CSDS: End-to-End Aerial Scenes Classification With Depthwise Separable Convolution and an Attention Mechanism

Xinyu Wang, Liming Yuan, Haixia Xu, and Xianbin Wen

***Abstract*—Compared with natural scenes, aerial scenes are usually composed of numerous objects densely distributed within the aerial view, and thus, more key local semantic features are needed to describe them. However, when existing CNNs are used for remote sensing image classification, they typically focus on the global semantic features of the image, and especially for deep models, shallow and intermediate features are easily lost. This article proposes a channel–spatial attention mechanism based on a depthwise separable convolution (CSDS) network for aerial scene classification to solve these challenges. First, we construct a depthwise separable convolution (DS-Conv) and pyramid residual connection architecture. DS-Conv extracts features from each channel and merges them, effectively reducing the number of necessary calculations, and the pyramid residual connections connect the features from multiple layers and create associations. Then, the channel–spatial attention algorithm causes the model to obtain more effective features in the channel and spatial domains. Finally, an improved cross-entropy loss function is used to reduce the impact of similar categories on backpropagation. Comparative experiments on three public datasets show that the CSDS network can achieve results comparable to those of other state-of-the-art methods. In addition, visualization of feature extraction results by the Grad-CAM algorithm and ablation experiments for each module reflect the powerful feature learning and representation capabilities of the proposed CSDS network.**

***Index Terms*—Channel–spatial attention, convolutional neural network (CNN), depthwise separable convolution (DS-Conv), scene classification.**

I. Introduction

REMOTE sensing and Earth observation, also called Earth vision, are important branches and applications of computer vision and image understanding [1]–[3]. With the rapid development of this field and the widespread use of satellite sensing technology in everyday life, aerial scene classification has received increasing attention [4] as an important application that has influenced the development of many fields, such as land use and land cover [5], [6], urban design [7], and vegetation surveying and mapping [8].

Manuscript received June 26, 2021; revised August 20, 2021 and September 13, 2021; accepted October 1, 2021. Date of publication October 6, 2021; date of current version October 27, 2021. This work was supported in part by the New-Generation AI Major Scientific and Technological Special Project of Tianjin (18ZXZNGX00150) and in part by the Special Foundation for Technology Innovation of Tianjin (21YDTPJC00250). *(Corresponding authors: Liming Yuan; Xianbin Wen.)*

The authors are with the School of Computer Science, and Engineering, and Key Laboratory of Computer Vision, and System of the Ministry of Education, Tianjin University of Technology, Tianjin 300384, China (e-mail: joywang1225@sina.cn; yuanleeming@163.com; xuhaixiax@163.com; wenxianbin@163.com).

Digital Object Identifier 10.1109/JSTARS.2021.3117857

A. Image Characteristics of Aerial Scenes

Aerial scene classification is challenging due to the characteristics of the sampled images.

1) Useless Background Information: The key object of the sample usually determines the label of the remote sensing image. However, there are often objects and areas in the image that have nothing to do with the actual label; these objects and areas are considered background information. Because remote sensing images are affected by factors, such as angle of view, illumination changes, and terrain [9], [10], it can be difficult to accurately locate key subjects within them. Therefore, to highlight the key objects and suppress redundant background information, local key features must be extracted to enhance the semantic representation of the aerial image. Research has shown that deeper network structures can be used to extract more semantic features with key information [11], [12].

2) Distribution of Key Objects: Remote sensing image acquisition is different from the plane acquisition of natural images, in part, because the position of the subject in the image is often random. In addition, due to the effect of gravity, the main direction angle of the key object in a natural scene image is often 90° with the ground. In contrast, the main direction angle of the key object in an aerial scene image can change greatly [see Fig. 1(b)]. Finally, due to the large shooting height and angle of aerial scenes, the distribution of key objects is different from the central distribution observed in natural scene images (see Fig. 1). These characteristics increase the difficulty in understanding remote sensing images. Some studies have shown that methods that are robust to changes in direction are usually suitable [3], [13], [14].

3) Complex Spatial Distribution: Aerial scenes typically contain many elements that have nothing to do with the true label of the image or many key elements that are either densely or diffusely scattered and randomly arranged at any position in the image [see Fig. 1(c)]. In contrast, because the key objects in natural images are centrally distributed, spatial structure information and key features are easier to capture. One effective way to simplify the extraction of key discriminative features from

Fig. 1. Natural scene image (left) and aerial scene image (middle and right). (a) Different background information. (b) Different object distribution. (c) Different spatial arrangement.

aerial images is to extract deep features while retaining some low-level features [15]–[17].

With the rapid development of deep learning technology, many proposed convolutional neural network (CNN) models have achieved impressive results in different fields. For aerial scenes, deep learning has demonstrated powerful feature extraction capabilities through the application of a number of classic CNN network, such as VGGNet, AlexNet [9], and GoogLeNet [10]. Certain improved networks have also achieved state-of-the-art performance [10], [13], [18], [19]. The success of CNN models demonstrates that deep features can be used to better describe images than traditional handcrafted features and midlevel features [2], [20], [21]; nevertheless, some problems remain.

1) Loss of Low-Level Features: Some commonly used CNN models, such as VGGNet and AlexNet, cannot retain shallow features during the training process. Unlike the deep semantic features used in scene classification, shallow features are not the key points that determine the final classification performance. However, retention of these features can help with extracting more discriminative features and improve classification performance. Some recently proposed methods for retaining low-level features are not end-to-end solutions and are, thus, difficult to adapt to different tasks and datasets [21], [22].

2) Weaknesses in Main Semantic Features: Due to the way they are obtained, remote sensing scenes usually contain different types of land cover. As shown in Fig. 2, from a human perspective, the characteristic information of the tennis court is the main basis for recognition, and other objects or backgrounds, such as grass and parking lots are secondary or irrelevant information [23]. However, traditional CNNs tend to focus on global semantics, making it difficult to extract the key features

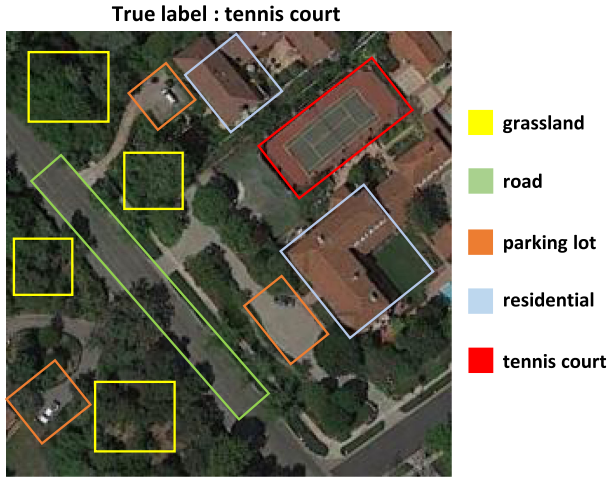


Fig. 2. Global and local semantics in an image of a tennis court.

of aerial scenes, which may reduce the ability to represent the scene and make it impossible to be accurately classified [11].

B. Motivation and Objectives

To solve the abovementioned problems, this article proposes a CNN for the classification of aerial scene images. The goals include the following.

1) Enhanced Local Key Feature Representation: Current CNN models are not sufficient for the representation of local semantic features. Due to the complex object distribution and spatial arrangement of remote sensing images, our model should demonstrate an improved ability to perform key feature extraction.

2) Fewer Calculations: Although increasing the number of CNN layers can improve the model's ability to express the depth semantic features of the image, more calculations will be required, resulting in high training costs and overfitting problems. Therefore, our model should minimize the amount of calculations and the number of parameters while ensuring efficient image feature extraction.

3) Retention of Features From Different Levels: Deep CNN models usually cannot retain shallower features, which could otherwise improve the discriminative power of the model. However, many existing solutions for retaining these shallow features are not flexible end-to-end architectures. Our model involves an end-to-end network for protecting different levels of functionality and effectively improving feature propagation and model classification performance.

In summary, we propose a new depth-separable convolution and residual connection network (CSDS) based on channel–spatial attention to achieve these goals. The research described in this article mainly includes the following aspects.

1) An end-to-end CNN model for aerial scene classification is proposed. Given that remote sensing scene image datasets are not large scale and the images therein contain complex semantic relations, our proposed CSDS uses a pyramid residual connection and a depthwise separable convolution (DS-Conv) architecture.In the spatial domain,

pyramid residual connection blocks can extract multilayer features and establish associations, making full use of limited samples. DS-Conv realizes decoupling channel correlation and spatial correlation, improves convolution efficiency, and greatly reduces the number of model training parameters and calculations.

2) A channel–spatial attention mechanism is used to improve the ability to extract key features for improving the overall accuracy (OA) of classification by suppressing the weights of secondary features.
3) The original cross-entropy function often only considers the highest index of a single category, easily leading to poor model generalization and overfitting. To reduce the impact of similar categories on the classification results, a cross-entropy loss function with a label smoothing constraint is added. This enables the model to focus on the loss of multiple categories, effectively suppress overfitting, and improve generalization.

The rest of this article is structured as follows. Section II introduces some related developments in the field. Section III introduces the proposed CSDS network in detail. Section IV reports and analyzes the relevant experimental results. In Section V, an ablation experiment is performed, and some network details are visualized. Finally, Section VI concludes this article and explains future research directions.

II. Related Work

A. Aerial Scene Classification

Based on the feature extraction method, remote sensing image classification algorithms can be roughly divided into low-level, middle-level, and deep feature extraction algorithms.

Some early research methods were mainly based on the extraction of low-level features, such as color (color histogram [9], [24], [25]), structure (scale-invariant feature transformation (SIFT) [26]), and texture (local binary pattern [27]). In addition, methods that fuse a variety of handcrafted features have achieved good results with images containing hyperspectral or spatial structure information [28], [29]. However, as the complexity of the remote sensing images increases, these low-level features become unable to assist in distinguishing categories.

Methods based on the extraction of middle-level features obtain global features by encoding the extracted local features and include bag-of-visual words [30], potential latent semantic analysis (PLSA) [31], and latent Dirichlet allocation [32]. However, these methods rely on a large amount of prior information and loose key local features and are, thus, not very suitable for remote sensing images.

Since 2010, deep learning methods (which, as the name implies, are based on deep features) have undergone rapid development. One group of deep learning methods, CNNs, has made remarkable achievements in a number of research fields, including aerial image scene classification [22], [33]–[36]. Compared with low- and middle-level features, deep features can better express the internal information within remote sensing images.

Remote sensing image classification methods based on deep learning can be divided into fine-tuning models, fully trained models, and convolutional feature extractors. Because CNNs require a large number of parameter weights during training, the sample size of aerial images is usually small. Therefore, fine-tuning methods tend to reuse models trained on larger target datasets (such as ImageNet) and incorporate changes to some experimental parameter settings. Some methods based on fine-tuning models have achieved good results [22], [37], [38]. Fully trained models are usually designed based on or outright use currently existing models for directly training with aerial image datasets (AIDs). The newly designed CNN model can extract features from aerial scenes, leveraging improvements to or redesigns of the classic convolutional neural algorithm to adapt to different remote sensing datasets. For example, DABNet [18] and PBNet [39] effectively solve the problem of difficult remote sensing image feature extraction due to the complex imaging principle, angle, and terrain, allowing them to outperform better than some existing CNNs [10], [13], [19]. Using a CNN as a feature extractor is another commonly used classification strategy; two examples include GLDBS [40] and TEX-Nets [41], which fuse deep features extracted from multiple convolutional layers. Although these methods are sometimes better than current CNN models, they are unable to describe local features and key objects.

B. Depthwise Separable Convolution

The CNN, proposed by LeCun *et al.* [42] in 1998, has been applied to numerous research field with remarkable results, particularly in the field of computer vision, where it has greatly surpassed traditional object classification and recognition algorithms [43]. In recent years, many deep learning-based methods have been used in remote sensing image processing. Hu *et al.* [22] used a pretrained CNN model as a feature extractor, comparing and verifying the impact of different layers on classification performance. Xu *et al.* [44] used a pretrained CNN model to extract features and fuse them with multilayer features to increase the global feature weight. Cheng *et al.* [45] proposed a feature extraction method named BoCF that constructs a visual vocabulary from the convolutional features of pretrained CNNs for aerial scene image classification. However, the better-performing CNN models usually rely on very large numbers of parameters and calculations, and very deep structures. In recent years, to improve efficiency and reduce costs, lightweight networks, such as MobileNet and ShuffleNet, have been produced. Notably, these models use DS-Conv, proposed by Sifre and Mallat[46], a convolution method that can effectively reduce the scale and number of parameters and calculations of the network while ensuring its accuracy. The general convolution operation is based on the joint mapping of channel correlation and spatial correlation of the 3-D filter (width, height, and channel) [47]. Different from traditional convolution, DS-Conv performs decoupled channel correlation and spatial correlation, which helps reduce computational complexity. Because CNNs often produce parameter redundancy, the accuracy loss from DS-Conv is minimal [46]. In the remote sensing field, Zhang *et al.* [48] used MobileNetV2 as the backbone and introduced channel attention to extract deep features and improve performance in

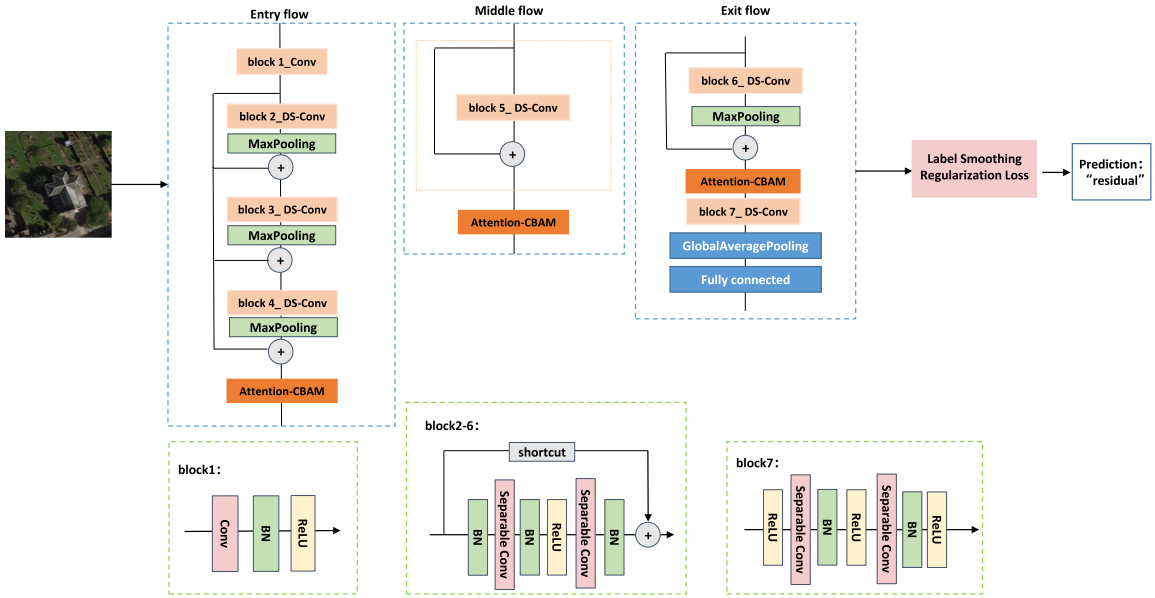


Fig. 3. Overall structure of the proposed CSDS model.

HSRRS image classification. Huang *et al.* [49] used light-CNN to extract features from the remote sensing image and obtained good accuracy, and an effective network scale and parameters.

C. Attention Mechanisms

The attention mechanism is an algorithm that can simulate human understanding and perception of images. It can quickly and accurately focus on the target object while suppressing useless information [23]. The development of the attention mechanism has provided new ideas for the field of remote sensing image research. Zhang *et al.* [50] used a saliency sampling strategy to extract the key features of remote sensing images. Hu *et al.* [26] fused the features of salient regions with key points to classify scene images. However, early saliency detection methods based on texture information cannot capture all key information well. Recently, some studies have proposed methods that can adaptively extract key attention features. Mnih *et al.* [51] combined the attention mechanism with a recurrent neural network, which reduced the difficulty of image feature extraction. Wang *et al.* [52] used the product of feature maps and attention weights to obtain high-level features describing image information, and proposed a recurrent attention network for scene classification. Haut *et al.* [53] used generated attention masks to multiply the corresponding regions to obtain attention features. However, these methods only add attention weights to a single domain and do not comprehensively consider channel and spatial domains. For the first time, Woo *et al.* [54] added attention weights to both the channel and spatial dimensions, which enabled the network to fully learn the key features of the images and achieved impressive results in classification tasks. However, due to the serious shortage of remote sensing image samples, the feature extractors in existing studies were typically pretrained models whose internal structure is difficult to change at will, thus limiting the development of scene classification.

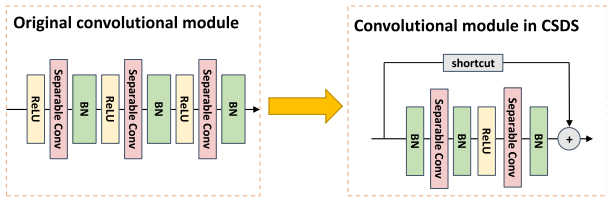


Fig. 4. Changes in the convolution modules.

III. Proposed Method

As shown in Fig. 3, the proposed CSDS network consists of attention layers, DS-Conv with pyramid residual connection layers, and a classifier containing a loss function based on label smoothing. To simplify model construction and visualization, we base the CSDS network on the structure of Xception [47] and divide it into three parts: 1) entry flow, 2) middle flow, and 3) exit flow. It should be noted that the overall size of the proposed network is much smaller than that of Xception. The Xception architecture consists of 36 convolutional layers that form the feature extraction base of the network, structured into 14 modules [47]. The original Xception model contains a large number of repeated linear stack convolution modules; we remove some redundant layers to obtain a smaller model and change the convolution module to a structure with pyramid residual connections, as shown in Fig. 4. Then, the channel–spatial attention module is added to better extract key features. The improved CSDS structure contains 13 convolutional layers and three attention layers, forming ten modules as the basis for feature extraction. Except for the first and last convolution modules, the seven remaining convolution modules are all pyramid residual connection structures and use an improved cross-entropy loss function with label smoothing for final classification.

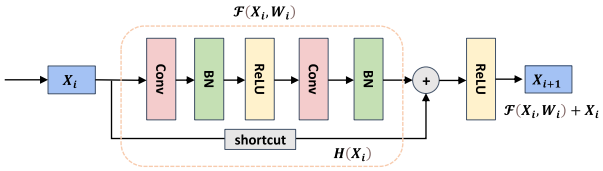


Fig. 5. Structure of the basic residual unit. Its execution order is Conv→BN→ReLU→Conv→BN.

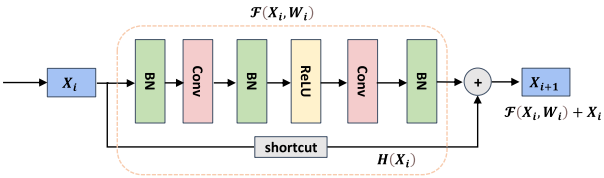


Fig. 6. Structure of the pyramid residual unit. Its execution order is BN → Conv→BN → ReLU→Conv → BN.

A. *Feature Extraction Backbone*

A very deep neural network has difficulty converging from the beginning and must contend with network degradation [55]. To solve these problems, we use the pyramid residual connection unit [56] in the network, shown in Fig. 5, where X_i represents the input to the residual unit, and X_{i+1} represents the output of the residual unit. F and H represent the residual operation and shortcut connection, respectively: if identity mapping is present, then $H(X_i) = X_i$. The mathematical meaning of the basic residual unit is as follows:

$$X_{i+1} = \mathcal{F}(X_i, W_i) + X_i. \tag{1}$$

Shortcut connections increase the depth of the network to a certain extent and improve the deep feature extraction ability of the network, and the diversity of advanced attributes while adding a reasonable number of parameters, accelerating the training efficiency, and effectively suppressing the network degradation problem [56]. The pyramid residual unit can be considered an improvement to the residual connection unit, offering greater advantages and better performance to the model structure, while deleting the last ReLU [57]. Batch normalization (BN) [58] is required before the first convolution operation in the residual pyramid unit. Han *et al.* [56] demonstrated that a large number of ReLUs will reduce the performance of the model; therefore, unnecessary ReLUs are deleted, and the ReLUs between the two convolution modules are retained to ensure nonlinearity. Adding the BN layer before convolution can improve the capabilities of the network structure and speed up convergence. The order of execution can be described as BN → Conv→BN → ReLU→Conv → BN (shown in Fig. 6). Note that we replace the standard convolution with DS-Conv, which helps to further reduce the number of parameters and calculations.

DS-Conv can be divided into two parts: 1) depthwise convolution and 2) point-by-point convolution (1×1 convolution). Fig. 7(b) shows the feature map processing steps. First, the input image is decoupled from its channel and spatial correlations through depthwise convolution, and each channel is convolved

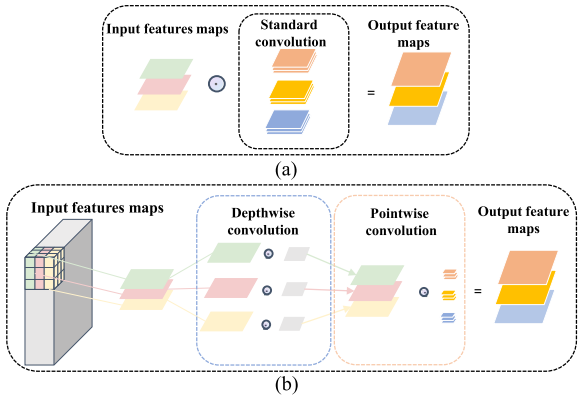


Fig. 7. Comparison of standard convolution and DS-Conv structures.

separately. Then, the features of each channel are combined through a 1×1 standard convolution.

DS-Conv effectively reduces the number of parameter calculations. A standard convolution operation [see Fig. 7(a)] can be defined as follows:

$$G_i = \sum_{i=1}^{M} H_i \cdot K_j^i + b_i, i = 1, 2, \ldots, N \tag{2}$$

where H_i is the ith mapping in input feature map H of size $D_{\text{input}} \times D_{\text{input}}$, G_i is the ith mapping in output feature map G of size $D_{\text{output}} \times D_{\text{output}}$, and b_i is the bias of G_i. K_j^i is the ith slice in the jth kernel, and the mathematical symbol $\cdot$ represents the convolution operation. The number of input channels and the number of output channels are M and N, respectively. If the size of the standard convolution kernel K is $k \times k$, then the total number of parameters P_1 and number of calculations F_1 can be expressed as

$$\begin{aligned} P_1 &= k \times k \times M \times N \\ F_1 &= k \times k \times M \times N \times D_{\text{output}} \times D_{\text{output}}. \end{aligned} \tag{3}$$

As shown in Fig. 7(b), the complete feature extraction process for DS-Conv is divided into two steps. The first step is a depthwise convolution based on a single channel, expressed as follows:

$$G_i' = H_i \cdot K_j + b_j, i, j = 1, 2, \ldots, M \tag{4}$$

where K_j is the jth depthwise convolutional kernel; it only filters the input channels without combining them to create new features. Therefore, the second step is the generation of the final features through a standard 1×1 convolutional layer connection [47], [59]. The overall number of parameters P_2 and calculations F_2 of DS-Conv are obtained by combining the abovementioned two steps and can be expressed as

$$\begin{aligned} P_2 &= k \times k \times M + M \times N \\ F_2 &= k \times k \times D_{\text{output}} \times D_{\text{output}} \times M \\ &\quad + D_{\text{output}} \times D_{\text{output}} \times M \times N. \end{aligned} \tag{5}$$

Then, the ratio of the number of calculations for the conventional convolution to that of the DS-Conv can be expressed as

$$\frac{P_2}{P_1} = \frac{1}{N} + \frac{1}{k^2}$$
$$\frac{F_2}{F_1} = \frac{1}{N} + \frac{1}{k^2}. \tag{6}$$

These simple calculations show that DS-Conv contains fewer parameters and calculations than the standard convolution. Therefore, this convolution method can reduce storage space requirements, computation time, and hardware computing power requirements.

B. Channel–Spatial Feature Attention Block for Feature Refinement

If the residuals connect features from different layers, the extracted feature parameters will be redundant. The attention mechanism can selectively focus on the main object in the image and extract key features while removing redundant information [23]. To improve the model's ability to adaptively extract key features and enrich the diversity of advanced features, Woo *et al.* [54] proposed an attention module based on channel and spatial dimensions (CBAM). The most important aspect of this module is that it considers both the channel and spatial information, and can adapt to any network structure. For remote sensing images with small category differences and complex spatial structures, CBAM can suppress unimportant, redundant background information, and secondary objects, and extract discriminative features that are conducive to the final classification. Therefore, we add the CBAM module to the CSDS model structure. To make full use of the fusion of the channel–spatial attention module and the residual unit without adding a large number of parameters, the attention mechanism is only added to the last residual module of each flow.

1) *The attention unit for the channel dimension in CBAM is shown in Fig. 8(a).* Under the joint action of the global AvgPool and MaxPool, input feature F with dimension $H \times W \times C$ will output two weights F^c_{avg} and F^c_{max} with dimensions $1 \times 1 \times C$. Then, the channel attention feature $M_c \in R^{C\times 1\times 1}$ is generated by a shared multilayer perceptron (MLP). To further reduce the number of parameters, the activation size of the hidden layer in the MLP is set to $R^{\frac{C}{r}\times 1\times 1}$, and the reduction rate is r. The channel feature M_c is multiplied with the original feature F to obtain the final feature F'. Then, M_c can be expressed as

$$\begin{aligned} M_c(F) &= \sigma\left(\text{MLP}\left(\text{AvgPool}(F)\right)\right. \\ &\quad \left.+\text{MLP}\left(\text{MaxPool}(F)\right)\right) \\ &= \sigma\left(W_1\left(W_0\left(F^c_{\text{avg}}\right)\right) + W_1\left(W_0\left(F^c_{\text{max}}\right)\right)\right) \end{aligned} \tag{7}$$

where σ represents the sigmoid operation, $W_0 \in R^{\frac{C}{r}\times C}$, $W_1 \in R^{C\times\frac{C}{r}}$, and the weights W_0 and W_1 in the MLP are shared.

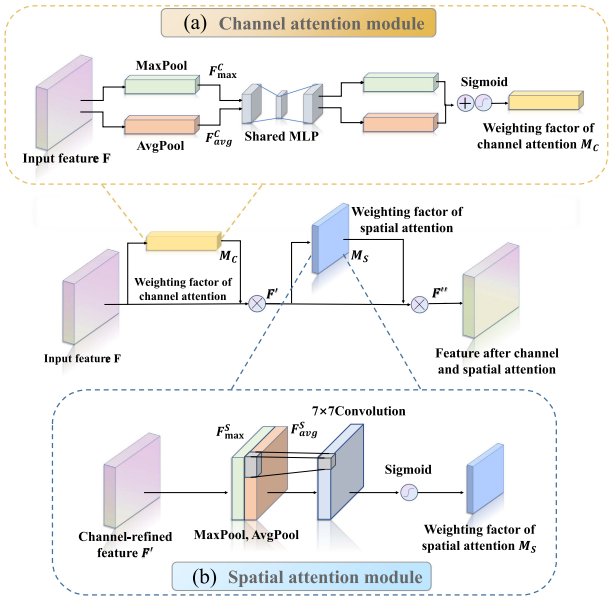


Fig. 8. Architecture of the channel–spatial attention module (CBAM).

2) *The attention unit for the spatial dimension in CBAM is shown in Fig. 8(b).* The channel feature map F' outputs F^s_{avg} and F^s_{max} through the AvgPool and MaxPool operations. Then, a concat connection is implemented, and dimensionality reduction is performed through a convolution operation. The weights after dimensionality reduction generate the spatial attention features through the sigmoid function. Finally, the feature of the module and the input feature F' are multiplied to obtain the final feature M_s, which can be expressed as

$$\begin{aligned} M_s(F) &= \sigma\left(f^{7\times 7}\left(\left[\text{AvgPool}\left[F\right];\text{MaxPool}\left[F\right]\right]\right)\right) \\ &= \sigma\left(f^{7\times 7}\left(\left[F^s_{\text{avg}};F^s_{\text{max}}\right]\right)\right) \end{aligned} \tag{8}$$

where $f^{7\times 7}$ represents a standard convolution operation with a convolution kernel size of 7×7. Woo *et al.* [54] showed that the model achieves the best performance by placing the channel attention module before the spatial attention module.

The channel–spatial attention module obtains relevant attention mapping through the two independent channel and spatial dimensions, and assigns adaptive weights to different features. The number of parameters is very small, which saves overhead and reduces the risk of overfitting.

C. Cross-Entropy Loss Function Based on Label Smoothing

When the features of key objects in different images are identical or their similarity is high, the CNN model will experience overfitting and become more prone to classification errors. Label smoothing can be used to constrain and modify the cross-entropy loss function, thus helping improve the accuracy of classification for similar categories and the generalizability of the model.

When a CNN model is used for image classification tasks, the softmax function is generally implemented in the classifier

layer to calculate the probability of each category in the dataset according to the "hard" label. In multiclassification tasks, the category vector is expressed in a one-hot form: for a dataset of n categories, the probability can be regarded as an array of length n, where the probability of the correct category is 1 and that of the incorrect categories is 0. In this way, there is an absolute nonone or zero relationship between the real category and other categories, and the probability gap is maximized. However, according to the bounded gradient, the CNN model can easily experience overfitting, limiting the generalizability of the category. The original cross-entropy loss function is corrected by label smoothing so that "hard" labels are transformed into "soft" labels. Each category has a certain probability; the probability of the positive category is the largest, and the probabilities of other categories decrease in order according to the correlation with the positive category. The original softmax function formula is as follows:

$$p_k = \frac{e^{f^T w_k}}{\sum_{l=1}^{L} e^{f^T w_l}} \tag{9}$$

where p_k represents the probability of each category in k_{th}, w_k represents the corresponding weight and deviation, and f represents the feature vector. According to the "hard" category label, the expected value of the minimization of the cross-entropy between the actual category y_k and the corresponding probability p_k is calculated by backpropagation

$$\text{Loss} = -\sum_{k=1}^{K} y_k \log p_k \tag{10}$$

y_k is "1" for the correctly classified class and "0" otherwise. y_k is expressed as

$$y_k = \begin{cases} 1, \text{if}(k = x) \\ 0, \text{if}(k \neq x). \end{cases} \tag{11}$$

It can be seen that the original cross-entropy loss function does not consider the loss of the wrong label but allows the model to learn in the direction of the largest difference. Remote sensing datasets tend to be small and contain many similar categories; thus, the original loss function is insufficient for addressing all sample characteristics. For example, in remote sensing image classification tasks, the UC Merced (UCM) dataset contains 2100 labeled samples, only half of which may be used for training, and is characterized by an uneven sample distribution. This makes the model prone to prediction bias and overfitting, which causes difficulties in correctly distinguishing similar categories.

As a regularization strategy, label smoothing reduces the output difference between positive and negative samples by using the hyperparameter α, and soft-one hot to add noise and constrain the loss function. The cross-entropy loss relationship between the corrected real label y_k^{LS} and the corresponding probability p_k is

$$y_k^{\text{LS}} = y_k (1 - \alpha) + \alpha u(K) = \begin{cases} 1 - \alpha + \alpha/K, \text{if}(k = x) \\ \alpha/K, \text{if}(k \neq x) \end{cases} \tag{12}$$

Fig. 9. UCM dataset examples.

where K represents the total number of categories, k is the index of a particular category, and $u(K)$ obeys a uniform distribution with respect to the K classes. Now, the new loss function can be expressed as

$$\begin{aligned} \text{Loss}' &= -\sum_{k=1}^{K} y_k^{\text{LS}} \log p_k = \begin{cases} (1 - \alpha) * \text{Loss}, \text{ if } (k = x) \\ \alpha * \text{Loss}, \text{ if } (k \neq x) \end{cases} \\ &= -(1 - \alpha + \alpha/K) \log y_x - \frac{\alpha}{K} \sum_{k \neq x} \log p_k. \end{aligned} \tag{13}$$

Label smoothing allows the cross-entropy loss function to not only evaluate the loss of the correct category but also reduce the difference with the wrong category, which helps improve the generalizability of the model to remote sensing datasets with small differences between classes.

IV. Experiments and Analysis

A. Dataset Description

We used three public remote sensing scene image datasets, the UCM dataset [60], the AID [10], and the NWPU-RESISC45 Dataset (NWPU) [9], to verify the classification performance of the proposed CSDS network. Table I shows the basic information of the datasets.

1) UC Merced Land-Use Dataset (UCM): The UCM dataset [60] consists of 2100 images and 21 categories, each category containing 100 256×256 pixel images. It is the first publicly available remote sensing image dataset, and all samples were captured with a civilian satellite platform. Therefore, the data are integrated as a common dataset in aerial scene classification tasks. Examples from the dataset are shown in Fig. 9.

2) AID: The AID [10] dataset is larger and has richer interclass diversity than the UCM dataset, consisting of 30 categories and 10 000 images (220–420 per category) measuring 600×600 pixels each. The dataset was initially collected from different regions of the world at different spatial resolutions and times, resulting in a more difficult classification task. Some example images are shown in Fig. 10.

3) NWPU-RESISC45 Dataset (NWPU): The NWPU dataset is a new large-scale image dataset released by Northwestern Polytechnical University [9]. It consists of a total of 45 categories and 31 500 images, 700 images per class at a resolution of 256×256 pixels. This dataset was collected from more than

TABLE I
BASIC INFORMATION OF THE THREE DATASETS

dataset	#classes	#images per class	Image size	Spatial Resolution (in meters)	Color space	Training ratio settings
UC Merced	21	100	256*256	0.3	RGB	50%, 80%
AID	30	220-400	600*600	0.5-8		20%, 50%
NWPU	45	700	256*256	0.2-30		10%, 20%

Fig. 10. AID dataset examples.

Fig. 11. NWPU dataset examples.

100 regions around the world, has a variety of spatial resolutions and similar categories, and represents a more challenging classification task than the UCM and AID datasets. Some example images are shown in Fig. 11.

B. Experimental Details

Training-to-test set ratio: To fairly compare the proposed method with other experimental methods, we stipulate that the training-to-test set ratio should be the same as that used in most previous experiments [9], [10], [20], [33], [61]–[63]. Table I shows the training-to-test set ratio for each dataset. We consider two different training-to-test set ratios to fully evaluate the proposed CSDS network.

Model initialization: In the CSDS network, the parameters of the Xception model pretrained on ImageNet are used as the initialization parameters of the depth-separable convolutional layer, and the other network layer parameters are initialized randomly. All offset parameters are initially set to 0.001.

Training process: All images are resized to 299×299 pixels as the input, and the batch size is set to 16. The Adam optimizer is implemented for parameter optimization, and the initial learning rate is set to 0.001. If the training loss does not decrease for five consecutive epochs, the learning rate is divided by 10, and training continues until the network converges.

Other experimental details: In this work, all our algorithms are implemented by the TensorFlow framework. All the implementations are evaluated on a workstation with a Xeon(R) Gold 5222 CPU and 64 GB memory, and a GeForce RTX2080Ti GPU was used for hardware acceleration.

C. Accuracy Evaluation Indices

The OA, average accuracy (AA), Kappa coefficient (Kappa), F1 score (F1), and confusion matrix (CM) are used in the experiment to describe the performance of the proposed CSDS network. The OA represents the performance of the model in predicting the image category and is calculated as the number of correctly classified images in the test set divided by the total number of test images, with a range from 0 to 1. The AA is the accuracy averaged across all scenario classes in the test set. The F1 is the harmonic average of precision and recall, with a maximum value of 1 and a minimum value of 0. The CM represents the actual classification result for each category. Each item x_{ij} in the matrix is the ratio of the ith predicted class to the jth true class. The CM can be directly visualized through information tables to quickly analyze misclassifications between different categories.

To obtain true and reliable experimental results, we randomly divide the three datasets according to the training-to-test set ratio and repeat the experiment ten times. The average value and standard deviation are calculated as the final experimental results for the proposed CSDS network.

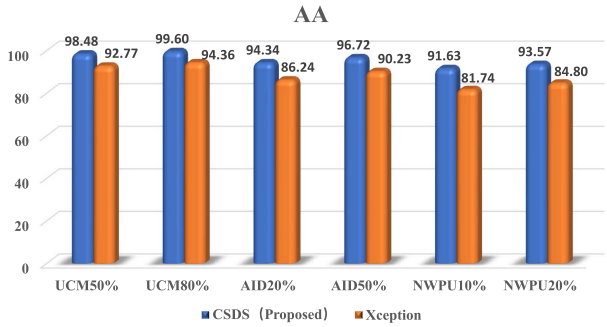
D. Experimental Results

Table II shows the classification performance of the CSDS and Xception networks in terms of the OA and Kappa for the three datasets at two different training-to-test set ratios.

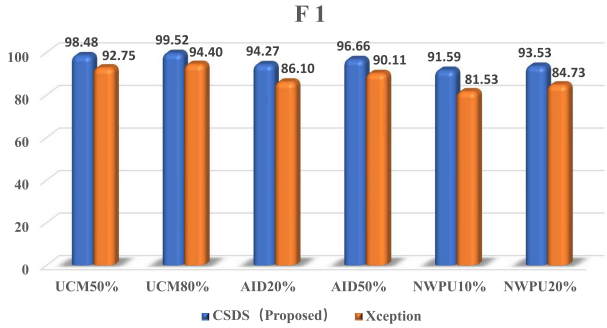
Table II shows that the performance of CSDS is significantly better than that of Xception. The best performance is obtained

TABLE II
COMPARISON OF THE OA AND KAPPA VALUES OF THE CSDS AND XCEPTION NETWORKS

Dataset	CSDS		Xception	
	OA(%)	Kappa(%)	OA(%)	Kappa(%)
UCM50%	98.48	98.28	92.76	92.57
UCM80%	99.52	99.39	94.40	94.21
AID20%	94.29	93.39	86.12	85.96
AID50%	96.7	96.21	90.14	89.76
NWPU10%	91.64	91.43	81.64	80.93
NWPU20%	93.59	93.27	84.79	84.12



(a)



(b)

Fig. 12. Performance comparison of CSDS and Xception. (a) AA. (b) F1.

TABLE III
EXPERIMENTAL RESULTS ON THE UCM DATASET (—: NOT REPORTED)

Method	Training-to-test set ratio	
	50%	80%
PLSA(SIFT) [10]	67.55±1.11	71.38±1.77
BoVW(SIFT) [10]	73.48±1.39	75.52±2.13
AlexNet [10]	93.98±0.67	95.02±0.81
VGGNet-16 [10]	94.14±0.69	95.21±1.20
GoogLeNet [10]	92.70±0.60	94.31±0.89
CaffeNet [10]	93.98±0.67	95.02±0.81
TEX-Net with VGG [41]	94.22±0.50	95.31±0.69
D-CNN with AlexNet [13]	—	96.67±0.10
Fine-tuned GoogLeNet [37]	—	97.1
Two-Stream Fusion [26]	96.97±0.75	98.02±1.03
SPP with AlexNet [19]	94.77±0.46	96.67±0.94
Gated attention [64]	94.64±0.43	96.12±0.42
CCP-net [65]	—	97.52±0.97
Fusion by addition [20]	—	97.42±1.79
DSFATN [61]	—	98.25
Deep CNN Transfer [22]	—	98.49
MIDC-Net [66]	95.41±0.40	97.40±0.48
DFAGCN [44]	—	98.48±0.42
Inception-v3-CapsNet [34]	97.59±0.16	99.05±0.24
Backbone (Xception) [47]	92.76±0.31	94.40±0.15
CSDS (ours)	98.48±0.21	99.52±0.13

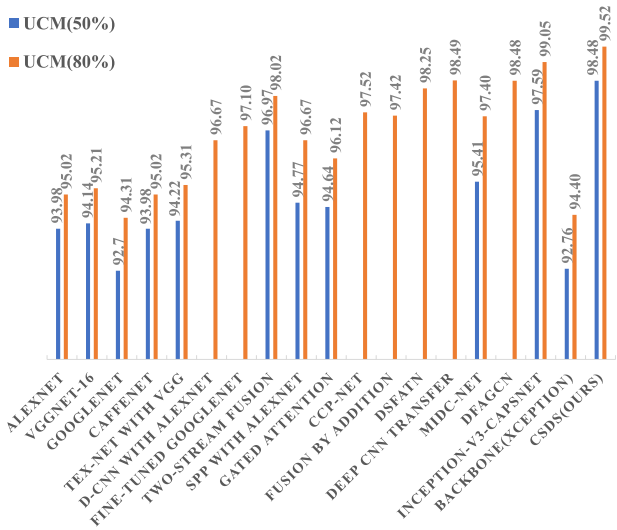


Fig. 13. Comparison of the accuracy of different methods in the UCM dataset (for those with an accuracy greater than 90%).

with the UCM dataset, with OA and Kappa values of 99.52% and 99.39%, respectively. With the AID and NWPU datasets, which are more difficult to classify, the performance of CSDS also far exceeds that of Xception. The OA values of the CSDS network are 5.72%, 8.17%, and 10.0% higher, respectively, than those of Xception with the smaller training ratio. In addition, when the training-to-test ratio of the three datasets is small, the advantages of CSDS are more prominent, illustrating the effectiveness and superiority of the proposed method.

The AA and F1 results are shown in Fig. 12. In Fig. 12(a), the AA values of the CSDS are all higher than those of Xception for the three datasets and the two different training-to-test set ratios. On the larger AID and NWPU dataset, the improved performance of CSDS is more obvious. With the smaller training ratio, the AA of CSDS is 8.10% and 9.89% higher than that of Xception, respectively. The F1 performance of the CSDS method, shown in Fig. 12(b), is also good, especially on AID20%, AID50%, NWPU10%, and NWPU20%, with scores 8.17%, 6.55%, 10.06%, and 8.8% higher than those of Xception, respectively.

1) Experiments on the UCM Dataset: The classification performance of the proposed CSDS and other state-of-the-art methods on the UCM dataset is shown in Table III. To facilitate visualization of this comparison, we provide a histogram for methods with a classification accuracy greater than 90% in Fig. 13.

1) The CSDS network has the highest OAs under the two training-to-test ratios, 98.48% and 99.52%.
2) The improvements of the CSDS model are prominent for the large (80%) than for the small training-to-test set ratio (50%).
3) Methods based on deep features are better than those based on handcrafted features [10] in aerial image classification.

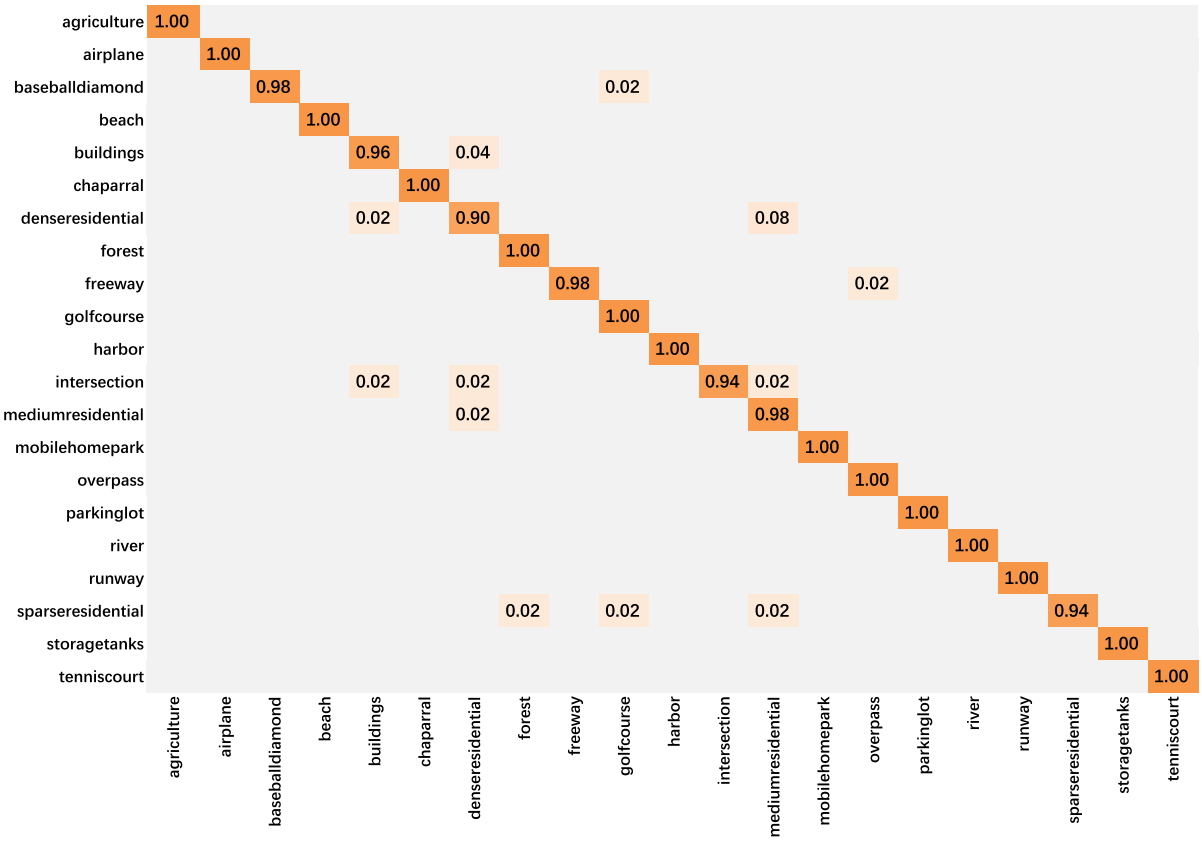


Fig. 14. CM of the UCM dataset for a 50% training-to-test set ratio (only interclass misclassifications greater than 0.01 are displayed).

4) Deeper and wider networks, such as the fine-tuned GoogLeNet [37], demonstrate powerful feature learning capabilities and achieved good classification results.
5) Xception, which serves as the backbone of CSDS, has good performance with this dataset but is still outperformed by our method and some other multifeature fusion methods.

The possible reasons for these findings are as follows.

1) Although existing CNNs tend to retain deeper features, CSDS retains features from multiple levels, especially shallow features.
2) By including channel–spatial attention, CSDS extracts more important weight information from the channel and spatial domains.
3) Transfer learning provides more parameter weights for the model, thereby improving classification performance.
4) Handcrafted feature-based models are unable to extract a large amount of spatial information from remote sensing images, and the feature learning ability is poor, which affects the classification accuracy. Deep features can enhance the semantic representation of aerial images.
5) A deeper and wider network can extract more global features, thereby improving the classification results.
6) Although the backbone Xception model has a deeper architecture than the CSDS model, it easily extracts features through a pure linear stacked convolution module, which affects the classification accuracy.

Fig. 14 shows the classification CM of the UCM dataset at a training-to-test set ratio of 50%. Twenty categories exceeded 94%, and half reached 100% accuracy; however, the "dense residential" and "medium residential" categories yielded large errors, possibly because they consist of highly similar spatial structures and objects.

2) Experiments on the AID Dataset: The classification performance of the proposed CSDS and other state-of-the-art methods on the AID dataset is shown in Table IV. The histogram in Fig. 15 shows methods with an accuracy greater than 80%.

1) The CSDS network performs best under both training-to-test set ratios, with OA values of 94.29% for the 20% ratio and 96.70% for the 50% ratio.
2) Comparing Tables III and IV shows that the same method achieves better classification accuracy with UCM than with AID.
3) Our proposed CSDS and other state-of-the-art methods [9], [34], [41], [66] outperform the baseline method [10] at both training-to-test set ratios.
4) As the number of categories increases, the classification accuracy of Xception drops substantially.

The possible reasons for these findings are as follows.

1) The UCM dataset is small, which can easily lead to model overfitting, comparatively, the AID dataset has more categories and training samples. Furthermore, a dataset with small differences between classes is more suitable for

TABLE IV
EXPERIMENTAL RESULTS ON THE AID DATASET (—: NOT REPORTED)

Method	Training-to-test set ratio	
	20%	50%
PLSA(SIFT) [10]	56.24±0.58	63.07±1.77
BoVW(SIFT) [10]	62.49±0.53	68.37±0.40
AlexNet [10]	86.86±0.47	89.53±0.31
VGGNet-16 [10]	86.59±0.29	89.64±0.36
GoogLeNet [10]	83.44±0.40	86.39±0.55
CaffeNet [10]	86.86±0.47	89.53±0.31
TEX-Net with VGG [41]	87.32±0.37	90.00±0.33
D-CNN with AlexNet [13]	85.62±0.10	94.47±0.12
SPP with AlexNet [19]	87.44±0.45	91.45±0.38
Two-Stream Fusion [26]	92.32±0.41	94.58±0.25
Fusion by addition [20]	—	91.87±0.36
MIDC-Net [66]	88.51±0.41	92.95±0.17
Gated attention [64]	87.63±0.44	92.01±0.21
DFAGCN [44]	—	94.88±0.22
TFADNN [67]	93.21±0.32	95.04±0.16
Inception-v3-CapsNet [34]	93.79±0.13	96.32±0.12
Backbone (Xception) [47]	86.12±0.28	90.14±0.52
CSDS (ours)	94.29±0.35	96.70±0.14

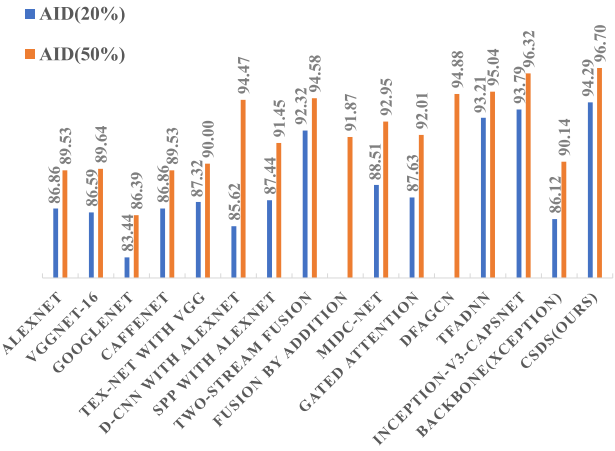


Fig. 15. Comparison of the accuracy of different methods in the AID dataset (for methods with an accuracy greater than 80%).

verifying the proposed method. Therefore, the improvement in the latest state-of-the-art methods is more obvious.

2) Due to the complexity of the AID dataset, simple networks cannot achieve a suitable performance, with classification accuracies less than 90%. However, methods based on feature fusion, such as two-stream fusion, can provide more features needed for network training to achieve better results.
3) The CSDS network has a pyramid residual connection that can retain more features, and the channel–spatial attention module can improve the network's ability to learn local semantic features. Therefore, the proposed method has the best effect on the more complex AID dataset.

The CM of the AID dataset for a training ratio of 20% for the CSDS network is shown in Fig. 16 . The CSDS achieved a classification accuracy of more than 90% for 80% of the 30 categories, while beaches, parking lots, and sparse houses were classified at 100% accuracy. Some categories with similar spatial distributions of key objects, such as "sparse residential," "medium residential," and "dense residential" categories, were also classified with very high accuracies (100%, 96%, and 94%, respectively). This shows that our proposed method can accurately extract and learn key features for similar categories. However, the classification accuracy for the "school" and "square" categories are relatively low, 68% and 80%, respectively. The "school" and "commercial" categories are easily confused due to their similar distributions of characteristics, and "square" images are often misclassified as "parks" and "churches" images due to the presence of similar objects, such as vegetation and houses. Fortunately, the proposed method achieves accuracies much higher than 49% and 67%, respectively, reported in [10]. Thus, this method still has a positive effect on the classification of highly similar categories.

TABLE V
EXPERIMENTAL RESULTS ON THE NWPU DATASET (—: NOT REPORTED)

Method	Training radio	
	10%	20%
BoVW(SIFT) [9]	41.72±0.21	44.97±0.28
AlexNet [9]	76.69±0.21	79.85±0.13
VGGNet-16 [9]	76.47±0.18	79.79±0.15
GoogLeNet [9]	76.19±0.38	78.48±0.26
BoCF [45]	82.65±0.31	84.32±0.17
Fine-tuned VGG-16 [9]	87.15±0.45	90.36±0.18
Fine-tuned AlexNet [9]	81.22±0.19	85.16±0.18
Fine-tuned GoogLeNet [9]	82.57±0.12	86.02±0.18
Triple networks [68]	—	92.33±0.20
D-CNN with AlexNet [13]	85.56±0.20	87.24±0.12
SPP with AlexNet [19]	82.13±0.30	84.64±0.23
Two-Stream Fusion [26]	80.22±0.22	83.16±0.18
MIDC-Net [66]	86.12±0.29	87.99±0.18
Gated attention [64]	84.94±0.22	86.62±0.22
DFAGCN [44]	—	89.29±0.28
TFADNN [67]	87.78±0.11	90.86±0.24
Inception-v3-CapsNet [34]	89.03±0.21	92.60±0.11
Backbone (Xception) [47]	81.64±0.32	84.79±0.26
CSDS (ours)	91.64±0.16	93.59±0.21

3) Experiments on the NWPU Dataset: NWPU is the most challenging dataset to classify. From Table V and Fig. 17, the following can be observed.

1) The CSDS network achieved the best accuracy rates, 92.70% and 94.58%, under training-to-test set ratios of 10% and 20%, respectively.
2) Inception-v3-CapsNet performed better than the baseline method but was outperformed by our CSDS.
3) Our proposed CSDS outperforms all baseline methods [9] and the three latest methods [13], [34], [66] for both training-to-test set ratios. Inception-v3-CapsNet [34] also outperformed other methods.
4) The performance of our proposed CSDS on the NWPU dataset is significantly higher than that of the backbone Xception network.

The possible reasons for these findings are as follows.

1) For samples past a certain size, deep learning methods are more advantageous than middle- and low-level feature-based methods, and achieve better overall accuracy.
2) Although Inception-v3-CapsNet [34] mainly utilizes multiscale information from within images, the extraction and utilization of key features are limited. Therefore, compared with the proposed CSDS, it demonstrates worse performance.

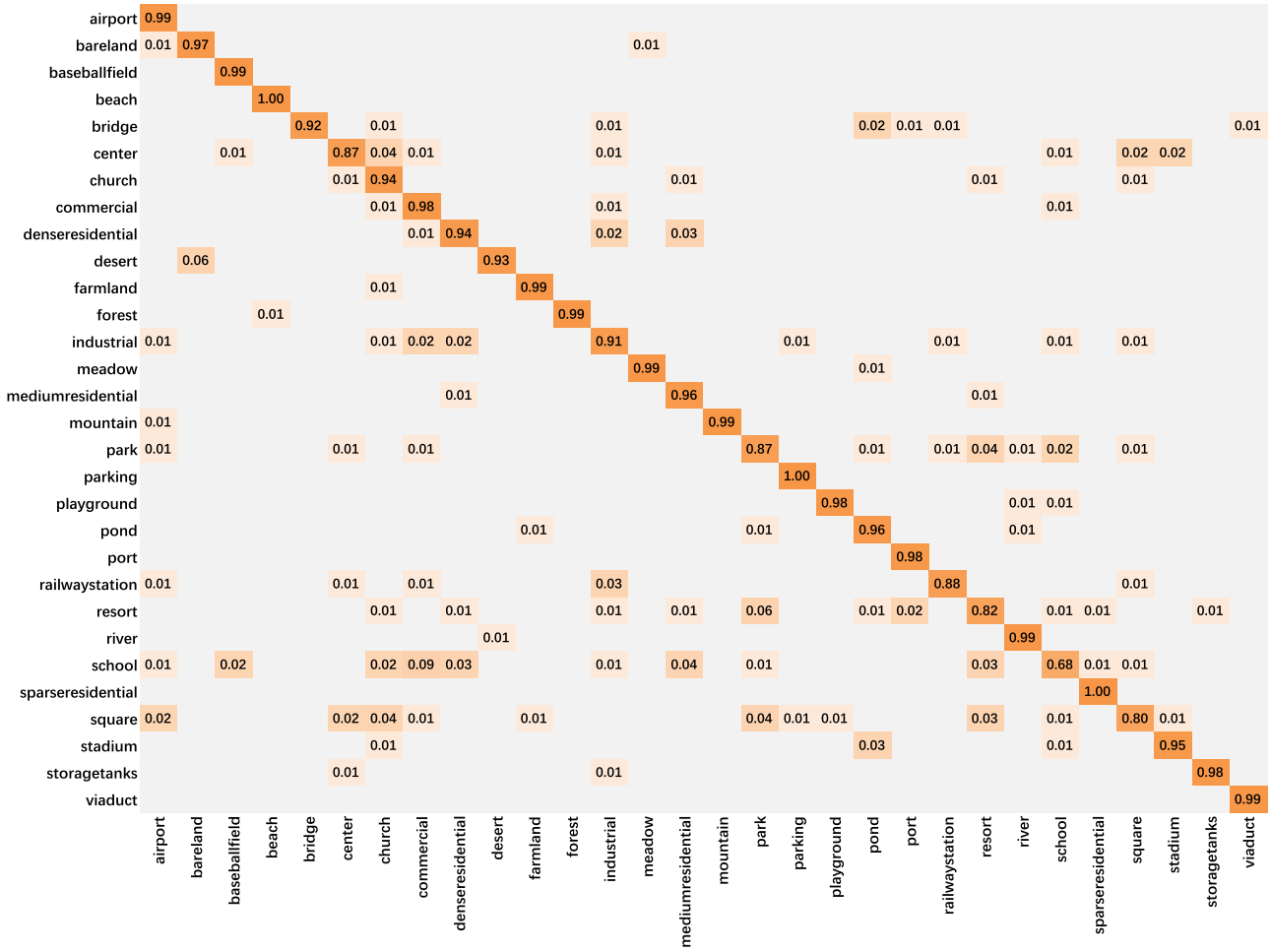


Fig. 16. CM of the AID dataset for a training-to-test set ratio of 20% (only interclass misclassifications greater than 0.01 are shown).

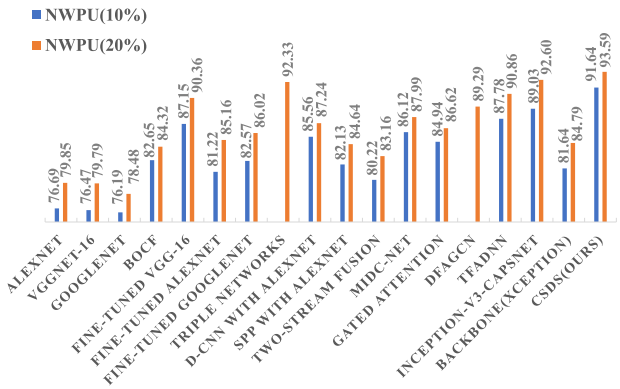


Fig. 17. Comparison of the accuracy of different methods in the NWPU dataset (for methods with an accuracy greater than 75%).

Fig. 18 shows the CM when using CSDS at a training rate of 10%. Among them, 33 of the 45 categories have classification accuracies higher than 90%. "Palace" and "church" are easy to confuse because the two have extremely similar structural features. However, compared with 56% and 47% [9], 67% and 72% still made substantial progress.

TABLE VI
NUMBER OF PARAMETERS AND MODEL SIZES OF THE PROPOSED CSDS AND THREE BASELINE MODELS

Method	Parameters (in millions)	Model size (in Mbytes)
AlexNet [9]	61	232.7
VGG-16 [9]	138.3	527.6
Inception-v3 [9]	24.7	94.2
Xception [47]	22.9	87.2
CSDS (ours)	10.9	41.5

Parameters and sizes: Table VI shows the following.

1) The size and number of parameters of our CSDS are far smaller than those of the other three CNN models that are widely used in remote sensing image classification.
2) Although our model is much smaller than the other three models, it still outperforms them in classifying remote sensing images (as shown in Tables III–V).

The possible reasons for these findings are as follows.

1) Using pyramid residual connections not only reduces the model size but also allows features and parameters to be reused, enhances the feature learning ability of the model and establishes connections with shallower features. Furthermore, the use of DS-Conv does not add an excessive

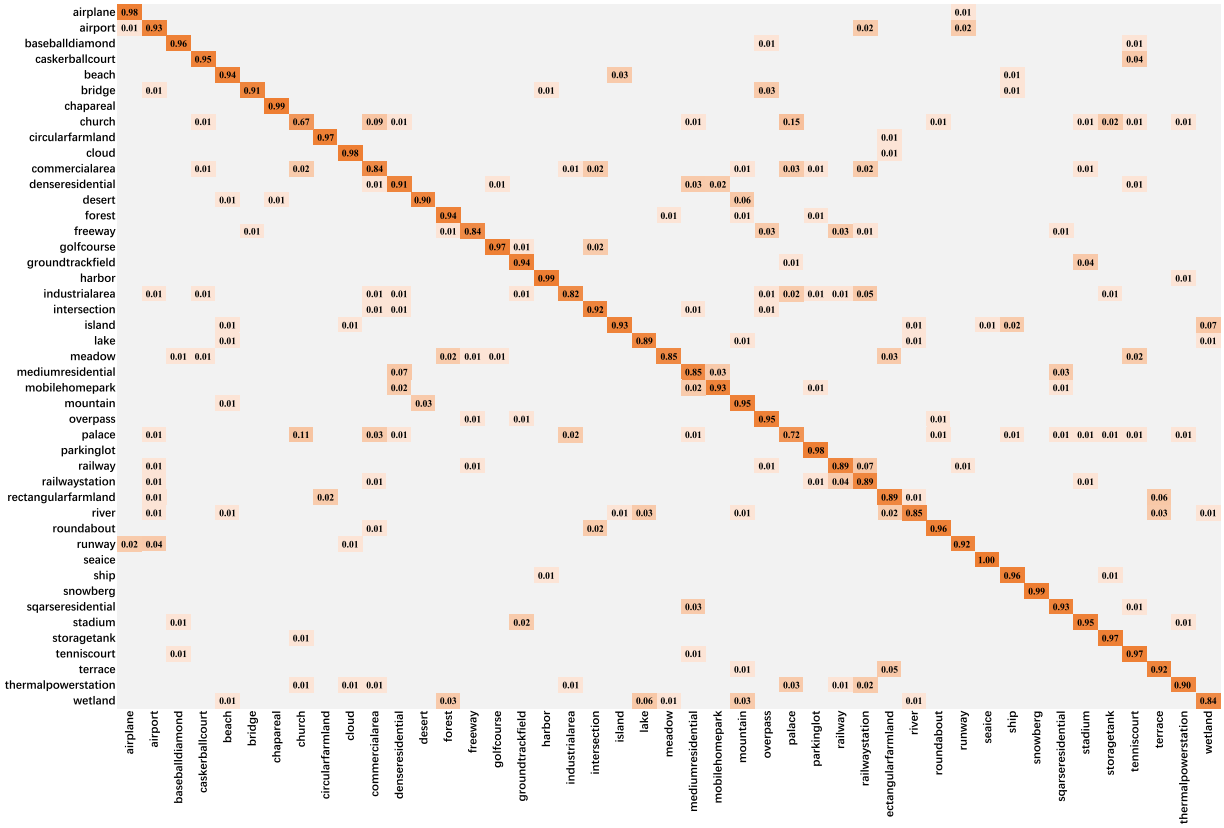


Fig. 18. CM of the NWPU dataset for a training-to-test set ratio of 10% (only for interclass misclassifications greater than 0.01).

number of parameters as the model depth increases; in other words, incorporation of pyramid residual connections ensures the model's ability to extract deep features while minimizing the increase in the number of parameters and calculations.

2) Channel–spatial attention enhances the ability of CSDS to learn local key semantic features in complex remote sensing images. Studies have shown that when local feature representation is enhanced, even if many of the convolutional layers are reduced, the network can still produce good results [69].
3) The attention module does not substantially increase the computational burden because it can be regarded as a variant of the fully connected layer. Additionally, the depth separable convolution reduces the number of calculations from the perspective of decoupling. Increasing the depth of the network does not increase the number of parameters to an adverse degree. This allows the network to obtain better feature extraction capabilities while mitigating the effect of increasing the number of parameters.
4) Compared with the original Xception model, the CSDS has fewer layers and includes multiple attention modules. While reducing the number of parameters by half, we increase the model's ability to extract local features.

Prediction time: Table VII shows the prediction time for a single image from each of the three datasets. The prediction time of CSDS is significantly shorter than that of both the three CNN [9] models widely used in aerial scene classification and Xception. These results demonstrate that the reduction of the model parameters was beneficial in improving the prediction time. The CSDS model guarantees a classification accuracy with much fewer parameters than the other models shown in Table V. The difference in the number of parameters between Xception and Inception-v3 is not large, but the prediction speed of the former is greater by an average of 0.017 s, which shows that DS-Conv is faster than standard convolution.

TABLE VII
COMPUTATION TIMES (IN SECONDS) OF DIFFERENT METHODS WITH THE THREE DATASETS

Method	UCM(50%)	AID(20%)	NWPU(10%)
AlexNet [9]	0.772	0.865	1.7
VGG-16 [9]	1.315	1.790	2.251
Inception-v3 [9]	0.051	0.068	0.074
Xception [47]	0.036	0.047	0.053
CSDS (ours)	0.011	0.029	0.044

V. DISCUSSION

A. CSDS Ablation Experiment

We conduct ablation studies with the AID and NWPU datasets to further demonstrate the performance of the CSDS network. Table VIII presents the classification performance of the CSDS network and its different parts.

Even without the attention module, the CSDS network outperforms most methods (compare Table VIII with

TABLE VIII
CSDS ABLATION STUDY RESULTS

Dataset	Without channel-spatial attention	Without label smoothing	Without pyramid residual	CSDS
AID (20%)	91.80±0.11	93.23±0.32	92.16±0.26	94.29±0.35
AID (50%)	94.85±0.32	96.24±0.21	95.82±0.17	96.70±0.14
NWPU (10%)	89.10±0.41	90.14±0.26	89.23±0.21	91.64±0.16
NWPU (20%)	90.37±0.20	92.79±0.35	91.52±0.18	93.59±0.21
Average	91.53	93.10	92.18	94.06

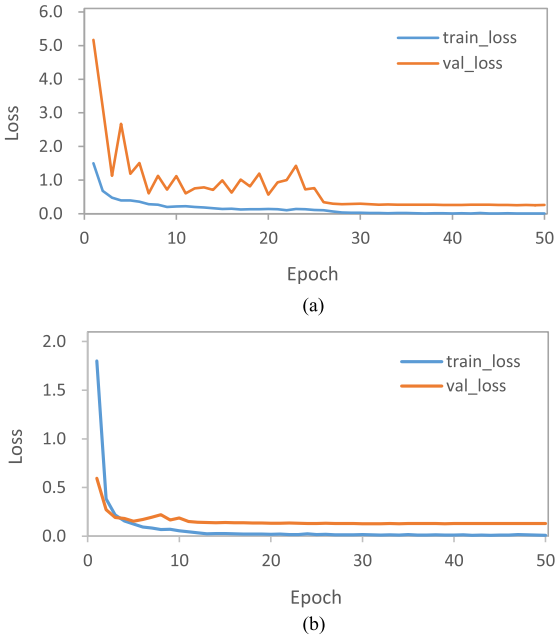


Fig. 19. Loss function images with the AID dataset (training-to-test set ratio 20%). (a) CSDS without pyramid residual connection unit. (b) CSDS.

Tables III–V). Table VIII shows that the attention module increases the performance of the CSDS by an average of 1.57%; this may be because channel–spatial attention enhances the representation of the main object features in the image. The loss function with label smoothing increases the classification accuracy by 0.86% on average; because label smoothing corrects the loss function, the generalization ability of the CSDS model is improved. The pyramid residual connection unit increases the accuracy by 1.88% on average, confirming that this unit improves the generalizability of the CSDS model while also helping to accelerate the convergence of the network model and improve the fitting effect [56], as shown in Fig. 19.

The figure shows the loss function images of the CSDS trained 50 times on the AID dataset (training-to-test set ratio 20%) with and without the pyramid residual connection unit. The oscillation in the loss function is reduced, and the convergence speed is very high.

B. Attention Maps on CSDS

Fig. 20 shows the attention map generated with the Grad-CAM algorithm, with the key feature area highlighted. Four categories are the most difficult to classify: 1) church, 2) airplane, 3) palace, and 4) square. The CSDS network with the

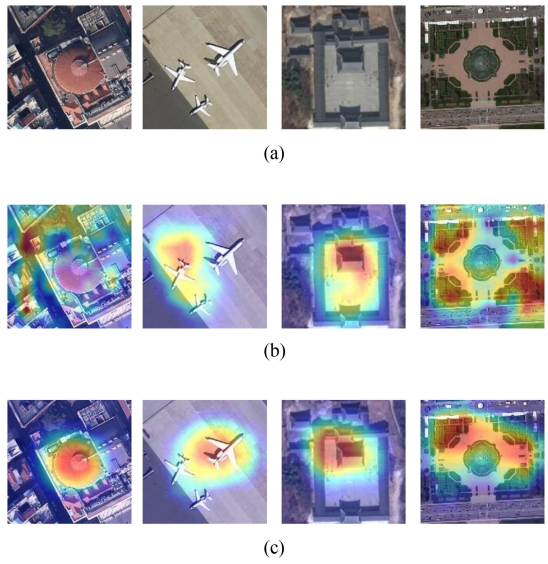


Fig. 20. Original images and attention maps. (a) Original scene images. (b) Attention maps without the channel-spatial attention. (c) Attention maps with the channel-spatial attention.

channel–spatial attention module extracts the key features of these scenes and accurately captures the key objects without being affected by other objects and background information. Even for the categories church, palace, and square, which are prone to classification errors, the CSDS network still accurately extracts the features of key areas. Therefore, the attention map fully illustrates the effectiveness of channel–spatial attention and the powerful image feature learning ability of CSDS. Channel–spatial attention can be used to effectively locate and extract the main objects and key local features in remote sensing images and increase the weight of decision classification information, thereby improving the final accuracy of the CSDS.

VI. CONCLUSION

Remote sensing images have complex spatial distributions and multiscale semantic relations, but these key semantic features cannot be extracted by traditional CNNs constructed with linear superposition. Therefore, this article proposes an end-to-end framework named CSDS to solve these problems. First, the CSDS model is a deep network that includes a pyramid residual unit and depth separable convolution, which reduce the number of calculations and parameters, prevent overfitting, and effectively extract multilayer spatial information. Second, the channel–spatial attention module in the network obtains highly effective local feature representation, increasing the weights of main features while suppressing the weights of secondary features. Finally, the cross-entropy loss function based on label smoothing considers the relationship between different classes to reduce the impact of classes with similar features on the scene image classification results. The proposed CSDS achieves better performance than other methods on three public remote sensing image classification datasets. To further verify the feature extraction and classification performance of CSDS, we generated an attention map based on the Grad-CAM algorithm and showed

the results visually. Although good results were achieved with the model, for some categories with very similar characteristics, the effect still needs to be improved. Future work will consider the integration of multiple features, the addition of self-attention mechanisms, and enhancement of the convolution modules to further improve the network learning capabilities.

REFERENCES

[1] Z. Zou and Z. Shi, "Random access memories: A new paradigm for target detection in high resolution aerial remote sensing images," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1100–1111, Mar. 2018.

[2] L. Zhang, Y. Han, Y. Yang, M. Song, S. Yan, and Q. Tian, "Discovering discriminative graphlets for aerial image categories recognition," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5071–5084, Dec. 2013.

[3] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu, "Learning RoI transformer for oriented object detection in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2849–2858.

[4] J. R. Taylor and S. T. Lovell, "Mapping public and private spaces of urban agriculture in Chicago through the analysis of high-resolution aerial images in Google Earth," *Landscape Urban Plan.*, vol. 108, no. 1, pp. 57–70, 2012.

[5] W. Chen, X. Li, H. He, and L. Wang, "Assessing different feature sets' effects on land cover classification in complex surface-mined landscapes by Ziyuan-3 satellite imagery," *Remote Sens.*, vol. 10, no. 1, pp. 23–43, 2018.

[6] 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.

[7] X. Zhang and S. Du, "A linear Dirichlet mixture model for decomposing scenes: Application to analyzing urban functional zonings," *Remote Sens. Environ.*, vol. 169, pp. 37–49, 2015.

[8] Q. Feng, J. Liu, and J. Gong, "UAV remote sensing for urban vegetation mapping using random forest and texture analysis," *Remote Sens.*, vol. 7, no. 1, pp. 1074–1094, 2015.

[9] 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.

[10] 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.

[11] Q. Zhu, Y. Zhong, L. Zhang, and D. Li, "Adaptive deep sparse semantic modeling framework for high spatial resolution image scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 6180–6195, Oct. 2018.

[12] H. Song, Z. Liu, H. Du, G. Sun, O. Le Meur, and T. Ren, "Depth-aware salient object detection and segmentation via multiscale discriminative saliency fusion and bootstrap learning," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4204–4216, Sep. 2017.

[13] 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.

[14] G.-R. Cai, P.-M. Jodoin, S.-Z. Li, Y.-D. Wu, S.-Z. Su, and Z.-K. Huang, "Perspective-SIFT: An efficient tool for low-altitude remote sensing image registration," *Signal Process.*, vol. 93, no. 11, pp. 3088–3110, 2013.

[15] Y. Wang *et al.*, "Learning a discriminative distance metric with label consistency for scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4427–4440, Aug. 2017.

[16] 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, 2020.

[17] Y. Tao, M. Xu, Z. Lu, and Y. Zhong, "DenseNet-based depth-width double reinforced deep learning neural network for high-resolution remote sensing image per-pixel classification," *Remote Sens.*, vol. 10, no. 5, pp. 779–806, 2018.

[18] Q. He, X. Sun, Z. Yan, and K. Fu, "DABNet: Deformable contextual and boundary-weighted network for cloud detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TGRS.2020.3045474.

[19] X. Han, Y. Zhong, L. Cao, and L. Zhang, "Pre-trained AlexNet architecture with pyramid pooling and supervision for high spatial resolution remote sensing image scene classification," *Remote Sens.*, vol. 9, no. 8, pp. 848–870, 2017.

[20] 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.

[21] E. Li, J. Xia, P. Du, C. Lin, and A. Samat, "Integrating multilayer features of convolutional neural networks for remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 10, pp. 5653–5665, Oct. 2017.

[22] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sens.*, vol. 7, no. 11, pp. 14680–14707, 2015.

[23] Y. Hu, G. Wen, M. Luo, D. Dai, J. Ma, and Z. Yu, "Competitive inner-imaging squeeze and excitation for residual network," Jul. 2018, *arXiv:1807.08920*.

[24] T. Gevers and A. W. Smeulders, "Pictoseek: Combining color and shape invariant features for image retrieval," *IEEE Trans. Image Process.*, vol. 9, no. 1, pp. 102–119, Jan. 2000.

[25] M. J. Swain and D. H. Ballard, "Color indexing," *Int. J. Comput. Vis.*, vol. 7, no. 1, pp. 11–32, 1991.

[26] 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.

[27] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[28] A. Avramović and V. Risojević, "Block-based semantic classification of high-resolution multispectral aerial images," *Signal, Image Video Process.*, vol. 10, no. 1, pp. 75–84, 2016.

[29] L. Guo, N. Chehata, C. Mallet, and S. Boukir, "Relevance of airborne lidar and multispectral image data for urban scene classification using random forests," *ISPRS J. Photogrammetry Remote Sens.*, vol. 66, no. 1, pp. 56–66, 2011.

[30] 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.

[31] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Mach. Learn.*, vol. 42, no. 1, pp. 177–196, 2001.

[32] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[33] Y. Yu and F. Liu, "Dense connectivity based two-stream deep feature fusion framework for aerial scene classification," *Remote Sens.*, vol. 10, no. 7, pp. 1158–1183, 2018.

[34] W. Zhang, P. Tang, and L. Zhao, "Remote sensing image scene classification using CNN-CapsNet," *Remote Sens.*, vol. 11, no. 5, pp. 494–516, 2019.

[35] E. Othman, Y. Bazi, N. Alajlan, H. Alhichri, and F. Melgani, "Using convolutional features and a sparse autoencoder for land-use scene classification," *Int. J. Remote Sens.*, vol. 37, no. 10, pp. 2149–2167, 2016.

[36] X. X. Zhu *et al.*, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.

[37] M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva, "Land use classification in remote sensing images by convolutional neural networks," Aug. 2015, *arXiv:1508.00092*.

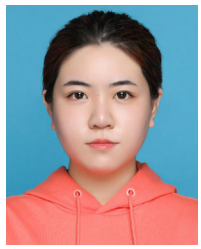
[38] K. Nogueira, O. A. Penatti, and J. A. D. Santos, "Towards better exploiting convolutional neural networks for remote sensing scene classification," *Pattern Recognit.*, vol. 61, pp. 539–556, 2017.

[39] X. Sun, P. Wang, C. Wang, Y. Liu, and K. Fu, "PBNet: Part-based convolutional neural network for complex composite object detection in remote sensing imagery," *ISPRS J. Photogrammetry Remote Sens.*, vol. 173, pp. 50–65, 2021.

[40] K. Xu, H. Huang, and P. Deng, "Remote sensing image scene classification based on global-local dual-branch structure model," *IEEE Geosci. Remote Sens. Lett.*, to be published, doi: 10.1109/LGRS.2021.3075712.

[41] R. M. Anwer, F. S. Khan, J. van de Weijer, M. Molinier, and J. Laaksonen, "Binary patterns encoded convolutional neural networks for texture recognition and remote sensing scene classification," *ISPRS J. Photogrammetry Remote Sens.*, vol. 138, pp. 74–85, 2018.

[42] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[43] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
[44] K. Xu, H. Huang, P. Deng, and Y. Li, "Deep feature aggregation framework driven by graph convolutional network for scene classification in remote sensing," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2021.3071369 .
[45] 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.
[46] L. Sifre and S. Mallat, "Rigid-motion scattering for image classification," Mar. 2014, *arXiv:1403.1687*.
[47] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1251–1258.
[48] 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.
[49] X. Huang, Q. Yang, and H. Qiao, "Lightweight two-stream convolutional neural network for SAR target recognition," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 4, pp. 667–671, Apr. 2021.
[50] 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.
[51] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, vol. 2, pp. 2204–2212
[52] 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.
[53] J. M. Haut, M. E. Paoletti, J. Plaza, A. Plaza, and J. Li, "Visual attention-driven hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 8065–8080, Oct. 2019.
[54] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.
[55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[56] D. Han, J. Kim, and J. Kim, "Deep pyramidal residual networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5927–5935.
[57] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 807–814.
[58] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
[59] A. G. Howard *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," Apr. 2017, *arXiv:1704.04861*.
[60] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2010, pp. 270–279.
[61] 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, pp. 410–434, 2018.
[62] 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, pp. 734–762, 2018.
[63] X. Bian, C. Chen, L. Tian, and Q. Du, "Fusing local and global features for high-resolution scene classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 6, pp. 2889–2901, Jun. 2017.
[64] M. Ilse, J. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2127–2136.
[65] 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, pp. 934–953, 2018.
[66] Q. Bi, K. Qin, Z. Li, H. Zhang, K. Xu, and G.-S. Xia, "A multiple-instance densely-connected ConvNet for aerial scene classification," *IEEE Trans. Image Process.*, vol. 29, pp. 4911–4926, 2020.
[67] K. Xu, H. Huang, P. Deng, and G. Shi, "Two-stream feature aggregation deep neural network for scene classification of remote sensing images," *Inf. Sci.*, vol. 539, pp. 250–268, 2020.
[68] 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.
[69] W. Brendel and M. Bethge, "Approximating CNNs with bag-of-local-features models works surprisingly well on ImageNet," Mar. 2019, *arXiv:1904.00760*.



Xinyu Wang received the B.S. degree in network engineering and the M.S. degree in computer application technology from Inner Mongolia Agricultural University, Hohhot, China, in 2017 and 2020, respectively. She is currently working toward the Ph.D. degree in computer science and technology with the School of Computer Science and Engineering, Tianjin University of Technology, Tianjin, China.

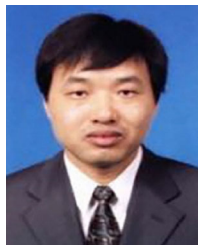
Her research interests include remote sensing image processing, computer vision, and deep learning.

Liming Yuan received the Ph.D. degree in computer science and technology from the Harbin Institute of Technology, Harbin, China, in 2014.

He is currently working as a Lecturer with the School of Computer Science and Engineering, Tianjin University of Technology, Tianjin, China. His research interests include machine learning and image processing.

Haixia Xu received the M.Sc. degree in applied mathematics and the Ph.D. degree in computer science and technology from Northwestern Polytechnical University, Xi'an, China, in 2006 and 2009, respectively.

She is currently an Associate Professor with the School of Computer Science and Engineering, Tianjin University of Technology, Tianjin, China. Her main research interests include image analysis, signal processing, and pattern recognition.



Xianbin Wen received the Ph.D. degree from Northwestern Polytechnical University, Xi'an, China, in 2005.

He is currently a Professor with the School of Computer Science and Engineering, Tianjin University of Technology, Tianjin, China. His research interests include image interpretation, machine learning, and information hiding.