remote sensing

MDPI

Article

Classification of Hyperspectral and LiDAR Data Using Multi-Modal Transformer Cascaded Fusion Net

Shuo Wang [1], Chengchao Hou [1], Yiming Chen [1], Zhengjun Liu [1,*], Zhenbei Zhang [2] and Geng Zhang [1]

[1] Chinese Academy of Surveying & Mapping, Beijing 100036, China; shuowang7738@163.com (S.W.); houchengchao@163.com (C.H.); chenym@casm.ac.cn (Y.C.); zg1989518@163.com (G.Z.)
[2] State Key Laboratory of Tibetan Plateau Earth System, Resources and Environment (TPESRE), Institute of Tibetan Plateau Research, Chinese Academy of Sciences, Beijing 100101, China; zhangzb@itpcas.ac.cn
* Correspondence: zjliu@casm.ac.cn

Abstract: With the continuous development of surface observation methods and technologies, we can acquire multiple sources of data more effectively in the same geographic area. The quality and availability of these data have also significantly improved. Consequently, how to better utilize multi-source data to represent ground information has become an important research question in the field of geoscience. In this paper, a novel model called multi-modal transformer cascaded fusion net (MMTCFN) is proposed for fusion and classification of multi-modal remote sensing data, Hyperspectral Imagery (HSI) and LiDAR data. Feature fusion and feature extraction are the two stages of the model. First, in the feature extraction stage, a three-branch cascaded Convolutional Neural Network (CNN) framework is employed to fully leverage the advantages of convolutional operators in extracting shallow-level local features. Based on this, we generated multi-modal long-range integrated deep features utilizing the transformer-based vectorized pixel group transformer (VPGT) module during the feature fusion stage. In the VPGT block, we designed a vectorized pixel group embedding that preserves the global features extracted from the three branches in a non-overlapping multi-space manner. Moreover, we introduce the DropKey mechanism into the multi-head self-attention (MHSA) to alleviate overfitting caused by insufficient training samples. Finally, we employ a probabilistic decision fusion strategy to integrate multiple class estimations, assigning a specific category to each pixel. This model was experimented on three HSI-LiDAR datasets with balanced and unbalanced training samples. The proposed model outperforms the other seven SOTA approaches in terms of OA performance, proving the superiority of MMTCFN for the HSI-LiDAR classification task.

Keywords: deep learning; multi-head self-attention (MHSA); multi-modal transformer cascaded fusion net (MMTCFN); HSI-LiDAR classification

Citation: Wang, S.; Hou, C.; Chen, Y.; Liu, Z.; Zhang, Z.; Zhang, G. Classification of Hyperspectral and LiDAR Data Using Multi-Modal Transformer Cascaded Fusion Net. *Remote Sens.* **2023**, *15*, 4142. https://doi.org/10.3390/rs15174142

Academic Editor: Dong Liu

Received: 17 July 2023
Revised: 16 August 2023
Accepted: 17 August 2023
Published: 24 August 2023

1. Introduction

In recent years, with the continued advancement of remote sensing technology and the enlargement of data-gathering sources, remote sensing imagery has recently emerged as one of the key methods for learning about the characteristics of the Earth's surface and has achieved remarkable success in several fields, including the distribution of water resources, vegetation cover, and land use [1–3]. Nowadays, it is normal practice to gather remote sensing data for the same area from many sources. The availability of these data is high, so it has become possible to utilize images acquired by different sensors for surface cover object description [4–6].

Multi-modal data can provide varied information, for example, hyperspectral imagery (HSI) can provide rich spectral information of features, while LiDAR data contains highly accurate three-dimensional topographic information about the terrain and features. HSI provides a more precise description of spectral characteristics, facilitating feature classification. However, HSI's spatial resolution is typically low, and its optical image can be

easily affected by environmental factors such as the atmosphere, clouds, rain, and snow, leading to unstable data quality. LiDAR is unaffected by light and climate [7,8], which can compensate for HSI's limitations and provide new approaches for classifying land cover in complicated scenarios. Moreover, LiDAR data also includes rich spatial information in three dimensions, and it can produce more accurate classification results for surface covers with similar spectral curves but differing heights. As a result, when it comes to characterizing the ground surface, HSI and LiDAR data are complementary to one another. The two of them also cooperate in the categorization of ground features, which can have an effect of "one plus one is greater than two" in the classification outcomes.

Due to the high dimensionality of HSI and LiDAR data after fusion, which may contain some irrelevant or redundant information, HSI must be downscaled before feeding the fused data into the classifier. Researchers have suggested a few dimensionality reduction techniques to address this issue, including principal component analysis (PCA) [9–11], linear discriminant analysis (LDA) [12,13], and Isometric Feature Mapping (ISOMAP) [14]. A popular strategy in the early stages of classifying HSI-LiDAR fusion data is to extract feature properties of features using machine learning, which can be pixel-based or object-based. Pixel-based methods, such as the well-known support vector machine (SVM) [15,16], random forest (RF) [17], and Artificial Neural Networks (ANN) [18], rely solely on the spectral features of each pixel in the scene, assigning a class to each pixel without considering the correlation between neighboring pixels. Object-based classification approaches, in contrast, merge neighboring pixels into an object as a unit for classification, such as Decision Tree (DT) [15] and Naive Bayes Classifier (Bayes) [17].

As the amount of data and computational power continue to increase, many algorithmic improvements have been discovered. Deep learning can better utilize massive data and computational resources, while reducing the need for human intervention and automatically learning deeper features from data, thus achieving better performance than traditional machine learning. In recent years, deep learning has been successfully applied to target detection [19,20], semantic segmentation [21,22], and super-resolution reconstruction of remote sensing images [23,24], as well as multi-modal data fusion and classification [25,26]. Convolutional Neural Network (CNN) plays a crucial role in enhancing classification accuracy and automatically learning features in HSI-LiDAR multi-modal data categorization. CNNs are classified as unsupervised and supervised, and when algorithms are trained without the use of labeled samples, they are referred to as unsupervised training, and when they are taught using labeled examples, they are referred to as supervised training. Patch-to-Patch Convolutional Neural Network (PToP CNN) is an unsupervised feature extraction network proposed by Zhang et al. [27] that intends to combine multi-scale features among various source data and categorize them. Rasti et al. [28] proposed a fusion model with unsupervised sparse and low-rank decomposition to extract low-rank fusion features from HSI-LiDAR data, allowing the classification map transition to be more natural and smooth, resulting in a softer, homogeneous, and coherent fused classification map. For supervised training techniques, Xia et al. [29] developed the semisupervised graph fusion (SSGF) approach to model the 3D spatial information of the major components of HSI and LiDAR, respectively, to obtain new features and categorize them.

Despite achieving some success, the single-branch-based CNN feature extraction method still has several flaws due to its limited receptive area and information loss. Researchers have suggested multi-branch-based feature extraction techniques to address these issues. Hang et al. [30] introduced a simple two-branch coupled CNN that extracts features from HSI and LiDAR separately and then combines these heterogeneous data for classification using feature-level and decision-level fusion approaches. Zhao et al. [31] introduced a new fused HSI-LiDAR classification network called hierarchical random walk network (HRWN), which integrates dual-channel feature extraction and the hierarchical random walk technique to increase classification accuracy. Unlike previous models, a similar double-concentrate network (SDCN) proposed by Zhu et al. [32] first uses the double-concentrate structure to extract the features in the HSI, and then integrates the

LiDAR information on top of it. This is conducted to highlight the distinctions between spatial and band features and boost the model's sensitivity to particular data.

The data dimension and information increase when HSI-LiDAR is used as the input data, yet there is a significant degree of overlap between this information. The attention mechanism was incorporated into the model by the researchers to aid in the model's capacity to pay greater attention to essential feature information and enhance its generalization ability [33–35]. Li et al. [36] utilized three components of a two-channel CNN, multi-scale attention model and long and short-term memory CNN, to form a network A(3) CLNN for fusion and classification of multi-modal data to improve the predictive ability of the model. Mohla et al. [37] proposed a feature fusion and extraction framework called FusAtNet, a model that uses a self-attention mechanism for HSI to highlight its spectral features and a cross-attention mechanism for LiDAR to highlight the morphological features in HSI. Wang et al. [38] used a network called multi-attentive hierarchical fusion net (MAHiDFNet), which employs a new Modal Attention Module (MA) for feature interaction and integration of spatial, spectral, and elevation information extracted from the three branches in order to generate integrated modal features for HSI- LiDAR feature-level fusion classification.

Additionally, the transformer model has been successfully used to classify data from multi-modal remote sensing data. The transformer's self-attention mechanism can be used to learn the global information of multi-modal data interaction. Building on this, Ding et al. [39] proposed the global–local transformer network (GLT-Net) for capturing the global–local correlation features of the input data, which is effective in improving the results. Zhang et al. [40] proposed a local information interaction transformer (LIIT) model to overcome the problem of insufficient or redundant complementary information between HSI and LiDAR data by dynamically fusing multi-modal features through the transformer, which also yielded good results.

Although feature extraction for HSI and LiDAR data has advanced due to the development of deep learning, there are still several issues with the fusion and categorization of multi-modal data. For instance, the model performs badly on a smaller number of classes as a result of the imbalance of labeled training samples [41,42], and high-dimensional data necessitates more computational resources, which also makes model training more challenging [43,44]. In this paper, as a starting point to address the aforementioned issue, a framework called multi-modal transformer cascaded fusion net (MMRCFN) is suggested, which can be applied to the categorization of multi-modal remote sensing data. This model combines two stages, feature extraction and feature fusion, and the former of which offers a three-branch cascaded CNN framework for extracting shallow characteristics from the combined HSI-LiDAR data, such as spectral, spatial, and 3D spatial features. We generate multi-modal long-range integrated information using the transformer-based VPGT module during the deep feature fusion stage, and the MASH in this module properly accounts for the correlation and heterogeneity between multi-modal data. Finally, we create a land cover categorization map by pixel-by-pixel estimating the probability distribution of each group.

The following is a summary of this essay's main points.

1. This paper proposes a model for classifying data from multiple modalities, including HSI-LiDAR. In the feature extraction stage, a three-branch cascaded CNN module is used to extract spatial–spectral-3D terrain data. During the feature fusion stage, the VPGT method takes into account the correlation and heterogeneity among the multi-modal data and generates fusion features to improve classification accuracy.
2. We introduce a straightforward yet efficient vectorized pixel group embedding in the feature fusion stage, which maintains detailed information of the feature maps in a non-overlapping multi-channel manner. Additionally, we employ the MASH with DropKey approach to address the issue of overfitting. The combination of these two techniques effectively captures long-range correlation information among the multi-modal features.

3. We conduct numerous balanced and unbalanced sample tests on three HSI-LiDAR datasets, and the results demonstrate that our proposed method outperforms the state-of-the-art (SOTA) method we compare it with.

The rest of the paper is organized as follows: Section 2 provides a detailed description of our proposed MMTCFN model. Section 3 presents the three HSI-LiDAR datasets used in this experiment, along with extensive experiments and analyses comparing them to seven other state-of-the-art (SOTA) methods. Finally, Section 4 presents the general conclusions drawn from the study.

2. Methodology

2.1. Over Architecture

The MMTCFN's framework overview diagram is shown in Figure 1. The network framework is designed to utilize a Convolutional Neural Network (CNN) feature extraction backbone and a transformer-based VPGT feature fusion backbone for fusion and classification of HSI and LiDAR data. The proposed feature extraction backbone primarily consists of three branches: the spectral branch (δ_{H-spec}) for HSI, the spatial branch (δ_{H-spa}) for HSI, and the LiDAR branch (δ_{LiDAR}). Among them, the δ_{H-spec} method employs 1D convolution as the feature extraction unit to extract the spectral data from the HSI, while the δ_{H-spa} and δ_{LiDAR} methods use 2D convolution to extract the spatial data from the HSI and LiDAR, respectively.

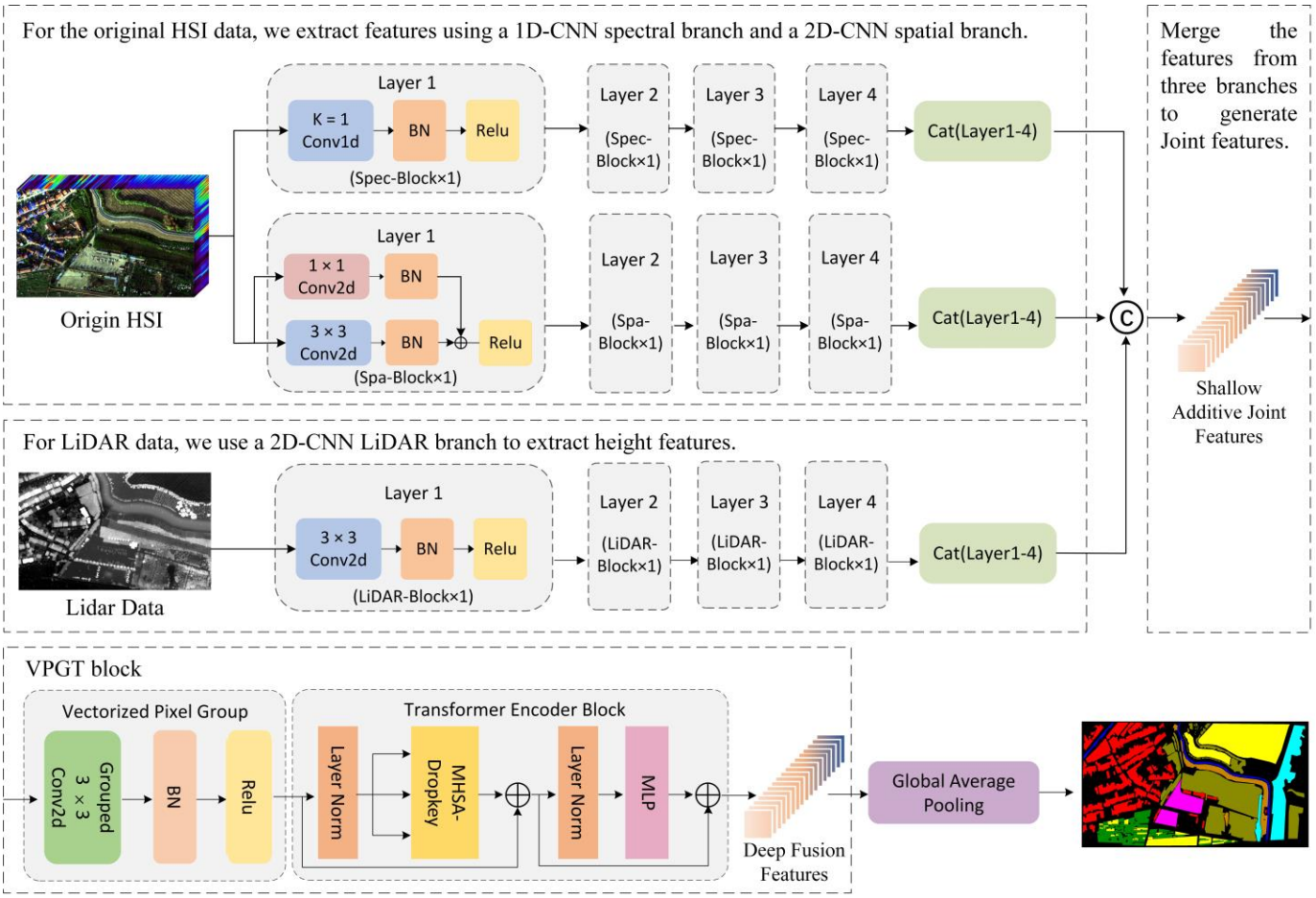


Figure 1. Overall architecture of the MMTCFN.

To extract the spectral, spatial, and elevation data from the raw data, we first extract the HSI-patch and LiDAR-patch that are centered on each pixel in the paired HSI-LiDAR data in both the height and width dimensions. We then input the two patches into the HSI

branch and LiDAR branch of the feature extraction backbone. Second, as the VPGT module is capable of capturing the global information of multi-modal features and adaptively learning the link between the features, we superimpose the feature maps produced by the three branches and employ the VPGT module for multi-modal feature fusion. Finally, to preserve the spatial integrity of the input data, we use a global average pooling layer instead of a fully connected layer to generate a classification probability map based on the number of categories.

The samples for this study were split into three groups: the training set, the validation set, and the test set. The training set is used to update the network model's parameters throughout the training process. The validation set is utilized to describe the network structure and alter the model parameters, and the test set is used to evaluate the model's performance and generalization ability.

2.2. Feature Extraction Backbone

Considering that HSI-LiDAR data have distinct features, in order to enhance the diversity and richness of feature extraction, we adopt a three-branch CNN structure as the backbone for HSI-LiDAR data. Each branch is responsible for extracting the spectral information, spatial information, and elevation information from the data. δ_{H-spa}, δ_{H-spec}, and δ_{Lidar} each consist of four layers. Layers 2, 3, and 4 are similar in design to Layer 1, with the exception that the output feature map's channel count is in the shape of an inverted pyramid. The distinction is that the number of output channels from Layer 1 to Layer 4 is 256, 128, 64, and 32.

The architectural diagram of the layer of the δ_{H-spa} branch of the HSI (shown in Figure 2) is used to illustrate its structure and role in detail. The layer of the δ_{H-spa} branch consists of a parallel structure of a 3 × 3 2D convolutional layer and a 1 × 1 2D convolutional layer, using a convolutional kernel of size three to expand the receptive field and improve the ability of localized feature extraction, and a 1 × 1 convolutional kernel for distinguishing the difference between different bands. The design of the parallel structure can improve the network's ability to perceive features of different complexity. To ensure that the gradient is more stable during the backpropagation process during training, and to avoid the phenomenon of gradient disappearance or gradient explosion, a BN (batch normalization) layer is added after each convolutional layer, and finally, the parallel structure is designed to increase the nonlinear relationship between layers through the ReLU (rectified linear unit) activation function, and we define the final output of this branch to be the feature extraction branch. We define the final output feature map of this branch as K_{H-spa}. Unlike the δ_{H-spa} branch, the δ_{H-spec} branch and the δ_{LiDAR} branch use a serial cascade structure to construct the serialized network. The δ_{H-spec} branch uses a one-dimensional convolutional layer with a convolutional kernel size of one, a one-dimensional BN layer, and a ReLU activation layer composition to extract the spectral information in the HSI. The δ_{LiDAR} branch uses a 3 × 3 2D convolution, a 2D BN layer, and an activation function to extract the terrain height features in the LiDAR data. Similarly, the final output feature maps of these two branches are defined as K_{H-spec} and K_{LiDAR}. Based on the ResNet [45] structure, we collect all the feature maps output from Layers 1–4, and here we represent the feature maps output from the three branches as Equation (1):

$$K_b = F_b + \sum_{i=0}^{L} \partial\left(K_b^{L-1} \infty W_b^L + r_b^L\right), \quad (1)$$

where b∈{H-spec, H-spa, LiDAR}, L∈{1, 2, 3, 4}, F_b denotes the features of the input three branches, K_b^L represents the feature map output from the Lth Layer of the bth branch. W_b^L represents the size of the convolution kernel. In the δ_{H-spec} branch of HSI, $W_b^l \in R^{k=1}$; in the δ_{H-spa} branch of HSI, $W_b^l \in R^{3\times3}$ & $R^{1\times1}$; and in the δ_{LiDAR} branch, $W_b^l \in R^{3\times3}$. '∞' represents the convolution operation, r_b^l is the increased bias, and '∂' represents activation function ReLU.

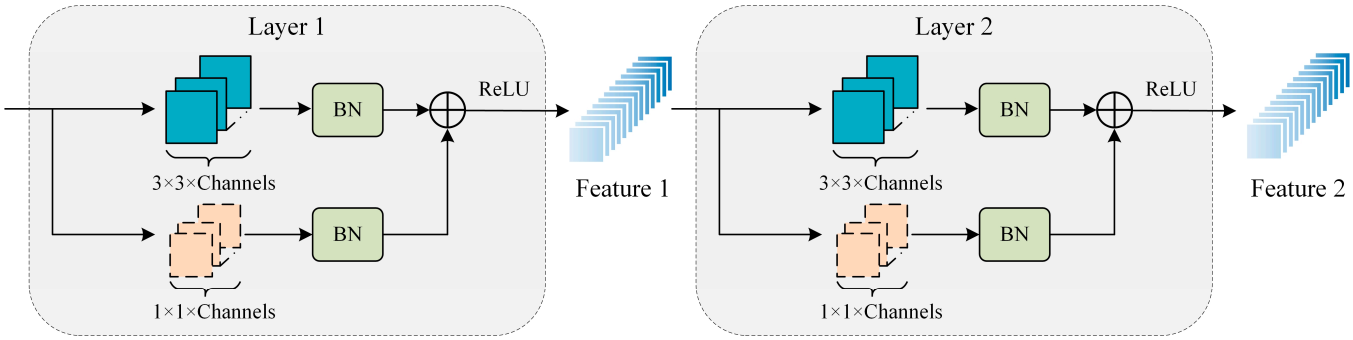


Figure 2. HSI spatial branch layer network architecture.

In order to extract the joint features of the HSI-LiDAR data more efficiently and to improve the recognition accuracy of the target objects by the advantages complementing each other, the output features of the three branches need to be superimposed. This generates a new superimposed joint feature vector K_{Joint}, which can be defined as:

$$K_{Joint} = K_{H-spec} \bowtie K_{H-spa} \bowtie K_{LiDAR}, \quad (2)$$

where the term '$\bowtie$' refers to concatenation. K_{Joint} contains a wealth of topographic, spectral, and spatial data. Next, the stacked joint features are fed into the VPGT block, capturing the correlations and interdependencies between the branches and automatically assigning different weights to different features while suppressing the response to noise and irrelevant information.

2.3. Feature Fusion Backbone

The CNN structure has a potent feature extraction capability that allows it to share the weight parameters and local perception of the image through the convolutional layer while also learning the shallow and deep features in the image automatically. However, CNN's local perception capability restricts to some extent its capacity to extract features from multi-modal fusion data. In contrast, transformer can learn the correlation between different source features through the self-attention mechanism and effectively capture the long-distance dependencies and interaction information between multi-modal data. Therefore, the feature fusion in this work uses the transformer-based vectorized pixel group transformer (VPGT) block.

The vectorized pixel group embedding and transformer encoder block are the two components of the VPGT block suggested in this study. The input feature maps are transformed into continuous visual embedding vectors using vectorized pixel group embedding, which are then used as inputs to the transformer encoder block to extract and fuse features. The multi-head self-attention mechanism (MHSA) with DropKey and the Multi-Layer Perceptron (MLP) layer are the two components of the transformer encoder block.

2.3.1. Vectorized Pixel Group Embedding

The embedding layer in the original Vision Transformer (VIT) [46] is to reduce the whole input image to zero, cut it into one non-overlapping patch, and feed one patch as a token into the model for processing. When the input data is a feature map, patch embedding is performed to split the whole feature map into a series of patch blocks, and the pixels within each patch block are expanded into a vector, and these vectors are concatenated to form a vector matrix. This method contains only a small portion of local information within each patch, and its contextual and interaction information is in blocks, which is not conducive for establishing inter-feature dependencies.

Instead of patch embedding, we suggest using a pixel embedding layer to address this issue. Each pixel in the entire multi-modal fusion feature map will be expanded into a vector through the vectorized pixel embedding process. This will allow each pixel to be

mapped one-to-one with the vector, allowing each pixel to be viewed as an independent entity rather than a component of the region as in the case of patch embedding. The vectorized pixel embedding may retain the detail information and long range dependence information of the feature map more completely, allowing the network to better capture the correlation between multi-modal features.

The embedding layer was originally meant to convert discrete pixel points into a continuous low-dimensional vector space. However, employing ordinary convolution as the embedding layer for fused data results in an excessive number of parameters, which lowers the model's training effectiveness. As a result, group convolution was used as the embedding layer in this study. The difference between grouped convolution and standard convolution is shown in Figure 3. Assuming the input feature map is $M \in \mathcal{R}^{B \times C \times H \times W}$, where B stands for the batchsize, C for the feature map's channel count, and H and W stand for the feature map's width and height, respectively. First, we need to divide the feature map M into n relatively independent groups, and the number of channels of both input and output can be evenly divisible by the number of groups n, that is:

$$M = \{M_1, M_2, M_3, \ldots, M_i, \ldots, M_n\}, \tag{3}$$

where n is the number of groups, and the number of channels in batch M_i is C/n, that is, $M_i \in \mathcal{R}^{B \times (C/n) \times H \times W}$. Then, a different convolution operation is performed on each group to obtain the output feature map of each group M_i^{out}:

$$M_i^{out} = M_i \infty W_i + r_i, \tag{4}$$

where W_i is the size of the convolution kernel for each group, and r_i is the increased bias of the ith group convolution. Finally, the n group output feature maps are then stitched together to generate the input feature map of the transformer encoder layer M^{out}:

$$M^{out} = \mathrm{Concat}\left(M_1^{out}, M_2^{out}, M_3^{out}, \ldots, M_i^{out}, \ldots, M_n^{out}\right), \tag{5}$$

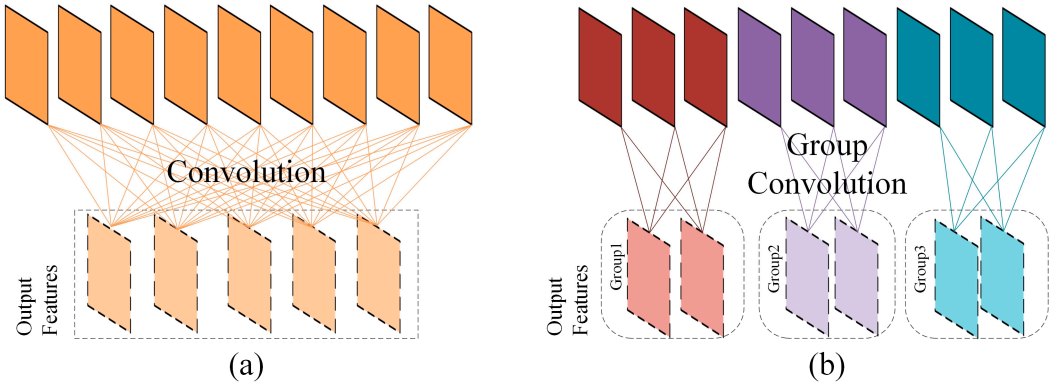


Figure 3. Comparison of standard convolution and grouped convolution. (**a**) Standard convolution. (**b**) Group convolution.

Group convolution reduces the number of parameters, which considerably improves computing performance when compared to ordinary convolution. When dealing with multi-modal fusion features, group convolution can better maintain the independent information across features, which is helpful for transferring all of the information into the transformer encoder layer for feature fusion and screening. We next create vectorized pixel group embedding layers using group convolution, and to increase the stability of the mapping process, we add a BN layer and activation function ReLU behind each group convolution layer.

2.3.2. Transformer Encoder Block

Mulit-Head Self-Attention with DropKey

One of the most important parts of the transformer encoder block is MHSA [47], which builds long-distance dependencies by determining the attentional weights between each position in the input sequence vectors and the other positions in order to better capture the contextual information in the sequence vectors. By using three independent linear transformations to divide each element of the input sequence vector into the three vector subspaces Q (query), K (key), and V (value), MHSA then learns the connections and features among the various subspaces using multiple attention heads. Finally, the output vector matrix of MHSA is obtained by stitching together the outputs of many attention heads and mapping them to the final output space using linear transformations. Figure 4b,c depict how the MHSA was implemented. Assuming for a moment that our input is a 3D matrix, $X^3 \in \mathcal{R}^{H \times W \times C}$, where H and W stand for the 3D matrix's length and breadth, respectively, and C for the number of channels. In order to create a 2D matrix patch $X^2 \in \mathcal{R}^{S \times C}$, we next compress this 3D matrix into a 2D space, where S is created by multiplying H by W. The patch X^2 is mapped by a linear transformation to three independent vectors, Q, K, and V, whose expressions are given below:

$$Q = X^2 W^Q, K = X^2 W^K, V = X^2 W^V, \tag{6}$$

where W^Q, W^K, and W^V are C × C matrix parameters, $Q \in \mathcal{R}^{S \times C}$, $K \in \mathcal{R}^{S \times C}$, and $V \in \mathcal{R}^{S \times C}$. Q, K, and V are then mapped into n groups into n vector subspaces, that is, there are a total of n attention heads with the following expression:

$$\begin{aligned} Q &= \{Q_1, Q_2, Q_3, , \ldots, Q_i, \ldots, Q_n\}, \\ K &= \{K_1, K_2, K_3, , \ldots, K_i, \ldots, K_n\}, \\ V &= \{V_1, V_2, V_3, , \ldots, V_i, \ldots, V_n\}, \end{aligned} \tag{7}$$

and at this time, Q_I, I, and $V_i \in \mathcal{R}^{S \times (C/n)}$. Next, we need to calculate the similarity between Q and K using softmax, the formula is shown in Equation (8):

$$Z_i = \text{Attention}(Q_i, K_i, V_i) = \text{Softmax}(\frac{Q_i K_i^T}{\sqrt{d_k}}) V_i, \tag{8}$$

where Z_i stands for the score that is weighed by the ith head's attention. The result of dot product of Q_i and K_i^T is the length of the projection of one vector on the other vector, which can be used to reflect the similarity between the two vectors. The gradient value of softmax backpropagation reaches an extremely small value and is vulnerable to gradient vanishing when the value of $Q_i K_i^T$ is high. To solve this problem, dividing $\sqrt{d_k}$ in Equation (8) controls the variance to obtain a value of one. The final step is to multiply the similarity vector produced from softmax by V_i to obtain the attention value for each head. The n heads need to be combined in the final output and then multiplied by a projection matrix W^o to obtain the output Z matrix of the MHSA with the following formula:

$$Z = \text{Concat}(Z_1, Z_2, \ldots, Z_i, \ldots, Z_n) W^o, \tag{9}$$

MHSA is prone to overfitting in few-shot scenarios, and people often introduce the commonly used Dropout method in CNNs to solve this problem. However, the use of such random dropout operations in MHSA may destroy the probability distribution of the attention weights, thus leading the model to over-focus on locally specific information. This paper introduces the recently proposed Dropkey approach as a result. The Dropkey methodology, in contrast to the conventional Dropout method, employs the key as its base unit and executes the Dropkey operation before computing the attention matrix (as illustrated in Figure 4c), that is, before softmax. This strategy enhances the model's generalizability by capturing significant information from a global perspective. The drop

ratio of Dropkey in the model also changes on different datasets, which is discussed in Section 3.4.2.

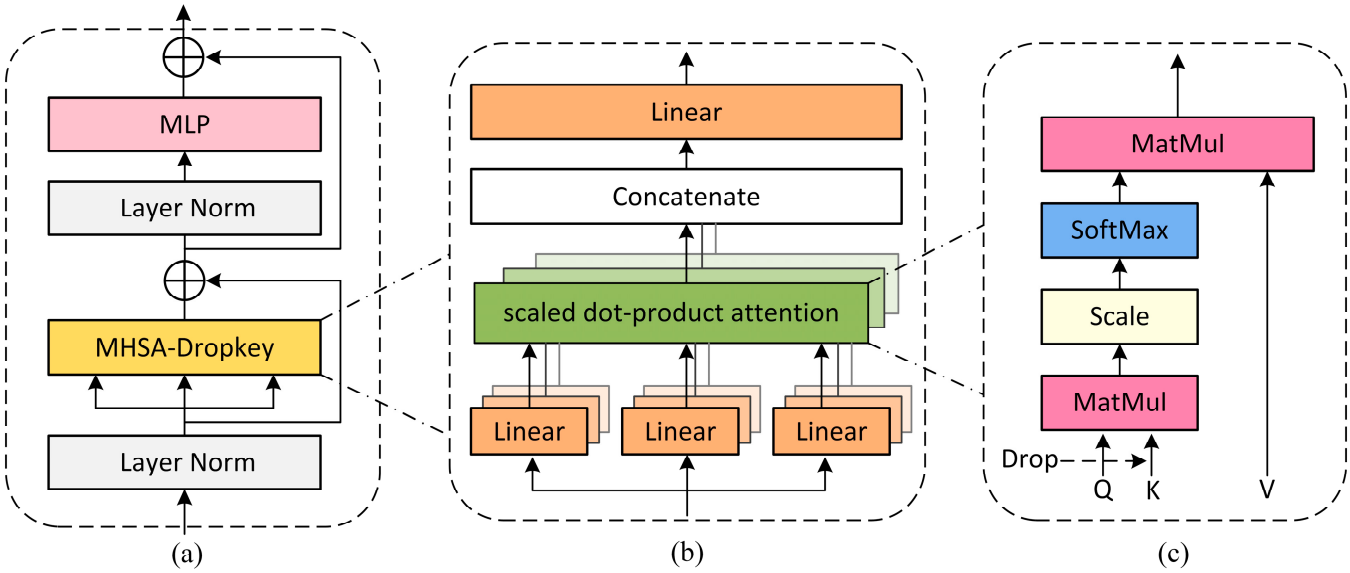


Figure 4. (**a**) Transformer encoder overview. (**b**) Structure of MHSA. (**c**) Self-attention layer with DropKey.

Multi-Layer Perceptron

To increase the expressiveness of the model, nonlinear transformations are applied to the MHSA results using the MLP layer. The MLP layer in transformer is typically composed of two linear transformations and an activation function, where the first linear transformation weights and sums the inputs, the second maps them to a new space, and the activation function nonlinearly transforms the output. The output of the MLP layer can be defined as:

$$\mathrm{MLP}(\mathrm{x}) = \eta(0, \mathrm{x}W_1 + b_1)W_2 + b_2, \tag{10}$$

where x is the input. W_1, b_1 and W_2, b_2 are the weights and biases of the first and second linear transformations, respectively. 'η' is the nonlinear activation function.

A layer normalization (LN) layer is added before the MHSA and MLP layers to normalize the input, which accelerates model convergence and improves its robustness. Meanwhile, the whole transformer encoder block is connected together by residual structures [45] (as shown in Figure 4a).

3. Experiments and Analysis

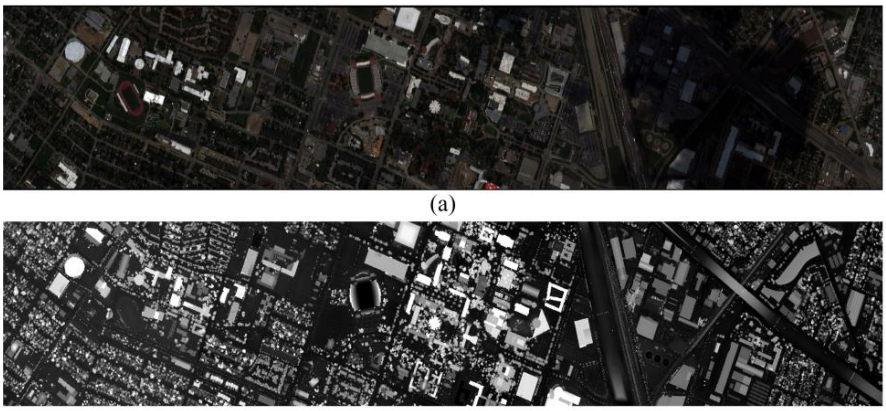
In order to validate the effectiveness of the proposed multi-modal fusion model MMTCFN under balanced and unbalanced samples, we conducted extensive experiments using three HSI-LiDAR datasets.

3.1. Experimental Datasets

Three HSI-LiDAR datasets in total are used in this study, including two common public datasets Houston dataset and MUUFL dataset, and one HSI-LiDAR dataset created by the authors, named Wuhan dataset. The HSI images, LiDAR-based DSM images, and category information of the three datasets are shown in Figures 5–7, and the detailed information of each of the three datasets is described below.

1. Houston dataset: This dataset was acquired at the GRSS Data Fusion Contest 2013 and covers the University of Houston campus and the surrounding urban area. The number of HSI image bands totaled 270, with wavelengths ranging from 364 to 1046 nm. The spatial size of the HSI and DSM is 1905 × 349 image elements, and the

spatial resolution is 2.5 m. The dataset has a total of 15 feature classes and the detailed information of each class is shown in Figure 5.



(a)

(b)

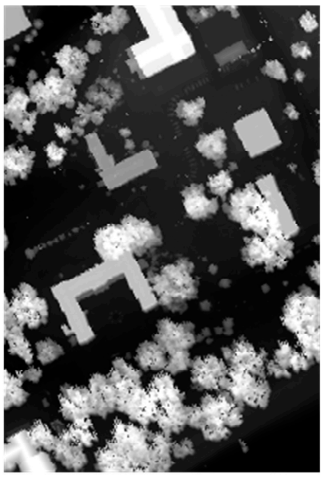
Label	Color	Class Name	Samples
1		Health grass	1374
2		Stressed grass	1454
3		Synthetic grass	759
4		Trees	1264
5		Soil	1298
6		Water	339
7		Residential	1476
8		Commercial	1354
9		Road	1554
10		Highway	1424
11		Railway	1566
12		Parking lot 1	1429
13		Parking lot 2	632
14		Tennis court	513
15		Running track	798
Totle			17,234

(c)

Figure 5. Original HSI, LiDAR-based DSM and number of samples in Houston dataset. (**a**) Original HSI. (**b**) LiDAR-based DSM. (**c**) Category details.

2. MUUFL dataset: The University of Southern Mississippi Gulf Park Campus is where the MUUFL Gulfport dataset, which includes HSI and LiDAR data for the campus area, was gathered. The raw HSI image has 72 bands and 325 × 337 pixels. There are 64 remaining bands after the noisy bands—the first four and the last four bands—are eliminated. The lower right corner of the original HSI image contains invalid regions that require cropping, and the cropped HSI and DSM dimensions are 325 × 220 × 64. The scene's pixels were manually classified into a total of 11 classifications. However, mostly grass (label 2) refers to a surface that is clearly covered with grass, and mixed ground surface (label 3) refers to a surface that may include a mixture of grass, soil, dirt, etc.

(a)



(b)

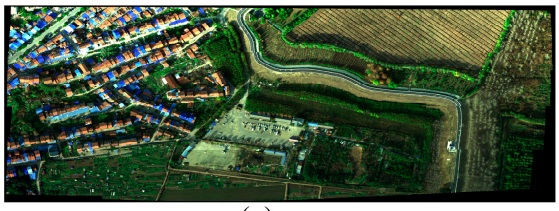
Label	Color	Class Name	Samples
1		Trees	23,246
2		Mostly grass	4270
3		Mixed ground surface	6882
4		Dirt and sand	1826
5		Road	6687
6		Water	466
7		Building shadow	2233
8		Building	6240
9		Sidewalk	1385
10		Yellow curb	183
11		Cloth panels	269
Totle			53,687

(c)

Figure 6. Original HSI, LiDAR-based DSM and number of samples in MUUFL dataset. (**a**) Original HSI. (**b**) LiDAR-based DSM. (**c**) Category details.

3. Wuhan dataset: This dataset was acquired on 14 March 2023, in Wuhan, Hubei Province, China. The raw HSI data underwent preprocessing, which included atmospheric, geometric, and radiometric correction. Before creating a comprehensive digital surface model, the raw point cloud data was filtered and downsampled to remove noise and unnecessary points. Finally, the HSI was positioned in accordance

with the LiDAR-based DSM utilizing ground-based individual control points to ensure spatial coherence between the two sources of data. The HSI images have a total of 270 bands with a wavelength range between 402–1031 nm and a spectral resolution of 3 nm. The spatial size of HSI and DSM is 2500 × 1140 pixels with a spatial resolution of 0.4 m. After comparing the features with Google Maps, a total of eight land cover features were tagged.



(a)

(b)

Label	Color	Class Name	Samples
1		Vegetation	377,569
2		Bare land	36,894
3		Dead wood	86,715
4		Plowland	128,006
5		Cropland	360,256
6		Parking lot	153,176
7		Road	73,455
8		Building	433,706
Totle			1,649,777

(c)

Figure 7. Original HSI, LiDAR-based DSM and number of samples in Wuhan dataset. (**a**) Original HSI. (**b**) LiDAR-based DSM. (**c**) Category details.

3.2. Experimental Setting

(1) Parameter settings: All experiments in this study were implemented using the Pytorch deep learning framework in Python. The PC used for all trials had an Intel(R) Xeon(R) Gold 5218R processor, an NVIDIA GeForce RTX 3080 graphics card, 128GB of system RAM, and Windows 10. To minimize the impact of unexpected errors and ensure the accuracy and stability of the results, we set the training epoch to 100 in the experiments and averaged all experimental results over 10 runs.

(2) Sample setup: To validate the effectiveness of our proposed model in dealing with balanced and unbalanced samples, we use unbalanced training samples from the Houston and Wuhan datasets and balanced training samples from the MUUFL dataset. In the Houston dataset and the Wuhan dataset, 5% and 1% of the total number of samples are considered as the training set, while 75% and 10% of the total number of samples are considered as the validation set, respectively. The MUUFL dataset uses 150 pixels per category as the training set and 95% of the total sample size as the validation set.

(3) Input patch size setting: The patch size in the input model influences the model's capacity to acquire receptive field and extract features from the original HSI-LiDAR data. A smaller patch can minimize the model's training time, but it will lose some contextual information. Larger patches can aid the model in detecting more spatial correlation information, but they can also reduce its computational speed and consume more memory. To establish the best patch input size, we conducted a series of experiments on three datasets, gradually increasing the patch size from 3 × 3 to 15 × 15, while accounting for the impacts of experimental accuracy and training time. Finally, in order to obtain the greatest compromise between experimental effect and training duration, we chose 11 × 11 as the patch input size for the three datasets.

(4) Learning rate setting: The learning rate controls the step size of the parameter update, and it controls the process of finding the minimum value of the loss function of the model in the parameter space. We tested five values of 1×10^{-3}, 1×10^{-4}, 5×10^{-4}, 1×10^{-5}, and 5×10^{-5}, and Figure 8 displays the performance of these three datasets under different learning rates. We set the learning rate to 1×10^{-4} for the Houston and

Wuhan datasets and 1×10^{-3} for the MUUFL dataset based on the results represented in the figure.

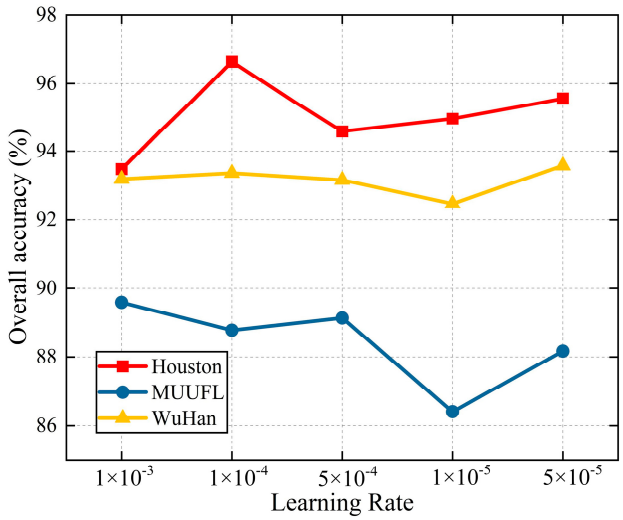


Figure 8. Comparison of the OA of three datasets under different learning rates.

(5) Evaluation criteria: In this experiment, the classification effect is evaluated using the overall accuracy (OA), average accuracy (AA), and Kappa coefficient. OA is the ratio of correctly categorized samples to all samples, and it is used to assess the classifier's overall classification ability. AA is the average accuracy of each category, which is capable of accurately reflecting how the classifier classified various categories. The confusion matrix, which may be used to assess the model's resilience and stability, is the basis for the calculation of the Kappa coefficient.

3.3. Comparison Methods

We chose seven traditional classifiers for comparison in order to verify the classification performance of the proposed multi-modal fusion framework, including 3D-CNN, DFFN, SSFTT, SpectralFormer, HRWN, MAHiDFNet, and GLT-Net.

1. 3D-CNN [48]: 3D convolution is created from 2D convolution to extract spectral and spatial features from the input data. In this study, a 3D-CNN with four convolutional layers and two pooling layers was used.
2. DFFN [49]: DFFN uses residual learning to optimize multiple convolutional layers in constant mappings, resulting in a deeper network while also facilitates network training.
3. SSFTT [50]: This model combines CNN and transformer to classify HSI data using spectral–spatial feature tokenization. Spatial–spectral feature extraction module first extracts shallow features, which are then turned into input features for transformer using a Gaussian weighted feature tokenizer, followed by feature learning and classification using transformer encoder.
4. SpectralFormer [51]: To learn the feature representation in HSI images, the SpectralFormer uses the transformer encoder. In contrast to VIT, which only takes into account spatial information, SpectralFormer takes into account both spectral and spatial information.
5. HRWN [31]: Hierarchical random walk network is a joint classifier for HSI and LiDAR data. HRWN employs a two-branch CNN to capture spectral and spatial information and uses a hierarchical random walk layer to explore the local similarity of pixel-level pairs.
6. MAHiDFNet [38]: The multi-attentive hierarchical fusion net realizes feature-level HSI image and LiDAR data fusion and classification by extracting the spatial–spectral information and elevation information separately through a three-branch network,

and then fusing the features using a hierarchical fusion strategy through Modal Attention Module (MA).

7. GLT-Net [39]: The global–local transformer network achieves full mining and utilization of global–local spectral spatial information and complementary information of multi-modal data by fully utilizing the advantages of convolution operators in characterizing locally relevant features and the potential of the transformer framework in learning long-range dependencies, thus realizing the fusion and classification of HSI and LiDAR data.

Based on the Houston dataset, Table 1 displays the quantitative evaluation findings of MMTCFN and seven other classical classifiers, and Figure 9 displays the classification maps produced by the eight methods. As noted in Table 1, MMTCFN outperformed all other classifiers in terms of accuracy, with average OA, AA, and Kappa coefficient accuracies of 96.63%, 96.09%, and 93.36%, respectively. However, SSFTT also performs well on this dataset, achieving the maximum precision in the categories of "Stressed grass (label 2)", "Synthetic grass (label 3)", "Soil (label 5)", and "Running track (label 15)". The classification findings of GLT-Net exhibit good consistency with the real labels and have high confidence, as indicated by the fact that its Kappa coefficient is the highest and is 0.76% higher than that of the suggested MMTCFN. It is worth mentioning that 3D-CNN performs quite poorly in the Houston dataset, earning an OA of only 65.13%, and that the average accuracy of all categories fail to exceed 80% during the validation process. This is because, when dealing with multi-modal fusion data with inadequate samples, the little amount of data and the diversity of data sources prevent 3D-CNN based on the conventional attention mechanism from producing the expected results. Nevertheless, by utilizing the self-attention mechanism, SSFTT, GLT-Net, and MMTCFN are able to adaptively correlate and model the variability between a variety of distinct data sources, improving classification accuracy. By analyzing Figure 9, it is clear that 3D-CNN has significant background noise issues. The multi-modal fusion data classifiers HRWN, MAHiDFNet, GLT-Net, and MMTCFN have a considerable advantage in extracting fused features, allowing them to provide a more fine-grained feature cover classification map.

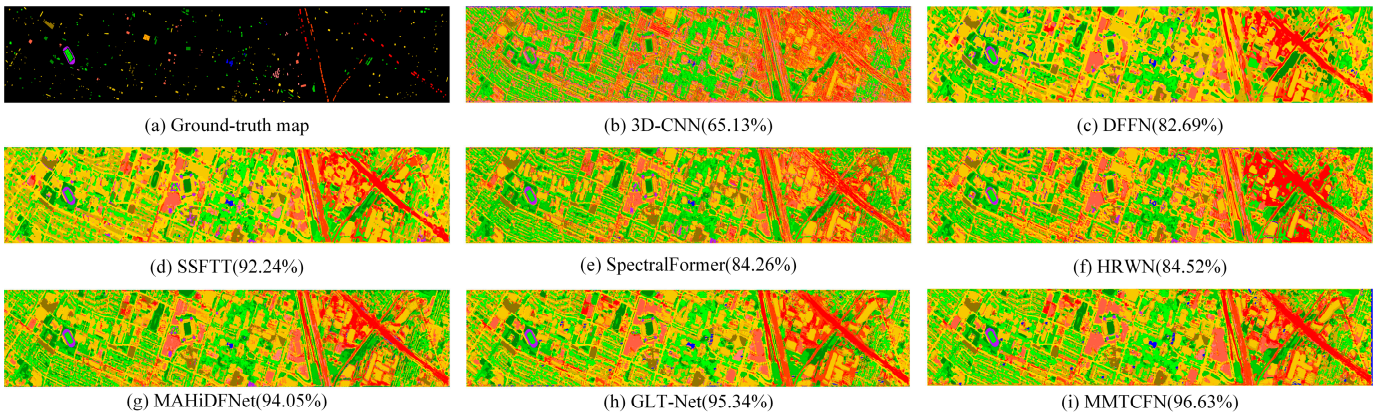


Figure 9. The land cover classification map generated through eight methods in the Houston dataset (5% of training set).

The accuracy evaluation and classification result graphs for the MUUFL dataset are displayed in Table 2 and Figure 10. We chose 150 pixels from each category to use as training samples in order to conduct the experiments with balanced samples. Table 2 shows that none of the methods achieve an OA of more than 90%, and our suggested method achieves better OA (89.59%), AA (91.29%), and Kappa coefficient (86.49%) than the others, demonstrating the proposed method's strong competitiveness in the classification of HSI-LiDAR multi-modal fusion data. Although GLT-Net, MAHiDFNet, and SSFTT achieved high accuracies that were second only to MMTCFN, their AAs were lower by 1.92%, 2.08%,

and 5.91%, respectively, than those of the suggested approaches. The classification diagram shows that for "Mostly grass" (label 2) and "Mixed ground surface" (label 3), there is a confounding phenomenon in all the approaches. This is because label 3 also has a ground surface that is partially covered in weeds.

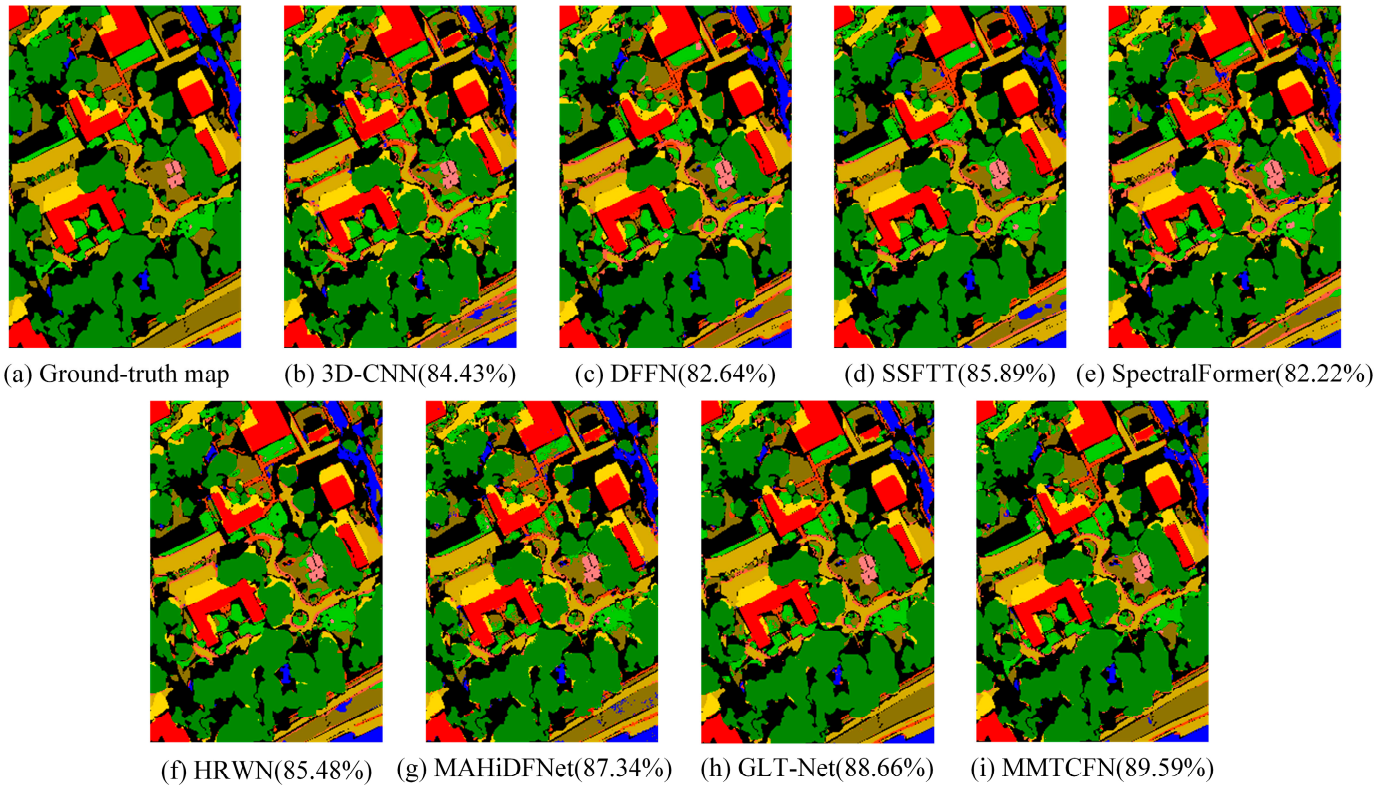


Figure 10. The land cover classification map generated through eight methods in the MUUFL dataset (consider 150 pixels as the training set).

The classification impact of the Wuhan dataset is displayed in Table 3 and Figure 11. This is the HSI-LiDAR dataset that we developed, with eight types of features manually highlighted and huge gaps between feature classes. We finally settled on using 0.1% of the total sample count as the training set and 10% of the total sample count as the validation and test sets after conducting numerous experiments. On this dataset, the approach proposed in this research produced the highest OA and Kappa coefficients, while GLT-Net produced the highest AA. However, the performance results of SpectralFormer were unsatisfactory, and overfitting occurred, and the integration of MMTCFN with Dropkey technology alleviated the problem of model non-convergence in the case of inadequate samples. Although 3D-CNN performs badly on the first two datasets, it performs well on the Wuhan dataset due to 3D-CNN's benefits in handling features with regular shapes and significant spectral variances. Figure 11 shows that, in terms of visual impact, the feature categorization maps created using all approaches are essentially similar with the information in Table 3. However, compared to models with a single input port, the HRWN, MAHiDFNet, and MMTCFN models with multiple branch input ports yield smoother and more precise results. This is because the multi-branch input decreases information loss, preserving and enhancing the model's capacity to learn from the data. This further demonstrates the importance of using multiple branches to extract and fuse spatial, spectral, and elevation information in the data, respectively.

Table 1. Quantitative evaluation of the Houston dataset using eight methods (%).

Label	Train	Test	Methods							
			3D-CNN	DFFN	SSFTT	SpectralFormer	HRWN	MAHiDFNet	GLT-Net	MMTCFN
1	68	961	80.12 ± 2.14	92.51 ± 3.78	93.44 ± 0.26	94.11 ± 1.08	91.59 ± 0.25	98.06 ± 0.70	94.15 ± 0.48	**98.65 ± 0.37**
2	72	1017	74.43 ± 2.57	85.72 ± 4.00	**98.74 ± 0.17**	98.70 ± 0.70	96.22 ± 0.05	97.09 ± 0.28	96.54 ± 0.21	98.33 ± 0.35
3	39	556	67.19 ± 1.61	99.03 ± 1.68	**99.89 ± 0.09**	96.47 ± 2.25	96.76 ± 0.30	97.27 ± 0.85	99.12 ± 1.88	98.13 ± 0.54
4	63	884	74.64 ± 3.98	86.61 ± 2.60	94.68 ± 0.21	89.86 ± 4.88	88.37 ± 0.38	97.17 ± 0.86	98.05 ± 1.09	**98.19 ± 0.46**
5	64	908	70.93 ± 2.44	96.81 ± 0.84	**99.80 ± 0.08**	98.04 ± 0.73	98.61 ± 0.11	99.76 ± 0.27	99.10 ± 0.75	99.74 ± 0.32
6	16	237	46.67 ± 2.87	64.47 ± 15.37	78.14 ± 1.54	75.70 ± 3.53	60.59 ± 0.34	83.21 ± 2.32	**97.14 ± 0.62**	89.70 ± 3.27
7	73	1033	76.67 ± 1.55	83.33 ± 2.76	92.74 ± 0.28	82.19 ± 2.17	88.11 ± 0.13	94.68 ± 0.46	94.64 ± 1.25	**94.83 ± 2.20**
8	67	947	56.24 ± 1.88	69.17 ± 6.55	79.66 ± 0.99	79.62 ± 2.56	64.73 ± 1.66	91.64 ± 1.16	90.59 ± 1.52	**95.97 ± 1.08**
9	77	1087	64.10 ± 1.58	77.44 ± 2.79	93.36 ± 0.27	74.06 ± 1.75	80.37 ± 0.35	89.46 ± 1.50	94.28 ± 0.33	**95.38 ± 1.49**
10	71	996	53.59 ± 4.47	57.75 ± 10.68	88.63 ± 0.39	81.99 ± 2.17	73.59 ± 0.47	91.27 ± 1.19	91.92 ± 1.94	**97.17 ± 0.36**
11	78	1096	57.66 ± 1.56	64.09 ± 5.34	94.29 ± 0.23	74.42 ± 1.13	80.53 ± 0.19	95.26 ± 0.40	96.54 ± 0.95	**97.10 ± 1.18**
12	71	1000	56.04 ± 4.59	81.06 ± 5.25	78.52 ± 0.56	81.86 ± 4.54	75.32 ± 0.39	86.88 ± 1.12	92.68 ± 1.77	**95.38 ± 2.93**
13	31	442	72.35 ± 1.30	76.02 ± 9.51	95.16 ± 0.44	30.09 ± 3.33	80.36 ± 0.33	89.50 ± 2.54	**98.40 ± 0.46**	83.71 ± 4.84
14	25	359	48.91 ± 2.01	89.36 ± 2.44	97.72 ± 0.48	91.03 ± 4.39	98.61 ± 0.00	96.10 ± 2.24	98.69 ± 1.18	**99.05 ± 0.67**
15	39	558	55.66 ± 1.11	96.16 ± 3.37	**100.00 ± 0.00**	99.07 ± 0.44	93.01 ± 0.16	98.42 ± 0.64	99.96 ± 1.38	99.96 ± 0.07
	OA		65.13 ± 0.20	82.69 ± 2.68	92.24 ± 0.15	84.26 ± 1.24	84.52 ± 0.17	94.05 ± 0.34	95.34 ± 0.19	**96.63 ± 0.35**
	AA		63.28 ± 0.25	81.30 ± 3.09	92.32 ± 0.14	83.15 ± 1.41	84.45 ± 0.16	93.72 ± 0.47	95.04 ± 0.48	**96.09 ± 0.43**
	K × 100		62.26 ± 0.22	79.12 ± 2.90	91.62 ± 0.17	82.98 ± 1.34	83.26 ± 0.19	93.57 ± 0.37	**94.12 ± 0.68**	93.36 ± 0.38

Table 2. Quantitative evaluation of the MUUFL dataset using eight methods (%).

Label	Train	Test	Methods							
			3D-CNN	DFFN	SSFTT	SpectralFormer	HRWN	MAHiDFNet	GLT-Net	MMTCFN
1	150	22,083	89.35 ± 0.05	88.24 ± 2.10	92.20 ± 0.03	89.15 ± 0.95	90.10 ± 0.69	89.95 ± 0.98	91.28 ± 0.59	**92.27 ± 0.81**
2	150	4056	81.04 ± 0.09	73.79 ± 1.00	78.50 ± 0.12	71.09 ± 2.16	79.70 ± 2.16	72.58 ± 5.77	79.63 ± 3.63	**83.08 ± 4.70**
3	150	6537	61.84 ± 0.10	63.15 ± 3.69	61.91 ± 0.08	61.32 ± 3.35	67.73 ± 6.63	78.19 ± 3.83	74.29 ± 1.84	**79.36 ± 1.21**
4	150	1734	88.02 ± 0.08	85.78 ± 2.50	**92.92 ± 0.21**	80.99 ± 6.04	90.72 ± 0.54	91.22 ± 3.51	91.48 ± 1.74	92.54 ± 3.65
5	150	6352	80.83 ± 0.06	79.21 ± 1.35	85.37 ± 0.06	77.02 ± 1.51	81.73 ± 1.64	86.97 ± 1.45	**88.90 ± 0.64**	87.74 ± 1.42
6	150	442	99.41 ± 0.11	98.73 ± 0.23	99.77 ± 0.00	**99.91 ± 0.18**	99.41 ± 0.18	99.28 ± 0.09	98.39 ± 0.34	99.50 ± 0.17

Table 2. *Cont.*

Label	Train	Test	Methods							
			3D-CNN	**DFFN**	**SSFTT**	**SpectralFormer**	**HRWN**	**MAHiDFNet**	**GLT-Net**	**MMTCFN**
7	150	2121	93.14 ± 0.18	88.28 ± 0.78	91.75 ± 0.25	87.92 ± 0.69	90.10 ± 2.01	**94.91 ± 1.21**	91.43 ± 1.54	94.38 ± 1.69
8	150	5928	94.62 ± 0.05	94.55 ± 0.55	95.46 ± 0.05	95.02 ± 0.22	94.77 ± 0.40	94.71 ± 1.32	**96.48 ± 1.09**	94.82 ± 0.78
9	150	1315	68.53 ± 0.33	53.32 ± 2.80	55.03 ± 0.13	54.10 ± 4.73	68.06 ± 6.16	78.81 ± 3.04	71.74 ± 1.39	**83.24 ± 2.58**
10	150	173	89.83 ± 0.28	87.75 ± 1.23	86.71 ± 0.52	87.86 ± 1.75	90.64 ± 0.85	96.18 ± 1.53	95.24 ± 1.71	**98.38 ± 0.57**
11	150	255	99.37 ± 0.19	98.75 ± 0.38	**99.61 ± 0.00**	98.20 ± 0.91	98.59 ± 0.19	98.51 ± 1.30	99.49 ± 1.52	98.90 ± 1.48
	OA		84.43 ± 0.03	82.64 ± 1.11	85.89 ± 0.02	82.22 ± 0.96	85.48 ± 1.39	87.34 ± 0.53	88.66 ± 0.12	**89.59 ± 0.38**
	AA		86.00 ± 0.06	82.87 ± 0.52	85.38 ± 0.07	82.05 ± 1.14	86.50 ± 1.44	89.21 ± 0.44	89.37 ± 0.59	**91.29 ± 0.90**
	K × 100		79.97 ± 0.04	77.72 ± 1.33	81.69 ± 0.03	77.08 ± 1.20	81.27 ± 1.75	83.65 ± 0.65	84.29 ± 1.17	**86.49 ± 0.48**

Table 3. Quantitative evaluation of the Wuhan dataset using eight methods (%).

Label	Train	Test	Methods							
			3D-CNN	**DFFN**	**SSFTT**	**SpectralFormer**	**HRWN**	**MAHiDFNet**	**GLT-Net**	**MMTCFN**
1	377	37,755	93.23 ± 0.46	90.92 ± 1.46	93.93 ± 0.37	89.04 ± 0.83	91.58 ± 0.43	95.21 ± 0.50	**95.41 ± 0.68**	94.70 ± 0.42
2	36	3688	35.82 ± 3.95	61.25 ± 17.37	77.99 ± 1.09	00.00 ± 0.0	34.84 ± 7.67	77.62 ± 3.46	76.94 ± 1.10	**79.37 ± 1.82**
3	86	8674	82.61 ± 0.91	81.36 ± 2.27	87.08 ± 0.46	55.25 ± 20.14	82.65 ± 0.46	88.10 ± 0.78	87.49 ± 0.70	**88.48 ± 1.26**
4	128	12,809	89.77 ± 0.59	91.73 ± 0.69	86.91 ± 0.27	89.89 ± 1.23	91.47 ± 0.32	**93.52 ± 0.93**	**93.52 ± 0.54**	92.50 ± 0.64
5	360	36,021	92.67 ± 0.12	90.47 ± 0.85	91.88 ± 0.37	88.59 ± 0.53	90.97 ± 0.21	**94.58 ± 0.58**	93.69 ± 0.16	93.42 ± 0.28
6	156	15,615	83.25 ± 1.48	84.60 ± 2.32	**92.22 ± 0.10**	64.37 ± 2.51	87.53 ± 0.29	89.56 ± 1.17	88.18 ± 0.82	90.61 ± 0.72
7	73	7344	84.90 ± 0.87	84.06 ± 1.76	88.01 ± 1.36	61.04 ± 6.04	84.55 ± 0.27	89.33 ± 0.76	87.24 ± 0.97	**90.20 ± 0.73**
8	433	43,376	94.86 ± 0.19	95.04 ± 0.22	97.24 ± 0.17	96.25 ± 0.46	95.78 ± 0.15	95.67 ± 0.37	96.66 ± 0.21	**97.30 ± 0.09**
	OA		90.12 ± 0.18	89.93 ± 0.76	92.70 ± 0.10	83.57 ± 1.22	90.11 ± 0.23	92.78 ± 0.18	92.22 ± 0.02	**93.36 ± 0.14**
	AA		82.14 ± 0.53	85.00 ± 2.48	89.42 ± 0.09	68.05 ± 3.14	82.42 ± 1.01	90.10 ± 0.68	**91.19 ± 0.34**	90.07 ± 0.33
	K × 100		87.74 ± 0.23	87.54 ± 0.95	90.96 ± 0.13	79.39 ± 1.61	87.75 ± 0.28	91.33 ± 0.23	91.60 ± 0.99	**91.78 ± 0.17**

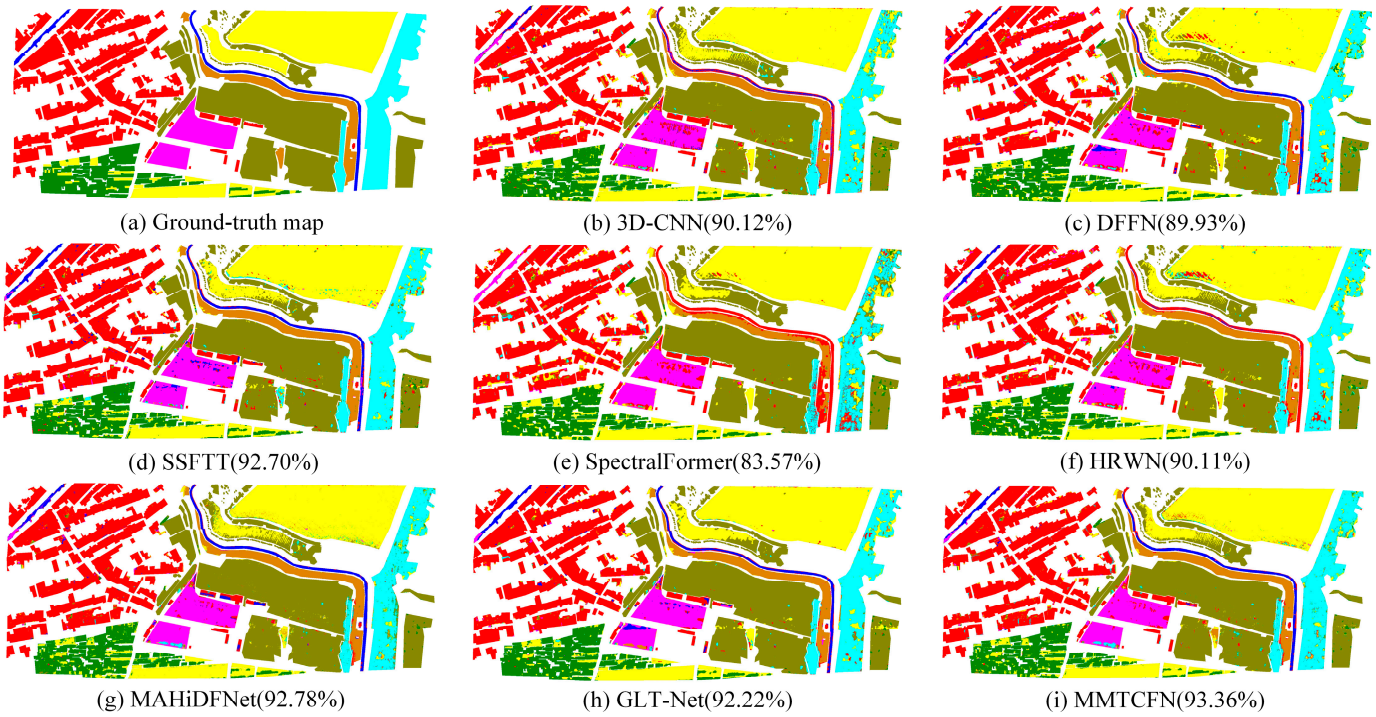


Figure 11. The land cover classification map generated through eight methods in the Wuhan dataset (0.1% of training set).

In conclusion, MMTCFN performs well across the three datasets and generates feature classification maps with the most visual resemblance to ground truth. Multiple experiments have shown that the fusion and categorization of HSI-LiDAR data can be of great potential with the help of MMTCFN.

3.4. Discussion

3.4.1. Ablation Experiments

We conducted various ablation experiments utilizing three datasets to confirm the validity of the whole proposed model. Next, this is covered in two parts.

(1) With/without LiDAR-based DSM: The outcomes with and without LiDAR-based DSM ablation trials are shown in Table 4 and Figure 12. According to the results in Table 4, the OA is increased in each of the three datasets with HSI-LiDAR data input by 2.71%, 2.94%, and 5.93% when compared to HSI data alone. In particular, the improvement is most significant in the Wuhan dataset. Figure 12 demonstrates the effect of having and not having LiDAR-based DSM on the classification accuracy for each category. The classification effect after adding LiDAR-based DSM is clearly superior to that of using simply HSI data for classification for the majority of features, as can be shown in the figure. However, for some highly insensitive features, such as "water (label 6)" in the Houston dataset and "mostly grass (label 2)" in the MUUFL dataset, their spatial variation is small, adding LiDAR-based DSM may introduce noise and unnecessary information, thus affecting the classification effect.

Table 4. Ablation experiments on the presence or absence of LiDAR-based DSM on three datasets. (%).

	Houston			**MUUFL**			**Wuhan**		
	OA	**AA**	**Kappa**	**OA**	**AA**	**Kappa**	**OA**	**AA**	**Kappa**
Without LiDAR-based DSM	93.92	93.46	93.43	86.65	90.31	82.84	87.43	80.46	84.43
With HSI-LiDAR	96.63	96.09	93.36	89.59	91.29	86.49	93.36	90.07	91.78

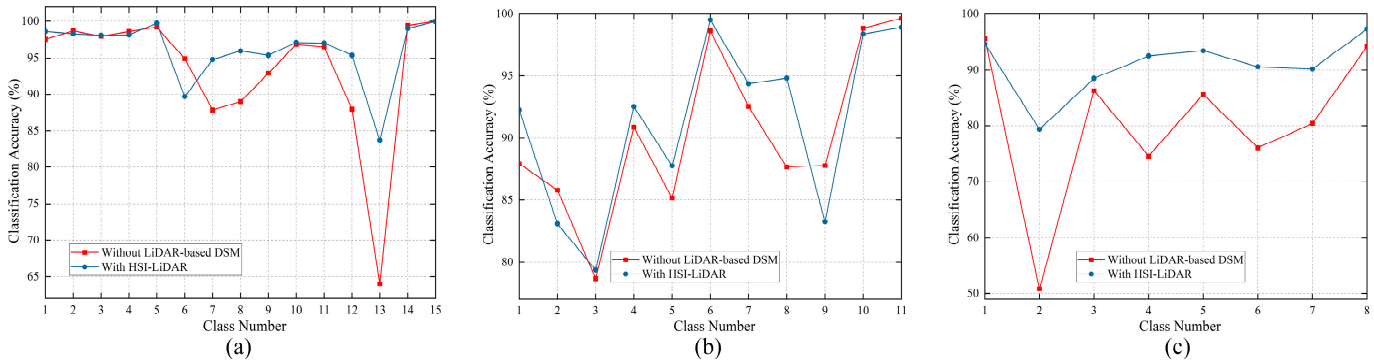


Figure 12. The impact of LiDAR-DSM on the classification results of MMTCFN. (**a**) Houston. (**b**) MUUFL. (**c**) Wuhan.

T-distributed stochastic neighbor embedding (t-SNE), a nonlinear dimensionality reduction approach, is used to visualize high-dimensional data in a two-dimensional space and determine the separability across categories through observation. The results of visualization without LiDAR-based DSM visualization are shown in Figure 13a–c, whereas the results of visualization with HSI-LiDAR are shown in Figure 13g–i. The three datasets show that the addition of LiDAR-based DSM creates a more distinct boundary of separability between features. This is because of the fact that HSI alone is not sensitive enough to the 3D morphological information of the features, whereas LiDAR-based DSM can provide highly accurate morphological and elevation features, and the combination of HSI and LiDAR-based DSM can obtain more comprehensive and detailed 3D spatial information while retaining the rich spectral information.

(2) With/without VPGT: The most crucial part of the proposed MMTCFN classifier is the VPGT module, which plays a vital role in the accuracy of the classification results by feature fusion and thus extracting deep features. The quantitative evaluation of the accuracy of with/without VPGT is shown in Table 5 and Figure 14, and the findings reveal that the VPGT plays a substantial role in enhancing the majority of the features from the three datasets' classification accuracy. As shown in Figure 14b, a total of eight feature classes had the highest OA in the results using VPGT in the MUUFL dataset. In addition, VPGT pays more attention to those features with similar spectral characteristics but different spatial information, in which case VPGT introduces the stereo features of LiDAR data into HSI to help distinguish between, for example, "Residential (label 7)" and "Commercial (label 8)" in the Houston dataset. Figure 13d–f display the visualization results without VPGT, allowing one to observe the serious feature mixing present in the MUUFL dataset and Wuhan dataset. In the Wuhan dataset, none of the features are separable from each other, and the correlation between each of the two features "vegetation (label 1)" and "parking lot (label 6)" is weak. The Kappa coefficient of the Houston dataset without VPGT is higher than that with VPGT, which indicates that the Houston dataset also has a strong classification ability without VPGT, and thus the boundary between different features in Figure 13d is clearer. Overall, the model including VPGT enhances feature fusion capability and successfully optimizes the results of feature categorization.

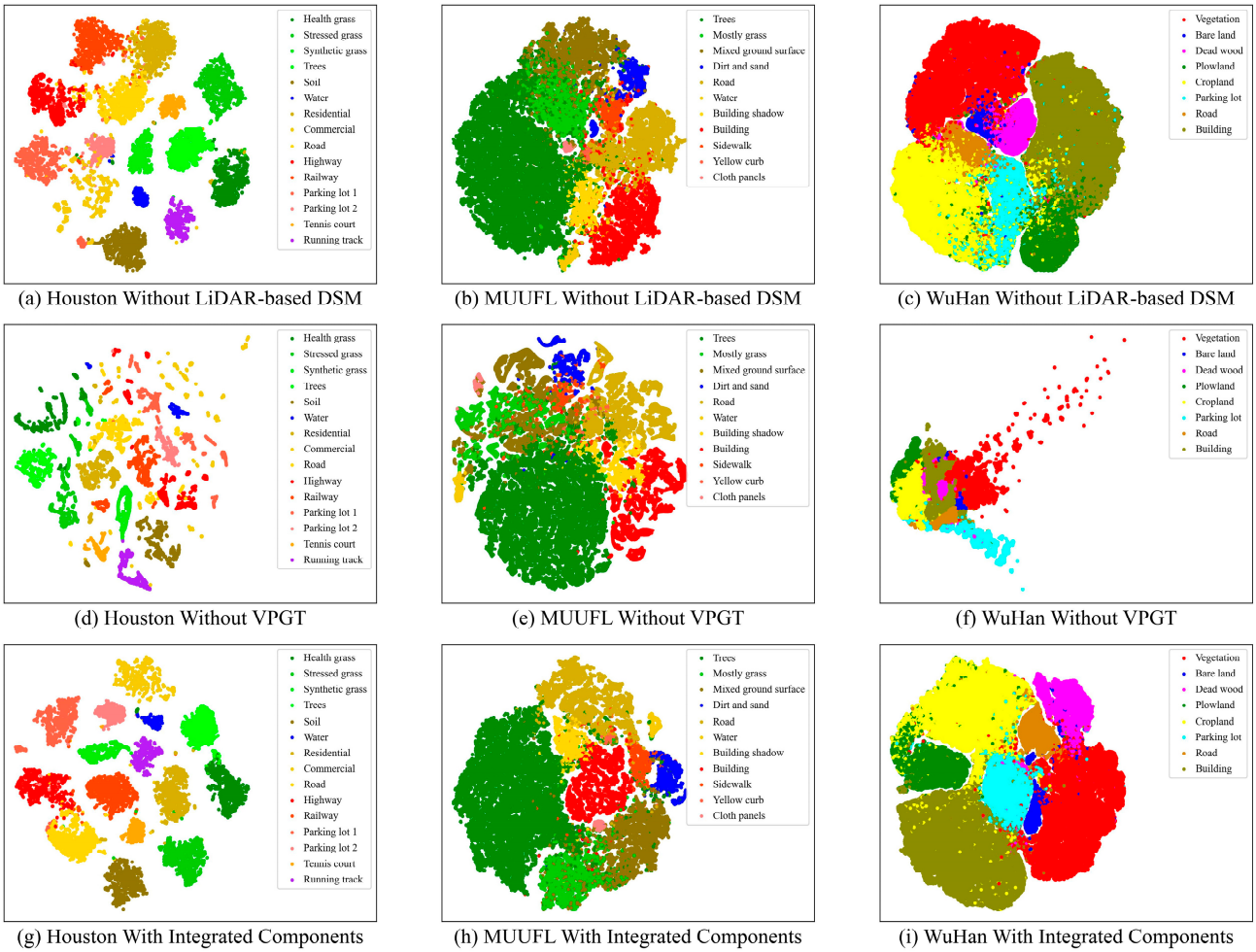


Figure 13. T-SNE visualization of dimensionality reduction for high-dimensional data3.

Table 5. Ablation experiments on the presence or absence of VPGT on the three datasets (%).

	Houston			MUUFL			Wuhan		
	OA	**AA**	**Kappa**	**OA**	**AA**	**Kappa**	**OA**	**AA**	**Kappa**
Without VPGT	94.59	93.75	94.40	87.01	88.08	83.77	90.64	88.01	89.62
With VPGT	96.63	96.09	93.36	89.59	91.29	86.49	93.36	90.07	91.78

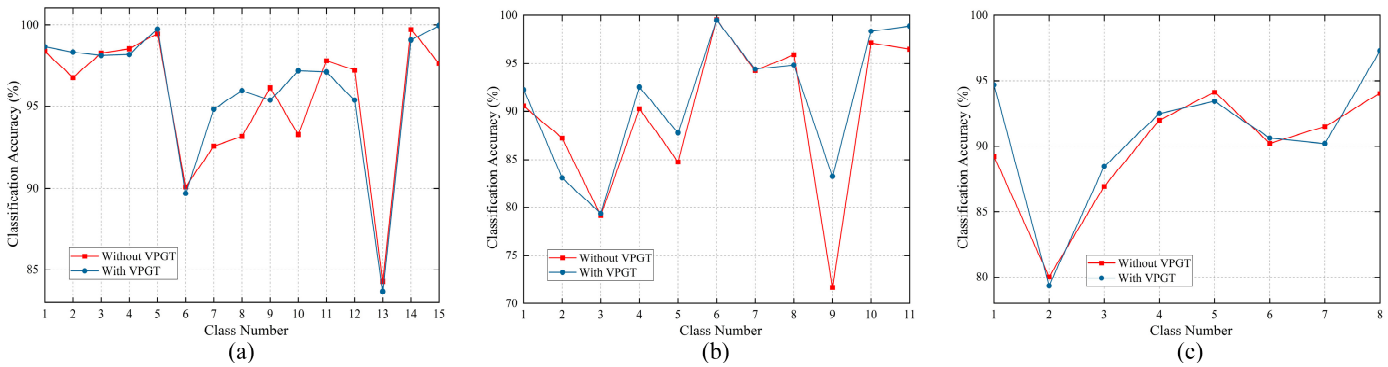


Figure 14. The impact of VPGT on the classification results of MMTCFN. (**a**) Houston. (**b**) MUUFL. (**c**) Wuhan.

3.4.2. Analyzing the Effect of DropKey Ratio

DropKey is a novel regularizer that can be used in MHSA to effectively mitigate the overfitting problem in the case of insufficient samples. The impact of various DropKey ratios on the classification accuracy of the three datasets, with DropKey ratios ranging from 0.1 to 0.9, is investigated in this experiment. The outcomes of the three datasets in OA are displayed in Table 6. The table clearly shows that as the DropKey ratio rises, the total classification accuracy of the three datasets tends to rise and subsequently fall. The best training outcomes were obtained with the MUUFL and Wuhan datasets at DropKeys of 0.7 and 0.6, respectively, whereas Houston had the highest OA at a DropKey of 0.3. Therefore, for the Houston, MUFFL, and Wuhan datasets, their DropKey ratios were set to 0.3, 0.7, and 0.6, respectively.

Table 6. The analysis was conducted on the use of different DropKey ratios across the three datasets (%).

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Houston	96.28	96.51	96.63	96.40	96.10	95.85	96.39	96.20	95.74
MUUFL	88.46	87.87	88.58	88.37	88.22	89.12	89.59	89.11	88.98
Wuhan	92.96	93.17	93.22	92.36	93.12	93.36	93.42	93.61	93.34

3.4.3. Robustness Evaluation

We conducted experiments on three datasets using progressively less training samples to compare with seven other distinct methods in order to verify the robustness of the proposed MMTCFN. For the three datasets, we used 75%, 50%, and 25% fewer training samples than originally planned. In particular, we used 0.075%, 0.05%, and 0.025% of the labeled samples as training sets for Wuhan and 3.75%, 2.5%, and 1.25% of the data as training samples for Houston. As balanced training samples, we chose 112, 75, and 38 pixels from each category in the MUUFL dataset. The overall accuracy results of this experiment are displayed in Figure 15. We observe that, when compared to the other seven examined approaches, the proposed MMTCFN performs better under various training sample ratios. We find that the proposed MMTCFN approach can significantly outperform the other methods at all four sample ratios, despite the Houston dataset having the fewest pixels in the labeled samples. Additionally, the accuracy of MMTCFN has the least declining trend as the training sample ratio gradually declines. This is particularly evident in the Wuhan dataset, further demonstrating the proposed MMTCFN's strong robustness.

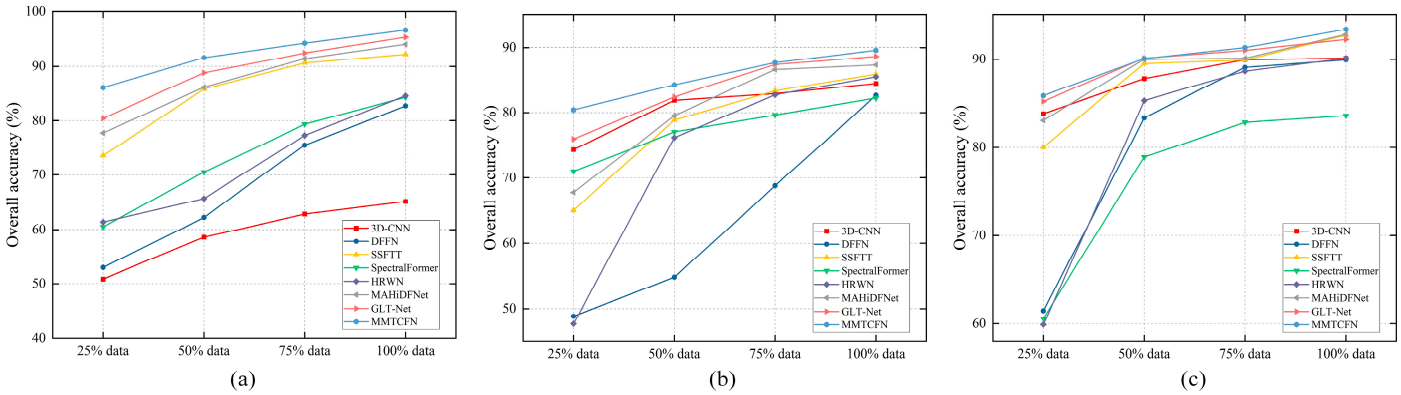


Figure 15. Performance of OA with different proportions of training samples on three datasets. (**a**) Houston. (**b**) MUUFL. (**c**) Wuhan.

4. Conclusions

In this study, the MMTCFN model is proposed for the fusion and classification of multimodal remote sensing data. Two stages of feature extraction and feature fusion are present in the model. To begin, the feature extraction module uses a three-branch cascade CNN

framework to extract shallow characteristics from HSI-LiDAR data, such as spatial features, spectral features, and 3D topography features. The recognition and classification accuracy can be improved by using the three-branch cascade CNN by obtaining more detailed and rich feature information. On this basis, we employ the VPGT block in the feature fusion stage to generate multi-modal long-range integrated features. We created a vectorized pixel group embedding for the VPGT block to preserve the global detail information of the feature map in the form of non-overlapping multiple groups. Additionally, we employ the transformer model, which combines MHSA and MLP, to fully exploit the correlation and heterogeneity among multi-modal features to interact with and integrate various features in order to provide more expressive and discriminative feature representations. Among them, we include the DropKey technique in MHSA to alleviate the overfitting issue. We contrast the proposed MMTCFN approach with seven additional SOTA algorithms on three HSI-LiDAR datasets. The experimental results demonstrate that the proposed approach performs better than existing methods and has tremendous potential for HSI-LiDAR data fusion and classification tasks.

Author Contributions: Conceptualization, S.W. and Z.L.; Methodology, S.W. and Z.L.; Validation, S.W. and Z.L.; Investigation, S.W., Z.Z. and G.Z.; Data curation, S.W. and C.H.; Writing—original draft, S.W.; Writing—review and editing, S.W. and Z.Z.; Supervision, S.W., Y.C. and Z.L.; Funding acquisition, Y.C. and Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China (2018YFB0504504) and the Funded Project of Fundamental Scientific Research Business Expenses of Chinese Academy of Surveying and Mapping (AR2201; AR2203).

Data Availability Statement: The Houston dataset used in this study is available at https://hyperspectral.ee.uh.edu/?page_id=1075 (accessed on 7 March 2023); the MUUFL dataset is available from https://github.com/GatorSense/MUUFLGulfport/ (accessed on 7 March 2023); the Wuhan dataset in this study is provided by Professor Wei Gong′s team at Wuhan University and is not shared with others.

Acknowledgments: We are extremely grateful to Professor Wei Gong′s team at Wuhan University for providing the Wuhan dataset. Their generous support and contribution have played a significant role in our research. This dataset will provide valuable resources and references for our study. Once again, we would like to express our heartfelt gratitude to Professor Wei Gong and his team.

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

References

1. Ding, Z.; Liao, X.; Su, F.; Fu, D. Mining Coastal Land Use Sequential Pattern and Its Land Use Associations Based on Association Rule Mining. *Remote Sens.* **2017**, *9*, 116. [CrossRef]
2. Luo, B.; Zhang, F.; Liu, X.; Pan, Q.; Guo, P. Managing Agricultural Water Considering Water Allocation Priority Based on Remote Sensing Data. *Remote Sens.* **2021**, *13*, 1536. [CrossRef]
3. Mo, K.; Cong, Z.; Lei, H. Optimal vegetation cover in the Horqin Sands, China. *Ecohydrology* **2015**, *9*, 700–711. [CrossRef]
4. Cao, W.; Dong, L.; Wu, L.; Liu, Y. Quantifying urban areas with multi-source data based on percolation theory. *Remote Sens. Environ.* **2020**, *241*, 111730. [CrossRef]
5. Cheng, Y.; Zhou, K.; Wang, J.; Yan, J. Big Earth Observation Data Integration in Remote Sensing Based on a Distributed Spatial Framework. *Remote Sens.* **2020**, *12*, 972. [CrossRef]
6. Tong, X.-Y.; Xia, G.-S.; Lu, Q.; Shen, H.; Li, S.; You, S.; Zhang, L. Land-cover classification with high-resolution remote sensing images using transferable deep models. *Remote Sens. Environ.* **2020**, *237*, 111322. [CrossRef]
7. Roy, S.K.; Deria, A.; Hong, D.; Ahmad, M.; Plaza, A.; Chanussot, J. Hyperspectral and LiDAR Data Classification Using Joint CNNs and Morphological Feature Learning. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5530416. [CrossRef]
8. Zhang, Z.; Li, T.; Tang, X.; Lei, X.; Peng, Y. Introducing Improved Transformer to Land Cover Classification Using Multispectral LiDAR Point Clouds. *Remote Sens.* **2022**, *14*, 3808. [CrossRef]
9. Liu, Q.; Xue, D.; Tang, Y.; Zhao, Y.; Ren, J.; Sun, H. PSSA: PCA-Domain Superpixelwise Singular Spectral Analysis for Unsupervised Hyperspectral Image Classification. *Remote Sens.* **2023**, *15*, 890. [CrossRef]
10. Uchaev, D.; Uchaev, D. Small Sample Hyperspectral Image Classification Based on the Random Patches Network and Recursive Filtering. *Sensors* **2023**, *23*, 2499. [CrossRef]

11. Uddin, P.; Mamun, A.; Hossain, A. Effective feature extraction through segmentation-based folded-PCA for hyperspectral image classification. *Int. J. Remote Sens.* **2019**, *40*, 7190–7220. [CrossRef]
12. Li, W.; Prasad, S.; Fowler, J.E.; Bruce, L.M. Locality-Preserving Discriminant Analysis in Kernel-Induced Feature Spaces for Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2011**, *8*, 894–898. [CrossRef]
13. Li, X.; Zhang, L.; You, J. Locally Weighted Discriminant Analysis for Hyperspectral Image Classification. *Remote Sens.* **2019**, *11*, 109. [CrossRef]
14. Sun, W.; Halevy, A.; Benedetto, J.J.; Czaja, W.; Liu, C.; Wu, H.; Shi, B.; Li, W. UL-Isomap based nonlinear dimensionality reduction for hyperspectral imagery classification. *ISPRS J. Photogramm. Remote Sens.* **2014**, *89*, 25–36. [CrossRef]
15. Bigdeli, B.; Samadzadegan, F.; Reinartz, P. Fusion of hyperspectral and LIDAR data using decision template-based fuzzy multiple classifier system. *Int. J. Appl. Earth Obs. Geoinf.* **2015**, *38*, 309–320. [CrossRef]
16. Li, Q.; Wong, F.K.K.; Fung, T. Mapping multi-layered mangroves from multispectral, hyperspectral, and LiDAR data. *Remote Sens. Environ.* **2021**, *258*, 112403. [CrossRef]
17. de Almeida, C.T.; Galvão, L.S.; de Oliveira Cruz e Aragão, L.E.; Ometto, J.P.H.B.; Jacon, A.D.; de Souza Pereira, F.R.; Sato, L.Y.; Lopes, A.P.; Lima de Alencastro Graça, P.M.; Silva, C.V.d.J.; et al. Combining LiDAR and hyperspectral data for aboveground biomass modeling in the Brazilian Amazon using different regression algorithms. *Remote Sens. Environ.* **2019**, *232*, 111323. [CrossRef]
18. Nimbalkar, P.; Jarocinska, A.; Zagajewski, B. Optimal Band Configuration for the Roof Surface Characterization Using Hyperspectral and LiDAR Imaging. *J. Spectrosc.* **2018**, *2018*, 6460518. [CrossRef]
19. Jiang, H.; Peng, M.; Zhong, Y.; Xie, H.; Hao, Z.; Lin, J.; Ma, X.; Hu, X. A Survey on Deep Learning-Based Change Detection from High-Resolution Remote Sensing Images. *Remote Sens.* **2022**, *14*, 1552. [CrossRef]
20. Li, K.; Wan, G.; Cheng, G.; Meng, L.; Han, J. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS J. Photogramm. Remote Sens.* **2020**, *159*, 296–307. [CrossRef]
21. Kuang, H.; Wu, J. Survey of Image Semantic Segmentation Based on Deep Learning. *Comput. Eng. Appl.* **2019**, *55*, 12–21+42.
22. Wang, J.-X.; Chen, S.-B.; Ding, C.H.Q.; Tang, J.; Luo, B. Semi-Supervised Semantic Segmentation of Remote Sensing Images With Iterative Contrastive Network. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 2504005. [CrossRef]
23. Bashir, S.M.A.; Wang, Y.; Khan, M.; Niu, Y. A comprehensive review of deep learning-based single image super-resolution. *PeerJ Comput. Sci.* **2021**, *7*, e621. [CrossRef]
24. Fu, Y.; Zhang, X.; Wang, M. Super-Resolution Reconstruction of Remote Sensing Images Using Generative Adversarial Network With Shallow Information Enhancement. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *15*, 8529–8540. [CrossRef]
25. Li, J.; Hong, D.; Gao, L.; Yao, J.; Zheng, K.; Zhang, B.; Chanussot, J. Deep learning in multimodal remote sensing data fusion: A comprehensive review. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *112*, 102926. [CrossRef]
26. Xu, P.; Zhu, X.; Clifton, D.A. Multimodal Learning with Transformers: A Survey. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*; IEEE: Piscataway, NJ, USA, 2023; pp. 1–20. [CrossRef] [PubMed]
27. Zhang, M.; Li, W.; Du, Q.; Gao, L.; Zhang, B. Feature Extraction for Classification of Hyperspectral and LiDAR Data Using Patch-to-Patch CNN. *IEEE Trans. Cybern.* **2018**, *50*, 100–111. [CrossRef] [PubMed]
28. Rasti, B.; Ghamisi, P.; Plaza, J.; Plaza, A. Fusion of Hyperspectral and LiDAR Data Using Sparse and Low-Rank Component Analysis. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 6354–6365. [CrossRef]
29. Xia, J.; Liao, W.; Du, P. Hyperspectral and LiDAR Classification with Semisupervised Graph Fusion. *IEEE Geosci. Remote Sens. Lett.* **2019**, *17*, 666–670. [CrossRef]
30. Hang, R.; Li, Z.; Ghamisi, P.; Hong, D.; Xia, G.; Liu, Q. Classification of Hyperspectral and LiDAR Data Using Coupled CNNs. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 4939–4950. [CrossRef]
31. Zhao, X.; Tao, R.; Li, W.; Li, H.-C.; Du, Q.; Liao, W.; 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]
32. Zhu, Y.; Li, W.; Zhang, M.; Pang, Y.; Tao, R.; Du, Q. Joint feature extraction for multi-source data using similar double-concentrated network. *Neurocomputing* **2021**, *450*, 70–79. [CrossRef]
33. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2021, Nashville, TN, USA, 20–25 June 2021; pp. 13708–13717.
34. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
35. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the Computer Vision—ECCV 2018, PT VII, Munich, Germany, 8–14 September 2018; pp. 3–19.
36. 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 Networks Learn. Syst.* **2020**, *33*, 747–761. [CrossRef]
37. Mohla, S.; Pande, S.; Banerjee, B.; Chaudhuri, S. FusAtNet: Dual Attention based SpectroSpatial Mul-ti-modal Fusion Network for Hyperspectral and LiDAR Classification. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 2020), Seattle, WA, USA, 14–19 June 2020; pp. 416–425.
38. Wang, X.; Feng, Y.; Song, R.; Mu, Z.; Song, C. Multi-attentive hierarchical dense fusion net for fusion classification of hyperspectral and LiDAR data. *Inf. Fusion* **2021**, *82*, 1–18. [CrossRef]

39. Ding, K.; Lu, T.; Fu, W.; Li, S.; Ma, F. Global–Local Transformer Network for HSI and LiDAR Data Joint Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5541213. [CrossRef]
40. Zhang, Y.; Peng, Y.; Tu, B.; Liu, Y. Local Information Interaction Transformer for Hyperspectral and LiDAR Data Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *16*, 1130–1143. [CrossRef]
41. Ghosh, K.; Bellinger, C.; Corizzo, R.; Branco, P.; Krawczyk, B.; Japkowicz, N. The class imbalance problem in deep learning. *Mach. Learn.* **2022**, 1–57. [CrossRef]
42. Lin, E.; Chen, Q.; Qi, X. Deep reinforcement learning for imbalanced classification. *Appl. Intell.* **2020**, *50*, 2488–2502. [CrossRef]
43. Sarker, I.H. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Comput. Sci.* **2021**, *2*, 420. [CrossRef]
44. Zhu, X.X.; Tuia, D.; Mou, L.; Xia, G.-S.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 8–36. [CrossRef]
45. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
46. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2021**, arXiv:2010.11929.
47. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
48. Ben Hamida, A.; Benoit, A.; Lambert, P.; Ben Amar, C. 3-D Deep Learning Approach for Remote Sensing Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 4420–4434. [CrossRef]
49. Song, W.; Li, S.; Fang, L.; Lu, T. Hyperspectral Image Classification with Deep Feature Fusion Network. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 3173–3184. [CrossRef]
50. Sun, L.; Zhao, G.; Zheng, Y.; Wu, Z. Spectral–Spatial Feature Tokenization Transformer for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5522214. [CrossRef]
51. Hong, D.; Han, Z.; Yao, J.; Gao, L.; Zhang, B.; Plaza, A.; Chanussot, J. SpectralFormer: Rethinking Hyperspectral Image Classification with Transformers. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5518615. [CrossRef]