remote sensing

MDPI

Article

Cross Attention-Based Multi-Scale Convolutional Fusion Network for Hyperspectral and LiDAR Joint Classification

Haimiao Ge [1,2], Liguo Wang [3,*], Haizhu Pan [1,2], Yanzhong Liu [1,2], Cheng Li [1,2], Dan Lv [1,2] and Huiyu Ma [1,2]

[1] College of Computer and Control Engineering, Qiqihar University, Qiqihar 161000, China; 01557@qqhru.edu.cn (H.G.); panhaizhu@qqhru.edu.cn (H.P.); 01514@qqhru.edu.cn (Y.L.); licheng@qqhru.edu.cn (C.L.); 02243@qqhru.edu.cn (D.L.); 03253@qqhru.edu.cn (H.M.)
[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 Minzu University, Dalian 116600, China
* Correspondence: wangliguo@hrbeu.edu.cn; Tel.: +86-131-0166-7835

Abstract: In recent years, deep learning-based multi-source data fusion, e.g., hyperspectral image (HSI) and light detection and ranging (LiDAR) data fusion, has gained significant attention in the field of remote sensing. However, the traditional convolutional neural network fusion techniques always provide poor extraction of discriminative spatial–spectral features from diversified land covers and overlook the correlation and complementarity between different data sources. Furthermore, the mere act of stacking multi-source feature embeddings fails to represent the deep semantic relationships among them. In this paper, we propose a cross attention-based multi-scale convolutional fusion network for HSI-LiDAR joint classification. It contains three major modules: spatial–elevation–spectral convolutional feature extraction module (SESM), cross attention fusion module (CAFM), and classification module. In the SESM, improved multi-scale convolutional blocks are utilized to extract features from HSI and LiDAR to ensure discriminability and comprehensiveness in diversified land cover conditions. Spatial and spectral pseudo-3D convolutions, pointwise convolutions, residual aggregation, one-shot aggregation, and parameter-sharing techniques are implemented in the module. In the CAFM, a self-designed local-global cross attention block is utilized to collect and integrate relationships of the feature embeddings and generate joint semantic representations. In the classification module, average polling, dropout, and linear layers are used to map the fused semantic representations to the final classification results. The experimental evaluations on three public HSI-LiDAR datasets demonstrate the competitiveness of the proposed network in comparison with state-of-the-art methods.

Keywords: HSI and LiDAR fusion classification; convolutional neural network; multi-scale feature extraction; cross attention

check for updates

Citation: Ge, H.; Wang, L.; Pan, H.; Liu, Y.; Li, C.; Lv, D.; Ma, H. Cross Attention-Based Multi-Scale Convolutional Fusion Network for Hyperspectral and LiDAR Joint Classification. *Remote Sens.* **2024**, *16*, 4073. https://doi.org/10.3390/rs16214073

Academic Editors: Riccardo Roncella, Mattia Previtali and Luca Perfetti

Received: 14 September 2024
Revised: 21 October 2024
Accepted: 29 October 2024
Published: 31 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

In recent years, remote sensing technology has played a crucial role in Earth observation tasks [1]. With the development of sensor technology, remote sensing imaging methods exhibit a diversified trend [2]. Although an abundance of multi-source data are now available, remote sensing data from each source captures only one or a few specific properties, which cannot fully describe the scenes observed [3,4]. Naturally, multi-source remote sensing data fusion techniques present a feasible resolution to the predicament. By integrating complementary information from multi-source data, tasks can be performed more reliably and accurately [5,6]. Specifically, light detection and ranging (LiDAR) data can provide additional elevation information to hyperspectral image (HSI) data. In this way, the joint land cover classification of HSI-LiDAR data becomes a promising approach that has received favorable results in practical tasks [7].

Depending on the sensor types of remote sensing data, multi-source remote sensing data can be categorized as homogeneous data or heterogeneous data [8]. HSI-LiDAR data are heterogeneous remote sensing data that contains two forms of characteristics, namely spatial–spectral features in the HSI and spatial–elevation features in the LiDAR data [9]. Depending on the hierarchical level of data fusion, multi-source remote sensing joint classification techniques can be further categorized into pixel-level, feature-level, and decision-level approaches [10]. Due to the vast variation in the target characteristics being observed, the joint HSI-LiDAR data fusion is frequently processed by feature-level or decision-level techniques. In general, a typical HSI-LiDAR fusion classification network consists of four components, including the HSI feature extraction module, LiDAR feature extraction module, feature embedding fusion module, and classification module, which is shown in Figure 1.

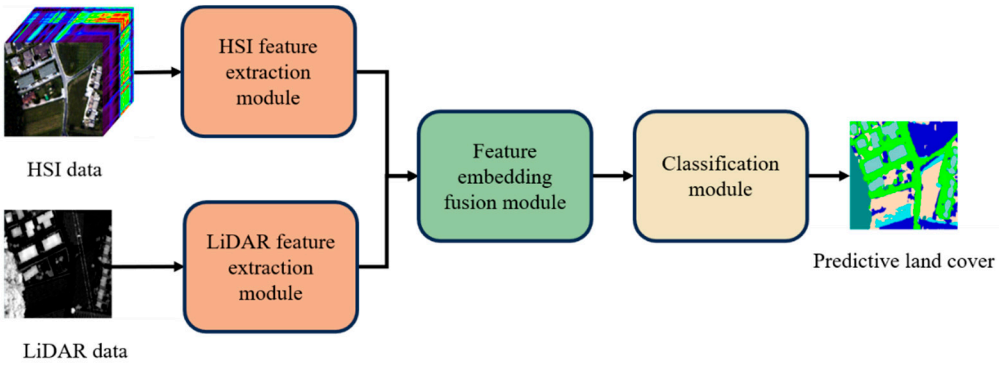


Figure 1. A typical HSI-LiDAR fusion classification network diagram.

For the HSI feature extraction module, the feature extraction approaches mainly include convolution-based techniques, recurrent-based techniques, transformer-based techniques, and attention-based techniques. Many convolutional neural network-based (CNN) approaches use 2D convolution to learn local contextual information from pixel-centric data cubes [11,12]. However, these methods devote insufficient attention to the spectral signatures and fail to consider the joint spatial–spectral information in HSI. Naturally, some scholars use 2D-3D convolutions to improve feature extraction modules to obtain joint spatial–spectral feature embeddings, which has achieved promising results in practical applications [13,14]. Nevertheless, 3D convolution exposes increased computation in spectral dimensions over 2D convolution, which can dramatically raise the number of parameters and boost the computational complexity. Moreover, 2D and 3D convolutions are restricted by the receptive field and omit the long-distance dependencies among features. To capture the long-distance dependencies of spectral signatures, some scholars regard the spectral signatures of HSI as time series signals and employ recurrent neural network (RNN) techniques to process them [15]. However, the sequential processing of RNN is time-consuming, which limits its application in practical tasks. In recent years, transformers and attention mechanisms have proposed to provide new thoughts for HSI feature extraction [16,17]. These approaches can capture long-distance dependencies of feature embeddings in parallel [18,19]. However, the attention mechanisms provide less ability to extract spatial information compared to the CNN models. Furthermore, the training and inference speed of transformer models is strongly influenced by data size and model structure. To solve these issues, the mamba models have emerged as a promising approach, which has strong long-distance modeling capabilities while maintaining a linear computational complexity [20,21]. However, it is hard for mamba models to provide an integrated spatial and spectral understanding of HSI features. For the LiDAR feature extraction module, 2D CNNs and attention mechanisms are widely used, and their high performance has been demonstrated in real tasks [22–24].

For the feature embedding fusion module, the fusion techniques mainly include concatenate fusion, hierarchical fusion, residual fusion, graph-based fusion, transformer-

and attention-based fusion. The concatenate fusion technique is introduced to achieve data fusion by combining feature embeddings of different input data into a joint feature [25,26]. However, this approach only performs simple stacking of multi-source features and only performs well on some small-scale datasets. In contrast, the hierarchical fusion technique is an effective improvement over the concatenate fusion approach. In hierarchical fusion, shallow and deep features, which are extracted in different hierarchies, interact with each other to perform fusion [23,27,28]. In the feature fusion process, different attention modules are used for hierarchical fusion to achieve complete complementarity of features, which are expressed as shallow fusion and deep fusion. However, the effectiveness of hierarchical fusion depends on the quality of the features extracted from the network at each level of the model, which limits the application of the method to complex tasks. Residual fusion uses CNN and residual aggregation to realize data fusion [29,30]. However, the method works poorly at handling the heterogeneous data fusion problem. To represent the relationship between multi-source features, scholars tried to use graph-based fusion techniques to achieve data fusion [9,31]. However, the computational complexity of the graph-based fusion technique is significantly influenced by the number of nodes and edges, which makes it unsuitable to apply to large-scale data. The transformer- and attention-based fusion techniques utilize attention mechanism to achieve data fusion [32,33]. This approach can capture long-distance dependencies of feature embeddings and provide a high-performance representation of the relationship between multi-source features [34,35]. However, the transformer and attention mechanisms are less sensitive to location information and lack the ability to collect local information. For the classification module, it serves to map the fused features to the final classification results. Linear layers and adaptive weighting techniques are always used in this component to enhance the robustness and generalization of the model [36,37].

In this paper, a cross attention-based multi-scale convolutional fusion network (CMCN) is proposed for HSI-LiDAR land cover classification. The approach majorly consists of three modules: spatial–elevation–spectral convolutional feature extraction module (SESM), cross attention fusion module (CAFM), and classification module. Two considerations are focused on improving the performance of the network. First, discriminative spatial–spectral and spatial–elevation features are extracted in diversified land cover conditions, and the correlative and complementary information is preserved. Second, the long-distance dependencies of feature embeddings are captured, and the representation of deep semantic relations for multi-source data is achieved. To capture discriminative spatial–spectral and spatial–elevation features and preserve the correlation and complementarity, improved multi-scale convolutional blocks are utilized in the SESM to extract features from HSI and LiDAR data. Spatial and spectral pseudo-3D [38] convolutions and pointwise convolutions are jointly used to extract spatial–elevation–spectral features and simplify the computation. Residual and one-shot aggregations are employed to maintain shallow features in deep layers and make the network easier to train. The parameter-sharing technique is used to exploit the correlation and complementarity. To capture the long-distance dependencies and achieve the relations of the feature embeddings, a local-global cross attention mechanism is applied in the CAFM to collect the local contexture features and integrate the global significant relational semantic information. A classification module is implemented to collect the fused features and translate them into the final classification results. More details about the proposed method are presented in Section 3. The main contributions are summarized as follows:

1. A multi-scale convolutional feature extraction module is designed to extract spatial–elevation–spectral features from HSI and LiDAR data. In this module, spatial and spectral pseudo-3D multi-scale convolutions and pointwise convolutions are jointly utilized to extract discriminative features, which can enhance the ability to extract ground characteristics in diversified environments. Residual and one-shot aggregations are employed to maintain the shallow features and ensure convergence. To capture the correlation and complementar-

ity of spatial and elevation information among HSI and LiDAR data, a parameter-sharing technique is applied to generate feature embeddings;

2. A local–global cross attention block is designed to collect and integrate effective information from multi-source feature embeddings. To collect the local information, local-based convolutional layers are implemented to perform the mapping transformation. After that, the global cross attention mechanism is applied to achieve long-distance dependencies and generate attention weights. Then, multiplication operation and residual aggregation are used to produce semantic representations and accomplish data fusion;

3. A novel cross attention-based multi-scale convolutional fusion network is proposed to achieve the joint classification of HSI and LiDAR data. A multi-scale CNN framework with parameter sharing and a local–global cross attention mechanism are combined to exploit joint deep semantic representations of HSI and LiDAR data and achieve data fusion. The classification module is implemented to perform classification results. Experimental results on three publicly available datasets are reported.

The rest of the paper is organized as follows. Section 2 introduces the related work, such as HSI and LiDAR data classification, residual and one-shot aggregations, and the cross attention mechanism. Section 3 presents the details of the proposed network. Section 4 gives the experimental results, and Section 5 makes some discussions. Section 6 gives the conclusion of this article and provides future work.

2. Related Work

2.1. HSI and LiDAR Data Classification

Different from traditional HSI classification models, the joint HSI and LiDAR classification models are required to consider how to capture correlative and complementary information and represent the relationship of HSI and LiDAR data while extracting discriminative features. Shallow learning methods use a stacking approach to combine features to achieve joint classification results. However, these methods only utilize the respective features of HSI and LiDAR data and fail to achieve genuine feature-level fusion. Improved approaches utilize dimensionality reduction [39] and related subspaces [40,41] technologies to align HSI and LiDAR data in a shared feature space, where the joint features of HSI and LiDAR can be well expressed. However, feature mapping is a challenging problem due to its ill-defined nature, making it difficult to find efficient ways to quantitatively map HSI and LiDAR data.

Unlike shallow learning methods, deep learning methods use deep neural networks to achieve feature extraction, feature fusion, and discriminative decisions. The original approaches use a combination of shallow feature extraction and deep learning to achieve HSI-LiDAR fusion classification. For example, Ghamisi et al. [42] propose an HSI-LiDAR data fusion framework using extinction profiles and deep learning. The fused features, which are extracted by extinction profiles, are fed to a deep learning-based classifier to ultimately produce the classification results. Later, the feedforward neural network (FNN), residual network (Resnet), and squeeze-and-excitation network (SEnet) are employed to implement HSI-LiDAR fusion classification. For example, Chen et al. [43] propose an HSI-LiDAR data fusion network, which utilizes a two-branch network to separately extract spectral–spatial–elevation features and then utilizes FNN to integrate these features for the final classification. Ge et al. [30] propose a deep Resnet-based fusion framework for HSI-LiDAR data. Three fusion methods are implemented to enhance the effectiveness of the method, which are the residual network-based deep feature fusion, the residual network-based probability reconstruction fusion, and the residual network-based probability multiplication fusion. Feng et al. [44] incorporate squeeze-and-excitation networks into the fusion step to adaptively realize the feature calibration. Although these deep learning-based fusion methods represent an improvement over traditional methods, the feature extraction, feature fusion, and discriminative decision processes are basic and rough, which constrains the practicality of these methods. To further improve the capability of the fusion classification network, many improved methods are proposed, such as CNN,

transformer, attention, and mamba techniques. For the CNN methods, Yu et al. [11] propose a simplified CNN architecture to classify HSI and 2D convolution blocks that are used to extract the spatial features of abundantly involved spectral information as training channels. Roy et al. [45] propose a hybrid 2D-3D spectral CNN for HSI classification. Li et al. [46] propose an HSI-LiDAR data fusion network based on a convolutional neural network and composite kernels. A three-stream CNN is designed to extract the spectral, spatial, and elevation features from HSI-LiDAR datasets, and a multi-sensor composite kernels scheme is designed to fuse the extracted features and produce the final classification results. In [47], a novel HSI-LiDAR classification method based on multi-view feature learning and multi-level information fusion is proposed. For the transformer methods, Zhang et al. [32] propose a transformer and multi-scale fusion network for HSI and LiDAR joint classification. Zhao et al. [33] propose a hierarchical CNN and transformer network for HSI and LiDAR joint classification. For the attention techniques, in [48], a dual-channel spatial, spectral, and multi-scale attention convolutional network is proposed for HSI-LiDAR data fusion. A novel composite attention learning mechanism is introduced to fully integrate the features in the data sources. Song et al. [49] propose a multi-scale pseudo-Siamese network with an attention mechanism to fuse HSI and LiDAR data. In [35], an attention-guided fusion and classification framework based on a convolutional neural network is proposed to classify the land cover of HSI and LiDAR data. The spectral attention mechanism is adopted to assign weights to the spectral channels, and the cross attention mechanism is introduced to impart significant spatial weights from LiDAR to HSI. Li et al. [50] propose a morphological convolution and attention calibration network for HSI-LiDAR classification. A dual attention module is designed to extract features of the input data. For the mamba methods, Li et al. [20] propose a mamba-based model to exploit long-range interaction of the whole image and spatial–spectral information for the HSI dataset. These methods use deep learning techniques to enhance the data fusion capabilities to some extent.

2.2. Residual Aggregation and One-Shot Aggregation

The depth of the neural network is critical to the performance of the model. When increasing the depth of the neural network, the model can perform more complicated feature mapping, which can theoretically provide better performance. However, the experiments find that the deep neural network has a degradation problem, i.e., the network accuracy saturates and even decreases when the depth of the network increases. To address this problem, residual aggregation is proposed to incorporate residual pathways by explicitly modifying the network structure and retaining shallow features using identity mapping. Given H as a hidden layer, $\mathbf{F}$ as a feature map, and $\oplus$ as a summation operator, the output feature map of the lth hidden layer can be expressed as

$$\mathbf{F}_l = H_l(\mathbf{F}_{l-1}) \oplus \mathbf{F}_{l-1} \tag{1}$$

However, experiments show that multiple summation operations will wash away the information embedded in the previous features. To better maintain the previous feature maps, one-shot aggregation [51] is proposed to use a concatenation operation instead of a summation operation and simultaneously aggregate the previous feature maps to the target lth layer at one time. Given $\mathrm{Cat}(\cdot)$ as a concatenation operator, the process can be expressed as

$$\mathbf{F}_l = Cat(H_1(\mathbf{F}_1), H_2(\mathbf{F}_2), \ldots, H_{l-1}(\mathbf{F}_{l-1})) \tag{2}$$

2.3. Cross Attention Mechanism

The attention mechanism [52] is a feature transformation method that is based on the human perceptual process. It is designed to focus more on the informative areas while taking into account nonessential areas to a lesser extent [53]. Specifically, given queries, keys, and values are input vectors, and the output is output vector. The attention mechanism is a process that maps queries and a set of key-value pairs to weights, which are used to transform the values to obtain output [54]. The attention mechanism can capture long-

range interactions of input vectors and has shown great potential in fields such as natural language processing and computer vision [38]. The cross attention mechanism [55,56] uses multi-source input vectors for queries, keys, and values. It generates contextual representations for the values by computing associations between queries and keys and helps the model understand the relationship between multi-source features. For instance, Liu et al. [57] propose a multi-scale cross-interaction attention network for HSI classification, and cross attention is implemented to detect spectral–spatial features at different scales. In [58], a spatial–spectral cross attention module is proposed to extract the interactive spatial–spectral fusion feature intra-transformer block. Yang et al. [59] introduce a cross attention spectral–spatial network for HSI classification. Cross-spectral and cross-spatial attention components are proposed to capture spectral and spatial features from HSI patches. In [60], the cross attention mechanism is used to select the HSI bands, which are guided by LiDAR data. In [61], cross HSI and LiDAR attention is introduced. In the approach, LiDAR patch tokens serve as queries, while keys and values are derived from HSI patch tokens. The fused features are used for the joint classification of HSI and LiDAR data.

3. Methodology

The proposed method is introduced in this section. First, the preliminary is given. After that, the SESM, CAFM, and classification modules are described in detail. Finally, the overall framework of the proposed model is discussed.

3.1. Preliminary

Mathematically, given an HSI dataset $\mathbf{X}_{HSI} \in \mathbb{R}^{H\times W\times c}$ and its corresponding LiDAR dataset $\mathbf{X}_{LiDAR} \in \mathbb{R}^{H\times W\times 1}$, where H and W refer to the height and width of the two datasets, c is the spectral size of HSI. The input HSI cube can be described as $\mathbf{x}^i_{HSI} \in \mathbb{R}^{f\times h\times w\times c}$, where $\mathbf{x}^i_{HSI}$ represents the cube-based HSI data of ith pixel, f is the number of feature maps (f is set to 1 when initializing the input data), $h \times w$ is the patch size, and $h = w$. The input LiDAR cube can be presented as $\mathbf{x}^i_{LiDAR} \in \mathbb{R}^{f\times h\times w\times 1}$, which is the cube-based LiDAR data of ith pixel. The ith input data of the proposed network can be denoted as $\mathbf{x}^i = \left(\mathbf{x}^i_{HSI}, \mathbf{x}^i_{LiDAR}\right)$. The ground-truth label of the ith input data is expressed as $y^i_{true} \in \{1, 2, \ldots, C\}$, where C is the number of classes. The output of the network is expressed as $\mathbf{y}^i_{pre} \in \mathbb{R}^{1\times C}$. For quick reference, the notations used in this paper are summarized in Table 1.

Table 1. A summary of notations used in this paper.

Symbols	Definitions
$\mathbf{X}_{HSI}, \mathbf{X}_{LiDAR}$	The original input data of HSI and LiDAR
H, W	The height and width of the original HSI and LiDAR
c	The spectral size of HSI
C	The number of classes
$\mathbf{x}^i_{HSI}, \mathbf{x}^i_{LiDAR}$	The cube-based input HSI and LiDAR data
$\mathbf{x}^i$	The merged cube-based input HSI and LiDAR data
h, w	The patch size of HSI and LiDAR data
f	The number of feature maps
$y^i_{true}, \mathbf{y}^i_{pre}$	The ground-truth and predictive label of the ith input data
$\mathcal{B}_{\mathcal{H}1}, \mathcal{B}_{\mathcal{H}2}, \mathcal{B}_{\mathcal{L}}$	The triple-branch of SESM
$\mathbf{F}_{\mathcal{H}1}, \mathbf{F}_{\mathcal{H}2}, \mathbf{F}_{\mathcal{L}}$	The corresponding output of the triple-branch
$\mathbf{F}_{HH}, \mathbf{F}_{HL}$	The output of stage 1 in CAFM
$\mathbf{F}_{HHL}$	The output of stage 2 in CAFM
r, h	The multiplicity of channel compression and number of heads in the attention mechanism
$\mathcal{L}$	The cross-entropy loss

3.2. The Overall Framework of the Proposed Method

The proposed framework mainly consists of three modules. Initially, the cube-based inputs are fed into the SESM. Pseudo-3D convolutions and pointwise convolutions are used to extract discriminative features. The parameter-sharing technique is employed to exploit correlation and complementarity. Then, the extracted features are passed through a local–global cross attention fusion module. Local-based convolutions and global-based cross attention mechanisms are combined to represent the relationships of multi-source data and achieve data fusion. Finally, the fused features are connected to the classification module to provide classification results. The overall structure of the proposed method is shown in Figure 2. Early stopping and dynamic learning rate technologies are implemented to reduce the training time and provide better network convergence. Cross-entropy loss is applied in the proposed network, which can be expressed as

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{j=1}^{C} y_{true}^{ij} log\left(y_{pre}^{ij}\right) \tag{3}$$

where y_{true}^{ij} represents the jth element of the label y_{true}^{i} and y_{pre}^{ij} represents the probability that the pixel belongs to the jth class.

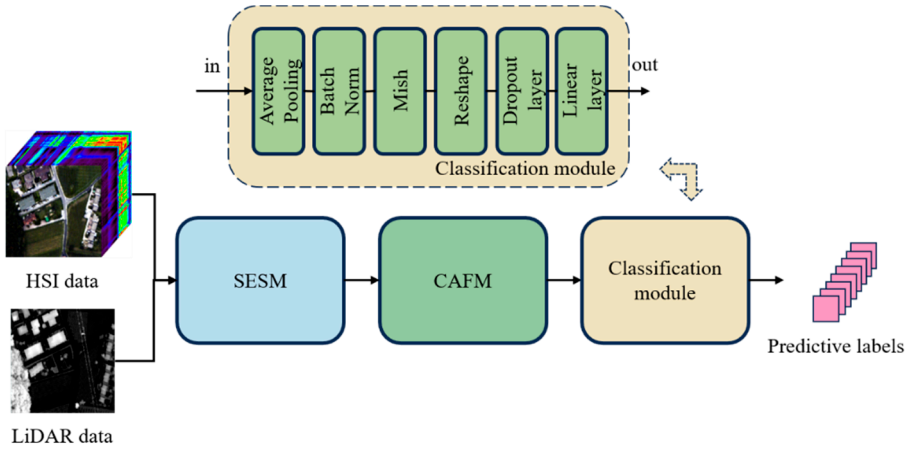


Figure 2. Structure of the proposed method.

3.3. Spatial–Elevation–Spectral Convolutional Feature Extraction Module

A triple-branch CNN architecture ($\mathcal{B}_{\mathcal{H}1}$, $\mathcal{B}_{\mathcal{H}2}$ and $\mathcal{B}_{\mathcal{L}}$) is applied to extract spatial–spectral and spatial–elevation features from HSI and LiDAR data. Specifically, $\mathcal{B}_{\mathcal{H}1}$ is used to extract the spectral features and $\mathcal{B}_{\mathcal{H}2}$ is used to extract the spatial features from HSI. $\mathcal{B}_{\mathcal{L}}$ is applied to extract the spatial–elevation features from LiDAR. To fully explore the discriminative features, multi-scale spatial and spectral pseudo-3D convolutional blocks are designed. Residual and one-shot aggregations are implemented to enhance the module. Parameter sharing is introduced to capture the correlation and complementarity between $\mathcal{B}_{\mathcal{H}2}$ and $\mathcal{B}_{\mathcal{L}}$, allowing the module to better exploit spatial and elevation features of HSI and LiDAR and reducing the number of free parameters. Figure 3 shows the detailed architecture of the SESM.

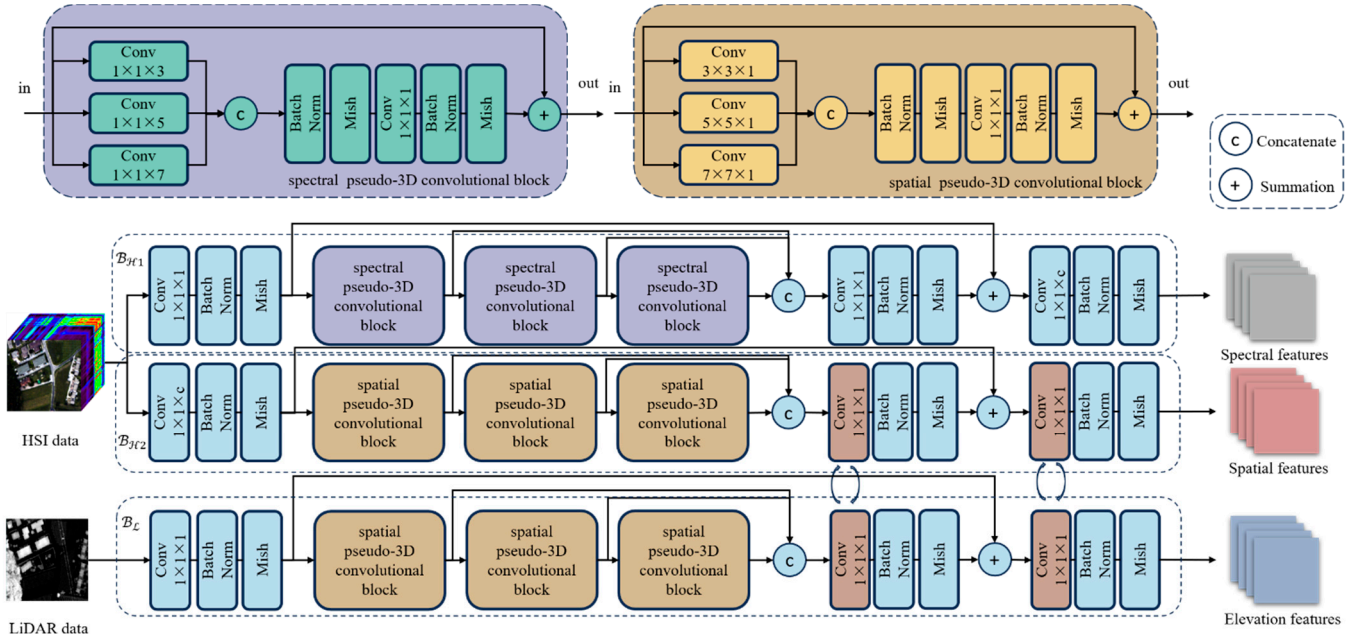


Figure 3. The architecture of the SESM.

For $\mathcal{B}_{\mathcal{H}1}$, the input is the HSI cube x^i_{HSI}. A $(1 \times 1 \times 1)$ convolutional layer is implemented to maintain the spectral size and increase the number of feature maps. The 3D convolution can be expressed as

$$\mathbf{F}_l(x,y,z) = \sum_{s_1=0}^{k_1} \sum_{s_2=0}^{k_2} \sum_{s_3=0}^{k_3} \mathbf{s}_l(s_1,s_2,s_3)\mathbf{F}_{l-1}(x+s_1, y+s_2, z+s_3) + \mathbf{b}_l \tag{4}$$

where $\mathbf{F}_l(x,y,z)$ represents the output of the lth layer at the position (x,y,z), $\mathbf{F}_{l-1}(x+s_1, y+s_2, z+s_3)$ represents the specific value of the feature maps at the $(x+s_1, y+s_2, z+s_3)$ position in $(l-1)$th layer, $\mathbf{s}_l$ represents the specific value of the convolution kernel of the lth layer at the position (s_1,s_2,s_3), k_1,k_2 and k_3 represent the size of the convolution kernel, respectively, $\mathbf{b}_l$ represents the bias of the lth layer.

Then, three spectral convolutional blocks are used to extract features. One-shot aggregation is used to collect the shallow and deep features to promote the presentation ability. The process can be expressed as

$$\mathbf{F}_l = Cat\big(H_{spe}(\mathbf{F}_{in1}), H_{spe}(\mathbf{F}_{in2}), H_{spe}(\mathbf{F}_{in3})\big) \tag{5}$$

where H_{spe} represents the spectral convolutional block, $\mathbf{F}_{in1}$, $\mathbf{F}_{in2}$ and $\mathbf{F}_{in3}$ are the corresponding input data, $Cat(\cdot)$ represents the concatenate operator.

After that, a $(1 \times 1 \times 1)$ convolutional layer is implemented to reduce the number of feature maps. The residual aggregation is used to enhance convergence. The residual aggregation can be expressed as

$$\mathbf{F}_l = H(\mathbf{F}_s) \oplus \mathbf{F}_s \tag{6}$$

where $\mathbf{F}_s$ represents the shallow feature map, $H(\cdot)$ represents the hidden intermediate layer. Finally, a $(1 \times 1 \times c)$ convolutional layer is used to reduce the spectral size. The dataflow of the $\mathcal{B}_{\mathcal{H}1}$ is shown in Table 2. All these layers are followed by the batch normalization [62] (Batch Norm) and Mish activation function [63] (Mish) to avoid overfitting and provide nonlinear capabilities for the module. The Batch Norm formula is

$$BN(\mathbf{x}) = \frac{\mathbf{x} - E(\mathbf{x})}{Var(\mathbf{x})} \tag{7}$$

where $BN(\cdot)$ is the batch normalization function, $E(\cdot)$ and $Var(\cdot)$ represent the mean and variance function in element dimension. The Mish activation function can be expressed as

$$M(\mathbf{x}) = tanh(\ln(1 + e^{\mathbf{x}}))\mathbf{x} \tag{8}$$

where $M(\cdot)$ represents the Mish function. Assume the following formula

$$f^{k_1 \times k_2 \times k_3}(\mathbf{x}) = M\Big(BN\Big(Conv^{k_1 \times k_2 \times k_3}(\mathbf{x})\Big)\Big) \tag{9}$$

where $f^{k_1 \times k_2 \times k_3}(\cdot)$ is a conforming function, $Conv^{k_1 \times k_2 \times k_3}(\cdot)$ represents the 3D convolutional layer, $(k_1 \times k_2 \times k_3)$ represents the kernel size. The whole process of $\mathcal{B}_{\mathcal{H}1}$ can be expressed as

$$\mathbf{F}_1^{\mathcal{H}1} = f^{1\times1\times1}\Big(\mathbf{x}_{HSI}^i\Big) \tag{10}$$

$$\mathbf{F}_{spe1}^{\mathcal{H}1} = H_{spe}\Big(\mathbf{F}_1^{\mathcal{H}1}\Big),\ \mathbf{F}_{spe2}^{\mathcal{H}1} = H_{spe}\Big(\mathbf{F}_{spe1}^{\mathcal{H}1}\Big),\ \mathbf{F}_{spe3}^{\mathcal{H}1} = H_{spe}\Big(\mathbf{F}_{spe2}^{\mathcal{H}1}\Big) \tag{11}$$

$$\mathbf{F}_{\mathcal{H}1} = f^{1\times1\times c}\Big(f^{1\times1\times1}\Big(Cat\Big(\mathbf{F}_{spe1}^{\mathcal{H}1}, \mathbf{F}_{spe2}^{\mathcal{H}1}, \mathbf{F}_{spe3}^{\mathcal{H}1}\Big)\Big) \oplus \mathbf{F}_1^{\mathcal{H}1}\Big) \tag{12}$$

where $\mathbf{F}_{\mathcal{H}1}$ represents the output of $\mathcal{B}_{\mathcal{H}1}$.

Table 2. The dataflow of the $\mathcal{B}_{\mathcal{H}1}$.

Input Size	Layer Name	Kernel	Stride	Padding	Filters	Output Size
$(1, h, w, c)$	Conv/BN/Mish	$(1, 1, 1)$	$(1, 1, 1)$	$(0, 0, 0)$	f	(f, h, w, c)
(f, h, w, c)	Spectral conv block	-	-	-	-	(f, h, w, c)
(f, h, w, c)	Spectral conv block	-	-	-	-	(f, h, w, c)
(f, h, w, c)	Spectral conv block	-	-	-	-	(f, h, w, c)
-	Concatenate	-	-	-	-	$(3f, h, w, c)$
$(3f, h, w, c)$	Conv/BN/Mish	$(1, 1, 1)$	$(1, 1, 1)$	$(0, 0, 0)$	f	(f, h, w, c)
-	Summation	-	-	-	-	(f, h, w, c)
(f, h, w, c)	Conv/BN/Mish	$(1, 1, c)$	$(1, 1, 1)$	$(0, 0, 0)$	f	$(f, h, w, 1)$

For $\mathcal{B}_{\mathcal{H}2}$ and $\mathcal{B}_{\mathcal{L}}$, the structure of the network is almost identical. The only difference is that $\mathcal{B}_{\mathcal{H}2}$ uses $(1 \times 1 \times c)$ convolutional layer to reduce the spectral size and increase the number of feature maps, and $\mathcal{B}_{\mathcal{L}}$ use $(1 \times 1 \times 1)$ convolutional layer to achieve the same function. After that, three spatial convolutional blocks are used to extract spatial and elevation features from HSI and LiDAR data. Two $(1 \times 1 \times 1)$ convolutional layers are used to encode the features and exploit the correlation and complementarity by parameter sharing, which can enhance the generalizability and robustness of the neural network. Residual and one-shot aggregation are used to enhance the convergence. Batch Norm and Mish are applied after these layers. The process of $\mathcal{B}_{\mathcal{H}2}$ is expressed as follows.

$$\mathbf{F}_1^{\mathcal{H}2} = f^{1\times1\times c}\Big(\mathbf{x}_{HSI}^i\Big) \tag{13}$$

$$\mathbf{F}_{spa1}^{\mathcal{H}2} = H_{spa}\Big(\mathbf{F}_1^{\mathcal{H}2}\Big),\ F_{spa2}^{\mathcal{H}2} = H_{spa}\Big(\mathbf{F}_{spa1}^{\mathcal{H}2}\Big),\ F_{spa3}^{\mathcal{H}2} = H_{spa}\Big(\mathbf{F}_{spa2}^{\mathcal{H}2}\Big) \tag{14}$$

$$\mathbf{F}_{\mathcal{H}2} = f^{1\times1\times1}\Big(f^{1\times1\times1}\Big(Cat\Big(\mathbf{F}_{spa1}^{\mathcal{H}2}, \mathbf{F}_{spa2}^{\mathcal{H}2}, \mathbf{F}_{spa3}^{\mathcal{H}2}\Big)\Big) \oplus \mathbf{F}_1^{\mathcal{H}2}\Big) \tag{15}$$

where $\mathbf{F}_{\mathcal{H}2}$ represents the output of $\mathcal{B}_{\mathcal{H}2}$. Similarly, the $\mathcal{B}_{\mathcal{L}}$ can be expressed as

$$\mathbf{F}_1^{\mathcal{L}} = f^{1\times1\times1}\Big(\mathbf{x}_{LiDAR}^i\Big) \tag{16}$$

$$\mathbf{F}_{spa1}^{\mathcal{L}} = H_{spa}\Big(\mathbf{F}_1^{\mathcal{L}}\Big),\ F_{spa2}^{\mathcal{L}} = H_{spa}\Big(\mathbf{F}_{spa1}^{\mathcal{L}}\Big),\ F_{spa3}^{\mathcal{L}} = H_{spa}\Big(\mathbf{F}_{spa2}^{\mathcal{L}}\Big) \tag{17}$$

$$\mathbf{F}_{\mathcal{L}} = f^{1\times1\times1}\left(f^{1\times1\times1}\left(Cat\left(\mathbf{F}^{\mathcal{L}}_{spa1}, \mathbf{F}^{\mathcal{L}}_{spa2}, \mathbf{F}^{\mathcal{L}}_{spa3}\right)\right) \oplus \mathbf{F}^{\mathcal{L}}_{1}\right) \tag{18}$$

where $\mathbf{F}_{\mathcal{L}}$ is the output of $\mathcal{B}_{\mathcal{L}}$. The parameter sharing occurs in the last two $f^{1\times1\times1}(\cdot)$. The dataflow of the $\mathcal{B}_{\mathcal{H}2}$ and $\mathcal{B}_{\mathcal{L}}$ is shown in Table 3.

Table 3. The dataflow of the $\mathcal{B}_{\mathcal{H}2}$ and $\mathcal{B}_{\mathcal{L}}$.

	Input Size	Layer Name	Kernel	Stride	Padding	Filters	Output Size
$\mathcal{B}_{\mathcal{H}2}$	$(1,h,w,c)$	Conv/BN/Mish	$(1,1,c)$	$(1,1,1)$	$(0,0,0)$	f	$(f,h,w,1)$
$\mathcal{B}_{\mathcal{L}}$	$(1,h,w,1)$	Conv/BN/Mish	$(1,1,1)$	$(1,1,1)$	$(0,0,0)$	f	$(f,h,w,1)$
$\mathcal{B}_{\mathcal{H}2}, \mathcal{B}_{\mathcal{L}}$	$(f,h,w,1)$	Spatial conv block	-	-	-	-	$(f,h,w,1)$
$\mathcal{B}_{\mathcal{H}2}, \mathcal{B}_{\mathcal{L}}$	$(f,h,w,1)$	Spatial conv block	-	-	-	-	$(f,h,w,1)$
$\mathcal{B}_{\mathcal{H}2}, \mathcal{B}_{\mathcal{L}}$	$(f,h,w,1)$	Spatial conv block	-	-	-	-	$(f,h,w,1)$
$\mathcal{B}_{\mathcal{H}2}, \mathcal{B}_{\mathcal{L}}$	-	Concatenate	-	-	-	-	$(3f,h,w,1)$
$\mathcal{B}_{\mathcal{H}2}, \mathcal{B}_{\mathcal{L}}$	$(3f,h,w,1)$	Conv/BN/Mish	$(1,1,1)$	$(1,1,1)$	$(0,0,0)$	f	$(f,h,w,1)$
$\mathcal{B}_{\mathcal{H}2}, \mathcal{B}_{\mathcal{L}}$	-	Summation	-	-	-	-	$(f,h,w,1)$
$\mathcal{B}_{\mathcal{H}2}, \mathcal{B}_{\mathcal{L}}$	$(f,h,w,1)$	Conv/BN/Mish	$(1,1,1)$	$(1,1,1)$	$(0,0,0)$	f	$(f,h,w,1)$

To capture discriminative features, multi-scale spectral and spatial pseudo-3D convolutional blocks are designed. The pseudo-3D convolutions and pointwise convolutions are jointly utilized to extract multi-scale features, which can adequately exploit the ground characteristics in complex environments. Furthermore, residual aggregation is applied to make the block easier to train. The two blocks present a similar structure. The difference is that the spectral convolutional block uses multi-scale spectral pseudo-3D convolutions to extract spectral features, and the spatial convolutional block applies multi-scale spatial pseudo-3D convolutions to focus on spatial features. Specifically, for the spectral convolutional block, three convolutional layers with $(1\times1\times3)$, $(1\times1\times5)$ and $(1\times1\times7)$ kernels are used to generate multi-scale feature maps. The number of data cubes is decreased to half the original number to reduce complexity. After that, the concatenation operator is applied to converge the features. A $(1\times1\times1)$ convolutional layer is implemented to reduce the number of data cubes, and the residual aggregation is conducted to maintain the original features and make the network easier to converge. Batch Norm and Mish are used after the convolutional layers to provide stability and nonlinearity for the block. The process can be expressed as

$$H_{spe}(\mathbf{x}) = \mathbf{x} \oplus \left(f^{1\times1\times1}(M\left(BN\left(Cat\left(Conv^{1\times1\times3}(\mathbf{x}), Conv^{1\times1\times5}(\mathbf{x}), Conv^{1\times1\times7}(\mathbf{x})\right)\right)\right)\right) \tag{19}$$

For the spatial convolutional block, the kernels of the pseudo-3D convolutions are $(3\times3\times1)$, $(5\times5\times1)$, and $(7\times7\times1)$. The following stages are the same as the spectral block. The spatial convolutional block can be expressed as

$$H_{spa}(\mathbf{x}) = \mathbf{x} \oplus \left(f^{1\times1\times1}(M\left(BN\left(Cat\left(Conv^{3\times3\times1}(\mathbf{x}), Conv^{5\times5\times1}(\mathbf{x}), Conv^{7\times7\times1}(\mathbf{x})\right)\right)\right)\right) \tag{20}$$

The dataflows of the spectral and spatial convolutional blocks are shown in Tables 4 and 5.

Table 4. The dataflow of the multi-scale spectral pseudo-3D convolutional block.

Input Size	Layer Name	Kernel	Stride	Padding	Filters	Output Size
(f,h,w,c)	Conv	$(1,1,3)$	$(1,1,1)$	$(0,0,1)$	$f/2$	$(f/2,h,w,c)$
(f,h,w,c)	Conv	$(1,1,5)$	$(1,1,1)$	$(0,0,2)$	$f/2$	$(f/2,h,w,c)$
(f,h,w,c)	Conv	$(1,1,7)$	$(1,1,1)$	$(0,0,3)$	$f/2$	$(f/2,h,w,c)$
-	Concatenate	-	-	-	-	$(3f/2,h,w,c)$
$(3f/2,h,w,c)$	BN/Mish	-	-	-	-	$(3f/2,h,w,c)$
$(3f/2,h,w,c)$	Conv/BN/Mish	$(1,1,1)$	$(1,1,1)$	$(0,0,0)$	f	(f,h,w,c)
-	Summation	-	-	-	-	(f,h,w,c)

Table 5. The dataflow of the multi-scale spatial pseudo-3D convolutional block.

Input Size	Layer Name	Kernel	Stride	Padding	Filters	Output Size
$(f,h,w,1)$	Conv	$(3,3,1)$	$(1,1,1)$	$(1,1,0)$	$f/2$	$(f/2,h,w,1)$
$(f,h,w,1)$	Conv	$(5,5,1)$	$(1,1,1)$	$(2,2,0)$	$f/2$	$(f/2,h,w,1)$
$(f,h,w,1)$	Conv	$(7,7,1)$	$(1,1,1)$	$(3,3,0)$	$f/2$	$(f/2,h,w,1)$
-	Concatenate	-	-	-	-	$(3f/2,h,w,1)$
$(3f/2,h,w,1)$	BN/Mish	-	-	-	-	$(3f/2,h,w,1)$
$(3f/2,h,w,1)$	Conv/BN/Mish	$(1,1,1)$	$(1,1,1)$	$(0,0,0)$	f	$(f,h,w,1)$
-	Summation	-	-	-	-	$(f,h,w,1)$

3.4. Cross Attention Fusion Module

To capture the long-distance dependencies and represent the relationships of the features, a local–global cross attention fusion module is designed to integrate the multi-source features and generate relational semantic representations to achieve data fusion. Different from the existing cross attention techniques, which commonly use maximization or average pooling to capture the global linear spectral and spatial dependencies between feature maps and ignore the local spatial cues in the feature maps, the proposed CAFM module uses a combination of convolutional layers and attention operations to achieve feature representation. Local-based convolutions are used to collect contextual cues of feature maps. A global-based cross attention mechanism is applied to exploit long-distance dependences. Residual aggregations are implemented to enhance convergence. The overall structure of the CAFM is shown in Figure 4.

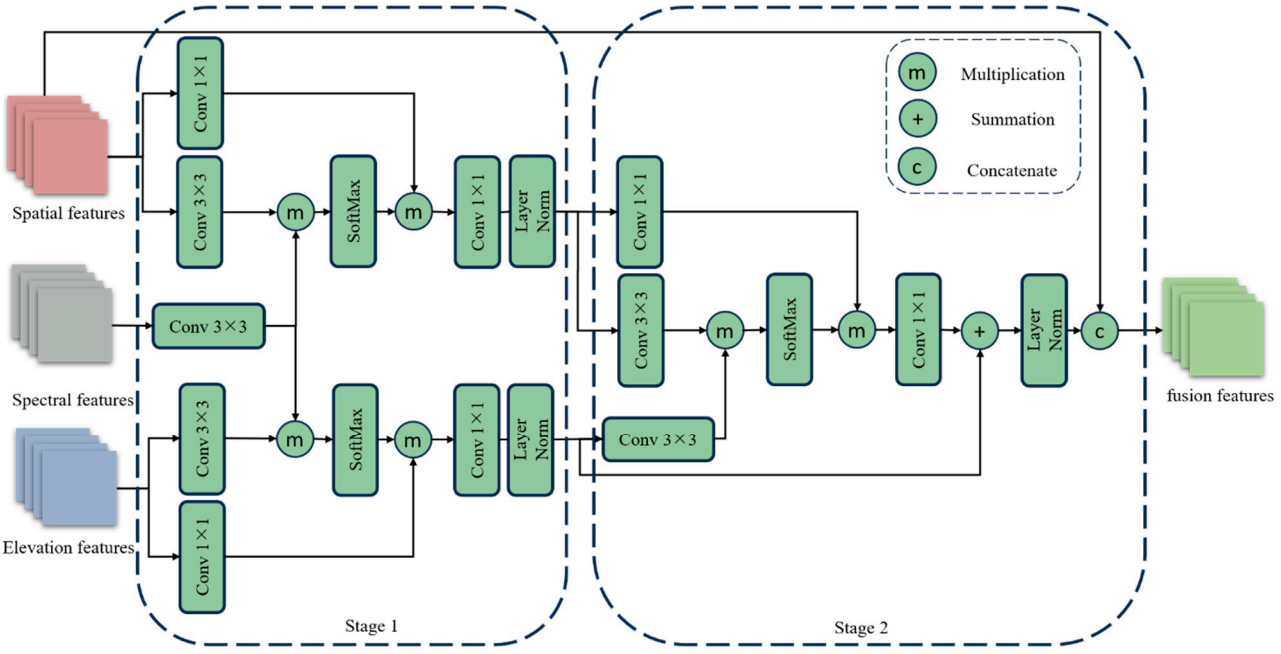


Figure 4. Structure of the CAFM.

In the attention mechanism, **Q** is the target of the query, **K** is the key feature representation, and **V** is the value of the key feature representation. First, the attention scores are obtained by the correlation between **Q** and **K**, which can be used to express the dependency of **Q** and **K**. Then the output is produced by weighted summation of **V** according to the attention scores. In the calculation process of the attention mechanism, **Q** is the subject of the query and is the core information that generates the attention scores. For the joint HSI and LiDAR datasets, we consider that the spectral signatures provided by HSI are the core information that provides more discriminative power than spatial and elevation features, which should be given a higher priority. Therefore, in the first stage of the CAFM module,

spectral features are chosen as **Q** instead of spatial features or elevation information. With this design, the spectral signatures are fully utilized. And the correlation expressions of the spectral signatures with spatial features and elevation information are obtained, respectively. Based on the same consideration, in the second stage of the CAFM module, the correlation expression of the spectral signatures and elevation information is employed as **Q** to obtain a joint expression of the spectral signatures, spatial features, and elevation information, which is treated as the final fusion features.

Specifically, the input's spatial features ($\mathbf{F}_{H2}$), spectral features ($\mathbf{F}_{H1}$), and elevation features ($\mathbf{F}_L$) are extracted by SESM. The shapes of the three features are (h, w, f), which are obtained by removing the spectral dimensions through the reshape operator. The whole process consists of two stages. For stage 1, spectral features are fused with spatial features and elevation features by cross attention mechanism, respectively; (3×3) convolutions are used to capture local contextual information of feature maps; (1×1) convolutions are applied to replace the traditional FFN to achieve linear transformation, which can effectively reduce the computational complexity. The initiation process is expressed as follows.

$$\mathbf{V}_H = Re\Big(Conv^{1\times1}(\mathbf{F}_{H2})\Big), \mathbf{K}_H = Re\Big(Conv^{3\times3}(\mathbf{F}_{H2})\Big), \mathbf{Q}_H = Re\Big(Conv^{3\times3}(\mathbf{F}_{H1})\Big) \tag{21}$$

$$\mathbf{V}_L = Re\Big(Conv^{1\times1}(\mathbf{F}_L)\Big), \mathbf{K}_L = Re\Big(Conv^{3\times3}(\mathbf{F}_L)\Big) \tag{22}$$

where $\mathbf{V}_H, \mathbf{K}_H, \mathbf{Q}_H, \mathbf{V}_L, \mathbf{K}_L$ represent the queries, keys and values, $Re(\cdot)$ represents the reshape operator, which are used to change the shape of the input data, $Conv^{k_1\times k_2}(\cdot)$ represents the 2D convolutional layer, $(k_1 \times k_2)$ represents the kernel size.

Layer normalization (Layer Norm) [64] are applied to stabilize the module, which is commonly observed in the attention mechanism. The Layer Norm formula is

$$LN(\mathbf{x}) = \frac{\mathbf{x} - E(\mathbf{x})}{Var(\mathbf{x})} \tag{23}$$

where $LN(\cdot)$ is the Layer normalization function, $E(\cdot)$ and $Var(\cdot)$ represent the mean and variance function in feature dimension.

To enhance flexibility, the multi-head approach is applied to provide cross attention weights, which can be expressed as

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{24}$$

$$\mathbf{head}_i = Attention\Big(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V\Big) \tag{25}$$

$$MultiHead(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = Cat(\mathbf{head}_1, \ldots, \mathbf{head}_h)\mathbf{W}^O \tag{26}$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ represent the queries, keys, and values, h represents the number of heads, $\mathbf{head}_i$ represents the output of ith head, $\mathbf{W}^O$ represents the output transform matrix, $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V$ represent the transform matrix of queries, keys, and values in ith head, $Attention(\cdot)$ is the attention function, d_k is the feature size of the key vector, $softmax(\cdot)$ represents the similarity normalization function. The process of stage 1 can be expressed as

$$\mathbf{F}_{HH} = LN\Big(Conv^{1\times1}(Re(MultiHead(\mathbf{Q}_H, \mathbf{K}_H, \mathbf{V}_H)))\Big) \tag{27}$$

$$\mathbf{F}_{HL} = LN\Big(Conv^{1\times1}(Re(MultiHead(\mathbf{Q}_H, \mathbf{K}_L, \mathbf{V}_L)))\Big) \tag{28}$$

where $\mathbf{F}_{HH}$ and $\mathbf{F}_{HL}$ represent the output of stage 1.

Furthermore, feature compression is used for local feature extraction to condense valuable information and reduce the computation. The dataflow of stage 1 is shown in Table 6.

Table 6. The dataflow of stage 1 of CAFM.

	Input Size	Layer Name	Kernel	Stride	Padding	Filters	Output Size
$\mathbf{F}_{H2}$	(f,h,w)	Conv/Reshape	$(1,1)$	$(1,1)$	$(0,0)$	f	$(h\times w,f)$
$\mathbf{F}_{H2}$	(f,h,w)	Conv/Reshape	$(3,3)$	$(1,1)$	$(1,1)$	f/r	$(h\times w,f/r)$
$\mathbf{F}_{H1}$	(f,h,w)	Conv/Reshape	$(3,3)$	$(1,1)$	$(1,1)$	f/r	$(f/r,h\times w)$
$\mathbf{F}_{L}$	(f,h,w)	Conv/Reshape	$(3,3)$	$(1,1)$	$(1,1)$	f/r	$(h\times w,f/r)$
$\mathbf{F}_{L}$	(f,h,w)	Conv/Reshape	$(1,1)$	$(1,1)$	$(0,0)$	f	$(h\times w,f)$
	-	Multiple/SoftMax	-	-	-	-	$(h\times w,h\times w)$
	-	Multiple/SoftMax	-	-	-	-	$(h\times w,h\times w)$
	-	Multiple	-	-	-	-	$(h\times w,f)$
	-	Multiple	-	-	-	-	$(h\times w,f)$
	$(h\times w,f)$	Reshape/Conv/LN	$(1,1)$	$(1,1)$	$(0,0)$	f	(f,h,w)
	$(h\times w,f)$	Reshape/Conv/LN	$(1,1)$	$(1,1)$	$(0,0)$	f	(f,h,w)

Stage 2 is designed to further interact with the fused features and exploit the relationship between multi-source data to generate joint deep semantic representations. To achieve the objective, a further cross attention approach is performed. Different from the process of stage 1, residual aggregations are implemented in stage 2 to maintain shallow features and provide convergence. The initialization steps can be expressed as follows.

$$\mathbf{V}_{HH}=Re\Big(Conv^{1\times1}(\mathbf{F}_{HH})\Big),\mathbf{K}_{HH}=Re\Big(Conv^{3\times3}(\mathbf{F}_{HH})\Big),\mathbf{Q}_{HL}=Re\Big(Conv^{3\times3}(\mathbf{F}_{HL})\Big) \tag{29}$$

where $\mathbf{V}_{HH},\mathbf{K}_{HH},\mathbf{Q}_{HL}$ represent the values, keys, and queries. The process of stage 2 can be expressed as follows.

$$\mathbf{F}_{HHL}=Cat\Big(\mathbf{F}_{H2},LN\Big(\mathbf{F}_{HL}\oplus Conv^{1\times1}(Re(MultiHead(\mathbf{Q}_H,\mathbf{K}_H,\mathbf{V}_H)))\Big)\Big) \tag{30}$$

where $\mathbf{F}_{HHL}$ represents the output of stage 2. The dataflow of stage 2 is shown in Table 7.

Table 7. The dataflow of stage 2 of CAFM.

Input Size	Layer Name	Kernel	Stride	Padding	Filters	Output Size
(f,h,w)	Conv/Reshape	$(1,1)$	$(1,1)$	$(0,0)$	f	$(h\times w,f)$
(f,h,w)	Conv/Reshape	$(3,3)$	$(1,1)$	$(1,1)$	f/r	$(h\times w,f/r)$
(f,h,w)	Conv/Reshape	$(3,3)$	$(1,1)$	$(1,1)$	f/r	$(f/r,h\times w)$
-	Multiple/SoftMax	-	-	-	-	$(h\times w,h\times w)$
-	Multiple	-	-	-	-	$(h\times w,f)$
$(p\times p,f)$	Reshape/Conv	$(1,1)$	$(1,1)$	$(0,0)$	f	(f,h,w)
-	Summation/LN	-	-	-	-	(f,h,w)
-	Concatenate	-	-	-	-	$(2f,h,w)$

3.5. The Classification Module

The classification module is used to process the extracted semantic features to generate the final classification result. The structure of the classification module is shown in Figure 2. The classification module consists of an average pooling layer, a Batch Norm layer, a Mish activation function, a reshape layer, a dropout layer, and a linear layer. Specifically, the average pooling layer is used to further integrate the deep semantic features into classification results. The Batch Norm layer and Mish layer are applied to normalize features and provide nonlinear mapping. The reshape layer is used to eliminate redundant dimensions. The dropout layer is implemented to enhance the generalization of the module. The linear layer is used to generate the predictive labels. The process of the classification module can be expressed as follows.

$$\mathbf{y}_{pre}^{i}=Linear(Dropout(Re(M(BN(Avgpool(\mathbf{F}_{HHL})))))) \tag{31}$$

where $\mathbf{y}_{pre}^{i}$ represents the predictive label of ith input data, $Linear(\cdot)$ represents the linear layer, $Dropout(\cdot)$ represents the dropout layer, $Avgpool(\cdot)$ represents the average pooling layer. The dataflow of the classification module is shown in Table 8.

Table 8. The dataflow of the classification module.

Input Size	Layer Name	Kernel	Stride	Padding	Filters	Output Size
$(2f, h, w)$	Avg pooling	-	-	-	-	$(2f, 1, 1)$
$(2f, 1, 1)$	BN/Mish	-	-	-	-	$(2f, 1, 1)$
$(2f, 1, 1)$	Reshape	-	-	-	-	$(2f)$
$(2f)$	Dropout	-	-	-	-	$(2f)$
$(2f)$	Linear	-	-	-	c	(c)

4. Experiment

4.1. Dataset Description

In the experiment, three HSI-LiDAR pair datasets with different land covers are used to evaluate the effectiveness of the proposed network, which are the Trento dataset, the MUUFL dataset, and the Houston2013 dataset. The brief views of the datasets are described as follows.

Trento Dataset [4]: The Trento dataset is an HSI-LiDAR pair dataset, where the HSI data were captured by an AISA Eagle sensor, and the LiDAR digital surface model (DSM) data were acquired by an Optech ALTM 3100EA sensor. The dataset is captured over a rural area south of the city of Trento, Italy. The spatial size of the Trento dataset is 166×600 and the spatial resolution is about 1 m. The HSI data contains 63 bands with a spectral wavelength ranging from 420 to 990 nm. The LiDAR DSM data can reflect the height of ground objects. The land covers are classified into six categories, including Apple trees, Buildings, Ground, Woods, Vineyard, and Roads. The pseudo-color image for the HSI data, the LiDAR DSM image, and the ground-truth map of the Trento dataset are shown in Figure 5. The classes, colors, and the number of samples for each class are exhaustively provided in Table 9.

MUUFL Dataset [65]: The MUUFL dataset was collected by the ITERS CASI-1500 sensor in November 2010 at the University of Southern Mississippi Gulf Park campus in Long Beach, Mississippi, which contains the HSI dataset and LiDAR dataset. The spatial size is 325×220, and the spatial resolution is 0.54×1.0 m. The HSI contains 64 available bands in the range of 375 to 1050 nm. The LiDAR data can reflect the height of the ground objects. The land covers are classified into 11 categories, including Trees, Mostly grass, Mixed ground surface, Dirt and sand, Road, Water, Building Shadow, Building, Sidewalk, Yellow curb, and Cloth panels. The pseudo-color image for the HSI data, the LiDAR DSM image, and the ground-truth map are shown in Figure 6. The classes, colors, and the number of samples for each class are provided in Table 9.

Houston2013 Dataset [66,67]: The Houston2013 dataset was acquired by the ITERS CASI-1500 sensor over the University of Houston campus, Houston, Texas, USA, and the neighboring urban area in 2012. It is composed of HSI data and LiDAR DSM data. The spatial size is 349×1905, and the spatial resolution is about 2.5 m. The HSI data contain 144 spectral bands in the 380 to 1050 nm region. The LiDAR data can reflect the height of the ground objects. The land covers are classified into 15 categories, including Healthy grass, Stressed grass, Synthetic grass, Trees, Soil, Water, Residential, Commercial, Road, Highway, Railway, Parking Lot 1, Parking Lot 2, Tennis Court, and Running track. The pseudo-color image for the HSI data, the LiDAR DSM image, and the ground-truth map of the Houston2013 dataset are shown in Figure 7. The classes, colors, and the number of samples for each class are provided in Table 9.

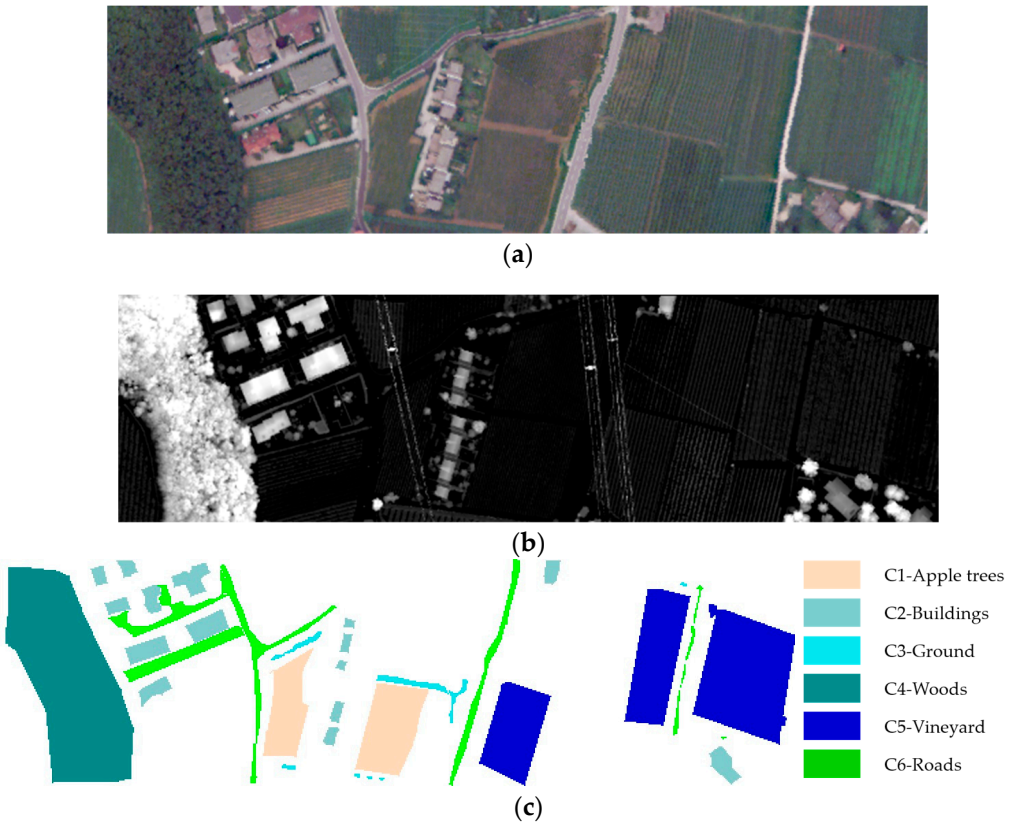


Figure 5. Trento dataset: (**a**) Pseudo-color image (31,14,2 bands); (**b**) LiDAR-derived DSM image; (**c**) Ground-truth map.

Table 9. Classes, colors, and number of samples of the Trento, MUUFL, and Houston2013 datasets.

Class	**Total**	**Train**	**Val**	**Test**	**Class**	**Total**	**Train**	**Val**	**Test**
C1-Apple trees	4034	41	41	3952	C5-Vineyard	10,501	106	106	10,289
C2-Buildings	2903	30	30	2843	C6-Roads	3174	32	32	3110
C3-Ground	479	5	5	469	Total	30,214	306	306	29,602
C4-Woods	9123	92	92	8939					
Class	Total	Train	Val	Test	Class	Total	Train	Val	Test
C1-Trees	23,246	233	233	22,780	C7-Building Shadow	2233	23	23	2187
C2-Mostly grass	4270	43	43	4184	C8-Building	6240	63	63	6114
C3-Mixed ground surface	6882	69	69	6744	C9-Sidewalk	1385	14	14	1357
C4-Dirt and sand	1826	19	19	1788	C10-Yellow curb	183	2	2	179
C5-Road	6687	67	67	6553	C11-Cloth panels	269	3	3	263
C6-Water	466	5	5	456	Total	53,687	541	541	52,605
Class	Total	Train	Val	Test	Class	Total	Train	Val	Test
C1-Healthy grass	1251	13	13	1225	C9-Road	1252	13	13	1226
C2-Stressed grass	1254	13	13	1228	C10-Highway	1227	13	13	1201
C3-Synthetic grass	697	7	7	683	C11-Railway	1235	13	13	1209
C4-Trees	1244	13	13	1218	C12-Parking Lot 1	1233	13	13	1207
C5-Soil	1242	13	13	1216	C13-Parking Lot 2	469	5	5	459
C6-Water	325	4	4	317	C14-Tennis Court	428	5	5	418
C7-Residential	1268	13	13	1242	C15-Running track	660	7	7	646
C8-Commercial	1244	13	13	1218	Total	15,029	158	158	14,713

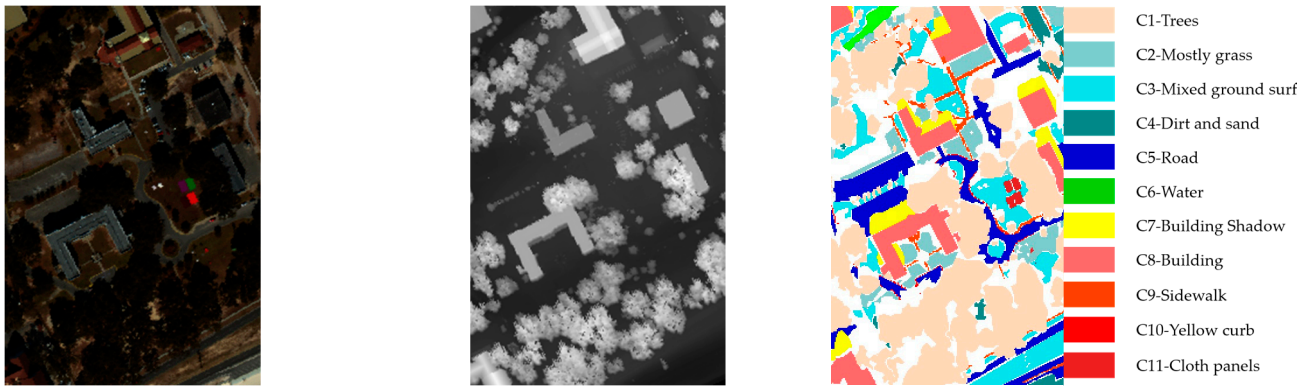


Figure 6. MUUFL dataset: (**a**) Pseudo-color image (31,16,6 bands); (**b**) LiDAR-derived DSM image; (**c**) Ground-truth map.

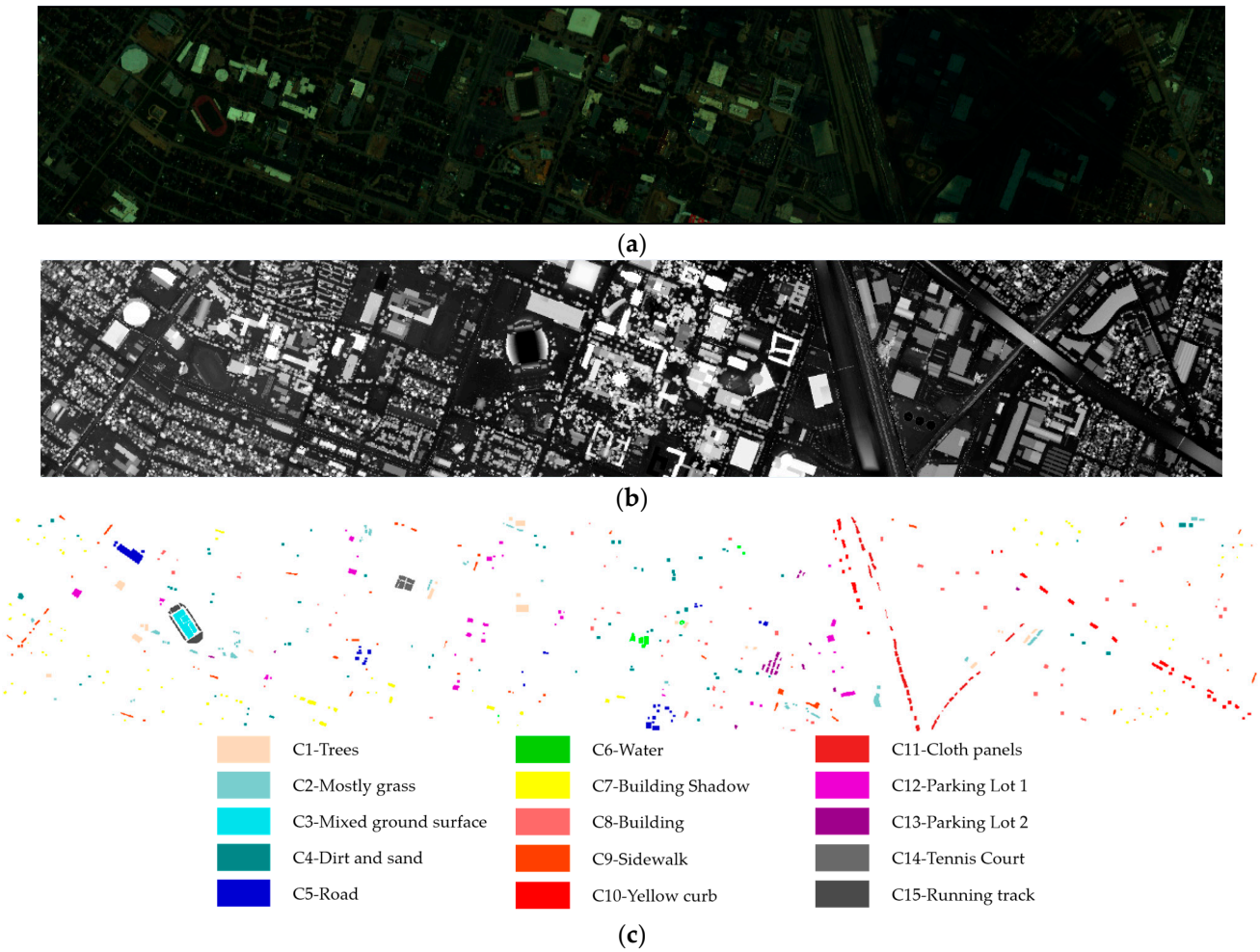


Figure 7. Houston2013 dataset: (**a**) Pseudo-color image (69,36,12 bands); (**b**) LiDAR-derived DSM image; (**c**) Ground-truth map.

4.2. Experimental Setup and Assessment Indices

To evaluate the performance of the proposed network on the multi-source remote sensing datasets, three HSI-LiDAR pair datasets with different land covers and spatial resolutions are introduced to our experiment. Ten representative methods are collected for comparison, including SVM [68], HYSN [45], DBDA [53], PMCN [69], FusAtNet [34],

CCNN [22], AM^{3}net [27], HCTnet [33], Sal^{2}RN [65], and MS2CAN [70]. Among these methods, SVM is adopted to represent the classical machine learning HSI classification methods based on spectral signatures; HYSN, DBDA, and PMCN are used to represent the classical deep learning HSI classification methods based on spatial–spectral information; and FusAtNet, CCNN, AM^{3}net, HCTnet, Sal^{2}RN, MS2CAN are employed to represent the state-of-the-art deep learning HSI-LiDAR classification methods based on spatial–elevation–spectral information. The details of the comparisons are described as follows:

1. SVM: this method finds the optimal hyperplane, which is determined by the support vectors, to achieve classification;

2. HYSN: this approach proposes a hybrid spectral convolutional neural network for HSI classification and uses spectral–spatial 2D-3D convolutions to extract features;

3. DBDA: this approach proposes a double-branch dual-attention mechanism network for HSI classification. CNNs and self-attentions are used to extract spectral–spatial features;

4. PMCN: this method uses multi-scale spectral-spatial convolutions to extract features from HSI data, and an attention mechanism is applied to enhance the performance;

5. FusAtNet: this method uses residual aggregation and attention blocks to achieve data fusion of HSI and LiDAR data;

6. CCNN: this approach proposes a coupled convolutional neural network for HSI and LiDAR classification. Multi-scale CNNs are used to capture spectral–spatial features from HSI and LiDAR, and a hierarchical fusion method is implemented to fuse the extracted features;

7. AM^{3}net: this approach uses CNNs to extract spectral–spatial–elevation features and the involution operator is specially designed for spectral features. A hierarchical mutual-guided module is proposed to fuse feature embeddings to achieve HSI-LiDAR data classification;

8. HCTnet: this method proposes a dual-branch approach to achieve HSI and LiDAR classification; 2D-3D convolutions are implemented to extract features, and a transformer network is used to fuse the features;

9. Sal^{2}RN: this approach uses CNNs to extract spectral–spatial features from HSI and LiDAR data and applies a cross attention mechanism to achieve fusion;

10. MS2CAN: this approach is a multiscale pyramid fusion framework based on spatial–spectral cross-modal attention for HSI and LiDAR classification.

SVM utilizes optimal parameters by experimental analysis. Other comparisons utilize the default parameters as specified in their original papers. The data enhancement of FusAtNet is removed in the actual experiment.

For the proposed CMCN, the patch size of the data cube is set to 7×7. The batch size is set to 32. The number of feature maps (f) is set to 24. The multiplicity of channel compression (r) and the number of heads (h) in multi-head attention, which are used in CAFM, are set to 2 and 2, respectively. The dropout rate is set to 0.5. Additional hyperparameters are listed as follows. The epoch is set to 200. The initial learning rate is set to 5.0×10^{-4}. The adaptive moment estimation (Adam) [71] optimizer is applied to train the network, where the attenuation rate is set to $(0.9, 0.999)$ and the fuzzy factor is set to 10^{-8}. The cosine annealing technology [72] is adopted with 200 epochs. The early stopping technology is used in the training process with 50 epochs; 1.00% of labeled samples are randomly selected as training samples and validation samples, respectively. The remaining labeled samples are collected as testing samples.

The overall accuracy (OA), average accuracy (AA), and Kappa coefficient (Kappa) [73] are introduced to quantitatively measure the performance of the competitors. All experiments are repeated 10 times independently, and the average values are reported as the final results. The experimental hardware environment is a workstation with Intel Xeon E5-2680v4 processor 2.4 GHz and NVIDIA GeForce RTX 2080Ti GPU. The software environment is CUDA v11.2, PyTorch 1.10, and Python 3.8.

4.3. Experimental Results

We first compare the performance of the various methods on the Trento dataset. The classification results are given in Table 10, and the full-factor classification maps are shown in Figure 8. The best classification accuracy for each category, as well as OA, AA, Kappa, and training time, are highlighted in bold in the tables. Observing the classification accuracies of each method, we can see that the Trento dataset is relatively easy to classify with sufficient training samples. SVM gives the lowest OA (85.39%), indicating that the pixel classification methods using only spectral signatures are less effective than the methods using spatial–spectral features. To be specific, the C2-Buildings and C6-Roads are hard to be classified in SVM (74.12% and 71.57%). It indicates that ground objects of the two categories are difficult to distinguish by spectral signatures. HYSN, DBDA, and PMCN provide higher OAs (95.50%, 96.55%, and 97.46%) than SVM. It indicates that the inclusion of spatial information is helpful for the improvement of classification accuracy. Viewing the classification accuracies of these spatial–spectral-based deep learning HSI classification methods in various categories, especially in C2-Buildings and C6-Roads, we can see that the accuracies increase gradually. It demonstrates that the usage of multi-scale convolution and attention mechanisms can effectively improve the generalization of the model. The HSI-LiDAR fusion classification methods (FusAtNet, CCNN, AM^{3}net, HCTnet, Sal^{2}RN, MS2CAN, and CMCN) yield higher OAs than those of HSI classification methods (SVM, HYSN, DBDA, and PMCN). It indicates that the elevation information provided by LiDAR data can provide additional discriminative features to help improve the classification accuracy. Checking the average classification accuracies of each category and the OAs of these HSI-LiDAR fusion classification methods, we can see that CCNN, MS2CAN, and CMCN obtain relatively higher classification accuracies. It further demonstrates that multi-scale feature extraction and multi-level feature fusion can effectively exploit discriminative characteristics in complex ground cover environments. In particular, the proposed method (CMCN) obtained the highest OA, which shows that the proposed method can effectively capture correlative and complementary information and fuse them to generate discriminative deep semantic features. Observing the full-factor classification maps and the ground truth map, we can see that some scattered buildings, ground, and roads are difficult to distinguish. Region I is a parcel of buildings, ground, and roads. The comparison shows that CMCN can express the detailed information of ground cover more precisely. In Region II, there are two trees on the road. It is also clearly resolved on the full-factor classification map given by CMCN. Viewing the training times of the competitors, SVM gives the best training time (4.63 s). Comparing the deep neural networks, HCTnet provides the shortest training time (10.07 s). In contrast, FusAtNet gives the longest training time (68.03 s). The training time of CMCN is 30.98 s.

Table 10. Classification OA (%), AA (%), and Kappa (×100) with standard deviation and training time (s) of the Trento dataset.

Class	SVM	HYSN	DBDA	PMCN	FusAtNet	CCNN	AM^{3}net	HCTnet	Sal^{2}RN	MS2CAN	CMCN
C1	74.22 ± 3.24	97.30 ± 0.23	98.29 ± 0.49	**98.95 ± 0.39**	98.44 ± 0.96	95.06 ± 1.73	97.06 ± 0.44	96.35 ± 2.17	95.72 ± 5.34	96.67 ± 0.34	97.67 ± 1.03
C2	74.12 ± 5.61	76.07 ± 0.20	79.97 ± 1.75	85.48 ± 3.62	89.14 ± 8.10	94.90 ± 0.56	95.85 ± 0.60	96.00 ± 0.65	**96.69 ± 0.67**	94.95 ± 0.93	96.02 ± 0.33
C3	78.25 ± 28.54	97.76 ± 0.11	**99.98 ± 0.01**	99.95 ± 0.10	95.15 ± 3.61	99.39 ± 0.79	99.80 ± 0.28	99.35 ± 0.36	98.67 ± 0.40	99.75 ± 0.24	99.90 ± 0.09
C4	95.47 ± 0.79	99.97 ± 0.01	99.82 ± 0.05	99.68 ± 0.02	99.98 ± 0.04	**99.99 ± 0.01**	99.83 ± 0.06	99.88 ± 0.01	99.93 ± 0.01	**99.99 ± 0.01**	**99.99 ± 0.01**
C5	89.57 ± 7.29	99.11 ± 0.03	99.02 ± 0.14	99.68 ± 0.11	99.36 ± 0.39	99.62 ± 0.10	99.62 ± 0.08	99.07 ± 0.14	99.24 ± 0.82	**99.84 ± 0.04**	99.55 ± 0.28
C6	71.57 ± 0.81	87.76 ± 0.33	93.68 ± 0.88	93.81 ± 3.78	**98.22 ± 1.60**	97.88 ± 0.64	97.38 ± 0.63	94.88 ± 0.96	96.73 ± 0.75	96.96 ± 0.53	97.14 ± 0.63
OA	85.39 ± 0.81	95.50 ± 0.02	96.55 ± 0.32	97.46 ± 0.13	98.06 ± 1.22	98.45 ± 0.31	98.74 ± 0.07	98.20 ± 0.28	98.39 ± 0.82	98.67 ± 0.06	**98.83 ± 0.11**
AA	80.53 ± 4.98	93.00 ± 0.03	95.13 ± 0.41	96.26 ± 0.12	96.72 ± 1.32	97.81 ± 0.32	98.26 ± 0.12	97.59 ± 0.28	97.83 ± 0.87	98.03 ± 0.07	**98.38 ± 0.08**
Kappa	80.57 ± 1.08	93.99 ± 0.03	95.39 ± 0.47	96.61 ± 0.19	97.41 ± 1.62	97.92 ± 0.41	98.31 ± 0.10	97.60 ± 0.38	97.85 ± 1.08	98.22 ± 0.08	**98.44 ± 0.14**
Time (s)	**4.63**	14.36	19.70	113.31	68.03	13.25	29.44	10.07	24.09	21.49	30.98

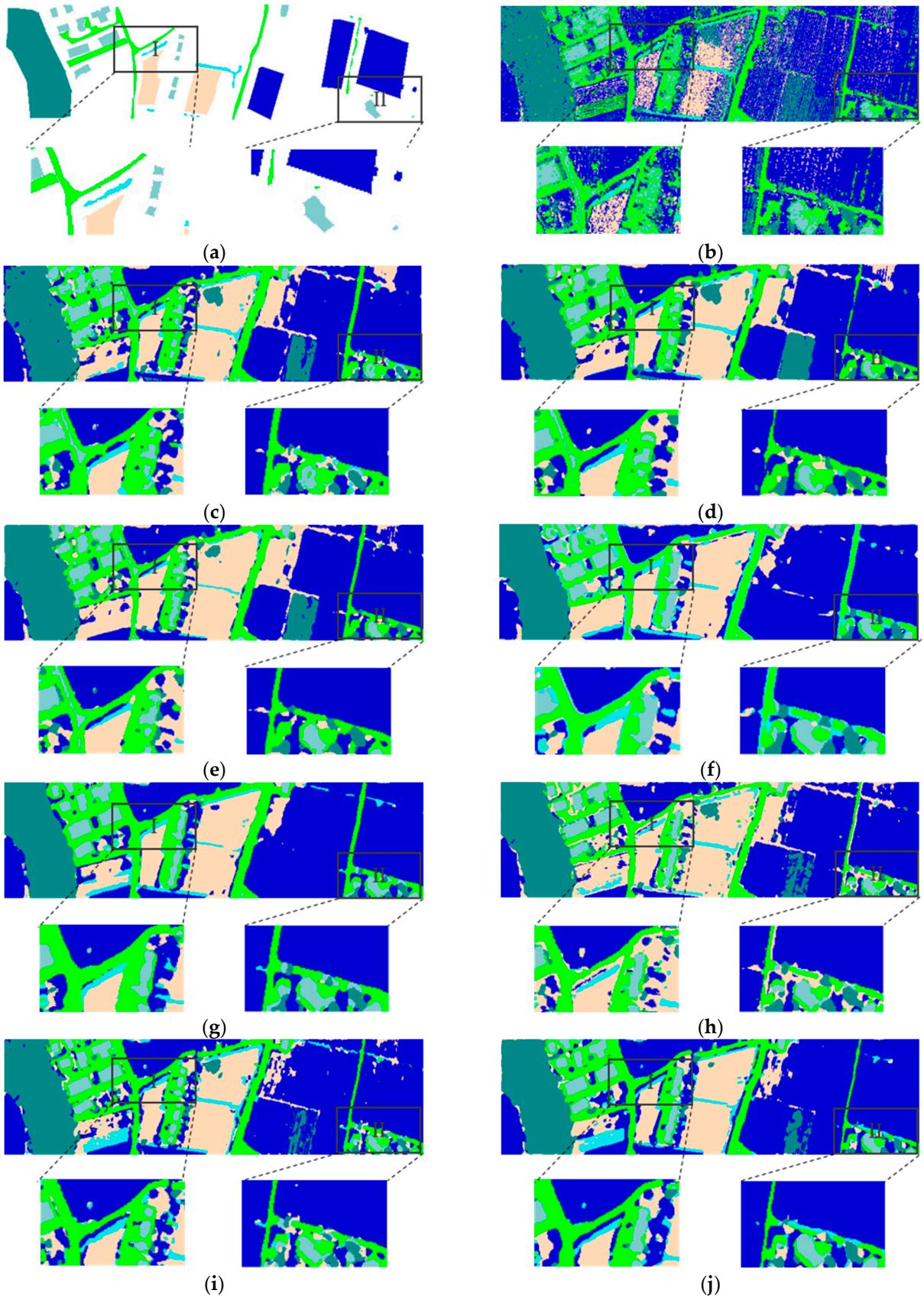
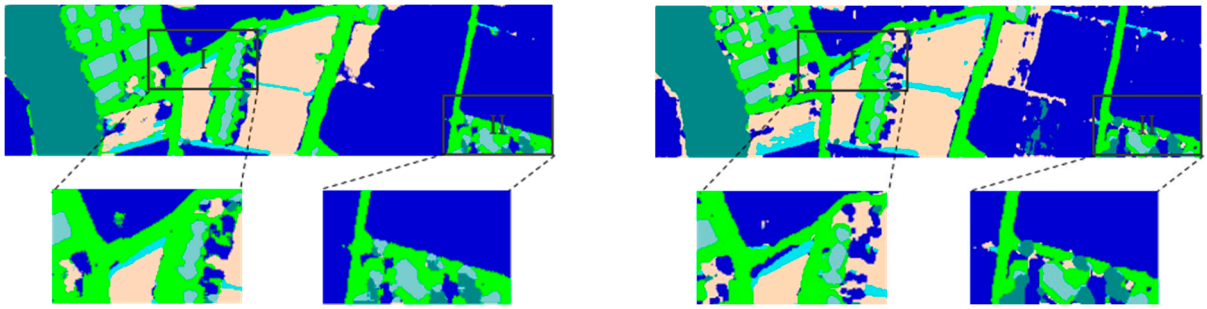


Figure 8. *Cont.*



(**k**) (**l**)

Figure 8. Full-factor classification maps for the Trento dataset: (**a**) Ground-truth map; (**b**) SVM; (**c**) HYSN; (**d**) DBDA; (**e**) PMCN; (**f**) FusAtNet; (**g**) CCNN; (**h**) AM^{3}net; (**i**) HCTnet; (**j**) Sal^{2}RN; (**k**) MS2CAN; (**l**) CMCN.

To further test the performance of the proposed method, experiments are implemented on the MUUFL dataset, which contains complex topographical landscapes and an unbalanced number of labeled samples. The classification results and full-factor classification maps are presented in Table 11 and Figure 9. Different from the experimental results on the Trento dataset, SVM obtains an OA of 83.87%, which is higher than those of HYSN and FusAtNet. For spatial–spectral-based deep learning classification methods, DBDA provides the highest OA (88.09%). The spatial–elevation–spectral-based fusion classification methods provide relatively higher classification accuracies than those of SVM, HYSN, and PMCN. MS2CAN and CMCN obtained the second-highest and the first-highest OAs (88.65% and 88.99%) for HSI-LiDAR classification. Reviewing the classification accuracies for each category, it can be seen that C7-Building Shadow, C9-Sidewalk, and C10-Yellow curb are hard to classify. It may be due to the dispersed distribution of ground cover and the similarity in elevation that leads to LiDAR data failing to provide additional valid discriminatory information. In addition, inadequate training of the deep neural network caused by an insufficient number of labeled samples for the C10-Yellow curb may also be one of the reasons for the difficulty in recognizing this category. Checking the classification accuracies of C7-Building Shadow and C9-Sidewalk, we can see that CCNN and CMCN received relatively high experimental results (83.35% and 80.91% for C7-Building Shadow; 74.94% and 75.98% for C9-Sidewalk), indicating that multi-scale feature extraction has better adaptability for remote sensing images with vast structural variations. For the C10-Yellow curb, we can see that SVM provides the highest accuracy (68.40%); meanwhile, all other methods give low classification accuracy on the C10-Yellow curb. The accuracy of the proposed method on the C10-Yellow curb is also poor (8.66%), indicating that the approach performs poorly with insufficient labeled samples and requires continuous efforts to improve it. Observing the full-factor classification maps and the ground truth map, the CMCN yields a relatively clearer map for land cover classification. In Region I and II, the building, building shadows and roads are smoothly recognized. However, we can see that the sidewalk is misclassified to be mostly grass in Region I. In Region II, the yellow curb is misclassified to be a road. SVM provides the shortest training time (11.75 s). FusAtNet gives the longest training time (225.94 s). The training time of CMCN is 66.53 s.

Table 11. Classification OA (%), AA (%), and Kappa (×100) with standard deviation and training time (s) of the MUUFL dataset.

Class	SVM	HYSN	DBDA	PMCN	FusAtNet	CCNN	AM^{3}net	HCTnet	Sal^{2}RN	MS2CAN	CMCN
C1	93.00 ± 0.96	92.24 ± 0.71	94.71 ± 0.45	92.00 ± 2.85	95.31 ± 1.20	95.75 ± 0.73	94.06 ± 0.62	94.16 ± 1.05	94.10 ± 2.24	93.59 ± 1.30	**95.77 ± 1.32**
C2	71.61 ± 3.61	66.64 ± 2.90	77.51 ± 1.13	**90.60 ± 2.92**	77.24 ± 4.73	82.83 ± 3.84	80.81 ± 9.09	78.22 ± 2.74	76.34 ± 6.31	79.66 ± 4.00	86.12 ± 5.67
C3	73.10 ± 2.47	67.25 ± 2.18	**82.90 ± 0.82**	75.35 ± 5.23	68.50 ± 4.63	73.71 ± 1.12	78.46 ± 3.81	73.16 ± 4.04	70.91 ± 8.57	80.35 ± 2.22	72.42 ± 3.34
C4	70.37 ± 4.53	76.84 ± 7.87	**89.18 ± 0.81**	82.71 ± 9.72	71.81 ± 3.43	81.06 ± 4.00	85.30 ± 2.40	79.25 ± 3.14	83.37 ± 6.82	83.06 ± 5.82	86.68 ± 5.31
C5	81.73 ± 2.12	77.04 ± 3.75	86.16 ± 0.28	83.37 ± 2.24	80.98 ± 3.34	**89.32 ± 1.64**	85.40 ± 1.10	83.06 ± 1.80	85.71 ± 2.67	86.56 ± 1.66	87.81 ± 2.21
C6	94.10 ± 7.27	51.29 ± 20.86	91.59 ± 2.95	79.12 ± 39.57	77.33 ± 11.26	75.68 ± 6.39	58.09 ± 7.67	43.28 ± 6.02	84.69 ± 8.73	75.13 ± 37.69	**92.31 ± 5.08**
C7	64.65 ± 5.36	62.67 ± 3.17	72.25 ± 0.91	75.57 ± 3.40	76.03 ± 6.03	83.35 ± 3.13	79.36 ± 3.21	79.77 ± 5.21	70.07 ± 9.79	**83.53 ± 5.88**	80.91 ± 6.91
C8	90.48 ± 2.67	74.27 ± 0.64	93.40 ± 0.57	85.74 ± 6.20	83.93 ± 5.46	87.96 ± 1.18	91.32 ± 1.91	87.40 ± 1.98	83.03 ± 5.53	95.25 ± 1.54	**96.12 ± 1.62**
C9	59.56 ± 10.58	51.39 ± 7.20	46.05 ± 0.65	48.41 ± 6.07	49.91 ± 9.68	74.94 ± 4.08	67.90 ± 3.82	72.57 ± 5.90	53.45 ± 4.92	47.4 ± 24.78	**75.98 ± 10.5**
C10	**68.40 ± 22.51**	2.19 ± 1.79	18.00 ± 1.97	0.00 ± 0.00	7.98 ± 3.58	15.86 ± 8.25	14.86 ± 7.56	5.98 ± 1.97	7.56 ± 5.63	0.00 ± 0.00	8.66 ± 7.83
C11	94.82 ± 5.10	91.85 ± 9.74	95.74 ± 0.96	84.23 ± 3.58	53.96 ± 7.06	82.04 ± 8.41	92.90 ± 7.17	88.15 ± 7.72	93.96 ± 1.98	17.31 ± 34.62	**98.51 ± 1.68**
OA	83.87 ± 0.55	80.00 ± 0.92	88.09 ± 0.07	85.41 ± 1.90	83.73 ± 1.36	88.09 ± 0.49	87.49 ± 0.72	85.50 ± 0.56	84.10 ± 2.16	88.65 ± 0.96	**88.99 ± 0.79**
AA	78.35 ± 2.23	64.88 ± 4.28	77.04 ± 0.45	72.46 ± 4.24	67.54 ± 2.17	76.59 ± 1.05	75.32 ± 1.90	71.36 ± 0.94	73.02 ± 2.40	67.44 ± 7.47	**80.12 ± 1.32**
Kappa	78.62 ± 0.69	73.35 ± 1.27	84.21 ± 0.11	80.41 ± 3.01	78.46 ± 1.84	84.20 ± 0.67	83.34 ± 0.99	80.71 ± 0.77	78.88 ± 2.86	84.84 ± 1.34	**85.39 ± 1.08**
Time (s)	**11.75**	58.57	47.70	115.9	225.94	23.46	54.17	18.63	32.96	49.4	66.53

To further test the performance of the proposed network, experiments are conducted on the Houston2013 dataset. The classification results and the full-factor classification maps are given in Table 12 and Figure 10. The spectral-based SVM obtains the lowest OA (75.70%). The spatial–spectral-based methods provide higher OAs (79.98%, 86.53%, and 89.09%) than SVM. PMCN obtains the highest OA among these spatial–spectral-based deep learning methods, which is composed of multi-scale convolution and attention mechanisms. The spatial–elevation–spectral-based methods obtain relatively higher OAs than those of SVM and spatial–spectral-based methods. CMCN achieves the highest OA (89.47%) among all competitors. Checking the classification accuracies of each category, we can see that C9-Road and C12-Parking Lot 1 are relatively difficult to classify. For C9-Road, the classification accuracies of all methods are below 90%. FusAtNet gives the lowest accuracy (53.10%), while CMCN obtains the highest result (85.28%). For C12-Parking Lot 1, SVM provides the lowest accuracy (44.75%), and MS2CAN gives the highest accuracy (93.83%). The proposed CMCN obtains an accuracy of 76.83%, which is not good among the competitors. It may be caused by the scattered distribution of labeled samples in C9-Road and C12-Parking Lot 1, where spatial and elevation information cannot be fully utilized. Observing the full-factor classification maps and the ground-truth map, we can see that CMCN provides more clear and smooth classification maps in most categories. In Region I and II, it can be seen that buildings, roads, and land covers can be clearly identified. However, some of the ground objects in Region II are still poorly recognized due to cloud obscuration. For the training time, SVM provides the shortest time (3.49 s), and FusAtNet obtains the longest time (91.21 s). The training time of CMCN is 24.2 s.

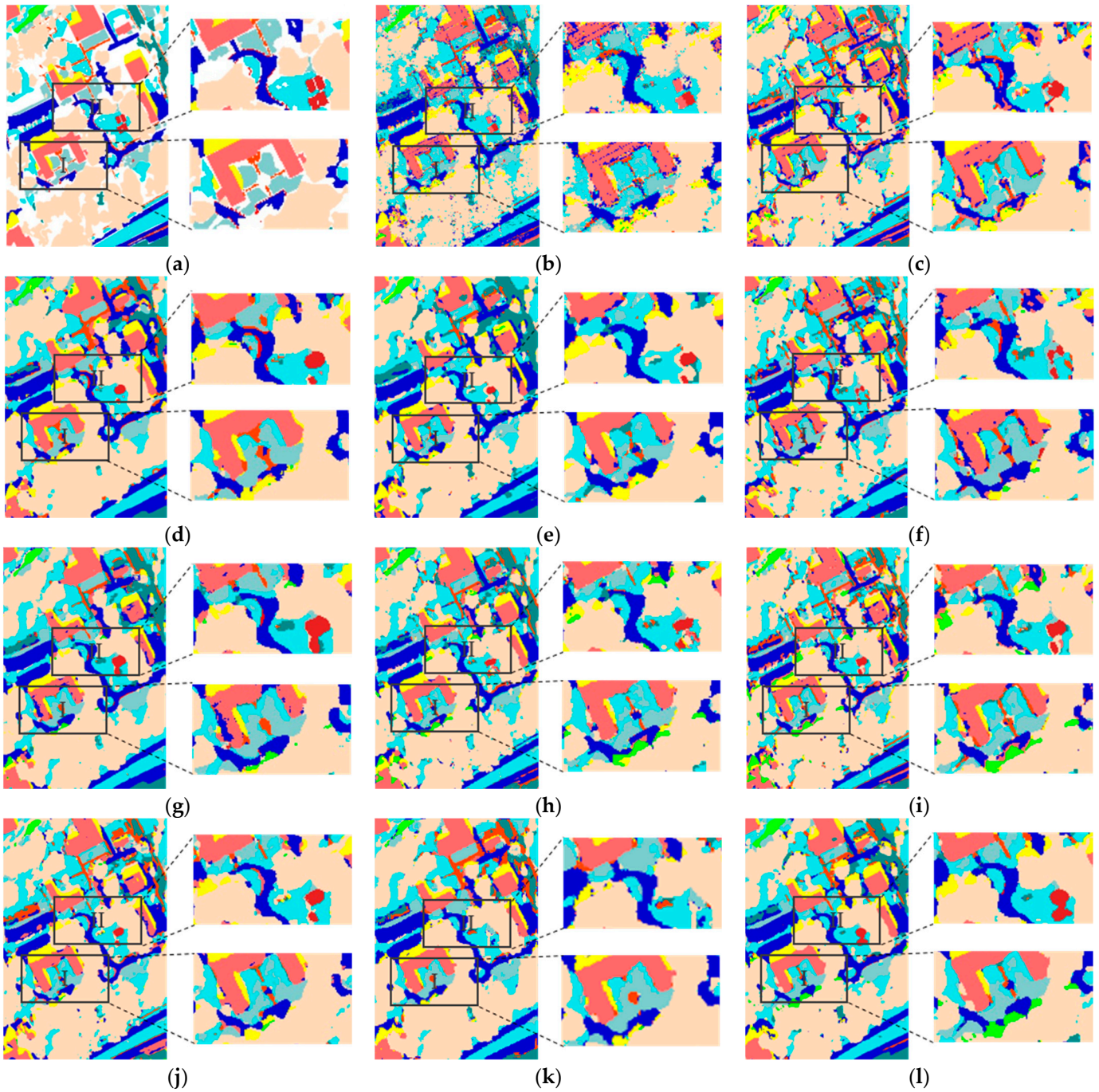


Figure 9. Full-factor classification maps for the MUUFL dataset: (**a**) Ground-truth map; (**b**) SVM; (**c**) HYSN; (**d**) DBDA; (**e**) PMCN; (**f**) FusAtNet; (**g**) CCNN; (**h**) AM^{3}net; (**i**) HCTnet; (**j**) Sal^{2}RN; (**k**) MS2CAN; (**l**) CMCN.

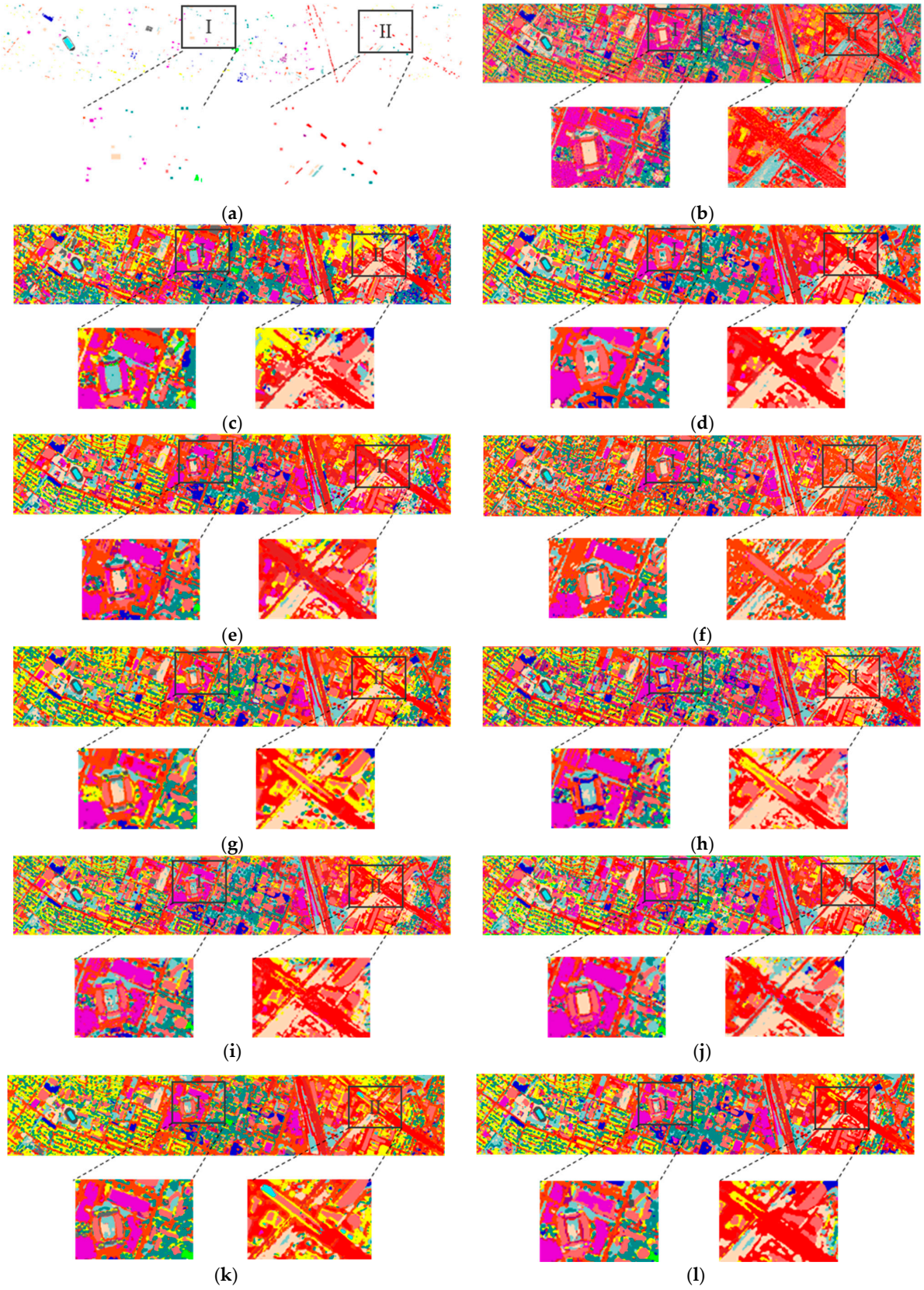


Figure 10. Full-factor classification maps for the Houston2013 dataset: (**a**) Ground-truth map; (**b**) SVM; (**c**) HYSN; (**d**) DBDA; (**e**) PMCN; (**f**) FusAtNet; (**g**) CCNN; (**h**) AM^{3}net; (**i**) HCTnet; (**j**) Sal^{2}RN; (**k**) MS2CAN; (**l**) CMCN.

Table 12. Classification OA (%), AA (%), and Kappa (×100) with standard deviation and training time (s) of the Houston2013 dataset.

Class	SVM	HYSN	DBDA	PMCN	FusAtNet	CCNN	AM^{3}net	HCTnet	Sal^{2}RN	MS2CAN	CMCN
C1	89.59 ± 6.06	82.83 ± 5.05	83.48 ± 0.99	87.24 ± 0.48	85.22 ± 6.19	85.06 ± 1.50	78.66 ± 4.85	87.75 ± 2.35	74.92 ± 12.21	85.38 ± 2.34	**91.69 ± 4.44**
C2	90.72 ± 4.06	91.82 ± 0.86	88.42 ± 0.44	**98.59 ± 0.15**	92.59 ± 11.8	80.46 ± 4.63	73.61 ± 3.18	82.13 ± 8.35	82.14 ± 7.48	80.66 ± 4.76	88.32 ± 1.79
C3	98.38 ± 2.89	96.42 ± 2.63	99.53 ± 0.11	**100.00 ± 0.00**	70.81 ± 13.6	99.94 ± 0.12	99.67 ± 0.65	99.97 ± 0.06	98.01 ± 0.26	94.20 ± 1.95	**100.00 ± 0.00**
C4	94.79 ± 2.76	81.61 ± 1.94	79.15 ± 1.70	95.57 ± 0.48	94.02 ± 4.55	92.92 ± 1.25	**97.24 ± 0.33**	96.40 ± 0.91	93.53 ± 1.42	92.61 ± 1.02	95.64 ± 1.92
C5	93.58 ± 2.93	84.68 ± 2.22	92.26 ± 1.17	**99.09 ± 0.19**	90.68 ± 12.4	90.26 ± 2.50	94.27 ± 1.33	95.37 ± 1.97	95.77 ± 1.67	96.27 ± 1.34	95.15 ± 0.37
C6	98.15 ± 5.38	86.97 ± 5.77	**100.00 ± 0.00**	99.42 ± 0.18	42.84 ± 9.91	98.98 ± 0.42	99.77 ± 0.19	99.86 ± 0.28	99.93 ± 0.14	94.65 ± 1.31	97.29 ± 0.86
C7	76.36 ± 5.91	63.07 ± 4.64	84.29 ± 0.57	87.04 ± 0.68	80.73 ± 4.70	**91.50 ± 3.7**	83.60 ± 1.08	85.52 ± 2.64	83.33 ± 1.07	83.20 ± 1.00	87.94 ± 1.94
C8	61.29 ± 13.62	75.66 ± 1.72	87.32 ± 0.42	84.46 ± 0.87	89.37 ± 5.20	83.68 ± 1.66	67.96 ± 2.22	88.65 ± 1.05	**91.27 ± 3.52**	88.94 ± 1.80	86.05 ± 7.76
C9	58.49 ± 11.04	72.15 ± 8.44	74.23 ± 0.40	75.45 ± 1.60	53.10 ± 4.33	79.28 ± 2.27	75.55 ± 1.83	75.42 ± 3.27	79.96 ± 4.76	80.96 ± 2.88	**85.28 ± 2.84**
C10	63.75 ± 8.62	73.77 ± 4.48	**93.32 ± 1.21**	88.44 ± 0.73	68.81 ± 12.1	82.83 ± 1.72	79.71 ± 1.97	85.19 ± 0.67	79.90 ± 3.39	87.15 ± 1.34	88.73 ± 1.75
C11	67.76 ± 10.33	79.48 ± 5.81	86.35 ± 0.44	77.32 ± 2.20	86.97 ± 5.99	90.10 ± 2.98	79.89 ± 1.33	89.05 ± 2.83	83.00 ± 4.18	**96.92 ± 1.39**	94.83 ± 1.63
C12	56.76 ± 5.44	75.51 ± 2.62	81.96 ± 0.20	87.90 ± 2.02	72.10 ± 4.42	91.47 ± 4.26	75.09 ± 2.47	83.15 ± 3.94	82.34 ± 5.84	**93.83 ± 3.14**	76.83 ± 2.05
C13	44.75 ± 23.69	79.65 ± 8.29	84.51 ± 1.11	89.26 ± 0.91	82.57 ± 8.42	85.06 ± 1.29	90.61 ± 3.49	97.03 ± 4.69	89.90 ± 7.51	97.34 ± 0.78	**99.15 ± 0.19**
C14	89.09 ± 11.03	90.73 ± 2.36	96.40 ± 3.07	**100.00 ± 0.00**	69.07 ± 8.08	99.43 ± 0.35	96.25 ± 1.91	99.29 ± 0.58	98.05 ± 1.43	91.53 ± 4.24	91.95 ± 0.39
C15	**98.03 ± 3.24**	88.13 ± 3.98	94.55 ± 0.22	94.17 ± 0.17	86.61 ± 3.30	91.02 ± 1.08	95.62 ± 0.89	95.15 ± 1.05	89.46 ± 6.37	91.10 ± 0.94	82.19 ± 0.44
OA	75.70 ± 1.51	79.98 ± 2.55	86.53 ± 0.10	89.09 ± 0.22	78.66 ± 1.11	87.90 ± 0.73	83.01 ± 0.40	88.50 ± 0.87	85.37 ± 2.25	89.18 ± 0.58	**89.47 ± 0.52**
AA	78.77 ± 2.36	81.50 ± 1.11	88.39 ± 0.09	90.93 ± 0.21	77.70 ± 0.83	89.46 ± 0.60	85.83 ± 0.51	**90.66 ± 0.58**	88.10 ± 1.53	90.32 ± 0.51	90.47 ± 0.45
Kappa	73.69 ± 1.64	78.36 ± 3.09	85.44 ± 0.11	88.20 ± 0.27	76.91 ± 1.22	86.92 ± 0.79	81.62 ± 0.44	87.56 ± 0.94	84.18 ± 2.44	88.30 ± 0.63	**88.62 ± 0.56**
Time (s)	**3.49**	13.76	14.64	56.65	91.21	8.82	29.28	9.50	14.59	15.20	24.20

5. Discussion

5.1. Impact of the Hyper-Parameters

To detect the impact of the hyper-parameters on the proposed network, four hyper-parameters are investigated in the experiments, including the patch size of the data cubes (h, w), the number of feature maps (f), the multiplicity of channel compression (r), and the number of heads (h). The experimental results are demonstrated in Figure 11.

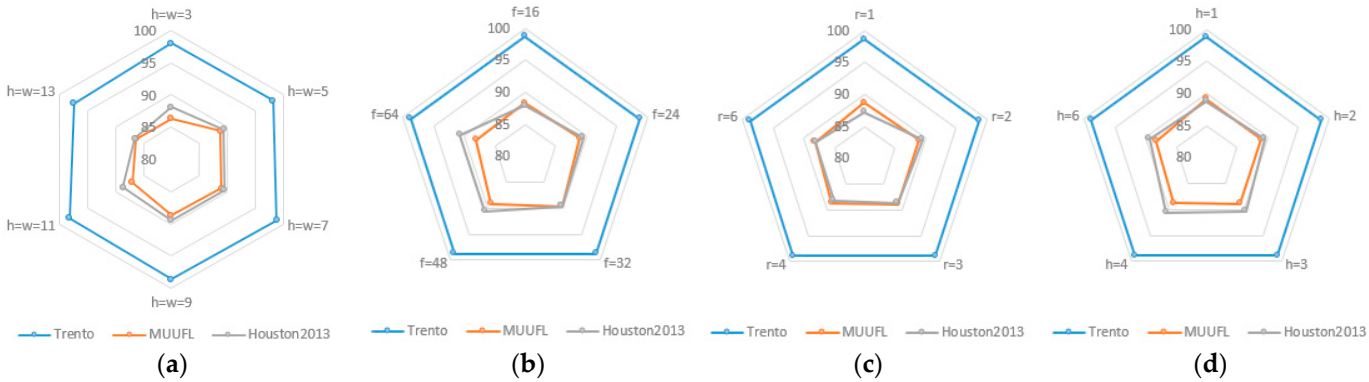


Figure 11. Impact of hyper-parameters on the OA of the CMCN in the Trento, MUUFL, and Houston2013 datasets: (**a**) Impact of the patch size of the data cubes (*patch*); (**b**) impact of the number of feature maps (f); (**c**) impact of the multiplicity of channel compression (r); (**d**) impact of the number of heads (h).

To be specific, Figure 11a illustrates the OAs of CMCN for Trento, MUUFL, and Houston2013 datasets under different patch sizes of the data cubes. We can see clearly

that the OAs of CMCN on the three datasets increase and then decrease as the patch sizes increase. This phenomenon is easy to appreciate since small patches cannot provide enough spatial information, while too large patches present more redundant data. As a result, we need to choose an appropriate patch size to optimize the classification results. In the experiment, the default value of patch size is set to $h = w = 7$.

Figure 11b illustrates the OAs of CMCN for three HSI-LiDAR pair datasets under different numbers of feature maps, which appear in the multi-scale convolutional blocks. A larger number of feature maps indicates more learnable parameters and a more complex network structure. From the experimental results, the best f is 24 for the Trento dataset (98.83%). For the MUUFL dataset, the best f is 32 (89.62%). For the Houston2013 dataset, the best f is 64 (90.73%). It shows that increasing the number of feature maps is helpful to boost the performance of the network to some extent. To strike a balance between performance and efficiency, we choose $f = 24$ as the default hyper-parameter setting for the actual experiments.

Figure 11c shows the OAs of CMCN under the different multiplicities of channel compression in the CAFM. A larger multiplicity indicates a smaller channel width. From the experimental results, it is pleased to find that higher compression multiplicity ($r = 4$) provides higher OA (98.92%) on the Trento, and there is a small decrease in OAs (0.07%, 0.32%, 0.85%; 0.83%, 1.26%, 1.40%) compared with $r = 2$ on the MUUFL and Houston2013 datasets. The experimental results encouraged us to increase the channel compression ratio in future research. In the experiment, the default value of the multiplicity of channel compression is set to $r = 2$.

Finally, we compare the performance of CMCN for different numbers of heads in CAFM, and the experimental results are displayed in Figure 11d. We can see that the number of heads appears to have a small impact on the Trento (98.77%, 98.83%, 98.85%, 98.95%, and 98.87%) and MUUFL (89.15%, 88.99%, 88.91%, 88.76%, and 88.21%) datasets and a large impact on the Houston2013 dataset (88.74%, 89.47%, 90.41%, 90.62%, and 89.37%). Theoretically, a larger number of heads could provide more flexibility in the attention mechanism. Considering the robustness of the model, we set the default value of the number of heads as $head = 2$ in the experiments.

5.2. Investigation of the Proportion of Training Samples

In this section, we examine the performance of the competitors in the context of different proportions of training samples on three HSI-LiDAR datasets. It is an important investigation since supervised learning methods are data-driven algorithms, and the number of training samples can directly determine the performance of the classification methods. To comprehensively analyze the performance of the competitors under different training sample conditions, we randomly selected 0.5%, 1%, 2%, 4%, and 5% labeled samples in each class to compose the training sample set. The classification results are reported in Figure 12. In general, a large number of training samples enables the supervised learning methods to be adequately trained and also provides sufficient discriminative information, thus helping the methods to improve the discriminative ability. It is confirmed in the experiment, where the OAs of the competitors are low when the proportion of training samples is small, and vice versa. When the proportion of training samples is small (0.5%), the OAs of MS2CAN and CMCN decrease less than those of other methods. It indicates that these methods can better extract effective information under small sample conditions. CMCN provides consistently competitive results with the increase of the training sample proportion. When the training sample proportion reaches 5%, the classification accuracies of all methods achieve the maximum values. Among them, CMCN obtained the highest classification accuracy in all three datasets, where the accuracies are saturated in the Trento (99.7%) and Houston2013 (98.83%) datasets. The accuracy is 94.01% in the MUUFL dataset. The experimental results demonstrate again the superiority of the multi-scale convolutional network coupled with the local–global cross attention mechanism in the task of pixel classi-

fication of remote sensing images and provide thoughts to design the fusion network for the multi-source dataset.

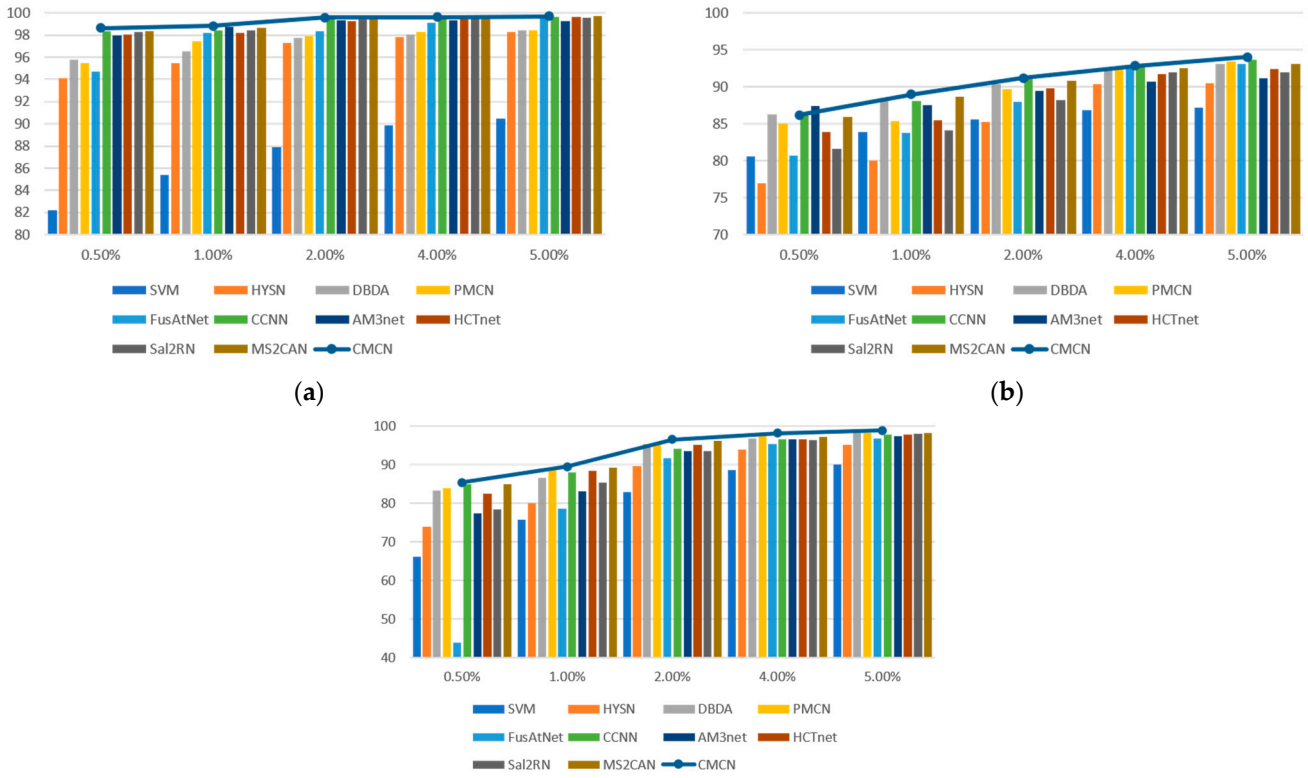


Figure 12. Investigation of the proportion of training samples: (**a**) Trento dataset; (**b**) MUUFL dataset; (**c**) Houston2013 dataset.

5.3. Computational Cost and Visualization of Data Features

In this section, we will discuss the computational complexity of the competitors and demonstrate the feature extraction ability of the proposed method in visualization. The number of learnable parameters (Par) and floating-point operations (FLOPs) of the competitors on three datasets are shown in Table 13. In general, more complex convolutional structures and larger input data will increase the number of learnable parameters and the FLOPs of the network. This phenomenon is observed in Table 13. For example, both HYSN and FusAtNet contain a large number of learnable parameters, and at the same time, their FLOPs are extensive. When the input data are the Houston2013 dataset (larger spectral size and more categories), the number of learnable parameters and FLOPs of HYSN and FusAtNet also increases. A similar situation is also found in the DBDA, CCNN, MS2CAN, and HCTnet, where their number of learnable parameters and FLOPs is relatively small. However, the variations of parameters and FLOPs of PMCN, AM^{3}net, Sal^{2}RN, and CMCN are different from those of the previous methods. AM^{3}net and Sal^{2}RN have a large number of parameters but relatively low FLOPs. This is because a large number of convolutional layers and attention blocks are used to compose the network in AM^{3}net, which will increase the learnable parameters. The input data are uniformly compressed as data preprocessing, which can reduce the FLOPs and make the parameters and FLOPs the same value on all datasets. In addition, AM^{3}net uses involution convolution to extract spectral features for single pixels rather than for the entire data cube, which can further reduce the FLOPs of the network. Sal^{2}RN employs dense blocks in the network, which leads to smaller FLOPs for computation. In contrast, the number of learnable parameters for PMCN and CMCN are small. However, the FLOPs of PMCN and CMCN are relatively extensive. Examining

the computational process of PMCN and CMCN, we can see that there is a large number of convolutional operations in the networks, which provide a significant number of FLOPs.

Table 13. Number of parameters (Par) (M) and FLOPs (G) of the competitors.

		HYSN	DBDA	PMCN	FusAtNet	CCNN	AM^{3}net	HCTnet	Sal^{2}RN	MS2CAN	CMCN
Trento	Par	1.00	0.13	0.08	36.24	0.11	2.65	0.58	0.91	0.19	0.12
	FLOPs	1.36	0.64	4.78	108.28	0.12	0.23	0.51	0.21	0.17	2.13
MUUFL	Par	1.02	0.13	0.08	36.26	0.12	2.65	0.59	0.91	0.19	0.12
	FLOPs	1.40	0.64	4.86	108.31	0.12	0.23	0.52	0.21	0.17	2.16
Houston2013	Par	1.76	0.28	0.13	36.90	0.14	2.65	0.96	0.97	0.22	0.17
	FLOPs	3.25	1.49	10.70	110.81	0.21	0.23	1.15	0.27	0.29	4.52

To intuitively investigate the feature extraction capability of the proposed CMCN, the t-Distributed Stochastic Neighbor Embedding (t-SNE) [74] is adopted to visualize the distributions of the original HSI data and the output data of the CMCN for the three datasets in low-dimensional feature space. The feature distributions are illustrated in Figure 13. We can see that the feature distribution of the original HSI data in the three datasets appears more chaotic. Some categories are characterized by overlapping distributions, such as C2-Buildings and C6-Roads, C1-Apple trees, and C5-Vineyard in the Trento dataset, C1-Trees, C6-Water, and C7-Building Shadow in the MUUFL dataset, and C8-Commercial, C10-Highway, C11-Railway, and C13-Parking Lot 2 in the Houston2013 dataset. The CMCN can be regarded as a feature mapping method, which can help map the features into distribution states that are easy to distinguish. As shown in Figure 13, the distributions of the HSI-LiDAR data features are more dispersed in the feature space after being processed by the CMCN network, which will assist in distinguishing different categories of ground objects.

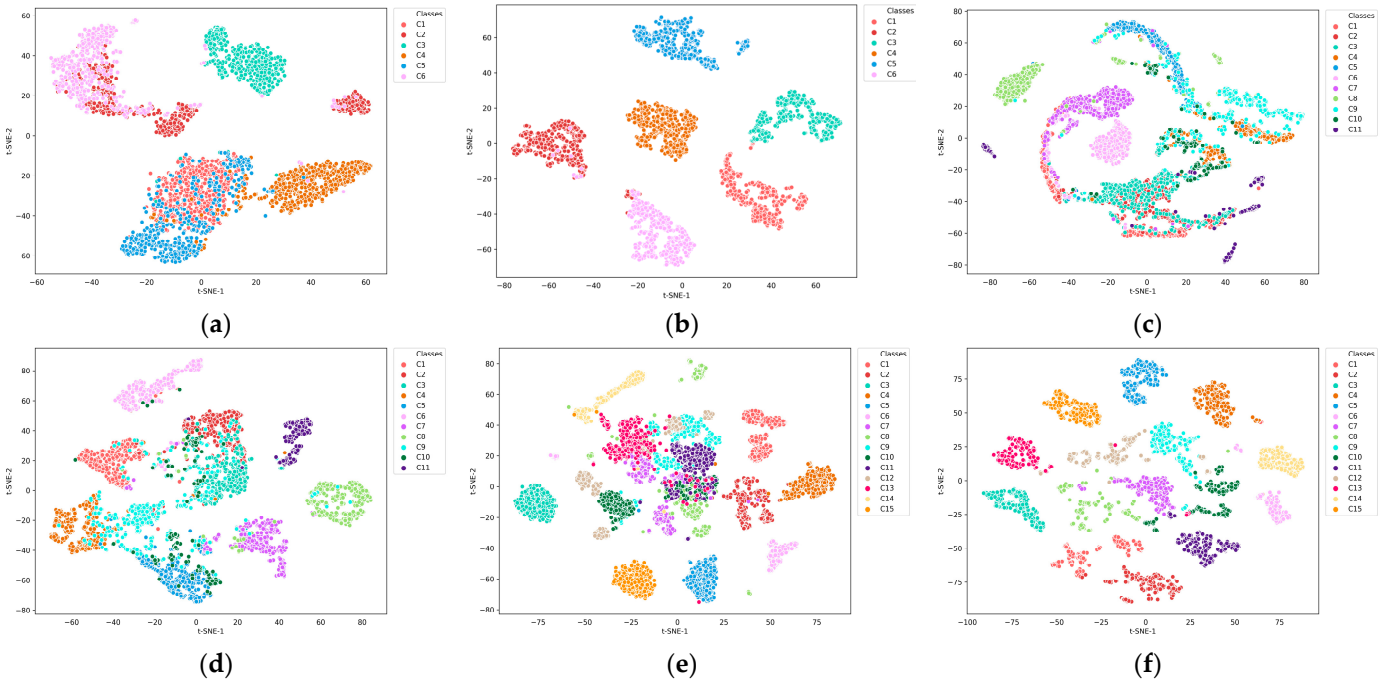


Figure 13. Visualization of the distributions of the original HSI and the output of the CMCN for the three datasets. (**a**) Original Trento. (**b**) Processed Trento. (**c**) Original MUUFL. (**d**) Processed MUUFL. (**e**) Original Houston2013. (**f**) Processed Houston2013.

5.4. Model Analysis

In this section, model analysis is performed to investigate the effectiveness of the components of the proposed CMCN. The performance of the model is evaluated by iteratively removing each module used in our model. For the triple-branch CNN architecture of SESM, the multi-scale feature extraction blocks of each branch are removed, and the modified models are called CMCN-re-$B_{\mathcal{H}1}$, CMCN-re-$B_{\mathcal{H}2}$, and CMCN-re-$B_{\mathcal{L}}$. For the CAFM, the local–global cross attention (CAt) block is removed, which is called CMCN-re-CAt. In the CMCN-re-CAt, the spatial features, spectral features, and elevation features are concatenated directly to fed into classification module. In addition, the one-shot aggregation (OS) technique and parameter sharing (PS) technique are iteratively removed to check the impact on the model, so the modified models are called CMCN-re-OS and CMCN-re-PS. The experimental results of the models on three datasets are shown in Table 14. The best OA, AA, and Kappa are highlighted in bold.

Table 14. Model analysis. The CMCN-re-$B_{\mathcal{H}1}$ is the model obtained by removing the $B_{\mathcal{H}1}$ branch from the CMCN. The CMCN-re-$B_{\mathcal{H}2}$ is the model obtained by removing the $B_{\mathcal{H}2}$ branch from the CMCN. The CMCN-re-$B_{\mathcal{L}}$ is the model obtained by removing the $B_{\mathcal{L}}$ branch from the CMCN. The CMCN-re-CAt is the model obtained by removing the local–global cross attention block from the CMCN. The CMCN-re-OS is the model obtained by removing the one-shot aggregation technique from the CMCN. The CMCN-re-PS is the model obtained by removing the parameter-sharing technique from the CMCN. The CMCN is the proposed model.

Method	Trento			MUUFL			Houston 2013		
	OA (%)	AA (%)	Kappa × 100	OA (%)	AA (%)	Kappa × 100	OA (%)	AA (%)	Kappa × 100
CMCN-re-$B_{\mathcal{H}1}$	98.60 ± 0.40	98.10 ± 0.48	98.13 ± 0.53	87.43 ± 0.88	79.35 ± 3.12	83.20 ± 1.18	87.16 ± 0.75	90.02 ± 0.37	86.11 ± 0.82
CMCN-re- $B_{\mathcal{H}2}$	98.61 ± 0.24	98.03 ± 0.35	98.15 ± 0.33	86.70 ± 1.64	78.36 ± 2.05	82.29 ± 2.14	87.05 ± 0.59	88.46 ± 0.45	86.00 ± 0.64
CMCN-re- $B_{\mathcal{L}}$	98.12 ± 0.70	97.40 ± 0.70	97.49 ± 0.93	86.01 ± 1.17	78.15 ± 1.84	81.16 ± 1.61	88.02 ± 0.46	88.77 ± 0.35	87.06 ± 0.49
CMCN-re-CAt	98.34 ± 0.82	97.77 ± 0.99	97.79 ± 1.09	88.32 ± 0.60	79.01 ± 1.58	84.32 ± 0.85	88.39 ± 0.28	90.24 ± 0.30	87.45 ± 0.31
CMCN-re-OS	98.72 ± 0.35	98.28 ± 0.48	98.21 ± 0.47	88.19 ± 0.84	77.97 ± 1.60	84.22 ± 1.13	88.93 ± 0.58	90.54 ± 0.30	88.03 ± 0.63
CMCN-re-PS	98.28 ± 0.37	97.57 ± 0.46	97.70 ± 0.49	88.05 ± 0.93	78.62 ± 1.77	84.11 ± 1.23	88.87 ± 0.89	90.46 ± 0.73	87.96 ± 0.96
CMCN	**98.83 ± 0.11**	**98.38 ± 0.08**	**98.44 ± 0.14**	**88.99 ± 0.79**	**80.12 ± 1.32**	**85.39 ± 1.08**	**89.47 ± 0.52**	**90.47 ± 0.45**	**88.62 ± 0.56**

Comparing CMCN-re-$B_{\mathcal{H}1}$, CMCN-re-$B_{\mathcal{H}2}$, and CMCN-re-$B_{\mathcal{L}}$ with our model, we can see that the classification accuracies of the models decrease in all three datasets. It suggests that the multi-scale feature extraction blocks in the three branches can exploit discriminative information for the model, thus enhancing the performance of the model to some extent. A further observation of the experimental results shows that the three branches perform differently on the three datasets. For Trento and MUUFL, $\mathcal{B}_{\mathcal{L}}$ makes the highest impact on OAs, decreasing by 0.71% and 2.98%, respectively. For Houston2013, $\mathcal{B}_{\mathcal{H}2}$ produced the highest drop, which is 2.42%. It is a consequence of the complex data characteristics of the remote sensing datasets, which encourages us to collect spatial, elevation, and spectral information in a comprehensive nature. Comparing the OAs of CMCN-re-CAt and our model, we can see that the OAs decrease on all three datasets (0.49%, 0.67%, and 1.08%) after removing the cross attention block. It indicates that the local–global cross attention mechanism can further boost the generalization of the model by integrating relationships of the feature embeddings. For CMCN-re-OS, we can see that there are small increases (0.11%, 0.80%, and 0.54%) in the OAs after applying the one-shot aggregation technique. It demonstrates that the performance of the model is improved by maintaining the shallow features to deep layers. Finally, we tested the impact of the parameter-sharing technique, which is CMCN-re-PS. Encouragingly, we obtained positive experimental results indicating that the parameter-sharing technique is beneficial. The increases of OAs are 0.55%, 0.94%, and 0.60%, respectively. The results encourage us to utilize flexible parameter-sharing techniques to collect correlative and complementary information in future studies.

5.5. Limitations of the Model

In this study, a deep learning-based model is proposed for HSI-LiDAR joint classification. However, there are still some limitations to be considered. First, the predictions of the model are influenced by the quality of the labeled samples. Errors or biases in the real data may affect the discriminatory power of the model. Second, the predictions for the categories with small sample sizes in the unbalanced data are inaccurate. Third, the FLOPs of the model are large, resulting in a longer training time than the competitors. Fourth, the number of adjustable parameters is relatively large, which increases the difficulty of optimizing the model. Fifth, the interpretability of the model may affect the application of the model to real tasks. When using the model to predict and analyze remote sensing data, we need to be aware of these limitations. And we need to continuously improve and refine the model to improve the prediction and interpretability.

6. Conclusions

In this paper, a cross attention-based multi-scale convolutional fusion network is proposed for pixel-wise HSI-LiDAR classification. The proposed model consists of three modules, which are SESM, CAFM, and classification module. The SESM is used to extract spatial, elevation, and spectral features of the HSI and LiDAR data. The CAFM is implemented to fuse the extracted HSI and LiDAR features in a cross-modal representation learning manner and generate joint semantic information. The classification module is employed to map the semantic features to classification results. Some techniques such as multi-scale convolution, cross attention mechanism, one-shot aggregation, residual aggregation, parameter sharing, batch normalization, layer normalization, and Mish activation function are implemented to improve the performance of the network. Three HSI-LiDAR datasets containing different land covers and spectral–spatial resolutions are used to verify the effectiveness of the proposed method. Ten relevant methods are invited for comparison. At the same time, the impact of the hyper-parameters, the proportion of training samples, computational cost, visualization of data features, model analysis, and the limitations are discussed.

In conclusion, our research contributes to the field of multi-source data fusion and classification by proposing an effective framework that combines multi-scale CNN and cross attention techniques. Compared with the state-of-the-art methods, the proposed CMCN provides competitive classification performance on widely used datasets, such as Trento, MUUFL, and Houston2013. The experimental results demonstrate the potential of these techniques in enhancing the ability to extract discriminative spatial–spectral features and capture correlation and complementarity between different data sources for HSI and LiDAR joint classification. Although the proposed method provides efficient performance in HSI-LiDAR classification, there are still several issues to be focused on, such as the quality of the labeled samples, unbalanced available data, FLOPs, parameter scale, and interpretability. In the future, we will aim to overcome these issues and further enhance the robustness and overall performance of our approach in multi-source data fusion and classification.

Author Contributions: Conceptualization, H.G.; Data curation, H.G., H.P., Y.L., C.L., D.L. and H.M.; Formal analysis, H.G. and H.P.; Funding acquisition, H.G., H.P. and L.W.; Investigation, H.G., H.P., Y.L., C.L., D.L. and H.M.; Methodology, H.G.; Project administration, H.G., H.P. and L.W.; Resources, H.G., H.P. and L.W.; Software, H.G.; Supervision, H.G., H.P. and L.W.; Validation, H.G.; Visualization, H.G.; Writing—original draft, H.G.; Writing—review and editing, H.G., H.P., Y.L., C.L., D.L. and H.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 62071084, in part by Heilongjiang Provincial Natural Science Foundation of China under Grant LH2023F050, and in part by Fundamental Research Funds in Heilongjiang Provincial Universities under Grant 145309208.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: The authors would like to thank the handling editor and anonymous reviewers for their insights and comments. The authors also would like to thank the Hyperspectral Image Analysis group at the University of Houston and the IEEE GRSS DFC2013 for providing the CASI University of Houston datasets.

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

Abbreviations

The following abbreviations are used in this manuscript:

HIS	Hyperspectral image
LiDAR	Light detection and ranging
CNN	Convolutional neural network
RNN	Recurrent neural network
CMCN	Cross attention-based multi-scale convolutional fusion network
SESM	Spatial–elevation–spectral convolutional feature extraction module
CAFM	Cross attention fusion module
Batch Norm	Batch normalization
Mish	Mish activation function
Layer Norm	Layer normalization
DSM	Digital surface model
SVM	Support vector machine
HYSN	Hybrid spectral convolutional neural network
DBDA	Double-branch dual-attention mechanism network
PMCN	Pyramidal multiscale convolutional network with polarized self-attention
FusAtNet	Dual attention-based spectro–spatial multimodal fusion network
CCNN	Coupled convolutional neural network
AM^3net	Adaptive mutual-learning-based multimodal data fusion network
HCTnet	Hierarchical CNN and transformer
Sal^2RN	Spatial–spectral salient reinforcement network
MS2CAN	Multi-scale spatial–spectral cross-modal attention network
Adam	Adaptive moment estimation
OA	Overall accuracy
AA	Average accuracy
Kappa	Kappa coefficient
Par	Parameters
FLOPs	Floating-point operations
t-SNE	t-Distributed stochastic neighbor embedding
CAt	Cross attention
OS	One-shot aggregation
PS	Parameter sharing

References

1. Tong, X.D. Promote the implementation of high-score projects and help the construction of the "Belt and Road" initiative. *Spacecr. Recovery Remote Sens.* **2018**, *39*, 18–25.
2. Sun, W.W.; Yang, G.; Chen, C.; Chang, M.H.; Huang, K.; Meng, M.Z.; Liu, L.Y. Development status and literature analysis of China's earth observation remote sensing satellites. *J. Remote Sens.* **2020**, *24*, 479–510. [CrossRef]
3. Hong, D.F.; Gao, L.R.; Yao, J.; Zhang, B.; Plaza, A.; Chanussot, J. Graph Convolutional Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 5966–5978. [CrossRef]
4. Rasti, B.; Ghamisi, P.; Gloaguen, R. Hyperspectral and LiDAR Fusion Using Extinction Profiles and Total Variation Component Analysis. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3997–4007. [CrossRef]
5. Chen, S.; Li, Q.; Zhong, W.S.; Wang, R.; Chen, D.; Pan, S.H. Improved Monitoring and Assessment of Meteorological Drought Based on Multi-Source Fused Precipitation Data. *Int. J. Environ. Res. Public Health* **2022**, *19*, 1542. [CrossRef]
6. Judah, A.; Hu, B.X. An Advanced Data Fusion Method to Improve Wetland Classification Using Multi-Source Remotely Sensed Data. *Sensors* **2022**, *22*, 8942. [CrossRef]

7. Li, S.T.; Li, C.Y.; Kang, X.D. Development Status and Future Prospects of Multi-source Remote Sensing Image Fusion. *J. Remote Sens.* **2021**, *25*, 148–166. [CrossRef]
8. Li, J.X.; Hong, D.F.; Gao, L.R.; 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]
9. Cai, J.H.; Zhang, M.; Yang, H.F.; He, Y.T.; Yang, Y.Q.; Shi, C.H.; Zhao, X.J.; Xun, Y.L. A novel graph-attention based multimodal fusion network for joint classification of hyperspectral image and LiDAR data. *Expert Syst. Appl.* **2024**, *249*, 123587. [CrossRef]
10. Schmitt, M.; Zhu, X.X. Data Fusion and Remote Sensing: An ever-growing relationship. *IEEE Geosci. Remote Sens. Mag.* **2016**, *4*, 6–23. [CrossRef]
11. Yu, C.Y.; Han, R.; Song, M.P.; Liu, C.Y.; Chang, C.I. A Simplified 2D-3D CNN Architecture for Hyperspectral Image Classification Based on Spatial-Spectral Fusion. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 2485–2501. [CrossRef]
12. Hao, J.; Dong, F.J.; Wang, S.L.; Li, Y.L.; Cui, J.R.; Men, J.L.; Liu, S.J. Combined hyperspectral imaging technology with 2D convolutional neural network for near geographical origins identification of wolfberry. *J. Food Meas. Charact.* **2022**, *16*, 4923–4933. [CrossRef]
13. Liu, D.X.; Han, G.L.; Liu, P.X.; Yang, H.; Sun, X.L.; Li, Q.Q.; Wu, J.J. A Novel 2D-3D CNN with Spectral-Spatial Multi-Scale Feature Fusion for Hyperspectral Image Classification. *Remote Sens.* **2021**, *13*, 4621. [CrossRef]
14. Zhao, J.L.; Wang, G.L.; Zhou, B.; Ying, J.J.; Liu, J. Exploring an application-oriented land-based hyperspectral target detection framework based on 3D-2D CNN and transfer learning. *Eurasip J. Adv. Signal Process.* **2024**, *2024*, 37. [CrossRef]
15. Hang, R.L.; Liu, Q.S.; Hong, D.F.; Ghamisi, P. Cascaded Recurrent Neural Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 5384–5394. [CrossRef]
16. Peng, Y.B.; Ren, J.S.; Wang, J.M.; Shi, M.L. Spectral-Swin Transformer with Spatial Feature Extraction Enhancement for Hyperspectral Image Classification. *Remote Sens.* **2023**, *15*, 2696. [CrossRef]
17. Sun, J.; Zhang, J.B.; Gao, X.S.; Wang, M.T.; Ou, D.H.; Wu, X.B.; Zhang, D.J. Fusing Spatial Attention with Spectral-Channel Attention Mechanism for Hyperspectral Image Classification via Encoder-Decoder Networks. *Remote Sens.* **2022**, *14*, 1968. [CrossRef]
18. Roy, S.K.; Deria, A.; Hong, D.F.; Rasti, B.; Plaza, A.; Chanussot, J. Multimodal Fusion Transformer for Remote Sensing Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5515620. [CrossRef]
19. Arshad, T.; Zhang, J.P.; Anyembe, S.C.; Mehmood, A. Spectral Spatial Neighborhood Attention Transformer for Hyperspectral Image Classification. *Can. J. Remote Sens.* **2024**, *50*, 2347631. [CrossRef]
20. Li, Y.P.; Luo, Y.; Zhang, L.F.; Wang, Z.M.; Du, B. MambaHSI: Spatial-Spectral Mamba for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5524216. [CrossRef]
21. Chen, H.Y.; Long, H.Y.; Chen, T.; Song, Y.J.; Chen, H.L.; Zhou, X.B.; Deng, W. M^{3}FuNet: An Unsupervised Multivariate Feature Fusion Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5513015. [CrossRef]
22. Hang, R.L.; Li, Z.; Ghamisi, P.; Hong, D.F.; Xia, G.Y.; Liu, Q.S. Classification of Hyperspectral and LiDAR Data Using Coupled CNNs. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 4939–4950. [CrossRef]
23. Wang, X.H.; Feng, Y.N.; Song, R.X.; Mu, Z.H.; Song, C.M. Multi-attentive hierarchical dense fusion net for fusion classification of hyperspectral and LiDAR data. *Inf. Fusion* **2022**, *82*, 1–18. [CrossRef]
24. Zhou, L.; Geng, J.; Jiang, W. Joint Classification of Hyperspectral and LiDAR Data Based on Position-Channel Cooperative Attention Network. *Remote Sens.* **2022**, *14*, 3247. [CrossRef]
25. Zhao, X.D.; Tao, R.; Li, W.; Li, H.C.; Du, Q.; Liao, W.Z.; Philips, W. Joint Classification of Hyperspectral and LiDAR Data Using Hierarchical Random Walk and Deep CNN Architecture. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 7355–7370. [CrossRef]
26. Zhang, H.T.; Yao, J.; Ni, L.; Gao, L.R.; Huang, M. Multimodal Attention-Aware Convolutional Neural Networks for Classification of Hyperspectral and LiDAR Data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *16*, 3635–3644. [CrossRef]
27. Wang, J.P.; Li, J.; Shi, Y.L.; Lai, J.H.; Tan, X.J. AM^{3}Net: Adaptive Mutual-Learning-Based Multimodal Data Fusion Network. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 5411–5426. [CrossRef]
28. Feng, Y.N.; Song, L.Y.; Wang, L.; Wang, X.H. DSHFNet: Dynamic Scale Hierarchical Fusion Network Based on Multiattention for Hyperspectral Image and LiDAR Data Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5522514. [CrossRef]
29. Li, H.; Ghamisi, P.; Rasti, B.; Wu, Z.Y.; Shapiro, A.; Schultz, M.; Zipf, A. A Multi-Sensor Fusion Framework Based on Coupled Residual Convolutional Neural Networks. *Remote Sens.* **2020**, *12*, 2067. [CrossRef]
30. Ge, C.R.; Du, Q.; Sun, W.W.; Wang, K.Y.; Li, J.J.; Li, Y.S. Deep Residual Network-Based Fusion Framework for Hyperspectral and LiDAR Data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 2458–2472. [CrossRef]
31. Arun, P.V.; Sadeh, R.; Avneri, A.; Tubul, Y.; Camino, C.; Buddhiraju, K.M.; Porwal, A.; Lati, R.N.; Zarco-Tejada, P.J.; Peleg, Z.; et al. Multimodal Earth observation data fusion: Graph-based approach in shared latent space. *Inf. Fusion* **2022**, *78*, 20–39. [CrossRef]
32. Zhang, M.Q.; Gao, F.; Zhang, T.E.; Gan, Y.H.; Dong, J.Y.; Yu, H. Attention Fusion of Transformer-Based and Scale-Based Method for Hyperspectral and LiDAR Joint Classification. *Remote Sens.* **2023**, *15*, 650. [CrossRef]
33. Zhao, G.R.; Ye, Q.L.; Sun, L.; Wu, Z.B.; Pan, C.S.; Jeon, B. Joint Classification of Hyperspectral and LiDAR Data Using a Hierarchical CNN and Transformer. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5500716. [CrossRef]
34. Mohla, S.; Pande, S.; Banerjee, B.; Chaudhuri, S. Fusatnet: Dual attention based spectrospatial multimodal fusion network for hyperspectral and lidar classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 92–93.

35. Huang, J.; Zhang, Y.H.; Yang, F.; Chai, L.; Tansey, K. Attention-Guided Fusion and Classification for Hyperspectral and LiDAR Data. *Remote Sens.* **2024**, *16*, 94. [CrossRef]
36. Gao, H.M.; Feng, H.; Zhang, Y.Y.; Xu, S.F.; Zhang, B. AMSSE-Net: Adaptive Multiscale Spatial–Spectral Enhancement Network for Classification of Hyperspectral and LiDAR Data. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5531317. [CrossRef]
37. Liu, Y.; Ye, Z.; Xi, Y.Q.; Liu, H.; Li, W.; Bai, L. Multiscale and Multidirection Feature Extraction Network for Hyperspectral and LiDAR Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 9961–9973. [CrossRef]
38. Pan, H.Z.; Liu, M.Q.; Ge, H.M.; Wang, L.G. One-Shot Dense Network with Polarized Attention for Hyperspectral Image Classification. *Remote Sens.* **2022**, *14*, 2265. [CrossRef]
39. Gu, Y.F.; Wang, Q.W. Discriminative Graph-Based Fusion of HSI and LiDAR Data for Urban Area Classification. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 906–910. [CrossRef]
40. Xia, J.S.; Yokoya, N.; Iwasaki, A. Fusion of Hyperspectral and LiDAR Data with a Novel Ensemble Classifier. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 957–961. [CrossRef]
41. Liu, H.; Jia, Y.H.; Hou, J.H.; Zhang, Q.F. Global-Local Balanced Low-Rank Approximation of Hyperspectral Images for Classification. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 2013–2024. [CrossRef]
42. Ghamisi, P.; Höfle, B.; Zhu, X.X. Hyperspectral and LiDAR Data Fusion Using Extinction Profiles and Deep Convolutional Neural Network. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *10*, 3011–3024. [CrossRef]
43. Chen, Y.S.; Li, C.Y.; Ghamisi, P.; Jia, X.P.; Gu, Y.F. Deep Fusion of Remote Sensing Data for Accurate Classification. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1253–1257. [CrossRef]
44. Feng, Q.L.; Zhu, D.H.; Yang, J.Y.; Li, B.G. Multisource Hyperspectral and LiDAR Data Fusion for Urban Land-Use Mapping based on a Modified Two-Branch Convolutional Neural Network. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 28. [CrossRef]
45. Roy, S.K.; Krishna, G.; Dubey, S.R.; Chaudhuri, B.B. HybridSN: Exploring 3-D-2-D CNN Feature Hierarchy for Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 277–281. [CrossRef]
46. Li, H.; Ghamisi, P.; Soergel, U.; Zhu, X.X. Hyperspectral and LiDAR Fusion Using Deep Three-Stream Convolutional Neural Networks. *Remote Sens.* **2018**, *10*, 1649. [CrossRef]
47. Feng, J.; Zhang, J.P.; Zhang, Y. Multiview Feature Learning and Multilevel Information Fusion for Joint Classification of Hyperspectral and LiDAR Data. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5528613. [CrossRef]
48. Li, H.C.; Hu, W.S.; Li, W.; Li, J.; Du, Q.; Plaza, A. A^3 CLNN: Spatial, Spectral and Multiscale Attention ConvLSTM Neural Network for Multisource Remote Sensing Data Classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 747–761. [CrossRef]
49. Song, D.M.; Gao, J.C.; Wang, B.; Wang, M.Y. A Multi-Scale Pseudo-Siamese Network with an Attention Mechanism for Classification of Hyperspectral and LiDAR Data. *Remote Sens.* **2023**, *15*, 1283. [CrossRef]
50. Li, Z.W.; Sui, H.; Luo, C.; Guo, F.M. Morphological Convolution and Attention Calibration Network for Hyperspectral and LiDAR Data Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *16*, 5728–5740. [CrossRef]
51. Lee, Y.; Hwang, J.W.; Lee, S.; Bae, Y.; Park, J. An Energy and GPU-Computation Efficient Backbone Network for Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–17 June 2019.
52. Meng, Q.Y.; Zhao, M.F.; Zhang, L.L.; Shi, W.X.; Su, C.; Bruzzone, L. Multilayer Feature Fusion Network with Spatial Attention and Gated Mechanism for Remote Sensing Scene Classification. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 6510105. [CrossRef]
53. Li, R.; Zheng, S.Y.; Duan, C.X.; Yang, Y.; Wang, X.Q. Classification of Hyperspectral Image Based on Double-Branch Dual-Attention Mechanism Network. *Remote Sens.* **2020**, *12*, 582. [CrossRef]
54. Pan, H.Z.; Zhu, Y.X.; Ge, H.M.; Liu, M.Q.; Shi, C.P. Multiscale cross-fusion network for hyperspectral image classification. *Egypt. J. Remote Sens. Space Sci.* **2023**, *26*, 839–850. [CrossRef]
55. Meng, X.C.; Zhu, L.Q.; Han, Y.L.; Zhang, H.C. We Need to Communicate: Communicating Attention Network for Semantic Segmentation of High-Resolution Remote Sensing Images. *Remote Sens.* **2023**, *15*, 3619. [CrossRef]
56. Wang, C.; Ji, L.Q.; Shi, F.; Li, J.Y.; Wang, J.; Enan, I.H.; Wu, T.; Yang, J.J. Collapsed Building Detection in High-Resolution Remote Sensing Images Based on Mutual Attention and Cost Sensitive Loss. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 8000605. [CrossRef]
57. Liu, D.X.; Wang, Y.R.; Liu, P.X.; Li, Q.Q.; Yang, H.; Chen, D.B.; Liu, Z.C.; Han, G.L. A Multiscale Cross Interaction Attention Network for Hyperspectral Image Classification. *Remote Sens.* **2023**, *15*, 428. [CrossRef]
58. Peng, Y.S.; Zhang, Y.W.; Tu, B.; Li, Q.M.; Li, W.J. Spatial-Spectral Transformer with Cross-Attention for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5537415. [CrossRef]
59. Yang, K.; Sun, H.; Zou, C.B.; Lu, X.Q. Cross-Attention Spectral-Spatial Network for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5518714. [CrossRef]
60. Yang, J.X.; Zhou, J.; Wang, J.; Tian, H.; Liew, A.W.C. LiDAR-Guided Cross-Attention Fusion for Hyperspectral Band Selection and Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5515815. [CrossRef]
61. Roy, S.K.; Sukul, A.; Jamali, A.; Haut, J.M.; Ghamisi, P. Cross Hyperspectral and LiDAR Attention Transformer: An Extended Self-Attention for Land Use and Land Cover Classification. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5512815. [CrossRef]
62. Wu, S.; Li, G.Q.; Deng, L.; Liu, L.; Wu, D.; Xie, Y.; Shi, L.P. *L*1-Norm Batch Normalization for Efficient Training of Deep Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 2043–2051. [CrossRef]
63. Wang, X.L.; Ren, H.E.; Wang, A.C. Smish: A Novel Activation Function for Deep Learning Methods. *Electronics* **2022**, *11*, 540. [CrossRef]

64. Cui, Y.Q.; Xu, Y.F.; Peng, R.M.; Wu, D.R. Layer Normalization for TSK Fuzzy System Optimization in Regression Problems. *IEEE Trans. Fuzzy Syst.* **2023**, *31*, 254–264. [CrossRef]
65. Li, J.J.; Liu, Y.Z.; Song, R.; Li, Y.S.; Han, K.L.; Du, Q. Sal^{2}RN: A Spatial-Spectral Salient Reinforcement Network for Hyperspectral and LiDAR Data Fusion Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5500114. [CrossRef]
66. Khodadadzadeh, M.; Li, J.; Prasad, S.; Plaza, A. Fusion of Hyperspectral and LiDAR Remote Sensing Data Using Multiple Feature Learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 2971–2983. [CrossRef]
67. Liu, Y.; Bioucas-Dias, J.; Li, J.; Plaza, A. Hyperspectral cloud shadow removal based on linear unmixing. In Proceedings of the IGARSS 2017—2017 IEEE International Geoscience and Remote Sensing Symposium, Fort Worth, TX, USA, 23–28 July 2017.
68. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [CrossRef]
69. Ge, H.M.; Wang, L.G.; Liu, M.Q.; Zhao, X.Y.; Zhu, Y.X.; Pan, H.Z.; Liu, Y.Z. Pyramidal Multiscale Convolutional Network with Polarized Self-Attention for Pixel-Wise Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5504018. [CrossRef]
70. Wang, X.H.; Zhu, J.H.; Feng, Y.N.; Wang, L. MS2CANet: Multiscale Spatial-Spectral Cross-Modal Attention Network for Hyperspectral Image and LiDAR Classification. *IEEE Geosci. Remote Sens. Lett.* **2024**, *21*, 5501505. [CrossRef]
71. Ghorbanian, A.; Ahmadi, S.A.; Amani, M.; Mohammadzadeh, A.; Jamali, S. Application of Artificial Neural Networks for Mangrove Mapping Using Multi-Temporal and Multi-Source Remote Sensing Imagery. *Water* **2022**, *14*, 244. [CrossRef]
72. Loshchilov, I.; Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *arXiv* **2016**, arXiv:1608.03983.
73. Zhang, S.Y.; Xu, M.; Zhou, J.; Jia, S. Unsupervised Spatial-Spectral CNN-Based Feature Learning for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5524617. [CrossRef]
74. van der Maaten, L.; Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.

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.