remote sensing

MDPI

Article

TransHSI: A Hybrid CNN-Transformer Method for Disjoint Sample-Based Hyperspectral Image Classification

Ping Zhang [1], Haiyang Yu [1,2,*], Pengao Li [1] and Ruili Wang [1]

[1] School of Surveying and Land Information Engineering, Henan Polytechnic University, Jiaozuo 454000, China; 311705000307@home.hpu.edu.cn (P.Z.); 212104020002@home.hpu.edu.cn (P.L.); 212104020072@home.hpu.edu.cn (R.W.)
[2] Key Laboratory of Mine Spatio-Temporal Information and Ecological Restoration, Henan Polytechnic University Ministry of Natural Resources, Jiaozuo 454000, China
* Correspondence: yuhaiyang@hpu.edu.cn

Abstract: Hyperspectral images' (HSIs) classification research has seen significant progress with the use of convolutional neural networks (CNNs) and Transformer blocks. However, these studies primarily incorporated Transformer blocks at the end of their network architectures. Due to significant differences between the spectral and spatial features in HSIs, the extraction of both global and local spectral–spatial features remains incomplete. To address this challenge, this paper introduces a novel method called TransHSI. This method incorporates a new spectral–spatial feature extraction module that leverages 3D CNNs to fuse Transformer to extract the local and global spectral features of HSIs, then combining 2D CNNs and Transformer to capture the local and global spatial features of HSIs comprehensively. Furthermore, a fusion module is proposed, which not only integrates the learned shallow and deep features of HSIs but also applies a semantic tokenizer to transform the fused features, enhancing the discriminative power of the features. This paper conducts experiments on three public datasets: Indian Pines, Pavia University, and Data Fusion Contest 2018. The training and test sets are selected based on a disjoint sampling strategy. We perform a comparative analysis with 11 traditional and advanced HSI classification algorithms. The experimental results demonstrate that the proposed method, TransHSI algorithm, achieves the highest overall accuracies and kappa coefficients, indicating a competitive performance.

Keywords: hyperspectral image (HSI) classification; spectral–spatial features; self-attention; convolutional neural network (CNN); Transformer

Citation: Zhang, P.; Yu, H.; Li, P.; Wang, R. TransHSI: A Hybrid CNN-Transformer Method for Disjoint Sample-Based Hyperspectral Image Classification. *Remote Sens.* **2023**, *15*, 5331. https://doi.org/10.3390/rs15225331

Academic Editors: Meiping Song, Silvia Liberata Ullo, Chunyan Yu, Yulei Wang, Weiying Xie, Enyu Zhao and Caixia Gao

Received: 13 September 2023
Revised: 10 November 2023
Accepted: 10 November 2023
Published: 12 November 2023

1. Introduction

The classification of HSIs is a crucial task in Earth observation missions. HSIs capture hundreds of bands per pixel in the spectral dimension, providing a vast amount of spectral–spatial information with a high spectral resolution [1]. HSI classification has enormous potential for various high-precision Earth observation applications, such as land cover identification [2], precision agriculture [3,4], change detection [5,6], environmental monitoring [7,8], and resource exploration [9].

1.1. Literature Review

The early approaches to HSI classification primarily relied on methods based on spectral features. These methods involved taking the spectral information from HSIs as input features, extracting features using classifier models, and ultimately performing a classification. Commonly used classifier models include K-nearest neighbor (KNN) [10], support vector machine (SVM) [11], random forest (RF) [12], and others. However, HSI acquisition is often affected by environmental factors, resulting in various forms of noise during the imaging process. Furthermore, the obtained spectral features may contain redundant information. As a result, many of the current research methods aim to address

these challenges by initially reducing the dimensionality of the data. Techniques such as principal component analysis (PCA) [13], singular value decomposition (SVD) [14], linear discriminant analysis (LDA) [15], and independent principal component analysis (ICA) [16] are commonly employed. To leverage the spectral–spatial information present in HSIs, researchers [17] have improved traditional classifier models. However, these manual HSI classification feature extraction methods require expert help and cannot represent a large amount of complex data.

In recent years, due to the powerful feature extraction and adaptive learning capabilities, deep learning techniques have been applied to HSI classification by many researchers. Table 1 lists some of the most recent state-of-the-art techniques, including CNNs, recurrent neural networks (RNNs), graph convolutional networks (GCNs), generative adversarial networks (GANs), and Transformer, and discusses their advantages and limitations.

Table 1. Strengths and limitations of recent state-of-the-art techniques for HSI classification.

Reference	Deep Learning Techniques	Strengths	Limitations
[18–22]	CNN	• CNN can capture local spectral–spatial features and effectively reduce the dimensionality of HSIs, automatically learning features within HSIs, without the need for manual feature extractors' design.	• The limited local receptive field, lack of consideration for spectral bands correlation, and failure to learn global contextual information.
[23–25]	RNN	• RNN can leverage spatial dependencies between non-adjacent patches and explore long-range spectral dependencies across different bands.	• RNN is unable to extract local spatial features from HSIs and is prone to gradient vanishing and exploding during training.
[26–29]	GCN	• GCN can effectively capture complex spatial relationships.	• A substantial amount of labeled data are needed to train the model.
[30–32]	GAN	• GAN can work effectively with a limited number of samples. The generator part can be used to learn useful feature representations.	• There are many issues during the training process, such as unstable convergence and gradient vanishing.
[33–35]	Transformer	• Transformer can learn long-range spectral–spatial information from HSIs and enables parallel computing, improving operational efficiency.	• Transformer cannot represent local semantic elements and cannot fully utilize spatial information.

As one of the mainstream backbone architectures, CNN demonstrates strong capabilities in extracting spatial structural information and local contextual information in HSIs through global parameter sharing and local connectivity. For instance, Hu et al. [18] used five convolutional layers for HSI classification. However, these approaches failed to sufficiently exploit the spatial features inherent to the data.

To better extract the spectral–spatial features of HSIs, researchers have introduced higher-dimensional convolutions. For example, Yang et al. [19] developed a two-branch network structure by combining 1D CNNs and 2D CNNs models. Chen et al. [20] introduced a method based on 3D CNNs combined with regularization to extract 3D features efficiently. Meanwhile, Zhong et al. [21] proposed a spectral–spatial residual network(SSRN) for HSI classification. This method leverages the information from the previous layer of features to supplement the latter layer of features, significantly improving the utilization rate of the features. Roy et al. [22] combined the characteristics of 3D CNNs and 2D CNNs to propose a hierarchical network structure. This method fully extracts spectral–spatial features, reduces computational complexity, and improves classification accuracy. Building upon the fusion

of 2D CNNs and 3D CNNs, Firat et al. [36] incorporated deep separable convolution to reduce network parameters effectively.

Most of the methods outlined above are based on CNN backbones and their variants, but they are often insufficient to detect subtle differences between spectral dimensions. As the mainstream backbone architecture, CNNs have shown a solid ability to extract spatial structure and local context information from HSIs, effectively improving classification performance. However, CNNs also struggle to capture sequence properties, mainly medium- and long-term dependencies. Therefore, when there are numerous categories for classification and the spectral features are highly similar, the classification performance tends to degrade [34]. The self-attention mechanism [37,38] may be a better choice, as it is not limited by distance and can focus on more valuable information, contributing to a more comprehensive capture of global contextual relationships.

Therefore, to better extract the spectral–spatial features of HSIs, researchers have introduced attention mechanisms. For example, Mei et al. [39] introduced the spectral–spatial attention networks (SSAN), which employ attention-based RNNs to extract essential features from continuous spectra and use attention-based CNNs to focus on significant spatial correlations between neighboring pixels. However, in real HSI scenarios, a large number of hyperspectral samples (or pixels) are typically present, and RNNs are not suitable for parallel training, reducing the efficiency of HSI classification.

The Transformer network [35] has been proposed to solve natural image classification tasks from a sequence data perspective. For example, Dosovitskiy et al. [40] applied Transformer to image classification tasks and proposed the Vision Transformer (ViT) network. This network utilizes a multi-head self-attention (MHSA) mechanism to efficiently process and analyze sequential data, allowing for the extraction of global information. He et al. [33] directly used the Transformer networks for HSI classification and proposed HSI-BERT, which has a global receiving domain and good generalization ability but only uses linear projection and does not consider local spatial context information. Therefore, several scholars [41,42] have combined CNNs and Transformer structures to jointly extract information on HSIs and take advantage of both benefits. For instance, He et al. [43] introduced the spatial–spectral Transformer (SST) model, which extracted spatial features using a network structure similar to that of VGGNet. The model further integrated a dense connection-enabled Transformer to capture sequential spectral relationships. Moreover, a classification task was accomplished by utilizing the multilayer perceptron (MLP). Sun et al. [44] developed the SSFTT model, which uses convolutional layers to extract shallow spectral and spatial features of HSIs and then introduces a Gaussian-weighted feature tokenizer into the ViT model for feature transformation, obtaining deep semantic information. In the studies mentioned above that integrate Transformer structures, it is customary to append Transformer modules solely at the end of the network architecture. However, due to substantial disparities between the spectral and spatial features in HSIs, the extraction of global and local spectral–spatial features remains incomplete.

To fully exploit the combined potential, scholars have studied the strategy of integrating CNNs and Transformer blocks. For example, Zhong et al. [45] introduced the spectral spatial transformation network (SSTN), which separately merged 2D CNNs and 3D CNNs into the self-attention mechanism. They constructed spatial attention and spectral association modules to surpass the limitations of convolutional kernels. However, the SSTN fails to further integrate the extracted local and global spectral–spatial features, making it challenging to capture the crucial information of the targets. Li et al. [46] proposed Next-ViT, a new generation of Vision Transformer that uses the next hybrid strategy (NHS) to stack next convolutional blocks (NCBs) and next transformer blocks (NTBs) in an effective hybrid paradigm to better capture local and global information, thereby improving the performance of various downstream tasks. However, this method has a relatively large number of parameters.

With the emergence of deep learning methods, the classification accuracy for commonly used HSI datasets has reached nearly 100% [36,47]. However, many of the above

methods suffer from a common issue: training and test sets are constructed using random sampling, with both training and test samples being drawn from the same image. The random selection of training samples overlaps neighboring test samples, leading to a relatively high correlation between them and overly optimistic classification results [48,49]. Recognizing this problem, researchers have proposed various separation sampling strategies to ensure the sampling of training and test sets from different regions [50,51]. For instance, Liang et al. [52] introduced a separate sampling strategy to control random sampling and minimize the overlap between training and test samples. Moreover, IEEE GRSS also provides disjoint training and test sets for HSI classification competitions.

1.2. Contribution

Inspired by these studies, we aim to leverage the advantages of the Transformer model based on the self-attention mechanism, that is, by capturing the spectral–spatial relationship of HSI sequences at a long distance and solving the problem of the limited ability of CNN methods to obtain deep semantic features. In this paper, considering the characteristics of HSI, Transformer Encode modules are introduced at both shallow and deep stages of spectral–spatial feature extraction in the network, enabling the comprehensive extraction of global and local spectral–spatial information in HSIs. A novel HSI classification method named TransHSI is proposed. The proposed method is introduced and evaluated using the disjoint training and test sets. The main contributions of this paper are summarized below:

(1) The TransHSI proposes a new spectral–spatial feature extraction module, in which the spectral feature extraction module combines 3D CNNs with different convolution kernel sizes and Transformer to extract the global and local spectral features of HSIs. In addition, the spatial feature extraction module combines 2D CNNs and Transformer to extract the global and local spatial features of HSIs. The module mentioned above thoroughly considers the disparities between spectral and spatial characteristics in HSIs, facilitating the comprehensive extraction of both the global and local spectral–spatial features in HSIs.

(2) A fusion module is proposed, which first cascades the extracted spectral–spatial features and the original HSIs after dimensionality reduction and captures relevant features from different stages of the network. Secondly, a semantic tokenizer is used to transform the features to enhance the discriminant ability of features. Finally, the features are represented and learned in the Transformer Encode module to fully utilize the image's shallow and deep features to achieve an efficient fusion classification of spectral–spatial features.

(3) In this paper, the effectiveness of TransHSI is verified using three publicly available datasets, and competitive results are obtained. Crop classification is assessed using the Indian Pines dataset, and urban land cover classification is assessed using the Pavia University dataset and the Data Fusion Contest 2018. These results provide a reference for future research focused on HSI classification.

The remainder of this paper is organized as follows. Section 2 describes in detail the architecture of the proposed classification method. Section 3 presents the experimental datasets used to evaluate the performance of TransHSI and the experimental setup. Section 4 compares the classification performance of TransHSI with that of the other methods. Section 5 discusses the validity of TransHSI structures, feature visualization, and the impact of training set proportions on classification accuracy. Section 6 summarizes the paper and looks toward future work.

2. Materials and Methods

2.1. CNNs

In 1D CNNs, the convolutional kernels slide primarily along the spectral dimension to extract spectral features in HSIs. On 2D CNNs, the kernels move in both the height and width dimensions of the image to capture spatial features. In the case of 3D CNNs, the kernels slide in three directions: height, width, and spectral dimensions. When performing

HSI classification tasks, this implies that using 3D CNNs allows for a more comprehensive utilization of the information embedded in HSIs. However, it is worth noting that 3D CNNs come with the largest parameter count, and to balance performance and computational complexity, this study employs both 2D CNNs and 3D CNNs to extract local features in the pixel spatial and spectral dimension. Within the neural network, the value $v_{i,j}^{\alpha,\beta,\gamma}$ at position (α, β, γ) on the jth feature cube in the ith layer can be represented using the following formula:

$$v_{i,j}^{\alpha,\beta,\gamma} = g\left(\sum_{k} \sum_{h=0}^{H_i-1} \sum_{w=0}^{W_i-1} \sum_{r=0}^{R_i-1} \omega_{i,j,k}^{h',w',r'} v_{i-1,k}^{\alpha+h',\beta+w',\gamma+r'} + b_{i,j}\right) \quad (1)$$

where H_i, W_i, R_i represent the width, height, and number of channels of the convolution kernel, respectively. $\omega_{i,j,k}^{h',w',r'}$ is the weight parameter of position (h', w', r') connected to the kth feature cube, and $b_{i,j}$ is the bias. The function $g(\cdot)$ denotes the activation function.

2.2. Transformer Encode

The Transformer Encode module, shown in Figure 1b, comprises an MHSA layer, an MLP layer, two normalization layers, and two residual connection structures. The MHSA possesses the ability to capture long-range dependencies and adaptive spatial clustering. It comprises multiple self-attention layers stacked and integrated, enabling parallel computations and improving operational efficiency. Compared to a standard CNN, the Transformer Encode module incorporates advanced components such as layer normalization (LN) and the MLP. These components enhance the functionality and expressive power of the model. To learn multiple meanings, three learnable weight matrices W_q, W_k, W_v are used to linearly map the flattened results, such as X_{flat1} into a 3D invariant matrix comprising queries Q, keys K, and values V. The attention score is calculated using all Q and K, and the weight of the score is calculated using the softmax function. In summary, the self-attention layer is expressed as in the following formula.

$$Z = \text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (2)$$

where Z represents the output from the attention layer and d represents the dimension of the keys K. Several different self-attention layers are combined into a single MHSA layer using Equation (2). The following formula expresses the MHSA layer.

$$\text{MHSA} = \text{Concat}(Z_1, Z_2, \cdots, Z_h)W \quad (3)$$

where h is the head number of MHSA, $h = 8$ in this paper, $W \in R^{h \times d \times t}$ is the matrix parameter, and t is the number of tokens.

Following the MHSA and MLP layers, there is an LN layer and a residual connection, respectively. The role of the LN layer and the residual connection is to enhance the model's stability, accelerate training convergence speed, and assist the network to better capture features and information. The MLP layer comprises two linear layers with a nonlinear activation function called the Gaussian Error Linear Unit (GELU) inserted between them. This activation function introduces the idea of random regularity in activation, thereby reducing gradient vanishing problems and enabling faster training.

2.3. Proposed Methodology

As shown in Figure 2, the classification method mainly includes three modules: the HSI pretreatment module, the spectral–spatial feature extraction module, and the fusion module, followed by classification through two linear layers. The spectral–spatial feature extraction module consists of a spectral feature extraction module and a spatial feature

extraction module, both of which are composed of CNNs and Transformer components. This subsection of the paper provides a detailed description of the TransHSI method.

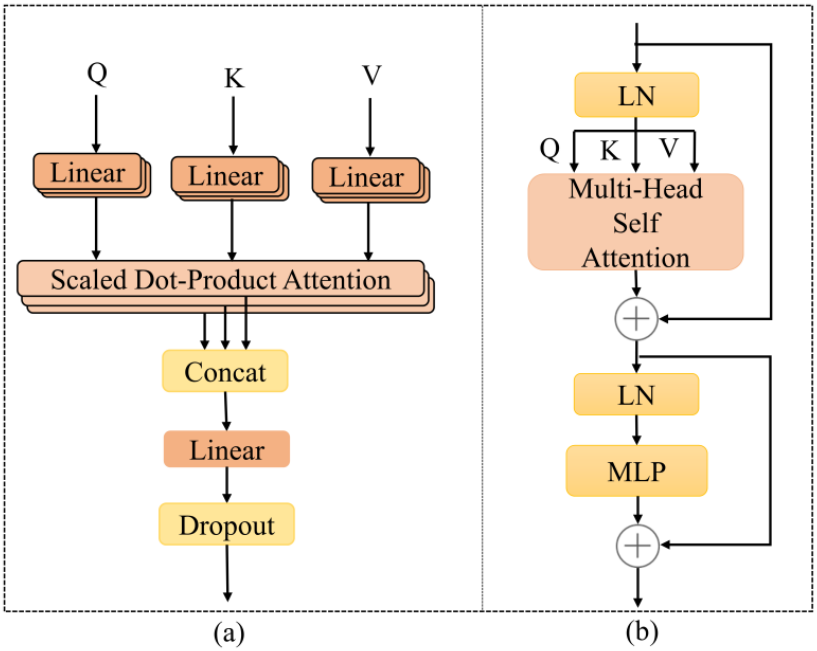


Figure 1. (**a**) The MHSA of Transformer Encode; (**b**) Transformer Encode.

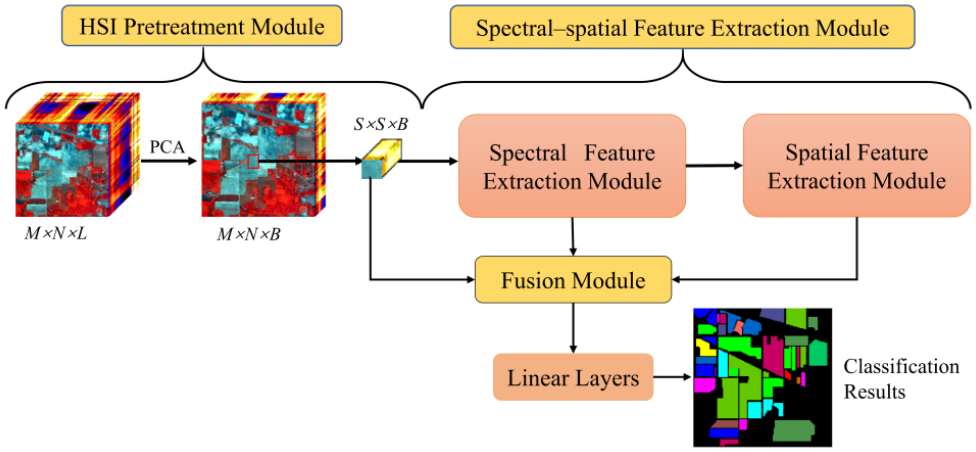


Figure 2. TransHSI classification network framework.

2.3.1. HSI Pretreatment Module

The original HSI data $I \in R^{M \times N \times L}$ are given, where M is the height, N is the width, and L is the number of spectral bands. Each pixel in I has L spectral bands, forming a single class vector $Y = (y_1, y_2, \cdots, y_C) \in R^{1 \times 1 \times C}$, where C is the number of land cover categories. To reduce the spectral dimensions and computational complexity while eliminating redundant information, HSIs are processed using PCA. The HSI after dimensionality reduction is expressed as $I_{pca} \in R^{M \times N \times B}$, where B is the number of spectral bands after PCA processing.

Then, 3D patches of the extraction of HSI data I_{pca} allow for the complete learning of spectral–spatial information around each pixel. The 3D patches $X \in R^{S \times S \times B}$ are created from I_{pca}, for which $S \times S$ represents the window size. The true label of each patch is determined by the label of the center pixel. A fill operation is performed on edge pixels when extracting a 3D patch around a single pixel. The width of the fill is $(S-1)/2$. In each iteration of training, the order of the training dataset is randomly shuffled to ensure the

diversity of training data, improve the generalization ability of the model, reduce the risk of overfitting, and increase the robustness of the model.

2.3.2. Spectral Feature Extraction Module

The spectral feature extraction module is shown in Figure 3a; the 3D patch $X \in R^{S\times S\times B}$ is initially processed through a 3D CNN with a kernel of $3 \times 3 \times 3$ and a step size of 1. In this paper, to ensure that the size of the input and output data remains unchanged, padding is applied to the input data using padding = (kernel_size $-$ 1)/2. The kernel_size represents the size of the convolutional kernel. Here, the 3D patch remains unchanged by utilizing the (1, 1, 1) padding strategy. The output of the 3D CNN is normalized using the batch normalization (BN) layer to obtain X_{3D1}. Then, X_{3D1} enters two 3D CNNs with a kernel of $5 \times 3 \times 3$ and a kernel of $7 \times 3 \times 3$, and the output is X_{3D2}. X_{3D1} is connected with the X_{3D2} residuals to obtain X'_{3D2}. The two 3D CNNs and residual connection structure are iterated again, and the result is X''_{3D2}. The utilization of multiple 3D CNNs is intended to thoroughly extract the local spectral information of 3D patches. The iteration result is input into a 2D CNN with a 1×1 kernel and 64 output channels to simplify the model and reduce the number of channels. Subsequently, the output of the 2D CNN is flattened into S^2 tokens. Finally, the tokens are input into the Transformer Encode module, and the output is denoted as X_{trans1}. The flattened result X_{flat1} is connected with X_{trans1} residuals to obtain the output of the spectral feature extraction module, marked as X_{spe}. The Transformer Encode module enhances the learning of spectral band sequence properties, compensating for the underutilization of spectral information. The above process is expressed as in the following formulas.

$$X_{3D1} = \mathrm{BN}(Conv3D_1(X)) \tag{4}$$

$$X_{3D2} = Conv3D_2(X_{3D1}) \tag{5}$$

$$X'_{3D2} = X_{3D1} + X_{3D2} \tag{6}$$

$$X_{flat1} = \mathrm{Flatten}\left(Conv2D_1\left(X''_{3D2}\right)\right) \tag{7}$$

$$X_{trans1} = \mathrm{Trans}\left(X_{flat1}\right) \tag{8}$$

$$X_{spe} = X_{flat1} + X_{trans1} \tag{9}$$

where $Conv3D$ denotes the 3D CNNs, $Conv2D$ denotes the 2D CNNs, $\mathrm{Flatten}(\cdot)$ denotes flattening and $\mathrm{Trans}(\cdot)$ denotes processing in the Transformer Encode module.

In the TransHSI network, there is a BN layer following the first 3D CNN, and after each subsequent convolutional layer, there is also a rectified linear unit (ReLU) layer and a dropout layer. The BN layer addresses the overfitting problem while speeding up network training and convergence. The ReLU layer adds nonlinearity to the model, thus improving its expressive power. The dropout layer enhances feature differentiation by increasing sparsity and helps to prevent overfitting. In this paper, the formula expression and illustration have been simplified, and the BN, ReLU, and dropout layers after the convolutional layer are not shown.

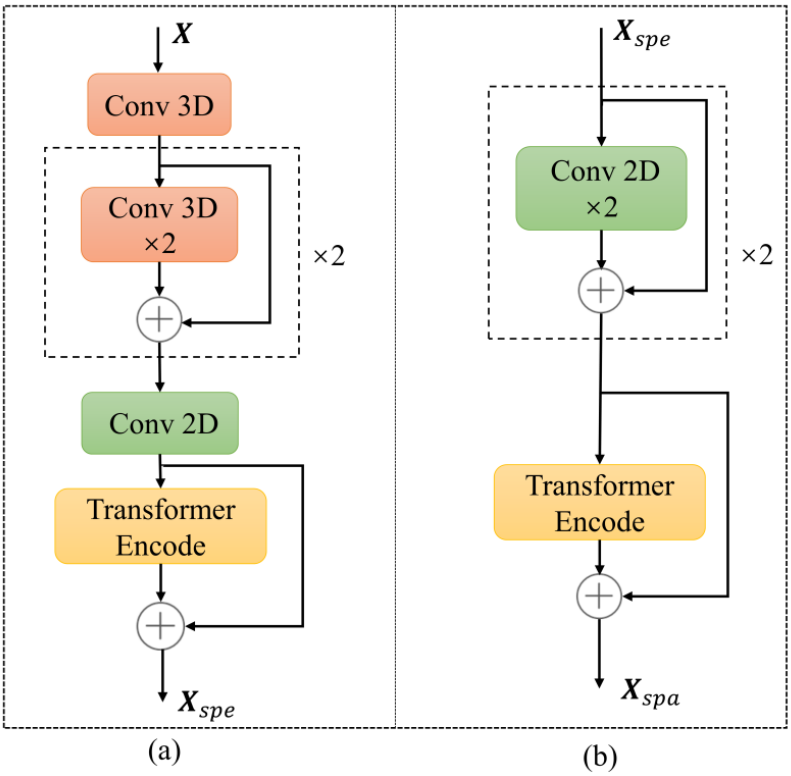


Figure 3. Spectral–spatial feature extraction module. (**a**) The spectral feature extraction module; (**b**) The spatial feature extraction module.

2.3.3. Spatial Feature Extraction Module

As shown in Figure 3b, the spatial feature extraction module initially employs 2D CNNs to capture local spatial information and subsequently utilizes the Transformer Encode module to learn long-range dependencies of pixels, effectively exploring the spatial information within HSIs. The module takes X_{spe} as the input and passes it through two 2D CNNs with 3×3 kernels, yielding an output denoted as X_{2D2}. Next, similar to the two 3D CNNs in the spectral feature extraction module, the two 2D CNNs undergo residual connections and iterative processing before being fed into the spectral feature extraction module. The output of the spatial feature extraction module is recorded as X_{spa}. The spatial feature extraction module is represented as in the following formulas.

$$X_{2D2} = Conv2D_2(X_{spe}) \tag{10}$$

$$X'_{2D2} = X_{spe} + X_{2D2} \tag{11}$$

$$X_{flat2} = \text{Flatten}(X''_{2D2}) \tag{12}$$

$$X_{trans2} = \text{Trans}\left(X_{flat2}\right) \tag{13}$$

$$X_{spa} = X_{flat2} + X_{trans2} \tag{14}$$

where X'_{2D2} is the result of connecting the X_{2D2} and X_{spe} residuals, and X''_{2D2} is the result of iterating over two 2D CNNs and the residual connection structure.

2.3.4. Fusion Module

The fusion module is shown in Figure 4, which includes a cascade layer, a 2D CNN, a tokenizer module, and a Transformer Encode module.

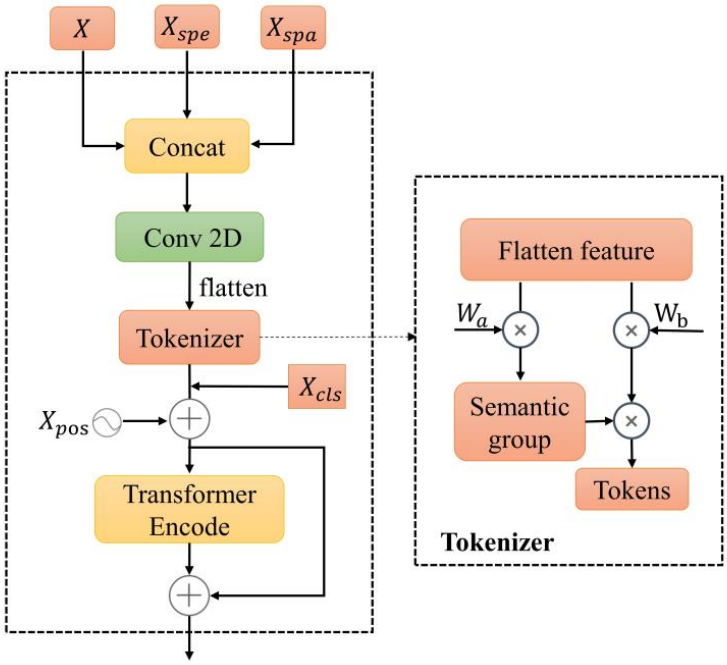


Figure 4. Fusion module.

The cascade layer plays a role in fusing the shallow and deep spectral–spatial features of HSIs. Specifically, it combines the original 3D patches X, the output X_{spe} of the spectral feature extraction module, and the output X_{spa} of the spatial feature extraction module. The fusion result is recorded as X_{cat}. To simultaneously learn fused feature results and reduce the number of channels, X_{cat} is subsequently fed into the 2D CNN with a kernel of 3×3. Then, the output of the 2D CNN is flattened to obtain $X_{in} \in R^{S^2 \times D}$, where S represents the size of the 3D patch in the spatial dimension, either height or width. And D is the number of channels. The above process is expressed as in the following formulas:

$$X_{cat} = \text{Concat}\left(X, X_{spe}, X_{spa}\right) \tag{15}$$

$$X_{in} = \text{Flatten}(Conv2D_3(X_{cat})) \tag{16}$$

To guide feature distribution regularization, the tokenizer operation uses two learnable weights initialized using the Xavier standard normal distribution, W_a and W_b, which are multiplied by X_{in} to extract critical features. That is, P feature vectors are selected from S^2 feature vectors for tokenization, where $0 < P < S^2$. The softmax function is employed to emphasize relatively significant information. This processing converts shallow and deep spectral–spatial features into tokenized semantic features, which aims to align the deep semantic features more closely with the distribution characteristics of the samples, thereby enhancing their separability [44]. The following equations summarize this process.

$$X_a = \text{softmax}(X_{in} \cdot W_a)^T \tag{17}$$

$$X_b = X_{in} \cdot W_b \tag{18}$$

$$X_{out} = X_a \cdot X_b \tag{19}$$

In this process, $W_a \in R^{D \times P}, W_b \in R^{D \times D}; X_a \in R^{P \times S^2}, X_b \in R^{S^2 \times D}$. $X_{out} \in R^{P \times D}$ is the output of the tokenizer. X_{out} consists of P tokens, which can be expressed as $[X_1, X_2, \cdots, X_P]$.

To achieve a better HSI classification, a classification token X_{cls} is introduced here. It dynamically updates during network training, aggregating global features from other to-

kens to prevent bias towards any individual token. Then, the positional coding information X_{pos} is added. The above process is expressed in the following formula.

$$X_{tokens} = [X_{cls}, X_1, X_2, \cdots, X_P] + X_{pos} \tag{20}$$

The positional encoding $X_{pos} \in R^{(P+1)\times D}$ is a learnable parameter initialized using a normal distribution. As the network trains, it learns the row and column information of the pixels. X_{cls} possesses a fixed positional encoding (at position 0), alleviating interference caused by positional encoding. $X_{tokens} \in R^{(P+1)\times D}$ is the output of the cascade X_{cls} and the addition of positional encoded information. Furthermore, the transformed features are passed through the Transformer Encode module to further learn the deep semantic features abstractly after fusion. Finally, the output is classified using two linear layers.

2.4. Implementation of TransHSI

As shown in Table 2, a detailed summary of the types of layers, output dimensions of maps, and the number of parameters for the TransHSI network is provided on the Pavia University dataset. The size of the 3D patch extracted using the Pavia University dataset is $15 \times 9 \times 9$. In the spectral feature extraction module of the TransHSI network, 3D CNN_1 consists of 32@3×3×3 convolutional kernels; 3D CNN_2 and 3D CNN_4 each consist of 64@5×3×3 convolutional kernels, and 3D CNN_3 and 3D CNN_5 each consist of 32@7×3×3 convolutional kernels. In addition, 2D CNN_1 comprises 64@1×1 convolutional kernels. In the spatial feature extraction module, 2D CNN_2 and 2D CNN_4 each consist of 128@3×3 convolutional kernels, and 2D CNN_3 and 2D CNN_5 each consist of 64@3×3 convolutional kernels. Preceding the fusion module, the Transformer Encoder modules Trans_1 and Trans_2 outputs are resampled into 64@9×9 feature maps. Post fusion, 143($64 \times 2 + 15$) @9×9 feature maps are input into 2D CNN_6 containing 128@3×3 convolutional kernels. Subsequently, the 128@9×9 feature maps from 2D CNN_6 are resampled into 81 tokens of dimension 128. After selecting 5 feature vectors, which include 4 tokens and 1 introduced classifiable token, they are further processed through the Transformer Encoder module to output a $128 \times 1 \times 1$ classifiable token. Finally, a linear layer generates a $9 \times 1 \times 1$ prediction vector. In terms of the number of parameters, the total trainable parameters in the network amount to 1,049,569, with the highest number of parameters being attributed to 3D convolutions, followed by 2D convolutions.

Table 2. Layer-wise summary of TransHSI network based on the Pavia University Dataset.

Layer (Type)	Output Shape	Parameter
Input_1 (InputLayer)	(1, 15, 9, 9)	0
3D CNN_1 (3D CNN)	(32, 15, 9, 9)	960
3D CNN_2 (3D CNN)	(64, 15, 9, 9)	92,352
3D CNN_3 (3D CNN)	(32, 15, 9, 9)	129,120
3D CNN_4 (3D CNN)	(64, 15, 9, 9)	92,352
3D CNN_5 (3D CNN)	(32, 15, 9, 9)	129,120
2D CNN_1 (2D CNN)	(64, 9, 9)	30,912
Flat_1 (Flatten)	(81, 64)	128
Trans_1 (Transformer Encoder)	(81, 64)	17,864
2D CNN_2 (2D CNN)	(128, 9, 9)	74,112
2D CNN_3 (2D CNN)	(64, 9, 9)	73,920
2D CNN_4 (2D CNN)	(128, 9, 9)	74,112
2D CNN_5 (2D CNN)	(64, 9, 9)	73,920
Flat_2 (Flatten)	(81, 64)	128
Trans_2 (Transformer Encoder)	(81, 64)	17,864
Cascade layer (2D CNN)	(128, 9, 9)	165,120
Tokenizer layer (Flatten)	(5, 128)	256

Table 2. *Cont.*

Layer (Type)	Output Shape	Parameter
Trans_3 (Transformer Encoder)	(5, 128)	68,488
Cls_token (Identity)	(128)	0
Linear_1 (Linear layer)	(64)	8256
Linear_2 (Linear layer)	(9)	585
Total Trainable parameters: 1,049,569		

3. Datasets and Experimental Setup

3.1. Experimental Datasets

In this study, three commonly used hyperspectral datasets are used for experiments: the Indian Pines dataset, the Pavia University dataset, and the Data Fusion Contest 2018. Moreover, the practice of randomly sampling training samples on the entire image is not adopted because in practical applications, the training and test samples are often collected from different locations. When the method uses the CNNs to process data, the receptive field will unconsciously contain the samples in the test set, meaning that the classification accuracy of the test set is set at a high level [1]. Therefore, experiments are performed using disjoint training and test sets. We downloaded the training sets for the Indian Pines and the Pavia University datasets are downloaded from the IEEE GRSS DASE website: http://dase.grss-ieee.org/, accessed on 6 October 2022. The training set for the Data Fusion Contest 2018 was manually selected. The number of training and test dataset samples for each class within the three datasets is shown in Table 3.

Table 3. Land cover categories and training and test sample sizes of the Indian Pines Dataset, the Pavia University Dataset, and Data Fusion Contest 2018.

No.	Indian Pines Dataset			Pavia University Dataset			Data Fusion Contest 2018		
	Class	Training	Test	Class	Training	Test	Class	Training	Test
1	Alfalfa	21	25	Asphalt	327	6304	Healthy grass	858	8941
2	Corn-notill	753	675	Meadows	503	18,146	Stressed grass	1954	30,548
3	Corn-mintill	426	404	Gravel	284	1815	Artificial turf	126	558
4	Corn	138	99	Trees	152	2912	Evergreen trees	1810	11,785
5	Grass-pasture	209	274	Painted metal sheets	232	1113	Deciduous trees	1073	3948
6	Grass-trees	376	354	Bare Soil	457	4572	Bare earth	600	3916
7	Grass-pasture-mowed	16	12	Bitumen	349	981	Water	74	192
8	Hay-windrowed	228	250	Self-Blocking Bricks	318	3364	Residential buildings	3633	36,139
9	Oats	10	10	Shadows	152	795	Non-residential buildings	18,571	205,181
10	Soybean-notill	469	503				Roads	2899	42,967
11	Soybean-mintill	1390	1065				Sidewalks	1962	32,067
12	Soybean-clean	311	282				Crosswalks	417	1101
13	Wheat	125	80				Major thoroughfares	3189	43,159
14	Woods	720	545				Highways	1518	8347
15	Buildings-Grass-Trees-Drives	287	99				Railways	708	6229
16	Stone-Steel-Towers	49	44				Paved parking lots	799	10,701
17							Unpaved parking lots	51	95
18							Cars	575	5972

Table 3. *Cont.*

No.	Indian Pines Dataset			Pavia University Dataset			Data Fusion Contest 2018		
	Class	Training	Test	Class	Training	Test	Class	Training	Test
19							Trains	155	5214
20							Stadium seats	556	6268
Total		5528	4721		2774	40002		41,528	463,328

3.1.1. Indian Pines Dataset

The Indian Pines dataset was collected in northwestern Indiana, USA, using an Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor. It comprises two-thirds agriculture and one-third forest or other natural perennials. The HSI comprises 145 × 145 pixels, has a ground sampling distance (GSD) of 20 m, and 220 spectral bands that cover a wavelength range of 400–2500 nm. After removing the 20 noise and water absorption bands, 200 spectral bands were retained, namely 1–103, 109–149, and 164–219. This study primarily has 16 land cover categories, some of which are not mutually exclusive. The false-color map, ground truth map, and distribution of the training and test sets for the Indian Pines dataset are shown in Figure 5.

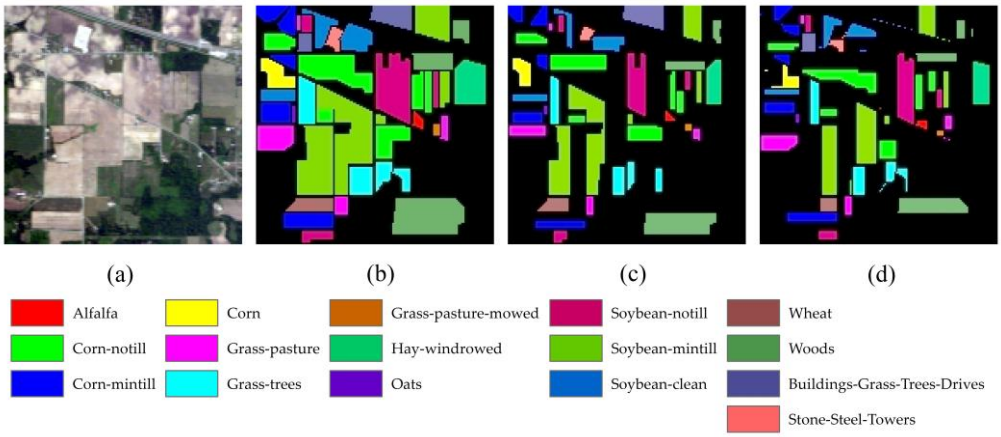


Figure 5. Indian Pines Dataset: (**a**) false-color map, (**b**) ground truth map, (**c**) training, (**d**) test.

3.1.2. Pavia University Dataset

The Pavia University dataset is made up of scene data from the University of Pavia and its surroundings obtained using the Reflective Optics System Imaging Spectrometer (ROSIS) sensor. The sensor includes 103 spectral bands at 430–860 nm, and the image consists of 610 × 340 pixels with a GSD of 1.3 m. The scene contains nine land cover classes. The false-color map, ground truth map, and distribution of the training and test sets for the Pavia University dataset are shown in Figure 6.

3.1.3. Data Fusion Contest 2018 (DFC 2018)

The DFC 2018 was acquired with onboard sensors in downtown Houston, Texas, USA. The image consists of 610 × 2384 pixels, with a GSD of 1 m, which covers 48 adjacent bands of 380–1050 nm. The dataset includes not only urban categories, such as various types of buildings, roads, cars, and trains, but also various vegetation types, including healthy or stressed grass, deciduous or evergreen trees, etc. The dataset is part of the 2018 Data Fusion Contest, with 20 land cover categories. The false-color map, ground truth map, and distribution of the training and test sets for the DFC 2018 are shown in Figure 7.

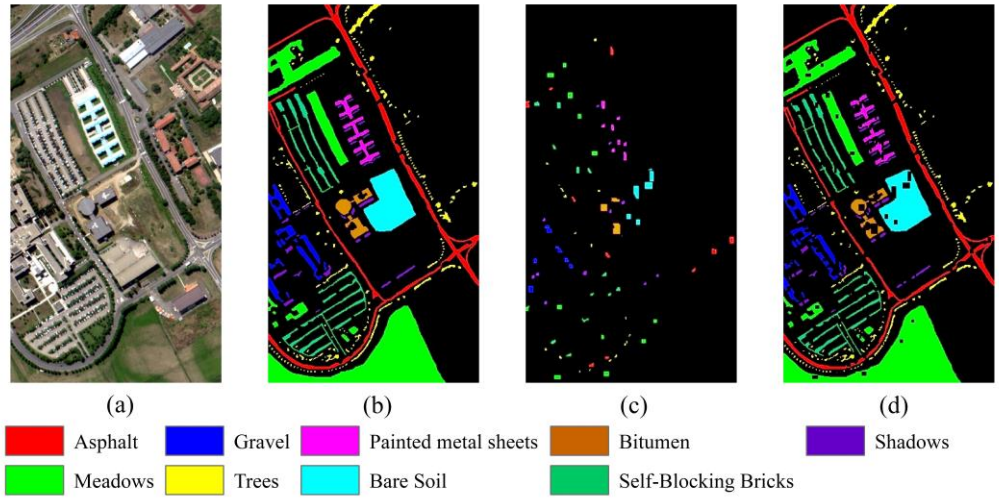


Figure 6. Pavia University Dataset: (**a**) false-color map, (**b**) ground truth map, (**c**) training, (**d**) test.

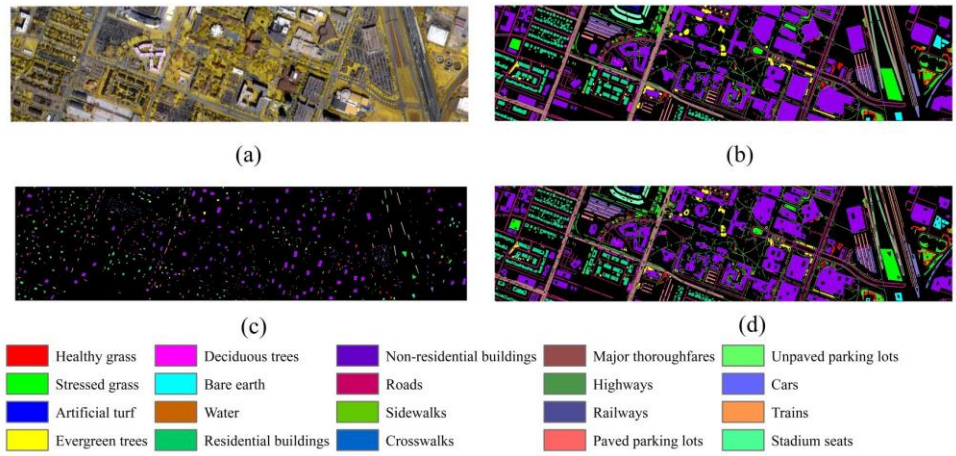


Figure 7. DFC 2018: (**a**) false-color map, (**b**) ground truth map, (**c**) training, (**d**) test.

3.2. Experimental Setup

In this paper, we used Python language and the PyTorch library to implement the classification method on the Windows 10 operating system. The experimental environment consists of the Intel(R) Core (TM) i9-9900K CPU @ 3.60 GHz, 32 GB memory, and NVIDIA GeForce RTX 2080 GPU. The learning rate is set to 0.001, and the batch size for both the training and test is set to 32. The experimental results are quantitatively evaluated using three indicators: overall accuracy (OA), average accuracy (AA), and the kappa coefficient (κ). The OA is the ratio of the total number of correctly classified samples to the total number of samples tested across all tests. AA is the average of the classification accuracy for all categories. The kappa coefficient is used to assess the categorical agreement of all classes and is calculated from the confusion matrix. If the sample size is the same for each category, OA and AA are equal. The strong difference between OA and AA may indicate a high percentage of specific categories being misclassified. The higher the value of the network on these three indicators, the better the classification ability of the network.

To fully utilize the spectral–spatial information of HSIs, as explained in Section 2.3.1, the original hyperspectral data undergo PCA processing. Then, 3D patches of a specific size around central pixels are selected as inputs for the network. The chosen size of a 3D patch significantly influences the amount of information utilized for classification and, to some extent, impacts the classification effect. As the patch size increases, more spectral–spatial information from the neighborhood may be introduced. However, a larger patch size can also introduce irrelevant information, leading to information redundancy and higher

computational costs. Therefore, when determining the optimal patch size for each dataset, a balance must be struck between computational cost and classification performance. The number of spectral bands after PCA (denoted as B) also follows a similar consideration. Due to significant differences in the sample sizes of the three datasets, we conducted experiments with different patch sizes and spectral band numbers, as illustrated in Figure 8 below. The OAs for the three datasets initially increased and then gradually decreased with an increase in input size. The change in OA was not consistent across datasets due to variations in data acquisition equipment and the spectral reflectance characteristics of ground objects. Specifically, for the Indian Pines dataset with a smaller sample size and a larger number of spectral bands (145 × 145 × 200), we used a patch size of 11 and B of 30. For the moderate-sized Pavia University dataset (610 × 340 × 103) and the DFC2018 with a larger sample size and fewer spectral bands (610 × 2384 × 48), we used a patch size of 9 and B of 15.

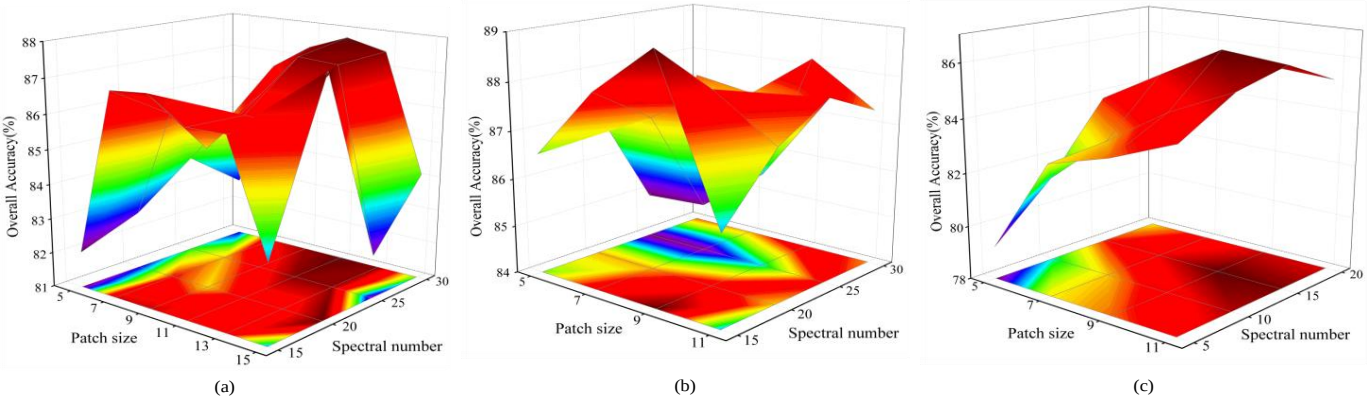


Figure 8. Impact of patch size and spectral number after PCA for OA (%). (**a**) The Indian Pines dataset; (**b**) the Pavia University dataset; (**c**) the DFC2018.

4. Experimental Result

This section uses several comparative experiments to verify the classification performance of the proposed method, TransHSI. The comparative methods include traditional classification methods like SVM [1] and RF [1]; CNN-based methods including 2D CNNs [20], 3D CNNs [20], HybirdSN [22], SSRN [21], and InternImage [53] based on deformable convolution (DCNv3); Transformer network-based methods such as ViT [40]; and methods combining CNNs and Transformer blocks, like Next-ViT [46], SSFTT [44], and SSTN [45]. To minimize experimental errors, each experiment was conducted three times, and the average of the three results was taken as the classification accuracy, as shown in Tables 4–6. In the TransHSI and comparative methods, the highest three comprehensive indicators are highlighted in bold.

Table 4. Classification accuracies of various methods in Indian Pines Dataset.

Class	SVM	RF	2D CNN	3D CNN	HybirdSN	SSRN	InternImage	ViT	Next -ViT	SSFTT	SSTN	TransHSI
1	0.00	18.67	4.00	66.67	81.33	89.33	75.27	80.00	80.00	76.00	13.33	88.00
2	61.78	64.55	76.54	76.45	79.16	74.27	88.45	68.59	77.58	79.55	70.86	90.82
3	48.43	43.48	70.54	85.81	79.79	76.57	78.71	82.92	79.46	76.98	81.68	79.87
4	22.22	23.57	90.24	34.68	51.52	60.61	71.52	65.66	71.38	37.71	60.60	59.59
5	78.71	71.53	82.36	88.93	88.56	88.93	88.03	83.94	90.88	89.17	87.10	91.61
6	93.50	96.80	96.33	99.06	98.12	89.27	90.96	97.18	86.53	97.55	92.37	92.37
7	38.89	58.33	11.11	86.11	80.56	33.33	59.44	33.33	75.00	86.11	0.00	97.22
8	100.00	100.00	99.87	98.67	99.60	100.00	99.85	100.00	99.87	98.67	100.00	96.93
9	0.00	26.67	6.67	50.00	50.00	0.00	65.92	0.00	96.67	83.33	0.00	96.67
10	37.84	27.70	62.29	67.40	78.99	90.26	71.28	85.29	74.55	77.47	78.80	85.15
11	84.63	83.38	80.66	76.90	89.36	82.82	81.08	76.43	80.00	82.47	87.80	83.44

Table 4. *Cont.*

Class	SVM	RF	2D CNN	3D CNN	HybirdSN	SSRN	InternImage	ViT	Next -ViT	SSFTT	SSTN	TransHSI
12	55.67	52.60	58.75	68.32	81.20	83.09	68.15	79.79	72.93	81.44	95.74	83.33
13	92.92	93.75	95.83	97.50	89.58	90.00	82.59	83.75	88.33	92.92	90.42	99.17
14	94.56	92.54	95.05	96.39	99.27	100.00	99.30	99.27	98.53	99.45	99.88	98.84
15	60.95	66.33	59.94	43.09	37.37	10.10	47.87	27.27	35.35	47.14	46.47	76.10
16	90.15	97.73	100.00	90.15	93.18	87.88	65.40	88.64	72.73	100.00	83.34	88.64
OA (%)	71.47 ± 10.27	69.93 ± 0.09	79.36 ± 0.29	80.63 ± 0.36	85.79 ± 0.67	83.51 ± 0.18	83.22 ± 0.98	81.61 ± 0.71	81.90 ± 1.99	83.97 ± 0.57	84.47 ± 2.00	**87.75 ± 0.35**
AA (%)	60.01 ± 5.83	63.60 ± 0.33	68.13 ± 1.76	76.63 ± 0.56	83.04 ± 1.85	72.28 ± 0.68	77.11 ± 3.77	72.00 ± 1.77	79.99 ± 5.33	81.62 ± 1.35	68.02 ± 2.24	**88.42 ± 1.37**
K (%)	66.97 ± 12.19	65.23 ± 0.10	76.48 ± 0.38	77.95 ± 0.42	84.53 ± 0.43	81.29 ± 0.20	80.91 ± 1.13	79.21 ± 0.81	80.30 ± 1.52	81.79 ± 0.67	82.34 ± 2.27	**86.11 ± 0.41**

Table 5. Classification accuracies of various methods in Pavia University Dataset.

Class	SVM	RF	2D CNN	3D CNN	HybirdSN	SSRN	InternImage	ViT	Next-ViT	SSFTT	SSTN	TransHSI
1	66.28	79.03	90.44	86.48	92.34	96.37	94.35	87.99	89.38	92.93	88.67	89.58
2	83.95	56.78	70.57	75.58	75.50	86.33	78.54	71.29	84.53	77.14	72.99	88.59
3	35.54	43.22	69.97	69.51	65.84	68.08	73.38	68.30	63.07	82.26	56.97	76.53
4	93.10	95.91	81.21	72.78	86.44	84.71	87.02	85.23	87.37	87.58	78.80	92.07
5	99.28	99.10	99.91	99.94	100.00	99.97	99.75	100.00	100.00	100.00	99.55	99.91
6	33.03	77.45	94.02	89.52	90.13	78.81	97.13	96.65	91.28	88.99	97.19	80.75
7	90.52	79.14	98.51	99.15	99.46	98.03	97.10	95.07	97.69	98.24	97.76	97.96
8	91.35	88.03	97.51	97.39	98.27	99.13	97.27	97.53	96.16	96.36	97.77	97.11
9	99.87	99.79	98.25	99.79	99.87	99.58	96.81	94.92	98.78	99.96	97.48	99.54
OA (%)	75.34 ± 0.00	70.09 ± 0.09	81.45 ± 0.24	81.98 ± 1.04	83.85 ± 0.91	88.11 ± 0.91	86.52 ± 0.14	81.76 ± 0.03	87.32 ± 1.11	85.20 ± 0.06	81.84 ± 1.16	**89.03 ± 1.72**
AA (%)	76.99 ± 0.00	79.83 ± 0.05	88.93 ± 0.57	87.79 ± 1.04	89.76 ± 0.48	90.11 ± 0.71	91.26 ± 0.41	88.55 ± 0.50	89.81 ± 0.56	**91.49 ± 0.16**	87.47 ± 1.24	91.34 ± 2.05
κ (%)	66.88 ± 0.00	62.76 ± 0.08	76.31 ± 0.20	76.71 ± 1.35	79.10 ± 1.03	84.21 ± 1.04	82.51 ± 0.19	76.71 ± 0.01	83.32 ± 1.38	80.78 ± 0.11	76.75 ± 1.41	**85.41 ± 2.16**

Table 6. Classification accuracies of various methods in DFC 2018.

Class	SVM	RF	2D CNN	3D CNN	HybirdSN	SSRN	InternImage	ViT	Next -ViT	SSFTT	SSTN	TransHSI
1	92.94	91.14	85.05	82.70	87.25	87.77	76.86	80.27	80.48	90.06	81.03	83.42
2	88.57	87.49	87.42	88.67	85.61	84.94	82.74	82.86	89.44	83.84	85.26	88.54
3	100.00	100.00	97.79	98.80	99.58	99.64	97.79	96.59	97.91	97.85	94.56	99.46
4	97.11	96.76	98.27	98.40	98.32	97.44	95.63	96.67	98.03	97.63	95.96	96.69
5	83.64	83.76	93.75	93.26	97.95	94.83	94.19	94.73	97.05	95.34	85.79	97.68
6	91.35	89.35	93.34	92.15	93.75	94.91	91.86	93.57	95.75	92.40	94.69	92.59
7	98.96	98.09	98.61	99.65	99.83	100.00	93.23	98.79	99.31	97.57	96.18	100.00
8	80.55	79.77	82.47	86.26	87.17	86.46	84.64	87.19	87.97	82.04	87.76	88.91
9	88.75	93.00	89.35	91.24	91.65	92.02	91.42	89.10	92.14	91.02	91.67	92.12
10	41.19	47.04	59.75	62.60	62.09	64.32	62.30	60.67	65.53	61.91	64.03	68.66
11	48.26	58.45	68.94	67.64	76.50	78.36	73.67	67.43	69.36	72.74	61.05	74.13
12	10.35	29.70	38.81	42.78	62.97	79.11	73.63	75.81	60.61	65.21	14.83	72.69
13	52.80	52.71	67.22	65.04	72.17	69.86	66.40	68.70	72.41	67.91	72.98	72.63
14	60.37	59.76	71.14	78.01	77.30	81.82	77.13	88.06	75.62	67.84	70.14	80.64
15	96.77	94.39	98.78	98.93	99.55	98.07	99.09	95.08	99.33	97.69	99.06	99.40
16	61.21	71.45	91.14	91.67	93.62	94.11	91.75	87.44	91.05	90.88	90.56	93.70
17	84.21	99.65	90.52	94.74	100.00	99.65	98.60	100.00	98.95	98.59	98.59	100.00
18	27.14	42.29	89.74	93.70	95.80	95.10	91.95	92.90	94.72	92.88	88.17	96.75
19	20.23	34.91	87.41	88.45	90.19	88.54	88.03	92.36	91.96	83.17	74.62	88.53
20	70.15	78.82	92.00	93.76	96.89	96.76	90.03	96.89	94.20	96.28	87.88	96.08
OA (%)	74.78 ± 0.00	78.44 ± 0.03	82.44 ± 0.20	83.80 ± 0.24	85.40 ± 0.10	85.64 ± 0.23	83.68 ± 0.22	82.81 ± 0.39	85.52 ± 0.28	83.55 ± 0.38	83.62 ± 0.58	**86.36 ± 0.65**
AA (%)	69.73 ± 0.00	74.43 ± 0.04	84.07 ± 0.65	85.42 ± 0.44	88.41 ± 0.58	**89.19 ± 0.85**	86.05 ± 0.40	87.25 ± 0.40	87.59 ± 1.15	86.14 ± 0.36	81.74 ± 1.37	89.13 ± 0.70
κ (%)	67.32 ± 0.00	71.73 ± 0.03	77.59 ± 0.23	79.26 ± 0.28	81.30 ± 0.10	81.60 ± 0.20	79.07 ± 0.29	78.15 ± 0.46	81.40 ± 0.34	78.90 ± 0.45	78.95 ± 0.76	**82.53 ± 0.77**

4.1. Classification Results for the Indian Pines Dataset

As can be observed from Table 4, among all the classification methods applied to the Indian Pines dataset, the traditional classification method RF exhibits the poorest results in terms of the classification indicators OA, AA, and the kappa coefficient. At the same time, the classification accuracies of SVM are also low. In contrast, high-dimensional CNN methods such as 2D CNN, 3D CNN, and ViT networks with the MHSA mechanisms have better classification effects than traditional methods. This is because traditional classification methods rely solely on spectral information, whereas deep learning methods can also leverage spatial features [1]. Moreover, methods like HybirdSN, which mixes 2D CNNs and 3D CNNs, SSRN, which adds residual connection structures to 3D CNNs, InternImage based on DCNv3, and Next-ViT, SSFTT, and SSTN methods, which integrate CNNs and Transformer blocks, achieve classification accuracies higher than 81% with better classification effects. This implies that, compared to solely relying on convolution and attention mechanisms, hybrid networks have the capacity to learn and capture a broader range of spectral features and spatial contextual information [54]. The proposed TransHSI method achieved an OA of 87.75%, AA of 88.42%, and kappa coefficient of 86.11%, yielding the best performance. Among the six types of land cover categories, the TransHSI method achieves the highest accuracy. Additionally, the highly accurate classification methods HybirdSN and SSTN achieved OAs of over 84%. However, TransHSI outperformed HybirdSN and SSTN in terms of OA, with improvements of 1.96% and 3.28%, respectively. Regarding the AA, TransHSI demonstrated significant improvements compared to HybirdSN, SSFTT, and SSTN, with increases of 5.38%, 6.8%, and 20.4%, respectively. TransHSI also achieved a higher kappa coefficient than HybirdSN and SSTN, with improvements of 1.58% and 3.77%, respectively.

Figure 9 displays the classification results of the 12 methods in the Indian Pines dataset. It is evident that the TransHSI method (Figure 9l) exhibits the least noise, while SVM (Figure 9a) and RF (Figure 9b) suffer from significant noise scattering and a high number of misclassified areas, performing noticeably worse than the other classification networks. Additionally, in categories with less satisfactory visual results, the misclassification areas of the same land cover types, such as soybean-notill, soybean-mintill, and soybean-clean, are relatively smaller in the results of HybirdSN (Figure 9f), SSRN (Figure 9g), SSTN (Figure 9k), and TransHSI (Figure 9l). In the land cover category "corn-notill", TransHSI has the fewest misclassified regions compared to the other three methods (Figure 9f,g,k). These observations align with the results presented in Table 4.

4.2. Classification Results for the Pavia University Dataset

As shown in Table 5 and Figure 10, the classification accuracies of SVM (Figure 10a) and RF (Figure 10b) for the Pavia University dataset are still very poor, with many misclassification areas in the visualization of the classification results. This is especially noticeable in the land cover categories of the "meadows" and "bare soil", which are somewhat similar. However, this situation improved in the subsequent methods (Figure 10d–k). The TransHSI method (Figure 10l) demonstrates exemplary performance in the classification map. The proposed TransHSI method achieved an OA of 89.03%, AA of 91.34%, and kappa coefficient of 85.41%. In the TransHSI classification results, although the highest accuracy is achieved for a single land cover class, the OA and kappa coefficient metrics rank first among all the methods. The OAs of the SSRN, Next-ViT, InternImage, and SSFTT methods all reach more than 85%, but the OA of the TransHSI method is higher than those of the SSRN, Next-ViT, InternImage, and SSFTT methods by 0.92%, 1.71%, 2.51%, and 3.83%. The AA of the TransHSI method is second only to that of SSFTT. The kappa coefficient of the TransHSI method is 1.20%, 2.09%, 2.90%, and 4.63% higher than those of the SSRN, Next-ViT, InternImage, and SSFTT methods.

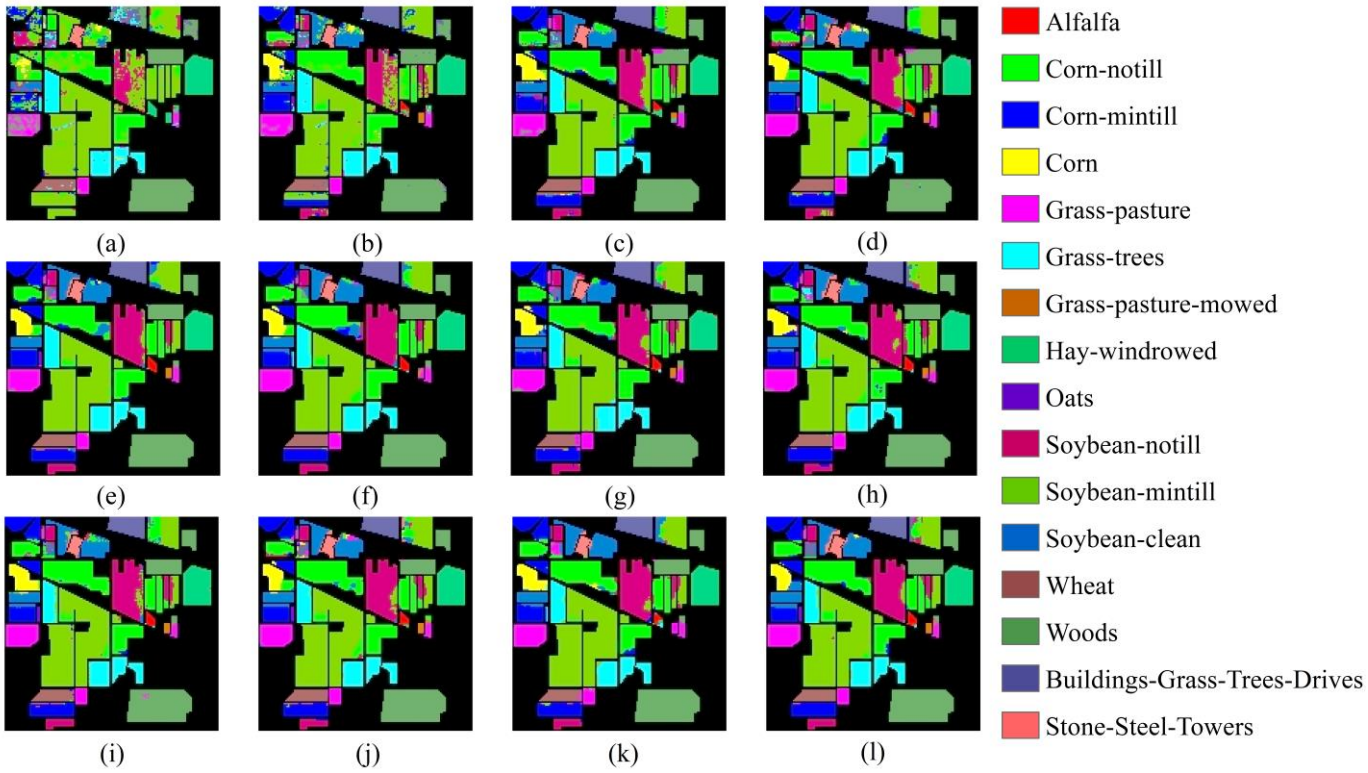


Figure 9. The classification results of the Indian Pines Dataset. (**a**) SVM; (**b**) RF; (**c**) 2D CNN; (**d**) 3D CNN; (**e**) HybirdSN; (**f**) SSRN; (**g**) InternImage; (**h**) ViT; (**i**) Next-ViT; (**j**) SSFTT; (**k**) SSTN; (**l**) TransHSI.

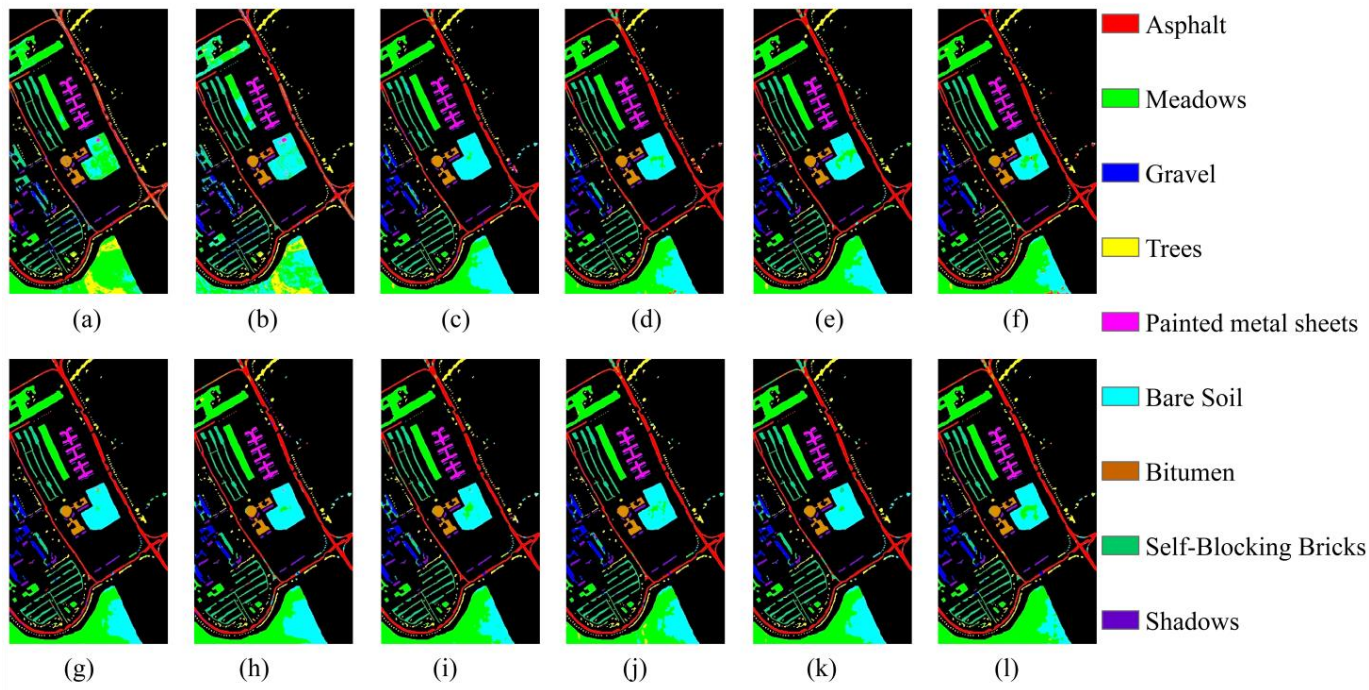


Figure 10. The classification results of the Pavia University dataset. (**a**) SVM; (**b**) RF; (**c**) 2D CNN; (**d**) 3D CNN; (**e**) HybirdSN; (**f**) SSRN; (**g**) InternImage; (**h**) ViT; (**i**) Next-ViT; (**j**) SSFTT; (**k**) SSTN; (**l**) TransHSI.

4.3. Classification Results for the DFC 2018

As can be seen from Table 6 and Figure 11, among all the classification methods applied to the DFC2018, the traditional classification method SVM has the lowest values in

terms of the classification indicators OA, AA, and kappa coefficient. RF has higher values for these three comprehensive evaluation indicators than SVM, but its OAs do not reach 80%. From the visual classification in Figure 11, it is evident that SVM (Figure 11a) and RF (Figure 11b) have a more significant number of misclassified regions. In comparison with the ground truth map in Figure 6b, it can be observed that the classification map produced by the TransHSI method (Figure 11l) yields more accurate results. In the case of other methods, there is still salt-and-pepper noise present in the boundary regions of the "stadium seats". The proposed TransHSI method achieves an OA of 86.36%, AA of 89.13%, and kappa coefficient of 82.53%. In the TransHSI classification results, although the accuracies of only five land cover categories reach the maximum values, both the OA and kappa coefficient indicators rank first among all networks. Secondly, the OAs of the SSRN, Next-ViT, and HybirdSN methods reach more than 85%, but their OAs are 0.72%, 0.84%, and 0.96% lower than that of the TransHSI method, respectively. The AA of the TransHSI method is second only to that of SSRN. The kappa coefficients of the SSRN, Next-ViT, and HybirdSN methods are lower than that of the TransHSI method by 0.93%, 1.13%, and 1.23%.

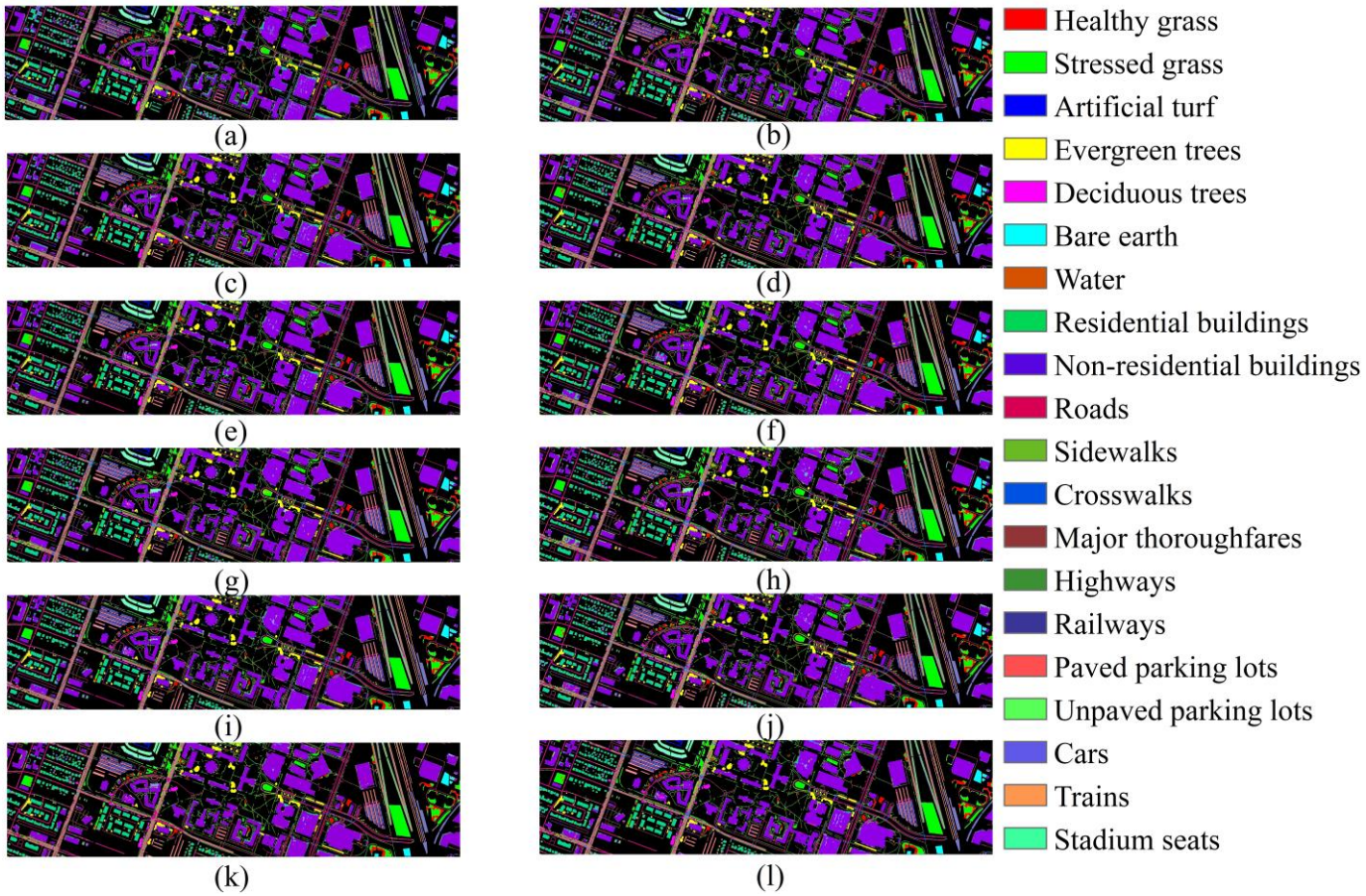


Figure 11. The classification results of DFC2018. (**a**) SVM; (**b**) RF; (**c**) 2D CNN; (**d**) 3D CNN; (**e**) HybirdSN; (**f**) SSRN; (**g**) InternImage; (**h**) ViT; (**i**) Next-ViT; (**j**) SSFTT; (**k**) SSTN; (**l**) TransHSI.

4.4. Visualization Analysis of TransHSI

As each method was subjected to three experiments on every dataset, the confusion matrices of the results were visualized by sorting them in ascending order of OAs and selecting the middle results. Figure 12 clearly illustrates that using TransHSI on all three datasets has led to the successful separation of the majority of the categories, as evidenced by the bar charts on the diagonal. However, some classes do not achieve ideal classification performance, such as "Corn" and "Buildings-Grass-Trees-Drives" on the Indian Pines dataset. Corn is often misclassified as "Soybean-clean", possibly due to the scene being

captured in June, where corn and soybean plants are in their early stages of growth, making them challenging to distinguish. Furthermore, as indicated in Table 4, the proposed method outperforms other comparative methods in terms of accuracy for "Buildings-Grass-Trees-Drives", which consists of categories with diverse land cover types.

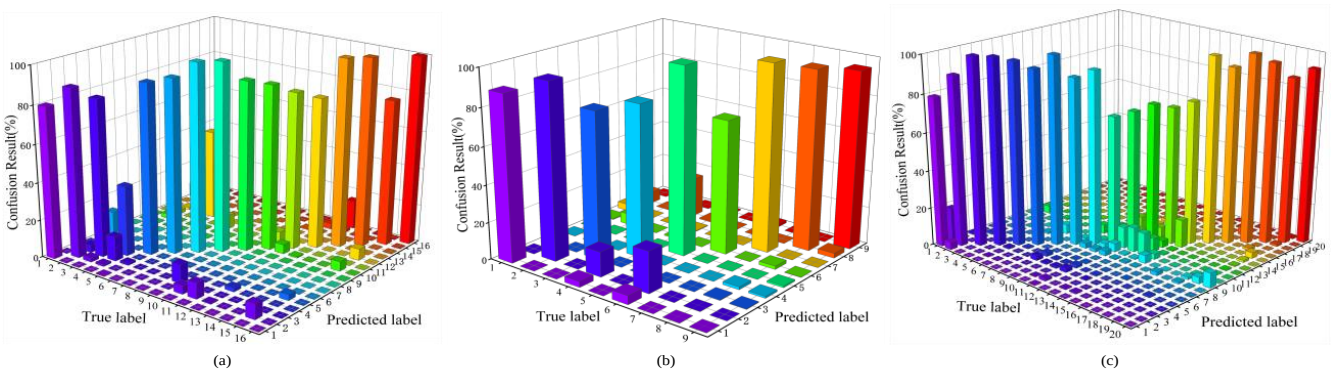
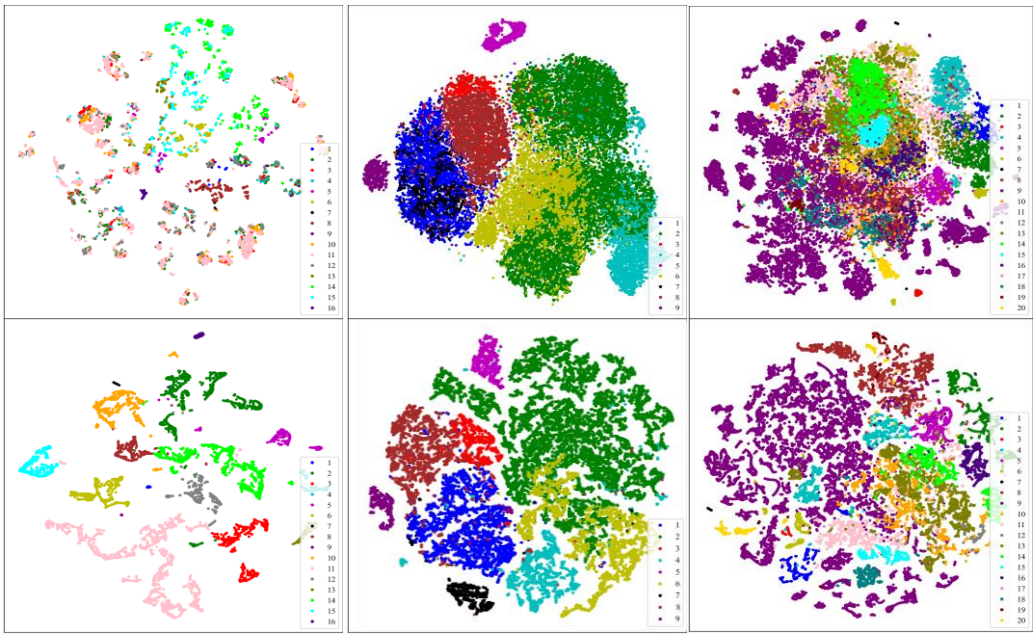


Figure 12. The 3D confusion matrix of TransHSI. (**a**) The Indian Pines dataset; (**b**) the Pavia University dataset; (**c**) the DFC2018.

On the Pavia University dataset, "Gravel" is frequently misclassified as "Self-Blocking Bricks". "Trees" and "Bare Soil" are commonly confused with "Meadows". The false-color image of the Pavia University dataset indicates that these categories indeed share similarities, making them challenging to distinguish. Similarly, on the DFC 2018, "Roads", "Sidewalks", "Crosswalks", "Major thoroughfares", and "Highways" also exhibit mutual misclassifications due to their common road-related attributes, high similarity, and, thus, a tendency to be confused. Tables 5 and 6 indicate that although the accuracy of most misclassified classes does not reach the highest values when compared to the other methods, their performance remains commendable.

The t-Distributed Stochastic Neighbor Embedding (t-SNE) [55] algorithm can map high-dimensional data to a lower-dimensional data, reducing the distances between similar categories, and increasing the gaps between different categories while preserving the local characteristics of the original data. To gain a more intuitive understanding of the feature learning process of the proposed method [56], we employed t-SNE to map the original sample and output layer features of three datasets into a two-dimensional space. As shown in Figure 13, the feature distribution of the original sample in the three datasets appears more chaotic, with higher levels of overlap, especially in datasets with a large number of categories like the Indian Pines dataset and DFC2018. Conversely, in the feature distribution of the output layer, the boundaries of most categories are clearly distinguished.



(a) (b) (c)

Figure 13. Visualization of the t-SNE algorithm in the TransHSI method for three datasets (top/original features and bottom/output features). (**a**) The Indian Pines dataset; (**b**) the Pavia University dataset; (**c**) the DFC2018.

5. Discussion

TransHSI achieved competitive classification results on the Indian Pines, Pavia University, and DFC2018, indicating a certain degree of generalization capability of the proposed method. In this section, we will discuss the TransHSI network's architecture and ablation experiment results on the Indian Pines and Pavia University datasets, analyzing the functionality of each network component. Additionally, on the three datasets, we will explore the impact of different training set percentages on the classification performance.

5.1. Ablation Experiments

5.1.1. Quantitative Comparison of Classification Results

This paper proposes a method of fusing convolution and Transformer models to extract the spectral–spatial features of hyperspectral data. To evaluate the effectiveness of each component of TransHSI, seven ablation experiments were conducted.The following is shown in Table 7.

The results of the TransHSI method and ablation experiments on the two hyperspectral datasets are shown in Tables 8 and 9. From the tables, it is evident that TransHSI outperforms all other methods regarding both OA and the kappa coefficient. Furthermore, TransHSI achieves the highest AA on the Indian Pines dataset. On the Pavia University dataset, the AA of the TransHSI method is not much different from the highest AA. These indicate that the proposed network possesses a certain degree of robustness.

Table 7. Network structure components for TransHSI and ablation experiments. (**a**) TransHSI removal of the spectral feature extraction module; (**b**) TransHSI removal of the spatial feature extraction module; (**c**) TransHSI removal of the cascading layer; (**d**) TransHSI removal of all Transformer Encode modules; (**e**) TransHSI removal of Transformer Encode from the spectral feature extraction module; (**f**) TransHSI removal of Transformer Encode from the spatial feature extraction module; (**g**) TransHSI removal of Transformer Encode from the fusion module. It is worth noting that in methods (**d**) and (**g**), a fully connected layer is inserted into the fusion module to generate the classification results.

Experiments	3D CNNs	Trans_1	2D CNNs	Trans_2	Concat	Trans_3
(a)	×	×	√	√	√	√
(b)	√	√	×	×	√	√
(c)	√	√	√	√	×	√
(d)	√	×	√	×	√	×
(e)	√	×	√	√	√	√
(f)	√	√	√	×	√	√
(g)	√	√	√	√	√	×
TransHSI	√	√	√	√	√	√

Table 8. The classification results of (a), (b), (c), (d), (e), (f), (g) and the TransHSI method in the Indian Pines Dataset.

Class	(a)	(b)	(c)	(d)	(e)	(f)	(g)	TransHSI
1	94.67 ± 4.62	98.67 ± 2.31	61.33 ± 53.12	60.00 ± 8.00	98.67 ± 2.31	100.00 ± 0.00	96.00 ± 6.93	88.00 ± 6.93
2	88.64 ± 5.71	85.73 ± 7.93	83.36 ± 5.12	81.33 ± 1.43	82.62 ± 2.06	84.30 ± 3.81	79.90 ± 10.33	90.82 ± 3.08
3	78.96 ± 1.73	87.21 ± 3.97	86.22 ± 5.51	82.10 ± 4.84	82.67 ± 1.87	80.45 ± 2.75	80.61 ± 5.88	79.87 ± 3.62
4	56.57 ± 11.91	44.11 ± 8.23	45.79 ± 2.54	84.18 ± 7.44	53.20 ± 23.26	40.40 ± 2.67	78.17 ± 3.55	59.59 ± 30.17
5	92.58 ± 0.56	91.24 ± 1.83	89.30 ± 2.20	89.05 ± 0.97	89.17 ± 0.56	92.95 ± 1.28	88.93 ± 1.12	91.61 ± 0.97
6	88.51 ± 0.91	98.68 ± 0.99	94.73 ± 2.27	96.23 ± 0.71	91.90 ± 1.88	95.29 ± 2.40	86.25 ± 2.12	92.37 ± 0.29
7	75.00 ± 0.00	86.11 ± 24.06	61.11 ± 53.58	86.11 ± 12.73	91.67 ± 14.43	77.78 ± 17.35	61.11 ± 52.93	97.22 ± 4.81
8	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	99.07 ± 1.62	100.00 ± 0.00	96.93 ± 5.31
9	46.67 ± 5.77	80.00 ± 34.64	46.67 ± 50.33	43.33 ± 11.55	30.00 ± 43.59	96.67 ± 5.77	0.00 ± 0.00	96.67 ± 5.77
10	70.97 ± 4.83	89.53 ± 5.81	81.38 ± 4.13	85.69 ± 3.81	78.99 ± 2.95	86.81 ± 5.62	93.64 ± 1.39	85.15 ± 3.54
11	82.75 ± 1.81	83.94 ± 5.77	91.67 ± 1.52	81.28 ± 2.27	82.63 ± 3.10	83.29 ± 1.93	84.63 ± 0.95	83.44 ± 2.16
12	87.71 ± 3.79	86.17 ± 4.09	91.25 ± 6.07	80.14 ± 1.07	93.15 ± 4.38	92.91 ± 0.62	95.98 ± 1.14	83.33 ± 5.84
13	93.75 ± 1.25	96.67 ± 2.60	96.25 ± 4.51	95.00 ± 4.33	95.83 ± 2.60	95.00 ± 1.25	90.00 ± 0.00	99.17 ± 0.72
14	99.69 ± 0.28	99.08 ± 0.66	99.45 ± 0.64	98.23 ± 2.76	97.49 ± 1.87	99.21 ± 0.56	100.00 ± 0.00	98.84 ± 2.01
15	24.24 ± 3.50	23.57 ± 16.45	14.14 ± 3.03	42.76 ± 6.17	68.69 ± 7.28	30.30 ± 20.28	34.68 ± 4.98	76.10 ± 2.10
16	95.46 ± 6.01	100.00 ± 0.00	66.67 ± 57.74	88.63 ± 3.94	85.61 ± 12.92	98.48 ± 2.63	76.51 ± 10.50	88.64 ± 10.41
OA (%)	84.68 ± 1.00	87.67 ± 0.33	87.25 ± 0.10	85.65 ± 0.42	85.92 ± 1.28	86.70 ± 0.23	86.86 ± 0.49	**87.75 ± 0.35**
AA (%)	79.76 ± 0.60	84.42 ± 4.26	75.58 ± 12.19	80.88 ± 0.93	82.64 ± 2.54	84.56 ± 0.69	77.90 ± 2.96	**88.42 ± 1.37**
κ (%)	82.59 ± 1.12	86.02 ± 0.38	85.48 ± 0.12	83.73 ± 0.49	84.02 ± 1.46	84.93 ± 0.23	85.10 ± 0.53	**86.11 ± 0.41**

Table 9. The classification results of (a), (b), (c), (d), (e), (f), (g) and the TransHSI method in the Pavia University Dataset.

Class	(a)	(b)	(c)	(d)	(e)	(f)	(g)	TransHSI
1	94.37 ± 1.40	93.24 ± 1.79	94.99 ± 2.30	94.22 ± 1.48	92.96 ± 4.82	93.23 ± 3.35	96.14 ± 0.56	89.58 ± 3.41
2	76.92 ± 2.98	77.88 ± 1.45	81.24 ± 0.75	85.95 ± 1.82	87.68 ± 4.63	81.91 ± 0.65	80.33 ± 0.17	88.59 ± 4.49
3	82.79 ± 1.32	73.96 ± 7.11	75.68 ± 1.38	63.73 ± 11.11	72.51 ± 6.12	80.88 ± 8.26	75.19 ± 2.93	76.53 ± 7.10
4	86.28 ± 3.86	89.09 ± 0.79	90.54 ± 1.38	89.62 ± 2.83	87.17 ± 5.22	87.48 ± 3.23	92.83 ± 0.02	92.07 ± 2.04
5	100.00 ± 0.00	99.73 ± 0.47	99.79 ± 0.36	100.00 ± 0.00	99.88 ± 0.21	99.94 ± 0.10	99.58 ± 0.05	99.91 ± 0.16
6	85.80 ± 4.31	91.06 ± 2.53	82.32 ± 2.62	80.68 ± 4.55	78.84 ± 5.32	90.96 ± 2.91	91.12 ± 0.83	80.75 ± 3.15
7	99.05 ± 0.66	99.22 ± 0.42	99.56 ± 0.15	98.61 ± 0.41	98.33 ± 1.05	99.39 ± 0.18	97.18 ± 0.21	97.96 ± 0.47
8	98.26 ± 0.33	97.61 ± 0.52	96.51 ± 0.74	99.62 ± 0.14	98.08 ± 0.36	95.80 ± 2.77	98.67 ± 0.06	97.11 ± 0.59
9	98.91 ± 0.70	99.54 ± 0.38	99.24 ± 0.33	99.46 ± 0.32	99.20 ± 0.41	99.25 ± 0.58	99.29 ± 0.14	99.54 ± 0.31
OA (%)	85.05 ± 0.86	85.67 ± 0.45	86.57 ± 0.66	88.03 ± 0.08	88.48 ± 0.70	87.53 ± 0.29	87.60 ± 0.14	**89.03 ± 1.72**
AA (%)	91.37 ± 1.02	91.26 ± 1.02	91.10 ± 0.39	90.21 ± 1.35	90.52 ± 1.29	92.09 ± 0.41	**92.26 ± 0.31**	91.34 ± 2.05
κ (%)	80.59 ± 1.02	81.39 ± 0.59	82.36 ± 0.86	84.13 ± 0.11	84.66 ± 0.82	83.68 ± 0.39	83.82 ± 0.16	**85.41 ± 2.16**

Specifically, in the Indian Pines and Pavia University datasets, we observed the following results from the ablation experiments: Method (a) exhibited the lowest accuracies,

indicating that the spectral feature extraction module significantly influences the improvement in the classification performance. Methods (b) and (c) had minimal impacts on the classification results for the Indian Pines dataset, but they both had substantial impacts on the results for the Pavia University dataset, possibly due to differences in the dataset resolution. Method (c), unlike TransHSI, directly used learned features for classification, whereas TransHSI incorporated the features extracted in earlier stages. This suggests that the TransHSI network's fusion of shallow and deep features effectively improves the classification accuracy. Method (d) removed all Transformer Encode modules, making it similar to the HybirdSN and SSRN methods, which primarily rely on CNN architectures. The classification accuracies of method (d) are higher than those of SSRN in the Indian Pines Dataset and slightly lower than those of HybirdSN. On the Pavia University Dataset, the classification accuracies of method (d) outperform those of HybirdSN but are slightly lower than those of SSRN. Its classification performance remains relatively stable across both datasets. In contrast, HybirdSN and SSRN achieved high accuracies on only one dataset each. Furthermore, when compared to the SSFTT and SSTN methods that combine CNN and Transformer modules, method (d) exhibits a slightly lower AA by approximately 1%, but a higher OA by about 1–6% and a higher kappa coefficient by 1–7%. This indicates that, in terms of overall correctly classified samples, method (d) holds a certain advantage over SSFTT and SSTN, which introduce long-range dependencies. However, deeper networks may not fully leverage their advantages if the specific characteristics of HSIs are not adequately considered. For example, when compared to Next-ViT (a method with a network depth of over 400 layers), method (d) still outperforms Next-ViT in all classification metrics on both datasets. Methods (e), (f), and (g), when compared to TransHSI, exhibited varying degrees of reduced accuracy. This suggests that integrating Transformer Encode modules into various components of the TransHSI network improved the extraction of global spectral–spatial information from HSIs, thereby enhancing the classification accuracy. Based on the analysis above, it is evident that each module in the TransHSI network plays a significant role in enhancing the HSI classification performances. Furthermore, it is worth noting that on the Pavia University dataset, methods (a), (f), and (g) achieve higher AA compared to TransHSI. This is because TransHSI performs less ideally in classifying the "Gravel" and "Bare Soil" categories, which are prone to misclassification. Additionally, the imbalance [57] in the sample categories in this dataset leads to TransHSI achieving the highest OA but a slightly lower AA.

5.1.2. Activation Maps Visualization

To further investigate the effectiveness of the TransHSI method and its ablation experiments, this paper conducted a qualitative experiment to visualize their activation maps. As shown in Figure 14, we used two different samples from the Pavia University dataset as inputs to compare the partial activation maps before the final Transformer Encode module. The red portion in the upper-left corner of sample 1 represents buildings, while the green portion in the lower-right corner represents trees. Sample 2 predominantly consists of painted metal sheets. In these two samples, this study observed that the activation maps of TransHSI (Figure 14a,e) exhibited a relatively uniform distribution of brighter areas, with high activation values in specific regions consistent with the targets (e.g., green trees in sample 1). In contrast, the activation maps from the ablation experiments (Figure 14b–d,f–h) displayed a more dispersed and chaotic distribution of high-activation-value areas, indicating a lower consistency with the targets. This is mainly attributed to the network structure of convolutional kernels [43]. When extracting spectral–spatial features, we introduced Transformer Encode modules separately, optimizing the extraction of global spectral–spatial features and key regions in HSIs. These feature visualization comparison results demonstrate that TransHSI can extract more effective spectral–spatial features.

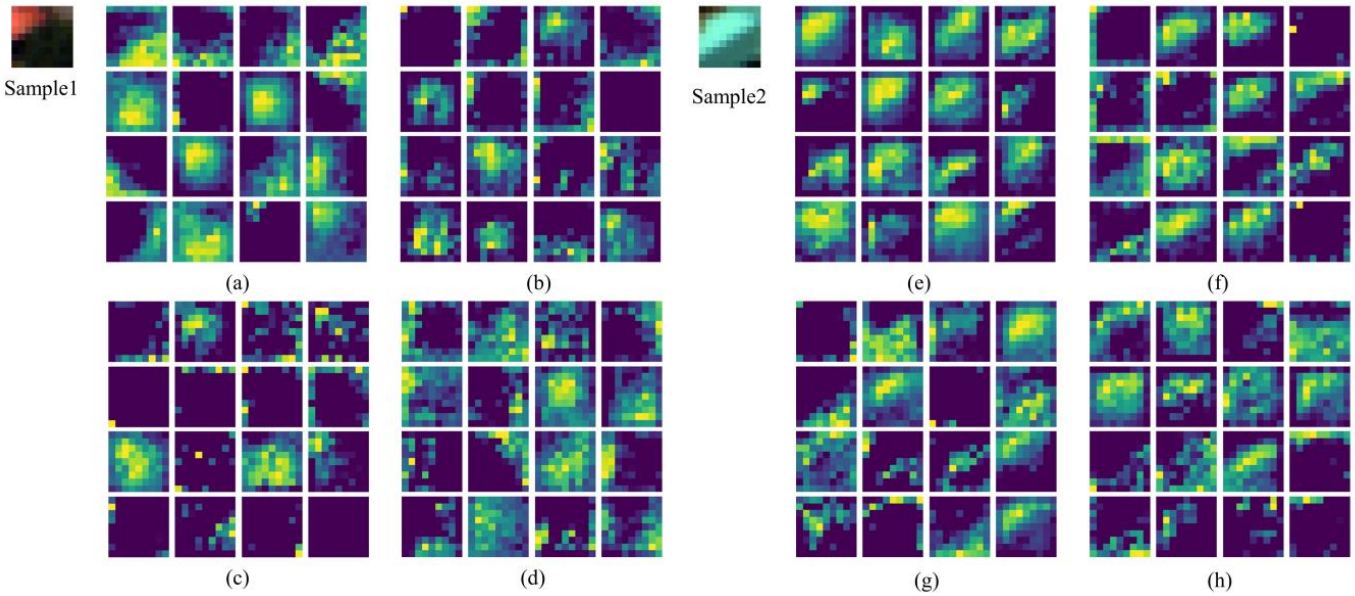


Figure 14. The partial activation maps of two HSI samples on the Pavia University Dataset. (**a**,**e**) represent the activation maps of the TransHSI method; (**b**,**f**) show the activation maps when the spectral feature extraction module is removed; (**c**,**g**) illustrate the activation maps when the spatial feature extraction module is removed; (**d**,**h**) show the activation maps when all Transformer Encode modules are removed. The yellow region corresponds to areas with high activation values, the green region corresponds to regions with moderate activation values, and the blue region corresponds to areas with low activation values.

5.2. Effect of Training Sample Percentages on Classification Results

We conducted experiments on three datasets with the Controlled Random Sampling Strategy [52], selecting different percentages of training samples for our proposed method, TransHSI, and algorithms based on CNNs and Transformer blocks. As shown in Figure 15, it is evident that as the training sample proportion increases, the classification accuracy of all methods improves. In comparison to TransHSI and Next-ViT, the classification performance of SSFTT and SSTN appears less satisfactory. This may be attributed to their failure to fully exploit the spectral–spatial information in HSI. The Next-ViT performs relatively poorly on the Indian Pines dataset when the training sample proportion is 5%, but it shows higher accuracy in other scenarios. Furthermore, in nearly all cases, TransHSI achieves the highest classification accuracy, even with a smaller number of samples, demonstrating the robustness of the proposed method.

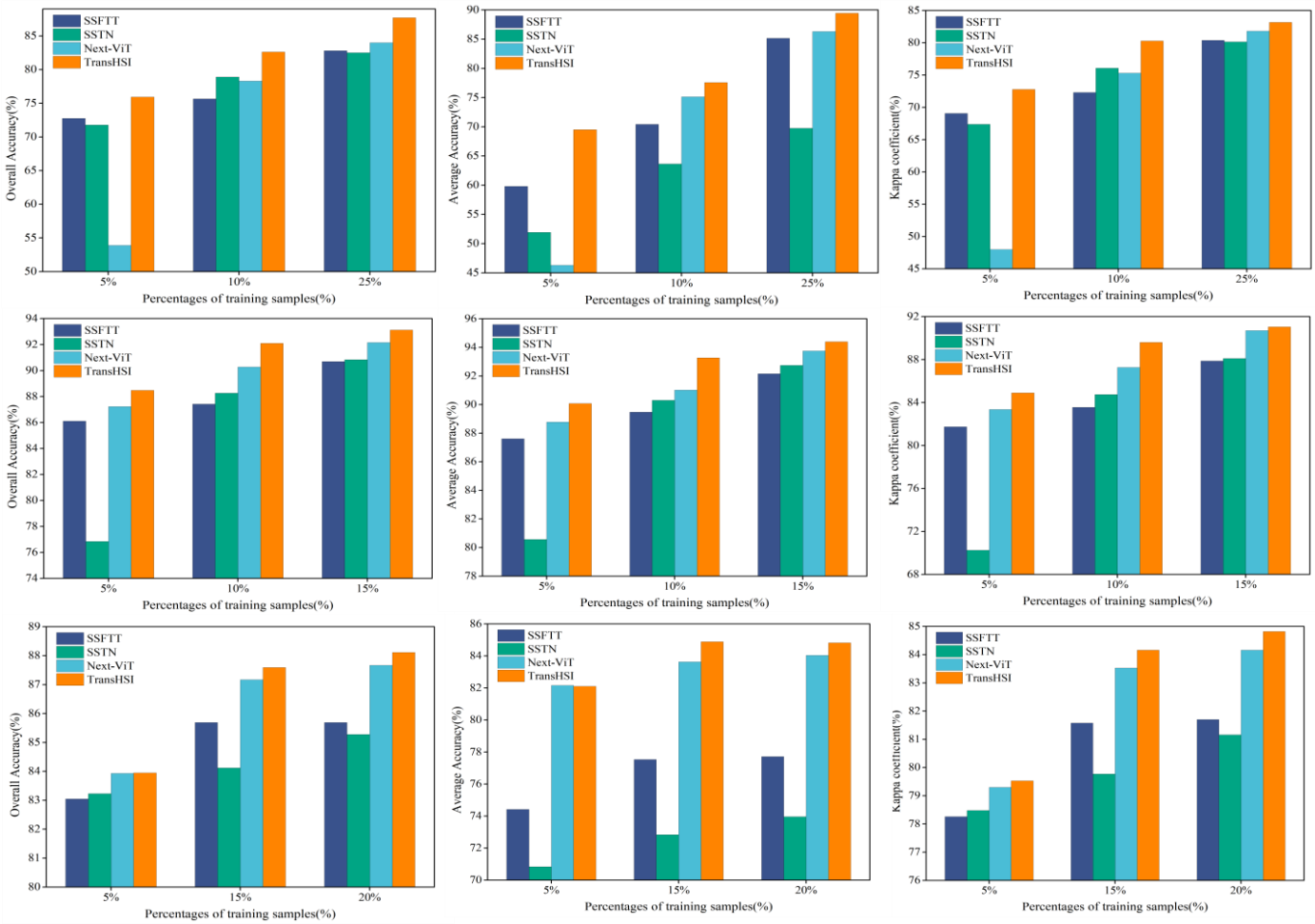


Figure 15. Effect of different percentages (**top**/Indian Pines, **bottom**/DFC2018, and **middle**/Pavia University) of training samples on the classification results.

6. Conclusions

This study developed a novel classification model called TransHSI. By integrating CNN and Transformer modules, we introduce a spectral–spatial feature extraction module designed to extract both global and local spectral and spatial information from HSIs. Additionally, the fusion module combines information from shallow and deep layers and introduces learnable parameters for feature transformation of flattened tokens, enhancing the features' discriminative capacity. Through experiments conducted on three publicly available hyperspectral datasets with disjoint training and test sets, we carried out comprehensive comparisons with other recently proposed HSI classification methods. The results demonstrate that TransHSI excels, particularly on the Indian Pines, a low-resolution, small-sample dataset, where it achieves the best performance regarding OA, AA, and kappa coefficient. On the high-resolution, large-sample datasets, Pavia University and DFC2018, TransHSI still exhibits a competitive advantage, with AAs comparable to those of other leading methods and the highest OAs and kappa coefficients, showcasing comparable results to other methods. Furthermore, the ablation experiments and feature visualization results further validate the designed model's effectiveness. In future research, we will optimize the network using lightweight module structures to reduce network complexity and enhance performance. Additionally, we will explore more neural network architectures and methods to further improve the accuracy and efficiency of HSI classification.

Author Contributions: All the authors made significant contributions to this work. Conceptualization, P.Z. and H.Y.; Methodology, P.Z. and H.Y.; Software, R.W.; Supervision, H.Y.; Validation, P.L.; Visualization, P.L.; Writing—Original Draft, P.Z.; Writing—Review and Editing, H.Y. and R.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (U1304402, 41977284) and Natural Science and Technology Project of Natural Resources Department of Henan Province (2019-378-16).

Data Availability Statement: The Indian Pines and the University of Pavia datasets are available online at http://dase.grss-ieee.org/, accessed on 6 October 2022. The Data Fusion Contest 2018 is available online at https://hyperspectral.ee.uh.edu/?page_id=1075, accessed on 10 October 2022. The source code for the TransHSI method is available at https://github.com/zpfx3/TransHSI, accessed on 6 March 2023.

Acknowledgments: The authors would like to thank the authors of all references used in this paper, the editors, and the anonymous reviewers for their detailed comments and suggestions.

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

References

1. Audebert, N.; Le Saux, B.; Lefevre, S. Deep Learning for Classification of Hyperspectral Data: A Comparative Review. *IEEE Geosci. Remote Sens. Mag.* **2019**, *7*, 159–173. [CrossRef]
2. Bhosle, K.; Musande, V. Evaluation of Deep Learning CNN Model for Land Use Land Cover Classification and Crop Identification Using Hyperspectral Remote Sensing Images. *J. Indian Soc. Remote Sens.* **2019**, *47*, 1949–1958. [CrossRef]
3. Fong, A.; Shu, G.; McDonogh, B. Farm to Table: Applications for New Hyperspectral Imaging Technologies in Precision Agriculture, Food Quality and Safety. In Proceedings of the Conference on Lasers and Electro-Optics, Washington, DC, USA, 10–15 May 2020; p. AW3K.2.
4. Lu, B.; Dao, P.D.; Liu, J.G.; He, Y.H.; Shang, J.L. Recent Advances of Hyperspectral Imaging Technology and Applications in Agriculture. *Remote Sens.* **2020**, *12*, 2659. [CrossRef]
5. Wang, Q.; Yuan, Z.; Du, Q.; Li, X. GETNET: A General End-to-End 2-D CNN Framework for Hyperspectral Image Change Detection. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 3–13. [CrossRef]
6. Zhan, T.; Song, B.; Sun, L.; Jia, X.; Wan, M.; Yang, G.; Wu, Z. TDSSC: A Three-Directions Spectral–Spatial Convolution Neural Network for Hyperspectral Image Change Detection. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 377–388. [CrossRef]
7. Zeng, D.; Zhang, S.; Chen, F.S.; Wang, Y.M. Multi-Scale CNN Based Garbage Detection of Airborne Hyperspectral Data. *IEEE Access* **2019**, *7*, 104514–104527. [CrossRef]
8. Lowe, A.; Harrison, N.; French, A.P. Hyperspectral image analysis techniques for the detection and classification of the early onset of plant disease and stress. *Plant Methods* **2017**, *13*, 80. [CrossRef]
9. Peyghambari, S.; Zhang, Y. Hyperspectral remote sensing in lithological mapping, mineral exploration, and environmental geology: An updated review. *J. Appl. Remote Sens.* **2021**, *15*, 031501. [CrossRef]
10. 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]
11. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [CrossRef]
12. Kang, X.; Li, S.; Fang, L.; Li, M.; Benediktsson, J.A. Extended random walker-based classification of hyperspectral images. *IEEE Trans. Geosci. Remote Sens.* **2014**, *53*, 144–153. [CrossRef]
13. Farrell, M.D.; Mersereau, R.M. On the impact of PCA dimension reduction for hyperspectral detection of difficult targets. *IEEE Geosci. Remote Sens. Lett.* **2005**, *2*, 192–195. [CrossRef]
14. Menon, V.; Du, Q.; Fowler, J.E. Fast SVD With Random Hadamard Projection for Hyperspectral Dimensionality Reduction. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 1275–1279. [CrossRef]
15. Jia, S.; Zhao, Q.; Zhuang, J.; Tang, D.; Long, Y.; Xu, M.; Zhou, J.; Li, Q. Flexible Gabor-Based Superpixel-Level Unsupervised LDA for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 10394–10409. [CrossRef]
16. Falco, N.; Benediktsson, J.A.; Bruzzone, L. A Study on the Effectiveness of Different Independent Component Analysis Algorithms for Hyperspectral Image Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2183–2199. [CrossRef]
17. Wang, Y.; Yu, W.; Fang, Z. Multiple Kernel-Based SVM Classification of Hyperspectral Images by Combining Spectral, Spatial, and Semantic Information. *Remote Sens.* **2020**, *12*, 120. [CrossRef]
18. Hu, W.; Huang, Y.; Wei, L.; Zhang, F.; Li, H. Deep Convolutional Neural Networks for Hyperspectral Image Classification. *J. Sens.* **2015**, *2015*, 258619. [CrossRef]
19. 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]

20. 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]
21. Zhong, Z.; Li, J.; Luo, Z.; Chapman, M. Spectral–Spatial Residual Network for Hyperspectral Image Classification: A 3-D Deep Learning Framework. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 847–858. [CrossRef]
22. 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]
23. Hang, R.L.; Liu, Q.S.; Hong, D.F.; Ghamisi, P. Cascaded Recurrent Neural Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 5384–5394. [CrossRef]
24. Mou, L.C.; Ghamisi, P.; Zhu, X.X. Deep Recurrent Neural Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3639–3655. [CrossRef]
25. Hao, S.Y.; Wang, W.; Salzmann, M. Geometry-Aware Deep Recurrent Neural Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 2448–2460. [CrossRef]
26. Ding, Y.; Zhang, Z.; Zhao, X.; Hong, D.; Cai, W.; Yu, C.; Yang, N.; Cai, W. Multi-feature fusion: Graph neural network and CNN combining for hyperspectral image classification. *Neurocomputing* **2022**, *501*, 246–257. [CrossRef]
27. Hong, D.; Gao, L.; Yao, J.; Zhang, B.; Plaza, A.; Chanussot, J. Graph Convolutional Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 5966–5978. [CrossRef]
28. Mou, L.; Lu, X.; Li, X.; Zhu, X.X. Nonlocal Graph Convolutional Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 8246–8257. [CrossRef]
29. Wan, S.; Gong, C.; Zhong, P.; Du, B.; Zhang, L.; Yang, J. Multiscale Dynamic Graph Convolutional Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 3162–3177. [CrossRef]
30. Wang, J.; Guo, S.; Huang, R.; Li, L.; Zhang, X.; Jiao, L. Dual-Channel Capsule Generation Adversarial Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5501016. [CrossRef]
31. Zhan, Y.; Hu, D.; Wang, Y.; Yu, X. Semisupervised Hyperspectral Image Classification Based on Generative Adversarial Networks. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 212–216. [CrossRef]
32. 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]
33. He, J.; Zhao, L.; Yang, H.; Zhang, M.; Li, W. HSI-BERT: Hyperspectral Image Classification Using the Bidirectional Encoder Representation from Transformers. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 165–178. [CrossRef]
34. 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]
35. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762. [CrossRef]
36. Fırat, H.; Asker, M.E.; Hanbay, D. Classification of hyperspectral remote sensing images using different dimension reduction methods with 3D/2D CNN. *Remote Sens. Appl. Soc. Environ.* **2022**, *25*, 100694. [CrossRef]
37. Zhang, X.; Sun, G.; Jia, X.; Wu, L.; Zhang, A.; Ren, J.; Fu, H.; Yao, Y. Spectral-Spatial Self-Attention Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5512115. [CrossRef]
38. Ge, H.M.; Wang, L.G.; Liu, M.Q.; Zhu, Y.X.; Zhao, X.Y.; Pan, H.Z.; Liu, Y.Z. Two-Branch Convolutional Neural Network with Polarized Full Attention for Hyperspectral Image Classification. *Remote Sens.* **2023**, *15*, 848. [CrossRef]
39. Sun, H.; Zheng, X.T.; Lu, X.Q.; Wu, S.Y. Spectral-Spatial Attention Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 3232–3245. [CrossRef]
40. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929. [CrossRef]
41. Roy, S.K.; Deria, A.; Hong, D.; Rasti, B.; Plaza, A.; Chanussot, J. Multimodal Fusion Transformer for Remote Sensing Image Classification. *arXiv* **2022**, arXiv:2203.16952. [CrossRef]
42. Yang, L.; Yang, Y.; Yang, J.; Zhao, N.; Wu, L.; Wang, L.; Wang, T. FusionNet: A Convolution-Transformer Fusion Network for Hyperspectral Image Classification. *Remote Sens.* **2022**, *14*, 4066. [CrossRef]
43. He, X.; Chen, Y.; Lin, Z. Spatial-Spectral Transformer for Hyperspectral Image Classification. *Remote Sens.* **2021**, *13*, 498. [CrossRef]
44. 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]
45. Zhong, Z.; Li, Y.; Ma, L.; Li, J.; Zheng, W.-S. Spectral–Spatial Transformer Network for Hyperspectral Image Classification: A Factorized Architecture Search Framework. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5514715. [CrossRef]
46. Li, J.; Xia, X.; Li, W.; Li, H.; Wang, X.; Xiao, X.; Wang, R.; Zheng, M.; Pan, X. Next-ViT: Next Generation Vision Transformer for Efficient Deployment in Realistic Industrial Scenarios. *arXiv* **2022**, arXiv:2207.05501. [CrossRef]
47. Firat, H.; Asker, M.E.; Bayindir, M.İ.; Hanbay, D. 3D residual spatial–spectral convolution network for hyperspectral remote sensing image classification. *Neural Comput. Appl.* **2022**, *35*, 4479–4497. [CrossRef]
48. Ahmad, M.; Ghous, U.; Hong, D.; Khan, A.M.; Yao, J.; Wang, S.; Chanussot, J. A Disjoint Samples-Based 3D-CNN With Active Transfer Learning for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5539616. [CrossRef]

49. Zhang, F.; Yan, M.; Hu, C.; Ni, J.; Zhou, Y. Integrating Coordinate Features in CNN-Based Remote Sensing Imagery Classification. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 5502505. [CrossRef]
50. Cao, X.; Liu, Z.; Li, X.; Xiao, Q.; Feng, J.; Jiao, L. Nonoverlapped Sampling for Hyperspectral Imagery: Performance Evaluation and a Cotraining-Based Classification Strategy. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5506314. [CrossRef]
51. Geib, C.; Aravena Pelizari, P.; Schrade, H.; Brenning, A.; Taubenbock, H. On the Effect of Spatially Non-Disjoint Training and Test Samples on Estimated Model Generalization Capabilities in Supervised Classification with Spatial Features. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 2008–2012. [CrossRef]
52. Liang, J.; Zhou, J.; Qian, Y.; Wen, L.; Bai, X.; Gao, Y. On the Sampling Strategy for Evaluation of Spectral-Spatial Methods in Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 862–880. [CrossRef]
53. Wang, W.; Dai, J.; Chen, Z.; Huang, Z.; Li, Z.; Zhu, X.; Hu, X.-h.; Lu, T.; Lu, L.; Li, H.; et al. InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–24 June 2022; pp. 14408–14419.
54. Ahmad, M.; Shabbir, S.; Roy, S.K.; Hong, D.; Wu, X.; Yao, J.; Khan, A.M.; Mazzara, M.; Distefano, S.; Chanussot, J. Hyperspectral Image Classification-Traditional to Deep Models: A Survey for Future Prospects. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 968–999. [CrossRef]
55. Van der Maaten, L.; Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
56. Liu, M.; Pan, H.; Ge, H.; Wang, L. MS3Net: Multiscale stratified-split symmetric network with quadra-view attention for hyperspectral image classification. *Signal Process.* **2023**, *212*, 109153. [CrossRef]
57. 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]