Embedded Self-Distillation in Compact Multibranch Ensemble Network for Remote Sensing Scene Classification

Qi Zhao, *Member, IEEE*, Yujing Ma, Shuchang Lyu, *Graduate Student Member, IEEE*, and Lijiang Chen

***Abstract*— Remote sensing image classification task is challenging due to the characteristics of complex composition, so different geographic elements in the same image will interfere with each other, resulting in misclassification. To solve this problem, we propose a multibranch ensemble network to enhance the feature representation ability by fusing final output logits and intermediate feature maps. However, simply adding branches will increase the complexity of models and decline the inference efficiency. To reduce the complexity of multibranch network, we make multibranch share more weights and add feature augmentation modules to compensate for the lack of diversity caused by weight sharing. To improve the efficiency of inference, we embed self-distillation (SD) method to transfer knowledge from ensemble network to main branch. Through optimizing with SD, the main branch will have close performance as an ensemble network. In this way, we can cut other branches during inference. In addition, we simplify the process of SD and totally adopt two loss functions to self-distill the logits and feature maps. In this article, we design a compact multibranch ensemble network, which can be trained in an end-to-end manner. Then, we insert an SD method on output logits and feature maps. Our proposed architecture (ESD-MBENet) performs strongly on classification accuracy with compact design. Extensive experiments are applied on three benchmark remote sensing datasets, AID, NWPU-RESISC45, and UC-Merced with three classic baseline models, VGG16, ResNet50, and DenseNet121. Results prove that ESD-MBENet can achieve better accuracy than previous state-of-the-art complex deep learning models. Moreover, abundant visualization analyses make our method more convincing and interpretable.**

***Index Terms*— Multibranch ensemble network, network pruning, remote sensing scene classification, self-distillation (SD).**

I. Introduction

REMOTE sensing scene classification is a recent popular task in practical application. It reveals the geographical characteristics, such as land utilization and vegetation coverage [1]. With the progress of RS scene classification, research on local land planning, tree planting, and afforestation can be realized more intelligent. In recent years, with the rapid development of deep learning technology [2], [3], methods for improving the remote sensing scene classification accuracy have been continuously proposed.

In remote sensing scene classification task, remote sensing images always have complex composition, large resolution, and large geographic coverage area. Therefore, the major problem is the interference from different characteristics of different geographical elements. It leads to misclassification. To solve this problem, previous works focus on fusing multilevel features to enhance the model's ability to represent complex structural information. The multilevel method cannot provide the diversity of feature augmentation. In this article, we integrate the ensemble learning method into CNN modules to construct a multibranch ensemble network. Toward different geographical elements in remote sensing images, we use more branches to provide sufficient representation. Each branch is added a feature augmentation module, and the whole network can provide the feature augmentation diversity. Through fusing the final logits of different branches, we combine all perspectives together and obtain a more convincing prediction. As shown in Fig. 1, we use the main branch and subbranch to construct our ensemble network.

Although the ensemble network is quite effective in dealing with the above-mentioned problem, the large memory and computation cost of multibranch structure cannot be ignored. Especially on embedded devices or mobile devices, multibranch models are cumbersome and hard to deploy. To construct a lighter yet high-efficient multibranch network, we explore the weight-sharing potential of multibranch networks. Sharing more weights will lead to the lack of diversity, and therefore, we add feature augmentation modules to multibranch networks. Then, we embed the self-distillation (SD) method in a multibranch ensemble network. The overview of ESD-MBENet is shown in Fig. 1.

To intuitively lighten the multibranch structure, we design one subbranch in ESD-MBENet-v1. Then, we split two branches and connect blocks in a zigzag manner. Designing like this, we can generate a multibranch network with only two branches. As shown in Fig. 1(a), we split the main branch into three blocks and subbranch into two blocks. The images are first fed into the first block of main branch ("main-block1"). The output feature maps then pass through "sub-block1∼sub-block2" and "main-block2." The output feature maps of "main-block2" will then pass through "sub-block2" and "main-block3." Finally, we obtain three output logits from three paths. If we set more split points, we can get

Manuscript received April 1, 2021; revised June 21, 2021 and September 10, 2021; accepted October 23, 2021. Date of publication November 8, 2021; date of current version February 15, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62072021 and in part by the Fundamental Research Funds for the Central Universities under Grant YWF-21-BJ-J-534. *(Corresponding author: Lijiang Chen.)*

The authors are with the Department of Electronics and Information Engineering, Beihang University, Beijing 100191, China (e-mail: zhaoqi@buaa.edu.cn; zy1902407@buaa.edu.cn; lyushuchang@buaa.edu.cn; chenlijiang@buaa.edu.cn).

Digital Object Identifier 10.1109/TGRS.2021.3126770

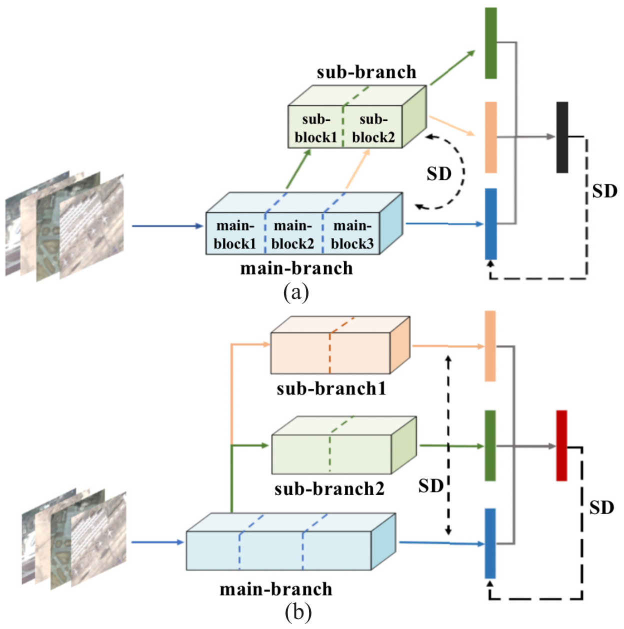


Fig. 1. Overview of ESD-MBENet structure. “SD” denotes self-distillation. (a) ESD-MBENet-v1. (b) ESD-MBENet-v2.

more diverse output logits. Besides ESD-MBENet-v1, we also design ESD-MBENet-v2 as in Fig. 1(b) to form the multibranch ensemble network because if we set first split point in deeper layer, the number of branches of ESD-MBENet-v1 will be reduced. In this way, the expression of multiperspective will be reduced and the classification accuracy will be affected. ESD-MBENet-v2 can flexibly construct multiple branches without being affected by the backward movement of split points. Essentially, we construct weight-sharing blocks in ESD-MBENet and maximally explore the representation ability of them.

Even though we use weight-sharing blocks to simplify multibranch network, ensemble network is also cumbersome during inference. To cast off subbranch during inference, we introduce an SD method. SD is carried out through the mutual learning method in [4]. It uses multiple loss functions to optimize the network, and it takes more time to adjust the weights of multiple loss functions. We make improvements to the optimization process. We totally use two loss functions when distilling the logits and feature maps. The weights adjustment of the loss functions is more convenient and concise. When the main branch can show comparable performance as the ensemble network, we can prune the subbranch and only adopt the main branch during inference. In ESD-MBENet, we embed the SD method into the ensemble network. Specifically, the final ensemble logits that are fused together by logits of every branch will be served as soft label to distill knowledge to the main branch. In intermediate feature maps, the ensemble feature map is used to guide the feature map of main branch. After optimizing with SD method, the main branch has close performance as an ensemble network. Therefore, we can prune subbranch and only use the main branch as an inference model.

Compared with previous single-branch networks [5]–[7], the inference speed of our proposed model does not become slower. Compared with previous multibranch networks [8]–[11], our proposed ESD-MBENet achieves better performance with more compact structure by enhancing the capability of main branch to learn more and better knowledge. Extensive experiments using VGG16 [12], ResNet50 [13], and DenseNet121 [14] as baseline models on remote sensing benchmark datasets (AID [15], NWPU-RESISC45 [16], and UC-Merced [17]) prove the effectiveness of our proposed ESD-MBENet. Classification results on ESD-MBENet surpass previous deep learning methods and reach the state-of-the-art level. To show the generalization of our model, we also conduct experiments on Million-AID.[1] The result is also encouraging. Our main contributions can be summarized as follows.

1) We propose a more compact yet efficient multibranch ensemble network, explore the weight-sharing potential of multibranch networks, and add the feature augmentation modules to compensate for the lack of diversity to overcome the interference of different geographical elements in remote sensing images.
2) We insert the SD method in the ensemble network to distill knowledge to the main branch, which can further simplify the inference network and reduce parameters.
3) The design of the knowledge distillation process is more concise. We totally adopt two loss functions to complete the knowledge distillation. It reduces the complexity of the distillation process.

The rest of this article is organized as follows. Section II briefly depicts the related works. Section III introduces the method of our proposed in detail. Section IV describes the experiment process, results, and visualization effects. Section V draws the conclusions.

II. Related Works

A. Remote Sensing Image Classification

With the development of artificial intelligence, remote sensing scene classification methods have transitioned from handcrafted feature extraction to deep learning feature extraction.

Handcrafted feature extraction is first applied in remote sensing image classification task. Typical handcrafted feature extraction methods are SIFT [18]–[20], HOG [20]–[23], and so on. Low-level information of remote sensing images can be extracted by these methods. Stehling *et al.* [24], dos Santos *et al.* [25], and Penatti *et al.* [26] proposed the border-interior pixel classification (BIC), which can calculate the border and interior pixels color histograms. Later, principal component analysis (PCA), K-means clustering, bag-of-visual-words (BoVW) [17], [27]–[29], and sparse encoding [30] are proposed to extract mid-level information of the images for remote sensing scene classification.

Due to the appearance of backpropagation neural network, deep learning has developed quickly. Simonyan and Zisserman [12], He *et al.* [13], Huang *et al.* [14], and

[1] https://rs.sensetime.com/competition/index.html

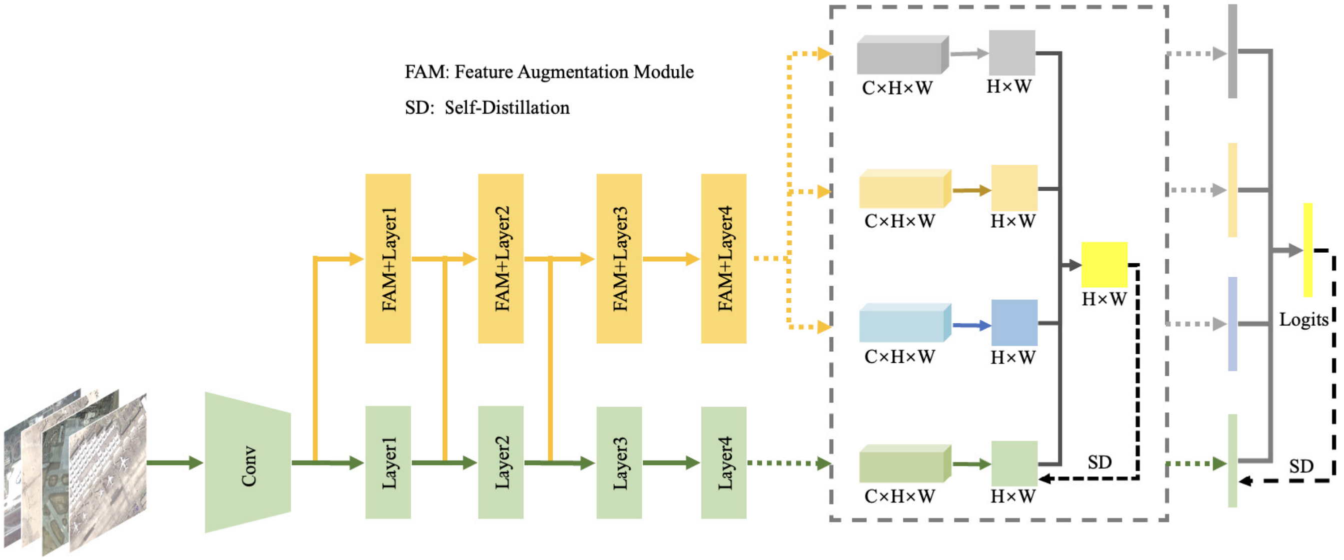


Fig. 2. Overall framework of the ESD-MBENet-v1. The subbranch and the main branch use the same structure. Compared with the main branch, the subbranch adds a feature augmentation module, such as SE or CAM or dropout to diversify the network outputs. We use SD on the final output logits and intermediate feature maps. In addition, hard labels are also used to optimize the network. To reduce the complexity of the model, only the main branch is used for inference.

Krizhevsky *et al.* [3] achieved amazing improvement in the image classification task. Based on these baseline models, RS image classification technologies have improved rapidly [31]–[33]. Zou *et al.* [34] proposed a deep-learning-based feature selection method to achieve feature abstraction of the remote sensing images. Liu and Yang [35] classified unlabeled remote sensing images, which improves the speed and accuracy of classification compared to traditional machine learning algorithms. Our proposed ESD-MBENet also uses the deep learning method for remote sensing scene classification.

B. Multibranch Network

A multibranch network can obtain abundant information from multiple perspectives of the input images, which helps the network to have a more comprehensive representation of the images and improves the generalization of the classifier. In a remote sensing scene classification task, many researchers explore the potential of multibranch networks by fusing features of different branches [8], [9], [36]. Teffahi and Yao [37], Ji *et al.* [38], and Wang *et al.* [39] achieved the feature fusion method by extracting multiple spectral and spatial features and concatenating them, which improves the accuracy of remote sensing image classification. Tan *et al.* [8] used a multibranch lightweight network to extract image features and built a graph model based on the learned features. Liu *et al.* [9] adopted fine-grained and coarse branches to obtain the features in images. Xi *et al.* [40] proposed an ensemble deep kernel extreme learning machine and utilized the strategies of decision fusion and weighted output layer fusion for efficient hyperspectral image (HSI) classification. Xi *et al.* [41] extracted the multistream features using multidirection samples to achieve the HSI classification.

Our proposed ESD-MBENet uses fewer modules and more weight-sharing blocks to build a multibranch network, which can obtain multiview information from multibranch so that the network can have more references when making final decisions.

C. Knowledge Distillation and SD

Knowledge distillation is a concept proposed by Hinton *et al.* [42]. The main purpose of knowledge distillation and SD is model compression. Knowledge distillation aims to guide simple and relatively poor student network learning from a complex but superior teacher network [43], [44]. The student network learns how the teacher network learns to improve its distinguishing performance for remote sensing image classification. SD mainly distills from its own network, without the assistance of external networks or models. The weighted combination of multiple teacher networks is proposed to guide students to learn from it in [45]. A knowledge distillation framework is proposed in [46], which makes the output of the student and teacher models match. The discriminative modality distillation approach is introduced in [47], the teacher is trained on multimodal data, and then, the student model learns from the teacher model to improve the performance of the remote sensing image classifications. To address the problem of network overfitting due to noisy data, a novel noisy label distillation method (NLD) is proposed in [48].

Regarding the SD method, there is little research in the RS image classification task. We propose an end-to-end compact multibranch ensemble network ESD-MBENet that uses SD to improve the main-branch performance.

III. Proposed Network

A. Overview of ESD-MBENet

To overcome the interference of different geographic elements in remote sensing images, we propose two versions of

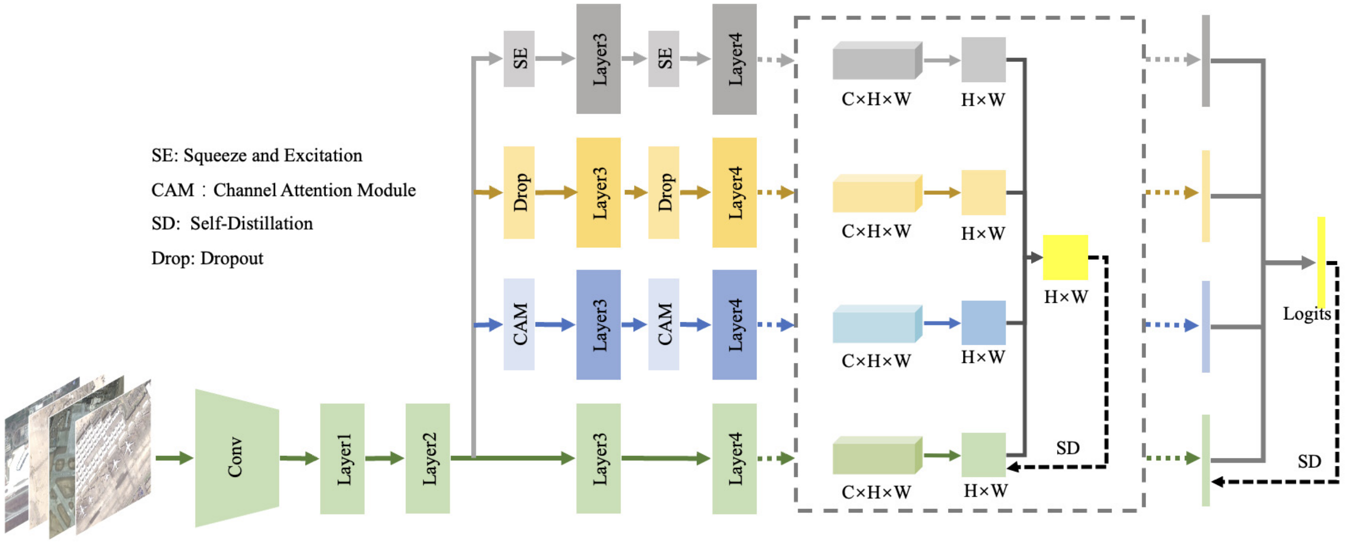
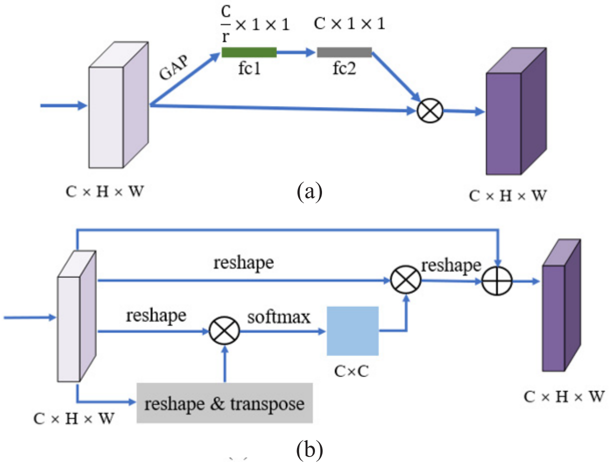


Fig. 3. Framework of the ESD-MBENet-v2. When the split points keep moving backward in ESD-MBENet-v1, the number of branches decreases. In order to maintain the diversity of the network, new branches are added as compensation. The process of training optimization and inference is the same as that of ESD-MBENet-v1.



(a)

(b)

Fig. 4. Structure of (a) SE and (b) CAM module.

ESD-MBENet. Fig. 2 shows the structure of ESD-MBENet-v1. For the purpose of using fewer modules and constructing a multibranch network, we set multiple split points in the network. First, we design a network of two branches, the main branch and the subbranch. The main branch and the subbranch use the same backbone. To resist the loss of multibranch diversity caused by weight sharing, feature augmentation modules, such as SE or CAM or dropout, are added to the subbranch. If the subbranch shares the first "conv" with the main branch, we can set split points in the main branch "layer1" and "layer2" and then send the main-branch feature maps to the subbranch network in the position of the split points. Of course, the main branch and the subbranch can also share "layer1," and the corresponding split points will move backward. In the training phase, multibranch output results can be obtained after fusing the output logits. To simplify the model complexity, we only use the main branch for inference.

As the split points move backward, more and more weights are shared by the network, and the number of branches reduces. It will cause the lack of network diversity and the deterioration of network performance. To solve the problem, we propose ESD-MBENet-v2. The network framework is shown in Fig. 3. The same structure is used in four branches, and SE, CAM, and dropout modules are added to each subbranch. There is no weight-sharing blocks in each subbranch. ESD-MBENet-v2 is not limited by the number of branches. At the corresponding split point, we can add multiple branches at will. Taking the number of parameters, model complexity, and performance improvement into account, we choose four branches in experiments. Similar to ESD-MBENet-v1, multiple branches are used for training, and the main branch is used for inference.

In short, ESD-MBENet-v1 uses fewer modules to construct multibranch structures, and ESD-MBENet-v2 can construct multibranch networks more flexibly. Both ESD-MBENet-v1 and ESD-MBENet-v2 consider using as few modules as possible to build multiple branches, that is, sharing as many weights as possible. Through experimental verification, the two versions of ESD-MBENet we proposed both have better remote sensing image classification performance than the previous deep learning methods.

B. Feature Augmentation Modules

The diversity of multibranch networks is very important for feature fusion. In ESD-MBENet, if the subbranch and main branch are exactly the same, it may lack diversity for image feature extraction, and it is impossible to describe image features from multiple perspectives. Compared with tasks such as image segmentation, image classification does not require so much attention between pixels. Therefore, we consider the enhancement of the attention between feature map channels.

We use SE and CAM modules proposed in SENet [49] and DANet [50] to add attention to the feature maps.

The structure of SE and CAM modules is shown in Fig. 4. The idea of SE is to take the obtained feature maps and pass them through the global average pooling layer, two fully connected layers, and the sigmoid function as (1) and then multiply it with the original input feature maps. The scale factor is 4, that is, we set "r" equal to 4 in Fig. 4(a). In this way, the global information of the images can be integrated into the feature maps, which improves the sensitivity of the network to the channel and makes the feature maps containing richer information

$$S(x) = \frac{1}{1+e^{-x}}. \tag{1}$$

The CAM module emphasizes the interdependent channel mappings by integrating the relevant features between all channel mappings. After the input feature maps are reshaped, transposed, and multiplied, the matrix of C × C is obtained, which is the channel attention map. Then, multiply the matrix, which should pass through the softmax layer with the input feature maps after reshaping. Finally, reshape the feature maps and add the feature maps with the original input feature maps. In this way, the attention mechanism is added to the feature maps. The specific process of SE and CAM module is consistent with [49] and [50].

In addition, for remote sensing image classification, one image corresponds to one category, but not all pixel values in the image can provide useful information for the classification results, and even some pixels may interfere with the judgment for the image. Therefore, we design to use the dropout module, and the probability of a random drop is 0.2. This helps a lot to improve the network generalization performance.

These three modules are all independent modules, which can be embedded anywhere in the network without affecting other structure of the network and have strong flexibility.

C. Multibranch Ensemble

To solve the interference between different geographic elements in remote sensing images, we propose ESD-MBENet. ESD-MBENet-v1 constructs multiple branches from two branches (main branch and subbranch) networks by setting different split points and then fuses the features in multiple branches. To ensure the feature diversity using the weight-sharing multibranch, the subbranch adds the feature augmentation module in front of each layer, such as SE or CAM or dropout. Assume that the main branch and the subbranch share the first "conv," we can build four branches as in Fig. 2. The construction of multibranch is as follows. The first branch is the main branch, and the second branch is the subbranch. The third branch is the feature maps obtained after "layer1" of the main branch passes through the rest of the subbranch, and the fourth branch is the feature maps after "layer2" of the main branch passes through the rest of the subbranch. Suppose that the main branch is divided into "conv," "main-layer1," "main-layer2," "main-layer3," and "main-layer4," mathematically expressed as f_0, f_1, f_2, f_3, and f_4, and the subbranch is divided into "sub-layer1," "sub-layer2," "sub-layer3," and "sub-layer4," mathematically expressed as g_1, g_2, g_3, and g_4. The four branches b_1, b_2, b_3, and b_4 divided by ESD-MBENet-v1 are $b_1 = f_4 f_3 f_2 f_1 f_0(x), b_2 = g_4 g_3 g_2 g_1 f_0(x), b_3 = g_4 g_3 g_2 f_1 f_0(x)$, and $b_4 = g_4 g_3 f_2 f_1 f_0(x)$, in which $\boldsymbol{x}$ is a batch of images. The ESD-MBENet-v1 ensemble multibranch algorithm is shown in Algorithm 1.

Algorithm 1 ESD-MBENet-v1 Multibranch Algorithm

```
Input: A batch of images $\boldsymbol{x}$. Define main-branch blocks function as a list $[f_0, f_1, \ldots, f_m]$, the sub-branch blocks function as a list $[g_1, g_2, \ldots, g_m]$, the number of blocks in main-branch $m$, the split points after the corresponding layer as a list $sp = [0, 1, 2, \ldots, n]$, $n \leq m-1$, main-branch fully connected layer function the $f_c$, $k^{\text{th}}$ sub-branch fully connected layer function $f_{c_k}$, the number of all branches $N$.
Output: the output logits list $[\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_N]$
$\boldsymbol{v}_0 = f_c f_m f_{m-1}, \ldots, f_0(x)$
if $i$ is the first split point and $i \in sp$ then
    for $k = i$ to $n$ do
        $\boldsymbol{v_k} = f_{c_k}\ g_m g_{m-1}, \ldots, g_{k+1} f_k f_{k-1}, \ldots, f_i f_{i-1}, \ldots, f_1 f_0(x)$
    end for
end if
```

When the split points of the ESD-MBENet-v1 gradually move backward, the weight-sharing blocks of multibranch continue to increase, and the number of branches that can be divided decreases, which is very unfriendly to the extraction of image features. Therefore, we propose ESD-MBENet-v2. Compared with ESD-MBENet-v1, ESD-MBENet-v2 can add branches flexibly and will not be affected by the movement of split points. Suppose that the split point of ESD-MBENet-v2 is set to "layer2," and we can build four branches, as shown in Fig. 3. To ensure the diversity of multibranch output features, although the same structure is used in four branches, each subbranch adds a different feature augmentation module. The main branch does not add any additional module, and the subbranch1 adds SE, sub-branch2 adds CAM, and sub-branch3 adds dropout. Assume that the main branch is divided into "conv," "main-layer1," "main-layer2," "main-layer3," and "main-layer4," mathematically expressed as f_0, f_1, f_2, f_3, and f_4. The subbranches are denoted as l_1, l_2, and l_3. The four branches b_1, b_2, b_3, and b_4 divided by ESD-MBENet-v2 are $b_1 = f_4 f_3 f_2 f_1 f_0(x), b_2 = l_0 f_2 f_1 f_0(x), b_3 = l_1 f_2 f_1 f_0(x)$, and $b_4 = l_2 f_2 f_1 f_0(x)$, in which $\boldsymbol{x}$ is a batch of images. The ESD-MBENet-v2 ensemble multibranch algorithm is shown in Algorithm 2.

D. Self-Distillation

Using a compact multibranch ensemble network, the accuracy of ESD-MBENet for remote sensing image classification can be significantly improved. To shorten the time of inference and simplify the complexity of ESD-MBENet, we use SD for ESD-MBENet to improve the inference performance of the main branch. Thus, we can prune all the subbranches and only use the main branch for inference.

TABLE I

DETAILED STRUCTURE OF BASELINE MODELS [12]–[14]. THE LAYER NAME CORRESPONDS TO THE LAYERS IN FIGS. 2 AND 3. WE APPLY VGG16, RESNET50, AND DENSENET121 AS BASELINE MODELS FOR REMOTE SENSING SCENE CLASSIFICATION. FOR VGG16, FC LAYER CHANNELS NUMBER IS MODIFIED TO 512

Layer Name	Network		
	VGG16	ResNet50	DenseNet121
Conv1	conv1-x × 2	7 × 7 conv, stride 2	
Pool1	2×2 max pool, stride 2	3 × 3 max pool, stride 2	
Layer1	conv2-x × 2	Bottleneck × 3	DenseBlock × 6
Layer2	conv3-x × 3	Bottleneck × 4	Transition,DenseBlock × 12
Layer3	conv4-x × 3	Bottleneck × 6	Transition,DenseBlock × 24
Layer4	conv5-x × 3	Bottleneck × 3	Transition,DenseBlock × 16
Pool2	average pool		
FC	FC1 × 2, 512 × num_cls	2048 × num_cls	1024 × num_cls

TABLE II

COMPARISON OF CLASSIFICATION RESULTS (%) ON AID, NWPU-RESISC45, AND UC-MERCED. "TR" DENOTES THE TRAINING RATE

Networks	Backbone	AID		NWPU-RESISC45		UC-Merced
		tr = 20%	tr = 50%	tr = 10%	tr = 20%	tr = 80%
Pre-trained VGGNet-16+SVM [51]	VGG16	89.33±0.23	96.04±0.13	87.15±0.45	90.36±0.18	97.14±0.10
Fusion by Addition [52]	VGG16	-	91.87±0.36	-	-	97.42±1.79
Triplet networks [53]	VGG16	-	-	-	-	97.99±0.53
ARCNet [54]	VGG16	88.75±0.40	93.10±0.55	-	-	99.12±0.40
HW-CNN [55]	-	-	96.98±0.33	-	94.38±0.16	-
DCNN [56]	VGG16	90.82±0.16	96.89±0.10	89.22±0.50	91.89±0.22	98.93±0.10
MSCP [57]	VGG16	92.21±0.17	96.56±0.18	88.07±0.18	90.81±0.13	98.40±0.34
RTN [58]	VGG16	92.44	-	89.90	92.71	98.96
MG-CAP [11]	VGG16	93.34±0.18	96.12±0.12	**90.83 ± 0.12**	92.95±0.13	99.0±0.10
SCCov [59]	VGG16	93.12±0.25	96.10±0.16	89.30±0.35	92.10±0.25	99.05±0.25
Hydra[60]	DenseNet121	-	-	92.44±0.34	94.51±0.21	-
KFBNet [51]	VGG16	94.27±0.02	97.19±0.07	90.27±0.02	92.54±0.03	**99.76 ± 0.24**
KFBNet [51]	DenseNet121	95.50±0.27	97.40±0.10	93.08±0.14	95.11±0.10	**99.88 ± 0.12**
ESD-MBENet-v1(ours)	VGG16	94.10±0.13	97.15±0.21	**90.29 ± 0.11**	**93.48 ± 0.06**	**99.81 ± 0.10**
ESD-MBENet-v2(ours)	VGG16	94.12±0.24	**97.3 ± 0.08**	90.25±0.21	**93.42 ± 0.15**	**99.86 ± 0.12**
ESD-MBENet-v1(ours)	ResNet50	**96.0 ± 0.15**	**98.54 ± 0.17**	**92.5 ± 0.22**	**95.58 ± 0.08**	**99.81 ± 0.10**
ESD-MBENet-v2(ours)	ResNet50	**95.81 ± 0.24**	**98.66 ± 0.2**	**93.03 ± 0.11**	**95.24 ± 0.23**	**99.86 ± 0.12**
ESD-MBENet-v1(ours)	DenseNet121	**96.2 ± 0.15**	**98.85 ± 0.13**	**93.24 ± 0.15**	**95.5 ± 0.09**	**99.86 ± 0.12**
ESD-MBENet-v2(ours)	DenseNet121	**96.39 ± 0.21**	**98.4 ± 0.23**	93.05±0.18	**95.36 ± 0.14**	**99.81 ± 0.10**

In ESD-MBENet, the SD includes the final output logits distillation and feature maps distillation. With respect to the output logits distillation, we can get the ensemble output logits from the multiple branches as (2) and then let them pass through the softmax function as (3), which can be as the teacher. The output logits of the main branch can be as the student. The Kullback–Leibler (KL) loss is used to optimize it. The SD output logits algorithm is shown in Algorithm 3

$$\boldsymbol{v}_t = \frac{1}{N} * \sum_{k=1}^{N} \boldsymbol{v}_k \tag{2}$$

$$p(\hat{y} = y_i|\boldsymbol{x}) = \frac{e^{v_t^i}}{\sum_{j=1}^{M} e^{v_t^j}} \tag{3}$$

where v_k is the logits of the kth branch, N is the number of all branches, M is the number of all classes, and $\boldsymbol{v}_t = \{\boldsymbol{v}_t^0, \boldsymbol{v}_t^1, \ldots, \boldsymbol{v}_t^M\}$.

With respect to the feature maps distillation, the output feature maps ($C \times H \times W$) of the multiple branches after "layer4" are added along the channel direction to obtain new feature maps ($H \times W$) as follows:

$$\boldsymbol{g}_k^{H*W} = \sum_{c=1}^{C} \boldsymbol{f}_{k_c}^{H*W}, \quad k = 1, 2, \ldots, N \tag{4}$$

where $\boldsymbol{f}_{k_c}^{H*W}$ is the kth branch feature map in the cth channel, C is the total number of channels, and $\boldsymbol{g}_k^{H*W}$ is the kth branch

Algorithm 2 ESD-MBENet-v2 Multibranch Algorithm

```
Input: A batch of input images $\boldsymbol{x}$. Define main-branch blocks
    function as a list $[f_0, f_1, \ldots, f_m]$, blocks number in main-
    branch $m$, split points after the corresponding layer as a
    list $sp = [0, 1, \ldots, n]$, $n \leq m-1$, the main-branch fully
    connected layer $f_c$, the $k^{\text{th}}$ sub-branch fully connected layer
    $f_{c_k}$, the function of the $k^{\text{th}}$ sub-branch $l_k$, the total branches
    number $N$.
Output: the output logits list $[\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_N]$
$\boldsymbol{v}_0 = f_c f_m f_{m-1}, \ldots, f_1 f_0(x)$
if $i$ is the split point and $i \in sp$ then
    for $k = 1$ to $N-1$ do
        $\boldsymbol{v}_k = f_{c_k} l_k f_i f_{i-1}, \ldots, f_1 f_0(x)$
    end for
end if
```

Algorithm 3 SD Output Logits Algorithm

```
Input: The total number of branches $N$, the $k^{\text{th}}$ branch output
    logits $\boldsymbol{v}_k$, main-branch output logits $\boldsymbol{v}_s$.
Output: the self-distillation loss $L_{em}^{KL}$ between ensemble out-
    put logits and main-branch output logits
Compute the $\boldsymbol{v}_t$ using Eq. 2 and Eq. 3
Compute the $L_{em}^{KL}$, $\boldsymbol{p}_e = \boldsymbol{v}_t$, $\boldsymbol{p}_m = \boldsymbol{v}_s$ using Eq. 13
```

new feature map

$$\boldsymbol{g}_k^{H*W} = \begin{bmatrix} g_{k,11} & \cdots & g_{k,1W} \\ \vdots & \ddots & \vdots \\ g_{k,H1} & \cdots & g_{k,HW} \end{bmatrix} \tag{5}$$

$$x_a = \frac{1}{H*W} * \sum_{i=1}^{H}\sum_{j=1}^{W} g_{k,ij} \tag{6}$$

$$x_s = \sqrt{\frac{1}{H*W} * \sum\nolimits_{i=1}^{H}\sum\nolimits_{j=1}^{W}\left(g_{k,ij} - x_a\right)^2} \tag{7}$$

$$F_{k,ij} = (g_{k,ij} - x_a)/x_s \tag{8}$$

$$\boldsymbol{F}_k^{H*W} = \begin{bmatrix} F_{k,11} & \cdots & F_{k,1W} \\ \vdots & \ddots & \vdots \\ F_{k,H1} & \cdots & F_{k,HW} \end{bmatrix}. \tag{9}$$

Then, normalize each of the feature maps ($H \times W$). The normalization process is each pixel value $g_{k,ij}$ on the feature map subtracts the mean value x_a and then divides the standard deviation x_s as (6)–(8). Then, we can obtain normalized feature maps $\boldsymbol{F}_k^{H*W}, k = 1, 2, \ldots, N$. We average these feature maps to obtain a teacher feature map $\boldsymbol{F}_e^{H*W}$ as (10). The student is the main-branch feature map after normalization $\boldsymbol{F}_m^{H*W}$. The distribution of the feature maps directly affects the output logits. If the main-branch feature map (student) can learn the equivalent knowledge to the multibranch ensemble feature map (teacher), the overall performance of the main branch will be improved. The mean square error (MSE) loss is used to optimize it. Compared with the feature map mutual learning mechanism proposed by previous researchers, we propose a simple feature map learning mechanism, as shown in Algorithm 4

$$\boldsymbol{F}_e^{H*W} = \frac{1}{N} * \sum_{k=1}^{N} \boldsymbol{F}_k^{H*W}. \tag{10}$$

E. Backward Propagation of ESD-MBENet

The ESD-MBENet is optimized by reducing the total loss objective function as (11). We use a cross-entropy loss as (12), KL divergence loss as (13), and MSE loss as (14) to make ESD-MBENet converge quickly

$$L_{\text{total}} = \sum_{k=1}^{N}(\alpha_k * L_{\text{ce}_k}) + \beta * L_{em}^{\text{KL}} + \lambda * L_{em}^{\text{MSE}} \tag{11}$$

$$L_{\text{ce}_k} = -\sum_{j=1}^{M}(y_{kj} * \log(p_{kj})) \tag{12}$$

$$\begin{aligned} L_{em}^{\text{KL}} &= L^{\text{KL}}(\boldsymbol{p}_e \| \boldsymbol{p}_m) = -\frac{1}{M} * \sum_{i=1}^{M} \boldsymbol{p}(\boldsymbol{x}_{ei}) \log \frac{\boldsymbol{p}(\boldsymbol{x}_{mi})}{\boldsymbol{p}(\boldsymbol{x}_{ei})} \\ &= H(\boldsymbol{p_e}, \boldsymbol{p_m}) - H(\boldsymbol{p_e}) \end{aligned} \tag{13}$$

$$\begin{aligned} L_{em}^{\text{MSE}} &= \frac{1}{T} * \sum_{t=1}^{T} L^{\text{MSE}}\big(\boldsymbol{F_e}(\boldsymbol{x}_t) \| \boldsymbol{F_m}(\boldsymbol{x}_t)\big) \\ &= \frac{1}{T} * \sum_{t=1}^{T}\sum_{i=1}^{H}\sum_{j=1}^{W}\big((f_e(\boldsymbol{x}_t))_{ij} - (f_m(\boldsymbol{x}_t))_{ij}\big)^2 \end{aligned} \tag{14}$$

where L_{ce_k} represents the cross-entropy loss function, and the cross-entropy loss is obtained from each branch. L_{em}^{KL} represents the KL loss between ensemble output logits $\boldsymbol{p}_e$ and the main-branch output logits $\boldsymbol{p}_m$ and L_{em}^{MSE} is the MSE loss between the ensemble feature map $\boldsymbol{F}_e(\boldsymbol{x})$ and the main-branch feature map $\boldsymbol{F}_m(\boldsymbol{x})$. α, β, and λ are the weight coefficients of each loss function. The ESD-MBENet network uses only two loss functions in the distillation process of output logits and feature maps, regardless of the number of subbranches. This greatly simplifies the process of tuning and optimization.

IV. Experiments

A. Datasets

We use three remote sensing datasets (AID, NWPU-RESISC45, and UC-Merced) to verify the effectiveness of ESD-MBENet. The AID dataset has 10 000 remote sensing images, including 30 categories, and the image size is 600 × 600. There are 220–420 images per category. The NWPU-RESISC45 dataset has 31 500 images, including 45 categories, each category has 700 images, and the image size is 256 × 256. The UC-Merced dataset has only 2100 images, including 21 categories. The image size is 256 × 256, and each category has 100 images. In addition, to verify the generalization of ESD-MBENet, we also select the Million-AID dataset to show the superiority of our method.

B. Implementation Details

We select three backbones (VGG16, ResNet50, and DenseNet121) on remote sensing datasets (AID, NWPU-RESISC45, and UC-Merced) to experiment. The detailed

Algorithm 4 SD Feature Maps Algorithm

```
Input: Feature maps from the $k^{\text{th}}$ branch $\boldsymbol{f}_k^{C*H*W}$. Define the
    $k^{\text{th}}$ branch the $c^{\text{th}}$ channel feature map $\boldsymbol{f}_{k_c}^{H*W}$, the total
    branches number $N$, the value in row $i$ and column $j$
    of the new feature map $g_{k,ij}$, after normalization the $k^{\text{th}}$
    branch feature map $\boldsymbol{F}_k^{H*W}$, the main-branch feature map
    after normalization $\boldsymbol{F}_m^{H*W}$.
Output: the self-distillation loss $L_{em}^{\text{MSE}}$ between ensemble feature map and main-branch feature map
1: Compute the new feature map of the $k^{\text{th}}$ branch using Eq. 4.
2: Compute the average value of the $k^{\text{th}}$ branch new feature map using Eq. 6.
3: Compute the standard deviation value of the $k^{\text{th}}$ branch new feature map using Eq. 7.
4: Compute the feature map of the $k^{\text{th}}$ branch after normalization using Eq. 8 and Eq. 9.
5: Compute the ensemble feature map of the all branches after normalization using Eq. 10.
6: Compute the $L_{em}^{\text{MSE}}$, $\boldsymbol{F}_e(\boldsymbol{x}) = \boldsymbol{F}_e^{H*W}, \boldsymbol{F}_m(\boldsymbol{x}) = \boldsymbol{F}_m^{H*W}$ using Eq. 14.
```

TABLE III

Results of ESD-MBENet-v1/v2 on the Million-AID Dataset.“tr” Denotes the Training Rate

Methods(backbone)	tr = 10%	tr = 20%
VGG16	92.1±0.23	94.63±0.15
ESD-MBENet-v1(VGG16)	92.76±0.22	95.03±0.27
ESD-MBENet-v2(VGG16)	92.47±0.23	95.02±0.2
ResNet50	94.74±0.12	96.45±0.11
ESD-MBENet-v1(ResNet50)	95.2±0.16	96.85±0.24
ESD-MBENet-v2(ResNet50)	95.26±0.13	97.0±0.13
DenseNet121	95.03±0.24	96.43±0.25
ESD-MBENet-v1(DenseNet121)	95.39±0.2	97.06±0.22
ESD-MBENet-v2(DenseNet121)	95.52±0.13	97.0±0.29

TABLE IV

Compared the Flops, Parameters, and FPS of Our Proposed Model With the Baseline Method During Training and Inference

Methods(backbone)	training		inference		
	FLOPs	parameters	FLOPs	parameters	FPS
VGG16	15.38G	15.75M	15.38G	15.75M	386
ESD-MBENet-v1(VGG16)	45.47G	31.6M	15.38G	15.75M	386
ESD-MBENet-v2(VGG16)	45.93G	60.46M	15.38G	15.75M	386
ResNet50	4.12G	25.56M	4.12G	25.56M	579
ESD-MBENet-v1(ResNet50)	13.72G	47.25M	4.12G	25.56M	579
ESD-MBENet-v2(ResNet50)	10.97G	92.49M	4.12G	25.56M	579
DenseNet121	2.88G	7.01M	2.88G	7.01M	491
ESD-MBENet-v1(DenseNet121)	8.01G	14.11M	2.88G	7.01M	491
ESD-MBENet-v2(DenseNet121)	5.19G	23.9M	2.88G	7.01M	491

structure of the backbone is shown in Table I. Since most previous notable methods use VGG16 as the backbone, we also select the VGG16 network. ResNet50 [13] has superior performance than VGG16 and is widely used. Therefore, we select ResNet50 as one of our backbones. For deeper networks, we select DenseNet121 following the setting of previous state-of-the-art methods of KFBNet [51]. The optimizer used in the experiment is the stochastic gradient descent (SGD) with momentum that is set to 0.9. The image is resized to 256×256 during the training of the AID dataset. The models are trained for 100 epochs in each experiment on the AID dataset, and the learning rate drops ten times at epochs 40, 70, and 90. The training images of the NWPU-RESISC45 and UC-Merced are resized to 224×224. The models are trained for 120 epochs in each experiment on the NWPU-RESISC45 dataset, and the learning rate drops ten times at epochs 70, 90, and 110. ImageNet pretrained parameters are loaded in each layer when training. The code is implemented using the Pytorch framework. The equipment used in the experiment is NVIDIA GTX 1080ti.

C. Results

The experimental results are shown in Table II. We compare the ESD-MBENet with the previous proposed excellent algorithm when using the same backbones and the same datasets. To reduce the experimental error, we did each experiment five times and reported the results as the mean and standard deviation of the five experiments.

1) Classification Results on the AID Dataset: The setting of ESD-MBENet in the AID dataset is the same as the previous methods; 20% or 50% of the data are randomly selected as the training set and the rest data are as the test set. If the backbone is VGG16, the results of ESD-MBENet-v1 and ESD-MBENet-v2 are 94.10% and 94.12% on 20% training data. It is very close to the state-of-the-art results (94.27%). On 50% training data, the results are 97.15% and 97.3%. When using ResNet50 as the backbone and 20% and 50% training set, ESD-MBENet-v1 can reach 96.0% and 98.54%, and ESD-MBENet-v2 can reach 95.81% and 98.66%. When DenseNet121 is selected as the backbone, ESD-MBENet-v1 can achieve the accuracy of 96.2% and 98.85%, and ESD-MBENet-v2 can achieve the accuracy of 96.39% and 98.4%. Compared with KFBNet, we did not introduce additional elements to the network in inference, that is, only the main branch is used for inference, but the results surpass about 1% in DenseNet121.

2) Classification Results on the NWPU-RESISC45 Dataset: In the experiment, we randomly select 10% or 20% of the data for training and the rest data for test. If VGG16 is the backbone, 10% and 20% data for training, the accuracy of the ESD-MBENet-v1 is 90.29% and 93.48%, and the accuracy of the ESD-MBENet-v2 is 90.25% and 93.42%. If ResNet50 is the backbone of the ESD-MBENet, we can achieve the accuracy of 92.5% and 95.58% in ESD-MBENet-v1 and the accuracy of 93.03% and 95.24% in ESD-MBENet-v2. ESD-MBENet-v1 can reach 93.24% and 95.5% and ESD-MBENet-v2 can reach 93.05% and 95.36% in DenseNet121. In most cases, ESD-MBENet exceeds the state-of-the-art results. In a few cases, it is very close to them.

3) Classification Results on the UC-Merced Dataset: The accuracy of the proposed methods previously in the UC-Merced dataset has reached the limit. The results of

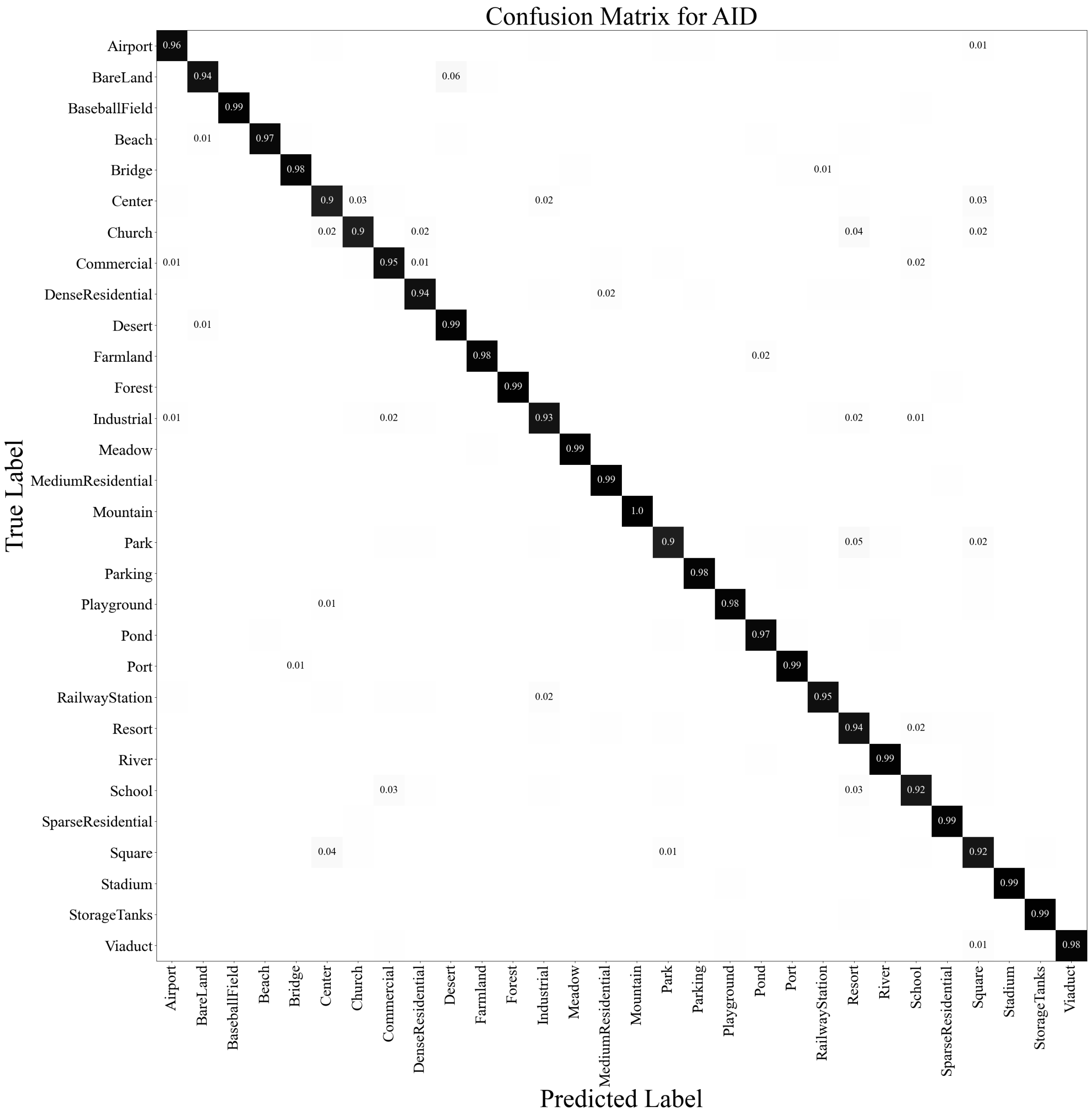


Fig. 5. ESD-MBENet-v1 confusion matrix of DenseNet121 on the 20% AID training data.

ESD-MBENet are the same. During the experiment, even if VGG16 is used as the backbone, the accuracy can reach 100% sometimes. There are a total of 420 test images. This result means that at most one image can be predicted incorrectly or all predictions are accurate in each experiment. The average accuracy can reach 99.81% or 99.86%.

4) Classification Results on the Million-AID Dataset: We randomly select 10% or 20% data for training, and the rest data for test. As shown in Table III, ESD-MBENet-v1 and ESD-MBENet-v2 show the well generalization on this dataset whatever the backbone is. The accuracy of our proposed method is higher than that of the baseline model.

5) Computation Complexity and Inference Time Results: As shown in Table IV, during training, to improve the accuracy, the computation of the proposed model is more complex than the baseline model. During inference, we only use the main branch that is the same as the baseline model, so the computation complexity and frames per second (FPS) are the same as the baseline model.

D. Confusion Matrix

In the remote sensing image scene classification, a confusion matrix is used as an evaluation criterion to judge the effect of the proposed algorithm. The confusion matrix can be used

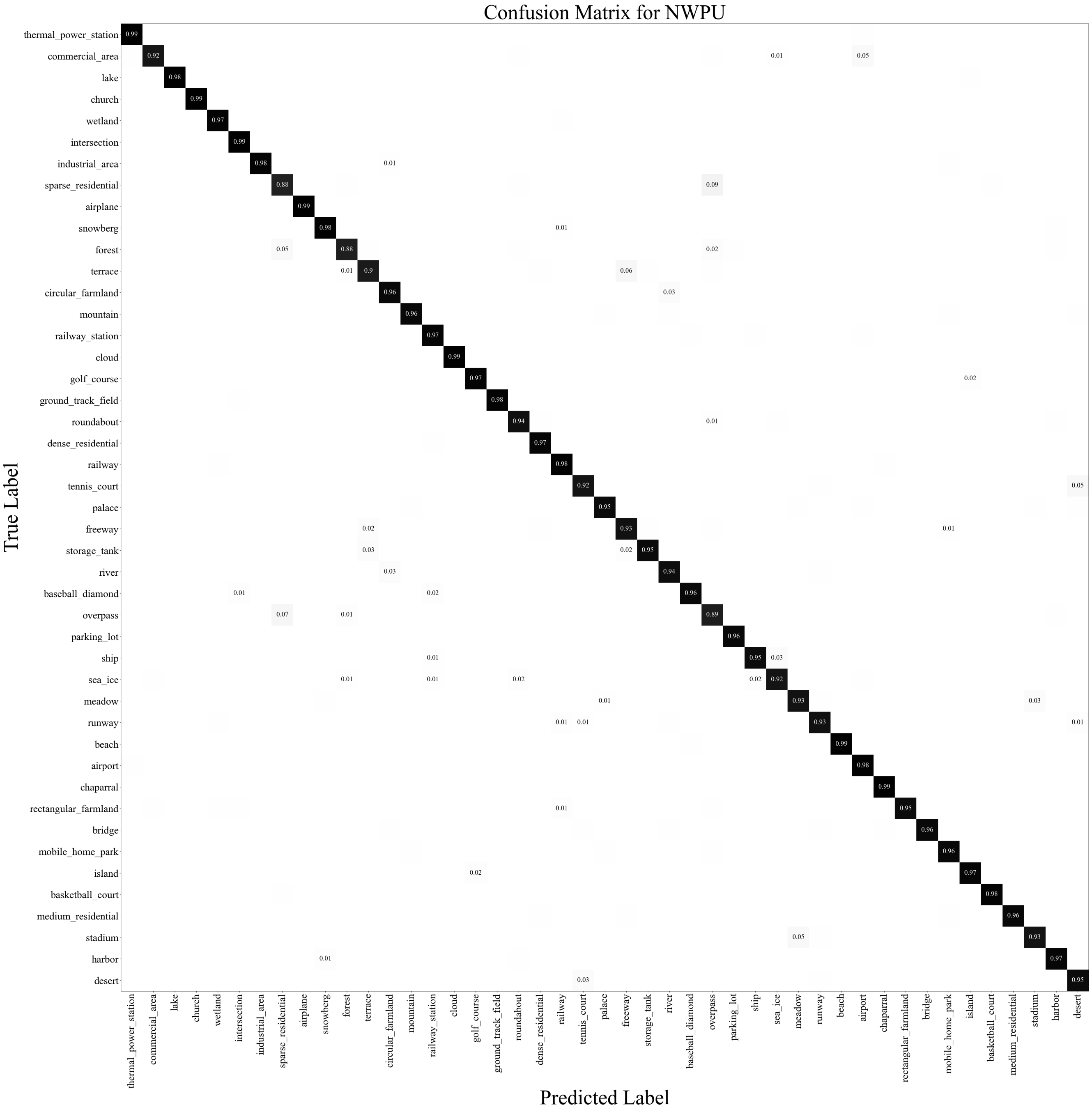


Fig. 6. ESD-MBENet-v1 confusion matrix of ResNet50 on the 20% NWPU-RESISC45 training data.

to represent the difference between the predicted label and the true label. The confusion matrix is very intuitive and clear, which is very useful for data analysis. In this article, a confusion matrix is used to check the effectiveness of ESD-MBENet. Fig. 5 shows the confusion matrix of 20% AID training data in the DenseNet121 network. It can be seen that ESD-MBENet-v1 has less than a 4% prediction error rate for almost all classes, and the prediction accuracy rate of some classes even reaches 99% and 100%. Fig. 6 shows the confusion matrix made by ESD-MBENet-v1(ResNet50) prediction results on the 20% NWPU-RESISC45 training dataset. The prediction error rate is less than 5% for almost all classes.

E. Ablation Study

To more effectively verify the effects of multibranch ensemble and SD of ESD-MBENet and the robustness of the network to add different feature augmentation modules to subbranch, we did the following ablation experiments.

1) Comparison Between ESD-MBENet and Baseline: To solve the interference from different characteristics of different geographical elements in remote sensing images, ESD-MBENet has introduced the method of multibranch ensemble network in the training process, allowing the network to explore image information from multiple perspectives. We introduce SD on the output logits and feature maps to

TABLE V

COMPARISON OF EXPERIMENTAL RESULTS BETWEEN BASELINE NETWORK AND ESD-MBENET ON THE AID AND NWPU-RESISC45 DATASETS. "v1-OD" AND "v2-OD" DENOTE ESD-MBENET ONLY SELF-DISTILL THE OUTPUT LOGITS. "v1/v2" DENOTES ESD-MBENET-v1/v2

(a)

Networks	AID		NWPU-RESISC45	
	tr = 20%	tr = 50%	tr = 10%	tr = 20%
VGG16	93.21±0.24	96.9±0.16	88.16±0.13	92.69±0.09
VGG16-v1-OD	93.82±0.15	97.08±0.15	89.98±0.21	93.15±0.17
VGG16-v1	**94.10 ± 0.13**	**97.15 ± 0.21**	**90.29 ± 0.11**	**93.48 ± 0.06**
VGG16-v2-OD	94.03±0.22	96.98±0.23	89.95±0.25	93.12±0.2
VGG16-v2	**94.12 ± 0.24**	**97.3 ± 0.08**	**90.25 ± 0.21**	**93.42 ± 0.15**

(b)

Networks	AID		NWPU-RESISC45	
	tr = 20%	tr = 50%	tr = 10%	tr = 20%
ResNet50	95.4±0.05	97.82±0.12	91.6±0.09	94.85±0.14
ResNet50-v1-OD	95.75±0.23	98.44±0.2	92.48±0.24	95.2±0.12
ResNet50-v1	**96.0 ± 0.15**	**98.54 ± 0.17**	**93.42 ± 0.15**	**95.58 ± 0.08**
ResNet50-v2-OD	95.80±0.22	98.22±0.13	92.91±0.19	95.15±0.21
ResNet50-v2	**95.81 ± 0.24**	**98.66 ± 0.2**	**93.03 ± 0.11**	**95.24 ± 0.23**

(c)

Networks	AID		NWPU-RESISC45	
	tr = 20%	tr = 50%	tr = 10%	tr = 20%
DenseNet121	95.46±0.13	98.2±0.07	91.77±0.22	94.34±0.15
DenseNet121-v1-OD	95.96±0.19	98.8±0.24	92.8±0.24	95.42±0.2
DenseNet121-v1	**96.2 ± 0.15**	**98.85 ± 0.13**	**93.24 ± 0.15**	**95.5 ± 0.09**
DenseNet121-v2-OD	96.30±0.18	98.38±0.06	92.92±0.22	95.23±0.16
DenseNet121-v2	**96.39 ± 0.21**	**98.4 ± 0.23**	**93.05 ± 0.18**	**95.36 ± 0.14**

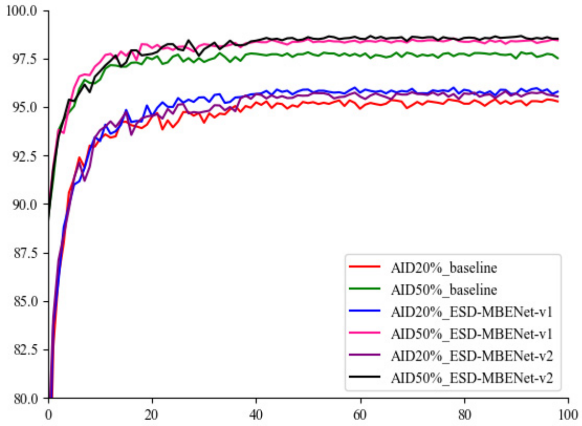


Fig. 7. Training curves of AID on ResNet50. The accuracy changes of the training process of baseline, and ESD-MBENet-v1 and ESD-MBENet-v2 are compared on the 20% and 50% AID training datasets, respectively. "AID20%" means that we select 20% data for training.

reduce the complexity of the model, and there is no difference in inference speed compared with the baseline network.

We use VGG16, ResNet50, and DenseNet121 as the backbones and do the following comparative experiments on the AID and NWPU-RESISC45 datasets. It can be seen from Table V that ESD-MBENet-v1 and ESD-MBENet-v2 both have more than 1% improvement compared with the baseline network. This can also verify that ESD-MBENet has indeed learned more image information through multibranch feature ensemble and SD, which is helpful for remote sensing image classification.

2) SD in ESD-MBENet Feature Maps: Distillation technology is essentially a process in which a student network learns and imitates a teacher network to achieve student knowledge enhancement. In this experiment, we use the ESD-MBENet itself as the teacher, so it can be called SD. We mainly use SD in output logits and intermediate feature maps. For students' learning, if the teacher directly tells the students the standard answer every time, let the students explore the learning process by themselves, in most cases, the students will learn very well. However, if the teacher also provides guidance and advice in the learning process, this may be more helpful to the students' learning. Therefore, we propose to use SD in feature maps. In order to effectively compare the effectiveness of the idea, we also did comparative experiments on the AID and NWPU-RESISC45 datasets on the VGG16, ResNet50, and DenseNet121 networks. As shown in Table V, students who not only self-distill the output logits but also self-distill the feature maps learn better than only self-distill the output logits.

3) Comparison of Multibranch Ensemble Output and Main-Branch Output: When students learn well, the teacher may also be inspired in distillation, although sometimes the inspiration is small. Therefore, we compare the multibranch ensemble output with the main-branch output, as well as the parameters and FLOPs in inference. As shown in Table VI, it can be seen that the ensemble output does indeed perform well than the main-branch output, but it consumes more parameters and FLOPs in inference. This is not friendly for practical applications. Therefore, we choose the main branch for inference.

4) Comparison of ESD-MBENet-v1 Subbranch Using Different Feature Augmentation Modules: To verify that ESD-MBENet-v1 is still effective even when the subbranch uses different feature augmentation modules, in the experiment, we mainly add the SE or CAM or dropout module in the subbranch. The experimental results are shown in Table VII. The multibranch network we constructed is robust to the addition of different feature augmentation modules.

F. Visualization and Analysis

1) Training Curves: To compare the convergence of ESD-MBENet with the baseline network more intuitively, we plot the curves of the experimental results. As shown in Fig. 7, we use ResNet50 to train 20% or 50% data of the AID dataset. It can be seen from the curves that the overall performance of the ESD-MBENet is better than the baseline. Also, the difference between the results of ESD-MBENet-v1 and ESD-MBENet-v2 is very small. We set a total of 100 epochs. The accuracy of baseline and ESD-MBENet networks is gradually flattening out around epoch 50. Compared with baseline posttraining, ESD-MBENet has less fluctuation and is more stable.

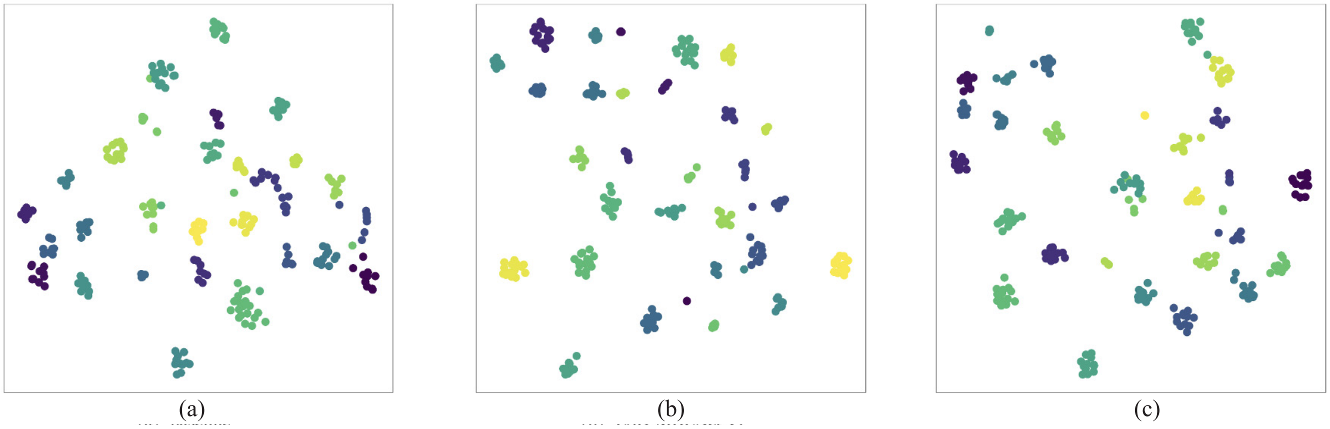


Fig. 8. T-SNE of the DenseNet121, ESD-MBENet-v1, and ESD-MBENet-v2. There are more interclass differences in ESD-MBENet-v1 and ESD-MBENet-v2 compared with the baseline network. Different colors indicate different categories, and the same categories are gathered into a small pile. The larger the distance between the small piles, the larger the gap in different classes, and the better the classification effect. (a) Baseline. (b) ESD-MBENet-v1. (c) ESD-MBENet-v2.

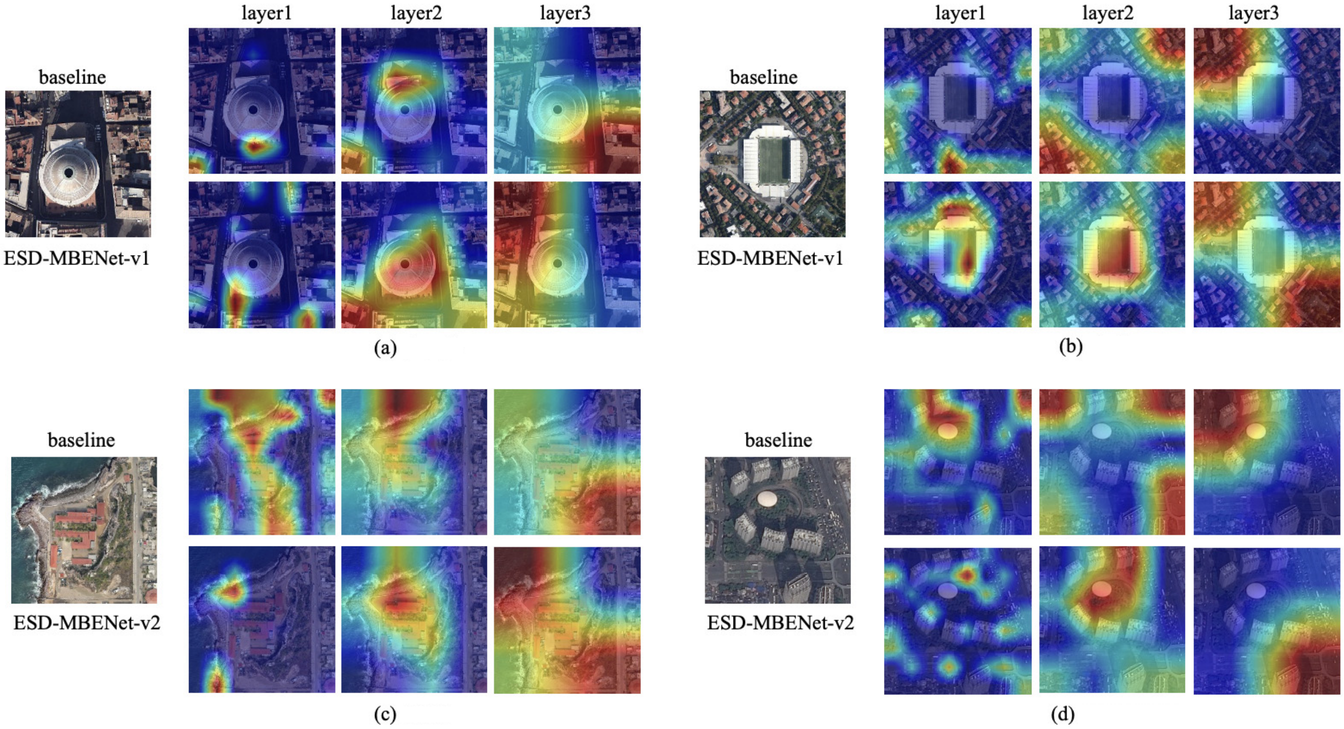


Fig. 9. Grad-CAM comparison of ResNet50 network and ESD-MBENet on the 20% AID training dataset. We randomly select four images from the AID dataset as representatives and show the focus of different parts of the baseline and ESD-MBENet, such as “layer1,” “layer2,” and “layer3” during the remote sensing image classification. The warmer the color, the higher the degree of attention. (a) Church. (b) Stadium. (c) Resort. (d) Commercial.

2) T-SNE: T-SNE technology can map data in high-dimensional space to low-dimensional space. We can clearly see the difference between different algorithms. Therefore, we use T-SNE to show a 2-D mapping representation of the final output results. As shown in Fig. 8, we compare the baseline, ESD-MBENet-v1, and ESD-MBENet-v2 networks, in which backbone is DenseNet121. ESD-MBENet can obtain larger interclass differences for remote sensing image classification. The more similar the category, the larger the gap between the categories is needed to achieve better classification, and the network will not be confused due to the large gap. Therefore, ESD-MBENet achieves a better classification effect than baseline.

3) Grad CAM: Grad-CAM is a popular visualization method, which can make it easier for us to understand how convolutional neural networks learn for a given task, such as image classification or image segmentation. We also visualized the Grad-CAM experimental effect of ESD-MBENet. Also, compared with the baseline network, in the experiment, the baseline, ESD-MBENet-v1, and ESD-MBENet-v2 models trained on the ResNet50 network using the 20% AID dataset are used to draw Grad-CAM on four randomly selected images. To compare the learning effect of the network at different stages, we have shown the Grad-CAM of the different depths of the network, such as “layer1,” “layer2,” and “layer3,” which can also represent the learning focus of the network in

TABLE VI

Compared the Multibranch Ensemble Output With Main-Branch Output on 20% Training Data of the NWPU-RESISC45 in Inference, Which Is Used DenseNet121 as the Backbone. "ESD-MBENet-v1-E" and "ESD-MBENet-v2-E" Denote the Multibranch Ensemble Output of the ESD-MBENet

Methods	Accuracy	Parameters	FLOPs
ESD-MBENet-v1-E	95.52±0.12	14.08M	7.96G
ESD-MBENet-v1	95.5±0.09	7.98M	2.87G
ESD-MBENet-v2-E	95.44±0.17	23.88M	5.16G
ESD-MBENet-v2	95.36±0.14	7.98M	2.87G

TABLE VII

Comparison Results of ESD-MBENet-v1 Subbranch Using Different Feature Augmentation Modules With ResNet50 as the Backbone. "ESD-MBENet-v1-SE" Means That the Module Added to the Subbranch Is SE. The TR of the AID and NWPU-RESISC45 Is 20%

Methods	NWPU-RESISC45	AID
ESD-MBENet-v1-SE	95.42±0.14	95.78±0.21
ESD-MBENet-v1-CAM	95.38±0.2	95.87±0.17
ESD-MBENet-v1-Dropout	95.58±0.08	96.0±0.15

the shallow stage and the deep stage. It can be seen from Fig. 9 that the ESD-MBENet network pays more attention to the objects to be classified at "layer2" than the baseline network. At "layer3," the focus of the ESD-MBENet is more than that of the baseline, which means that ESD-MBENet can extract more information of the images and then transfer it to the deeper network. This is more conducive to network learning.

V. Conclusion

In this article, we design ESD-MBENet-v1 and ESD-MBENet-v2 to construct a compact multibranch ensemble network to solve the interference from different characteristics of different geographical elements in remote sensing images. ESD-MBENet-v1 uses as few modules as possible to build as many branches as possible, but as the split points move backward, the number of branches decreases. Therefore, we propose ESD-MBENet-v2, which can build multiple branches flexibly. ESD-MBENet-v2 achieves the greatest possible weight sharing. Due to the multibranch construction, although the network performance has been greatly improved, in the inference stage, the model is too complex to reduce the inference efficiency and speed. Thus, we propose SD, distilling logits, and intermediate feature maps, to make the main-branch network reach the performance of the whole network. In this way, only the main branch is used for inference. Through experimental verification, our proposed ESD-MBENet network achieves better classification results than previous state-of-the-art deep learning networks on remote sensing datasets. Meanwhile, it is easy to transfer our methods to pixel-wise classification and multispectral images task. For pixel-wise classification, we can design a multibranch encoder–decoder network or modify the output according to our proposed method. For a multispectral images task, we can use different branches to extract the spectral and spatial features and fuse them.

References

[1] P. Zhong and R. Wang, "Jointly learning the hybrid CRF and MLR model for simultaneous denoising and classification of hyperspectral imagery," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 7, pp. 1319–1334, Jul. 2014, doi: 10.1109/TNNLS.2013.2293061.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 26th Annu. Conf. Neural Inf. Process. Syst.*, P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., Lake Tahoe, NV, USA, 2012, pp. 1106–1114. [Online]. Available: https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

[4] T.-B. Xu and C.-L. Liu, "Deep neural network self-distillation exploiting data representation invariance," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Oct. 19, 2020, doi: 10.1109/TNNLS.2020.3027634.

[5] U. Muhammad, W. Wang, S. P. Chattha, and S. Ali, "Pre-trained VGGNet architecture for remote-sensing image scene classification," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 1622–1627.

[6] X. Liu, M. Chi, Y. Zhang, and Y. Qin, "Classifying high resolution remote sensing images by fine-tuned VGG deep networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2018, pp. 7137–7140.

[7] R. Cao, L. Fang, T. Lu, and N. He, "Self-attention-based deep feature fusion for remote sensing scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 1, pp. 43–47, Jan. 2021.

[8] Y. Tan, S. Xiong, and P. Yan, "Multi-branch convolutional neural network for built-up area extraction from remote sensing image," *Neurocomputing*, vol. 396, pp. 358–374, Jul. 2020, doi: 10.1016/j.neucom.2018.09.106.

[9] C. Liu *et al.*, "Remote sensing images feature learning based on multi-branch networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Waikoloa, HI, USA, Sep./Oct. 2020, pp. 2057–2060, doi: 10.1109/IGARSS39084.2020.9323967.

[10] A. Raza, H. Huo, S. Sirajuddin, and T. Fang, "Diverse capsules network combining multiconvolutional layers for remote sensing image scene classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5297–5313, 2020.

[11] S. Wang, Y. Guan, and L. Shao, "Multi-granularity canonical appearance pooling for remote sensing scene classification," *IEEE Trans. Image Process.*, vol. 29, pp. 5396–5407, 2020, doi: 10.1109/TIP.2020.2983560.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, pp. 1–14, Sep. 2014.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[14] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[15] G.-S. Xia *et al.*, "AID: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3965–3981, Jul. 2017.

[16] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: Benchmark and state of the art," *Proc. IEEE*, vol. 105, no. 10, pp. 1865–1883, Oct. 2017.

[17] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geograp. Inf. Syst.*, Nov. 2010, pp. 270–279.

[18] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[19] Y. Yang and S. Newsam, "Geographic image retrieval using local invariant features," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 2, pp. 818–832, Feb. 2013, doi: 10.1109/TGRS.2012.2205158.

[20] V. Risojević and Z. Babić, "Fusion of global and local descriptors for remote sensing image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 4, pp. 836–840, Jul. 2013.

[21] Z. Shi, X. Yu, Z. Jiang, and B. Li, "Ship detection in high-resolution optical imagery based on anomaly detector and local shape feature," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 8, pp. 4511–4523, Aug. 2014, doi: 10.1109/TGRS.2013.2282355.

[22] W. Zhang, X. Sun, H. Wang, and K. Fu, "A generic discriminative part-based model for geospatial object detection in optical remote sensing images," *ISPRS J. Photogramm. Remote Sens.*, vol. 99, pp. 30–44, Jan. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271614002573

[23] G. Cheng, P. Zhou, X. Yao, C. Yao, Y. Zhang, and J. Han, "Object detection in VHR optical remote sensing images via learning rotation-invariant HOG feature," in *Proc. 4th Int. Workshop Earth Observ. Remote Sens. Appl. (EORSA)*, Jul. 2016, pp. 433–436.

[24] R. O. Stehling, M. A. Nascimento, and A. X. Falcão, "A compact and efficient image retrieval approach based on border/interior pixel classification," in *Proc. ACM CIKM Int. Conf. Inf. Knowl. Manage.* McLean, VA, USA, Nov. 2002, pp. 102–109, doi: 10.1145/584792.584812.

[25] J. A. D. Santos, O. A. B. Penatti, and R. da Silva Torres, "Evaluating the potential of texture and color descriptors for remote sensing image retrieval and classification," in *Proc. 5th Int. Conf. Comput. Vis. Theory Appl. (VISAPP)*, vol. 2, P. Richard and J. Braz, Eds. Angers, France: INSTICC Press, May 2010, pp. 203–208.

[26] O. A. B. Penatti, E. Valle, and R. da Silva Torres, "Comparative study of global color and texture descriptors for web image retrieval," *J. Vis. Commun. Image Represent.*, vol. 23, no. 2, pp. 359–380, 2012, doi: 10.1016/j.jvcir.2011.11.002.

[27] L. Zhao, P. Tang, and L. Huo, "A 2-D wavelet decomposition-based bag-of-visual-words model for land-use scene classification," *Int. J. Remote Sens.*, vol. 35, no. 6, pp. 2296–2310, 2014.

[28] Q. Zhu, Y. Zhong, B. Zhao, G.-S. Xia, and L. Zhang, "Bag-of-visual-words scene classifier with local and global features for high spatial resolution remote sensing imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 6, pp. 747–751, Jun. 2016.

[29] R. Bahmanyar, S. Cui, and M. Datcu, "A comparative study of bag-of-words and bag-of-topics models of EO image patches," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 6, pp. 1357–1361, Jun. 2015.

[30] G. Cheng and J. Han, "A survey on object detection in optical remote sensing images," *CoRR*, vol. abs/1603.06201, pp. 1–32, Mar. 2016.

[31] R. Sedona, G. Cavallaro, J. Jitsev, A. Strube, M. Riedel, and J. Benediktsson, "Remote sensing big data classification with high performance distributed deep learning," *Remote Sens.*, vol. 11, no. 24, p. 3056, Dec. 2019, doi: 10.3390/rs11243056.

[32] T. He and S. Wang, "Multi-spectral remote sensing land-cover classification based on deep learning methods," *J. Supercomput.*, vol. 77, no. 3, pp. 2829–2843, Mar. 2021, doi: 10.1007/s11227-020-03377-w.

[33] S. K. Meher, "Granular space, knowledge-encoded deep learning architecture and remote sensing image classification," *Eng. Appl. Artif. Intell.*, vol. 92, Jun. 2020, Art. no. 103647, doi: 10.1016/j.engappai.2020.103647.

[34] Q. Zou, L. Ni, T. Zhang, and Q. Wang, "Deep learning based feature selection for remote sensing scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 11, pp. 2321–2325, Nov. 2015.

[35] J.-M. Liu and M.-H. Yang, "Deep learning-based classification of remote sensing image," *J. Comput.*, vol. 13, no. 1, pp. 44–48, Jan. 2018, doi: 10.17706/jcp.13.1.44-48.

[36] K. Yang, Z. Liu, Q. Lu, and G. Xia, "Multi-scale weighted branch network for remote sensing image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPR)*. Long Beach, CA, USA: Computer Vision Foundation, Jun. 2019, pp. 1–10. [Online]. Available: http://openaccess.thecvf.com/content_CVPRW_2019/html/DOAI/Yang_Multi-Scale_Weighted_Branch_Network_for_Remote_Sensing_Image_Classification_CVPRW_2019_paper.html

[37] H. Teffahi and H. Yao, "RS-MSSF frame: Remote sensing image classification based on extraction and fusion of multiple spectral-spatial features," in *Proc. 19th Pacific-Rim Conf. Multimedia*, in Lecture Notes in Computer Science, vol. 11165, R. Hong, W. Cheng, T. Yamasaki, M. Wang, and C. Ngo, Eds. Hefei, China: Springer, Sep. 2018, pp. 582–595, doi: 10.1007/978-3-030-00767-6_54.

[38] H. Ji, L. Tian, J. Li, R. Tong, Y. Guo, and Q. Zeng, "Spatial–spectral fusion of HY-1C COCTS/CZI data for coastal water remote sensing using deep belief network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 1693–1704, 2021, doi: 10.1109/JSTARS.2020.3045516.

[39] Y. Wang, X. Zhou, C. Li, Y. Chen, and L. Yang, "Bathymetry model based on spectral and spatial multifeatures of remote sensing image," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 1, pp. 37–41, Jan. 2020, doi: 10.1109/LGRS.2019.2915122.

[40] B. Xi, J. Li, Y. Li, R. Song, W. Sun, and Q. Du, "Multiscale context-aware ensemble deep KELM for efficient hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 5114–5130, Jun. 2021, doi: 10.1109/TGRS.2020.3022029.

[41] B. Xi *et al.*, "Multi-direction networks with attentional spectral prior for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, early access, Jan. 14, 2021, doi: 10.1109/TGRS.2020.3047682.

[42] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *CoRR*, vol. abs/1503.02531, pp. 1–9, Mar. 2015.

[43] G. Urban *et al.*, "Do deep convolutional nets really need to be deep and convolutional?" in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, Toulon, France, Apr. 2017, pp. 1–13. [Online]. Available: https://openreview.net/forum?id=r10FA8Kxg

[44] G. Chen, W. Choi, X. Yu, T. X. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., Long Beach, CA, USA, Dec. 2017, pp. 742–751. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/e1e32e235eee1f970470a3a6658dfdd5-Abstract.html

[45] T. Fukuda, M. Suzuki, G. Kurata, S. Thomas, J. Cui, and B. Ramabhadran, "Efficient knowledge distillation from an ensemble of teachers," in *Proc. Interspeech, 18th Annu. Conf. Int. Speech Commun. Assoc.*, F. Lacerda, Ed. Stockholm, Sweden: ISCA, Aug. 2017, pp. 3697–3701. [Online]. Available: http://www.isca-speech.org/archive/Interspeech_2017/abstracts/0614.html

[46] G. Chen *et al.*, "Training small networks for scene classification of remote sensing images via knowledge distillation," *Remote Sens.*, vol. 10, no. 5, p. 719, May 2018, doi: 10.3390/rs10050719.

[47] S. Pande, A. Banerjee, S. Kumar, B. Banerjee, and S. Chaudhuri, "An adversarial approach to discriminative modality distillation for remote sensing image classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 4571–4580.

[48] R. Zhang *et al.*, "Remote sensing image scene classification with noisy label distillation," *Remote Sens.*, vol. 12, no. 15, p. 2376, Jul. 2020, doi: 10.3390/rs12152376.

[49] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Salt Lake City, UT, USA: IEEE Computer Society, Jun. 2018, pp. 7132–7141. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Hu_Squeeze-and-Excitation_Networks_CVPR_2018_paper.html

[50] J. Fu *et al.*, "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Long Beach, CA, USA: Computer Vision Foundation, Jun. 2019, pp. 3146–3154. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/Fu_Dual_Attention_Network_for_Scene_Segmentation_CVPR_2019_paper.html

[51] F. Li, R. Feng, W. Han, and L. Wang, "High-resolution remote sensing image scene classification via key filter bank based on convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 11, pp. 8077–8092, Nov. 2020.

[52] S. Chaib, H. Liu, Y. Gu, and H. Yao, "Deep feature fusion for VHR remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4775–4784, Aug. 2017, doi: 10.1109/TGRS.2017.2700322.

[53] Y. Yu and F. Liu, "Dense connectivity based two-stream deep feature fusion framework for aerial scene classification," *Remote Sens.*, vol. 10, no. 7, p. 1158, Jul. 2018, doi: 10.3390/rs10071158.

[54] Q. Wang, S. Liu, J. Chanussot, and X. Li, "Scene classification with recurrent attention of VHR remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 1155–1167, Feb. 2019, doi: 10.1109/TGRS.2018.2864987.

[55] Y. Liu, C. Y. Suen, Y. Liu, and L. Ding, "Scene classification using hierarchical Wasserstein CNN," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 5, pp. 2494–2509, May 2019, doi: 10.1109/TGRS.2018.2873966.

[56] G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, "When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 2811–2821, May 2018.

[57] N. He, L. Fang, S. Li, A. Plaza, and J. Plaza, “Remote sensing scene classification using multilayer stacked covariance pooling,” *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 12, pp. 6899–6910, Dec. 2018.

[58] Z. Chen, S. Wang, X. Hou, and L. Shao, “Recurrent transformer network for remote sensing scene categorisation,” in *Proc. Brit. Mach. Vis. Conf.*, 2018, p. 266.

[59] N. He, L. Fang, S. Li, J. Plaza, and A. Plaza, “Skip-connected covariance network for remote sensing scene classification,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 5, pp. 1461–1474, May 2020.

[60] R. Minetto, M. P. Segundo, and S. Sarkar, “Hydra: An ensemble of convolutional neural networks for geospatial land classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6530–6541, Sep. 2019.

Qi Zhao (Member, IEEE) received the Ph.D. degree in communication and information system from Beihang University, Beijing, China, in 2002.

She was a Visiting Scholar with the Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, PA, USA, from 2014 to 2015. She is currently a Professor with Beihang University. Since 2016, she has been working on wearable device-based first-view image processing and deep learning-based image recognition. Her research interests include one-shot semantic segmentation, communication signal processing, and target tracking.

Yujing Ma received the B.S. degree in information engineering from the China University of Mining and Technology, Beijing, China, in 2019. She is currently pursuing the M.S. degree with the School of Electronics and Information Engineering, Beihang University, Beijing.

Her research interests include deep learning, remote sensing, and active learning.

Shuchang Lyu (Graduate Student Member, IEEE) received the B.S. degree in communication and information from Shanghai University, Shanghai, China, in 2016, and the M.S. degree in communication and information system from the School of Electronic and Information Engineering, Beihang University, Beijing, China, in 2019, where he is currently pursuing the Ph.D. degree.

His research interests include deep learning, image classification, one-shot semantic segmentation, and object detection.

Lijiang Chen received the B.S. and Ph.D. degrees from the School of Electronic and Information Engineering, Beihang University, Beijing, China, in 2007 and 2012, respectively.

He was a Hong Kong Scholar with the City University of Hong Kong, Hong Kong, from 2015 to 2017. He is currently an Assistant Professor with the School of Electronic and Information Engineering, Beihang University. His research interests include pattern recognition, image processing, and human–computer interaction.