International Journal of Geo-Information

MDPI

Article

Satellite Image for Cloud and Snow Recognition Based on Lightweight Feature Map Attention Network

Chaoyun Yang [1], Yonghong Zhang [1,2,*], Min Xia [1,2], Haifeng Lin [3], Jia Liu [1] and Yang Li [1]

[1] Collaborative Innovation Center on Atmospheric Environment and Equipment Technology, Nanjing University of Information Science and Technology, Nanjing 210044, China; 202083250021@nuist.edu.cn (C.Y.); xiamin@nuist.edu.cn (M.X.); liujia@nuist.edu.cn (J.L.); liy163@chinatelecom.cn (Y.L.)
[2] Jiangsu Key Laboratory of Big Data Analysis Technology, Nanjing University of Information Science and Technology, Nanjing 210044, China
[3] College of Information Science and Technology, Nanjing Forestry University, Nanjing 210037, China; haifeng.lin@njfu.edu.cn
* Correspondence: zyh@nuist.edu.cn

Abstract: Cloud and snow recognition technology is of great significance in the field of meteorology, and is also widely used in remote sensing mapping, aerospace, and other fields. Based on the traditional method of manually labeling cloud-snow areas, a method of labeling cloud and snow areas using deep learning technology has been gradually developed to improve the accuracy and efficiency of recognition. In this paper, from the perspective of designing an efficient and lightweight network model, a cloud snow recognition model based on a lightweight feature map attention network (Lw-fmaNet) is proposed to ensure the performance and accuracy of the cloud snow recognition model. The model is improved based on the ResNet18 network with the premise of reducing the network parameters and improving the training efficiency. The main structure of the model includes a shallow feature extraction module, an intrinsic feature mapping module, and a lightweight adaptive attention mechanism. Overall, in the experiments conducted in this paper, the accuracy of the proposed cloud and snow recognition model reaches 95.02%, with a Kappa index of 93.34%. The proposed method achieves an average precision rate of 94.87%, an average recall rate of 94.79%, and an average F1-Score of 94.82% for four sample recognition classification tasks: no snow and no clouds, thin cloud, thick cloud, and snow cover. Meanwhile, our proposed network has only 5.617M parameters and takes only 2.276 s. Compared with multiple convolutional neural networks and lightweight networks commonly used for cloud and snow recognition, our proposed lightweight feature map attention network has a better performance when it comes to performing cloud and snow recognition tasks.

Keywords: cloud and snow recognition; convolutional neural network; lightweight feature map; attention network

Citation: Yang, C.; Zhang, Y.; Xia, M.; Lin, H.; Liu, J.; Li, Y. Satellite Image for Cloud and Snow Recognition Based on Lightweight Feature Map Attention Network. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 390. https://doi.org/10.3390/ijgi11070390

Academic Editors: Godwin Yeboah and Wolfgang Kainz

Received: 18 May 2022
Accepted: 7 July 2022
Published: 12 July 2022

1. Introduction

In recent years, with the deepening of meteorological remote sensing research, cloud and snow recognition technology has been widely used in meteorological research. However, in certain highland areas in China, the snow-covered areas in remote sensing images can interfere with the cloud identification and produce false detection, which reduces the detection accuracy. Additionally, the larger snowfall in the winter and spring seasons in China's highland areas will produce a snow accumulation phenomenon. At the same time, China's livestock-based Inner Mongolia plateau and plantation-based Yunnan-Guizhou plateau, etc., are very sensitive to the environmental changes of cloud snow and ice cold. Therefore, the study of cloud-snow identification is of great importance [1–3].

So far, traditional research on cloud image analysis or cloud and snow recognition has mainly been based on multi-spectral threshold analysis and artificially designed clustering methods using multiple functions. Zhang et al. (2019) [4] proposed a detection method using multi-featured high-resolution remote sensing images, including the color, texture, and shape of ground objects, and used methods such as minimum cross-entropy threshold and transform filter multi-scale image decomposition to increase the accuracy. Wang et al. (2011) [5] divided the MODIS data into two categories based on landmark spectral analysis and combined them with the K-means method, and proposed an optimized detection method combining K-means clustering and multispectral threshold, which can effectively detect a small area of cloud pixels and exclude the interference of the underlying surface. Sekrecka et al. (2019) [6] mainly examines and improves the panchromatic sharpened image for object detection, which is used to detect the resolution of different spaces, enhancing the target detection and the spectral quality of the fusion image. Candra et al. (2018) [7] proposed cloud color and edge features and a two-color space cloud detection method. After overlaying analysis and linear stretching of cloud area maps with different saturation and colors, a complete cloud area picture was formed. However, there are still some problems with traditional detection methods. For example, it is difficult to solve the problem of the inaccurate distinction between cloud and snow at the pixel level caused by the consistent color distribution of clouds and snow and the characteristics of similar local textures [8]. There are still no small challenges to the requirements of high-precision detection.

Because of the limitations of the above-mentioned traditional detection methods in terms of detection accuracy, many research scholars combined deep learning methods to propose a cloud image detection method based on deep learning, thereby improving the accuracy of cloud and snow recognition and enhancing practicability [9–12]. Xia et al. (2019) [13] proposed an improved multi-dimensional input deep residual network (M-ResNet), which can effectively extract the image features and spectral information of satellite images in the process of cloud and snow recognition. To apply the advantages of deep convolutional neural networks to cloud classification tasks, Ye et al. (2017) [14] proposed cloud classification with the help of deep convolution, and Fisher vector coding was applied to perform spatial feature aggregation and high-dimensional feature mapping of the original deep convolutional features to achieve better results. Li et al. (2018) [15] proposed a multi-scale convolutional feature fusion (MSCFF) cloud detection method based on deep convolutional neural networks. To extract multi-scale and advanced spatial features, a trainable filter bank was used to increase the feature map. It extracted multi-scale and high-level spatial features, up-sampled and connected feature maps of multiple scales, and then merged features of different scales for output to achieve the task of cloud detection. Bai et al. (2016) [16] sought to improve the accuracy of resolution satellite images. Machine learning and multi-feature fusion technology were used to compare the typical spectrum, texture, and other characteristic differences of clouds and backgrounds, which had good scalability. Li et al. (2019) [15] proposed a cloud detection method based on a deep learning method called multi-scale convolutional feature fusion (MSCFF) that can provide local and global context to fetch multi-scale and high-level spatial features. Additionally, a novel multi-scale feature fusion module was designed to fuse features from different scales to obtain the output. Yang et al. (2018) [17] achieved multi-level cloud detection using multiple convolutional neural networks and an adaptive simple linear iterative clustering algorithm in satellite image segmentation. At the same time, a new multi-convolutional neural network was designed to achieve high-precision detection targets in order to extract multi-scale features from each superpixel. Zhan et al. (2017) [8] used a fully convolutional neural network to classify pixel-level clouds and snow,solving the shortcomings of traditional methods that cannot accurately distinguish pixel-level clouds and snow from images. The deep model of cloud and snow detection, and the introduction of a multi-scale prediction strategy, then manually marked the cloud and snow data set for training. Xie et al. (2017) [18] segmented the image into high-quality superpixels. They

improved the simple linear iterative clustering (SLIC) method and designed a two-branch deep convolutional neural network to extract multiple superpixels from each superpixel. The scale featured and gave detection results to the prediction of superpixels in the image, which improved the accuracy and robustness of detection. Liu et al. (2020) [19] proposed a new depth model to learn heterogeneous depth features (HDFs) for multimodal ground remote sensing cloud classification. They fed the graph into a graph convolutional network (GCN) extractor to obtain GCN-based features that could capture the correlation between multimodal cloud samples using the graph convolutional layer and concatenate features based on heterogeneity under different networks to represent multimodal cloud samples with high accuracy.

Although the detection of the above-mentioned new method can accomplish the target task, it still has many shortcomings. The efficiency of cloud and snow detection still needs to be improved [20,21]. Therefore, it is necessary to design an efficient and lightweight network model, and this work proposes a lightweight feature mapping attention network, which is combined with the cloud and snow recognition model to reduce network parameters and improve training efficiency while ensuring the performance of the cloud and snow recognition model. An accurate and efficient cloud and snow detection model will be achieved.

2. Lightweight Feature Map Attention Network

2.1. Multi-Scale Fusion Attention Network

To solve the problems of misdetection, low classification accuracy, and weak generalization ability in the cloud and snow recognition task, a multi-scale fusion attention network (MfaNet) is proposed. Its structure is shown in Figure 1. It is composed of a densely connected network as the backbone network and combined with a multi-scale fusion module and an attention mechanism. This network can reduce the problem of false detection and missed detection in cloud snow recognition and improve the generalization recognition ability.

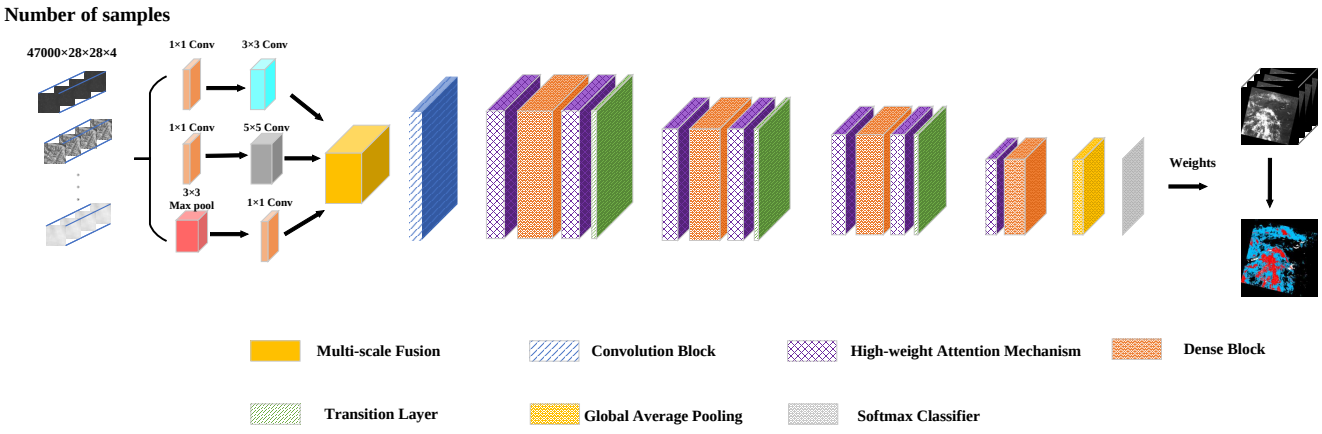

Figure 1. Multi-scale fusion attention network.

We propose using the multi-scale fusion module mainly to be able to pre-cluster the highly correlated cloud-snow features together while reducing some of the redundant cloud-snow feature information. In the multi-scale fusion module, three-branch operations are performed. The first branch consists of 1×1 and 3×3 convolution operations, the second branch consists of 1×1 and 5×5 convolution operations, and the third branch consists of a 3×3 max-pooling operation with a step size of 1 and a 1×1 convolution operation, and each convolution is followed by an activation function. After the branching operations, the output is stitched into a deeper feature map, resulting in a richer cloud snow feature map.

After generating the feature map via the multi-scale fusion module, in order to address the random and complex features of cloud snow features, a high-weight attention mechanism is added to recalibrate the features of the feature map by assigning high and low weights to the useful and useless information, respectively.

After that, we enter the densely connected network, which consists of four dense blocks and three transition layers. Unlike traditional convolutional neural networks, in which L layers correspond to L connections, in densely connected networks, the input of each layer is stacked on the output of all previous layers, so that $L(L+1)/2$ connections can be obtained. In order to get the same number of feature channels per layer, the four dense blocks have 2, 3, 3, and 2 layers, respectively, and the dense blocks are able to fully propagate the feature information by interconnecting all the layers and add 32 feature maps to each layer while keeping the feature map size constant. Additionally, the network uses a transition layer that is designed to reduce the dimensionality of the densely connected neural network. The transition layer contains a batch normalization layer, a 1×1 convolutional layer, and a 1×1 average pooling layer, which can reduce the feature map of the previous dense block to half of its original size.

Finally, the feature map is transformed into one-dimensional data using global average pooling and convolution operations, and the Softmax classifier outputs the probability that each cloud snow training sample belongs to each class.

However, this network utilizes a densely connected network as the backbone network, which consumes memory during training, and the training speed is slow. Because of the densely connected network, to ensure the propagation and reuse of features, a more radical connection method is adopted: the output of all the previous layers and the output of the current layer are spliced, and all are passed to the next layer. This brings huge memory consumption because every splicing operation needs to open up a new memory to store features. For an L-layer dense block, it consumes the memory equivalent to $L(L+1)/2$-layer convolutional neural network. This method not only slows down the training speed of the network but also requires higher computing power and hardware. In addition, the various modules in the multi-scale fusion attention network have not considered factors such as the number of parameters and the test time.

Therefore, this paper aims to design an efficient and lightweight cloud and snow recognition model based on ensuring the accuracy of the cloud and snow recognition model.

2.2. Network Structure

As the depth of the convolutional neural network becomes deeper and deeper, the network model usually requires tens of millions of parameters to participate in the calculation, which leads to problems such as a large amount of calculation, a complex network structure, and a slow test speed [22]. Therefore, reducing the number of model parameters and calculation loss has become an important research direction. Lightweight issues are generally considered from the two perspectives of network compression and network structure optimization. As far as network compression is concerned, the usual methods include channel pruning [23,24], model quantification [25], tensor decomposition [26], knowledge distillation [27], and other methods. The lightweight feature mapping attention network (Lw-fmaNet) in this paper is studied from the perspective of network structure optimization.

In existing studies, DenseNet is usually used as a backbone network because the stitching operation feature of this network can be exploited so that the feature input of each layer contains the output of the previous layer and the input of all layers before that layer, thus enhancing the propagation and replication of features to fully extract the feature information of the image. However, this network has a large amount of floating-point calculations and a large number of memory accesses, which is not suitable for the construction of a lightweight network model. At the same time, the radical dense connection method, which means that the output of all the previous layers and the output of the current layer are stitched together and are all passed to the next layer, will definitely

bring the redundancy of the cloud and snow characteristics. For the selection of the basic network of the cloud and snow recognition model in this paper, we focus on the most classic ResNet. Further, we have conducted comparative experiments on ResNet18, ResNet34, and ResNet50, where the number in the ResNet network represents the number of layers of the sum of convolutional and fully connected layers. The comparative data are shown in Table 1. The residual block in ResNet18 and ResNet34 is composed of two layers of 3×3 convolution, while the residual block of ResNet50 contains two layers of 1×1 convolution and one layer of 3×3 convolution. It should be emphasized here that all models in this paper are based on the Pytorch framework, a common neural network framework. Unlike the commonly used Keras framework, the Pytorch framework has better training accuracy and speed, which is more helpful for lightweight and efficient purposes.

Table 1. Comparison of basic network related parameters.

Basic Network	Accuracy (%)	Kappa Index (%)	Parameter (M)	Floating Point Calculation (G)	Testing Time (s)
ResNet18	89.85	86.42	11.182	0.035	1.354
ResNet34	88.74	84.94	21.290	0.072	3.165
ResNet50	90.12	86.78	23.519	0.081	2.705

From Table 1, we can see that although ResNet50 has the highest accuracy rate, the accuracy rate of ResNet18 is only 0.27% lower. Taking into account other factors such as the number of parameters and test time, we finally choose ResNet18 as the base network of the model in this paper. For the original ResNet18, it is not a lightweight model, and neither accuracy nor generalization can achieve satisfactory results when applied to cloud and snow recognition tasks. Therefore, we make a series of modifications to its basis, changing the 7×7 convolutional layer in the original network to a shallow feature extraction module, changing the two-layer 3×3 convolution in the residual module to a two-layer intrinsic feature mapping module, and adding a lightweight adaptive attention mechanism behind them. Through the above modifications, a lightweight feature map attention network is constructed.

Next, we introduce the network structure of the lightweight feature map attention network in detail. The network first uses the shallow feature extraction module to extract shallow features from the cloud and snow dataset. In this module, we use two layers of depth separable convolution and one layer of hybrid depth convolution and use the Leaky ReLU activation function after the three layers of convolution instead of the traditional ReLU activation. In the depth separable convolution of the first layer, the channel-by-channel convolution size is 3×3, and the point-by-point convolution size is $1 \times 1 \times 32$. In the depth separable convolution of the second layer, the channel-by-channel convolution size is 5×5, and the point-by-point convolution size is $1 \times 1 \times 64$. In the third layer of hybrid depth convolution, we divide the 64 channels into four groups. The first group has 32 channels, the second group has 16 channels, and the third and fourth groups have 8 channels. The sizes of the convolution kernels are: 3×3, 5×5, 7×7, 9×9. The main purpose of this module is to extract the cloud and snow features of multi-level and multiple receptive fields, so as to reduce the occurrence of missed detection. In the shallow feature extraction module, the core convolution is channel-by-channel convolution, which can greatly reduce the number of parameters. If used together with point-by-point convolution, it can fuse information between channels and compress or expand channel dimensions.

After the cloud and snow dataset completes the learning of the shallow feature extraction module, it enters the L2 to L5 layers of the lightweight feature mapping attention network for deep feature learning. In these layers of modules, we changed the original two-layer 3×3 convolution block in ResNet18 to a two-layer intrinsic feature mapping module and added a lightweight adaptive attention mechanism behind them. It should be noted that we retain the residual connection in ResNet, which aims to alleviate the phenomenon of vanishing gradient. In the internal feature mapping module, the first half of the original

feature maps are generated by conventional convolution, and the remaining half of the feature maps are generated by function mapping, then the two parts of the feature maps are spliced, which is the output of this layer. We can see that the number of channels and size of the output feature maps obtained by the intrinsic feature mapping module and the ordinary convolution operation are the same, but the parameter amount and computational complexity are reduced. The lightweight adaptive attention mechanism mainly focuses on the mutual influence between channels. Through one-dimensional convolution, the attention learning between channels is completed. This module uses almost negligible parameters to achieve the weight of the feature channel. Figure 2 shows the specific network structure of the L3 layer in the lightweight feature mapping attention network.

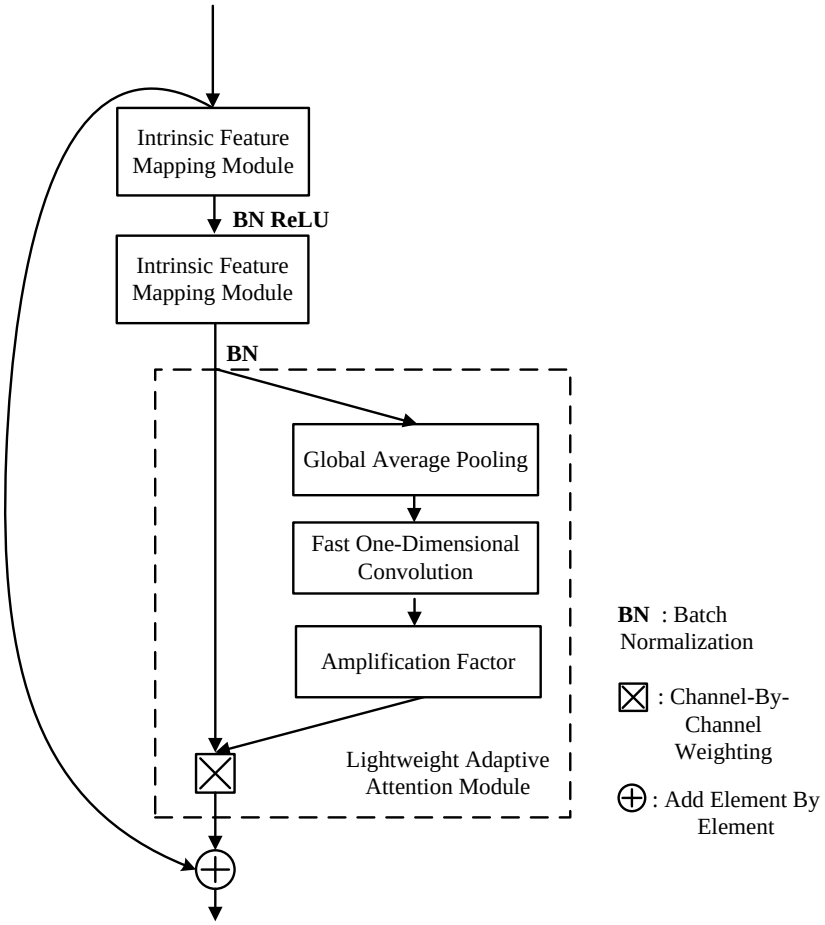

Figure 2. Detail map of L3 layer in lightweight feature attention network.

After completing the learning of the shallow and deep features, we use the global average pooling layer and the fully connected layer to map the feature space to the label category space, and finally use the Softmax classifier to output the probability of each cloud and snow training sample belonging to each category, thereby obtaining the classification result of the cloud and snow sample. Table 2 shows the comparison between the ResNet18 network structure and the modified lightweight feature map attention network structure. Compared with ResNet18, the advantage of our network is that we have much higher accuracy for the same number of layers, while the specific parameter comparison will be given in Section 3.

Table 2. Comparison of ResNet18 and lightweight feature mapping attention network structure.

	ResNet18	**Lightweight Feature Map Attention Network**
Level	Layer	Layer
L1	conv 7×7, 64, step2	Shallow feature extraction module
L2	3×3 max pool, step2	3×3 max pool, step2
	$\begin{bmatrix} conv3 \times 3, & 64 \\ conv3 \times 3, & 64 \end{bmatrix} \times 2$	$\begin{pmatrix} Intrinsic\ feature\ mapping\ module \\ Intrinsic\ feature\ mapping\ module \\ Lightweight\ adaptive\ attention\ mechanism \end{pmatrix} \times 2$
L3	$\begin{bmatrix} conv3 \times 3, & 128 \\ conv3 \times 3, & 128 \end{bmatrix} \times 2$	$\begin{pmatrix} Intrinsic\ feature\ mapping\ module \\ Intrinsic\ feature\ mapping\ module \\ Lightweight\ adaptive\ attention\ mechanism \end{pmatrix} \times 2$
L4	$\begin{bmatrix} conv3 \times 3, & 256 \\ conv3 \times 3, & 256 \end{bmatrix} \times 2$	$\begin{pmatrix} Intrinsic\ featuremappingmodule \\ Intrinsic\ featuremappingmodule \\ Lightweight\ adaptive\ attention\ mechanism \end{pmatrix} \times 2$
L5	$\begin{bmatrix} conv3 \times 3, & 512 \\ conv3 \times 3, & 512 \end{bmatrix} \times 2$	$\begin{pmatrix} Intrinsic\ feature\ mapping\ module \\ Intrinsic\ feature\ mapping\ module \\ Lightweight\ adaptive\ attention\ mechanism \end{pmatrix} \times 2$
L6	Global average pooling layer Fully connected layer Softmax classifier	Global average pooling layer Fully connected layer Softmax classifier

2.3. Shallow Feature Extraction Module

As we all know, when we use convolutional neural networks to learn features of samples, the shallow features usually learn the detailed information and location information of the image [28]. For the cloud and snow data set in this article, the shallow features mainly learn the texture, spectrum, space, and other information in the remote sensing image. The deep features usually contain more abstract features. In the cloud and snow recognition task, the judgment of the cloud and snow category is biased. How to conduct detailed and comprehensive learning of the shallow features such as texture, spectrum, and space in remote sensing images is very important for the task of cloud and snow recognition.

The purpose of the shallow feature extraction module is to extract detailed and rich shallow features, thereby reducing the missed detection of the cloud and snow recognition model and improving the recognition accuracy of the model. This module is a modification made on the L1 layer of ResNet18 in Table 2. We changed the original 7×7 convolution in the L1 layer of ResNet18 to a shallow feature extraction module that includes two layers of depth separable convolution and one layer of hybrid depth convolution.

Depth separable convolution is a common convolution operation in lightweight network models [29]. Compared with the conventional convolution operation, its parameter amount and operation cost are much lower. Depth separable convolution divides a complete convolution operation into two steps through channel-wise convolution and point-wise convolution. Figure 3 is a schematic diagram of the convolution process of depth separable convolution.

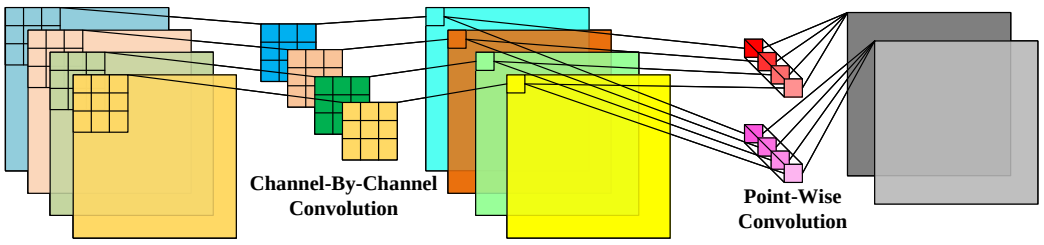

Figure 3. Schematic diagram of depth separable convolution.

In the depth separable convolution, we first perform feature learning on the input feature map without changing the number of feature channels through channel-by-channel convolution. Different from the conventional convolution operation, the channel-by-channel convolution splits all the input multi-dimensional feature maps into single-channel feature maps and then performs single-channel convolution on them separately, which means that a convolution kernel is only responsible for one feature channel.

After the feature map is convolved channel by channel, the number of channels of the newly generated feature map remains unchanged. There are two problems with this operation: first, the feature map after single-channel convolution can only learn the local semantic information of the channel; second, the channel-by-channel convolution performs independent convolution on each channel of the input layer. Operationally, semantic information in the same spatial location but on different feature channels cannot be correlated with each other, and the spatial information between them cannot be effectively used. Therefore, the depth separable convolution requires the second step; that is, the point-by-point convolution operation. Point-by-point convolution performs a convolution operation with a convolution kernel size of $1 \times 1 \times M$, where M is the number of channels in the previous layer. Through the $1 \times 1 \times M$ convolution kernel, the depth separable convolution completes the feature fusion of the output feature map in the channel-by-channel convolution. At the same time, this operation also changes the number of channels of the output feature. If there are n convolution kernels in the point-by-point convolution, then the output feature map has n channels.

Next, we introduce the second type of convolution in the shallow feature extraction module, the hybrid deep convolution [30]. The convolution aims to mix convolution kernels of different sizes into one convolution operation so that receptive fields of different sizes can be used to capture more shallow semantic features. The role of hybrid deep convolution in the cloud and snow recognition task is similar to the multi-scale fusion module that was previously introduced. We used the multi-scale fusion module to extract multi-scale features from the cloud and snow dataset. This module learns the multi-scale receptive field features and provides rich shallow semantic information. The ablation experiment shows that the module can improve the accuracy of cloud and snow recognition tasks, and proves the feasibility of multi-scale feature learning in cloud and snow recognition tasks. Compared with the multi-scale fusion module we designed previously, the parameter amount and floating-point calculation amount of hybrid deep convolution are smaller, and a wider range of receptive fields can be obtained. Figure 4 shows the structure of hybrid depth convolution. DW in the figure represents channel-by-channel convolution.

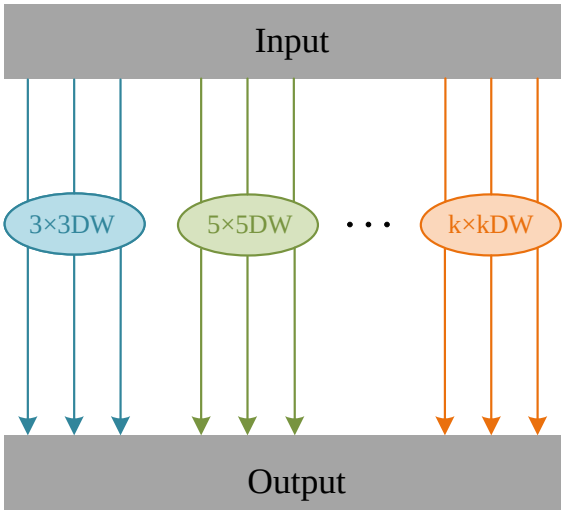

Figure 4. Schematic diagram of hybrid depth convolution.

The hybrid depth convolution first divides the feature map into multiple groups according to the channel. In general, the feature map is divided into 1–5 groups. In this paper, the feature maps are divided into four groups. After dividing into groups, we need to use different sizes of convolution kernels for each group to perform convolution, where we use channel-by-channel convolution. For the size of each group of convolution kernels, we set the size of the convolution kernel sequentially in the manner of $(2i+1) \times (2i+1)$, i represents the number of each group. Since the feature maps are divided into four groups, i is incremented from one in the first group to four in the fourth group. As for the division of convolution kernel channels, there are two division methods. The first type is equal division; that is, the number of channels of each group of convolution kernels is the same; the second type is divided by exponents. For the first $i-1$ group, the number of channels in each group accounts for 2^{-i} of the total number of channels. The number of channels in the last group is the remaining channels. In the model in this paper, the method of dividing by index is selected. After completing the above operations, we spliced these channels together to complete the hybrid depth convolution operation.

In the shallow feature extraction module in this article, the first layer is depth separable convolution, where the size of the convolution kernel for channel-by-channel convolution is 3×3, and the size of the convolution kernel for point-by-point convolution is $1 \times 1 \times 32$. In this way, the number of parameters involved in the convolution in this layer is $4 \times 3 \times 3 + 4 \times 32 = 164$. The second layer is still depth separable convolution, but the size of the convolution kernel of channel-wise convolution becomes 5×5, and the size of the convolution kernel of point-wise convolution becomes $1 \times 1 \times 64$. In this way, the number of parameters involved in the convolution in this layer is $32 \times 5 \times 5 + 32 \times 64 = 2848$. The above two layers of convolution are the same as the 7×7 receptive field in the range of the receptive field. The third layer is hybrid depth convolution. We divide the 64 channels of the second layer into 32, 16, 8, and 8 according to the index. The corresponding convolution kernel size is 3×3, 5×5, 7×7, and 9×9. For the number of parameters involved in this layer of convolution, there are $3 \times 3 \times 32 + 5 \times 5 \times 16 + 7 \times 7 \times 8 + 9 \times 9 \times 8 = 1728$. In summary, the number of parameters of the shallow feature extraction module is $164 + 2848 + 1728 = 4740$. For the L1 layer of ResNet18, we have $4 \times 7 \times 7 \times 64 = 12,544$ parameters. The number of parameters of the shallow feature extraction module of the lightweight feature map attention network is less than that of the $L1$ layer of ResNet18.

After the original 7×7 convolutional layer of ResNet18, the author added the ReLU activation function. We visualize the feature map after the activation function operation, as shown in Figure 5. From this group of pictures, we can see that the feature map after ReLU activation has a lot of loss. If the activation function is added in the shallow feature extraction of the model, it will not be conducive to the subsequent feature learning.

This phenomenon is determined by the nature of ReLU. If the input feature information is positive, then the output feature activated by ReLU is the restored input value. In this case, we can understand it as a linear transformation. However, if the input feature information is negative, then the feature information of the channel will be cleared, which will lose the feature information. In the shallow feature extraction module in this section, we did not choose the ReLU activation function after the three-layer convolution, but added the batch normalization and the Leaky ReLU activation function. The output of this function has a small slope for negative input. In this paper, it is set to $a = 0.01$, so that the output tends to be activated in the negative area instead of being cleared.

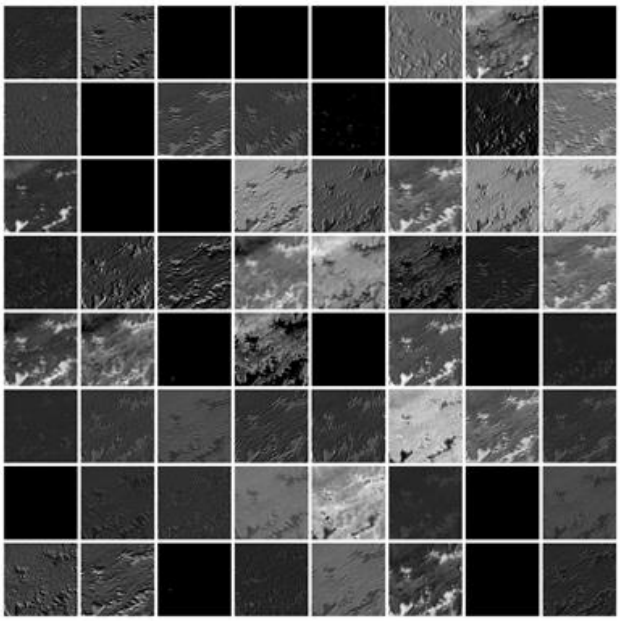

Figure 5. Feature map after ReLU activation function.

2.4. Intrinsic Feature Mapping Module

In an excellent convolutional neural network model, to ensure a comprehensive understanding of the input data, it will contain rich and even redundant feature maps [31]. These feature maps support the network architecture of the convolutional neural network and are an indispensable part of the network model. In the cloud and snow recognition task, we visualize the feature map of the $L1$ layer in the ResNet18 structure, as shown in Figure 6. To facilitate comparison, we present it in the form of a heat map. From this set of heat maps, we can see that for the 64 feature maps in this layer of convolution, some heat maps have similar details. For these similar feature maps, we can understand that they are formed through some kind of linear or non-linear transformation. For example, in an ideal situation, the feature maps of a certain layer are all learning edge features. There is horizontal edge information, left edge information, right edge information, etc. These feature maps can all pass through a feature map after conversion.

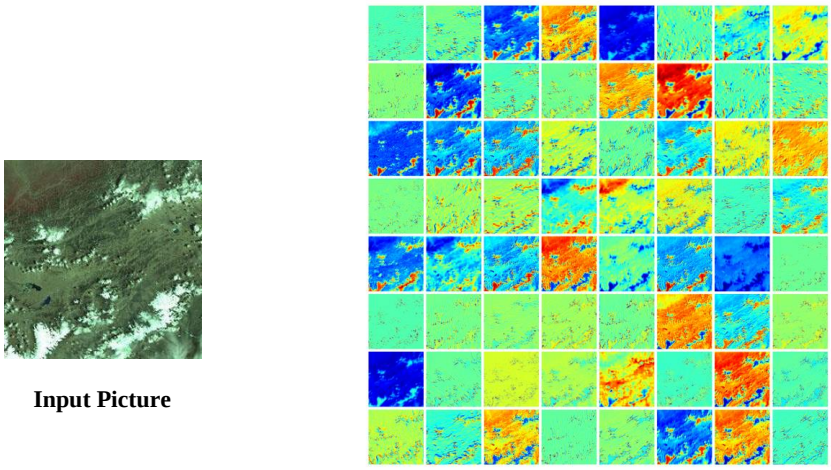

Figure 6. L1 layer feature heat map in ResNet18.

Meanwhile, for redundant feature maps in the network, instead of racking our brains to find ways to avoid redundant features, we might as well map these similar features from the existing feature maps through a cheap calculation. In this way, without changing the number of output feature maps, the number of network parameters and calculations is also reduced. This is the core idea of the internal feature mapping module. The intrinsic feature mapping module of this article is shown in Figure 7. This module uses two steps to obtain the same number of feature maps as conventional convolution.

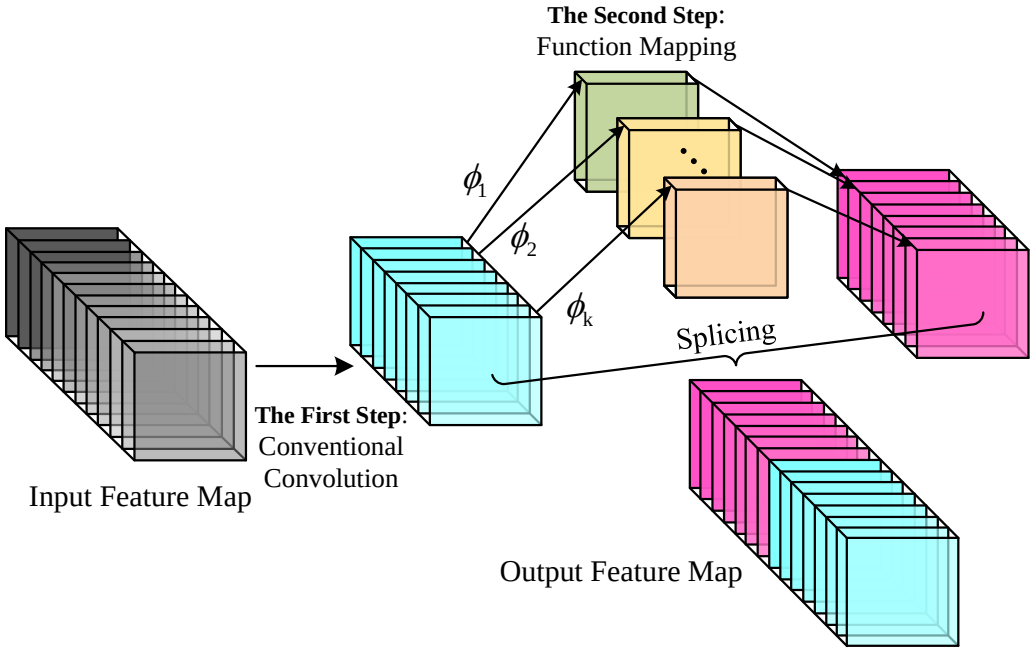

Figure 7. Intrinsic feature mapping module detailed diagram.

In traditional convolution, for a given input $h \times w \times c$, c is the number of channels, h and w are the height and width of the input feature map, respectively, when the size of the convolution kernel is $k \times k \times c$. When there are n convolution kernels, the size of the output feature map is $h' \times w' \times n$. Then, the convolution operation of this layer can be expressed as:

$$Y = X * f + b \tag{1}$$

where $*$ represents a convolution operation, X represents an input feature map, $f \in R^{c \times k \times k \times n}$ represents a convolution kernel in the layer, b represents a deviation term, and $Y \in R^{h' \times w' \times n}$ represents an output feature map of n channels.

In the internal feature mapping module, our first step is to perform regular convolution operations, but the number of convolution kernels will be strictly regulated. In the first step, we only extract m original feature maps for ordinary convolution operation, and m in this step is set to half of the normal convolution kernel, namely:

$$Y' = X * f' + b \tag{2}$$

Among them, $f' \in R^{c \times k \times k \times m}$ are the m convolution kernels used by the module. In this layer of convolution, the hyperparameters such as the size of the convolution kernel and the step length are the same as those in the ordinary convolution. The only difference is the number m of convolution kernels. For example, 64 convolution kernels are normally used in the network; we use 32 here.

After the original feature map is generated, in order for it to be the same as the ordinary convolution, we need to obtain another m feature maps. Next, we carry out the second step, through cheap linear or non-linear calculations: mapping to generate the remaining half of the feature map, Figure 7. ϕ_1 and ϕ_k represent the second step of the intrinsic feature mapping module. We use a series of cheap linear or non-linear operations on each original feature in Y'. In theory, each original feature can be mapped to generate s new feature maps:

$$y_{ij} = \phi_{i,j}(y'_i), \forall i = 1, \ldots, m, j = 1, \ldots, s \tag{3}$$

Among them, y'_i represents the i-th original feature map in Y' , the function $\phi_{i,j}$ represents the linear or non-linear function used for mapping to generate the j-th new feature map, and s represents how many times the original feature map has been linearly mapped. In other words, one or more feature maps can be mapped theoretically. By using cheap operations, we can get up to $m \times s$ new feature maps $\begin{bmatrix} y_{11} & y_{12} & \cdots & y_{mc} \end{bmatrix}$. The cheap operations here can use affine transformation, wavelet transformation, exponential transformation, etc., but these operations are too single and difficult to complete multi-feature conversion. In the end, we chose 3×3 channel-by-channel convolution, because the convolution operation contains many linear operations, such as smoothing, blurring, and motion. In this way, it is more intuitive to ensure that an original feature map corresponds to the mapping to generate a new feature map. After completing the above operations, we splice the feature maps generated in the first and second steps as the output feature map of the internal feature mapping module. The output feature map includes m original feature maps and m feature maps generated by feature mapping. It can be seen that the output feature maps obtained by the intrinsic feature mapping module are as many as those obtained by ordinary convolution, but the amount of parameters is less than that of ordinary convolution and is cost effective and efficient.

2.5. Lightweight Adaptive Attention Mechanism

We introduce the high-weight attention mechanism, combined with the densely connected backbone network, to reduce the false detection of cloud and snow samples in convolutional neural networks. The experimental results show that this mechanism can improve the discrimination ability of the cloud and snow recognition model, and the addition of this module can achieve higher accuracy, which is of great significance to the task of cloud and snow recognition. When learning the weight of each feature channel, the high-weight attention mechanism uses two layers of full connection, and the first layer of full connection contains the construction of scaling coefficient and activation function. Although this configuration can achieve more non-linearity, and thus better fit the correlation between the channels, the simple and crude way of scaling coefficients to reduce the number of parameters and reduce the amount of calculation destroys the most primitive and direct correspondence between the feature channels and their weights, and the interdependence in the feature graph is also disturbed [32].

Based on the high-weight attention mechanism, this paper further optimizes the model structure in the attention mechanism, reduces the parameters, and proposes a lightweight adaptive attention mechanism, whose structure diagram is as shown in Figure 8. The mechanism generates channel attention through one-dimensional convolution so that there is a direct correspondence between the channel and its weight. The re-weighted feature channel has a guiding significance for the subsequent feature learning. For the whole model, it can reduce the false detection phenomenon in the task of cloud and snow recognition.

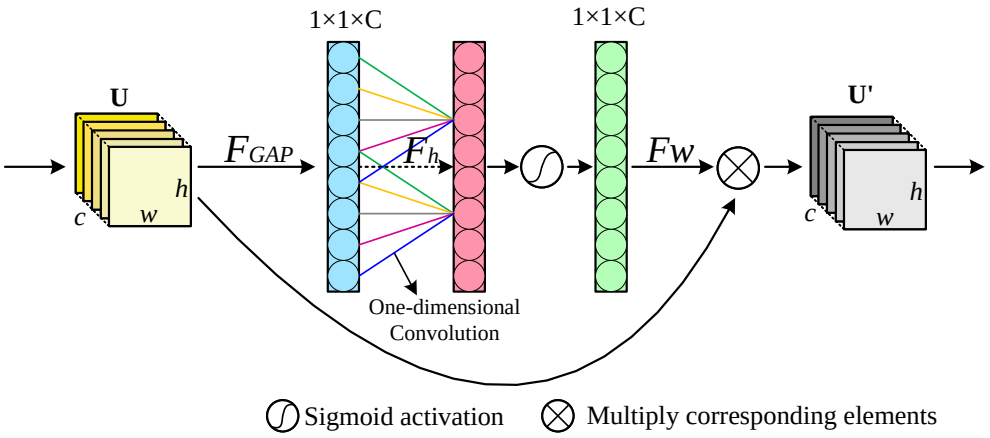

Figure 8. Lightweight adaptive attention mechanism.

In the lightweight adaptive attention mechanism, for a given input feature map U, the size is $H \times w \times C$. We still perform the averaging operation on each channel independently; that is, to make the two-dimensional feature map one-dimensional. Through this operation, each feature channel becomes a real number. The purpose of this is also to characterize the global receptive field of its corresponding feature map $(h \times w)$ to some extent. The formula is as follows:

$$Z_c = F_{GAP}(x_c) = \frac{1}{w \times h} \sum_{i=1}^{w} \sum_{j=1}^{h} x_c(i,j) \tag{4}$$

where x_c represents the c-th feature map in the input U. z_c represents the output of the c-th channel. After the above formula is applied to each channel in turn, the size of the output feature map becomes $1 \times 1 \times C$. After that, instead of learning the relationship between channels through the fully connected layer, we construct a cross-channel weight matrix. The re-learning of the channel feature weights is completed.W_k The expression form of the cross-channel weight matrix is shown in the Formula (5). That is, in the cross-channel weight matrix, the 1st to kth items in the first row are non-zero items, and the other items are all zero. In row 2, terms 2 through $k+1$ are non-zero, the others are zero, and so on. The cross-channel weight matrix is used to capture the interrelationships between feature channels, where k represents the range of influence of cross-channel interactions. This coefficient can be understood as: when calculating the importance of a channel, there will be k neighboring channels to multiply with the cross-channel weight matrix, which means that only these K neighboring channels participate in the attention prediction of the channel. From the perspective of the number of parameters, compared with the method of learning weights with two fully connected layers, the cross-channel weight matrix contains only $k \times c$ parameters.

$$\begin{bmatrix} w^{1,1} & \cdots & w^{1,k} & 0 & 0 & \cdots & \cdots & 0 \\ 0 & w^{2,2} & \cdots & w^{2,k+1} & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & w^{c,c-k+1} & \cdots & w^{c,c} \end{bmatrix} \tag{5}$$

Further, the attention weight of the cth channel in the feature map U can be expressed as:

$$w_c = F_h(w,z) = \sum_{j=1}^{k} w_c^j z_c^j, z_c^j \in \Omega_c^k \tag{6}$$

Among these, w_c^j represents a cross-channel weight matrix coefficient corresponding to the channel, Ω_c^k expresses z_c, a corresponding set of k adjacent characteristic chan-

nels, w_c represents the corresponding weight learned by the cth feature channel, and $W_c = [w_1\ w_2\ \cdots\ w_c]$.In order to further reduce the parameters and ensure that the weights of each channel and its k neighboring channels can be optimized at the same time, we let all the feature channels share the weight information. (7) Updated to:

$$w_c = F_h(w, z) = \sum_{j=1}^{k} w^j z_c^j, z_c^j \in \Omega_c^k \tag{7}$$

At this point, the number of parameters of the lightweight adaptive attention mechanism becomes k. For formula (7), we can do this with a one-dimensional convolution. Therefore, in the lightweight adaptive attention mechanism, we finally complete the information exchange between feature channels through one-dimensional convolution with convolution kernel size k, and the formula can be written as:

$$w_c = C1D_k(z) \tag{8}$$

Where $C1D$ represents a one-dimensional convolution. It should be noted here that to ensure the one-to-one correspondence between the characteristic channel and its attention prediction weight, the padding term in the one-dimensional convolution needs to be set to $k/2$ and rounded down (Figure 9). It represents the real process of weight learning by one-dimensional convolution when the influence range of cross-channel interaction in the lightweight adaptive attention mechanism is $k = 5$.

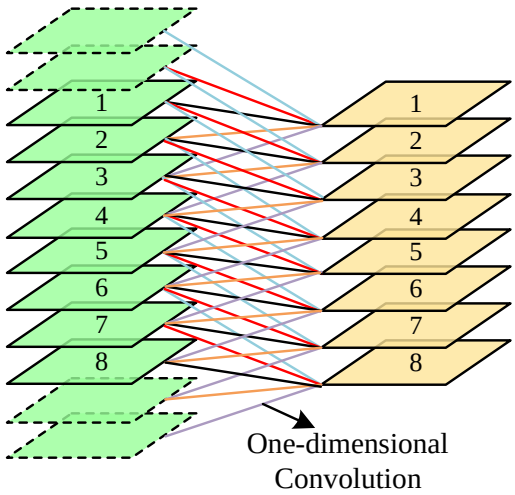

Figure 9. Process diagram before and after one-dimensional convolution operation (K = 5).

The influence range k of cross-channel interaction—that is, the kernel size of one-dimensional convolution—is an important parameter of the lightweight adaptive attention mechanism. In different convolutional neural network models, we can manually adjust the parameter k to achieve the best coverage. However, manual tuning through cross-validation is computationally expensive and time consuming. We know that the more input feature channels, the greater the interaction of spatial information between channels, which means that the influence range K of cross-channel interaction is greater. In other words, the dimension C of the characteristic channel is proportional to the parameter k. Further, there is a certain mapping between the dimension C of the characteristic channel and the influence range k of the cross-channel interaction $\phi(k)$:

$$C = \phi(k) \tag{9}$$

We know that the simplest mapping is a linear function, namely $\phi(k) = \gamma * k - b$. However, the relationship between input and output that can be characterized by linear function is limited, and the phenomenon of under-fitting is serious. In addition, the dimension C of the characteristic channel is usually artificially set to a power of two. Therefore, we will use the linear function $\phi(k) = \gamma * k - b$. The expansion is a non-linear function, namely:

$$C = \phi(k) = 2^{(r*k-b)} \tag{10}$$

Then, given the characteristic channel dimension C, we calculate the characteristic channel dimension using the formula. The method adaptively determines the kernel size k:

$$k = \psi(C) = |\frac{\log_2 C}{\gamma} + \frac{b}{\gamma}|_{odd} \tag{11}$$

Among these, $|t|_{odd}$ represents the odd number closest to t. In this section, $\gamma = 2$, $b = 1$. Obviously, by non-linear mapping $\psi(C)$, the input high-dimensional feature channel can interact with more neighboring feature channels in a wider range.

After doing this, we add the Sigmoid activation function to get a normalized weight between 0 and 1. However, because the weights of the normalized characteristic channels are between 0 and 1, the correlation and difference of the relationship between some characteristic channels are not obvious. Therefore, we further amplify the weights learned through one-dimensional convolution to highlight the characteristics of the channel itself. In Formula (12), n is the weight amplification coefficient; for example, n is 10, that is, the weight interval is 0–10, which makes the important weight more significant, and the formula is:

$$F_w = n \cdot \sigma(W_c) \tag{12}$$

Additionally, σ indicates Sigmoid is active. Finally, this weight is weighted to the features of each channel as follows:

$$U' = F_w \cdot U \tag{13}$$

The suppression or enhancement of the feature channel weight is completed through the above operations, thereby improving the overall recognition capability of the cloud and snow recognition model.

2.6. Introduction to the Dataset

The datasets in this paper are four remote sensing images in the visible spectral band taken by CCD cameras from the Chinese HJ-1A/1B satellites dedicated to Earth observation for environmental and disaster monitoring. Both HJ-1A and HJ-1B carry two CCD cameras, which are capable of multispectral detection. The difference is that HJ-1A carries an additional hyperspectral imager (HSI), while HJ-1B carries an infrared camera (IRS). Table 3 shows the specific parameters of the visible spectral band of the satellite.

Table 3. HJ-1A/1B satellite 4 spectrum parameters.

Platform	Payload	Spectrum Number	Spectral Range (μm)	Spatial Resolution (m)	Width (km)
HJ-1A/B	CCD(2)	1	0.43–0.52	30	360 (1 unit), 700 (2 units)
		2	0.52–0.60		
		3	0.63–0.69		
		4	0.76–0.90		

The images in this paper are randomly collected remote sensing images of the Qinghai–Tibetan Plateau and Yunnan–Guizhou Plateau regions in China. In the dataset of this paper, there are 11,000 samples of thick cloud, thin cloud and snow cover, and there are 14,000 samples of no snow and no clouds.

3. Experimental Results and Analysis

All models in this paper are neural networks based on the Pytorch framework under the Windows system. The hardware configuration of the computer is: GeForce RTX 2070 8 GB discrete graphics, 16 GB memory, Intel Core i7-8700 processor. There are the following reasons for changing the framework. First, keras, due to the framework itself, has a relatively advanced encapsulation, and cannot modify the training details, so it is not suitable for algorithm research. Second, at present, many researchers have found that Pytorch's training speed and accuracy are better than Keras's after comparing the performance of various frameworks through experiments.

In the analysis of the experimental results in this paper, we first use the accuracy and Kappa index to quantitatively analyze the performance of various networks in cloud and snow recognition tasks. The accuracy and Kappa index of six convolutional neural networks and eight lightweight networks in cloud and snow recognition tasks are listed in Table 4. From Table 4, we can see that in the convolutional neural network model, ResNeXT18 [33], the test accuracy and Kappa index were the lowest, 89.30% and 85.68%, respectively (Xception) [34]. The test accuracy and Kappa index were the highest, 93.40% and 91.18%. However, compared with the test accuracy and Kappa index of the lightweight feature mapping attention network, it is 1.62% and 2.16% lower, respectively.

Table 4. Accuracy and Kappa index of each cloud and snow recognition model.

Model	Accuracy Rate	Kappa Index	Model	Accuracy Rate	Kappa Index
ResNet18	0.8985	0.8642	MobileNetV1	0.9242	0.8986
ResNeXT18	0.8930	0.8568	MobileNetV2	0.9242	0.8986
Res2net18	0.9021	0.8691	MobileNetV3	0.9215	0.8949
Vgg16	0.9117	0.8819	MobileNetV1	0.9328	0.9101
Inceptionv3	0.9146	0.8857	MobileNetV2	0.9370	0.9157
Xception	0.9340	0.9118	EfficientNet	0.9128	0.8833
SqueezeNet	0.8855	0.8468	Lw-fmaNet	0.9502	0.9334

We apply the current mainstream lightweight model to the task of cloud and snow recognition and get the experimental results. In lightweight networks, we have SqueezeNet [35]. The test accuracy and Kappa index are the lowest: the test accuracy is 88.55% and the Kappa index is 84.68%. This is followed by EfficientNet [36]. Its test accuracy is 91.28%, and its Kappa index is 88.33%. The test accuracy of other lightweight models is over 92%, and the Kappa index is over 89%. In these lightweight network models, for the cloud and snow recognition task, MobileNet V2 [37], the test accuracy and Kappa index are 94.26% and 92.31%, respectively. However, compared with the lightweight feature mapping attention network we designed, there is still a certain gap. The test accuracy of the lightweight feature mapping attention network is 95.02%, and the Kappa index is 93.34%, both of which are the highest among all the cloud and snow recognition models in this paper.

Next, we use the three indicators of accuracy, recall, and F1-score to further analyze the four types of samples in each model: no snow, no cloud, thin cloud, and thick cloud and snow. Relevant experimental results are shown in Tables 5 and 6 The highest value of each indicator in the four categories of samples is shown in bold font.

Table 5. Classification and evaluation index analysis of cloud and snow recognition network (1).

Sample \ Model		RNet2Net	Vgg16	Inception V3	Xception	EfficientNet	Lw-fmaNet
No snow and no clouds	Precision rate	0.9897	0.9822	0.9793	0.9870	0.9829	0.9831
	Recall rate	0.9707	0.9801	0.9754	0.9866	0.9754	0.9889
	F1-Score	0.9801	0.9812	0.9773	0.9868	0.9791	0.9860
Thin Cloud	Precision rate	0.9429	0.9577	0.9584	0.9670	0.9563	0.9716
	Recall rate	0.9567	0.9490	0.9549	0.9613	0.9480	0.9658
	F1-Score	0.9498	0.9533	0.9566	0.9641	0.9522	0.9687
Thick Cloud	Precision rate	0.8006	0.8381	0.8508	0.8777	0.8244	0.9015
	Recall rate	0.8793	0.8482	0.8508	0.8816	0.8802	0.9328
	F1-Score	0.8381	0.8431	0.8508	0.8796	0.8514	0.9169
Snow cover	Precision rate	0.8633	0.8538	0.8556	0.8928	0.8767	0.9386
	Recall rate	0.7847	0.8534	0.8630	0.8944	0.8320	0.9040
	F1-Score	0.8221	0.8536	0.8593	0.8936	0.8538	0.9210

Table 6. Classification and evaluation index analysis of cloud and snow recognition network (2).

Sample \ Model		SqueezeNet	MobileNet V1	MobileNet V2 V3	MobileNet V3	ShuffleNet V1	ShuffleNet V2
No snow and no clouds	Precision rate	0.9805	0.9777	0.9788	0.9815	0.9838	0.9841
	Recall rate	0.9634	0.9841	0.9852	0.9819	0.9884	0.9855
	F1-Score	0.9719	0.9809	0.9820	0.9817	0.9861	0.9848
Thin cloud	Precision rate	0.9089	0.9615	0.9668	0.9527	0.9746	0.9660
	Recall rate	0.9640	0.9558	0.9549	0.9640	0.9635	0.9576
	F1-Score	0.9356	0.9586	0.9608	0.9583	0.9691	0.9618
Thick cloud	Precision rate	0.7468	0.8563	0.8901	0.8300	0.8679	0.8553
	Recall rate	0.9310	0.8838	0.9199	0.9110	0.8918	0.9372
	F1-Score	0.8288	0.8699	0.9048	0.8686	0.8797	0.8944
Snow cover	Precision rate	0.9345	0.8901	0.9277	0.9165	0.8943	0.9409
	Recall rate	0.6627	0.8589	0.8999	0.8138	0.8739	0.8553
	F1-Score	0.7755	0.8742	0.9136	0.8621	0.8840	0.8960

Through the data of these three indicators, we find that in the snow-free and cloud-free samples, the accuracy rate of the model Res2Net [38] is the highest among all models, at 98.97%, and the accuracy rate of the lightweight feature mapping attention network is 98.31%. Among the thin cloud samples, the model with the highest accuracy rate is ShuffleNet V1 [39]. The accuracy rate of the lightweight feature mapping attention network is 0.3% lower. In the model ShuffleNetV2 [40], the recall rate of the thick cloud sample is the highest in all models, at 93.72%; the accuracy of snow samples is also the highest, at 94.09%. For the comprehensive classification evaluation index of the F1 score, the lightweight feature mapping attention network is superior to all other networks in the thin cloud, thick cloud, snow samples, and the F1 score of no snow and no cloud is very little different from that of Xception. In addition, we found that, although the test accuracy and Kappa index of MobileNet V2 are second only to the lightweight feature mapping attention network, its three indicators are not outstanding. To sum up, for cloud and snow recognition tasks, the classification performance of the lightweight feature mapping attention network is better than that of other networks.

Next, we give the generalization effect of cloud and snow recognition for different models (Figure 10). In this paper, we select a satellite cloud image of the plateau area with high complexity and many interference scenes and compare the lightweight feature mapping attention network with the popular convolutional neural network. Figure 10a is a satellite cloud image. Figure 10b shows the generalization effect of cloud and snow recognition of Res2Net18; the network improves the original residual block by connecting

the hierarchical residuals within the residual block and increasing the range of the hierarchical receptive field by 3 × 3 convolution. Figure 10c for the generalization effect of Vgg16, the network constructs a 16-layer convolutional neural network using a 3 × 3 convolution kernel and a maximum pooling operation. Figure 10d for the generalization effect of cloud and snow recognition of InceptionV3, the network adds an RMSProp optimizer and label smoothing based on InceptionV2. Figure 10e for the generalization effect of Xception, the model changes the convolution in Inception v3 into depth separable convolution and adds residual connection. Figure 10f generalization effect of cloud and snow recognition for a lightweight feature mapping attention network.

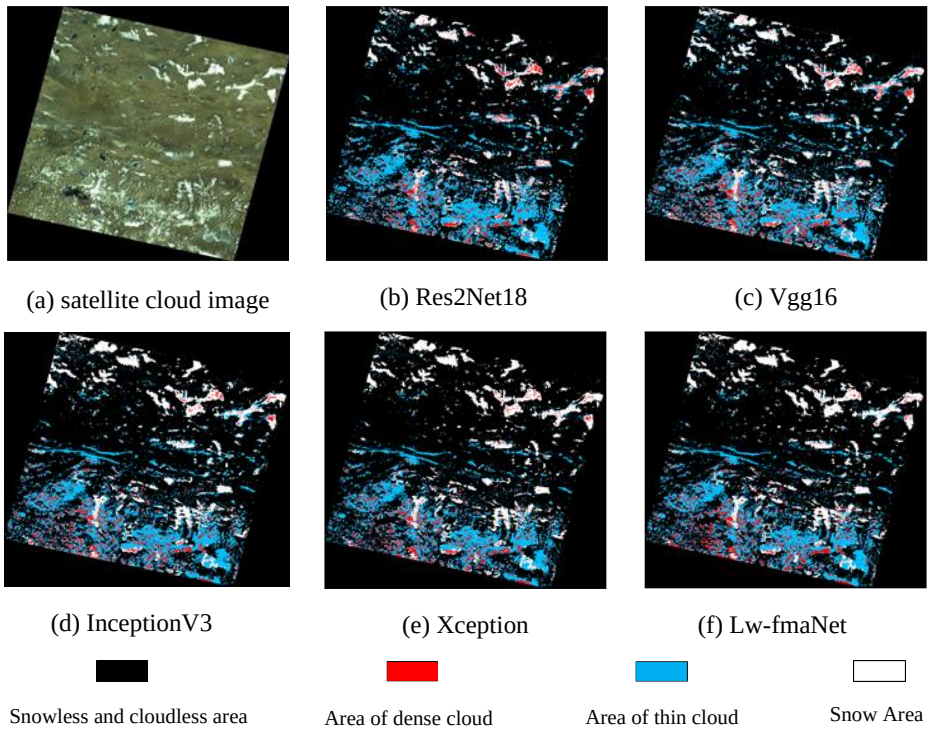

Figure 10. Cloud and snow recognition effect map of different Models in plateau area.

From this set of generalization results, we can see that the generalization effect of Res2Net18 and Vgg16 is the worst for the snow area in the upper right corner of the satellite cloud image. In the generalization maps of the two models, the area was widely misdetected as dense clouds. InceptionV3 and Xception have relatively few false detections, and the lightweight feature mapping attention network has the least false detections. In the lower-left corner of the satellite image, for this area, the generalization ability of the lightweight feature mapping attention network is the strongest, and other models have a serious phenomenon of missing detection. For the interference information of lakes and salt lakes, the model designed in this paper also has a good recognition, and classifies them as samples without snow and clouds. So, the generalization ability of the lightweight feature mapping attention network in the cloud and snow recognition task is better than that of the existing convolutional neural network.

Further, we compare the generalization effect graph of the lightweight feature mapping attention network with that of the lightweight model, as shown in Figure 11. The cloud and snow generalization effect map of SqueeeNet is formed through the accumulation of Fire modules. The module has two layers; the first layer passes through a 1 × 1 convolutional compression channel, and the second layer passes through 1 × 1 and 3 × 3 convolutional expansion channels. Figure 11c, for the cloud and snow generalization effect map of EfficientNet, a new network size scaling method is proposed, and the optimal network structure is searched through NAS. Figure 11d is the cloud and snow generalization effect

map of ShuffleNetV2. The network divides the channel into two branches; one branch performs the same mapping, and the other branch contains three convolutions. After the two branch operations are completed, the feature maps obtained by them are connected in series. Figure 11e is the cloud and snow generalization effect map of MobileNet V2. The network uses depth separable convolution and proposes inverted residual blocks with bottleneck structure. Figure 11f is the cloud and snow generalization effect map of the lightweight feature mapping attention network. In this set of generalization results, we can see that there is a large area of thick clouds in the lower left of the satellite cloud image of the plateau area, but there are a lot of false detections in SqueeeNet, EfficientNet, and ShuffleNetV2, and some areas of these models are mistakenly detected as snow, resulting in poor generalization effect in this area. The lightweight feature mapping attention network has less false detection in this area, so the recognition generalization effect in this area is the best. For the generalization effect of the whole graph, we can find that, compared with the other four lightweight models, in the generalization graph of the lightweight feature mapping attention network, the edge contour of each region is clear, which is closest to the real result.

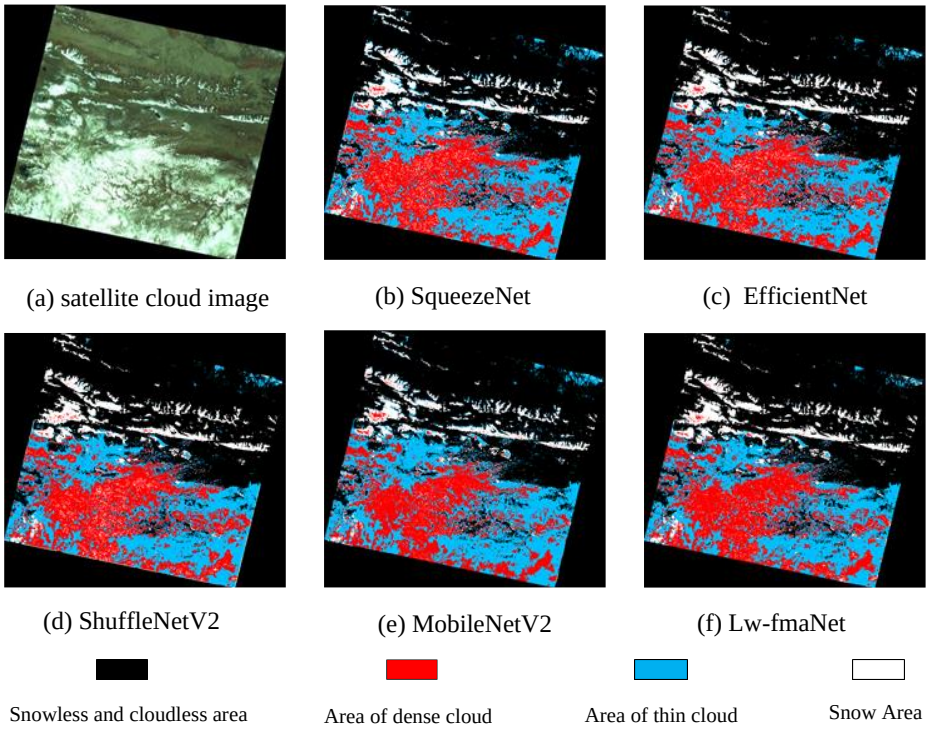

Figure 11. Generalization effect of cloud and snow recognition based on lightweight model.

In order to ensure the rigor of the experiment, we also give a set of generalization effect maps of the lightweight feature mapping attention network and other lightweight models, such as those shown in Figure 12. From this set of generalization maps, we can see that there are large areas of dense clouds in the upper right corner and lower right corner of the satellite cloud images, and compared with the other four lightweight networks, the lightweight feature mapping attention network model achieves the best generalization effect in these areas. From the overall generalization effect, SqueezeNet's generalization result is the worst, and a large number of dense clouds are missed. The generalization ability of lightweight feature mapping attention network is the strongest, and the recognition of details in satellite cloud images is also the best, which can better generalize the subtleties.

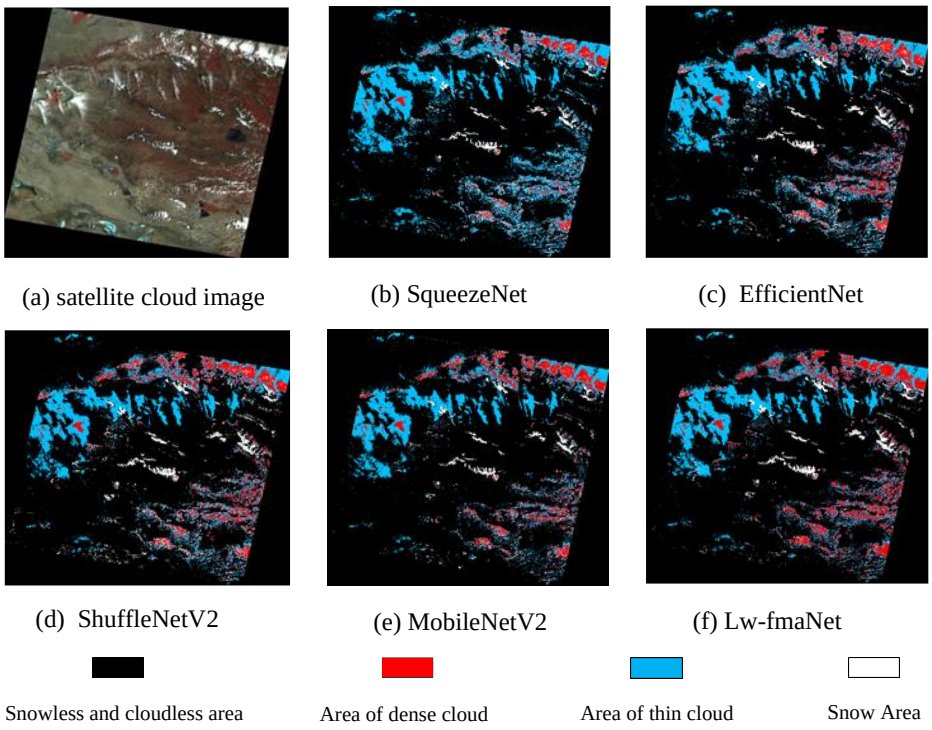

Figure 12. Generalization effect of cloud and snow recognition based on lightweight model.

Next, we analyze the model size and complexity of the lightweight feature mapping attention network and other convolutional neural networks. The number of parameters, the amount of floating-point computation, and the test time for each model are shown in Table 7.

Table 7. The number of parameters, the amount of floating-point computation, and the test time of each model.

Model	Number of Parameters (M)	Floating-Point Computation (G)	Test Time (s)
ResNeXT18	25.110	0.645	3.578
res2net18	15.881	0.052	4.337
Vgg16	15.247	0.209	3.086
inceptionv3	22.122	2.513	10.294
Xception	20.818	0.918	4.981
EfficientNet	8.413	0.018	4.136
Lw-fmaNet	5.617	0.234	2.276

From Table 7, we can see that the lightweight feature mapping attention network has the least number of parameters, only 5.617 M, followed by EfficientNet, which has 8.413 M. The model with the largest number of parameters is ResNeXT18, and its number of parameters reaches 25.110 M. In terms of floating-point computation, the model of inceptionv3 has the highest complexity of 2.513 G, and the model of EfficientNet has the lowest complexity. The floating-point computation is 0.018 G. The floating-point computation of the lightweight feature mapping attention network is 0.234 G, and the model complexity is in the middle position among the seven networks. In the test time, the lightweight feature mapping attention network takes the least time, only 2.276 s, and the Inceptionv3 takes the longest time, 10.294 s. Combining the three indicators, the lightweight feature mapping attention network has the best performance.

Finally, for the lightweight feature mapping attention network proposed in this paper, we specifically analyze the classification results in the cloud and snow recognition task. Table 8 maps the confusion matrix of the attention network for the lightweight feature.

Table 8. Confusion matrix for the results of the lightweight feature mapping attention network model.

	No Snow and No Clouds	Thin Cloud	Thick Cloud	Snow Cover	Total	Drawing Accuracy (%)
No snow and no clouds	2732	21	1	9	2763	98.89
Thin cloud	40	2119	33	2	2194	96.58
Thick cloud	0	32	2095	119	2246	93.28
Snow cover	7	9	195	1986	2197	90.40
Total	2779	2181	2324	2116	9400	/
User precision (%)	98.31	97.16	90.15	93.86	/	/

From Table 8, we can see that the snowless and cloudless samples have the lowest probability of being misrecognized, followed by the thin cloud samples. For snow samples, the mapping accuracy is 90.40%, which is the lowest among the four types of samples. The actual number of snow samples is 2197, the number of correct samples predicted by the lightweight feature mapping attention network is 1986, and 195 samples are incorrectly identified as thick clouds. For dense cloud samples, the mapping accuracy is 93.28%. Among the 2246 real dense cloud samples, 2095 samples are correctly predicted by the model in this section, and 119 samples are incorrectly identified as snow. These two sets of sample data show that due to the similarity of spectral information between dense clouds and snow, false detection will occur.

4. Summary

In this study, a lightweight feature mapping attention network (Lw-fmaNet) is proposed for efficient and accurate cloud snow recognition tasks. Our proposed network is adapted from the ResNet18 network architecture, including a shallow feature extraction module, an eigenfeature mapping module, and a lightweight adaptive attention mechanism. The shallow feature extraction module contains two layers of depth separable convolution and one layer of mixed depth convolution, aiming to extract more perceptual fields and reduce the occurrence of missed detection. Then in the intrinsic feature mapping module, the correlation and redundancy between cloud and snow feature maps are considered, so a small number of essential feature maps are first generated by convolution, and then new feature maps are generated by inexpensive computation, reducing the number of parameters and computation to achieve high efficiency and light weight. After the intrinsic feature mapping module, a lightweight adaptive attention mechanism is introduced. This mechanism is able to generate channel attention via fast one-dimensional convolution, so that there is a direct correspondence between the channel and its weights, with the aim of reducing the false detection phenomenon in cloud and snow recognition. In addition, the kernel size of the 1D convolution in this mechanism, i.e., the influence range of cross-channel interactions, can be determined adaptively by a non-linear mapping of channel sizes, without the need for manual adjustment of parameters. In the experiments of this paper, the method in this paper can ensure the accuracy and generalization ability of cloud snow identification while reducing the number of network parameters and computational effort. Moreover, our proposed lightweight feature map attention network has better performance in performing cloud snow recognition tasks compared with multiple convolutional neural networks and lightweight networks commonly used for cloud snow recognition.

However, there are still some shortcomings.

(1) In this paper, the recognition targets are thick clouds, thin clouds, snow cover, and no snow and no clouds, but in the practical application in real highland areas, the influence of ground conditions, such as lakes and rivers, needs to be considered, so more classes of data sets need to be added in the next research study.

(2) The confusion matrix in the article shows that the phenomenon of false detection still exists, so on top of the new computer hardware development, more optimized and advanced network models need to be designed to be able to perform the cloud map recognition task more accurately.

Author Contributions: Conceptualization, Min Xia, Chaoyun Yang and Yonghong Zhang; methodology, Min Xia and Chaoyun Yang; software, Chaoyun Yang; validation, Chaoyun Yang and Yonghong Zhang; formal analysis, Min Xia, Jia Liu and Haifeng Lin; investigation, Yonghong Zhang, Yang Li and Haifeng Lin; resources, Min Xia and Yonghong Zhang; data curation, Min Xia and Haifeng Lin; writing—original draft preparation, Chaoyun Yang; writing—review and editing, Min Xia and Jia Liu; visualization, Chaoyun Yang and Yang Li; supervision, Min Xia and Yonghong Zhang; project administration, Min Xia; funding acquisition, Min Xia. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key special projects of the national key R & D plan "Intergovernmental International Scientific and Technological Innovation Cooperation" (2021YFE0116900), National Natural Science Foundation of China (42175157), NUIST Students' Platform for Innovation and Entrepreneurship Training Program (XJDC202110300629), Key Projects of Jiangsu College Students' Innovation and Entrepreneurship Plan (202110300051).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data and the code of this study are available from the corresponding author upon request (zyh@nuist.edu.cn).

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

References

1. Qu, Y.; Xia, M.; Zhang, Y. Strip pooling channel spatial attention network for the segmentation of cloud and cloud shadow. *Comput. Geosci.* **2021**, *157*, 104940. [CrossRef]
2. Xia, M.; Qu, Y.; Lin, H. PANDA: Parallel asymmetric network with double attention for cloud and its shadow detection. *J. Appl. Remote Sens.* **2021**, *15*, 046512. [CrossRef]
3. Chen, B.; Xia, M.; Qian, M.; Huang, J. MANet: A multi-level aggregation network for semantic segmentation of high-resolution remote sensing images. *Int. J. Remote Sens.* **2022**, *13*. [CrossRef]
4. Zhang, J.; Zhou, Q.; Shen, X.; Li, Y. Cloud detection in high-resolution remote sensing images using multi-features of ground objects. *J. Geovis. Spat. Anal.* **2019**, *3*, 14. [CrossRef]
5. Wang, W.; Song, W.G.; Liu, S.X.; Zhang, Y.M.; Zheng, H.Y.; Tian, W. A cloud detection algorithm for MODIS images combining Kmeans clustering and multi-spectral threshold method. *Spectrosc. Spectr. Anal.* **2011**, *31*, 1061–1064.
6. Sekrecka, A.; Kedzierski, M.; Wierzbicki, D. Pre-processing of panchromatic images to improve object detection in pansharpened images. *Sensors* **2019**, *19*, 5146. [CrossRef]
7. Huang, W.; Wang, Y.; Chen, X. Cloud detection for high-resolution remote-sensing images of urban areas using colour and edge features based on dual-colour models. *Int. J. Remote Sens.* **2018**, *39*, 6657–6675. [CrossRef]
8. Zhan, Y.; Wang, J.; Shi, J.; Cheng, G.; Yao, L.; Sun, W. Distinguishing cloud and snow in satellite images via deep convolutional network. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1785–1789. [CrossRef]
9. Xia, M.; Wang, K.; Song, W.; Chen, C.; Li, Y. Non-intrusive load disaggregation based on composite deep long short-term memory network. *Expert Syst. Appl.* **2020**, *160*, 113669. [CrossRef]
10. Xia, M.; Zhang, X.; Weng, L.; Xu, Y. Multi-stage feature constraints learning for age estimation. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 2417–2428. [CrossRef]
11. Miao, S.; Xia, M.; Qian, M.; Zhang, Y.; Liu, J.; Lin, H. Cloud/shadow segmentation based on multi-level feature enhanced network for remote sensing imagery. *Int. J. Remote Sens.* **2022**. [CrossRef]
12. Lu, C.; Xia, M.; Qian, M.; Chen, B. Dual-Branch Network for Cloud and Cloud Shadow Segmentation. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5410012. [CrossRef]
13. Xia, M.; Liu, W.; Shi, B.; Weng, L.; Liu, J. Cloud/snow recognition for multispectral satellite imagery based on a multidimensional deep residual network. *Int. J. Remote Sens.* **2019**, *40*, 156–170. [CrossRef]
14. Ye, L.; Cao, Z.; Xiao, Y. DeepCloud: Ground-based cloud image categorization using deep convolutional features. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 5729–5740. [CrossRef]
15. Li, Z.; Shen, H.; Cheng, Q.; Liu, Y.; You, S.; He, Z. Deep learning based cloud detection for medium and high resolution remote sensing images of different sensors. *ISPRS J. Photogramm. Remote Sens.* **2019**, *150*, 197–212. [CrossRef]

16. Bai, T.; Li, D.; Sun, K.; Chen, Y.; Li, W. Cloud detection for high-resolution satellite imagery using machine learning and multi-feature fusion. *Remote Sens.* **2016**, *8*, 715. [CrossRef]
17. Chen, Y.; Fan, R.; Bilal, M.; Yang, X.; Wang, J.; Li, W. Multilevel cloud detection for high-resolution remote sensing imagery using multiple convolutional neural networks. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 181. [CrossRef]
18. Xie, F.; Shi, M.; Shi, Z.; Yin, J.; Zhao, D. Multilevel cloud detection in remote sensing images based on deep learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *10*, 3631–3640. [CrossRef]
19. Liu, S.; Duan, L.; Zhang, Z.; Cao, X.; Durrani, T.S. Multimodal ground-based remote sensing cloud classification via learning heterogeneous deep features. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 7790–7800. [CrossRef]
20. Song, L.; Xia, M.; Jin, J.; Qian, M.; Zhang, Y. SUACDNet: Attentional change detection network based on siamese U-shaped structure. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, *105*, 102597. [CrossRef]
21. Lu, C.; Xia, M.; Lin, H. Multi-scale strip pooling feature aggregation network for cloud and cloud shadow segmentation. *Neural Comput. Appl.* **2022**, *34*, 6149–6162. [CrossRef]
22. Xia, M.; Wang, Z.; Lu, M.; Pan, L. MFAGCN: A new framework for identifying power grid branch parameters. *Electr. Power Syst. Res.* **2022**, *207*, 107855. [CrossRef]
23. He, Y.; Zhang, X.; Sun, J. Channel pruning for accelerating very deep neural networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1389–1397.
24. Pang, K.; Weng, L.; Zhang, Y.; Liu, J.; Lin, H.; Xia, M. SGBNet: An Ultra Light-weight Network for Real-time Semantic Segmentation of Land Cover. *Int. J. Remote. Sens.* **2022**. [CrossRef]
25. Wu, J.; Leng, C.; Wang, Y.; Hu, Q.; Cheng, J. Quantized convolutional neural networks for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4820–4828.
26. Zhou, Z.; Li, S.; Shao, Y. Object-oriented crops classification for remote sensing images based on convolutional neural network. In *Proceedings of the Image and Signal Processing for Remote Sensing XXIV*; SPIE: Bellingham WA, USA, 2018; Volume 10789, p. 1078922.
27. Prakosa, S.W.; Leu, J.S.; Chen, Z.H. Improving the accuracy of pruned network using knowledge distillation. *Pattern Anal. Appl.* **2021**, *24*, 819–830. [CrossRef]
28. Gao, J.; Weng, L.; Xia, M.; Lin, H. MLNet: Multichannel feature fusion lozenge network for land segmentation. *J. Appl. Remote Sens.* **2022**, *16*, 016513. [CrossRef]
29. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
30. Tan, M.; Le, Q.V. Mixconv: Mixed depthwise convolutional kernels. *arXiv* **2019**, arXiv:1907.09595.
31. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1580–1589.
32. Zhang, Q.L.; Yang, Y.B. Sa-net: Shuffle attention for deep convolutional neural networks. In Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing, Toronto, ON, Canada, 6–11 June 2021; pp. 2235–2239.
33. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.
34. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
35. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
36. Tan, M.; Le, Q.V. EfficientNet: Rethinking model scaling for convolutional neural networks. *arXiv* **2020**, arXiv:1905.11946.
37. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
38. Gao, S.; Cheng, M.M.; Zhao, K.; Zhang, X.Y.; Yang, M.H.; Torr, P.H. Res2net: A new multi-scale backbone architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 652–662. [CrossRef]
39. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
40. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.