


# A Deep Neural Network Combined CNN and GCN for Remote Sensing Scene Classification

Jiali Liang , Yufan Deng, and Dan Zeng, *Member, IEEE*

**Abstract**—Learning powerful discriminative features is the key for remote sensing scene classification. Most existing approaches based on convolutional neural network (CNN) have achieved great results. However, they mainly focus on global-based visual features while ignoring object-based location features, which is important for large-scale scene classification. There are a large number of scene-related ground objects in remote sensing images, as well as Graph convolutional network (GCN) has the potential to capture the dependencies among objects. This article introduces a novel two-stream architecture that combines global-based visual features and object-based location features, so as to improve the feature representation capability. First, we extract appearance visual features from whole scene image based on CNN. Second, we detect ground objects and construct a graph to learn the spatial location features based on GCN. As a result, the network can jointly capture appearance visual information and spatial location information. To the best of authors' knowledge, we are the first to investigate the dependencies among objects in remote sensing scene classification task. Extensive experiments on two datasets show that our framework improves the discriminative ability of features and achieves competitive accuracy against other state-of-the-art approaches.

**Index Terms**—Convolutional neural networks (CNNs), deep learning, feature representation, graph convolutional network (GCN), remote sensing scene classification.

## I. INTRODUCTION

REMOTE sensing scene classification aims to automatically classify remote sensing images into specific categories based on semantic content, which has attracted great attention in recent years due to the wide range of applications, such as natural disaster monitoring, land cover analysis, and urban planning [1]–[4]. Up to now, there are a variety of approaches have been proposed for remote sensing scene classification. According to the form of feature extractor, they can be divided into three groups, i.e., low-level, mid-level, and high-level feature descriptors.

In the early days, most of approaches are based on low-level features, which extract the color or texture like histograms of

oriented gradients [5], local binary patterns (LBPs) [6], and scale invariant feature transform [7]. However, these approaches based on hand-crafted features may not well represent semantic information. A major shortcoming of these hand-crafted features is that they demand complex engineering skills that rely on expert experience.

In contrast to low-level features, the mid-level features are the coding and fusion of local visual features, such as bag-of-visual-words (BoVWs) [8], spatial pyramid matching [9], vector of locally aggregated descriptors [10], and Fisher vector [11]. The most commonly used approach is BoVW, which encodes the local features of image patches into visual dictionaries by  $k$ -means clustering [12], [13]. Although these mid-level features are highly efficient, they ignore the spatial distribution information of remote sensing scene image leading to poor representation capability. As a result, the abovementioned approaches only perform well on some scenes with regular texture or spatial arrangements, but have limited performance in dealing with complex and challenging scene images.

Due to the impressive feature representation power of convolutional neural networks (CNNs), it has been widely applied to image classification [14], object detection [15], semantic segmentation [16]. Meanwhile, the rapid development of deep learning technologies accelerates the progress in remote sensing scene classification. Many works employ networks pretrained on the ImageNet dataset [17] as feature extractors for scene classification, such as visual geometry group net (VGGNet) [18], AlexNet [19], GoogLeNet [20]. And not only that, there are many novel networks are designed for remote sensing scene classification [21]–[31]. Wang *et al.* [25] employed the rich hierarchical features of a CNN to form a discriminative image representation for scene classification, which incorporates from low-level, middle-level, and high-level features simultaneously. Liu *et al.* [28] introduced a Siamese CNN model that combines verification and identification models to boost the performance. To allow the input images to be of arbitrary sizes, Xie *et al.* [29] proposed a scale-free CNN to preserve key information in high spatial resolution images. A multiscale CNN [30] is proposed to merge the feature maps of different layers based on feature maps selection algorithm and region covariance descriptor. Several recent works have paid attention to local semantic feature learning. Wang *et al.* [31] utilized attention mechanism to adaptively select a series of critical parts of images, and then to generate powerful features. There is also a research regarded scene classification as a multiple-instance learning (MIL) problem. Bi *et al.* [32] proposed an MIL framework to

Manuscript received May 8, 2020; revised June 29, 2020; accepted July 7, 2020. Date of publication July 27, 2020; date of current version August 11, 2020. This work was supported by the National Natural Science Foundation of China under Grant 61572307. (Corresponding author: Dan Zeng.)

Jiali Liang and Dan Zeng are with the Key Laboratory of Specialty Fiber Optics and Optical Access Networks, Joint International Research Laboratory of Specialty Fiber Optics and Advanced Communication, Shanghai Institute of Advanced Communication and Data Science, Shanghai University, Shanghai 200444, China (e-mail: liangjl@shu.edu.cn; dzeng@shu.edu.cn).

Yufan Deng is with the Shenzhen Middle School, Shenzhen 518001, China (e-mail: 540106758@qq.com).

Digital Object Identifier 10.1109/JSTARS.2020.3011333

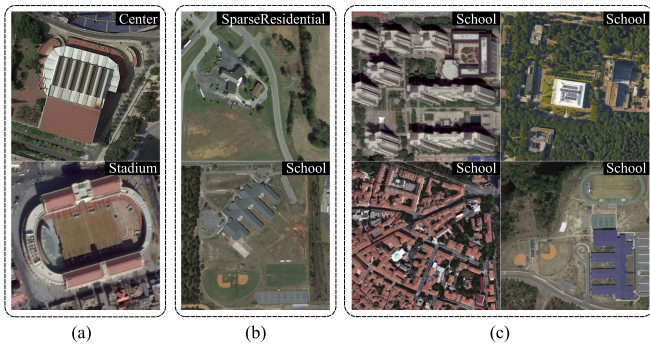


Fig. 1. Examples of AID datasets illustrate the challenge of remote sensing scene classification: large intraclass variance and small interclass variance. (a) and (b) Different scenes show high visual similarity. (c) Diversity within the same scene varies largely.

highlight the local semantics relevant to the scene label. These methods effectively discarded the useless information, thereby enhancing the capability of local semantic representation.

Remote sensing scene classification is a challenging task due to the diversity, multiresolution and complex spatial distributions of remote sensing image data [33], which result in large intraclass variance and small interclass variance. As shown in Fig. 1, both center and stadium scenes show high appearance similarity and the diversity within the school scene varies largely. The abovementioned methods are mainly to learn the global representation of images and neglect the local information. Although these existing CNN-based methods have achieved a great performance to some extent, some complex scene classes are still easily misclassified since only visual information is utilized. The most of the previous methods [21]–[30] only learn the global features representation of images, which may neglect the local details. Even though there are several works [31]–[32] attempt to focus on the critical local image patches and discard the useless information, they still only utilize the visual information. All these methods ignore the spatial location and distribution information. To overcome the drawback, we consider learning the spatial information by exploring the dependencies among ground objects in scene images. Our proposed method takes advantage of appearance visual information as well as spatial location and distribution cues to make better predictions for remote sensing scene classification.

Recently, a novel model called graph neural network [34] has drawn wide attention due to its ability to deal with the data in graph domain. The graph convolutional network (GCN) [35] is an extension of the CNN, which aims to operate convolutional on non-Euclidean space. In contrast with the classical CNN method, GCN would be more effective for learning the feature representation of the graph-structured data. GCN has been widely used in the tasks with rich relational structure, such as recommender systems [36], relation detection [37], multi-label image recognition [38], and 3-D point cloud classification [39]. GCN aggregates information from the neighbors of each node [40], which can be utilized to explore the dependencies among objects in remote sensing scene images. There are a large number of scene-related ground objects in remote sensing

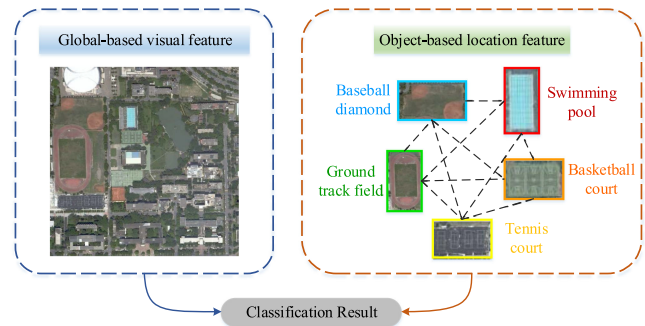


Fig. 2. Overview of the proposed approach. We propose to classify scene categories by jointly learning feature from whole image and local graph structure.

images, which represent different scene semantic categories by different combinations and spatial arrangements. Therefore, we are motivated to employ the effective architecture GCN to model the dependencies among objects and explore the potential spatial location and distribution information.

To further improve the performance of the remote sensing scene classification, we propose a novel two-stream architecture, which can jointly learn both the global-based visual features and the object-based location features. Motivated by the powerful feature representation capability of CNN and the potential of relation inference of GCN, our proposed model combined CNN and GCN as shown in Fig. 2. First, we extract appearance visual features from whole scene image based on CNN. Second, we detect ground objects and construct a graph to learn the spatial location features based on GCN. Unlike the classical CNN-based methods, our proposed model can take advantage of spatial location and distribution cues to make better predictions on scene category. The main contribution of this article is jointly learning appearance visual information and spatial location information for remote sensing scene classification. In summary, the major contributions of this article are as follows.

- 1) To further improve classification accuracy, we propose a novel two-stream architecture, which combined CNN and GCN to learn both the global-based visual features and the object-based location features.
- 2) Our network allows the input images to be of arbitrary sizes by using global average pooling layer and region of interest (ROI) pooling layer, which can preserve detailed information in high spatial resolution images as much as possible.
- 3) By combining the CNN and GCN, our proposed method improves the discriminative ability of features and achieves competitive accuracy against other state-of-the-art approaches.

The rest of this article is organized as follows. Section II introduces our proposed architecture combined CNN and GCN in detail. The experimental results and analysis are presented in Section III. Finally, Section IV concludes this article.

## II. PROPOSED METHOD

The overall architecture of our proposed model is illustrated in Fig. 3, which composes of three parts: CNN-based branch,

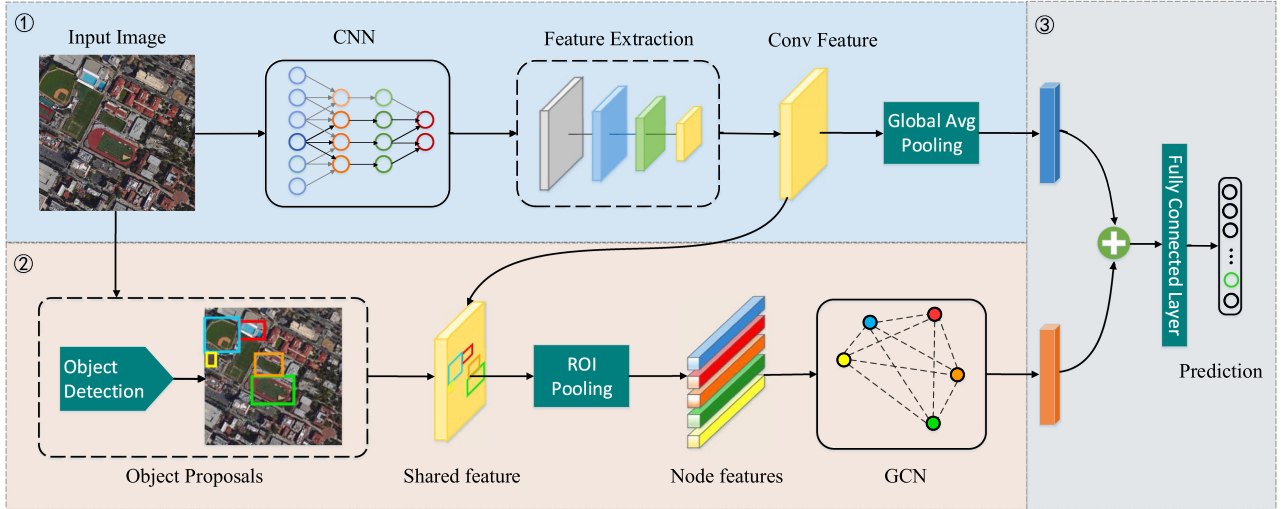


Fig. 3. Architecture of our proposed method. The model consists of: 1) CNN-based branch: the visual appearance feature is modeled by CNN. 2) GCN-based branch: we detect object proposals and use ROI pooling layer to form the node features, which are sent into GCN to capture the spatial location information. 3) Feature fusion: the feature descriptor extracted from the two branches are integrated and delivered to the classifier.

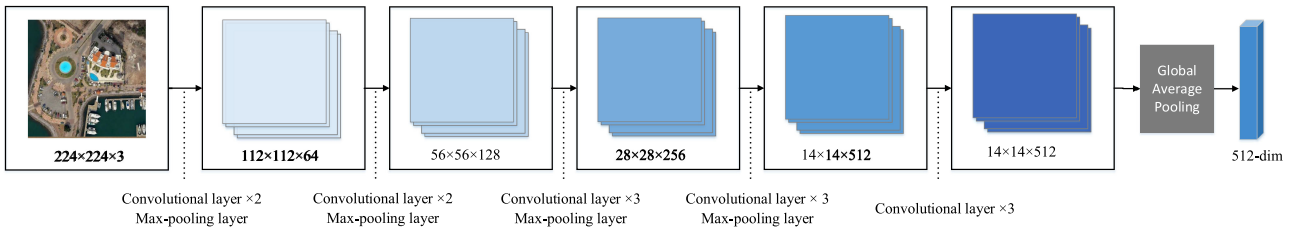


Fig. 4. Architecture of CNN-based branch, which is modified from VGG16.

GCN-based branch, and feature fusion. For the CNN-based branch, the input image goes forward through the VGG16 backbone network to extract the global feature. For the GCN branch, we detect ground objects by Faster R-CNN [41] algorithm in each scene image, and the object proposals are fed into ROI Pooling layer [42] to extract the regional feature representation. Then, we construct a graph to model the dependencies among objects. Finally, the fusion of features extracted from the two branches is delivered to the classifier to make the final prediction. By combining the CNN and GCN, our framework improves the discriminative ability of features. The three parts of our model will be further elaborated, respectively.

#### A. CNN-Based Branch

It is critical for remote sensing scene classification to learn powerful appearance visual feature, especially there are some categories of images do not contain objects, such as beach, desert, and forest. For these images, we can only utilize the appearance visual information. In general, remote sensing scene classification can use any classification network as the backbone. As shown in Fig. 4, our CNN-based branch network is modified from the simple and effective model VGG16 [18]. The convolutional layers are assembled within five convolution blocks each ending with a max-pooling operation. The size of feature map

output of convolution block is defined as  $h \times w \times d$ , where  $h$  and  $w$  are spatial dimensions and  $d$  is the number of channels.

To be specific, two modifications have been made in our CNN-based branch network. First, we remove the max-pooling layer in last convolution blocks. Max Pooling is a down-sample operation that chooses the maximum element from the region of the feature map covered by the filter. As the spatial size of the features reduced, some detailed information is missing. In consideration of the global convolutional feature map will be shared with GCN-based feature to extract the local object feature, we did not reduce the feature map size too much. Generally, the size of feature map performed by standard VGG16 network is 1/32 of the input image. It can be seen that our network makes the feature map size 1/16 that of the input image, which effectively preserves detailed information in high spatial resolution scene images. Second, we remove the last three fully connected layers and add a global average pooling layer [43]. Global average pooling layers are used to reduce the spatial dimensions by computing the mean of the height and width dimensions of the feature map. The equations of global average pooling can be given as follows:

$$g_{c,i} = \frac{\sum_{w=1}^W \sum_{h=1}^H x_{w,h,c,i}}{W \times H} \quad (1)$$

where  $x$  is the input feature map,  $w$ ,  $h$ , and  $c$  are the width, height, and the number of channels, respectively. It can be seen

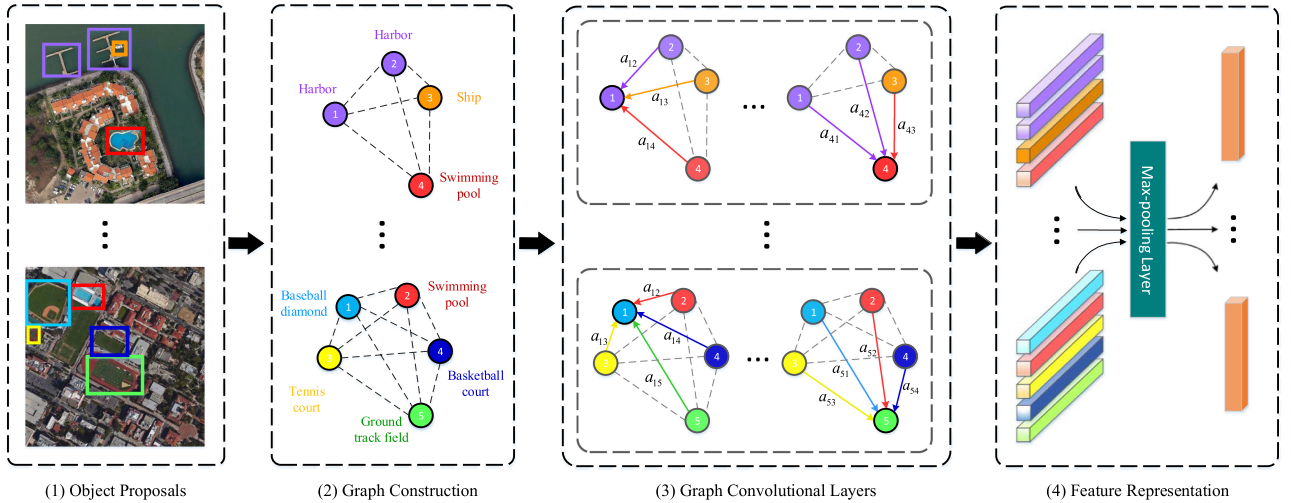


Fig. 5. Architecture of GCN-based branch. 1) A set of objects are extracted in the scene image. 2) Construct a graph with node and edge. 3) Graph convolution network is applied to update each node from the neighbor nodes. 4) The graph-based features are generated by max-pooling layer.

that the global average pooling layer converts a 3-D tensor with the size of  $14 \times 14 \times 512$  to a 1-D feature vector with the size of 512-dim. Our network allows the input remote sensing scenes to be of arbitrary size since the global average pooling layer, which can preserve detailed information in remote sensing images as much as possible. In addition, the fully connected layers have a large number of weight parameters, which cause high computational cost. By replacing fully connected layer and by global average pooling layer, our model eliminates a large number of parameters.

### B. GCN-Based Branch

In our work, we view remote sensing scene classification task as a graph classification problem. GCN is a novel neural network that learns feature by gradually aggregating information in the neighborhood. Fig. 5 shows the architecture of our GCN-based branch. The process is described as follows. First, we detect the ground objects in scene images through the object detection algorithm. Second, the scene images can be modeled by graphs, the nodes of which stand for the detected objects and the edges of which represent distance between the nodes. Then, we use graph convolutional operation to capture the dependencies among objects. Finally, the object-based location features are generated by the pooling layer.

1) *Object Proposals*: In remote sensing images, a scene is a combination of objects and backgrounds. There are scene-related local objects in remote sensing images. For instance, in school scene image, there are usually ground objects such as baseball diamond, tennis court, basketball court, and track field. And in the port scene image, there are usually ground objects such as ship and harbor. These objects form different scene semantic categories in different combinations and spatial arrangements. The prerequisite of learning spatial dependency is accurate object detection. Although object detection has achieved great successes in natural images, it is hard to apply

to remote sensing image due to the large scale variations and arbitrary orientations of objects. Therefore, the object detectors learned from natural images are not suitable for aerial images.

In this work, our object detection model is trained on dataset for object detection in aerial images (DOTA) dataset [44], which can recognize 15 object categories, including plane, ship, storage tank, baseball diamond, tennis court, basketball court, ground track field, harbor, bridge, large vehicle, small vehicle, helicopter, roundabout, soccer ball field, and swimming pool. Due to the high accuracy of faster R-CNN [41], we adopt it to detect the ground objects in remote sensing scene images. When the training is finished, we employ the model to detect objects on aerial image dataset (AID) and NWPU-RESISC45 datasets for graph construction. Each object proposal is associated with a spatial region  $\{x_i, y_i, w_i, h_i\}$ , where  $x_i$  and  $y_i$  are the coordinates of the top-left corner and  $w_i$  and  $h_i$  are the width and height of bounding box, respectively. In addition to location information, each object proposal has a confidence score, which stands for the probabilities of the coordinates of objects. In this article, we set the threshold as 0.3. For scores over our chosen threshold, we choose the corresponding object proposals detected in the image as our nodes in the graph.

2) *Graph Construction*: For the scene images containing objects, we build a graph based on the object location. The graph structure consists of nodes and edges, where the nodes of which stand for the detected objects and the edges of which represent distance between the nodes. Therefore, we extract the initial node features and compute the adjacency matrix.

Each object proposal  $i$  is associated with a spatial region, and then we use ROI pooling layer [42] to generate the node feature representation  $n_i$ . The ROI pooling layer takes two inputs: a global feature map and spatial region location. The operation of ROI Pooling is as follows:

$$n_i = R(f_i, x_i, y_i, w_i, h_i) \quad (2)$$

where  $f_i$  is the global feature shared from the CNN-based branch. Each object proposal location is defined by a four-tuple  $\{x_i, y_i, w_i, h_i\}$ . The ROI pooling layer outputs feature  $n_i$  for each proposal with fixed size and the kernel size as  $3 \times 3$  in our work. Finally, the node feature is constructed with regional feature and location, i.e.,  $\{n_i, x_i, y_i, w_i, h_i\}$ .

GCN works by aggregating features between nodes based on the adjacency matrix [45]. In this article, we build this adjacency matrix through Euclidean distance. The dependencies among objects are highly correlated to the distance among them, we define the adjacency matrix according to the spatial coordinates. We first calculate the center coordinates of object proposal as follows:

$$c_i = \left( x_i + \frac{w_i}{2}, y_i + \frac{h_i}{2} \right). \quad (3)$$

Then, adjacency matrix is established by Euclidean distance and normalized with Gaussian kernel, which can be defined as

$$a_{i,j} = \begin{cases} e^{-\frac{\|c_i - c_j\|^2}{2\gamma^2}} & \text{if } \|c_i - c_j\|^2 > \tau \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where the parameter  $\gamma$  is a scale factor, where we set it to 1.6. Considering node is assumed to be connected to each other in our work, the adjacency matrix is symmetric. In this way, we embed the spatial information into the adjacency matrix. If two nodes  $n_i$  and  $n_j$  are so far away in the space that distance exceeds  $\tau$ , the corresponding  $a_{i,j}$  will be set to zero.

3) *GCN*: Since graph is a non-Euclidean structure data, we leverage the GCNs to learn the spatial dependencies among objects. A graph can be defined as  $G = (V, E)$ , where  $V = \{v_i \mid i \in \{1, \dots, N\}\}$  and  $E = \{e_{i,j} \mid \forall i, j \in \{1, \dots, N\}\}$  are the sets of nodes and edges, respectively. The initial feature descriptions of nodes  $V$  are extracted by ROI pooling and defined as  $X \in R^{n \times d}$ . Every node is assumed to be connected to each other and the adjacency matrix is defined as  $A \in R^{n \times n}$ . Here,  $n$  denotes the number of nodes and  $d$  denotes the feature dimension. In our GCN-based branch model, we stacked two graph convolutional layers, which take graph feature description  $X$  and the adjacency matrix  $A$  as inputs. Mathematically, we can represent the output feature of first layer of the graph convolution as

$$h^{(1)} = \sigma(AXW^{(0)}) \quad (5)$$

where  $A$  is the adjacency matrix,  $W^{(0)} \in R^{m \times d}$  is a weight matrix of trainable parameters at first layer, and  $\sigma$  is the activation function, which is implemented by rectified linear units (ReLU) in our model. By stacking layers, we can aggregate higher-order feature from neighbors as follows:

$$h^{(l+1)} = \sigma(\tilde{A}h^{(l)}W^{(l)}) \quad (6)$$

where  $l$  denotes the number of graph convolution layer and  $h^{(0)} = X$ .  $W^{(l)}$  is the weight matrix of trainable parameters at  $l$ th layer. GCN propagates messages on a graph structure and aggregates feature information from the neighbors of each node. Actually, GCN updates the hidden state of nodes by a weighted

---

**Algorithm 1:** The Proposed Method for Remote Sensing Scene Classification.

---

**Input:**

Input image; number of iterations  $T$ ; learning rate  $\eta$ ;  
number of graph convolutional  $l$ ; parameter  $\gamma$ .

**Output:**

Predict label for input image.

**Algorithm:**

- 1: Fine-tune VGG16 model on training datasets.
  - 2: Forward inference and generate global-based feature map.
  - 3: Generate object proposals  $\{x_i, y_i, w_i, h_i\}_{i=1}^N$ .
  - 4: Compute the adjacency matrices  $A$  according to (4).
  - 5: ROI Pooling extracts regional feature  $f_i$  and generates node features  $n_i$ .
  - 6: Construct the graph  $G$  for input image.
  - 7: **for**  $t = 1$  to  $T$  **do**
  - 8:   Calculate the outputs of the  $l$ th layer  $h^{(l)}$ .
  - 9:   Calculate the loss according to (9).
  - 10:   Update the weight matrices  $W$  using gradient descent.
  - 11: **end for**
  - 12: Calculate the network output according to (8) and conduct label prediction.
- 

sum of the features of their neighbors as follows:

$$\tilde{A}h^{(l)} = \sum_{j=1}^n A_{i,j}h_j^{(l)}. \quad (7)$$

Therefore, the complex dependencies among nodes can be modeled by gradually aggregating information in the neighborhood. Then, we employ simple global max-pooling strategy on graph features. GCN passes information among neighbor nodes and updates each node according to the predefined adjacency matrix [46], which allows us to effectively capture the spatial dependencies among objects.

### C. Feature Fusion

In order to enable our network to simultaneously obtain the appearance characteristics of the whole image and the spatial dependency between objects, we design a simple feature fusion block. The feature descriptor extracted from the two branches is integrated and delivered to the final classifier. The operation of feature fusion can be described as follows:

$$z_i = W^{(f)} \cdot (g_i \oplus \lambda h_i) \quad (8)$$

where  $g_i$  is the features of CNN-based branch and  $h_i$  is the features of GCN-based branch,  $W^{(f)}$  is the weight matrix of trainable parameters at final fully connected layer,  $\oplus$  represents element-wise addition, and  $\lambda$  is a parameter that controls the fusion ratio. We adjusted  $\lambda$  manually based on experience and set it to one. To fine-tune the VGG network on remote sensing datasets, we replace the final fully connected layer of the CNN model to the number of scene category. The fused feature is

fed into the softmax layer to obtain the probability that each image belonging to each class. During the training stage, the cross-entropy loss function is used to optimize the parameter values in our model, which is

$$L = - \sum_{i=1}^N y_i \log \frac{e^{z_i}}{\sum_{j=1}^N e^{z_j}} + (1 - y_i) \log \left( 1 - \frac{e^{z_i}}{\sum_{j=1}^N e^{z_j}} \right) \quad (9)$$

where  $y_i$  denotes the ground truth label of  $i$ th class, with 1 representing the image belonging to the corresponding class while 0 for not.  $N$  is the number of scene category and  $z_i$  is the outputted probability from the model. And during the testing stage, the feature descriptor is used to make the final prediction of scene category. The process of our proposed method is summarized in Algorithm 1. By combining the CNN and GCN, our proposed method effectively improves the discriminative ability of features.

### III. EXPERIMENT

In this section, we conduct exhaustive experiments on two challenging datasets and make comparison between the proposed method and state-of-the-art approaches, where two metrics including overall accuracy and confusion matrix are adopted.

#### A. Experimental Datasets

We choose two large-scale datasets AID [47] and NWPU-RESISC45 [48] to demonstrate the effectiveness of our method. In detail, AID dataset contains 10 000 images and 30 scene categories with  $600 \times 600$  pixels, where the number of images in each category ranges from 220 to 420. The spatial resolution ranges from 8 to 0.5 m/pixel. NWPU-RESISC45 dataset is the largest remote sensing scene dataset up to now, which contains 31 500 images and 45 scene categories with  $256 \times 256$  pixels. Each category contains 700 images with spatial resolution ranging from 30 to 0.2 m/pixel. These two datasets all contain rich scene categories, which can provide a large number of scene-related ground objects to learn the spatial information. The images of two datasets are collected from Google Earth imagery, which are captured from different remote imaging satellites at different times, so as to rich image variations and diversity. In particular, some confusing scene categories with high similarity like basketball courts and tennis courts are designed in NWPU-RESISC45, which intensifies the characteristic of small interclass variance.

#### B. Evaluation Protocol

For the task of remote sensing image scene classification, overall accuracy and confusion matrix are two common quantitative evaluation metrics. The overall accuracy is a holistic measure to show the classification performance on the whole dataset, and the confusion matrix is a more detailed table used for visualizing the performance of each category. To obtain reliable results, we randomly select the training samples and repeat it

TABLE I  
NUMERICAL RESULTS (%) OF OUR OBJECT DETECTION MODEL ON EACH CATEGORY OF THE DOTA TESTING SET

Category	AP	Category	AP
Plane	88.7	Baseball diamond	75.7
Basketball court	77.9	Bridge	51.7
Ground field track	73.8	Harbor	67.6
Helicopter	58.3	Large vehicle	71.3
Roundabout	59.2	Ship	78.8
Small vehicle	64.5	Soccer ball field	50.1
Storage tank	81.3	Swimming pool	69.2
Tennis court	89.5		

10 times. The mean and standard deviation of overall accuracies are also reported.

#### C. Implementation Details

We implement our proposed architecture with the PyTorch framework. All experiments are performed on Intel Core i7 2.93 GHz CPU with NVIDIA GeForce GTX 1080 GPU for acceleration. We use the pretrained VGG16 model on ImageNet dataset and fine-tune it on remote sensing scene dataset. We use stochastic gradient descent [49] optimizer with a momentum of 0.9 and weight decay of  $5e-4$ . The initial learning rate starts at 0.0001 and is divided by 10 when epoch reaches 20, 40, and 60. We train our model for 80 epochs with a mini-batch size of 4. We also adopt some effective augmentation schemes like shuffle the whole dataset to use the random order of the images during training. Moreover, all input images are augmented by randomly rotating  $90^\circ$ ,  $180^\circ$ ,  $270^\circ$ , flipping horizontally, and flipping vertically. As same as the previous works [23], we adopt two training ratios for each dataset for a fair comparison. For the AID dataset, we set the ratios of training set to 20% and 50%, and the rest 80% and 50% for testing. Similarly, for the NWPU-RESISC45 dataset, the training ratios are set to 10% and 20%, and the rest 90% and 80% for testing.

#### D. Object Detection Results

Our object detection model achieves 70.5% mAP on the DOTA testing set and the numerical results of each category are shown in Table I. Some object detection results of AID dataset and NWPU-RESISC45 dataset are presented in Figs. 6 and 7. It can be seen that a number of planes, baseball diamonds, and storage tanks are accurately detected in airport, baseball field, and storage tanks, respectively. Our detection algorithm still performs well even in the challenging scene classes, such as school, park, and viaduct. In school scene images, there are usually ground objects such as baseball diamond, tennis court, basketball court, and ground track field. In park scene images, there are usually ground objects such as bridge and ground track field. And in port scene image, there are usually ground objects such as ships and harbor. The different combinations and spatial arrangements represent specific scene

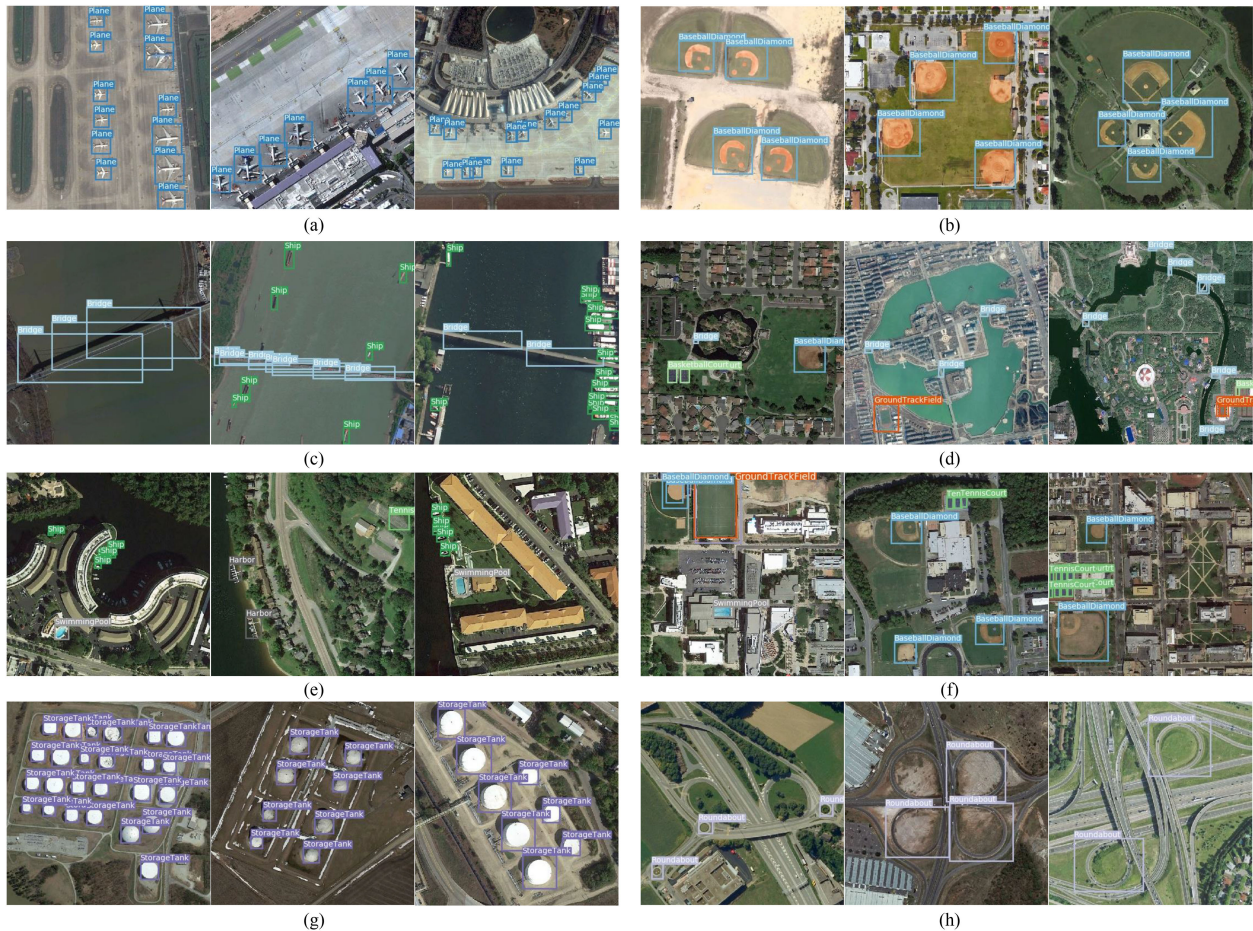


Fig. 6. Examples of object detection results on AID dataset. (a) Airport. (b) Baseball diamond. (c) Bridge. (d) Park. (e) Resort. (f) School. (g) Storage tanks. (h) Viaduct.

categories. Our method tries to explore the spatial dependencies based on the rich object instances contained in scene image.

### E. Parameter Analysis

Three parameters are tested to analyze how these parameters affect the classification result, including the kernel size of ROI pooling layer, the hidden dimension of graph convolution layer, and the scale factor of Gaussian kernel. We test on AID dataset and NWPU-RESISC45 dataset under the training ratio of 50% and 20%, respectively. Fig. 8 shows the influence on the classification accuracy with different parameter settings. We set the other parameters to the best value when evaluating each parameter.

1) *Kernel Size of ROI Pooling Layer*: We use ROI pooling layer to generate the regional feature representation of object proposal with fixed size. Four different kernel sizes are tested to analyze the effect on classification accuracy, including  $1 \times 1$ ,  $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$ . From Fig. 8(a), we can find that increasing the kernel size can improve classification accuracy. For the NWPU-RESISC45 dataset, the performance drops slightly when the kernel size set to  $5 \times 5$ . This can be explained that most

small objects contained in images are better suited for small kernel size. Combining the results of the two data sets, we set the kernel size of ROI pooling layer as  $3 \times 3$ .

2) *Scale Factor of Gaussian Kernel*: GCN works by aggregating features between nodes based on the adjacency matrix. The dependencies among objects are highly correlated to the distance among them, we define the adjacency matrix according to the spatial coordinates. We build this adjacency matrix through Euclidean distance and normalize it with Gaussian kernel in our method. Therefore, the scale factor  $\gamma$  is also defined as the adjustment parameter to control the Gaussian kernel. As the scale factor increases, the values in adjacency matrix gradually decrease. From Fig. 8(b), we can find that the highest classification accuracy is obtained when the scale factor of Gaussian kernel is set to 1.6.

3) *Distance Threshold of Adjacency Matrix*: We define the adjacency matrix according to the spatial coordinates in the GCN-based branch. If the distance of two nodes are so far away in the space that exceeds threshold  $\tau$ , the corresponding will be set to zero. Because nodes that are far apart almost not affect the feature learning of each other. The main aim is to reduce computation and speed up the running rate. From Fig. 8(c), we set the distance threshold as  $1e-5$ .



Fig. 7. Examples of object detection results on NWPU-RESISC45 dataset. (a) Airplane. (b) Baseball diamond. (c) Basketball court. (d) Bridge. (e) Ground track field. (f) Harbor. (g) Roundabout. (h) Storage tank.

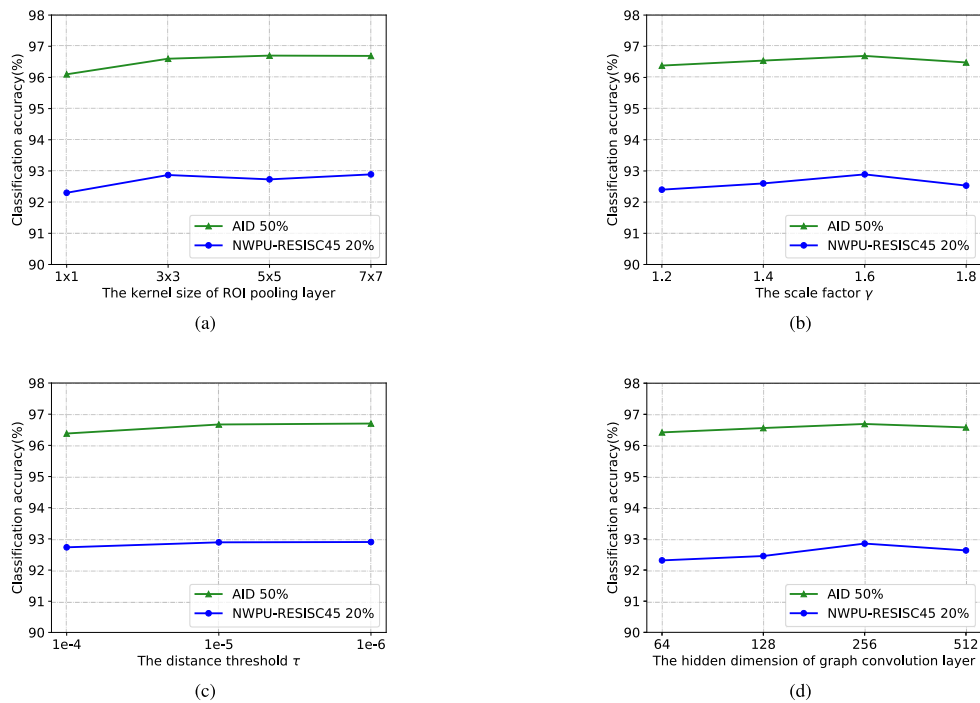


Fig. 8. Parameter evaluation of our proposed method. (a) Kernel size of ROI pooling layer. (b) Hidden dimension of graph convolution layer. (c) Distance threshold  $\tau$  of adjacency matrix. (d) Scale factor  $\gamma$  of Gaussian kernel.



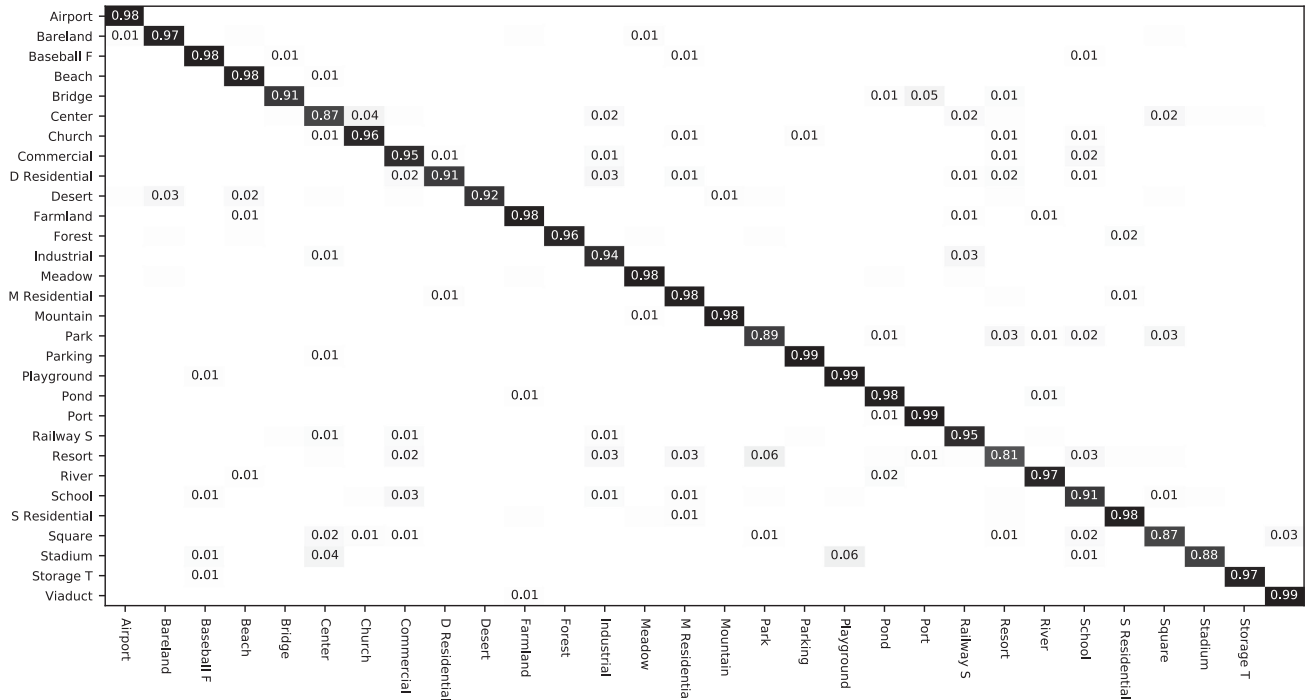


Fig. 9. Confusion matrix of our method on AID dataset by fixing the training ratio as 20%.

TABLE II  
OVERALL ACCURACY (%) OF OUR PROPOSED METHOD AND THE  
COMPARISON METHODS UNDER THE TRAINING RATIOS OF 50% AND 20%  
ON THE AID DATASET

Methods	50% Training Set	20% Training Set
CaffeNet [47]	89.53 ± 0.31	86.86 ± 0.47
GoogLeNet [47]	86.39 ± 0.55	83.44 ± 0.40
VGG16 [47]	89.64 ± 0.36	86.59 ± 0.29
salM <sup>3</sup> LBP-CLM [50]	89.76 ± 0.45	86.92 ± 0.35
Fusion by addition [51]	91.87 ± 0.36	-
TEX-Net-LF [52]	92.96 ± 0.18	90.87 ± 0.11
VGG-VD16+MSCP [53]	94.42 ± 0.17	91.52 ± 0.21
Two-Stream Fusion [24]	94.58 ± 0.25	92.32 ± 0.41
SFCNN [29]	96.66 ± 0.11	93.60 ± 0.12
D-CNN with VGGNet-16 [54]	96.89 ± 0.10	90.82 ± 0.16
<b>Ours</b>	<b>96.89 ± 0.10</b>	<b>94.93 ± 0.31</b>

The best performance is highlighted in bold.

4) *Hidden Dimension of Graph Convolution Layer*: We stacked two graph convolutional layers in our GCN-based branch model. By stacking layers, we can aggregate feature information from the neighbors of each node. GCN updates the hidden state of nodes by a weighted sum of the features of their neighbors. From Fig. 8(d), we can find that the highest classification accuracy is obtained when the hidden dimension between two graph convolutional layers is set to 256.

#### F. Experimental Results and Analysis

1) *AID Dataset*: To illustrate the superiority of the proposed method, a comparative evaluation against several state-of-the-art classification methods on the AID dataset is shown in Table II.

We select eight mainstream methods based on the deep learning network and compare the performance of scene classification. As can be seen from Table II, our proposed method, by combining the CNN and GCN, achieved the highest overall accuracy of 96.70% and 94.93% using 50% and 20% training ratios, respectively. It is worth mentioning that our method outperformed the SFCNN [29] with increases in the overall accuracy of 1.27% under the training ratio of 20%. The classification performance of our method verifies the effectiveness of combining global-based visual features and object-based location features on AID dataset.

Figs. 9 and 10 show the confusion matrix generated by our proposed method with the 20% and 50% training ratio. AID dataset contains 30 scene categories, including airport, bare land, baseball field, beach, bridge, center, church, commercial, dense residential, desert, farmland, forest, industrial, meadow, medium residential, mountain, park, parking, playground, pond, port, railway station, resort, river, school, sparse residential, square, stadium, storage tanks, and viaduct. As can be seen from Fig. 9, the classification accuracy of 24 categories is greater than 90% and 17 categories is greater than 95% in AID dataset. The most notable confusion occurs between resort and school. Specifically, 6% of images from resort are mistakenly classified as park, 3% of images from school are mistakenly classified as commercial. These two categories are very confusing because of the appearance similarity, so that other methods usually get much lower accuracy. For example, SFCNN [29] only achieves 70% for the class of resort but our method gets 75%. As can be seen from Fig. 10, the classification accuracy of 29 categories is greater than 90%. The categories of school and resort had relatively high classification accuracies with 90% and 91%. It confirms that our method is very good at capture spatial location

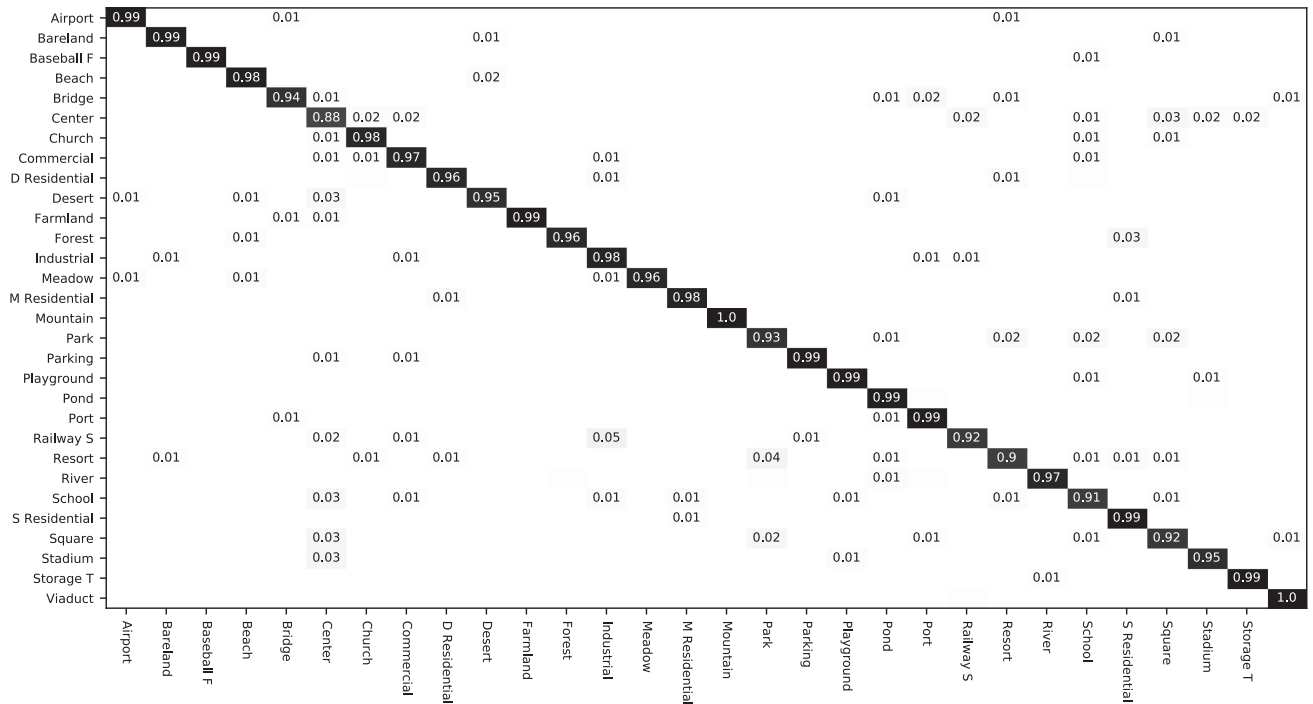


Fig. 10. Confusion matrix of our method on AID dataset by fixing the training ratio as 50%.

TABLE III

OVERALL ACCURACY (%) OF OUR PROPOSED METHOD AND THE COMPARISON METHODS UNDER THE TRAINING RATIOS OF 20% AND 10% ON THE NWPU-RESISC45 DATASET

Methods	20% Training Set	10% Training Set
GoogLeNet [48]	78.48 ± 0.26	76.19 ± 0.38
VGG16 [48]	79.79 ± 0.15	76.47 ± 0.18
AlexNet [48]	79.85 ± 0.13	76.69 ± 0.21
Two-Stream Fusion [24]	83.16 ± 0.18	80.22 ± 0.22
BoCF [55]	84.32 ± 0.17	82.65 ± 0.31
Fine-tuned AlexNet [48]	85.16 ± 0.18	81.22 ± 0.19
Fine-tuned GoogLeNet [48]	86.02 ± 0.18	82.57 ± 0.12
VGG-VD16+MSCP [53]	88.93 ± 0.14	85.33 ± 0.17
Fine-tuned VGG16 [48]	90.36 ± 0.18	87.15 ± 0.45
D-CNN with VGGNet-16 [54]	91.89 ± 0.22	89.22 ± 0.50
Triple networks [56]	92.33 ± 0.20	-
SFCNN [29]	92.55 ± 0.14	89.89 ± 0.16
<b>Ours</b>	<b>92.87 ± 0.13</b>	<b>90.75 ± 0.21</b>

The best performance is highlighted in bold.

information to distinguish these categories with small interclass variance.

2) *NWPU-RESISC45 Dataset*: We also evaluate our method on the more challenging dataset, NWPU-RESISC45. The comparison of the proposed method and the existing state-of-the-art classification methods on the NWPU-RESISC45 dataset is shown in Table III. We select ten mainstream methods based on the deep learning network and compare the performance of scene classification. As we can see, our classification method outperforms all of the comparison methods, which achieved the overall accuracy of 90.75% and 92.87% using 10% and 20% training

ratios, respectively. Specifically, our method outperforms the SFCNN [29] with increases in the overall accuracy of 0.86% under the training ratio of 20%. The classification performance of our method demonstrates the effectiveness of combining global-based visual features and object-based location features.

From the experimental results, we can find that NWPU-RESISC45 is much more difficult than the AID dataset. Therefore, it is absolutely essential to analyze the experimental result through the confusion matrix. Figs. 11 and 12 show the confusion matrix generated by the proposed method with the 10% and 20% training ratio. There are 45 scene categories, including airplane, airport, baseball diamond, basketball court, beach, bridge, chaparral, church, circular farmland, cloud, commercial area, dense residential, desert, forest, freeway, golf course, ground track field, harbor, industrial area, intersection, island, lake, meadow, medium residential, mobile home park, mountain, overpass, palace, parking lot, railway, railway station, rectangular farmland, river, roundabout, runway, sea ice, ship, snow berg, sparse residential, stadium, storage tank, tennis court, terrace, thermal power station, and wetland. As can be seen from Fig. 11, the classification accuracy of 27 categories is greater than 90% and 14 categories is greater than 95% in NWPU-RESISC45 dataset. It is seen that the church and palace are two confusing categories that get the lowest classification accuracy. Specifically, 24% of images from church are mistakenly classified as palace and 12% of images from palace are mistakenly classified as church. There are many similarities between these two categories, which leads many existing works to be unable to get a better performance. For example, SFCNN [29] only achieves 67% for the class of palace but our method gets 70%. By analyzing the confusion matrix on our

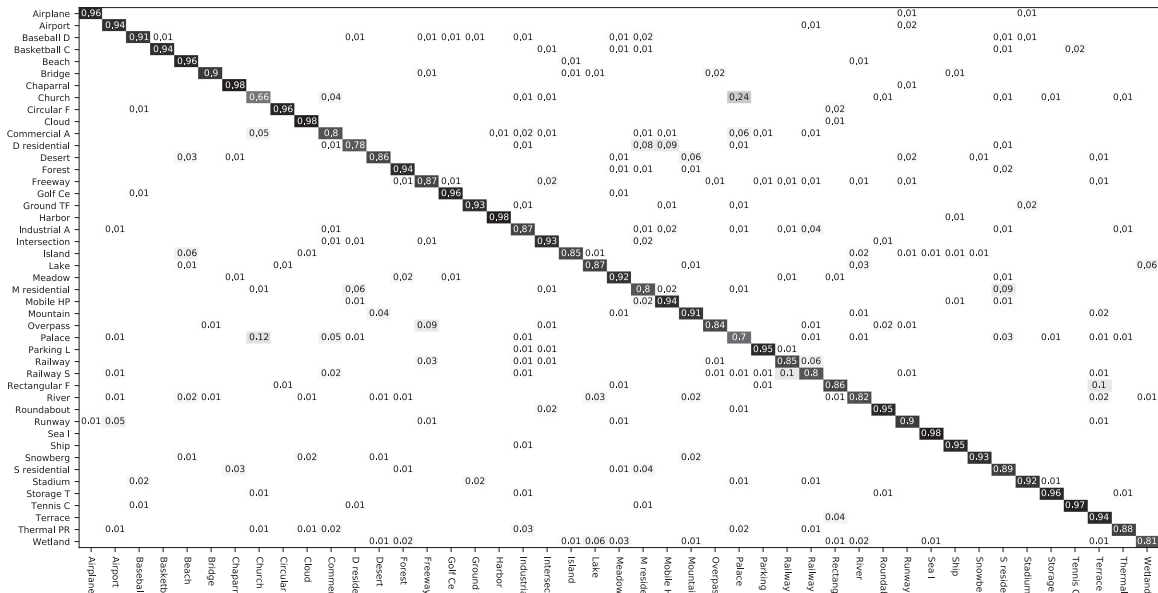


Fig. 11. Confusion matrix of our method on NWPU-RESISC45 dataset under the training ratio of 10%.

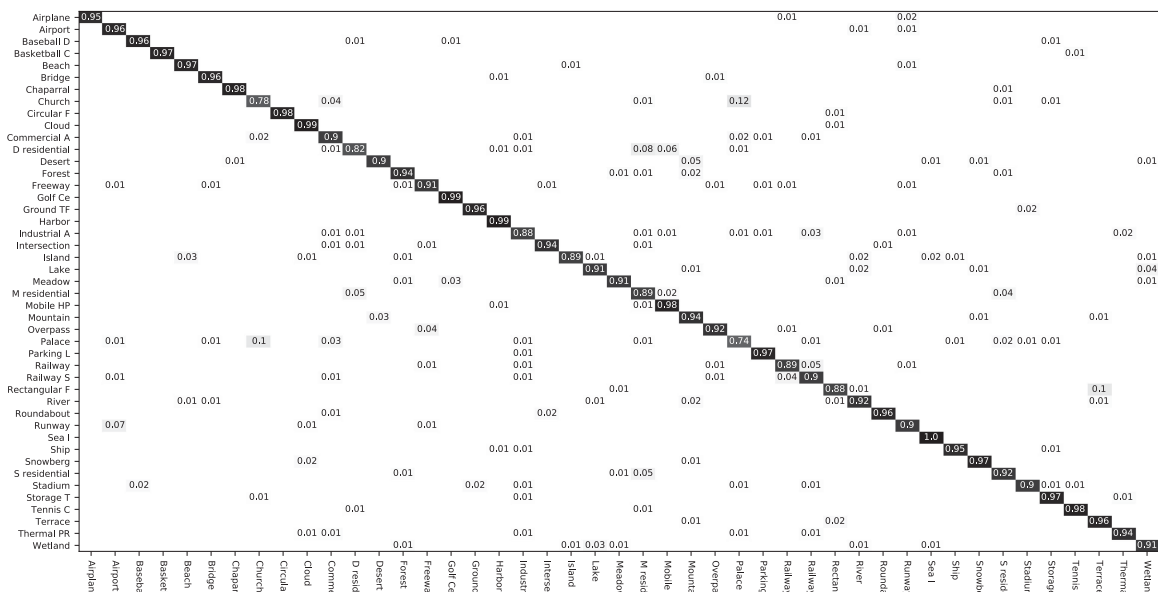


Fig. 12. Confusion matrix of our method on NWPU-RESISC45 dataset under the training ratio of 20%.

method, we can see that the number of misclassified categories is relatively reduced. The abovementioned experimental results demonstrate our proposed method works well on the large-scale NWPU-RESISC45 dataset, which can reduce visual confusion and improve the discriminative ability of feature descriptors.

### G. Ablation Study

We further perform several extra ablation studies to verify the effectiveness of our proposed model. Table IV represents the classification performance of our proposed method with

and without graph structure, to analyze the effectiveness of CNN-based branch, and their collaborative representation. We also report the results of without modification of VGGNet. By removing the max-pooling layer and adding a global average pooling layer, we effectively preserve detailed information in high spatial resolution scene images. Compared to the CNN branch, there are almost 0.82% and 0.75% improvements yielded under the two training ratios on the AID dataset, and 0.98% and 1.43% improvements yielded under the two training ratios on the NWPU-RESISC45 dataset. Our proposed method achieved a better performance compared to only using the CNN-based

TABLE IV  
ABLATION STUDIES ON THE AID AND NWPU-RESISC45 DATASETS

Methods	AID		NWPU-RESISC45	
	50%	20%	20%	10%
VGG16	94.18	92.25	90.57	87.81
CNN	95.88	94.08	91.89	89.32
CNN-GCN	96.70	94.93	92.87	90.75

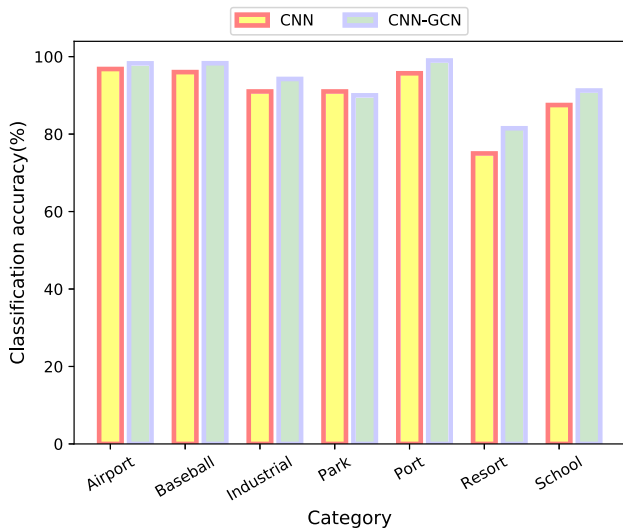


Fig. 13. Classification performance of certain categories on the AID dataset with the training ratio of 20%.

branch method, which was a result of combining global-based visual features by CNN and object-based location features by GCN. Therefore, the experimental results reveal that a more discriminative feature representation of the fusion of appearance visual information and the spatial dependencies among objects can generate superior performance.

Besides the abovementioned comparison results, the classification performance for certain scene categories is also shown in Fig. 13. It is worth noting that these categories containing scene-related local objects in remote sensing images, such as school and resort, achieved promising improvement. In addition, there are also several categories performance reduced after combining GCN. This can be explained that some error detection results bring negative effect on the classification result. On the whole, our proposed method improves the classification accuracy by combining the CNN and GCN, especially these categories contained a large number of scene-related ground objects.

#### H. Evaluation of Size of Model

We also compared the number of parameters and floating point operations per second (FLOPs) with other methods, representing the size of model and the computation complexity, respectively. The results are listed in Table V. During the process of implementation, we saved the object detection results offline. Therefore, we evaluated the detection module and the classification

TABLE V  
PARAMETERS COMPARISON WITH DIFFERENT METHODS

Methods	Parameters	FLOPs
CaffeNet [47]	60.97M	715M
VGG16 [47]	138.36M	15.5G
GoogLeNet [47]	7M	1.5G
Detection (ours)	523M	172G
Classification (ours)	15.28M	71.9G

module separately. It can be seen the size of our proposed two stream architecture is small. But our model has a higher calculation consumption, which is one of the improvement directions in the future.

#### IV. CONCLUSION

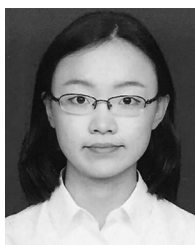
In this work, to explore the potential spatial dependencies among objects, we propose a novel two-stream remote sensing scene classification architecture. By combining CNN and GCN, our proposed network jointly learns global-based visual features and object-based location features. It not only utilizes the excellent feature extraction characteristics of the deep learning network but also introduces graph structure model into our task. Our network simultaneously obtains the appearance characteristics of the whole image and the spatial dependency between objects, which effectively reduces visual confusion and improves the discriminative ability of features. Experiments are performed on two challenging large-scale datasets, and the experimental results prove the spatial location and distribution information of objects is crucial for scene classification. In future work, dynamic graph convolutional can be used to further improve performance by reducing the impact of a negative predefined graph. The approach that we presented in this article is based on rich object categories and accurate object location. In the future, we will annotate more object categories relevant to the scene label on public datasets, so that more scene images can be modeled by graph. And then we can expand our work into hyperspectral images, which usually contain a large number of pixels corresponding to many land-cover classes, respectively. Therefore, our method is also suitable for classifying the pixels of a hyperspectral image into certain land-cover categories.

#### REFERENCES

- [1] R. K. Jaiswal, R. Saxena, and S. Mukherjee, "Application of remote sensing technology for land use/land cover change analysis," *J. Indian Soc. Remote Sens.*, vol. 27, no. 2, pp. 123–128, Jun. 1999.
- [2] L. Gmez-Chova, D. Tuia, G. Moser, and G. Camps-Valls, "Multimodal classification of remote sensing images: A review and future directions," *IEEE Proc.*, vol. 103, no. 9, pp. 1560–1584, Sep. 2015.
- [3] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in vHR optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7405–7415, Dec. 2016.
- [4] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state-of-the-art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016.
- [5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, Jun. 2005, pp. 886–893.

- [6] J. Ren, X. Jiang, and J. Yuan, "Learning LBP structure by maximizing the conditional mutual information," *Pattern Recognit.*, vol. 48, pp. 3180–3190, Feb. 2015.
- [7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, pp. 91–110, Nov. 2004.
- [8] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, Jun. 2005, pp. 524–531.
- [9] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, New York, NY, USA, Jun. 2006, pp. 2169–2178.
- [10] H. Jgou, F. Perronnin, M. Douze, J. Snchez, P. Prez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1704–1716, Sep. 2012.
- [11] B. Zhao, Y. Zhong, L. Zhang, and B. Huang, "The fisher kernel coding framework for high spatial resolution scene classification," *Remote Sens.*, vol. 8, no. 2, Feb. 2016, Art. no. 19.
- [12] 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.*, San Jose, CA, USA, Nov. 2010, pp. 270–279.
- [13] 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.
- [14] K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.
- [15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Ohio, CO, USA, Jun. 2014, pp. 580–587.
- [16] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 3431–3440.
- [17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 248–255.
- [18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, San Diego, CA, USA, May 2015, pp. 1–14.
- [19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, NY, USA, Dec. 2012, pp. 1097–1105.
- [20] C. Szegedy, W. Liu, and Y. Jia, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 1–9.
- [21] F. Luus, B. Salmon, F. Van Den Bergh, and B. Maharaj, "Multiview deep learning for land-use classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 12, pp. 2448–2452, Dec. 2015.
- [22] W. Zhao and S. Du, "Learning multiscale and deep representations for classifying remotely sensed imagery," *ISPRS J. Photogrammetry Remote Sens.*, vol. 113, pp. 155–165, 2016.
- [23] 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.
- [24] Y. Yu and F. Liu, "A two-stream deep fusion framework for high-resolution aerial scene classification," *Comput. Intell. Neurosci.*, vol. 2018, Jan. 2018, Art. no. 8639367.
- [25] G. L. Wang, B. Fan, S. M. Xiang, and C. Pan, "Aggregating rich hierarchical features for scene classification in remote sensing imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 9, pp. 4104–4115, Sep. 2017.
- [26] E. Z. Li, J. S. Xia, P. J. 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.
- [27] Q. S. Liu, R. L. Hang, H. H. Song, and Z. Li, "Learning multiscale deep features for high-resolution satellite image scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 117–126, Jan. 2018.
- [28] X. Liu, Y. Zhou, J. Zhao, R. Yao, B. Liu, and Y. Zheng, "Siamese convolutional neural networks for remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens. Lett.*, vol. 16, no. 8, pp. 1200–1204, Aug. 2019.
- [29] J. Xie, N. He, L. Fang, and A. Plaza, "Scale-free convolutional neural network for remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6916–6928, Sep. 2019.
- [30] Y. Liu, Y. Zhong, and Q. Qin, "Scene classification based on multiscale convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 7109–7121, Sep. 2018.
- [31] 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.
- [32] Q. Bi, K. Qin, Z. Li, H. Zhang, K. Xu, and G. Xia, "A multiple-instance densely-connected ConvNet for aerial scene classification," *IEEE Trans. Image Process.*, vol. 29, pp. 4911–4926, Mar. 2020.
- [33] K. Nogueira, O. A. B. Penatti, and J. A. dos Santos, "Towards better exploiting convolutional neural networks for remote sensing scene classification," *Pattern Recognit.*, vol. 61, pp. 539–556, Jan. 2017.
- [34] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.
- [35] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, Toulon, France, Apr. 2017, pp. 1–10.
- [36] C. Feng, Z. Liu, S. Lin, and T. Q. S. Quek, "Attention-based graph convolutional network for recommendation system," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Brighton, U.K., May 2019, pp. 7560–7564.
- [37] H. Zhang, Z. Kyaw, S. F. Chang, and T. S. Chua, "Visual translation embedding network for visual relation detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, Jul. 2017, pp. 3107–3115.
- [38] Z. Chen, X. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, Jun. 2019, pp. 5172–5181.
- [39] Y. Zhang and M. Rabbat, "A graph-CNN for 3D point cloud classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Calgary, AB, Canada, Apr. 2018, pp. 6279–6283.
- [40] J. Chen, T. Ma, and C. Xiao, "FastGCN: Fast learning with graph convolutional networks via importance sampling," in *Proc. Int. Conf. Learn. Representations*, Vancouver, BC, Canada, Apr./May 2018, pp. 1–15.
- [41] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
- [42] R. Girshick, "Fast R-CNN," 2015, in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 1440–1448.
- [43] M. Lin, Q. Chen, and S. Yan, "Network in network," in *Proc. Int. Conf. Learn. Representations*, Apr. 2014, pp. 1–10.
- [44] G. S. Xia *et al.*, "DOTA: A large-scale dataset for object detection in aerial images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 3974–3983.
- [45] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proc. AAAI Conf. Artif. Intell.*, Honolulu, HI, USA, Jan./Feb. 2019, pp. 7370–7377.
- [46] R. Liao, Z. Zhao, R. Urtasun, and R. Zemel, "Lanczosnet: Multi-scale deep graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, New Orleans, LA, USA, May 2019, pp. 1–18.
- [47] 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.
- [48] G. Cheng, J. W. Han, and X. Q. Lu, "Remote sensing image scene classification: Benchmark and state of the art," *Proc. IEEE*, vol. 105, no. 10, pp. 1865–1883, Oct. 2017.
- [49] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *IEEE Proc.*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [50] 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.
- [51] S. Chaib, H. Liu, Y. Gu, and H. Yao, "Deep feature fusion for VHR remote sensing scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 55, no. 8, pp. 4775–4784, Aug. 2017.
- [52] R. M. Anwer, F. S. Khan, J. van deWeijer, M. Monlinier, and J. Laaksonen, "Binary patterns encoded convolutional neural networks for texture recognition and remote sensing scene classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 138, pp. 74–85, Apr. 2017.

- [53] N. He, L. Fang, S. Li, A. Plaza, and J. Plaza, "Remote sensing scene classification using multilayer stacked covariance pooling," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 12, pp. 6899–6910, Dec. 2018.
- [54] 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.
- [55] 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.
- [56] 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.



**Jiali Liang** received the B.E. degree in communication engineering, in 2018 from Shanghai University, Shanghai, China, where she is currently working toward the M.E. degree in communication and information system.

Her research interests include remote sensing scene classification and person reidentification.



**Yufan Deng** is currently studying in the Shenzhen Middle School, Shenzhen, China. He has completed the Chinese University of Hong Kong-Shenzhen, Shenzhen, China, and University of Pennsylvania, Philadelphia, PA, USA, 2019 Robotics Research Camp and studied in Computer Vision and Pattern Recognition Laboratory, Shanghai University, Shanghai, China, during the summer of 2019.



**Dan Zeng** (Member, IEEE) received the B.S. degree in electronic Science and technology and the Ph.D. degree in circuits and systems from the University of Science and Technology of China, Hefei, China, in 2003 and 2008, respectively.

Since 2008, she has been teaching in the Communication Engineering Department, Shanghai University, Shanghai, China, where she is currently a Professor. From 2001 to 2004, she was a Visiting Scholar with the School of computer science, and the National Scholarship Council, University of Texas, San Antonio, San Antonio, TX, USA. Her research interests include computer vision,

machine learning, and multimedia technology.