remote sensing

MDPI

Article

Learn by Yourself: A Feature-Augmented Self-Distillation Convolutional Neural Network for Remote Sensing Scene Image Classification

Cuiping Shi [1,2,*], Mengxiang Ding [1], Liguo Wang [3] and Haizhu Pan [4]

[1] College of Communication and Electronic Engineering, Qiqihar University, Qiqihar 161000, China; 2021910321@qqhru.edu.cn
[2] College of Information and Engineering, Huzhou University, Huzhou 313000, China
[3] College of Information and Communication Engineering, Dalian Nationalities University, Dalian 116000, China; wangliguo@hrbeu.edu.cn
[4] College of Computer and Control Engineering, Qiqihar University, Qiqihar 161000, China; panhaizhu@qqhru.edu.cn
* Correspondence: shicuiping@qqhru.edu.cn

Abstract: In recent years, with the rapid development of deep learning technology, great progress has been made in remote sensing scene image classification. Compared with natural images, remote sensing scene images are usually more complex, with high inter-class similarity and large intra-class differences, which makes it difficult for commonly used networks to effectively learn the features of remote sensing scene images. In addition, most existing methods adopt hard labels to supervise the network model, which makes the model prone to losing fine-grained information of ground objects. In order to solve these problems, a feature-augmented self-distilled convolutional neural network (FASDNet) is proposed. First, ResNet34 is adopted as the backbone network to extract multi-level features of images. Next, a feature augmentation pyramid module (FAPM) is designed to extract and fuse multi-level feature information. Then, auxiliary branches are constructed to provide additional supervision information. The self-distillation method is utilized between the feature augmentation pyramid module and the backbone network, as well as between the backbone network and auxiliary branches. Finally, the proposed model is jointly supervised using feature distillation loss, logits distillation loss, and cross-entropy loss. A lot of experiments are conducted on four widely used remote sensing scene image datasets, and the experimental results show that the proposed method is superior to some state-ot-the-art classification methods.

Keywords: remote sensing scene image; knowledge distillation; convolutional neural network; auxiliary branch; soft label

Citation: Shi, C.; Ding, M.; Wang, L.; Pan, H. Learn by Yourself: A Feature-Augmented Self-Distillation Convolutional Neural Network for Remote Sensing Scene Image Classification. *Remote Sens.* **2023**, *15*, 5620. https://doi.org/10.3390/rs15235620

Academic Editors: Junmin Liu, Xile Zhao, Tieyong Zeng, Bin Zhao and Claudia Paris

Received: 15 October 2023
Revised: 21 November 2023
Accepted: 30 November 2023
Published: 4 December 2023

1. Introduction

Remote sensing scene classification is the task of assigning a label to a specific scene. It has received extensive attention in recent years and is mainly used in urban planning, environmental surveying, natural disaster detection, and land use [1–4]. High-resolution remote sensing images have the characteristics of complex content, diverse semantics, and multi-scale targets. Remote sensing scene image classification is widely used, but due to the characteristics of remote sensing images, it is difficult to accurately classify remote sensing scene images. Therefore, the way in which to improve the classification accuracy of remote sensing scene images has become a research hotspot in the field of remote sensing. Traditional feature extraction uses hand-crafted features (e.g., texture features [5,6], spectral features [7,8], color features [9,10], and shape features [11,12]). Traditional classifiers are support vector machines [13] and decision trees [14]. Because it is difficult for manual features to fully describe the information of high-resolution remote sensing scene images,

traditional classifiers cannot classify the information of manual features well, and the classification performance of traditional models cannot meet our requirements. With the development of the deep convolutional neural network (DCNN) [15], DCNN-based classification methods have become more and more popular. At this stage, many methods have been proposed to distinguish remote sensing scene images [16–19]. The role of the feature extractor is to map the remote sensing scene image to appropriate visual features, while the role of the classifier is to classify the visual features into various semantic classes. Convolutional neural networks (CNNs) are outstanding in expressive feature learning and have achieved good performance in remote sensing scene classification. In conventional CNNs, one-hot ground-truth labels are used to guide feature learning. However, one-hot ground-truth labels only bring category information (i.e., which category the input image belongs to), but cannot provide the relationship between categories. For example, "Dense Residential Area" has the same distance to "Medium Residential Area" and "Airport", but "Dense Residential Area" is closer to "Medium Residential Area" than to "Airport". In this case, category information alone cannot accurately describe images, which leads to insufficient supervision in training. As shown in Figure 1, the label of the scene is "bridge". In addition to "bridge", there will be "river", "tree", "car", and "house" in the scene. If only bridge information is considered in the process of feature learning, other semantic information will reduce the discrimination of learned features.

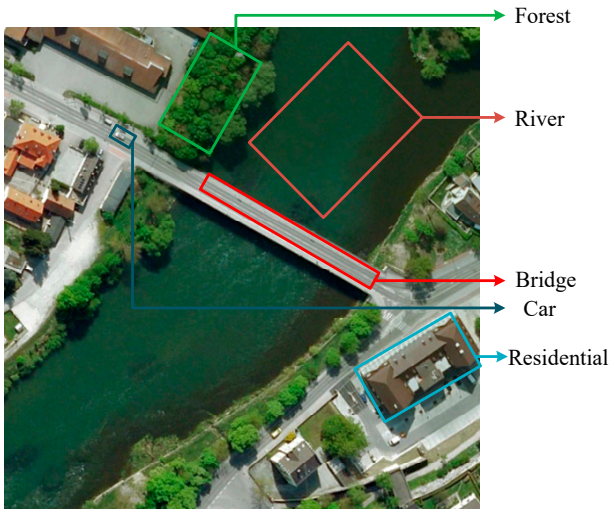

Figure 1. The remote sensing scene image above is manually semantically labeled as a bridge. There are multiple different land covers besides the bridge, including "River", "Forest", "Car", "Residential". If only bridges are considered in the feature learning process, the content corresponding to other semantics will reduce the discriminative degree of the learned features.

Knowledge distillation is a method of transferring the knowledge of the pre-trained teacher network to the student network, so that the small network can replace the large teacher network during the network deployment stage. The concept of knowledge distillation [20] was originally proposed by Hinton et al. and has been widely used in various fields and tasks. The basic principle of knowledge distillation is to learn the knowledge of a larger and more complex model by training a smaller and more lightweight model. Typically, complex models are referred to as "teacher models", while simplified models are referred to as "student models". The teacher model can be a deep neural network or other complex model, while the student model is usually a shallower or narrower layer neural network. By taking the output of the teacher model and its corresponding labels as the training target of the student model, the student model can gain more knowledge from the teacher model, and gradually approach or exceed the performance of the teacher model during the learning process. One of the main advantages of knowledge distillation is that it can significantly reduce model complexity and computational resource requirements

while maintaining relatively high performance. This makes knowledge distillation have a broad application potential in resource-constrained environments such as mobile devices, embedded systems, and edge computing. In addition, knowledge distillation can also be used as a method of model compression to reduce the cost of storage and inference by transferring the knowledge of complex models into simplified models.

Self-distillation (SD) [21] is a technique based on knowledge distillation. SD extracts knowledge from an already trained model and uses this knowledge to retrain the same model, thereby further improving the performance of the model. The core idea of the self-distillation method is to improve performance by letting the model learn its own knowledge. During training, the model uses its own soft labels as targets instead of hard labels.

Auxiliary classifiers [22,23] can enhance the performance of the main classifier by providing additional information. In this paper, an auxiliary classifier is used to learn the distribution of the classification results output by the classifier. During the training phase, the auxiliary classifier is trained together with the main classifier. Auxiliary classifiers can provide additional supervisory signals to help the main classifier better understand the data distribution. Due to the high cost of remote sensing image acquisition and the relatively small dataset, the use of auxiliary classifiers can help alleviate the overfitting problem of the main classifier. By introducing auxiliary classifiers, additional regularization effects can be provided to help reduce the degree of model fitting. The introduction of auxiliary classifiers can also increase the diversity of the model. If the auxiliary classifiers provide different predictions than the main classifier, the difference between them can help improve the overall classification performance. In general, the basic principle of the auxiliary classifier is to strengthen training, reduce overfitting, and improve diversity by providing additional information.

In order to train a compact model to achieve high classification performance and overcome the drawbacks of traditional distillation, a new self-distillation framework is proposed.

The main contributions of this paper are as follows.

1. This paper proposes a new self-distillation framework that effectively combines feature distillation and logits distillation to solve the problem of losing fine-grained information in traditional hard-label supervised models. This enables the backbone network to extract more representative features of the image and improve the generalization performance and adversarial nature of the model. Extensive experiments on four commonly used remote sensing scene image classification datasets have demonstrated the effectiveness of the proposed method.
2. In order to complement the advantages of multi-level features, a feature augmentation pyramid module is carefully designed, which fuses the top-level features with the low-level features through deconvolution to increase the richness of the features, so that the semantic features extracted by the deep network can be learned by the underlying network.
3. A method of adding two auxiliary classifiers in the middle layer is proposed, which is trained through distillation to provide additional supervisory information and help the network converge faster. In order to ensure that the shallow auxiliary classifier and the main classifier share similar feature representations, a bottleneck structure is added to the middle layer of the backbone network to encourage them to learn similar features.

The rest of this paper is organized as follows. Section 2 briefly introduces the research on remote sensing scene classification, and Section 3 provides a description of our proposed method in detail. Section 4 shows the experimental results and discussion. In Section 5, the conclusion and prospects are given.

2. Related Works

2.1. Classification of Remote Sensing Scene Images

Over the past few decades, many methods for image classification of remote sensing scenes have been proposed. Initially, these methods were mainly based on hand-crafted features, such as gradient histograms [24], scale-invariant feature transforms [25], and the bag-of-visual-word (BoVW) [26]. Although these methods yield impressive representations, handcrafted features cannot fully capture the complex content of remote sensing scenes. In recent years, the convolutional neural network (CNN) has performed well in extracting representational features and is widely used in image classification and target detection. It has achieved great success in remote sensing scene image classification, and many CNN-based methods have been proposed. For example, Li et al. [27] proposed a deep feature fusion network for remote sensing scene classification. Zhao et al. [28] proposed a structure that combines local spectral features, global texture features, and local structural features to fuse features. Wang et al. [29] use an attention mechanism to adaptively select key regions of an image, and then fuse features to produce more representative features. The key filter bank network (KFBNet) [30] uses a key filter bank to capture discriminative local details while preserving local features. Shi et al. [31] proposed a multi-branch fusion attention network, which fuses spatial attention and channel attention into the ResNet backbone network. Shi et al. [32] proposed a dense fusion of multi-level features, through 3×3 depthwise separable convolution and 1×1 standard convolution, to extract the information of the current layer and fuse it with the features extracted from the previous layer. Deng et al. [33] proposed a deep neural network incorporating contextual features, using the pre-trained VGG-16 as a feature extractor to obtain feature maps. Then, the feature map is input into two parallel modules, global average pooling (GAP) and long short-term memory (LSTM), to extract global and contextual features, respectively, and finally splicing global features and contextual features. Meng et al. [34] proposed a multi-layer feature fusion network based on spatial attention and a gating mechanism, using a backbone network to extract multi-layer convolutional features, and then using a spatial attention module to aggregate multi-layer features for classification. Wang et al. [35] proposed an enhanced feature pyramid network based on deep semantic embeddings. Using multi-level and multi-scale features, a feature fusion module is introduced to fuse the two branch features. Zhang et al. [36] proposed a distributed convolutional neural network.

2.2. Knowledge Distillation

The idea of knowledge distillation (Original Knowledge Distillation) is to guide the training of the student model by using the soft targets of the teacher model. The teacher network usually produces class probabilities by using a "softmax" output layer with a temperature hyperparameter applied, which is used to convert the logits generated by each class calculation, that is, z_i, into a probability q_i, and the calculation process can be represented as

$$q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_i/T)} \tag{1}$$

where T is the temperature, usually set to 1. Using a higher T value produces a softer distribution on the output of the classification.

As shown in Figure 2, an image is input into the network, the resulting output is fed into Softmax with temperature hyperparameters, and then a soft-label output is obtained. In the output soft label, we can see that for an input image labeled as a forest, in addition to the probability of the forest category, there will also be a certain probability of other categories in the output.

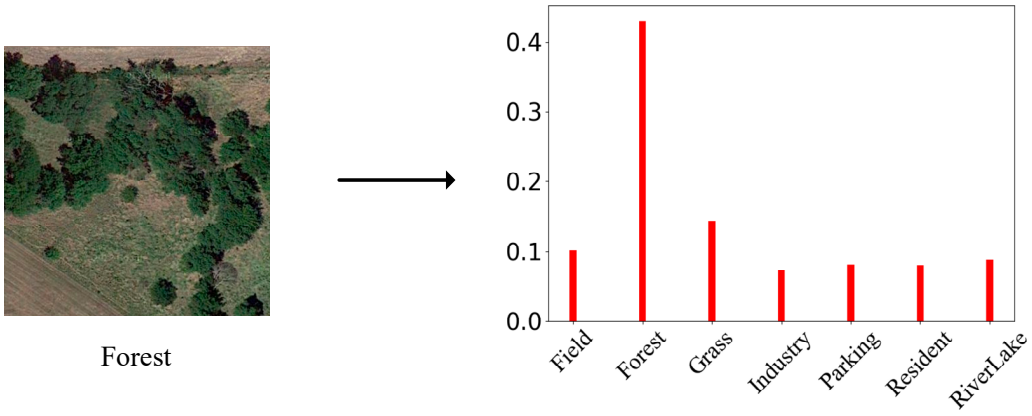

Figure 2. Network output soft-label histogram.

Soft labels are the probability distributions output by the teacher model, which can provide richer information to help the student model learn. To transfer knowledge effectively, an appropriate loss function needs to be defined to measure the difference between the output of the student model and the output of the teacher model. Commonly used loss functions include the mean squared error [37], cross-entropy loss [37], and *KL* divergence [38]. The mean square error loss function measures the numerical difference of the output, while the cross-entropy loss function measures the difference in the probability distribution of the output. *KL* divergence (Kullback–Leibler divergence), also known as relative entropy, is an indicator used to measure the difference between two probability distributions. The calculation process can be represented as

$$KL(P \parallel Q) = \sum P(i) log \frac{P(i)}{Q(i)} \tag{2}$$

where $P(i)$ represents the distribution predicted by the teacher network, and $Q(i)$. represents the distribution predicted by the student network.

KL divergence measures the loss of information from the true distribution to the model distribution. The real distribution is simulated using the output of the teacher network. In the process of knowledge distillation, a large teacher model is usually used for training first, and then the teacher model is adopted to generate soft labels, which are used together with the output of the student model to train the student model. During the training process, different weights can be used to balance the relative importance of hard and soft objects. Choosing an appropriate teacher model is crucial to the effect of knowledge distillation. In general, the teacher model should be complex and accurate enough to provide high-quality soft targets. A commonly used teacher model is a pre-trained deep neural network model. The student model is usually more lightweight and simplified than the teacher model for deployment where computing resources are constrained. Some common student model design strategies include using shallow network structures and reducing the number of network parameters. With the deepening of research, many improved and extended knowledge distillation methods have emerged. For example, the FitNets method [39] introduced the concept of intermediate layer alignment to align the intermediate layer outputs of the teacher model and the student model. The attention transfer method [40] learned knowledge from the teacher network by having the student network imitate the attention map of the teacher network. The relational knowledge distillation method [41] exploited relational modeling to improve knowledge distillation. A comprehensive overhaul of the feature distillation method [42] adopted the feature distillation, designed a new distillation loss, distilled features before the ReLU function, and retained negative values before distillation. Ahn et al. [43] proposed a variational information distillation framework, which transfers the knowledge learned by the convolutional network to the multi-layer perceptron (MLP)

and maximizes the mutual information of the two neural networks by maximizing the variational lower bound.

Due to the difficulty of selecting a teacher network and training a large teacher network, some studies have proposed self-distillation algorithms. The self-distillation framework distills knowledge within the network itself. The network is first divided into several parts. Then, the knowledge from the deep layers of the network is squeezed into the shallow layers. Zhang et al. [44] proposed a self-distillation framework using ResNet as the backbone network to extract the output of the intermediate layer through the bottleneck structure. The output obtained by the deep layer is used as a soft label to supervise the distribution of the shallow layer so that the shallow layer of the network learns the distribution of the deep layer. Ji et al. [45] proposed a self-distillation framework for feature refinement, which enhances feature maps through lateral convolutions for the purpose of self-knowledge distillation. Hu et al. [46] proposed a hierarchical self-distillation feature learning framework. The distribution generated by the shallow network is supervised by the distribution generated by the deep network. And a gradient separation and fusion module is proposed, and the gradient generated by the final classification output is not returned to the backbone network in reverse.

Influenced by the above work, knowledge distillation methods have been introduced into remote sensing image analysis, and Wei et al. proposed MSH-Net [47] to assist models with missing modalities by reconstructing complete modality-shared features from incomplete inference modality reasoning. Among them, the Joint Adaptive Distillation (JAD) method guides the model to learn modality-shared knowledge from multimodal models by matching the joint probability distribution between the representation and the ground truth. Hu et al. [48] proposed variational self-distillation to distill deep and shallow layers through Variational Knowledge Transfer (VKT), using the prediction vector of class entanglement information as supplementary class information. Li et al. proposed dual knowledge distillation [49], designing dual attention and a spatial structure. The two designed loss functions can effectively transfer the knowledge learned by the teacher network to the student network. Liu et al. proposed cross-model knowledge distillation, using the RGB image pre-trained model as a teacher model to guide multispectral scene classification [50]. Lin et al. [51] proposed a pyramid network, which used an interpolation method to generate high-resolution feature maps. Unlike these methods, we propose a feature-augmented self-distillation network. In the network architecture, the teacher network is an extension of the student network, which belongs to the same network architecture. We employ a pyramid module to fuse the top-level feature maps with the underlying multi-level features through deconvolution. The feature maps of the backbone network are then supervised using feature distillation with the fused features. At the same time, we add two auxiliary branches to the backbone network, and use the soft-label distillation loss of the auxiliary branch to supervise the shallow network to learn representational features. For the classifier of the backbone network, a combination of soft and hard labels for supervision is adopted.

3. Methodology

In this paper, a feature-augmented self-distillation convolutional neural network (FASDNet) is proposed, which is shown in Figure 3. It consists of the backbone classifier network in the gray area in the middle of the picture, the self-teacher network in the green area below the picture, and two auxiliary branches in the blue area above the picture. ResNet34 is utilized as the backbone network to generate multi-layer features. Then, multi-layer features are sent to the feature augmentation pyramid module to generate a refined feature map. The reason why green areas are called self-teacher networks is that in the same model, green areas are branches that extend from the backbone network. Green areas generate more refined feature maps, which can guide model learning through feature distillation. The core of self-teaching networks is that models use self-generated labels or targets for training. The backbone network contains 4 convolution blocks, and each

convolution block from bottom to top generates feature maps $S1$, $S2$, $S3$, and $S4$. The shape of $S1$ is $B \times C \times H \times W$, and B, C, H, and W represent the batch size, number of channels, and width and height of the feature map, respectively. The self-teacher network takes the horizontal feature map of the backbone network as input, and each convolutional block from the bottom to the top sequentially generates feature maps $T1$, $T2$, $T3$, $T4$. The shape of $T1$ is $B \times C \times 2 \times H \times W$, and the number of horizontal convolution kernels is set to $2C$, so that the number of channels of the feature map T is twice the number of channels of the feature map S. Two auxiliary branches adopt the convolutional bottleneck structure to further learn the representation of shallow features. The final loss is jointly determined using feature distillation loss, soft-label distillation loss, and ground truth loss. By combining the loss function to apply strong supervision to the network, the probability of network overfitting is greatly reduced.

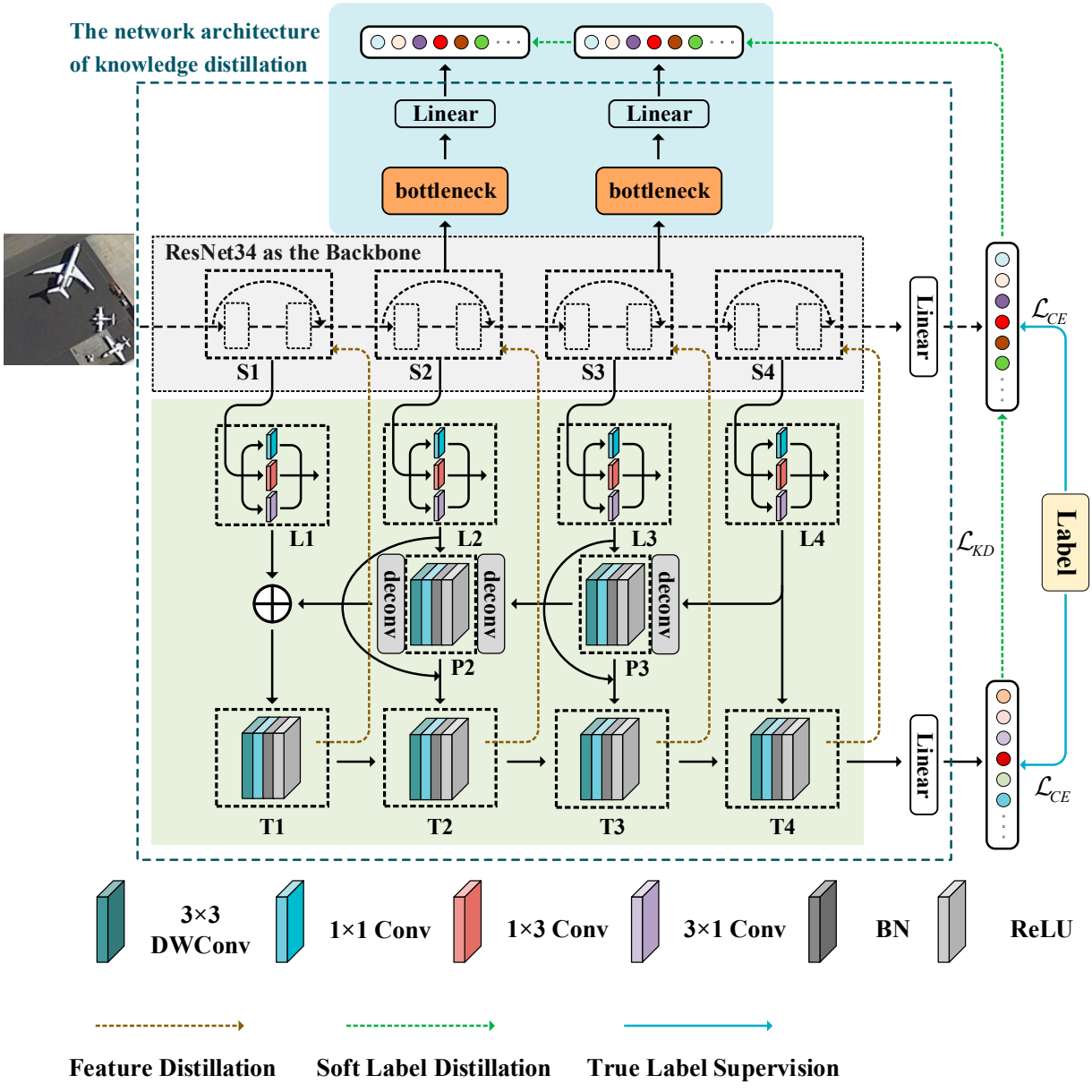

Figure 3. The overall framework of the proposed FASDNet.

3.1. Self-Distillation

The idea of the self-distillation method is to introduce some mechanisms to allow the model to generate some information by itself, and to use this information to perform self-

learning operations. It can be used for model compression. During the process of training, the self-teacher network will be included. During the testing stage, only the backbone classifier network needs to be deployed to compress the complex model into a smaller and lighter model, thereby reducing resource constraints. By introducing soft labels, the model pays more attention to the distribution of learning data instead of just hard labels, which helps improve the generalization performance of the model. The self-distillation method can also sustain the disturbance of input data, thus improving the robustness of the model. The proposed FASDNet adopts the green area part in Figure 3 as the self-teacher network, where the feature map is represented by T_i, and the output soft label is represented by $\hat{p}_t$. The gray area is utilized as the backbone network, and the feature map in it is represented by S_i. The feature map of the self-teacher is used to guide the feature map of the classifier network. In other words, it aims to learn the feature representation of the self-teacher network through a classifier network. For feature distillation, the distillation loss can be represented as

$$\mathcal{L}_{\mathrm{F}}(T, S; \theta_c, \theta_t) = \sum_{i=1}^{n} ||\varphi(T_i) - \varphi(S_i)||_2 \tag{3}$$

where φ represents pooling of feature maps along the channel dimension and L_2 normalization, θ_c represents the parameters of the classification network, θ_t represents the parameters of the self-teacher network, S represents the feature map of the classification network, and T represents the feature map of the self-teacher network. $\mathcal{L}_F$ enables the classification network to learn the enhanced feature map of the self-teacher network. This training can reduce the gap between the classification network and the self-teaching network. At the same time, the self-distillation method also uses soft labels for distillation, and the distillation loss is

$$\mathcal{L}_{\mathrm{KD}}(x; \theta_c, \theta_t, T) = D_{KL}(softmax(\frac{f_c(x; \theta_c)}{T}) || softmax(\frac{f_t(x; \theta_t)}{T})) \tag{4}$$

where f_c is the classifier network, f_t is the teacher network, D_{KL} represents the KL divergence between two distributions, $\mathcal{L}_{\mathrm{KD}}$ represents knowledge distillation loss, x represents the tensor after input data augmentation, θ_c represents the parameters of the classification network, θ_t represents the parameters of the self-teacher network, and T represents the temperature hyperparameter. In addition to the distillation loss, the classifier network and the self-teacher network adopt cross-entropy loss to learn the true labels. The cross-entropy loss can be represented as

$$\mathcal{L}_{CE}(x; \theta_c, \theta_t) = -\sum_{i=1}^{N} y_i \log(p_i^S(x; \theta_c) + (-\sum_{i=1}^{N} y_i \log(p_i^T(x; \theta_t)) \tag{5}$$

Among them, N represents the training sample, y_i represents the real label, p_i^S represents the output obtained by the backbone network after passing through the fully connected layer and then using Softmax, and p_i^T represents the output obtained from the teacher network after passing through the fully connected layer and then using Softmax.

3.2. Feature Augmentation Pyramid Module (FAPM)

The purpose of the teacher network is to provide an excellent feature map and soft labels for the classifier network. The input of the self-teacher network is the feature map $S_1, S_2, \ldots, S_n$ of the classifier network. Assume that the classifier network is divided into n blocks. The overall architecture of the feature augmentation pyramid module (FAPM) is shown in Figure 4. We refer to the green blocks in Figure 4 as the feature augmentation pyramid module, which mainly consists of deconvolution and convolution. Deconvolution is used to upsample deep features with rich semantic information, and then fuse them with the features obtained from horizontal convolution. The fused features are further processed using convolution.

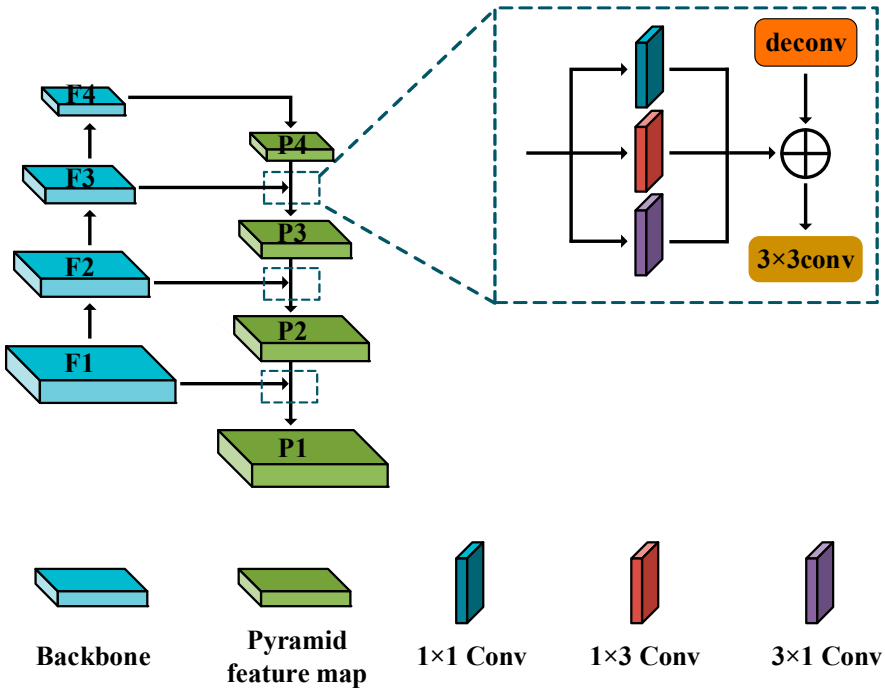

Figure 4. The overall architecture of FAPM.

The feature pyramid network can generate multi-scale and multi-level feature maps. In this paper, deconvolution is utilized as an upsampling technique. Deconvolution upsamples the feature map with rich semantic information twice. Compared to upsampling interpolation methods, deconvolution can enable the model to learn how to generate high-resolution features with more semantic significance from training data. Through deconvolution, the semantic-information-rich features are combined with the underlying features of the neural network to achieve spatial feature enhancement. Specifically, a top-down and bottom-up path design is adopted. The horizontal convolution layer is used before using the top-down path as follows.

$$L_i = Conv(S_i; d_i) \tag{6}$$

where $Conv$ is the convolution, batch normalization, and $ReLU$ activation function operations. The convolution includes parallel 1×1 convolution, 1×3 convolution, and 3×1 convolution. The $Conv$ output dimension is d_i, and we design $\mathrm{d}_i = w \times c_i$, where w is a width hyperparameter, which is set to 2 here. Among them, the 1×3 and 3×1 convolutions have direction sensitivity, and the 1×3 and 3×1 convolution kernels have different weights in the horizontal and vertical directions, which can better capture the directional features in the input data. Compared with the traditional 3×3 convolution kernel, the 1×3 and 3×1 convolution kernels have fewer parameters. When the number of input and output channels is both 1, 1×3 and 3×1 require 6 weight parameters, while 3×3 convolution requires 9 weight parameters. This can reduce the complexity and computational cost of the model. The process of top-down path is

$$\mathrm{P}_i = Conv'(w^{\mathrm{P}}_{i,1} \cdot L_i + w^{\mathrm{P}}_{i,2} \cdot Deconv(P_{i+1}); d_i) \tag{7}$$

The downsampling process adopts a combination of maximum pooling and 1×1 convolution, which can be represented as

$$T_{i+1} = Conv_{1\times1}(maxpool(T_i); \varphi_i) \tag{8}$$

where $Conv_{1\times1}$ represents a 1×1 convolution, T_i represents the feature map of each level of the self-teacher network, and φ_i represents the parameters of the convolution kernel. The process of the bottom-up path is

$$T_i = Conv'(w_{i,1}^T \cdot L_i + w_{i,2}^T \cdot P_i + w_{i,3}^T \cdot Conv_{1\times1}(maxpool(T_{i-1}); \varphi_i); d_i) \quad (9)$$

where P_i represents the output of the middle layer of the i layer in the top-down path, and T_i represents the output of the i layer in the bottom-up path. w^p and w^T represent normalized parameters. The convolution kernel size of deconvolution is 2×2, and the step size is 2. The feature map obtained after deconvolution is added element by element with the feature map generated using horizontal convolution. $Conv'$ represents the combination of convolution, batch normalization, and *ReLU* activation functions. The calculation process of $Conv'$ can be represented as

$$Conv' = ReLU(BN(Conv_{1\times1}(Conv_{dsc}(x)))) \quad (10)$$

$Conv_{dsc}$ represents 3×3 depth-separable convolution, 1×1 point convolution is used to interact with the feature maps in the channel dimension, and then the batch normalization and *ReLU* activation function are performed.

3.3. Auxiliary Classifier

Two additional branches are introduced into the middle layer of the network to assist in the training task, providing additional supervised information during the training process, accelerating model convergence, and improving model generalization. The distillation loss between the backbone network classifier and the auxiliary classifier can provide additional supervised signals for the model, which helps the gradient propagate back to the shallower layers of the network more effectively, thereby improving the training effect of the network. Introducing auxiliary branches can also serve as a regularization method by introducing additional tasks in the middle layer, forcing the network to learn effective representations for multiple tasks, thereby improving the generalization performance of the model. The position of shallow auxiliary classifiers and main classifiers in the network may lead to them learning different feature representations of the data. This leads to inconsistent weight updates for different parts. In this case, the weight adjustment between the shallow auxiliary classifier and the main classifier may not be coordinated, resulting in inconsistent classification results. To ensure that the shallow auxiliary classifier and the main classifier share similar feature representations, we added a bottleneck structure in the middle layer of the backbone network to encourage them to learn similar features. Soft labels are used instead of hard labels to supervise the network in the process of training shallow classifiers. A good shallow classifier can obtain more discriminative features, which in turn improves the performance of deep classifiers. The bottleneck structure of the auxiliary classifier we designed is shown in Figure 5. The blue square in the figure represents a 3×3 depth-separable convolution, which is utilized for further extracting local spatial features, and reducing parameters compared to ordinary convolution. The orange square represents a 1×1 point convolution, which is used to increase the dimension of the feature map. The purple square represents the batch normalization layer and the yellow square represents the activation function, which are used to finally pass through a global average pooling layer. The bottleneck structure can be represented as

$$Downsample = Avgpool2d(ConvBNReLU(S_i)) \quad (11)$$

where $ConvBNReLU$ represents the stacking of convolution, batch normalization, and *ReLU* activation functions, as shown in Figure 5. $Avgpool2d$ represents global average pooling. Using the bottleneck structure for downsampling can reduce the difference

between the shallow classifier and the deep classifier. The losses for supervising the two auxiliary classifiers with soft labels are

$$\mathcal{L}_{Aux1}(x;\theta_c,T) = D_{KL}(softmax(\frac{f_c(x)}{T})||softmax(\frac{f_{Aux1}(x)}{T})) \tag{12}$$

$$\mathcal{L}_{Aux2}(x;\theta_c,T) = D_{KL}(softmax(\frac{f_{Aux1}(x)}{T})||softmax(\frac{f_{Aux2}(x)}{T})) \tag{13}$$

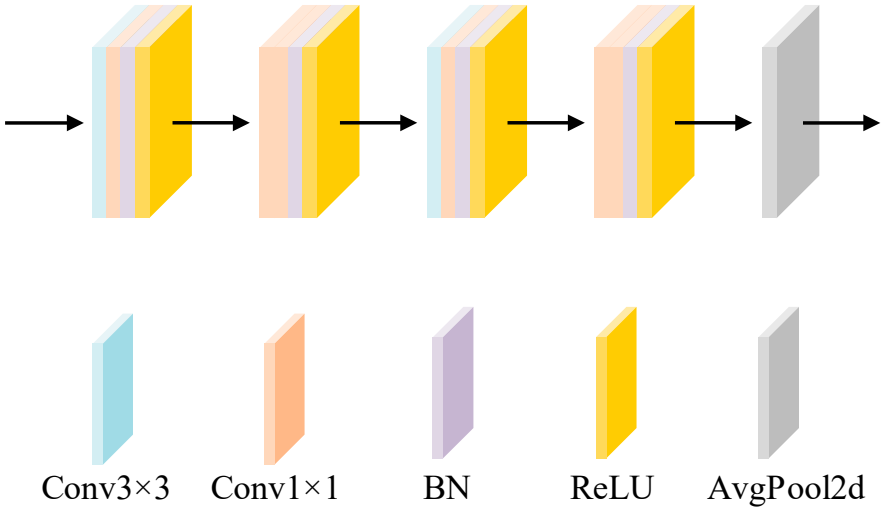

Figure 5. The bottleneck convolution structure of the auxiliary classifier.

The total loss of the auxiliary classifier is

$$\mathcal{L}_{Aux}(x;\theta_c,T) = \mathcal{L}_{Aux1}(x;\theta_c,T) + \mathcal{L}_{Aux2}(x;\theta_c,T) \tag{14}$$

where f_c represents the classifier network, and f_{Aux1} and f_{Aux2} represent the auxiliary classifier network. KL divergence is used as a metric distance to make shallow classifiers learn the distribution of deep classifier outputs. Among them, $\mathcal{L}_{Aux1}(x;\theta_c,T)$ represents the KL distance between the classification output of the backbone network and the deep auxiliary classifier. $\mathcal{L}_{Aux2}(x;\theta_c,T)$ denotes the KL distance between the deep auxiliary classifier and the shallow auxiliary classifier.

The supervised loss of the whole network consists of four parts. The first part is the feature distillation loss between the backbone network feature map and the self-teacher network feature map. The second part is the logits distillation loss between the backbone network classifier and the self-teacher classifier. The third part is the logits distillation loss between the backbone network classifier and the auxiliary classifier. The fourth part is the cross-entropy loss of the classifier network and the real label and the cross-entropy loss of the self-teacher network and the real label. The overall loss function can be expressed as

$$Loss = \mathcal{L}_F(T,F;\theta_c,\theta_t) + \mathcal{L}_{KD}(x;\theta_c,\theta_t,T) + \mathcal{L}_{Aux}(x;\theta_c,T) + \mathcal{L}_{CE}(x;\theta_c,\theta_t) \tag{15}$$

3.4. Implementation Details

The process of the proposed FASDNet is as follows. Firstly, the original remote sensing scene image is preprocessed. Then, the data are input into the backbone network to obtain feature maps S_1, $S2$, S_3, and S_4 at different stages. Using horizontal convolution to process feature maps S_1, S_2, S_3, and S_4, feature maps L_1, L_2, L_3, and L_4 are obtained. Following this, the feature map S_i is input into 1×1 convolution, 1×3 convolution, and 3×1 convolution for processing, i = 1, 2, 3, 4. The output features of the three parallel branches are added element by element to obtain the aggregated features. Among them, 1×3 and 3×1 convolutions have direction sensitivity and can better capture the directional features of the

input data. The deconvolution in the feature augmentation pyramid module upsamples features with rich semantic information and fuses the upsampled features with the features obtained using horizontal convolution. The fused features are further processed through convolution to obtain an enhanced feature map, represented by P_i. After a top-down path, the multi-level feature map enhanced by the pyramid module is obtained. Next is the bottom-up path. First, the feature maps of L_1 and P_1 are fused to obtain T_1, and then T_1 is fused with L_2 and P_2 through downsampling to obtain T_2. Through this bottom-up approach, the feature maps T_1, T_2, T_3, and T_4 are obtained, and then T_4 passes through a linear layer to obtain the output of the self-teacher network. Two auxiliary branches are added after the middle two layers of feature maps of the backbone network. The auxiliary branches consist of a bottleneck downsampling structure and an auxiliary classifier. Additional supervised information can be provided through distillation between auxiliary branches and the backbone network. The final supervised loss consists of four parts: feature distillation between the self-teacher network and the backbone network, logits distillation between the output of the self-teacher network and the output of the backbone network, logits distillation between the output of the backbone network and the auxiliary branch output, and cross-entropy loss between the output of the self-teacher network and the backbone classifier network and the real label. By updating the model parameters through the total loss, the trained model is ultimately obtained. The specific process of the FASDNet is shown in Algorithm 1.

Algorithm 1. The process of the proposed FASDNet

1. Data preprocessing to obtain the input tensor x.
2. Input x to the backbone network to obtain feature maps $S1$, $S2$, $S3$, $S4$ at different stages.
3. Use horizontal convolution to enhance the feature map obtained in step 2, $L_i = Conv(S_i; d_i)$
4. The feature map obtained in step 3 is input into the feature augmentation pyramid module, $\mathrm{P}_i = Conv'(w^{\mathrm{P}}_{i,1} \cdot L_i + w^{\mathrm{P}}_{i,2} \cdot \mathrm{D}econv(P_{i+1}); d_i)$
5. Combine the resulting enhanced feature map with the feature map of the horizontal convolution and the feature map after the maximum pooling $T_i = Conv'(w^T_{i,1} \cdot L_i + w^T_{i,2} \cdot P_i + w^T_{i,3} \cdot Conv_{1\times1}(maxpool(T_{i-1}); \varphi_i); d_i)$
6. The feature maps of the middle two layers of the backbone network are sent to the auxiliary branch, $Downsample = Avgpool2d(ConvBNReLU(S_i))$, then the output of the auxiliary classifier is obtained.
7. Calculate the overall supervised loss $Loss = \mathcal{L}_F(T, F; \theta_c, \theta_t) + \mathcal{L}_{KD}(x; \theta_c, \theta_t, T) + \mathcal{L}_{Aux}(x; \theta_c, T) + \mathcal{L}_{CE}(x; \theta_c, \theta_t)$
8. Updating model parameters through the overall supervised loss.
9. Obtain the output of the classifier.

4. Experiments

To evaluate the effectiveness of the proposed FASDNet, some experiments are performed on four public and challenging datasets, i.e., UC-Merced dataset [26], RSSCN7 dataset [52], AID [53], and NWPU-RESISC45 dataset [54], and the proposed FASDNet is compared with some advanced classification methods proposed in recent years. The experimental results show that the classification performance of the proposed method is superior to those of some state-of-the-art methods on all datasets.

4.1. Datasets

In this section, the four datasets used in the experiments are introduced briefly. Some examples selected in these datasets are shown in Figure 6.

Due to the significant difference in the number of images between different datasets, we used a larger training ratio for smaller datasets and a smaller training ratio for larger datasets. The training ratio on the four datasets is the same as that used in previous work [55–57]. The information of the four datasets is described in Table 1.

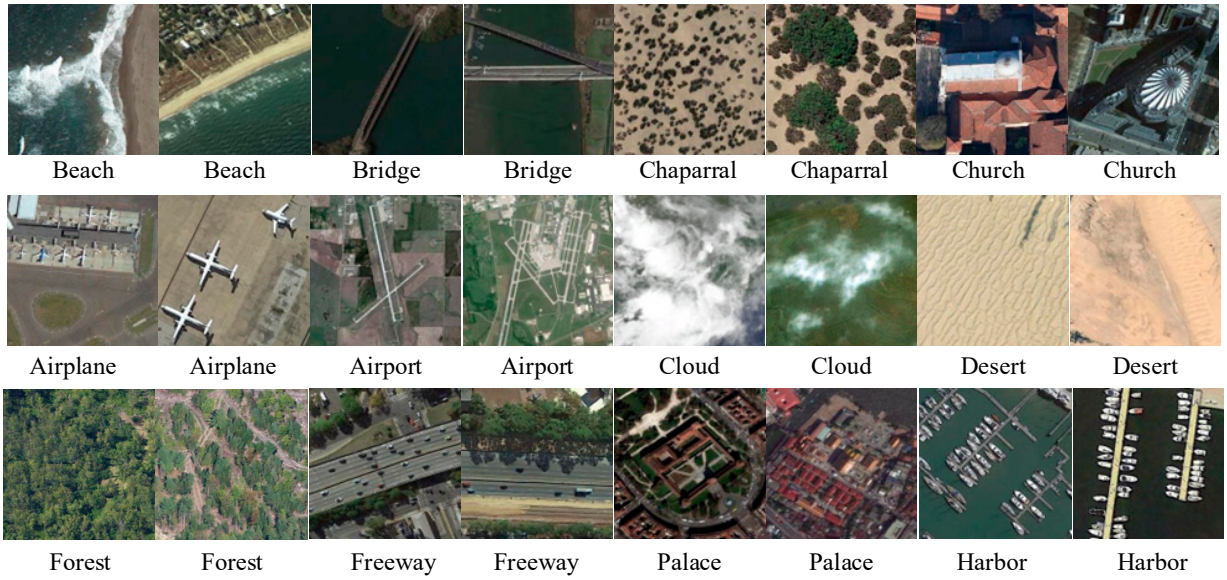

Figure 6. Randomly selected sample images from the four datasets.

Table 1. Data information of the four datasets.

Datasets	Number of Images per Class	Number of Scene Categories	Total Number of Images	Spatial Resolution (m)	Image Size
UC-Merced	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
NWPU-45	700	45	31,500	0.2–30	256 × 256

(1) UC-Merced

UC-Merced is a commonly used remote sensing image dataset for object classification tasks. This dataset was created and provided by the University of California, Merced. The UC-Merced dataset contains 21 different object categories and each category has 100 images, containing a total of 2100 images. Each image has a resolution of 256 × 256 pixels and is a color image (RGB format). The images cover different types of ground features such as cities, farmlands, forests, rivers, and parks. For the UC-Merced dataset, we divide the proportion of training into 50% and 80%, and the remaining 50% and 20% are used for testing.

(2) RSSCN7

RSSCN7 (Remote Sensing Scene Classification using Convolutional Networks) is a dataset for remote sensing scene classification. The RSSCN7 dataset contains seven common remote sensing scene categories, namely: Buildings, Forest, Farmland, River, Lake, Meadow, and Roads. Each category contains about 400 images, for a total of about 2800 images. Each image has a resolution of 256 × 256 pixels and is a color image (RGB format). For the RSSCN7 dataset, we divide the training ratio into 50%, and the remaining 50% is used for testing.

(3) AID

The AID (Aerial Image Dataset) is a widely used dataset for aerial image analysis, mainly for remote sensing image classification and target detection tasks. The AID was created by the Institute of Automation of the Chinese Academy of Sciences and contains 30 different object categories, covering cities, farmland, forests, grasslands, roads, rivers, lakes, buildings, and other types of objects. Each category contains approximately 200–400 images, for a total of approximately 10,000 images. Each image has a resolution of

600 × 600 pixels and is a color image (RGB format). For the AID, we divide the training ratio into 20% and 50%, and the remaining 80% and 50% are used for testing.

(4) NWPU-RESISC45

NWPU-RESISC45 is a widely used remote sensing image dataset for remote sensing image classification tasks. This dataset is provided by Northwestern Polytechnical University (NWPU) in China and is one of the datasets for the RESISC45 (Remote Sensing Image Scene Classification) competition. The NWPU-RESISC dataset contains 45 different remote sensing image scene categories and each category has 700 images, containing a total of 31,500 images. Each image has a resolution of 256 × 256 pixels and is a color image (RGB format). The images cover different geographical environments and scenes, including cities, farmlands, rivers, forests, grasslands, airports, etc. For the experiment on the NWPU-RESISC45 dataset, we divide the training ratio into 10% and 20%, and the remaining 90% and 80% are used as the test set.

4.2. Experimental Details

All experiments are implemented using Pytorch on a workstation with a GeForce RTX 3090. ResNet34 in the network is initialized with parameters pre-trained on ImageNet [58], and the rest of the network uses randomly initialized parameters. The adaptive moment estimation is adopted to optimize the model, the initial learning rate is set to 0.0001, and the training is 150 epochs. The cosine decay learning rate adjustment is used, and the learning rate decays to 0.1 times the original at the 30th epoch, 50th epoch, and 100th epoch. We first resize the image to 448 × 448. Random horizontal flip, random vertical flip, and random rotation with fixed angles are adopted to enhance the image. In addition, a color enhancement method is adopted to enhance the image. In the experiments, the overall accuracy (OA) is adopted to evaluate the effectiveness of our proposed method. OA is the number of correctly predicted images in the test set divided by the total number of images in the test set. To ensure the accuracy of the experimental results, the final results are obtained by averaging 10 experiments. In addition, the confusion matrix is adopted to analyze the prediction results of different categories.

4.3. Experimental Results and Analysis

To evaluate the effectiveness of our proposed method, a series of experiments are conducted on four datasets. Some advanced classification methods using multi-layer feature aggregation and global deep features are used for comparison. The experimental results are listed in Table 2.

Table 2. The remote sensing scene classification methods studied in recent years to be compared, where * indicates a classification method based on global deep features, · indicates a classification method based on multi-layer feature aggregation, and † indicates an image classification method using knowledge distillation.

Methods	Year
GoogleNet/VGGNet-16 [53] *	TGRS2017
VGG-VD16 + MSCP + MRA [59] ·	TGRS2018
VGG-16-CapsNet [60] *	RS2019
SCCov [19] ·	TNNLS2019
GBNet + global feature [57] ·	TGRS2020
MG-CAP(Sqrt-E) [61] *	TIP2020
MIDC-Net_CS [62] *	TIP2020
ACR-MLFF [63] ·	GRSL2021
ACNet [64] *	JSTARS2021
MSA-Network [65] *	JSTARS2021
RANet [66] ·	JSTARS2021
EFPN-DSE-TDFF [35] ·	TGRS2021

Table 2. *Cont.*

Methods	Year
DFAGCN [67] ·	TNNLS2021
EMTCAL [68] *	TGRS2022
MLF2Net_SAGM [34] *	RS2022
CFDNN [33]	RS2022
MBFANet [69] *	GRSL2023
SAGN [55] *	TGRS2023
VSDNet-ResNet34 [48] †	TGRS2022

(1) Classification results on the UC-Merced dataset

Some methods with good classification performance on the UC-Merced dataset in recent years are chosen to compare with the proposed FASDNet. The experimental results are listed in Table 3. We can see that the classification accuracy of the proposed method reaches 99.90% when the training ratio is 80%, which exceeds all comparison methods. The OA of the proposed FASDNet is 0.33% higher than that of EMTCAL, which also uses a ResNet34 backbone. The OA of the proposed FASDNet is 0.08% higher than that of the SAGN method that uses a dense network to extract underlying features and then uses graph convolution to further aggregate features. The OA of the proposed FASDNet is 0.33% higher than that of VSDNet-ResNet34, which also uses the distillation method.

Table 3. Comparison of our proposed method with some methods proposed in recent years on the UC-Merced dataset.

Method	OA (50%)	OA (80%)
GoogleNet [53]	92.70 ± 0.60	94.31 ± 0.89
VGG-16 [53]	94.14 ± 0.69	95.21 ± 1.20
VGG-16-CapsNet [60]	95.33 ± 0.18	98.81 ± 0.22
SCCov [19]	-	99.05 ± 0.25
VGG-VD16 + MSCP + MRA [59]	-	98.40 ± 0.34
GBNet + global feature [57]	97.05 ± 0.19	98.57 ± 0.48
MIDC-Net_CS [62]	95.41 ± 0.40	97.40 ± 0.48
EFPN-DSE-TDFF [35]	96.19 ± 0.13	99.14 ± 0.22
RANet [66]	97.80 ± 0.19	99.27 ± 0.24
DFAGCN [67]	-	98.48 ± 0.42
MG-CAP(Sqrt-E) [61]	-	99.00 ± 0.10
MSA-Network [65]	97.80 ± 0.33	98.96 ± 0.21
ACR-MLFF [63]	97.99 ± 0.26	99.37 ± 0.15
EMTCAL [68]	98.67 ± 0.16	99.57 ± 0.28
MBFANet [69]	-	99.66 ± 0.19
SAGN [55]	-	99.82 ± 0.10
VSDNet-ResNet34 [48]	98.49 ± 0.18	99.67 ± 0.18
FASDNet (ours)	**98.71 ± 0.13**	**99.90 ± 0.10**

The confusion matrix obtained in the case of the 80% training ratio is shown in Figure 7. From Figure 7, we can see that each category has been well classified. The above experimental results on the UC-Merced dataset indicate that the proposed FASDNet method exhibits excellent classification performance.

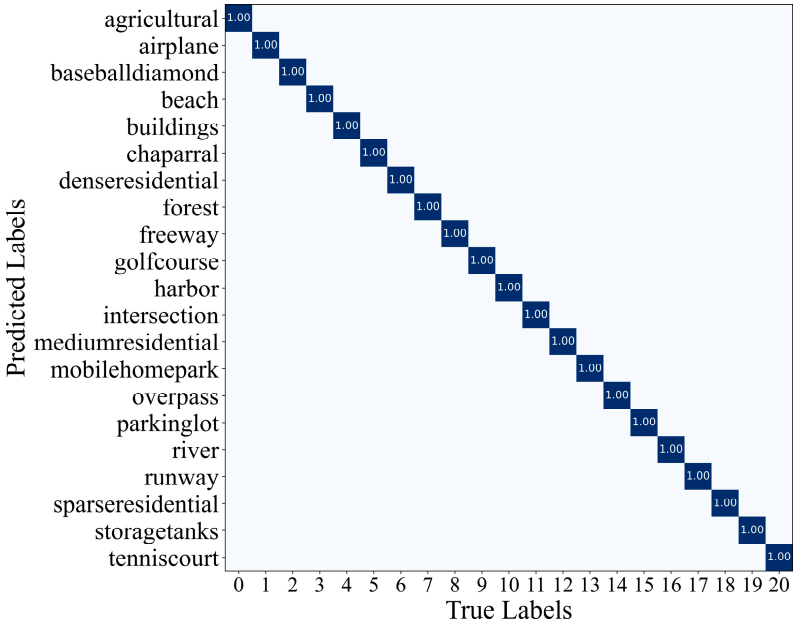

Figure 7. Confusion matrix on the UC-Merced dataset with 80% training ratio.

(2) Classification results on the RSSCN7 dataset

The proposed method is compared with some methods proposed in recent years on the RSSCN7 dataset. The experimental results are shown in Table 4. The OA of our proposed method is 97.79%, which is 1.78%, 1.81%, and 2.25% higher than those of MLF2Net_SAGM, PCANet, and Contourlet CNN, respectively. The experimental results prove that our proposed method has good feature representation ability.

Table 4. Comparison of our proposed method with some methods proposed in recent years on the RSSCN7 dataset.

Method	OA (50%)
BoVW(SIFT) [53]	81.34 ± 0.55
Tex-Net-LF_VGG-M [70]	91.25 ± 0.57
Resnet50 [70]	93.12 ± 0.55
WSPM-CRC-ResNet152 [71]	93.9
Tex-Net-LF_Resnet50 [70]	94.00 ± 0.57
DFAGCN [67]	94.14 ± 0.44
SE-MDPMNet [72]	94.71 ± 0.15
Contourlet CNN [17]	95.54 ± 0.71
PCANet [18]	95.98 ± 0.56
MLF2Net_SAGM [34]	96.01 ± 0.23
FASDNet (ours)	**97.79 ± 0.14**

The confusion matrix of the proposed FASDNet on the RSSCN7 dataset is shown in Figure 8, which shows the proposed method can provide good classification performance. The classification accuracy rate of all scenes is up to 97%, and the classification accuracy rate of the "forest" scene can reach 99%. It can be seen from the figure that some "Field" scenes are misclassified as "Grass", and some "Grass" scenes are misclassified as "Field". This is due to the strong inter-class similarity between the "Grass" and "Field" scenarios. There is also a misclassification between the "Industry" and "Parking" scenes, because the "Industry" scene contains many parking areas, while the "Parking" scene contains many industrial-area-style buildings. This makes it difficult for our proposed method to distinguish them as well. Nevertheless, the proposed method still achieved excellent classification performance.

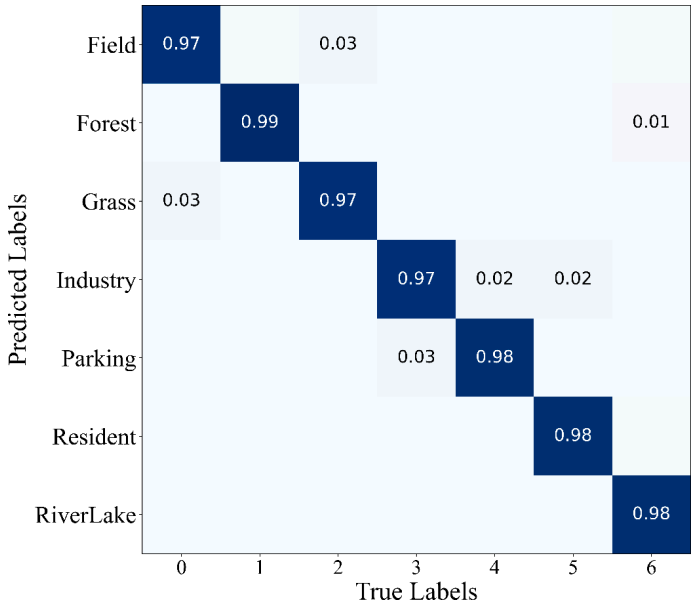

Figure 8. The confusion matrix obtained under the 50% training ratio of the RSSCN7 dataset.

(3) Classification results on the AID:

Some methods proposed in recent years are selected to compare with our proposed method, and the experimental results are shown in Table 5. With a training ratio of 20% on the AID, the classification accuracy of the proposed FASDNet is 95.68%. It is 0.78% higher than that of the SAGN method, 1.97% higher than that of the MBFANet method, and 1.26% higher than that of the EMTCAL method. When the training ratio of the AID is 50%, the classification accuracy of the proposed FASDNet is 97.84%, which is 1.07% higher than that of the SAGN method, 0.91% higher than that of the MBFANet method, and 1.43% higher than that of the EMTCAL method. The experimental results fully demonstrate the effectiveness of our proposed method.

Table 5. Comparison of our proposed method with some methods proposed in recent years on the AID.

Method	OA (20%)	OA (50%)
GoogleNet [53]	83.44 ± 0.40	86.39 ± 0.55
VGG-16 [53]	86.59 ± 0.29	89.64 ± 0.36
VGG-16-CapsNet [60]	91.63 ± 0.19	94.74 ± 0.17
SCCov [19]	93.12 ± 0.25	96.10 ± 0.16
VGG-VD16 + MSCP + MRA [59]	92.21 ± 0.17	95.56 ± 0.18
GBNet + global feature [57]	92.20 ± 0.23	95.48 ± 0.12
MIDC-Net_CS [62]	88.51 ± 0.41	92.95 ± 0.17
EFPN-DSE-TDFF [35]	94.02 ± 0.21	94.50 ± 0.30
ACNet [64]	92.71 ± 0.14	95.31 ± 0.37
DFAGCN [67]	-	94.88 ± 0.22
MG-CAP(Sqrt-E) [61]	93.34 ± 0.18	96.12 ± 0.12
MSA-Network [65]	93.53 ± 0.21	96.01 ± 0.43
ACR-MLFF [63]	92.73 ± 0.12	95.06 ± 0.33
EMTCAL [68]	94.69 ± 0.14	96.41 ± 0.23
MBFANet [31]	93.98 ± 0.15	96.93 ± 0.16
SAGN [55]	95.17 ± 0.12	96.77 ± 0.18
VSDNet-ResNet34 [48]	96.00 ± 0.18	97.28 ± 0.14
FASDNet (ours)	**96.05 ± 0.13**	**97.84 ± 0.12**

The confusion matrix diagram under the 50% training ratio on the AID is shown in Figure 9. Among the 30 categories, 28 categories have an accuracy of more than 90%, and only two categories have an accuracy that does not reach 90%. The two categories are "Resort" and "Square". The "Resort" scene category is mainly misclassified as schools and parking lots. The "Square" scene category is mainly misclassified into parking lots, central areas, and schools. This is due to the high inter-class similarity of scene categories. Further improving the performance of FASDNet is our future work.

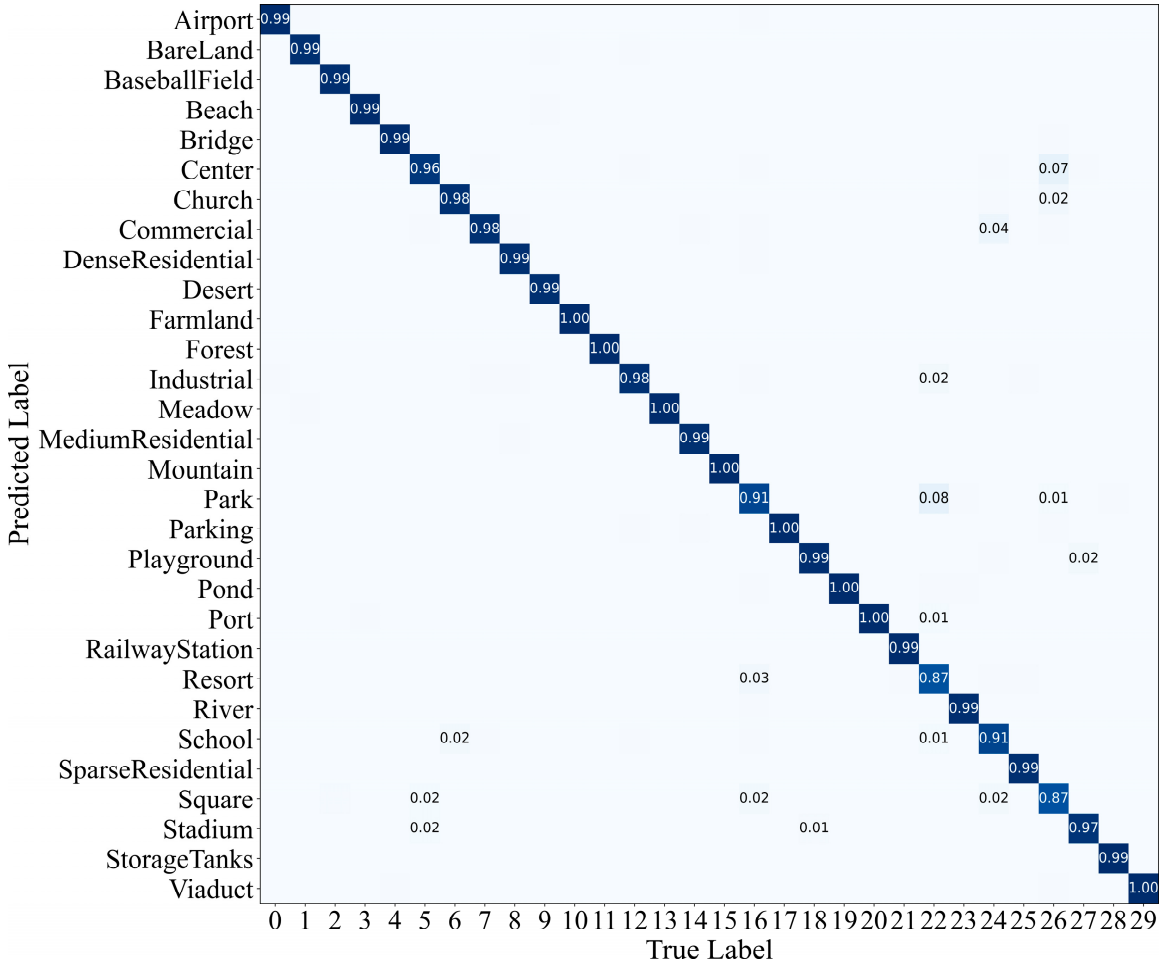

Figure 9. The confusion matrix obtained under the 50% training ratio of the AID.

(4) Classification results on the NWPU dataset:

The experimental results of different classification methods are summarized in Table 6. The overall accuracy reaches 92.89% under the 10% training ratio and 94.95% under the 20% training ratio. Compared with other methods, our proposed method achieves the best classification results under training ratios of 10% and 20%. At the 10% training ratio, the proposed FASDNet is 0.76% higher than that of the VSDNet-ResNet34 method, 1.16% higher than that of the SAGN method, 1.28% higher than that of the MBFANet method, and 1.26% higher than that of the EMTCAL method. At the 20% training ratio, the proposed FASDNet is 0.27% higher than that of the VSDNet-ResNet34 method, 1.46% higher than that of the SAGN method, 0.94% higher than that of the MBFANet method, and 1.30% higher than that of the EMTCAL method. These experimental results fully validate the effectiveness of our proposed method on the remote sensing scene classification task.

The confusion matrix of FASDNet on the NWPU dataset is shown in Figure 10. We can see that under a training ratio of 20%, only 4 of the 45 categories did not achieve 90% accuracy, and 29 categories achieved more than 95% accuracy. Among these categories,

the "palace" category is mainly misclassified as the "church" category. And the "church" category is misclassified as "palace" just as much. The reason is that the architectural styles of churches and palaces have great similarities.

Table 6. Comparison of our proposed method with some methods proposed in recent years on the NWPU dataset.

Method	OA (10%)	OA (20%)
GoogleNet [53]	76.19 ± 0.38	78.48 ± 0.26
VGG-16 [53]	76.47 ± 0.18	79.79 ± 0.15
VGG-16-CapsNet [60]	85.08 ± 0.13	89.18 ± 0.14
SCCov [19]	89.30 ± 0.35	92.10 ± 0.25
VGG-VD16 + MSCP + MRA [59]	88.07 ± 0.18	90.81 ± 0.13
MIDC-Net_CS [62]	86.12 ± 0.29	87.99 ± 0.18
ACNet [64]	91.09 ± 0.13	92.42 ± 0.16
DFAGCN [67]	-	89.29 ± 0.28
MG-CAP(Sqrt-E) [61]	90.83 ± 0.12	92.95 ± 0.13
MSA-Network [65]	90.38 ± 0.17	93.52 ± 0.21
ACR-MLFF [63]	90.01 ± 0.33	92.45 ± 0.20
EMTCAL [68]	91.63 ± 0.19	93.65 ± 0.12
MBFANet [31]	91.61 ± 0.14	94.01 ± 0.08
SAGN [55]	91.73 ± 0.18	93.49 ± 0.10
VSDNet-ResNet34 [48]	92.13 ± 0.16	94.68 ± 0.13
FASDNet (ours)	**92.89 ± 0.13**	**94.95 ± 0.12**

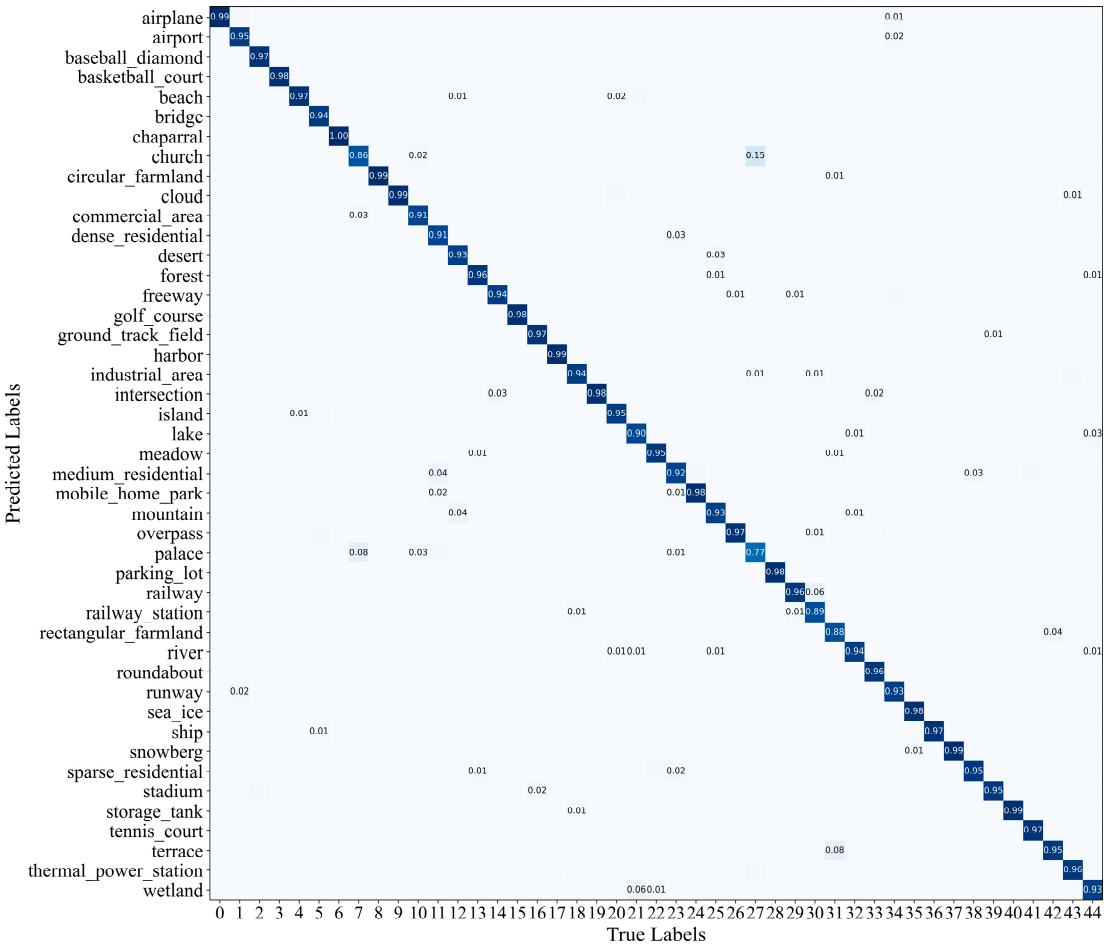

Figure 10. The confusion matrix obtained under the 20% training ratio of the NWPU dataset.

4.4. Evaluation of Size of Models

The floating-point operations (FLOPs) and parameter quantities of some network models are listed in Table 7. The FLOPs measure the complexity of the model. Table 7 shows that compared with EMTCAL, the proposed method has advantages in FLOPs and parameter quantity, with a classification accuracy of 1.43% higher than that of EMTCAL, demonstrating the advantages of the proposed method. Compared with methods such as GoogLeNet, SE-MDPMNet, and Contourlet CNN, although they have disadvantages in terms of parameter quantity and FLOPs, they greatly surpass these models in terms of classification accuracy. It is worth mentioning that our model only needs to deploy a backbone classifier network during the deployment phase. In this way, the model complexity and computational resource requirements are reduced. From Table 7, it can also be seen that the number of FLOPs and parameters of the model during deployment is less than that during training.

Table 7. Complexity evaluation of some models.

The Network Model	OA (%)	Number of Parameter	FLOPs
GoogLeNet [53]	85.84	6.1 M	24.6 M
CaffeNet [53]	88.25	60.97 M	715 M
VGG-VD-16 [53]	87.18	138 M	15.5 G
SE-MDPMNet [72]	92.46	5.17 M	3.27 G
Contourlet CNN [17]	95.54	12.6 M	2.1 G
EMTCAL [68]	96.41	27.8 M	4.3 G
FASDNet (Our training model)	97.84	24.7 M	3.8 G
FASDNet (Our deployment model)	97.84	21.8 M	3.6 G

4.5. Discussion

In order to comprehensively evaluate the effectiveness of our proposed method, some ablation experiments and heat map analysis are conducted. Grad-CAM can make full use of the features of the last layer of neural convolution to generate an attention map, also called a heat map, to display important areas in the image. In these experiments, some scene images are randomly selected, such as "parking lot", "residential", and "river", in the RSSCN7 dataset. The heat maps obtained by only the backbone network and the backbone network combined with the distillation method are shown in Figure 11.

We can see from the figure that for the "Parking" scene, the method using only the backbone network cannot accurately focus on the parking area. In addition to the parking area, the network also focused on other parts. After combining with our proposed distillation method, the network is obviously more focused on the parking area. For the "resident" scene, only using the backbone network method, the network is partially biased in the region of interest and ignores similar surrounding objects, which can only use limited features for classification. However, our proposed method focuses on the target region very well. For the "RiverLake" scene, the method without distillation can only focus on the edge information of the scene, and cannot fully extract the target, which will affect the classification accuracy. After using the distillation method, the network can focus more on the complete region of interest.

To verify the effectiveness of our three proposed modules, some ablation experiments were conducted on four datasets. The results of the ablation experiment are shown in Tables 8–11. Each experimental result given in the table is the average of 10 repeated experimental results. In the first case, the network only includes the backbone network, resulting in the model with the worst classification performance. In the second case, the classification accuracy is improved by combining distillation methods with networks. In the third case, adding distillation methods and feature augmentation pyramid modules to the network further improves the classification accuracy compared to distillation-only methods. The fourth case adds distillation methods and auxiliary branches to the network, which improves the classification accuracy compared to using only distillation methods.

The last case connects all modules, i.e., distillation methods, feature augmentation pyramid modules, and auxiliary branches. It can be seen that from the four tables, when the network includes these three modules, the highest classification accuracy can be achieved. The ablation study has fully demonstrated the effectiveness of the main modules in FASDNet.

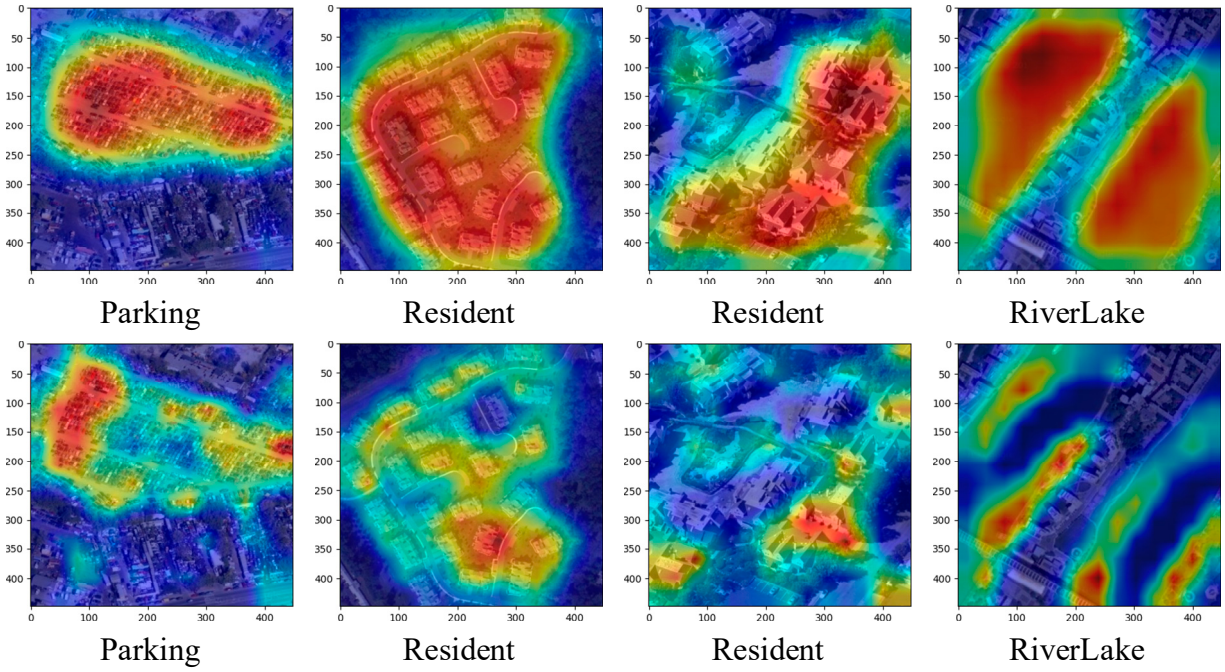

Figure 11. The heat maps obtained using different methods. The first row shows the heat maps obtained using the backbone network combined with the distillation method. The second row shows the heat maps obtained with only the backbone network.

Table 8. Some ablation experiments of the proposed FASDNet on the UC-Merced dataset.

Condition	KD	FAPM	Auxiliary	OA
1				98.57 ± 0.18
2	✓			99.29 ± 0.12
3	✓	✓		99.76 ± 0.15
4	✓		✓	99.55 ± 0.12
5	✓	✓	✓	99.90 ± 0.10

Table 9. Some ablation experiments of the proposed FASDNet on the RSSCN dataset.

Condition	KD	FAPM	Auxiliary	OA
1				94.50 ± 0.28
2	✓			95.79 ± 0.21
3	✓	✓		97.12 ± 0.14
4	✓		✓	97.38 ± 0.15
5	✓	✓	✓	97.79 ± 0.14

Table 10. Some ablation experiments of the proposed FASDNet on the AID.

Condition	KD	FAPM	Auxiliary	OA
1				96.34 ± 0.22
2	✓			97.12 ± 0.17
3	✓	✓		97.54 ± 0.14
4	✓		✓	97.40 ± 0.23
5	✓	✓	✓	97.84 ± 0.12

Table 11. Some ablation experiments of the proposed FASDNet on the NWPU dataset.

Condition	KD	FAPM	Auxiliary	OA
1				91.97 ± 0.15
2	✓			94.12 ± 0.10
3	✓	✓		94.78 ± 0.13
4	✓		✓	94.56 ± 0.14
5	✓	✓	✓	94.95 ± 0.12

To verify the effect of temperature hyperparameters on model performance, some ablation experiments are carried out using the proposed FASDNet on the AID with a training ratio of 50%. The temperature hyperparameters are divided into 1, 2, 4, 6, and 8. The experimental results are listed in Table 12. It can be seen from Table 12 that the highest classification accuracy is achieved when T is 4. When T is greater than 4, the network performance becomes worse as the temperature increases. When T is less than 4, as the temperature increases, the network performance continues to improve. Therefore, we use 4 as the temperature hyperparameter when training on other datasets.

Table 12. The experimental results obtained by the proposed FASDNet under the 50% training ratio on the AID when the temperature hyperparameters are 1, 2, 4, 6, and 8.

T	1	2	4	6	8
Accuracy	97.44	97.52	97.66	97.58	97.48

5. Conclusions

In this paper, a novel remote sensing scene classification method is proposed, named FASDNet. It mainly comprises three new designed modules, including the feature augmentation pyramid module, the self-teacher network, and the auxiliary classifier. First, ResNet34 is utilized as the backbone network to learn the multi-layer features of the model. Then, a feature augmentation pyramid module is designed to fuse rich deep semantic information and shallow features step by step through transposed convolution. Next, the backbone network learns the aggregated features through feature distillation, and then Logits distillation is used as a regularization method to reduce the confidence of the network prediction, thereby improving the robustness of the model. Finally, auxiliary branches are added after the feature maps S2 and S3 generated by the backbone network. For the auxiliary branch, the knowledge distillation method is also added, which can provide additional supervision information and help the model to learn more effectively. The proposed FASDNet is verified on four widely used remote sensing classification datasets. The experimental results show that, compared with other advanced classification methods, the proposed FASDNet has significant advantages in the classification of remote sensing scene images.

Although our proposed FASDNet method achieves excellent performance, it still has some shortcomings. In future work, we will integrate the three proposed modules into other advanced networks to improve their generalization. In addition, striving to design specialized networks that are more suitable for remote sensing scene classification is also one of our ongoing efforts.

Author Contributions: Conceptualization, C.S.; data curation, C.S. and M.D.; formal analysis, H.P.; methodology, C.S.; software, M.D.; validation, C.S. and M.D.; writing—original draft, M.D.; 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 (42271409), in part by the Heilongjiang Science Foundation Project of China under Grant LH2021D022, in part by the Leading Talents Project of the State Ethnic Affairs Commission, and in part by the Fundamental Research Funds in Heilongjiang Provincial Universities of China under Grant 145209149.

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. Grigorescu, S.E.; Petkov, N.; Kruizinga, P. Comparison of texture features based on Gabor filters. *IEEE Trans. Image Process.* **2002**, *11*, 1160–1167. [CrossRef] [PubMed]
6. Tang, X.; Jiao, L.; Emery, W.J. SAR image content retrieval based on fuzzy similarity and relevance feedback. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.* **2017**, *10*, 1824–1842. [CrossRef]
7. Mei, S.; Ji, J.; Hou, J.; Li, X.; Du, Q. Learning sensor-specific spatial–spectral features of hyperspectral images via convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 4520–4533. [CrossRef]
8. Jiao, L.; Tang, X.; Hou, B.; Wang, S. SAR images retrieval based on semantic classification and region-based similarity measure for Earth observation. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.* **2015**, *8*, 3876–3891. [CrossRef]
9. Sergyan, S. Color histogram features based image classification in content-based image retrieval systems. In Proceedings of the 6th International Symposium on Applied Machine Intelligence and Informatics, Herlany, Slovakia, 21–22 January 2008; pp. 221–224.
10. Tang, X.; Jiao, L. Fusion similarity-based reranking for SAR image retrieval. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 242–246. [CrossRef]
11. Soltanian-Zadeh, H.; Rafiee-Rad, F.; Pourabdollah-Nejad, S. Comparison of multiwavelet, wavelet, haralick, and shape features for microcalcification classification in mammograms. *Pattern Recognit.* **2004**, *37*, 1973–1986. [CrossRef]
12. Tang, X.; Jiao, L.; Emery, W.J.; Liu, F.; Zhang, D. Two-stage reranking for remote sensing image retrieval. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 5798–5817. [CrossRef]
13. Zhang, L.; Zhou, W.; Jiao, L. Wavelet support vector machine. *IEEE Trans. Syst. Man Cybern. B Cybern.* **2004**, *34*, 34–39. [CrossRef] [PubMed]
14. Friedl, M.A.; Brodley, C.E. Decision tree classification of land cover from remotely sensed data. *Remote Sens. Environ.* **1997**, *61*, 399–409. [CrossRef]
15. Khan; Sohail, A.; Zahoora, U.; Qureshi, A.S. A survey of the recent architectures of deep convolutional neural networks. *Artif. Intell. Rev.* **2020**, *53*, 5455–5516. [CrossRef]
16. 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]
17. Liu, M.; Jiao, L.; Liu, X.; Li, L.; Liu, F.; Yang, S. C-CNN: Contourlet Convolutional Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 2636–2649. [CrossRef]
18. Zhang, D.; Li, N.; Ye, Q. Positional Context Aggregation Network for Remote Sensing Scene Classification. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 943–947. [CrossRef]
19. 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.* **2020**, *31*, 1461–1474. [CrossRef]
20. Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv* **2015**, arXiv:1503.02531.
21. Zhang, R.; Li, X.; Liu, W. Self-distillation with label refining for human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10736–10745.
22. Woloszynski, T.; Ruta, D.; Schaefer, G. Combining multiple classifiers with dynamic classifier selection. *Pattern Recognit.* **2013**, *46*, 3054–3066.
23. Yu, L.; Liu, H.; Motoda, H. Selective ensemble with application to data classification. *Inf. Fusion* **2014**, *15*, 17–26.
24. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, 20–26 June 2005; Volume 1, pp. 886–893.
25. Yang, Y.; Newsam, S. Comparing SIFT descriptors and Gabor texture features for classification of remote sensed imagery. In Proceedings of the 15th IEEE International Conference on Image Processing, San Diego, CA, USA, 12–15 October 2008; pp. 1852–1855.
26. 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 (GIS), San Jose, CA, USA, 2–5 November 2010; pp. 270–279.

27. 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.
28. 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.* **2015**, *54*, 2108–2123. [CrossRef]
29. 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]
30. Li, F.; Feng, R.; Han, W.; Wang, L. High-resolution remote sensing image scene classification via key filter bank based on convolutional neural network. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 8077–8092. [CrossRef]
31. Shi, J.; Liu, W.; Shan, H.; Li, E.; Li, X.; Zhang, L. Remote Sensing Scene Classification Based on Multibranch Fusion Attention Network. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 3001505. [CrossRef]
32. Shi, C.; Zhang, X.; Sun, J.; Wang, L. Remote sensing scene image classification based on dense fusion of multi-level features. *Remote Sens.* **2021**, *13*, 4379. [CrossRef]
33. Deng, P.; Huang, H.; Xu, K. A Deep Neural Network Combined with Context Features for Remote Sensing Scene Classification. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 8000405. [CrossRef]
34. Meng, Q.; Zhao, M.; Zhang, L.; Shi, W.; Su, C.; Bruzzone, L. Multilayer Feature Fusion Network with Spatial Attention and Gated Mechanism for Remote Sensing Scene Classification. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 6510105. [CrossRef]
35. Wang, X.; Wang, S.; Ning, C.; Zhou, H. Enhanced feature pyramid network with deep semantic embedding for remote sensing scene classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 7918–7932. [CrossRef]
36. Zhang, T.; Wang, Z.; Cheng, P.; Xu, G.; Sun, X. DCNNet: A Distributed Convolutional Neural Network for Remote Sensing Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5603618. [CrossRef]
37. Goodfellow, H.J.I.; Bengio, Y.; Courville, A. Deep learning: The MIT Press, 2016, 800 pp, ISBN: 0262035618. *Genet. Program. Evolvable Mach.* **2018**, *19*, 305–307.
38. Kullback, S.; Leibler, R.A. On information and sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86. [CrossRef]
39. Romero, A.; Ballas, N.; Kahou, S.E.; Chassang, A.; Gatta, C.; Bengio, Y. FitNets: Hints for thin deep nets. *arXiv* **2014**, arXiv:1412.6550.
40. Zagoruyko, S.; Komodakis, N. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv* **2016**, arXiv:1612.03928.
41. Park, W.; Kim, D.; Lu, Y.; Cho, M. Relational knowledge distillation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 3967–3976.
42. Heo, B.; Kim, J.; Yun, S.; Park, H.; Kwak, N.; Choi, J.Y. A comprehensive overhaul of feature distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1921–1930.
43. Ahn, S.; Hu, S.X.; Damianou, A.; Lawrence, N.D.; Dai, Z. Variational information distillation for knowledge transfer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 9163–9171.
44. Zhang, L.; Song, J.; Gao, A.; Chen, J.; Bao, C.; Ma, K. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3713–3722.
45. Ji, M.; Shin, S.; Hwang, S.; Park, G.; Moon, I.C. Refine myself by teaching myself: Feature refinement via self-knowledge distillation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 10664–10673.
46. Hu, Y.; Jiang, X.; Liu, X.; Luo, X.; Hu, Y.; Cao, X.; Zhang, B.; Zhang, J. Hierarchical Self-Distilled Feature Learning for Fine-Grained Visual Categorization. *IEEE Trans Neural Netw Learn Syst.* **2021**. [CrossRef] [PubMed]
47. Wei, S.; Luo, Y.; Ma, X.; Ren, P.; Luo, C. MSH-Net: Modality-Shared Hallucination With Joint Adaptation Distillation for Remote Sensing Image Classification Using Missing Modalities. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 4402615. [CrossRef]
48. Hu, Y.; Huang, X.; Luo, X.; Han, J.; Cao, X.; Zhang, J. Variational Self-Distillation for Remote Sensing Scene Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5627313. [CrossRef]
49. Li, D.; Nan, Y.; Liu, Y. Remote Sensing Image Scene Classification Model Based on Dual Knowledge Distillation. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 4514305. [CrossRef]
50. Liu, H.; Qu, Y.; Zhang, L. Multispectral Scene Classification via Cross-Modal Knowledge Distillation. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5409912. [CrossRef]
51. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
52. 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]
53. 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]
54. Cheng, G.; Han, J.; Lu, X. Remote sensing image scene classification: Benchmark and state of the art. *Proc. IEEE* **2017**, *105*, 1865–1883. [CrossRef]

55. Yang, Y.; Tang, X.; Cheung, Y.-M.; Zhang, X.; Jiao, L. SAGN: Semantic-Aware Graph Network for Remote Sensing Scene Classification. *IEEE Trans. Image Process.* **2023**, *32*, 1011–1025. [CrossRef] [PubMed]
56. 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]
57. 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]
58. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Proc. Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1–9. [CrossRef]
59. He, N.; Fang, L.; Li, S.; Plaza, A.; Plaza, J. Remote sensing scene classification using multilayer stacked covariance pooling. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 6899–6910. [CrossRef]
60. Zhang, W.; Tang, P.; Zhao, L. Remote sensing image scene classification using CNN-CapsNet. *Remote Sens.* **2019**, *11*, 494. [CrossRef]
61. 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]
62. Bi, Q.; Qin, K.; Li, Z.; Zhang, H.; Xu, K.; Xia, G.-S. A multiple-instance densely-connected ConvNet for aerial scene classification. *IEEE Trans. Image Process.* **2020**, *29*, 4911–4926. [CrossRef]
63. Wang, X.; Duan, L.; Shi, A.; Zhou, H. Multilevel feature fusion networks with adaptive channel dimensionality reduction for remote sensing scene classification. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 1–5. [CrossRef]
64. Tang, X.; Ma, Q.; Zhang, X.; Liu, F.; Ma, J.; Jiao, L. Attention consistent network for remote sensing scene classification. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.* **2021**, *14*, 2030–2045. [CrossRef]
65. Zhang, G.; Xu, W.; Zhao, W.; Huang, C.; Yk, E.N.; Chen, Y.; Su, J. A multiscale attention network for remote sensing scene images classification. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.* **2021**, *14*, 9530–9545. [CrossRef]
66. Wang, X.; Duan, L.; Ning, C.; Zhou, H. Relation-attention networks for remote sensing scene classification. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.* **2021**, *15*, 422–439. [CrossRef]
67. Xu, K.; Huang, H.; Deng, P.; Li, Y. Deep feature aggregation framework driven by graph convolutional network for scene classification in remote sensing. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 5751–5765. [CrossRef] [PubMed]
68. Tang, X.; Li, M.; Ma, J.; Zhang, X.; Liu, F.; Jiao, L. EMTCAL: Efficient Multiscale Transformer and Cross-Level Attention Learning for Remote Sensing Scene Classification. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5626915. [CrossRef]
69. 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. [CrossRef]
70. Anwer, R.M.; Khan, F.S.; Van De Weijer, J.; Molinier, M.; Laaksonen, J. Binary patterns encoded convolutional neural networks for texture recognition and remote sensing scene classification. *ISPRS J. Photogramm. Remote Sens.* **2018**, *138*, 74–85. [CrossRef]
71. 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]
72. 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]