International Journal of
Geo-Information

Article

DFFAN: Dual Function Feature Aggregation Network for Semantic Segmentation of Land Cover

Junqing Huang [1,2], Liguo Weng [1,2,*], Bingyu Chen [1,2] and Min Xia [1,2]

[1] Jiangsu Key Laboratory of Big Data Analysis Technology, Nanjing University of Information Science and Technology, Nanjing 210044, China; hjq@nuist.edu.cn (J.H.); 20191222015@nuist.edu.cn (B.C.); xiamin@nuist.edu.cn (M.X.)

[2] Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology, Nanjing University of Information Science and Technology, Nanjing 210044, China

* Correspondence: 002311@nuist.edu.cn

Abstract: Analyzing land cover using remote sensing images has broad prospects, the precise segmentation of land cover is the key to the application of this technology. Nowadays, the Convolution Neural Network (CNN) is widely used in many image semantic segmentation tasks. However, existing CNN models often exhibit poor generalization ability and low segmentation accuracy when dealing with land cover segmentation tasks. To solve this problem, this paper proposes Dual Function Feature Aggregation Network (DFFAN). This method combines image context information, gathers image spatial information, and extracts and fuses features. DFFAN uses residual neural networks as backbone to obtain different dimensional feature information of remote sensing images through multiple downsamplings. This work designs Affinity Matrix Module (AMM) to obtain the context of each feature map and proposes Boundary Feature Fusion Module (BFF) to fuse the context information and spatial information of an image to determine the location distribution of each image's category. Compared with existing methods, the proposed method is significantly improved in accuracy. Its mean intersection over union (MIoU) on the LandCover dataset reaches 84.81%.

Keywords: land cover; semantic segmentation; convolution neural network

check for updates

Citation: Huang, J.; Weng, L.; Chen, B.; Xia, M. DFFAN: Dual Function Feature Aggregation Network for Semantic Segmentation of Land Cover. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 125. https://doi.org/10.3390/ijgi10030125

Academic Editor: Wolfgang Kainz

Received: 7 December 2020
Accepted: 23 February 2021
Published: 1 March 2021

1. Introduction

Land cover monitoring and evaluation is crucial in land planning and natural resource management. With the advancement of science and technology, we can obtain detailed land use information by analyzing remote sensing data. Land cover information involves many aspects, including but not limited to city planning, water area change, and vegetation coverage. By analyzing land cover data, we can study urbanization rates, forestry and agriculture, and the changes of other natural environments.

Most of the remote sensing data uses multispectral satellite images, but satellite data is low in resolution, typically at 10 m to 30 m [1]. Compared with satellite images, aerial photographs have higher pixel resolution but fewer bands, the common pixel size is usually at 25 cm to 50 cm. High resolution aerial images are very important in detecting buildings, forests, and waters areas. Professional image classification tools are usually used in the detection process [2,3], and GIS programs always provide relevant tools. However, it is time-consuming to use professional image classification tools to detect objects, and when the workload is heavy, errors often occur. In order to solve this problem, related research attempts to use machine learning (ML). In the past ten years, many scholars used machine learning algorithms to analyze remote sensing images and made considerable achievements. For example: Xu et al. [4] employed a nonparametric rule-based classifier, which is based on decision tree learning. They used decision tree regression to estimate the classification ratio of mixed pixels in remote sensing images and compared its classification accuracy with the Maximum Likelihood Classifier and supervised version of fuzzy c-means classifier. Samaniego et al. [5] proposed Modified kNN based on k-nearest neighbor (kNN), the difference between Modified kNN and kNN lies in finding the embedded spaces and their corresponding metrics. In Modified kNN, the basic condition to find an embedding space is that the cumulative variance of a given class label for a given portion of the closest pairs of observations should be minimum. Gislason et al. [6] employed Random Forest (RF) to remote sensing images, the RF classifier uses bagging, or bootstrap aggregating, to form an ensemble of classification. Melgani et al. [7] studied the potentially critical issue of applying binary support vector machine (SVM) to multiclass problems in hyperspectral data. In land cover classification and detection, machine learning showed excellent performance.

In recent years, with the improvement of computer hardware and the increasing demand for image processing in practical work, deep learning (DL) has made great progress in the field of security [8], handwritten digit recognition [9], human action recognition [10], financial trading [11], remote image processing [12–17], and others [18–22]. According to the study of Kussul et al. [23] in processing land cover remote sensing images, deep learning algorithms are significantly better than machine learning algorithms such as the SVM. Convolutional Neural Network (CNN) [24] is a representative algorithm of deep learning, and CNN is a feedforward neural network that includes convolution calculations and deep structures. CNN is widely used in computer vision.

This paper proposes Dual Function Feature Aggregation Network (DFFAN) based on multiple semantic segmentation models. In view of the complex characteristics of the spectral environment in land cover segmentation, DFFAN not only aggregates contextual information but also fuses spatial information of remote sensing images, thus improving the accuracy of semantic segmentation. Experimental comparisons show that DFFAN has better performance, and its mean intersection over union (MIoU) and Kappa are higher than other semantic segmentation networks. We make our code publicly available https://github.com/jqbetter/DFFANet, accessed on 20 December 2020. In general, there are three contributions in this work: (1) An Affinity Matrix Module is proposed to aggregate contextual semantic information. (2) Boundary Feature Fusion is proposed to fuse the boundary information of each feature map. (3) Feature Channels Maximum Element is proposed to strengthen the class location information.

The rest of the paper is organized as follows: Section 2 introduces the related work of CNN in the field of computer vision. Section 3 describes the structure of DFFAN and the function of each module. Section 4 presents the experimental setup and data details. Section 5 summarizes the corresponding work of this article and proposes future research directions.

2. Related Work

Early CNN networks were often used to process image classification tasks, such as the VGG [25] and the ResNet [26]. However, a land cover detection task could not be treated as an image classification task, and the image semantic segmentation method was usually used for the land cover detection problem.

The Fully Convolutional Network (FCN) [27] proposed by Jonathan Long et al. in 2015 was a groundbreaking semantic segmentation algorithm. FCN classified images at the pixel level and solved the problem of semantic segmentation. However, FCN used high-level features of spatial information as the basis for pixel classification, which led to the neglection of low-level features with rich semantic information, and thus resulted in the facts that the FCN did poorly in processing multi-images and its segmentation was very rough. For the shortcomings of the FCN, Ronneberger proposed the U-Net [28]. U-Net network used a U-type structure to strengthen low-level features. RefineNet [29] and SegNet [30] and U-Net used a similar network structure. This network structure used an encoder–decoder structure, the encoder was used to extract the feature information from images, and the decoder restored the extracted features. The decoder will made up the lost information in the encoding process, fused the low-level features, and improved the segmentation accuracy of the network. However, these models were limited by convolution layer structures, which made the aggregation of context information insufficient, resulting in poor prediction results. In order to obtain more accurate segmentation results, the model needed to be able to aggregate the relevance of context information as much as possible. There were two common methods for aggregating context information, they were the global average pooling method based on the pyramid model and the aggregation method based on the attention mechanism. These methods tended to ignore the dependency of context information. The CPNet [31] proposed by Changqian Yu et al. could aggregate spatial information and considered context information, and thus had better prediction performance in multiclass distributed pictures. They designed the Context Prior Layer in CPNet, which was used to aggregate the intracontext and intercontext for each pixel. Meanwhile, the Aggregation Module was designed to aggregate the spatial information for reasoning. However, Context Prior Layer and Aggregation Module in the CPNet increased the amount of calculation geometrically, so that as convolutional layers became deep, gradient would disappear, which affected the accuracy of semantic segmentation.

3. Proposed Method

In this section, the overall structure of DFFAN is described first. Second, the functions of each module in DFFAN is introduced. Finally, the Affinity Matrix Module (AMM), the Boundary Feature Fusion Module (BFF) and the related function modules are introduced in detail.

3.1. Model Overview

In tasks of land cover semantic segmentation, it is necessary to distinguish some confusing categories, and furthermore, some objects which are quite different in their appearances should be put into the same category. Therefore, in the process of semantic segmentation, it is necessary to improve the recognition ability of various types of features in remote sensing images. In DFFAN, the AMM is used to construct the affinity matrix to distinguish the classification of pixels. The affinity matrix supervises the priority mapping of the context and classifies pixels accurately. The AMM improves the accuracy of classification in remote sensing images. The BFF is used to aggregate the spatial information of images, BFF module uses high-dimensional spatial information to guide low-dimensional semantic information, this feature fusion is very efficient. DFFAN adopts a U-shaped structure, and the overall structure is shown in Figure 1. Furthermore, DFFAN is composed of a backbone, a AMM, and a BFF modules. The backbone uses improved ResNet and extracts features of different dimensions through 4×, 8×, 16× and 32× downsamplings, the 32× downsampling layers contains abundant spatial information. Pyramid pooling [32] uses a parallel structure, and it takes into account the characteristics of multiple receptive fields, and thus has a better recognition of the target. The pyramid pooling module can capture the context information, which has a very positive impact on segmentation results.

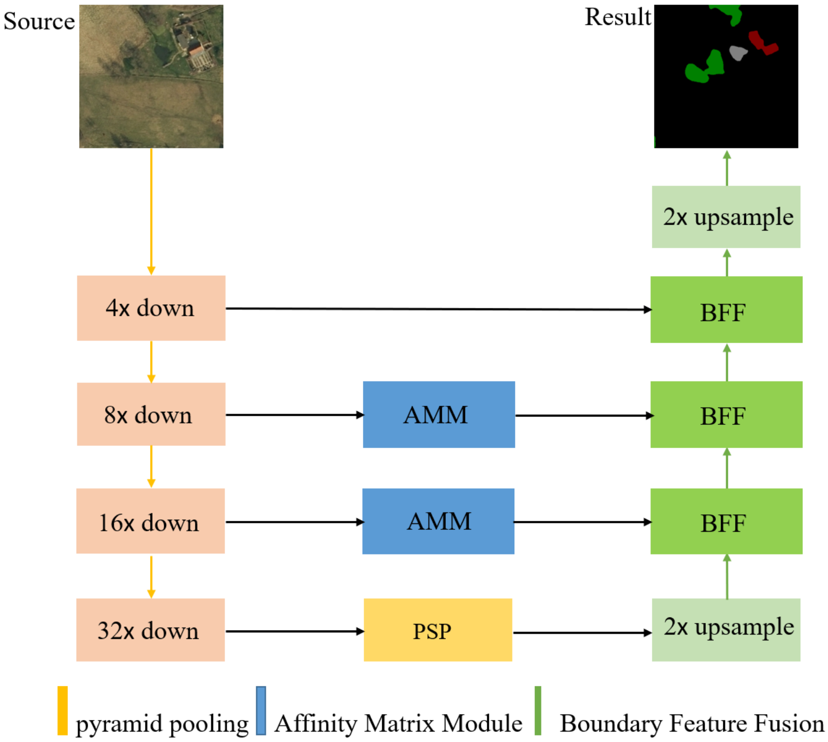


Figure 1. The structure of the Dual Function Feature Aggregation Network (DFFAN).

3.2. *Backbone*

In the process of land cover semantic segmentation, it is very important to extract high-precision feature information of remote sensing images, so it is very important to select a suitable depth convolution neural network for the whole segmentation task. Classical deep convolution neural networks include VGG, ResNet, DenseNet [33], MobileNet [34], and Inception [35]. It is well known that as the convolution layer increases, more feature information could be extracted. However, if there are too many convolution layers in the network, the gradient would disappear and the error would propagate backward. The ResNet solves this problem by using a residual connection structure. Figure 2a shows the original residual block in ResNet, and Figure 2b shows the residual block after the improvement. The modified residual block uses the activation function Mish [36] instead of ReLU.

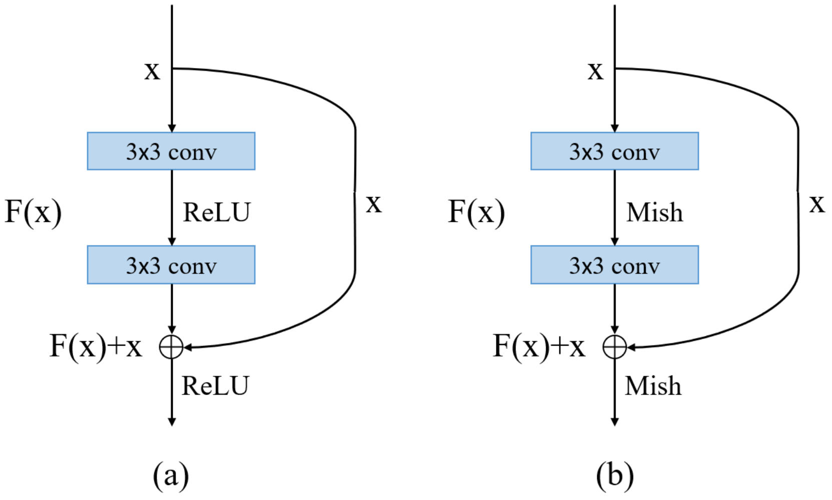


Figure 2. Comparison of two residual blocks: (**a**) original residuals (**b**) modified residuals.

The identity mapping of deep residual networks can effectively solve the problem of training accuracy saturation caused by network layer increasement. In the original residual block, the ReLU [37] is used. The function of ReLU is to activate the weight in the original network, which is described as:

$$f(x) = \max(0, x). \tag{1}$$

We know from Formula (1), convolutional neural network local weights are set to 0 when using the ReLU; therefore, ReLU will affect neuron renewals during reverse propagation. In order to make the network update its data easily, we use Mish instead of ReLU, the mish expression is as follows:

$$f(x) = x \tanh(\ln(1 + e^{x})). \tag{2}$$

The graphs of ReLU and Mish are illustrated in Figure 3, comparing Formula (1) with Formula (2), we can get the following conclusion: when x is positive, Mish and ReLU can reach an infinite value, which can avoid saturation caused by thresholds. However, when x is negative, Relu drives the function value to 0 abruptly, whereas Mish does it more gently, the output of the Mish function is smooth and continuous, and thus it has a better gradient stream. Mish allows the deep neural network to obtain better information, and thus the Mish function has better accuracy and generalization.

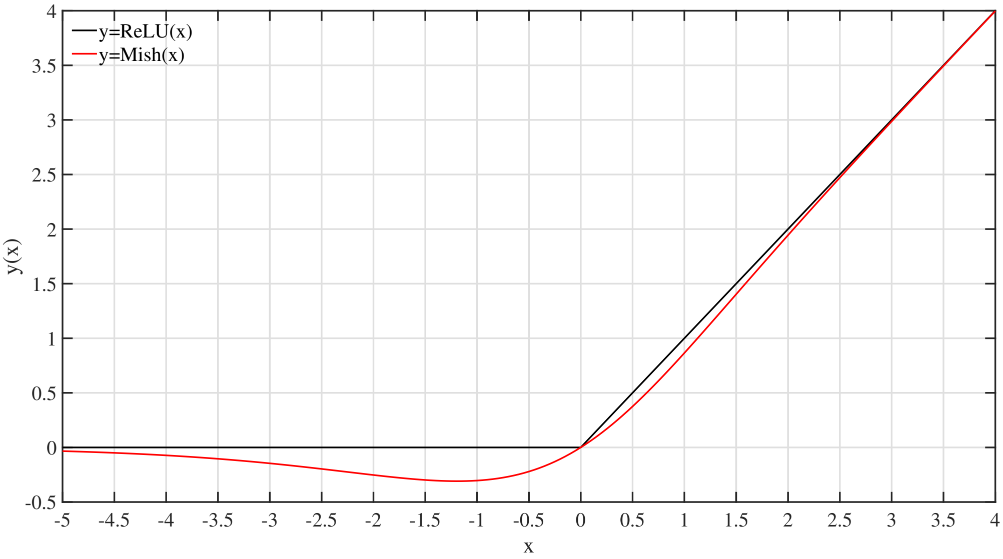


Figure 3. Graph of Mish and ReLU activation functions.

3.3. Affinity Matrix Module

In tasks of remote sensing land cover segmentation, the spectrum and radiance of different samples are different. For example, in building segmentation, isolated buildings need to be identified completely and the features of the subsidiary buildings around large buildings could not to be ignored during segmentation. However, in woodland segmentation, few isolated shrubs should be ignored and edges of woodland need to be segmented accurately to prevent misclassification. Based on the above problems, in tasks of land cover segmentation, we should fully consider relationships between each pixel and its context information. We propose AMM in this paper, this module integrates intraclass and interclass relationships to capture context dependencies within and between classes. The Affinity Matrix Module is shown in Figure 4.

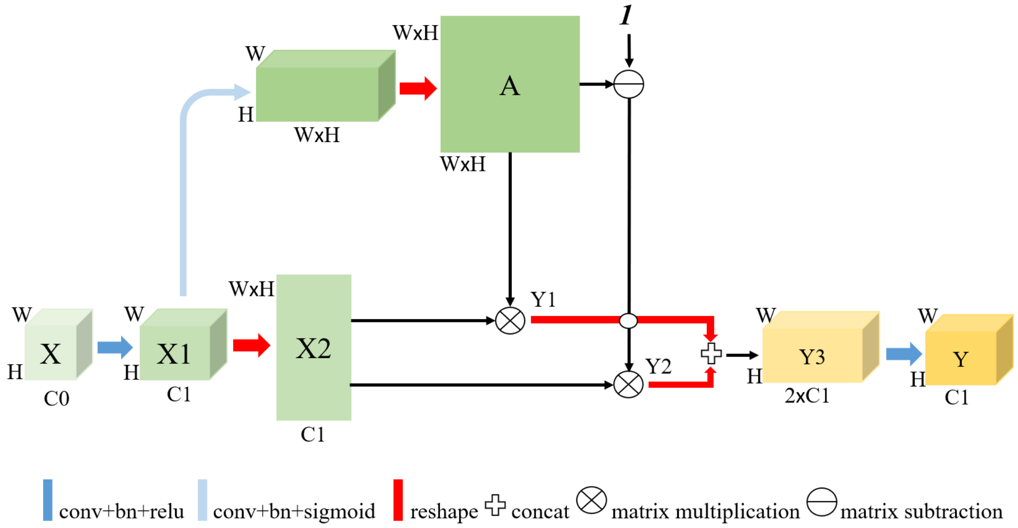


Figure 4. Affinity Matrix Module.

$X \in R^{H \times W \times C0}$ is an input feature matrix, Affinity Matrix Module changes the feature channels of X, we use CNN to adapt X to $X_1 \in R^{H \times W \times C1}$:

$$X_1 = \sigma(bn(f^{k \times k}(X))), \tag{3}$$

where $f^{k \times k}$ represents a convolution with a kernel of $k \times k$ (same as below), k is adjusted according to the scale of X. When the X is small, the calculation cost can be reduced, but when the X is large, it can have a large enough receptive field. bn represents Batch Normalization (same as below), and σ represents ReLU activation function (same as below). The number of feature channels of an input feature graph is increased by Equation (3), X_1 aggregates rich high-dimensional spatial information. In order to construct a priori graph that can indicate the same class of pixels in context, we calculate X_1 as follows:

$$X_2 {=} reshape(X_1), X_2 \in R^{N \times C1}, \tag{4}$$

$$A = reshape(\delta(bn(f^{1 \times 1}(X_1)))), A \in R^{N \times N}, \tag{5}$$

where δ represents Sigmoid activation function, $N = H \times W$. A is the Affinity Matrix, The function of Affinity Matrix is to distinguish whether the pixels belong to the same category or not and to establish relationships between pixels of the same category. We use Sigmoid activation function instead of ReLU activation function. We encode the intraclass pixels and interclass pixels by using A:

$$Y_1 = reshape(X_2 \otimes A), Y_1 \in R^{H \times W \times C1}, \tag{6}$$

$$Y_2 = reshape(X_2 \otimes (1 - A)), Y_2 \in R^{H \times W \times C1}, \tag{7}$$

where $\otimes$ represents matrix multiplication (same as blow), 1 represents an identity matrix. The function of *reshape* is to change the shape of the input feature. Y_1 represents intraclass context information, Y_2 represents interclass context information. To distinguish the context information of each pixel, we concatenate Y_1 and Y_2:

$$Y = \sigma(bn(f^{1 \times 1}(concat(Y_1, Y_2)))), Y \in R^{H \times W \times C1}, \tag{8}$$

where $concat(\cdot, \cdot)$ represents concatenating two maps (same as below). After Equation (8), we calculate Y, the size of Y is $H \times W \times C1$. Y is extracted network features of different dimensions.

3.4. Boundary Feature Fusion Module

At 32-fold downsample, 16-fold downsample, and 8-fold downsample of DFFAN, pyramid pooling and AMM are used to obtain spatial feature information of different dimensions. In order to fuse the information step by step, BFF is designed in this paper, the structure of the BFF module is shown in Figure 5. BFF module contains two function modules feature channels maximum element (FCME) and Information Extraction (IE), which will be described in details later in this section.

BFF has two feature map inputs, X_1 with size $H \times W \times C0$ and X_2 with size $H \times W \times C1$. Pretreatment of X_1 performed at a M_channel and a S_channel, and then the results from the two channels are added together. In the M_channel, X_1 convolutes twice, but the sizes of X_1 does not change, its corresponding result is X_M:

$$X_M = \sigma(bn(f^{3\times3}(\sigma(bn(f^{3\times3}(X_1)))))). \tag{9}$$

In S_channel, FCME is used to extract the largest element and its location at each feature channel in X_1, and the result is X_S. The output Y of BFF is obtained as in Formula (10):

$$Y = upsample(IE(concat(X_2, (X_M + X_S)))), \tag{10}$$

where *upsample* represents upsampling.

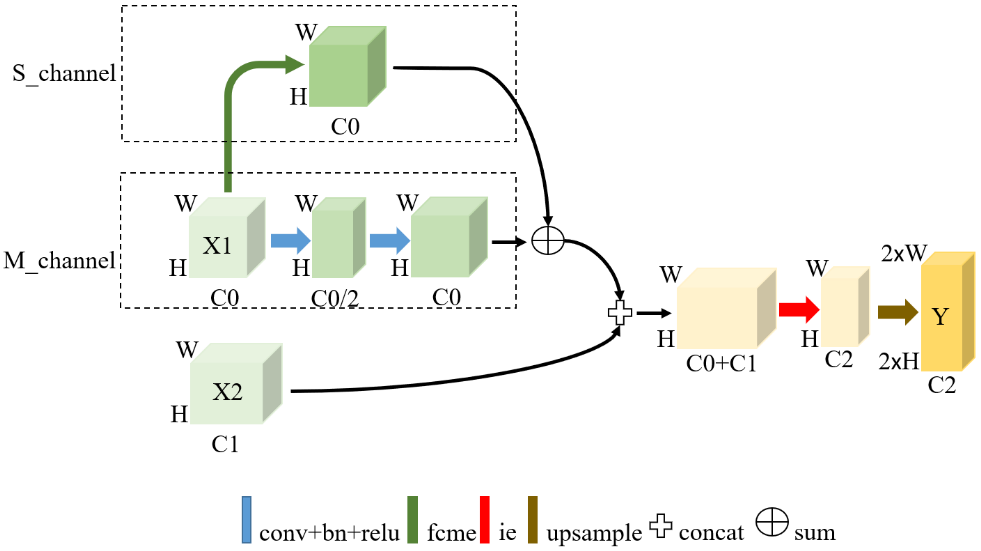
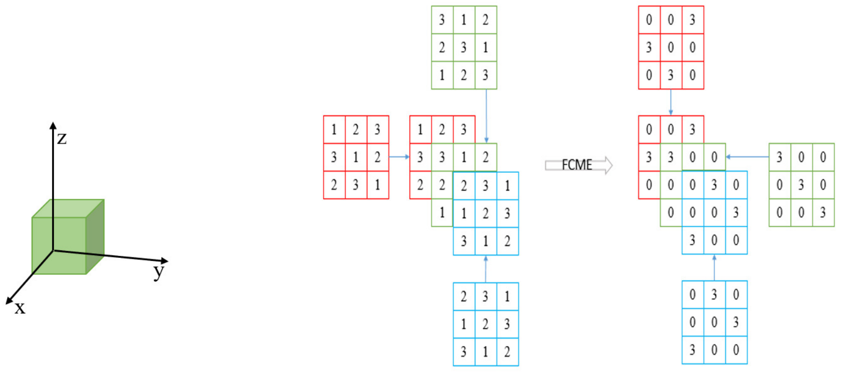


Figure 5. The structure of Boundary Feature Fusion Module (BFF).

Feature channels maximum element (FCME) function module keeps the maximum value at the same position of each channel and clears the remaining values to 0. The FCME and channel attention mechanism (CAM) are different. CAM uses global maximum pooling to extract the maximum value of each feature channel; however, the function of FCME is to compare the values of each feature channel in the input feature map at the same position and to select the maximum value. For a three-dimensional feature map, see Figure 6a. The function of CAM is to save the maximum value in x-y plane, but the function of FCME is to save the maximum value in z axis. The effect of FCME is shown in Figure 6b. The function of FCME is to strengthen position information of each category contained in input feature maps, especially to enhance the edge prediction.



(a) (b)

Figure 6. The Dual Function Feature Aggregation Network (FCME) function diagram. (**a**) The three-dimensional feature map. (**b**) The effect of FCME.

Information Extraction (IE) function module uses separable convolutions to extract local spatial information of input feature maps in the depth dimension. In this way, we can infer the semantic relevance of each element. Meanwhile, IE module uses aggregated high-dimensional spatial information to guide the classification of elements in feature maps. The structure of IE functional module is shown in Figure 7.

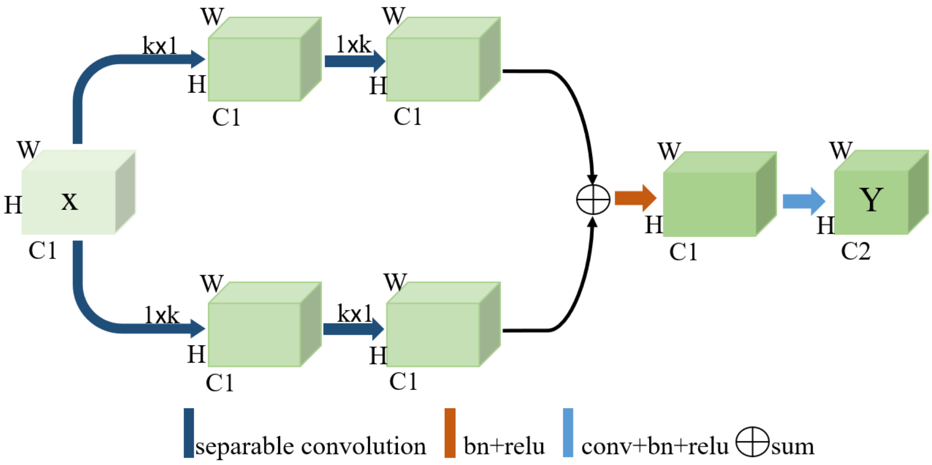


Figure 7. The structure of IE function module.

Input feature map $X \in R^{H \times W \times C1}$, IE uses two groups of asymmetric separable convolutional networks to aggregate spatial information. We mark these two results as y_1 and y_2. The calculation process of IE functional module is as follows:

$$y_1 = f^{1\times k}(f^{k\times 1}(X)), \tag{11}$$

$$y_2 = f^{k\times 1}(f^{1\times k}(X)), \tag{12}$$

$$y_3 = \sigma(bn(y_1 + y_2)), \tag{13}$$

$$Y = \sigma(bn(f^{1\times 1}(y_3))), Y \in R^{H\times W\times C2}. \tag{14}$$

In IE functional module, we use a $k \times 1$ convolution and a $1 \times k$ convolution in two steps instead of a $k \times k$ convolution. In this way, the computation is reduced by half and the receptive field of the original convolution is retained.

4. Experiment and Result Analysis

This chapter compares experimental results of each model; experiments are carried on land cover datasets. The results show that DFFAN is better than other models.

4.1. LandCover Dataset

The land cover aerial dataset published by Boguszewski.[38] was used in the experiment. The landcover dataset collects more than 200 square kilometers of land cover images in Poland. And this dataset has 41 remote sensing images, 8 of them have resolution of 50 cm/pixel, and the remaining 33 have resolution of 25 cm/pixel. We used Python to cut all the images into pixel blocks and got a total of 7938 pictures, each picture is 512 × 512 pixels. According to a ratio of 7:3, 5557 pictures were chosen as the training set and 2381 pictures as the validation set. The RGB values of dataset labels are shown in Table 1, and the cropped pictures and their labels are shown in Figure 8.

The dataset has three classes, including woodland, water, and building. More explicitly, Woodland includes neither single trees nor very small shrubs that are not connected. Ditches and dry riverbeds are excluded from water. In the water segmentation task, different remote sensing images have different spectral and radiometric. Building classification and water classification are similar, but attention should also be paid to the subordinate buildings of the main building and small unobvious buildings. In woodland segmentation task, a few shrubs are often classified as woodlands because the pixel values of trees in the same picture are often not very different and thus misclassification could happen.

Table 1. The RGB values of dataset labels.

	R	**G**	**B**
Void	0	0	0
Building	128	0	0
Woodland	0	128	0
Water	0	0	128

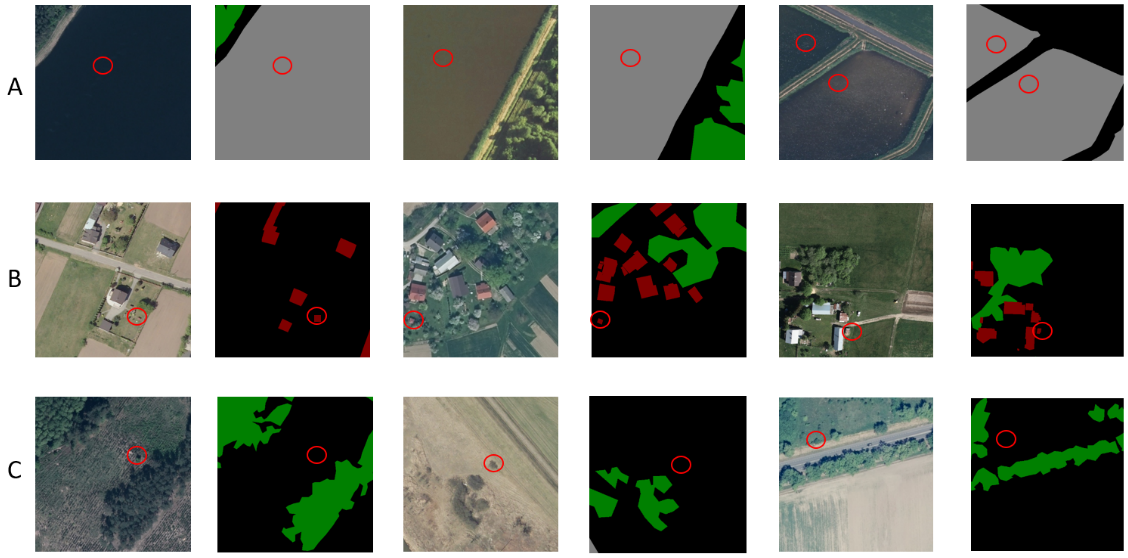


Figure 8. Image and label example from land cover. In (**A**) Line, the red circle area is the water, in different regions, even in the same region, but the remote sensing shooting angle is different, the spectral of water area will be different. In (**B**) Line, the red circle area is the subsidiary buildings of the main building and small buildings with weak characteristics. In (**C**) Line, the red circle area indicates one shrub or a few shrubs, which are not marked on the label.

4.2. Evaluation Metric

In this experiment, we selected four evaluation metrics: including mean pixels accuracy (MPA), mean intersection over union (MIoU), frequency weighted intersection over union (FWIoU), and Kappa. They are as follows:

$$MPA = \frac{1}{k+1} \sum_{i=0}^{k} \frac{P_{ii}}{\sum_{j=0}^{k} P_{ij}}, \tag{15}$$

$$MIoU = \frac{1}{k+1} \sum_{i=0}^{k} \frac{P_{ii}}{\sum_{j=0}^{k} P_{ij} + \sum_{j=0}^{k} P_{ji} - P_{ii}}, \tag{16}$$

$$FWIoU = \frac{1}{\sum_{i=0}^{k} \sum_{j=0}^{k} P_{ij}} \sum_{i=0}^{k} \frac{P_{ii}}{\sum_{j=0}^{k} P_{ij} + \sum_{j=0}^{k} P_{ji} - P_{ii}}, \tag{17}$$

$$Pa = \frac{\sum_{i=0}^{k} P_{ii}}{\sum_{i=0}^{k} \sum_{j=0}^{k} P_{ij}}, \tag{18}$$

$$Pe = \frac{\sum_{i=0}^{k} (\sum_{j=0}^{k} P_{ij}) \times (\sum_{j=0}^{k} P_{ji})}{(\sum_{i=0}^{k} \sum_{j=0}^{k} P_{ij}) \times (\sum_{i=0}^{k} \sum_{j=0}^{k} P_{ij})}, \tag{19}$$

$$Kappa = \frac{Pa - Pe}{1 - Pe}, \tag{20}$$

where k represents number of categories. P_{ij} is a pixel whose correct label is i but its prediction result is j. If the correct label is i, when $i \neq j$, P_{ii} is true positive, P_{ij} is false negative, P_{ji} is false positive and P_{jj} is true negative.

In this evaluation metric, MPA, $MIoU$, and $FWIoU$ are common evaluation metric in semantic segmentation, and $Kappa$ measures classification accuracy. $Kappa$ is different from Pa (pixels accuracy). Pa can directly reflect the proportion of correctly classified pixels, and is easy to calculate. However, if the number of samples of different categories is unbalanced in the data set, the prediction results of the model tend to prefer the categories with more samples and ignore the categories with less samples. For example, if the pixels of a certain category reaches 90% of the total and even if all prediction drops into this category, the Pa would be as high as 0.9. Because the pixels of woodland or water could cover very large proportions in cropped images, Pa cannot objectively reflect the accuracy of segmentation. We need an index that can punish the bias to replace Pa, in Formula (19) and Formula (20), it is noted that the more unbalanced the sample distribution, the higher the Pe would be and the lower the $Kappa$ would be, in consideration of that, we choose $Kappa$ as the evaluation index.

4.3. Experiment Setting and Training

In this paper, we chose FCN, LEDNet, PSPNet, BiSeNet, DeepLabv3+, and UNet as comparison models. During the training phase, we selected the SGD optimizer and set the batch size as 4; we adjusted the learning rate in each iteration, using the formula:

$$new\_lr = lr \times (1 - \frac{iter}{total\_iter})^{0.9}, \tag{21}$$

where lr is the initial learning rate, new_lr is the new learning rate, $iter$ represents the $iter$ iteration. $total\_iter$ is the total number of iterations. All models were trained for 300 epochs with a batch size of 4. All experiments were carried out on Ubuntu16.04 LTS with a Intel(R) Core(TM)i5-9400F CPU @2.90 GHz, 16 G of memory (RAM), and a NVIDIA GeForce RTX 2060S (8 GB). Python 3.6 was used and the experiments were based on the pytorch 1.0.1 programming framework with CUDA10.1 and cudnn7.6.5. We used the cross-entropy loss function to calculate the loss of a neural network, as shown in Formula (22):

$$loss = \sum_{i=1}^{n} p(x_i)\log(p(x_i)) - \sum_{i=1}^{n} p(x_i)\log(q(x_i)), \tag{22}$$

where x_i is the sample; $p(\cdot)$ and $q(\cdot)$ are two separate probability distributions of random variables, n is the number of samples. The training process used a gradient descent algorithm; by comparing labels and predictions, the parameters were updated continuously by using back propagation. All optimal parameters of training models were saved.

For data sets with less data, cross-validation technique or repeat the split into training/testing images several times and report the average performance with a standard deviation can be used to improve the performance of the model. However, the dataset in this paper contains a large amount of data, and the number of pictures in the randomly selected test set can cover various possible situations. Therefore, we use simple split validation to assess the quality of the proposed method.

4.4. Result Analysis

In order to compare the performance of each model, the models were tested under the same conditions. The evaluation metrics of each model are illustrated in Table 2, and the prediction results are shown in Figure 9. Because the FCN and LEDNet prediction images are not ideal, there is no figure of them. It can be seen that the evaluation indexes of other models are higher than those of FCN and LEDNet; all metrics of DFFAN are better than the other models. On the other hand, the *MIoU*, *FWIoU*, and *Kappa* of UNet are higher than those of PSPNet and DeepLabv3+. The *MPA* of DeepLabv3+ is higher than that of UNet and PSPNet, but the *Kappa* of DeepLabv3+ is lower than that of UNet and PSPNet. This explanation would be that the excessive aggregation of spatial information and the neglection of the overfitting caused by context information make the prediction results of DeepLabv3+ on images with unbalanced positive and negative sample distribution biased.

Table 2. Four evaluation metrics of models.

	MPA	*FWIoU*	*MIoU*	*Kappa*
FCN	0.8461	0.8032	0.7289	0.7865
LEDNet	0.843	0.7764	0.721	0.7613
BiSeNet	0.8757	0.8613	0.8028	0.8534
PSPNet	0.8907	0.8767	0.8284	0.8714
DeepLabv3+	0.8938	0.8705	0.8337	0.8641
UNet	0.8788	0.8814	0.836	0.8755
DFFAN	0.9064	0.8921	0.8481	0.8872

It can be seen from Figure 9 that there were some observations in the prediction images. First, when the distribution of different categories of the target to be identified is scattered and the distribution of the same category is centralized, the prediction results of each model are roughly consistent with their labels. It is obvious from the prediction images of line A. Second, classifications of subsidiary buildings and the main building, which were marked by red circles in line B, were not completely consistent with the original image. On the other hand, a few shrubs in the image were also classified as woodland, it was marked by a red circle in line C(d). Finally, misclassification also occurred in areas where multiple classifications intersected. As shown in the yellow circles in line B(d), the blue circle in line B(e) and the yellow circle in line C(c). In summary, each model in this experiment had some shortcomings in land cover segmentation, but the prediction of DFFAN network was the best, as shown in Figure 9 column (g). This is because that the AMM in DFFAN can monitor the context information of each feature graph and guide its classification. The FCME function module in BFF has the function of saving spatial information of feature graphs and enhancing the ability to predict the edges of various categories.

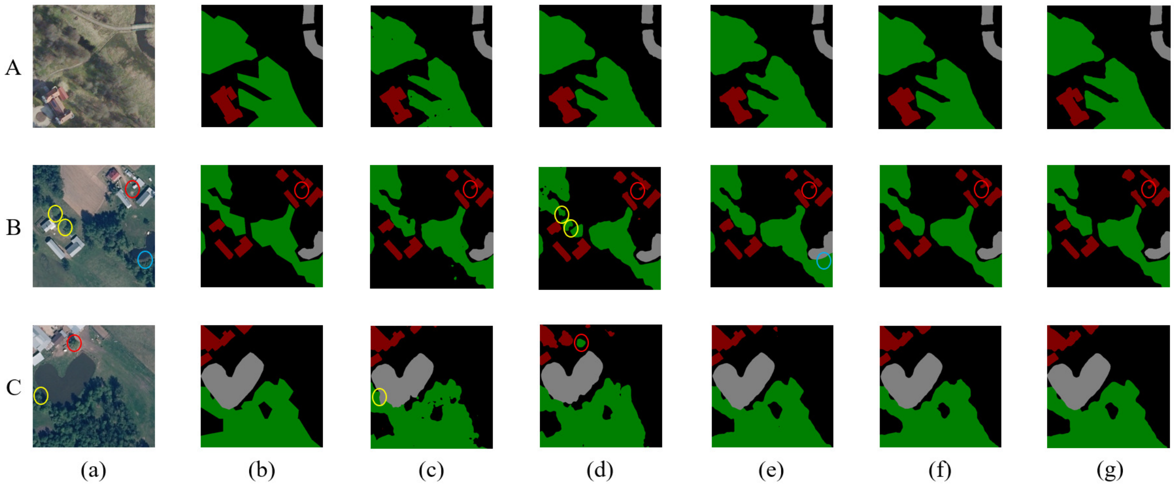


Figure 9. The prediction results of some models. The (**A**) line indicates the distribution of each category. The (**B**) line indicates the inside of forest and the situation that the ancillary buildings are difficult to identify. The (**C**) line indicates that the edge of each category is prone to classification errors. (**a**) Real image. (**b**) label. (**c**) BiSeNet. (**d**) PSPNet. (**e**) DeepLabv3+. (**f**) UNet. (**g**) DFFAN.

The loss curves of all models are shown in Figure 10. The convergence speed of DFFAN was slow, and it can be stabilized after 200 epochs of trainings, but its stability performance was better than the other modules. After the network is stable, the loss fluctuates less. BiSeNet was the opposite of DFFAN. BiSeNet converged very quickly before the 150th epoch. However, from the results, the metric of DFFAN better than BiSeNet.

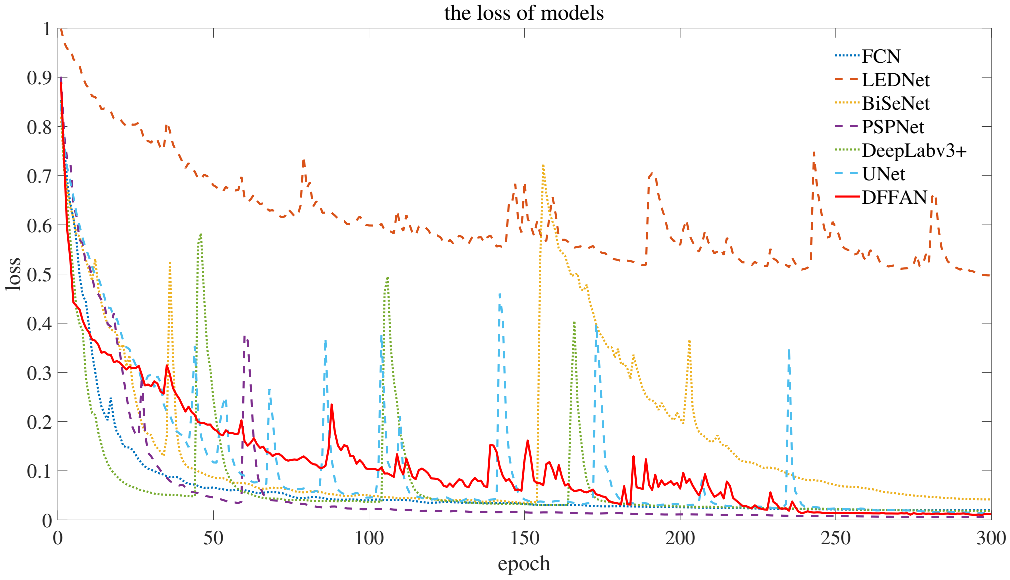


Figure 10. The loss curves of models.

The *MIoU* curves of models are shown in Figure 11. At the beginning of the experiment, the *MIoU* of PSPNet rose very rapidly. However, the *MIoU* of DFFAN exceeded the other models after the 153th epoch and kept the highest since then. This showed that, in the long run, DFFAN was more suitable for land cover segmentation.

In consideration that there is a large deviation of positive and negative in samples of the dataset, we use *Kappa* to evaluate the performance of models. Figure 12 shows the *Kappa* curves of models, each model's *Kappa* trend is similar to its *MIoU*. The *Kappa* of UNet is higher than that of other models at the beginning of the experiment, and the *Kappa* of UNet leads until it is overtaken by DFFAN.

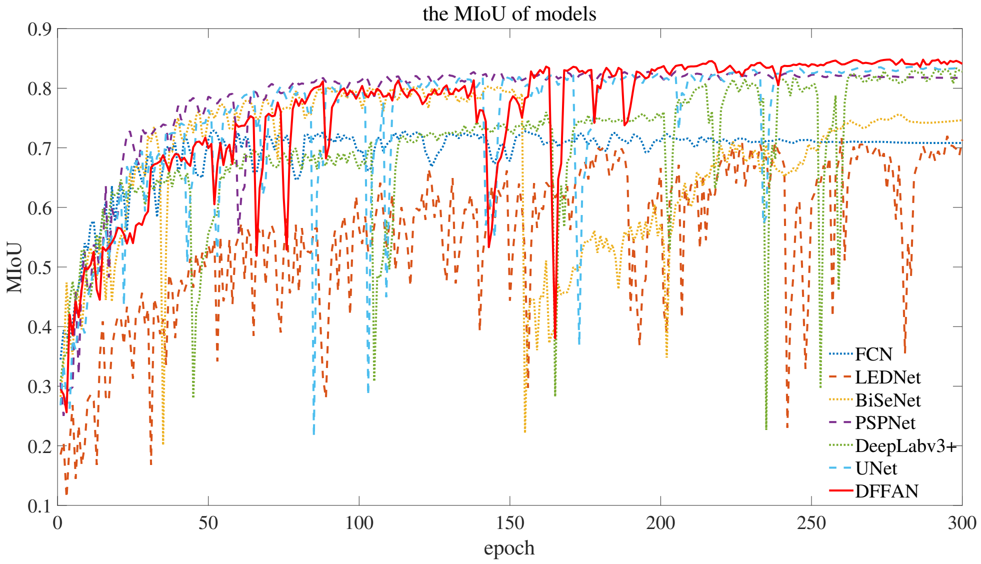


Figure 11. The MIoU curves of models.

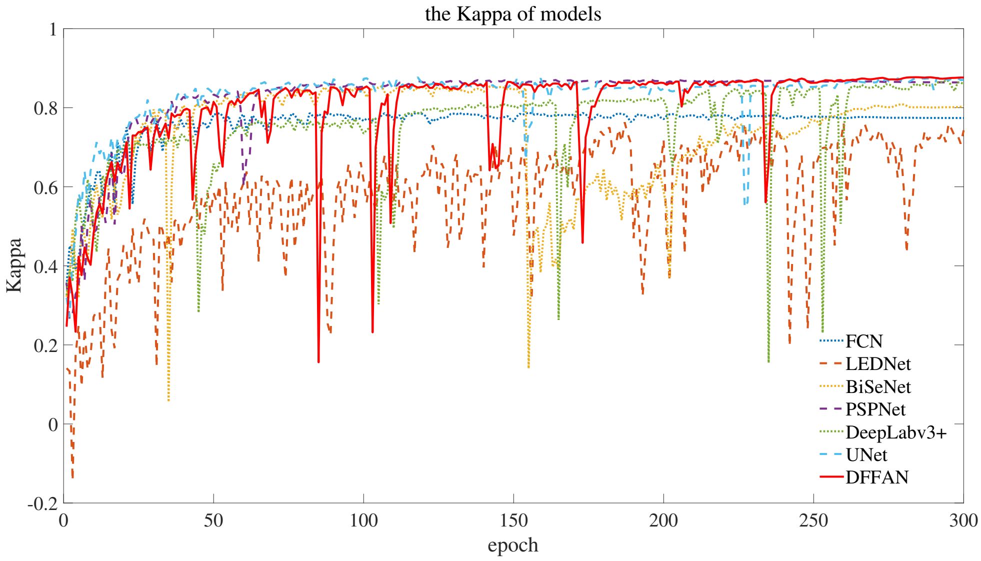


Figure 12. The Kappa curves of models.

To observe the advantages of DFFAN in predicting pictures when there are large differences between positive and negative samples and to verify the generalization of DFFAN we choose pictures that have dominant categories and use trained models to predict them. The prediction results are shown in Figure 13. In addition, these pictures we selected did not appear in the training set nor test set.

The yellow circles in Figure 13 are the parts that are difficult to predict. Although prediction results of DFFAN cannot be exactly the same as labels, its prediction ability is better than the other networks. This is because AMM can extract the context information of each feature map, and the BFF module can fuse the context information of the picture and extract the location information of the picture. It can better predict edge areas and distribution positions of each category.

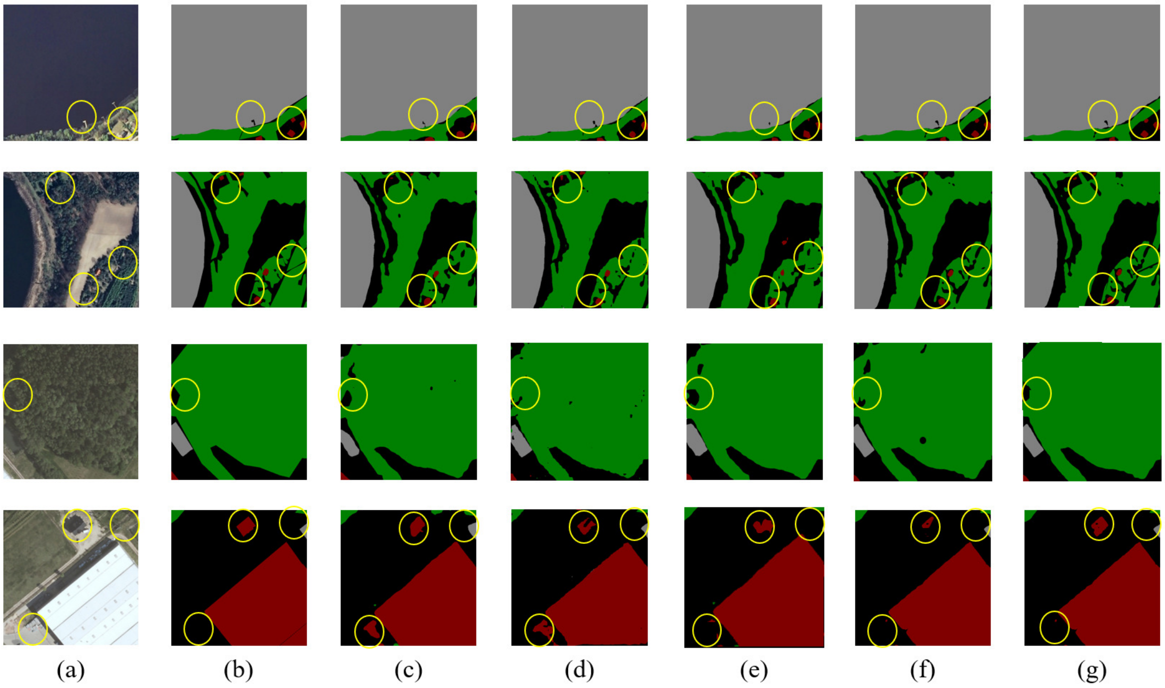


Figure 13. The prediction results of each model for special images. (**a**) Real image. (**b**) label. (**c**) BiSeNet. (**d**) PSPNet. (**e**) DeepLabv3+. (**f**) UNet. (**g**) DFFAN.

4.5. Generalization Experiment

To further verify the generalization abilities of the models proposed in this paper, AISD, a public dataset, is selected for further experiment. The AISD has three categories, namely building, road, and background. This work cuts the images into 512 × 512 pixels, and there are 30,000 pictures. According to a ratio of 7:3, 21,000 pictures are chosen as the training set and 9000 pictures are chosen as the validation set. With the SGD optimizer, the initial learning rate is 0.0001, the weight attenuation rate is 0.0005, the training batch batch-size is 4, and the iteration is 300 times.

The generalization experiment results are shown in Table 3. The results show that the metric of DFFAN is better than other models. Therefore, the generalization performance and effectiveness of the proposed network is verified.

Table 3. The results on the AISD dataset.

	MPA	$FWIoU$	$MIoU$	$Kappa$
FCN	0.8195	0.6955	0.6903	0.7171
LEDNet	0.8104	0.6823	0.6787	0.7039
BiSeNet	0.8584	0.7526	0.7493	0.7785
PSPNet	0.8595	0.7543	0.7524	0.78
DeepLabv3+	0.8668	0.7654	0.7621	0.791
UNet	0.8519	0.7429	0.7371	0.7673
DFFAN	0.8672	0.7661	0.763	0.7915

5. Conclusions

Land cover segmentation is one of the important applications of remote sensing image processing, it has important significance in agriculture, forestry, and public land planning. In order to explore the effect of convolutional neural networks in land cover semantic segmentation, this paper proposed DFFAN and conducted experiments on the LandCover. DFFAN uses ResNet as the backbone to extract different levels of features from remote sensing images. Furthermore, DFFAN uses the AMM to construct the context prior of each feature map to distinguish the contextual relevance of each pixel. Meanwhile, DFFAN uses FCME function module in BFF to extract spatial position information and uses asymmetric depth separate convolutions to aggregate spatial information and semantic information. In this way, BFF infers the spatial distribution of each category. The experimental results show that the evaluation metrics of DFFAN are better than those of comparing networks, and its prediction results are better in some edge areas.

However, DFFAN still has some shortcomings. First, DFFAN's convergence speed is slow, and it often takes 150 epochs before its evaluation metrics exceed the other networks. Second, for some small buildings, DFFAN can only mark their locations, but the outlines do not fit perfectly. This situation is the most obvious when predicting buildings are isolated in woodland. Finally, in edge areas of woodland and water, the prediction results are slightly different from labels.

To improve some shortcomings of DFFAN, the following methods can be considered. First, self-attention mechanism can be added to AMM, which can enhance the aggregation capability of context information. Secondly, smoothing ground truth in the training process can increase the training effect. Finally, we consider adding a classifier to the output of AMM, calculating the loss of classifier's output and ground truth, and setting the loss as auxiliary loss. In addition to the above methods, we will refer to relevant papers and learn from some ideas of state-of-the-art methods to improve the DFFAN.

Author Contributions: Conceptualization, Junqing Huang and Liguo Weng; methodology, Junqing Huang and Liguo Weng; software, Junqing Huang; validation, Junqing Huang, Liguo Weng and Bingyu Chen; formal analysis, Bingyu Chen; investigation, Bingyu Chen; resources, Min Xia and Liguo Weng; data curation, Bingyu Chen; writing—original draft preparation, Junqing Huang and Bingyu Chen; writing—review and editing, Min Xia; visualization, Junqing Huang; supervision, Min Xia; project administration, Min Xia; funding acquisition, Min Xia. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported in part by the National Natural Science Foundation of PR China (42075130,41875027).

Data Availability Statement: The data and the code of this study are available from the corresponding author upon request (002311@nuist.edu.cn).

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

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Networks
DFFAN	Dual Function Feature Aggregation Network
AMM	Affinity Matrix Module
BFF	Boundary Feature Fusion Module
RF	Random Forest
SVM	Support Vector Machine
FCN	Fully Convolutional Network
PSP	Pyramid Pooling
FCME	Feature Channels Maximum Element
IE	Information Extraction Function Module
CAM	Channel Attention Mechanism
MIoU	Mean Intersection over Union

References

1. Wulder, A.; Masek, G.; Cohen, B.; Loveland, T.R.; Woodcock, C.E. Opening the archive: How free data has enabled the science and monitoring promise of Landsat. *Remote Sens. Environ.* **2012**, *122*, 2–10. [CrossRef]
2. Blaschke, T. Object based image analysis for remote sensing. *ISPRS J. Photogramm. Remote Sens.* **2010**, *65*, 2–16. [CrossRef]
3. Reza, K.; Giorgos, M.; Stephen, V. A meta-analysis of remote sensing research on supervised pixel-based land-cover image classification processes: General guidelines for practitioners and future research. *Remote Sens. Environ.* **2016**, *177*, 89–100.

4. Xu, M.; Watanachaturaporn, P.; Varshney, P.; Arora, K. Decision tree regression for soft classification of remote sensing data. *Remote Sens. Environ.* **2005**, *97*, 322–336. [CrossRef]
5. Samaniego, L.; Bardossy, A.; Schulz, K. Supervised Classification of Remotely Sensed Imagery Using a Modified k-NN Technique. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 2112–2125. [CrossRef]
6. Gislason, P.; Benediktsson, J.; Sveinsson, J. Random forests for land cover classification. *Pattern Recognit. Lett.* **2006**, *27*, 294–300. [CrossRef]
7. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [CrossRef]
8. Xia, M.; Zhang, X.; Liu, W.; Weng, L.; Xu, Y. Multi-stage Feature Constraints Learning for Age Estimation. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 2417–2428. [CrossRef]
9. Ahlawat, S.; Amit, C.; Anand, N.; Saurabh, S.; Byungun, Y. Improved handwritten digit recognition using convolutional neural networks (CNN). *Sensors* **2020**, *20*, 3344. [CrossRef] [PubMed]
10. Lee, S.; Xiong, W.; Bai, Z. Human action recognition based on supervised class-specific dictionary learning with deep convolutional neural network features. *Comput. Mater. Contin.* **2020**, *63*, 243–262.
11. Sezer, O.; Ozbayoglu, A. Financial trading model with stock bar chart image time series with deep convolutional neural networks. *Intell. Autom. Soft Comput.* **2020**, *26*, 323–334. [CrossRef]
12. Qian, J.; Xia, M.; Zhang, Y.; Liu, J.; Xu, Y. TCDNet: Trilateral Change Detection Network for Google Earth Image. *Remote Sens.* **2020**, *12*, 2669. [CrossRef]
13. Xia, M.; Tian, N.; Zhang, Y.; Xu, Y.; Zhang, X. Dilated Multi-scale Deep Forest for Satellite Cloud Image Detection. *Int. J. Remote. Sens.* **2020**, *41*, 7779–7800. [CrossRef]
14. Weng, L.; Xu, Y.; Xia, M.; Zhang, Y.; Liu, J.; Xu, Y. Wate-'r Areas Segmentation from Remote Sensing Images Using a Separable Residual SegNet Network. *Int. J. Geo-Inf.* **2020**, *9*, 256. [CrossRef]
15. Xia, M.; Cui, Y.; Zhang, Y.; Liu, J.; Xu, Y. DAU-Net: A Novel Water Areas Segmentation Structure for Remote Sensing Image. *Int. J. Remote Sens.* **2021**, *42*, 2594–2621. [CrossRef]
16. Xia, M.; Wang, T.; Zhang, Y.; Liu, J.; Xu, Y. Cloud/shadow segmentation based on global attentionfeature fusion residual network for remote sensing imagery. *Int. J. Remote Sens.* **2021**, *42*, 2022–2045. [CrossRef]
17. Chen, B.; Xia, M.; Huang, J. MFANet: A Multi-Level Feature Aggregation Network for Semantic Segmentation of Land Cover. *Remote Sens.* **2021**, *13*, 731. [CrossRef]
18. Xia, M.; Liu, W.; 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]
19. Lee, S.; Ahn, Y.; Kim, H. Predicting concrete compressive strength using deep convolutional neural network based on image characteristics. *Comput. Mater. Contin.* **2020**, *65*, 1–17. [CrossRef]
20. Janarthanan, A.; Kumar, D. Localization based evolutionary routing (lober) for efficient aggregation in wireless multimedia sensor networks. *Comput. Mater. Contin.* **2019**, *60*, 895–912. [CrossRef]
21. Yang, W.; Li, J.; Peng, W.; Deng, A. A rub-impact recognition method based on improved convolutional neural network. *Comput. Mater. Contin.* **2020**, *63*, 283–299. [CrossRef]
22. Fang, W.; Zhang, W.; Zhao, Q.; Ji, X.; Chen, W. Comprehensive analysis of secure data aggregation scheme for industrial wireless sensor network. *Comput. Mater. Contin.* **2019**, *61*, 583–599. [CrossRef]
23. Kussul, N.; Lavreniuk, M.; Skakun, S.; Shelestov, A. Deep Learning Classification of Land Cover and Crop Types Using Remote Sensing Data. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 778–782. [CrossRef]
24. Li, Y.; Zhang, H.; Xue, X.; Jiang, Y.; Shen, Q. Deep learning for remote sensing image classification: A survey. *Wiley Interdiplinary Rev. Data Min. Knowl. Discov.* **2018**, *12*, e1264. [CrossRef]
25. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556
26. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition(CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
27. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
28. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional networks for biomedical image segmenation. In Proceedings of the 18th International Conference on Medical Image Computing and Computer Assisted lntervention(MICCAI), Munich, Germany, 5–9 October 2015; pp. 234–241.
29. Lin, G.; Milan, A.; Shen, C.; Reid, I. RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5168–5177.
30. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef]
31. Yu, C.; Wang, J.; Gao, C.; Yu, G.; Shen, C.; Sang, N. Context Prior for Scene Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2020, Seattle, WA, USA, 14–19 June 2020; arXiv:2004.01547 .
32. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition(CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6230–6239.

33. Huang, G.; Zhuang, L.; Laurens, M.; Weinberger, Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
34. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L. MobileNetV2: Inverted residuals and linear bottleneck. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition(CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4510–4520.
35. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
36. Diganta, M. Mish: A Self Regularized Non-Monotonic Activation Function. *arXiv* **2020**, *10*, arXiv:1908.08681.
37. Krizhevsky, A.; Sutskever, I.; Hinton, E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
38. Boguszewski, A.; Batorski, D.; Ziemba-Jankowska, N.; Zambrzycka, A.; Dziedzic, T. LandCover.ai: Dataset for Automatic Mapping of Buildings, Woodlands and Water from Aerial Imagery. *arXiv* **2020**, arXiv:2005.02264.