remote sensing

Article

Multi-Feature Cross Attention-Induced Transformer Network for Hyperspectral and LiDAR Data Classification

Zirui Li [1], Runbang Liu [1], Le Sun [2,3] and Yuhui Zheng [3,*]

[1] Ocean College, Jiangsu University of Science and Technology, Zhenjiang 212100, China; 212241803624@just.edu.cn (Z.L.); liurunbang212@just.edu.cn (R.L.)
[2] Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science and Technology, Nanjing 210044, China; sunlecncom@nuist.edu.cn
[3] School of Artificial Intelligence, Nanjing University of Information Science and Technology, Nanjing 210044, China
* Correspondence: zheng_yuhui@nuist.edu.cn

Abstract: Transformers have shown remarkable success in modeling sequential data and capturing intricate patterns over long distances. Their self-attention mechanism allows for efficient parallel processing and scalability, making them well-suited for the high-dimensional data in hyperspectral and LiDAR imagery. However, further research is needed on how to more deeply integrate the features of two modalities in attention mechanisms. In this paper, we propose a novel Multi-Feature Cross Attention-Induced Transformer Network (MCAITN) designed to enhance the classification accuracy of hyperspectral and LiDAR data. The MCAITN integrates the strengths of both data modalities by leveraging a cross-attention mechanism that effectively captures the complementary information between hyperspectral and LiDAR features. By utilizing a transformer-based architecture, the network is capable of learning complex spatial-spectral relationships and long-range dependencies. The cross-attention module facilitates the fusion of multi-source data, improving the network's ability to discriminate between different land cover types. Extensive experiments conducted on benchmark datasets demonstrate that the MCAITN outperforms state-of-the-art methods in terms of classification accuracy and robustness.

Keywords: hyperspectral imagery; LiDAR data; cross-attention; transformer; classification

Citation: Li, Z.; Liu, R.; Sun, L.; Zheng, Y. Multi-Feature Cross Attention-Induced Transformer Network for Hyperspectral and LiDAR Data Classification. *Remote Sens.* **2024**, *16*, 2775. https://doi.org/10.3390/rs16152775

Academic Editor: Kevin Tansey

Received: 18 June 2024
Revised: 24 July 2024
Accepted: 26 July 2024
Published: 29 July 2024

1. Introduction

Hyperspectral image classification (HSIC) [1] is of great significance in the field of remote sensing, and it is widely used in agriculture [2–4], environmental monitoring [5,6], urban planning [7–9], military reconnaissance [10,11], and other fields. HSI can provide detailed spectral features by capturing spectral information in multiple continuous bands, helping distinguish different types of ground objects [12]. However, due to the high dimensionality and complexity of hyperspectral data, classification solely relying on HSI faces challenges such as data redundancy and noise interference [13,14]. For this reason, joint classification, combined with LiDAR data, has become an effective solution. LiDAR data provides high-resolution spatial and structural information, which makes up for the lack of spatial resolution of HSI and complements the spectral information of hyperspectral data. The combination of the two types of data can significantly improve classification performance [15]. By jointly utilizing the three-dimensional spatial information of LiDAR and the spectral information of HSI, we can more accurately identify and classify ground objects and reduce confusion. The fusion of hyperspectral and LiDAR data can provide rich information in multiple dimensions, making the classification results more comprehensive and reliable. Therefore, it is very necessary to combine HSI and LiDAR data for joint classification.

In the past five years, HSIC methods have made significant progress [16], mainly reflected in the widespread application of deep learning technology and the development of multi-source data fusion methods [17]. Recently, Yang et al. proposed an HSIC method based on a multi-level feature fusion network of interactive transformer and convolutional neural networks (CNNs) [18]. In addition, Yang et al. also proposed a method based on deformable dilated convolution pyramid feature extraction [19]. Cao et al. investigated the use of convolutional neural networks (CNNs) combined with active learning for classifying HSI [20]. Xue et al. explored a self-calibrating convolution [21] for collaborative classification of hyperspectral and LiDAR data. From the perspective of development history, HSIC methods have experienced a transformation from traditional machine learning methods to deep neural network methods. Early HSIC methods mainly relied on machine learning algorithms such as SVM [22–24] and random forest (RF) [25–27]. These methods improved the accuracy of classification to a certain extent. However, with the increase in data volume and computing power, deep learning methods [28–30] have gradually become the mainstream of HSIC. Deep learning models such as CNN, RNN, and GAN [31] have greatly improved the performance of HSIC by automatically extracting multi-level feature representations. Although deep learning methods have made significant progress, there are still some problems, such as dependence on large amounts of annotated data, a high computational cost, and insufficient generalizability to different data sets. To overcome these limitations, combining multi-source data (such as LiDAR data) for classification becomes an effective solution.

Deep learning has demonstrated remarkable capabilities in extracting features from raw data and adjusting parameters, particularly through its multi-layered network structure that can automatically capture complex feature representations from data [32–34]. Common deep learning structures include RNN [35], LSTM [36], CNN [37], etc. Among these, CNNs have particularly strong feature extraction capabilities and can automatically learn deep semantic features from images [38,39]. Some approaches based on CNN depth features have emerged. For instance, Li et al. [40] proposed a spatial-spectral saliency reinforcement network (Sal2RN) to enhance joint classification performance. Despite these advancements, the mainstream methods still face challenges due to the very different dimensions and feature distribution of HSI and LiDAR data [41,42]. To address this, Gao et al. [43] proposed an adaptive, multiscale spatial-spectral enhancement network (AMSSE-Net) that includes an adaptive feature-fusion module. In addition to CNNs, other advanced network structures have been used for joint HSI and LiDAR data classification to improve accuracy, such as autoencoders (SAEs) [44], GCNs [45], GANs [46], etc. While these classical deep learning methods can effectively extract local features from images, they are not as effective in dealing with global relationships, and they lack a consideration of location information. To address this, the transformer network has been applied to joint classification [47].

In learning overall sequence features, transformers [48–51] rely on their global modeling capability, self-attention mechanism, adaptability to different-length sequences, and multitasking ability. They are widely used in the field of hyperspectral LiDAR classification; combining HSI and LiDAR for land-cover classification at the same time can establish long-term dependencies and help make full use of spectral information and global features. Through multiple-branch networks, and combining self- and cross-guided attention mechanisms, effective fusion and classification of hyperspectral and LiDAR data are achieved. Ni et al. [52] proposed a model called the Multiscale Head Selection Transformer. Through a multiscale head selection mechanism, the transformer network selects and integrates hyperspectral and LiDAR features at various scales. This mechanism allows MHST to capture features at different scales, enhancing classification accuracy and robustness. Yang et al. [53] proposed a LiDAR-guided cross-attention fusion method. This approach uses LiDAR data to guide band selection in HSI and employs a cross-attention mechanism to fuse LiDAR and hyperspectral data, thereby enhancing classification performance. Roy et al. [54] proposed a Cross-HL Attention Transformer model, extending the self-attention mechanism by cross-attending to hyperspectral and LiDAR data. This

approach facilitates the effective fusion and feature extraction from multiple data sources. The transformer network enables end-to-end classification processing, yielding superior classification results. The implementation of cross-attention demonstrates the capability to integrate information from diverse data sources in hyperspectral and LiDAR classification, thereby enhancing the performance and accuracy of classification models. By employing multi-feature fusion, the multidimensional features of land cover are captured more comprehensively, improving the classifier's ability to recognize complex scenes, and thus achieving more precise hyperspectral and LiDAR data classification.

In a word, the contributions of the proposed MCAITN method can be summarized into the following threefold list:

(1) The proposed method introduces a novel architecture that leverages the strengths of transformer networks to enhance classification accuracy in HSI and LiDAR data fusion. The MCAITN effectively captures the complementary information between the two modalities, leading to improved feature representation and classification performance.

(2) The MCAITN incorporates a cross-attention module that selectively focuses on the most relevant features from each modality. This targeted attention mechanism enables the network to emphasize informative features from both HSI and LiDAR data, enhancing its ability to discriminate between different land-cover types and improving the overall classification accuracy.

(3) Comprehensive experiments on benchmark datasets reveal that MCAITN exceeds current SOTA methods in classification accuracy, resilience to noise and data variability, and computational efficiency.

2. Materials and Methods

The MCAITN architecture is illustrated in Figure 1. Initially, PCA was applied to reduce the dimensionality of the original HSI data. Then, LiDAR data and the reduced dimensionality HSI data were segmented into small three-dimensional blocks as the input of the shallow CNN feature-extraction module. Next, the extracted joint spectral–spatial features were input into the HSI and LiDAR branches for processing so as to preserve the local context and crucial information. Finally, they were input into the cross-feature enhanced-attention transformer encoder for comprehensive feature cross-learning, the classification tags were extracted for final classification, and the number of encoders was N.

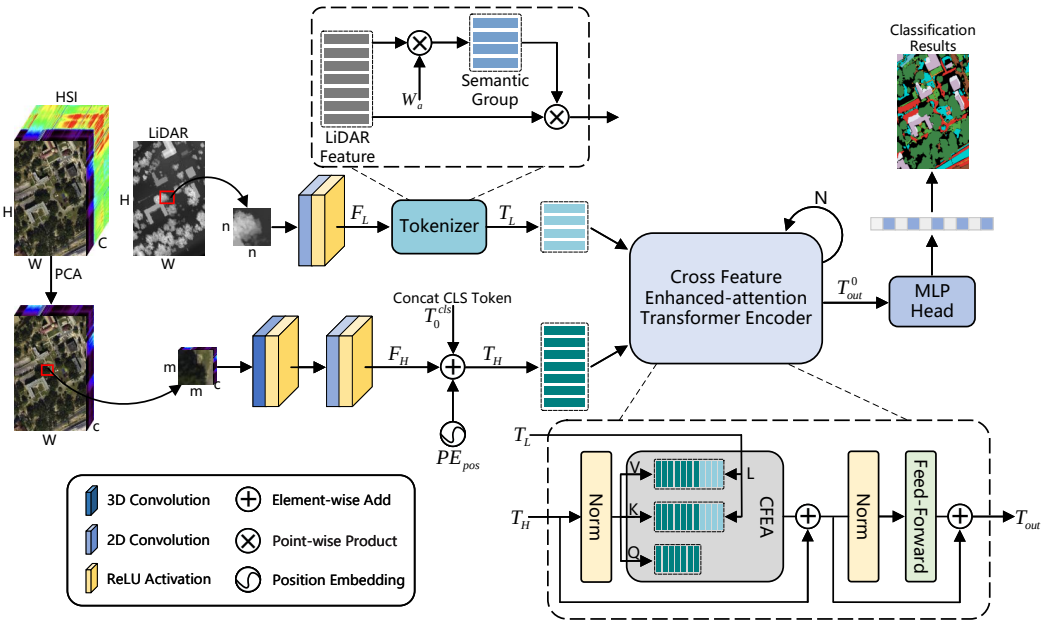

Figure 1. Overall architecture of MCAITN.

2.1. HSI and LiDAR-Data Preprocessing

Assume a hyperspectral data cube and LiDAR data, denoted as $X \in \mathbb{R}^{W\times H\times C}$, $Y \in \mathbb{R}^{W\times H}$, respectively, where X and Y are the original inputs to the whole classification framework, W and H correspond to the width and height of the HSI, and C represents the number of bands.

Typically, HSI is rich in spectral bands, which contain very valuable information but also introduce a lot of redundancy. To reduce the calculation complexity, PCA is used to reduce the dimension of HSI data. The label of each pixel in X is denoted as a one-shot vector, $Z \in \mathbb{R}^{1\times 1\times B}$, where B denotes the number of land cover analogs. Then, PCA is performed along the spectral dimension on the HSI data X. After the dimensionality reduction is performed, the spatial resolution of X remains unchanged, but the number of spectral bands is reduced from C to c, i.e., $X_{pca} \in \mathbb{R}^{W\times H\times c}$. In other words, PCA eliminates the redundant spectral information in HSI and retains the spatial information without degradation.

Then, we divided X_{pca} into small, overlapping three-dimensional adjacent patches. Each patch was denoted as $X_{patch} \in \mathbb{R}^{m\times m\times c}$, where $m \times m$ represents the spatial size of the patch, and c represents the number of spectral bands. The label for each patch was derived from the ground-truth label of the central pixel within that patch. For all adjacent patches, we took the patch X^{ij}_{patch} with the central pixel position (i, j) as an example; its spatial coverage ranged from $i-(m-1)/2$ to $i(m-1)/2$ in width and from $j-(m-1)/2$ to $j+(m-1)/2$ in height. It contained all spectra within this spatial range. It should be emphasized that, when generating patches for edge pixels, one side of these pixels is smaller than $(m-1)/2)$ due to the asymmetric overlay dimensions; therefore, a padding operation is required. The remaining patches were divided into training and test sets according to the proportion.

And for LiDAR image Y, a similar operation was performed. It was segmented into small overlapping patches. We denoted each small patch as $Y_{patch} \in \mathbb{R}^{n\times n}$, where $n \times n$ denotes the size of the patch.

2.2. Shallow CNN Feature-Extraction Module

In recent years, CNNs have become extensively utilized in HSIC, primarily due to their remarkable proficiency in extracting local features, setting them apart as a dominant approach in the field. In MCAITN, we use a shallow CNN feature-extraction module to effectively extract the spectral and spatial information of HSI. For LiDAR data, we use a two-dimensional convolution block to extract its features. The shallow CNN feature extraction module mainly consists of a 3D convolution block and a 2D convolution block. Such a structure helps effectively integrate spectral and spatial information in the early stage of feature extraction.

First, we introduce a 3D convolution-based block to process the input 3D neighborhood patch. The convolution block consists of a 3D convolution layer with a kernel size of 8@$3\times 3\times 3$ and a nonlinear activation layer. The stride and padding are 1 and 0, respectively. Specifically, the 3D convolution layer performs convolution operations along the spectral and spatial dimensions to generate a 3D feature map containing spectral–spatial features. The calculation process of the 3D convolution block is as follows:

$$F_{3\text{d}} = \Phi\left(X_{\text{patch}}\Theta w_{3\text{d}} + b_{3\text{d}}\right) \tag{1}$$

where F_{3d} represents the three-dimensional feature map, and w_{3d} and b_{3d} represent the weight and bias parameters, respectively. Θ is the three-dimensional convolution operator, and Φ is the activation function.

Then, the obtained feature map is flattened along the spectral dimension and used as the input of the 2D convolution block. Similarly, the 2D convolution is composed of a 2D convolution block with a kernel size of 64@$3\times 3\times 3$ and a subsequent nonlinear activation

layer. The 2D convolution block performs convolution along the spatial dimension to extract more discriminative spatial information. The calculation process is as follows.

$$F_{2d} = \Phi(f(F_{3d}) \odot w_{2d} + b_{2d}) \tag{2}$$

where F_{2d} represents the 2D feature map, f denotes the flattening operation, and w_{2d} and b_{2d} denote the weight and bias parameters, respectively. $\odot$ is the 2D convolution operator, and Φ is the activation function.

For LiDAR data, since they are two-dimensional, we use the two-dimensional convolution block in the shallow CNN feature-extraction module to extract their spatial features. The calculation process is as follows:

$$Y_{2d} = \Phi(Y_{pach} \odot w_{2d} + b_{2d}) \tag{3}$$

Finally, we flatten the 2D feature map of F_{2d} and Y_{2d} along the spatial dimension and then output the features F_H and F_L. Through this step, we achieve the exploration of spatial and spectral information in the data at a relatively low computational cost.

2.3. HSI Semantic Tokenizer and LiDAR Gaussian-Weighted Feature Tokenizer

As shown in Figure 1, for HSI, we use position embedding to sign the position information of each semantic token for the feature F2d extracted via the shallow CNN feature-extraction module. Each token is represented by $[F_{H1}, F_{H2}, \ldots, F_{Hw}]$. These tokens are connected together with a learnable classification token, T_0^{cls}, for classification tasks. The position information, PE_{pos}, encoding is then attached to the token representation. The resulting semantic tag embedding sequence is as follows:

$$T_H = [T_0^{cls}, F_{H1}, F_{H2}, \cdots, F_{Hw}] + PE_{pos} \tag{4}$$

For LiDAR data, we apply semantic labels to the LiDAR data to enable the representation and processing of advanced semantic ideas at the level of LiDAR features. The flattened feature map of the input is defined as $F_L \in \mathbb{R}^{nn \times z}$, where nn is the size, and z is the number of channels. The feature token is defined as $T \in \mathbb{R}^{w \times z}$, where w denotes the number of tokens. For the feature mapping of F_L, T_L can be obtained from the following formula:

$$T_L = \underbrace{\text{softmax}(F_L W_a)^T}_{A} F_L \tag{5}$$

where W_a represents the weight matrix initialized with Gaussian distribution, and $F_L W_a$ represents performing 1×1 dot product to map F_L to a group. At this juncture, the size of the semantic group is denoted as A. Following this, A undergoes specialization, and $softmax(.)$ is employed to emphasize the relatively crucial semantic components. Subsequently, it is combined with F_L to yield the semantic sequence $T_L = [F_{L1}, F_{L2}, \cdots, F_{Lw}]$.

Finally, we input the obtained T_H and T_L into the CFETE module to learn the relationship between high-level semantic features.

2.4. Cross-Feature Enhanced-Attention Transformer Encoder

As shown in Figure 1, CFETE is mainly composed of a cross-feature enhanced attention (CFEA) block and a simple, fully connected feed-forward network (FFN).

The original MHSA mechanism aims to establish global, long-range dependencies between input feature sequences. Now, in order to provide MHSA with more valuable context information, we extend the traditional MHSA mechanism to CFEA and take HSI and LiDAR semantic-feature sequences as the input of CFEA, which can be expressed as follows:

$$T_{in} = [T_H, T_L] \tag{6}$$

In the CFEA module, firstly, the two feature sequences are linearly transformed to obtain five different matrices: query $Q_H \in \mathbb{R}^{m\times d}$, key $K_H \in \mathbb{R}^{m\times d}$, the value $V_H \in \mathbb{R}^{m\times d}$ of HSI, and $K_L \in \mathbb{R}^{n\times d}$, $V_L \in \mathbb{R}^{n\times d}$ of the LiDAR. The linear transformation process is defined as follows:

$$\begin{aligned} Q_H &= T_H W_q \\ K_H &= T_H W_k \\ V_H &= T_H W_v \\ K_L &= T_L W_k \\ V_L &= T_L W_v \end{aligned} \tag{7}$$

where m and n are the number of HSI feature vectors T_H and LiDAR feature vectors T_L, respectively, and d represents their dimensions. W_q, W_v, and W_k are learnable weight matrices. Then, Q, K, and V are divided into h parts along the d dimension, expressed as follows:

$$\begin{aligned} Q_H &= [Q_{H1}, Q_{H2}, \ldots, Q_{Hh}] \\ K_H &= [K_{H1}, K_{H2}, \ldots, K_{Hh}] \\ V_H &= [V_{H1}, V_{H2}, \ldots, V_{Hh}] \\ K_L &= [K_{L1}, K_{L2}, \ldots, K_{Lh}] \\ V_L &= [V_{L1}, V_{L2}, \ldots, V_{Lh}] \end{aligned} \tag{8}$$

where h represents the number of the attention heads, $Q_H i \in \mathbb{R}^{m\times(d/h)}$, $K_H i \in \mathbb{R}^{m\times(d/h)}$, $V_H i \in \mathbb{R}^{m\times(d/h)}$, $K_L i \in \mathbb{R}^{n\times(d/h)}$, and $V_L i \in \mathbb{R}^{n\times(d/h)}$. Next, we use K_L and V_L to expand K_H and V_H; the process is as follows:

$$K_i' = Concat(K_{Hi}, K_{Li}) V_i' = Concat(V_{Hi}, V_{Li}) \tag{9}$$

Since K_L and V_L represent the projection matrix of LiDAR data, they preserve local context and salient spatial feature representation. We utilize the extended K and V matrices and integrate them into the self-attention mechanism. This allows the model to consider not only the spatial–spectral characteristics of HSI but also the feature representation of LiDAR data during the self-attention operation, thereby incorporating a broader range of contextual information. This helps provide a more comprehensive set of information, allowing the model to better understand the relationships and dependencies between different parts of the input sequence. Furthermore, employing a multi-head mechanism allows the model to process different subspaces of information concurrently. The extended parts of K and V of each head provide additional information, enhance the cross-fusion between different data source information, and improve the model's generalization ability and prediction performance with location data.

Following this, the attention scores between each Q and K in each attention head are calculated, and then these scores are converted into attention weights using the $softmax$ function. Finally, these weights are multiplied by V. The calculation process for each head is as follows:

$$H_i = \text{Attention}\left(Q_i, K_i', V_i'\right) = \text{Softmax}\left(\frac{Q_i K_i'^T}{\sqrt{d}}\right) V_i' \tag{10}$$

The final output of CFEA is constructed by concatenating the attention results from all attention heads and further projecting them. We represent it as follows:

$$CFEA(T_{in}) = Concat(H_1, H_1, \ldots, H_h) W_o \tag{11}$$

Among them, W_o is the parameter matrix.

Then, the output of CFEA is used as the input of FFN. The feed-forward layer consists of two FC layers with a Gaussian error linear unit (GELU) inserted in between. It is defined as follows:

$$FFN(X) = FC_2(GELU(FC_1(X))) \tag{12}$$

In summary, the entire calculation process of CFETE can be summarized as follows:

$$\begin{aligned} z_l^{'} &= CFEA(LN(z_l)) + z_l \\ z_{l+1}^{'} &= FFN(LN(z_l^{'})) + z_l^{'} \end{aligned} \tag{13}$$

where LN is the layer norm, which alleviates the gradient-vanishing and -exploding problems, thereby speeding up the training process. z_l represents the input of the lth layer of CFETE.

2.5. Classification Head

To achieve the final classification, we use a multi-layer perceptron (MLP) head. Typically, an MLP consists of multiple FC layers, and the MLP head refers to its last layer. In this paper, the MLP head consists of layer norms and FC layers. The classification tokens are extracted from the output of MATE and used as the input for the MLP head; the output dimension of the MLP head is equal to the total number of classes predicted in the end. The unit with the highest value in this output corresponds to the predicted label for that pixel. Algorithm 1 outlines the entire execution process of the method.

Algorithm 1 MCAITN network.

Require:
HSI data $X \in \mathbb{R}^{W \times H \times C}$, LiDAR data $Y \in \mathbb{R}^{W \times H}$; PCA bands number c; patch size S; train rate α%.
Ensure:
Predicted labels of the test set.
Obtain the HSI after PCA transformation, denoted as X_{pca}.
Obtain patches from X_{pca} and Y, respectively, and divide the patches into the train set and test set.
Set batch size $bs = 32$; learning rate $lr = 5e - 4$; epochs $e = 100$.
for $i = 1$ to e do **do**
  Perform Conv3D and Conv2D on the HSI patch to obtain spatial spectral features F_{3d} and F_{2d}; perform Conv2D on the LiDAR patch to obtain spatial feature Y_{2d}.
  Flatten the 2D feature map to obtain F_L and F_H.
  Use Gaussian-Weighted Feature Tokenizer to generate feature sequence T_L for F_L.
  Obtain T_H by adding position information and an additional classification token to F_H.
  Input features T_H and T_L into CFETE for feature learning.
  for $j = 1$ to L do **do**
    Perform CFEA operation according to Equation (11).
    Perform FFN operation according to Equation (13).
  end for
  Input the first classification token T_{out}^0 into the MLP head to obtain the category label.
end for
Use the trained model to predict the test set.

3. Experiments

To validate the proposed method's effectiveness, we conducted experiments on three classic hyperspectral and LiDAR joint classification datasets and compared them with current mainstream methods. In the experiments, three classification metrics, overall accuracy (OA), average accuracy (AA), and the kappa coefficient, were used to quantitatively assess the experimental performance.

3.1. Dataset Description

3.1.1. MUUFL

The MUUFL was obtained using a reflective optical system-imaging spectrometer sensor, capturing the area around the Gulf Park University of Southern Mississippi campus. The

spatial dimensions of both the HSI and LiDAR data are 325 × 220 pixels, with 325 representing the height and 220 the width. After noisy bands were filtered out, the HSI data were reduced to 64 spectral bands. This dataset categorizes the land into 11 different classes. Figure 2 displays the specific situation of the dataset.

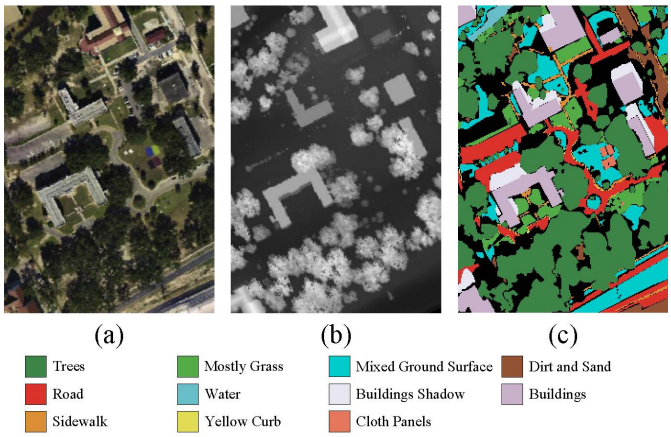

Figure 2. MUUFL. (**a**) Pseudo-color image of HSI. (**b**) Gra-image of the LiDAR-based DSM. (**c**) Ground-truth map.

3.1.2. Trento

The Trento dataset features HSI and LiDAR data collected from across a rural area south of Trento, Italy. The dataset boasts a spatial resolution of 1 m and dimensions of 600 × 166 pixels. It includes 63 spectral bands in the HSI data, with wavelengths ranging from 0.42 to 0.99 µm. There are six distinct land-cover classes within this dataset. Figure 3 shows the specific situation of the dataset.

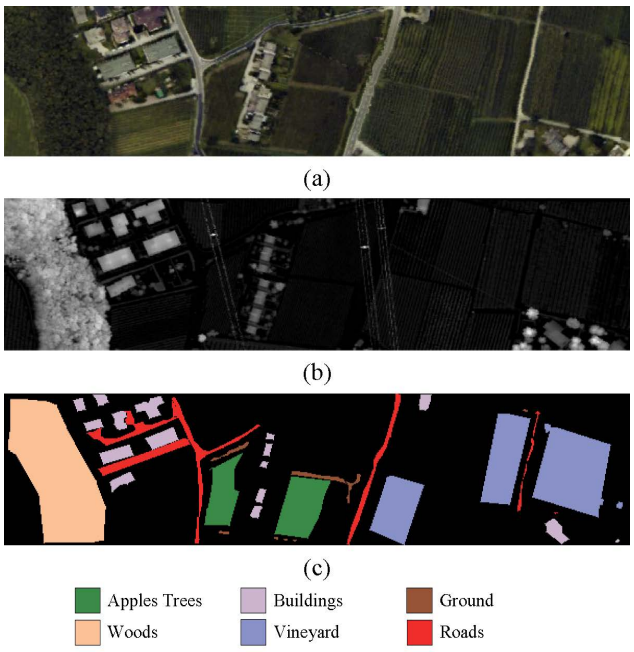

Figure 3. Trento. (**a**) Pseudo-color image of HSI. (**b**) Gray image of the LiDAR-based DSM. (**c**) Ground-truth map.

3.1.3. Augsburg

The Augsburg dataset includes HSI data and LiDAR-based DSM data collected from across the city of Augsburg, Germany. The dataset's spatial dimensions are 332 × 485 pixels, representing height and width. The HSI data comprise 180 spectral bands, with wave-

lengths ranging from 0.4 to 2.5 µm. This dataset encompasses seven land-cover classes. Figure 4 presents the specific situation of the dataset.

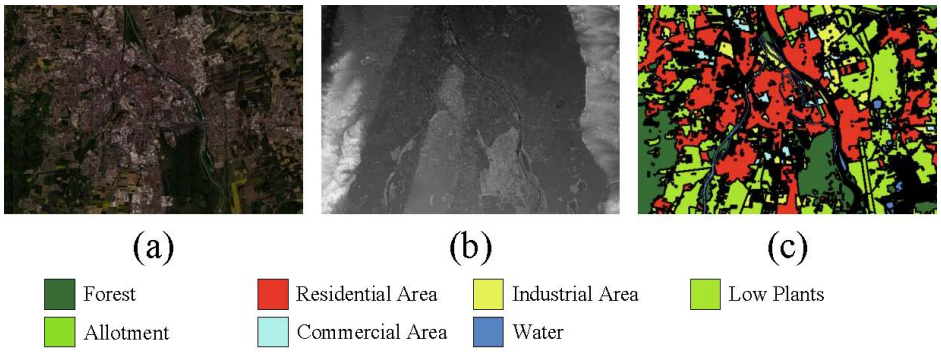

Figure 4. Augsburg. (**a**) Pseudo-color image of HSI. (**b**) Gray image of the LiDAR-based DSM. (**c**) Ground-truth map.

The datasets are available at (accessed on 1 January 2024): https://github.com/AnkurDeria/MFT. The names of land categories, along with the numbers of training and testing samples used in the experiments for the three datasets mentioned above, are presented in Table 1.

Table 1. Training and test samples in the MUUFL, Augsburg, and Trento datasets.

No.	MUUFL			Augsburg			Trento		
	Class	**Training.**	**Test.**	**Class**	**Training.**	**Test.**	**Class**	**Training.**	**Test.**
C01	Trees	150	23,096	Forest	675	12,832	Apple Trees	129	3905
C02	Mostly Grass	150	4120	Residential Area	1516	28,813	Buildings	125	2778
C03	Mixed Ground Surface	150	6732	Industrial Area	192	3659	Ground	105	374
C04	Dirt and Sand	150	1676	Low Plants	1342	25,515	Woods	154	8969
C05	Road	150	6537	Allotment	28	547	Vineyard	184	10,317
C06	Water	150	316	Commercial Area	82	1563	Roads	122	3052
C07	Buildings Shadow	150	2083	Water	76	1454			
C08	Buildings	150	6090						
C09	Sidewalk	150	1235						
C10	Yellow Curb	150	33						
C11	Cloth Panels	150	119						
-	Total	1650	52,037	Total	3911	74,383	Total	819	29,395

3.2. Experimental Setup

3.2.1. Evaluation Indicators

We selected four widely used evaluation metrics to quantitatively assess the classification performance of all methods: single-class accuracy, overall accuracy (OA), average accuracy (AA), and the kappa coefficient (κ). Higher values for each metric signify better classification performance.

3.2.2. Configurations

To ensure a fair comparison of the classification performance of the models, both the proposed method and the comparison methods were implemented using the PyTorch framework. All training and testing experiments were conducted on Intel Xeon Silver 4210 and an NVIDIA GeForce RTX 2080Ti GPU. Additionally, the parameters for the comparison methods were maintained as per their original settings to achieve optimal performance. For our method, network parameters were updated using the Adam optimizer, with a batch size of 32 and training epochs set to 100.

3.3. Classification Results and Analysis

In this subsection, we will quantitatively and qualitatively analyze the comparison results between the proposed MCAITN method and the current mainstream methods. These methods include SVM [22], S2FL [55], EndNet [44], MDL [56], LSAF [57], CCRNet [58], CoupledCNN [59], and HCT [12].

3.3.1. Quantitative Results and Analysis

Tables 2–4 present the quantitative results for the three classic datasets Trento, MUUFL, and Augsburg, along with the standard deviations for each metric. From Table 2, it is evident that traditional machine learning methods, such as SVM, have a lower joint classification accuracy, achieving only an OA value of 72.23. In contrast, neural network methods perform relatively better, with methods like CCRNet, CoupleCNN, and HCT having OA values mostly above 83% and AA values around 90%. MCAITN demonstrates significant improvements over competing methods across all evaluation metrics (OA, AA, and Kappa), reaching an OA of 90.43%, an AA of 91.94%, and a Kappa of 0.8745. Additionally, the table shows that the standard deviations for the OA, AA, and Kappa values of our method are relatively low, indicating that the proposed method consistently produces stable classification results across ten random experiments.

Table 2. Performance of various classifiers with the MUUFL dataset (best results are in boldface).

No.	SVM [22]	S2FL [55]	EndNet [44]	MDL [56]	LSAF [57]	CCRNet [58]	CoupledCNN [59]	HCT [12]	MCAITN
1	74.81 ± 01.79	81.02 ± 00.74	84.21 ± 00.96	89.42 ± 03.88	88.70 ± 00.99	84.81 ± 01.91	89.16 ± 01.85	91.12 ± 01.43	91.75 ± 01.57
2	72.94 ± 01.91	76.99 ± 02.36	83.28 ± 02.09	74.57 ± 13.12	85.29 ± 02.48	85.81 ± 01.57	86.96 ± 00.94	85.47 ± 02.35	86.62 ± 03.42
3	57.59 ± 02.06	66.16 ± 01.51	71.85 ± 02.07	77.96 ± 04.67	78.97 ± 02.26	66.15 ± 03.63	81.60 ± 02.32	81.53 ± 04.88	82.21 ± 03.20
4	63.39 ± 01.34	82.56 ± 02.98	87.65 ± 01.49	90.83 ± 09.13	97.16 ± 01.41	94.39 ± 02.98	94.36 ± 02.97	96.07 ± 00.44	96.92 ± 00.89
5	79.06 ± 00.93	84.46 ± 01.22	88.96 ± 01.86	75.93 ± 04.39	87.72 ± 01.28	85.77 ± 02.63	89.66 ± 02.69	87.57 ± 04.96	89.26 ± 01.97
6	92.51 ± 01.31	94.49 ± 00.62	94.38 ± 01.53	99.79 ± 00.18	100 ± 00.00	99.03 ± 00.64	98.92 ± 00.64	99.45 ± 00.69	99.50 ± 00.48
7	82.45 ± 01.03	84.19 ± 01.23	88.70 ± 01.52	90.01 ± 07.08	94.46 ± 01.40	88.54 ± 01.82	91.84 ± 01.65	94.23 ± 00.42	93.64 ± 01.78
8	66.16 ± 01.50	79.49 ± 01.72	80.56 ± 02.05	96.32 ± 02.05	95.59 ± 00.35	94.48 ± 00.72	96.71 ± 01.29	95.15 ± 02.92	96.84 ± 00.84
9	79.48 ± 01.73	71.55 ± 02.78	75.39 ± 03.02	70.23 ± 11.80	77.14 ± 02.78	65.45 ± 02.27	72.71 ± 02.56	78.38 ± 01.55	80.53 ± 02.33
10	82.93 ± 03.49	92.33 ± 03.89	97.31 ± 02.48	84.85 ± 08.02	93.94 ± 05.25	87.72 ± 03.08	94.32 ± 02.85	95.45 ± 02.62	95.67 ± 04.58
11	75.09 ± 02.46	85.82 ± 02.53	98.18 ± 01.18	100.00 ± 00.00	99.72 ± 00.48	97.95 ± 01.81	98.47 ± 00.59	99.02 ± 01.06	98.44 ± 01.32
OA (%)	72.23 ± 01.37	78.31 ± 00.18	82.92 ± 00.64	85.58 ± 00.45	88.18 ± 00.43	83.12 ± 01.01	88.73 ± 00.39	88.93 ± 00.97	**90.43 ± 00.67**
AA (%)	75.13 ± 01.53	81.73 ± 01.85	86.41 ± 00.87	86.36 ± 01.23	90.79 ± 00.50	86.37 ± 00.99	90.43 ± 01.34	91.22 ± 01.57	**91.94 ± 00.52**
$\kappa \times 100$	65.41 ± 01.42	72.47 ± 00.33	77.82 ± 01.04	81.17 ± 00.37	84.60 ± 00.52	78.25 ± 01.17	85.16 ± 01.03	85.29 ± 00.85	**87.45 ± 00.83**

Table 3. Performance of various classifiers with the Augsburg dataset (best results are in boldface).

No.	SVM [22]	S2FL [55]	EndNet [44]	MDL [56]	LSAF [57]	CCRNet [58]	CoupledCNN [59]	HCT [12]	MCAITN
1	95.78 ± 00.39	97.18 ± 00.15	92.49 ± 00.49	95.56 ± 03.04	99.15 ± 00.21	96.44 ± 00.97	97.47 ± 00.97	98.98 ± 00.17	98.97 ± 00.27
2	89.41 ± 01.27	72.29 ± 01.26	88.61 ± 00.53	93.82 ± 03.88	98.53 ± 00.05	96.69 ± 00.76	97.71 ± 00.65	98.69 ± 00.26	98.82 ± 00.29
3	06.47 ± 01.35	32.25 ± 03.09	41.38 ± 03.13	79.42 ± 07.18	89.22 ± 03.02	82.76 ± 03.83	84.71 ± 03.57	88.33 ± 04.20	88.92 ± 03.16
4	67.32 ± 01.39	87.45 ± 01.04	94.25 ± 00.53	99.74 ± 00.06	99.22 ± 00.31	98.02 ± 00.41	97.56 ± 00.53	98.94 ± 00.29	99.07 ± 00.25
5	06.86 ± 01.82	40.34 ± 05.19	31.75 ± 03.13	56.26 ± 13.26	87.08 ± 05.64	41.69 ± 06.06	69.43 ± 03.05	80.04 ± 08.55	86.66 ± 08.75
6	10.90 ± 01.87	39.97 ± 02.81	28.32 ± 04.21	57.34 ± 21.99	54.36 ± 05.16	33.38 ± 05.05	72.84 ± 02.36	70.38 ± 02.06	74.89 ± 03.78
7	53.27 ± 01.85	70.35 ± 01.29	50.65 ± 02.07	47.58 ± 12.57	70.02 ± 02.64	59.39 ± 06.24	61.98 ± 02.58	72.81 ± 04.62	72.91 ± 03.00
OA (%)	76.01 ± 00.83	78.77 ± 00.37	86.77 ± 00.56	93.50 ± 01.46	96.85 ± 00.23	94.35 ± 00.74	95.59 ± 00.75	97.08 ± 00.21	**97.34 ± 00.15**
AA (%)	47.15 ± 00.78	62.83 ± 01.06	61.06 ± 00.95	75.68 ± 00.51	85.37 ± 01.33	72.63 ± 02.80	83.1 ± 01.90	86.88 ± 01.07	**88.61 ± 01.14**
$\kappa \times 100$	64.82 ± 01.09	70.87 ± 00.71	80.73 ± 00.58	90.63 ± 02.08	95.48 ± 00.33	93.09 ± 00.61	93.92 ± 00.79	95.82 ± 00.29	**96.18 ± 00.21**

From the quantitative results of Tables 3 and 4, conclusions similar to those in Table 2 can be drawn; that is, the MCAITN method proposed in this paper achieved the best quantitative indicators in terms of OA, AA, and kappa. The joint-classification methods related to deep learning are significantly better than traditional classification methods such as SVM classifiers, mainly due to the powerful nonlinear feature-extraction capabilities of neural networks. Although the MCAITN method improved on various classification indicators over the second-best method, HCT, in the Augsburg and Trento databases, the OA only increased by 0.26% and 0.11%, the AA increased by 1.73% and 0.21%, and the kappa increased by 0.36 and 0.17. It can be seen that the improvement in the MCAITN method was the least for the Trento dataset. The main reason for this may be that the features in the MUUFL are mainly intertwined buildings and vegetation, and the elevation information in the LiDAR data has a better positive effect on the classification results; meanwhile for the Augsburg dataset,

there is mainly vegetation, and the auxiliary classification ability of elevation information is limited. In the Trento dataset, there are also fields, houses, roads, and trees, but they are more scattered, and better results can be obtained simply through hyperspectral information.

Table 4. Performance of various classifiers with the Trento dataset (best results are in boldface).

No.	SVM [22]	S2FL [55]	EndNet [44]	MDL [56]	LSAF [57]	CCRNet [58]	CoupledCNN [59]	HCT [12]	MCAITN
1	80.05 ± 01.08	80.35 ± 00.71	87.52 ± 00.62	98.06 ± 01.39	99.66 ± 00.09	99.13 ± 00.91	99.32 ± 00.21	99.59 ± 00.17	99.41 ± 00.44
2	77.03 ± 03.13	80.32 ± 01.23	87.43 ± 00.82	99.37 ± 00.74	98.85 ± 00.79	96.74 ± 01.41	97.87 ± 00.29	98.49 ± 01.15	99.32 ± 00.32
3	85.64 ± 02.57	90.47 ± 00.71	98.22 ± 01.04	99.07 ± 00.27	97.86 ± 00.92	94.17 ± 02.11	98.39 ± 00.31	100.00 ± 00.00	100.00 ± 00.00
4	92.48 ± 01.23	93.14 ± 00.31	98.37 ± 00.32	100.00 ± 00.00	100.00 ± 00.00	99.97 ± 00.04	100.00 ± 00.00	100.00 ± 00.00	100.00 ± 00.00
5	82.43 ± 01.01	82.14 ± 00.39	93.66 ± 00.36	99.98 ± 00.03	99.87 ± 00.19	99.95 ± 00.05	100.00 ± 00.00	99.98 ± 00.02	100.00 ± 00.00
6	82.21 ± 01.39	80.78 ± 01.22	86.68 ± 01.65	96.16 ± 02.14	98.31 ± 00.60	96.46 ± 01.49	97.96 ± 00.89	97.96 ± 01.01	98.56 ± 01.01
OA (%)	84.43 ± 00.51	85.14 ± 00.48	93.01 ± 00.31	99.27 ± 00.19	99.31 ± 00.19	98.68 ± 00.63	99.04 ± 00.33	99.59 ± 00.09	**99.70 ± 00.09**
AA (%)	83.31 ± 01.72	84.53 ± 00.31	91.98 ± 00.80	98.78 ± 00.30	99.09 ± 00.08	97.74 ± 00.81	98.92 ± 00.25	99.34 ± 00.15	**99.55 ± 00.14**
$\kappa \times 100$	79.45 ± 00.68	80.25 ± 00.65	90.55 ± 00.29	99.02 ± 00.26	99.23 ± 00.13	98.19 ± 00.57	98.97 ± 00.26	99.44 ± 00.12	**99.61 ± 00.12**

3.3.2. Visual Evaluation and Analysis

To further assess the performance of the proposed MCAITN method and the other methods, a qualitative visual analysis was conducted using representative samples from the MUUFL, Augsburg, and Trento databases; the results are illustrated in Figures 5–7.

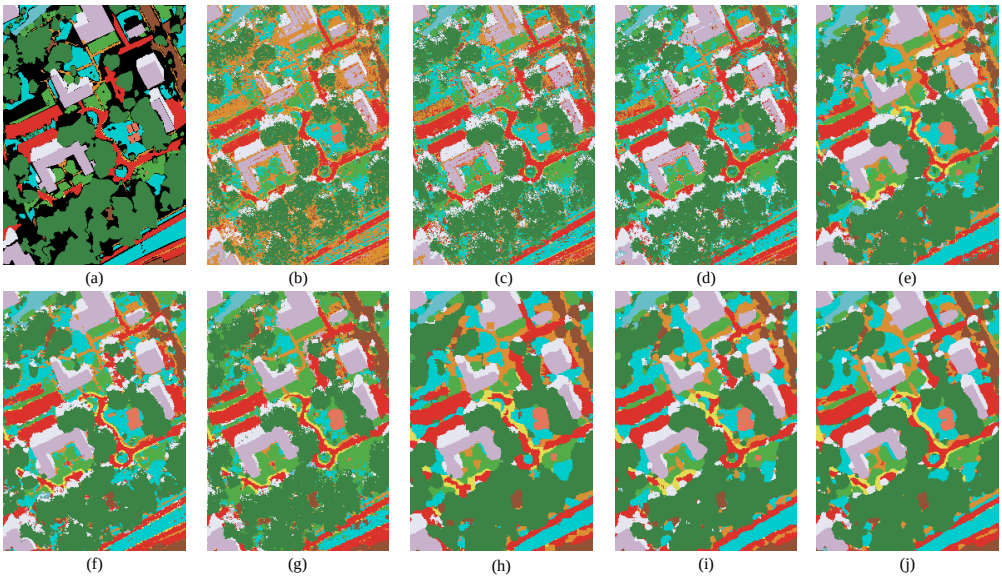

Figure 5. Classification maps of the MUUFL dataset: (**a**) ground truth, (**b**) SVM, (**c**) S2FL, (**d**) EndNet, (**e**) MDL, (**f**) LSAF, (**g**) CCRNet, (**h**) CoupledCNN, (**i**) HCT, and (**j**) MCAITN.

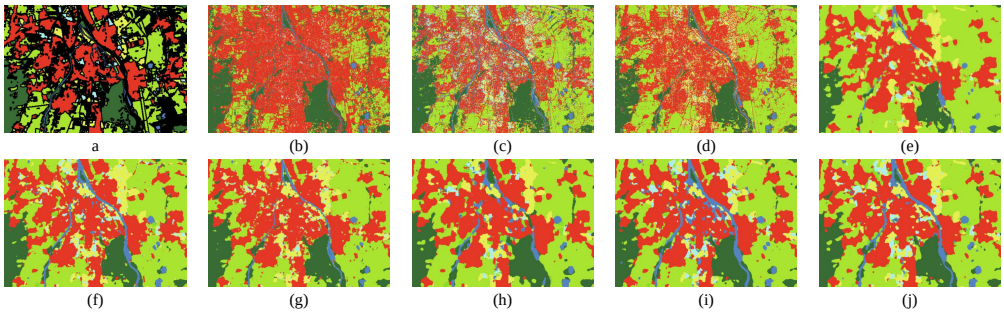

Figure 6. Classification maps of the Augsburg dataset: (**a**) ground truth, (**b**) SVM, (**c**) S2FL, (**d**) EndNet, (**e**) MDL, (**f**) LSAF, (**g**) CCRNet, (**h**) CoupledCNN, (**i**) HCT, and (**j**) MCAITN.

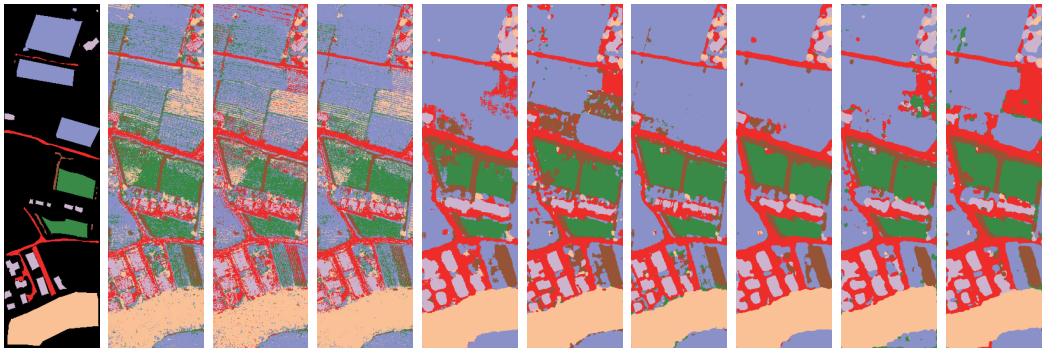

Figure 7. Classification maps of the Trento dataset: (**a**) ground truth, (**b**) SVM, (**c**) S2FL, (**d**) EndNet, (**e**) MDL, (**f**) LSAF, (**g**) CCRNet, (**h**) CoupledCNN, (**i**) HCT, and (**j**) MCAITN.

The visual results indicate that the MCAITN method is capable of producing more accurate and detailed classifications compared to the other methods. In the MUUFL dataset, for instance, the MCAITN method was able to distinctly classify various land cover types such as grassland, forests, and buildings. The other methods often struggled to differentiate between these classes, resulting in more overlapping classifications.

With the Augsburg dataset, the MCAITN method accurately captured the roads, buildings, and vegetation, especially in terms of edge delineation. The other methods either produced less clear classifications or misclassified some of the areas.

With the Trento dataset, which is characterized by high complexity and varying texture information, the MCAITN method once again demonstrated its robustness by identifying different land cover types more accurately than the other methods. The complex nature of the dataset posed a challenge for some of the methods, leading to confusion in classifications.

In summary, both the quantitative and qualitative analyses indicate that the MCAITN method provides better results compared to the current mainstream methods for HSI and LiDAR data classification tasks.

4. Discussion

4.1. Parameter Analysis

HSIs have a very high number of spectral dimensions, and directly processing these high-dimensional data significantly increases computational complexity. Moreover, adjacent bands often exhibit high correlation, leading to a large amount of redundant information, which adversely affects classification performance. Therefore, we considered a set of candidate values {10, 20, 30, and 40} for the retained spectral dimensions and fixed other hyperparameters to explore their impact on classification performance. As shown in Figure 8, with an increase in spectral dimensions, the classification performance of all three datasets initially rose and then stabilized. Considering both classification performance and computational complexity, we set the spectral dimension to 30.

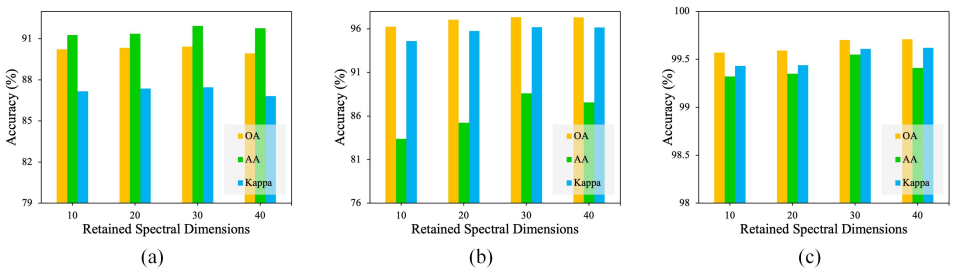

Figure 8. The impact of retained spectral dimensions on OA, AA, and the kappa coefficient. (**a**) MUUFL. (**b**) Augsburg. (**c**) Trento.

On one hand, directly processing the entire hyperspectral and LiDAR images consumes significant computational resources and memory. By dividing the images into smaller patches, we can reduce the amount of data processed at each step, thereby improving computational efficiency. On the other hand, HSI and LiDAR images have different spectral and spatial resolutions, so the patch size for these images can also impact classification performance. We fixed other hyperparameters and considered a set of candidate patch sizes {5, 7, 9, 11, and 13} for both types of image inputs. As shown in Figure 9, with the increase in the HSI patch size, the classification performance for all three datasets initially improved and then stabilized. Considering both computational complexity and classification performance, we set the HSI patch size to 11.

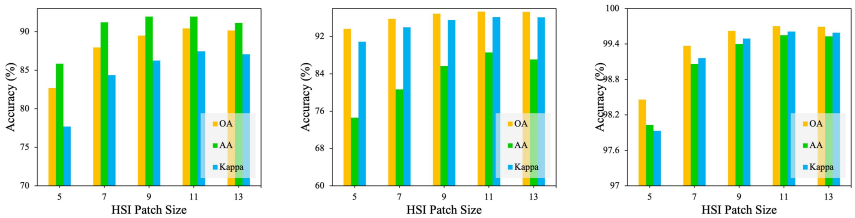

Figure 9. The impact of the HSI patch size on OA, AA, and the kappa coefficient. (**a**) MUUFL. (**b**) Augsburg. (**c**) Trento.

As illustrated in Figure 10, it is evident that the MUUFL, Augsburg, and Trento databases achieved the best classification performance with a LiDAR patch size of 5, 13, and 7, respectively.

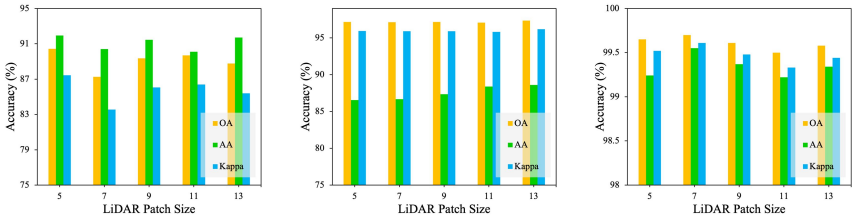

Figure 10. The impact of the LiDAR patch size on OA, AA, and the kappa coefficient. (**a**) MUUFL. (**b**) Augsburg. (**c**) Trento.

HSIs are typically high-dimensional and sparse data, and an appropriate learning rate helps the model find stable feature representations in such data, enhancing classification performance. Moreover, a suitable learning rate balances the convergence speed and stability, enabling the model to reach the optimal solution in a shorter time. We kept other hyperparameters unchanged and considered a set of candidate learning rates {1e-5, 5e-5, 1e-4, 5e-4, and 1e-3}. As shown in Figure 11, with an increasing learning rate, the AA for the MUUFL database gradually increased, while OA and kappa first increased and then decreased, achieving the best performance at 1e-4. For the Augsburg and Trento databases, the best classification performance was achieved at 5e-4 and 1e-4, respectively.

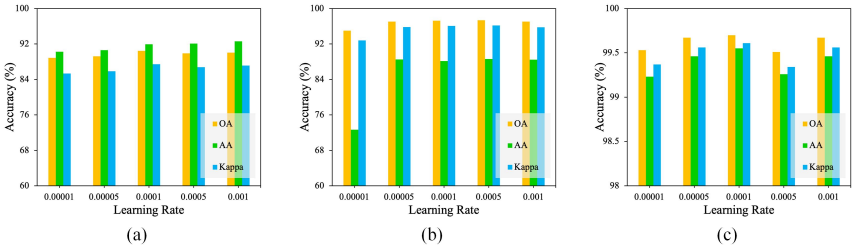

Figure 11. The impact of the learning rate on OA, AA, and the kappa coefficient. (**a**) MUUFL. (**b**) Augsburg. (**c**) Trento.

4.2. Ablation Study

To validate the effectiveness of each component in our proposed network on classification performance, we conducted ablation experiments on the MUUFL database involving four components: Conv3D, Conv2D, LiDAR-branch, and CFEA-TE. The results are listed in Table 5.

Table 5. Evaluating model components: ablation analysis with the MUUFL database (the best results are in boldface).

Cases	Component				Indicators		
	Conv3D	Conv2D	Lidar-Branch	CFEA-TE	OA (%)	AA (%)	$\kappa \times 100$
1	√	-	√	√	87.57	88.14	83.75
2	-	√	√	√	86.89	87.78	82.93
3	-	-	√	TE	55.61	50.44	43.16
4	√	√	-	TE	88.69	90.63	85.19
5	√	√	√	√	**90.43**	**91.94**	**87.45**

In Case 1 and Case 2, we removed the 2D convolution block and the 3D convolution block, respectively. The results showed a decrease in the model's classification performance in both scenarios. However, the OA, AA, and kappa in Case 1 were slightly higher than those in Case 2, suggesting that the 3D convolution block, which performs joint convolution in both spatial and spectral dimensions, is more effective at feature extraction compared to the 2D convolution block, which only performs spatial convolution. In Case 3 and Case 4, we performed classification experiments using only the LiDAR branch and the HSI branch, respectively, with the encoder utilizing a conventional transformer encoder. The results show a significant decrease in performance when only LiDAR data were used for classification, while using only HSI data yielded better classification performance. This suggests that the information contained in the LiDAR data is considerably less than that in the HSI data, and solely using LiDAR data is insufficient for classification. Finally, Case 5 represents our complete proposed classification model. Compared to Case 3 and Case 4, it achieves the best classification performance, demonstrating that our cross-feature enhanced-attention transformer encoder effectively integrates LiDAR and HSI data for joint classification. In summary, each component of the proposed MCAITN network positively contributes to the final classification performance.

5. Conclusions

This paper has introduced a novel, multi-feature, cross-attention transformer classification network named MCAITN for the joint classification of HSI and LiDAR data. The innovation of this method lies in its effective coupling of hyperspectral features with LiDAR data features through the Q, K, and V vectors in the cross-attention mechanism. It further integrates the two discriminative features iteratively, adaptively adjusting their respective advantageous features to enhance classification accuracy. The experimental results show that, compared to mainstream joint classification methods for hyperspectral and LiDAR data, the MCAITN method can better fuse the features of the two modalities, achieving an average classification accuracy improvement of about 1% at a 3% sampling rate. Another advantage of this type of method is that its architecture can easily be generalized to feature extraction for the fusion of more modalities.

In the future, directions for improvement include altering the way Q, K, and V connections are established between the two types of data markings (currently concatenation) to enable a more effective fusion of the features from the two modalities, thus further enhancing accuracy. Additionally, designing a more lightweight network architecture is also a direction for research.

Author Contributions: Conceptualization, Z.L., L.S. and Y.Z.; methodology, R.L.; software, Z.L.; validation, L.S. and R.L.; writing—original draft preparation, Z.L. and L.S.; writing—review and editing, Y.Z.; visualization, Z.L.; supervision, L.S. and Y.Z.; funding acquisition, R.L. 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 under grants No. 62076137.

Data Availability Statement: Suggested Data Availability Statements are available in Section 3.1 Dataset Description.

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

Abbreviations

MCAITN	Multi-feature, cross-attention-induced transformer network
CNNs	Convolutional neural networks
SVN	Support vector machine
RF	Random forest
RNNs	Recurrent neural networks
GANs	generative adversarial networks
LSTM	Long short-term memory
HSI	Hyperspectral image
HSIC	Hyperspectral image classification
IP-CNN	Interleaving perception convolutional neural network
Sal2RN	Saliency reinforcement network
DSHFNet	Dynamic-scale hierarchical fusion network
AMSSE-Net	Adaptive multiscale spatial–spectral enhancement network
CMSE	Cross-modal semantic enhancement
SAEs	Autoencoders
GCNs	Graph convolutional networks
CFEA	Cross-feature enhanced attention
FFN	Feed-forward network
MLP	Multi-layer perceptron
FC	Fully connected

References

1. He, L.; Li, J.; Liu, C.; Li, S. Recent Advances on Spectral–Spatial Hyperspectral Image Classification: An Overview and New Guidelines. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 1579–1597. [CrossRef]
2. Teke, M.; Deveci, H.S.; Haliloğlu, O.; Gürbüz, S.Z.; Sakarya, U. A short survey of hyperspectral remote sensing applications in agriculture. In Proceedings of the 2013 6th International Conference on Recent Advances in Space Technologies (RAST), Istanbul, Turkey, 12–14 June 2013 ; pp. 171–176. [CrossRef]
3. Agilandeeswari, L.; Prabukumar, M.; Radhesyam, V.; Phaneendra, K.L.N.B.; Farhan, A. Crop Classification for Agricultural Applications in Hyperspectral Remote Sensing Images. *Appl. Sci.* **2022**, *12*, 1670. [CrossRef]
4. Lu, B.; Dao, P.D.; Liu, J.; He, Y.; Shang, J. Recent Advances of Hyperspectral Imaging Technology and Applications in Agriculture. *Remote Sens.* **2020**, *12*, 2659. [CrossRef]
5. Camps-Valls, G.; Tuia, D.; Bruzzone, L.; Benediktsson, J.A. Advances in Hyperspectral Image Classification: Earth Monitoring with Statistical Learning Methods. *IEEE Signal Process. Mag.* **2014**, *31*, 45–54. [CrossRef]
6. Stuart, M.B.; Davies, M.; Hobbs, M.J.; Pering, T.D.; McGonigle, A.J.S.; Willmott, J.R. High-Resolution Hyperspectral Imaging Using Low-Cost Components: Application within Environmental Monitoring Scenarios. *Sensors* **2022**, *22*, 4652. [CrossRef]
7. Weber, C.; Aguejdad, R.; Briottet, X.; Avala, J.; Fabre, S.; Demuynck, J.; Zenou, E.; Deville, Y.; Karoui, M.; Benhalouche, F.; et al. Hyperspectral Imagery for Environmental Urban Planning. In Proceedings of the IGARSS 2018—2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 1628–1631. [CrossRef]
8. Brabant, C.; Alvarez-Vanhard, E.; Laribi, A.; Morin, G.; Thanh Nguyen, K.; Thomas, A.; Houet, T. Comparison of Hyperspectral Techniques for Urban Tree Diversity Classification. *Remote Sens.* **2019**, *11*, 1269. [CrossRef]
9. Nisha, A.; Anitha, A. Current Advances in Hyperspectral Remote Sensing in Urban Planning. In Proceedings of the 2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT), Kannur, India, 11–12 August 2022; pp. 94–98. [CrossRef]
10. Shimoni, M.; Haelterman, R.; Perneel, C. Hypersectral Imaging for Military and Security Applications: Combining Myriad Processing and Sensing Techniques. *IEEE Geosci. Remote Sens. Mag.* **2019**, *7*, 101–117. [CrossRef]
11. Zhao, J.; Zhou, B.; Wang, G.; Ying, J.; Liu, J.; Chen, Q. Spectral Camouflage Characteristics and Recognition Ability of Targets Based on Visible/Near-Infrared Hyperspectral Images. *Photonics* **2022**, *9*, 957. [CrossRef]
12. Zhao, G.; Ye, Q.; Sun, L.; Wu, Z.; Pan, C.; Jeon, B. Joint Classification of Hyperspectral and LiDAR Data Using a Hierarchical CNN and Transformer. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 1–16. [CrossRef]

13. Sun, L.; He, C.; Zheng, Y.; Wu, Z.; Jeon, B. Tensor cascaded-rank minimization in subspace: A unified regime for hyperspectral image low-level vision. *IEEE Trans. Image Process.* **2022**, *32*, 100–115. [CrossRef]
14. Sun, L.; Cao, Q.; Chen, Y.; Zheng, Y.; Wu, Z. Mixed noise removal for hyperspectral images based on global tensor low-rankness and nonlocal SVD-aided group sparsity. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 1–17. [CrossRef]
15. Song, T.; Zeng, Z.; Gao, C.; Chen, H.; Li, J. Joint Classification of Hyperspectral and LiDAR Data Using Height Information Guided Hierarchical Fusion-and-Separation Network. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 1–15. [CrossRef]
16. 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]
17. Song, W.; Li, S.; Fang, L.; Lu, T. Hyperspectral Image Classification with Deep Feature Fusion Network. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 3173–3184. [CrossRef]
18. Yang, H.; Yu, H.; Zheng, K.; Hu, J.; Tao, T.; Zhang, Q. Hyperspectral Image Classification Based on Interactive Transformer and CNN With Multilevel Feature Fusion Network. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 1–5. [CrossRef]
19. Yang, J.; Li, A.; Qian, J.; Qin, J.; Wang, L. A Hyperspectral Image Classification Method Based on Pyramid Feature Extraction with Deformable–Dilated Convolution. *IEEE Geosci. Remote Sens. Lett.* **2024**, *21*, 1–5. [CrossRef]
20. Cao, X.; Yao, J.; Xu, Z.; Meng, D. Hyperspectral Image Classification with Convolutional Neural Network and Active Learning. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 4604–4616. [CrossRef]
21. Xue, Z.; Yu, X.; Tan, X.; Liu, B.; Yu, A.; Wei, X. Multiscale Deep Learning Network with Self-Calibrated Convolution for Hyperspectral and LiDAR Data Collaborative Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–16. [CrossRef]
22. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [CrossRef]
23. Baassou, B.; He, M.; Mei, S. An accurate SVM-based classification approach for hyperspectral image classification. In Proceedings of the 2013 21st International Conference on Geoinformatics, Kaifeng, China, 20–22 June 2013; pp. 1–7. [CrossRef]
24. Xie, L.; Li, G.; Xiao, M.; Peng, L.; Chen, Q. Hyperspectral Image Classification Using Discrete Space Model and Support Vector Machines. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 374–378. [CrossRef]
25. Amini, S.; Homayouni, S.; Safari, A. Semi-supervised classification of hyperspectral image using random forest algorithm. In Proceedings of the 2014 IEEE Geoscience and Remote Sensing Symposium, Quebec City, QC, Canada, 13–18 July 2014; pp. 2866–2869. [CrossRef]
26. Wang, S.; Dou, A.; Yuan, X.; Zhang, X. The airborne hyperspectral image classification based on the random forest algorithm. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; pp. 2280–2283. [CrossRef]
27. Zhang, Y.; Cao, G.; Li, X.; Wang, B. Cascaded Random Forest for Hyperspectral Image Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 1082–1094. [CrossRef]
28. Yang, X.; Ye, Y.; Li, X.; Lau, R.Y.K.; Zhang, X.; Huang, X. Hyperspectral Image Classification with Deep Learning Models. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 5408–5423. [CrossRef]
29. 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]
30. Ullah, F.; Ullah, I.; Khan, R.U.; Khan, S.; Khan, K.; Pau, G. Conventional to Deep Ensemble Methods for Hyperspectral Image Classification: A Comprehensive Survey. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 3878–3916. [CrossRef]
31. Zhu, L.; Chen, Y.; Ghamisi, P.; Benediktsson, J.A. Generative Adversarial Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 5046–5063. [CrossRef]
32. Deng, X.; Dragotti, P.L. Deep Convolutional Neural Network for Multi-Modal Image Restoration and Fusion. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 3333–3348. [CrossRef] [PubMed]
33. 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*, 1–16. [CrossRef]
34. 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*, 1–14. [CrossRef]
35. Liang, L.; Zhang, S.; Li, J. Multiscale DenseNet Meets With Bi-RNN for Hyperspectral Image Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 5401–5415. [CrossRef]
36. Hu, W.S.; Li, H.C.; Pan, L.; Li, W.; Tao, R.; Du, Q. Spatial–Spectral Feature Extraction via Deep ConvLSTM Neural Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 4237–4250. [CrossRef]
37. Huang, L.; Chen, Y. Dual-Path Siamese CNN for Hyperspectral Image Classification with Limited Training Samples. *IEEE Geosci. Remote Sens. Lett.* **2021**, *18*, 518–522. [CrossRef]
38. Yu, C.; Han, R.; Song, M.; Liu, C.; Chang, C.I. Feedback Attention-Based Dense CNN for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–16. [CrossRef]
39. Bhatti, U.A.; Yu, Z.; Chanussot, J.; Zeeshan, Z.; Yuan, L.; Luo, W.; Nawaz, S.A.; Bhatti, M.A.; Ain, Q.U.; Mehmood, A. Local Similarity-Based Spatial—Spectral Fusion Hyperspectral Image Classification with Deep CNN and Gabor Filtering. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–15. [CrossRef]

40. Li, J.; Liu, Y.; Song, R.; Li, Y.; Han, K.; Du, Q. Sal²RN: A Spatial—Spectral Salient Reinforcement Network for Hyperspectral and LiDAR Data Fusion Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 1–14. [CrossRef]
41. Zhang, Y.; Xu, S.; Hong, D.; Gao, H.; Zhang, C.; Bi, M.; Li, C. Multimodal Transformer Network for Hyperspectral and LiDAR Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 1–17. [CrossRef]
42. Wang, X.; Zhu, J.; Feng, Y.; Wang, L. MS2CANet: Multiscale Spatial—Spectral Cross-Modal Attention Network for Hyperspectral Image and LiDAR Classification. *IEEE Geosci. Remote Sens. Lett.* **2024**, *21*, 1–5. [CrossRef]
43. Gao, H.; Feng, H.; Zhang, Y.; Xu, S.; Zhang, B. AMSSE-Net: Adaptive Multiscale Spatial–Spectral Enhancement Network for Classification of Hyperspectral and LiDAR Data. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 1–17. [CrossRef]
44. Hong, D.; Gao, L.; Hang, R.; Zhang, B.; Chanussot, J. Deep Encoder—Decoder Networks for Classification of Hyperspectral and LiDAR Data. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]
45. Du, X.; Zheng, X.; Lu, X.; Doudkin, A.A. Multisource Remote Sensing Data Classification with Graph Fusion Network. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 10062–10072. [CrossRef]
46. Dam, T.; Anavatti, S.G.; Abbass, H.A. Mixture of Spectral Generative Adversarial Networks for Imbalanced Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]
47. Zhang, Y.; Peng, Y.; Tu, B.; Liu, Y. Local Information Interaction Transformer for Hyperspectral and LiDAR Data Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *16*, 1130–1143. [CrossRef]
48. Sun, L.; Zhang, H.; Zheng, Y.; Wu, Z.; Ye, Z.; Zhao, H. MASSFormer: Memory-Augmented Spectral-Spatial Transformer for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 8257–8268. [CrossRef]
49. Fu, L.; Zhang, D.; Ye, Q. Recurrent thrifty attention network for remote sensing scene recognition. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 1–17. [CrossRef]
50. Ding, K.; Lu, T.; Fu, W.; Li, S.; Ma, F. Global–Local Transformer Network for HSI and LiDAR Data Joint Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–13. [CrossRef]
51. Zhang, M.; Gao, F.; Zhang, T.; Gan, Y.; Dong, J.; Yu, H. Attention Fusion of Transformer-Based and Scale-Based Method for Hyperspectral and LiDAR Joint Classification. *Remote Sens.* **2023**, *15*, 650. [CrossRef]
52. Ni, K.; Wang, D.; Zheng, Z.; Wang, P. MHST: Multiscale Head Selection Transformer for Hyperspectral and LiDAR Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 5470–5483. [CrossRef]
53. Yang, J.X.; Zhou, J.; Wang, J.; Tian, H.; Liew, A.W.C. LiDAR-Guided Cross-Attention Fusion for Hyperspectral Band Selection and Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 1–15. [CrossRef]
54. Roy, S.K.; Sukul, A.; Jamali, A.; Haut, J.M.; Ghamisi, P. Cross Hyperspectral and LiDAR Attention Transformer: An Extended Self-Attention for Land Use and Land Cover Classification. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 1–15. [CrossRef]
55. Hong, D.; Hu, J.; Yao, J.; Chanussot, J.; Zhu, X.X. Multimodal remote sensing benchmark datasets for land cover classification with a shared and specific feature learning model. *ISPRS J. Photogramm. Remote Sens.* **2021**, *178*, 68–80. [CrossRef]
56. Hong, D.; Gao, L.; Yokoya, N.; Yao, J.; Chanussot, J.; Du, Q.; Zhang, B. More Diverse Means Better: Multimodal Deep Learning Meets Remote-Sensing Imagery Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 4340–4354. [CrossRef]
57. Feng, M.; Gao, F.; Fang, J.; Dong, J. Hyperspectral and Lidar Data Classification Based on Linear Self-Attention. In Proceedings of the 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, Brussels, Belgium, 11–16 July 2021; pp. 2401–2404. [CrossRef]
58. Wu, X.; Hong, D.; Chanussot, J. Convolutional Neural Networks for Multimodal Remote Sensing Data Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–10. [CrossRef]
59. Hang, R.; Li, Z.; Ghamisi, P.; Hong, D.; Xia, G.; Liu, Q. Classification of Hyperspectral and LiDAR Data Using Coupled CNNs. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 4939–4950. [CrossRef]