remote sensing

MDPI

Article

A Novel Dual-Encoder Model for Hyperspectral and LiDAR Joint Classification via Contrastive Learning

Haibin Wu [1], Shiyu Dai [1,2], Chengyang Liu [1], Aili Wang [1,*] and Yuji Iwahori [3]

[1] Heilongjiang Province Key Laboratory of Laser Spectroscopy Technology and Application, Harbin University of Science and Technology, Harbin 150080, China
[2] Artificial Intelligence R&D Center, Nuctech Jiang Su Company Limited, Changzhou 213000, China
[3] Department of Computer Science, Chubu University, Aichi 487-8501, Japan
* Correspondence: aili925@hrbust.edu.cn

Abstract: Deep-learning-based multi-sensor hyperspectral image classification algorithms can automatically acquire the advanced features of multiple sensor images, enabling the classification model to better characterize the data and improve the classification accuracy. However, the currently available classification methods for feature representation in multi-sensor remote sensing data in their respective domains do not focus on the existence of bottlenecks in heterogeneous feature fusion due to different sensors. This problem directly limits the final collaborative classification performance. In this paper, to address the bottleneck problem of joint classification due to the difference in heterogeneous features, we innovatively combine self-supervised comparative learning while designing a robust and discriminative feature extraction network for multi-sensor data, using spectral–spatial information from hyperspectral images (HSIs) and elevation information from LiDAR. The advantages of multi-sensor data are realized. The dual encoders of the hyperspectral encoder by the ConvNeXt network (ConvNeXt-HSI) and the LiDAR encoder by Octave Convolution (OctaveConv-LiDAR) are also used. The adequate feature representation of spectral–spatial features and depth information obtained from different sensors is performed for the joint classification of hyperspectral images and LiDAR data. The multi-sensor joint classification performance of both HSI and LiDAR sensors is greatly improved. Finally, on the Houston2013 dataset and the Trento dataset, we demonstrate through a series of experiments that the dual-encoder model for hyperspectral and LiDAR joint classification via contrastive learning achieves state-of-the-art classification performance.

Keywords: hyperspectral image; light detection and ranging (LiDAR); multi-sensor; contrastive learning; contrastive loss

Citation: Wu, H.; Dai, S.; Liu, C.; Wang, A.; Iwahori, Y. A Novel Dual-Encoder Model for Hyperspectral and LiDAR Joint Classification via Contrastive Learning. *Remote Sens.* **2023**, *15*, 924. https://doi.org/10.3390/rs15040924

Academic Editor: Costas Panagiotakis

Received: 7 December 2022
Revised: 3 February 2023
Accepted: 3 February 2023
Published: 7 February 2023

1. Introduction

Hyperspectral remote sensing refers to the science and technology of the acquisition, processing, analysis, and application of remote sensing data with high spectral resolution. Different from multispectral remote sensing, hyperspectral remote sensing can obtain information on surface objects in hundreds of continuous spectrum segments, providing rich spectrum information to enhance the ability to enhance the expression of features [1]. Hyperspectral remote sensing has been widely used in surface classification, target detection, agricultural monitoring, mineral mapping, environmental management, and other fields [2].

Remote sensing image classification is an essential part of hyperspectral remote sensing image processing and application, and its ultimate goal is to assign a unique category identifier to each pixel in the image. In the past decades, a variety of HSI classification methods have been proposed [3,4] and have mainly focused on spectral or spatial–spectral information. For classification based on spectral information, a TabNet with spatial attention (TabNets) was designed for hyperspectral image classification in the study [3]. To

fully explore the spatial features of HSIs, many methods have been proposed, such as an encoder–decoder with a residual network (EDRN) [4], a study that combines hyperspectral and panchromatic remote sensing images to extract the representative deep features of codes. However, the high-dimensional characteristics, high correlation between bands, and spectral mixing make the classification of hyperspectral remote sensing images face significant challenges [5]. Thanks to the development of remote sensing technology, it is now possible to measure different aspects of the same object on the Earth's surface [6]. Hyperspectral data are easily disturbed by environmental factors such as clouds, shadows, etc., which can easily lead to the confusion of information. For example, if building roofs and roads are made of concrete, it is difficult to distinguish them using hyperspectral data alone, because their spectral responses are similar. However, light detection and ranging (LiDAR) uses pulsed lasers to measure distances and is an active remote sensing method [7,8]. It is not susceptible to weather conditions and not only provides height and shape information of the scene but also has excellent accuracy and flexibility [9,10] and can accurately classify these two categories. Conversely, LiDAR data cannot distinguish two roads composed of different materials (e.g., asphalt and concrete) with the same height [11,12]. Therefore, the two kinds of data are deeply integrated to realize the complementary advantages of multi-sensor remote sensing, break through the performance bottleneck of single remote sensing data (such as "different objects with the same spectrum" or "same objects with different spectrum"), and finally achieve the purpose of improving the accuracy of object classification [13,14].

Most traditional classification models first perform feature extraction on multi-sensor data and then distinguish them through a classifier [15]. Among the feature extraction methods, knowledge-guided feature extraction is based on the understanding of spectral features to perform mathematical operations on relevant bands to obtain deep-level information. However, expert knowledge is often hard to obtain. Furthermore, traditional classification methods rely heavily on hand-designed features, which limit the representation of models [16].

With the development of deep-learning techniques, convolutional neural networks (CNNs) [17–23] have been widely used in computer vision tasks, such as image classification [24,25], object detection [26], semantic segmentation [17], etc. Research methods based on the CNN model have been widely used in the field of remote sensing image classification and have become the mainstream methods in this field [27,28]. Convolutional neural networks have shown excellent feature extraction capabilities in this field and are gradually replacing artificial feature-based methods.

For example, in [29], a Two-Branch CNN combines the separately extracted spatial and spectral features of HSIs with those extracted by LiDAR. EndNet (encoder–decoder network) [30] is a deep encoder–decoder network architecture that reconstructs multi-sensor inputs by encoding and decoding fused features via an autoencoder. In [31], multimodal deep learning middle fusion (MDL-Middle) is an intermediate fusion CNN model.

Based on all these advanced approaches mentioned above, we have the following reflections. First, the ability to fully integrate different models is critical, and this can be fully reflected by the complementary nature of HSIs and LiDAR features. This is because multi-sensor data are naturally correlated. If this complementarity is fully exploited in the encoding process, the extracted multi-sensor encoding will be more robust and comprehensive. Brain studies also reaffirm that the human brain begins to interact with multi-sensor data in the primary perceptual cortex.

Second, the design of feature extraction networks specifically for different sensor data is often critical and needs to fully take into account the characteristics of different sensor domains, which is often important for downstream tasks.

Therefore, this paper takes two aspects and explores their solutions.

Self-supervised learning has recently emerged as an effective approach, self-supervised learning in the multimodal field is very effective and can achieve downstream task performance comparable to supervised pre-training in tasks such as action recognition, in-

formation retrieval, and video question answering. For example, many self-supervised methods [32–36] exploit contrastive objectives (e.g., comparison) to facilitate multimodality such as visual–linguistic learning. For example, MoCo [34] further improves this scheme by storing the representation of the future quantum encoder in a dynamic dictionary with queues. MoCov2 [36] borrows the multilayer perceptron (MLP) head design and shows significant improvements.

From the success of self-supervised learning within the multimodal domain, we can also realize that self-supervised learning can be used in the multi-sensor domain as a way to help the deep information fusion of features from different sensors, which can further improve the performance of multi-sensor downstream tasks.

To solve the second aspect, we investigated a large number of state-of-the-art networks and their improved algorithms. Recently, the application of the ConvNeXt [37] network in the visual direction has become a hot spot. On the basis of maintaining the CNN structure, the ConvNeXt network draws on the design concepts of methods such as the Swin Transformer [38]. Swin Transformer is a landmark work in the transformer direction, which for the first time demonstrates that transformers can be used as general-purpose vision backbones and achieve state-of-the-art performance in a range of computer vision tasks. ConvNeXt uses a larger kernel size to simulate long-distance modeling capabilities while maintaining the local sensitivity of the CNN, ensuring the global information of the network. However, the spectrum of an HSI is a kind of sequence data, which usually contains hundreds of spectral bands. Through the feature extraction advantages of the local information and global information of the ConvNeXt network, it can not only complete the extraction of global spectral–spatial information but also overcome the mixed pixel band problems such as decreased accuracy.

Refocusing on the convolutional neural network, the receptive field is used to represent the size of the range of perception of the original image by neurons at different positions within the network. The larger the value of the neuron's receptive field, the larger the range of the original image it can touch, which also means that it may contain more global and higher semantic features, while the smaller the value, the more features it contains, tending to be localized and contain more detail. So, the receptive field is very important for the network.

The introduction of dilated convolution [39] introduces a dilation rate parameter in the convolution kernel, and the dilation rate defines the spacing between the convolution kernels (where the spacing is defined as r). In other words, the dilation convolution is similar to the traditional convolution, but the number of weights in the dilation convolution kernel remains the same. Only the weights of the convolution kernels are spaced by r positions, i.e., the kernels of the dilation convolution layer are sparse. This allows the convolution filter to obtain a larger perceptual field without reducing the spatial resolution or increasing the kernel size to improve the recognition of downstream tasks in the network. The traditional convolutional neural network can be regarded as cascading a large number of convolution operators to encode the input information, representing the characteristics of different frequency components of the input sample. However, there is no effective fusion process between frequencies, and the interaction between frequencies is very critical for the encoding of LiDAR information. OctaveConv [40] decomposes the input convolutional feature map into two sets of feature maps with different spatial frequencies and processes different convolutions at corresponding frequencies, which helps each layer to obtain a larger receptive field to capture more contextual information.

The main contributions are summarized as follows:

1. We introduce a multi-sensor pair training framework for the HSI-LiDAR classification task. Our multi-sensor training framework can exploit intrinsic data properties in each modality and simultaneously extract semantic information from cross-modal correlations. It can not only encode the two modalities independently to capture more modality-specific information but also complete the deep fusion of the two sensors'

information and learn the alignment between different modalities and learn deep fusion for HSI-LiDAR classification tasks;

2. It is well known that information is conveyed at different frequencies, where higher frequencies are typically used for fine detail encoding and lower frequencies are typically used for global structure encoding. The Digital Surface Model (DSM) of LiDAR has rich depth information, that is, high- and low-frequency information. We propose a new LiDAR encoder network structure with Octave convolution. The output maps of a convolutional layer can also be factorized and grouped by their spatial frequency. OctaveConv focuses on reducing the spatial redundancy in CNNs and is designed to replace vanilla convolution operations. In this way, the high- and low-frequency information of the DSM is fully utilized from the aspect of feature extraction;
3. Due to the spectral redundancy and low spatial resolution of HSIs, we propose the Spectral-Aware Trident network in parallel and the ConvNeXt network in series. In both networks, dilated convolution that can improve the receptive field is used. Recently, the application of the ConvNeXt network in the visual field has become a hot spot. On the basis of maintaining the CNN structure, the ConvNeXt network borrows the design concepts of Transformer and other methods. While maintaining the local sensitivity of the CNN, a larger kernel size is used to simulate the long-range modeling ability, which ensures the global information of the network. The spectrum of an HSI is a sequence of data that typically contains hundreds of bands. Through the feature extraction advantages of the local information and the global information of the ConvNeXt network, we can not only complete the extraction of global spectral–spatial information but also overcome the problems of accuracy degradation caused by mixed pixels.
4. In the training method of the network, we show the use of a stagewise training strategy, which trains the HSI branch, LiDAR branch, and HSI-LiDAR classification tasks in stages. The method of training the HSI and LiDAR branches in stages can provide better model parameter initialization for the HSI-LiDAR classification model, which usually leads to better generalization performance and accelerates convergence on this downstream task.

2. Methods

2.1. Overview

Figure 1 shows the overall architecture of the proposed HSI-LiDAR joint classification algorithm. The training method adopts the stagewise training strategy. Firstly, HSI and LiDAR data are preprocessed and online multi-scale data enhancement is performed separately according to the given HSI-LiDAR sample pairs. The HSI/LiDAR downstream classification task that is consistent with the final HSI-LiDAR classification task is selected as the supervision to train the HSI and LiDAR branches separately. The respective depth representations can be extracted for different sensor data. Secondly, the HSI-LiDAR (HL) deep coding model is introduced to obtain the depth fusion features between multi-sensor data. The feature representation of the multi-sensor feature encoding model is also enhanced by using HSI-LiDAR self-supervised contrastive learning (see Figure 1, top right). Finally, we use the pre-trained model obtained from the training. The features are extracted by the joint HSI encoder, LiDAR encoder, and HL fusion encoder. We selected the HSI/LiDAR downstream classification task and HSI-LiDAR self-supervised contrastive learning as auxiliary supervision to jointly train the final HSI-LiDAR classification task.

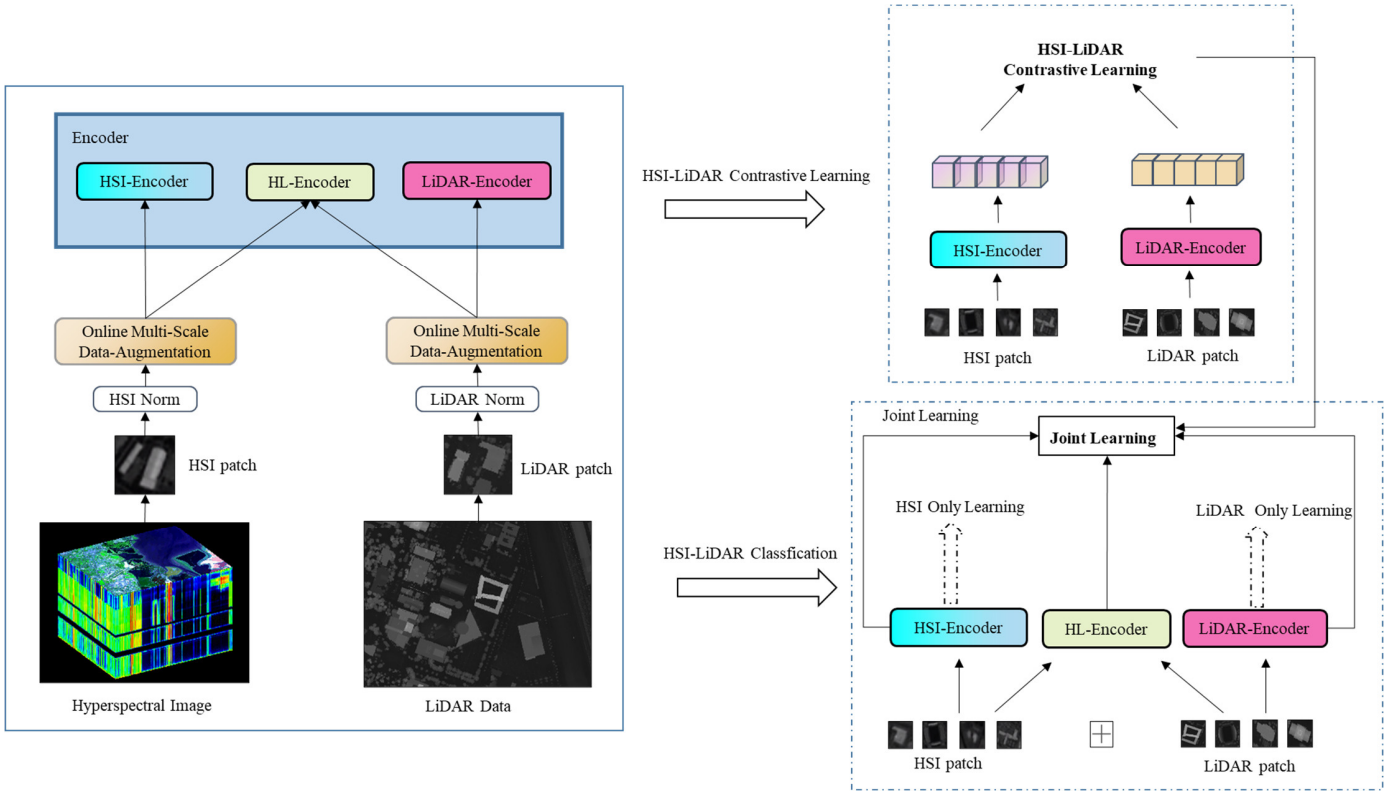


Figure 1. The proposed classification framework of joint HSI-LiDAR classification.

2.2. HSI and LiDAR Encoder

In this paper, we perform online data augmentation on HSI and LiDAR data to generate new and diverse instances in each training batch, enriching positive and negative sample features in a training batch to improve the performance and results of deep-learning models. The data augmentation methods in this paper are normalization, random noise addition, multiscale transformation, and random horizontal mirroring. Among them, multiscale transformation is a multiscale transformation operation on the input samples: in this paper, the training samples are multiscale-transformed with four window sizes: 11×11, 9×9, 7×7, and 5×5.

2.2.1. HSI Encoder

Hyperspectral remote sensing technology can obtain more spectral–spatial information, and because the number of imaging channels is greatly increased, the ability to distinguish and identify ground objects is greatly improved. Therefore, improving the effectiveness of hyperspectral image feature extraction is crucial to improve the overall classification accuracy of images in complex scenes.

To enhance the effectiveness of hyperspectral image feature extraction, there are two ideas to design the HSI encoder in terms of network design.

1. Trident-HSI

In view of the spectral redundancy and low spatial resolution of HSIs, the Spectral-Aware Trident network is proposed. As shown in Figure 2, a three-branch structure is introduced in the network, and the three-branch structure can group input features. Each branch uses dilated convolutions with different expansion rates in the second Conv layer but has the same parameter value. Here, the dilated Conv layers with an expansion rate of n are abbreviated as Dn-Conv layers.

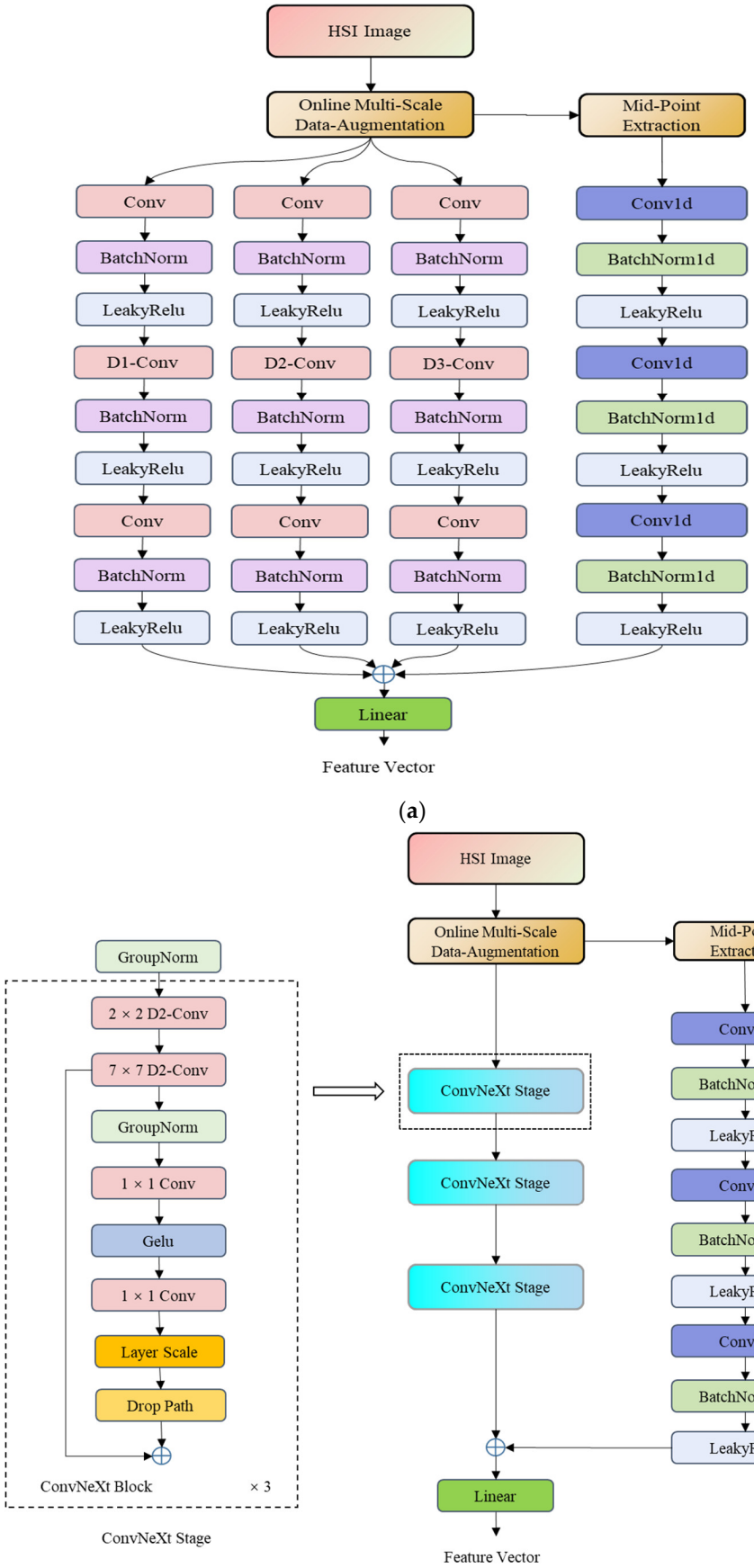


Figure 2. Two architectures of HSI networks. (**a**) Trident-HSI. (**b**) ConvNeXt-HSI.

In this way, different branches have different perceptual field sizes to be responsible for different scales of targets. This improves the spatial resolution of the HSI, increases the effectiveness of network feature extraction, and reduces the redundancy of HSI hyperspectral information. Finally, the feature extraction is performed for the median points of the HSI patch. The median point feature extraction branch consists of three consecutive one-dimensional Conv layers, one-dimensional BatchNorm layers, and LeakyRelu layers. By encoding the median points with the features extracted from the Trident branch, the information representation of the HSI encoder is further enhanced and the classification accuracy is improved.

Since the HSI patch extraction method is a multi-scale window extraction on the true value points, the median point has the strongest information. The median point features are merged with those extracted by the Trident branch to further enhance the performance of the HSI encoder. The formula is as follows:

$$F_{Trident}(X;K) = \sum_i f_i(X;K) + g_{mid\_point}(X;K) \quad (1)$$

The feature map obtained by the first convolutional layer is defined as $X \in R^{C\times H\times W}$, where C is the number of channels of the feature map, and H and W are the height and width of the feature map. $f_i(X;K)$ denotes the ith Trident parallel branch, K is the weight of the neural network, and $g_{mid_{point}}(X;K)$ is the median point feature extraction branch.

The median point feature extraction branch is composed of three groups of continuous modules, including a one-dimensional Conv layer, a one-dimensional BatchNorm layer, and a LeakyRelu layer. The feature encoding of the median point further enhances the information expression of the network and improves the classification accuracy.

2. ConvNeXt-HSI

The Trident network uses a three-branch parallel form to improve the effectiveness of network feature extraction, while the ConvNeXt network uses a serial form to improve the effectiveness of network feature extraction. The Stem layer reduces the redundancy of HSI hyperspectral information, and the ConvNeXt network uses a large convolution kernel size of 7×7 to increase the network receptive field. Higher-level networks can extract richer HSI semantic information, which is very critical for hyperspectral image classification. Therefore, the HSI encoder in this paper uses a concatenated ConvNeXt network to improve the accuracy of HSI feature extraction. As shown in Figure 2b, we improved ConvNeXt into a ConvNeXt-HSI network that is more suitable for HSI classification tasks. First, we reduced the depths of ConvNeXt from level 4 to level 2 to reduce the risk of overfitting due to the network being too deep. The number of convolution kernels in the widths of ConvNeXt ensures network performance while reducing the number of network parameters and improving the operation speed. Finally, in the first and second Conv layers of each stage, dilated convolution is used to ensure HSI spatial resolution. Similar to the Trident network, we also extract the median points of the HSI patch to enhance the ConvNeXt-HSI network features. The formula is as follows:

$$F_{ConvNext}(X;K) = H_i(X;K) + g_{mid\_point}(X;K)$$
$$H_i(X;K) = ConvNextStage(H_{i-1}(X;K)) \quad (2)$$

The feature map obtained by the first convolutional layer is defined as $X \in R^{C\times H\times W}$, where C is the number of channels in the feature map, H and W are the height and width of the feature map, $H_i(X;K)$ denotes the ith serial form branch with parameter K, $g_{mid_{point}}(X;K)$ is the median point feature extraction branch with parameter K, and K is the weight of the neural network.

The ConvNeXt stage consists of GroupNorm, a 2×2 dilated Conv layer with a dilated rate of 2, and three consecutive ConvNeXt Blocks. The ConvNeXt Block consists of a 7×7 dilated Conv layer with a dilated rate of 2, GroupNorm, a 1×1 Conv layer, Gelu, a 1×1 Conv layer, a Layer Scale layer, and a Drop Path layer. The 7×7 Conv is the depthwise convolution, and the depthwise convolution performs a self-attention mechanism in each convolution channel, which can obtain the spatial information in the convolution channel

and enhance the information expression of the network. The Layer Scale layer multiplies the input feature layer by the trainable parameters, and the number of training parameter elements is the same as the number of channels in the feature layer; that is, scaling the data of each channel can not only accelerate the convergence in a larger depth network but also improve network accuracy. The Drop Path layer is a regularization method that randomly "deletes" the multi-branch structure in the deep-learning model, which can ensure the depth of the network and reduce the probability of overfitting.

Like the Trident network, we also extract the median point of the HSI patch to enhance the features of the ConvNeXt-HSI network. The network structure is also consistent with the Trident network. The feature encoding of the median point further enhances the information expression of the network and improves the classification accuracy.

2.2.2. LiDAR DSM Encoder

The DSM of LiDAR includes ground elevation models of object heights such as buildings, bridges, and trees on the ground, with rich elevation information and surface information. From the DSM image itself, the elevation information and surface information means that the image contains rich high- and low-frequency information. The traditional convolutional neural network can be regarded as cascading a large number of convolutional operators, which can be seen as cascading different frequency components in the input samples to obtain feature encoding. However, traditional convolutional operators do not have an effective fusion process between high and low frequencies, and it is known that high frequencies focus on local details and low frequencies focus on global structure. The interaction between high and low frequencies allows the high and low frequencies to complement each other and extract richer local and global information, which is crucial for the encoding of LiDAR information. The principle of Octave convolution is as shown in Figure 3. The green arrows correspond to the information update between high frequency–high frequency and low frequency–low frequency, while the blue arrows represent the information exchange between the two frequency bands. Finally, the two frequency bands' information is fused to obtain a multi-frequency feature representation, which can not only enrich the feature representation but also reduce feature redundancy.

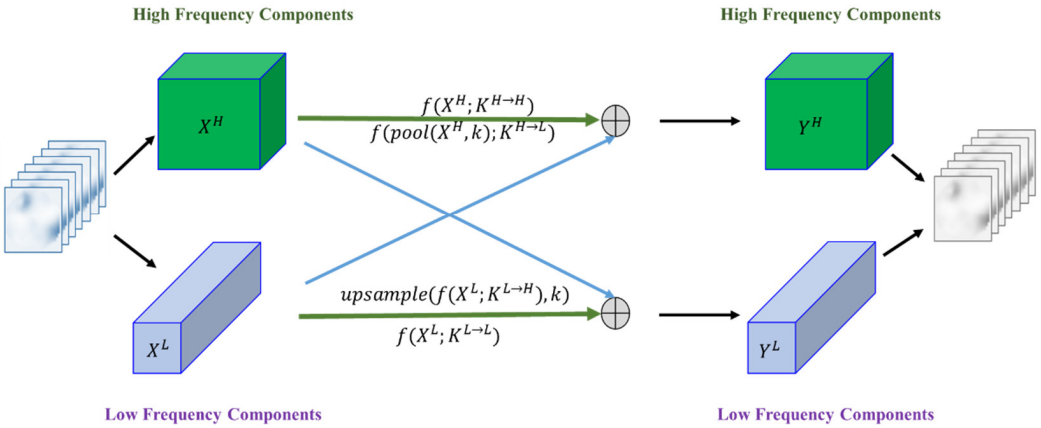


Figure 3. The principle of Octave convolution.

The following are the details of Octave convolution. The feature map obtained by the first convolutional layer is defined as $X \in R^{C\times H\times W}$, where C is the number of channels in the feature map, and H and W are the height and width of the feature map. The feature map X is first divided into two parts by a convolution operation, $\{X^H, X^L\}$, where X^H is the relatively high frequency and X^L is the lower frequency. Here, the channel of the output feature map is $\frac{1}{ratio}$; that is, α. The number of high-frequency channels is $R^{\alpha\times C\times H\times W}$,

and the number of high-frequency channels is $R^{(1-\alpha)\times C\times H\times W}$. We set $\alpha = 0.5$. It can be formulated as:

$$Y^H = f(X^H; K^{H\to H}) + upsample(f(X^L; K^{L\to H}), k)$$
$$Y^L = f(X^L; K^{L\to L}) + f(pool(X^H, k); K^{H\to L}) \quad (3)$$

where $f(X; K)$ denotes a convolution with parameter K, $pool$ (X, k) is an average pooling operation with kernel size $k \times k$ and stride k, and $upsample$ (X, k) is an up-sampling operation by a factor of $k = 2$ via nearest interpolation.

Figure 4 shows the overall architecture of the LiDAR encoder.

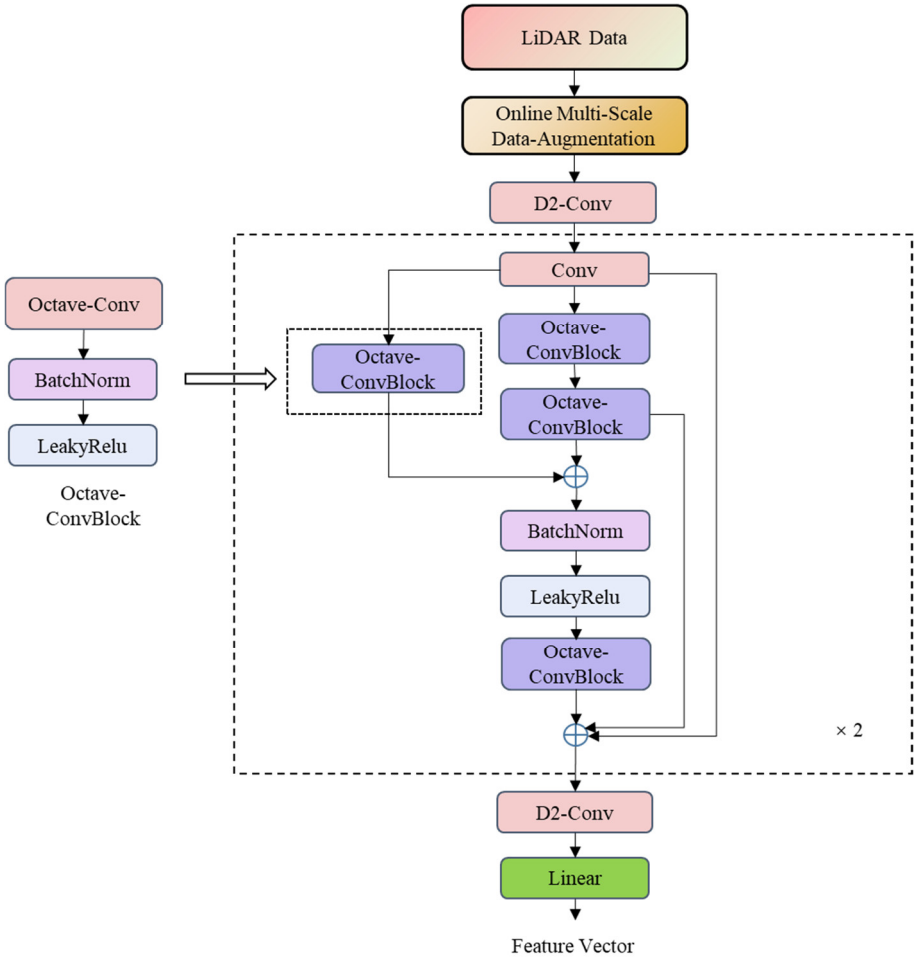
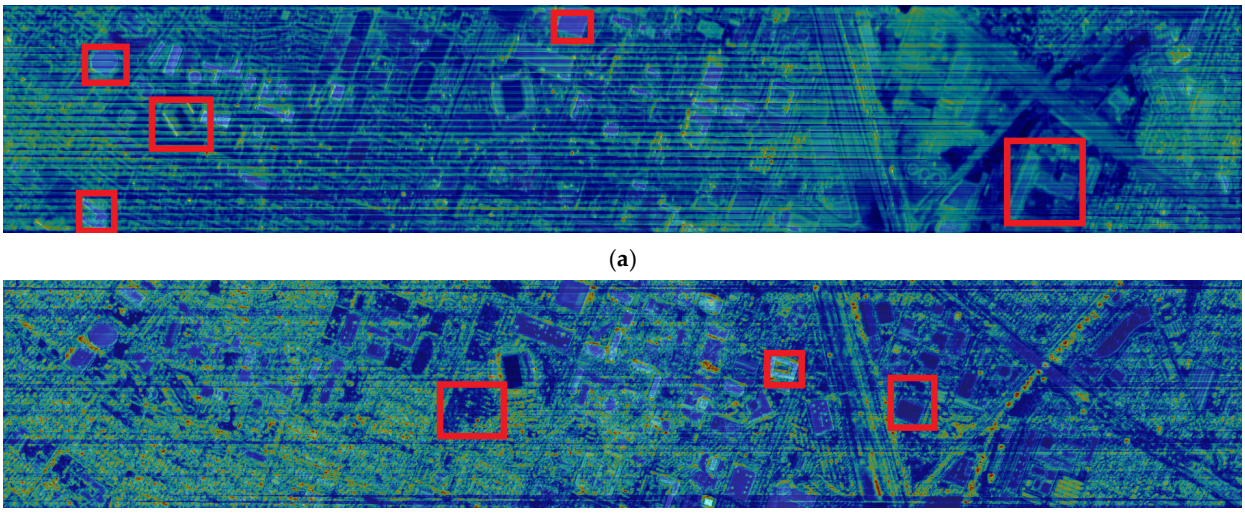


Figure 4. The architecture of OctaveConv-LiDAR DSM encoder.

The encoder consists of a Conv layer, a BatchNorm layer, a LeakyRelu layer, and multiple OctaveConv Blocks. In order to ensure the receptive field of the network, the first Conv layer and the last Conv layer are dilated Conv operations with a dilated rate of 2. An OctaveConv Block is composed of OctaveConv, a BatchNorm layer, and a LeakyRelu layer, which can extract rich high- and low-frequency information. We use skip-layer connections in the middle of the network, such as merging the second Conv layer with the features of the second OctaveConv Block and the third concatenated OctaveConv Block. Through the joint learning of low-, medium-, and high-level features, the information flow of deep and shallow networks can better improve the representation of semantic and spatial information.

2.2.3. Feature Visualization and Analysis

To further analyze the accuracy of the features extracted by ConvNeXt-HSI and OctaveConv-LiDAR, we visualize the features in the last convolutional layer of each of the two encoders, as shown in Figure 5.



(**a**)

(**b**)

Figure 5. Feature visualization. (**a**) ConvNeXt-HSI feature map. (**b**) OctaveConv-LiDAR.

Figure 5a,b show the features of the ConvNeXt-HSI and OctaveConv-LiDAR encoders. It can be seen that with the ConvNeXt-HSI encoder, the spatial-spectral information of the HSI is fully utilized. As shown in the red box in Figure 5a, the feature map response is larger for features such as grass and house roof fields, and a clearer feature response can be obtained due to the rich HSI spectral information. In Figure 5b, it can be seen that the features have a stronger response for regions in the image with sharp high- and low-frequency changes, such as the edges of the features, by the OctaveConv-LiDAR encoder, as shown in the red box in Figure 5b.

2.3. HSI-LiDAR Contrastive Learning

During the training process, HSI-LiDAR contrastive learning was introduced for N HSI-LiDAR pairs in the same batch, and the outputs were obtained features from the HSI encoder and LiDAR encoder, respectively. After the linear operation, dropout, and normalization operation, we obtained HSI vectors $\{h_i^v\}_{i=1}^N$ and LiDAR vectors $\{l_i^w\}_{i=1}^N$ in a training batch to compute HSI-to-LiDAR and LiDAR-to-HSI similarities, which can simultaneously take into account the respective modal depth encoding of HSI and LiDAR and the deep hybrid encoding between the two modalities. HSI-LiDAR contrastive learning can be formulated as:

$$s_{i,j}^{h2l} = {h_i^v}^T l_j^w, \quad s_{i,j}^{l2h} = {l_i^w}^T h_j^v \tag{4}$$

$$l_i^{(h\to l)} = -log\frac{\exp(s_{i,i}^{h2l}/\tau)}{\sum_{j=1}^N \exp(s_{i,j}^{h2l}/\tau)}, \quad l_i^{(l\to h)} = -log\frac{\exp(s_{i,i}^{l2h}/\tau)}{\sum_{j=1}^N \exp(s_{i,j}^{l2h}/\tau)} \tag{5}$$

where $s_{i,j}^{h2l}$ represents the HSI-to-LiDAR similarity in the ith HSI sample and the jth LiDAR sample in the sample pair, $s_{i,j}^{l2h}$ represents the LiDAR-to-HSI similarity in the ith LiDAR sample and the jth HSI sample in the sample pair, and $\tau \in R$ represents a temperature parameter. We set $\tau = 2$. We used softmax cross-entropy loss to compute $l_i^{(h\to l)}$ (HSI-to-LiDAR) and $l_i^{(l\to h)}$ (LiDAR-to-HSI) similarities during the traning phase.

2.4. Stagewise Training Strategy

In order to provide better model parameter initialization for the HSI-LiDAR classification model, we used a staged training method, as shown in Figure 6. Firstly, train the HSI (HSI-only learning) and freeze the parameters of the LiDAR encoder. After the HSI encoder training is completed, the parameters of the HSI encoder are frozen to optimize the LiDAR encoder (LiDAR-only learning). The losses of the two encoders are both softmax cross-entropy losses. Finally, the HSI encoder and LiDAR encoder are used to pre-train the model parameters to finetune the final HSI-LiDAR classification model. Our final joint-training losses are multi-weight losses, including the softmax cross-entropy loss of the HSI, the softmax cross-entropy loss of LiDAR, and the contrastive loss of HSI-LiDAR. These three losses are used as auxiliary losses. The main loss is the softmax cross-entropy loss of HSI-LiDAR, written as follows:

$$\begin{aligned} L &= \frac{1}{N}\sum_{i=1}^{N}\left(\lambda_1 l_{contrast} + \lambda_2 l_{CE-HSI} + \lambda_3 l_{CE-LiDAR} + \lambda_4 l_{CE-all}\right) \\ l_{CE} &= -\sum_{k=1}^{C} y_k log(S_k), S_k = softmax(X_k) = \frac{e^{x_k}}{\sum_{i=1}^{C} e^{x_i}} \end{aligned} \tag{6}$$

where $l_{contrast} = (l_i^{(h\to l)} + l_i^{(l\to h)})/2$, $\lambda_{1\sim4} \in [0,1]$ is a scalar weight.

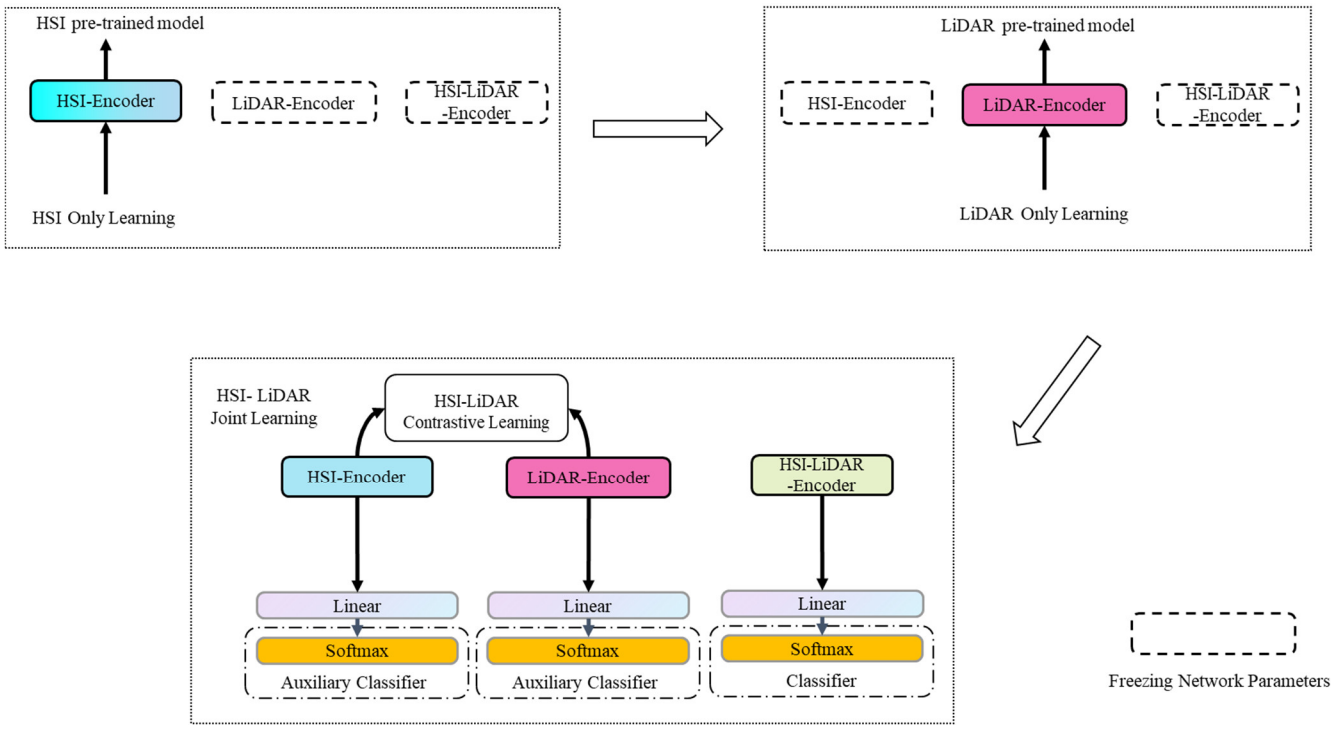


Figure 6. The framework of our proposed stagewise training strategy.

3. Results

To evaluate the performance effectiveness of the proposed model, we used two different datasets for evaluation: Houston2013 and Trento. All deep models were implemented in the Pytorch 1.9 framework. All experiments were carried out in the same hardware environment; that is, Ubuntu16.04, Tesla K80 device.

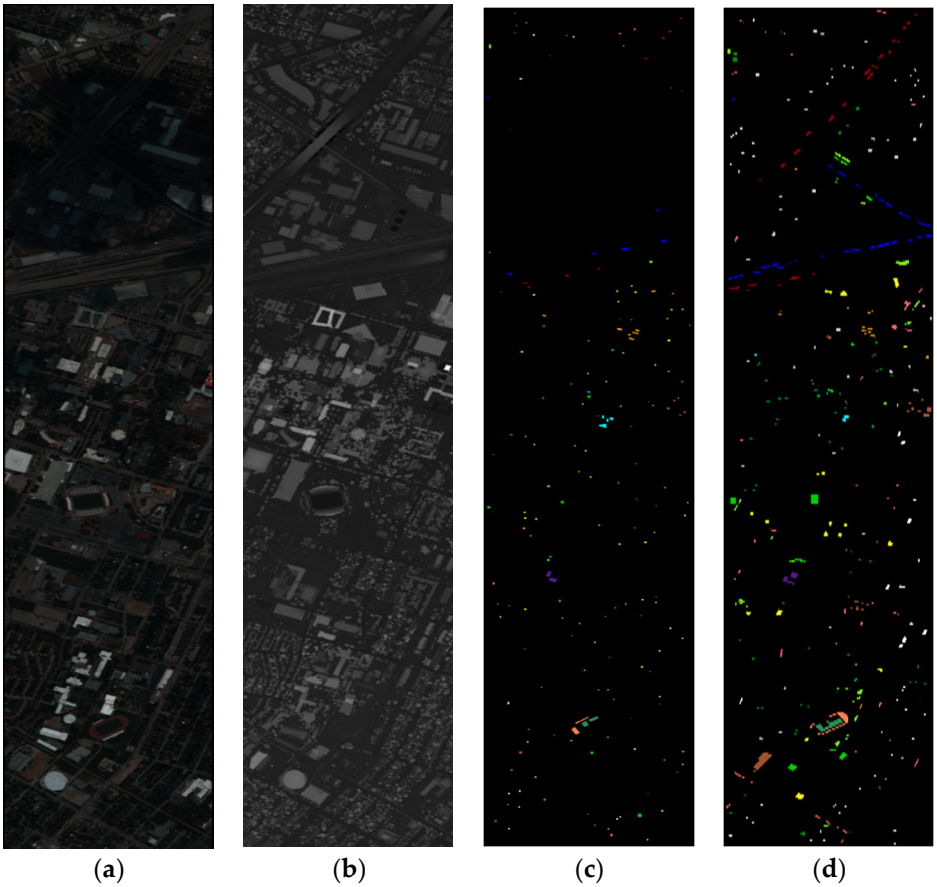
3.1. Experimental Datasets Description

Houston 2013 dataset: This dataset involves two datasets—hyperspectral images and a LiDAR-derived DSM, both consisting of 349 × 1905 pixels with the same spatial resolution (2.5 m). The data were acquired by the NSF-funded Center for Airborne Laser Mapping

(NCALM) over the University of Houston's 2013 campus and the neighboring urban area. The HSI has 144 spectral bands in the 380 nm to 1050 nm region, including 15 classes. Table 1 lists the number of samples of different classes and the color of each class. Figure 7 gives the visualization results of the Houston2013 dataset. These data and reference classes can be obtained online from the IEEE GRSS website (http://dase.grss-ieee.org/ (accessed on 10 September 2022)).

Table 1. Houston2013 dataset: the numbers of training and testing samples for each class.

Class	Class Name	Train Num	Test Num	Color
C1	Healthy Grass	198	1053	
C2	Stressed Grass	190	1064	
C3	Synthetic Grass	192	505	
C4	Trees	188	1056	
C5	Soil	186	1056	
C6	Water	182	143	
C7	Residential	196	1072	
C8	Commercial	191	1053	
C9	Road	193	1059	
C10	Highway	191	1036	
C11	Railway	181	1054	
C12	Parking Lot1	192	1041	
C13	Parking Lot2	184	285	
C14	Tennis Court	181	247	
C15	Running Track	187	473	
-	Total	2832	12,197	



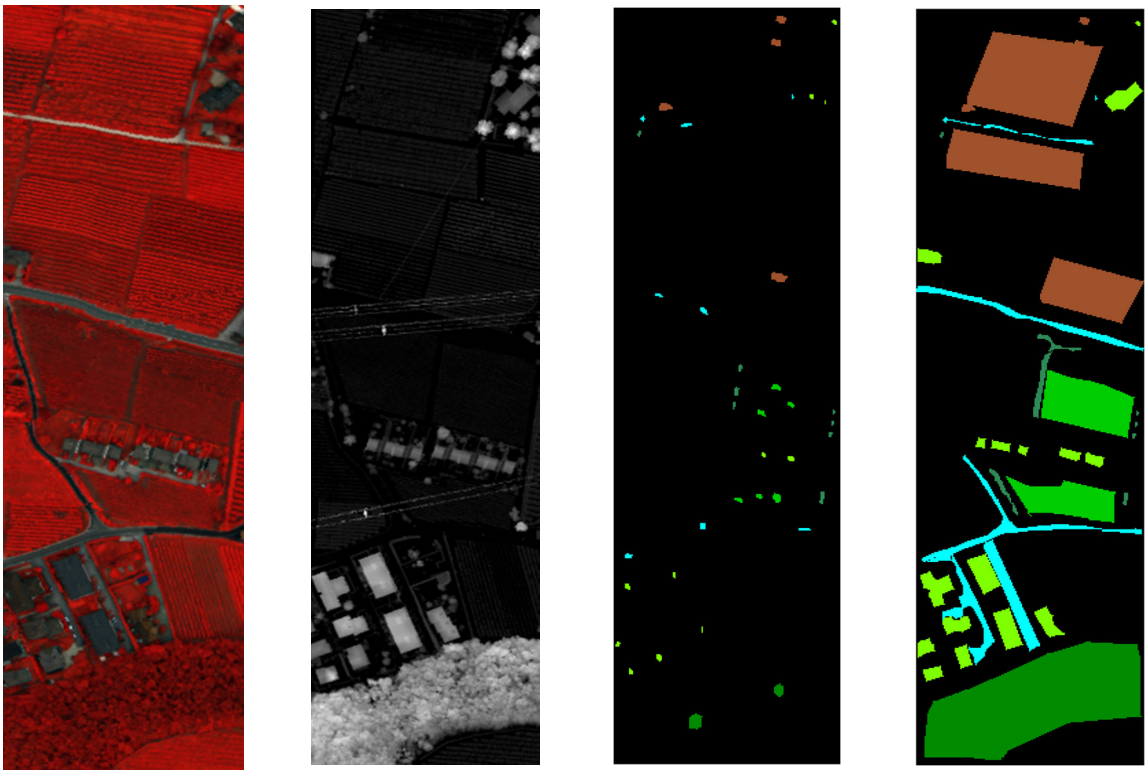
(**a**) (**b**) (**c**) (**d**)

Figure 7. The visualization of the Houston2013 dataset. (**a**) Pseudo color map of an HSI. (**b**) DSM of LiDAR. (**c**) Training samples map. (**d**) Testing samples map.

Trento dataset: This dataset involves two datasets—hyperspectral images and LiDAR-derived DSM, both consisting of 600 × 166 pixels with the same spatial resolution (1 m). The data were acquired by the AISA Eagle sensor, and the LiDAR DSM was produced using first- and last-point cloud pulses obtained by the Optech ALTM 3100EAsensor. The HSI has 63 spectral bands covering the 402.89 to 989.09 nm region and includes six classes. Table 2 lists the number of samples of different classes and the color of each class. Figure 8 gives the visualization results of the Trento dataset.

Table 2. Trento dataset: the numbers of training and testing samples for each class.

Class	Class Name	Train Number	Test Number	Color
C1	Apples	129	3905	
C2	Buildings	15	2778	
C3	Ground	105	374	
C4	Woods	154	8969	
C5	Vineyard	184	10,317	
C6	Roads	122	29,395	
-	Total	819	55,738	



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

Figure 8. The visualization of the Trento dataset. (**a**) Pseudo color map of an HSI. (**b**) DSM of LiDAR. (**c**) Training samples map. (**d**) Testing samples map.

3.2. Experimental Setup

The proposed network was implemented on the Pytorch platform. The models were trained on the training set by randomly dividing the original training set into a training set and a validation set in each epoch. The ratio of training set to validation set was 8:2. In the training phase, we used an SGD optimizer with a weight decay of 1×10^{-4}, a momentum of 0.9, and a batch size of 64 on a NVIDIA TESLA K80 GPU. We used the "step" learning rate strategy. Then, the HSI encoder and LiDAR encoder were trained with an initial learning rate of 0.03 and 4×10^{-5}, respectively, and, finally, the joint HSI and LiDAR encoder was trained with an initial learning rate of 0.01. All training epochs were 100.

In terms of details, we normalized all encoder feature vectors before calculating their dot products in the contrastive losses, where τ was set as 0.07. For trade-off parameters in the final loss, we set λ_1 as 0.5, λ_2 as 0.2, λ_3 as 0.5, and λ_4 as 1.0. All the auxiliary losses were less than 1.0. In the auxiliary losses, after pre-training the encoder, we found that the classification performance of the LiDAR encoder was not good enough compared to the HSI. In order to degrade the performance of the LiDAR encoder, we set the auxiliary classification loss of LiDAR to 0.5 and the auxiliary classification loss of HSI to 0.3. We used uniform hyperparameters for all datasets.

To evaluate the performance effectiveness of the proposed model, we used two different datasets for evaluation and evaluated the effectiveness of the model through four metrics: the overall accuracy (OA), average accuracy (AA), and Kappa coefficient. The overall accuracy (OA) defines the ratio of all correctly classified pixels to the total number of pixels in the test set. Average accuracy (AA) is the average probability that the accuracies for each class of elements are summed and divided by the number of classes. The Kappa coefficient was also used to evaluate the classification accuracy, checking the consistency of the remote sensing classification result map with the ground-truth map.

3.3. Experimental Results

To demonstrate the effectiveness of the proposed model, several representative multi-sensor joint classification model methods were selected for comparison experiments with the proposed model, including Two-Branch CNN, EndNet, and MDL-Middle. Here, Two-Branch CNN performs feature fusion by combining the spatial and spectral features extracted from the HSI branch with the LiDAR data features extracted from the cascaded network. EndNet is a deep encoder–decoder network architecture that fuses multi-sensor information by enhancing fused features. MDL-Middle performs multi-sensor feature fusion on the middle layer of the CNN model. We also compared single-sensor classification models: Trident-HSI, CNN-LiDAR, ConvNeXt-HSI, and OctaveConv-LiDAR, Trident-HSI, ConvNeXt-HSI, CNN-LiDAR, and OctaveConv-LiDAR. In order to ensure the validity of the comparative experiments, the verification data were uniformly used as the Houston2013 and the Trento dataset, and the training set and test set of each dataset were completely consistent.

3.3.1. Classification Results of the Houston2013 Dataset

Table 3 shows the detailed classification results of eight models in terms of OA, AA, and Kappa coefficients on the Houston2013 dataset. The best results are shown in bold. As can be seen from Table 3, our proposed method shows obvious improvement in OA, AA, and Kappa coefficients compared with Two-Branch CNN, EndNet, and MDL-Middle. The classification performance of Soil, Road, Railway, Parking Lot1, Tennis Court, and Running Track are all better than those of these three models, especially the recognition accuracy of 88.41% of Healthy Grass and 94.97% of Railway. Compared with these three models, the maximum is 6.83% and 11.19% improvement.

The following factors are also evident from Table 3. Firstly, all single-sensor classification performance is lower than multi-sensor classification. The performance using only HSI data is significantly higher than that of only LiDAR data. For example, in the Houston2013 dataset, the OctaveConv-LiDAR classification model has an OA of 67.58%, AA of 65.29%, and Kappa of 64.92%. These three metrics are all lower than the ConvNeXt-HSI classification model performance, which is 87.12%, 88.17%, and 86.02%, respectively. When combined with dual-sensor joint classification, the OA increased to 88.14%, AA increased to 88.14%, and Kappa increased to 87.16%. Secondly, the performance of the ConvNeXt-HSI and OctaveConv-LiDAR classification models proposed in this paper is better than the corresponding Trident-HSI and CNN-LiDAR classification models. Among them, the OA, AA, and Kappa of ConvNeXt-HSI were higher than that of Trident-HSI, at 5.77%, 6.14%, and 6.17%. The OA, AA, and Kappa of OctaveConv-LiDAR are higher than those of CNN-LiDAR, at 4.41%, 4.69%, and 4.71%.

Table 3. Classification accuracy of different models on the Houston2013 dataset.

Class	Two-Branch	EndNet	MDL-Middle	Trident-HSI	CNN-LiDAR	ConvNeXt-HSI	OctaveConv-LiDAR	Proposed
C1	83.1	81.58	83.1	82.72	52.42	83.1	46.63	88.41
C2	84.1	83.65	85.06	84.4	35.06	84.77	54.79	81.3
C3	100	100	99.6	98.02	83.96	99.8	86.14	100
C4	93.09	93.09	91.57	92.99	82.95	92.9	81.63	99.53
C5	100	99.91	98.86	99.91	44.03	100	47.72	100
C6	99.3	95.1	100	93.71	64.34	96.5	71.33	97.2
C7	92.82	82.65	97.64	82.28	88.06	82.74	91.98	91.98
C8	82.34	81.29	88.13	70.85	94.87	76.63	87.84	85.28
C9	84.7	88.29	85.93	78.09	48.63	83.76	64.02	90.46
C10	65.44	89	74.42	41.89	52.7	64.38	56.27	75.1
C11	88.24	83.78	84.54	61.86	87.95	91.75	87.86	94.97
C12	89.53	90.39	95.39	96.73	27.28	97.79	35.45	96.54
C13	92.28	82.46	87.37	76.81	68.77	82.46	70.18	91.23
C14	96.76	100	95.14	100	78.14	97.57	80.57	100
C15	99.79	98.1	100	100	67.44	97.25	84.57	98.31
OA (%)	87.98	88.52	89.55	81.35	63.17	87.12	67.58	91.37
AA (%)	90.11	89.95	91.05	81.76	60.6	88.17	65.29	91.33
K × 100	86.98	87.59	87.59	79.83	60.21	86.02	64.92	90.64

To sum up, the method proposed in this paper is better than all other models, and it is proved that the recognition effect of a multi-sensor is better than that of a single sensor.

Figure 9 shows the classification diagrams of the different models of Two-Branch CNN, EndNet, MDL-Middle, Trident-HSI, CNN-LiDAR, ConvNeXt-HSI, OctaveConv-LiDAR, and the proposed model. In this figure, different colors represent different classes of objects. From the perspective of a single sensor, for the single-sensor HSI method (for example, d and f in Figure 9), rich spectral information can provide more detailed ground-object information for the target to be detected, but it is difficult to identify similar objects (such as grass and shrubs); methods based on single-sensor LiDAR data (such as Figure 9e,g), using elevation information, can distinguish objects of different heights, but it is difficult to classify objects of the same height. In contrast, in Figure 9h, our proposed HSI-LiDAR joint classification model combines multi-sensor and self-supervised learning algorithms and compares other three advanced algorithms, Two-Branch CNN, EndNet, and MDL-Middle (for example, Figure 9a–c), which can obtain more detailed information and smooth classification results (for example, railways) and can achieve high-precision classification tasks in complex scenes.

3.3.2. Classification Results of the Trento Dataset

Table 4 shows the detailed classification results of eight models in terms of OA, AA, and Kappa coefficients on the Trento dataset. Compared with Two-Branch CNN, EndNet, and MDL-Middle, our proposed method also has obvious improvements in OA, AA, and Kappa coefficients. The classification performance of Apples, Buildings, and Roads is superior to these three models, especially the recognition accuracy of Buildings at 99.10%, which is up to a 1.17% improvement compared to these three models.

Moreover, all single-sensor classification performance is lower than multi-sensor classification. The performance using only HSI data is significantly higher than that of only LiDAR data. For example, in the Trento dataset, the OctaveConv-LiDAR classification model has an OA of 91.85%, AA of 83.57%, and Kappa of 89.21%. These three metrics are all lower than the performance of the ConvNeXt-HSI classification model, for which they are 96.40%, 92.91%, and 95.20%, respectively. When the dual-sensor joint classification is combined, the OA is increased to 88.14%, the AA is increased to 88.14%, and the Kappa is increased to 87.16%.

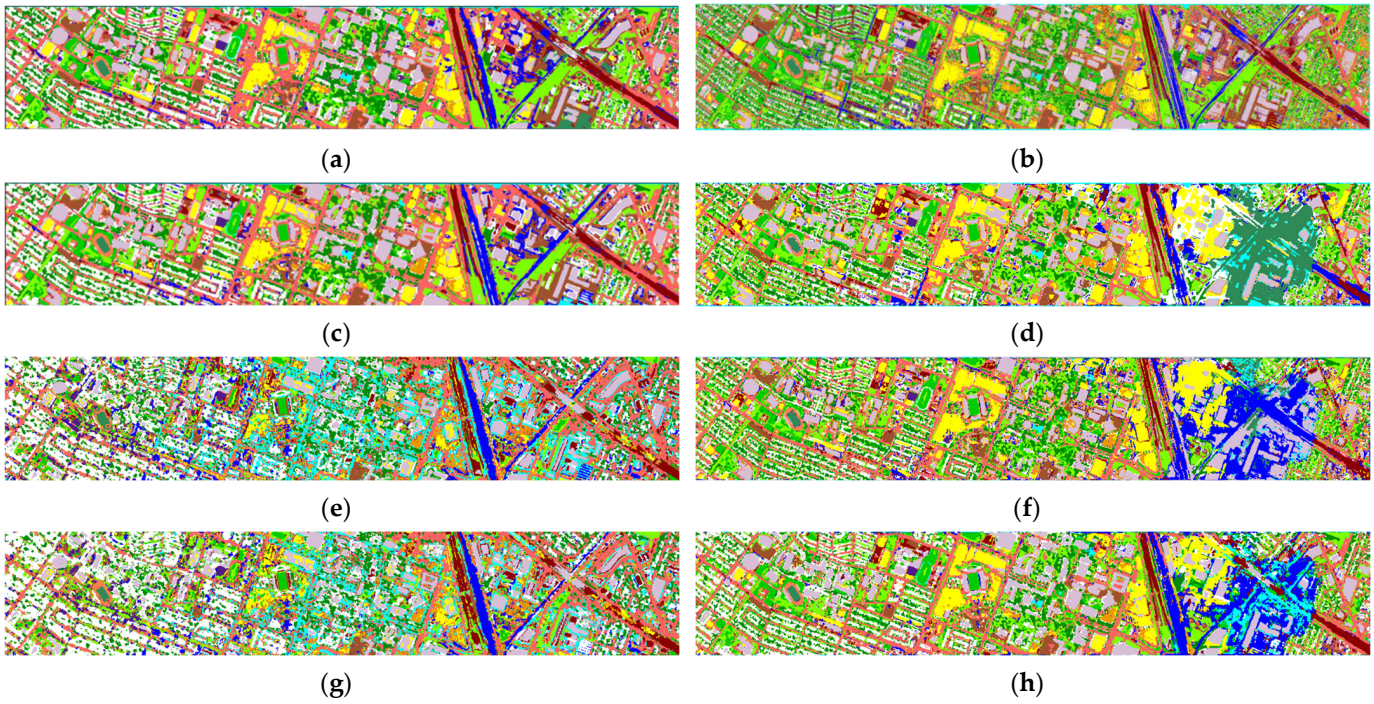


Figure 9. Classification maps of the Houston2013 dataset. (**a**) Two-Branch. (**b**) EndNet. (**c**) MDL-Middle. (**d**) Trident-HSI. (**e**) CNN-LiDAR. (**f**) ConvNeXt-HSI. (**g**) OctaveConv-LiDAR. (**h**) Proposed.

Table 4. Classification accuracy of different models on the Trento dataset.

Class	Two-Branch	EndNet	MDL-Middle	Trident-HSI	CNN-LiDAR	ConvNeXt-HSI	OctaveConv-LiDAR	Proposed
C1	99.78	88.19	99.5	97.87	99.5	98.17	99.78	99.88
C2	97.93	98.49	97.55	87.39	95.25	98.35	96.49	99.1
C3	99.93	95.19	99.1	98.75	78.91	98.96	75.78	96.87
C4	99.46	99.3	99.9	97.94	93.73	98.09	94.38	99.87
C5	98.96	91.96	99.71	99.21	91.68	99.39	91.48	99.08
C6	91.68	90.14	92.25	78.01	67.62	77.22	74.01	94.77
OA (%)	98.36	94.17	98.73	95.28	90.95	96.4	91.85	98.92
AA (%)	97.96	93.88	98	90.17	81.98	92.91	83.57	98.4
K × 100	97.83	92.22	98.32	92.7	88.07	95.2	89.26	98.61

Similarly, comparing the accuracy of a single branch horizontally, ConvNeXt-HSI compared to Trident-HSI and OctaveConv-LiDAR compared to CNN-LiDAR, the classification performance is relatively good. The OA, AA, and Kappa of ConvNeXt-HSI are higher than those of Trident-HSI, at 1.12%, 2.74%, 2.5%. The OA, AA, and Kappa of OctaveConv-LiDAR are higher than those of CNN-LiDAR, at 0.9%, 1.59%, and 1.19%.

Figure 10 shows the classification diagrams of the different models of Two-Branch CNN, EndNet, MDL-Middle, Trident-HSI, CNN-LiDAR, ConvNeXt-HSI, OctaveConv-LiDAR, and the proposed model. In this figure, different colors represent different classes of objects. In the Trento dataset, we can obtain the same conclusion as the Houston2013 dataset. From a single-sensor point of view, for single-sensor HSI methods (for example, Figure 10d,f), it is difficult to identify similar objects (for example, Apples and Woods); for methods based on single-sensor LiDAR data (for example, Figure 10e,g), objects with the same height are difficult to classify (such as Buildings and Roads). In Figure 10h, our proposed method can obtain more detailed information and smooth classification results (e.g., Vineyard) compared with the other three advanced algorithms (e.g., Figure 10a–c).

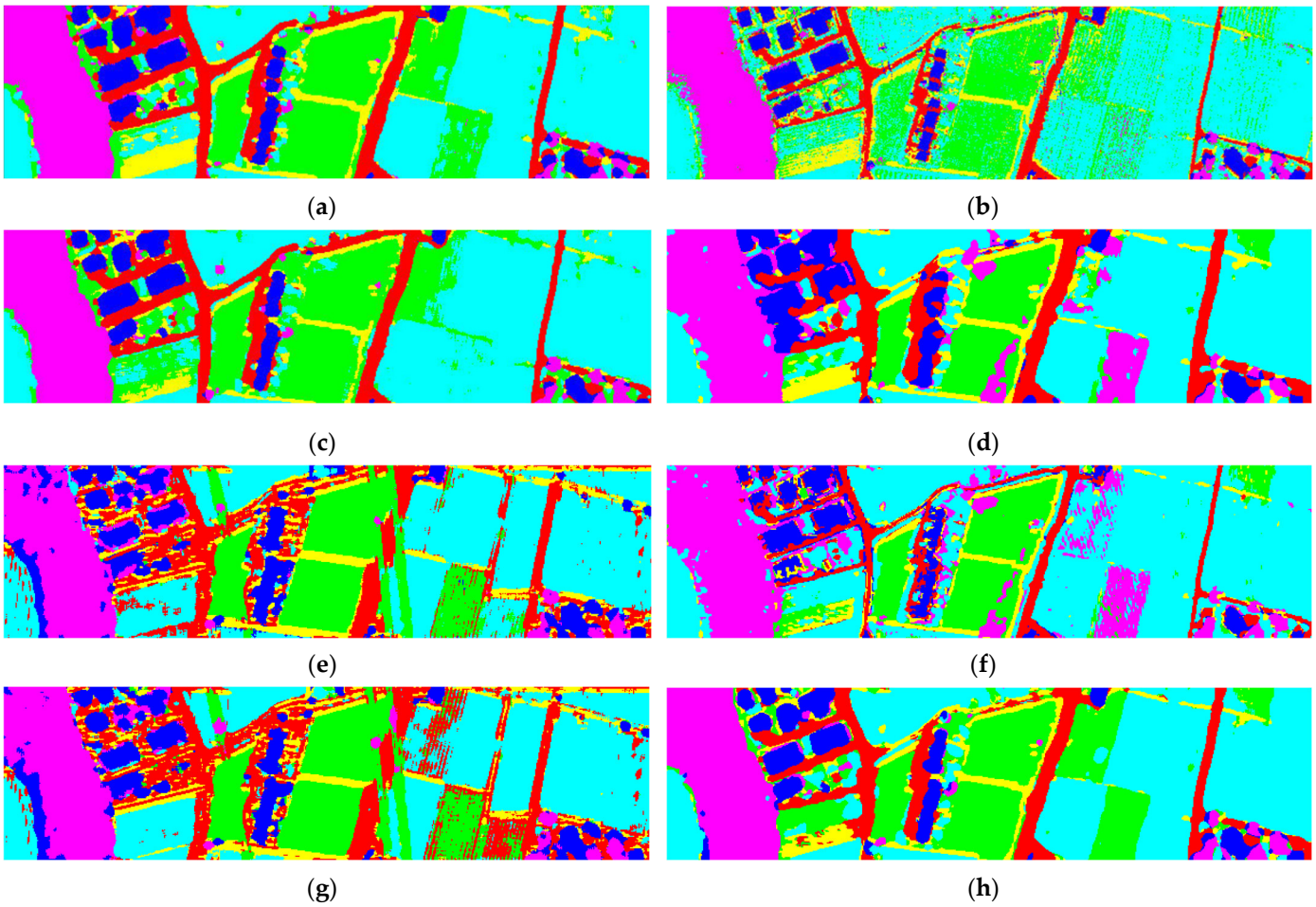


Figure 10. Classification maps of the Trento dataset using different models. (**a**) Two-Branch. (**b**) EndNet. (**c**) MDL-Middle. (**d**) Trident-HSI. (**e**) CNN-LiDAR. (**f**) ConvNeXt-HSI. (**g**) OctaveConv-LiDAR. (**h**) Proposed.

3.3.3. Computational Complexity Analysis

Table 5 shows the model complexity analysis for the different models. The model complexity analysis is represented by two important metrics, which are floating point operations (FLOPs) and the number of model parameters (#param.) #param. FLOPs refers to the number of floating point operations that occur for the input of a single sample (one image) and for the model to complete one forward propagation, i.e., the time complexity of the model. #param refers to how many parameters the model contains, which directly determines the size of the model and also affects the amount of memory used for inference, i.e., the spatial complexity of the model.

Table 5. The numbers of FLOPs and parameters of different classification models.

Methods	#param. (M)	FLOPs (M)
Two-Branch	0.25	4.7
EndNet	0.27	4.9
MDL-Middle	0.25	4.7
Proposed	50	10

Because EndNet does not consider neighborhood information, the spatial and temporal complexity of EndNet is small. Although using a single pixel as input reduces the model complexity, ignoring neighborhood information leads to a decrease in accuracy. The model

proposed in this paper uses multiple encoders and the network is deeper than other models, which greatly increases the computational cost but also improves the performance of the network.

4. Discussion

We further performed different ablation studies to highlight improved aspects of our model. In our proposed model, there are two modules that are critical for classification performance. They are the online multi-scale data augmentation module and the contrastive learning module, respectively. In order to prove the effectiveness of these modules, we successively removed these modules to conduct a series of ablation experiments.

4.1. Effect of the Online Multi-Scale Data Augmentation Module

The online multi-scale data augmentation module is a very critical factor. After we removed the online multi-scale data augmentation module, the fixed patch extraction scale was 11 × 11. Table 6 gives the specific experimental results of the multi-scale data augmentation module on classification accuracy. Our proposed model with the online multi-scale data augmentation module has a certain improvement on the two datasets, from the view of OA, AA, and Kappa indexes. Compared with the offline data augmentation method, this module does not need to synthesize augmented data, thus saving data storage space and having high flexibility.

Table 6. Effect of the multi-scale data augmentation module on classification accuracy.

Dataset	Online Multi-Scale	OA (%)	AA (%)	K × 100
Houston2013	×	89.6	89.41	88.55
	√	91.37	91.33	90.64
Trento	×	97.88	97.54	96.96
	√	98.92	98.4	98.61

4.2. Effect of the Contrastive Learning Module

The contrastive learning module is also a very critical factor. The multi-sensor joint classification with the addition of the contrastive learning module significantly improved the classification performance of the models. Through self-supervised learning, the complementary advantages of multi-sensor remote sensing data are realized. The difference in feature representation affects the performance of multi-sensor joint classification. This approach enables the network to understand higher-level semantic information and significantly improves network classification performance. We removed the contrastive learning module to train the classification model. The specific experimental results are shown in Table 7. It can be clearly seen that the contrastive learning module was added, and the OA, AA, and Kappa of the two datasets were improved to a certain extent. Moreover, this module is only used in the training phase, does not increase the model parameters, and does not affect the speed of inference. This module is plug-and-play, flexible, and effective.

Table 7. Effect of the contrastive learning module on the Houston2013 and Trento datasets.

Dataset	Contrastive Learning	OA (%)	AA (%)	K × 100
Houston2013	×	88.14	89.12	87.16
	√	91.37	91.33	90.64
Trento	×	98.55	98.23	97.94
	√	98.92	98.4	98.61

4.3. Effect of the Stagewise Training Strategy

We proposed a staged pre-training strategy that leverages large-scale pure HSIs and pure LiDAR data to improve HSI-LiDAR multi-sensor classification models. We first performed HSI-encoder pre-training on pure HSI data and then performed LiDAR-encoder pre-training on pure LiDAR data to learn general HSI and LiDAR representations. The last stage trains HSI-LiDAR pre-training, learning the alignment of HSI and LiDAR information while training the classification task. As shown in Figure 11, the pre-training model provides better initialization of model parameters, which brings better generalization performance to the final model and accelerates the convergence on this downstream task.

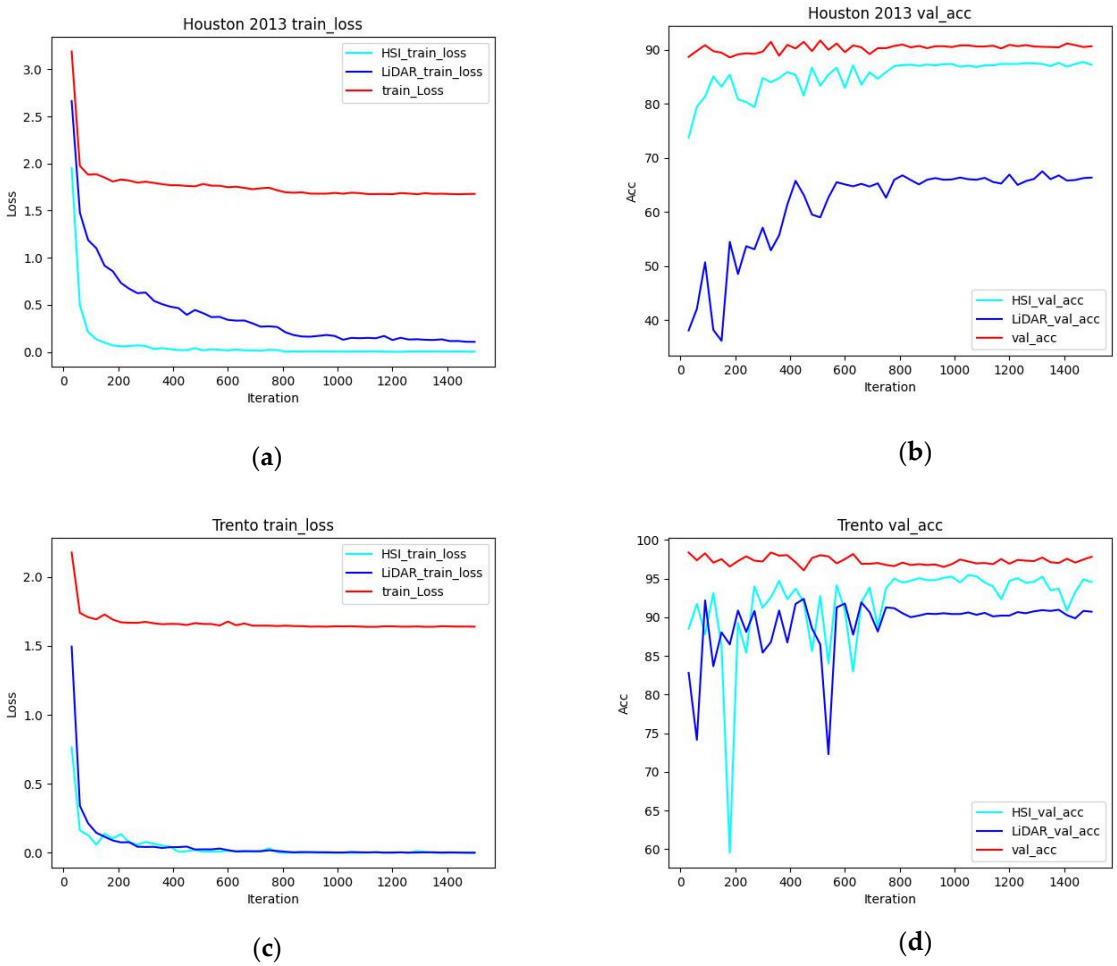


Figure 11. Stagewise training strategy. (**a**) train_loss on the Houston2013 dataset. (**b**) val_acc on the Houston2013 dataset. (**c**) train_loss on the Trento dataset. (**d**) val_acc on the Trento dataset.

5. Conclusions

In this work, we proposed a method for the joint classification of hyperspectral and LiDAR data by fully mining spectral–spatial features and depth information from image data acquired using different sensors. The advantages of multi-sensor remote sensing data are complemented by self-supervised contrastive learning. It solves problems such as affecting the joint multi-sensor classification performance due to differences in the feature representation of different sensors and improves the classification accuracy. The experimental results show that the proposed dual-encoder HSI-LiDAR joint classification model via contrastive learning achieves state-of-the-art classification performance, including on the Houston2013 dataset and Trento dataset. However, the deeper dual-encoder network leads to high model complexity, which is challenging for realtime performance.

In the future, we will explore methods such as model compression, pruning, and quantization to reduce the complexity of the proposed model and improve the real-time performance without compromising the classification performance.

Author Contributions: Conceptualization, A.W., H.W. and Y.I.; methodology, S.D.; software, S.D. and C.L.; validation, S.D. and C.L.; writing—review and editing, S.D., A.W., H.W. and Y.I. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the high-end foreign experts introduction program (G2022012010L) and the Reserved Leaders of Heilongjiang Provincial Leading Talent Echelon 2021.

Data Availability Statement: http://dase.grss-ieee.org/ (accessed on 10 September 2022).

Acknowledgments: Iwahori's research is supported by the JSPS Grant-in-Aid for Scientific Research (C) (20K11873) and the Chubu University Grant.

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

References

1. Schaepman, M.E.; Ustin, S.L.; Plaza, A.J.; Painter, T.H.; Verrelst, J.; Liang, S. Earth system science related imaging spectroscopy—An assessment. *Remote Sens. Environ.* **2009**, *113*, S123–S137. [CrossRef]
2. Hong, D.; Yokoya, N.; Chanussot, J.; Zhu, X.X. CoSpace: Common subspace learning from hyperspectral-multispectral correspondences. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 4349–4359. [CrossRef]
3. Shah, C.; Du, Q.; Xu, Y. Enhanced TabNet: Attentive Interpretable Tabular Learning for Hyperspectral Image Classification. *Remote Sens.* **2022**, *14*, 716. [CrossRef]
4. Zhao, R.; Du, S. An Encoder–Decoder with a Residual Network for Fusing Hyperspectral and Panchromatic Remote Sensing Images. *Remote Sens.* **2022**, *14*, 1981. [CrossRef]
5. Shahshahani, B.M.; Landgrebe, D.A. The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon. *IEEE Trans. Geosci. Remote Sens.* **1994**, *32*, 1087–1095. [CrossRef]
6. Dalponte, M.; Bruzzone, L.; Gianelle, D. Tree species classification in the Southern Alps based on the fusion of very high geometrical resolution multispectral/hyperspectral images and LiDAR data. *Remote Sens. Environ.* **2012**, *123*, 258–270. [CrossRef]
7. Kuras, A.; Brell, M.; Rizzi, J.; Burud, I. Hyperspectral and lidar data applied to the urban land cover machine learning and neural-network-based classification: A review. *Remote Sens.* **2021**, *13*, 3393. [CrossRef]
8. Mäyrä, J.; Keski-Saari, S.; Kivinen, S.; Tanhuanpää, T.; Hurskainen, P.; Kullberg, P.; Poikolainen, L.; Viinikka, A.; Tuominen, S.; Kumpula, T. Tree species classification from airborne hyperspectral and LiDAR data using 3D convolutional neural networks. *Remote Sens. Environ.* **2021**, *256*, 112322. [CrossRef]
9. Dalponte, M.; Bruzzone, L.; Gianelle, D. Fusion of hyperspectral and LIDAR remote sensing data for classification of complex forest areas. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 1416–1427. [CrossRef]
10. Mountrakis, G.; Im, J.; Ogole, C. Support vector machines in remote sensing: A review. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, 247–259. [CrossRef]
11. Ghamisi, P.; Rasti, B.; Yokoya, N.; Wang, Q.; Hofle, B.; Bruzzone, L.; Bovolo, F.; Chi, M.; Anders, K.; Gloaguen, R. Multisource and multitemporal data fusion in remote sensing: A comprehensive review of the state of the art. *IEEE Geosci. Remote Sens. Mag.* **2019**, *7*, 6–39. [CrossRef]
12. Debes, C.; Merentitis, A.; Heremans, R.; Hahn, J.; Frangiadakis, N.; van Kasteren, T.; Liao, W.; Bellens, R.; Pižurica, A.; Gautama, S. Hyperspectral and LiDAR data fusion: Outcome of the 2013 GRSS data fusion contest. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2405–2418. [CrossRef]
13. Geng, J.; Deng, X.; Ma, X.; Jiang, W. Transfer learning for SAR image classification via deep joint distribution adaptation networks. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 5377–5392. [CrossRef]
14. Feng, Q.; Zhu, D.; Yang, J.; Li, B. 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]
15. Jia, X.; Kuo, B.-C.; Crawford, M.M. Feature mining for hyperspectral image classification. *Proc. IEEE* **2013**, *101*, 676–697. [CrossRef]
16. Gao, S.-H.; Cheng, M.-M.; Zhao, K.; Zhang, X.-Y.; Yang, M.-H.; Torr, P. Res2net: A new multi-scale backbone architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 652–662. [CrossRef] [PubMed]
17. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
18. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
19. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
20. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.

21. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
23. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.
24. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
25. Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; Torralba, A. Places: A 10 million image database for scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 1452–1464. [CrossRef] [PubMed]
26. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems 28*; MIT Press: Cambridge, MA, USA, 2015.
27. Bazi, Y.; Bashmal, L.; Rahhal, M.M.A.; Dayil, R.A.; Ajlan, N.A. Vision transformers for remote sensing image classification. *Remote Sens.* **2021**, *13*, 516. [CrossRef]
28. Miao, W.; Geng, J.; Jiang, W. Semi-Supervised Remote-Sensing Image Scene Classification Using Representation Consistency Siamese Network. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–14. [CrossRef]
29. Xu, X.; Li, W.; Ran, Q.; Du, Q.; Gao, L.; Zhang, B. Multisource remote sensing data classification based on convolutional neural network. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 937–949. [CrossRef]
30. Hong, D.; Gao, L.; Hang, R.; Zhang, B.; Chanussot, J. Deep encoder-decoder networks for classification of hyperspectral and LiDAR data. *IEEE Geosci. Remote Sens. Lett.* **2020**, *19*, 5500205. [CrossRef]
31. Hong, D.; Gao, L.; Yokoya, N.; Yao, J.; Chanussot, J.; Du, Q.; Zhang, B. More diverse means better: Multimodal deep learning meets remote-sensing imagery classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 4340–4354. [CrossRef]
32. Bachman, P.; Hjelm, R.D.; Buchwalter, W. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems 32*; MIT Press: Cambridge, MA, USA, 2019.
33. Oord, A.v.d.; Li, Y.; Vinyals, O. Representation learning with contrastive predictive coding. *arXiv* **2018**, arXiv:1807.03748.
34. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9729–9738.
35. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In Proceedings of the International Conference on Machine Learning, Vienna, Austria, 12–18 July 2020; pp. 1597–1607.
36. Chen, X.; Fan, H.; Girshick, R.; He, K. Improved baselines with momentum contrastive learning. *arXiv* **2020**, arXiv:2003.04297.
37. Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A convnet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 11976–11986.
38. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022.
39. Wang, P.; Chen, P.; Yuan, Y.; Liu, D.; Huang, Z.; Hou, X.; Cottrell, G. Understanding convolution for semantic segmentation. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 1451–1460.
40. Chen, Y.; Fan, H.; Xu, B.; Yan, Z.; Kalantidis, Y.; Rohrbach, M.; Yan, S.; Feng, J. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3435–3444.