remote sensing

MDPI

Article

A Lightweight Convolutional Neural Network Based on Channel Multi-Group Fusion for Remote Sensing Scene Classification

Cuiping Shi [1,*], Xinlei Zhang [1] and Liguo Wang [2]

[1] College of Communication and Electronic Engineering, Qiqihar University, Qiqihar 161000, China; 2020935682@qqhru.edu.cn
[2] College of Information and Communication Engineering, Dalian Nationalities University, Dalian 116000, China; wangliguo@hrbeu.edu.cn
* Correspondence: shicuiping@qqhru.edu.cn

Abstract: With the development of remote sensing scene image classification, convolutional neural networks have become the most commonly used method in this field with their powerful feature extraction ability. In order to improve the classification performance of convolutional neural networks, many studies extract deeper features by increasing the depth and width of convolutional neural networks, which improves classification performance but also increases the complexity of the model. To solve this problem, a lightweight convolutional neural network based on channel multi-group fusion (LCNN-CMGF) is presented. For the proposed LCNN-CMGF method, a three-branch downsampling structure was designed to extract shallow features from remote sensing images. In the deep layer of the network, the channel multi-group fusion structure is used to extract the abstract semantic features of remote sensing scene images. The structure solves the problem of lack of information exchange between groups caused by group convolution through channel fusion of adjacent features. The four most commonly used remote sensing scene datasets, UCM21, RSSCN7, AID and NWPU45, were used to carry out a variety of experiments in this paper. The experimental results under the conditions of four datasets and multiple training ratios show that the proposed LCNN-CMGF method has more significant performance advantages than the compared advanced method.

Keywords: remote sensing scene image classification (RSSIC); channel fusion; convolutional neural network (CNN); channel multi-group fusion (CMGF); lightweight; downsampling

Citation: Shi, C.; Zhang, X.; Wang, L. A Lightweight Convolutional Neural Network Based on Channel Multi-Group Fusion for Remote Sensing Scene Classification. *Remote Sens.* **2022**, *14*, 9. https://doi.org/10.3390/rs14010009

Academic Editors: Kun Tan, Jie Feng, Qian Du and Xue Wang

Received: 21 November 2021
Accepted: 20 December 2021
Published: 21 December 2021

1. Introduction

The goal of remote sensing scene image classification is to correctly classify the input remote sensing images. Due to the wide application of remote sensing image classification in natural disaster detection, land cover analysis, urban planning and national defense security [1–4], the classification of remote sensing scene images has attracted extensive attention. In order to improve the performance of remote sensing scene classification, many methods have been proposed. Among them, convolutional neural networks have become one of the most successful deep learning methods with their strong feature extraction ability. Convolutional neural networks are widely used in image classification [5] and target detection [6]. Many excellent neural networks have been designed for image classification. For example, Li et al. [7] proposed a deep feature fusion network for remote sensing scene classification. Zhao et al. [8] proposed a multi-topic framework combining local spectral features, global texture features and local structure features to fuse features. Wang et al. [9] used an attention mechanism to adaptively select the key parts of each image, and then fused the features to generate more representative features.

In recent years, designing a convolutional neural network to achieve the optimal trade-off between classification accuracy and running speed has become a research hotspot. SqueezeNet [10] designed a lightweight network by squeezing and extending modules to

reduce parameter weight. In the SqueezeNet structure, three strategies are mainly used to reduce the parameters of the model. Firstly, partial 3×3 convolution is replaced by 1×1 convolution, then the number of input channels of 3×3 convolution kernel is reduced, and downsampling is carried out in the later part of the network to provide a larger features map for the convolution layer. The traditional convolution is decomposed in MobileNetV1 [11] to obtain the depthwise separable convolution. The depthwise separable convolution is divided into two independent processes: a lightweight depthwise convolution for spatial filtering and a 1×1 convolution for generating features, which separates spatial filtering from the feature generation mechanism. MobileNetV2 [12] added a linear bottleneck and inverted residuals structure on the basis of MobileNetV1, which further improves the performance of the network. SENet [13] proposed the SE module, which consists of extrusion and expansion. Firstly, extrusion is accomplished by global average pooling, which transforms the input two-dimensional feature channel into a real number with a global receptive field. Then, expansion is accomplished through the full connection layer, and a set of weighting parameters is obtained. Finally, channel-by-channel weighting is multiplied to complete the re-calibration of the original feature on the channel dimension. NASNet [14] used the enhanced learning and model search structure to learn a network unit in a small dataset, then stacked the learned units on a large dataset, which solves the problem that the previous neural network search structure cannot be applied to a large dataset. MobileNetV3 [15] added a SE module and used a neural structure search to search for network configuration and parameters. ResNet [16] solved performance degradation due to network depth using residual connectivity, and presents an efficient bottleneck structure with satisfactory results. Xception [17] replaced the convolution operation in the Inception module with depthwise separable convolution and achieves better performance. GoogleNet [18] used the Inception module to make the network deeper and wider. The Inception module consists of three convolution branches and one pooled branch, and finally four branches were fused through channels.

Grouped convolution was first used in AlexNet [19]. Due to the limitations of hardware conditions at that time, it was used in AlexNet to slice the network, which made it run parallel with two GPUs and achieved good performance. The validity of grouping convolution is well demonstrated in ResNeXt [20]. ResNeXt highly modularizes the network structure, building a network architecture by repeating stacked modules. The module is composed of several bottleneck structures, which improves the accuracy of the model without increasing the number of parameters. Traditional channel grouping uses a single grouping form (for example, the number of channels of input features is C, g is the number of groups, and the number of channels in each group is C/g). Using a single channel group is not conducive to feature extraction; to solve this problem, we proposed a channel multi-group convolution structure. The structure classifies the input features into two types of grouping, the number of channels for each set of features in the first type is C/g, and the number of channels for each set of features in the other type is $2C/g$. The channel multi-group structure increases the diversity of features while decreasing the parameters further. To reduce the loss of feature information during the grouping convolution process, residual connection is added to the channel multi-group structure, which can effectively avoid the disappearance of gradient due to network deepening. In order to solve the problem of network performance degradation caused by lack of information interaction between individual groups during group convolution, channel fusion of adjacent features is carried out to increase information interaction and improve network feature representation ability.

The main contributions of this study are as follows.

(1) In the shallow layer of the network, a shallow feature extraction module is constructed. The module is composed of three branches. Branch 1 uses two consecutive 3×3 convolution for downsampling and feature extraction, branch 2 utilizes max-pooling and 3×3 convolution for downsampling and feature extraction. Branch 3 is a shortcut branch. The fused features of branch 1 and branch 2 are shortcut with branch 3.

The module can fully extract the shallow feature information, so as to accurately distinguish the target scene.

(2) In the deep layer of the network, a channel multi-group fusion module is constructed for the extraction of deep features, which divided the input features into features with channel number of C/g and channel number of $2C/g$, increasing the diversity of features.

(3) To solve the problem of lack of information interaction for features between groups due to group convolution, in the channel multi-group module, the channel fusion of adjacent features is utilized to increase the information exchange, which significantly improves the performance of the network.

(4) A lightweight convolutional neural network is constructed based on channel multi-group fusion (LCNN-CMGF) for remote sensing scene image classification, which includes shallow feature extraction module and channel multi-group fusion module. Moreover, a variety of experiments are carried out under the conditions of four datasets of UCM21, RSSCN7, AID and NWPU45, and the experimental results prove the proposed LCNN-CMGF method achieves the trade-off between model classification accuracy and running speed.

The rest of this paper is as follows. In Section 2, the overall structure, shallow feature extraction module and channel multi-group module of the proposed LCNN-CMGF method are described in detail. Section 3 provides the experimental results and analysis. In Section 4, several visualization methods are adopted to evaluate the proposed LCNN-GMGF method. The conclusion of this paper is given in Section 5.

2. Methods

2.1. The Overall Structure of Proposed LCNN-CMGF Methods

As shown in Figure 1, the proposed network structure is divided into eight groups, the first three being used to extract shallow information from remote sensing images. Groups 1 and 2 adopt a proposed shallow downsampling structure, which is introduced in Section 2.2 in detail. Group 3 uses a hybrid convolution method combining standard convolution and depthwise separable convolution for feature extraction. Depthwise separable convolution has a significant reduction in the number of parameters compared with standard convolution. Assuming that the input feature size is $H \times W \times C_1$, the convolution kernel size is $H_1 \times W_1 \times C_1$ and the output feature size is $H \times W \times C_2$, the parameter quantity of standard convolution is:

$$params_{conv} = H_1 \times W_1 \times C_1 \times C_2 \tag{1}$$

The parameter quantities of depthwise separable convolution is:

$$params_{dsc} = H_1 \times W_1 \times C_1 + C_1 \times C_2 \tag{2}$$

The ratio $params_{dsc}/params_{conv}$ of depthwise separable convolution to standard convolution is:

$$\frac{params_{dsc}}{params_{conv}} = \frac{H_1 \times W_1 \times C_1 + C_1 \times C_2}{H_1 \times W_1 \times C_1 \times C_2} = \frac{1}{C_2} + \frac{1}{H_1 \times W_2} \tag{3}$$

According to Equation (3), when the convolution kernel size $H_1 \times W_2$ is equal to 3×3, due to $C_2 \gg H_1 \times H_2$, the parameter quantity of standard convolution is approximately 9 times that of depthwise separable convolution, and when the convolution kernel size $H_1 \times W_2$ is equal to 5×5, the parameter of standard convolution is approximately 25 times that of depthwise separable convolution. With the increase in convolution kernel size, the parameter will be further reduced. However, depthwise separable convolution can inevitably lead to the loss of some feature information while significantly reducing the amount of parameters, and then make the learning ability of the network decline. Therefore, we propose to use the hybrid convolution of standard convolution and depthwise separable convolution for feature extraction, which not only reduces the weight parameters, but also

improves the learning ability of the network. From group 4 to group 7, channel multi-group fusion structure is used to further extract deep feature information. Channel multi-group fusion structure can generate a large number of features with a few parameters to increase the feature diversity. Assuming that the input feature size is $H \times W \times C_1$, the convolution kernel size is $H_1 \times W_1 \times C_1$ and the output feature size is $H \times W \times C_2$, the parameter quantity of standard convolution is

$$params_{conv} = H_1 \times W_1 \times C_1 \times C_2 \tag{4}$$

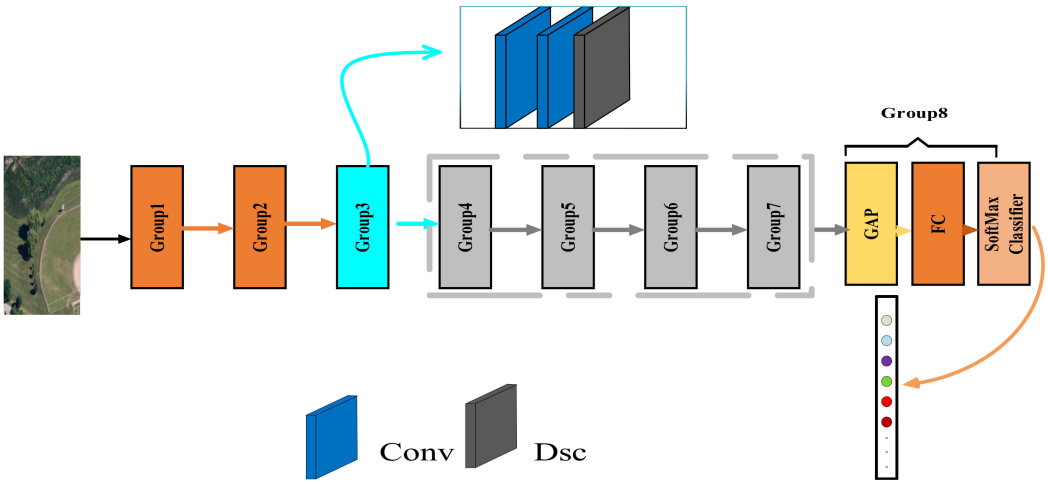


Figure 1. The proposed LCNN-CMGF network structure. Groups 1 and 2 are shallow downsampling modules. Group 3 is hybrid convolution. Groups 4–7 are channel multi-group fusion modules. Group 8 is the global average pooling, fc and classifier.

Divide the input features into t groups along the channel dimension, then the feature size of each group is $H \times W \times \frac{C_1}{t}$, the corresponding convolution kernel size is $H_1 \times W_1 \times \frac{C_1}{t}$, and the output feature size of each group is $H \times W \times \frac{C_2}{t}$. Connect the obtained t group features along the channel dimension to obtain that the final output feature size is $H \times W \times C_2$. The parameter quantity of the whole process is

$$params_{gconv} = H_1 \times W_1 \times \frac{C_1}{t} \times \frac{C_2}{t} \times t = H_1 \times W_1 \times C_1 \times C_2 \times \frac{1}{t} \tag{5}$$

As shown in Equations (4) and (5), the parameter quantity of group convolution is $1/t$ of the standard convolution parameter quantity, that is, under the condition of the same parameter quantity, the number of features obtained by group convolution is t times of the standard convolution, which increases the feature diversity and effectively improves the classification accuracy. The details are described in Section 2.3. Group 8 consists of a global average pooling layer (GAP), a fully connected layer (FC), and a softmax classifier to convert the convolutionally extracted feature information into probabilities for each scenario. Because features extracted by convolution contain spatial information, which is destroyed if features derived by convolution are directly mapped to the feature vector through a fully connected layer, and global average pooling does not. Assuming that the output of the last convolution layer is $O = [o_1; o_2; \ldots; o_i; \ldots o_N] \in \mathbb{R}^{N \times H \times W \times C}$, $[;;;]$ represents cascading operations along the batch dimension, and $\mathbb{R}$ represents the set of real numbers. In addition, N, H, W, C represent the number of samples per training, the height of the feature, the width of the feature, and the number of channels, respectively. Suppose the result of global average pooling is $P = [p_1; p_2; \ldots; p_i \ldots; p_N] \in \mathbb{R}^{N \times 1 \times 1 \times C}$, then the

processing process of any $P = [p_1; p_2; \ldots; p_i \ldots; p_N] \in \mathbb{R}^{N \times 1 \times 1 \times C}$ with the global average pooling layer can be represented as

$$p_i = \frac{1}{H \times W} \sum_{h=1}^{H} \sum_{w=1}^{W} o_i \tag{6}$$

As shown in Equation (6), global average pooling more intuitively maps the features of the last layer convolution output to each class. Additionally, the global average pooling layer does not require weight parameters, which can avoid overfitting phenomena during training the model. Finally, a softmax classifier is used to output probability values for each scenario.

2.2. The Three-Branch Shallow Downsampling Structure

Max-pooling downsampling is a nonlinear downsampling method. For small convolutional neural networks, better nonlinearity can be obtained by using maximum pool downsampling. On the contrary, for deep neural networks, multi-layer superimposed convolutional downsampling can learn better nonlinearity than max-pooling according to the training set, as shown in Figure 2. Figure 2a,b represents convolution downsampling and max-pooling downsampling, respectively. The convolution downsampling in Figure 2a first uses the 3×3 convolution with step size of 1 for feature extraction of the input data, and then uses the 3×3 convolution with step size of 2 for downsampling. In the max-pooling downsampling in Figure 2b, the input feature is extracted by 3×3 convolution with step size of 1, and then the max-pooling downsampling with step size of 2 is adopted. Combining max-pooling downsampling and convolutional downsampling, we propose a three-branch downsampling structure as shown in Figure 3 for feature extraction, and use the input features to compensate the downsampling features, which can not only extract strong semantic features, but also retain shallow information.

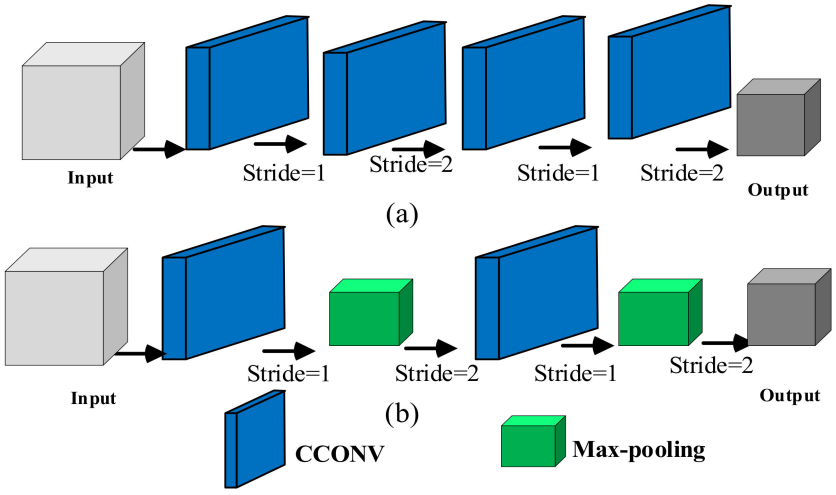


Figure 2. The structure of traditional shallow sampling module. (**a**) Convolution downsampling. (**b**) Max-pooling downsampling.

In groups 1 and 2 of the network, we use the structure shown in Figure 3 to extract shallow features. The structure is divided into three branches. The first branch uses 3×3 convolution with step size of 2 to obtain f_{down}, and then uses 3×3 convolution with step size of 1 to extract the shallow features of the image to obtain $f_1(x)$. That is

$$f_{down}(x) = \delta(BN(F * K_{s=2})) \tag{7}$$

$$f_1(x) = \delta(BN(f_{down}(x) * K_{s=1})) \tag{8}$$

In Equations (7) and (8), δ represents the activation function $Rule$, BN represents batch standardization, F represents the input characteristics, K_s represents the 3×3 convolution kernel with step size s, and $*$ represents the convolution operation.

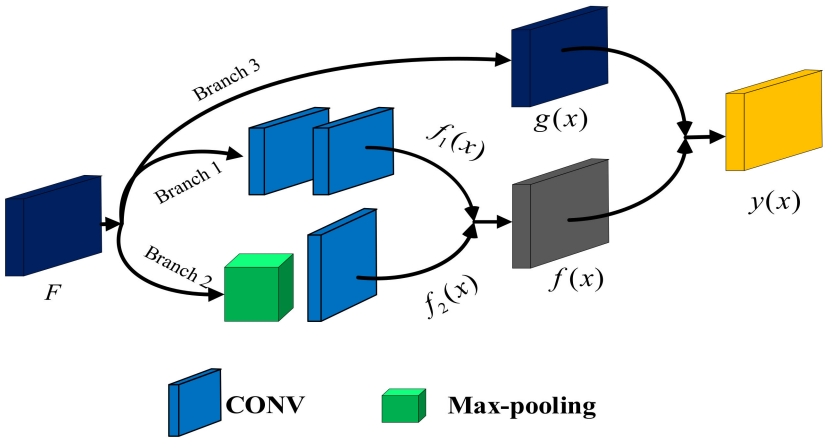


Figure 3. Three-branch shallow downsampling structure.

The second branch uses the max-pooling with step size of 2 to downsample the input features to obtain f_{mij}. The most responsive part of the max-pooling selection features enters the next layer, which reduces the redundant information in the network and makes the network easier to be optimized. The max-pooling downsampling can also reduce the estimated mean shift caused by the parameter error of the convolution layer, keep more texture information. Then, the shallow features $f_2(x)$ are extracted by 3×3 convolution with step size of 1. That is

$$f_{mij} = \max_{(s,t) \in R_{ij}} x_{mst} \tag{9}$$

$$f_2(x) = \delta(BN(f_{\mathrm{mij}}(x) * K_{s=1})) \tag{10}$$

In Equation (9), f_{mij} represents the max-pooling output value in rectangular area R_{ij} related to the m-th feature, and x_{mst} represents the element at the (s, t) position in rectangular area R_{ij}.

The fused feature $f(x)$ is obtained by fusing the features from Branch 1 and Branch 2. To reduce the loss of feature information caused by the first two branches, a residual branch is constructed to compensate for the loss of information. The fused feature $f(x)$ and the third branch are fused to generate the final output feature $y(x)$. That is

$$y(x) = g(x) + f(x) \tag{11}$$

The $g(x)$ in Equation (11) is a residual connection implemented by 1×1 convolution.

2.3. Channel Multi-Group Fusion Structure

The proposed channel multi-group fusion structure is shown in Figure 4. It divides the input features with the number of channels C into two parts, one part is composed of 4 features with the number of channels $\frac{C}{4}$, and the other part is composed of 2 features with the number of channels $\frac{C}{2}$. First, the convolution operations are performed for features with the number of channels $\frac{C}{4}$, the adjacent two convolution results are channel concatenated, the number of feature channels after concatenate is $\frac{C}{2}$. Then, the convolution operations are performed on features with the number of channels $\frac{C}{2}$, the adjacent two features convolution results are channel concatenated, the number of channels of each feature after fusion was C. The convolution operations are performed on features with the number of channels C, and the convolution results are fused to obtain the output features. This process can be described as follows.

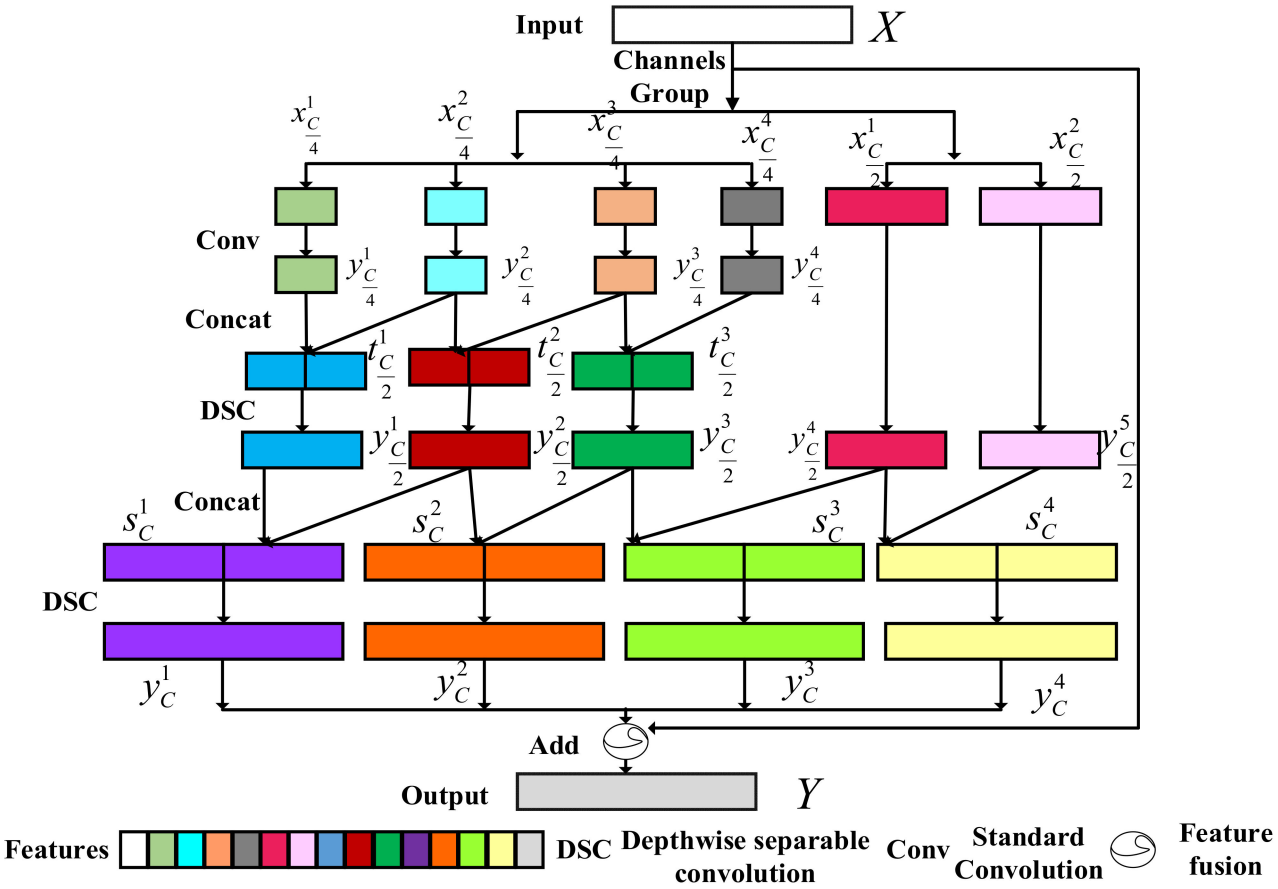


Figure 4. Channel multi-group fusion structure.

Suppose that the input feature is $X = [x_1, x_2, \ldots, x_C] \in \mathbb{R}^{W \times H \times C}$, $x^i_{\frac{C}{4}} \in \mathbb{R}^{W \times H \times \frac{C}{4}}$ represents the i-th feature with the number of channels $\frac{C}{4}$, and $x^i_{\frac{C}{2}} \in \mathbb{R}^{W \times H \times \frac{C}{2}}$ represents the i-th feature with the number of channels $\frac{C}{2}$. After channel grouping, the input features can be represented as $x^1_{\frac{C}{4}} = [x_1, \ldots, x_{\frac{C}{4}}]$, $x^2_{\frac{C}{4}} = [x_{\frac{C}{4}}, \ldots, x_{\frac{C}{2}}]$, $x^3_{\frac{C}{4}} = [x_{\frac{C}{2}}, \ldots, x_{\frac{3C}{4}}]$, $x^4_{\frac{C}{4}} = [x_{\frac{3C}{4}}, \ldots, x_C]$, $x^1_{\frac{C}{2}} = [x_1, \ldots, x_{\frac{C}{2}}]$, $x^2_{\frac{C}{2}} = [x_{\frac{C}{2}}, \ldots, x_C]$. The convolution operation is performed first for the features $x^1_{\frac{C}{4}}$, $x^2_{\frac{C}{4}}$, $x^3_{\frac{C}{4}}$, and $x^4_{\frac{C}{4}}$, where the number of channels is $\frac{C}{4}$, and after convolution the results are $y^1_{\frac{C}{4}} \in \mathbb{R}^{W \times H \times \frac{C}{4}}$, $y^2_{\frac{C}{4}} \in \mathbb{R}^{W \times H \times \frac{C}{4}}$, $y^3_{\frac{C}{4}} \in \mathbb{R}^{W \times H \times \frac{C}{4}}$, and $y^4_{\frac{C}{4}} \in \mathbb{R}^{W \times H \times \frac{C}{4}}$, respectively. Here, $y^i_{\frac{C}{4}}$ can be represented as

$$y^i_{\frac{C}{4}} = f_{conv}(x^i_{\frac{C}{4}}, W) = \mathrm{Re}LU(BN(W \cdot x^i_{\frac{C}{4}})) \qquad i = 1, 2, 3, 4 \tag{12}$$

$y^i_{\frac{C}{4}}$ represents the convolution result of the feature $x^i_{\frac{C}{4}}$, and $y^i_{\frac{C}{4}} = [y^i_{\frac{C}{4}}(1), y^i_{\frac{C}{4}}(2), \ldots, y^i_{\frac{C}{4}}(m), \ldots, y^i_{\frac{C}{4}}(\frac{C}{4})]$. $y^i_{\frac{C}{4}}(m)$ represents the m-th channel of the i-th feature with the number of channels $\frac{C}{4}$, $f_{conv}(\cdot)$ represents the convolution operation, W represents the convolution weight, ReLU represents the activation function, and BN represents batch normalization.

The use of grouping convolution can reduce the requirement of computing power, but it will also lead to the lack of information interaction between group features, which makes the extracted features incomplete. The information interaction is enhanced through channel concatenate of two adjacent features $(y^1_{\frac{C}{4}}, y^2_{\frac{C}{4}})$, $(y^2_{\frac{C}{4}}, y^3_{\frac{C}{4}})$ and $(y^3_{\frac{C}{4}}, y^4_{\frac{C}{4}})$. The number

of feature channels after channel concatenate is $\frac{C}{2}$, and $t^i_{\frac{C}{2}} \in \mathbb{R}^{W\times H\times \frac{C}{2}}$ represents the i-th feature after channel concatenate, the channel concatenate operation of feature ς and feature ϖ is represented by $\sum\sum([\varsigma, \varpi])$, where $t^i_{\frac{C}{2}}$ is calculated as

$$t^1_{\frac{C}{2}} = \sum_{m=1}^{\frac{C}{4}} \sum_{n=1}^{\frac{C}{4}} ([y^1_{\frac{C}{4}}(m), y^2_{\frac{C}{4}}(n)]) \tag{13}$$

$$t^2_{\frac{C}{2}} = \sum_{m=1}^{\frac{C}{4}} \sum_{n=1}^{\frac{C}{4}} ([y^2_{\frac{C}{4}}(m), y^3_{\frac{C}{4}}(n)]) \tag{14}$$

$$t^3_{\frac{C}{2}} = \sum_{m=1}^{\frac{C}{4}} \sum_{n=1}^{\frac{C}{4}} ([y^3_{\frac{C}{4}}(m), y^4_{\frac{C}{4}}(n)]) \tag{15}$$

The features $x^1_{\frac{C}{2}}$, $x^2_{\frac{C}{2}}$, $t^1_{\frac{C}{2}}$, $t^2_{\frac{C}{2}}$, $t^3_{\frac{C}{2}}$ with the number of channel $\frac{C}{2}$ are processed by depthwise separable convolution and the results after convolution are $y^1_{\frac{C}{2}} \in \mathbb{R}^{W\times H\times \frac{C}{2}}$, $y^2_{\frac{C}{2}} \in \mathbb{R}^{W\times H\times \frac{C}{2}}$, $y^3_{\frac{C}{2}} \in \mathbb{R}^{W\times H\times \frac{C}{2}}$, $y^4_{\frac{C}{2}} \in \mathbb{R}^{W\times H\times \frac{C}{2}}$, $y^5_{\frac{C}{2}} \in \mathbb{R}^{W\times H\times \frac{C}{2}}$ respectively. $y^i_{\frac{C}{2}}$ is calculated as

$$y^i_{\frac{C}{2}} = f_{dsc}(x^i_{\frac{C}{2}}, W) = \mathrm{Re}LU(BN(W \cdot x^i_{\frac{C}{2}})) \quad i = 1, 2 \tag{16}$$

$$y^i_{\frac{C}{2}} = f_{dsc}(t^j_{\frac{C}{2}}, W) = \mathrm{Re}LU(BN(W \cdot t^j_{\frac{C}{2}})) \quad i = 3, 4, 5, \quad j = 1, 2, 3 \tag{17}$$

where $y^i_{\frac{C}{2}}$ represents the convolution result of the features $x^i_{\frac{C}{2}}$ and $t^i_{\frac{C}{2}}$, and $y^i_{\frac{C}{2}} = [y^i_{\frac{C}{2}}(1), y^i_{\frac{C}{2}}(2), \ldots, y^i_{\frac{C}{2}}(m), \ldots, y^i_{\frac{C}{2}}(\frac{C}{2})]$, $y^i_{\frac{C}{2}}(m)$ represents the m-th channel of the i-th feature with the number of channels $\frac{C}{2}$, and $f_{dsc}(\cdot)$ represents the depthwise separable convolution operation. Then, the adjacent features $(y^1_{\frac{C}{2}}, y^2_{\frac{C}{2}})$, $(y^2_{\frac{C}{2}}, y^3_{\frac{C}{2}})$, $(y^3_{\frac{C}{2}}, y^4_{\frac{C}{2}})$ and $(y^4_{\frac{C}{2}}, y^5_{\frac{C}{2}})$ are concatenated in the channel dimension. The number of feature channels after concatenate is C, and $s^i_C \in \mathbb{R}^{W\times H\times C}$ represents the i-th feature with the number of channels after concatenate C. The calculation process of s^i_C is

$$s^1_C = \sum_{m=1}^{\frac{C}{2}} \sum_{n=1}^{\frac{C}{2}} ([y^1_{\frac{C}{2}}(m), y^2_{\frac{C}{2}}(n)]) \tag{18}$$

$$s^2_C = \sum_{m=1}^{\frac{C}{2}} \sum_{n=1}^{\frac{C}{2}} ([y^2_{\frac{C}{2}}(m), y^3_{\frac{C}{2}}(n)]) \tag{19}$$

$$s^3_C = \sum_{m=1}^{\frac{C}{2}} \sum_{n=1}^{\frac{C}{2}} ([y^3_{\frac{C}{2}}(m), y^4_{\frac{C}{2}}(n)]) \tag{20}$$

$$s^4_C = \sum_{m=1}^{\frac{C}{2}} \sum_{n=1}^{\frac{C}{2}} ([y^4_{\frac{C}{2}}(m), y^5_{\frac{C}{2}}(n)]) \tag{21}$$

The features s^1_C, s^2_C, s^3_C and s^4_C with the number of channels C are processed by depthwise separable convolution, respectively. The convolution results are y^1_C, y^2_C, y^3_C and y^4_C. The calculation process of y^i_C is

$$y^i_C = f_{dsc}(s^i_C, W) = \mathrm{Re}LU(BN(W \cdot s^i_C)) \quad i = 1, 2, 3, 4 \tag{22}$$

Next, the features y^1_C, y^2_C, y^3_C, y^4_C are fused and the fusion results and input feature X are shortcut to obtain the final output result $Y \in \mathbb{R}^{W\times H\times C}$, where $\odot$ denotes feature fusion.

$$Y = y^1_C \odot y^2_C \odot y^3_C \odot y^4_C \odot X \tag{23}$$

3. Experiment

In this section, the proposed LCNN-CMGF method is evaluated from multiple perspectives using different indicators. The four most commonly used remote sensing scene datasets, UCM21, RSSCN7, AID and NWPU45, are used to carry out a variety of experi-

ments in this paper. The experimental results under the conditions of four datasets and multiple training ratios show that the proposed LCNN-CMGF method has more significant performance advantages than the compared advanced method.

3.1. Dataset Settings

To verify the performance of the proposed LCNN-CMGF method, a series of experiments were performed on four datasets, i.e., UCM21 [21], RSSCN7 [22], AID [23], NWPU45 [24]. In addition to the complex spatial structure, remote sensing scene images also have high intra-class differences and similarities between classes, which make these four datasets very challenging. Details of the four datasets are shown in Table 1, including the number of images per class, the number of scene categories, the total number of images, the spatial resolution of images, and the image size. In addition, we select a scene image from each scene category of the four datasets for display, as shown in Figure 5. Due to the inconsistent size of the image, in order to avoid memory overflow in the training process, the bilinear interpolation method is used to adjust the size of the training image to 256×256.

Table 1. Detailed description of four datasets.

Datasets	The Number of Images per Class	The Number of Scene Categories	The Total Number of Images	The Spatial Resolution of Images (m)	Image Size
UCM21	100	21	2100	0.3	256×256
RSSCN7	400	7	2800	-	400×400
AID	200–400	30	10,000	0.5–0.8	600×600
NWPU45	700	45	31,500	0.2–30	256×256

3.2. Setting of the Experiments

When dividing the dataset, a stratified sampling method is adopted. The stratified sampling can effectively avoid the risk of sampling deviation. In stratified sampling, a random seed is set to ensure that the same images are chosen in each experiment. In addition, in order to improve the reliability of the experimental results, the average value of 10 experimental results is taken as the final result. According to previous work on remote sensing scene image classification, the datasets are divided as follows: the UCM21 [21] dataset is divided into training:test = 8:2, that is, 1680 scene images are used for training, and the remaining 420 scene images are used for testing; the RSSCN7 [22] dataset is divided into training:test = 5:5, that is, 1400 scene images are used for training, and the remaining 1400 scene images are used for testing; the AID30 [23] dataset is divided into training: test = 2:8 and training:test = 5:5, respectively. When training:test = 2:8, 2000 scene images for training and 8000 scene images for testing; When training:test = 5:5, 5000 scene images for training and 5000 scene images for testing; and the NWPU45 [24] dataset is divided into training:test = 1:9 and training:test = 2:8, respectively. When training:test = 1:9, 3150 scene images for training and 28,350 scene images for testing; When training:test = 2:8, 6300 scene images for training and 25,200 scene images for testing. As shown in Table 2, the input and output sizes of each group of features from group 1 to group 8 in the LCNN-CMGF method are listed. Table 3 shows the experimental environment and parameter setting.

3.3. Experimental Result

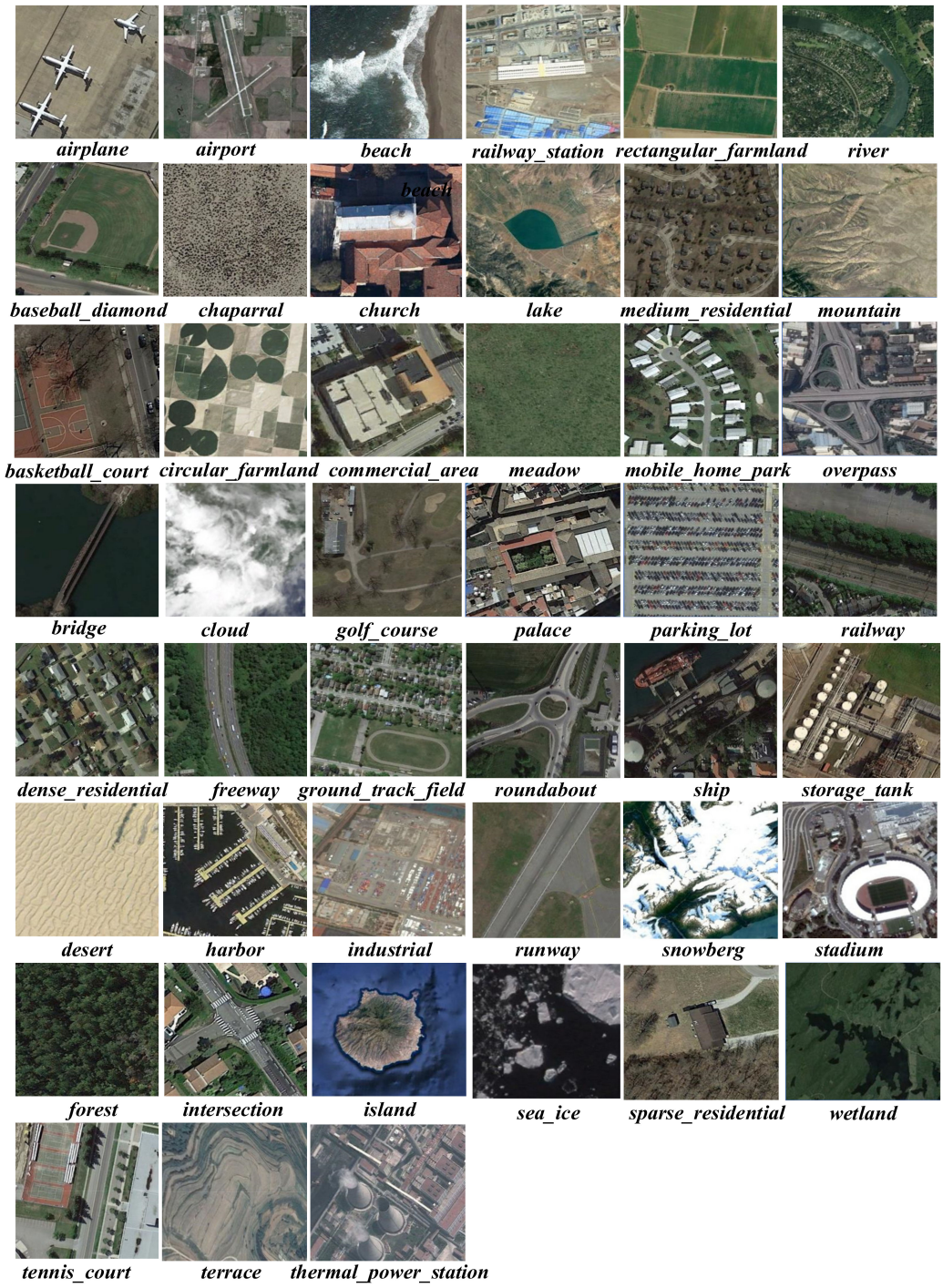
To verify the performance of the proposed method, evaluation indexes such as overall accuracy (OA), kappa coefficient (kappa), confusion matrix, and weighting parameters were used for experimental comparison.

(**a**) Scene images from the UCM21 dataset.

(**b**) Scene images from the RSSCN7 dataset.

(**c**) Scene images from the AID dataset.

Figure 5. *Cont.*



(**d**) Scene images from the NWPU45 dataset.

Figure 5. Different scene images in four datasets. (**a**) Scene images from the UCM21 dataset. (**b**) Scene images from the RSSCN7 dataset. (**c**) Scene images from the AID dataset. (**d**) Scene images from the NWPU45 dataset.

3.3.1. Experimental Results on the UCM21 Dataset

The methods with good classification performance on the UCM21 dataset from 2019 to 2020 are selected for comparison with the proposed LCNN-CMGF method. The experimental results are shown in Table 4. Under the condition that the training proportion of the UCM21 dataset was 80%, the classification accuracy of the proposed LCNN-CMGF method reaches 99.52%, which exceeds all the comparison methods. The proposed LCNN-CMGF

method is 0.6% higher than the Lie group (LiG) with sigmoid kernel [25], 0.55% higher than the Contourlet CNN method [26], and 3.19% higher than the MobileNet method [27]. Table 5 lists the kappa coefficient of the proposed LCNN-CMGF method and the comparison methods. The kappa coefficient of the proposed method is 99.50%, 6.13% higher than EfficientNet [28], and 2.58% higher than Fine tune Mobilenet V2 [29], which proves the effectiveness of our method.

Table 2. Network architecture of the LCNN-CMGF method.

Input Size	Groups	Output Size
256 × 256 × 3	1	128 × 128 × 64
128 × 128 × 64	2	64 × 64 × 128
64 × 64 × 128	3	32 ×3 2 × 128
32 × 32 × 128	4	16 × 16 × 128
16 × 16 × 128	5	8 × 8 × 256
8 × 8 × 256	6	8 × 8 × 256
8 × 8 × 256	7	8 × 8 × 512
8 × 8 × 512	Avgpooling	1 × 1 × 512
1 × 1 × 512	Dense	1 × 1 × 7

Table 3. Experimental environment and parameter setting.

Item	Contents
Processor	AMD Ryzen 7 4800 H with Radeon Graphics@2.90 GHz
Memory	16 G
Operating system	Windows10
Solid state hard disk	512 G
Software	PyCharm Community Edition 2020.3.2
GPU	NVIDIA GeForce RTX2060 6G
Keras	v2.2.5
Initial study rate	0.01
Momentum	0.9

Table 4. OA(%) of eighteen methods and the LCNN-CMGF method at the training ratio of 80% in the UCM21 dataset.

Method	OA(80%)	Year
LiG with Sigmoid Kernel [25]	98.92 ± 0.35	2020
Contourlet CNN [26]	98.97 ± 0.21	2020
MobileNet [27]	96.33 ± 0.15	2020
EfficientNet [28]	94.37 ± 0.14	2020
Fine-Tune MobileNet V2 [29]	98.13 ± 0.33	2019
VGG-16-CapsNet [30]	98.81 ± 0.12	2019
ResNet + WSPM-CRC [31]	97.95	2019
Positional Context Aggregation [32]	99.21 ± 0.18	2020
LCNN-BFF Method [33]	99.29 ± 0.24	2020
DDRL-AM Method [34]	99.05 ± 0.08	2020
Skip-Connected CNN [35]	98.04 ± 0.23	2020
Gated Bidirectiona + Global Feature [36]	98.57 ± 0.48	2020
Feature Aggregation CNN [37]	98.81 ± 0.24	2019
Aggregated Deep Fisher Feature [38]	98.81 ± 0.51	2019
HABFNet [39]	99.29 ± 0.35	2020
EfficientNetB3-Attn-2 [40]	99.21 ± 0.22	2021
VGG_VD16 with SAFF [41]	97.02 ± 0.78	2020
Semi-Supervised Representation Learning [42]	94.05 ± 1.2	2020
Proposed	99.52 ± 0.34	2021

Table 5. Kappa (%) of six methods and the LCNN-CMGF method at the training ratio of 80% in the UCM21 dataset.

Method	OA (80%)	Kappa (%)
LiG with Sigmoid Kernel [25]	98.92 ± 0.35	97.63
Contourlet CNN [26]	98.97 ± 0.21	97.81
MobileNet [27]	96.33 ± 0.15	94.91
EfficientNet [28]	94.37 ± 0.14	93.37
Fine-Tune MobileNet V2 [29]	98.13	96.92
SE-MDPMNet [29]	98.95	97.74
Proposed	99.52 ± 0.34	99.50

As shown in Figure 6, in the UCM21 dataset, except that the classification accuracy of two scenes of 'storagetanks' and 'tenniscourt' is 95%, the classification accuracy of other scenes is 100%. It is proved that this method has good performance on the UCM21 dataset.

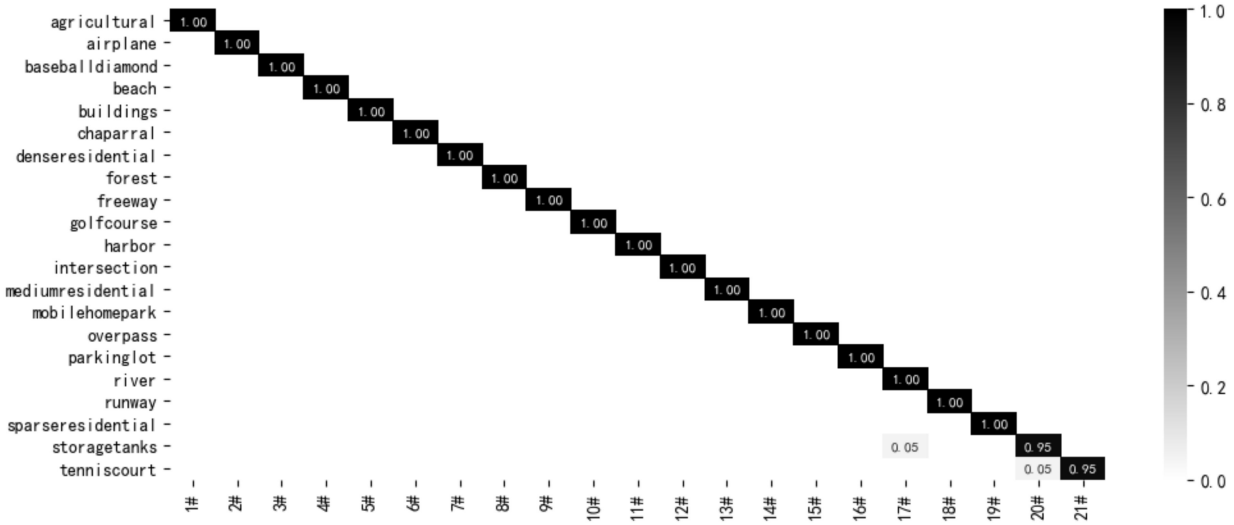


Figure 6. Confusion matrix of the LCNN-CMGF method on the UCM21 dataset (80/20).

3.3.2. Experimental Results on the RSSCN7 Dataset

The comparison of experimental results of the proposed methods and some state-of-the-art methods proposed in the last two years on RSSCN7 datasets are shown in Table 6. The OA of our proposed method is 97.50%, which is 0.85%, 3.9% and 1.52% higher than that of VGG-16-CapsNet [30], WSPM-CRC [31] and the Positional Context Aggregation method [32], respectively. It is proved that our method has better feature representation ability.

Table 6. OA (%) of seven kinds of methods and the LCNN-CMGF method under the training ratios of 50% in the RSSCN7 dataset.

Method	OA (50%)	Year
Contourlet CNN [26]	95.54 ± 0.17	2020
VGG-16-CapsNet [30]	96.65 ± 0.23	2019
SPM-CRC [31]	93.86	2019
WSPM-CRC [31]	93.60	2019
Positional Context Aggregation [32]	95.98 ± 0.56	2020
ADFF [38]	95.21 ± 0.50	2019
LCNN-BFF [33]	94.64 ± 0.12	2020
SE-MDPMNet [29]	92.46 ± 0.66	2019
Variable-Weighted Multi-Fusion [43]	89.1	2019
Proposed	97.50 ± 0.21	2021

The confusion matrix of the proposed LCNN-CMGF method on the RSSCN7 dataset is shown in Figure 7. The proposed method has good classification accuracy on this dataset. The classification accuracy of all scenarios can reach more than 95%, and the classification accuracy of three scenarios, 'Forest', 'RiverLake', and 'Resident', can reach 99%. The classification accuracy of 'Field' scenes is the lowest 95%, and some of them are incorrectly classified into 'Grass' scenes, which is due to the strong class similarity between 'Grass' and 'Field' scenes.

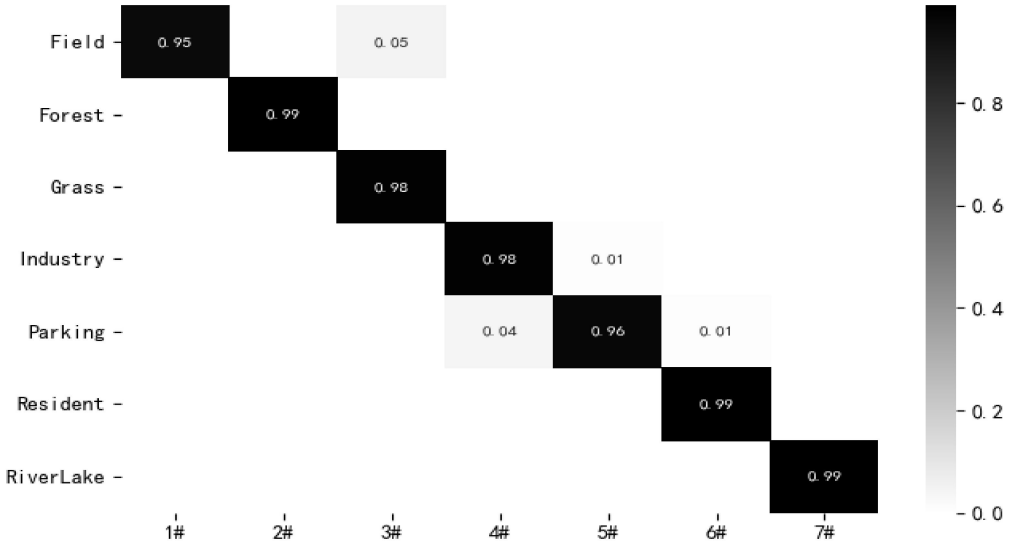


Figure 7. Confusion matrix of the LCNN-CMGF method on the RSSCN7 dataset (50/50).

3.3.3. Experimental Results on the AID Dataset

Some excellent CNN-based methods on the AID dataset from 2018 to 2020 are selected for comparison with the proposed method. The experimental results are shown in Table 7. Under the condition that the training proportion of the AID dataset was 20%, the classification accuracy of the proposed LCNN-CMGF method is 93.63%, which is 1.57% higher than that of LCNN-BFF [33], 2.07% higher than that of DDRL-AM Method [34], 2.53% higher than that of Skip-Connected CNN [35], and 1.43% higher than that of GBNet + Global Feature [36]. Under the condition that the training proportion of the AID dataset was 50%, the OA of the proposed method is reaching 97.54%, which is 2.09% higher than that of Feature Aggregation CNN [37], 2.28% higher than that of Aggregated Deep Fisher Feature [38], 0.79% higher than that of HABFNet [39], 2.15% higher than that of EfficientNetB3-Attn-2 [40], and 1.56% higher than that of VGG_VD16 with SAFF Method [41]. The experimental results show that the proposed method is very effective. For remote sensing scene images with rich image variation, high similarity between classes and strong intra-class differences, the proposed method can capture more representative features. As shown in Table 8, the Kappa coefficient of this method is 97.45% when the training proportion is 50%, which is 1.95% higher than that of Semi-Supervised Representation Learning [42], 1.33% higher than that of Variable-Weighted Multi-Fusion [43], 3.49% higher than that of TSDFF [44], and 3.19% higher than that of Discriminative+AlexNet [45]. The Kappa coefficient results demonstrate that the predicted and actual results of the proposed method are more consistent.

Table 7. Kappa (%) of fourteen methods and the LCNN-CMGF method at the training ratio of 50% in the AID dataset.

Method	OA (20/80)(%)	OA (50/50)(%)	Year
MobileNet [27]	88.53 ± 0.17	90.91 ± 0.18	2020
EfficientNet [28]	86.56 ± 0.17	88.35 ± 0.16	2020
Feature Aggregation CNN [37]	-	95.45 ± 0.11	2019
TSDFF [44]	-	94.65	2018
Discriminative + AlexNet [45]	85.62 ± 0.10	94.47 ± 0.12	2018
InceptionV3 [46]	93.27 ± 0.17	95.07 ± 0.22	2020
Bidirectional Adaptive Feature Fusion [47]	-	93.56	2019
MG-CAP(Bilinear) [48]	92.11 ± 0.15	95.14 ± 0.12	2020
LCNN-BFF [33]	92.06 ± 0.36	94.53 ± 0.24	2020
DDRL-AM Method [34]	91.56 ± 0.49	94.08 ± 0.35	2020
GBNet [36]	90.16 ± 0.24	93.72 ± 0.34	2020
GBNet + Global Feature [36]	92.20 ± 0.23	95.48 ± 0.12	2020
Aggregated Deep Fisher Feature [38]	92.78 ± 0.57	95.26 ± 0.84	2019
HABFNet [39]	93.01 ± 0.43	96.75 ± 0.52	2020
EfficientNetB3-Attn-2 [40]	92.48 ± 0.76	95.39 ± 0.43	2021
VGG_VD16 with SAFF Method [41]	92.05 ± 0.34	95.98 ± 0.70	2020
ResNet50 [46]	92.39 ± 0.15	94.69 ± 0.19	2020
VGG19 [46]	87.73 ± 0.25	91.71 ± 0.24	2020
Skip-Connected CNN [35]	91.10 ± 0.15	93.30 ± 0.13	2020
Proposed	93.63 ± 0.10	97.54 ± 0.25	2021

Table 8. Kappa (%) of seven methods and the LCNN-CMGF method at the training ratios of 50% in the AID dataset.

Method	OA (50%)	Kappa (%)
Two-Stage Deep Feature Fusion [44]	94.65	93.41
MobileNet [27]	90.91 ± 0.18	89.53
EfficientNet [28]	88.35 ± 0.16	87.21
Semi-Supervised Representation Learning [42]	95.63	95.50
TSDFF [44]	94.65	93.96
Discriminative + AlexNet [45]	94.47 ± 0.12	94.26
Variable-Weighted Multi-Fusion [43]	96.23 ± 0.35	96.12
Two-Stream Deep Fusion Framework [49]	94.58	93.34
InceptionV3 [46]	95.07 ± 0.22	94.83
ResNet50 [46]	94.69 ± 0.19	93.47
VGG19 [46]	91.71 ± 0.24	90.06
Proposed	97.54 ± 0.25	97.45

The confusion matrix of the LCNN-CMGF method on the 50/50 AID dataset is shown in Figure 8. The classification accuracy of all scenes has reached more than 90%, among which the classification accuracy of 'Meadow', 'Viaduct' and 'Sparse Residential' has reached 100%. In the training proportion of 50%, the classification accuracy of 'School' scenes is the lowest, which is 93%. Some school scenes are incorrectly classified into three scenes: 'Industrial', 'Church' and 'Commercial'. The reason is that there are similar buildings among the four scenes of 'School', 'Industrial', 'Church' and 'Commercial'. The high inter-class similarity leads to the low classification accuracy of the 'School' scene. For all that, the proposed method still achieved better classification performance compared with the previously proposed advanced method.

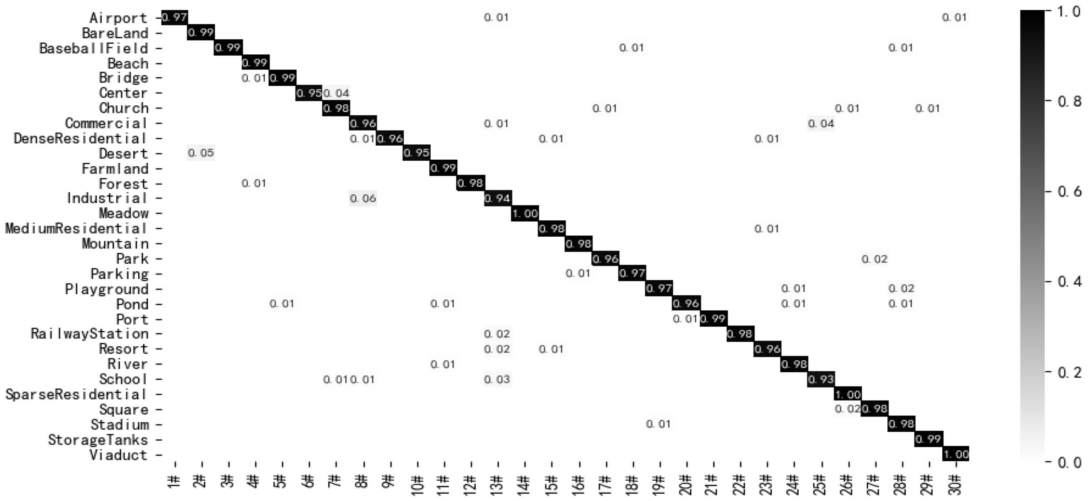


Figure 8. Confusion matrix of the LCNN-CMGF method on the AID (50/50) dataset.

3.3.4. Experimental Results on the NWPU45 Dataset

Similar to the AID dataset, some excellent neural networks on the NWPU45 dataset from 2018 to 2020 are selected for experimental comparison. The experimental results are shown in Table 9. When training:test = 1:9, the OA of the proposed method reached 92.53%, which is 6% higher than that of the LCNN-BFF method [33], 8.15% higher than that of VGG_VD16 with the SAFF method [41], 3.31% higher than that of Discriminative + VGG16 [45], 11.19% higher than that of VGG19 [46] and 0.97% higher than that of MSDFF [50], respectively. When training:test = 2:8, the OA of the proposed method is 6.6% and 2.45% higher than that of Contourlet CNN [26] and the LCNN-BFF method [33], respectively. Meanwhile, the OA of the proposed method is 8.2% higher than that of Skip-Connected CNN [35] and 3.31% higher than that of Discriminative + VGG16 [45]. This indicates that the proposed method performs better on both training ratios on NWPU45 datasets. Under the condition that the training proportion is 20% in the NWPU45 dataset, the kappa coefficient contrast results for the proposed LCNN-CMGF method and the contrast method are shown in Table 10. The Kappa value of this method is the highest among all comparison methods, reaching 94.04%. It is 1.12%, 5.69% and 10.72% higher than that of LiG with sigmoid kernel [25], Contourlet CNN [26] and MobileNet [27], respectively.

On the NWPU45 dataset, when training:test = 2:8, the confusion matrix obtained by the proposed LCNN-GWHA method is shown in Figure 9. Because the NWPU45 dataset contains rich content variations and a complex spatial structure, there are no fully recognized scenes when classifying the dataset. However, the classification accuracy of the proposed method for 43 scenes reached more than 90%. The lowest classification accuracy was for 'palace' and 'church', which were 87% and 88%, respectively. The main reason is that the two scenarios, 'palace' and 'church', have similar architectural styles and are easy to confuse when extracting features, resulting in classification errors.

Table 9. OA (%) of seventeen methods and the LCNN-CMGF method at the training ratios of 20% and 10% in the NWPU45 dataset.

Network Model	OA (10/90)(%)	OA (20/80)(%)	Year
Contourlet CNN [26]	85.93 ± 0.51	89.57 ± 0.45	2020
MG-CAP with Biliner [48]	89.42 ± 0.19	91.72 ± 0.16	2020
EfficientNet [28]	78.57 ± 0.15	81.83 ± 0.15	2020
LiG with RBF Kernel [51]	90.23 ± 0.13	93.25 ± 0.12	2020
LiG with Sigmoid Kernel [25]	90.19 ± 0.11	93.21 ± 0.12	2020
VGG19 [46]	81.34 ± 0.32	83.57 ± 0.37	2020
ResNet50 [46]	86.23 ± 0.41	88.93 ± 0.12	2020
InceptionV3 [46]	85.46 ± 0.33	87.75 ± 0.43	2020
MobileNet [27]	80.32 ± 0.16	83.26 ± 0.17	2020
Discriminative + VGG16 [45]	89.22 ± 0.50	91.89 ± 0.22	2018
Discriminative + AlexNet [45]	85.56 ± 0.20	87.24 ± 0.12	2018
Skip-Connected CNN [35]	84.33 ± 0.19	87.30 ± 0.23	2020
LCNN-BFF Method [33]	86.53 ± 0.15	91.73 ± 0.17	2020
VGG-16-CapsNet [30]	85.05 ± 0.13	89.18 ± 0.14	2019
VGG_VD16 with SAFF Method [41]	84.38 ± 0.19	87.86 ± 0.14	2020
MSDFF [50]	91.56	93.55	2020
R.D [52]	-	91.03	2019
Proposed	92.53 ± 0.56	94.18 ± 0.35	2021

Table 10. Kappa (%) of ten methods and the LCNN-CMGF method at the training ratio of 20% in the NWPU45 dataset.

Network Model	OA (20%)	Kappa (%)
Contourlet CNN [26]	89.57 ± 0.45	88.35
EfficientNet [28]	81.83 ± 0.15	79.53
LiG with RBF Kernel [51]	93.25 ± 0.12	93.02
LiG with Sigmoid Kernel [25]	93.21	92.92
VGG19 [46]	83.57	82.17
ResNet50 [46]	88.93	87.61
InceptionV3 [46]	87.75	86.46
MobileNet [27]	83.26	81.72
Fine-Tune MobileNet V2 [29]	93.00	92.93
LCNN-BFF Method [33]	91.73	91.54
Proposed	94.18 ± 0.35	94.04

3.4. Comparison of the Computational Complexity of Models

In addition, to further demonstrate the advantages of the proposed methods in terms of speed, MobileNetV2 [12], CaffeNet [23], VGG-VD-16 [23], GoogleNet [23], Contourlet CNN [26], SE-MDPMNet [29], Inception V3 [46], ResNet50 [46], LiG with RBF kernel [51], and LGRIN [53] were used for comparison with the proposed LCNN-CMGF method. Some experiments were carried out on the AID dataset. The size of Giga Multiply-Accumulation operations per second (GMACs) was used as the evaluation index in the experiments. The GMACs measures the computational complexity of a model. The comparison of experimental results of these methods on the AID dataset with training:test = 5:5 is shown in Table 11. As shown in Table 11, the OA of the proposed LCNN-CMGF method is 97.54%, the parameter quantity is 0.8 M, and the GMACs value is 0.0160 G. Compared with other lightweight models LiG with RBF kernel [51] and MobleNetV2 [12], the proposed method achieves higher classification accuracy with less than half of the parameters of the two methods. Although the accuracy is slightly lower than that of LGRIN [53], the number of parameters is 3.83 M less than that of LGRIN [53], and the GMACs value is 0.4773 G less than that of LGRIN [53]. The proposed LCNN-CMGF method achieves a good trade-off between model complexity and classification accuracy.

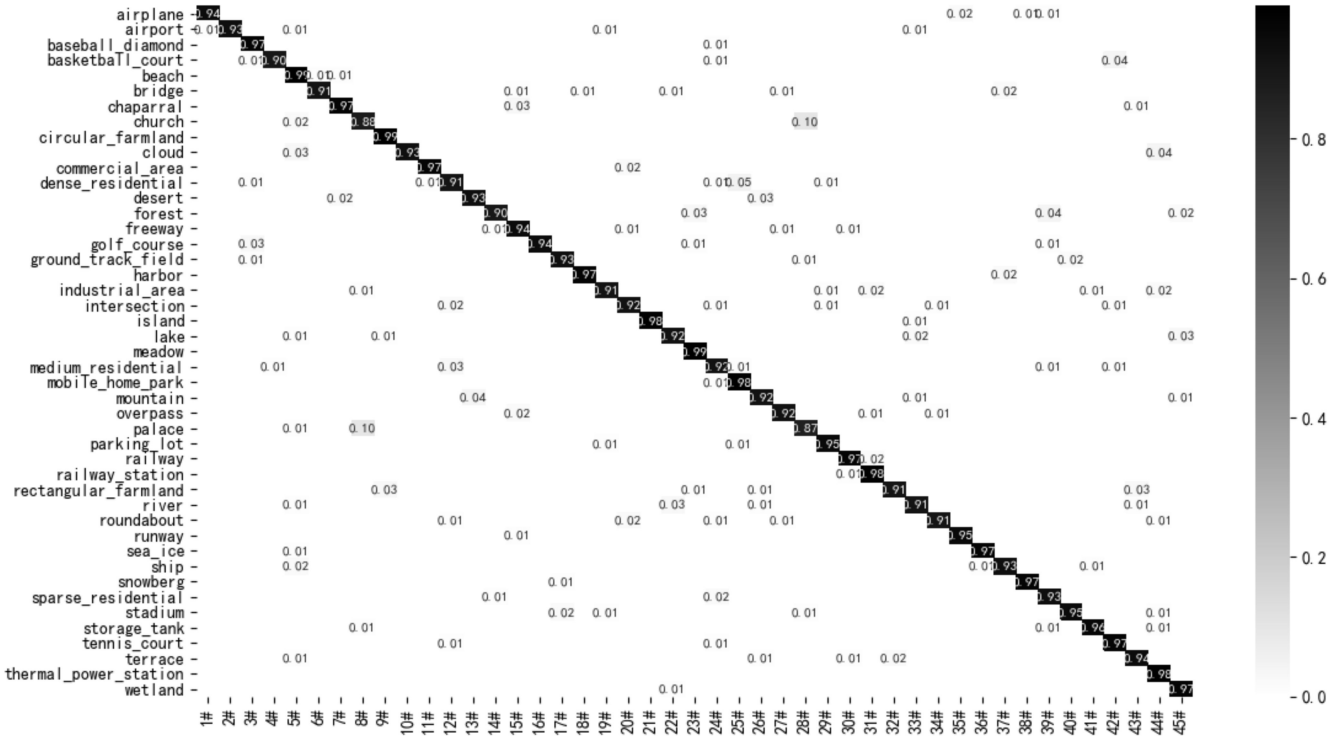


Figure 9. Confusion matrix of the LCNN-CMGF method on the NWPU45 (20/80) dataset.

Table 11. Evaluation values of ten methods and the LCNN-CMGF method at the training ratio of 50% in the AID dataset.

Method	OA (%)	Parameters	GMACs
Contourlet CNN [26]	89.57	12.6 M	1.0583 G
SE-MDPMNet [29]	97.14	5.17 M	0.9843 G
LiG with RBF Kernel [51]	96.19	2.07 M	0.2351 G
InceptionV3 [46]	95.07	45.37 M	2.4356 G
ResNet50 [46]	94.69	25.61 M	1.8555 G
MobileNetV2 [12]	95.96	3.5 M	0.3451 G
VGG-VD-16 [23]	89.64	138.36 M	7.7500 G
CaffeNet [23]	89.53	60.97 M	3.6532 G
GoogleNet [23]	86.39	7 M	0.7500 G
LGRIN [53]	97.65	4.63 M	0.4933 G
Proposed	97.54	0.8 M	0.0160 G

3.5. Comparison Results of Shallow Feature Extraction Modules

Convolutional neural network first extracts the shallow feature of images. With the deepening of the network, the extracted features are more abstract and contain more semantic content, as shown in Figure 10. Figure 10a is the original remote sensing scene image. After the feature extraction of convolutional neural network, the shallow feature map is shown in Figure 10b. With the deepening of the network, more complex features are extracted, as shown in Figure 10c. Compared with the features in Figure 10b, the features in Figure 10c are more complex and have more semantic information. For the classification of remote sensing images, both shallow and deep features are very useful. The traditional methods for extracting shallow features of images are shown in Figure 2a,b. The two methods are not sufficient to extract the shallow features of the image, and some information will be lost during the feature extraction process. Therefore, we propose a three-branch downsampling structure to extract the shallow features of images. The three-branch downsampling structure has great performance advantages compared with the traditional

convolution method. In order to prove the effectiveness of the proposed three-branch downsampling structure, the feature extraction capabilities of the two traditional downsampling structures (as shown in Figure 2a,b) and the proposed three-branch downsampling structure (as shown in Figure 3) are experimentally compared. The comparison process is as follows. In the first experiment, the downsampling structure in Figure 2a is used to replace the three-branch downsampling structure in group 1 and group 2 ofthe proposed method, represented by method 1. In the second experiment, the downsampling structure in Figure 2b is used to replace the three-branch downsampling structure in group 1 and group 2 of the proposed method, represented by method 2. In the third experiment, the three-branch downsampling structure is preserved, which was represented by method 3. The specific experimental parameter settings are shown in Section 2.2. Some comparative experiments were conducted on the AID dataset with training:test = 5:5. For a fair comparison, the three experiments were carried out under the same experimental conditions. The experimental results are listed in Table 12. As shown in Table 12, the three methods have no obvious difference in parameter quantity and model complexity. However, compared with method 3, the other two methods have lower classification accuracy. Specifically, the classification accuracy of method 3 is 1.08% and 1.57% higher than that of method 1 and method 2, respectively.

Figure 10. Visualization results of feature maps. (**a**) The original remote sensing scene image. (**b**) Visualization results of shallow features. (**c**) Visualization results of deep features.

Table 12. Comparison results of feature extraction ability of three methods.

Method	OA(%)	Parameters	GMACs
method 1	96.46 ± 0.36	0.79 M	0.0156 G
method 2	95.97 ± 0.16	0.77 M	0.0152 G
method 3	97.54 ± 0.25	0.8 M	0.0160 G

3.6. Ablation Experiment

In this section, the effectiveness of the number of channels of each group in a channel multi-group fusion structure on network performance is analyzed. Firstly, the group with 2/C channels of the designed multi-grouping fusion structure is removed, and then the structure diagram with channel number C/4 is obtained, which is shown in Figure 11a. Secondly, the group with C/4 channels of the designed multi-grouping fusion structure is removed, and then the structure diagram with channel number C/2 is obtained, which is shown in Figure 11b. Finally, the complete multi-group fusion structure is used for comparison. Some comparative experiments were conducted on the AID dataset with a training proportion of 50%. OA, parameters and GMACs were adopted as evaluation indexes in

the experiment. In the three experiments, to make a fair comparison, the experimental equipment and experimental parameter settings are all the same. The experimental results are listed in Table 13. As shown in Table 13, when the grouping structure with the number of channels C/4 is adopted, as shown in Figure 11a, the OA is 96.09%, the parameter quantity is 0.57 M, and the GMACs value is 0.0153 G. When the grouping structure with the number of channels C/2 is adopted, as shown in Figure 11b, the OA is 95.20%, the parameter quantity is 0.53 M, and the GMACs value is 0.0149 G. The two structures are similar in parameter quantity and model complexity, but the OA of the grouping structure with channel number C/4 is higher than that of the grouping structure with channel number C/2 because it increases the diversity of features. However, there is still a large gap in classification performance between the two methods and the proposed multi-group fusion structure, which further proves the effectiveness of the proposed method.

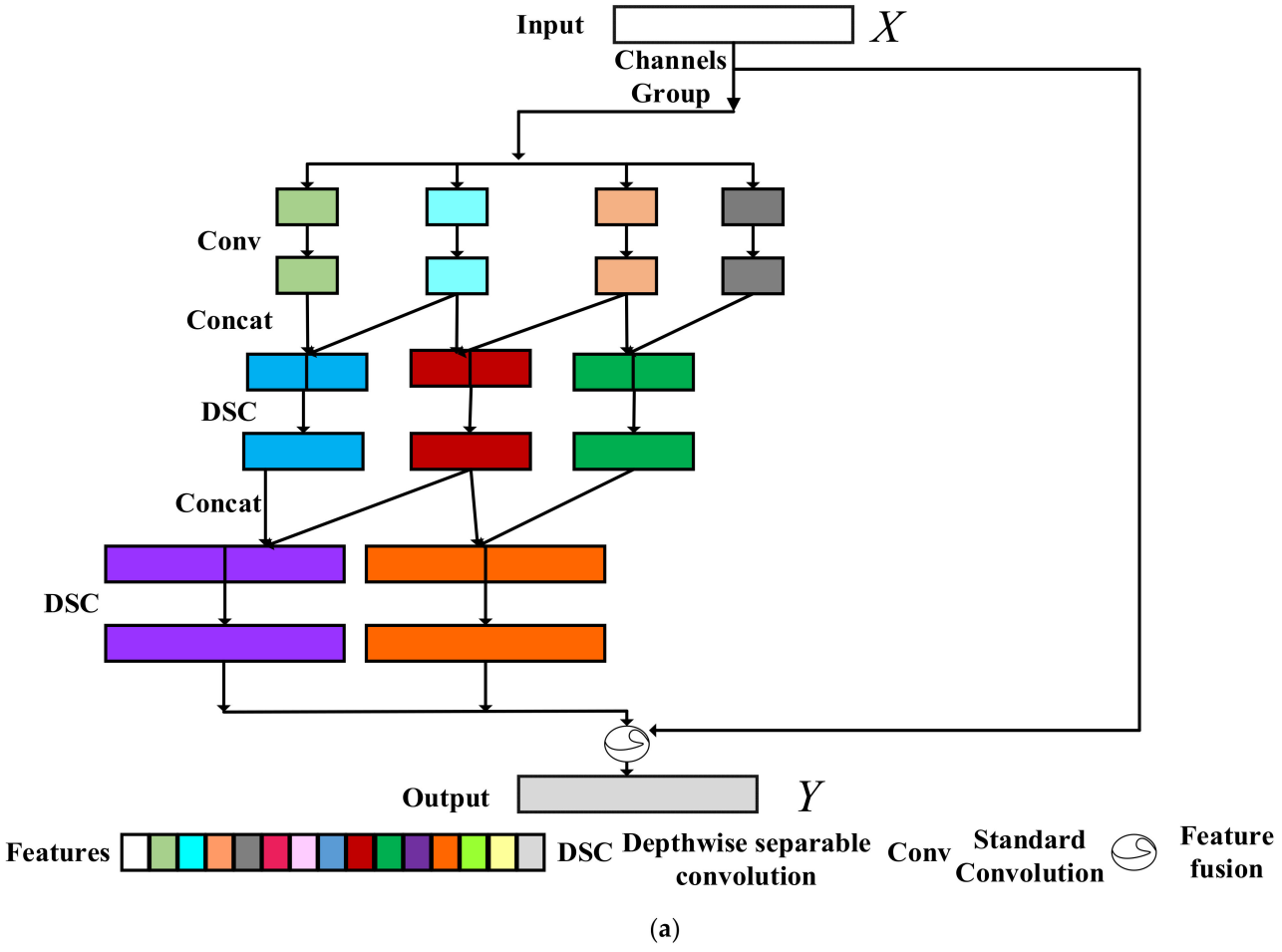


Figure 11. *Cont.*

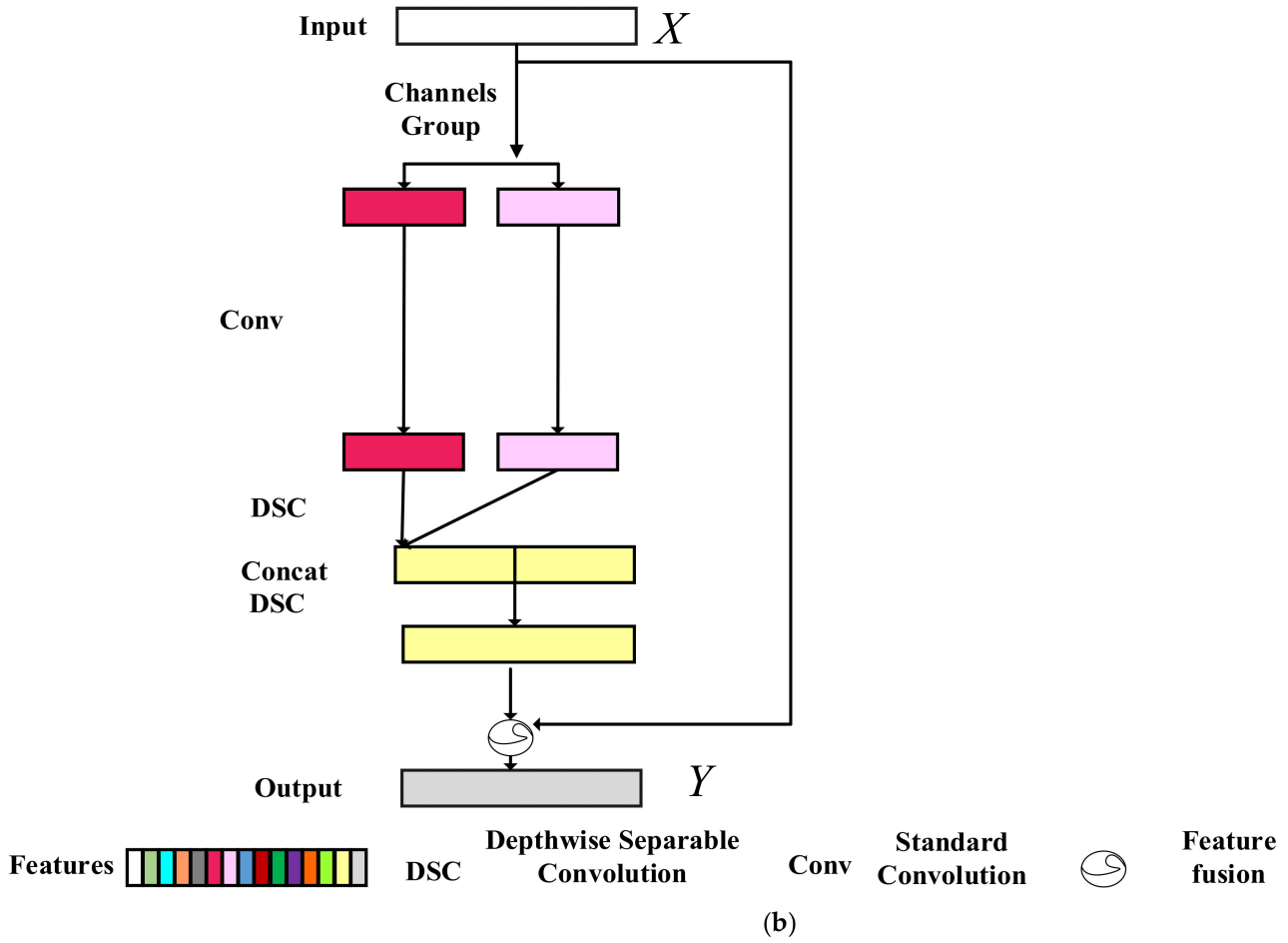


Figure 11. The proposed structure with different groups. (**a**) Structure diagram with channel number C/4. (**b**) Structure diagram with channel number C/2.

Table 13. Comparison results of the proposed structure with different groupings on network performance.

Group	OA(%)	Parameters	GMACs
4/C	96.09 ± 0.61	0.57 M	0.0153 G
2/C	95.20 ± 0.15	0.53 M	0.0149 G
2/C + 4/C	97.54 ± 0.25	0.8 M	0.0160 G

4. Discussions

In order to display the feature extraction ability of the proposed method more intuitively, a series of visualization methods are adopted to evaluate the proposed method. Firstly, the feature extraction ability of the proposed method is presented by using the visualization method of Class Activation Map (CAM). The CAM method displays important areas of the image predicted by the model by generating a rough attention map from the last layer of the convolutional neural network. Some images in the UCM21 dataset are chosen for visualization experiments, and the visualization results are shown in Figure 12. As shown in Figure 12, the proposed LCNN-CMGF method can highlight semantic objects corresponding to real categories. This shows that the proposed LCNN-CMGF method has a strong ability to locate and recognize objects. In addition, the proposed LCNN-CMGF method can better cover semantic objects and has a wide highlight range.

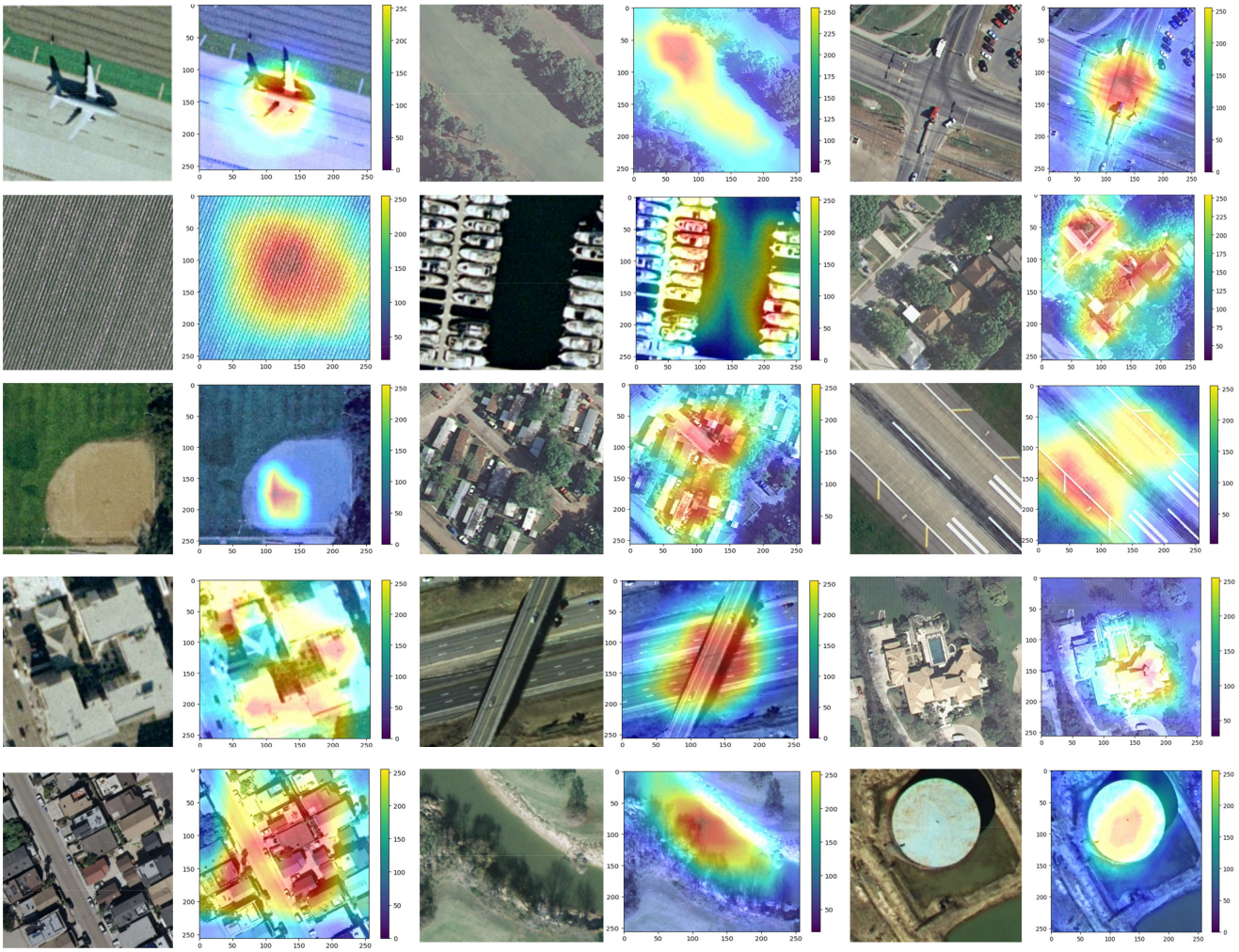
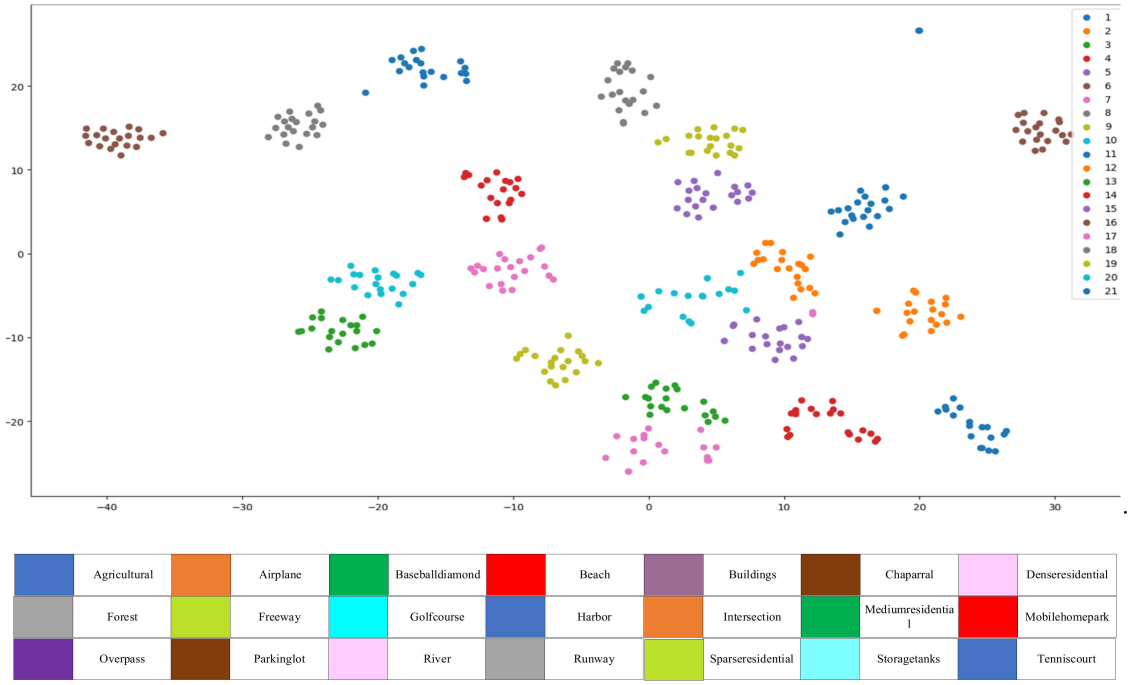


Figure 12. Cam visualization results of the LCNN-CMGF method on the UCM21 dataset.

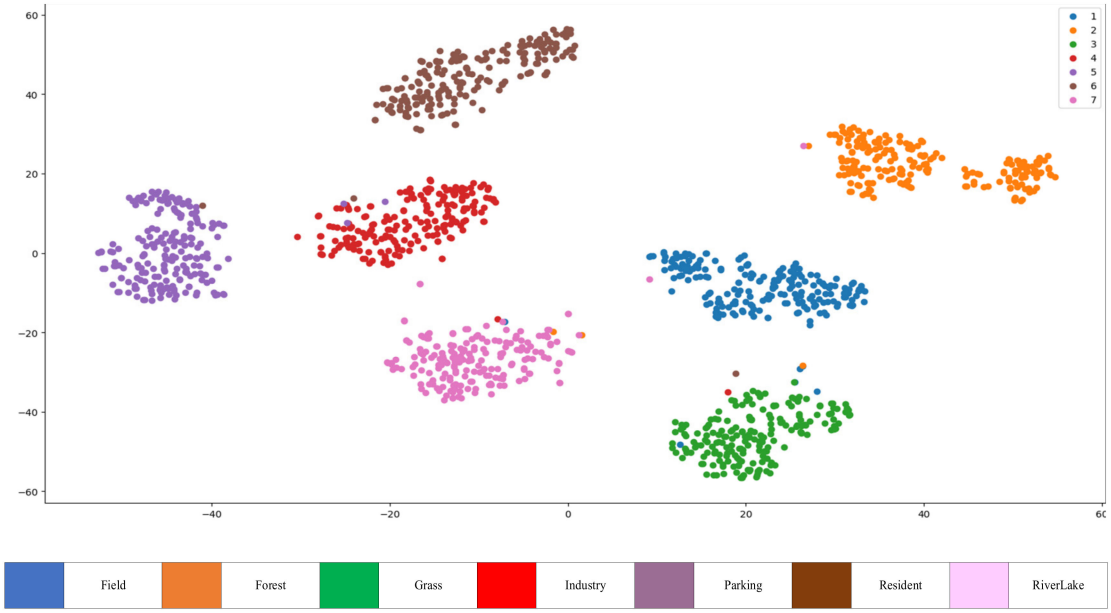
Next, t-distributed stochastic neighbor embedding visualization (T-SNE) method is used to visualize the proposed LCNN-CMGF method and further evaluate the performance of the proposed LCNN-CMGF method. T-SNE is a nonlinear dimensionality reduction algorithm, which usually maps high dimensions to two-dimensional or three-dimensional space for visualization, which can well evaluate the classification effect of the model. The RSSCN7 and UCM21 datasets are used for visualization experiments, and the experimental results are shown in Figure 13.

As shown in Figure 13, there is no confusion between single semantic clusters on the UCM21 dataset and the RSSCN7 dataset, which means that the proposed LCNN-CMGF method has better global feature representation, increases the separability and relative distance between single semantic clusters, and can more accurately extract the features of remote sensing scene images and improve the classification accuracy.

In addition, a randomized prediction experiment is performed on the UCM21 dataset using the LCNN-CMGF method. The results are shown in Figure 14. From Figure 14, we can see that the LCNN-CMGF method has more than 99% confidence in remote sensing image prediction, and some of the predictions even reach 100%. This further proves the validity of the proposed method for remote sensing scene image classification.



(**a**)



(**b**)

Figure 13. T-SNE visual analysis results. (**a**) The T-SNE visualization results of the proposed method on UCM21 datasets. (**b**) The T-SNE visualization results of the proposed method on RSSCN7 datasets.

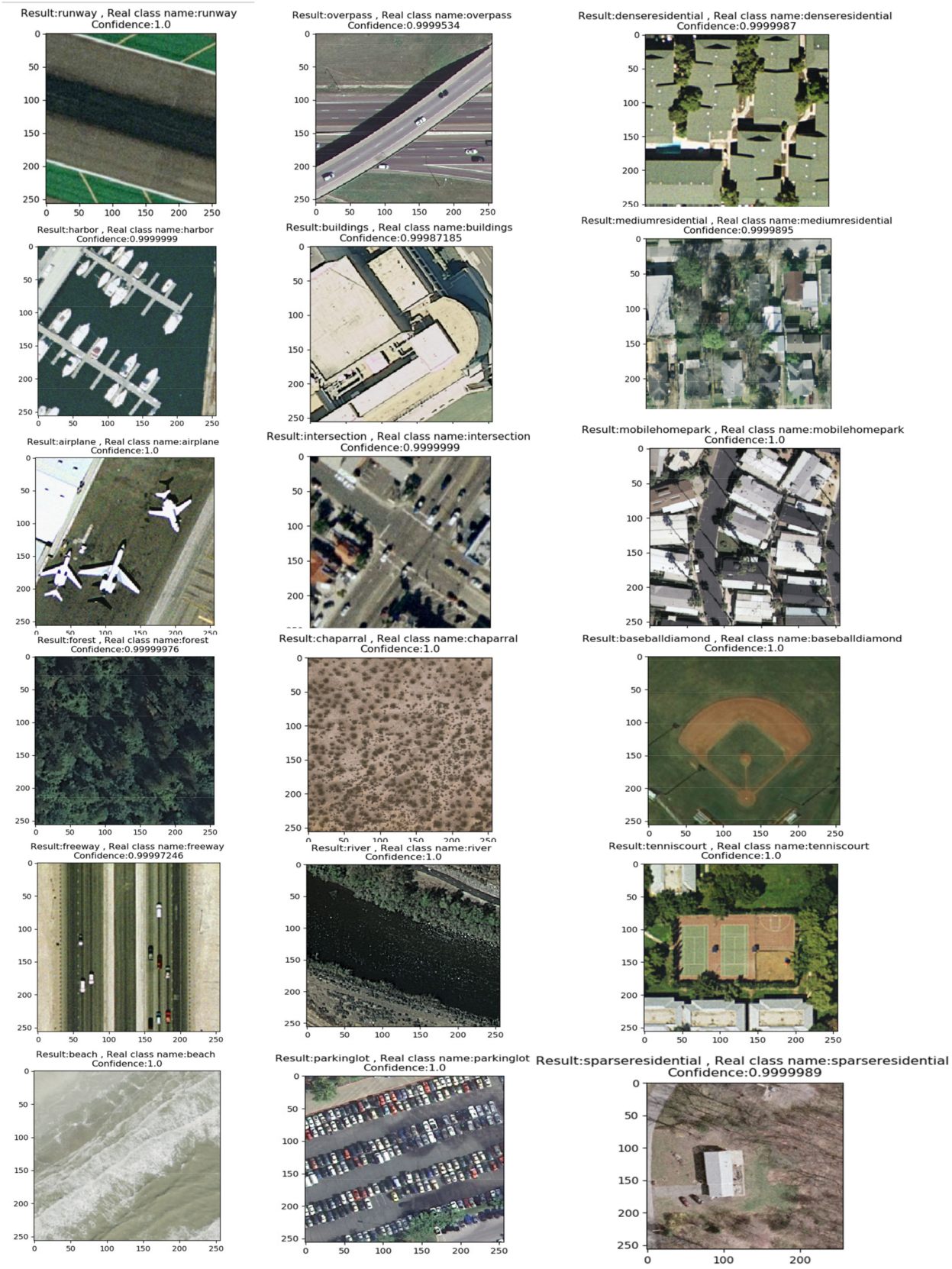


Figure 14. Random classification prediction results.

5. Conclusions

In this paper, a lightweight convolutional neural network based on channel multi-group fusion (LCNN-CMGF) was proposed to classify remote sensing scene images. In the proposed LCNN-CMGF method, the three-branch downsampling structure was designed to extract shallow features from remote sensing images. Channel multi-group fusion structure was presented to efficiently extract deep and abstract features of remote sensing images. The channel multi-group fusion structure utilizes channel fusion of adjacent features to reduce the lack of information exchange between groups caused by group convolution. The experimental results show that the proposed LCNN-CMGF method can achieve higher classification accuracy with fewer parameters and computational complexity than some state-of-the-art methods. Especially on UCM21 datasets, the OA value of this method is as high as 99.52%, which surpasses most of the existing advanced methods. Future work is to find a more effective convolution method to reduce the loss of feature information as much as possible while preserving the lightweight of the network.

Author Contributions: Conceptualization, C.S.; data curation, C.S. and X.Z.; formal analysis, L.W.; methodology, C.S.; software, X.Z.; validation, C.S. and X.Z.; writing—original draft, X.Z.; writing—review and editing, C.S. and L.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Natural Science Foundation of China (41701479, 62071084), in part by the Heilongjiang Science Foundation Project of China under Grant LH2021D022, and in part by the Fundamental Research Funds in Heilongjiang Provincial Universities of China under Grant 135509136.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data associated with this research are available online. The UC Merced dataset is available for download at http://weegee.vision.ucmerced.edu/datasets/landuse.html. RSCCN dataset is available for download at https://sites.google.com/site/qinzoucn/documents. NWPU dataset is available for download at http://www.escience.cn/people/JunweiHan/NWPU-RESISC45.html. AID dataset is available for download at https://captain-whu.github.io/AID/ (accessed on 15 December 2021).

Acknowledgments: We would like to thank the handling editor and the anonymous reviewers for their careful reading and helpful remarks.

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

References

1. Jaiswal, R.K.; Saxena, R.; Mukherjee, S. Application of remote sensing technology for land use/land cover change analysis. *J. Indian Soc. Remote Sens.* **1999**, *27*, 123–128. [CrossRef]
2. Chova, L.G.; Tuia, D.; Moser, G.; Valls, G.C. Multimodal classification of remote sensing images: A review and future directions. *IEEE Proc.* **2015**, *103*, 1560–1584. [CrossRef]
3. Cheng, G.; Zhou, P.; Han, J. Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 7405–7415. [CrossRef]
4. Zhang, L.; Zhang, L.; Du, B. Deep learning for remote sensing data: A technical tutorial on the state-of-the-art. *IEEE Geosci. Remote Sens. Mag.* **2016**, *4*, 22–40. [CrossRef]
5. Feng, J.; Li, D.; Gu, J.; Cao, X.; Shang, R.; Zhang, X.; Jiao, L. Deep Reinforcement Learning for Semisupervised Hyperspectral Band Selection. *IEEE Trans. Geosci. Remote Sens.* **2021**, 1–19. [CrossRef]
6. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
7. Li, Y.; Wang, Q.; Liang, X.; Jiao, L. A Novel Deep Feature Fusion Network for Remote Sensing Scene Classification. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 5484–5487.
8. Zhao, B.; Zhong, Y.; Xia, G.-S.; Zhang, L. Dirichlet-Derived Multiple Topic Scene Classification Model for High Spatial Resolution Remote Sensing Imagery. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 2108–2123. [CrossRef]

9. Wang, Q.; Liu, S.; Chanussot, J.; Li, X. Scene Classification with Recurrent Attention of VHR Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 1155–1167. [CrossRef]
10. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. Squeezenet: Alexnet-level accuracy with 50× fewer parameters and <1 mb model size. *arXiv* **2016**, arXiv:1602.07360.
11. 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.
12. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L. MobileNetV2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520. [CrossRef]
13. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
14. Li, Y.; Jin, X.; Mei, J.; Lian, X.; Yang, L.; Xie, C.; Yu, Q.; Zhou, Y.; Bai, S.; Yuille, A.L. Neural Architecture Search for Lightweight Non-Local Networks. In Proceedings of the CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 10294–10303. [CrossRef]
15. Howard, A.; Sandler, M.; Chen, B.; Wang, W.; Chen, L.-C.; Tan, M.; Chu, G.; Vasudevan, V.; Zhu, Y.; Pang, R.; et al. Searching for MobileNetV3. In Proceedings of the International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 1314–1324. [CrossRef]
16. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity Mappings in Deep Residual Networks. In Proceedings of the 14th European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 630–645. [CrossRef]
17. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1800–1807. [CrossRef]
18. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
19. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
20. Xie, S.N.; Girshick, R.; Dollar, P.; Tu, Z.W.; He, K.M. Aggregated Residual Transformations for Deep Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5987–5995. [CrossRef]
21. Yang, Y.; Newsam, S. Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 3–5 November 2010; pp. 270–279. [CrossRef]
22. Zou, Q.; Ni, L.; Zhang, T.; Wang, Q. Deep Learning Based Feature Selection for Remote Sensing Scene Classification. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 2321–2325. [CrossRef]
23. Xia, G.-S.; Hu, J.; Hu, F.; Shi, B.; Bai, X.; Zhong, Y.; Zhang, L.; Lu, X. AID: A Benchmark Data Set for Performance Evaluation of Aerial Scene Classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3965–3981. [CrossRef]
24. Cheng, G.; Han, J.; Lu, X. Remote Sensing Image Scene Classification: Benchmark and State of the Art. *Proc. IEEE.* **2017**, *105*, 1865–1883. [CrossRef]
25. Xu, C.; Zhu, G.; Shu, J. Robust Joint Representation of Intrinsic Mean and Kernel Function of Lie Group for Remote Sensing Scene Classification. *IEEE Geosci. Remote Sens. Lett.* **2021**, *18*, 796–800. [CrossRef]
26. Liu, M.; Jiao, L.; Liu, X.; Li, L.; Liu, F.; Yang, S. C-CNN: Contourlet Convolutional Neural Networks. *IEEE Trans. Neural Networks Learn. Syst.* **2021**, *32*, 2636–2649. [CrossRef] [PubMed]
27. Pan, H.; Pang, Z.; Wang, Y.; Wang, Y.; Chen, L. A New Image Recognition and Classification Method Combining Transfer Learning Algorithm and MobileNet Model for Welding Defects. *IEEE Access* **2020**, *8*, 119951–119960. [CrossRef]
28. Pour, A.M.; Seyedarabi, H.; Jahromi, S.H.A.; Javadzadeh, A. Automatic Detection and Monitoring of Diabetic Retinopathy Using Efficient Convolutional Neural Networks and Contrast Limited Adaptive Histogram Equalization. *IEEE Access* **2020**, *8*, 136668–136673. [CrossRef]
29. Zhang, B.; Zhang, Y.; Wang, S. A Lightweight and Discriminative Model for Remote Sensing Scene Classification with Multidilation Pooling Module. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 2636–2653. [CrossRef]
30. Zhang, W.; Tang, P.; Zhao, L. Remote sensing image scene classification using CNN-CapsNet. *Remote Sens.* **2019**, *11*, 494. [CrossRef]
31. Liu, B.-D.; Meng, J.; Xie, W.-Y.; Shao, S.; Li, Y.; Wang, Y. Weighted Spatial Pyramid Matching Collaborative Representation for Remote-Sensing-Image Scene Classification. *Remote Sens.* **2019**, *11*, 518. [CrossRef]
32. Zhang, D.; Li, N.; Ye, Q. Positional Context Aggregation Network for Remote Sensing Scene Classification. *IEEE Geosci. Remote Sens. Lett.* **2019**, *17*, 943–947. [CrossRef]
33. Shi, C.; Wang, T.; Wang, L. Branch Feature Fusion Convolution Network for Remote Sensing Scene Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 5194–5210. [CrossRef]
34. Li, J.; Lin, D.; Wang, Y.; Xu, G.; Zhang, Y.; Ding, C.; Zhou, Y. Deep Discriminative Representation Learning with Attention Map for Scene Classification. *Remote Sens.* **2020**, *12*, 1366. [CrossRef]

35. He, N.; Fang, L.; Li, S.; Plaza, J.; Plaza, A. Skip-Connected Covariance Network for Remote Sensing Scene Classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 1461–1474. [CrossRef] [PubMed]
36. Sun, H.; Li, S.; Zheng, X.; Lu, X. Remote Sensing Scene Classification by Gated Bidirectional Network. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 82–96. [CrossRef]
37. Lu, X.; Sun, H.; Zheng, X. A Feature Aggregation Convolutional Neural Network for Remote Sensing Scene Classification. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 7894–7906. [CrossRef]
38. Li, B.; Su, W.; Wu, H.; Li, R.; Zhang, W.; Qin, W.; Zhang, S. Aggregated Deep Fisher Feature for VHR Remote Sensing Scene Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 3508–3523. [CrossRef]
39. Yu, D.; Guo, H.; Xu, Q.; Lu, J.; Zhao, C.; Lin, Y. Hierarchical Attention and Bilinear Fusion for Remote Sensing Image Scene Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 6372–6383. [CrossRef]
40. Alhichri, H.; Alswayed, A.S.; Bazi, Y.; Ammour, N.; Alajlan, N.A. Classification of Remote Sensing Images Using EfficientNet-B3 CNN Model with Attention. *IEEE Access* **2021**, *9*, 14078–14094. [CrossRef]
41. Cao, R.; Fang, L.; Lu, T.; He, N. Self-Attention-Based Deep Feature Fusion for Remote Sensing Scene Classification. *IEEE Geosci. Remote Sens. Lett.* **2021**, *18*, 43–47. [CrossRef]
42. Yan, P.; He, F.; Yang, Y.; Hu, F. Semi-Supervised Representation Learning for Remote Sensing Image Classification Based on Generative Adversarial Networks. *IEEE Access* **2020**, *8*, 54135–54144. [CrossRef]
43. Zhao, F.; Mu, X.; Yang, Z.; Yi, Z. A novel two-stage scene classification model based on feature variable significance in high-resolution remote sensing. *Geocarto Int.* **2019**, *35*, 1–12. [CrossRef]
44. Liu, Y.; Liu, Y.; Ding, L. Scene classification based on two-stage deep feature fusion. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 183–186. [CrossRef]
45. Cheng, G.; Yang, C.; Yao, X.; Guo, L.; Han, J. When Deep Learning Meets Metric Learning: Remote Sensing Image Scene Classification via Learning Discriminative CNNs. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 2811–2821. [CrossRef]
46. Li, W.; Wang, Z.; Wang, Y.; Wu, J.; Wang, J.; Jia, Y.; Gui, G. Classification of High-Spatial-Resolution Remote Sensing Scenes Method Using Transfer Learning and Deep Convolutional Neural Network. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 1986–1995. [CrossRef]
47. Lu, X.; Ji, W.; Li, X.; Zheng, X. Bidirectional adaptive feature fusion for remote sensing scene classification. *Neurocomputing* **2019**, *328*, 135–146. [CrossRef]
48. Wang, S.; Guan, Y.; Shao, L. Multi-Granularity Canonical Appearance Pooling for Remote Sensing Scene Classification. *IEEE Trans. Image Process.* **2020**, *29*, 5396–5407. [CrossRef] [PubMed]
49. Yu, Y.; Liu, F. A Two-Stream Deep Fusion Framework for High-Resolution Aerial Scene Classification. *Comput. Intell. Neurosci.* **2018**, *2018*, 1–13. [CrossRef]
50. Xue, W.; Dai, X.; Liu, L. Remote Sensing Scene Classification Based on Multi-Structure Deep Features Fusion. *IEEE Access* **2020**, *8*, 28746–28755. [CrossRef]
51. Xu, C.; Zhu, G.; Shu, J. A Lightweight Intrinsic Mean for Remote Sensing Classification with Lie Group Kernel Function. *IEEE Geosci. Remote Sens. Lett.* **2021**, *18*, 1741–1745. [CrossRef]
52. Zhou, Y.; Liu, X.; Zhao, J.; Ma, D.; Yao, R.; Liu, B.; Zheng, Y. Remote sensing scene classification based on rotation-invariant feature learning and joint decision making. *EURASIP J. Image Video Process.* **2019**, *1*, 1–11. [CrossRef]
53. Xu, C.; Zhu, G.; Shu, J. A Lightweight and Robust Lie Group-Convolutional Neural Networks Joint Representation for Remote Sensing Scene Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–15. [CrossRef]