remote sensing

MDPI

Article

A Novel Transformer Network with a CNN-Enhanced Cross-Attention Mechanism for Hyperspectral Image Classification

Xinyu Wang [1], Le Sun [1,2], Chuhan Lu [3] and Baozhu Li [4,*]

[1] School of Computer Science, Nanjing University of Information Science and Technology, Nanjing 210044, China; cs_xywang@nuist.edu.cn (X.W.); sunlecncom@nuist.edu.cn (L.S.)
[2] Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science and Technology, Nanjing 210044, China
[3] School of Atmospheric Sciences, Nanjing University of Information Science and Technology, Nanjing 210044, China; luchuhan@nuist.edu.cn
[4] Internet of Things & Smart City Innovation Platform, Zhuhai Fudan Innovation Institute, Zhuhai 519031, China
* Correspondence: baozhuli@fudan-zhuhai.org.cn

Abstract: Recently, with the remarkable advancements of deep learning in the field of image processing, convolutional neural networks (CNNs) have garnered widespread attention from researchers in the domain of hyperspectral image (HSI) classification. Moreover, due to the high performance demonstrated by the transformer architecture in classification tasks, there has been a proliferation of neural networks combining CNNs and transformers for HSI classification. However, the majority of the current methods focus on extracting spatial–spectral features from the HSI data of a single size for a pixel, overlooking the rich multi-scale feature information inherent to the data. To address this problem, we designed a novel transformer network with a CNN-enhanced cross-attention (TNCCA) mechanism for HSI classification. It is a dual-branch network that utilizes different scales of HSI input data to extract shallow spatial–spectral features using a multi-scale 3D and 2D hybrid convolutional neural network. After converting the feature maps into tokens, a series of 2D convolutions and dilated convolutions are employed to generate two sets of Q (queries), K (keys), and V (values) at different scales in a cross-attention module. This transformer with CNN-enhanced cross-attention explores multi-scale CNN-enhanced features and fuses them from both branches. Experimental evaluations conducted on three widely used hyperspectral image (HSI) datasets, under the constraint of limited sample size, demonstrate excellent classification performance of the proposed network.

Keywords: convolutional neural network (CNN); hyperspectral image classification; transformer; multi-scale feature

check for updates

Citation: Wang, X.; Sun, L.; Lu, C.; Li, B. A Novel Transformer Network with a CNN-Enhanced Cross-Attention Mechanism for Hyperspectral Image Classification. *Remote Sens.* **2024**, *16*, 1180. https://doi.org/10.3390/rs16071180

Academic Editor: Farid Melgani

Received: 23 January 2024
Revised: 13 March 2024
Accepted: 26 March 2024
Published: 28 March 2024

1. Introduction

Hyperspectral imaging (HSI) has emerged as a powerful technique for remote sensing and the analysis of the Earth's surface [1,2]. By capturing and analyzing a large number of narrow and contiguous spectral bands, HSI data provides rich and detailed information about the composition and properties of observed objects [3,4]. The ability to differentiate between different land cover types and detect subtle variations in materials has made HSI classification a crucial task in various fields, including agriculture [5], environmental monitoring [6], mineral exploration [7], and military reconnaissance [8]. HSI classification has become a hot research topic [9–13].

Currently, several HSI classification methods based on traditional machine learning algorithms have been proposed. These methods include Support Vector Machines (SVMs) [14,15] and Random Forest (RF) [16]. In addition, the k-Nearest Neighbors (k-NN) [17] algorithm is a non-parametric classification method that is based on the assump-

tion of similar feature values. It assigns the class of an unlabeled pixel as the most frequent class among its k-nearest-neighboring pixels in the feature space. Linear Discriminant Analysis (LDA) [18] is a supervised dimensionality reduction and classification algorithm. It aims to find a linear transformation that maximizes the differences between different classes and minimizes the within-class scatter, resulting in discriminative features used for pixel classification. The Endmember Extraction and Classification Algorithm (EMAP) [19] is a comprehensive algorithm that combines endmember extraction and classification in hyperspectral image analysis. It involves extracting endmembers, which are pure spectral signatures, and using a linear mixing model to classify pixels based on their linear combinations of endmembers. EMAP enables the accurate characterization of materials present in hyperspectral data.

Traditional machine learning methods for hyperspectral classification have limitations in feature extraction, high-dimensional data, and modeling nonlinear relationships [20]. In contrast, deep learning offers advantages such as automatic feature learning, strong nonlinear modeling capabilities, compact data representation, and data augmentation for improved generalization [21]. These benefits make deep learning well-suited to handling high-dimensional, nonlinear, and complex hyperspectral data, leading to enhanced classification accuracy and robustness.

Due to the popularity of deep learning research, deep learning methods have also been applied to HSI classification tasks. Initially, researchers only used convolutional layers to solve classification tasks, such as 1D-CNN [22], 2D-CNN [23], and 3D-CNN [24]. However, more complex and deeper networks have been designed. He et al. [25] discovered that HSI differed significantly from 3D object images due to its combination of 2D spatial and 1D spectral features. Existing deep neural networks cannot be directly applied to HSI classification tasks. To address this issue, they proposed a Multiscale 3D Deep Convolutional Neural Network (M3D-CNN), which jointly learned both two-dimensional multiscale spatial features and one-dimensional spectral features from HSI data in an end-to-end manner. To achieve better classification performance by combining two types of convolutions, Roy et al. [26] effectively combined 3D-CNN with 2D-CNN. Zhu et al. [27] discovered the remarkable capabilities of Generative Adversarial Networks (GANs) in various applications. As a result, they explored the application of GANs in the field of HSI classification and designed a CNN for discriminating samples and another CNN for generating synthetic input samples. Their approach achieved superior classification accuracy compared to previous methods. Due to the sequential nature of hyperspectral pixels, Mou et al. [28] applied Recurrent Neural Networks (RNNs) to HSI classification tasks. Then, they proposed a novel RNN model that effectively analyzed HSI pixels as sequential data. Their research demonstrated the significant potential of RNNs in HSI classification tasks. Traditional CNN models can only capture fixed receptive fields for HSI, making it challenging to extract feature information with different object distributions. To address this issue, Wan et al. [29] applied Graph Convolutional Networks (GCNs) to HSI classification tasks. They designed a multi-scale dynamic GCN (MDGCN) that updated the graph dynamically during the convolution process, leveraging multiscale features in HSI.

With the introduction of attention mechanisms, Haut et al. [30] combined CNNs and Residual Networks (ResNets) with visual attention. Visual attention effectively assisted in identifying the most representative parts of the data. Experimental results demonstrated that deep attention models had a strong competitive advantage. Sun et al. [31] discovered that CNN-based methods, due to the presence of interfering pixels, weaken the discriminative power of spatial–spectral features. Hence, they proposed a Spectral–Spatial Attention Network (SSAN) that captured discriminative spatial–spectral features from attention areas in HSI. To leverage the diverse spatial–spectral features inherent in different regions of the training data, Hang et al. [32] proposed a novel attention-aided CNN. It consisted of two subnetworks responsible for extracting spatial and spectral features, respectively. Both subnetworks incorporated attention modules to assist in constructing a discriminative network. To mitigate the interference between spatial and spectral features during

the extraction process, Ma et al. [33] designed a Double-Branch Multi-Attention mechanism network (DBMA). It employed two branches, each focusing on extracting spatial and spectral features, respectively, thereby reducing mutual interference. Subsequently, Zhu et al. [34] discovered that the equal treatment of all spectral bands using deep neural networks restricted feature learning and was not conducive to classification performance in HSI. Therefore, they proposed a Residual Spectral–Spatial Attention Network (RSSAN) to address this issue. The RSSAN took raw 3D cubes as input data and employed spectral attention and spatial attention to suppress irrelevant components and emphasize relevant components, achieving adaptive feature refinement.

Recently, with the introduction of Vision Transformer [35] into image processing, which originated from the transformer model in natural language processing, more and more efficient transformer structures have been designed [36]. To fully exploit the sequential properties inherent in the spectral feature of HSI, Hong et al. [37] proposed a new classification network called SpectralFormer. It can learn the spectral sequence information. Similarly, He et al. [38] also addressed this issue and designed a classification framework called Spatial–Spectral Transformer to capture the sequential spectral relationships in HSI. Due to the limited ability of CNN to capture deep semantic features, Sun et al. [39] discovered that transformer structures can effectively complement this drawback. They proposed a method called Spectral–Spatial Feature Tokenization Transformer (SSFTT). It combined CNNs and transformers to extract abundant spectral–spatial features. Mei et al. [40] found that the features extracted using the current transformer structures exhibited excessive discretization and, thus, proposed a Group-Aware Hierarchical Transformer (GAHT) based on group perception. This network used a hierarchical structure and achieved a significant improvement in classification performance. Fang et al. [41] introduced a Multi-Attention Joint Representation with Lightweight Transformer (MAR-LWFormer) for scenarios with extremely limited samples. They employed a three-branch structure to extract multi-scale features and demonstrated excellent classification performance. To utilize morphological features, Roy et al. [42] proposed a novel transformer (morphFormer) that combined morphological convolutional operations with attention mechanisms.

In the current research, most models are capable of effectively extracting spatial–spectral information from HSI. However, training on fixed-size sample cubes constrained the model's ability to extract multi-scale features. Additionally, in practical applications, there is often a scarcity of labeled samples in HSI datasets [43]. Therefore, it is crucial to develop a network model that can adequately extract spatial–spectral features from HSI even in scenarios with limited samples.

The TNCCA model proposed by us offers the following three main contributions:

- Taking blocks of different sizes from HSI, we employ a mixed fusion multi-scale extraction shallow spatial–spectral feature module to process shallow features. This module primarily consists of two multi-scale convolutional neural networks designed for different-sized data. The network utilizes convolutional kernels of varying sizes to extract shallow feature information at different scales.
- An efficient transformer encoder was designed in which we apply 2D convolution and dilated convolution to tokens to obtain two sets of Q, K, and V with different scale information. This enables the transformer architecture with cross-attention to not only learn deeper feature information and promote the interaction of deep semantic information but also effectively fuse feature information of different sizes from the two branches.
- We designed an innovative dual-branch network specifically for classification tasks in small-sample scenarios. This network efficiently integrates a multi-scale CNN with a transformer encoder to fully exploit the multi-scale spatial–spectral features of HSI. We validated this network on three datasets, and the experimental results indicated that our proposed network was competitive compared to state-of-the-art methods.

2. Materials and Methods

In Figure 1, we illustrate an overview diagram of the proposed TNCCA model, which is an efficient dual-branch deep learning network for HSI classification. The network consists of the following sub-modules: the data preprocessing module for HSI, the shallow feature extraction module that utilizes different fusion methods to combine multi-scale spatial–spectral features, the module that converts the shallow features into tokens with different quantities assigned to different sizes, and the transformer module with CNN-enhanced cross-attention. Finally, there is the classifier head, which takes the input pixels and outputs the corresponding classification labels.

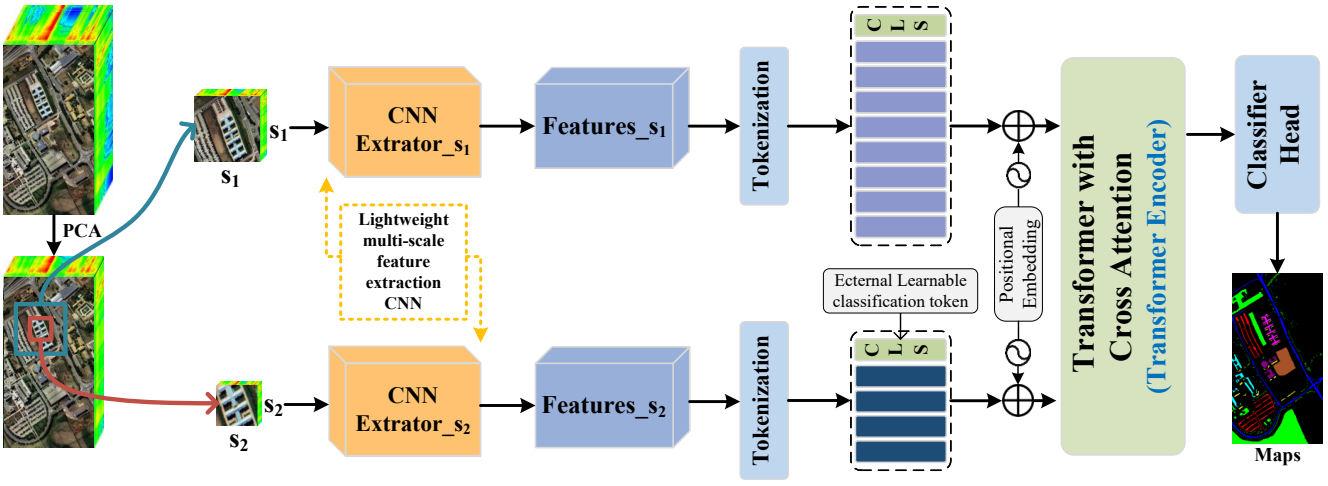

Figure 1. Overview diagram of the proposed TNCCA model.

In summary, the TNCCA model consists of the following five components: HSI data preprocessing, a dual-branch multi-scale shallow feature extraction module, a feature-maps-to-tokens conversion module, a transformer with a CNN-enhanced cross-attention module, and a classifier head.

2.1. HSI Data Preprocessing

The processing of the original HSI ($\boldsymbol{X} \in \mathbb{R}^{a \times b \times l}$) is described in this section, where a and b represent different spatial sizes, and l represents the spectral dimension. Due to the typically large number of spectral dimensions in HSI, it increases computational complexity and consumes significant computational resources. Therefore, we use the PCA operation to solve this problem by reducing the dimensionality of the original image from l to r.

To obtain information at different scales, we extract two square patches of different sizes, $\boldsymbol{X}_1^p \in \mathbb{R}^{s_1 \times s_1 \times r}$ and $\boldsymbol{X}_2^p \in \mathbb{R}^{s_2 \times s_2 \times r}$ ($s_1 > s_2$), centered at each pixel. We combine these two variables into a dataset and feed it into the network together. Finally, the set of data generated via each pixel is placed into a collection, A, and the training and test sets are randomly partitioned from A based on the sampling rate. Each group of training and testing data contains the corresponding ground truth labels. The labels, denoted as $\boldsymbol{Y} \in \mathbb{R}^{a \times b}$, are obtained from the set of ground truth labels.

2.2. Dual-Branch Multi-Scale Shallow Feature Extraction Module

As shown in Figure 2, a group of cubes, denoted as $\boldsymbol{X}_1^p$ and $\boldsymbol{X}_2^p$, with different sizes are fed into the network. Firstly, they pass through a 3D convolutional layer. In the first branch, a larger-sized cube is processed, and 8 convolutional kernels are allocated. The size of each kernel is ($3 \times 5 \times 5$). In the second branch, a cube with smaller dimensions is processed, and 4 convolutional kernels are allocated. The size of each kernel is ($1 \times 3 \times 3$).

To maintain the original size of the cubes, padding is applied. The above process can be represented in the following equation:

$$X_1^{3d} = Conv3D_{(3\times5\times5)}(X_1^p) \qquad X_2^{3d} = Conv3D_{(1\times3\times3)}(X_2^p) \tag{1}$$

where Conv3D and Conv2D represent 3D convolutional layers and 2D convolutional layers with different kernel sizes, respectively.

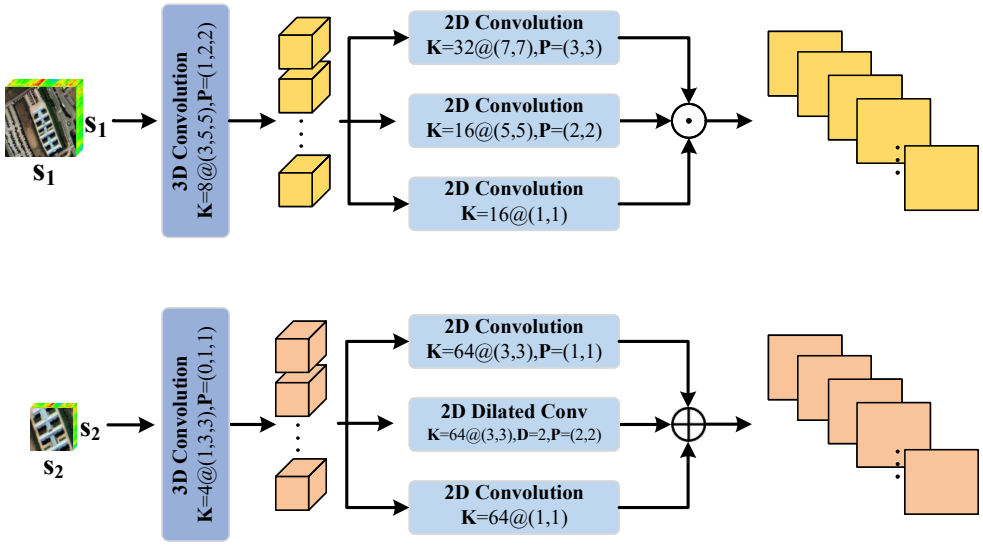

Figure 2. Dual-branch multi-scale shallow feature extraction module.

After passing through a 3D convolutional layer, we extract shallow spatial features at different scales using multi-scale 2D convolutional layers. Similarly, we use different numbers of convolutional kernels and different kernel sizes in different branches. In the first branch, we use 32 2D convolutional kernels of size (7×7), 16 kernels of size (5×5), and 16 kernels of size (1×1). The information from these three different scales is fused through the Concatenation operation. In the second branch, smaller kernel sizes are used to extract shallow spatial features. Specifically, we use 64 2D convolutional kernels of size (3×3), 64 2D dilated convolutional kernels with a dilation rate of 2 and size (3×3), and 64 2D convolutional kernels of size (1×1). The information from these three different scales is fused through element-wise addition.

Finally, we obtain two sets of 2D features, F_1 and F_2, respectively. This process can be represented in the following equations:

$$\begin{aligned} F_1 &= Conv2D_{(7\times7)}(X_1^{3d}) \odot Conv2D_{(5\times5)}(X_1^{3d}) \odot Conv2D_{(1\times1)}(X_1^{3d}) \\ F_2 &= Conv2D_{(3\times3)}(X_2^{3d}) \oplus Dilated\ Conv2D_{(3\times3)}(X_2^{3d}) \oplus Conv2D_{(1\times1)}(X_2^{3d}) \end{aligned} \tag{2}$$

2.3. Feature-Maps-to-Tokens Conversion Module

After obtaining the multi-scale 2D feature information from the dual-branch shallow feature extraction module, in order to better adapt to the structure of the Transformer, these features need to be tokenized.

The flattened feature maps are denoted as F_1^{flat} and F_2^{flat}, respectively. These two variables can be represented in the following equation:

$$F_1^{flat} = \mathcal{TS}(Flatten(F_1)) \qquad F_2^{flat} = \mathcal{TS}(Flatten(F_2)) \tag{3}$$

where $\mathcal{TS}(\cdot)$ is a transpose function. Next, F_1^{flat} is multiplied by a learnable weight matrix W_1 using a 1×1 operation, and similarly, F_2^{flat} is multiplied by a learnable weight matrix W_2 using a 1×1 operation. We use weight matrices of different shapes to achieve the purpose of assigning a different number of tokens. Then, the feature maps are transformed into feature tokens multiplied by themselves. The above process can be achieved using the following equation:

$$T_1^f = (softmax(\mathcal{TS}(F_1^{flat} W_1)))F_1^{flat} \qquad T_2^f = (softmax(\mathcal{TS}(F_2^{flat} W_2)))F_2^{flat} \tag{4}$$

To accomplish the classification task, we also embed a learnable classification token consisting of all zeros. Then, to preserve the original positional information, positional information is embedded into the tokens. The tokens of the two branches can be obtained from the following equation:

$$T_1 = (T_1^f \odot T_1^{cls}) \oplus T_1^f \qquad T_2 = (T_2^f \odot T_2^{cls}) \oplus T_2^f \tag{5}$$

2.4. Transformer with CNN-Enhanced Cross-Attention Module

The transformer possesses powerful feature-information-mining capabilities, as it can capture long-range dependencies and acquire global contextual information. To further explore the deep feature information contained in the data and fully integrate the multi-scale feature information extracted via the two branches, we embed a cross-attention in the transformer structure.

As shown in Figure 3, We utilize different convolutional layers to obtain the attention mechanism's **Q**, **K**, and **V** tensors from one of the outputs T_1 obtained from the previous module. Firstly, we apply a 2D convolutional layer with kernel sizes of (3×3) and padding of 1 to obtain Q_1. Next, a 2D convolutional layer with kernel sizes of (5×5) and padding of 2 is used to obtain K_1. Finally, we employ a dilated convolutional layer with kernel sizes of (3×3), padding of 2, and a dilation rate of 2 to obtain V_1.

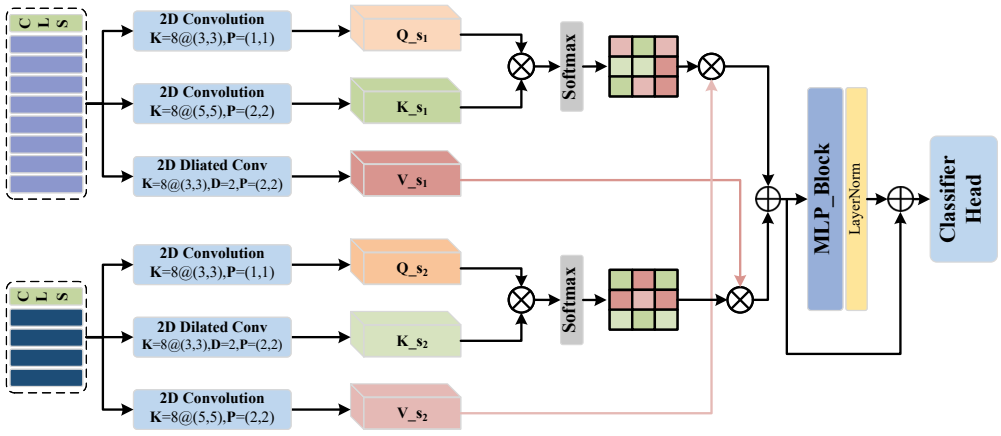

Figure 3. Transformer with CNN-enhanced cross-attention module.

Next, we apply similar multi-scale convolutions to another output, T_2, to obtain Q_2, K_2, and V_2. Firstly, we use a 2D convolutional layer with a kernel size of (3×3) and padding of 1 to obtain Q_2. Then, we employ a dilated convolutional layer with a kernel size of (3×3), padding of 2, and a dilation rate of 2 to obtain K_2. Finally, we utilize a 2D convolutional layer with a kernel size of (5×5) and padding of 2 to obtain V_2. Once we have obtained these tensors, we perform element-wise multiplication among them to

obtain deep features $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ that have undergone the attention mechanism. The process can be represented in the following formula:

$$\boldsymbol{A}_1 = softmax\left(\frac{\boldsymbol{Q_1}(\boldsymbol{\mathcal{TS}}(\boldsymbol{K_1}))}{\sqrt{d_{K_1}}}\right)\boldsymbol{V_2} \tag{6}$$

$$\boldsymbol{A}_2 = softmax\left(\frac{\boldsymbol{Q_2}(\boldsymbol{\mathcal{TS}}(\boldsymbol{K_2}))}{\sqrt{d_{K_2}}}\right)\boldsymbol{V_1} \tag{7}$$

where d_{K_1} is the dimension of $\boldsymbol{K}_1$, and d_{K_2} is the dimension of $\boldsymbol{K}_2$. We obtain the deep features from two branches and sum them pixel-wise. Then, we pass the summed features through a multi-layer perceptron block using a residual structure to obtain the final deep feature, $\boldsymbol{DF}$. This can be obtained using the following equation:

$$\boldsymbol{DF} = \boldsymbol{LN}[\boldsymbol{MLP}[\boldsymbol{A}_1 \oplus \boldsymbol{A}_2]] \oplus (\boldsymbol{A}_1 \oplus \boldsymbol{A}_2) \tag{8}$$

where $\boldsymbol{MLP}[\cdot]$ is the multi-layer perceptron, and $\boldsymbol{LN}$ is the abbreviation for layer normalization. The MLP mainly includes two linear layers, with the addition of the Gaussian Error Linear Unit (GELU) activation function in between.

2.5. Classifier Head

We extract the learnable classification token, $\boldsymbol{T}_{cls}^{DF}$, from the output tokens, $\boldsymbol{DF}$, of the transformer encoder. Then, we pass it through a linear layer to obtain a one-dimensional vector, denoted as $\boldsymbol{I} \in \mathbb{R}^{1\times c}$, where c represents the number of classes. The softmax function is used to ensure that the total activation of each output unit is 1. By selecting the corresponding maximum value, we obtain the class label for that pixel. The entire process can be represented in the following equation:

$$Label = max(\underbrace{Softmax(Linear(\boldsymbol{T}_{cls}^{DF}))}_{\boldsymbol{I}}) \tag{9}$$

The complete procedure of the TNCCA method, as proposed, is outlined in Algorithm 1.

Algorithm 1 Multi-scale Feature Transformer with CNN-Enhanced Cross-Attention Model

Input: Input HSI data $\boldsymbol{X} \in \mathbb{R}^{a\times b\times l}$ and ground truth labels $\boldsymbol{Y} \in \mathbb{R}^{a\times b}$; the original data are reduced in spectral dimension to r = 30 using PCA operation. A set of small cubes with sizes s_1 = 13 and s_2 = 7 is then extracted. Subsequently, the training set of the model is randomly sampled at a sampling rate of 1%.

Output: Predicted labels for the test dataset.

1: Set the batch size of the training data to 64, and use the Adam optimizer with a learning rate of $lr = 5 \times 10^{-4}$. Decay the learning rate to lr * 0.9 every 50 steps. Set the total number of training epochs to ϵ = 500.
2: After the dimensionality reduction of the original HSI using PCA, cubes corresponding to each pixel are extracted with the pixel as the center. Subsequently, each extracted set of data, $\boldsymbol{X}_1^p$ and $\boldsymbol{X}_2^p$, is placed into a collection. Then, the collection is divided into a training set and a testing set according to Table 1.
3: Create training and test data loaders. Each group of training and testing data will obtain corresponding ground truth labels from $\boldsymbol{Y}$.
4: **for** $i = 1$ to ϵ **do**
5: The dual-branch, multi-scale shallow feature extraction module is used to extract the multi-scale shallow spatial–spectral features $\boldsymbol{F}_1$ and $\boldsymbol{F}_2$.
6: The outputs of the feature maps to the token conversion module are used as inputs for the next module, denoted as $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$.
7: Passing tokens through a transformer encoder with cross-attention yields deep semantic features, referred to as deep semantic features, $\boldsymbol{DF}$.
8: Extracting a learnable classification token, $\boldsymbol{T}_{cls}^{DF}$, from $\boldsymbol{DF}$ and feeding it into a classification head yields the predicted class for the current pixel.
9: **end for**
10: Apply the trained model to the test dataset to generate predicted labels.

Table 1. Explanation of the division of training samples and test samples in the Houston2013 dataset, the Trento dataset, and the Pavia University dataset.

NO.	Houston2013 Dataset			Trento Dataset			Pavia University Dataset		
	Class	Training (1%).	Test.	Class	Training (1%).	Test.	Class	Training (1%).	Test.
#1	Healthy Grass	13	1238	Apple Trees	40	3994	Asphalt	66	6565
#2	Stressed Grass	13	1241	Buildings	29	2874	Meadows	186	18,463
#3	Synthetic Grass	7	690	Ground	5	474	Gravel	21	2078
#4	Tree	12	1232	Woods	91	9032	Trees	31	3033
#5	Soil	12	1230	Vineyard	105	10,396	Metal Sheets	13	1332
#6	Water	3	322	Roads	31	3143	Bare Soil	50	4979
#7	Residential	13	1255				Bitumen	13	1317
#8	Commercial	12	1232				Bricks	37	3645
#9	Road	13	1239				Shadows	9	938
#10	Highway	12	1215						
#11	Railway	12	1223						
#12	Parking Lot 1	12	1221						
#13	Parking Lot 2	5	464						
#14	Tennis Court	4	424						
#15	Running Track	7	653						
	Total	150	14,879	Total	301	29,913	Total	426	42,350

3. Results

3.1. Data Description

The proposed TNCCA model was tested on three widely used datasets. Below, we introduce these three datasets one by one.

Houston2013 dataset: The Houston2013 dataset was jointly provided by the research group at the University of Houston and the National Mapping Center of the United States. It contained a wide range of categories and has been widely used by researchers. The dataset consisted of 144 bands and contained 349 × 1905 classified pixels. There were 15 different classification categories. Figure 4 displayed the pseudocolored image and ground truth map of the Houston2013 dataset.

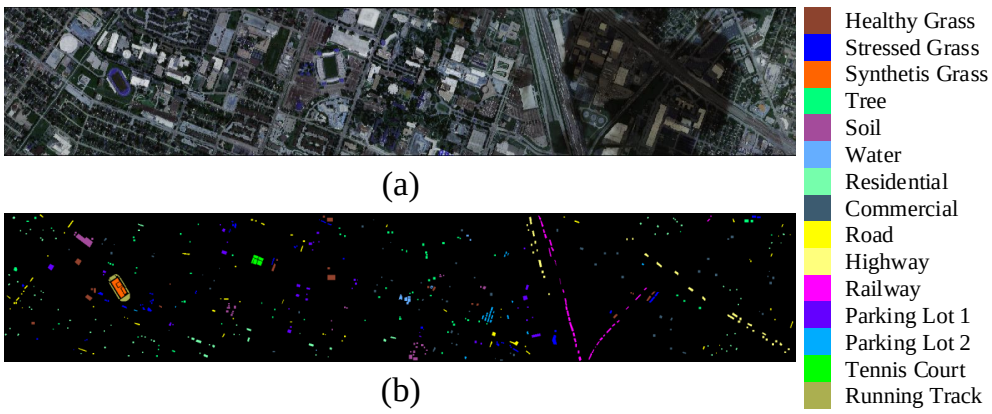

Figure 4. Presentation of the Houston2013 dataset. (**a**) Pseudo-color image composed of three spectral bands. (**b**) Ground truth map.

Trento dataset: The Trento dataset was captured in the southern region of Trento, Italy. It was an HSI obtained using the Airborne lmaging Spectrometer for Application (AISA) Eagle sensor. The dataset consisted of 63 spectral bands and had dimensions of 600 × 166 pixels for classification. It included six different categories of ground objects. Figure 5a,b respectively display the pseudocolored image and ground truth map.

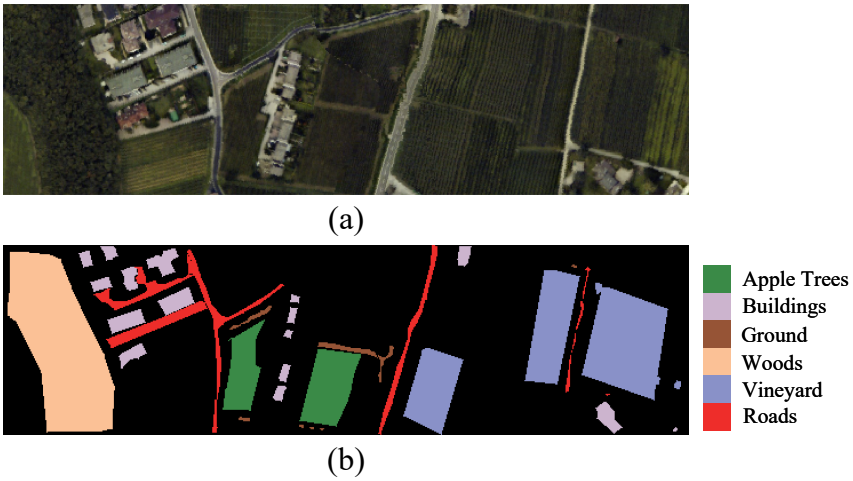

Figure 5. Presentation of the Trento dataset. (**a**) Pseudo-color image composed of three spectral bands. (**b**) Ground truth map.

Pavia University dataset: The Pavia University dataset was a collection of HSI taken in 2001, specifically at Pavia University in Italy. The dataset was an HSI obtained using a Reflective Optics System Imaging Spectrometer (ROSIS) sensor. The image comprised 115 bands and had dimensions of 610 × 340 classified pixels. There were a total of nine land cover classification categories. To reduce the interference of noise, we removed 12 bands that contained noise. Figure 6 displays the pseudocolored image and ground truth map of the dataset.

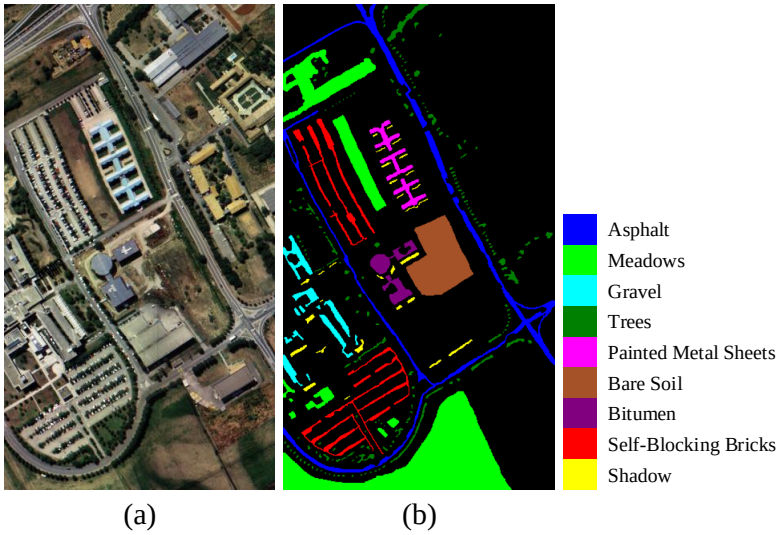

Figure 6. Presentation of the Pavia University dataset. (**a**) Pseudo-color image composed of three spectral bands. (**b**) Ground truth map.

We present the division of training and test samples for the three datasets in Table 1, which includes the specific data for each category. For each category, we used 1% of the total number of samples as the training set.

3.2. Parameter Analysis

In the model we proposed, there was a set of hyperparameters, such as batch size, the size of the first cubic patch, and the size of the second cubic patch. We conducted experimental analysis on these parameters to ensure that their values were optimal. The analysis results are shown in Figures 7–9.

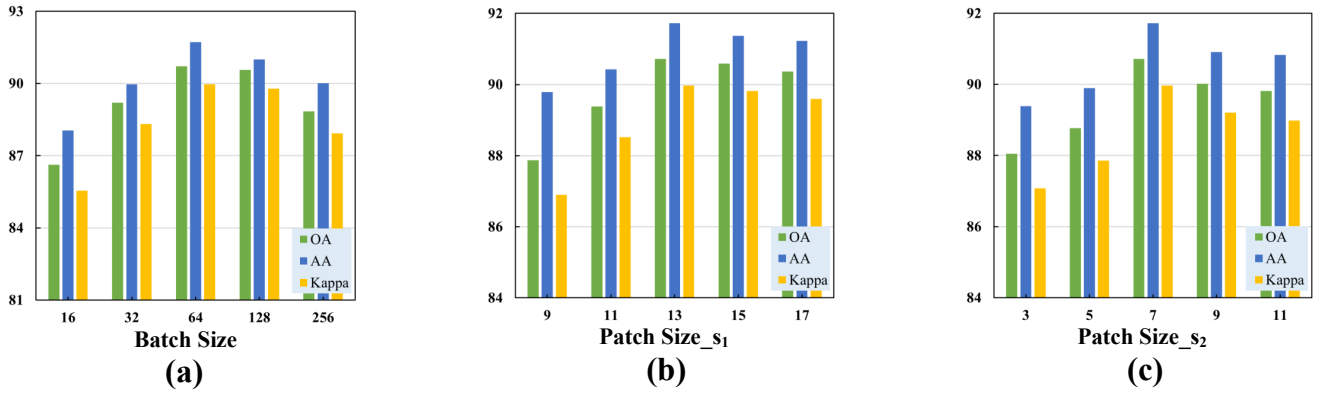

Figure 7. Validation of the optimal hyperparameters with different classification metrics for the Houston2013 dataset. (**a**) Batch size. (**b**) Size of the cubic patch in the first branch. (**c**) Size of the cubic patch in the second branch.

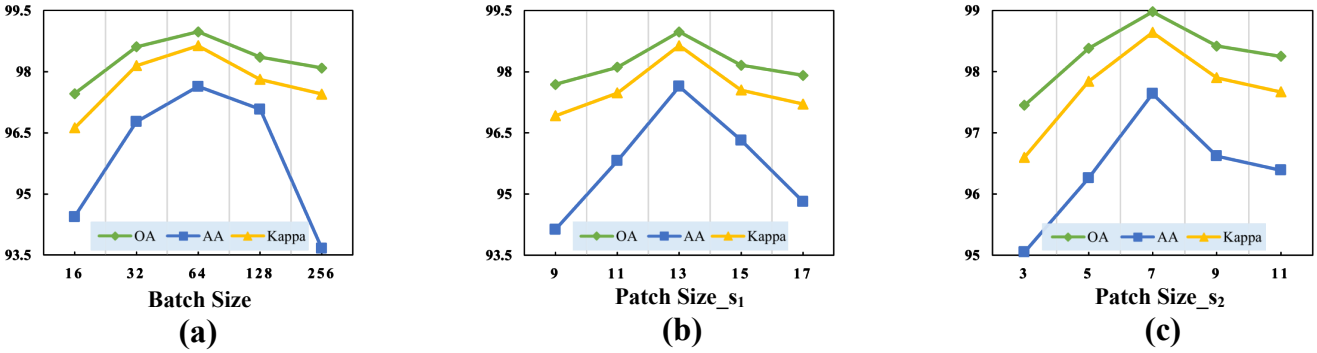

Figure 8. Validation of the optimal hyperparameters with different classification metrics for the Trento dataset. (**a**) Batch size. (**b**) Size of the cubic patch in the first branch. (**c**) Size of the cubic patch in the second branch.

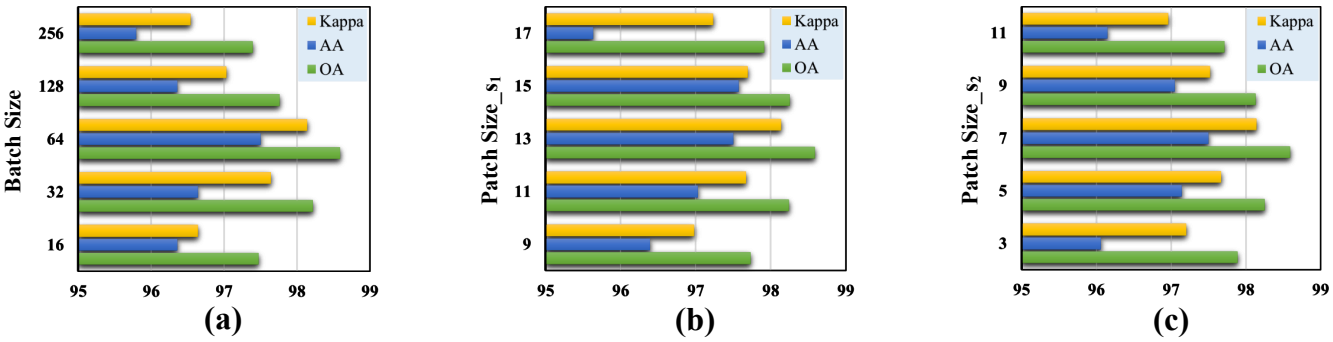

Figure 9. Validation of the optimal hyperparameters with different classification metrics for the Pavia University dataset. (**a**) Batch size. (**b**) Size of the cubic patch in the first branch. (**c**) Size of the cubic patch in the second branch.

(1) Batch Size: Due to our observation that the performance of the transformer architecture was highly sensitive to the batch size, different sizes resulted in varying classification performance. We set the batch size to the following candidate values: $\{16, 32, 64, 128, 256\}$. Additionally, we experimentally determined the batch size that yielded the best performance for our proposed model.

(2) Patch Size: Since the cubic patch served as the input to the model, selecting a patch size that was too small could limit the model's receptive field, while choosing a

size that was too large could result in excessive data volume and increased computational complexity. Our proposed TNCCA selected two different sizes of cubic patches to extract multi-scale features, for which the size of the cubic patch in the first branch was slightly larger than that in the second branch. These two cubic patches served as inputs to the model, and their sizes significantly impacted the classification accuracy. Therefore, we conducted experiments on these two hyperparameters.

We first selected the parameter for the first branch from the set $\{9, 11, 13, 15, 17\}$, and the experimental results showed that the model achieved the best classification performance when its value was 13. Then, for the second branch, we selected the parameter from the set $\{3, 5, 7, 9, 11\}$. From Figures 7–9, it can be observed that the model achieved the highest classification metrics when its value was 7.

3.3. Classification Results and Analysis

We explored eight advanced classification models, and in this section, we describe the conducted experiments and analyze them to compare the classification performance of our proposed model with these models. They comprised SVM [14], 1D-CNN [22], 3D-CNN [24], M3D-CNN [25], 3D-DLA [44], Hybrid [26], SSFTT [39], and morphFormer [42]. To maintain the original performance of the comparative models, we used the training strategies described in their respective papers. The number of training and testing samples for each model was the same as the numbers listed in Table 1, and random sampling was employed. If you wish to reproduce our experiments, you can download the code from the following link: https://github.com/cupid6868/TNCCA.git (accessed on 25 March 2024).

(1) Quantitative results and analysis: We present the results in Tables 2–4, where we demonstrate the superior performance of our proposed model. We highlight the best results for each metric. We conducted experiments on three datasets: the Houston2013 dataset, the Trento dataset, and the Pavia University dataset. The comparative classification metrics included overall accuracy (OA), average accuracy (AA), the Kappa coefficient (κ), and class-wise accuracy. The data in the tables clearly indicate that our proposed TNCCA outperformed the other seven models on the experimental datasets. Let us take the Houston2013 dataset as an example. The proposed TNCCA exhibited the best classification performance for classes such as 'Synthetic Grass', 'Soil', 'Water', 'Commercial', 'Parking Lot 2', 'Tennis Court', and 'Running Track'. Additionally, for classes like 'Healthy Grass', 'Stressed Grass', and 'Parking Lot 1', although our model's performance was not the best, it still ranked among that of the top methods. In contrast, SVM and 1D-CNN showed extremely low classification performance for certain classes. This clearly demonstrated that, in the context of small sample sizes, our proposed model effectively utilized multi-scale feature information and fully exploited the spatial–spectral characteristics in HSI.

(2) Visual evaluation and analysis: We present the aforementioned experimental results in the form of classification maps, shown in Figures 10–12. By comparing the spatial contours of the classification maps with the noise contained in the images, we can clearly observe the superior classification performance of the proposed TNCCA compared to other models.

In the classification maps, it is obvious that the classification map of TNCCA exhibited the clearest spatial contours and contained the least amount of noise. Conversely, the classification maps of the other models showed more instances of misclassifications and interfering noise. Let us take the classification map of the Houston2013 dataset as an example. The classification map of our proposed model closely resembles the ground truth map. On the other hand, the classification maps of SVM, 1D-CNN, 3D-CNN, M3D-CNN, and 3D-DLA exhibited more misclassifications and noise. In the zoomed-in window, we can clearly observe the high classification performance of our proposed model for classes such as 'Parking Lot 2', 'Road', and 'Synthetic Grass'.

Table 2. Comparison of classification performance using the Houston2013 dataset with different methods (The optimal results are shown in bold, and the names of land-covers are shown in italics).

Instances	SVM [14]	1D-CNN [22]	3D-CNN [24]	M3D-CNN [25]	3D-DLA [44]	Hybrid [26]	SSFTT [39]	morphFormer [42]	TNCCA
HealthyGrass	85.78 ± 0.00	85.70 ± 0.00	73.99 ± 6.96	94.74 ± 5.10	85.11 ± 0.28	89.68 ± 2.88	85.78 ± 6.71	**96.66 ± 1.79**	94.82 ± 2.49
StressedGrass	1.39 ± 2.41	0.00 ± 0.00	41.94 ± 0.17	81.35 ± 5.60	75.10 ± 7.70	83.42 ± 2.56	89.79 ± 6.94	**96.21 ± 1.69**	96.13 ± 1.78
SyntheticGrass	0.00 ± 0.00	0.00 ± 0.00	47.89 ± 4.20	90.33 ± 3.02	92.89 ± 1.01	73.04 ± 11.29	92.41 ± 10.15	98.26 ± 0.72	**99.34 ± 0.32**
Tree	42.77 ± 28.19	37.85 ± 8.81	48.98 ± 16.58	84.68 ± 3.12	**93.37 ± 3.11**	72.64 ± 21.09	90.99 ± 3.37	93.15 ± 1.05	90.85 ± 2.58
Soil	61.76 ± 52.71	95.09 ± 0.92	78.78 ± 2.06	87.66 ± 8.48	96.61 ± 1.74	99.72 ± 0.46	99.75 ± 0.21	92.62 ± 5.57	**100 ± 0.00**
Water	0.00 ± 0.00	0.00 ± 0.00	16.77 ± 2.63	38.61 ± 7.19	38.81 ± 16.97	78.98 ± 11.47	82.60 ± 1.35	79.60 ± 3.43	**91.55 ± 3.31**
Residential	87.94 ± 1.98	**95.75 ± 0.04**	48.96 ± 2.08	53.30 ± 6.21	49.61 ± 3.50	53.01 ± 3.43	74.42 ± 3.87	77.71 ± 4.59	82.54 ± 2.94
Commercial	30.65 ± 12.64	0.00 ± 0.00	29.54 ± 3.55	54.49 ± 7.91	45.83 ± 2.60	70.94 ± 2.03	69.23 ± 2.59	67.28 ± 1.78	**82.88 ± 3.53**
Road	7.02 ± 6.74	86.54 ± 2.98	39.87 ± 16.20	58.59 ± 6.57	68.44 ± 0.42	55.82 ± 0.80	87.27 ± 3.51	**88.86 ± 3.91**	84.57 ± 4.02
Highway	17.55 ± 30.41	0.21 ± 0.38	45.59 ± 7.91	60.90 ± 4.45	57.17 ± 33.02	77.91 ± 3.03	**95.08 ± 1.07**	87.57 ± 9.54	91.59 ± 3.60
Railway	23.73 ± 28.90	0.00 ± 0.00	39.98 ± 10.29	38.37 ± 8.64	55.79 ± 30.36	72.03 ± 5.33	**92.58 ± 5.11**	87.18 ± 1.06	81.26 ± 5.85
ParkingLot1	7.88 ± 13.66	2.48 ± 3.55	39.68 ± 13.03	73.16 ± 9.21	73.32 ± 16.06	**89.62 ± 2.33**	83.48 ± 5.40	74.50 ± 3.30	89.46 ± 2.26
ParkingLot2	0.50 ± 0.69	0.00 ± 0.00	41.48 ± 4.72	39.87 ± 10.37	29.45 ± 8.02	52.94 ± 7.05	84.69 ± 5.40	84.77 ± 3.14	**90.77 ± 3.21**
TennisCourt	0.00 ± 0.00	0.00 ± 0.00	40.33 ± 26.68	51.02 ± 6.60	74.92 ± 11.14	**100 ± 0.00**	99.76 ± 0.23	90.09 ± 3.42	**100 ± 0.00**
RunningTrack	62.68 ± 54.52	0.00 ± 0.00	67.99 ± 15.59	85.96 ± 7.39	97.54 ± 1.86	**100 ± 0.00**	**100 ± 0.00**	97.65 ± 0.57	**100 ± 0.00**
OA (%)	33.24 ± 5.37	33.63 ± 0.67	48.39 ± 2.48	68.44 ± 1.81	70.49 ± 1.27	77.29 ± 1.19	87.85 ± 1.20	87.17 ± 0.80	**90.72 ± 0.89**
AA (%)	28.64 ± 4.57	26.91 ± 0.54	46.78 ± 1.52	66.20 ± 1.54	68.93 ± 0.42	77.98 ± 1.21	88.52 ± 0.82	87.47 ± 0.79	**91.72 ± 0.74**
$\kappa \times 100$	27.46 ± 5.69	27.58 ± 0.73	44.12 ± 2.63	65.82 ± 1.95	68.06 ± 1.37	75.44 ± 1.29	86.87 ± 1.29	86.13 ± 0.87	**89.97 ± 0.97**

Table 3. Comparison of classification performance using the Trento dataset with different methods (The optimal results are shown in bold, and the names of land-covers are shown in italics).

Instances	SVM [14]	1D-CNN [22]	3D-CNN [24]	M3D-CNN [25]	3D-DLA [44]	Hybrid [26]	SSFTT [39]	morphFormer [42]	TNCCA
AppleTrees	0.37 ± 0.32	0.00 ± 0.00	78.58 ± 34.28	97.72 ± 0.55	86.28 ± 4.62	99.04 ± 0.56	**99.64 ± 0.23**	99.49 ± 0.23	99.58 ± 0.25
Buildings	66.96 ± 5.56	73.56 ± 0.67	75.59 ± 11.99	80.15 ± 3.32	82.65 ± 1.42	67.16 ± 12.93	98.08 ± 0.38	91.66 ± 1.63	**98.32 ± 0.31**
Ground	0.00 ± 0.00	0.00 ± 0.00	45.44 ± 16.43	71.49 ± 13.08	57.63 ± 18.06	35.43 ± 14.97	51.26 ± 2.53	91.20 ± 5.05	**97.79 ± 1.90**
Woods	92.87 ± 0.93	89.39 ± 0.53	98.11 ± 2.59	98.82 ± 0.55	97.75 ± 0.40	**100 ± 0.00**	**100 ± 0.00**	99.97 ± 0.01	**100 ± 0.00**
Vineyard	75.15 ± 1.61	84.40 ± 1.00	99.54 ± 0.13	99.49 ± 0.45	99.49 ± 0.07	**100 ± 0.00**	99.91 ± 0.09	99.92 ± 0.11	**100 ± 0.00**
Roads	67.53 ± 2.70	70.08 ± 1.67	81.61 ± 8.08	82.04 ± 4.15	80.40 ± 2.62	66.89 ± 2.86	89.71 ± 2.49	92.84 ± 1.33	**93.17 ± 1.48**
OA (%)	67.74 ± 0.52	70.75 ± 0.22	91.27 ± 6.45	94.91 ± 0.56	92.91 ± 0.65	92.21 ± 1.19	97.88 ± 0.25	98.20 ± 0.12	**98.98 ± 0.22**
AA (%)	50.48 ± 1.17	52.90 ± 0.01	79.81 ± 10.23	88.28 ± 3.10	84.03 ± 2.45	78.09 ± 0.34	89.77 ± 0.61	95.85 ± 0.93	**97.64 ± 0.62**
$\kappa \times 100$	55.45 ± 0.80	59.46 ± 0.29	88.22 ± 8.81	93.21 ± 0.76	90.49 ± 0.88	89.54 ± 1.59	97.17 ± 0.33	97.60 ± 0.16	**98.64 ± 0.30**

Table 4. Comparison of classification performance using the Pavia University dataset with different methods (The optimal results are shown in bold, and the names of land-covers are shown in italics).

Instances	SVM [14]	1D-CNN [22]	3D-CNN [24]	M3D-CNN [25]	3D-DLA [44]	Hybrid [26]	SSFTT [39]	morphFormer [42]	TNCCA
Asphalt	94.76 ± 0.61	91.32 ± 0.27	83.24 ± 3.03	94.44 ± 1.69	88.32 ± 5.04	92.46 ± 0.93	97.91 ± 0.66	96.75 ± 0.98	**98.61 ± 0.57**
Meadows	92.45 ± 1.20	95.58 ± 1.22	93.89 ± 4.27	98.14 ± 1.35	96.42 ± 1.06	99.95 ± 0.07	98.39 ± 0.33	99.75 ± 0.20	**99.98 ± 0.02**
Gravel	0.00 ± 0.00	0.00 ± 0.00	54.52 ± 20.93	68.65 ± 5.04	80.95 ± 1.39	**94.80 ± 0.50**	82.53 ± 1.10	82.17 ± 1.63	87.11 ± 0.87
Trees	15.81 ± 2.28	60.44 ± 4.77	66.00 ± 21.73	95.57 ± 1.52	91.10 ± 1.73	76.81 ± 4.40	95.73 ± 1.67	96.03 ± 1.11	**98.48 ± 0.55**
MetalSheets	99.07 ± 0.18	99.44 ± 0.17	90.29 ± 15.20	99.62 ± 0.52	97.99 ± 1.36	86.76 ± 19.29	**100 ± 0.00**	99.82 ± 0.30	**100 ± 0.00**
Baresoil	18.51 ± 6.58	9.58 ± 1.72	78.17 ± 8.13	77.51 ± 11.15	74.37 ± 2.27	99.43 ± 0.87	99.66 ± 0.42	99.16 ± 1.18	**99.69 ± 0.14**
Bitumen	0.00 ± 0.00	0.00 ± 0.00	57.27 ± 5.24	81.87 ± 7.21	81.67 ± 6.23	81.67 ± 21.37	99.16 ± 0.62	79.87 ± 4.37	**99.56 ± 0.31**
Bricks	86.91 ± 2.98	92.42 ± 1.27	73.79 ± 8.03	92.83 ± 2.12	77.66 ± 10.19	72.84 ± 7.42	95.40 ± 1.81	95.70 ± 1.19	**95.93 ± 1.50**
Shadows	0.00 ± 0.00	**98.36 ± 0.53**	57.78 ± 21.11	96.97 ± 1.61	94.34 ± 2.30	64.81 ± 14.40	82.37 ± 7.04	93.85 ± 1.60	98.11 ± 0.70
OA (%)	68.90 ± 0.76	74.54 ± 0.28	82.68 ± 1.84	92.56 ± 1.48	89.36 ± 1.32	92.72 ± 1.96	96.96 ± 0.41	96.99 ± 0.47	**98.59 ± 0.12**
AA (%)	45.28 ± 0.61	60.79 ± 0.23	72.77 ± 1.63	89.51 ± 2.76	86.98 ± 2.04	85.50 ± 6.27	94.57 ± 0.84	93.68 ± 0.97	**97.50 ± 0.17**
$\kappa \times 100$	56.26 ± 0.98	64.42 ± 0.24	76.85 ± 2.48	90.04 ± 2.07	85.81 ± 1.77	90.29 ± 2.63	95.98 ± 0.54	96.01 ± 0.63	**98.14 ± 0.16**

In conclusion, our proposed model outperformed the compared models and demonstrated the best classification performance. It highlighted the model's capability of extracting features effectively in small sample scenarios.

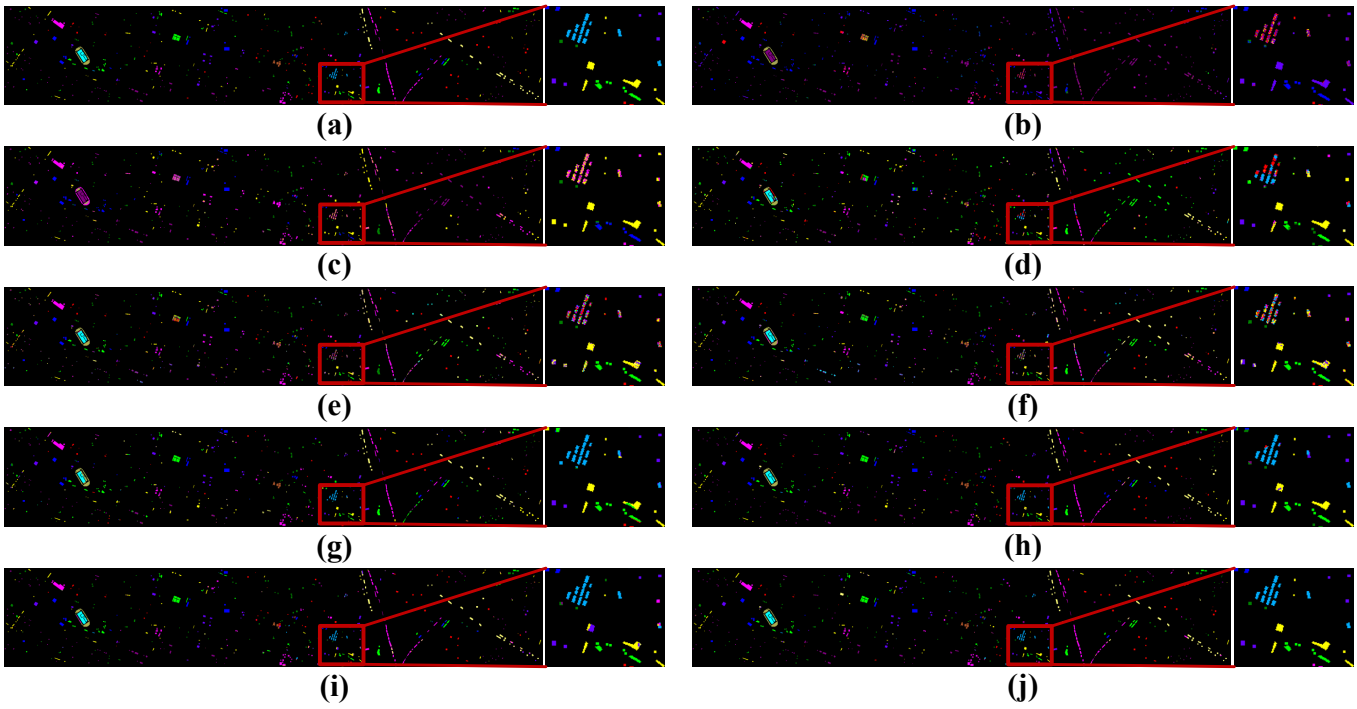

Figure 10. Visualization of classification results using different classification methods with the Houston2013 dataset. (**a**) Ground truth map, (**b**) SVM (OA = 33.24%), (**c**) 1D-CNN (OA = 33.63%), (**d**) 3D-CNN (OA = 48.39%), (**e**) M3D-CNN (OA = 68.44%), (**f**) 3D-DLA (OA = 70.49%), (**g**) hybrid (OA = 77.29%), (**h**) SSFTT (OA = 87.85%), (**i**) morphFormer (OA = 87.17%), and (**j**) the proposed method (OA = 90.72%).

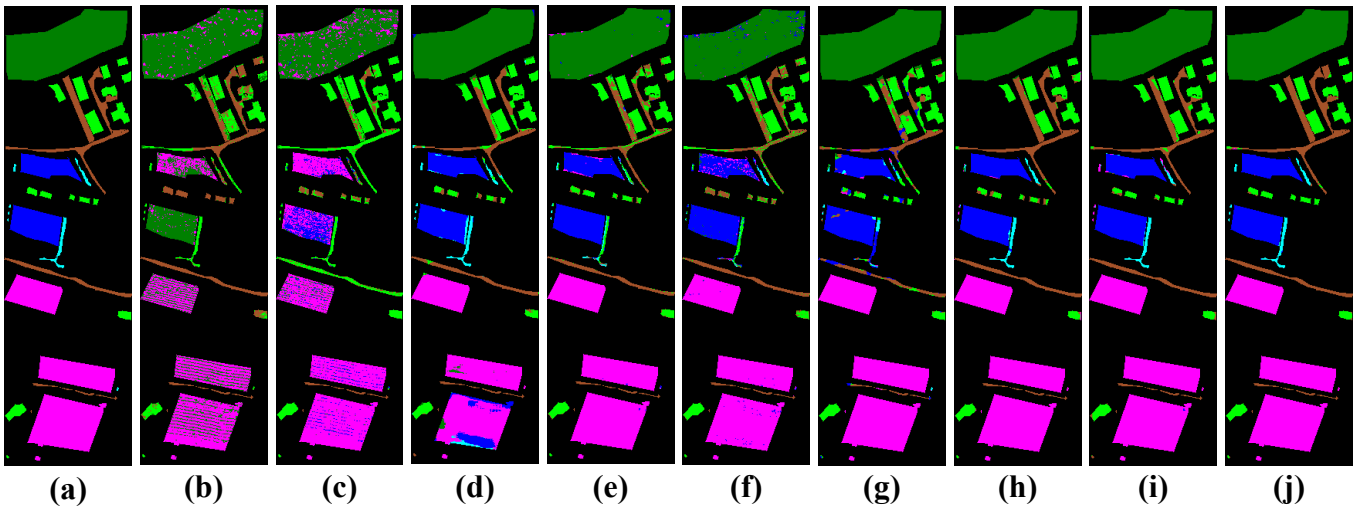

Figure 11. Visualization of classification results using different classification methods with the Trento dataset. (**a**) Ground truth map, (**b**) SVM (OA = 67.74%), (**c**) 1D-CNN (OA = 70.75%), (**d**) 3D-CNN (OA = 91.27%), (**e**) M3D-CNN (OA = 94.91%), (**f**) 3D-DLA (OA = 92.91%), (**g**) hybrid (OA = 92.21%), (**h**) SSFTT (OA = 97.88%), (**i**) morphFormer (OA = 98.20%), and (**j**) the proposed method (OA = 98.98%).

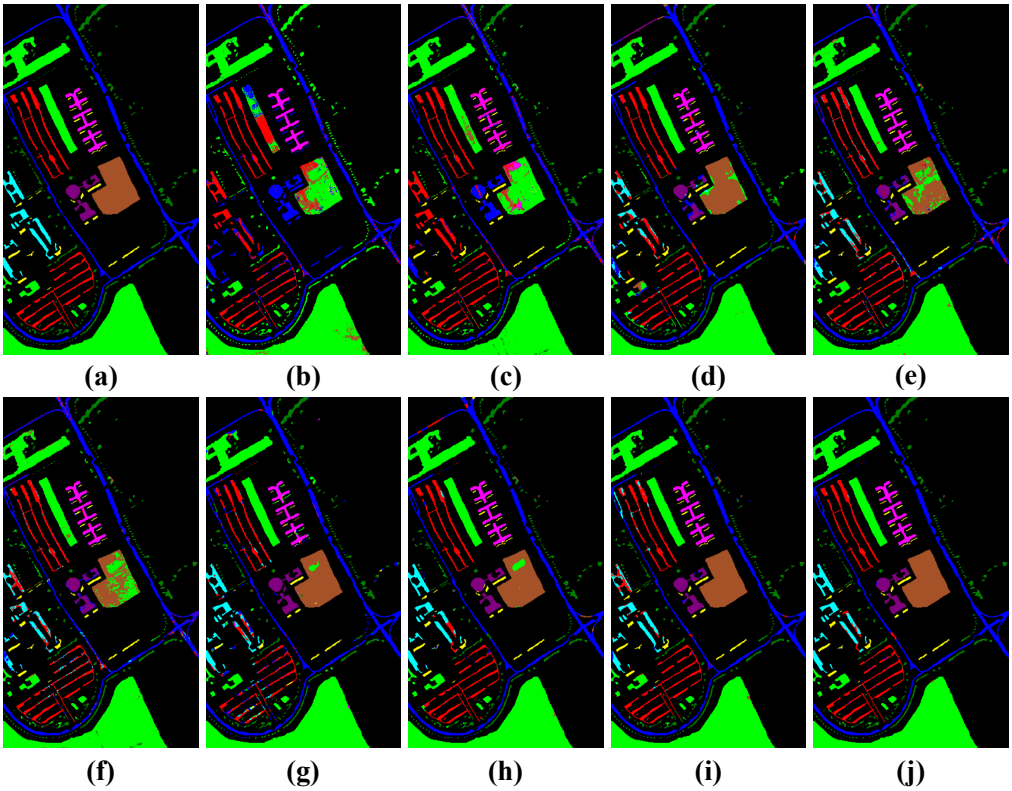

Figure 12. Visualization of classification results using different classification methods with the Pavia University dataset. (**a**) Ground truth map, (**b**) SVM (OA = 33.24%), (**c**) 1D-CNN (OA = 33.63%), (**d**) 3D-CNN (OA = 48.39%), (**e**) M3D-CNN (OA = 68.44%), (**f**) 3D-DLA (OA = 70.49%), (**g**) hybrid (OA = 77.29%), (**h**) SSFTT (OA = 87.85%), (**i**) morphFormer (OA = 87.17%), and (**j**) the proposed method (OA = 90.72%).

3.4. Analysis of Inference Speed

To demonstrate the inference speed of our proposed model, TNCCA, we present the training time and testing time of the model with different datasets in Table 5. The data show that our training speed is fast, as the model can complete 500 epochs in a very short period. To facilitate the observation of model performance during the training process, we adopted a training strategy of conducting a test after each epoch. This resulted in a significantly longer testing time compared to the training time. Additionally, we employed dynamic learning rates to accelerate the convergence speed.

Table 5. The inference speed of TNCCA on different datasets (epoch = 500).

Dataset	Houston2013		Trento		Pavia University	
	Train.	Test.	Train.	Test.	Train.	Test.
Time (min)	0.58	13.85	0.91	23.47	1.26	33.39

Among the three tested datasets, the Pavia University dataset, which had larger spatial dimensions and higher spectral dimensions, took the longest time, with 1.26 min for training and only 0.153 s per epoch. The training times for the other datasets were shorter. From this table, it is easy to conclude that our proposed model not only achieved high classification accuracy but also trained at a fast speed, demonstrating high efficiency.

3.5. Ablation Analysis

To validate the effectiveness of each module in our proposed model, we conducted ablation experiments on the four modules using the Houston2013 dataset. These four modules comprised a 3D convolutional layer (3D-Conv), a multi-scale 2D convolutional module (Ms2D-Conv), a feature map tokenization module (Tokenizer), and a transformer encoder module (TE). We evaluated their performance in terms of OA, AA, and κ by considering five different combinations of these modules. The results are listed in Table 6.

Specifically, we first kept only the 3D convolutional layer, and it was evident that the performance was extremely poor. In the next step, we removed the transformer encoder with the CNN-enhanced cross-attention mechanism, which was one of the main innovations of this paper. The results showed a significant decrease in classification performance. The OA, AA, and κ values of the model decreased by 4.71%, 5.89%, and 5.32%, respectively, compared to TNCCA. Next, we removed the 3D convolutional layer and replaced the multi-scale 2D convolutional module with a regular 2D convolutional layer. In this configuration, the model's OA decreased by 1.14%, and its AA decreased by 1.66%, compared to TNCCA. Then, we removed the 3D convolutional layer, which resulted in the loss of rich spectral information in the HSI. We observed that the model's OA decreased by 0.64%, and its AA decreased by 1.8%, compared to TNCCA. Finally, we replaced only the multi-scale 2D convolutional module with a regular 2D convolutional layer. In this case, the model's OA decreased by 0.17%, and its AA decreased by 0.20%, compared to TNCCA. This clearly demonstrated the positive contributions of these four modules in enhancing the accuracy of network classification.

Table 6. Conducting ablation experiments on different modules (using the Houston2013 dataset).

Cases	Components				Indicators		
	3D-Conv	**Ms2D-Conv**	**Tokenizer**	**TE**	**OA (%)**	**AA (%)**	**$\kappa \times 100$**
1	√	×	×	×	48.39	46.78	44.12
2	√	√	√	×	85.81	85.63	84.65
3	×	2D-Conv	√	√	89.58	90.06	88.73
4	√	2D-Conv	√	√	90.55	91.52	89.56
5	√	√	√	√	90.72	91.72	89.97

4. Conclusions

The paper has introduced a novel dual-branch deep learning classification model that effectively captures spatial–spectral feature information from HSI and achieves high classification performance in small sample scenarios. The two branches of the model utilize cubic patches of different sizes as inputs to fully exploit the limited samples and extract features at different scales. First, we employed a 3D convolutional layer and a multi-scale 2D convolutional module to extract shallow-level features. Then, the obtained feature maps were transformed into tokens, assigning a larger number of tokens to the larger cubic patches. Next, we utilized a transformer with CNN-enhanced cross-attention to delve into the deep-level feature information and fuse the different-scale information from the two branches. Finally, through extensive experiments, we demonstrated that the proposed TNCCA model exhibits superior classification performance.

In our future work, we aim to explore the rich multi-scale spatial–spectral features in HSI from different perspectives to improve classification accuracy. However, as the classification accuracy improves, there is an increasing demand for lightweight operations and reducing the computational complexity of the models. We will utilize more novel lightweight operations to design more efficient classification models.

Author Contributions: Methodology, X.W. and B.L.; conceptualization, X.W. and L.S.; software, X.W. and L.S.; validation, X.W. and C.L.; investigation, B.L. and X.W.; writing—original draft preparation, X.W.; writing—review and editing, B.L. and L.S.; visualization, C.L. and X.W.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Jiangsu key R&D plan, no. BE2022161.

Data Availability Statement: The data presented in this study are available in the article.

Acknowledgments: The authors thank the anonymous reviewers and the editors for their insightful comments and helpful suggestions that helped improve the quality of our manuscript.

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

Abbreviations

The following abbreviations are used in this manuscript:

HSI	Hyperspectral image
RF	Random Forest
SVM	Support Vector Machine
LDA	Linear Discriminant Analysis
PCA	Principal Component Analysis
CNN	Convolutional Neural Network
GAN	Generative Adversarial Network
GCN	Graph Convolutional Network
RNN	Recurrent Neural Network
ResNet	Residual Network
TE	Transformer encoder
Q	Queries
K	Keys
V	Values
MLP	Multi-layer perceptron
LN	Normalization layers

References

1. He, C.; Cao, Q.; Xu, Y.; Sun, L.; Wu, Z.; Wei, Z. Weighted Order-p Tensor Nuclear Norm Minimization and Its Application to Hyperspectral Image Mixed Denoising. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 5510505. [CrossRef]
2. Sun, L.; Wang, Q.; Chen, Y.; Zheng, Y.; Wu, Z.; Fu, L.; Jeon, B. CRNet: Channel-Enhanced Remodeling-Based Network for Salient Object Detection in Optical Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5618314. [CrossRef]
3. Gao, H.; Zhang, Y.; Chen, Z.; Xu, S.; Hong, D.; Zhang, B. A Multidepth and Multibranch Network for Hyperspectral Target Detection Based on Band Selection. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5506818. [CrossRef]
4. Gao, H.; Zhang, Y.; Chen, Z.; Xu, F.; Hong, D.; Zhang, B. Hyperspectral Target Detection via Spectral Aggregation and Separation Network With Target Band Random Mask. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5515516. [CrossRef]
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. Gong, P.; Li, Z.; Huang, H.; Sun, G.; Wang, L. ICESat GLAS Data for Urban Environment Monitoring. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 1158–1172. [CrossRef]
7. 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; pp. 1–4. [CrossRef]
8. Ardouin, J.P.; Levesque, J.; Rea, T.A. A demonstration of hyperspectral image exploitation for military applications. In Proceedings of the 2007 10th International Conference on Information Fusion, Québec, QC, Canada, 9–12 July 2007; pp. 1–8. [CrossRef]
9. Su, Y.; Gao, L.; Jiang, M.; Plaza, A.; Sun, X.; Zhang, B. NSCKL: Normalized Spectral Clustering With Kernel-Based Learning for Semisupervised Hyperspectral Image Classification. *IEEE Trans. Cybern.* **2023**, *53*, 6649–6662. [CrossRef] [PubMed]
10. Su, Y.; Chen, J.; Gao, L.; Plaza, A.; Jiang, M.; Xu, X.; Sun, X.; Li, P. ACGT-Net: Adaptive Cuckoo Refinement-Based Graph Transfer Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5521314. [CrossRef]
11. Yu, H.; Gao, L.; Liao, W.; Zhang, B.; Zhuang, L.; Song, M.; Chanussot, J. Global Spatial and Local Spectral Similarity-Based Manifold Learning Group Sparse Representation for Hyperspectral Imagery Classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 3043–3056. [CrossRef]

12. Gao, H.; Yang, Y.; Li, C.; Gao, L.; Zhang, B. Multiscale Residual Network With Mixed Depthwise Convolution for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 3396–3408. [CrossRef]
13. Yan, L.; Fan, B.; Liu, H.; Huo, C.; Xiang, S.; Pan, C. Triplet Adversarial Domain Adaptation for Pixel-Level Classification of VHR Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 3558–3573. [CrossRef]
14. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [CrossRef]
15. Ye, Q.; Huang, P.; Zhang, Z.; Zheng, Y.; Fu, L.; Yang, W. Multiview Learning With Robust Double-Sided Twin SVM. *IEEE Trans. Cybern.* **2022**, *52*, 12745–12758. [CrossRef] [PubMed]
16. Ham, J.; Chen, Y.; Crawford, M.; Ghosh, J. Investigation of the random forest framework for classification of hyperspectral data. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 492–501. [CrossRef]
17. Guo, Y.; Han, S.; Li, Y.; Zhang, C.; Bai, Y. K-Nearest Neighbor combined with guided filter for hyperspectral image classification. *Procedia Comput. Sci.* **2018**, *129*, 159–165. [CrossRef]
18. Bandos, T.V.; Bruzzone, L.; Camps-Valls, G. Classification of Hyperspectral Images With Regularized Linear Discriminant Analysis. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 862–873. [CrossRef]
19. Dalla Mura, M.; Villa, A.; Benediktsson, J.A.; Chanussot, J.; Bruzzone, L. Classification of Hyperspectral Images by Using Extended Morphological Attribute Profiles and Independent Component Analysis. *IEEE Geosci. Remote Sens. Lett.* **2011**, *8*, 542–546. [CrossRef]
20. Li, S.; Song, W.; Fang, L.; Chen, Y.; Ghamisi, P.; Benediktsson, J.A. Deep Learning for Hyperspectral Image Classification: An Overview. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6690–6709. [CrossRef]
21. Lu, W.; Wang, X.; Sun, L.; Zheng, Y. Spectral–Spatial Feature Extraction for Hyperspectral Image Classification Using Enhanced Transformer with Large-Kernel Attention. *Remote Sens.* **2024**, *16*, 67. [CrossRef]
22. Hu, W.; Huang, Y.; Wei, L.; Zhang, F.; Li, H. Deep convolutional neural networks for hyperspectral image classification. *J. Sens.* **2015**, *2015*, 258619. [CrossRef]
23. Zhao, W.; Du, S. Spectral–Spatial Feature Extraction for Hyperspectral Image Classification: A Dimension Reduction and Deep Learning Approach. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 4544–4554. [CrossRef]
24. 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]
25. He, M.; Li, B.; Chen, H. Multi-scale 3D deep convolutional neural network for hyperspectral image classification. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 3904–3908. [CrossRef]
26. 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]
27. 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]
28. Mou, L.; Ghamisi, P.; Zhu, X.X. Deep Recurrent Neural Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3639–3655. [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. Haut, J.M.; Paoletti, M.E.; Plaza, J.; Plaza, A.; Li, J. Visual Attention-Driven Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 8065–8080. [CrossRef]
31. Sun, H.; Zheng, X.; Lu, X.; Wu, S. Spectral–Spatial Attention Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 3232–3245. [CrossRef]
32. Hang, R.; Li, Z.; Liu, Q.; Ghamisi, P.; Bhattacharyya, S.S. Hyperspectral Image Classification With Attention-Aided CNNs. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 2281–2293. [CrossRef]
33. Ma, W.; Yang, Q.; Wu, Y.; Zhao, W.; Zhang, X. Double-Branch Multi-Attention Mechanism Network for Hyperspectral Image Classification. *Remote Sens.* **2019**, *11*, 1307. [CrossRef]
34. Zhu, M.; Jiao, L.; Liu, F.; Yang, S.; Wang, J. Residual Spectral–Spatial Attention Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 449–462. [CrossRef]
35. 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.
36. Sun, L.; Wang, X.; Zheng, Y.; Wu, Z.; Fu, L. Multiscale 3-D–2-D Mixed CNN and Lightweight Attention-Free Transformer for Hyperspectral and LiDAR Classification. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 2100116. [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.* **2022**, *60*, 5518615. [CrossRef]
38. He, X.; Chen, Y.; Lin, Z. Spatial-Spectral Transformer for Hyperspectral Image Classification. *Remote Sensing* **2021**, *13*, 498. [CrossRef]
39. 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]

40. Mei, S.; Song, C.; Ma, M.; Xu, F. Hyperspectral Image Classification Using Group-Aware Hierarchical Transformer. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5539014. [CrossRef]
41. Fang, Y.; Ye, Q.; Sun, L.; Zheng, Y.; Wu, Z. Multiattention Joint Convolution Feature Representation With Lightweight Transformer for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5513814. [CrossRef]
42. Roy, S.K.; Deria, A.; Shah, C.; Haut, J.M.; Du, Q.; Plaza, A. Spectral–Spatial Morphological Attention Transformer for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5503615. [CrossRef]
43. Gao, H.; Chen, Z.; Xu, F. Adaptive spectral-spatial feature fusion network for hyperspectral image classification using limited training samples. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *107*, 102687. [CrossRef]
44. Ben Hamida, A.; Benoit, A.; Lambert, P.; Ben Amar, C. 3-D Deep Learning Approach for Remote Sensing Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4420–4434. [CrossRef]