SEMSDNet: A Multiscale Dense Network With Attention for Remote Sensing Scene Classification

Tian Tian, Lingling Li, Weitao Chen, and Huabing Zhou

***Abstract*—Remote sensing image scene classification plays an important role in remote sensing image interpretation. Deep learning brings prosperity to the research in this field, and numerous deep learning models are proposed in order to improve the performance of scene classification. However, images of different remote sensing scenes vary a lot, showing similar or diverse textures and simple or complex contents. Using a fixed convolutional neural network framework to classify scene images is performance-limited and not practice-flexible. To address this issue, in this article, we propose the SEMSDNet (multiscale dense networks with squeeze and excitation attention). The framework multiscale dense convolutional network (MSDNet) with multiple classifiers and dense connections can automatically transform between a small network and a deep network according to the complexity of test samples and the limitation of computational resources. Moreover, in order to extract more effective features, the squeeze-and-excitation (SE) attention mechanism is introduced into the framework to process the features of various scenes self-adaptively. In addition, considering the limited computing resources, we impose two settings with computational constraints at the test time: budgeted batch classification, which is a fixed computational budget setting for sample classification, and anytime prediction, which forces the network to output a prediction at any given point-in-time. Experimental results on several public datasets show that the proposed SEMSDNet method is superior to the state-of-the-art methods on both performance and efficiency. Experiments also reveal its capability to treat samples of different classification difficulties with uneven resource allocation and flexible network architecture, showing its potentials in practical applications.**

***Index Terms*—Attention mechanism, dense connection, multiscale, remote sensing scene classification.**

I. Introduction

With the rapid development of space remote sensing technology, a large number of high-quality remote sensing scene images are easy to obtain [1]. Remote sensing scene classification is an important method for remote sensing image interpretation, which has a significant application value in urban planning, geographic image retrieval, and land-use classification [2].

Fig. 1. Left image depicts the desert with simple textures, which is easy to classify, whereas the right image shows the industrial scene with complex contents, which is difficult to classify.

Remote sensing scene classification automatically assigns a specific label to each remote sensing image based on the content of scene [3], and the results of remote sensing scene classification usually depend on the features extracted from the images [4]. In recent decades, many scholars have done a great deal of research works on the classification of remote sensing scenes, and a review of them can be found in literature [5], [6]. There are three mainstream methods for remote sensing scene classification according to the level of features, which will be detailed in Section II. Deep convolutional neural network (CNN) has achieved great success in many visual applications due to its good biological basis and hierarchically abstract expression ability [7], which has been widely concerned by academia and industry. In recent years, CNN has also been widely used in remote sensing scene classification [8]–[10]. Without manually designed features, more complex and higher level abstract semantic features can be automatically extracted directly by superimposing a series of convolutional and pooling layers.

Although previous CNN models have remarkably improved performances on remote sensing scene classification, they also have some limitations. These CNN models generally use a uniform pipeline to extract image features, namely, they treat different scene categories equally. However, the classification difficulty of remote sensing images varies greatly among different scene categories. To illustrate this point, different scene images in Aerial Image Dataset (AID) [5] are taken as an example. As shown in Fig. 1, the left image depicts the desert with simple textures and the single content is obviously much easier to identify, whereas the right one shows the industrial scene with more complex objects and contents is more difficult to recognize.

Manuscript received November 27, 2020; revised February 25, 2021 and April 11, 2021; accepted April 17, 2021. Date of publication April 22, 2021; date of current version June 8, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 42071339, Grant U1711266, Gtart 41925007, Grant U1803117, and Grant 42071430. *(Corresponding author: Weitao Chen.)*

Tian Tian is with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: tiantian_hust@foxmail.com).

Lingling Li and Weitao Chen are with the School of Computer Science, China University of Geosciences, Wuhan 430074, China (e-mail: 2986614532@qq.com; wtchen@cug.edu.cn).

Huabing Zhou is with the School of Wuhan Institute of Technology, Wuhan 430205, China (e-mail: zhouhuabing@gmail.com).

Digital Object Identifier 10.1109/JSTARS.2021.3074508

Computationally intensive models are necessary to process more complex samples, but they may be wasteful and even overfitting when applied to simple samples such as the desert image on the left. In other words, it is a waste of computational resources for no gains to employ a complex network on the classification of simple scenes, whereas the effect of a simple and shallow network structure may be not good enough for complex scenes. Therefore, using a fixed framework to classify remote sensing images is not flexible enough. Moreover, the computational resources are not always rich in practical applications, which require different processing schemes for different images: less processing for simple samples, and more allocation of resources for complex samples as far as possible. This may be a good idea to achieve a comprehensive optimal classification performance under the condition of limited computational resources.

In view of the above characteristics of remote sensing scene images, we propose to use a new framework, MSDNet [11], on remote sensing scene image classification, which facilitates early prediction outputs with multiple classifiers. MSDNet is a CNN with multiscale features and dense connections. The multiscale feature extraction is used to maintain both coarse and fine feature representation within the network structure and benefit the prediction on early layers, and the dense connection is adopted to alleviate the impact of early classifiers on later ones. In our scheme and application, simple scene images can output classification results directly from the shallow classifier, while complex scenes can be further processed at a deeper classifier in the network. In other words, when the test image is simple or the computational resources are limited, the model can automatically form a small and simple network; when the sample is difficult or resources are abundant, the model can be used as a deeper and more complex network structure. Considering practical conditions, we impose two settings with computational constraints at the test time: budgeted batch classification, where a fixed computational budget is shared across a large set of examples, which can be spent unevenly across "easy" and "hard" image samples; anytime prediction, where the network can be forced to output a prediction at any given point-in-time. The budgeted batch classification setting is ubiquitous in large-scale machine learning applications, so it is in the remote sensing field. In the budgeted batch classification setting, a model can improve the average accuracy by reducing the amount of computation spent on easy samples to save up computation for hard samples. Anytime prediction is also necessary in real applications, because an output must be obtained before the computational resources run out.

In addition, the remote sensing image scene is rich in spatial information and complex background, where a lot of redundant features exist, which will interfere with the effective feature extraction for scene classification. In the feature extraction process, if the network treats each feature equally, the feature discrimination ability of the model is obviously insufficient. Therefore, it is necessary to effectively extract a variety of features and establish the relationship between features, and finally extract the key features that are most important to the classification task. Without introducing a new spatial dimension to fuse the feature channels, we use the SE attention mechanism [12] to establish the relationship between feature channels, which can explicitly model the interdependence between channels and automatically obtain the importance of each feature channel. This attention mechanism allows the model to enhance the feature channels with useful information, while suppresses those channels that are not useful for classification tasks. The essence of this attention module is to learn the weights of feature maps, and we think this attention mechanism will be highly beneficial for MSDNet on remote sensing scene classification. Our experimental results in Section IV also show that the SE attention mechanism has indeed improved the classification effects. To sum up, the contributions of our work include the following three aspects.

1) We adopt the multiscale dense framework with multiple classifiers and propose SEMSDNet for remote sensing scene classification. The proposed method has achieved state-of-the-art performances with better efficiency on three public datasets. As far as we know, this is the first time to employ this framework on remote sensing scene classification.
2) Considering the complexity of remote sensing scenes, different processing and different allocation of computing resources can be used for simple or difficult samples to achieve the optimal comprehensive performance. The proposed approach uses attention mechanism enhanced MSDNet with multiple classifiers to implement different processing pipelines under a unified framework.
3) We research two common settings under limited computing resources. No matter budgeted batch classification or anytime prediction, the proposed method is able to achieve better classification accuracies and require less computation consumption, showing the ability to make a good balance and a flexible processing on various samples.

The rest of this article is organized as follows. Section II reviews the related literature work of this study. Section III introduces the proposed SEMSDNet for remote sensing scene classification. Section IV shows the experimental results of the proposed SEMSDNet on several public benchmark datasets. Section V summarizes the full article, and puts forward some opinions and suggestions for the future research.

II. Related Work

In this section, we first review the traditional methods of remote sensing scene classification. Then, we also discuss the attention mechanism and implementation methods.

A. Methods for Remote Sensing Scene Classification

In recent years, with the rapid development of image processing and pattern recognition, many scholars have done a lot of work on remote sensing scene classification. The major research works of remote sensing image scene classification focus on feature extraction and semantic classification. To obtain high-accuracy classification results, effective and discriminative feature representation plays a very important role [13]. According to the level of feature, the remote sensing image classification methods can be divided into three categories: 1) low-level features; 2) mid-level features; and 3) high-level semantic features.

The original classification methods mainly depend on the extraction of manually designed low-level features. These methods first describe low-level features such as the texture, shape, space, color, and spectral information. Then, the whole scene image through the feature description route of local feature extraction, middle-level coding, and global expression is described, and finally, classification results through the feature classifier are obtained. Specifically, Li *et al.* [14] proposed morphological texture descriptors to extract useful contents from remote sensing images. Aptoula *et al.* [15] utilized a color code method to accelerate scene classification performance. However, methods based on low-level feature extraction depend on the features of manual design, which cannot capture the rich semantic information of remote sensing images; therefore, their performances on scene classification tasks are unsatisfactory. Methods based on mid-level feature extraction use handcrafted feature descriptors of low level to extract local image features and use high-order statistical patterns to encode these features [16], [17]. The bag-of-visual word is the pioneer of this kind of methods, and then, it is enhanced by spatial pyramid matching [18], [19] and sparse coding [20], [21] to increase the constraint of feature distribution and reduce the complexity of models. Although they have made progress on this research, these methods can still not extract high-level semantic features, resulting in inaccurate classification performances. In recent years, deep learning technology has been successfully applied to remote sensing image scene classification. The most popular deep learning models are the autoencoder (AE) [22] and the CNN [23]. Zhang *et al.* [22] utilized the AE to train a feature extraction model for scene classification, although methods based on AE need a pretraining layer and are time-consuming to train. CNNs can directly train a deep network without pretraining, meanwhile achieve better classification accuracy of remote sensing scenes, so they have attracted more and more attention from the researchers.

Most of the early CNN-based methods train the networks with a fixed scale. However, for the classification of remote sensing scenes with complex surface coverage and various scales of detail textures, it is not appropriate to use a deep abstract feature of a single scale. Convolutional kernels of fixed size are insufficient to extract scene features of different scales, inevitably losing effectively discriminative information. Moreover, features of different scales in deep learning provide descriptions of different levels, and making full use of scale information is beneficial to the remote sensing image classification.

To solve the above problem, some multiscale classification methods have been proposed successively. Liu *et al.* [24] integrated CNN and the kernel method to fuse the multiscale features extracted from images. Suhui *et al.* [25] used CNN to extract the multiscale features of images, and used multicore support vector machine to improve the scene classification. Liu *et al.* [26] proposed a multiscale CNN (MCNN) framework for remote sensing classification. Different from the previous methods of training networks on fixed scale images, MCNN constructs a new network structure including two network branches (fixed-scale branch and varied-scale branch), which can train network models with multiscale images simultaneously. In the above scene classification methods, only the local multiscale is used, but the multiscale features of images and the depth of networks are not fully utilized. Therefore, Yang *et al.* [27] proposed a new remote sensing scene classification method based on multiscale feature fusion (MSFF). MSFF combines multiscale features and multiscale input images for the first time, and the hierarchical features extracted from different levels are fused for classification. Wang *et al.* [28] proposed an enhanced feature pyramid network to extract multiscale and multilevel features, and a feature fusion module called two-branch deep feature fusion is introduced to aggregate the features at different levels in an effective way.

However, the abovementioned methods treat remote sensing scene categories equally. In fact, similar to natural image classification, some input images are easy to classify by the networks, while some are difficult. Taking the classification difference of natural images and the budget balance under limited resources into account, MSDNet uses a cascade of intermediate classifiers throughout the network. For simple images, the results can be obtained in advance from a previous classifier, while complex images can be transmitted to deeper classifiers. Although the early exit idea has been presented in some literatures [29], [30], the performance degradation of predictions are observed with the insertion of early classifiers. Huang *et al.* [11] implemented comprehensive experiments of early exits and concluded the two problems that the direct design idea of multiple classification layers brings. 1) The traditional neural networks learn fine features at the front layers and coarse features at the back layers. The abstract semantic features of the last layers are important to classify the content of an image into a given class, so the classification accuracy is highly dependent on the position of the classifier. Due to the lack of coarse scales, classifiers at shallow layers may produce poor classification results. 2) The addition of early classifiers will interfere with later classifiers. Intermediate classifiers in the early layers intend to optimize the short term rather than the last layer. This optimization improves the accuracy of the intermediate classifiers, but harms the information needed to generate high-quality features in the later layers. This effect becomes more apparent when the first classifier is connected to an earlier layer [11].

For the first problem, MSDNet proposes to employ multiscale feature maps, and all classifiers only use coarse-level features. A feature map at a particular layer and scale is obtained by combining the following one or two convolutions: 1) apply regular convolution to the same scale features of the previous layers (horizontal connection); 2) apply strided convolution to the fine-scale feature map from the previous layers (vertical connections). The horizontal connections preserve high-resolution information, which helps to build high-quality coarse features in subsequent layers. And the vertical connections produce coarse features that are conducive to classify throughout the process. For the second problem, MSDNet proposes to adopt dense connection. Dense connections [31] connect each layer to all subsequent layers and allow subsequent layers to bypass the feature of short-term optimization to keep the final classifier high precision. If an earlier layer collapses information to generate short-term features, the lost information can be recovered through the direct connection to its previous layers. Performance

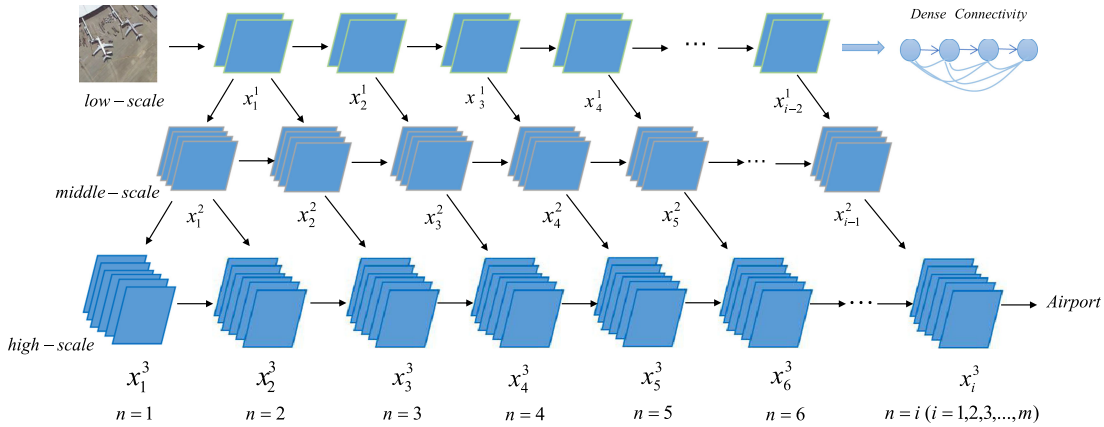

Fig. 2. Multiscale architecture for remote sensing scene classification.

of the final classifiers will become more independent of the location of the intermediate classifiers. MSDNet has validated the importance of dense connectivity for early-exit classifiers in deep networks on natural images.

B. Attention Mechanism and Methods

The attention mechanism stems from the study of human vision. In cognitive science, because of the bottleneck of information processing, human beings selectively focus on some of the information and ignore other visible information. CNNs use the attention mechanism to focus on a certain part of the given information, and each pixel has an independent weight, highlighting the distinguishing features that are effective and weakening the information that is not conducive to classification [32]. Attention can be seen as a way to allocate available computing resources to the most informative signal components [33], [34].

Attention mechanism has been used successfully in many tasks including localization and understanding in images [35], sequence learning [36], [37], lip reading [38], and image captioning [39]. In these applications, it can be merged into a single operator, following one or more layers that represent higher levels of semantic abstraction. Attention mechanism has also been successfully applied to remote sensing scene classification and significantly improved the classification effect. For example, Tang *et al.* [40] proposed a parallel-attention model to capture the local information from the spatial and spectral aspects. Some researches focus on the combined use of spatial attention and channel attention [41], [42]. Different from the previous studies that pay attention to the spatial attention with a large number of weight parameters [43], the squeeze-and-excitation (SE) block contains a lightweight gating mechanism that enhances the representational power of the network by modeling channel relationships in a computationally efficient manner [12]. In addition, SE blocks can be integrated into traditional network architectures such as VGGNet [44] by inserting the nonlinearity operation after each convolution. Moreover, the number of parameters of SE module is very small, which will not cause network overfitting. These advantages make the SE block become a very popular attention processing component in current network frameworks. In this article, we utilize the SE attention mechanism to learn the relationship between channels and improve the classification performance.

III. Method

This section first introduces the structure of MSDNet in terms of scale and depth, and then introduces the SE attention module and the new network SEMSDNet that integrates the two components. Budgeted batch classification and anytime prediction methods are introduced as well to illustrate the modes of application under limited resource condition.

A. Scale Structure of the MSDNet

Different scale information has different levels of feature representation. It is of great significance to make full use of scale information in remote sensing scene classification. Fig. 2 shows the three scales of MSDNet's framework. The horizontal direction corresponds to the layer direction of the network, and deep learning model is performed continuously on the same scale to extract the depth features. The vertical direction corresponds to the scale of the feature maps. Each column represents a layer of the network. Horizontal arrows indicate a regular convolution operation. The diagonal and vertical arrows represent down-sample performed by the stride convolution to make the feature from fine to coarse with different scales. As shown in Fig. 2, the low-scale feature map x_1^1 is down sampled to obtain middle-scale feature map x_1^2. The middle-scale feature map x_2^2 summarizes the middle-scale map x_1^2 and low-scale feature map x_1^1. Feature maps of different scales can be aggregated by this way. In the horizontal direction, feature map x_2^1 is directly generated by x_1^1 with the same scale.

Table I shows the details of feature fusion in Fig. 2. Symbol x_n^s corresponds to feature map at the layer n and scale sth. h_n^s is a regular convolution operator, and $\tilde{h_n^s}(.)$ represents the down-sampling operation with stride convolution. The symbol [...] refers to the concatenation aggregation operation. Each feature map in the subsequent layers is a cascade of different scales. In the direction of scale and depth, the feature information of the network flows diagonally from the previous layer.

TABLE I
FEATURE AGGREGATION TABLE CORRESPONDING TO MULTISCALE STRUCTURE

x_n^s	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$\cdots$	$n=i$
$s=1$	$h_1^1(x_0^1)$	$h_2^1(x_1^1)$	$h_3^1(x_1^1, x_2^1)$	$h_4^1(x_1^1, x_2^1, x_3^1)$	$h_5^1(x_1^1, x_2^1, x_3^1, x_4^1)$	$\cdots$	
$s=2$	$\widetilde{h}_1^2(x_1^1)$	$\begin{bmatrix}\widetilde{h}_2^2([x_1^1]) \\ h_2^2([x_1^2])\end{bmatrix}$	$\begin{bmatrix}\widetilde{h}_3^2([x_1^1, x_2^1]) \\ h_3^2([x_1^2, x_2^2])\end{bmatrix}$	$\begin{bmatrix}\widetilde{h}_4^2([x_1^1, x_2^1, x_3^1]) \\ h_4^2([x_1^2, x_2^2, x_3^2])\end{bmatrix}$	$\begin{bmatrix}\widetilde{h}_5^2([x_1^1, x_2^1, x_3^1, x_4^1]) \\ h_5^2([x_1^2, x_2^2, x_3^2, x_4^2])\end{bmatrix}$	$\cdots$	
$s=3$	$\widetilde{h}_1^3(x_1^2)$	$\begin{bmatrix}\widetilde{h}_2^3([x_1^2]) \\ h_2^3([x_1^3])\end{bmatrix}$	$\begin{bmatrix}\widetilde{h}_3^3([x_1^2, x_2^2]) \\ h_3^3([x_1^3, x_2^3])\end{bmatrix}$	$\begin{bmatrix}\widetilde{h}_4^3([x_1^2, x_2^2, x_3^2]) \\ h_4^3([x_1^3, x_2^3, x_3^3])\end{bmatrix}$	$\begin{bmatrix}\widetilde{h}_5^3([x_1^2, x_2^2, x_3^2, x_4^2]) \\ h_5^3([x_1^3, x_2^3, x_3^3, x_4^3])\end{bmatrix}$	$\cdots$	$\begin{bmatrix}\widetilde{h}_i^3([x_1^2, x_2^2, x_3^2, x_4^2, \cdots, x_{i-1}^2]) \\ h_i^3([x_1^3, x_2^3, x_3^3, x_4^3, \cdots, x_{i-1}^3])\end{bmatrix}$

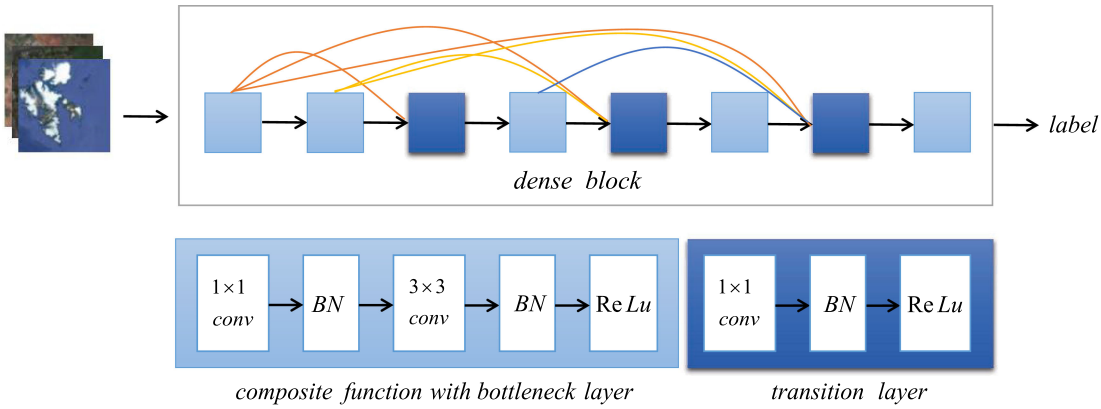

Fig. 3. Illustration of the dense structure with one block in the depth direction. The dense block includes five composite functions and two transition layers. The light blue module is the composite function with the bottleneck layer, and the deep blue module is the transition layer.

The horizontal connection saves and processes high-resolution information of remote sensing scene images, which is conducive to establish high-quality coarse features in subsequent layers. The vertical connection produces rough features throughout the stride convolution, which is convenient for remote sensing scene classification.

B. Depth Structure of the MSDNet

Traditional CNN produces fine features in early layers and coarse features in later layers [45]. Since the learning process is progressive, fine features tend to be lost in the process of convolution even if the skip connection is used. In addition, with the increase of network depth, the network complexity and training parameters also increase greatly. Therefore, for the classification of remote sensing scenes, the training time is often long and the convergence speed is slow. Moreover, the input information and the gradient tend to disappear when the multilayer transmission reaches the network terminal if the horizontal network structure is too deep. Therefore, when designing horizontal network, we must pay attention to the appropriate feature aggregation methods.

DenseNet uses dense connection to solve this problem [31]. By connecting all layers directly, it ensures maximum information flow between layers. Each layer receives information from the earlier layer as the input, and then transmits its own feature maps to the subsequent layers. In the depth direction, MSDNet utilizes dense connectivity to build the network framework (see Fig. 3). A dense block consists of composite functions with a bottleneck layer and a transition layer. The composite function with bottleneck layer includes BN, ReLU and convolutional layers with 1×1 and 3×3 convolution kernel, which defines the connection between input and output. The transition layer includes BN, ReLU, and 1×1 convolution kernel, which controls the number of channels and unifies the size of feature maps in each dense block. Therefore, there will be no problem of size inconsistency in concatenation. The model training process can be accelerated and the accuracy of remote sensing scene classification can be increased.

C. SE Block

MSDNet can be simply seen as a network framework based on multiscale DenseNet. Since DenseNet has strong feature representation in spatial domain, we introduce SE attention mechanism in the channel domain into the network model to learn the relationship between channels. With the help of channel attention mechanism, the network can automatically obtain the importance of each feature channel by learning, and then will improve or suppress the features according to their importance for the current task. A diagram illustrating the structure of an SE block is shown in Fig. 4. The SE attention mechanism includes two stages: "squeeze" and "excitation."

1) Squeeze: Global Information Embedding: In order to learn the dependencies between channels, the global average pooling method is used to compress the feature channels on the spatial dimension, and then, each feature channel is compressed into a real number with a global receptive field. Finally, all

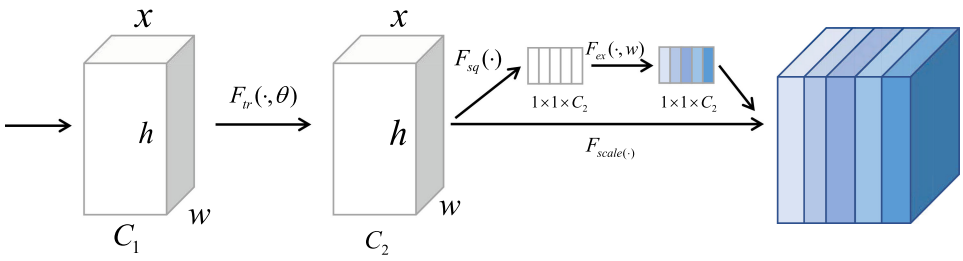

Fig. 4. SE block.

feature channels are compressed into a 1-D real number set. Each real number on the real number set represents a global feature channel, and the length of the real number set is the same as the number of feature channels. As shown in Fig. 4, F_{tr} is a normal convolution operation, mapping input X to feature maps U with the size of $H \times W \times C$. Equation (1) is the squeeze operation, where u_c is the c_{th} 2-D matrix in feature maps U, and the subscript c indicates the serial number of the channel. F_{sq} represents the squeeze operation, which converts the input feature maps U to the output with size of $1 \times 1 \times C$. C represents the number of feature channels, H and W are the height and width of the feature maps, and i and j are the elements of the feature maps, respectively.

$$z_c = F_{\mathrm{sq}}(u_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i,j) \tag{1}$$

2) Excitation—Adaptive Recalibration: The compression operation is equivalent to the analysis of the numerical distribution of feature maps, or global information. Then, the excitation operation is implemented, which is designed to fully capture the channel dependencies. As shown in (2), the excitation module exploits a gating mechanism with a sigmoid activation:

$$s_c = F_{\mathrm{ex}}(z, W) = \sigma(g(z, W)) = \sigma(W_2 \delta(W_1 z)) \tag{2}$$

where F_{ex} is the fully connected layer and z is the output of the squeeze operation. W_1 represents the full connection layer. The dimension of W_1 is $\frac{C}{r} \times C$ and r is the scaling parameter, whose function is to reduce the number of channels so as to reduce the amount of computation. Then, the activation function δ is used to activate the output with dimension unchanged, and then multiply the output with W_2. W_2 is also a fully connected layer, whose dimension is $C \times \frac{C}{r}$. As a result, the output dimension is $1 \times 1 \times C$. σ is the sigmoid function. Finally, as shown in (3), u_c is the original 2-D feature map. s_c is the output of the excitation operation, which also can be seen as the weight vector. $F_{\mathrm{scale}}(u_c, s_c)$ refers to channel-wise multiplication between weight vector s_c and feature map $u_c \in R^{H \times W}$. It is equivalent to assigning different weights to different feature channels.

$$\widetilde{x}_c = F_{\mathrm{scale}}(u_c, s_c) = s_c u_c \tag{3}$$

The parameters of the SE attention mechanism model are very small, only 0.22 M. It can be used as a pluggable tool in the mainstream network frameworks. Moreover, it can help the network to learn discriminant features meanwhile alleviate overfitting.

D. Framework of the SEMSDNet for Remote Sensing Scene Classification

Fig. 5 shows the overall framework of the proposed SEMSDNet with the scale structure and deep structure, where the first four layers are depicted in details. The horizontal direction corresponds to the depth (layer) direction of network, and the vertical direction corresponds to the scale of the feature maps. The horizontal arrows represent a regular convolution operation, whereas diagonal and vertical arrows represent a strided convolution operation. Classifiers operate on the second, third, and fourth layers. Fig. 6 shows the details of the horizontal structure of the SEMSDNet, which explains where the attention mechanism functions. The SE module is inserted after the transition layer of each dense block (depicted in Fig. 3) in the depth (horizontal) direction. Through this fusion mechanism, the network can not only realize the lossless transmission of the original input information, but also automatically learn the importance of each feature channel, which enables the feature channel adaptive calibration by enhancements and suppressions of the beneficial features and the useless features.

The structures and parameter settings of key layers of the network are described as follows.

1) *First layer.* The first layer ($n = 1$) is unique as it includes vertical connections. Its main purpose is to produce representations on all S scales. We can see the first layer's vertical layout as a miniature "S-layers" convolutional network. We denote the output feature maps at the layer n and scale s as x_n^s and the original input image as x_0^1. Feature maps with coarse scales are obtained by down-sampling. The first layer consists of the convolution operation with 3×3 convolution kernels, BN, and ReLU.
2) *Subsequent layers.* Subsequent layers produced the output feature maps x_n^s with scales s at subsequent layers n, which are a concatenation of feature maps from previous features maps with s and $s-1$. The nth layer of network outputs as a set of features at S scales $\{x_n^1, x_n^2, \ldots, x_n^S\}$.
3) *Classifiers.* The classifier at layer n with the coarsest scale utilizes all the features $[x_1^S, x_2^S, \ldots, x_n^S]$. Each classifier consists of two convolutional layers, followed by one average pooling layer and one linear layer. Classifiers are only attached to some of the intermediate layers,

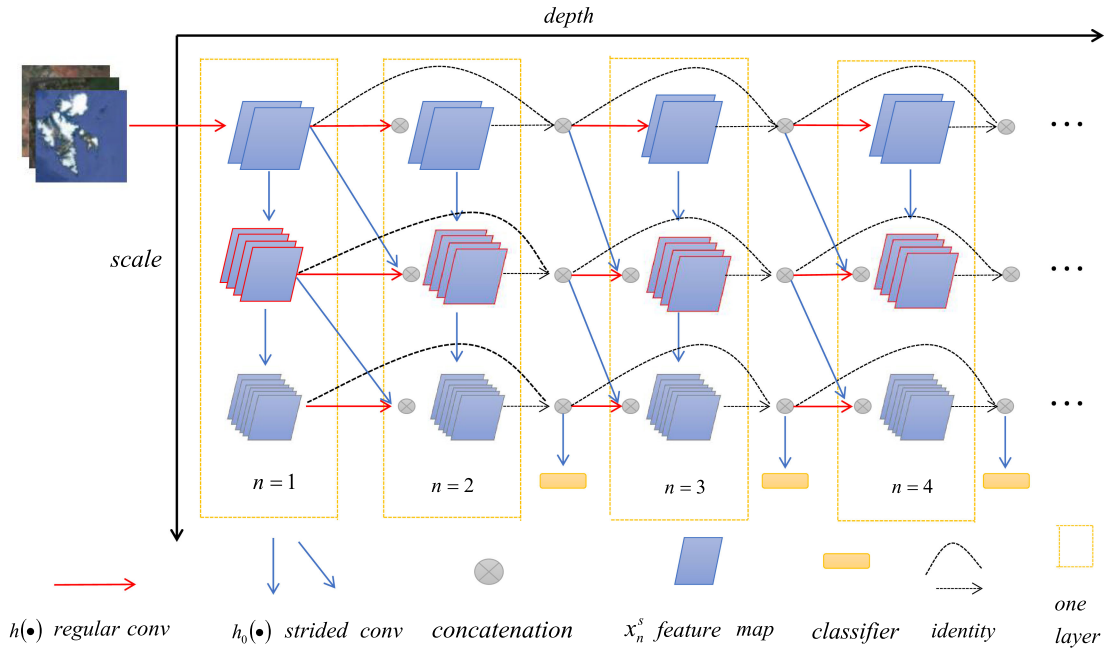

Fig. 5. Framework of the proposed SEMSDNet with the scale and deep structure.

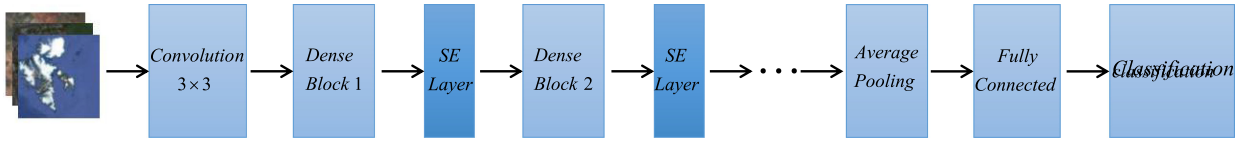

Fig. 6. Illustration of the horizontal structure of the proposed SEMSDNet. The SE layer is added after each transition layer in the depth direction.

and let $f_k(.)$ denote the kth classifier. The function of the classifier depends on the network settings. In the anytime setting at test time, we propagate an image through the network until we run out of budget and output the latest predictions. During testing in the batch budget setting, a probability threshold θ_k is set in advance. When the example traverses the network, if the prediction confidence of a certain classifier exceeds this threshold, the classification result can be directly output. Due to the limited experimental conditions, our main concern is the batch budget setting to save computing resources.

4) *Loss functions.* During the training, cross entropy loss functions $L(f_k)$ (f_k denotes the kth classifier) are used for all classifiers to minimize a weighted cumulative loss: $\frac{1}{|D|}\sum_{(x,y)\in D}\sum_k w_k L(f_k)$. D denotes the training set and w_k is the weight of the kth classifier. We use the same weight for all loss functions.

E. Budgeted Batch Classification

In the budgeted batch classification setting, the model needs to classify a batch of image samples $D_{\text{test}} = \{X_1, \ldots, X_M\}$ within a finite computational budget $B > 0$ that is known in advance. We aim to minimize the loss across all image samples in D_{test} within the computational budget B, which we denote by $L(f(D_{\text{test}}), B)$ for the loss function $L(.)$. SEMSDNet will probably spend less than B/M computation on classifying an "easy" image sample while use more than B/M computation on classifying a "difficult" one. Therefore, the budget B considered here is a soft constraint when we have a batch of testing image samples.

F. Anytime Prediction

In the anytime prediction setting, there is a finite computational budget $B > 0$ for each test image sample x. The computational budget is nondeterministic and varies according to each test sample. It is determined by whether the model needs to output a prediction immediately. We assume that the budget subject to some joint distribution $P(x, B)$. In some applications, $P(B)$ may be independent of $P(x)$ and can be estimated. We denote the loss of a model $f(x)$ that has to output a prediction for image sample s within budget B by $L(f(x), B)$. The goal of the anytime setting is to minimize the expected loss under the budget distribution: $L(f) = E[L(f(x), B)]_{P(x,B)}$. Here, $L(.)$ represents the loss function. The expectation under $P(x, B)$ can be estimated by an average over image samples from $P(x, B)$.

IV. Experiments

To verify the effectiveness of SEMSDNet model, we have performed a set of comprehensive experiments on three public benchmark datasets of remote sensing scenes, and the dataset

Fig. 7. Class samples of the UC Merced Land-Use dataset.

descriptions, experimental settings and results are presented in the following sections.

A. Dataset Descriptions

1) UC Merced Land-Use Data Set: The UC Merced Land-Use dataset [46] consists of 2100 images divided into 21 land-use classes, including 1) agricultural, 2) intersection, 3) mobile home park, 4) dense residential, 5) baseball diamond, 6) forest, 7) medium residential, 8) sparse residential, 9) chaparral, 10) overpass, 11) harbor, 12) buildings, 13) freeway, 14) parking lot, 15) storage tanks, 16) beach, 17) airplane, 18) runway, 19) golf course, 20) tennis courts, and 21) river. Each scene category contains 100 images with 256 × 256 pixels. Fig. 7 shows some examples of scenes in the UC Merced Land-Use dataset.

2) Aerial Image Dataset: The AID [5] is a big remote sensing image dataset with an image pixel size of 600 × 600. It contains 30 scene categories, each of which has about 220–420 aerial images, for a total of 10 000 aerial images, including 1) railway station, 2) school, 3) square, 4) storage tanks, 5) dense residential, 6) meadow, 7) forest, 8) park, 9) playground, 10) industrial, 11) baseball field, 12) center, 13) church, 14) farmland, 15) mountain, 16) port, 17) resort, 18) sparse residential, 19) beach, 20) parking, 21) commercial, 22) airport, 23) medium residential, 24) pond, 25) bridge, 26) river, 27) desert, 28) bare land, 29) stadium, and 30) viaduct. The dataset was published by the Huazhong University of Science and Technology and Wuhan University in 2017. Some examples of the AID are shown in Fig. 8.

3) NWPU-RESISC45 Dataset: The NWPU-RESISC45 dataset [6] contains 31 500 images, which are divided into 45 scene categories, including 1) basketball court, 2) baseball diamond, 3) beach, 4) church, 5) bridge, 6) chaparral, 7) harbor, 8) thermal power station, 9) sparse residential, 10) stadium, 11) medium residential, 12) railway, 13) parking lot, 14) desert, 15) railway station, 16) forest, 17) runway, 18) rectangular farmland, 19) ship, 20) freeway, 21) industrial area, 22) river, 23) cloud, 24) snow berg, 25) terrace, 26) golf course, 27) commercial area, 28) tennis court, 29) dense residential, 30) mobile home park, 31) meadow, 32) overpass, 33) storage tank, 34) roundabout, 35) circular farmland, 36) intersection, 37) island, 38) mountain, 39) sea ice, 40) ground track field, 41) airport, 42) lake, 43) palace, 44) airplane, and 45) wetland.

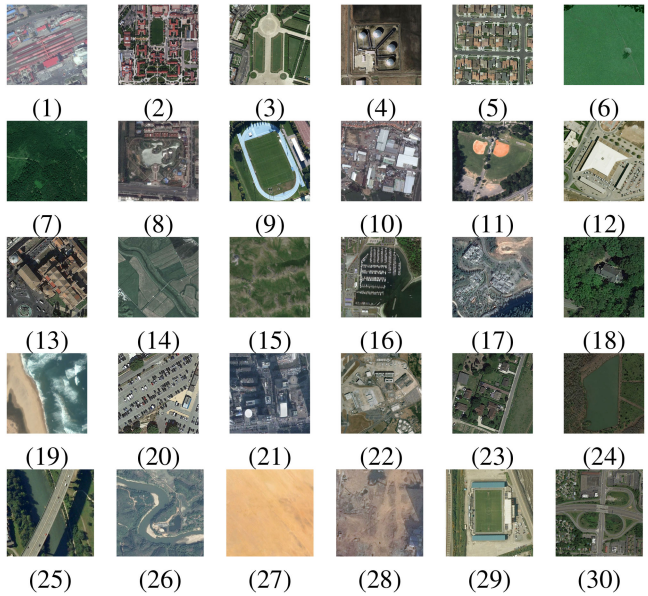

Fig. 8. Class samples of the AID dataset.

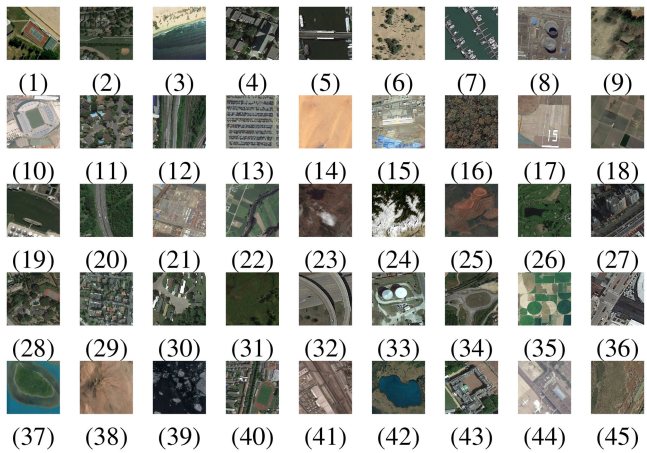

Fig. 9. Class samples of the NWPU-RESISC45 dataset.

The spatial resolution ranges from about 30 to 0.2 m/pixel. Each scene category contains 700 images with the size of 256 × 256 pixels. This dataset includes more than 100 urban areas worldwide. It is one of the largest remote sensing scene datasets in terms of the number and category of images with greater intraclass differences and interclass similarities than the former two datasets. Therefore, the classification of this dataset is the most difficult. Fig. 9 shows some examples of the NWPU-RESISC45 dataset.

B. Experimental Settings

1) Dataset Setting and Parameter Settings: We randomly divide the training set and the test set by conventional proportions. For the UC Merced Land-Use dataset, the training ratio is 80% and 50%, respectively. For the AID dataset, the training ratio is 50% and 20%, respectively. For the NWPU-RESISC45 dataset,

TABLE II
CLASSIFICATION PERFORMANCE OBTAINED BY DIFFERENT METHOD ON UC MERCED LAND-USE DATASET

Methods	80 % Training Ratio	50 % Training Ratio
CaffeNet [5]	95.02±0.81	93.98±0.67
GoogLeNet [5]	94.31±0.89	92.70±0.60
VGG-16 [5]	95.21±1.20	94.14±0.69
Fusion by addition [47]	97.42±1.79	/
CCP-net [48]	97.52±0.97	/
Two-Stream Fusion [7]	98.02±1.03	96.97±0.75
DSFATN [49]	98.25	/
GCFs+LOFs [50]	99±0.35	97.37±0.44
Inception-v3-CapsNet [51]	99.05±0.24	97.59±0.16
D-CNN [52]	98.93±0.10	/
SEMSDNet(ours)	**99.41±0.14**	**98.80±0.59**

the training ratio is 20% and 10%, respectively. These ratios have been selected based on previous studies in the literature so that we can compare with the latest methods. The training images are enhanced by horizontal flip and vertical flip.

In terms of network implementation, we use the PyTorch framework to implement our method. The network parameters and settings are as follows We use stochastic gradient descent and Nesterov momentum with a momentum weight of 0.9 to train all models. The training epochs are 150 and the batch size is 256. The initial learning rate is 0.1, which is divided by a factor of 10 after 40, 80, and 120 epochs. To facilitate the training, we resize all images to 256 × 256. In order to obtain reliable results on all datasets, we repeat the experiment 10 times for each training ratio, and calculate the average and standard deviation of the results. In addition, all experiments are implemented on a PC with CPU i7-10875H, 32 GB of RAM, 1 T SSD, and two GPUs (GTX 1080 Ti).

2) Evaluation Metrics: We use the overall accuracy and confusion matrix as the evaluation metric, which are also the two most common quantitative evaluation metrics of remote sensing scene classification. Overall accuracy is defined as the total number of correctly classified images divided by the number of images, which reflects the overall classification performance of the network. A confusion matrix is an information table used to analyze errors and confusions between different categories. It is obtained by counting the correct and incorrect classifications of each type of test images, and accumulating the results in the table. The confusion matrix can show the detail classification results. It focuses on the wrong classification, and it is a supplementary evaluation tool for overall accuracy. In addition, in order to evaluate the performance of the model, we choose the computation amount as the evaluation metric. The computation amount refers to the number of floating point operations for a complete forward propagation of the network model, which cannot fully represent the length of the prediction time of the model.

C. Experiment Results

1) Experiment Results of UC Merced Land-Use Dataset: As shown in Table II, after adding the attention module, the proposed SEMSDNet achieves the highest classification performance with 99.41% and 98.80% for the 80% and 20% training ratios, respectively. Furthermore, as shown in Fig. 10, 19 scene

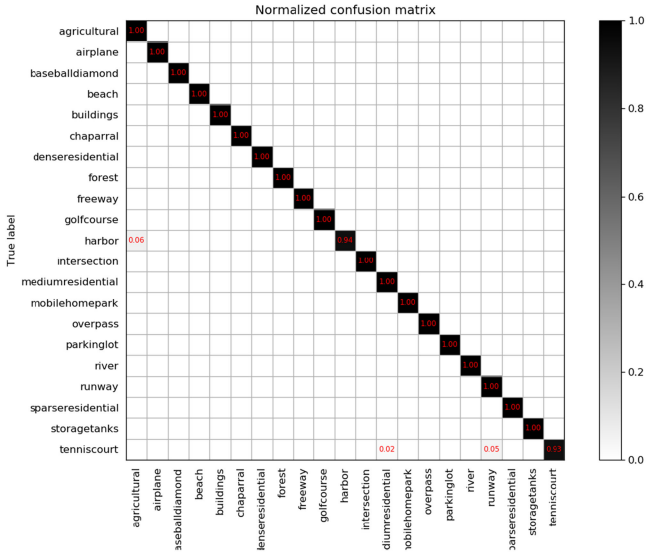

Fig. 10. Confusion matrix of the UC Merced Land-Use dataset under the training ratio of 80% using SEMSDNet.

TABLE III
CLASSIFICATION PERFORMANCE OBTAINED BY DIFFERENT METHOD ON AERIAL DATA SET

Methods	50 % Training Ratio	20 % Training Ratio
CaffeNet [5]	89.53±0.31	86.86±0.47
GoogLeNet [5]	86.39±0.55	83.44±0.40
VGG-16 [5]	89.64±0.36	86.59±0.29
TEX-Net-LF [53]	92.96±0.18	90.87±0.11
Fusion by addition [47]	91.87±0.36	/
Two-Stream Fusion [7]	94.58±0.25	92.32±0.41
GCFs+LOFs [50]	96.85±0.23	92.48±0.38
Inception-v3-CapsNet [51]	96.32±0.12	93.79±0.13
D-CNN [52]	96.89±0.10	90.82±0.16
SEMSDNet(ours)	**97.64±0.51**	**94.23±0.63**

categories are classified correctly in the test set. There are only two scene categories misclassified (harbor is misclassified as agricultural; tennis court is misclassified as runway). Most scene categories can be classified correctly.

2) Experiment Results of AID dataset: Table III shows the overall accuracy obtained by several different models. The SEMSDNet obtains the best performance with overall accuracy of 97.64% and 94.23% for the 50% and 20% training ratios, respectively. From the confusion matrix in Fig. 11, the five categories, which are school, center, park, resort, and square, are easily misclassified. This is due to the high similarity among these categories, which is difficult to distinguish even for human beings.

3) Experiment Results of NWPU-RESISC45 Dataset: As shown in Table IV, the proposed SEMSDNet method achieves the best classification performance of 93.89% and 91.68% for the 20% and 10% training ratios, respectively, which shows that the network architecture is very effective in extracting the category features of remote sensing scene images compared with the contrast algorithms. The confusion matrix in Fig. 12 shows that some pairs of categories (such as wetlands and lakes, terraces and rectangular farmland, palaces, and churches, etc.) are easily confused due to their high interclass similarity. In addition, some

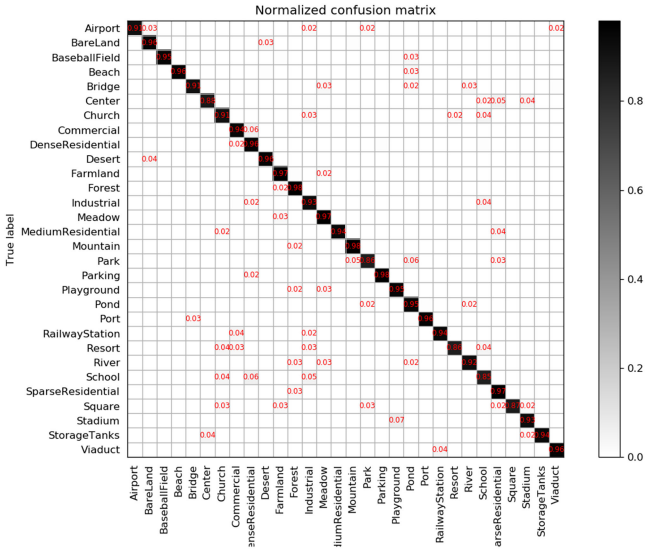

Fig. 11. Confusion matrix of the AID under the training ratio of 50% using SEMSDNet.

TABLE IV
CLASSIFICATION PERFORMANCE OBTAINED BY DIFFERENT METHOD ON NWPU-RESISC45 DATASET

Methods	**20 % Training Ratio**	**10 % Training Ratio**
GoogLeNet [19]	78.48±0.26	79.19±0.38
VGG-16 [19]	79.79±0.15	76.47±0.18
AlexNet [19]	79.85±0.13	76.69±0.21
Two-Stream Fusion [7]	83.16±0.18	80.22±0.22
BoCF [6]	83.16±0.18	82.65±0.31
Fine-tuned AlexNet [19]	85.16±0.18	81.22±0.19
Fine-tuned GoogLeNet [19]	86.02±0.18	82.57±0.12
Fine-tuned VGG-16 [19]	90.36±0.18	87.15±0.45
Triple networks [54]	92.33±0.20	/
Inception-v3-CapsNet [51]	92.60±0.11	89.03±0.21
D-CNN [52]	91.89±0.22	89.22±0.50
SEMSDNet(ours)	**93.89±0.63**	**91.68±0.39**

categories (such as thermal power station, palace, and freeway) are still hard to classify due to the high intraclass diversity.

4) Per-Class Comparisons: We draw a per-class histogram to better analyze the performance of our proposed network SEMSDNet on different scene categories. We take the AID dataset for example and employ two benchmark models for comparisons. From the per-class results on the AID dataset (Fig. 13), we can see that our method (SEMSDNet) outperforms other methods on distinguishing difficult scene categories. With the SE attention mechanism, our method can learn discriminative features and obtain better classification results. Some difficult scenes such as church, industrial, park, and river can be well distinguished. Particularly, tremendous improvements can be observed on some classes such as center, dense residential, resort, and square.

5) Ablation Studies for SEMSDNet: In order to verify the effect of the SE attention mechanism, we perform ablation experiments on the AID dataset and NWPU-RESISC dataset. Table V shows the classification results. For both datasets, obvious enhancement of classification accuracies can be observed under different training ratios. For example, the classification

TABLE V
ABLATION STUDY OF SEMSDNET ON THE AID AND NWPU-RESISC45 DATASET

	MSDNet (without SE Attention)	SEMSDNet
AID(50%)	95.75±0.30	97.64±0.51
AID(20%)	92.36±0.11	94.23±0.63
NWPU-RESISC45(20%)	89.14±0.31	93.89±0.63
NWPU-RESISC45(10%)	87.71±0.48	91.68±0.39

TABLE VI
COMPARISON OF MODEL PARAMETERS, COMPUTATIONAL COMPLEXITY AND PREDICTION TIME OF DIFFERENT MODELS ON AID TEST SET

Methods	Parameter Quantity (MB)	Computational Complexity (GFLOPS)	Prediction Time (s)
SEMSDNet	3.66	2.11	0.93
DenseNet	7.98	3.76	0.63
ResNet-18	11.69	8.75	0.81

accuracies of SEMSDNet are about 2% higher than MSDNet for AID dataset, and 4% for NWPU-RESISC dataset. This improvement is due to the fact that the SE attention mechanism enhances the representation of important features and suppresses the representation of less useful features.

6) Model Complexity and Lightweight Analysis: Theoretically and empirically, the deeper the network layer is, the deeper the semantic features can be extracted, which will be more helpful for the classification task. However, as the network layer deepens, the number of model parameters will continue to increase, and the computational complexity will also increase. Therefore, when we design the network structure, a tradeoff should be made between the classification ability of the model and the computational complexity. Table VI shows the parameter indexes of different network models on the AID dataset, including the number of parameters, the amount of computation, and the prediction time. The computation amount refers to the number of floating point operations for a complete forward propagation of the network model, which cannot fully represent the length of the prediction time of the model. The prediction time refers to the test time when an image is input under the condition that the computing platform is consistent, which depends on the computing power and the upper limit of bandwidth of the platform. In this article, RTX2080TI is selected as the computation platform for the experiments. The number of model parameters and the computational complexity are calculated through the OpCounter module of the open source code. As shown in Table VI, the SEMSDNET model has the lowest number of parameters, while other involved baseline models has much more parameters and higher computational complexity. The three models are basically on the same level in terms of the prediction time.

7) Analysis Under Budgeted Batch Classification Setting: In order to evaluate the classification performance of different networks in the budget batch classification setting, a comparative experiment including ResNet-18, DenseNet-121, and SEMSDNet is conducted on the AID test set, and the results are shown in Fig. 14. The abscissa represents the total limit B of computing resources for a batch of test images, and the ordinate represents the classification results. It can be seen from the figure that the

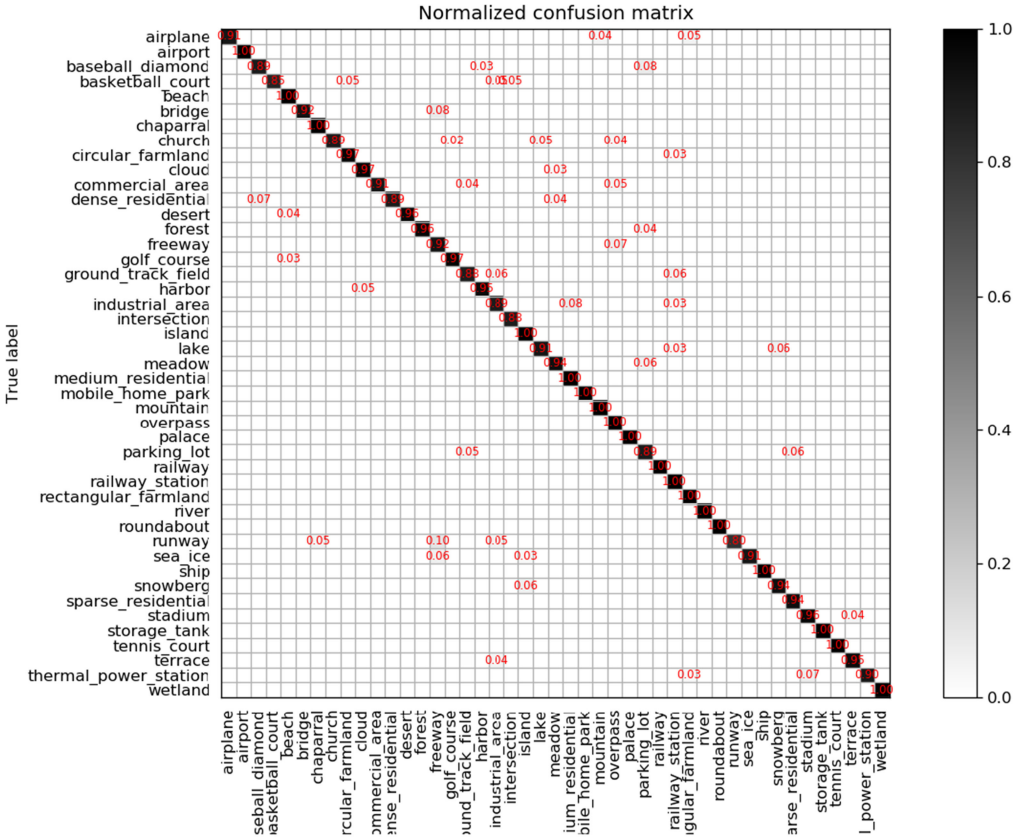

Fig. 12. Confusion matrix of the NWPU-RESISC45 dataset under the training ratio of 20% using SEMSDNet.

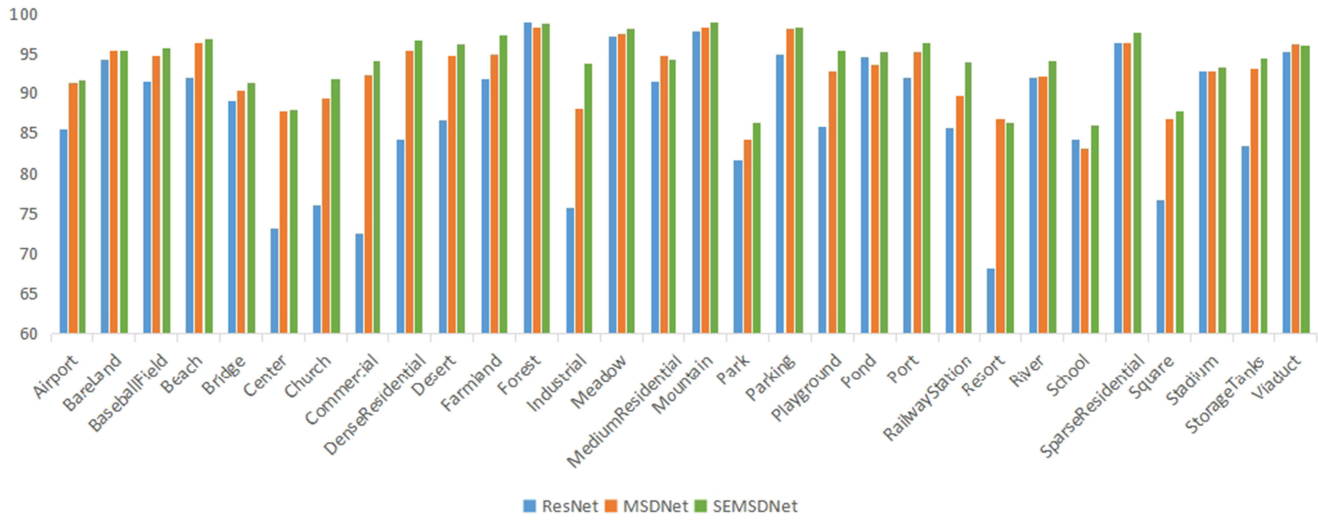

Fig. 13. Per-class accuracies of the proposed method and two benchmark references on the AID dataset.

classification performance of SEMSDNet with dynamic evaluation is better than that of ResNet and DenseNet with the same amount of computation. For example, when the average budget is 1.25×10^8 flops, the classification accuracy of the SEMSDNet reaches about 97%, which is 6% higher than that of the DenseNet and about 7% higher than ResNet. The DenseNet and ResNet use two times more computation than the SEMSDNet to achieve the same classification accuracy. It can be seen that when the average budget is small, the classification performance of SEMSDNet is much better than those of ResNet-18 and DenseNet-121.

8) Analysis Under Anytime Prediction Setting: In order to evaluate the classification performance of different networks in real-time prediction settings, we also select ResNet-18 and DenseNet-121 as baseline methods to carry out comparative

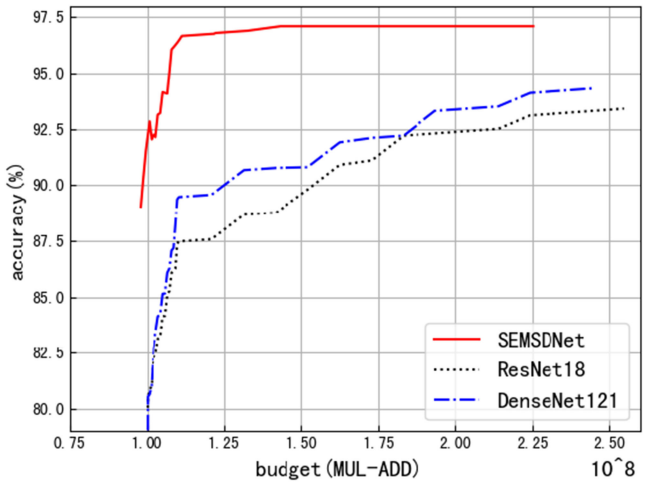

Fig. 14. Results of different networks on AID datasets under budgeted batch classification setting.

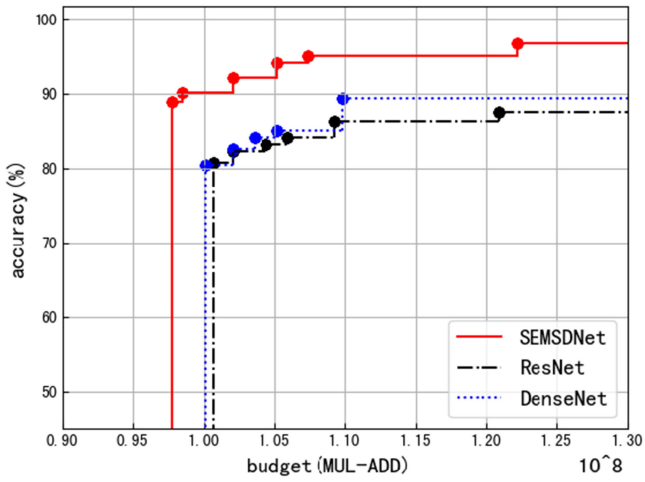

Fig. 15. Results of different networks on AID datasets under anytime prediction setting.

experiments on AID test set. The results are shown in Fig. 15. It is seen that under the same computation amount, the classification accuracy of the SEMSDNet model is much higher than ResNet-18 and DenseNet-121. When the average budget is 1.0×10^8 flop, the classification accuracy of SEMSDNet reaches about 90%, which indicates that compared with ResNet-18 and DenseNet-121, SEMSDNet model can output better classification results before computing resources are exhausted.

9) Predictive Visualization: To illustrate the ability of our approach on balancing the computational requirements of classifying examples of different complexities, we show six randomly sampled test images from AID dataset classes in Fig. 16. The top row shows "easy" image examples that are correctly classified and output by the first classifier. The bottom row shows "difficult" image examples that have been incorrectly classified by the first classifier but correctly classified by the latter classifiers. This figure suggests that early classifiers can recognize easy examples and leave the difficult ones for the deeper classifiers

Fig. 16. Sampled images of different complexities from AID dataset.

in accordance with our cognition. In other words, our proposed method is able to consider the difficulty and semantic level of scene images, and realize the automatic adaptation on scene classification.

V. Conclusion

In this article, we propose a new SEMSDNet model for remote sensing scene classification. With the architecture of multiscale dense connection and multiple classifiers, the MSDNet framework can automatically use small and simple networks when test images are easy or computational resources are limited, and use deep and complex networks when test images are hard or resources are is abundant. In order to avoid the interference of redundant features of remote sensing scene images, we introduce the SE attention mechanism to explicitly model the interdependence between the channels and automatically enhance the features of vital importance. Considering the limited computing resources of practical applications, we impose two settings with computational constraints at test time: budgeted batch classification and anytime prediction. Experimental results on multiple public datasets have validated that the proposed SEMSDNet model is superior to several baseline and state-of-the-art classification methods on classification accuracy, model lightweight, and computation requirement under different conditions. It is observed that some difficult scene images including complex semantic information are still misclassified. In future work, we will focus on extracting better semantic features and designing more effective networks to improve the performance of remote sensing scene classification.

References

[1] W. Tong, W. Chen, W. Han, X. Li, and L. Wang, "Channel-attention-based DenseNet network for remote sensing image scene classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 4121–4132, Jul. 2020.

[2] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[3] B. Zhang, Y. Zhang, and S. Wang, "A lightweight and discriminative model for remote sensing scene classification with multidilation pooling module,"

IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 12, no. 8, pp. 2636–2653, Aug. 2019.

[4] J. Xie, N. He, L. Fang, and A. Plaza, "Scale-free convolutional neural network for remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6916–6928, Sep. 2019.

[5] G. 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.

[6] G. Cheng, Z. Li, X. Yao, L. Guo, and Z. Wei, "Remote sensing image scene classification using bag of convolutional features," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 10, pp. 1735–1739, Oct. 2017.

[7] J. Hu, G.-S. Xia, F. Hu, H. Sun, and L. Zhang, "A comparative study of sampling analysis in scene classification of high-resolution remote sensing imagery," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2015, pp. 2389–2392.

[8] Y. Yu, Z. Gong, C. Wang, and P. Zhong, "An unsupervised convolutional feature fusion network for deep representation of remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 1, pp. 23–27, Jan. 2018.

[9] P. Zhong, Z. Gong, S. Li, and C. B. Schonlieb, "Learning to diversify deep belief networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 6, pp. 3516–3530, Jun. 2017.

[10] X. Yao, J. Han, G. Cheng, X. Qian, and L. Guo, "Semantic annotation of high-resolution satellite images via weakly supervised learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3660–3671, Jun. 2016.

[11] G. Huang, D. Chen, T. Li, F. Wu, L. van der Maaten, and K. Q. Weinberger, "Multi-scale dense networks for resource efficient image classification," 2018, *arXiv:1703.09844*.

[12] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, 2019.

[13] C. Xu, G. Zhu, and J. Shu, "A lightweight and robust lie group-convolutional neural networks joint representation for remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TGRS.2020.3048024.

[14] H. Li, H. Gu, Y. Han, and J. Yang, "Object-oriented classification of high-resolution remote sensing imagery based on an improved colour structure code and a support vector machine," *Int. J. Remote Sens.*, vol. 31, no. 6, pp. 1453–1470, 2010.

[15] E. Aptoula, "Remote sensing image retrieval with global morphological texture descriptors," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 3023–3034, May 2014.

[16] N. He, L. Fang, S. Li, and A. J. Plara, "Covariance matrix based feature fusion for scene classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2018, pp. 3587–3590.

[17] L. Fang, N. He, S. Li, P. Ghamisi, and J. A. Benediktsson, "Extinction profiles fusion for hyperspectral images classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1803–1815, Mar. 2018.

[18] 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.

[19] C. Gong, 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.

[20] A. M. Cheriyadat, "Unsupervised feature learning for aerial scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 439–451, Jan. 2014.

[21] J. Zou, W. Li, C. Chen, and Q. Du, "Scene classification using local and global features with collaborative representation fusion," *Inf. Sci.*, vol. 348, no. 348, pp. 209–226, 2016.

[22] F. Zhang, B. Du, and L. Zhang, "Saliency-guided unsupervised feature learning for scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 2175–2184, Apr. 2015.

[23] M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva, "Land use classification in remote sensing images by convolutional neural networks," *Acta Ecologica Sinica*, vol. 28, no. 2, pp. 627–635, 2015.

[24] Q. Liu, R. Hang, H. Song, and Z. Li, "Learning multi-scale deep features for high-resolution satellite image classification," 2016, *arXiv:1611.03591*.

[25] X. Suhui, M. Xiaodong, Z. Peng, and M. Ji, "Scene classification of remote sensing image based on multi-scale feature and deep neural network," *Acta Geodaetica et Cartographica Sinica*, vol. 45, no. 7, pp. 834–840, 2016.

[26] Y. Liu, Y. Zhong, and Q. Qian, "Scene classification based on multiscale convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 12, pp. 7109–7121, Dec. 2018.

[27] Z. Yang, X.-D. Mu, and F.-A. Zhao, "Scene classification of remote sensing image based on deep network and multi-scale features fusion," *Optik*, vol. 171, pp. 287–293, 2018.

[28] X. Wang, S. Wang, C. Ning, and H. Zhou, "Enhanced feature pyramid network with deep semantic embedding for remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TGRS.2020.3044655.

[29] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.

[30] A. R. Zamir *et al.*, "Feedback networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1308–1317.

[31] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.

[32] X. Mei *et al.*, "Spectral-spatial attention networks for hyperspectral image classification," *Remote Sens.*, vol. 11, no. 8, p. 963, 2019.

[33] B. A. Olshausen, C. H. Anderson, and V. Essen, "A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information." *J. Neurosci.*, vol. 13, no. 11, pp. 4700–19, 1993.

[34] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[35] M. Jaderberg, K. Simonyan, K. Zisserman, Andrew, and Koray, "Spatial transformer networks," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015 pp. 2017–2025.

[36] T. Bluche, "Joint line segmentation and transcription for end-to-end handwritten paragraph recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 838–846.

[37] A. Miech, I. Laptev, and J. Sivic, "Learnable pooling with context gating for video classification," 2017, *arXiv:1706.06905*.

[38] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, "Lip reading sentences in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3444–3453.

[39] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.

[40] X. Tang, Q. Ma, X. Zhang, F. Liu, J. Ma, and L. Jiao, "Attention consistent network for remote sensing scene classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 2030–2045, 2021.

[41] F. Wang *et al.*, "Residual attention network for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6450–6458.

[42] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.

[43] 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.

[44] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[45] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.

[46] Q. Weng, Z. Mao, J. Lin, and W. Guo, "Land-use classification via extreme learning classifier based on deep convolutional features," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 5, pp. 704–708, 2017.

[47] 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.

[48] K. Qi, Q. Guan, C. Yang, F. Peng, S. Shen, and H. Wu, "Concentric circle pooling in deep convolutional networks for remote sensing scene classification," *Remote Sens.*, vol. 10, no. 6, 2018, Art. no. 934.

[49] X. Gong, Z. Xie, Y. Liu, X. Shi, and Z. Zheng, "Deep salient feature based anti-noise transfer network for scene classification of remote sensing imagery," *Remote Sens.*, vol. 10, no. 3, 2018, Art. no. 410.

[50] D. Zeng, S. Chen, B. Chen, and S. Li, "Improving remote sensing scene classification by integrating global-context and local-object features," *Remote Sens.*, vol. 10, no. 5, 2018, p. 734.

[51] X. Lu, H. Wu, and Y. Yuan, "Double constrained NMF for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2746–2758, May 2014.

[52] 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.

[53] F. Zhang, B. Du, and L. Zhang, "Scene classification via a gradient boosting random convolutional network framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1793–1802, Mar. 2016.
[54] Y. Liu and C. Huang, "Scene classification via triplet networks," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 1, pp. 220–237, Jan. 2018.

Tian Tian received the B.S. degree in electronic information engineering and Ph.D. degrees in control science and engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2009 and 2015, respectively.

From 2012 to 2014, she was visiting Oakland University, MI, USA, as a Ph.D. Candidate sponsored by China Scholarship Council. In 2015, she joined the School of Computer Sciences, China University of Geosciences, Wuhan, China, as a Postdoctoral Lecturer. She is currently an Associate Professor with the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology. Her major research interests include remote sensing image processing and computer vision and applications.

Lingling Li received the B.S. degree from Wuhan University, Wuhan, China, in 2018. She is currently working toward the master's degree in computer science and technology with the School of Computer Sciences, China University of Geosciences, Wuhan, China.

Her main research interests include computer vision, machine learning, and remote sensing image classification.

Weitao Chen was born in Wugang, Henan Province, China. He received the B.E. degree from the Jiaozuo Institute of Technology, Jiaozuo, China, in 2003, and the M.E. and Ph.D degrees in environmental science and engineering from China University of Geosciences (CUG), Wuhan, China in 2006 and 2012, respectively.

He is currently an Associate Professor with the School of Computer Science, CUG. His main research interests include machine learning and remote sensing of environment.

Huabing Zhou received the B.S. and M.S. degrees in computer science and technology from the Wuhan Institute of Technology, Wuhan China, in 2005 and 2008, respectively, and the Ph.D. degree in control science and engineering form the Huazhong University of Science and Technology, Wuhan, China, in 2012.

From 2009 to 2010, he was a Research Intern with the Chinese Academy of Surveying and Mapping. From 2018 to 2019, he was a Visiting Scholar with Temple University, Philadelphia, PA, USA. He is currently an Associate Professor with the School of Computer Science and Engineering, Wuhan Institute of Technology. His research interests include computer vision, remote sensing image analysis, and intelligent robot.