remote sensing

Communication

Attention Fusion of Transformer-Based and Scale-Based Method for Hyperspectral and LiDAR Joint Classification

Maqun Zhang [1,2], Feng Gao [1,2], Tiange Zhang [1,2], Yanhai Gan [1,2], Junyu Dong [1,2,*] and Hui Yu [3]

[1] College of Information Science and Engineering, Ocean University of China, Qingdao 266100, China
[2] Institute of Marine Development, Ocean University of China, Qingdao 266100, China
[3] Faculty of Creative & Cultural Industries, University of Portsmouth, Portsmouth PO1 2DJ, UK
[*] Correspondence: dongjunyu@ouc.edu.cn; Tel.: +86-66781123

Abstract: In recent years, there have been many multimodal works in the field of remote sensing, and most of them have achieved good results in the task of land-cover classification. However, multi-scale information is seldom considered in the multi-modal fusion process. Secondly, the multimodal fusion task rarely considers the application of attention mechanism, resulting in a weak representation of the fused feature. In order to better use the multimodal data and reduce the losses caused by the fusion of different modalities, we proposed a TRMSF (Transformer and Multi-scale fusion) network for land-cover classification based on HSI (hyperspectral images) and LiDAR (Light Detection and Ranging) images joint classification. The network enhances multimodal information fusion ability by the method of attention mechanism from Transformer and enhancement using multi-scale information to fuse features from different modal structures. The network consists of three parts: multi-scale attention enhancement module (MSAE), multimodality fusion module (MMF) and multi-output module (MOM). MSAE enhances the ability of feature representation from extracting different multi-scale features of HSI, which are used to fuse with LiDAR feature, respectively. MMF integrates the data of different modalities through attention mechanism, thereby reducing the loss caused by the data fusion of different modal structures. MOM optimizes the network by controlling different outputs and enhances the stability of the results. The experimental results show that the proposed network is effective in multimodality joint classification.

Keywords: transformer; hyperspectral image; LiDAR; cross-modal data fusion

check for updates

Citation: Zhang, M.; Gao, F.; Zhang, T.; Gan, Y.; Dong, J.; Yu, H. Attention Fusion of Transformer-Based and Scale-Based Method for Hyperspectral and LiDAR Joint Classification. *Remote Sens.* **2023**, *15*, 650. https://doi.org/10.3390/rs15030650

Academic Editors: Gemine Vivone, Liqiang Zhang, Hairong Qi and Ying Qu

Received: 12 December 2022
Revised: 18 January 2023
Accepted: 19 January 2023
Published: 21 January 2023

1. Introduction

1.1. Multisource Remote Sensing Classification

The mission of fusing HSI and LiDAR images for land-cover classification has received a lot of attention in recent years. HSI consists of many spectral channels, which can reflect the spectral details of ground. LiDAR technology is a ranging technology. A LiDAR image taken at high altitude can reflect the distance between the ground and the shooting point, from which we can obtain the height information of the ground. Experimental studies have demonstrated that the combination of HSI and LiDAR images can significantly improve the accuracy of land-cover classification. In recent years, the research tendency of fusing HSI and LiDAR images often treats HSI and LiDAR images as similar positions to build a network, thereby judging the feature of those two equally.

The fusion of HSI and LiDAR images has been widely studied in recent years. H. Li et al. [1] proposed a three-way network to restrict hyperspectral features, LiDAR features, and fusion features. Each single way was optimized by an independent loss function. R. Hang et al. [2] proposed a dual-channel feature extraction network to extract hyperspectral features and LiDAR features. After getting these two modal features, they fuse the two and create a third way to obtain fusion features. The fusion strategies in this network are feature-level fusion using shared weights and decision-level fusion, which use ratios to adjust three outputs and finally obtain a good result. M. Zhang et al. [3] proposed a two-way autoencoder to rebuild HSI and LiDAR images, and adopted features generated by this two-way autoencoder to predict the land-cover classes. This method of extracting core features using autoencoders has achieved good results on the datasets. M. Zhang et al. also proposed a multi-scale parallel convolution in Ref. [4] called PToP CNN to extract the fusion features of HSI and LiDAR images. It is worth noting that its fusion method implemented by using PToP CNN to predict HSI to LiDAR images combines features from different layers in the test phase to get the final result. These two papers assumed that there is a common pattern between HSI and LiDAR images, and then used different strategies (two-way autoencoder and multi-scale convolution) to approach this pattern. Finally, they made the test data pass through the model carrier to generate the feature to be used in prediction.

In addition, HSI has rich spectral information, and the data extraction of HSI has gradually attracted the attention of scholars in recent years. In recent years, the outstanding work in unsupervised, semi-supervision, and self-supervision has included Refs. [5–7]. In addition, Ref. [8] also uses aggregate attention to solve the universal problem of inadequate fusion for global information and contextual information. Ref. [9] uses an Adaptive Filters method to combine information from different graphs network. Ref. [10] builds different branches to extract rich image information, and fully integrates them during the next process, and finally achieves good results.

1.2. Transformer

Transformer has been deeply studied in the community in recent years and is considered to be the most likely strategy to replace CNN's position in computer vision. Its main strategy is to use the attention mechanism to replace the adjacent region-based calculation method represented by the traditional CNN. It calculates the attention between the embeddings, so that the focus of the model is on the decisive element in the image, and uses those decisive element to extract the main features. Through this strategy, one can simulate human intuition to focus on the core characteristics and ignore the unrelated details and make the model pay more attention to the detailed information related to the task goals and curb other information. In recent years, the potential of the Transformer in multimodal fusion has been gradually realized. This is because any data of different modalities are main and secondary to the task goals; when one combines attention mechanism with the important information used for a task goal of different modalities, it can significantly reduce the interference of redundant information.

In recent years, there have been numerous papers related to Transformer. A.Vaswani et al. [11] for the first time proposed the Transformer structure in the field of natural language processing; it uses a form of embedding, which can be obtained through words, to calculate the attention of every embedding and every other embeddings. The Transformer therefore quantifies the connections between words. After the proposal of the Transformer, it soon gained the attention of relevant scholars in the field of computer vision. A. Dosovitskiy et al. [12] applied Transformer to the computer vision field first in the method of embedding image patches, which can be obtained by splitting an original image instead of words to calculate the attention between every two patches in the image. This method has an accuracy that is comparable to the convolutional neural network. Subsequently, related works for Transformer have sprung up in the field of computer vision. Because vision tasks needs to consider different scales, which is different from the natural language processing tasks, Z.Liu et al. [13] proposed Swin Transformer according to the CNN processing mode of images. It mainly uses the Shifted Windows to divide the image to generate patches, and finally they got a good result in experiments.

The success of the Transformer in multiple directions such as natural language processing and computer vision has attracted attention in the field of multimodality [14–18]. Among these attempts, there are works that simply use the Transformer structure to replace other structures, such as VisualBERT [16], Unicoder-VL [19], VL-BERT [20], and Uniter [21], There are also works that combine the Transformer structure with the structure of its own field, such as CLIP [22], Pixel-BERT [23], etc. There are also some multimodal works on remote sensing imagery, such as Refs. [14,17,18]. At present, the main scientific research direction is to use Transformer to combine with their respective advantageous models in different fields, such as recurrent neural networks, convolutional neural networks, etc. This is mainly due to the fact that Transformer, as a structure that only uses attention for feature extraction, lacks the prior knowledge (or inductive bias) of each domain itself [12]. Feature extraction using domain-specific feature extractors with prior knowledge and then using other means for information fusion is considered simple and effective [24,25].

Different modalities in multimodal mission have different composition structures. It is difficult to find a feature extraction and fusion method that is suitable for all modalities from these composition structures. The transformer extracts features from the form of attention, which is only related to the content of the data and has nothing to do with the form. This provides a universal feature extraction method for different modal data. Many scholars have published related works, such as H.wang et al. [26], who proposed a multi-channel attention calculation method of dual-path, using self-attention and cross-attention fusion to enhance the stability of the network.

1.3. Multi-Scale Method

The multi-scale method is an enhancement method of fusion with different scale feature information to simulate the visual phenomenon. Research results in recent years indicate that multi-scale strategies are effective in deep learning. In general, multi-scale methods are often used to build different scale feature maps or image pyramids using the difference in the pixel receptive field sizes, then fusing those different scale feature maps, so that the models can obtain the characteristic information about different sizes or distances. Similar to multimodality tasks, the multi-scale fusion strategies can also be divided into early fusion and late fusion. Early fusion is used to fuse multi-scale features before the final classification operation of the network. The advantage of this method is that the diversity of characteristics can be completely considered before decision-making. Later fusion is used to integrate features after the final classification operation. The multi-scale features obtained will inevitably have redundant information, which weakens the representation ability of the features, and later fusion can reduce the loss of this redundant data to the result. The disadvantage is that it is difficult to fully consider the features of different scales during classification, which results in an inaccurate outcome in the single decision-making process.

The multi-scale method is a method used to coordinate the global and local information of the image. It avoids the inadequate extraction of high and low frequency information used in traditional methods. Ren et al. [27] proposed a method based on different convolution kernels to extract different scale features, which controls the receptive field size of pixels in a feature map by using those convolution kernels of different sizes . Gong et al. [28] used one-dimensional, two-dimensional, and three-dimensional MS-CNN to extract multi-scale spectral features and spatial features to improve the representation ability of the feature maps. They eventually integrated the multi-scale features obtained above. More and more research results have proved that multi-scale methods have become one of the most important means to enhance the ability of feature representation and improve the robustness of models in practice.

1.4. Multi-Output Method

The multi-output structure converts the single-way network into a multi-way network, and learns different features in different layers by constraining multiple outputs. Multi-output can be implemented using different loss functions, and a single loss function can be used as well. The multi-output structure of the network is conducive to being optimized in different layers, which can partially avoid the optimization loss caused by the gradient backward. It can also design different networks to build different angles of features similar to the twin network to enhance the robustness of the network.

1.5. Contribution

The main contributions of this paper are as follows:

- TRMSF network is proposed for multimodal fusion classification task;
- In order to solve the problem of inadequate extraction of multi-scale information from multi-source remote sensing data, we build multi-scale features and fuse them into a feature in the process of multimodal fusion. A multi-scale attention enhancement module (MSAE) is proposed for feature fusion between different scales and different modalities. This module enhances the representation of multi-scale semantic information;
- In order to solve the problem of inadequate fusion of multimodal features, we introduced the attention mechanism to refine the multimodal features for fusion, and use those features to reduce the contradiction and redundancy from different modalities. We propose a module named fusion transformer (FUTR) for multimodal fusion using cross attention and experiments to prove that this module can significantly enhance the representation ability of fusion features;
- Aiming at solving incomplete feature extraction problem caused by a single output, this paper designs a multi-output module, and constructs a multi-level loss function to avoid the optimization loss caused by a backwards gradient.

2. Materials and Methods

The network proposed in this article contains three modules: multi-scale attention enhancement module (MSAE), multimodal fusion module (MFM), and multi-output modules (MOM) or loss strategy. As shown in Figure 1. MSAE is used to construct multi-scale feature maps and uses the attention mechanism to extract the main features of the image; MFM module uses the advantage of Transformer in multimodal fusion to calculate the cross-attention between two heterogeneous data and fuse them. MOM uses multi-level loss functions to adjust the emphasis on different modules and to avoid optimized losses caused by gradient backward transmission.

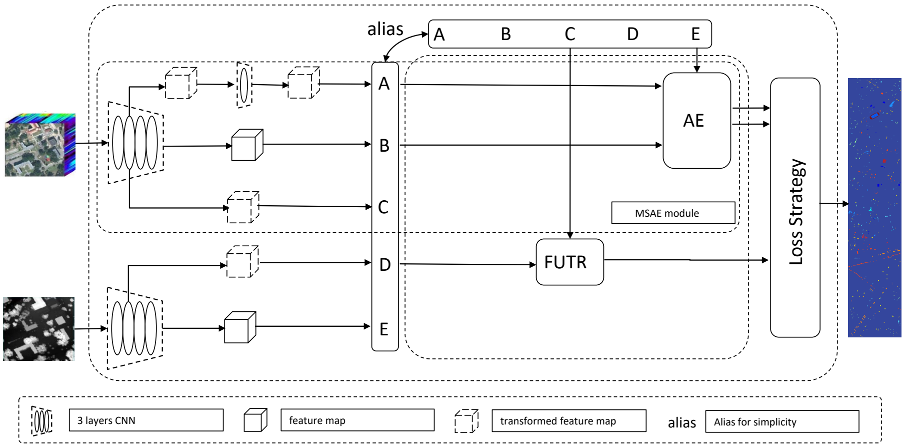


Figure 1. Structure of the network. In order to avoid cross-connection, we use the matrix form to represent the data flow. It is mainly composed of three parts: MSAE, FUTR, and loss strategy (MOM). The feature map extraction part is appropriately simplified in the figure, and the number of ellipses represents the number of convolutions. For ease of presentation, the five inputs of A-E are set with five more identical aliases. In addition, before extracting features from HSI, PCA technology is used to reduce the channel dimension of HSI. For brevity, the original input size in the above figure has been reduced, and the final output represents the true size.

2.1. MSAE

Image features of different scales have different effects on the results when carrying out a joint classification of HSI and LiDAR images. Some categories need to pay attention to the detail information of the image block during classification, and some need to pay more attention to the global information. For example, the type of grassland needs to be paid attention to for more comprehensive information of the image block, while roads and railways need to be paid attention to for more detailed information. However, this type of information, which needs to be considered, is often ignored in fusion. Secondly, current networks often perform approximately equal status in the used methods for the two modalities in the algorithm, such as in Refs. [1,2]. This partly causes an insufficient fusion of the scale information in the images. Therefore, this article uses shallow level and deep level features at the same time, and puts forward MSAE.

MSAE consists of two part: multi-scale feature construction(MSFC) and attention enhancement (AE).

Multi-scale features are obtained by multi-layer convolution extraction for deep features and image transformation (cutting, resizing, etc.) from the second layer for shallow features. AE is composed of two parallel channel attention extractors. The whole process is described in detail as follows.

First of all, because the HSI consists of many spectral channels, it requires a lot of calculation resources to deal with this spectral information. Therefore, we first use the PCA strategy to compress the spectral channels to 30, then we use HybridSN [29] as the backbone to extract the hyperspectral feature from input. Since the land-cover classification needs to predict each pixel of the original image, it is necessary to cut the neighborhood of the pixel to be predicted from the original image. Suppose that x is the processed HSI feature, and we obtain deep features x_{md} through feature extractor(md means multi-scale deep feature).

$$x_{md} = HybridSN(PCA(x)) \tag{1}$$

For LiDAR images, we also use CNN for feature extraction. Suppose that the processed LiDAR image is y, and we have

$$y_f = 2DCNN^3(y) \tag{2}$$

where the power exponent represents the number of convolutional layers and 2DCNN stands for two-dimensional convolution operation, i.e.,

$$Y_{i,j,k} = \sum_{c \in C} \sum_{\Delta_i, \Delta_j \in (-w,w)} X_{i+\Delta_i, j+\Delta_j, c,k} K_{o_i+\Delta_i, o_j+\Delta_j, c,k} \tag{3}$$

Here, i, j are the pixel indexes of the image or feature map, X represents the input, w represents the width or height size of the convolution kernel K, and k represents the index of the convolution kernel, C is the set of input channels. In a convolutional neural network, the parameters of the convolution kernel are self-learned. For brevity, the formula omits edge processing and bias terms.

The MSFC needs to consider different scale features for enhancement of network. There are many ways to build multi-scale features, including image transformations and design convolution kernels of different sizes, or feature fusion at different levels, etc. It is found through experiments that the improvement effect of three methods is almost the same. Considering the efficiency of calculation, this article uses image transformations and multi-layered convolutions to control the receptive size of feature map pixels and therefore constructs feature maps of different scales.

To achieve the operations above, we keep the shallow level features x_s obtained by the second convolution layer in the stage that uses the feature extractor of the HybridSN network to do the feature extraction, and then we transform it to get multi-scale feature maps for the next extraction. Finally we get the extracted shallow level features x_{ms} (ms means multi-scale shallow feature)

$$x_{ms} = 2DCNN(\mathscr{F}_t(x_s)) \tag{4}$$

where $\mathscr{F}_t$ represents transformation.

After we get the feature maps of different scales, we fuse them with LiDAR features by the way of attention mechanism. We calculate attention by the method of considering common attention. Specifically, we get a common attention by concatenation operation after applying global pooling operation, respectively, and then calculate two attention vectors for original feature maps, through linear mapping and apply those two attentions vectors to original feature maps channel-wise (see Figure 2). Specifically, we have

$$(x_{f1}, y_{f1}), (x_{f2}, y_{f2}) = AE(x_{ms}, y_f), AE(x_{md}, y_f) \tag{5}$$

$$f_{ems}, f_{emd} = [x_{f1}, y_{f1}], [x_{f2}, y_{f2}] \tag{6}$$

where [a,b] means concat a and b in channel dimension and AE means attention enhancement strategy, which in detail uses global pooling operation to convert the feature map to the attention vector, and then generate two attention vectors based on it. We can find its definition in formula (9). y_f, x_{ms}, x_{md} come from Formulas (1), (2) and (4). Two values in () means we split the output into two values. Finally, those two attention vectors are applied to feature maps channels of different modalities channel-wise. Assuming the inputs are feature maps u, v. the AE operation can be described as

$$F_{uni} = [\mathscr{F}_{GP}(u), \mathscr{F}_{GP}(v)] \tag{7}$$

$$w_u, w_v = W_u F_{uni}, W_v F_{uni} \tag{8}$$

$$AE(u, v) = w_u * u, w_v * v \tag{9}$$

where $\mathscr{F}_{GP}$ represents the global pooling operation and * represents multiplication channelwise (assign weights to different channels). W_u, W_v are the two linear layers. The main purpose of the AE operation is to combine data from different modalities to derive attention representation, therefore avoiding the insufficient extraction of multi-modal features due to the domination of single modality.

Next, apply pooling operation and output the result,

$$output_{ms}, output_{md} = AMCP(f_{ems}), AMCP(f_{emd}) \tag{10}$$

where AMCP means AMCPOOL operation (average pooling and max pooling for concat operation), which can enhance the stability of the pooling operations. In detail, we use both average pooling and maximum pooling and concat them. Suppose that the input is m, we have

$$AMCPOOL(m) = [\mathscr{F}_{AP}(m), \mathscr{F}_{MP}(m))] \tag{11}$$

where $\mathscr{F}_{AP}$ represents average pooling while $\mathscr{F}_{MP}$ represents maximum pooling.

Finally, we get $output_{ems}, output_{emd}$ in this section.

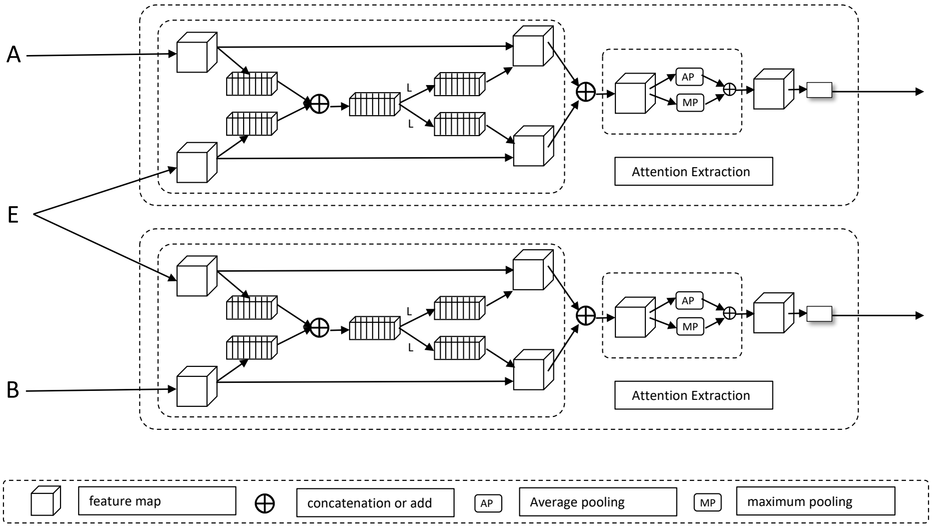


Figure 2. Structure of the AE module. L represents the dense layer. First, the multimodal unified representation is used to obtain the attentions and then applies it to the respective original features, and then the improved pooling operation is used to obtain the output. LiDAR data are fused with hyperspectral data at different scales to obtain more robust features.

2.2. MFM

Ref. [12] shows that CNN has a strong inductive bias in image processing, which allows the CNN to converge quickly during the training stage. However, this inductive bias is no better than what the Transformer learns by itself when pre-trained on a large amount of data. The results of the literature [30] show that using feature extractors of different modalities with inductive biases in multi-modal fusion tasks can significantly improve the feature extraction capability and extraction efficiency. Fusion of these extracted features using the Transformer can significantly improve the model performance.

The multi-modal fusion module is different from the multi-scale module in the fusion strategy, as it mainly uses cross attention for multimodal attention interaction.

First, we obtain features from the hyperspectral and LiDAR images for fusion.

$$x_{ca}, y_{ca} = 3DCNN^2(PCA(x)), 2DCNN^2(y) \tag{12}$$

Similar to 2DCNN, 3DCNN is

$$Y_{i,j,l,k} = \sum_{c\in C} \sum_{\substack{\Delta_i \in (-w,w) \\ \Delta_j \in (-w,w)'}} X_{i+\Delta_i, j+\Delta_j, l+\Delta_l, c, k} K_{o_i+\Delta_i, o_j+\Delta_j, l+\Delta_l, c, k} \tag{13}$$

The only difference between 3DCNN and 2DCNN is that the kernels of 3DCNN are three dimensional.

Next, the two outputs in Formula (12) are fused using the FUTR operation (fusion transformer operation) and then passed through the AMPPOOL operation (average pooling and max pooling for plus operation) to obtain the output (see Figure 3).

$$output_{futr} = W_f(AMPP(FUTR(x_{ca}, y_{ca}))) \tag{14}$$

where W_f represent the linear layer. The only difference between AMPP and AMPC is that we used plus operation instead of concat in the AMPC operation.

The FUTR operation is a method of multi-modal fusion using the attention calculation module of the transformer, and we can find its definition from Formula (18). Its main structure is shown in Figure 3. Assuming that the two inputs are i1 and i2, we first use the dense layer W_1, W_2 to convert them to q, k, v (query, key, value) vectors,

$$(q1, k1, v1), (q2, k2, v2) = W_1(i1), W_2(i2) \tag{15}$$

In the two sets of features, different groups of features are selected to form q, k, v, and are then fed into the encoder of the transformer for attention operation and the result is obtained through residual connection.

$$Attention(q, k, v) = softmax(\frac{qk^T}{\sqrt{d}})v \tag{16}$$

$$A = Attention(q, k, v) + i1 + i2 \tag{17}$$

$$output_{futr} = FUTR(i1, i2) = W_{ca1}W_{ca2}A + A \tag{18}$$

W_{ca1}, W_{ca2} represent two linear layers. In Formula (16) [11], $\sqrt{d}$ means the regularization term. Softmax is the softmax activation function.

There are two ways to implement the FUTR module. One is a single channel as shown in the upper side of Figure 3. This method selects the appropriate q, k, v from the first and second groups of q, k, v and then performs the attention calculation. The advantages of this strategy are higher computational efficiency and less redundancy in attention extraction. The second is to use a dual-path strategy. As shown in the lower part of the Figure 3, two attention groups are used to construct the attention calculation strategy. The advantage of this method is that it can calculate self-attention and fusion attention at the same time, and the features are fully considered, and it can even simultaneously calculate H2L (hyperspectral to LiDAR), L2H (LiDAR to hyperspectral), H2H (hyperspectral itself), and L2L (LiDAR itself)—four kinds of attention—and perform feature extraction for multiple attentions. Its disadvantage is that it is slow. Considering the problem of efficiency, we will use the former method to conduct the experiments.

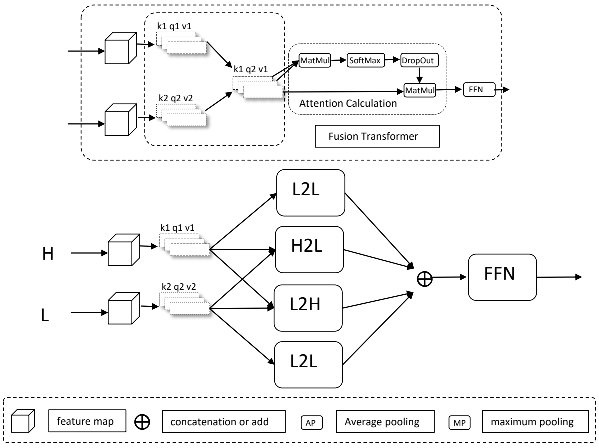


Figure 3. Structure of the FUTR module.

2.3. Multi-Output Modules

In the multi-scale attention enhancement module and multi-modal fusion module, we get three outputs, namely $output_{ms1}$, $output_{ms2}$ and $output_{futr}$. those three outputs are obtained in different ways. $output_{ms1}$ and $output_{ms2}$ represent features of different scales, while $output_{futr}$ represents the fusion features. In order to optimize different features separately and exclude the influence of features in different branches, we use the late fusion strategy [1,2]. Calculating the loss [1], we use the cross entropy of the three outputs respectively, and get $Loss_{ms1}, Loss_{ms2}, Loss_{futr}$. There are three ways to select the loss that requires the backward gradient. One is to calculate the common loss in the form of weighting, the other is to return only according to the smallest loss, and the third is to return according to the largest loss. After the experiments, the loss strategy of the first experiment is relatively stable, so this paper adopts the experimental results obtained this way.

$$Loss = \lambda_1 * Loss_{ms1} + \lambda_2 * Loss_{ms2} + \lambda_3 * Loss_{futr} \tag{19}$$

Here $*$ stands for ordinary multiplication and $\lambda_1, \lambda_2, \lambda_3$ are the parameters to be chosen or learned.

3. Experimental Results and Analysis

We use the Houston dataset and the Trento dataset to conduct the experiments. First, we conduct comparative experiments with several well-known models, and then construct several networks for the ablation experiments based on the two components proposed in this paper. Then, a visual experiment was carried out to analyze the difference between the model proposed in this paper and other models, and finally we use the Houston dataset to analyze the impact of three loss functions on the results. In order to ensure credibility, the experimental results are partly taken from other papers [31–38], except for those related to the model proposed in this paper. In the network parameter settings, we use a batch size of 128 and the learning rate is set to 0.0001.

3.1. Dataset Description

3.1.1. Houston Dataset

We use the Houston 2013 dataset for the experiments in this section, The Houston dataset contains image data of the University of Houston campus and the surrounding areas taken at high altitude, including hyperspectral data and LiDAR data, Its spectrum covers a range from 380 nm to 1050 nm, with 144 spectral channels. The image resolution is 2.5 m, and it consists of 16 types of objects in total.

3.1.2. Trento Dataset

The Trento dataset is divided into six categories, namely apple trees, buildings, ground, trees, vineyards, and roads. It has a size of 166 × 600 and was taken in southern Trento, Italy. The LiDAR data was acquired using an Optech ALTM 3100EA sensor, while the hyperspectral data was acquired by an AISA Eagle sensor with 63 spectral channels and a spatial resolution of 1 m.

3.2. Evaluation Indicators

In order to accurately evaluate the gap between the models, we selected three indicators to evaluate the models, namely the Overall Accuracy (OA), the Average Accuracy (AA), and Kappa coefficient.

3.2.1. Overall Accuracy

The Overall Accuracy is the proportion of samples that are accurately classified in all samples. Assuming that the total number of samples is N, and the number of samples accurately classified by the model is n, the overall accuracy rate is:

$$OA = \frac{n}{N} \tag{20}$$

The overall accuracy rate is a direct measure of the classification accuracy of the model, which can intuitively reflect the classification performance of the model.

3.2.2. Average Accuracy

OA has certain defects when the number of samples of different types is quite different. When the number of samples of a certain type is much smaller than the number of other samples, even if the samples of this type are not considered at all, a high OA can be obtained, which is incorrect in practical applications. So the average accuracy rate came into being, which can model the accuracy of each type of sample. Assuming that C is a set of sample categories, the total number of samples in the $i-th$ category is N_i , and the number of accurately classified samples is n_i , then the AA indicator is:

$$AA = \frac{1}{|C|} \sum_{i \in C} \frac{n_i}{N_i} \tag{21}$$

3.2.3. Kappa Coefficient

The Kappa coefficient is used for the consistency test, which is proposed to evaluate the consistency between the real sample category and the predicted sample category. The higher the Kappa coefficient, the closer the two results are. Its value is $[-1, 1]$, and the larger the value, the more consistent the predicted result is closer to the actual result. Assuming that C is a set of sample categories, the number of real samples of the i-th class is a_i, the number of predicted samples of the i-th class is b_i, and the total number of samples is N. We first calculate

$$p_e = \frac{\sum_{i \in C} a_i \times b_i}{n^2}, p_0 = OA \tag{22}$$

Then the Kappa coefficient is

$$Kappa = \frac{p_0 - p_e}{1 - p_e} \tag{23}$$

3.3. Comparative Experiment

3.3.1. Houston Dataset

The experimental result on the Houston dataset is shown in Table 1. We selected several well-known models in the industry for experimental comparison [1–4]. As can be seen from the table, our model performs significantly better than other models involved in the Houston dataset.

Table 1. Comparative experiment of the Houston dataset.

Class	SVM	CNN-PPF	CXC	TBC	CRNN	CC	CNN-MRF	EndNet	IP-CNN	PToP CNN	S2ENet	Ours
Healthy grass	82.43	83.57	84.89	83.10	83.00	**98.51**	85.77	78.54	85.77	85.77	82.72	98.29
Stressed grass	82.05	98.21	87.40	84.10	79.41	97.83	86.28	96.33	87.34	87.08	**100.00**	89.77
Synthetic grass	99.80	98.42	99.86	**100.00**	99.80	70.60	99.00	**100.00**	**100.00**	99.57	99.60	**100.00**
Trees	92.80	97.73	93.49	93.09	90.15	**99.06**	92.85	88.26	94.26	94.13	95.74	94.08
Soil	98.48	96.50	**100.00**	**100.00**	99.71	**100.00**	**100.00**	**100.00**	98.42	**100.00**	99.81	98.05
Water	95.10	97.20	98.77	99.30	83.21	41.11	98.15	**100.00**	99.91	99.38	97.20	99.14
Residential	75.47	85.82	82.81	92.82	88.06	83.14	91.64	83.02	94.59	87.38	91.23	**95.78**
Commercial	46.91	56.51	78.78	82.34	88.61	**98.39**	80.79	79.96	91.81	97.35	91.55	94.45
Road	77.53	71.20	82.51	84.70	66.01	94.81	91.37	93.30	89.35	90.81	**95.94**	93.89
Highway	60.04	57.12	59.41	65.44	52.22	**92.98**	73.35	92.28	72.43	72.21	84.75	87.18
Railway	81.02	80.55	83.24	88.24	81.97	90.88	98.87	85.86	96.57	**100.00**	94.31	94.84
Parking Lot 1	85.49	62.82	92.13	89.53	69.83	91.02	89.38	**99.81**	95.60	98.13	97.79	95.73
Parking Lot 2	75.09	63.86	94.88	92.28	79.64	97.09	92.75	83.16	94.37	92.11	89.47	**98.55**
Tennis Court	**100.00**	**100.00**	99.77	96.76	**100.00**	**100.00**	**100.00**	**100.00**	99.86	99.30	**100.00**	99.92
Running Track	98.31	98.10	98.79	99.79	**100.00**	97.85	**100.00**	**100.00**	99.99	**100.00**	**100.00**	99.54
OA(%)	80.49	83.33	86.90	87.98	88.55	90.43	90.61	90.71	92.06	92.48	93.99	**94.62**
AA(%)	83.37	83.21	89.11	90.11	90.3	90.22	92.01	92.03	93.35	93.55	94.67	**95.95**
KAPPA(%)	78.98	81.88	85.89	86.98	87.56	89.68	89.87	89.92	91.42	91.87	93.48	**94.16**

All data in the table are percentage values, we omit the % symbol for simplicity. In the table, TBC means Two Branch CNN and CxC means Contextual CNN, CC means Coupled CNN. The meaning in bold is the highest value of all networks.

3.3.2. Trento Dataset

The data on the Trento dataset is shown in the Table 2. We choose the same model as the previous section for comparison. It can be seen from the table that our model obviously surpasses other models in OA and Kappa coefficient, but is slightly insufficient in AA. After experimental verification, this is due to the occlusion caused by some wires in the lower right part of the LiDAR data in Trento dataset. In other parts of the Trento dataset, our model clearly outperforms other models in classification accuracy (see the visualization section).

Table 2. Comparative experiment of the Trento dataset.

Class	SVM	CNN-PPF	CXC	TBC	CRNN	CC	CNN-MRF	EndNet	IP-CNN	PToP CNN	S2ENet	Ours
Apple trees	88.62	90.11	99.26	98.07	98.39	99.87	**99.95**	99.90	99.00	99.60	99.90	96.88
Buildings	94.04	83.34	86.81	95.21	90.46	83.84	89.97	99.03	**99.40**	93.90	98.88	96.38
Ground	93.53	71.13	97.91	93.32	99.79	87.09	98.33	85.83	99.10	**100.00**	86.36	88.09
Woods	98.90	99.04	97.31	99.93	96.96	99.98	**100.00**	**100.00**	99.92	99.27	**100.00**	99.89
Vineyard	88.96	99.37	99.82	98.78	**100.00**	99.61	**100.00**	99.31	99.66	**100.00**	99.21	99.84
Roads	91.75	89.73	84.63	89.98	81.63	**98.75**	97.86	90.83	90.21	97.28	91.32	96.69
OA(%)	92.77	94.76	96.11	97.92	97.30	97.69	98.40	98.52	98.58	98.34	98.53	**98.63**
AA(%)	92.63	88.97	94.29	96.19	94.54	94.86	97.04	95.81	**97.88**	97.53	95.94	96.30
KAPPA(%)	95.85	93.04	94.81	96.81	96.39	96.91	97.86	98.01	**98.17**	97.79	98.03	**98.17**

All data in the table are percentage values, we omit the % symbol for simplicity. In the table, TBC means Two Branch CNN and CxC means Contextual CNN, CC means Coupled CNN. The meaning in bold is the highest value of all networks.

3.4. Ablation Study

We performed ablation experiments on the two datasets, respectively. Base, Msnet, Trnet, and TRMSF were used for experimental comparison. Base represents a network that uses only HybridSN and 2DCNN for the feature extraction of HSI and LiDAR images, Msnet represents a network that uses only the MSAE module based on Base, Trnet represents a network that uses only the FUTR module based on Base, and the last TRMSF represents the MSAE and FUTR modules network used together.

3.4.1. Houston Dataset

The experimental results are shown in the Table 3. From the data in the table, the results of Msnet and Trnet are slightly higher than Base. Among them, the improvement of Msnet is the most obvious.

Table 3. Ablation study result on the Houston dataset

	Base	Msnet	Trnet	TRMSF
OA(%)	92.31	94.09	93.17	94.62
AA(%)	94.72	95.67	94.86	95.95
kappa(%)	91.65	93.58	92.58	94.16

3.4.2. Trento Dataset

The results of the Trento data are shown in Table 4, and a changing trend similar to that of the Houston dataset can be observed.

Table 4. Ablation study result on the Trento dataset

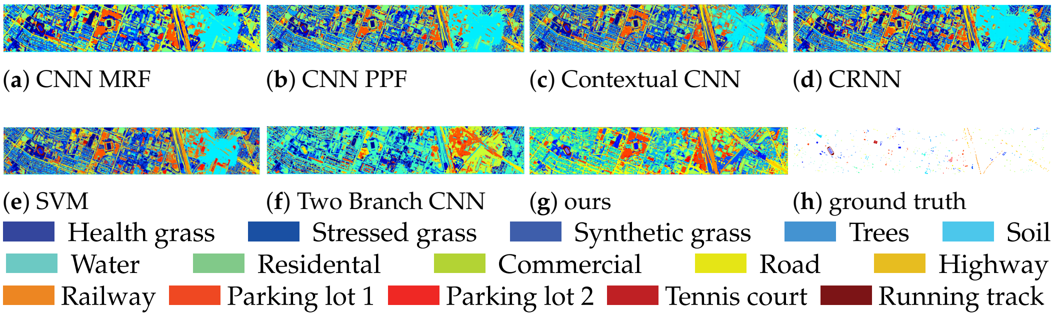
	Base	Msnet	Trnet	TRMSF
OA(%)	96.43	97.63	97.71	98.63
AA(%)	94.01	95.82	95.41	96.30
kappa(%)	95.25	96.83	97.41	98.17

3.5. Visualization Experiment

We still use the Houston and Trento datasets for the visualization experiment.

3.5.1. Houston Dataset

Figure 4 shows the visualization results of the several models that we selected and this model. It can be seen from the figure that the classification results in the right part of the image of our model are significantly more refined than other models. The classification of roads, bridges, and grasslands is particularly obvious.



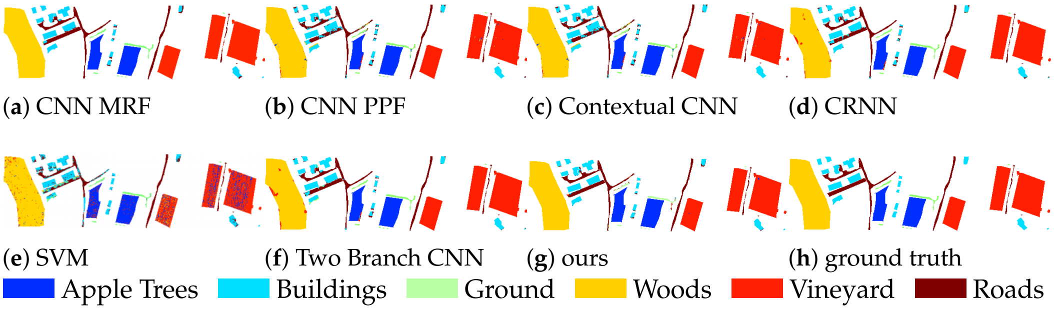
(**a**) CNN MRF (**b**) CNN PPF (**c**) Contextual CNN (**d**) CRNN

(**e**) SVM (**f**) Two Branch CNN (**g**) ours (**h**) ground truth

Figure 4. Visualization result of the Houston dataset in certain models.

3.5.2. Trento Dataset

Figure 5 shows the experimental results of several models. As can be seen from the figures, the classification accuracy of our model is significantly higher than other models, and it has fewer noise points and misclassification points.



(**a**) CNN MRF (**b**) CNN PPF (**c**) Contextual CNN (**d**) CRNN

(**e**) SVM (**f**) Two Branch CNN (**g**) ours (**h**) ground truth

Figure 5. Visualization result of the Trento dataset in certain models.

3.6. Convergence Experiment

In order to better show the training process of our model, we have measured the loss and OA generated during the model training process, and the training process is shown below. In order to better compare the convergence, we still use HybridSN and Msnet for comparison. The dataset used for the experiment is the Houston dataset. See Figure 6.

3.7. Loss Ratio Selection

The multi-output module has multiple outputs, and adjusting their ratios can have an impact on the results. This article uses the Houston dataset for the experiments to compare the results of different ratios. Throughout the experiments it was found that using high, medium, and low strategies for the three outputs can obtain better results. We split the ratio by the condition, so that the total is 1, and tested the network by 0.1, 0.3, and 0.6 shares. The test results are shown in the table below. The first and second positions in the index of three numbers represent the calculation results of shallow multi-scale features and deep multi-scale features, respectively, and the third position represents the calculation results of the fusion features. See Table 5.

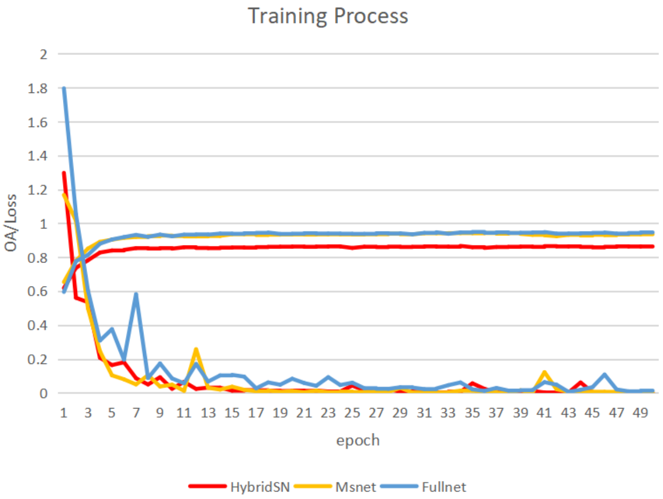


Figure 6. OA and loss in the training process.

Table 5. Experiment on ratio selection.

	Base	0.1,0.1,0.1	0.1,0.6,0.3	0.6,0.1,0.3	0.1,0.3,0.6	Aver
OA(%)	92.31	93.86	92.96	93.91	94.62	93.84
AA(%)	94.72	95.38	94.74	95.70	95.95	95.44
kappa(%)	91.65	93.33	92.36	93.39	94.16	93.31

It can be clearly seen that the optimization of fusion features has a certain effect on the improvement of the results.

4. Discussion and Conclusions

This paper proposes a multi-scale fusion technique and cross attention mechanism for the joint classification of hyperspectral data and LiDAR data. The model is divided into three parts: a multi-scale attention enhancement module, a multi-modal fusion module, and a multi-output module. The feature fusion module integrates the multimodal information together by fusing attention. The multi-scale attention enhancement module improves the performance by integrating multi-scale information to obtain richer semantic information. The multi-output module is used to realize multi-level loss and enhance the stability of the network. Overall, our model outperforms other models involved in the experiments and achieves satisfactory accuracy. By enhancing multi-scale data and fusing attentions from different modalities, our model has achieved 94.62% of OA on the Houston dataset, and 98.63% of OA is obtained on the Trento dataset. Through the ablation experiments, we proved the effectiveness of the MSAE module and the MFM module, and we also carried out experiments for ratio of the MOM module, The results prove that our model can achieve good results.

Compared to other work related to multisource remote sensing classification, our network considers rich multi-scale information and uses the attention mechanism to refine the fused features. This allows our network to capture more stable and richer features, which in turn improves the representative capacity of the fused features. In addition, the overall accuracy of the prediction results of our model is obviously higher than that of the other models involved. Through visualization experiments we can clearly see that our model can better capture information at different scales and can achieve finer classification results. Our model also gives finer classification results for large areas that are difficult to be classified by other models in the Houston dataset.

Next, our work is mainly devoted to eliminating the problem of misclassification caused by the occlusion and noise in the classification process. In the visualization results, the classification results have relatively regular misclassification problems in some local positions, and the reason for this problem is that there is a certain degree of occlusion in the LiDAR image. We found in our experiments that this problem can be partially avoided by enhancing the hyperspectral channel information or adjusting the usage of spectral channels.

Author Contributions: Conceptualization, M.Z. and F.G.; methodology, M.Z.; software, M.Z.; validation, M.Z.; investigation, M.Z; resources, F.G.; writing—original draft preparation, M.Z.; writing—review and editing, M.Z., F.G., J.D., Y.G., T.Z. and H.Y.; visualization, M.Z.; supervision, J.D. and F.G.; project administration, J.D. and F.G.; funding acquisition, J.D. All authors have read and agreed to the published version of the manuscript

Funding: This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0100602, and in part by the National Natural Science Foundation of China under Grant 41927805.

Data Availability Statement: The Houston dataset from https://hyperspectral.ee.uh.edu/?page_id=459.

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

References

1. Li, H.C.; Hu, W.S.; Li, W.; Li, J.; Du, Q.; Plaza, A. A 3 CLNN: Spatial spectral and multiscale attention ConvLSTM neural network for multisource remote sensing data classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *33*, 1–15. [CrossRef] [PubMed]
2. 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]
3. Zhang, M.; Li, W.; Tao, R.; Li, H.; Du, Q. Information fusion for classification of hyperspectral and LiDAR data using IP-CNN. *IEEE Trans. Geosci. Remote. Sens.* **2021**, *60*, 1–12. [CrossRef]
4. 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]
5. Ding, Y.; Zhang, Z.; Zhao, X.; Cai, W.; Yang, N.; Hu, H.; Cai, W. Unsupervised self-correlated learning smoothy enhanced locality preserving graph convolution embedding clustering for hyperspectral images. *IEEE Trans. Geosci. Remote. Sens.* **2022**, *60*, 1–16. [CrossRef]
6. Ding, Y.; Zhao, X.; Zhang, Z.; Cai, W.; Yang, N.; Zhan, Y. Semi-supervised locality preserving dense graph neural network with ARMA filters and context-aware learning for hyperspectral image classification. *IEEE Trans. Geosci. Remote. Sens.* **2021**, *60*, 1–12. [CrossRef]
7. Ding, Y.; Zhang, Z.; Zhao, X.; Cai, Y.; Li, S.; Deng, B.; Cai, W. Self-supervised locality preserving low-pass graph convolutional embedding for large-scale hyperspectral image clustering. *IEEE Trans. Geosci. Remote. Sens.* **2022**, *60*, 1–16. [CrossRef]
8. Ding, Y.; Zhao, X.; Zhang, Z.; Cai, W.; Yang, N. Graph sample and aggregate-attention network for hyperspectral image classification. *IEEE Geosci. Remote. Sens. Lett.* **2021**, *19*, 1–5. [CrossRef]
9. Ding, Y.; Zhang, Z.; Zhao, X.; Hong, D.; Li, W.; Cai, W.; Zhan, Y. AF2GNN: Graph convolution with adaptive filters and aggregator fusion for hyperspectral image classification. *Inf. Sci.* **2022**, *602*, 201–219. [CrossRef]
10. Yao, D.; Zhi-li, Z.; Xiao-feng, Z.; Wei, C.; Fang, H.; Yao-ming, C.; Cai, W.W. Deep hybrid: Multi-graph neural network collaboration for hyperspectral image classification. *Def. Technol.* **2022**, *in press*. [CrossRef]
11. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *2017*, 30.
12. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
13. Liu Z, Lin Y, Cao Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 10012–10022.
14. Xue, Z.; Tan, X.; Yu, X.; Liu, B.; Yu, A.; Zhang, P. Deep Hierarchical Vision Transformer for Hyperspectral and LiDAR Data Classification. *IEEE Trans. Image Process.* **2022**, *31*, 3095–3110. [CrossRef]
15. Li, L.H.; Yatskar, M.; Yin, D.; Hsieh, C.J.; Chang, K.W. Visualbert: A simple and performant baseline for vision and language. *arXiv* **2019**, arXiv:1908.03557.
16. Lu, J.; Batra, D.; Parikh, D.; Lee, S. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Adv. Neural Inf. Process. Syst.* **2019**, *2019*, 32.
17. Zhao, X.; Zhang, M.; Tao, R.; Li, W.; Liao, W.; Tian, L.; Philips, W. Fractional fourier image transformer for multimodal remote sensing data classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, 1–13. [CrossRef]
18. Yuxuan, H.; He, H.; Weng, L. Hyperspectral and LiDAR Data Land-Use Classification Using Parallel Transformers. In Proceedings of the IGARSS 2022–2022 IEEE International Geoscience and Remote Sensing Symposium, Kuala Lumpur, Malaysia, 17–22 July 2022.

19. Li, G.; Duan, N.; Fang, Y.; Gong, M.; Jiang, D. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, No. 07.
20. Su, W.; Zhu, X.; Cao, Y.; Li, B.; Lu, L.; Wei, F.; Dai, J. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv* **2019**, arXiv:1908.08530.
21. Chen, Y.C.; Li, L.; Yu, L.; El Kholy, A.; Ahmed, F.; Gan, Z.; Liu, J. Uniter: Universal Image-Text Representation Learning. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Cham, Switzerland, 2020.
22. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sutskever, I. Learning transferable visual models from natural language supervision. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual , 18–24 July 2021.
23. Huang, Z.; Zeng, Z.; Liu, B.; Fu, D.; Fu, J. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *arXiv* **2020**, arXiv:2004.00849.
24. Zhen, L.; Hu, P.; Wang, X.; Peng, D. Deep supervised cross-modal retrieval. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 10394–10403.
25. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Cham, Switzerland, 2020.
26. Wang, H.; Zhu, Y.; Adam, H.; Yuille, A.; Chen, L.C. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 5463–5474.
27. Ren, P.; Li, C.; Wang, G.; Xiao, Y.; Du, Q.; Liang, X.; Chang, X. Beyond Fixation: Dynamic Window Visual Transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022.
28. Gong, Z.; Zhong, P.; Yu, Y.; Hu, W.; Li, S. A CNN with multiscale convolution and diversified metric for hyperspectral image classification. *IEEE Trans. Geosci. Remote. Sens.* **2019**, *57*, 3599–3618 [CrossRef]
29. Roy, S.K.; Krishna, G.; Dubey, S.R.; Chaudhuri, B.B. HybridSN: Exploring 3-D–2-D CNN feature hierarchy for hyperspectral image classification. *IEEE Geosci. Remote. Sens. Lett.* **2019**, *17*, 277–281. [CrossRef]
30. Hu, R.; Amanpreet, S. Unit: Multimodal multitask learning with a unified transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021.
31. Cao, X.; Zhou, F.; Xu, L.; Meng, D.; Xu, Z.; Paisley, J. Hyperspectral image classification with Markov random fields and a convolutional neural network. *IEEE Trans. Image Process.* **2018**, *27*, 2354–2367. [CrossRef] [PubMed]
32. Li, W.; Wu, G.; Zhang, F.; Du, Q. Hyperspectral image classification using deep pixel-pair features. *IEEE Trans. Geosci. Remote. Sens.* **2016**, *55*, 844–853. [CrossRef]
33. Lee, H.; Kwon, H. Going deeper with contextual CNN for hyperspectral image classification. *IEEE Trans. Image Process.* **2017**, *26*, 4843–4855. [CrossRef] [PubMed]
34. Wu, H.; Prasad, S. Convolutional recurrent neural networks for hyperspectral data classification. *Remote. Sens.* **2017**, *9*, 298. [CrossRef]
35. 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]
36. 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]
37. Hong, D.; Gao, L.; Hang, R.; Zhang, B.; Chanussot, J. Deep encoder-decoder networks for classification of hyperspectral and LiDAR data. *IEEE Geosci. Remote. Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]
38. Fang, S.; Li, K.; Li, Z. S^{2}ENet: Spatial–spectral cross-modal enhancement network for classification of hyperspectral and LiDAR data. *IEEE Geosci. Remote. Sens. Lett.* **2022**, *19*, 1–5.

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