remote sensing

MDPI

Article

A Hierarchical Coarse–Fine Adaptive Fusion Network for the Joint Classification of Hyperspectral and LiDAR Data

Haizhu Pan [1,2,*], Xuan Li [1], Haimiao Ge [1,2], Liguo Wang [3] and Cuiping Shi [4]

[1] College of Computer and Control Engineering, Qiqihar University, Qiqihar 161000, China; 2022935773@qqhru.edu.cn (X.L.); 01557@qqhru.edu.cn (H.G.)
[2] Heilongjiang Key Laboratory of Big Data Network Security Detection and Analysis, Qiqihar University, Qiqihar 161000, China
[3] College of Information and Communication Engineering, Dalian Nationalities University, Dalian 116000, China; wangliguo@hrbeu.edu.cn
[4] College of Information Engineering, Huzhou University, Huzhou 313000, China; shicuiping@qqhru.edu.cn
* Correspondence: panhaizhu@qqhru.edu.cn

Abstract: Hyperspectral image (HSI) and light detection and ranging (LiDAR) data have gained significant attention due to their excellent complementarity, which can be combined to improve the accuracy of land cover classification. However, there are still many challenges, such as unrelated physical characteristics, different data structures, and a lack of labeled samples. Many methods fail to leverage the full potential of multi-source data, particularly hierarchical complementary information. To address these problems, a hierarchical coarse–fine adaptive (HCFA) fusion network with dynamic convolution and a transformer is proposed for multi-source remote sensing land cover classification. Although fusing hierarchical information can improve the classification accuracy of the model, improper hierarchical feature selection and optimization may negatively affect the classification results. Therefore, a coarse–fine mutual learning strategy is proposed to dynamically fuse hierarchical information. Additionally, the disparity between multi-source data continues to prevent the realization of effective fusion. To tackle this challenge, cross-tokenization and cross-token attention are implemented to enhance information interaction. Furthermore, to improve the model representation with limited computational cost, we combine the advantages of dynamic convolution with a transformer. Validation on three standard datasets demonstrates that HCFA achieves high accuracy with just 1% of the training set while maintaining low computational costs.

Keywords: coarse–fine grained features; dynamic convolution; transformer; mutual learning; multi-source

check for updates

Citation: Pan, H.; Li, X.; Ge, H.; Wang, L.; Shi, C. A Hierarchical Coarse–Fine Adaptive Fusion Network for the Joint Classification of Hyperspectral and LiDAR Data. *Remote Sens.* **2024**, *16*, 4029. https://doi.org/10.3390/rs16214029

Academic Editor: Salah Bourennane

Received: 1 September 2024
Revised: 27 October 2024
Accepted: 28 October 2024
Published: 30 October 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The availability of multi-source remote sensing (RS) image data for the same geographic area is gradually expanding as RS imaging technology continues to advance, providing researchers with a broader and more diverse data repository [1]. Different sensor technologies possess the capability to capture diverse characteristics of land cover materials, thus yielding a variety of RS data. For instance, hyperspectral imagery (HSI) encapsulates abundant spectral and geospatial details [2], whereas light detection and ranging (LiDAR) imagery furnishes data on elevation and object height [3]. At the same time, synthetic aperture radar (SAR) imagery offers geometric intricacies regarding both the amplitude and phase of ground objects [4]. Through the integration of diverse data sources, we gain access to more intricate information, allowing for the creation of more comprehensive feature representations. In particular, among these data types, the complementary strengths between HSI and LiDAR data have attracted a great deal of attention. HSI captures rich spectral and spatial information and shows their unique potential for land cover targets of similar depth and intensity but different materials. Nevertheless, when it comes to classifying

certain objects, such as roads and rooftops, the limitations of similar reflected spectra pose challenges to accurate classification [5]. On the other hand, LiDAR data provides accurate measurements of target distances and elevation information, making it ideally suited to distinguishing between objects that have different heights, such as roads and roofs [6]. However, distinguishing objects of the same height, such as grass and roads, becomes more challenging. HSI and LiDAR data exhibit distinct characteristics. HSI provides detailed spectral information, while LiDAR offers elevation data, making them complementary for land cover classification. The integrated classification approach, which combines the spatial–spectral information of HSI with the elevation data from LiDAR, capitalizing on the advantages of each data type while addressing their respective limitations, has emerged as a prominent trend in contemporary research. This approach finds application across a multitude of domains, including target detection [7,8], semantic segmentation [9], land cover classification [10,11], environmental monitoring [12], precision agriculture [13], urban planning [14], medical diagnosis [15], and mineral exploration [16].

With the rapid advancement of RS imaging technologies, some scholars have initiated exploration into the classification of multi-source RS data. Due to their excellent complementarity, the fusion of HSI and LiDAR data has garnered significant attention. Gu et al. [17] used the graph structure fusion method to fuse graphs consisting of different heterogeneous feature sets and used a support vector machine (SVM) for classification with the Gaussian radial basis function kernel. Rasti et al. [18] employed sparse low-rank techniques to estimate low-rank fusion features and utilized a random forest (RF) classifier for classification. Ghamisi et al. [19] designed a multi-sensor composite extreme learning machine (ELM) and estimated features with hyperspectral Stein's unbiased risk estimator for efficient fusion of heterogeneous features. However, traditional machine learning methods rely on handcrafted features and may not adequately explore deep features, thus failing to effectively capture the complex nonlinear relationships between terrain features in HSI and LiDAR data. This limitation could constrain their ability to achieve higher classification accuracy in hyperspectral and LiDAR fusion tasks.

Deep learning algorithms also show considerable potential in collectively classifying both HSI and LiDAR data. To achieve a more compact fusion of information and more effective information transmission, Hong et al. [20] introduced an architecture for deep encoder–decoder networks. The model has successfully inspired us to consider multi-source data classification methods, but due to its simple baseline, it is difficult to obtain high network performance. Moreover, owing to its simplistic baseline, it has encountered challenges in attaining higher network performance. Xu et al. [21] introduced a dual-branch fusion network for classifying HSI and LiDAR data. Leveraging the dual-branch architecture, the network extracts spectral–spatial features from HSI and spatial features from LiDAR separately. Ultimately, by cascading the fusion of features from multiple sources, the classification task is completed. Nevertheless, the feature integration in this model is rather conventional and fails to facilitate effective information interaction among heterogeneous data. To facilitate the information exchange between heterogeneous data sources, Hang et al. [22] introduced a coupled convolutional neural network (CNN). They utilized both feature-level fusion and decision-level fusion strategies concurrently to comprehensively integrate heterogeneous information and improve classification performance. Additionally, for multispectral image (MSI) and SAR data, various self-supervised contrastive learning frameworks have been proposed, effectively alleviating the issue of limited labeled samples in real-world applications [23,24]. The fusion strategies and classification frameworks used vary for heterogeneous RS data. Hong et al. [25] proposed a unified multimodal deep learning framework, which extensively investigates and summarizes multiple-stage and multi-source fusion strategies, thereby offering comprehensive guidance for multimodal feature learning and data fusion processing.

All the methods mentioned above are methods based on the CNN framework. Although CNNs excel at capturing local features of images, they cannot typically process the global contextual information of images. Furthermore, the sequential attributes of spectral

features in remote sensing data are not well exploited and represented by CNNs. In contrast, using a transformer addresses these limitations well. Hong et al. [26] developed a novel spectral transformer model that successfully captures the sequential attributes of spectral features. This model constructs cross-layer transformer encoder modules and captures spectral features from neighboring bands within groups. However, these works only utilize transformer technology for spectral information processing. By introducing the Fourier transform into the transformer, the model size and efficiency of the deep transformer can be enhanced. Zhao et al. effectively reduced redundancy and information loss from shallow to deep layers by connecting contextual features in multiple frequency domains [27]. Sun et al. [28] proposed a spectral–spatial feature tokenization transformer approach, leveraging transformer encoders to capture spectral–spatial features as well as higher-level semantic features. However, the method based on a single transformer suffers from high computational cost and insufficient local feature extraction capability [29]. Therefore, hybrid models have gradually attracted attention. Li et al. [30] proposed a hybrid model combining a CNN and Transformer and achieved excellent results on hyperspectral datasets. The study shows that the hybrid model can not only effectively improve the accuracy but also reduce the complexity of the model.

The above methods do not enable enough interaction between heterogeneous data, limiting the use of their complementary strengths. The mutual learning strategy increases the information interaction between the data. However, a single-hierarchical information exchange is still insufficient. To address this, hierarchical complementary information can be obtained by combining both coarse-grained and fine-grained information through the mutual learning strategy.

Building on the above research, a hierarchical coarse–fine adaptive (HCFA) fusion network with dynamic convolution and a cross-tokenization transformer was developed. In this framework, the coarse–fine joint representation network is combined with a feature fusion structure to elevate the classification accuracy of multi-source data fusion. A method combining HSI mixed convolution with a LiDAR 2D dynamic convolution branch network is designed, achieving simultaneous extraction of spectral features from HSI and elevation features from LiDAR. Subsequently, the parallel cross-tokenization module converts the two sets of features into feature tokens. Following this, two sets of feature tokens are used as inputs to the transformer encoder with cross-token attention for feature learning. Finally, the coarse and fine features are fused and mutually guided to obtain comprehensive classification information. The main contributions are outlined as follows:

(1) We design a coarse–fine mutual learning strategy that fuses coarse decisions and fine decisions and continuously adjusts the weights of both using adaptive loss. This approach enables discriminative classification decisions, synergistically enhancing the representational capacity of the network architecture.
(2) A cross-tokenization method is designed to effectively fuse information from HSI and LiDAR data in a novel manner. Cross-token attention is designed to enhance feature interaction learning among HSI and LiDAR feature tokens, in which the semantic correlations within dual-source RS data are thoroughly explored.
(3) To comprehensively extract and understand spatial–spectral features from HSI and elevation features from LiDAR, we design a two-branch coarse–fine mutual learning network. By combining the advantages of a dynamic CNN and transformer, the computational cost is effectively controlled, and more comprehensive and effective features are obtained.
(4) Extensive experiments conducted on three real HSI and LiDAR datasets demonstrate that this network framework is effective in alleviating the few sample classification problems of multi-source RS data and outperforms other state-of-the-art methods.

The rest of this paper is structured as follows: Section 2 delves into the related work, while Section 3 outlines the framework of the proposed method. Experimental results are provided in Section 4. Section 5 provides some discussion. Finally, Section 6 provides the conclusions and future prospects of the article.

2. Related Works

2.1. Multi-Source Remote Sensing Data Fusion Method

Multi-source RS data fusion is the process of integrating information from different types of RS data sources to achieve a more comprehensive analysis [31]. The technique is particularly suitable for fusing data with different spatial and spectral resolutions, such as HSI with LiDAR data or HSI with MSI. To effectively integrate these diverse data sources for classification tasks, multi-source RS data fusion strategies are generally categorized into three main types: pixel-level, feature-level, and decision-level fusion [32]. Pixel-level fusion is performed directly on RS raw data to enhance data quality through the fusion of directional observation data [33]. This fusion method retains both original data information and can improve the accuracy of the classifier. However, this method is susceptible to noise. Feature-level fusion involves integrating features extracted from multiple sources of data to generate new features or feature vectors, facilitating subsequent applications [34]. Compared with pixel-level fusion, feature-level fusion reduces the data dimensions, decreases the computational complexity, and exhibits greater flexibility in dealing with noise. Methods for feature-level fusion can be categorized into single-level feature fusion structures and multi-level feature fusion structures [25]. Single-level feature fusion methods entail merging features extracted from various sensors, either early or late in the feature extraction stage, at different locations [35]. Nevertheless, these single-level fusion methods often neglect the correlation and heterogeneity among different sensor data, failing to fully leverage all the available information. The multi-level feature fusion approach involves integrating hierarchical features extracted from different sensors at various stages of the feature extraction process. Multi-level feature fusion fully utilizes features at each level, enhances information interaction and fusion effects, and can better utilize complementary information from different data sources [36]. Decision-level fusion involves interpreting land cover features using various sensor data to obtain separate categories of land cover features. Subsequently, fusion is carried out utilizing specific decision rules. Its primary purpose is to address the issue of inconsistent results generated by disparate datasets, thereby yielding more dependable decision outcomes [37]. Given the distinct imaging mechanisms of these data sources, feature-level and decision-level fusion techniques are commonly employed [38]. The article introduces a hybrid fusion coarse–fine architecture that combines feature-level and decision-level fusion methods. This approach enables the comprehensive extraction and integration of features from multiple sources and perspectives. It dynamically adjusts the interaction between different data and perspectives, thereby obtaining a robust and significantly effective model.

2.2. Dynamic Convolution

CNN has found widespread application in multi-source RS image fusion classification due to its strong feature expression capability and natural adaptability to visual data [39]. Lightweight convolutional neural networks have emerged to reduce computational costs. However, lightweight networks also lead to performance deterioration because their constrained computational budget limits the depth and width of the CNN, resulting in limited representation capability [40]. Dynamic convolution can increase the representation capability of a model without increasing the depth and width of the network [41]. Dynamic convolution dynamically aggregates multiple convolutional kernels through attention mechanisms. Here, attention dynamically adjusts the weights of individual convolutional kernels based on the input, thereby generating adaptive dynamic convolutions. Non-linearly superimposing convolutional kernels through attention enhances the representational capability. Compared with traditional static convolution, dynamically stacking multiple convolution kernels based on attention not only significantly improves the representation capability, but the additional computational cost is also small. Figure 1 illustrates the structure of the dynamic convolutional layer. Dynamic convolution utilizes a set of k parallel convolutional kernels per layer instead of just a single kernel. These convolutional kernels dynamically aggregate each individual input x through attention π_k

that is relevant to the input. The biases are aggregated using identical attention mechanisms. The definition of dynamic convolution can be formulated as follows:

$$\begin{cases} [\pi_1, \pi_2, \cdots, \pi_k] = f_\pi(x) \\ d = \sum_{i=1}^{k} \pi_i(x)(W_i x + b_i), \end{cases} \tag{1}$$

where x and d are the input and output of dynamic convolution, W and b represent the weight matrix and bias vector, respectively, $f_\pi(\cdot)$ represents an attention block composed of a global average pooling layer, a ReLU activation function, and two fully connected layers, and $\{\pi_i\}_{i=1}^{k}$ denotes the generative attentional weights for the attention block, where k is the number of convolution kernels.

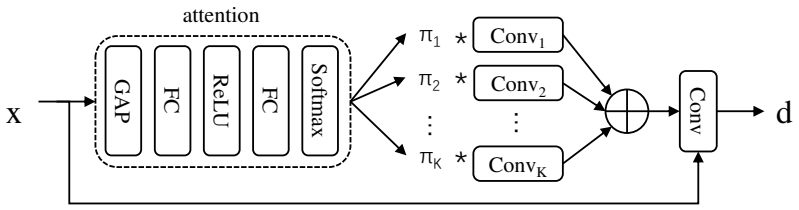


Figure 1. Structure of dynamic convolutions.

3. Methodology

In this paper, we consider a hyperspectral image denoted as $X_H \in R^{M\times N\times B}$ and the corresponding LiDAR image denoted as $X_L \in R^{M\times N\times 1}$, where M and N denote the width and height of the two images, respectively, while B denotes the number of raw HSI channels. After performing a dimensionality reduction operation on the input HSI, the number of compressed HSI channels reduced to b is denoted as $X_H \in R^{M\times N\times b}$. To fully leverage the spatial information of the image, an s×s dimensional adjacency cube centered on each pixel is used for classification. Each pixel undergoes both 3D and 2D block extraction procedures to obtain the small block cube $X_H^P \in R^{s\times s\times b}$ and the small block $X_L^P \in R^{s\times s}$, respectively. The index of the central pixel is used to label each patch. For edge pixels, we perform padding with a width of $(s-1)/2$. We assume that there are T labeled training samples, denoted as $X = \left\{X^i | (X_H^i, X_L^i)\right\}_{i=1}^{T}$ with corresponding labels as $Y = \left\{y^i\right\}_{i=1}^{T}$, where $y^i \in \{1, 2, \ldots, C\}$ and C represents the quantity of land cover types in RS data. Following the removal of zero-labelled pixel blocks, the remaining sample blocks are partitioned into training, validation, and testing sets. The HCFA method consists of three modules: a coarse feature learning (CFL) module, a fine feature learning (FFL) module, and a coarse–fine mutual-guided (CFM) module. The detailed architecture of the proposed HCFA is shown in Figure 2. The pseudo-code for training the HCFA model for HSI and LiDAR data fusion classification is shown in Algorithm 1.

Coarse-grained and fine-grained classification can provide levels from broad to fine in image classification tasks, allowing the model to more accurately distinguish and recognize subtle differences in images [42]. The HCFA network is a hierarchical structure in which the lower layers are mainly responsible for providing coarse-grained information, while the higher layers are mainly responsible for representing fine-grained information. Coarse-grained features refer to broad, general characteristics that are used to perform initial classifications of land cover categories. These features encompass major, easily distinguishable categories and provide an overarching view of the landscape, facilitating the identification of dominant land cover categories before delving into finer details. Fine-grained features refer to specific, detailed characteristics within a broader category identified during the coarse-grained classification phase. In Figure 2, an initial coarse-grained classification is first performed using dynamic convolution to identify the main land cover types and areas of change within a wide area. Subsequently, the features identified by the dynamic convolution are used as inputs to the transformer model to perform a

fine-grained classification of the areas identified in the initial classification to obtain more detailed information.

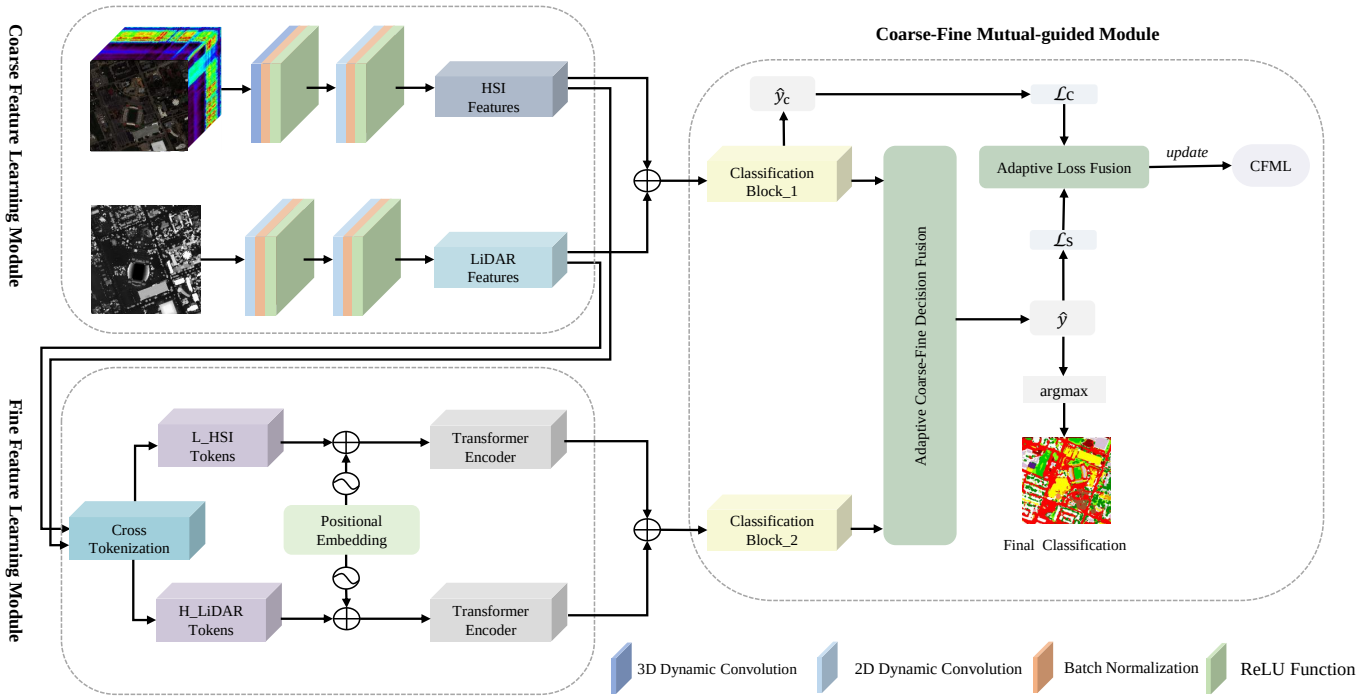


Figure 2. Overall structure of the coarse–fine mutual learning fusion network.

Algorithm 1 Pseudo-code of the HCFA.

X_H, X_L: Dimension-reduced HSI and corresponding LiDAR data;
Parameters: Epochs = 100, Batch Size = 64, training sample rate = 1%;
X_H^P, X_L^P: preprocess original input data to 7×7 patch data; Y: ground-truth label
for(X_H^P, X_L^P, Y) in data_loader: # load a mini-batch data
    # Coarse Feature Learning (CFL) Module
    # $X_H^{DyConv} : 7 \times 7 \times 64$; $X_L^{DyConv} : 7 \times 7 \times 64$
    $X_H^{DyConv}, X_L^{DyConv}$ = Feature_HSI(X_H^P), Feature_LiDAR(X_L^P) # Equations (2) and (3)
    # Fine Feature Learning (FFL) Module
    # HSI and LiDAR Feature Cross-Tokenization
    # T_{HL}^p, T_{LH}^p : (num_tokens + 1) $\times$ 64
    T_{HL}^p, T_{LH}^p = Cross-Tokenization$(X_H^{DyConv}, X_L^{DyConv})$ # Equations (4)–(13)
    # LiDAR and HSI Cross Token Attention (CTA)
    # Y_{LH}, Y_{HL} : (num_tokens + 1) $\times$ 64
    $Y_{LH}, Y_{HL} = \text{CTA}_1(T_{LH}^p), \text{CTA}_2(T_{HL}^p)$ # Equations (14)–(20)
    # Coarse–Fine Mutual-guided (CFM) Module
    # Adaptive Coarse–Fine Decision Fusion (ADF)
    CLS_*: map **(lambda t: t[:0])**; CLF: classification block
    $\hat{y}_c = \text{CLF}_1(X_H^{DyConv}, X_L^{DyConv})$
    $\hat{y}_f = \text{CLF}_2(CLS_y_{LH}, CLS_y_{HL})$
    $\hat{y} = \text{ADF}(\hat{y}_c, \hat{y}_f)$ # Equations (21)
    # Adaptive Loss Fusion (ALF)
    loss = ALF$(\hat{y}_c, \hat{y})$ # Equations (22), (23), and (24)
    # Update Trainable Parameters
return Classification Map;

3.1. CFL Module

Based on the outstanding ability of the CNN to simulate spatial contextual features, it performs excellently in HSI classification tasks. Additionally, the feature extraction mechanism of a 3D CNN enables local modeling of high-dimensional spectral information

and flexible control over the depth of output feature maps. To enhance the representation capability of convolution while maintaining a lower computational budget, we introduce dynamic convolution. Unlike the static and shared parameters in conventional convolution, dynamic convolution dynamically combines multiple parallel convolutional kernels based on input attention, allowing adaptive learning of the weight associated with each input. In the coarse feature learning module, we employ dual-branch dynamic convolution to extract spatial–spectral features and elevation information separately from HSI data and LiDAR data.

As shown in Figure 2, a sequence of consecutive layers utilizing 3D dynamic convolution and 2D dynamic convolution is employed to extract distinct spatial and spectral features from HSI data. Each HSI patch cube, denoted as X_H^P of size $(s \times s \times b)$, is utilized as the input training data for the 3D dynamic convolutional layer. In the 3D dynamic convolutional layer, the kernel size is set to $(3 \times 3 \times 3)$. After the 3D dynamic convolution operation, the generated feature cube is flattened along the spatial dimensions into a vector, which serves as the input for the next layer. Subsequently, a 2D dynamic convolution operation is performed using a (3×3) kernel, resulting in 64 two-dimensional feature maps as output. Unlike the processing of HSI data, LiDAR data employs two 2D dynamic convolutional layers for extracting elevation features. Each LiDAR patch X_L^P with size $(s \times s)$ is used as input to the two 2D dynamic convolution layers. In these two layers, the kernel size is set to (3×3). In addition, batch normalization (BN) and rectified linear unit (ReLU) activation layers are applied consecutively after each dynamic convolutional layer. BN is utilized to alleviate the overfitting that results from the limited number of training samples and enhance training efficiency. Additionally, ReLU serves a crucial function by introducing nonlinearity to the output feature mapping, ensuring the seamless and effective propagation of the loss gradient. The HSI and LiDAR features obtained by the dynamic convolution component can be represented as follows:

$$X_H^{Dyconv} = \mathrm{DyConv}_{2D}(\mathrm{DyConv}_{3D}(X_H^P)), \tag{2}$$

$$X_L^{DyConv} = \mathrm{DyConv}_{2D}(\mathrm{DyConv}_{2D}(X_L^P)), \tag{3}$$

where DyConv_{2D} denotes a 2D dynamic convolution component consisting of 2D dynamic convolution, BN, and ReLU. DyConv_{3D} denotes a 3D dynamic convolution component consisting of 3D dynamic convolution, BN, and ReLU. X_H^{DyConv} and X_L^{DyConv} represent heterogeneous coarse-grained feature information obtained through two different layers of dynamic convolution.

3.2. FFL Module

Fine-grained features play a crucial role in image processing. Although the CFM module can extract some local information, such as edges and textures in the image, it cannot model the fine-grained features well. To address this issue, we input the coarse-grained features into the transformer model to obtain more abundant fine-grained information. The fine feature learning module consists of HSI and LiDAR feature cross-tokenization and a transformer encoder with cross-token attention, aiming to extract fine-grained information for heterogeneous RS data fusion.

3.2.1. HSI and LiDAR Feature Cross-Tokenization

To obtain fine-grained information in multi-source RS images, we combine feature-level fusion and tokenization, proposing a novel cross-tokenization (CT) method. The traditional approach is to use HSI data as input, with other source data connected to the HSI data to capture complementary features that enrich the feature representation.. This fusion method increases the complexity of linear projection and may lead to opportunities for overfitting. The CT method effectively reduces the projection complexity and improves the accuracy and efficiency of data fusion by encoding different source data independently at the initial stage and then integrating the multi-source information with a cross-attention

mechanism. The method we proposed continues the dual-branch strategy. In the HSI branch, LiDAR features serve as complementary features, with the LiDAR characteristics abstracted into an external classification (CLS) token. This approach enriches the abstract representation of the CLS token without adding any computational overhead. The feature map is unfolded into a set of vectors, as shown in Figure 3. The $(7 \times 7 \times 64)$ HSI cube blocks are unfolded into feature blocks of shape (49×64). Similarly, the LiDAR branch follows the same procedure.

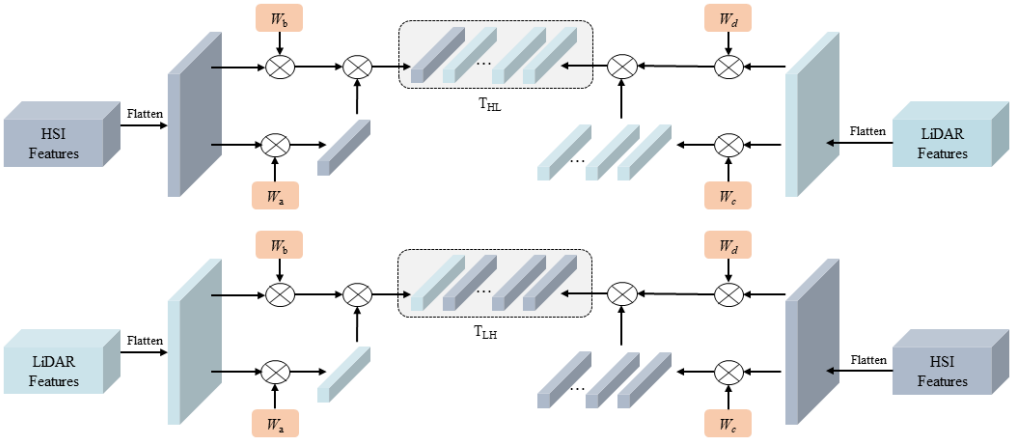


Figure 3. Cross-tokenization procedure for HSI and LiDAR features.

The unfolded HSI and LiDAR feature maps are represented, respectively, by $X_h^{DyConv} \in R^{u_h v_h \times z_h}$ and $X_l^{Dyconv} \in R^{u_l v_l \times z_l}$, where u_h and v_h are the height and width of the HSI feature map, u_l and v_l are the height and width of the LiDAR feature map, and z_h and z_l are the number of feature channels in the HSI and LiDAR feature maps, respectively. The HSI cross-feature tokens are represented by $T_{LH} \in R^{w_h \times z_h}$, where $w_h = n_h + 1$, and n_h is the number of HSI tokens. The LiDAR cross-feature tokens are represented by $T_{HL} \in R^{w_l \times z}$, where $w_l = n_l + 1$, n_l is the number of LiDAR tokens. Four trainable weights are employed in the cross-tokenization process. From Figure 3, W_a and W_b operate on secondary input features to extract complementary attributes, while W_c and W_d operate on primary input features to extract prominent characteristics. The following equation summarizes the process:

$$X_l^{Dyconv} = Flatten(X_L^{Dyconv}), \tag{4}$$

$$X_h^{Dyconv} = Flatten(X_H^{Dyconv}) \tag{5}$$

$$cls_L = \text{softmax}(X_l^{Dyconv} W_a)'(X_l^{Dyconv} W_b), \tag{6}$$

$$T_H = \text{softmax}(X_h^{DyConv} W_c)'(X_h^{DyConv} W_d), \tag{7}$$

$$T_{LH} = cls_L \cup T_H, \tag{8}$$

where $(\cdot)'$ denotes the transpose operation, $X_l^{DyConv} \in R^{49 \times 64}$ and $X_h^{Dyconv} \in R^{49 \times 64}$, $W_a \in R^{64 \times 1}$ and $W_b \in R^{1 \times n}$, $cls_L \in R^{1 \times 64}$,$W_c \in R^{64 \times n}$ and $W_d \in R^{64 \times n}$, and $T_H \in R^{n \times 64}$ and $T_{LH} \in R^{(n+1) \times 64}$.

On the LiDAR branch, the operations are similar to those on the HSI branch. The entire process can be summarized as follows:

$$cls_H = \text{softmax}(X_h^{Dyconv} W_a)'(X_h^{Dyconv} W_b), \tag{9}$$

$$T_L = \text{softmax}(X_l^{DyConv} W_c)'(X_l^{DyConv} W_d), \tag{10}$$

$$T_{HL} = cls_H \cup T_L. \tag{11}$$

To retain the semantic context information of the image cubes within the image blocks, trainable positional embeddings are incorporated into the image block embeddings. Positional embeddings are added to all tokens element-wise. Subsequently, a dropout layer is applied to mitigate network overfitting. The described process can be represented by the following equation:

$$T_{HL}^{p} = Dro(T_{HL} \oplus pos), \tag{12}$$

$$T_{LH}^{p} = Dro(T_{LH} \oplus pos), \tag{13}$$

where *Dro* is the dropout layer, and *pos* is the learnable position embedding.

3.2.2. Transformer Encoder with Cross-Token Attention

The integration of features at the level of features is crucial for effectively capturing the multi-source feature representation of RS data. To overcome the semantic disparity of multi-source data and achieve full fusion between the data, the multi-source attention in the existing transformer takes the whole tokens sequence of single-source data as Q (or K, V), which is computationally intensive. While classical self-attention uses a single data source, cross-token attention (CTA) combines inputs from multiple sources, integrates information more comprehensively, and performs better in multi-source remote sensing data fusion. CTA takes the CLS obtained by the CT method as the unique Q and takes the mixed tokens containing information from multiple data sources, such as K and V. This method reduces the computational effort while achieving full interaction between multi-source data. The entire process occurs within transformer encoder blocks, each comprising both multi-head cross-token attention (MHCTA) residual blocks and multi-layer perceptron (MLP) residual blocks. The cross-token attention residual block begins with layer normalization (LN) operations, followed by the CTA layer, and then the output is added to the input of LN element-wise.

As shown in Figure 4, the transformer image illustration with MHCTA with LiDAR features abstracted as CLS token is demonstrated, where cls_L is used as the unique q and T_{LH}^{p} is used as k and v. To understand the relationships among feature tokens, three learnable weights are predefined, including W_q, W_k and W_v. By multiplying k, q, and v with the three learnable weights, three different matrices are obtained. Namely, the query Q, key K, and value V. Attention scores are obtained by multiplying Q with K and scaling by Scale$(\frac{1}{\sqrt{h_d}})$. The softmax function is utilized to transform the scores into weight probabilities. In summary, the CTA can be represented by the following equation:

$$Q = cls_L W_q, K = T_{LH}^{p} W_k, V = T_{LH}^{p} W_v, \tag{14}$$

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V, \tag{15}$$

$$\mathrm{CTA_i} = Attention(Q_i, K_i, V_i), \tag{16}$$

where cls_L represents the classification label of LiDAR data, T_{LH}^{p} represents the fusion label of the two types of RS data, and d_k represents the dimension of K.

When assigning features to W_q, W_k, and W_v, the MHCTA module employs various sets of weights and executes identical computational procedures to obtain MHCTA scores. Subsequently, the resulting attention scores are then concatenated as follows:

$$\mathrm{MHCTA}(Q, K, V) = Concat(CTA_1, \ldots, CTA_h)W_o, \tag{17}$$

where h represents the number of attention heads, W represents the parameter matrix, $W_o \in R^{h \cdot d_k \times d_w}$, and $d_w = n$ represents the number of tokens.

Next, the MHCTA scores are passed through a dropout layer to yield Y_{LH}^{p}. Following this, it undergoes a residual connection and an LN operation, completing the MHCTA

block. Finally, the feature Y_{LH} is obtained by an MLP residual block. The above steps can be expressed as follows:

$$Y_{LH}^{p} = Dro(LN(\text{MHCTA})), \tag{18}$$

$$Y_{LH}^{R} = \text{MHCTA} \oplus Y_{LH}^{p}, \tag{19}$$

$$Y_{LH} = Y_{LH}^{R} \oplus \text{LN}(\text{MLP}(y_{LH}^{R})), \tag{20}$$

where the MLP consists of linear layers, activation functions, and dropout layers, enhancing the nonlinear representation of the model.

In another branch, HSI features are treated as prominent features, while LiDAR features serve as supplementary features. All other operations follow similar steps as described above. At this stage, we obtain fine-grained feature information from two integrated multi-source RS datasets, namely Y_{LH} and Y_{HL}.

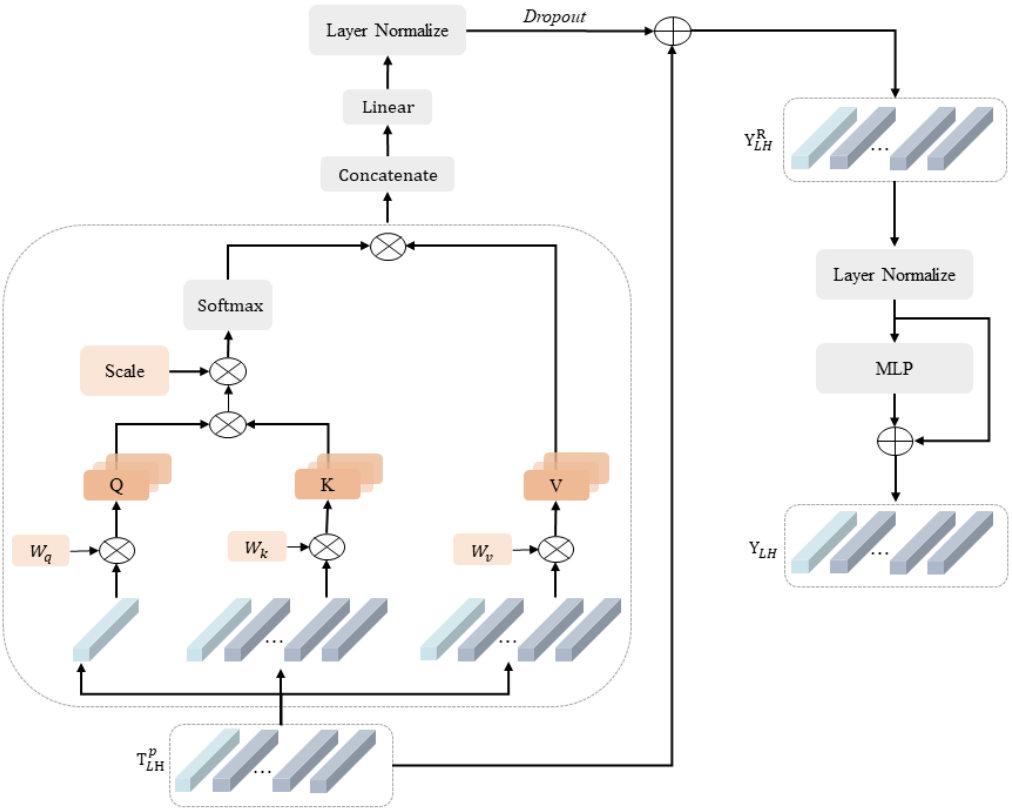


Figure 4. Illustration of a transformer diagram with multi-head cross-token attention.

3.3. FFL Module

To achieve superior classification performance by better balancing the coarse-grained and fine-grained information as well as the local and global information, we design the CFM module. This module mainly consists of two strategies: adaptive coarse–fine decision fusion and adaptive loss fusion.

3.3.1. Adaptive Coarse–Fine Decision Fusion (ADF)

Existing methods often adopt feature-level fusion strategies, ignoring decision-level fusion. Decision-level fusion is a process rooted in decision rules, which integrates multiple decisions to be more precise and effective than single decisions [43]. Meanwhile, to balance the contribution of different decisions and improve the fault tolerance of the decision function, we design the ADF strategy. Coarse-grained features and fine-grained features have strong complementary relationships in representing information of different categories. To

obtain reliable decision results, we fuse the two different granularity decision information to bolster the robustness of the model. The outputs X_h^{Dyconv} and X_l^{Dyconv} from the CFL are passed to the classification block to obtain coarse classification results. During this process, X_h^{Dyconv} and X_l^{Dyconv} are first subjected to average pooling operations to reduce dimensionality and extract key features. Afterward, the extracted characteristics are fed into a linear layer, with its output size adjusted to correspond to the total count of classes. Finally, the two output probability vectors are combined to derive coarse classification results. On the other hand, the two types of CLS tokens from the FFL outputs are separately sent into the classification block to obtain fine classification results. Similar to the coarse classification stage, these tokens also traverse through a linear layer with an output size matching the total number of classes. Then, the two output probability vectors from the linear layer are added together to obtain fine classification results. Finally, an adaptive parameter α is introduced to adaptively fuse coarse predictions with fine predictions. The resulting fused prediction probability vector $\widehat{y}$ is represented as follows:

$$\widehat{y} = \alpha \cdot \widehat{y}_c + (1 - \alpha) \cdot \widehat{y}_f, \tag{21}$$

where $\widehat{y}_c$ denotes the coarse prediction probability vector, $\widehat{y}_f$ denotes the fine prediction probability vector, and α and $1 - \alpha$ denote the different weights for coarse and fine prediction. This parameter is adjusted by the loss function during the learning phase. For a comprehensive understanding of the training process, please refer to Section 5.1, which is described and analyzed in detail.

3.3.2. Adaptive Loss Fusion (ALF)

Although the CNN-transformer structure can effectively utilize the local information extracted by CNN and the global information learned by the transformer, the differences in feature representations between them may lead to an imbalance between local and global information, thereby affecting the performance of the model. To address this imbalance and the potential decrease in network classification performance, we propose the ALF strategy. This strategy aims to guide the feature learning process by appropriately fusing the local information extracted by the CNN with the global information. Specifically, during the training phase, we use the cross-entropy loss function as the foundation and introduce an adaptive parameter β to dynamically adjust the balance between local information and global information. For the ith training sample, our loss function can be represented by the following mathematical formula:

$$L_c(y^i, \widehat{y}_c^i) = -\sum^{n} y^i \log \widehat{y}_c^i, \tag{22}$$

$$L_s(y^i, \widehat{y}^i) = -\sum_{i=1}^{n} y^i \log \widehat{y}^i, \tag{23}$$

$$L_{total} = \beta \cdot L_c + (1 - \beta) \cdot L_s, \tag{24}$$

where y^i is the real feature class that the pixel belongs to, y_c^i denotes the output of the coarse prediction, $\widehat{y}^i$ signifies the output of the final prediction, and L_{total} is the final fusion loss function. Similar to α, β is adopted to dynamically tune the formulation of the ultimate fusion loss function, as described in Section 5.1.

Once the proposed HCFA network is well-trained, for any test pixel X^{test} and its corresponding prediction vector $\widehat{y}^{test} \in R^{1 \times C}$, the label of the pixel X^{test} can be determined based on the maximum probability as follows:

$$Class(X^{test}) = argmax(\widehat{y}^{test}), \tag{25}$$

where $Class(X^{test}) \in [1, \ldots, C]$, and C denotes the number of land cover classes. In this manner, all pixels follow identical rules, resulting in the creation of a complete classification map.

4. Experiment

4.1. Dataset Description

In this section, extensive experiments are conducted on three widely used HSI and LiDAR datasets. These three datasets are described below.

(1) The Houston2013 dataset was captured by the ITRES CASI-1500 imaging sensor, manufactured by ITRES Research Limited, located in Calgary, AB, Canada. The data was collected in the vicinity of the University of Houston campus and its surrounding rural areas in Houston, TX, USA. This dataset, publicly available for the IEEE GRSS DFC 2013 (IEEE GRSS DFC 2013: https://technical-community-spotlight.ieee.org/ieee-grss-announces-plans-for-2013-data-fusion-contest/, accessed on 1 September 2024) , consists of two primary data sources. The hyperspectral data comprises 144 spectral bands, spanning from 0.38 to 1.05 µm in spectral resolution, while the LiDAR data consists of a single band. Both hyperspectral and LiDAR data cover an area of 349 × 1905 pixels, with a spatial resolution of 2.5 m, resulting in a total of 15 distinct classes [44]. Further information is detailed in Table 1 and Figure 5.

(2) The Trento dataset contains hyperspectral and LiDAR data, with hyperspectral imagery acquired through the AISA Eagle sensor, manufactured by Specim, Spectral Imaging Ltd, located in Oulu, Finland, and LiDAR DSM data captured by the Optech ALTM 3100 EA sensor, manufactured by Teledyne Optech, located in Vaughan, ON, Canada. This dataset is derived from a larger dataset gathered in a rural region located south of the city of Trento, Italy. It encompasses six unique categories, with an image dimension of 600 × 166 pixels and a spatial resolution of 1.0 m [45]. The HSI data contains 63 bands with spectral wavelengths ranging from 420 to 990 nm. Detailed information is provided in Table 1 and Figure 6.

(3) The MUUFL dataset, acquired in November 2010, was captured near the University of South Mississippi Gulf Park Campus using the Reflective Optics System Imaging Spectrometer sensor, manufactured by NASA's Goddard Space Flight Center, located in Greenbelt, MD, USA. This dataset contains hyperspectral data with 72 spectral bands ranging from 0.38 to 1.05 µm, and LiDAR data consists of two gratings with a wavelength of 1.06 µm. To mitigate excessive noise, we remove the first and last eight spectral bands, leaving 64 spectral bands for the updated HSI data. The dataset consists of 325 × 220 pixels with a spatial resolution of 1.0 m, covering 11 distinct land cover classes [46,47]. Detailed information can be found in Table 1 and Figure 7.

Table 1. Sizes of training, validation, and test samples for the Houston2013, Trento, and MUUFL datasets.

NO.	Houston2013 Dataset				Trento Dataset				MUUFL Dataset			
	Class Name	Training	Validation	Test	Class Name	Training	Validation	Test	Class Name	Training	Validation	Test
C01	Healthy Grass	13	13	1225	Apple Trees	41	41	3952	Trees	233	233	22,780
C02	Stressed Grass	13	13	1228	Buildings	30	30	2843	Mostly Grass	43	43	4184
C03	Synthetic Grass	7	7	683	Ground	5	5	469	Mixed Ground Surface	69	69	6744
C04	Trees	13	13	1218	Woods	92	92	8939	Dirt and Sand	19	19	1768
C05	Soil	13	13	1216	Vineyard	106	106	10,289	Road	67	67	6753
C06	Water	4	4	317	Roads	32	32	3110	Water	5	5	456
C07	Residential	13	13	1242					Buildings Shadow	23	23	2187
C08	Commercial	13	13	1218					Buildings	63	63	6114
C09	Road	13	13	1226					Sidewalk	14	14	1357
C10	Highway	13	13	1201					Yellow Curb	2	2	179
C11	Railway	13	13	1209					Cloth Panels	3	3	263
C12	Parking Lot1	13	13	1207								
C13	Parking Lot2	5	5	459								
C14	Tennis Court	5	5	418								
C15	Running Track	7	7	646								
-	Total	158	158	14,713	Total	306	306	29,602	Total	541	541	52,785

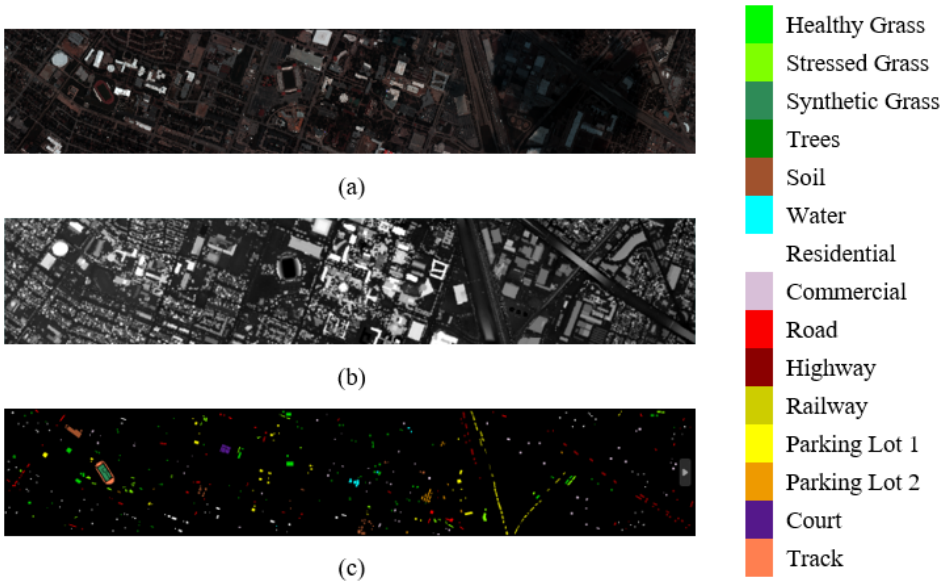


Figure 5. Houston2013 dataset: (**a**) pseudo-color composite image (64, 30, and 20 bands); (**b**) LiDAR-derived DSM image; (**c**) ground-truth map.

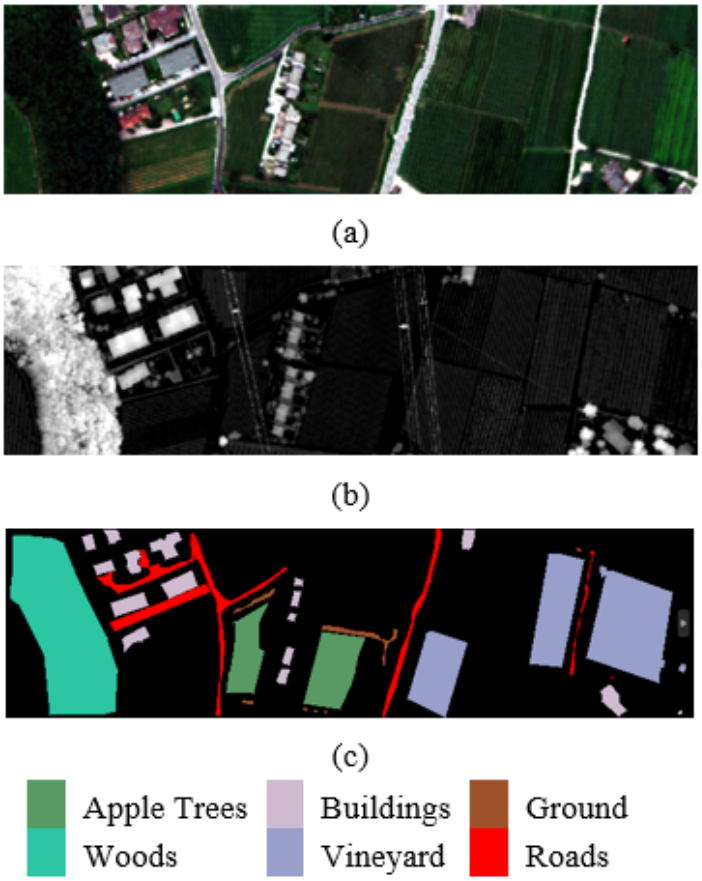


Figure 6. Trento dataset: (**a**) pseudo-color composite image (64, 30, and 20 bands); (**b**) LiDAR-derived DSM image; (**c**) ground-truth map.

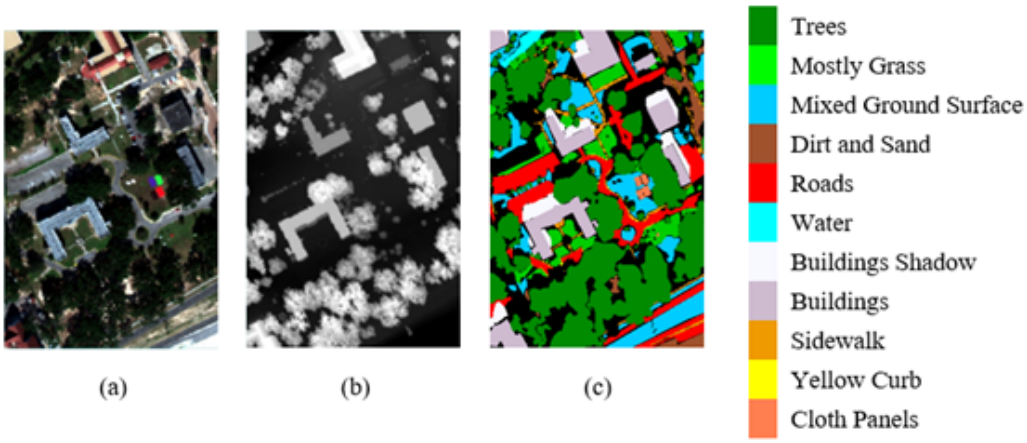


Figure 7. MUUFL dataset: (**a**) pseudo-color composite image (30, 15, and 7 bands); (**b**) LiDAR-derived DSM image; (**c**) ground-truth map.

4.2. Experiment Setting

To assess the classification performance of the proposed framework and other existing models, we calculate and analyze several metrics, including the overall accuracy (OA), average accuracy (AA), Kappa coefficient (Kappa), and per-class accuracy. The experiments involving the proposed method and other deep learning techniques are conducted using the PyTorch platform. The computations were performed on a laptop equipped with an AMD Ryzen7-6800H with Radeon Graphics 3.20 GHz CPU, 16 GB RAM, and an NVIDIA GeForce RTX 3060 Laptop GPU with 6 GB of VRAM. To optimize the network, we initially selected the Adam optimizer. During the training phase, we set the mini-batch size to 64 and the number of training epochs to 100.

4.3. Experimental Results and Analysis

To validate the efficacy of the proposed framework, a series of experiments will contrast it with various representative classification methods. These methods include SVM [48], CoupledCNN [22], the recently proposed OSDN [49], SSFTT [28], AM^{3}Net [50], HCT [51], and S^{2}Enet [52]. Among them, SVM, OSDN, and SSFTT represent machine learning methods, convolutional neural networks, and transformers, respectively, representing hyperspectral classification methods. Coupled CNNs represent early multi-source RS classification methods, while AM^{3}Net, HCT, and S^{2}Enet represent state-of-the-art fusion networks. All compared methods utilize the default parameters provided in the original files. Each experiment is independently replicated 10 times, and the average is documented as the conclusive outcome. As demonstrated in Table 1, the training sample set, validation sample set, and test sample set are randomly chosen for all methods to ensure an impartial comparison. Tables 2–4 correspondingly display the OA, AA, Kappa, and per-class accuracy of all these methods across the Houston2013, Trento, and MUUFL datasets. Bold font is used to denote superior outcomes within each table.

We first compare the performance of various methods on the Houston2013 dataset. The classification results are shown in Table 2, and the full-pixel classification map is presented in Figure 8. The best values for OA, AA, Kappa, and per-class accuracy are highlighted in bold in the table. Overall, the proposed HCFA network achieves excellent classification results on all four evaluation metrics in most cases compared to other methods. The methods based on deep learning obtain significantly higher OA than SVM (79.61%). Specifically, C08 (56.86%), C12 (62.51%), and C13 (22.24%) are particularly challenging for SVM classification. However, other deep learning methods achieve over 70% accuracy in both categories. This indicates that the introduction of spatial features and higher-level features contributes to improving the recognition rate of land cover. Our proposed HCFA network achieves the highest accuracy among all comparison algorithms for classes C12 and C13. This is attributed to the coarse–fine fusion classification method, which gives the

network access to more comprehensive information. OSDN provides a higher OA (87.19%) compared to SVM, attributed to its utilization of two polarized attentions that fully exploit spatial–spectral features. Observing the classification accuracy of various methods, it can be seen that overall, methods such as CoupledCNN, AM^{3}Net, HCT, S^{2}Enet, and HCFA, which adopt HSI-LiDAR input, achieve higher accuracy compared to classification methods in which the HSI is only inputted. This indicates that the introduction of LiDAR information effectively enhances classification accuracy. In comparative fusion classification methods, the proposed HCFA achieves the highest OA, with over half of per-class accuracies reaching their peaks. This is because the proposed feature fusion method effectively integrates heterogeneous information between HSI and LiDAR, obtaining more discriminative features. From Figure 8, the classification maps generated by the proposed HCFA are smoother and exhibit fewer misclassifications compared to other classification maps. Since SVM only utilizes spectral features of each pixel for classification, the classification result map is filled with a considerable amount of salt and pepper noise. Other methods utilize spatial information, resulting in less salt and pepper noise in the generated classification maps. However, some methods still suffer from significant misclassification. For instance, AM^{3}Net loses many spatial and elevation characteristics. Conversely, due to the adoption of carefully designed multi-source features, decision fusion, and adaptive strategies, most land cover classifications in HCFA classification maps are more consistent than those in AM^{3}Net.

Table 2. Performance of various methods in classifying the Houston2013 dataset (Optimal results are in bold).

NO.	HSI Input			HSI and LiDAR Input				
	SVM	**OSDN**	**SSFTT**	**CoupledCNN**	**AM^{3}Net**	**HCT**	**S^{2}ENet**	**HCFA**
C01	92.67 ± 05.05	85.08 ± 01.06	83.68 ± 01.87	89.54 ± 02.62	77.50 ± 04.90	89.00 ± 01.13	89.07 ± 00.12	**94.50 ± 00.08**
C02	90.98 ± 04.45	83.54 ± 00.70	83.90 ± 03.58	83.92 ± 05.61	85.54 ± 05.40	**96.83 ± 00.72**	93.10 ± 00.69	87.02 ± 00.10
C03	98.99 ± 00.69	**100.00 ± 00.00**	99.76 ± 00.12	**100.00 ± 00.00**	98.52 ± 01.51	**100.00 ± 00.00**	99.97 ± 00.06	**100.00 ± 00.00**
C04	93.24 ± 03.40	88.23 ± 00.20	90.60 ± 01.07	94.55 ± 00.80	91.98 ± 03.14	**97.05 ± 00.04**	90.13 ± 00.32	94.54 ± 00.00
C05	97.45 ± 01.81	86.94 ± 00.04	93.45 ± 01.04	94.42 ± 01.11	97.98 ± 00.80	94.76 ± 01.58	83.52 ± 00.30	**99.24 ± 00.20**
C06	83.79 ± 05.72	90.24 ± 00.12	97.16 ± 01.36	98.19 ± 00.67	94.46 ± 03.09	**100.00 ± 00.00**	**100.00 ± 00.00**	99.23 ± 00.00
C07	72.99 ± 05.56	82.90 ± 00.63	84.22 ± 00.17	87.74 ± 02.06	87.85 ± 02.23	**92.82 ± 01.20**	84.78 ± 00.08	89.25 ± 00.05
C08	56.86 ± 07.93	**96.79 ± 00.20**	88.23 ± 01.06	89.37 ± 02.88	81.39 ± 00.99	78.14 ± 00.38	85.21 ± 00.52	86.29 ± 00.01
C09	79.18 ± 07.77	73.35 ± 00.12	74.25 ± 00.20	74.49 ± 03.64	72.70 ± 02.71	71.95 ± 01.56	**87.26 ± 00.41**	81.63 ± 00.24
C10	76.13 ± 13.48	90.08 ± 00.35	85.42 ± 01.57	85.45 ± 03.03	86.24 ± 05.71	83.85 ± 01.18	90.47 ± 00.04	**95.13 ± 00.27**
C11	69.74 ± 12.57	**93.80 ± 00.67**	91.78 ± 00.21	89.24 ± 02.99	93.33 ± 01.62	86.40 ± 02.22	91.87 ± 00.45	88.84 ± 00.28
C12	62.51 ± 08.86	81.52 ± 00.58	83.86 ± 00.52	87.51 ± 06.64	87.12 ± 02.01	79.16 ± 01.25	83.67 ± 00.68	**91.80 ± 00.52**
C13	22.24 ± 04.36	93.89 ± 00.79	93.06 ± 00.07	96.19 ± 00.89	91.44 ± 02.49	89.65 ± 00.82	93.48 ± 00.80	**97.08 ± 00.54**
C14	89.56 ± 12.59	88.56 ± 00.00	97.88 ± 00.47	99.50 ± 00.76	83.20 ± 14.44	**100.00 ± 00.00**	**100.00 ± 00.00**	**100.00 ± 00.00**
C15	99.11 ± 00.41	98.48 ± 00.00	91.75 ± 01.28	92.00 ± 02.14	95.36 ± 06.77	97.29 ± 00.29	95.06 ± 00.34	**99.35 ± 00.06**
OA(%)	79.61 ± 01.30	87.19 ± 0.03	87.50 ± 00.67	88.79 ± 00.80	86.69 ± 01.00	88.47 ± 00.27	89.46 ± 00.09	**91.92 ± 00.04**
AA(%)	79.03 ± 01.68	88.89 ± 0.03	89.26 ± 00.51	90.81 ± 00.81	88.31 ± 00.82	90.46 ± 00.24	91.17 ± 00.02	**93.59 ± 00.02**
Kappa × 100	77.93 ± 01.42	86.16 ± 00.04	86.48 ± 00.72	87.88 ± 00.87	85.61 ± 01.08	87.53 ± 00.30	88.60 ± 00.10	**91.26 ± 00.04**

Table 3. Performance of various methods in classifying the Trento dataset (Optimal results are in bold).

NO.	HSI Input			HSI and LiDAR Input				
	SVM	**OSDN**	**SSFTT**	**CoupledCNN**	**AM^{3}Net**	**HCT**	**S^{2}ENet**	**HCFA**
C01	73.70 ± 07.76	98.52 ± 00.44	97.81 ± 00.78	96.28 ± 01.65	96.34 ± 00.73	97.58 ± 01.53	**99.13 ± 00.46**	98.95 ± 00.02
C02	70.27 ± 11.17	86.09 ± 08.75	82.80 ± 02.65	95.10 ± 00.44	94.50 ± 00.63	94.79 ± 00.73	95.78 ± 00.36	**96.59 ± 00.03**
C03	82.81 ± 08.77	**100.00 ± 00.00**	99.90 ± 00.19	99.59 ± 00.70	95.80 ± 02.64	93.09 ± 13.18	**100.00 ± 00.00**	99.77 ± 00.00
C04	95.11 ± 00.77	99.95 ± 00.03	99.77 ± 00.15	99.98 ± 00.05	99.83 ± 00.11	99.97 ± 00.02	99.91 ± 00.05	**99.98 ± 00.00**
C05	81.79 ± 03.32	99.6 ± 00.19	99.28 ± 00.13	99.61 ± 00.13	98.90 ± 00.46	99.27 ± 00.30	99.65 ± 00.11	**99.80 ± 00.00**
C06	70.26 ± 10.45	86.42 ± 11.73	86.37 ± 01.62	97.98 ± 00.49	**98.43 ± 00.60**	94.57 ± 01.72	97.26 ± 00.53	97.80 ± 00.04
OA (%)	82.43 ± 00.68	96.28 ± 01.37	96.26 ± 00.36	98.65 ± 00.19	98.30 ± 00.21	98.17 ± 00.36	99.04 ± 00.11	**99.22 ± 00.01**
AA (%)	78.99 ± 01.31	95.10 ± 00.99	94.33 ± 00.52	98.09 ± 00.28	97.30 ± 00.45	96.54 ± 02.02	98.62 ± 00.14	**98.81 ± 00.01**
Kappa × 100	76.64 ± 00.88	95.03 ± 01.83	95.01 ± 00.48	98.19 ± 00.25	97.73 ± 00.28	97.56 ± 00.48	98.71 ± 00.15	**98.96 ± 00.01**

Table 4. Performance of various methods in classifying the MUUFL dataset (Optimal results are in bold).

NO.	HSI Input			HSI and LiDAR Input				
	SVM	OSDN	SSFTT	CoupledCNN	AM^{3}Net	HCT	S^{2}ENet	HCFA
C01	92.86 ± 01.01	93.48 ± 01.01	89.59 ± 00.03	94.54 ± 00.54	94.76 ± 00.50	94.10 ± 00.39	**94.83 ± 00.85**	94.44 ± 00.02
C02	68.68 ± 03.56	79.39 ± 01.09	74.57 ± 00.10	76.64 ± 05.86	73.32 ± 02.45	81.50 ± 02.41	80.09 ± 02.35	**82.99 ± 00.10**
C03	73.62 ± 02.95	82.42 ± 01.29	73.59 ± 00.05	79.01 ± 03.66	75.58 ± 02.33	77.96 ± 02.07	81.81 ± 02.28	**83.21 ± 00.06**
C04	67.35 ± 04.58	83.89 ± 03.01	76.90 ± 00.10	80.91 ± 02.65	82.93 ± 03.62	79.04 ± 03.75	**92.57 ± 04.83**	85.61 ± 00.00
C05	81.61 ± 00.63	84.61 ± 02.19	78.49 ± 00.01	77.67 ± 04.39	89.13 ± 00.90	86.54 ± 00.75	88.75 ± 00.96	**89.21 ± 00.01**
C06	**99.00 ± 00.83**	95.96 ± 01.70	93.31 ± 00.14	91.75 ± 05.03	79.10 ± 04.29	89.05 ± 02.74	90.62 ± 07.47	98.47 ± 00.10
C07	65.17 ± 04.61	72.74 ± 02.87	68.75 ± 00.20	79.73 ± 03.85	65.46 ± 02.23	72.70 ± 03.86	**82.03 ± 03.75**	73.77 ± 00.01
C08	89.74 ± 02.32	96.71 ± 00.27	89.17 ± 00.06	96.43 ± 00.90	97.15 ± 00.31	**97.66 ± 00.28**	96.74 ± 00.24	97.13 ± 00.01
C09	53.01 ± 12.31	52.11 ± 11.85	**71.91 ± 00.11**	62.99 ± 03.78	67.11 ± 03.32	65.80 ± 05.95	52.26 ± 00.72	59.01 ± 00.18
C10	52.98 ± 13.07	13.32 ± 04.38	33.33 ± 00.00	05.71 ± 11.43	05.80 ± 00.84	07.94 ± 02.64	15.95 ± 02.45	**100.00 ± 00.00**
C11	94.14 ± 05.00	77.36 ± 04.05	91.69 ± 00.53	69.47 ± 07.77	**95.41 ± 07.36**	91.72 ± 01.72	91.72 ± 02.13	89.59 ± 00.14
OA (%)	83.27 ± 00.53	87.78 ± 00.66	83.50 ± 00.02	87.06 ± 00.54	87.02 ± 00.17	88.18 ± 00.43	89.52 ± 00.15	**89.92 ± 00.01**
AA (%)	76.20 ± 01.27	75.64 ± 00.93	76.48 ± 00.06	74.08 ± 01.52	75.07 ± 00.97	76.43 ± 01.18	78.85 ± 00.49	**86.68 ± 00.02**
Kappa × 100	77.84 ± 00.68	83.74 ± 00.89	77.83 ± 00.03	82.86 ± 00.72	82.77 ± 00.21	84.28 ± 00.54	86.07 ± 00.24	**86.63 ± 00.01**

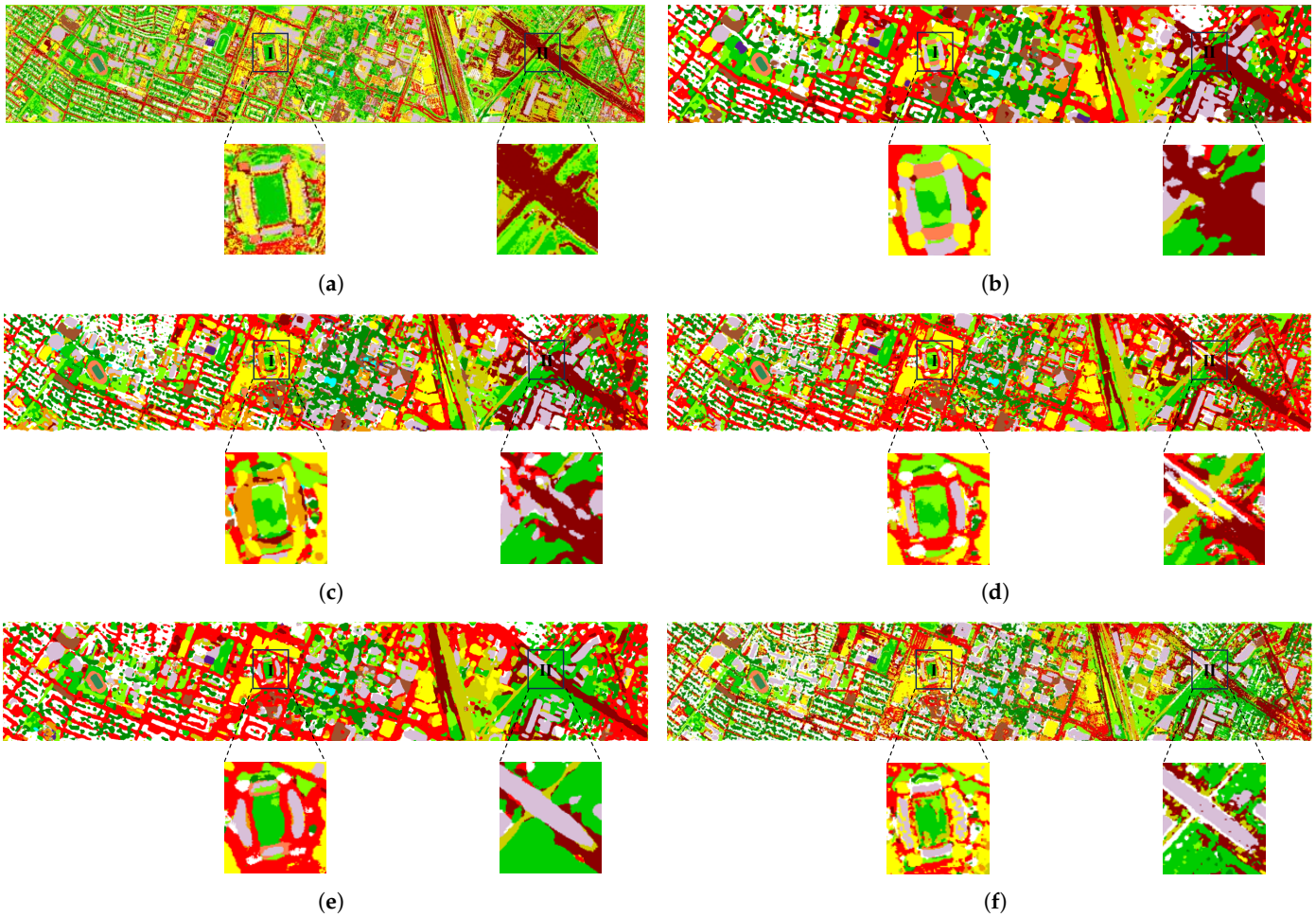
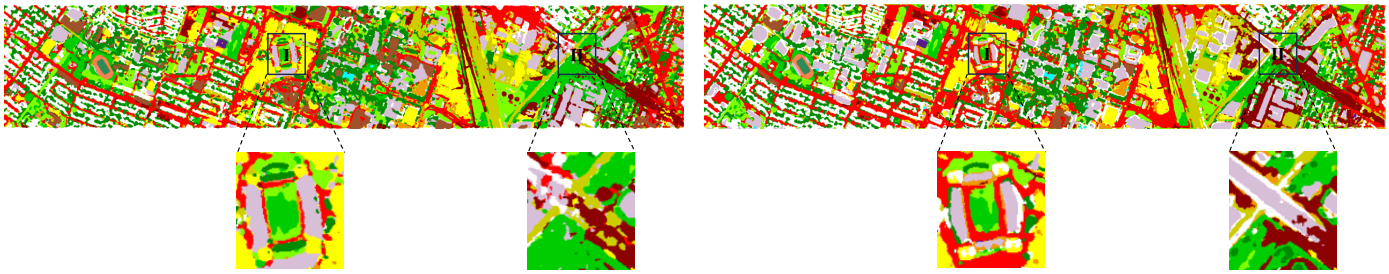


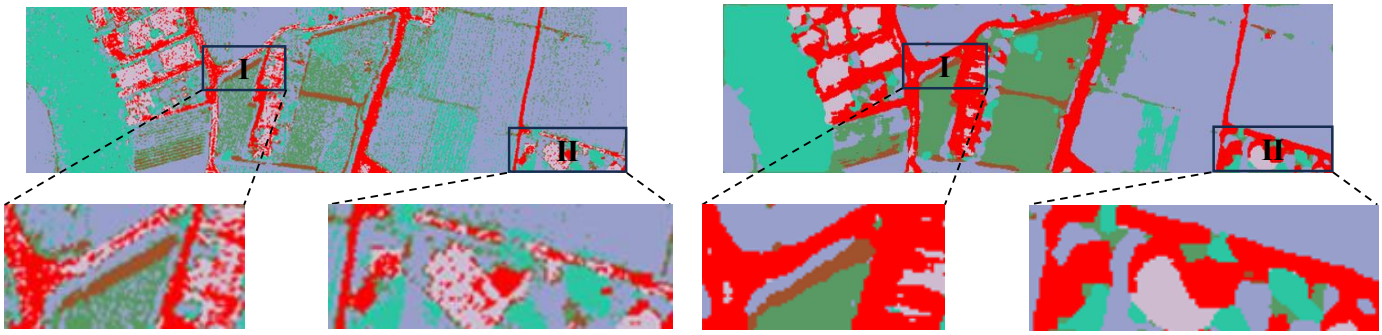
Figure 8. *Cont.*



(**g**) (**h**)

Figure 8. Full-pixel classification maps for the Houston2013 dataset. (**a**) SVM. (**b**) OSDN. (**c**) SSFTT. (**d**) CoupledCNN. (**e**) AM^{3}Net. (**f**) HCT. (**g**) S^{2}ENet. (**h**) HCFA.The regions marked as I and II indicate enlarged areas for clearer observation of differences.

In the experiments on the Trento dataset, there are enough training samples with fewer categories of training samples. The specific classification results and the full-pixel classification map are shown in Table 3 and Figure 9, respectively. Observing the classification accuracy of each method, SVM obtains the lowest OA (82.43%). The OAs of other single-source methods, OSDN and SSFTT, are 96.28% and 96.26%, respectively. Further observation of the per-class accuracies reveals that with the introduction of spatial information, the accuracies of C02 and C06 increased from 70% to 80%. With the subsequent inclusion of elevation information, the accuracies of these classes further rose from 80% to 90%. This indicates that the addition of LiDAR data is advantageous for the classification of these two classes. However, compared with the single HSI input OSDN and SSFTT methods, the accuracies of C01 and C03 decreased with the introduction of elevation information in Coupled CNN, AM^{3}Net, and HCT methods. This suggests that the introduction of elevation information may affect the utilization of spatial information for classes C01 and C03. According to Figure 9, the apple tree (C01) can be seen surrounded by the ground (C03). There are some misclassification cases for methods such as such as CoupledCNN, AM^{3}Net, and HCT. This indicates that only by fully fusing the features of two types of data can the advantages of both types of data be leveraged, thus steadily improving the recognition of land features. In this regard, the proposed HCFA method performs well, with OA, AA, Kappa, and per-class accuracy reaching decent levels. Observing Figure 9, it becomes apparent that the full-pixel classification map generated by HCFA appears cleaner and smoother. Particularly, the boundaries of the corresponding areas for C01 and C03 are remarkably distinct, further corroborating the classification performance of the proposed method.



(**a**) (**b**)

Figure 9. *Cont.*

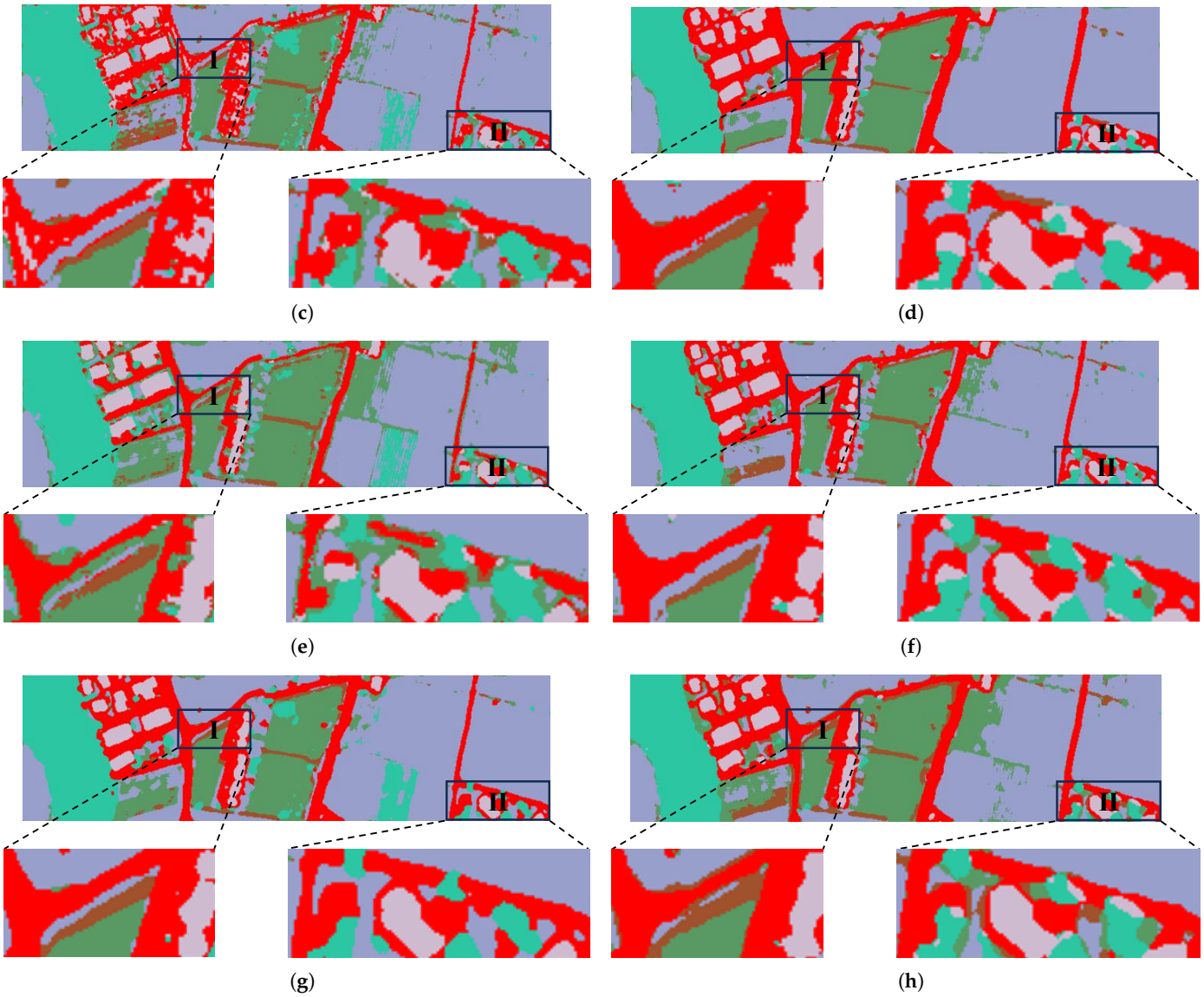


Figure 9. Full-pixel classification maps for the Trento dataset. (**a**) SVM. (**b**) OSDN. (**c**) SSFTT. (**d**) CoupledCNN. (**e**) AM^{3}Net. (**f**) HCT. (**g**) S^{2}ENet. (**h**) HCFA.The regions marked as I and II indicate enlarged areas for clearer observation of differences.

The MUUFL dataset contains complex terrain landscapes and exhibits an imbalance in the number of labeled samples. The classification results and the full-pixel classification map are shown in Table 4 and Figure 10. Observing the classification accuracy of each method, SVM obtains the lowest OA (83.27%), while the proposed HCFA method achieves the highest OA (89.92%), with a difference of 6.65% between them. Additionally, the OA of HCFA is only 0.4% lower than the second-best method. This is because the MUUFL dataset has a larger number of samples, allowing most methods to better capture the distribution characteristics of the data. Further observation of the accuracy of each class reveals that classes C03, C07, C09, and C10 are difficult to classify for most methods. However, for C03 and C10, the proposed HCFA method performs the best. This is because the coarse–fine decision fusion strategy integrates coarse-grained and fine-grained information and continuously optimizes weight representations. From Figure 10, it can be seen that compared to other methods, the full-pixel classification map generated by the HCFA method is cleaner and smoother. This consistency with the classification results indicates that HCFA can better capture the useful features of the HSI and LiDAR data.

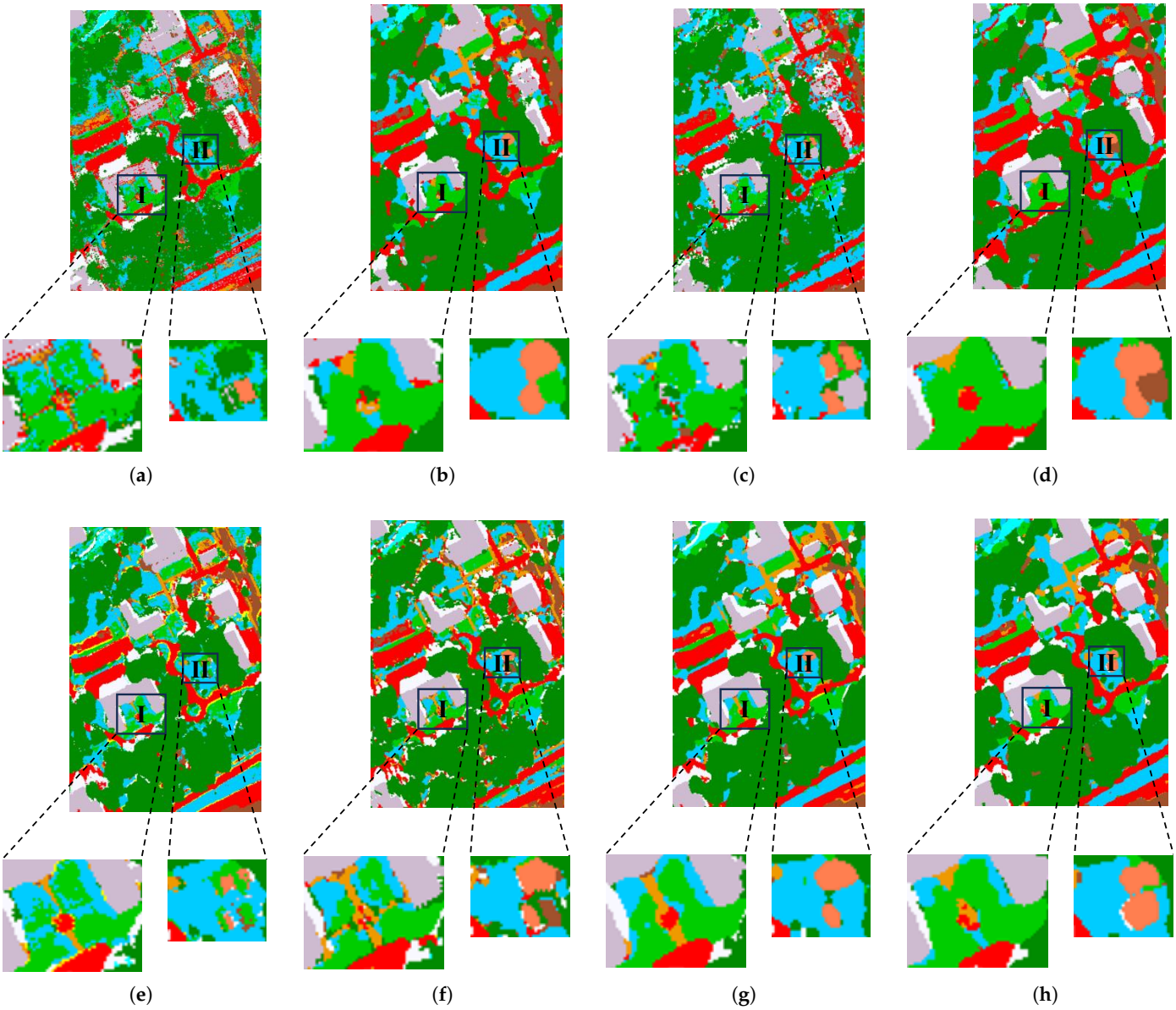


Figure 10. Full-pixel classification maps for the MUUFL dataset. (**a**) SVM. (**b**) OSDN. (**c**) SSFTT. (**d**) CoupledCNN. (**e**) AM^{3}Net. (**f**) HCT. (**g**) S^{2}ENet. (**h**) HCFA.The regions marked as I and II indicate enlarged areas for clearer observation of differences.

5. Discussion

5.1. Parameter Analysis

Various parameters that may influence the classification performance and training process are examined, encompassing the learning rate, the patch size of the input data, the tokens number, the encoder depth, the number of attention heads, the coarse–fine fusion parameter α, and the loss fusion parameter β. All parameter analyses are based on the method of controlling variables. Below, we will provide a detailed overview of the impact analysis of these parameters.

(1) Learning rate: The learning rate is a critical parameter in deep learning models, exerting significant influence on both the training process and the final classification results. With appropriate learning rate settings, model parameters are effectively guided towards optimal updates, aiding the target function to rapidly converge to local minima. To assess the effectiveness of the learning rate, we experiment with different learning rates selected from the candidate set {0.01, 0.005, 0.001, 0.0005,

0.0001, and 0.00005}. Figure 11a illustrates the OA obtained by our proposed method on three datasets with different learning rate settings. The results reveal that the optimal learning rates for the Houston2013, Trento, and MUUFL datasets are 0.001, 0.001, and 0.0001, respectively. This result indicates that the optimal learning rate may vary for different datasets.

(2) Patch size: In the field of RS image classification, accurately determining the appropriate patch size is crucial for the effective capture of features. In the experiment, we select the most suitable patch size from the candidate set {5, 7, 9, 11, 13, and 15}. As shown in Figure 11b, we investigate the influence of different patch sizes on OA. When the patch size increases from 5×5 to 7×7, OA shows an increasing trend. At the patch size of 7, all three datasets reach the optimal OA value. However, OA shows a decreasing trend as the patch size increases further. This is because excessively large patch sizes may increase the number of pixels inconsistent with the label of the center pixel, making it difficult to extract small-scale terrain features, thereby reducing classification performance.

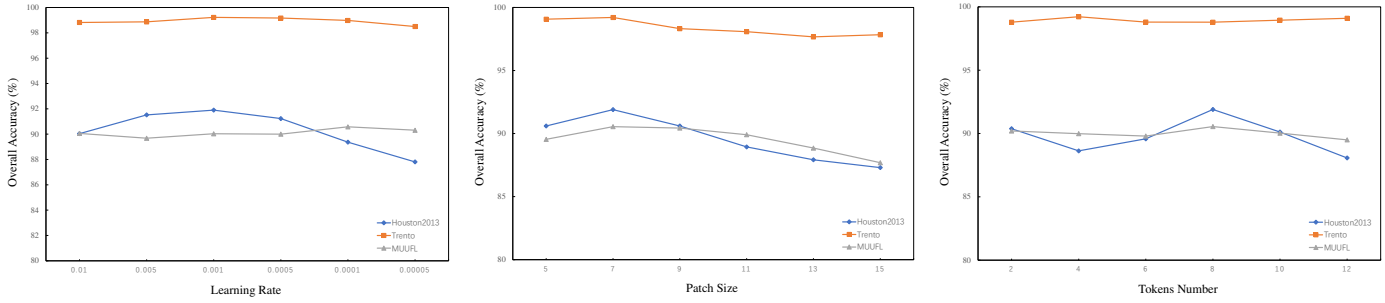
(3) Token number: Different datasets have different characteristics and data distributions, so the appropriate number of tokens needs to be determined. In Figure 11c, we give the influence of the number of tokens on OA on three datasets. The number of tokens is selected from the candidate set {2, 4, 6, 8, 10, and 12}. The results show that on the Houston2013 dataset and the MUUFL dataset, the optimal OA value is achieved when the number of tokens is 8. On the Trento dataset, the optimal OA value is achieved when the number of tokens is 4. From Figure 11c, it can be concluded that too many tokens may cause the model to overly focus on some irrelevant features and ignore the more important features for the classification task, leading to a decrease in classification performance.

(4) Depth of encoder: The depth of the encoder can impact the capability to represent features. To assess the impact of encoder depth on classification performance, we define the candidate set as {1, 2, 3, 4, and 5}. Figure 12a shows the various OAs with encoder depths across the three datasets. The proposed network demonstrates optimal classification performance with an encoder depth of 2. As the encoder depth increases, the performance of the network decreases on all three datasets. This phenomenon indicates that deeper networks do not necessarily lead to better performance.

(5) The number of attention heads: Increasing the number of attention heads improves the representational capacity of the model, but also increases the computational demand and the possibility of overfitting. Therefore, it is necessary to determine the number of appropriate attention heads. Based on Figure 12b, the results of testing different numbers of attention heads on three datasets are shown. The candidate set for the number of attention heads is {2, 4, 8, 16, and 32}. Experiments demonstrate that on the Houston2013 dataset and MUUFL dataset, the model achieves the optimal OA value when the number of attention heads is eight. Whereas on the Trento dataset, the optimal number of attention heads is four.

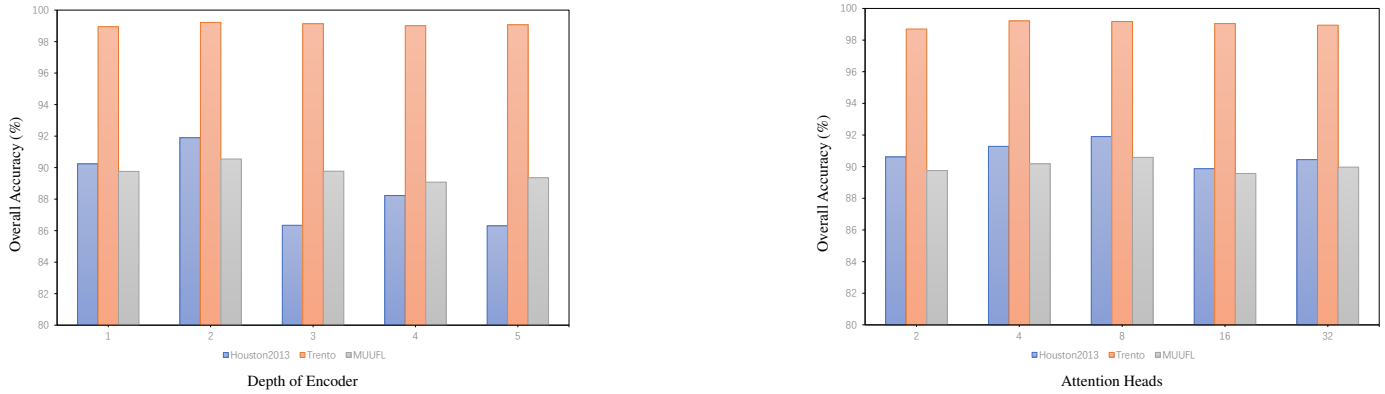
(6) Coarse–fine fusion parameters and loss fusion parameters: Figure 13 illustrates the impact of coarse-grained decisions and fine-grained decisions on three datasets. Here, train_acc represents training accuracy, val_acc represents validation accuracy, α represents the weight of the coarse-grained decision, and $1 - \alpha$ represents the weight of the fine-grained decision. On the Houston dataset shown in Figure 13a, it can be seen that when the number of epochs is nearly 50, the training accuracy gradually stabilizes. However, the validation accuracy fluctuates slightly and remains lower than the training accuracy when the number of epochs is greater than 50. It may be inferred that there is a slight overfitting phenomenon in the network. When the network stabilizes, it is observed that α is almost equal to $1 - \alpha$, suggesting that coarse-grained decisions and fine-grained decisions provide equivalent representational information on the Houston2013 dataset. On the Trento dataset shown in Figure 13b, near epoch 10, the training accuracy and validation accuracy tend to be very close. These two values

tend to stabilize with the increase in the iteration. This indicates that the network has successfully learned features from the Trento dataset. In this process, α is very close to $1-\alpha$. This indicates that coarse-grained fine-grained information plays important roles in land cover classification. On the MUUFL dataset shown in Figure 13c, it can be seen that the training accuracy and validation accuracy gradually stabilize as the difference between α and $1-\alpha$ slowly increases. This indicates that ADF dynamically adjusts the weights of coarse-grained and fine-grained information to achieve better classification performance. Figure 14 illustrates the influence of the composition of the loss function on network performance in three data sets. Here, β represents the local loss based on dynamic convolution, while $1-\beta$ represents the loss of the final decision of the network. The fusion aims to balance local and global information to train the optimal network performance. On the Houston dataset shown in Figure 14a, it can be observed that β is larger than $1-\beta$ when the epoch number is 10, while β is smaller than $1-\beta$ when the epoch number is 20. This indicates that the ALF module is dynamically adjusting the weights of the local and global information. In Figure 14b,c, for the Trento and MUUFL datasets, as the network accuracy stabilizes, the difference between β and $1-\beta$ gradually increases, and β is less than $1-\beta$. This suggests that the network gradually weakens the influence of local loss β. With the integration of the ALF module, the network achieves a suitable balance between local and global information across different datasets, thereby achieving higher classification performance.



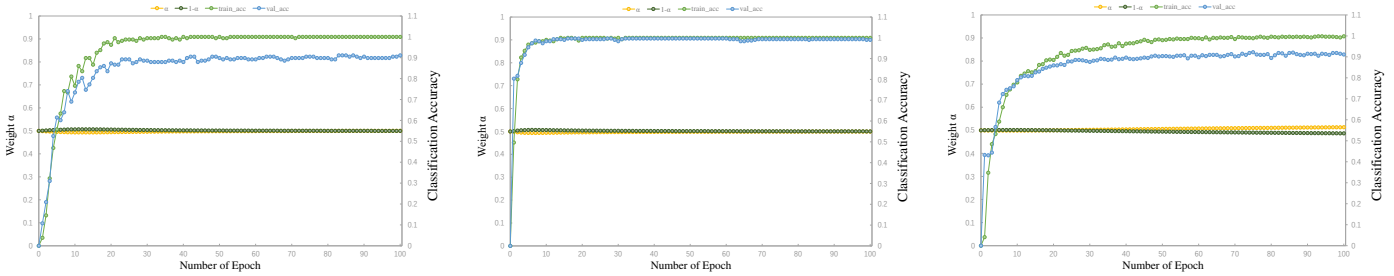
(**a**) (**b**) (**c**)

Figure 11. The impact of different parameters on overall accuracy across three datasets. (**a**) Learning rate. (**b**) Patch size. (**c**) Token number.



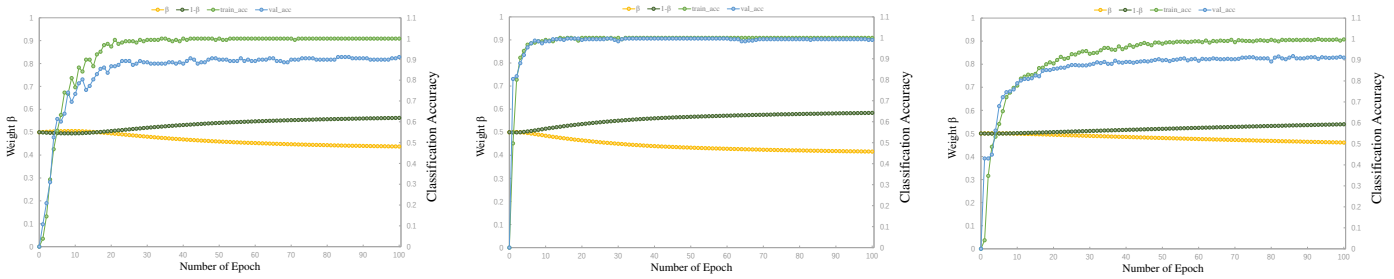
(**a**) (**b**)

Figure 12. The impact of different parameters on overall accuracy across three datasets. (**a**) Depth of encoder. (**b**) The number of attention heads.



(**a**) (**b**) (**c**)

Figure 13. Impact of weights α and $1-\alpha$ based on the proposed method in three datasets: (**a**) Houston2013, (**b**) Trento, and (**c**) MUUFL.



(**a**) (**b**) (**c**)

Figure 14. Impact of weights β and $1-\beta$ based on the proposed method in three datasets: (**a**) Houston2013, (**b**) Trento, and (**c**) MUUFL.

5.2. Ablation Analysis

In this section, we show the ablation analysis for different data sources and different components, respectively. Additionally, we visualize key ablation experimental results to intuitively demonstrate their impact.

5.2.1. Ablation Analysis of Inputs from Different Data Sources

Regarding the impact of heterogeneous data on model classification performance, three sets of experiments are conducted: using single HSI data input, single LiDAR data input, and combined HSI-LiDAR input. The experimental results are presented in Table 5. Observing the results, it is evident that omitting LiDAR data results in a noticeable decrease in OA, AA, and Kappa across all three datasets. This suggests that incorporating LiDAR data effectively enhances the accuracy of land cover classification. However, when only HSI data are removed, the proposed method shows significant decreases in classification performance on the Houston2013, Trento, and MUUFL datasets, with OA decreasing by 45.84%, 6.54%, and 26.69%, respectively. For the Trento dataset, the decrease in accuracy is not significant. For the Houston and MUUFL datasets, the decrease in accuracy is huge. This phenomenon suggests a greater reliance on spectral–spatial information for the classification of difficult datasets such as Houston and MUUFL. This confirms that both types of data have contributed positively to enhancing classification performance, thus affirming that the designed network effectively leverages complementary information from various RS data sources.

Table 5. Ablation analysis of inputs from different data sources (Optimal results are in bold).

Cases	Houston2013			Trento			MUUFL		
	OA (%)	AA (%)	Kappa × 100	OA (%)	AA (%)	Kappa × 100	OA (%)	AA (%)	Kappa × 100
Only HSI	89.11	91.19	88.23	97.03	95.49	96.04	88.98	77.82	85.39
Only LiDAR	43.27	41.18	39.03	92.68	89.00	90.15	63.23	34.79	49.90
HSI-LiDAR	**91.92**	**93.59**	**91.26**	**99.22**	**98.81**	**98.96**	**89.92**	**86.68**	**86.63**

5.2.2. Ablation Analysis of Different Components

Since the proposed network benefits from multiple components, we analyze the contribution of each component to the joint classification performance in the ablation experiments. The proposed framework comprises three primary parts: dynamic Conv 3D and dynamic Conv 2D for processing HSI and LiDAR data, respectively; the CT method for the feature fusion of two types of data; the CFM module, which consists of two components: ADF and ALF.

According to the results in Table 6, we demonstrate five variants of the proposed method on three datasets, which remove different combinations of components. We analyze the classification accuracy of these variants to evaluate the impact of different components on model performance. Across the three datasets, the first variant, which removes dynamic convolution, shows decreases in OA by 3.6%, 0.46%, and 0.73%, respectively, compared to the full component model. This confirms the importance of dynamic convolution in the feature extraction stage, which can enhance the feature extraction capability of the HCFA network in coarse feature learning. The second variant lacks the ALF component, meaning the absence of the CFM component. The decrease in OA in this variant validates the effectiveness of ALF in balancing local and global information for better classification performance. The third variant lacks the ADF component, meaning the absence of the CFM component. Compared to the full component, the classification performance of this variant significantly decreases, confirming that ADF effectively integrates coarse-grained and fine-grained information for comprehensive and accurate classification. The fourth variant lacks all components of the CFM module and has poor classification accuracy. This sufficiently proves the crucial role of the CFM module in reducing information loss during the feature abstraction process, thereby improving classification accuracy. The fifth variant shows a decrease in OA when CT is not used, indicating the effective role of CT in overcoming obstacles related to heterogeneous data structures and unrelated physical properties. This analysis further confirms the effectiveness of the proposed network framework and demonstrates the positive role of all its major components.

Table 6. Ablation analysis of different components in the proposed model on three datasets (Optimal results are in bold).

Cases	Component					Houston2013			Trento			MUUFL		
	Dynamic Conv	ALF	ADF	CFM	CT	OA (%)	AA (%)	Kappa × 100	OA (%)	AA (%)	Kappa × 100	OA (%)	AA (%)	Kappa × 100
1	Conv	✓	✓	✓	✓	88.32	90.6	87.37	98.76	98.24	98.35	89.19	77.68	85.64
2	✓	-	✓	-	✓	90.40	92.23	89.61	98.98	98.55	98.64	89.42	79.07	85.91
3	✓	✓	-	-	✓	90.14	91.94	89.34	99.08	98.67	98.77	89.43	77.83	85.95
4	✓	-	-	-	✓	88.77	90.95	87.86	98.67	98.18	98.22	89.12	78.34	85.50
5	✓	✓	✓	✓	-	88.89	91.09	87.99	98.81	98.21	98.41	89.21	77.45	85.67
6	✓	✓	✓	✓	✓	**91.92**	**93.59**	**91.26**	**99.22**	**98.81**	**98.96**	**89.92**	**86.68**	**86.63**

5.2.3. T-SNE Visual Evaluation and Analysis

To more intuitively scrutinize the outstanding feature learning capability of the proposed HCFA, we utilize the t-distributed stochastic neighbor embedding (t-SNE) algorithm. This algorithm facilitates the mapping of high-dimensional sample features into a two-dimensional space, allowing us to assess the distinctiveness between various categories through observation. In the experiments, all t-SNE parameters were set to their default values, such as n_components = 2, perplexity = 30.0, and n_iter = 1000. The results of visualization without using LiDAR methods are shown in Figure 15a–c, the results without using the CFM module visualization are shown in Figure 15d–f, and the results of visualization using the full components are shown in Figure 15g–i. Comparing Figures 15a–c and 15d–f, it can be seen that the separability boundaries between features are more pronounced with the addition of LiDAR on all three datasets. This arises from the fact that relying solely on HSI lacks the sensitivity required to capture the three-dimensional shape details of objects. Conversely, LiDAR offers a high-precision shape and elevation features. By integrating HSI with LiDAR, we can gather a more comprehensive and detailed understanding of

three-dimensional spatial information. The discernible impact of the CFM module on classification performance further underscores this synergy. From Figure 15g–i, it can be observed that the distances between similar classes are relatively closer. This is because the CFM module aims to minimize information loss during the feature abstraction process, thereby enabling it to acquire more comprehensive and accurate information.

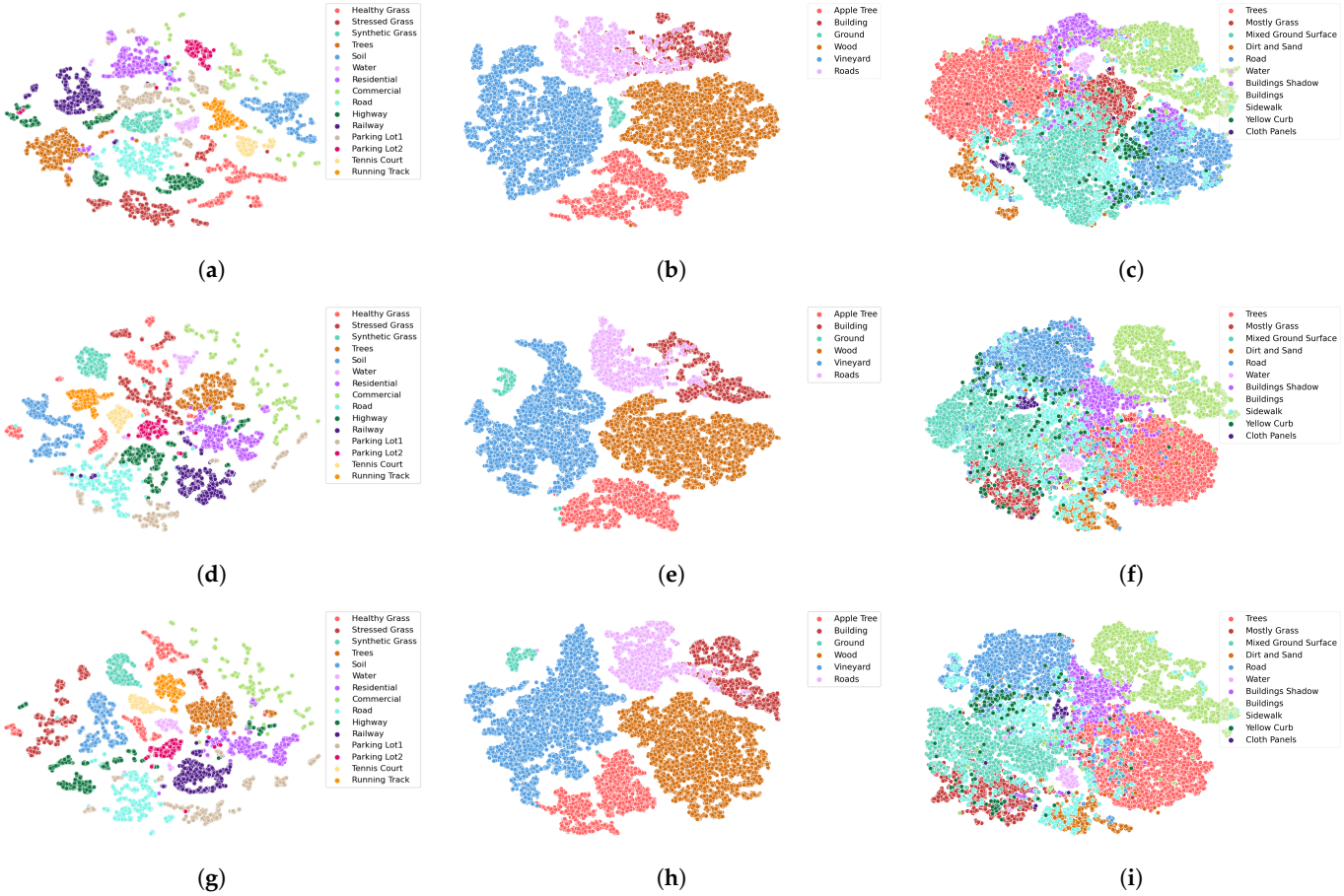
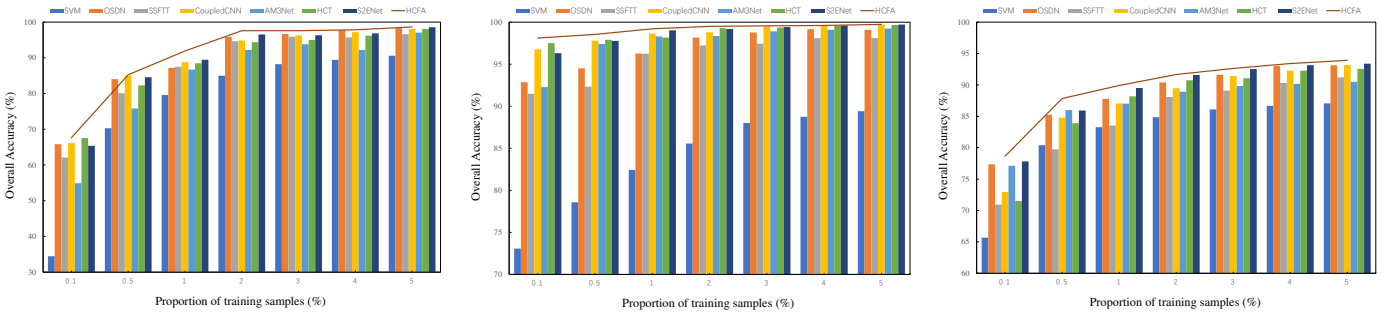


Figure 15. Distribution of the HCFA feature space for different cases given by the t-SNE algorithm: (**a**) Houston2013 without LiDAR; (**b**) Trento without LiDAR; (**c**) MUUFL without LiDAR; (**d**) Houston2013 without CFM; (**e**) Trento without CFM; (**f**) MUUFL without CFM; (**g**) Houston2013 with full component; (**h**) Trento with full component; (**i**) MUUFL with full component.

5.3. Investigation of the Proportion of Training Samples

The classification task in multi-source RS images usually faces the challenge of insufficient labelled samples due to a priori knowledge, manual labeling, and time constraints [53]. Robust algorithms can be adapted to a variety of conditions, including classification tasks, using a limited training set. To verify the robustness of HCFA, we randomly select 0.1%, 0.5%, 1%, 2%, 4%, and 5% labeled samples on three datasets to form the training set and compare it with other methods. The experimental results are shown in Figure 16. In general, a large number of training samples helps supervised learning methods to be adequately trained and provide sufficient discriminative information, thus improving the classification accuracy. Experimental results confirm this by showing that the OA of each algorithm is lower when the proportion of training samples is small and vice versa. With a training sample proportion of only 0.1%, the SVM performs the worst because, with a small number of training samples or an uneven dataset, SVM may not be able to adequately capture data complexity. In particular, in the Houston2013 dataset, SVM can only perform two-fold cross-validation due to the small number of training samples. The OA of the other deep learning methods varies by 10%. HCFA achieved the highest OA on all three datasets, indicating that it is better at extracting valid data features from a few training samples.

With the increase in training sample proportion, HCFA always delivers competitive results. The maximum classification accuracy for all methods is achieved when the training sample proportion reaches 5%. HCFA obtained the highest classification accuracy on all three datasets, 99.74% for Trento, 98.64% for Houston2013 and 93.92% for MUUFL. The experimental results demonstrate the robustness of HCFA in multi-source remote sensing image classification tasks.



(**a**) (**b**) (**c**)

Figure 16. Investigation of the proportion of training samples: (**a**) Houston2013 dataset; (**b**) Trento dataset; (**c**) MUUFL dataset.

5.4. Coarse–Fine-Grained Analysis

In this section, to more clearly verify the difference between coarse-grained and fine-grained features, the output of the Classification_1 block in Figure 2 is used as the coarse-grained classification, the output of Classification_2 is used as the fine-grained classification, and the output of the ALF block is used as the coarse–fine-grained fusion classification. In most cases, the OA, AA, and Kappa values of fine-grained classification are better than those of the coarse-grained classification, as shown in Table 7. This indicates that for most land cover classifications, fine-grained information is more effective than coarse-grained information. On the MUUFL dataset, although the OA values for coarse-grained classification are lower than those for coarse-grained classification, the AA values are higher, which suggests that coarse-grained information is helpful in classifying some covered land. For the Houston 2013, Trento, and MUUFL datasets, the OA values of coarse–fine-grained fusion classification are improved by 1.3%, 0.72%, and 1.47%, respectively, compared to coarse-grained classification. This indicates that coarse–fine-grained fusion classification outperforms single coarse-grained classification and fine-grained classification, validating the effectiveness of the fusion of coarse-grained and fine-grained classification methods.

Table 7. Comparison of coarse–fine-grained classification on the Houston 2013, Trento, and MUUFL datasets (Optimal results are in bold).

NO.	Houstom2013			Trento			MUUFL		
	OA (%)	**AA (%)**	**Kappa × 100**	**OA (%)**	**AA (%)**	**Kappa × 100**	**OA (%)**	**AA (%)**	**Kappa × 100**
Coarse	90.62 ± 00.16	92.35 ± 00.45	89.86 ± 00.18	98.49 ± 00.04	97.83 ± 00.05	97.99 ± 00.05	88.45 ± 00.01	81.86 ± 00.02	84.45 ± 00.01
Fine	91.86 ± 00.01	93.43 ± 00.01	91.20 ± 00.01	98.66 ± 00.00	98.17 ± 00.00	98.21 ± 00.00	89.14 ± 00.00	78.34 ± 00.01	85.52 ± 00.01
Coarse–Fine	**91.92 ± 00.04**	**93.59 ± 00.02**	**91.26 ± 00.04**	**99.22 ± 00.01**	**98.81 ± 00.01**	**98.96 ± 00.01**	**89.92 ± 00.01**	**86.68 ± 00.02**	**86.63 ± 00.01**

5.5. Complexity Analysis

The number of parameters and floating-point operations (FLOPs) of different methods on three datasets are shown in Table 8. The number of parameters and FLOPs are calculated with the batch size of 1, and all data in the table are computed using methods provided by the THOP library. It can be observed that the size of the dataset also affects the number of parameters and FLOPs. In particular, the results of all methods on the Trento dataset are similar to those on the MUUFL dataset, as the sizes of these two datasets are similar. In contrast, the data of AM^{3}Net seems to be unaffected by the size of the dataset because this network initially uses a (1×1) convolution for dimension transformation. Our proposed

HCFA performs moderately in terms of the number of parameters but excels in FLOPs. This is attributed to the adoption of dynamic convolutions and efficient network structures, resulting in achieving higher representational power with smaller FLOPs.

Table 8. Number of parameters and flops of the competitors (Optimal results are in bold).

Dataset	Metric	OSDN	SSFTT	CoupledCNN	AM^{3}Net	HCT	S^{2}ENet	HCFA
Houston2013	Parameters (M)	**0.06**	0.67	0.14	2.65	0.96	0.27	0.41
	FLOPs (MMac)	29.99	6.86	6.62	7.12	7.76	13.27	**1.27**
Trento	Parameters (M)	**0.04**	0.30	0.12	2.65	0.58	0.18	0.14
	FLOPs (MMac)	12.91	3.00	3.79	7.05	3.90	8.69	**0.83**
MUUFL	Parameters (M)	**0.04**	0.30	0.11	2.65	0.59	0.18	0.15
	FLOPs (MMac)	12.91	3.05	3.83	7.05	3.95	8.75	**0.83**

6. Conclusions

In this article, we propose a coarse–fine mutual learning fusion network based on a dynamic CNN and transformer, aiming to achieve accurate joint classification of HSI and LiDAR data. First, coarse-grained feature extraction is performed using dual-branch dynamic convolution. Compared to traditional CNNs, this method not only significantly enhances feature representation capability but also reduces additional computational costs. Second, to better integrate multi-source RS information, a transformer based on cross-token attention is designed. It can make full use of the complementarities between the different modalities and obtain discriminative fine-grained fusion features, enabling interactive learning of multi-source RS information excellently. In addition, a coarse and fine adaptive mutual learning strategy is used to achieve optimal balanced classification between local and global information, as well as coarse and fine granularity information. Experimental results on three sets of public HSI and LiDAR datasets and comparisons with existing classification methods demonstrate that the proposed approach has good classification performance and superior efficiency. A series of ablation experiments verify the effectiveness of dynamic convolution, CT, and CFM modules in multi-source RS classification within the HCFA method. The impact of hyperparameters, computational costs, and the visualization of data features are also discussed. In the future, we will further explore the coarse–fine mutual learning strategy with more heterogeneous RS data to achieve broader and deeper analysis and applications on the RS data.

Author Contributions: Conceptualization, H.P. and X.L.; data curation, X.L.; formal analysis, H.P., X.L., H.G. and L.W.; methodology, H.P., X.L., H.G. and L.W.; software, X.L.; writing original draft, X.L.; writing review and editing, H.P., H.G., L.W. and C.S. All authors have read and agree to the published version of the manuscript.

Funding: This research was funded by the Heilongjiang Provincial Natural Science Foundation of China, grant number LH2023F050, Fundamental Research Funds in Heilongjiang Provincial Universities under Grant, grant number 145309208, and the National Natural Science Foundation of China, grant number 42271409.

Data Availability Statement: Data available in a publicly accessible repositories: Houston 2013 dataset (https://www.grss-ieee.org/resources/tutorials-documents/, accessed on 1 September 2024), MUUFL dataset (https://github.com/GatorSense/MUUFLGulfport/, accessed on 1 September 2024), and Trento dataset (https://drive.google.com/drive/folders/1HK3eL3loI4Wd-RFr1psLLmVLTVDLctGd, accessed on 1 September 2024).

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

References

1. Rast, M.; Painter, T.H. Earth observation imaging spectroscopy for terrestrial systems: An overview of its history, techniques, and applications of its missions. *Surv. Geophys.* **2019**, *40*, 303–331. [CrossRef]
2. Fauvel, M.; Tarabalka, Y.; Benediktsson, J.A.; Chanussot, J.; Tilton, J.C. Advances in spectral-spatial classification of hyperspectral images. *Proc. IEEE* **2012**, *101*, 652–675. [CrossRef]

3. Wang, Z.; Menenti, M. Challenges and opportunities in Lidar remote sensing. *Front. Remote Sens.* **2021**, *2*, 641723. [CrossRef]
4. Mondini, A.C.; Guzzetti, F.; Chang, K.T.; Monserrat, O.; Martha, T.R.; Manconi, A. Landslide failures detection and mapping using Synthetic Aperture Radar: Past, present and future. *Earth-Sci. Rev.* **2021**, *216*, 103574. [CrossRef]
5. Wang, X.; Feng, Y.; Song, R.; Mu, Z.; Song, C. Multi-attentive hierarchical dense fusion net for fusion classification of hyperspectral and LiDAR data. *Inf. Fusion* **2022**, *82*, 1–18. [CrossRef]
6. Yan, W.Y.; Shaker, A.; El-Ashmawy, N. Urban land cover classification using airborne LiDAR data: A review. *Remote Sens. Environ.* **2015**, *158*, 295–310. [CrossRef]
7. Sharma, M.; Dhanaraj, M.; Karnam, S.; Chachlakis, D.G.; Ptucha, R.; Markopoulos, P.P.; Saber, E. YOLOrs: Object detection in multimodal remote sensing imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *14*, 1497–1508. [CrossRef]
8. Yu, L.; Yang, Q.; Dong, L. Aircraft target detection using multimodal satellite-based data. *Signal Process.* **2019**, *155*, 358–367. [CrossRef]
9. Pan, B.; Xu, X.; Shi, Z.; Zhang, N.; Luo, H.; Lan, X. DSSNet: A simple dilated semantic segmentation network for hyperspectral imagery classification. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 1968–1972. [CrossRef]
10. 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]
11. Li, Y.; Zhou, Y.; Zhang, Y.; Zhong, L.; Wang, J.; Chen, J. DKDFN: Domain knowledge-guided deep collaborative fusion network for multimodal unitemporal remote sensing land cover classification. *ISPRS J. Photogramm. Remote Sens.* **2022**, *186*, 170–189. [CrossRef]
12. Cheng, M.; Li, B.; Jiao, X.; Huang, X.; Fan, H.; Lin, R.; Liu, K. Using multimodal remote sensing data to estimate regional-scale soil moisture content: A case study of Beijing, China. *Agric. Water Manag.* **2022**, *260*, 107298. [CrossRef]
13. Jurado, J.M.; López, A.; Pádua, L.; Sousa, J.J. Remote sensing image fusion on 3D scenarios: A review of applications for agriculture and forestry. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *112*, 102856. [CrossRef]
14. Audebert, N.; Le Saux, B.; Lefèvre, S. Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks. *ISPRS J. Photogramm. Remote Sens.* **2018**, *140*, 20–32. [CrossRef]
15. Dremin, V.; Marcinkevics, Z.; Zherebtsov, E.; Popov, A.; Grabovskis, A.; Kronberga, H.; Geldnere, K.; Doronin, A.; Meglinski, I.; Bykov, A. Skin complications of diabetes mellitus revealed by polarized hyperspectral imaging and machine learning. *IEEE Trans. Med. Imaging* **2021**, *40*, 1207–1216. [CrossRef]
16. Peyghambari, S.; Zhang, Y. Hyperspectral remote sensing in lithological mapping, mineral exploration, and environmental geology: An updated review. *J. Appl. Remote Sens.* **2021**, *15*, 031501. [CrossRef]
17. Gu, Y.; Wang, Q. Discriminative graph-based fusion of HSI and LiDAR data for urban area classification. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 906–910. [CrossRef]
18. Rasti, B.; Ghamisi, P.; Plaza, J.; Plaza, A. Fusion of hyperspectral and LiDAR data using sparse and low-rank component analysis. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 6354–6365. [CrossRef]
19. Ghamisi, P.; Rasti, B.; Benediktsson, J.A. Multisensor composite kernels based on extreme learning machines. *IEEE Geosci. Remote Sens. Lett.* **2018**, *16*, 196–200. [CrossRef]
20. 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.* **2020**, *19*, 1–5. [CrossRef]
21. Xu, X.; Li, W.; Ran, Q.; Du, Q.; Gao, L.; Zhang, B. Multisource remote sensing data classification based on convolutional neural network. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 937–949. [CrossRef]
22. 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]
23. Camalan, S.; Cui, K.; Pauca, P.; Alqahtani, S.; Silman, M.; Chan, R.; Dethier, E.; Fernandez, L.E.; Lutz, D.A.; Plemmons, R. Detecting change due to alluvial gold mining in peruvian rainforest using recursive convolutional neural networks and contrastive learning. *AGU Fall Meet. Abstr.* **2022**, *2022*, B52G–0905.
24. Prexl, J.; Schmitt, M. Multi-modal multi-objective contrastive learning for sentinel-1/2 imagery. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 2136–2144. [CrossRef]
25. 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.* **2020**, *59*, 4340–4354. [CrossRef]
26. Hong, D.; Han, Z.; Yao, J.; Gao, L.; Zhang, B.; Plaza, A.; Chanussot, J. SpectralFormer: Rethinking hyperspectral image classification with transformers. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–15. [CrossRef]
27. Zhao, X.; Zhang, M.; Tao, R.; Li, W.; Liao, W.; Tian, L.; Philips, W. Fractional Fourier image transformer for multimodal remote sensing data classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *35*, 2314–2326. [CrossRef]
28. Sun, L.; Zhao, G.; Zheng, Y.; Wu, Z. Spectral–spatial feature tokenization transformer for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–14. [CrossRef]
29. Yuan, F.; Zhang, Z.; Fang, Z. An effective CNN and Transformer complementary network for medical image segmentation. *Pattern Recognit.* **2023**, *136*, 109228. [CrossRef]
30. Li, Z.; Huang, W.; Wang, L.; Xin, Z.; Meng, Q. CNN and Transformer interaction network for hyperspectral image classification. *Int. J. Remote Sens.* **2023**, *44*, 5548–5573. [CrossRef]
31. Zhang, J. Multi-source remote sensing data fusion: Status and trends. *Int. J. Image Data Fusion* **2010**, *1*, 5–24. [CrossRef]

32. Khaleghi, B.; Khamis, A.; Karray, F.O.; Razavi, S.N. Multisensor data fusion: A review of the state-of-the-art. *Inf. Fusion* **2013**, *14*, 28–44. [CrossRef]
33. Liu, Y.; Chen, X.; Wang, Z.; Wang, Z.J.; Ward, R.K.; Wang, X. Deep learning for pixel-level image fusion: Recent advances and future prospects. *Inf. Fusion* **2018**, *42*, 158–173. [CrossRef]
34. Jahan, F.; Zhou, J.; Awrangjeb, M.; Gao, Y. Inverse coefficient of variation feature and multilevel fusion technique for hyperspectral and LiDAR data classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 367–381. [CrossRef]
35. Farooque, G.; Xiao, L.; Sargano, A.B.; Abid, F.; Hadi, F. A dual attention driven multiscale-multilevel feature fusion approach for hyperspectral image classification. *Int. J. Remote Sens.* **2023**, *44*, 1151–1178. [CrossRef]
36. Mu, C.; Guo, Z.; Liu, Y. A multi-scale and multi-level spectral-spatial feature fusion network for hyperspectral image classification. *Remote Sens.* **2020**, *12*, 125. [CrossRef]
37. Kalluri, H.R.; Prasad, S.; Bruce, L.M. Decision-level fusion of spectral reflectance and derivative information for robust hyperspectral land cover classification. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 4047–4058. [CrossRef]
38. Li, J.; Hong, D.; Gao, L.; Yao, J.; Zheng, K.; Zhang, B.; Chanussot, J. Deep learning in multimodal remote sensing data fusion: A comprehensive review. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *112*, 102926. [CrossRef]
39. Shao, Z.; Cai, J. Remote sensing image fusion with deep convolutional neural network. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 1656–1669. [CrossRef]
40. Zhou, Y.; Chen, S.; Wang, Y.; Huan, W. Review of research on lightweight convolutional neural networks. In Proceedings of the 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 12–14 June 2020; pp. 1713–1720. [CrossRef]
41. Chen, Y.; Dai, X.; Liu, M.; Chen, D.; Yuan, L.; Liu, Z. Dynamic convolution: Attention over convolution kernels. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11030–11039. [CrossRef]
42. Huo, Y.; Lu, Y.; Niu, Y.; Lu, Z.; Wen, J.R. Coarse-to-fine grained classification. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 1033–1036. [CrossRef]
43. Zhang, T.; Xiao, S.; Dong, W.; Qu, J.; Yang, Y. A mutual guidance attention-based multi-level fusion network for hyperspectral and LiDAR classification. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 1–5. [CrossRef]
44. Debes, C.; Merentitis, A.; Heremans, R.; Hahn, J.; Frangiadakis, N.; van Kasteren, T.; Liao, W.; Bellens, R.; Pižurica, A.; Gautama, S.; et al. Hyperspectral and LiDAR data fusion: Outcome of the 2013 GRSS data fusion contest. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2405–2418. [CrossRef]
45. Jia, S.; Zhan, Z.; Xu, M. Shearlet-based structure-aware filtering for hyperspectral and LiDAR data classification. *J. Remote Sens.* **2021**. [CrossRef]
46. Gader, P.; Zare, A.; Close, R.; Aitken, J.; Tuell, G. *MUUFL Gulfport Hyperspectral and LiDAR Airborne Data Set*; Tech. Rep. REP-2013-570; University of Florida: Gainesville, FL, USA, 2013.
47. Du, X.; Zare, A. *Technical Report: Scene Label Ground Truth Map for MUUFL Gulfport Data Set*; University of Florida: Gainesville, FL, USA, 2017.
48. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [CrossRef]
49. Pan, H.; Liu, M.; Ge, H.; Wang, L. One-shot dense network with polarized attention for hyperspectral image classification. *Remote Sens.* **2022**, *14*, 2265. [CrossRef]
50. Wang, J.; Li, J.; Shi, Y.; Lai, J.; Tan, X. AM^{3}Net: Adaptive Mutual-Learning-Based Multimodal Data Fusion Network. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 5411–5426. [CrossRef]
51. 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.* **2022**, *61*, 1–16. [CrossRef]
52. Fang, S.; Li, K.; Li, Z. S^{2}ENet: Spatial–spectral cross-modal enhancement network for classification of hyperspectral and LiDAR data. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 1–5. [CrossRef]
53. Jia, S.; Jiang, S.; Lin, Z.; Li, N.; Xu, M.; Yu, S. A survey: Deep learning for hyperspectral image classification with few labeled samples. *Neurocomputing* **2021**, *448*, 179–204. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.