


Figure 1. Overall framework of the proposed AMHFN. Specifically, the AMHFN comprises a stem layer to extract shallow features and three stages to capture the local and global spatial-spectral representations. The stem layer consists of two convolution operations to obtain the shallow local features. Each stage includes LPEM, MSCE, and MSGE to achieve the subtle spatial-spectral information. It is noted that MS is an abbreviation for multi-scale, MSCE is an abbreviation for multi-scale convolution extraction, and MSGE is an abbreviation for multi-scale global extraction.

Suppose $X \in \mathbb{R}^{P\times P\times C}$ is the input HSI, where P denotes the patch size and C is the number of channels. And $X_stem \in \mathbb{R}^{P\times P\times C_1}$ is the output of the "Stem" layer. Thus, X_stem can be obtained by

$$X_{stem} = Stem(X). \tag{1}$$

where the X_{stem} denotes the stem layer, which comprises two 2D convolutional layers to extract local features. The stem layer is used to extract features from HSI inputs, reduce the spectral-spatial dimensionality, and perform feature mapping.

And $X_1 \in \mathbb{R}^{P\times P\times C_1}$, $X_2 \in \mathbb{R}^{P\times P\times C_1}$, and $X_3 \in \mathbb{R}^{P\times P\times C_1}$ are the outputs of the first, second, and third stages, where C_1 denotes the channel number of each stage. All outputs from three layers are concatenated, and the features re-weighted using a linear operation. It is noted that the raw inputs are connected using a global residual connection layer. Finally, the outputs of each stage can be obtained by

$$X_{LPEM} = LPEM(X), \tag{2}$$

$$X_{MSCE} = MSCE(X_{LPEM}), \tag{3}$$

$$X_{MSGE} = MSGE(X_{MSCE}), \tag{4}$$

$$X_L = F_L < X_{LPEM}, X_{MSCE}, X_{MSGE} >, \tag{5}$$

$$X_i = X_L + X. \tag{6}$$

where $LPEM(\cdot)$ denotes the LPEM module, $MSCE(\cdot)$ denotes the MSCE module, and $MSGE(\cdot)$ denotes the MSGE module. F_L is the linear operation and $< \cdot >$ denotes the "concate" layer. After the "Stage 3" layer, the output features are fed into the "Pooling" layer to predict the raw pixel inputs.

In the following passage, we introduce the details of the proposed modules.

Algorithm 1 AMHFN Implementation Process.

Require: HIS image data $X \in \mathbb{R}^{H\times W\times C}$, label $Y \in \mathbb{R}^{H\times W}$, spatial size $s = 11$, training sample rate μ%.

Ensure: Classification map and four performance evaluation metrics.

1: Set batch size B to 64, optimizer Adam (learning rate: 1×10^{-3}), number of epochs E to 100.
2: Extract the input $X_{in} \in \mathbb{R}^{P\times P\times C}$ from X and divide it into a training dataset and test dataset.
3: **for** $i = 1$ to E **do**
4: Perform the "Stem" layer for shallow feature extraction.
5: **for** stage = 1 to 3 **do**
6: The input $x \in \mathbb{R}^{P\times P\times C_1}$; perform LPEM, MSCE, MSGE; and obtain X_{LPEM}, X_{MSCE}, X_{MSGE}, respectively
7: $x = \text{Linear}(\text{Concate}(X_{LPEM}, X_{MSCE}, X_{MSGE})).\text{transpose}(P, P, C_1) + x.$
8: Perform the "Pooling" layer and "Linear" layer to predict the result.
9: Use the softmax function to identify the labels.
10: Obtain the output by testing the trained model on the test dataset.

Local-Pixel Embedding module: The proposed $LPEM$ is a grouped convolutional operation used to capture deep spatial-spectral features from HSI. Specifically, a grouped convolution layer applies n kernels to the input, whose size is $X \in \mathbb{R}^{h\times w\times C/n}$. Following a grouped convolution, batch normalization and ReLU activation are applied.

$$X_{LPEM} = BN(ReLU(GroupedConv(X))), \tag{7}$$

where X_{LPEM} is the output of LPEM.

After extracting the deep spatial-spectral features, we utilize a linear operation to project the extracted features to the desired dimension.

$$X_{out} = F_{linear}(X_{LPEM}), \tag{8}$$

where F_{linear} is the linear operation. In this study, we adapt nn.Linear, which is a module provided by PyTorch that applies a linear transformation to the incoming data.

3.1. Multi-Scale Convolutional Layer

Figure 1 shows the Multi-Scale (MS) convolutional layers, divided into a convolution layer, a multi-scale convolution layer, and an aggregate layer. The convolution layer adjusts input channels, the multi-scale convolution layer has four layers with varying receptive fields, and the aggregate layer fuses features to generate the final output. This design captures local spatial-spectral information and aggregates multi-scale, hierarchical features. The MS module can be formulated as per Figure 1.

$$X_0 = Conv(X), \tag{9}$$

$$Output = Conv_{1\times1}(MS - Conv(X_0)), \tag{10}$$

where $X \in R^{C\times H\times W}$ denotes the inputs, and $MS - Conv$ denotes the multi-scale convolution layer.

X_O_i denotes the output of the i-th multi-scale convolution branch. It utilizes 1×1, 3×3, 5×5 convolution layers and an average pooling layer. The output is obtained using these layers and passing through an average pooling layer, as shown in the provided equation.

$$X\_O_1 = F_1(X_0), \tag{11}$$

$$X\_O_2 = F_1(F_{pool}(X_0)), \tag{12}$$

$$X\_O_3 = F_{3,3}((F_1(X_0))), \tag{13}$$

$$X\_O_4 = F_{5,5}((F_1(X_0))), \tag{14}$$

We fuse and re-weight the multi-scale features by applying a 1×1 convolutional layer to produce the final output X_O.

$$X\_O = Concat([X_0, X\_O_1, X\_O_2, X\_O_3, X\_O_4]), \tag{15}$$

$$X\_O = F_1(X\_O), \tag{16}$$

The MS not only captures multi-scale local contextual information, but also explores global dependence across the spectral dimension. It achieves adaptability in both spatial and spectral dimensions.

3.2. Multi-Scale Convolutional Extraction Module

The proposed $MSCE$ module, as shown in Figure 1, uses multi-scale convolutional layers for extracting local spatial-spectral features, ECA for capturing the refined spectral information, and ESA for enhancing and refining the spatial information.

3.2.1. ECA-Based Layer

The ECA (as shown in Figure 2) uses global average pooling on input features, followed by 1D convolution with kernel size k and a Sigmoid activation to obtain channel weights. k represents the involvement of k adjacent channels in inter-channel information interaction. The output from Sigmoid is recalculated by re-weighting channels.

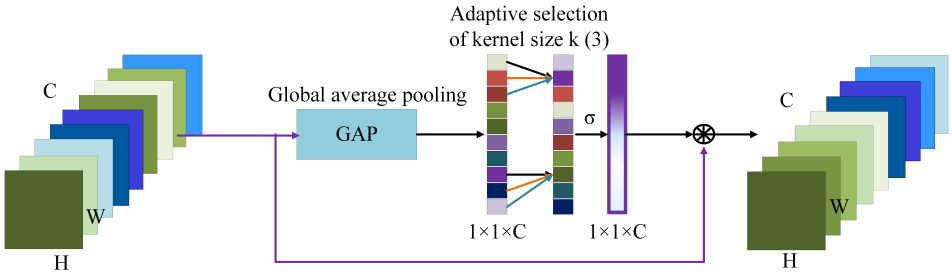


Figure 2. The structure of the ECA, where H, W, and C represent the height, width, and number of channels of the feature map, respectively. σ represents the operation of the Sigmoid activation function.

$$\begin{aligned} Attention &= \sigma(g(X)), \\ X_{out} &= Attention \cdot X. \end{aligned} \tag{17}$$

where $g(X) = \frac{1}{KK}\sum_{j=1,j=1}^{K,K} X_{ij}$ is channel-wise global average pooling (GAP). σ is a Sigmoid function.

3.2.2. ESA-Based Layer

Recently, some researchers [42] proposed Partial Convolution (PConv) and Efficient Spatial Attention (ESA) in the field of natural images, which can reduce computational redundancy and speed up operations.

Figure 3 illustrates the operational process of PConv. The original feature map $I \in \mathbb{R}^{h\times w\times C}$ is given, where h, w, and C represent the height, width, and number of channels of the original feature map, respectively. PConv utilizes conventional convolution operations on a select region of the original image, identified as a region feature map $i1 \in \mathbb{R}^{h\times w\times c}$, with "c" symbolizing the channels involved ($c < C$), to perform feature extraction. This ensures that both the spatial dimensions and the channel count of the output feature map $o1$ are congruent with the input region $i1$. Then, the ultima feature map, obtained by concatenating $o1$ with the non-convolved part ($i2 \in \mathbb{R}^{h\times w\times (C-c)}$), maintains the same spatial dimensions and channel numbers as the original image I. PConv delivers an efficient method for feature extraction by diminishing computational redundancy and memory requirements. Finally, PConv is formulated as follows:

$$\text{PConv} = \text{Concat}(o1, i2), \tag{18}$$

where Concat stands for concatenation in the channel dimension.

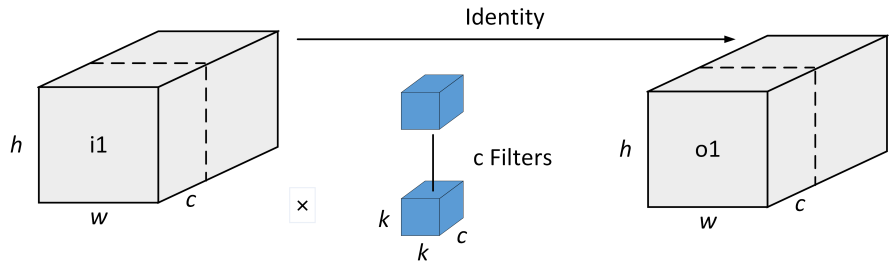


Figure 3. The operation process of Partial Convolution (PConv). "×" represents the operation of convolution.

As shown in Figure 4, the ESA is built upon one PConv layer and two PWconv layers. The PConv layer is used to capture local spatial information, and the PWconv layer is utilized to capture local features along with spatial-spectral dimension. Specifically, ESA balances low latency and feature diversity by connecting the two PWconv layers with

normalization and activation, instead of adding them after each convolution. Then, the combination of features extracted by PConv and not extracted by PConv occurs in each PWconv layer of ESA by increasing the feature map's dimensionality along the channel axis, then reducing it back to the initial channel dimension. Therefore, the output of ESA can be summarized as follows:

$$X_{out} = \text{PWconv}(\text{ReLU}(\text{BN}(\text{PWconv}(\text{PConv}(\text{X}))))), \tag{19}$$

where PWconv and PConv are the PWconv and PConv operations.

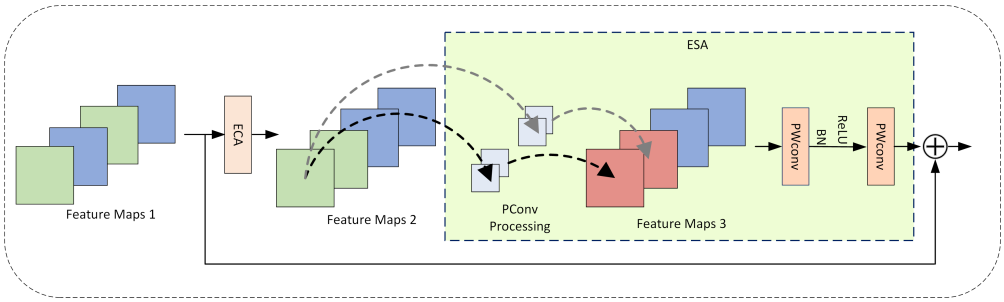


Figure 4. Structur of ESA mechanism.

Finally, ECA (Efficient Channel Attention) is used in hyperspectral image classification to capture refined spectral information by emphasizing important spectral channels, which helps in distinguishing subtle differences between spectral signatures. ESA (Enhanced Spatial Attention) focuses on enhancing and refining spatial information by giving attention to relevant spatial features, improving the ability to identify spatial patterns and structures within the image. Together, ECA and ESA effectively balance and enhance spectral and spatial information, leading to more accurate and detailed classification results. The proposed MSCE module is formulated as

$$X_{out} = ECA(X) + ESA(X). \tag{20}$$

3.3. Multi-Scale Global Extraction Module

The $MSCE$ could extract multi-scale local features but fails in exploring the global dependencies. To capture the global dependencies from HSIs, we design an $MSGE$ module to enhance the representation learning. Specifically, the $MSGE$ module integrates multi-head attention and MLP to effectively capture and refine complex graph relationships, enhancing the model's ability to learn rich and nuanced representations for improved performance of HSI classification tasks. The proposed $MSGE$ module uses multi-scale convolutional layers and a transformer, a self-attention mechanism (see Figure 5), to enhance performance. The transformer encoder incorporates multiple multi-head self-attention layers and a position-wise fully connected feed-forward network. The input and output of this module are a sequence of feature maps.

Self-attention (SA) is a mechanism enabling models to focus on relationships between different positions in a sequence. SA computes relationships between a query and a set of key–value pairs, generalizing the dot-product attention common in NLP tasks. SA can improve model performance by helping it focus on important relationships in input sequences.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_K}}\right)V, \tag{21}$$

where Q, K, and V are matrices, and d_K is the dimension of K.

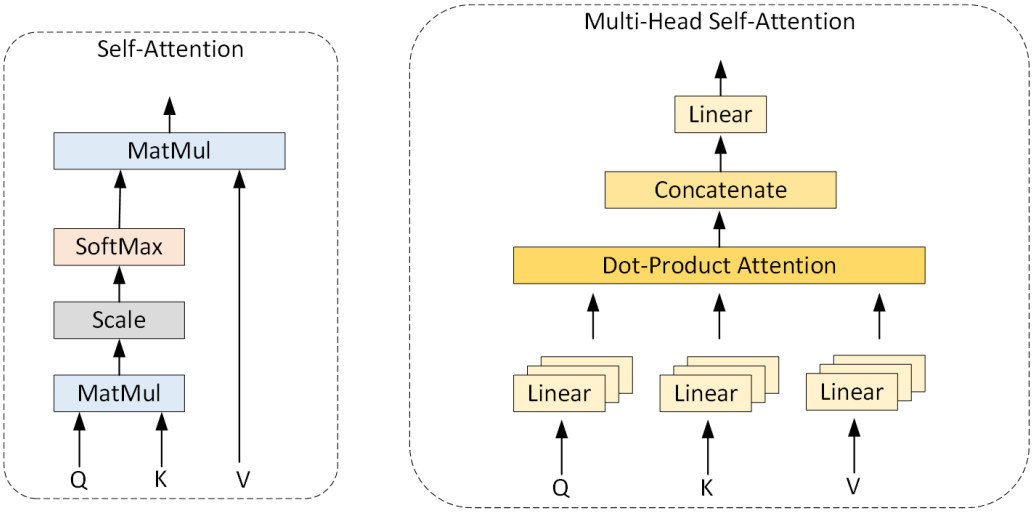


Figure 5. Structure of self-attention (SA) and $MHSA$ mechanisms.

Multi-head attention (MHSA) is a method that divides the input sequence into multiple sub-sequences and applies self-attention to each sub-sequence. It is a generalization of SA. Specifically, MHSA is a method that uses multiple SA mechanisms to explore different relationships in the input sequence. Each SA mechanism is denoted by SA_i, where i denotes the i-th head. The output of MHSA can be formulated as follows:

$$\mathrm{MHSA}(Q, K, V) = \mathrm{Concat}(SA_1, SA_2, \ldots, SA_h)W, \tag{22}$$

where h is the head number, W is the parameter matrix, $W \in \mathbb{R}^{d \times d}$, and $SA_1, SA_2, \ldots, SA_h \in \mathbb{R}^{n \times d/h}$.

The $MHSA$ input goes through a linear layer with non-linear activation, which reduces dimensionality and enhances the model's ability to learn nonlinear relationships.

The $MHSA$ output is passed through a linear layer with a non-linear activation function, and then divided into h chunks. Each chunk is fed into a separate SA mechanism. The h outputs are concatenated and multiplied by a weight matrix W. Then, the output passes through a final linear layer with a non-linear activation function to reduce dimensionality and enhance the model's ability to learn nonlinear relationships. The multiple consecutive transformer blocks can be formulated as above.

$$\begin{aligned} \widetilde{Y}_l &= \mathrm{MHSA}(\mathrm{LayerNorm}(Y_l)) + Y_l \\ Y_{l+1} &= \mathrm{MLP}(\mathrm{LayerNorm}(\widetilde{Y}_l)) + \widetilde{Y}_l \end{aligned} \tag{23}$$

where $\widetilde{Y}_l$ and Y_l denote the output features of the $MHSA$ module and MLP for block l.

4. Experiments

We selected three HSI datasets, including WHU-Hi-LongKou, Pavia University, and Houston 2013, to evaluate our proposed method. The experiments included parameter analysis, ablation experiments, and classification of results.

4.1. Datasets

4.1.1. WHU-Hi-LongKou Dataset

The WHL dataset was acquired from an 8-mm focal length Headwall Nano-Hyperspec imaging sensor mounted on a DJI Matrice 600 Pro UAV flying at 500 m altitude. The resulting imagery was 550 × 400 pixels with 270 bands from 400–1000 nm, at 0.463 m spatial resolution. The dataset contains 204,542 labeled samples across 9 land cover classes. In our experiments, we used 2% for training and 98% for testing, as shown in Table 1.

Table 1. Number of training and testing samples for the WHU-Hi-LongKou dataset.

Class No.	Class Name	Training	Testing
1	Corn	690	33,821
2	Cotton	167	8207
3	Sesame	61	2970
4	Broad-leaf soybean	1264	61,948
5	Narrow-leaf soybean	83	4068
6	Rice	237	11,617
7	Water	1341	65,715
8	Roads and houses	142	6982
9	Mixed weed	105	5124
Total		4090	200,452

4.1.2. Pavia University Dataset

The Pavia University (PU) dataset was acquired in 2001 using the ROSIS sensor. It covers 115 spectral bands from 380 nm to 860 nm. After discarding noisy bands, 103 bands remained for research. The dataset is an image with 610 × 340 pixels resolution. It contains 42,776 labeled samples across 9 land cover types. Only 5% of samples were used for training, while the remaining 95% are for testing. This split ensures rigorous model evaluation and comprehensive performance understanding, as shown in Table 2.

Table 2. Number of training and testing samples for the Pavia University dataset.

Class No.	Class Name	Training	Testing
1	Asphalt	332	6299
2	Meadows	932	17,717
3	Gravel	105	1994
4	Trees	153	2911
5	Painted metal sheets	67	1278
6	Bare Soil	251	4778
7	Bitumen	67	1263
8	Self-Blocking Bricks	184	3498
9	Shadows	47	900
Total		2138	40,638

4.1.3. Houston 2013 Dataset

The publicly available Houston 2013 (H2) dataset was collected using an Airborne Laser Mapping (ALM) system with a 2.5 µm wavelength laser. It was gathered during summer 2013 in Houston, Texas, USA, and initially used for the 2013 IEEE GRSS Data Fusion Competition. The dataset is an image with 949 × 1905. It was acquired from an airplane flying at 500 m between 12:30 and 16:30 on 18 June 2013 and covers 15 distinct land covers with 15,029 labeled samples. In our experiments, 10% of the samples were used for training and 90% for testing, as shown in Table 3.

4.2. Experimental Setup

4.2.1. Evaluation Indicators

In evaluating the proposed method's classification performance, we used three common indicators: Kappa coefficient (κ), overall accuracy, and average accuracy. The Kappa coefficient measures the agreement between two sets of data, with higher values indicating better agreement. The overall accuracy calculates the percentage of correct predictions, while the average accuracy determines the accuracy for each class. These metrics provide a comprehensive assessment of the method's performance, with higher values signifying better performance.

$$\kappa = \frac{P_o - P_e}{1 - P_e}, \tag{24}$$

where P_o and P_e are the observed and expected accuracies, respectively.

Table 3. Number of training and testing samples for the Houston 2013 dataset.

Class No.	Class Name	Training	Testing
1	Healthy Grass	125	1126
2	Stressed Grass	125	1129
3	Synthetic Grass	70	627
4	Trees	124	1120
5	Soil	124	1118
6	Water	33	292
7	Residential	127	1141
8	Commercial	124	1120
9	Road	125	1127
10	Highway	123	1104
11	Railway	123	1112
12	Parking Lot 1	123	123
13	Parking Lot 2	47	422
14	Tennise Court	43	385
15	Running Track	66	594
Total		1502	13,527

$$OA = \frac{\sum_{i=1}^{C} N_i \times A_i}{\sum_{i=1}^{C} N_i}, \tag{25}$$

where N_i is the number of samples of each class, and A_i is the accuracy of each class.

$$AA = \frac{\sum_{i=1}^{C} N_i \times A_i}{N} \tag{26}$$

where N is the total number of samples.

$$A_i = \frac{TP_i}{TP_i + FP_i} \tag{27}$$

where TP_i and FP_i are the true positive and false positive of each class, respectively.

4.2.2. Implementation Details

Experiments were conducted on the HSI dataset, containing 10 classes with 100 samples each. The dataset had 16 spectral bands, and images were 64 × 64 pixels. The experiments were run on an Intel(R) Xeon(R) Gold 6230R CPU and NVIDIA RTX A5000 GPU using the PyTorch deep learning framework. The Adam optimizer was used with an initial learning rate of 1×10^{-3}, a minibatch size of 64, and 100 epochs. These parameters remained consistent across all experiments.

4.2.3. Comparison with State-of-the-Art Backbone Methods

A range of cutting-edge classification networks based on CNN and transformer architectures were employed to validate our proposed method: 2D-CNN ([43]), 3D-CNN ([44]), HybridSn ([9]), ViT ([19]), PiT ([45]), HiT ([36]), GAHT ([22]). The 2D-CNN and 3D-CNN methods incorporate 2-D or 3-D convolutional layers, BN layers, activation functions, and linear layers. HybridSn combines 3-D and 2-D convolutional blocks, linear layers, and pooling layers. The ViT method uses a linear-projection component and transformer encoders. PiT includes four transformer encoder blocks, three pooling layers, and a linear-projection component. The HiT method combines a spectral-adaptive 3-D convolution projection (SACP) module and the Convolutional Permutator (Conv-Permutator) module.

The GAHT method uses a new Grouped Pixel Embedding Module to limit the Multi-head Self-Attention (MHSA) mechanism within a local spectral context, overcoming the issue of excessive dispersion in MHSA. Finally, the AMHFN method incorporates two Feature Hierarchical Blocks and a Retention Block to extract important and secondary feature information from the spectral space and learn long-range correlations between pixels and bands.

4.3. Ablation Studies

4.3.1. Ablation Study of the Input Patch Size

The proposed method is based on a spatial-spectral approach, where the patch size directly reflects the extent to which the central pixel can utilize spatial-spectral information from neighboring pixels. Hence, patch size plays a crucial role in determining AMHFN performance. The optimal patch size for different datasets is demonstrated using the AA. For WHL and PU, it is 7×7, and for H2 it is 11×11 (Table 4). This is perhaps because WHL and PU have denser pixel distributions, and smaller patches can fully utilize spatial spectral information in HSI. In contrast, H2 has a very sparse pixel distribution, requiring larger patches to acquire more sufficient information.

Table 4. Impact of different patch sizes for the AA on three datasets.

Patch Size	7×7	9×9	11×11	13×13	15×15
WHL	96.28	95.46	94.45	93.90	91.86
PU	97.58	96.93	96.25	95.52	94.61
H2	97.77	98.40	98.56	97.87	97.44

4.3.2. Ablation Study of the Kernel Size in the ECA Block

The proposed method utilizes an ECA block, a variant of the SE block. The ECA block contains two parts: a global normalization layer and a 1D convolution layer. Thus, the kernel size in the ECA block affects the proposed method's performance. We experimented to evaluate different kernel sizes' impact on AA, setting the kernel size to 1, 3, 9, and 15. It is noted that "1" indicates no inter-channel interaction and direct channel shuffle. Figure 6 illustrates different kernel size effects on AA across datasets. We first observe that a kernel size of 3 yielded the best AA on all HSI datasets. We also find varying decreases in AA with larger kernel sizes. This may result from introducing additional noise by expanding the channel interaction range, decreasing AA. Based on the kernel size analysis, we set the kernel size to 3 in the experiments.

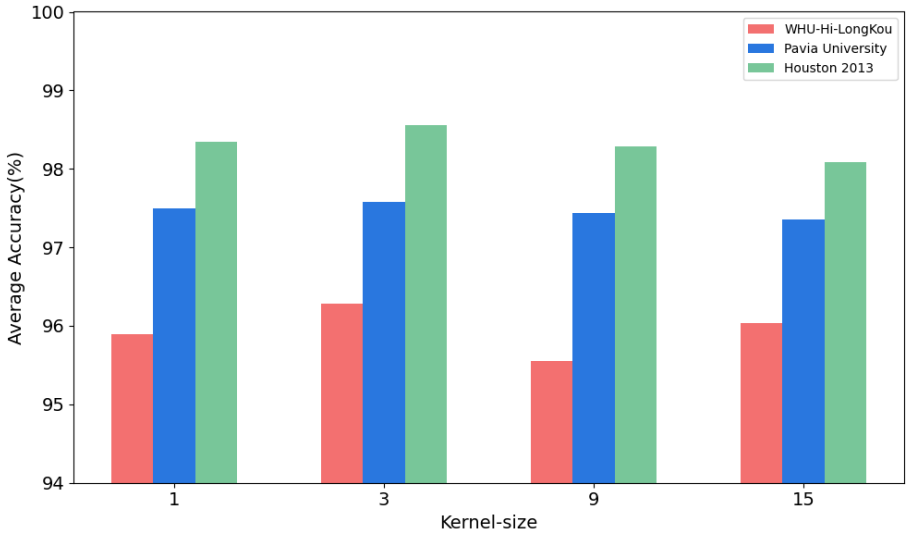


Figure 6. Impact of different kernel sizes for the AA on three datasets.

4.3.3. Ablation Study of the Proposed Multi-Feature Hierarchical Module

The proposed model AMHFN is built on the MSCE module, which is used to capture the prominent multi-scale local features and aggregate the subtle local contextual features. In this ablation study, we employed three modules, which divided the channels into three parts using Pconv to correspondingly separate feature information into prominent, moderate, and subtle parts. We compared the performance of the models to those with two components on three datasets using three performance metrics. Table 5 indicates that the results based on three modules are all inferior to the proposed AMHFN, where the performance of only ECA produces worse results than the baseline method. This may be attributed to the excessive fine-grained feature representation, impeding the model from fully capturing effective features, akin to the phenomenon of loss function overfitting.

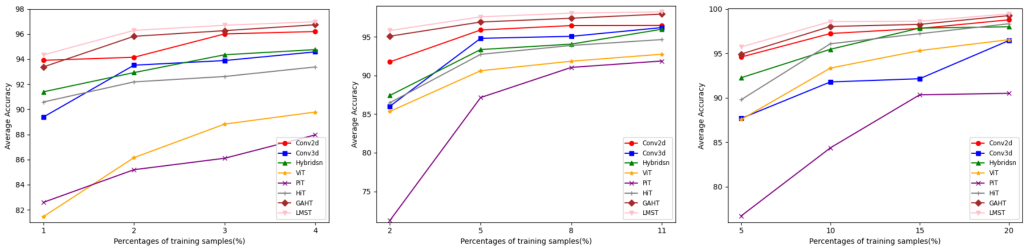
Table 5. Ablation study of the proposed multi-scale module on Houston 2013.

No.	ECA	ESA	κ	OA	AA
1	×	×	97.94	98.09	98.01
2	✓	×	97.50	97.69	98.06
3	×	✓	97.98	98.13	98.29
4	✓	✓	**98.32**	**98.45**	**98.71**

4.3.4. Ablation Study of the Numbers of the Training Samples

The robustness and stability of the proposed AMHFN were evaluated through a comprehensive set of experiments on various training samples. Different HSI datasets require different training sample percentages, ranging from 1–4% on the WHU-Hi-LongKou dataset, 5–20% on the Houston2013 dataset, and 1–7% on the Pavia University dataset.

The experimental results, meticulously depicted in Figure 7, offer illuminating insights. Most notably, a clear pattern of improvement emerges across all methodologies with increasing training samples. Consequently, deep learning methods, with intricate architectures, require substantive training data for optimal functionality. Of particular interest is the promising performance of the newly introduced AMHFN, which stands out by delivering superior results compared to well-established techniques, even maintaining the same training proportion. This significant observation highlights not only AMHFN's efficacy but also its resilience amid data-driven analysis demands.



(a) WHU-Hi-LongKou (b) Pavia University (c) Houston2013

Figure 7. AA of different models with different percentages of training samples on three datasets.

4.4. Classification Results

We comprehensively evaluated the proposed method and comparison methods on three HSI datasets. The experimental results in Tables 6–8 show performance metrics for each method, with optimal results in bold.

Table 6. Classification results of the WHU-Hi-LongKou dataset with 2% training samples.

Class No.	CNNs			Transformers					
	2D-CNN	3D-CNN	HybridSn	ViT	PiT	HiT	SSFTT	GAHT	AMHFN (Ours)
1	91.38	94.29	92.85	89.72	89.36	89.83	95.71	95.9	95.85
2	93.12	93.54	93.44	63.64	87.43	90.17	94.20	94.82	95.31
3	92.63	86.03	82.46	75.12	48.18	88.01	90.34	94.44	96.16
4	93.06	94.57	93.54	90.14	89.62	90.61	96.06	95.64	96.25
5	94.91	90.68	89.65	78.96	76.77	91.62	91.27	95.75	94.47
6	96.81	97.39	97.05	96.36	95.45	96.12	98.29	97.87	98.09
7	97.92	98.64	98.26	97.55	97.55	97.57	99.01	99.03	99.03
8	94.86	93.93	95.46	94.11	91.79	93.34	97.41	96.89	97.15
9	92.6	92.53	93.58	89.72	90.59	92.33	92.54	91.98	92.76
κ (%)	93.10	94.40	93.50	88.93	89.20	91.20	95.82	95.86	**96.17**
OA (%)	94.67	95.70	94.99	91.45	91.65	93.18	96.80	96.82	**97.07**
AA (%)	94.14	93.51	92.92	86.15	85.19	92.18	94.98	95.81	**96.12**

Table 7. Classification results of the Pavia University dataset with 1% training samples.

Class No.	CNNs			Transformers					
	2D-CNN	3D-CNN	HybridSn	ViT	PiT	HiT	SSFTT	GAHT	AMHFN (Ours)
1	92.40	83.43	85.32	85.53	85.91	82.27	87.81	94.36	93.95
2	91.58	88.69	86.23	79.99	82.95	85.65	92.33	92.82	93.07
3	55.15	39.51	60.64	51.40	13.47	43.79	76.23	81.81	84.70
4	96.27	93.54	95.32	86.22	63.40	87.54	93.93	95.02	96.04
5	99.70	91.22	98.27	98.57	98.57	98.80	100.00	99.55	99.70
6	94.50	69.29	80.32	73.55	31.37	67.78	79.78	92.89	91.75
7	69.17	59.15	60.67	51.48	20.65	60.14	72.89	77.68	78.13
8	84.44	55.67	67.22	77.56	49.19	76.84	89.16	80.05	90.29
9	99.89	94.45	100.00	99.25	78.76	98.72	94.13	98.72	100.00
κ (%)	86.62	73.81	78.07	73.12	57.75	74.35	85.39	88.83	**90.20**
OA (%)	89.73	79.97	83.04	79.05	68.09	80.26	88.88	91.46	**92.51**
AA (%)	87.01	74.99	81.55	78.17	58.25	77.95	87.36	90.32	**91.96**

The proposed AMHFN outperformed other methods on three datasets, primarily due to specialized modules capturing deep spatial-spectral features and enhancing spatial-spectral information. Interestingly, CNNs-based methods generally outperform transformer-based methods. Also, 3D-CNN outperforms 2D-CNN on the WHL dataset, possibly attributed to 3D convolution's advantage in extracting spectral information from 200 channels. However, 3D-CNN underperforms on the Houston2013 dataset, likely due to insufficient training samples. Surprisingly, transformer-based methods do not perform better than CNN-based ones. For example, ViT only achieves 88.93%, 91.45%, and 86.15% in terms of κ, OA, and AA. This might be because of their architecture specialized for natural images rather than spatial-spectral exploration. The PiT method's use of a pooling layer in the final part might lead to loss of critical feature information and inferior classification performance. However, HiT and GAHT are customized transformer models for HSI classification tasks, achieving satisfactory results compared to ViT and PiT. For instance, GAHT achieves outstanding OA and AA exceeding 98% on the Houston2013 dataset, demonstrating MHSA's effectiveness when confined to a local spatial-spectral context. Finally, our proposed AMHFN exhibited superior classification performance over other methods, with AA exceeding GAHT by 0.67% on the PU dataset.

Table 8. Classification results of the Houston 2013 dataset with 10% training samples.

Class No.	CNNs			Transformers					
	2D-CNN	3D-CNN	HybridSn	ViT	PiT	HiT	GAHT	SSFTT	AMHFN (Ours)
1	98.58	95.74	98.4	96.98	96.89	98.13	97.51	98.40	98.76
2	99.38	98.76	98.32	98.85	96.63	97.96	99.91	98.66	99.38
3	100	99.36	100	98.72	98.09	99.84	99.84	99.68	100
4	99.11	98.3	99.64	98.84	96.61	98.93	98.66	98.48	97.14
5	99.11	97.41	99.02	96.6	90.88	97.32	99.28	98.64	98.75
6	89.73	76.37	85.27	88.7	83.9	89.73	91.78	97.67	99.32
7	97.55	92.11	95	96.84	89.66	96.49	96.49	97.90	98.60
8	93.21	84.46	90.98	92.86	82.77	94.82	95.45	97.53	96.12
9	93.08	87.93	90.24	89.97	81.01	94.14	96.72	97.83	98.05
10	99.09	92.66	94.29	93.12	68.48	95.38	99.91	99.08	99.18
11	96.4	86.42	90.11	90.56	80.13	95.68	97.66	98.39	97.21
12	99.19	90.72	92.7	95.5	81.71	97.3	98.11	99.27	98.29
13	93.84	78.67	99.05	65.4	44.79	85.55	98.82	96.63	99.76
14	100	98.96	99.22	97.66	87.53	99.74	100	99.86	100
15	100	98.82	99.49	99.49	86.53	100	100	99.19	100
κ (%)	97.28	91.86	94.99	93.95	84.58	96.23	97.94	97.92	**98.32**
OA (%)	97.49	92.47	95.36	94.40	85.73	96.51	98.09	98.07	**98.45**
AA (%)	97.22	91.78	95.45	93.34	84.37	96.07	98.01	98.01	**98.71**

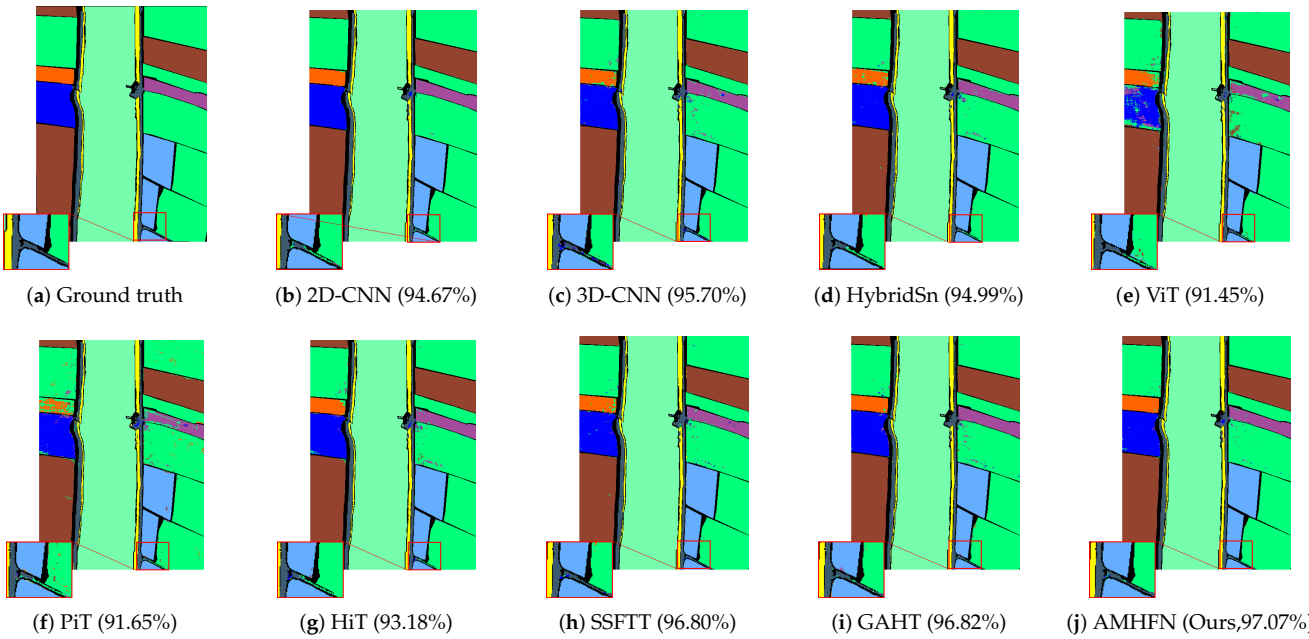
The Figures 8–10 show classification maps generated by various methods on different datasets. Methods using convolutional neural networks, specifically 2D-CNN, produce notably smooth maps with reduced salt and pepper noise, indicating enhanced classification accuracy for single, large ground features. Techniques using the transformer model are adept at capturing global dependencies in hyperspectral images (HSIs), yielding comparable results to 2D-CNN for HSI classification. The innovative AMHFN method excels at maximally harnessing hierarchical features and enhancing refined spatial-spectral information. It also demonstrates impressive capability to explore global dependencies from HSIs, achieving accurate and detailed classification maps.

Figure 11 shows t-SNE data distribution from the Houston 2013 Dataset, analyzed by six methods. Our novel method has impressively low inter-class confusion, precisely distinguishing between classes. There is minimal overlap between classes 1 and 2, highlighting the method's precision. In contrast, the GAHT method shows confusion, especially between classes 1 and 4. Other methods exhibit high confusion levels. However, our innovative method excels with superior clustering performance. It maintains vast inter-class distances while minimizing intra-class distances, enhancing data clustering and analytics. The visual analysis endorses our proposed methodology.

4.5. Discussion

From extensive experiments, we can find that the strengths of AMHFN are feature differentiation and hierarchical processing. By using LPEM and MSCE, AMHFN effectively differentiates between significant and subtle features, which is essential in dealing with the complex and redundant nature of HSI data. Meanwhile, the hierarchical structure allows for a more organized and detailed analysis of features, enhancing the model's ability to classify images accurately.

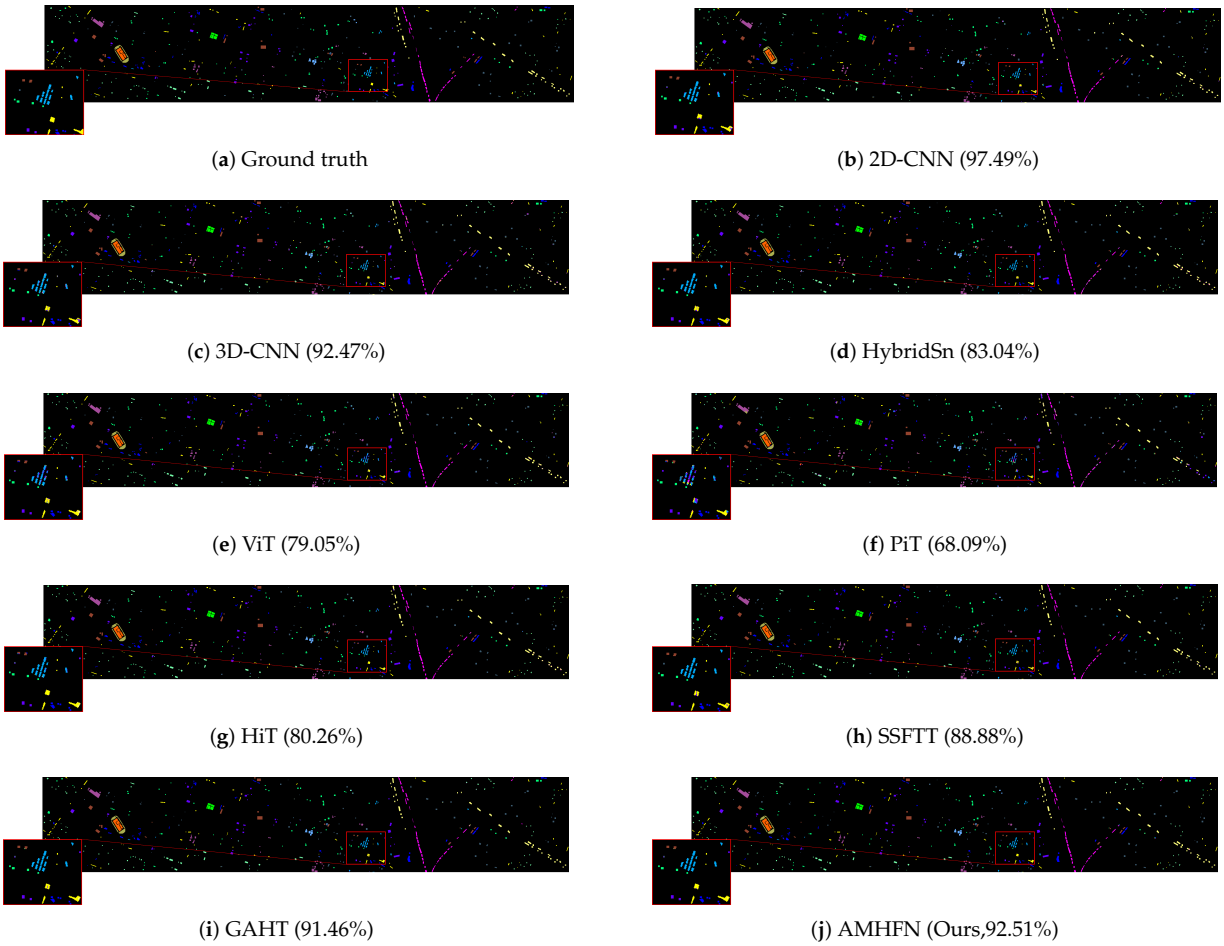
In summary, the AMHFN approach appears to be a sophisticated and well-validated method for hyperspectral image classification. By integrating techniques like LPEM and MSCE within a hierarchical framework, it addresses key challenges in feature differentiation and redundancy. The results from extensive testing support its efficacy and highlight its potential for practical applications in HSI analysis.



(**a**) Ground truth (**b**) 2D-CNN (94.67%) (**c**) 3D-CNN (95.70%) (**d**) HybridSn (94.99%) (**e**) ViT (91.45%)

(**f**) PiT (91.65%) (**g**) HiT (93.18%) (**h**) SSFTT (96.80%) (**i**) GAHT (96.82%) (**j**) AMHFN (Ours,97.07%)

Figure 8. Classification maps obtained using different methods on the WHU-Hi-LongKou dataset (with 2% training samples).



(**a**) Ground truth (**b**) 2D-CNN (97.49%)

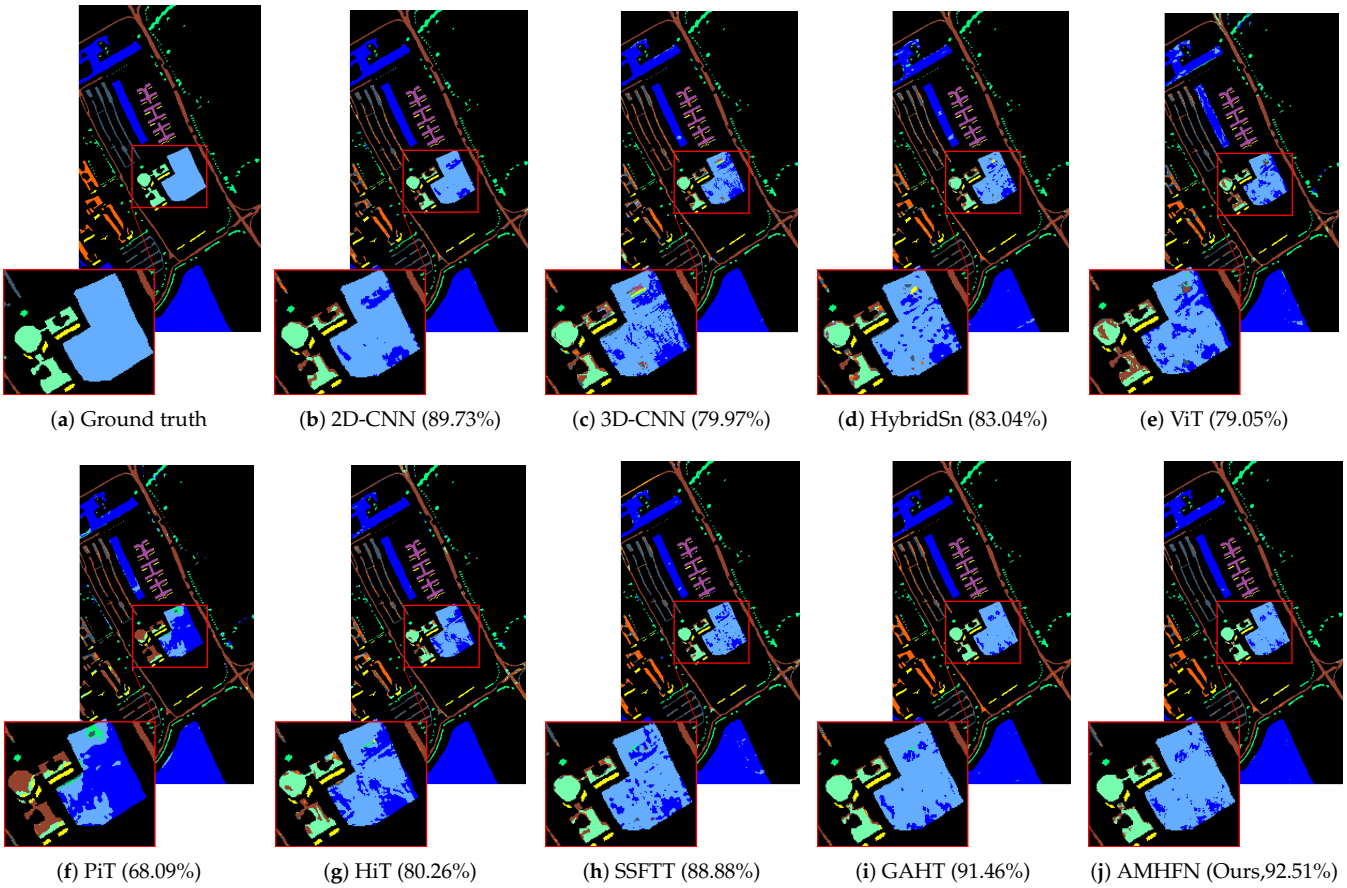
(**c**) 3D-CNN (92.47%) (**d**) HybridSn (83.04%)

(**e**) ViT (79.05%) (**f**) PiT (68.09%)

(**g**) HiT (80.26%) (**h**) SSFTT (88.88%)

(**i**) GAHT (91.46%) (**j**) AMHFN (Ours,92.51%)

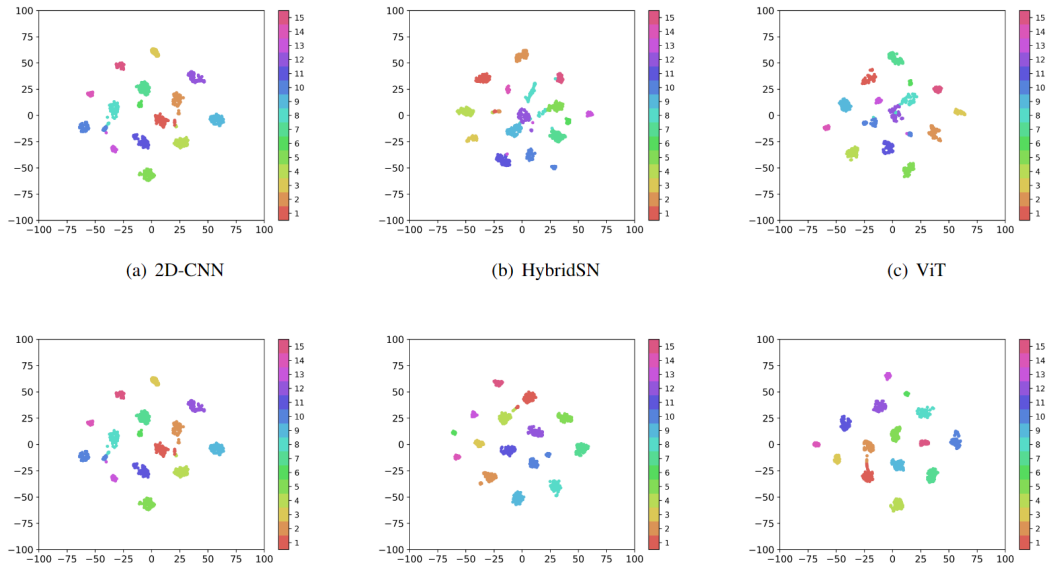
Figure 9. Classification maps obtained using different methods on the Houston 2013 dataset (with 10% training samples).



(**a**) Ground truth (**b**) 2D-CNN (89.73%) (**c**) 3D-CNN (79.97%) (**d**) HybridSn (83.04%) (**e**) ViT (79.05%)

(**f**) PiT (68.09%) (**g**) HiT (80.26%) (**h**) SSFTT (88.88%) (**i**) GAHT (91.46%) (**j**) AMHFN (Ours,92.51%)

Figure 10. Classification maps obtained by different methods on the Pavia University dataset (with 1% training samples).



(a) 2D-CNN (b) HybridSN (c) ViT

(d) HiT (e) GAHT (f) AMHFN (Ours)

Figure 11. Visualization of t-SNE data analysis on the Houston 2013 dataset.

5. Conclusions

In this paper, we present a new approach, which we call AMHFN, for the HSI classification task. The proposed AMHFN involves a gradual reduction in the channels of the feature maps, which is made possible by using a technique known as LPEM. The LPEM

makes the proposed AMHFN easier for the subsequent multi-scale to distinguish between significant features and more nuanced ones. The MSCE is especially adept at working with the abundant redundant information that is inherent in HSI, a process that involves differentiating feature information into two distinct categories: prominent and subtle aspects. Moreover, the strategic use of a hierarchical structure within the framework of our model significantly aids MSGE, which is based on the AMHFN algorithm. This is achieved by separating the two kinds of features and subsequently directing them to two distinct MSGEs. This process ensures that the more nuanced feature information does not get overlooked. The results of our extensive experiments and their subsequent analyses serve as a testament to the exceptional performance of our proposed model. This is true not only across multiple public HSI datasets but also in the broader context of HSI classification.

Future study will focus on improving the transformer architecture, such as transfer learning, and mutual learning with various networks (CNNs and transformers). Then, a standardized and universal method will be established for HSI classification based on transformers.

Author Contributions: Methodology, X.Y.; Writing—original draft, Y.L.; Writing—review & editing, Z.Z. (Zheng Zhou) and H.T.; Project administration, X.Y.; Funding acquisition, X.Y., Z.Z. (Zhen Zhang) and D.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the following projects: National Natural Science Foundation of China (NSFC) Fund under Grant 62301174. Guangzhou basic and applied basic research topics under Grant 2024A04J2081. Research Project of Guangzhou University under Grant 69-6239855.

Data Availability Statement: The Houston2013 dataset could be downloaded from at http://www.ehu.eus/ccwintco/uploads/6/67/Houston_2013.tar.gz, accessed on 1 September 2024.

Acknowledgments: This work was jointly supported by the following projects: National Natural Science Foundation of China (NSFC) Fund under Grant 62301174, Guangzhou basic and applied basic research topics under Grant 2024A04J2081, Research Project of Guangzhou University under Grant 69-6239855.

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

References

1. Hestir, E.L.; Brando, V.E.; Bresciani, M.; Giardino, C.; Matta, E.; Villa, P.; Dekker, A.G. Measuring freshwater aquatic ecosystems: The need for a hyperspectral global mapping satellite mission. *Remote Sens. Environ.* **2015**, *167*, 181–195. [CrossRef]
2. Sun, L.; Wu, F.; Zhan, T.; Liu, W.; Wang, J.; Jeon, B. Weighted Nonlocal Low-Rank Tensor Decomposition Method for Sparse Unmixing of Hyperspectral Images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 1174–1188. [CrossRef]
3. Wang, J.; Zhang, L.; Tong, Q.; Sun, X. The Spectral Crust project—Research on new mineral exploration technology. In Proceedings of the 2012 4th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), Shanghai, China, 4–7 June 2012; IEEE: Shanghai, China, 2012; pp. 1–4.
4. Noor, S.S.M.; Michael, K.; Marshall, S.; Ren, J.; Tschannerl, J.; Kao, F. The properties of the cornea based on hyperspectral imaging: Optical biomedical engineering perspective. In Proceedings of the 2016 International Conference on Systems, Signals and Image Processing (IWSSIP), Bratislava, Slovakia, 23–25 May 2016; IEEE: Bratislava, Slovakia 2016; pp. 1–4.
5. Gevaert, C.M.; Suomalainen, J.; Tang, J.; Kooistra, L. Generation of Spectral–Temporal Response Surfaces by Combining Multispectral Satellite and Hyperspectral UAV Imagery for Precision Agriculture Applications. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 3140–3146. [CrossRef]
6. Ma, L.; Crawford, M.M.; Tian, J. Local Manifold Learning-Based *k*-Nearest-Neighbor for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 4099–4109. [CrossRef]
7. Song, W.; Li, S.; Kang, X.; Huang, K. Hyperspectral image classification based on KNN sparse representation. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; IEEE: Beijing, China, 2016; pp. 2411–2414.
8. Ham, J.; Chen, Y.; Crawford, M.M.; Ghosh, J. Investigation of the random forest framework for classification of hyperspectral data. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 492–501. [CrossRef]
9. 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.* **2020**, *17*, 277–281. [CrossRef]
10. Pasolli, E.; Melgani, F.; Tuia, D.; Pacifici, F.; Emery, W.J. SVM active learning approach for image classification using spatial information. *IEEE Trans. Geosci. Remote Sens.* **2013**, *52*, 2217–2233. [CrossRef]

11. Li, S.; Jia, X.; Zhang, B. Superpixel-based Markov random field for classification of hyperspectral images. In Proceedings of the 2013 IEEE International Geoscience and Remote Sensing Symposium-IGARSS, Melbourne, Australia, 21–26 July 2013; IEEE: Melbourne, Australia 2013; pp. 3491–3494.
12. Chen, Y.; Lin, Z.; Zhao, X.; Wang, G.; Gu, Y. Deep Learning-Based Classification of Hyperspectral Data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2094–2107. [CrossRef]
13. Hu, W.; Huang, Y.; Wei, L.; Zhang, F.; Li, H. Deep convolutional neural networks for hyperspectral image classification. *J. Sens.* **2015**, *2015*, 258619. [CrossRef]
14. Ran, L.; Zhang, Y.; Wei, W.; Yang, T. Bands sensitive convolutional network for hyperspectral image classification. In Proceedings of the International Conference on Internet Multimedia Computing and Service, Xi'an, China, 19–21 August 2016; pp. 268–272.
15. Mei, S.; Ji, J.; Hou, J.; Li, X.; Du, Q. Learning sensor-specific spatial-spectral features of hyperspectral images via convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 4520–4533. [CrossRef]
16. Yang, J.; Zhao, Y.Q.; Chan, J.C.W. Learning and transferring deep joint spectral–Spatial features for hyperspectral classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 4729–4742. [CrossRef]
17. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jégou, H. Training data-efficient image transformers & distillation through attention. In Proceedings of the International Conference on Machine Learning, Online, 18–24 July 2021; PMLR: San Diego, CA, USA, 2021; pp. 10347–10357.
18. Jiang, Y.; Chang, S.; Wang, Z. Transgan: Two pure transformers can make one strong gan, and that can scale up. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 14745–14758.
19. 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* **2021**, arXiv:2010.11929.
20. Graham, B.; El-Nouby, A.; Touvron, H.; Stock, P.; Joulin, A.; Jégou, H.; Douze, M. Levit: A vision transformer in convnet's clothing for faster inference. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Online, 11–17 October 2021; pp. 12259–12269.
21. He, X.; Chen, Y.; Lin, Z. Spatial-Spectral Transformer for Hyperspectral Image Classification. *Remote Sens.* **2021**, *13*, 498. [CrossRef]
22. Mei, S.; Song, C.; Ma, M.; Xu, F. Hyperspectral image classification using group-aware hierarchical transformer. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–14. [CrossRef]
23. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11534–11542.
24. Slavkovikj, V.; Verstockt, S.; De Neve, W.; Van Hoecke, S.; Van de Walle, R. Hyperspectral image classification with convolutional neural networks. In Proceedings of the 23rd ACM International Conference on Multimedia, Brisbane, Australia, 26–30 October 2015; pp. 1159–1162.
25. Xu, X.; Li, W.; Ran, Q.; Du, Q.; Gao, L.; Zhang, B. Multisource remote sensing data classification based on convolutional neural network. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 937–949. [CrossRef]
26. Li, Y.; Zhang, H.; Shen, Q. Spectral–Spatial Classification of Hyperspectral Imagery with 3D Convolutional Neural Network. *Remote Sens.* **2017**, *9*, 67. [CrossRef]
27. Hang, R.; Liu, Q.; Hong, D.; Ghamisi, P. Cascaded recurrent neural networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 5384–5394. [CrossRef]
28. Mei, X.; Pan, E.; Ma, Y.; Dai, X.; Huang, J.; Fan, F.; Du, Q.; Zheng, H.; Ma, J. Spectral-spatial attention networks for hyperspectral image classification. *Remote Sens.* **2019**, *11*, 963. [CrossRef]
29. 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]
30. Maggiori, E.; Tarabalka, Y.; Charpiat, G.; Alliez, P. Fully convolutional neural networks for remote sensing image classification. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; pp. 5071–5074. [CrossRef]
31. Zhu, L.; Chen, Y.; Ghamisi, P.; Benediktsson, J.A. Generative adversarial networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 5046–5063. [CrossRef]
32. Zhan, Y.; Hu, D.; Wang, Y.; Yu, X. Semisupervised hyperspectral image classification based on generative adversarial networks. *IEEE Geosci. Remote Sens. Lett.* **2017**, *15*, 212–216. [CrossRef]
33. Paoletti, M.E.; Haut, J.M.; Fernandez-Beltran, R.; Plaza, J.; Plaza, A.; Li, J.; Pla, F. Capsule networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 2145–2160. [CrossRef]
34. Hong, D.; Gao, L.; Yao, J.; Zhang, B.; Plaza, A.; Chanussot, J. Graph convolutional networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 5966–5978. [CrossRef]
35. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 03762.
36. Yang, X.; Cao, W.; Lu, Y.; Zhou, Y. Hyperspectral image transformer classification networks. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5528715. [CrossRef]
37. 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.* **2021**, *60*, 1–15. [CrossRef]

38. Sun, L.; Zhao, G.; Zheng, Y.; Wu, Z. Spectral–spatial feature tokenization transformer for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5522214. [CrossRef]
39. Qi, W.; Huang, C.; Wang, Y.; Zhang, X.; Sun, W.; Zhang, L. Global–local 3-D convolutional transformer network for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 1–14. [CrossRef]
40. Tu, B.; Liao, X.; Li, Q.; Peng, Y.; Plaza, A. Local Semantic Feature Aggregation-Based Transformer for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**,*60*, 1–15. [CrossRef]
41. Ouyang, E.; Li, B.; Hu, W.; Zhang, G.; Zhao, L.; Wu, J. When Multigranularity Meets Spatial–Spectral Attention: A Hybrid Transformer for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 1–18. [CrossRef]
42. Chen, J.; Kao, S.h.; He, H.; Zhuo, W.; Wen, S.; Lee, C.H.; Chan, S.H.G. Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 12021–12031.
43. Liu, B.; Yu, X.; Zhang, P.; Tan, X.; Yu, A.; Xue, Z. A semi-supervised convolutional neural network for hyperspectral image classification. *Remote Sens. Lett.* **2017**, *8*, 839–848. [CrossRef]
44. Sharma, V.; Diba, A.; Tuytelaars, T.; Van Gool, L. Hyperspectral CNN for image classification & band selection, with application to face recognition. In *Technical Report KUL/ESAT/PSI/1604*; KU Leuven, ESAT: Leuven, Belgium, 2016.
45. Heo, B.; Yun, S.; Han, D.; Chun, S.; Choe, J.; Oh, S.J. Rethinking spatial dimensions of vision transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 11936–11945.