Enhanced Feature Pyramid Network with Deep Semantic Embedding for Remote Sensing Scene Classification

Xin Wang, Shiyi Wang, Chen Ning, and Huiyu Zhou

***Abstract*—Recent progress on remote sensing scene classification is substantial, benefiting mostly from the explosive development of convolutional neural networks (CNNs). However, different from the natural images in which the objects occupy most of the space, objects in remote sensing images are usually small and separated. Therefore, there is still a large room for improvement of the vanilla CNNs that extract global image-level features for remote sensing scene classification, ignoring local object-level features. In this paper, we propose a novel remote sensing scene classification method via enhanced feature pyramid network with deep semantic embedding. Our proposed framework extracts multi-scale multi-level features using an enhanced feature pyramid network (EFPN). Then, to leverage the complementary advantages of the multi-level and multi-scale features, we design a deep semantic embedding (DSE) module to generate discriminative features. Third, a feature fusion module, called two-branch deep feature fusion (TDFF), is introduced to aggregate the features at different levels in an effective way. Our method produces state-of-the-art results on two widely used remote sensing scene classification benchmarks, with better effectiveness and accuracy than the existing algorithms. Beyond that, we conduct an exhaustive analysis on the role of each module in the proposed architecture, and the experimental results further verify the merits of the proposed method.**

***Index Terms*—Remote sensing image, scene classification, convolutional neural network, feature pyramid network, deep semantic embedding.**

I. Introduction

SCENE classification in remote sensing (RS) images, referred to as the task of assigning a specific semantic label to a RS scene, has received wide interests in recent years, since it can be used in a wide range of practical applications, such as urban planning, environment prospecting, natural disaster detection, and land-use classification [1]-[3].

Over the past decades, an extremely rich set of remote sensing scene classification algorithms has been developed. Earlier methods were mainly based on various hand-crafted features and classical classifiers, such as support vector machine (SVM) [4], random forest [5] and boosting [6]. In general, these methods are divided into two categories: methods relying on low-level features and methods using mid-level representations. The representative low-level features include histogram of oriented gradient (HOG) [7], scale-invariant feature transform (SIFT) [8], local binary pattern (LBP) [9] and gray-level co-occurrence matrix [10], etc. They perform well on images with simple objects and high contrasts between objects and the background, but fail to depict the characteristics of complex remote sensing scenes.

Compared with the low-level methods, mid-level approaches attempt to develop a holistic scene representation by coding the low-level local features. The popular mid-level methods include bag-of-visual-word (BoVW) [11], locality-constrained linear coding (LLC) [12], spatial pyramid matching (SPM) [13], improved fisher kernel (IFK) [14], etc. As the most popular mid-level approach, BoVW represents the image by using a histogram of visual word occurrences [11]. LLC uses the locality constraint to project each descriptor into its local-coordinate system and integrates the projected coordinates by max pooling to produce the final representation [12]. SPM builds a spatial pyramid coding of local image descriptors by using a sequence of increasingly coarser grids [13]. IFK applies Gaussian mixture model based probability densities to encoding local image features [14]. Although mid-level methods produce more impressive representations for remote sensing scenes, their performance essentially relies on low-level features. Furthermore, lacking the flexibility in discovering highly intricate structures, these methods also carry little semantic meaning [15]-[17].

Recently, convolutional neural networks (CNNs) have successfully broken the limits of traditional hard-crafted features in a variety of computer vision tasks, such as object detection [18], semantic segmentation [19], edge detection [20], and image classification [21]. AlexNet [22], VGGNet [23], GoogLeNet [24] and ResNet [25] are four of the most commonly used backbones. For instance, in [16], an end-to-end learning system was proposed to learn a feature representation with the aid of convolution layers so as to shift the burden of feature determination from hand-engineering knowledge to a deep convolutional neural network. In [23], very deep convolutional networks were investigated to extract very deep

Manuscript received January 7, 2020; revised September 7, 2020; accepted December 8, 2020. X. Wang is supported in part by the Fundamental Research Funds for the Central Universities under Grant B210202077, Six Talents Peak Project of Jiangsu Province under Grant XYDXX-007, Jiangsu Province Government Scholarship for Studying Abroad. H. Zhou is supported in part by Royal Society-Newton Advanced Fellowship under Grant NA160342, and European Union's Horizon 2020 Research and Innovation Program through the Marie-Sklodowska-Curie under Grant 720325. *(Corresponding author: Xin Wang.)*

X. Wang and S. Wang are with the College of Computer and Information, Hohai University, Nanjing 211100, China (e-mail: wang_xin@hhu.edu.cn, zhhch8815@163.com).

C. Ning is with the School of Computer and Electronic Information, Nanjing Normal University, Nanjing, Jiangsu 210023, China (e-mail: ningchen@njnu.edu.cn).

H. Zhou is with the School of Informatics, University of Leicester, Leicester LE1 7RH, United Kingdom (e-mail: hz143@leicester.ac.uk).

features for large-scale image recognition. In [25], a residual learning framework was presented to extract feature maps from input data for image recognition. In [26], an architecture using stacked autoencoders is proposed to extract high-level features for hyperspectral data classification. In [27], a pre-trained deep CNN model was selected as a feature extractor, and then the initial feature maps were fed into the CapsNet to obtain the final classification result. In [28], different global features using different CNN-based models were reported for aerial scene classification. In [29], an end-to-end convolution neural network was adopted to extract global-context features for remote sensing scene classification.

Although progress has been made in feature extraction by CNNs, there is still a large room for improvement of the generic CNN models that extract global image-level features for remote sensing scene classification, ignoring local object-level features [30]-[34]. Different from the natural images in which the objects occupy most of the space, remote sensing scene images generally contain a diversity of objects which are smaller and more decentralized than the background, as shown in Fig. 1. Due to the highly complex spatial patterns and geometric structures in remote sensing scene images, they have larger intra-class variations and smaller inter-class dissimilarity. For instance, the left subfigure of Fig. 1 shows a 'Commercial' scene, while the right subfigure illustrates a 'Dense Residential' scene. Both of these two categories of scenes contain houses, roads, trees, cars, as well as other kinds of objects. The differences between them are merely reflected in the spatial layouts and the density distributions of the objects. Hence, accurate remote sensing scene classification needs to extract not only the global image-level features, but also the local object-level ones.

To overcome the drawbacks of the vanilla CNNs for remote sensing scene classification, in this paper, we propose a new remote sensing scene classification method via enhanced feature pyramid network with deep semantic embedding. By introducing the enhanced feature pyramid network (EFPN), deep semantic embedding (DSE) module, and two-branch deep feature fusion (TDFF) module into the unified framework, the performance of remote sensing scene classification can be intrinsically improved.

We summarize our contributions as follows:

1) To address the problem that many previous CNN-based algorithms only capture global image-level features but ignore local object-level features for remote sensing scene classification, a novel pyramid-like network called EFPN is proposed to extract multi-scale multi-level features simultaneously.
2) To leverage the complementary advantages of multi-scale multi-level features, a deep semantic embedding module, DSE, is proposed. By mapping the semantics of higher-level but coarser-resolution features into lower-level with finer-resolutions, both of the stronger semantics as well as higher spatial resolutions could be reserved, so that more reliable features can be generated.
3) A two-branch deep feature fusion module named TDFF is proposed. With this module, the features at different levels can be aggregated to get complete and accurate descriptions of complex scenes.

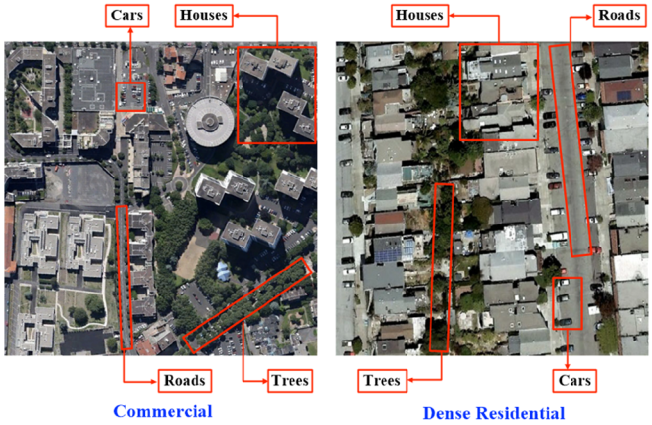


Fig. 1. Remote sensing scenes contain a diversity of objects. The highly complex spatial patterns and geometric structures in remote sensing scene images bring smaller inter-class dissimilarity for instance.

4) We evaluate our method and compare it against a number of state-of-the-art methods on two well-known benchmark datasets. Results show that our method performs favorably over all the others. Also, we provide a very comprehensive ablation study to demonstrate the effectiveness of each module in our method.

The rest of the paper is organized as follows. Section II introduces the details of the proposed method. In Section III, the experimental results are reported. Section IV discusses the effectiveness of each module in the proposed method. Finally, conclusions are drawn in Section V.

II. Proposed Method

The overall architecture of our proposed method is illustrated in Fig. 2. It contains four main modules. The first module is the enhanced feature pyramid network, which is used to produce initial feature maps at multi-levels and multi-scales. The second one, i.e., the deep semantic embedding, is designed for boosting the ability to generate features with rich semantics and high spatial resolutions. The third one is the two-branch deep feature fusion, in which two branches, namely the top branch and down branch, are designed to process and fuse different levels of features. The fused deep features are fed into the last module for RS scene classification.

A. Enhanced Feature Pyramid Network

Motivation. Current CNNs based methods prefer to cast the RS scene classification as an end-to-end problem and learn a global image-level representation from the raw image data [30], [35]-[37]. Nevertheless, the insightful consensus has pointed out that neurons in high layers respond to the whole image, while neurons in low layers are more likely to be activated by local patterns [38]. This manifests that it is necessary to utilize local object-level features extracted from low layers to further enhance the performance of RS scene classification.

To this end, we propose a pyramid-like network, which can capture both the global image-level features and local object-level features for scene reasoning. Our architecture is based on the well-known feature pyramid network (FPN) [31], which has been proposed for object detection tasks. FPN can produce a feature pyramid at multiple scales and multiple levels. Howev-

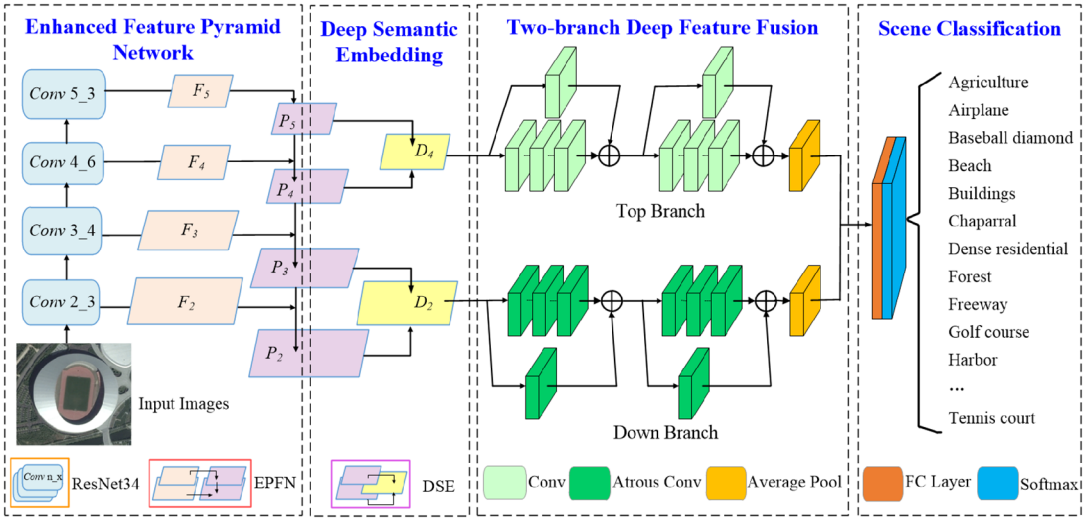


Fig. 2. The overall architecture of our proposed method.

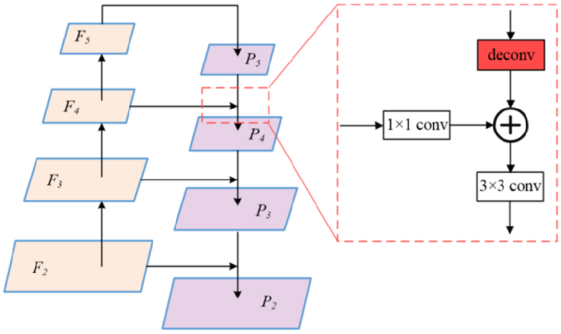


Fig. 3. Illustration of our proposed enhanced feature pyramid network.

er, FPN adopts nearest neighbors or bilinear interpolation to generate higher resolution feature maps, leading to the lack of high-frequency components of the higher resolution features, discontinuous phenomena production, as well as blurred edges of objects [32], which may influence the generation of precise feature maps for complex remote sensing scenes. This motivates us to utilize a more effective technique, that is, deconvolution, for upsampling. Compared to nearest neighbor or bilinear interpolation, deconvolution, as a vital tool in super-resolution, motion deblurring and semantic segmentation [33], can effectively complement the lost details caused by the convolutional layers in FPN and at the same time, suppress blurry edges or noise. We call our proposed pyramid-like network "enhanced feature pyramid network" (EFPN).

Enhanced Feature Pyramid Network. Fig. 3 illustrates the architecture of our enhanced feature pyramid network. It consists of a bottom-up pathway, a top-down pathway and lateral connections. It is worth noting out that, in the top-down pathway, as shown in Fig. 3, the spatial resolution is increased by deconvolution. The specific description of our EFPN is given as follows.

In the bottom-up pathway, layers of the backbone which generate feature maps of the same resolutions are defined as a stage. Considering that feature maps generated by different stages should be multi-scale and multi-level, we choose ResNet34 who has five hierarchies [39] as the basic backbone. Let $Q=\{(I_n, L_n), n=1,2,...,N\}$ denote the remote sensing scene image dataset for training, where N represents the number of training images, I_n denotes the input image, and L_n is the class label for I_n. For each image I_n, we feed it into ResNet34 [40], and calculate the outputs of each stage's last residual block. Formally, let $X, Y \in \mathbb{R}^{H\times W\times C}$ denote the input and output tensors of the last convolutional layer for a certain stage of ResNet34, where H and W denote the spatial dimensions, and C is the number of feature maps or channels. Let $\omega \in \mathbb{R}^{K\times K\times C}$ denote a $K\times K$ convolution kernel with C channels. Each feature map in $Y_{p,q} \in \mathbb{R}^{C}$ can be calculated by

$$Y_{p,q}=\mathcal{F}(X,\omega)=\sum_{i,j\in\Omega_K}\omega_{i+\frac{K-1}{2},j+\frac{K-1}{2}}^{\top}X_{p+i,q+j} \quad (1)$$

where $\mathcal{F}$ denotes a convolution layer, (p,q) represents the location coordinate and

$$\Omega_K=\left\{(i,j): i=\left\{-\frac{K-1}{2},...,\frac{K-1}{2}\right\}, j=\left\{-\frac{K-1}{2},...,\frac{K-1}{2}\right\}\right\} \quad (2)$$

defines a local neighborhood. For simplicity, here we assume that K is an odd number. Based on ResNet34, the outputs of conv2_3, conv3_4, conv4_6, conv5_3 are used as the initial bottom-up feature maps for I_n, which are denoted by $F_i \in \mathbb{R}^{H_i\times W_i\times C_i}, i=2,3,4,5$, where i represents the i-th stage of ResNet34.

In the lateral connections, to reduce the channel dimensions, we apply a 1×1 convolutional layer to each bottom-up map F_i as below

$$L_i=\mathcal{F}_1(F_i,\omega_1), i=2,3,4,5 \quad (3)$$

where $\mathcal{F}_1(\cdot,\omega_1)$ denotes a 1×1 convolution with parameters ω_1. Then, based on lateral connections, more precise locations of features can be passed from the finer levels of the bottom-up maps to the top-down ones.

In the top-down pathway, considering that the semantically stronger feature maps are spatially coarser, a deconvolution processing block is designed (dashed box in Fig. 3), in which a

deconvolutional layer is, in effect, followed by a batch normalization (BN) and a rectified linear unit (ReLU), that is Deconv-BN-ReLU. The deconvolution processing block aims to upsample the spatial resolution by a factor of 2 with a coarser-resolution feature map. The deconvolution process can be simply expressed as

$$T_i = \mathcal{G}\left(P_i, \varphi_i\right), i = 3,4,5 \tag{4}$$

where $\mathcal{G}\left(\cdot, \varphi_i\right)$ refers to a deconvolutional layer with a kernel size of 3×3 and parameters φ_i. The upsampled map is then merged with the corresponding bottom-up map by element-wise addition. Besides, to reduce the aliasing effect of upsampling, a 3×3 convolution is subsequently applied to the outputs of element-wise addition operation. Finally, the EFPN feature maps $P_i, i = 2,3,4$ can be generated.

$$P_i = \mathcal{F}_2\left(T_{i+1} \oplus L_i, \omega_2\right) \tag{5}$$

where $\oplus$ represents the element-wise addition operation and $\mathcal{F}_2\left(\cdot, \omega_2\right)$ denotes a 3×3 convolution with parameters ω_2. Note that in the top-down pathway, the coarsest resolution map P_5 is directly generated from F_5 through 1×1 convolution operation. The final outputs of EFPN are referred to as $\{P_2, P_3, P_4, P_5\}$, which correspond to $\{F_2, F_3, F_4, F_5\}$.

B. Deep Semantic Embedding

Motivation. Due to repeated downsampling and pooling operations in the bottom-up pathway, the resolutions of top feature maps are reduced. The loss of spatial details makes them unable to extract clear boundaries of small-scale objects. Our framework aggregates semantics of features by incorporating high responses of bottom features and strong activations of top features based on the fact that high responses to instances is helpful for accurately localizing objects and strong activations to semantics is an indicator for exactly understanding scenes. For this reason, we build a light-weighted and simple module, called deep semantic embedding (DSE), to aggregate features from different levels. Through this module, spatial information can be directly propagated into the target map without crossing dozens of layers. By integrating the fine details of lower-level but finer-resolution features with the semantics of higher-level but coarser-resolution features, DSE can make full use of the complementary information and learn more reliable features.

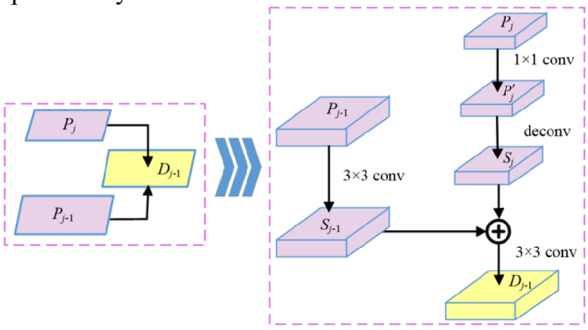


Fig. 4. Illustration of our proposed deep semantic embedding module.

Deep Semantic Embedding. Fig. 4 shows the architecture of DSE. The two-stream inputs of a DSE module are $P = \{P_j, P_{j-1}\}, j \in \{3,5\}$, where $P_j \in \mathbb{R}^{H_j \times W_j \times C_j}$ corresponds to the j-th level feature maps of EFPN. First, to capture more accurate representations, we apply two convolutional layers, with the kernel sizes of 3×3 and 1×1 respectively, to adjacent-level features for cross-channel information interaction and integration. Then, by applying a deconvolution operation, we upsample the higher-level feature maps to the scale of lower-level ones. The process can be represented by

$$S_{j-1} = \mathcal{F}_2\left(P_{j-1}, \omega_2\right) \tag{6}$$

where $\mathcal{F}_2\left(\cdot, \omega_2\right)$ denotes the 3×3 convolution with parameters ω_2.

$$S_j = \mathcal{G}\left(\mathcal{F}_1\left(P_j, \omega_1\right), \varphi_j\right) \tag{7}$$

where $\mathcal{F}_1\left(\cdot, \omega_1\right)$ denotes the 1×1 convolution with parameters ω_1 and $\mathcal{G}\left(\cdot, \varphi_j\right)$ refers to the deconvolutional layer in DSE.

Subsequently, we embed the strong semantic information of the upsampled features into the lower-level features by using element-wise addition. Similarly, to alleviate the aliasing effect, a 3×3 convolution is applied to the merged maps to get the final feature maps $D_{j-1}, j = \{3,5\}$ of DSE. The final outputs of DSE can be computed by

$$D_{j-1} = \mathcal{F}_3\left(S_j \oplus S_{j-1}, \omega_3\right) \tag{8}$$

where $\mathcal{F}_3\left(\cdot, \omega_3\right)$ denotes the 3×3 convolution with parameters ω_3.

C. Two-branch Deep Feature Fusion

Motivation. In the generic FPN, a proposal on a specific level is chosen for recognition according to the size of objects, since object detection only needs to assign a specific category to an individual object. Albeit simple and efficient, it cannot meet the demand of RS scene classification, because we infer the scene label via recognizing the combined characteristics of multiple discriminative objects rather than a single object.

Our motivation stems from the fact that, when manually classifying RS scene images, specialists always set the semantic labels based on the global characteristics of scenes as well as the local features of objects [2], [18]. Therefore, we believe that global image-level and local object-level features are two important representations for distinguishing RS scenes. To be specific, the higher-level feature maps generated by global receptive fields give the strongly semantic features. Making lower-level features access them will better absorb meaningful contextual information for prediction. On the contrary, the lower-level feature maps generated by local receptive fields reflect refined details for locating objects. Such features can help higher-level features to complement their loss of spatial information, which is beneficial for classification. Therefore, based on the above analysis, we present a

two-branch deep feature fusion module to fuse the features at different levels in a more effective way.

Two-branch Deep Feature Fusion. This architecture consists of two branches to deal with the higher-level and lower-level feature maps, respectively. To process various levels and at the same time, enlarge the receptive fields so as to incorporate multi-level contextual information without increasing computational cost, we advocate the use of the combination of convolution and atrous convolution. Thereinto, atrous convolution, also known as dilated convolution, has been verified to be a powerful tool for dense prediction tasks [41], [42]. In addition, to avoid network degradation that may be caused by excessive depth, we also introduce skip connection into the architecture. Further, the last layers in the architecture are two global average pooling (GAP) layers, which are used to generate image-level representation features. A high-level illustration of the presented two-branch deep feature fusion module is shown in Fig. 5.

Top Branch: This branch receives the higher-level feature maps D_4 produced by DSE. It is equipped with two residual blocks and one global average pooling layer. Table I shows the details of the residual blocks in this top branch, in which the 1×1, 3×3 and 3×3 convolutional layers are arranged orderly to learn deep feature efficiently. Note that each layer is also followed by a batch normalize layer and a ReLU layer for nonlinear transformation, and the outputs of internal paths for a residual block are combined by element-wise addition.

The output of one residual block can be represented by

$$Y=\Psi\left(\mathcal{F}_4\left(X,\omega_4\right)\oplus\mathcal{F}\left(X,\{\omega_i\}\right)\right) \tag{9}$$

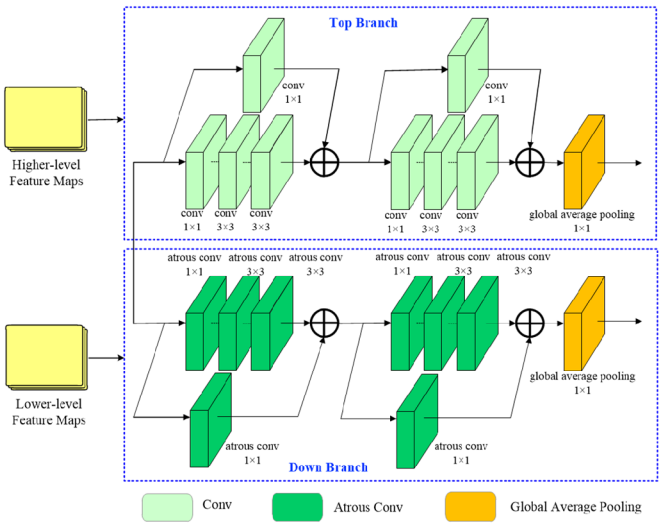


Fig. 5. Illustration of our proposed two-branch deep feature fusion module.

TABLE I
DETAILS OF THE RESIDUAL BLOCKS IN THE TOP BRANCH.

Residual Blocks	Layer	Kernel_ Size	Padding	In_ Channels	Out_ Channels
Main Pipeline	Conv1	1×1	0	256	64
	Conv2	3×3	1	64	64
	Conv3	3×3	1	64	256
Secondary Pipeline	Conv4	1×1	0	256	256

where

$$\mathcal{F}\left(X,\{\omega_i\}\right)=\mathcal{F}_3\left(\mathcal{F}_2\left(\mathcal{F}_1\left(X,\omega_1\right),\omega_2\right),\omega_3\right) \tag{10}$$

where X and Y denote the input and output of the residual block, respectively. $\mathcal{F}_1$, $\mathcal{F}_2$, $\mathcal{F}_3$ and $\mathcal{F}_4$ denote the 1×1, 3×3, 3×3 and 1×1 convolutional layers in TDFF. ω_1, ω_2, ω_3 and ω_4 are the corresponding parameters. Ψ represents the ReLU function, as shown in Eq. 11. In Eq. 9, the BN and ReLU are omitted for simplifying notations.

$$\Psi\left(x\right)=\max\left(x,0\right) \tag{11}$$

We can now write the outputs of the two residual blocks as

$$Y_t^1=\Psi\left(\mathcal{F}_4^1\left(D_4,\omega_4^1\right)\oplus\mathcal{F}_3^1\left(\mathcal{F}_2^1\left(\mathcal{F}_1^1\left(D_4,\omega_1^1\right),\omega_2^1\right),\omega_3^1\right)\right) \tag{12}$$

$$Y_t^2=\Psi\left(\mathcal{F}_4^2\left(Y_t^1,\omega_4^2\right)\oplus\mathcal{F}_3^2\left(\mathcal{F}_2^2\left(\mathcal{F}_1^2\left(Y_t^1,\omega_1^2\right),\omega_2^2\right),\omega_3^2\right)\right) \tag{13}$$

where Y_t^1 and Y_t^2 denote the outputs of the first and second residual blocks in the top branch, respectively. Note that, in Eq. 12, the input is the higher-level feature maps D_4 and in Eq. 13, the input is the output of the first residual block Y_t^1.

After two residual blocks, GAP [43] is introduced to strengthen the correspondences between categories and feature maps, and generate deep features. Ultimately, the output features of the top branch can be described as

$$Branch_t=\sigma(Y_t^2)=\frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W}Y_{i,j}^l \tag{14}$$

where σ denotes the global average pooling operation and $Y^l\in\mathbb{R}^{H\times W}$ is a feature map with height H and width W for the l-th channel of the input Y_t^2.

Down Branch: Compared with the top branch, the down branch replaces the convolutional layers in each residual block with atrous convolutional ones, because of the different scales of the inputs. By virtue of atrous convolution, our down branch is able to enlarge the receptive fields, thus capturing objects or useful image contextual information for classifying complex RS scenes. The information of residual blocks in the down branch is given in Table II.

Compared with standard convolution, atrous convolution can enlarge the kernel via integrating holes between pixels in kernels [44]. In the down branch, we utilize atrous convolution to increase the receptive field of the output units without increa sing the kernel size. Generally, an atrous convolution with a ke-

TABLE II
DETAILS OF THE ATROUS RESIDUAL BLOCKS IN THE DOWN BRANCH.

Atrous Residual Blocks	Layer	Kernel_ Size	Dilation	In_ Channels	Out_ Channels
Main Pipeline	AtrousConv1	1×1	1	256	64
	AtrousConv2	3×3	3	64	64
	AtrousConv3	3×3	5	64	256
Secondary Pipeline	AtrousConv4	1×1	1	256	256

rnel size $K \times K$ and atrous rate r has a receptive filed of

$$M = K + (r-1)(K-1) = rK - r + 1 \tag{15}$$

As a result, the output of $M \times M$ convolution, which can be calculated by Eq. 1, may be used as the result of $K \times K$ atrous convolution. In detail, suppose $U, V \in \mathbb{R}^{H \times W \times C}$ are the input and output tensors of an atrous convolutional layer, where H and W denote the spatial dimensions, and C is the number of channels. Each feature map in $V_{p,q} \in \mathbb{R}^{C}$ with the location coordinate (p,q) can be computed by

$$V_{p,q} = \sum_{i,j \in \Omega_M} \mu_{i+\frac{M-1}{2}, j+\frac{M-1}{2}}^{\top} U_{p+i,q+j} \tag{16}$$

where

$$\Omega_M = \left\{ (i,j) : i = \left\{ -\frac{M-1}{2}, ..., \frac{M-1}{2} \right\}, j = \left\{ -\frac{M-1}{2}, ..., \frac{M-1}{2} \right\} \right\} \tag{17}$$

denotes a local neighborhood and μ are the parameters of atrous convolution.

Based on atrous convolution, the output of one residual block in the down branch can be calculated by

$$V = \Psi\left(\mathcal{H}_4(U, \mu_4) \oplus \mathcal{H}(U, \{\mu_i\}) \right) \tag{18}$$

where

$$\mathcal{H}(U, \{\mu_i\}) = \mathcal{H}_3\left(\mathcal{H}_2\left(\mathcal{H}_1(U, \mu_1), \mu_2 \right), \mu_3 \right) \tag{19}$$

where U and V denote the input and output of the residual block in the down branch, respectively. $\mathcal{H}_1$, $\mathcal{H}_2$, $\mathcal{H}_3$ and $\mathcal{H}_4$ denote the 1×1, 3×3, 3×3 and 1×1 atrous convolutional layers in TDFF. μ_1, μ_2, μ_3 and μ_4 are the corresponding parameters.

Similarly, we can obtain the outputs of the two residual blocks as

$$V_d^1 = \Psi\left(\mathcal{H}_4^1\left(D_2, \mu_4^1 \right) \oplus \mathcal{H}_3^1\left(\mathcal{H}_2^1\left(\mathcal{H}_1^1\left(D_2, \mu_1^1 \right), \mu_2^1 \right), \mu_3^1 \right) \right) \tag{20}$$

$$V_d^2 = \Psi\left(\mathcal{H}_4^2\left(V_d^1, \mu_4^2 \right) \oplus \mathcal{H}_3^2\left(\mathcal{H}_2^2\left(\mathcal{H}_1^2\left(V_d^1, \mu_1^2 \right), \mu_2^2 \right), \mu_3^2 \right) \right) \tag{21}$$

where V_d^1 and V_d^2 denote the outputs of the first and second residual blocks in the down branch, respectively. Note that, in Eq. 20, the input is the lower-level feature maps D_2 and in Eq. 21, the input is the output of the first residual block V_d^1.

Subsequently, the global average pooling is applied to V_d^2 to acquire the deep features for the down branch. As a result, the output features of the down branch can be described as

$$Branch_d = \sigma(V_d^2) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} V_{i,j}^{l} \tag{22}$$

where $V^l \in \mathbb{R}^{H \times W}$ is a feature map with height H and width W for the l-th channel of the input V_d^2.

Ultimately, TDFF with atrous convolutions effectively captures two-branch deep features, i.e., $Branch_t$ and $Branch_d$. The serial feature fusion scheme is adopted to concatenate these two different types of features together, so as to obtain more significant and informative features to represent the RS scene images.

D. Scene Classification

The fused deep features are subsequently fed into the next scene classification module. This module, composed of the fully connected layer and softmax layer, is utilized to predict the class label for the input image.

Suppose the output of the fully connected layer is $Z = \{z_i, i = 1, 2, ..., m\}$, where m is the total number of class labels. The softmax function is defined as:

$$\theta_i = \frac{\exp(z_i)}{\sum_{j=1}^{m} \exp(z_j)} \tag{23}$$

$$\theta = \max(\theta_1, \theta_2, ..., \theta_m) \tag{24}$$

where θ_i represents the probability that the input image belongs to the i-th class. The final predicted label is determined by θ. Besides, during the process of classification, our loss function is the cross-entropy loss [45], which is given by

$$Loss = -\frac{1}{N} \sum_{n=1}^{N} \sum_{j=1}^{m} \mathbf{1}\{y^n = j\} \log \theta_j \tag{25}$$

where y is the real scene label, m is the number of scene categories, N denotes the size of mini-batch, and $\mathbf{1}\{*\}$ represents an indicator function. Mathematically, if y^n is equal to i, $\mathbf{1}\{y^n = i\} = 1$, else $\mathbf{1}\{y^n = i\} = 0$.

The proposed method is summarized in Algorithm 1.

III. Experiments

In this section, we evaluate our proposed method on two publicly available datasets for remote sensing scene classification. Firstly, the datasets used in experiments are described. Secondly, we give an introduction of the experimental setup. Finally, the proposed architecture is compared with a number of state-of-the-art algorithms.

A. Datasets

We test the proposed method on two different remote sensing scene datasets. One is the well-known UCMerced Land-Use dataset (referred to as UCM) [46], and the other is the Aerial Image dataset (referred to as AID) [28].

UCM: This dataset has 2100 remote sensing scene images, each of which is categorized into a certain class. These scene images with the RGB color space, which originate from 20 diverse regions, are all provided by United States Geological Survey (USGS) National Map. There are 21 scene labels in total, including Agricultural, Airplane, Baseball Diamond, etc. Some example images of all the categories in UCM are illustrated in Fig. 6. Each class consists of 100 images with the size of 256×256, and spatial resolution of 30 cm per pixel.

AID*:* This dataset is from Wuhan University, which is also available online. All images are categorized into 30 classes: Airport, BareLand, Baseball Field, etc. A total of 10000 images

are in the dataset; however, the number of images in each category varies from 220 to 420. Table III illustrates the detailed information of AID, and some example images of all categories are shown in Fig. 7. Each image has the size of 600×600, but the spatial resolution ranges from 1m to 8m.

B. Experimental Setup

Algorithm 1: The Proposed Method

Step 1 Enhanced Feature Pyramid Network
Input: Full Image I
Output: Enhanced Feature Map P
1: Input I to the pre-trained ResNet34, the convolutional feature maps of different stages are reserved as F_2, F_3, F_4, F_5.
2: $P_5 \equiv L_5 \leftarrow \mathcal{F}_1(F_5, \omega_1)$
3: **for** i = 4; i ≥ 2; i ← i-1 **do**
4: $L_i \leftarrow \mathcal{F}_1(F_i, \omega_1)$
5: $T_{i+1} \leftarrow \mathcal{G}(P_{i+1}, \varphi_{i+1})$
6: $P_i \leftarrow \mathcal{F}_2(T_{i+1} \oplus L_i, \omega_2)$
7: **end for**
Step 2 Deep Semantic Embedding
Input: Enhanced Feature Map P
Output: Deep Semantic Embedding Feature Map D
8: **for** j = 3; j ≤ 5; j ← j + 2 **do**
9: $S_{j-1} \leftarrow \mathcal{F}_2(P_{j-1}, \omega_2)$
10: Sj $\leftarrow \mathcal{G}(\mathcal{F}_1(P_j, \omega_1), \varphi_j)$
11: $D_{j-1} \leftarrow \mathcal{F}_3(S_j \oplus S_{j-1}, \omega_3)$
12: **end for**
Step 3 Two-branch Deep Feature Fusion
Input: Deep Semantic Embedding Feature Map D
Output: Fused Deep Feature I_{out}
13: **for** k = 1; k ≤ 2; k ← k + 1 **do**
14: **if** k == 2 **do** $D_4 = Y_t^1$, $D_2 = V_d^1$
15: $Y_t^k \leftarrow \Psi\left(\mathcal{F}_4^k\left(D_4, \omega_4^k\right) \oplus \mathcal{F}_3^k\left(\mathcal{F}_2^k\left(\mathcal{F}_1^k\left(D_4, \omega_1^k\right), \omega_2^k\right), \omega_3^k\right)\right)$
16: $V_d^k \leftarrow \Psi\left(\mathcal{H}_4^k\left(D_2, \mu_4^k\right) \oplus \mathcal{H}_3^k\left(\mathcal{H}_2^k\left(\mathcal{H}_1^k\left(D_2, \mu_1^k\right), \mu_2^k\right), \mu_3^k\right)\right)$
17: **end for**
18: $Branch_t \leftarrow \sigma(Y_t^2)$
19: $Branch_d \leftarrow \sigma(V_d^2)$
20: Concatenate $Branch_t$ and $Branch_d$ in channel dimension to form the final fused deep feature I_{out}
Step 4 Scene Classification
Input: Fused Deep Feature I_{out}
Output: Predicted Label L
21: Calculate the output of the fully connected layer Z
22: **for** i = 1; i ≤ m; i ← i + 1 **do**
23: $\theta_i \leftarrow \dfrac{\exp(z_i)}{\sum_{j=1}^{m} \exp(z_j)}$
24: **end for**
25: $\theta \leftarrow \max(\theta_1, \theta_2, ..., \theta_m)$
26: The final predicted label L is determined by θ.

Training/Testing. To make a comprehensive evaluation, the training-testing ratios for UCM are set to 80%-20% and 50%-50%, and the training-testing ratios for AID are set to 50%-50 and 20%-80%. We randomly select the samples from each scene category for training and leave the remaining images for testing. All images are resized to 224×224. Besides, to enhance the generalization ability of our method, some data augmentation techniques, including random horizontal flipping and random rotation, are adopted.

Fig. 6. Example images of UCM dataset.

TABLE III
THE NUMBER OF IMAGES FOR EACH CLASS OF AID DATASET.

Class	Number	Class	Number	Class	Number
Airport	360	Farmland	370	Port	380
Bare land	310	Forest	250	Railway station	260
Baseball field	220	Industrial	390	Resort	290
Beach	400	Meadow	280	River	410
Bridge	360	Medium residential	290	School	300
Center	260	Mountain	340	Sparse residential	300
Church	240	Park	350	Square	330
Commercial	350	Parking	390	Stadium	290
Dense residential	410	Playground	370	Storage tanks	360
Desert	300	Pond	420	Viaduct	420

Fig. 7. Example images of AID dataset.

Implementation Details. The proposed method is constructed based on the Pytorch library on Google Colaboratory, which is a cloud platform with a NVIDIA Tesla T4 GPU and 16G memory. Parameters of the backbone (ResNet34) are initialized from the official model pre-trained via ImageNet. In our framework, we discard the last global average pooling and fully connected layers of ResNet34 and introduce several lateral connections and DSE. Among them, the weights of 1×1 convolutional layers are initialized as 0.1, and the biases are initialized as 0. The kernel size, stride, padding and out padding of the deconvolutional layers are set to 3, 2, 1 and 1, respectively. In TDFF, the dilation_rates of the atrous convolutional layers in a block are set to 1, 3, and 5 orderly and the other parameters are initialized as the PyTorch default settings. We use the stochastic gradient descent (SGD) approach to optimize the proposed model. The initial learning rate is set to 1e-2 and is divided by 10 every 80 epochs, while the momentum is set to 0.9. The minibatch of SGD depends on the running data. In the testing phase, the batch sizes are 10 and 36 for the UCM and AID datasets, respectively, while in the training phase, the batch sizes of both datasets are set to 72. All images are resized to 224×224.

Evaluation Metrics. Two widely used metrics, i.e., overall accuracy and confusion matrix [47]-[49], are selected here to evaluate the performance of our proposed algorithm. Overall accuracy is the proportion of the number of correct predictions to the total number of samples, reflecting the overall classification performance of a classification method. Furthermore, to analyze the detailed classification errors between different classes, a specific table layout named confusion matrix is used, each column/row of which denotes the instances in a predicted/actual class. It is worth pointing out that, to achieve reasonable results of evaluation metrics as well as reduce the influence of randomness, all experiments are repeated ten times.

C. Comparison with State-of-the-arts

The proposed method, called EFPN-DSE-TDFF, is compared with a set of state-of-the-art algorithms, including SCK [11], BoVW [28], Dense-SIFT [50], IFK [28], salM3LBP-CLM [4], CaffeNet [28], GoogLeNet [28], VGG-VD-16 [28], TEX-Net-LF [51], DCA with concatenation [52], CaffeNet-DCF [53], VGG-VD16-DCF [53], CNN-NN [36], AlexNet+VGG16 [47], VGG-16-CapsNet [27], Fusion by addition [52], and Fusion by concatenation [52].

Thereinto, SCK, BoVW, Dense-SIFT, IFK and salM3LBP-CLM are the remote sensing scene classification methods based on midlevel scene features, while the other comparing approaches are based on high-level deep features. For those models with available code, we train and test those models using their default settings. For those models without released code, we used their results proposed in their original works. Also, for a fair comparison, the same ratios were applied in the following experiments according to the experimental settings in the related works. For the UCM dataset, the ratios of the number of the training set are set to 80% and 50%, respectively, while for the AID dataset, the ratios are fixed at 50% and 20%, respectively.

1) Experimental results on UCM dataset

In this section, we compare our proposed method with a number of state-of-the-arts on the widely used UCM dataset. The experimental results and analysis consists of four parts: the overall accuracy results under the training ratios of 80% and 50%, and the confusion matrix results under the training ratios of 80% and 50%.

We randomly select the fixed percent of the images to construct the training set by repeating ten times on the UCM dataset, and then compute the means and standard deviations of overall accuracy. The results are given in Table IV. From this table, we can find that, among the four kinds of midlevel methods, i.e., SCK, BoVW, Dense-SIFT, and salM3LBP-CLM, salM3LBP-CLM performs much better than others. However, its performance is still inferior to most of the other high-level deep feature methods shown in Table IV. This indicates that the midlevel methods have limited abilities for RS scene classification. On the contrary, benefiting from the superiority of deep neural networks, significant improvement of scene classification performance has been made by deep feature methods for remote sensing images.

Comparing the results of various deep feature methods, our method, EFPN-DSE-TDFF, achieves the best performance with the ratios of 80% and 50%. VGG-16-CapsNet and AlexNet+VGG16 obtain the 2nd best results with the training ratio of 80%, while under the training ratio of 50%, VGG-16-CapsNet, TEX-Net-LF, VGG-VD16-DCF, and CaffeNet-DCF achieve competitive performances. In addition, our method also has much smaller standard deviations with both ratios.

All the above phenomenon indicates that our proposed strategy of combining enhanced feature pyramid network, deep semantic embedding, and two-branch deep feature fusion in a unified framework is effective to improve the classification performance for RS scenes.

Besides the overall accuracy, we also compute the confusion matrix for the proposed method. With different fixed training ratios, we choose to show the best results. Fig. 8 shows the confusion matrix with the training ratio of 80%. From Fig. 8, we can see that most scene categories obtain the classification accuracy equal to 1. Categories with classification accuracy lo-

TABLE IV
OVERALL ACCURACY (%) OF DIFFERENT METHODS WITH THE TRAINING RATIOS OF 80% AND 50% ON UCM DATASET

Methods	Overall Accuracy	
	80% Training Ratio	50% Training Ratio
SCK [11]	72.52	/
BoVW [28]	74.12 ± 3.30	71.90 ± 0.79
Dense-SIFT [50]	81.67 ± 1.23	/
salM^{3}LBP-CLM [4]	95.75 ± 0.80	94.21 ± 0.75
CaffeNet [28]	95.02 ± 0.81	93.78 ± 0.67
GoogLeNet [28]	94.31 ± 0.89	92.70 ± 0.60
VGG-VD-16 [28]	95.21 ± 1.20	94.14 ± 0.69
TEX-Net-LF [51]	96.62 ± 0.49	95.89 ± 0.37
CNN-NN [36]	97.19	/
CaffeNet-DCF [53]	96.79 ± 0.66	95.26 ± 0.50
VGG-VD16-DCF [53]	97.10 ± 0.85	95.42 ± 0.71
AlexNet+VGG16 [47]	98.81 ± 0.38	/
VGG-16-CapsNet [27]	98.81 ± 0.12	95.33 ± 0.18
EFPN-DSE-TDFF (Ours)	99.14 ± 0.22	96.19 ± 0.13

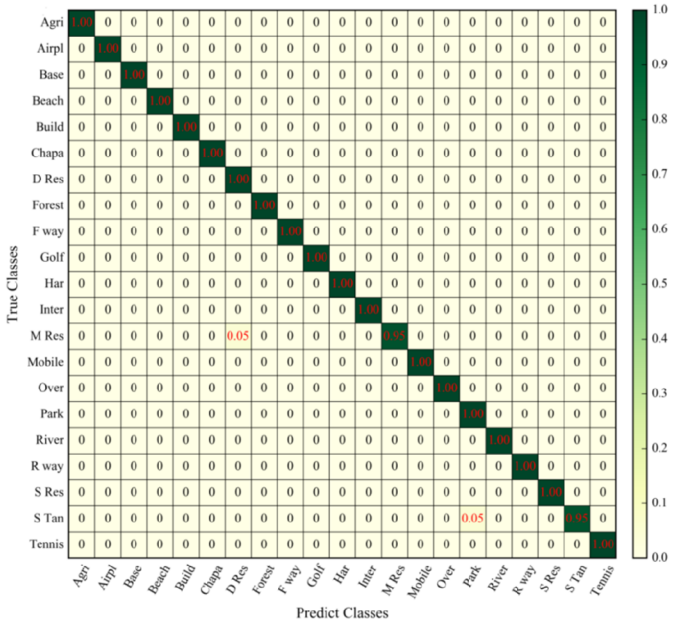


Fig. 8. Confusion matrix of the proposed method under the training ratio of 80% on UCM dataset.

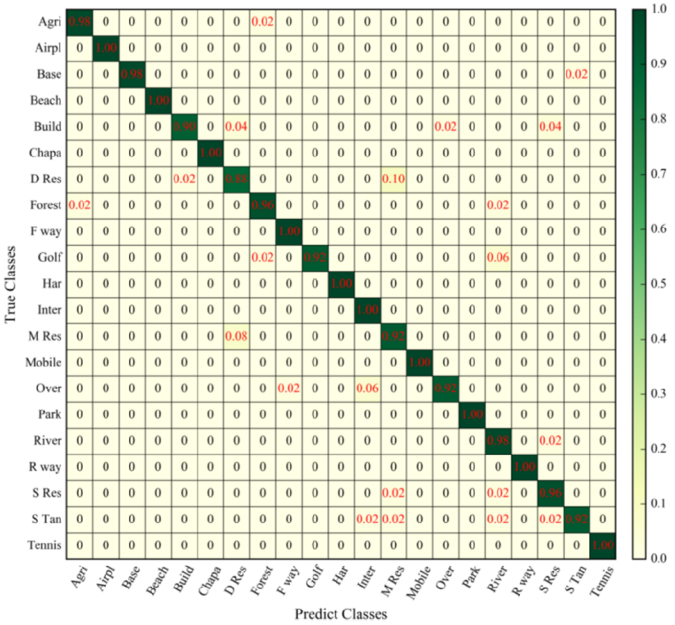


Fig. 9. Confusion matrix of the proposed method under the training ratio of 50% on UCM dataset.

wer than 1 only includes 'Medium Residential' (0.95) and 'Storage Tanks' (0.95). As is known, for the UCM dataset, the most confused scene types are 'Dense Residential', 'Medium Residential', and 'Sparse Residential', due to their similar spatial patterns and the common objects shared by them such as buildings and trees. In our confusion matrix, 5% images from 'Medium Residential' are mistakenly classified as 'Dense Residential'. Besides, 5% images from 'Storage Tanks' are mistakenly classified as 'Parking Lot', which may attribute to their similar land cover types.

Fig. 9 presents the confusion matrix with the training ratio of 50%. From this confusion matrix, we can observe that 15 of the 21 categories achieve the classification accuracy over 95%. Apart from these, the scene categories with the classification accuracy rate of more than 90% include 'Building' (0.90), 'Golf Course' (0.92), 'Medium Residential' (0.92), 'Overpass' (0.92), and 'Storage Tanks' (0.92). The most obvious confusion is still between 'Dense Residential' and 'Medium Residential'. As can be seen, 10% images from 'Dense Residential' are mistakenly classified as 'Medium Residential', while 8% images from 'Medium Residential' are classified as 'Dense Residential' by mistake. Besides, 6% images from 'Overpass', are mistakenly classified as 'Intersection', due to their similar appearances. And 6% images from 'Golf Course', are mistakenly classified as 'River', for both of them contains large areas of trees.

2) Experimental results on AID dataset

In this experiment, we use the publicly available remote sensing scene dataset, i.e., AID dataset to evaluate the effectiveness of the proposed method.

The comparative results of our method against a number of state-of-the-art scene classification algorithms over the AID 30-class scenes are shown in Table V. The overall accuracy of midlevel methods, i.e., BoVW, IFK, and salM3LBP-CLM are 67.65%, 77.33%, and 89.76% respectively under the training ratio of 50%, and 61.40%, 70.60%, 86.92% respectively with the training ratio of 20%. Compared with the midlevel methods, the methods based on deep features achieve far better performance, which indicates that the deep features are more informative and discriminative than the hand-crafted descriptors.

TABLE V
OVERALL ACCURACY (%) OF DIFFERENT METHODS WITH THE TRAINING RATIOS OF 50% AND 20% ON AID DATASET

Methods	Overall Accuracy	
	50% Training Ratio	20% Training Ratio
BoVW [28]	67.65 ± 0.49	61.40 ± 0.41
IFK [28]	77.33 ± 0.37	70.60 ± 0.42
salM^{3}LBP-CLM [4]	89.76 ± 0.45	86.92 ± 0.35
CaffeNet [28]	89.53 ± 0.31	86.86 ± 0.47
GoogLeNet [28]	86.39 ± 0.55	83.44 ± 0.40
VGG-VD-16 [28]	89.64 ± 0.36	86.59 ± 0.29
DCA with concatenation [52]	89.71 ± 0.33	/
Fusion by addition [52]	91.87 ± 0.36	/
Fusion by concatenation [52]	91.86 ± 0.28	/
TEX-Net-LF [51]	92.96 ± 0.18	90.87 ± 0.11
VGG16+CapsNet [27]	94.74 ± 0.17	91.63 ± 0.19
EFPN-DSE-TDFF (Ours)	94.50 ± 0.30	94.02 ± 0.21

Moreover, among all the deep feature methods, our method, i.e., EFPN-DSE-TDFF, and VGG16+CapsNet achieve comparable results (94.50% and 94.74%) with the training ratio of 50%. When the training ratio drops to 20%, our method still performs better (94.03%), while the overall accuracy of VGG16+CapsNet drops to 91.63%. It indicates that the strong discriminative power of our EFPN-DSE-TDFF compared to VGG16+CapsNet, providing a more robust representation for remote sensing scenes from AID.

Due to limited space, we only report the confusion matrixes of our method under the training ratios of 50% and 20% in Figs. 10 and 11, respectively. As shown in Fig. 10, 25 of the 30 categories achieve the classification accuracy of more than

90%. Categories with classification accuracy lower than 0.9 include 'Center' (0.88), 'Industrial' (0.88), 'Resort' (0.69), 'School' (0.79), and 'Square' (0.89). Over the AID dataset, the major confusions occur between 'Resort' and 'Park', 'School' and 'Commercial', 'Stadium' and 'Playground', or 'BareLand' and 'Desert'. These results are explained by the fact that these categories have similar ground object distributions or geometrical structures. However, some types with large inter-class similarity, such as 'Medium Residential' (0.93) and 'Sparse Residential' (0.99), can be accurately classified. In addition, 'Bridge', 'Port', and 'River', which have the analogous objects and image textures, also achieve high overall accuracy of 0.97, 0.96, and 1.00.

Fig. 11 demonstrates the confusion matrix with the training ratio of 20%, in which the row reflects the producer's accuracy, while the column reflects the user's accuracy. From this figure, we can see that most of the classes obtain a satisfactory classification result over 90%, and only five categories including 'Center', 'Park', 'Resort', 'School', 'Square' have a bit severe misclassification. Although the number of the training images has decreased dramatically, some categories that are easily confused can be still effectively classified, such as 'Medium Residential' (0.92) and 'Sparse Residential' (0.99), or 'Bridge' (0.98), 'Port' (0.98), and 'River' (0.99). The reason why our proposed method obtains effective scene classification results can attribute to different modules including enhanced feature pyramid network, deep semantic embedding, two-branch deep feature fusion in the unified framework.

From the above experimental results, we can summarize some interesting observations as follows. By comparing the proposed method with different scene classification algorithms, it can be seen that midlevel methods achieve worse performances, while deep feature methods perform better on all the datasets. Also, among the deep feature methods, our proposed framework has better performance, indicating the progress in RS scene classification.

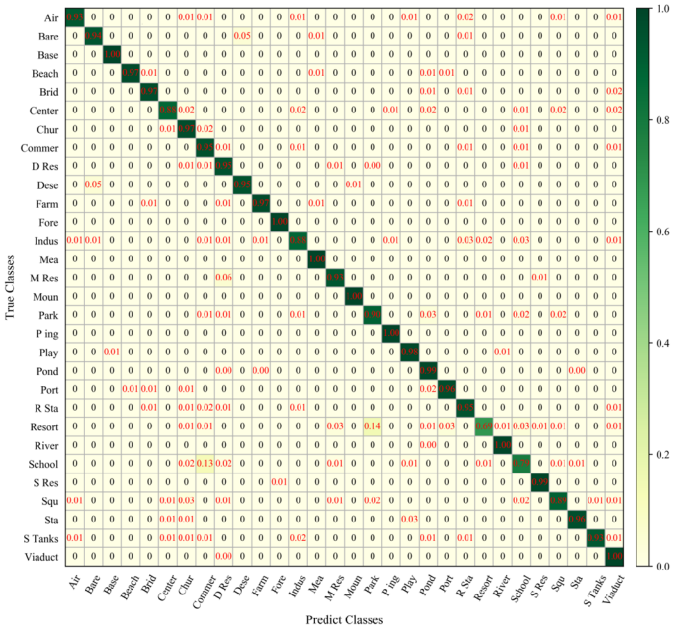


Fig. 10. Confusion matrix of the proposed method under the training ratio of 50% on AID dataset.

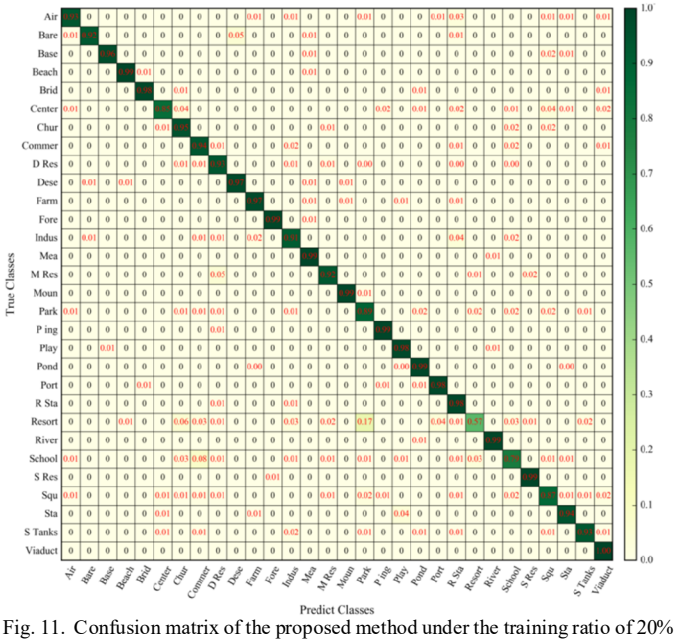


Fig. 11. Confusion matrix of the proposed method under the training ratio of 20% on AID dataset.

IV. Discussion

To comprehensive evaluate the effectiveness of our proposed method, various ablation experiments are performed by using different connection patterns or different design options.

A. Impact of Data Augmentation

Data augmentation has been proven to be very useful in many learning based vision tasks [45]. Therefore, in our experiments, we also employ data augmentation to generate enhanced data to train an effective model. The input images are augmented by randomly horizontal flipping and random rotation during training, which results in an augmented image set richer than the original one. We compare the methods with and without data augmentation so as to verify the effectiveness of data augmentation. The comparison results are shown in Table VI. In this table, ResNet34+ represents our proposed method that adopts data augmentation, while ResNet34* represents the corresponding method without data augmentation. From experimental results, we can find that adopting data augmentation improves the overall accuracy by more than 0.6%.

B. Scalability

There are two popular backbone networks (i.e., VGG-VD-16 and ResNet34) that are utilized in scene classification. To further validate the scalability of our proposed method, we conduct comparison experiments using different backbones in Table VI. For both of backbones, we choose the last outputs of each stage (except stage 1) as the initial inputs of top-down pathway. We keep all other settings the same. The comparison results are shown in Table VI. As can be seen, with the same training set, the performance of our ResNet34 based architectures (ResNet34+) are much better than that of the VGG-VD-16 based architecture.

TABLE VI
ABLATION ANALYSIS ON DATA AUGMENTATION AND DIFFERENT BACKBONES

Backbone	UCM 80%	UCM 50%	AID 50%	AID 20%
ResNet34^{+}	99.14	96.19	94.50	94.02
ResNet34*	98.48	95.37	93.75	92.33
VGG-VD-16	95.71	94.76	90.24	87.62

TABLE VII
OVERALL ACCURACY (%) OF DIFFERENT ARCHITECTURES WITH THE TRAINING RATIOS OF 80% AND 50% ON UCM DATASET.

Scheme	Architecture	Overall Accuracy	
		80% Training Ratio	50% Training Ratio
1	Without EFPN	91.38 ± 0.53	89.81 ± 0.39
2	Without DSE	96.36 ± 0.80	94.68 ± 0.52
3	Without TDFF	97.02 ± 0.54	95.91 ± 0.33
4	EFPN-DSE-TDFF (Ours)	99.14 ± 0.22	96.19 ± 0.13

C. Effects of Different Modules

Our framework contains three main modules, i.e., EFPN, DSE and TDFF. To analyze the importance of each main module, a series of ablation experiments are conducted with different architecture designs. For each architecture, we adopt the controlling method that only omits one module at a time. The concise illustration of various architectures is shown in Fig. 12, and Fig. 12 (a), (b) and (c) represent the architectures in which EFPN, DSE and TDFF are omitted, respectively.

For fair comparisons, all the results are tested on the UCM dataset. The overall accuracy of different architectures under the training ratios of 80% and 50% on UCM dataset is listed in Table VII, and the following can be seen from the results.

Effects of EFPN: Results from Scheme 1 are the worst, which is because the EFPN is omitted from this architecture, while the function of EFPN is to initially strengthen semantics of all level feature maps. Compared with Scheme 1, since Schemes 2, 3 and 4 contain the EFPN, the overall accuracy of them increases by 4.98%, 5.64%, 7.76% under 80% training ratio respectively, and 4.87%, 6.1%, 6.38% under 50% training ratio respectively. These results demonstrate that the EFPN we address is indeed beneficial for RS scene classification.

Effects of DSE: Scheme 2 directly links the TDFF to the outputs of EFPN without deep semantic embedding. Compared with it, by using DSE, our method, i.e., Scheme 4 achieve better performance, with an increase in overall accuracy of 2.78% and 1.51% under 80% and 50% training ratios, respectively. This phenomenon strongly validates the effectiveness and superiority of our proposed DSE.

Effects of TDFF: In Scheme 3, we delete the TDFF and replace it with a simple global average pooling layer, GAP for the subsequent scene classification. From Table VII, one can find that despite better performance is obtained by Scheme 3 when comparing with Schemes 1 and 2, there are still slight decreases when comparing with Scheme 4. In detail, there are slight decreases in overall accuracy of 2.12% and 0.28% under 80% and 50% training ratios.

Besides overall accuracy, we also report the per-class classification accuracies of these four different architectures in Fig. 13. As can be seen from Fig. 13 (a) and (b), no matter whether the training ratio is 80% or 50%, our method achieves the best performance, which indicates the effectiveness and superiority of our proposed architecture.

Moreover, we report the accuracy comparison achieved after convergence of different architectures on UCM dataset, as shown in Fig. 14. As we can see, the accuracy of our method is much more stable and higher than those of other three architectures under the training ratios of 80% as well as 50%.

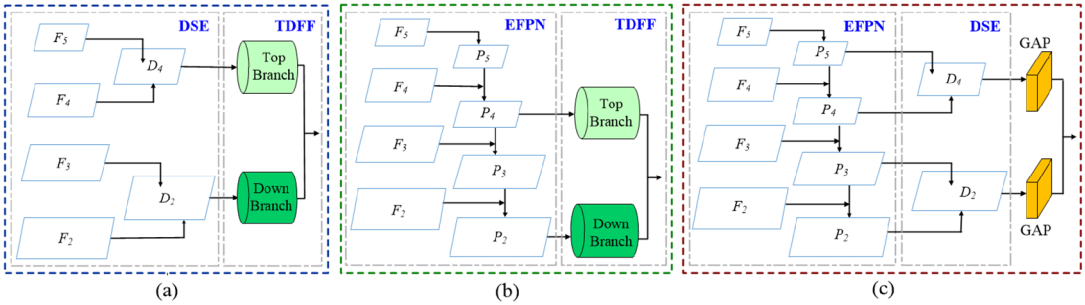


Fig. 12. Concise illustration of various architectures. (a) Without EFPN. (b) Without DSE. (c) Without TDFF.

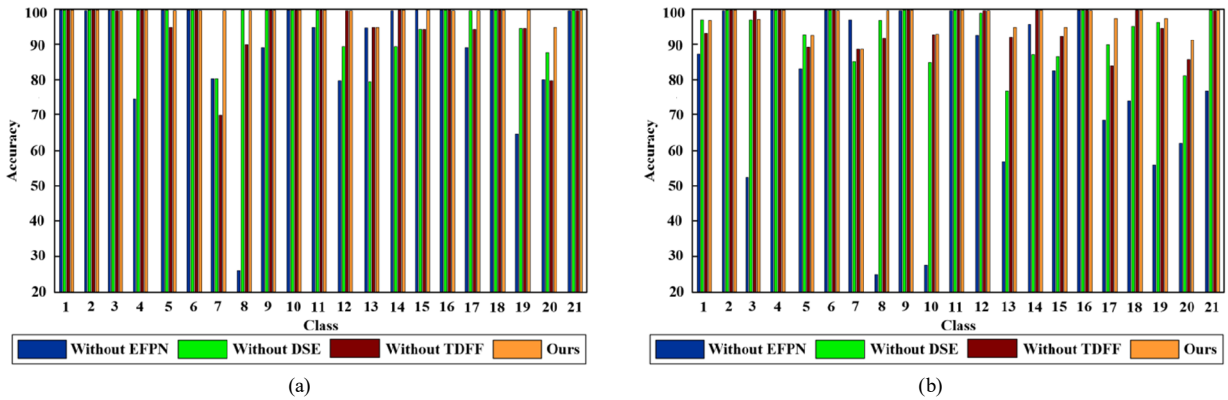
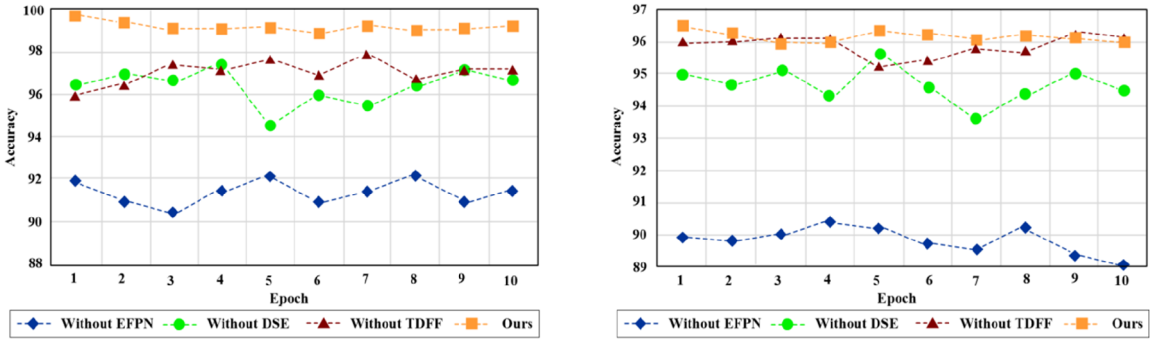


Fig. 13. Per-class classification performance of different architectures under the training ratios of 80% and 50% on UCM dataset. (a) Training ratio = 80%. (b) Training ratio = 50%.



(a) (b)

Fig. 14. Comparison of accuracy achieved after convergence of different architectures under the training ratios of 80% and 50% on UCM dataset. (a) Training ratio = 80%. (b) Training ratio = 50%.

TABLE VIII
EVALUATION ON THE EFFECTS OF UPSAMPLING STRATEGIES ON UCM DATASET (80% TRAINING RATION)

Mode	Top-Down Pathway	Deep Semantic Embedding	Overall Accuracy (%)
1	Nearest	Nearest	96.35 ± 0.67
2	Nearest	Bilinear	96.83 ± 0.70
3	Nearest	Deconv	97.02 ± 0.52
4	Bilinear	Nearest	97.88 ± 0.36
5	Bilinear	Bilinear	98.19 ± 0.61
6	Bilinear	Deconv	98.33 ± 0.36
7	Deconv	Nearest	98.57 ± 0.26
8	Deconv	Bilinear	98.83 ± 0.17
9	Deconv	Deconv	99.14 ± 0.22

D. *Different Upsampling Strategies*

In our framework, we propose to use the deconvolution technique (referred to as Deconv) instead of the commonly used upsampling strategies, such as nearest neighbor (referred to as Nearest) and bilinear interpolation (referred to as Bilinear). To verify this improvement, we do the experiments correspondingly. Specifically, we change the up-sampling methods in either the top-down pathway or deep semantic embedding module. For simplicity, here we merely demonstrate the experimental results using the UCM dataset with the 80% training ratio.

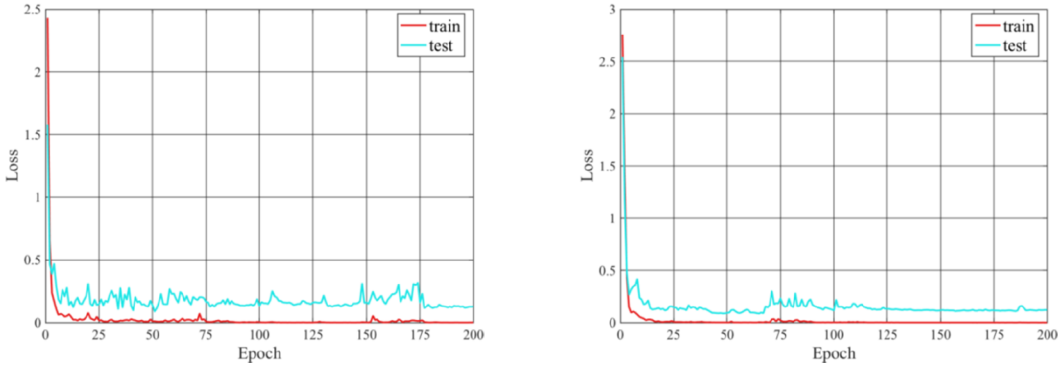
Table VIII reveals the details of various combinations of upsampling techniques used in the top-down pathway and deep semantic embedding module. There are totally nine different modes, and for instance, the first mode means that both of the top-down pathway and the deep semantic embedding module utilize the nearest neighbor method for upsampling.

We first take Modes 1, 4 and 7 for example to illustrate the importance of upsampling strategies in the top-down pathway. By comparing Modes 1, 4 and 7, in which all the deep semantic embedding modules use 'Nearest' for upsampling, while the upsampling schemes are set to 'Nearest', 'Bilinear', and 'Deconv' in the top-down pathways, we can find that the overall accuracy has changed from 96.35%, 97.88% to 98.57%.

Second, we take Modes 1, 2 and 3 for instance to show the importance of upsampling strategies in the deep semantic embedding module. In these three modes, the top-down pathways adopt 'Nearest' for upsampling, while the deep semantic embedding modules use 'Nearest', 'Bilinear', and 'Deconv' respectively for upsampling. The increasing overall accuracy, changing from 96.35%, 96.83% to 97.02%, illustrate the effectiveness of our proposed upsamping scheme. The similar conclusion can be also drawn by comparing the Modes 4, 5 and 6.

Third, by comparing Modes 7, 8 and 9, in which all the top-down pathways use 'Deconv' for upsampling, while the upsampling schemes are set to 'Nearest', 'Bilinear', and 'Deconv' in the deep semantic embedding module, we can see that our final method, i.e., Mode 9, which uses 'Deconv' in both the top-down pathway and deep semantic embedding module achieves the highest overall accuracy, i.e., 99.14%.

In addition, in the light of the better performance of Modes 7, 8 and 9, we further select them for comparison to illustrate the superiority of our 'Deconv' upsampling technique by the loss curve. Figs. 15, 16, and 17 display the loss curves of different



(a) (b)

Fig. 15. The loss curves of Mode 7 (Deconv + Nearest). (a) Training ratio = 80%. (b) Training ratio = 50%.

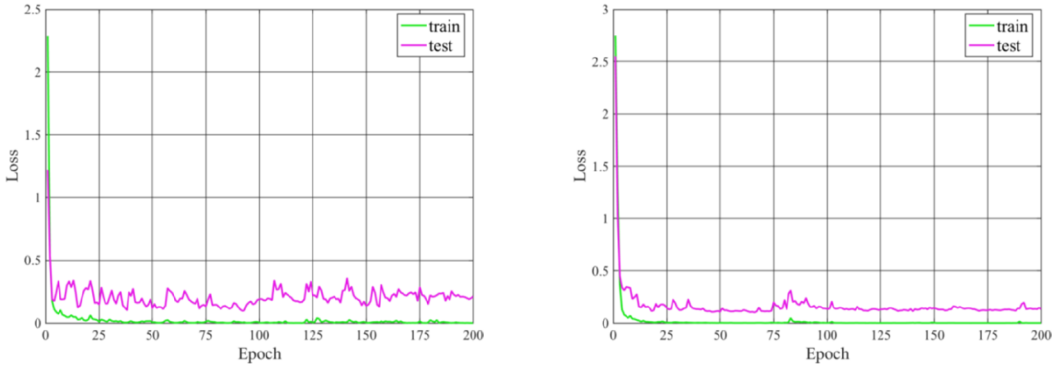


Fig. 16. The loss curves of Mode 8 (Deconv + Bilinear). (a) Training ratio = 80%. (b) Training ratio = 50%.

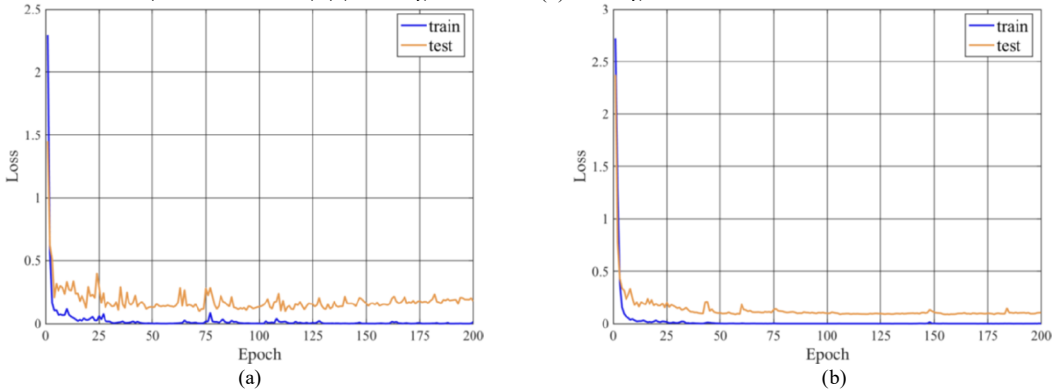


Fig. 17. The loss curves of Mode 9 (Deconv + Deconv). (a) Training ratio = 80%. (b) Training ratio = 50%.

modes under various training-testing ratios. The dataset is the UCM dataset. From these figures, we note that the losses of our proposed method (Mode 9) for both training-testing ratios converge faster than Modes 7 and 8. Also, the values of losses of our proposed method are much lower than those of Modes 7 and 8. In summary, these suggest that our upsampling scheme is beneficial to the whole architecture.

E. Benefits of Atrous Convolution

We further investigate the contribution of our atrous convolution. The study of different numbers and modes in atrous residual units is shown in Fig. 18, and the corresponding classification results are listed in Table IX. It can be seen that Shallow-Atrous Unit achieves the worst results. The performance of Parallel-Atrous Unit is a bit better, but the increase of the number of channels may lead to the increase of learning parameters. The other two units, Deep-Atrous and Deeper-Atrous obtain suboptimal results. Our Standard-Atrous Unit wins the competition on the RS classification task.

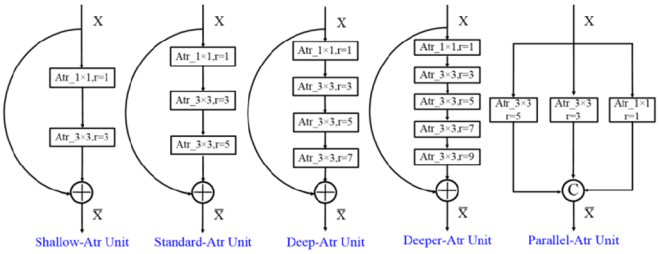


Fig. 18. The ablation study of different numbers and modes in atrous residual unit.

TABLE IX
OVERALL ACCURACY (%) OF DIFFERENT ARRANGES OF ATROUS CONVOLUTION ON UCM DATASET

Scheme	Architecture	Overall Accuracy	
		80% Training Ratio	50% Training Ratio
1	Shallow-Atrous	95.23 ± 0.53	92.21 ± 0.38
2	Standard-Atrous(ours)	99.14 ± 0.22	96.19 ± 0.13
3	Deep-Atrous	99.07 ± 0.27	96.15 ± 0.14
4	Deeper-Atrous	98.29 ± 0.54	95.34 ± 0.46
5	Parallel-Atrous	97.32 ± 1.14	94.61 ± 0.57

V. Conclusion

In this paper, we first detailed the challenges for the task of scene classification in remote sensing images. In order to overcome these problems, we constructed a unified deep learning framework, called EFPN-DSE-TDFF, in which three main contributions were made. In consideration of different rich characteristics of different level features, we designed an enhanced feature pyramid network to extract multi-scale multi-level feature maps and initially strengthen semantics. Besides, a deep semantic embedding module has been introduced to map the semantics of higher-level but coarser-resolution features into lower-level but finer-resolution ones so as to learn more reliable features. A two-branch deep feature fusion module is also employed for processing and aggregating the features at different levels together. We have compared the proposed method with many representative remote sensing scene classification approaches on several well-known RS scene datasets.

REFERENCES

[1] Y. Yu and F. Liu, "A two-stream deep fusion framework for high-resolution aerial scene classification," *Computational Intelligence and Neuroscience*, vol. 2018, article id. 8639367, Jan. 2018.

[2] X. Wang, X. Xiong, and C. Ning, "Multi-label remote sensing scene classification using multi-bag integration," *IEEE Access*, vol. 7, pp. 120399-120410, Aug. 2019.

[3] Ning C, W. Liu, G. Zhang, J. Yin, and X. Ji, "Enhanced synthetic aperture radar automatic target recognition method based on novel features," *Applied Optics*, vol. 55, no. 31, pp. 8893-8904, Nov. 2016.

[4] X. Bian, C. Chen, L. Tian, and Q. Du, "Fusing local and global features for high-resolution scene classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 6, pp. 2889-2901, Jun. 2017.

[5] Stumpf and N. Kerle, "Object-oriented mapping of landslides using Random Forests," *Remote Sensing of Environment*, vol. 115, no. 10, pp. 2564-2577, Oct. 2011.

[6] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *International Journal of Computer Vision*, vol. 81, no. 1, pp. 2-23, Jan. 2009.

[7] Cheng, P. Zhou, J. Han, L. Guo, and J. Han, "Auto-encoder-based shared mid-level visual dictionary learning for scene classification using very high resolution remote sensing images," *IET Computer Vision*, vol. 9, no. 5, pp. 639-647, Sept. 2015.

[8] Y. Yang and S. Newsam, "Comparing sift descriptors and gabor texture features for classification of remote sensed imagery," *In Proceedings of IEEE International Conference on Image Processing*, Oct. 2008, pp. 1852-1855.

[9] J. Ren, X. Jiang, and J. Yuan, "Learning LBP structure by maximizing the conditional mutual information," *Pattern Recognition*, vol. 48, no. 10, pp. 3180-3190, Oct. 2015.

[10] B. Luo, S. Jiang, and L. Zhang, "Indexing of remote sensing images with different resolutions by multiple features," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 6, no. 4, pp. 1899-1912, Apr. 2013.

[11] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," *In Proceedings of SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Nov. 2010, pp. 270-279.

[12] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3360-3367.

[13] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Oct. 2006, pp. 2169-2178.

[14] Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," *In Proceedings of European Conference on Computer Vision*, 2010, pp. 143-156.

[15] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834-848, Apr. 2017.

[16] P. S. Luus, B. P. Salmon, F. van den Bergh, and B. T. Maharaj, "Multiview deep learning for land-use classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 12, pp. 2448-2452, Oct. 2015.

[17] X. Zheng, Y. Yuan, and X. Lu, "A deep scene representation for aerial scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 7, pp. 4799-4809, Feb. 2019.

[18] Cheng, J. Han, P. Zhou, and L. Guo, "Multi-class geospatial object detection and geographic image classification based on collection of part detectors," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 98, pp. 119-132, Dec. 2014.

[19] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431-3440.

[20] Z. Yu, C. Feng, M. Y. Liu, and S. Ramalingam, "Casenet: Deep category-aware semantic edge detection," *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5964-5973.

[21] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "CNN-RNN: A unified framework for multi-label image classification," *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2285-2294.

[22] Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *In Advances in Neural Information Processing Systems*, 2012, pp. 1097-1105.

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

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1-9.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770-778.

[26] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2094-2107, Jun. 2014.

[27] W. Zhang, P. Tang, and L. Zhao, "Remote sensing image scene classification using cnn-capsnet," *Remote Sensing*, vol. 11, no. 5, pp. 494, Feb. 2019.

[28] S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, and X. Lu, "Aid: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3965-3981, Apr. 2017.

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

[30] X. Wang, X. Xiong, C. Ning, A. Shi, and G. Lv, "Integration of heterogeneous features for remote sensing scene classification," *Journal of Applied Remote Sensing*, vol. 12, no. 1, 015023, Mar. 2018.

[31] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117-2125.

[32] Y. Sa, "Improved bilinear interpolation method for image fast processing," *In Proceedings of IEEE Conference on Intelligent Computation Technology and Automation*, Oct. 2014, pp. 308-312.

[33] L. Xu, J. S. Ren, C. Liu, and J. Jia, "Deep convolutional neural network for image deconvolution," *In Advances in Neural Information Processing Systems*, 2014, pp. 1790-1798.

[34] X. Wang, S. Shen, C. Ning, F. Huang, and H. Gao, "Multi-class remote sensing object recognition based on discriminative sparse representation," *Applied optics*, vol. 55, no. 6, pp. 1381-1394, Feb. 2016.

[35] Wang, B. Fan, S. Xiang, and C. Pan, "Aggregating rich hierarchical features for scene classification in remote sensing imagery," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 9, pp. 4104-4115, May 2017.

[36] E. Othman, Y. Bazi, N. Alajlan, H. Alhichri, and F. Melgani, "Using convolutional features and a sparse autoencoder for land-use scene classification," *International Journal of Remote Sensing*, vol. 37, no. 10, pp. 2149-2167, Apr. 2016.

[37] Ghiasi and C. C. Fowlkes, "Laplacian pyramid reconstruction and refinement for semantic segmentation," *In Proceedings of European Conference on Computer Vision*, Sept. 2016, pp. 519-534.

[38] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2018, 8759-8768.

[39] Y. Chen, H. Fang, B. Xu B, Z. Yan, and Y. Kalantidis, "Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution," *arXiv preprint arXiv:1904.05049*, 2019.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *In Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 770-778.

[41] Y. Hua, L. Mou, and X. X. Zhu, "Lahnet: A convolutional neural network fusing low-and high-level features for aerial scene classification," *In Proceedings of IEEE International Geoscience and Remote Sensing Symposium*, Jul. 2018, pp. 4728-4731.

[42] L. C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, Dec. 2017.

[43] J. Ji, T. Zhang, L. Jiang, W. Zhong, and H. Xiong, "Combining Multilevel Features for Remote Sensing Image Scene Classification With Attention Model," *IEEE Geoscience and Remote Sensing Letters*, DOI: 10.1109/LGRS.2019.2949253, Nov. 2019.

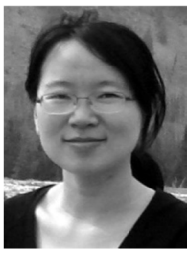
[44] S. Azimi, E. Vig, F. Kurz, and P. Reinartz, "Segment-and-count: Vehicle Counting in Aerial Imagery using Atrous Convolutional Neural Networks," *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, pp. 19-23, Sept. 2018.

[45] Q. Hou, M. M. Cheng, X. Hu, a. Borji, Z. Tu, and P. Torr, "Deeply supervised salient object detection with short connections," *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3203-3212.

[46] Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: Benchmark and state of the art," *Proceedings of the IEEE*, vol. 105, no. 10, pp. 1865-1883, Apr. 2017.

[47] E. Li, J. Xia, P. Du, C. Lin, and A. Samat, "Integrating multilayer features of convolutional neural networks for remote sensing scene classification," *IEEE*

Transactions on Geoscience and Remote Sensing, vol. 55, no. 10, pp. 5653-5665, Jun. 2017.
[48] 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 Sensing*, vol. 7, no. 11, pp. 14680-14707, Nov. 2015.
[49] 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 Sensing*, vol. 9, no. 8, 848, Aug. 2017.
[50] M. Cheriyadat, "Unsupervised feature learning for aerial scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 1, pp. 439-451, Mar. 2013.
[51] 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 Journal of Photogrammetry and Remote Sensing*, vol. 138, pp. 74-85, Apr. 2018.
[52] S. Chaib, H. Liu, Y. Gu, and H. Yao, "Deep feature fusion for VHR remote sensing scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 8, pp. 4775-4784, May 2017.
[53] N. Liu, X. Lu, L. Wan, H. Huo, and T. Fang, "Improving the separability of deep features with discriminative convolution filters for RSI classification," *ISPRS International Journal of Geo-Information*, vol. 7, no. 3, 95, Mar. 2018.



Xin Wang was born in Fuyang, Anhui, China in 1981. She received the B.S. and M.S. degrees in signal and information processing from Anhui University, Hefei, in 2006 and the Ph.D. degree in computer application technology from Nanjing University of Science and Technology, Nanjing, in 2010.

From 2010 to 2013, she was an Assistant Professor with the College of Computer and Information in Hohai University. Since 2014, she has been an Associate Professor with the College of Computer and Information, Hohai University. She is the author of three books, more than 80 articles, and more than 50 patents. Her research interests include image processing and analysis, computer vision, pattern recognition, and computer vision.

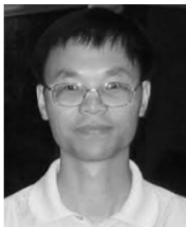
Dr. Wang was a recipient of the Science and Technology Progress Awards in 2014, 2015, 2017 and 2020.

Shiyi Wang was born in Qidong, Jiangsu, China in 1996. He received the B.S. degrees in communication engineering from Hohai University, Jiangsu, in 2018. He is currently working toward the M.S. degree in Signal and Information Processing from the College of Computer and Information, Hohai University, Nanjing, China.

His research interests include deep learning, remote sensing image processing and analysis, computer vision, and pattern recognition.

Chen Ning was born in Fuyang, Anhui, China in 1978. He received the B.S. degree in communication engineering from Anhui University, Hefei, in 2000 and the M.S. degree in signal and information processing from University of Science and Technology of China, Hefei, in 2003.

Since 2010, he was an Assistant Professor with the School of Physics and Technology, Nanjing Normal University. He is the author of 20 articles, and more than 10 patents. His research interests include image processing and analysis, computer vision, pattern recognition, and deep learning.



Huiyu Zhou received the B.Eng. degree in radio technology from the Huazhong University of Science and Technology, Wuhan, China, the M.S. degree in biomedical engineering from the University of Dundee, Dundee, U.K., and the Doctor of Philosophy degree in computer vision from Heriot- Watt University, Edinburgh, U.K.

He is currently a Reader with the Department of Informatics, University of Leicester, Leicester, U.K. He has taken part in the consortiums of a number of research projects in medical image processing, computer vision, intelligent systems, and data mining.