

## A semi-supervised generative framework with deep learning features for high-resolution remote sensing image scene classification

Wei Han, Ruyi Feng, Lizhe Wang\*, Yafan Cheng

School of Computer Science, China University of Geosciences, Wuhan 430074, PR China  
Hubei Key Laboratory of Intelligent Geo-Information Processing, China University of Geosciences, Wuhan 430074, PR China



### ARTICLE INFO

#### Article history:

Received 30 May 2017

Received in revised form 25 October 2017

Accepted 4 November 2017

Available online 14 November 2017

#### Keywords:

Scene classification

Deep learning

Self-label

High resolution remote sensing images

### ABSTRACT

High resolution remote sensing (HRSS) image scene classification plays a crucial role in a wide range of applications and has been receiving significant attention. Recently, remarkable efforts have been made to develop a variety of approaches for HRSS scene classification, wherein deep-learning-based methods have achieved considerable performance in comparison with state-of-the-art methods. However, the deep-learning-based methods have faced a severe limitation that a great number of manually-annotated HRSS samples are needed to obtain a reliable model. However, there are still not sufficient annotation datasets in the field of remote sensing. In addition, it is a challenge to get a large scale HRSS image dataset due to the abundant diversities and variations in HRSS images. In order to address the problem, we propose a semi-supervised generative framework (SSGF), which combines the deep learning features, a self-label technique, and a discriminative evaluation method to complete the task of scene classification and annotating datasets. On this basis, we further develop an extended algorithm (SSGA-E) and evaluate it by exclusive experiments. The experimental results show that the SSGA-E outperforms most of the fully-supervised methods and semi-supervised methods. It has achieved the third best accuracy on the UCM dataset, the second best accuracy on the WHU-RS, the NWPU-RESISC45, and the AID datasets. The impressive results demonstrate that the proposed SSGF and the extended method is effective to solve the problem of lacking an annotated HRSS dataset, which can learn valuable information from unlabeled samples to improve classification ability and obtain a reliable annotation dataset for supervised learning.

© 2017 International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS). Published by Elsevier B.V. All rights reserved.

## 1. Introduction

The presently available technologies (e.g. multi/hyper-spectral, synthetic aperture radar) for earth observation generate many types of airplane and satellite images with high resolutions (spatial resolution, spectral resolution, and temporal resolution) (Cheng et al., 2017; Plaza et al., 2011; Gamba, 2013; Cantaloube and Nahum, 2013; Lu et al., 2017; Li et al., 2016; Yuan et al., 2017; Liu et al., 2017). The main task transfers to intelligent earth observation through massive high-resolution remote sensing (HRSS) images, which smart classify land use and land cover scenes (LULC) from airborne or space platforms (Gómez-Chova et al., 2015). Remote sensing image scene classification, which plays an important role in earth observation and is receiving significant attention,

categorizes scene images into an independent set of semantic-level LULC class labels according to image contents. During the past few decades, a lot of remarkable efforts have been made to develop various methods for the task of HRSS image scene classification in a wide range of applications (Wang et al., 2016; Li and Wang, 2015; Dou et al., 2014; Cheng and Han, 2016; Ma et al., 2016; Yu et al., 2016; Dópido et al., 2013; Li et al., 2014), such as LULC determination urban planning, environmental protection, and crop monitoring.

Deep-learning-based methods, which achieve many improvements over state-of-the-art records in many research fields, have been widely applied in natural images classification, object recognition, natural language, and text processing (Chatfield et al., 2014; Simonyan and Zisserman, 2015; He et al., 2016; Krizhevsky et al., 2012; Szegedy et al., 2015). Due to their remarkable performance, these methods are used to analyze HRSS images, and have achieved more impressive results than the traditional shallow methods for scene classification (Castelluccio et al.; Hu et al.,

\* Corresponding author at: School of Computer Science, China University of Geosciences, Wuhan 430074, PR China.

E-mail address: [lizhe.wang@gmail.com](mailto:lizhe.wang@gmail.com) (L. Wang).

2015; Zhang et al., 2016; Zhao and Du, 2016; Luo et al., 2017; Wang et al., 2017; Cheng et al., 2016). Though satellite and aerial images dramatically increase in both quality and quantity, deep-learning-based methods in a fully-supervised learning fashion (Zhang et al., 2015) require a large scale, artificially-annotated dataset to obtain ideal classifiers. However, there is no HRRS dataset with as comparative a scale as ImageNet (Deng et al., 2009) to meet the requirements of the deep-learning-based methods in remote sensing. Additionally, in contrast to natural images, an annotated HRRS dataset needs to be labeled by experts and engineers, which greatly increases the difficulty of acquiring a large scale annotation dataset of HRRS images.

The acquisition of unlabeled images is much easier compared to acquiring a manually-annotated dataset. Hence, the use of the original, unlabeled data to generate labeled data could solve the problem of lacking labeled samples. Self-label technique (Triguero et al., 2015) is an available solution that aims to obtain an enlarged annotation dataset from unlabeled samples via semi-supervised learning. However, the existing self-label methods have a significant weakness in that they annotate samples by employing handcrafted features. Handcrafted features are designed by experts and engineers to solve the classification tasks. Traditional handcrafted features contain many severe limitations. On the one hand, extensive of domain expertise and engineering skills are needed to design handcrafted features. On the other hand, the representational capability of handcrafted features is significantly influenced by human ingenuity in feature designing. The ideal features should be automatically generated with powerful representation ability. Fortunately, deep learning features, which are spontaneously learned from data by a deep architecture neural network with remarkable performance, could be an option to address the limitations of the handcrafted features. Therefore, we replace the handcrafted features of the self-label techniques with deep learning features to carry out our work.

We propose a semi-supervised, generative framework with deep learning features (SSGF) for HRRS image scene classification to solve the problem of lacking sufficient annotation HRRS datasets. The details of this framework are summarized below:

1. Deep convolutional neural network (CNN) features are transferred to replace the traditionally handcrafted features due to their powerful representation ability. It enables the discovery of sufficient diversities and variations hidden in the HRRS images and provides a better understanding of scene classes.
2. A co-training (Blum and Mitchell, 1998) self-label method is used to learn valuable information from unlabeled samples and obtain an annotated dataset. It not only makes use of the low-confidence samples, but suppresses the problem of misclassification.
3. A discriminative evaluation method enhances classification of the confusion classes with similar texture structures and visualized features, which further improves the reliability of generative samples.

By combining the three techniques, the proposed SSGF is able to learn effective information from unlabeled data for the improvement of classification ability. Therefore, with a limited number of annotation samples and a significant number of unlabeled samples, the ideal model can be obtained. Hence, the enlarged set generated by the model is available for supervised learning. To evaluate the performance of SSGF, we further develop an extended algorithm (SSGA-E). The major contributions of this work are summarized as follows:

1. Focusing on the problem of insufficient annotation datasets in remote sensing, we propose a semi-supervised generative

framework. It can instantly improve the capability of scene classification by learning unlabeled instances, and generate a reliable annotation dataset for supervised learning.

2. On the basis, we further develop an extended algorithm. We have performed extensive experiments to evaluate the proposed method over four public HRRS datasets. The experimental results show that the proposed method outperforms most of the fully-supervised methods, and it has achieved the third best accuracy on the UCM dataset, the second best accuracy on the WHU-RS, the NWPU-RESISC45, and the AID datasets. The experimental results demonstrate that the proposed SSGA-E is effective in solving the problem of insufficient annotated datasets for HRRS image scene classification.

The remainder of this paper is organized as follows: In Section 2, we briefly review some related works about deep learning methods and scene classification. In Section 3, the deep neural networks used in this works are introduced briefly. The semi-supervised generative framework and an extended algorithm are proposed and explained in detail in Section 4. We display and discuss the experimental results in Section 5. Finally, conclusions are drawn in Section 6.

## 2. Related work and background

In the early 1970s, the spatial resolution of satellite images was extremely coarse and pixel sizes were similar in size or bigger than the interest objects (Janssen and Middelkoop, 1992). Therefore, available methods for analysis of remote sensing images have been based on pixel level since the early 1970s (Blaschke et al., 2008; Blaschke, 2010). With the advance of remote sensing technology, a higher number of HRRS images are obtainable, such the UCMerced Land Use dataset (Yang and Newsam, 2010) and WHU-RS19 dataset (Xia et al., 2010), in which many pixels compose of objects of interest and the texture and color information of the object is displayed more clearly. The HRRS images contain multiple scene classes, sufficient diversities, and variations compared to early remote sensing images. In this case, it is challenging for traditional pixel-based methods to recognize HRRS image. To solve the limitations of the pixel-based methods, object-level methods have been designed to recognize and classify HRRS images. These methods enable better understanding of the contents of the HRRS images at a semantic-level. Since scene classification is generally effected by feature maps, effective feature representation plays a crucial role in high performance classification methods.

The existing scene classification methods are divided into three categories (Cheng et al., 2017) according to their features: handcrafted-feature-based methods, unsupervised-feature-learning-based methods, and deep feature learning methods. The handcrafted-feature-based methods primarily use a remarkable amount of domain expertise and engineering skills to construct numerous human engineering features such as GIST (Oliva and Torralba, 2001), SIFT (Lowe, 2004), and HOG(Dalal and Triggs, 2005). Their representation capabilities are significantly influenced by the involvement of human ingenuity in feature design. Additionally, as HRRS images become finer and more complex, the description abilities of the handcrafted-feature-based methods may become limited. The unsupervised-feature-based methods focus on learning a set of basis functions used for feature encoding instead of relying on manually designed features (Fu et al., 2015; Lu et al., 2017). By doing this, we can obtain the more valuable and discriminative features that are often more applicable to practical problems. The unsupervised methods contain k-means clustering, sparse coding (Olshausen and Field, 1997), and principal

component analysis (PCA) (Jolliffe, 2002). Due to lack of the semantic information given by the labels, the discriminative ability of the unsupervised methods is not guaranteed. The deep feature learning methods, which contain a trainable multi-layer network, have shown impressive feature representation capability and achieved unbelievable improvement beyond state-of-the-art feature methods in many applications such as object detection, natural language, and scene classification. The existing deep learning models involve deep belief net (DBN) (Hinton et al., 2006), stacked auto-encoder (SAE) (Vincent et al., 2010), and convolutional neural networks (CNNs) (Chatfield et al., 2014; Simonyan and Zisserman, 2015; He et al., 2016; Krizhevsky et al., 2012; Szegedy et al., 2015). Though satellite and aerial images have significantly enhanced in both quantity and quality, the deep-feature-learning methods have encountered a serious challenge in that these approaches often require numerous training samples with accurate manual annotation to train an effective classifier. But there is usually not an artificially-labeled dataset with a similar scale like ImageNet to finish the training tasks in most practical applications.

Self-label techniques are used to tackle semi-supervised classification (SSC). Focusing on obtaining an enlarged annotation dataset, they annotate the most confident prediction to the unlabeled instances. This was first proposed to process English text for sense disambiguation (Yarowsky, 1995). After that, self-label techniques have achieved significant developments and generated many modified methods, such as co-training (Blum and Mitchell, 1998), CoForest (Li and Zhou, 2007), and SNNRCE (Wang et al., 2010). Self-label techniques are typically divided into self-training and co-training. In self-training, a classifier, initially training on a limited number of labeled samples, classifies unlabeled instances with the most confident labels and merges them with the initial samples into an enlarged set. Next, it is retrained on an enlarged dataset. This scheme only accepts that its own predictions are correct. Self-training is simple and fast method to obtain the enlarged set, but the reliability is not guaranteed. Standard co-training assumes that feature space could be divided into two different and independent views, and that each view enables correct prediction of the classes. It respectively trains one classifier in each feature view, and then two classifiers teach each other the most confident examples. Co-training achieves better annotated results than self-training while inevitably increasing complexity and computational time. However, existing self-label methods are based on hand-crafted features, which may cause misclassification, and the learned model rapidly deviates from the actual model during iterative learning. There are already some works that use deep learning features and semi-supervised methods to generate annotated samples (Yao et al., 2016; Han et al., 2015). But the CNNs that are suitable for images processed to extract high-level features, are not used. Semi-supervised generative adversarial nets (semi-supervised GANs) proposed recently are also used to generate specified image samples that currently contain SGAN (Odena, 2016) and CatGAN (Springenberg, 2015). They are the extended version of the generative adversarial net (GAN) by forcing the discriminator networks to output class labels. However, because existing semi-supervised GANs have only been applied in the simple datasets as MNIST and CIRAR-10, their superiority has not been proven in complex and diverse remote sensing images.

Different from the existing generative methods for HRRS images scene classification, we replace the handcrafted features of the self-label techniques with deep learning features. Focusing on the lack of sufficient annotation datasets, we develop a semi-supervised generative framework to solve the problem: (1) The deep learning features extracted from the pre-trained CNNs are used to replace the traditionally handcrafted feature for a better representation capability for unlabeled samples. (2) A co-training-based self-label technique and a discriminative evaluation are used to reduce

the problem of misclassification and obtain a reliable annotation dataset.

### 3. Deep Convolutional Neural Networks (CNNs)

In this section, we first discuss the typical structure of a CNN and the back propagation algorithm used to optimize the gradient to refer to the weight parameters of the network. Next, all CNNs used in this paper are briefly introduced.

#### 3.1. Convolutional neural networks

A Convolutional Neural Network (CNN) consists of one or many convolutional layers (always with nonlinearity and pooling layers) and followed by several full connection layers to output the recognition results. The structure of a typical CNN is suitable for the 2D shape of an input image. Fig. 1 illustrates that how an image is processed by CNN. It is implemented with local connections and shared weights followed by an activation function and pooling, which leads to translation invariant features. An input image is  $m \times m \times r$ , where  $m$  is the width and height properties and  $r$  is the number of color channels. A convolutional layer has  $k$  kernels of size  $n \times n \times q$ , where  $n$  is smaller than the height of the image and  $q$  can be smaller than or equal to  $r$  and may be different for each kernel. These kernels generate a local connection structure and are convolved with the images to construct  $k$  feature maps with a size of  $m - n + 1$ . After that, a non-linearity function is performed on each feature map between the convolutional and pooling layers. Next, each feature map is subsampled with maximum pooling over  $p \times p$  local regions where  $p$  is usually smaller than 5. After the convolutional layers, some fully-connected layers are placed to output the classification results.

In order to optimize the gradient, back propagation is employed to calculate the weight parameters of each layer. We denote  $\delta(l+1)$  as the loss term for the  $(l+1)_{th}$  layer in the network with a cost function  $J(W, b; x, y)$ , where  $(W, b)$  is the weight parameter and  $(x, y)$  is an instance and label as a training example. For a convolution layer, the loss function is formulated as

$$\delta_k^{(l)} = \text{upsample}((W_k^{(l)})^T \delta_k^{(l+1)}) * f'(z_k^{(l)}) \quad (1)$$

where  $k$  is the order of filter kernel and  $f'(z_k^{(l)})$  is the derivative of the activation function. The upsample needs to propagate the loss through the subsample layer by computing the loss of every unit input to the subsample layer. To compute the gradient of filter maps, the border-handling convolution operation is applied and the loss matrix  $\delta_k^{(l)}$  is flipped. The gradients are calculated:

$$\begin{cases} A \nabla_{W_k^{(l)}} J(W, b; x, y) = \sum_{i=1}^m (a_i^{(l)}) * \text{rot90}(\delta_k^{(l+1)}, 2) \\ \nabla_{b_k^{(l)}} J(W, b; x, y) = \sum_{a,b} (\delta_k^{(l+1)})_{a,b} \end{cases} \quad (2)$$

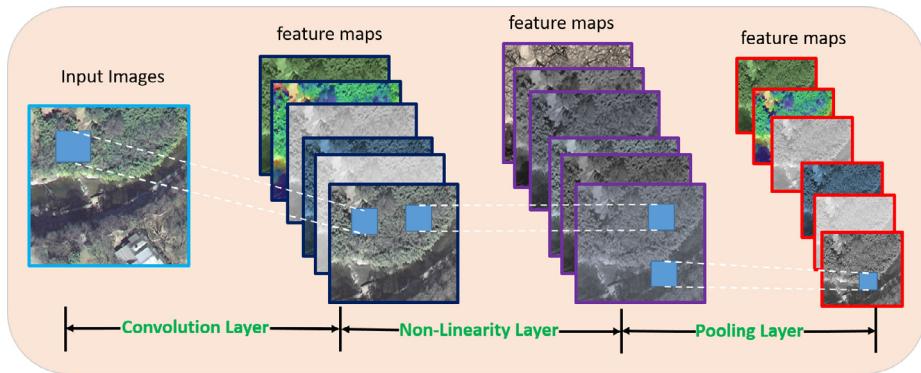
where  $a^{(l)}$  is the input vector to the  $l$ th layer, so  $a^{(1)}$  is a training instance. The operation  $(a_i^{(l)} * \delta_k^{(l+1)})$  is the available convolution between  $i_{th}$  the input map in the  $l$ th layer and the loss related to the  $k$ th filter kernel.

If the  $l_{th}$  belongs to a softmax or fully connection layer, then the loss of  $l_{th}$  layer is defined as

$$\delta^{(l)} = ((W^{(l)})^T \delta^{(l+1)}) * f'(z^{(l)}) \quad (3)$$

and the gradients are computed as

$$\begin{cases} \nabla_{W^{(l)}} J(W, b; x, y) = \delta^{(l+1)} (a^{(l)})^T \\ \nabla_{b^{(l)}} J(W, b; x, y) = \delta^{(l+1)} \end{cases} \quad (4)$$



**Fig. 1.** Illustration of a typical convolutional layer processing an image.

### 3.2. AlexNet and CaffeNet

AlexNet (Krizhevsky et al., 2012), a creative deep neural network, won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012. In contrast to the early neural networks, AlexNet is a deep architecture in a real sense, and contains convolutional layers followed by subsample layers and three fully connected layers. The remarkable performance of AlexNet is contributed to a set of practical techniques, such as data augmentation, Rectified Linear Units (ReLU) activation function and dropout. Subsequently, CaffeNet (Jia et al., 2014) is implemented by Yangqing Jia in Convolutional Architecture for Fast Feature Embedding (Caffe) which has a identical structure with AlexNet with the exception of two minor modifications: training without any data augmentation and exchanging the form of subsample and normalization layers. Because of the similar performance of AlexNet and CaffeNet, we only apply CaffeNet in the experimental section.

### 3.3. GoogLeNet

GoogLeNet (Szegedy et al., 2015), the champion in ILSVRC 2014, greatly increases the number of layers to 22. The outstanding classification capability of GoogLeNet is due to the “inception module,” which significantly reduces the number of parameters and the complexity of the 3D filters of the convolutional layers. GoogLeNet uses many auxiliary classifiers connected to intermediate layers. The “inception module” entails two advantages: maintains more specific spatial information by using the filters with different sizes at each convolutional layer, and the number of free parameter is dramatically reduced, making it less possible to overfit and be deeper.

### 3.4. VGG-Networks

Three CNN architectures VGG-F, VGG-M, and VGG-S are implemented based on Caffe (Chatfield et al., 2014). These models are used to compare the performance of different CNN models, each of which analyzes the accuracy/computation time trade-off. We make a simple comparison:

1. VGG-F, the fast CNN, has a similar structure with AlexNet. A smaller stride and a smaller number of filters in all convolutional layers are the main difference.
2. VGG-M employs a smaller pooling size and stride in the first convolutional layer and a small amount of filters in the fourth convolutional layer. It is constructed for balancing computational speed and classification capability.
3. VGG-S is the slow CNN architecture and is implemented based on the accuracy OverFeat model (Sermanet et al.). It maintains the first five convolutional layers in the specific accurate Over-

Feat model and uses fewer filters in the fifth layer. In contrast to VGG-M, a smaller stride is adopted in the second convolutional layer of VGG-S and the larger pooling size is used in first and fifth convolutional layers.

In ILSVRC-2014, the authors proposed two very deep CNNs (Simonyan and Zisserman, 2015), VGG-VD16 and VGG-VD19. VGG-VD16 has thirteen convolutional layers and three full connection layers, and VGG-VD 19 contains sixteen convolutional layers and three full connected layers. The VGG-VD group implemented remarkable performance and won runner-up in the competition. The VGG-Group Networks have been used in many works for extracting high-level CNN features as classical models. In this paper, we abbreviate VGG-VD16, VGG-VD19 as VGG-16 and VGG-19, respectively.

### 3.5. ResNet

Residual network (ResNet) (He et al., 2016), one of the widely-used neural networks, has aroused the considerable attention in deep learning. The most impressive characteristic of ResNet is depth. Compared with VGG Net, which contains a structure of sixteen or nineteen layers, ResNet is implemented with 50, 101, and even 152 layers (50 layers of architecture are used in this work, denoted as ResNet-50). Surprisingly, the performance of ResNet constantly grows as the depth increases because residual connections are used to reduce the effect of the vanishing gradient problem. ResNet is also prone to training because it has fewer extra parameters in each layer. A structure of residual connection is shown in Fig. 2. Mathematically, given an input image  $x$ , the transformation operations of convolutional layers  $f(x)$ , the merge operation (an addition operation), and some activation function such as ReLU  $r(x)$ , the output  $y$  is formulated as

$$y = r(x + f(x)) \quad (5)$$

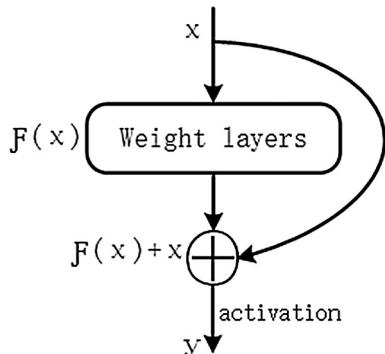
This structure enables passing the important information from the previous layer to the next layers, which insignificantly fastens the training process of the network and prevents the vanishing gradient problem.

## 4. A semi-supervised generative framework for HRRS scene classification

In this section, we first mathematically define the problem. Next, we propose the semi-supervised generative framework and an extended algorithm in detail.

### 4.1. Problem definition

In this section, we provide a set of specific definitions, and formulate the problem.



**Fig. 2.** Residual connection.

A labeled instance  $I = \langle x, y \rangle$  is two-tuples, where  $x$  is an image,  $y \in \{0, 1, 2, \dots, c\}$  is the corresponding class label, and  $c$  is the maximum index of categories. Therefore, a labeled set is composed of a series of annotated images, denoted as  $L = \{I_1, I_2, \dots, I_n\}$ , where  $n$  is the total number of images. For an unlabeled instance  $I' = \langle x' \rangle$ , only image  $x'$  is stored without the real label. Correspondingly, an unlabeled set is denoted as  $U = \{I'_1, I'_2, \dots, I'_m\}$ , where  $m$  is the number of unlabeled instances. In this work, the objective is to learn a map function  $f$  for an unlabeled image  $I'_i \in U$  to generate a predicted label  $\hat{y}_i$ ,

$$f : x'_i \rightarrow \hat{y}, \quad x'_i \in I'_i \quad (6)$$

while minimizing the mistake possibility over the actual distribution of original data,

$$\min P(f(x'_i) \neq f^*(x'_i)) \quad (7)$$

where  $f$  is the function learned by our method,  $f^*$  is the actual distribution function.

We solve the problem with the idea of bootstrapping (Narayanan et al., 2012), wherein an initial function  $f_0$  is learned from a limited number of labeled samples in  $L$ , then improved by training on the self-labeled samples from  $U$  in an iteration fashion. However, the naive bootstrapping category possibly causes semantic drift (Davenport and Cronin, 2000) and the target function  $f$  learned from  $U$  rapidly deviates from  $f^*$ . Semantic drift occurs when the semantics and usages of word change over time. If the error samples are introduced, the meanings of category labels are different from the original semantics. To prevent the problem of semantic drift, we use a set of approaches to improve the reliability of label samples. First, we replace the traditional handcrafted features with deep learning features. Then a co-training-based self-label method and a discriminative evaluation are used to jointly label samples which not only guarantee classification accuracy but also improve the model's discriminative capability by learning informative samples. The details of the several techniques are covered later in the section.

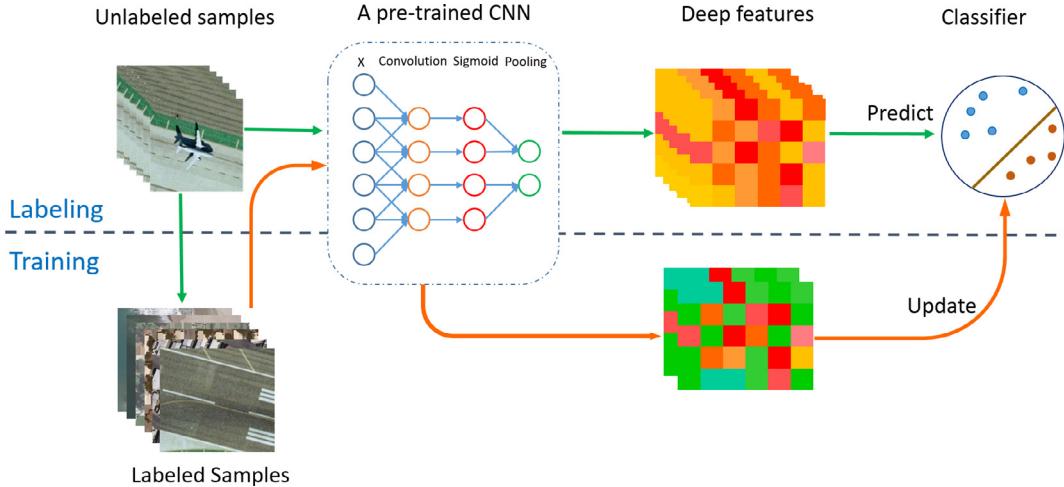
#### 4.2. A deep-learning-feature-based self-label method

As mentioned above, self-label via bootstrapping is an available solution for learning a significant number of unlabeled samples and obtaining sufficient annotation datasets for fully-supervised learning. In existing published works over text and image fields, the self-label approaches, which are handcrafted-

feature-based, have demonstrated their considerable performance for generating annotation datasets. However, the existing handcrafted-feature-based methods entail several severe limitations. First, powerful handcrafted features are designed by experts and engineers in the field, which is time-consuming. Additionally, in accordance with the growth of the scale of HRRS images, the diversities and variations of HRRS image datasets are significantly increasing, making it more difficult to design effective and robust handcrafted features. Ideal features should be automatically-generated and suitable for the characteristics of large-scale HRRS image datasets. Therefore, deep learning features that meet the requirements can solve the limitations of handcrafted features. Deep learning features are generated by a deep CNN, which have shown unbelievable performance in large scale scene classification and computer vision tasks. Therefore, we replace handcrafted features with deep features to maintain excellent representation capability and reduce the amount of manual workload. Since deep features are high-dimension vectors, SVM is effective in classification problems with high-dimension feature spaces. Additionally, it is also able to achieve desired accuracy and theoretically prevent overfitting. Due to advantages, it has been used in many related studies (Castelluccio et al.; Hu et al., 2015). Therefore, we select SVM as the classifier in this work. Through the above analysis, we propose the deep-learning-feature-based self-label method.

As shown in Fig. 3, this approach contains two main processes: training and labeling. In the training process, the annotated dataset is fed to a pre-trained CNN to generate the output vectors of the second full connection layer as high-level image representations. Image representations are the deep features of input images. Next, the features are used to train a *one vs rest* SVM classifier. If the SVM has been created, its parameters are updated. In the labeling stage, a proportion of unlabeled samples from the unlabeled pool are fed to the pre-trained CNN to output the high-dimensional features. The SVM classifier labels the samples according to its most confident predictions. Those samples with the predicted labels are merged into the original labeled set, which is regarded as the new training set.

The overall process is depicted detailedly in Algorithm 1, where the training and labeling processes are performed in an iterative fashion. Initially, the input is labeled set  $L$ , unlabeled set  $U$ , and batch size  $batch\_size$  (denotes the number of samples to be processed in each iterative round). Then  $EL$  is used to save the learned samples with predicted labels added, and  $LC$  keeps the remainder, which is difficult to classify.  $clf$ , an SVM evaluator, is initially trained on  $L$  (see line 4 in Algorithm 1). A set of samples from  $U$  are learned by  $clf$  and the annotated results are added into  $EL$ . Next,  $EL$  and  $L$  are merged as a new training set to update  $clf$ . The whole unlabeled set  $U$  is processed by continuously performing the training and labeling process (see line 7–12 in Algorithm 1).  $LC$ , containing a lot of previously unpredictable samples, is relearned to enlarge  $EL$  until the number of samples in  $EL$  does not change (line 13–19 in Algorithm 1). In our method, the additional mechanism of unlabeled samples is more similar to incremental (Triguero et al., 2015) mode than batch mode, wherein a part of an unlabeled set is learned to update the evaluator. Compared with the incremental model, a class label is not assigned to an instance during the labeling process in the batch mode. With this method is difficult to obtain an annotated dataset to improve the evaluator. Furthermore, the batch mode causes more significant time complexity than the incremental mode.



**Fig. 3.** Illustration of the deep-learning-feature-based self-label method.

**Algorithm 1.** Description of the deep learning based self-label method

```

1: Input: labeled set  $L$ , unlabeled set  $U$ , batch size  $batch\_size$ 
2: Output: enlarge set  $EL$ 
3: initialize  $EL, LC$  as empty,  $count = 0$   $\triangleright EL$  saves the
   samples from unlabeled set,  $LC$  remains the discard part,
    $count$  is number of iterations
4:  $clf \leftarrow training(L)$   $\triangleright clf$  is initialized by the labeled set  $L$ 
5:  $batch\_set \leftarrow return\_batch\_set(U, batch\_size, count)$ 
    $\triangleright batch\_set$  keeps data to be processed
6:  $labeling(clf, batch\_set, EL, LC)$   $\triangleright$  the function predicts
   samples and saves the results to  $EL$  and  $LC$ 
7: while  $batch\_size * count < len(U)$  do  $\triangleright$  processes the
   whole unlabeled set
8:    $count \leftarrow count + 1$ 
9:    $clf \leftarrow update\_training(clf, EL + L)$   $\triangleright$  updates the  $clf$  by
   training on  $EL$  and  $L$ 
10:   $batch\_set \leftarrow return\_batch\_set(U, batch\_size, count)$ 
11:   $labeling(clf, batch\_set, EL, LC)$ 
12: end while
13: while  $len(EL)$  increases do  $\triangleright$  processes the  $LC$  set until
   the sample number of  $EL$  doesn't increase
14:    $clf \leftarrow update\_training(clf, EL + L)$ 
15:    $batch\_set \leftarrow LC$ 
16:    $empty(LC)$ 
17:    $labeling(clf, batch\_set, EL, LC)$ 
18: end while
19: Return  $EL$ 

```

#### 4.3. A co-training-based self-label method

In Narayanan et al. (2012), a critical assumption is made that the objective function  $f$  learned from a limited number of labeled samples is a reasonable approximation of the actual map  $f^*$ , and an example from the unlabeled set  $U$ , which is classified with high confidence, would not be informative enough during the re-training process. Given an instance  $I' \in U$ , its image  $x'$  is considered an informative sample and only meets the condition,

$$P(f(x') = y) < P_{thresh}, \quad y \in \{0, 1, \dots, c\} \quad (8)$$

where  $y$  is the real label of  $x'$ ,  $P(f(x') = y)$  is the confidence scores returned by the SVM evaluator,  $P_{thresh}$  is an artificial hyperparameter. Correct low-confidence samples are informative to enlarge the feature space of the model and bring it closer to the true distribu-

tion  $f^*$ . However, error samples cause a significant degree of semantic drift. Taking into account the two scenarios, we utilize the co-training based algorithm. In traditional co-training (Triguero et al., 2015), two evaluators that are trained respectively on two views of the data help each other learn its own most confident instances. Our method is different in that the samples are promoted on the condition that both the evaluators agree. Additionally, new added samples are used to train both two independent evaluators. In order to ensure the accuracy of predictions, multiple views should have diversity and be complementary. We have chosen features from different CNN models to meet the requirements, which are introduced in detail later. Mathematically, a given unlabeled instance  $I' \in U$  and its image  $x'$  are regarded as a training sample that only meets

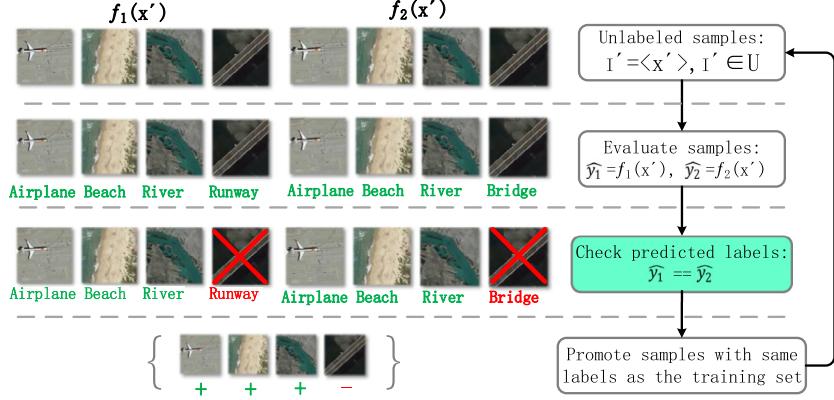
$$f_1(x') = f_2(x') \quad (9)$$

where  $f_1$  and  $f_2$  are two independent evaluators. We show the details in Fig. 4. A set of unlabeled samples are respectively fed to the two independent classifiers  $f_1$  and  $f_2$ ,  $\hat{y}_1$  and  $\hat{y}_2$  are respectively the predicted labels of  $f_1$  and  $f_2$ . Though  $f_1$  and  $f_2$  separately make predictions, the samples with two identical predicted labels are promoted as training data, while others with different results are discarded. One advantage of the annotation method is that the hyperparameter  $P_{thresh}$  need not be artificially selected, which reduces the impact of human intervention, and low-confidence samples are fully used to improve the generalizability of the evaluators.

#### 4.4. A discriminative evaluation for distinguishing the confusion classes

Classes with similar texture structures and visualized-feature representation are easily confused with each other during the labeling process. Additionally, the co-training-based self-label method makes it difficult to eliminate incorrect samples when using low-confidence samples. Encouraging the samples of such classes further contributes to semantic drift. In order to prevent this problem, we employ a discriminative evaluation based on the fact that confusion classes are easily collected from a confusion matrix over a validation dataset. The confusion information from the confusion matrix is used to train a set of one-vs-one classifiers, which could discriminate pairs of the confusion classes. Specifically, a pair of classes  $i_{th}$  and  $j_{th}$  are defined as the confusion classes:

$$\begin{cases} C(i,j) > C_{thresh} \\ \text{or} \\ C(j,i) > C_{thresh} \end{cases} \quad i \neq j, \quad \forall i, j \in \{1, 2, \dots, N\} \quad (10)$$



**Fig. 4.** Illustration of the co-training-based self-label method.

where  $C(i,j)$  is the ratio of the results that class  $i_{th}$  is wrongly predicted as class  $j_{th}$ , and  $C_{thresh}$  is a hyperparameter, which is used to determine the confusion classes.

For these confusion classes, we employ a discriminative evaluator  $g_{ij}$  to enforce discriminative constraints.  $g_{ij}$  is defined as a binary classifier, which is used only to discriminate the  $i_{th}$  and  $j_{th}$  class samples. Accordingly, it is only trained on the samples with categories of  $i_{th}$  and  $j_{th}$ . For given the independent evaluator's prediction function  $f$ , an instance  $I' = \langle x' \rangle, I' \in U$  is labeled as training data only if

$$f(x') = g_{ij}(x'), \quad \forall f(x') \in (i,j) \quad (11)$$

If an unlabeled sample is predicted as  $i_{th}$  or  $j_{th}$ , it would be discriminated by  $g_{ij}$ . Only the samples that are agreed upon by the discriminative evaluator  $g_{ij}$  and the independent evaluator  $f$  are promoted as training data. By employing a discriminative evaluator, annotating samples becomes more accurate, and the cases of misclassification are effectively reduced.

We expand two classes to multiple classes. A set of *one-vs-one* classifiers are built to match every pair of confusion classes. They are initialized in the same way and use one common CNN together to extract high-level features. Each discriminator would predict a sample if its condition is satisfied. Specially, when a sample involves two discriminators, the predicted results should be consistent for promoting it as a training sample. Therefore,  $C_{thresh}$  is an important parameter. A small threshold would identify more pairs of confused classes, at the cost of more pairwise discriminative evaluators. This is discussed further in the experimental section.

#### 4.5. A semi-supervised generative framework for HRRS scene classification

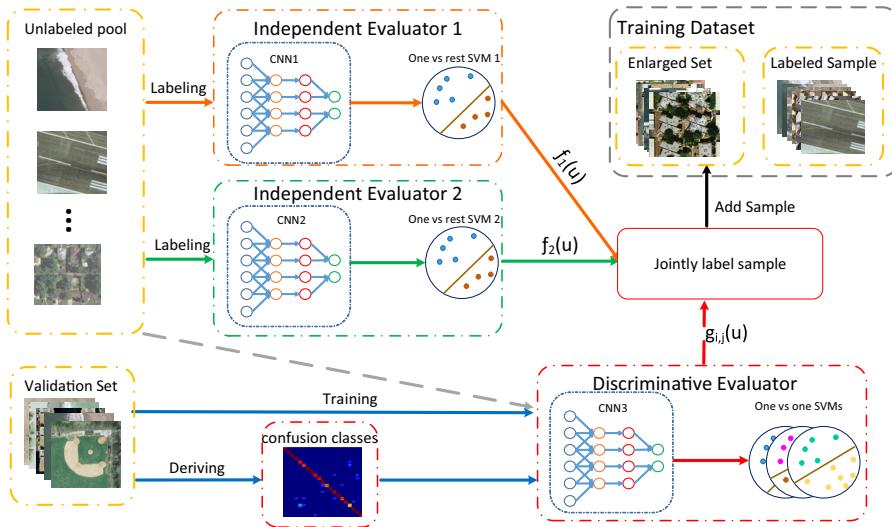
In this section, we propose a semi-supervised generative framework (SSGF) for HRRS image scene classification by combining the three techniques mentioned above. As shown in Fig. 5, two independent classifiers and one discriminative evaluator are involved. The independent classifiers, each of which is constructed by a deep feature extractor CNN and a *one vs rest* SVM classifier, jointly label a sample in the manner of co-training. In order to make the co-training evaluator more effective and robust, the features generated from two kinds CNNs should be very different in structure and depth to form complementary information. This setting is beneficial for evaluating scene classes in variant feature views and enhancing the model's classification capability. Therefore, it is an important question on how the available network is chosen, and we introduce our settings in Section 4.6. The SVM classifiers of independent evaluators are the same. The discriminative evaluator is used to enhance classification of the confusion classes, which

consist of a deep feature extractor and a set of *one vs one* SVM classifiers. In this framework, two independent classifiers are initialized by training on the labeled samples. In order to obtain the information of confusion classes, we test the classification results of the two independent classifiers on the validation set. The information of the confusion classes is derived from the confusion matrix based on the classification results. Therefore, the discriminative classifiers are accessed according to the information of the confusion classes by trained on the corresponding part of the validation set. Next, a batch of unlabeled examples are fed to the CNNs of independent evaluators to generate high-level features. Then the independent classifiers together determine the predicted labels according to the high-level features. Additionally, the discriminative classifiers jointly label a sample, only if the classification labels of two independent classifiers belong to the confusion classes. The instances with the same predicted labels are promoted as training examples, which are emerged into EL. As a result, all samples of unlabeled set are annotated during the process of iterative learning. In order to evaluate the performance of the proposed SSGF, we develop an extended algorithm.

#### 4.6. An extended algorithm

On the basis of the SSGF, we develop an extended algorithm by analyzing different architectures of deep CNNs. Many works have demonstrated that different architecture networks learn variant feature maps, while some networks are more receptive to texture others are more receptive to color information. Because they are designed for different applications and each of them differs with features like kernel size, activation function, depth degree, loss function, and amount of hidden units. Accordingly, the features generated by different deep CNNs may compose complementary information for scene classification.

As mentioned above, the two independent evaluators are used to jointly label samples and improve classification capability while learning the unlabeled samples. Therefore, the features generated from two very different networks are beneficial for evaluating scene classes in variant feature views, which helps to enhance the classification capability of the model. ResNet-50 has the greatest depth and the best classification performance, but keeps fewer parameters and less complexity than VGG-16 and VGG-19. Hence, we set ResNet-50 as the feature extractor. VGG-S is an eight-layer network, with  $7 \times 7$  filters in convolutional layers, quite different from the  $3 \times 3$  filters of ResNet. It is the modified version of the well-known AlexNet. Due to the differences in the structure of ResNet, VGG-S is set as the other feature extractor. Diversely, the discriminative evaluator is used to enforce classifying confusion samples and is not improved. We select features provided by VGG-16 for discriminative evaluation because it maintains a simi-



**Fig. 5.** Illustration of the proposed SSGF.

lar architecture and achieves comparative performance with VGG-19, but keeps fewer parameters.

#### Algorithm 2. Description of the proposed SSGA-E

```

1: Input: labeled set  $L$ , unlabeled set  $U$ , validation set  $V$ ,  
batch size  $batch\_size$   
2: Output: enlarge set  $EL$   
3: initialize  $EL, LC$  as empty,  $count = 0$   
4:  $clf_1, clf_2 \leftarrow multi\_training(L)$   $\triangleright$ the independent  
classifiers are initially trained on  $L$   
5:  $conf\_class \leftarrow derive\_conf\_class(V, clf_1, clf_2)$   $\triangleright$ derives the  
confusion classes  
6:  $g \leftarrow (V, conf\_class)$   $\triangleright$ discriminative classifier  $g$  is  
trained on the confusion classes  
7:  $batch\_set \leftarrow return\_batch\_set(U, batch\_size, count)$   
8:  $co\_labeling(clf_1, clf_2, g, batch\_set, EL, LC)$   $\triangleright$ in  $co\_labeling$ ,  
two independent evaluators make predictions  
9: while  $batch\_size * count < len(U)$  do  $\triangleright$ processes the  
whole unlabeled set  
10:    $count \leftarrow count + 1$   
11:    $clf_1, clf_2 \leftarrow update\_multi\_training(clf_1, clf_2, EL + L)$   
 $\triangleright$ updates the independent classifiers  
12:    $batch\_set \leftarrow return\_batch\_set(U, batch\_size, count)$   
13:    $co\_labeling(clf_1, clf_2, g, batch\_set, EL, LC)$   
14: end while  
15: while  $len(EL)$  increases do  $\triangleright$ processes the  $LC$  set until  
the sample number of  $EL$  doesn't change  
16:    $clf_1, clf_2 \leftarrow update\_multi\_training(clf_1, clf_2, EL + L)$   
17:    $batch\_set \leftarrow LC$   
18:    $empty(LC)$   
19:    $co\_labeling(clf_1, clf_2, g, batch\_set, EL, LC)$   
20: end while  
21: Return  $EL$ 
```

We propose an extended algorithm on the basis of the SSGF, the extended semi-supervised generative algorithm (SSGA-E), as follows based on the SSGF. ResNet, VGG-S are set as the feature extractors, and each of them forms an independent evaluator with a *one vs rest* linear SVM. VGG-16 with a set of *one vs one* SVMs constructs the discriminative evaluators. The process is displayed in algorithm 2.

In algorithm 2, the initial parameters are similar to Algorithm 1. Then, two independent classifiers  $clf_1, clf_2$  are initially trained on the labeled set  $L$ . The  $clf_1$  and  $clf_2$  form the co-training-based evaluator. We derive the information of confusion classes by testing the co-training-based evaluator over the validation dataset (line 5 in Algorithm 2). Hence, the discriminative evaluators  $g$  are developed on the confusion classes from the validation set. A set of samples from  $U$  are labeled by the co-training-based evaluator and the discriminative evaluator  $g$ . Those samples agreed by all evaluators are merged into  $EL$  as new training data. In the next loop, the co-training-based evaluator is updated by training on  $EL$  and  $L$ , then continuously finish the labeling task. The unlabeled samples are learned in this iterative manner (line 9–14 in Algorithm 2). The low-confidence set is continuously learned until the number of samples in  $EL$  does not increase (line 15–20 in Algorithm 2). In order to evaluate the proposed method, we have performed extensive experiments over four public HRRS image datasets in the experimental section.

## 5. Experimental results

In this section, we detail the series of experiments conducted to evaluate the performance of the proposed SSGA-E for annotation of remote sensing images over four HRRS image datasets. The detailed experimental setup and experimental results with reasonable analysis are presented below.

### 5.1. Experiment setup

Four public remote sensing datasets are used in the experimental section. The notable UCMerced Land Use dataset (Yang and Newsam, 2010) (UCM for short), includes aerial optical images with low-level characteristics similar to those of the Imagenet. In recent years, many researchers used this dataset to perform an extensive comparison of results. WHU-RS19 dataset (Xia et al., 2010) is collected from Google Earth and is a new public available dataset. NWPU-RESISC45 (Cheng et al., 2017) is a very large scale benchmark for remote sensing scene classification, and was created by Northwestern Polytechnical University (NWPU). Since it is accessible only as of recently, limited results are available. The last dataset is Aerial Image Dataset (AID)(Xia et al., 2017). Since samples in AID are from different sensors and contains various pixel resolutions, it becomes a challenge for scene classification.

All experiments were carried out on a work bench equipped with an Intel CPU i7 5820 k, an NVIDIA Geforce Maxwell Titan X GPU, and 32 GB DDR4 memory. Operation system is Ubuntu 16.10. The CNNs used includes CaffeNet, GoogLeNet, VGG-F, VGG-S, VGG-M, VGG-16, VGG-19, and ResNet-50, which are pre-trained on the ImageNet dataset (Deng et al., 2009).<sup>1</sup> During feature extraction, the images were resized to meet the input were of each CNN ( $224 \times 224$  or  $227 \times 227$ ). For fair comparison, the labeling image was carried out using linear support vector machines (SVMs), which are implemented by Scikit-learn (Pedregosa et al., 2011). We used the default setting in linear SVM ( $C = 1$ ). To make the experimental descriptions clearer, dataset usage and classifier settings are respectively presented in the different experimental stages. All experiments were performed through fivefold validation, by averaging over the five folds. Data augmentation was not used.

Overall accuracy and confusion matrix are widely-available standard evaluation metrics in image classification. Overall accuracy is calculated as the number of correctly classified samples divided by the total number of samples. The confusion matrix is an informative table showing all the errors and confusions between the different classes, and is gained by collecting each type of correct and incorrect classification of the test samples and accumulating the results. The metrics of overall accuracy and the confusion matrix are used to analyze the experimental results.

In the SSGA-E, ResNet and VGG-S are the feature extractor, and each of them respectively forms an independent evaluator with a *one vs rest* SVM. VGG-16 with a series of *one vs one* SVMs constructs the discriminative evaluators. We carried two independent experiments to evaluate our algorithm for each dataset in Sections 5.2.3, 4, 5.5. In the first experiment, the classification capability of the proposed method is evaluated. The comparison algorithms are the eight kinds of supervised methods, and three semi-supervised methods, including two self-training methods and one co-training based method. The supervised methods are eight different kinds of CNNs with the SVM classifier. For the supervised methods, the dataset is split into 60% training set, 20% validation set, and 20% test set. In order to compare with the semi-supervised methods, we added the test stage for the supervised methods, where only 10% training set is used, while the validation set and test set remain consistent. In the self-training methods, ResNet or VGG-S and an SVM form the evaluator to label samples. It should be noted that the co-training method here is distinct to traditional definition of co-training method. And it is regarded as the SSGA-E without the discriminative evaluator, wherein two independent evaluators jointly label samples, and they are trained on the same datasets and generative samples. For all semi-supervised methods, including SSGA-E, self-training, and co-training methods, the dataset is split into 10% for labeled training, 20% for validation of the independent evaluators, 50% for unlabeled training, and the remaining 20% for the test. Batch size *batch\_size* is set to 20% of the unlabeled set in UCM and WHU-RS datasets, and 10% in NWPU-RESISC45 and the AID datasets.  $C_{thresh}$  was set to 10%. In the second experiment, the impact of the number of unlabeled samples was evaluated. The classification results of the co-training and SSGA-E are compared. The settings of the labeled training set, validation set and test set are the same as the first experiment. We increased the ratio of unlabeled samples from 10% to 50% at 10% intervals. Taking the UCM dataset as an example, the sample number for each category is 100. As the number of unlabeled samples of each class increased 10 to 50 at intervals of 10, changes of classification accuracy are observed.

## 5.2. Experiment (1): UCM dataset

The first dataset is UCM, which was extracted from large optical images (RGB color space) from the US Geological Survey, taken over various regions of the United States. 2100 images of  $256 \times 256$  pixels were selected and manually labeled as belonging to 21 land use classes, 100 for each class. Fig. 6 shows one example image for each class. Due to their nature and relatively high resolution (30 cm), these images share many low-level features with general-purpose optical images, which makes features extracted by the CNNs pre-trained on ImageNet efficient for scene classification of the UCM dataset.

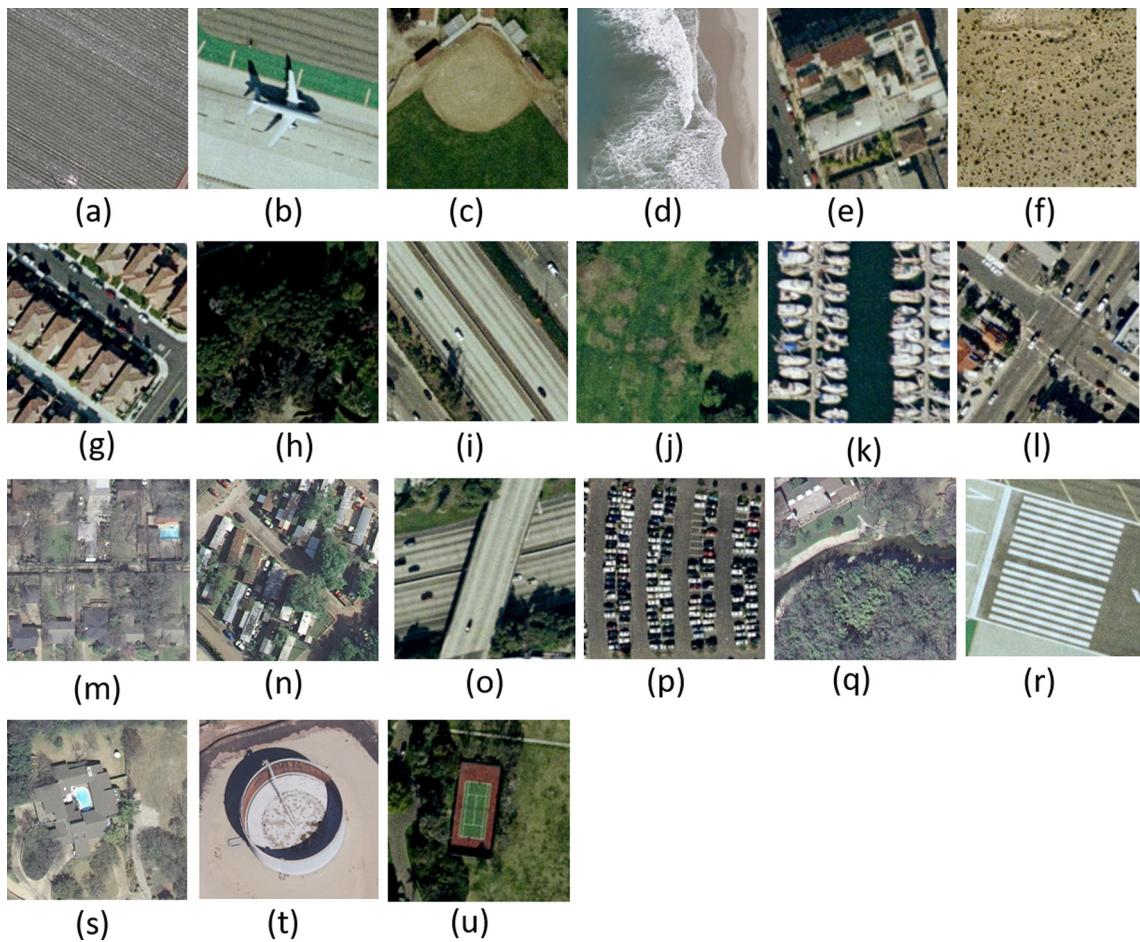
We carried out the first experiment to evaluate the classification capability of the SSGA-E. Table 1 shows the overall accuracy of the proposed method, and the comparison methods and Fig. 7 illustrates the accuracy curve of the SSGA-E for the validation set and test set. The proposed method significantly improves classification capability. It achieves a better classification accuracy than all the supervised methods with 10% training set and the third highest classification accuracy compared with the supervised method with 60% training set, only inferior to ResNet and VGG-S. For the semi-supervised methods, co-training outperforms the self-training methods by about 2.18%. The results prove that features learned by different networks are complementary and beneficial to enhance the model's classification capability. The SSGA-E is better than the co-training method by about 0.77%. It demonstrates that the discriminative evaluators are available to improve the recognition ability of SSGA-E. In Fig. 7, the SSGA-E continuously improves the performance of the classification as the number of iterations increases. The accuracy results of the validation set are better test set because the discriminative evaluators train on the confusion classes of the validation set. Therefore, it is reasonable that the discriminative capability of SSGA-E on validation set is better than that on the test. Fig. 8 shows the confusion matrices, where the entry in the  $i_{th}$  row and  $j_{th}$  column denotes the rate of the test samples from the  $i$ th class that are classified as the  $j$ th class. Classes with similar visual features may be classified to an incorrect category. Incorrect classification is significantly reduced after the entire unlabeled set is learned, such as the accuracy of storage tanks and buildings increases from 70% to 90%. The confusion matrix proves that the proposed method is superior to recognize the confusion classes.

Next, we evaluate the impact of the number of unlabeled samples. The experimental results of the SSGA-E and co-training methods are compared in Table 2. The accuracy of co-training is improved from 89.75% to 93.75% and the accuracy of SSGA-E is improved from 91.42% to 94.52%. The two algorithms are able to improve recognition ability by learning from unlabeled data. When the ratio of the unlabeled set is 10%, the SSGA-E outperforms co-training by about 1.77%. The difference is 0.78% when the ratio of unlabeled set is 50%. The results prove that the discriminative evaluator is able to provide more information to improve recognition ability. As the number of unlabeled samples increases, the positive impact of the discriminative evaluator reduces. In addition, the size of the unlabeled dataset significantly affects the performance of the SSGA-E.

## 5.3. Experiment (2): WHU-RS19 dataset

The second dataset for experiment is the WHU-RS dataset (Xia et al., 2010). Collected from Google Earth, it was accessed via free public download, and consists of 950 images,  $600 \times 600$  pixels each, distributed across 19 scene classes. Some examples are shown in Fig. 9. Compared with the UCM dataset, WHU-RS is more complicated, such as in the variation of illumination, scale, resolu-

<sup>1</sup> Pre-trained CNN models are available at <https://github.com/BVLC/caffe/wiki/Model-Zoo>.



**Fig. 6.** The UCM land-use dataset. (a)–(u) Agricultural, airplane, baseball diamond, beach, buildings, chaparral, dense residential, forest, freeway, golf course, harbor, intersection, medium residential, mobile home park, overpass, parking lot, river, runway, sparse residential, storage tanks, and tennis courts, respectively.

**Table 1**  
The classification results of all methods on the UCM dataset.

Method name	Model name	Classification accuracy(%)	
		10% training set	60% training set
Supervised learning	CaffeNet	84.56 ± 0.92	93.9 ± 0.62
	GoogLeNet	83.64 ± 2.29	93.15 ± 0.89
	VGG-F	85.61 ± 1.06	94.14 ± 1.53
	VGG-S	85.46 ± 0.7	94.85 ± 0.57
	VGG-M	85.14 ± 0.85	93.95 ± 0.81
	VGG-16	84.19 ± 2.05	94.09 ± 1.48
	VGG-19	84.52 ± 1.19	93.65 ± 0.64
	ResNet	<b>88.9 ± 2.48</b>	<b>95.52 ± 0.228</b>
Self-training	VGG-S	<b>86.14 ± 1.87</b>	
	ResNet	<b>91.57 ± 2.00</b>	
Co-training	ResNet&VGG-S	<b>93.75 ± 1.42</b>	
SSGA-E	ResNet&VGG-S&VGG-16	<b>94.52 ± 1.38</b>	

tion, viewpoint, and viewpoint-dependent appearance in some categories. These all make the WHU-RS more challenging than UCM dataset.

**Fig. 10** and **Table 3** show the overall classification results of the proposed method and the comparison methods. ResNet-50 achieves the best accuracy of 98% with 60% training set and 92.1% with 10% training set, and GoogLeNet achieves the worst results of 93.41% and 83.68% in the same categories, respectively. The average accuracy for WHU-RS19 is higher than for UCM. The

SSGA-E, achieving the second best precision of 96.84%, outperforms CaffeNet, GoogLeNet and VGG on the 60% training set and all supervised methods on the 10% training set. For the semi-supervised methods, the co-training method achieves accuracy of 96.41%, better than self-training methods. The differences of accuracy between co-training and SSGA-E on WHU-RS19 is less than the results on the UCM dataset. This could be because WHU-RS19 contains highly sufficient between-class diversity, which makes categorization simpler. In **Fig. 10**, the initial accuracy of the proposed method is 92% and rises with the iteration number. At the last iterative round, the proposed method achieves the best accuracy of 96.84% on the test set, and the accuracy on validation test was higher than 97%. In **Fig. 11**, the wrong classification is significantly reduced after carrying out iterative learning. The classification results of forest, industrial, and meadow are entirely correct. The experimental results demonstrate once again that the SSGF is an effective framework to improve classification accuracy by extracting valuable information from unlabeled sets and obtaining an available annotation dataset.

The impact of the number of the unlabeled samples on the WHU-RS19 dataset is evaluated in the second experiment. The experimental results are shown in **Table 4**. With the 10% unlabeled training set, co-training and SSGA-E reached high levels of accuracy at 93.68% and 94.76%, respectively. The accuracy of co-training is improved from 93.68% to 96.41% and the accuracy of SSGA-E is promoted from 94.42% to 96.84%. The SSGA-E outperforms co-training by about 1.08% with the ratio of the unlabeled set at 10%. The difference drops to 0.43% when the ratio is set at

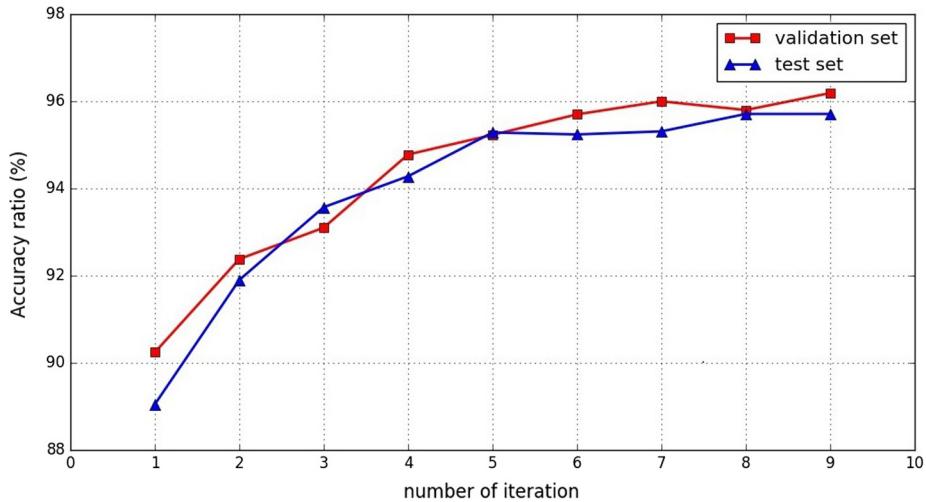


Fig. 7. The accuracy results of the proposed method on the UCM dataset.

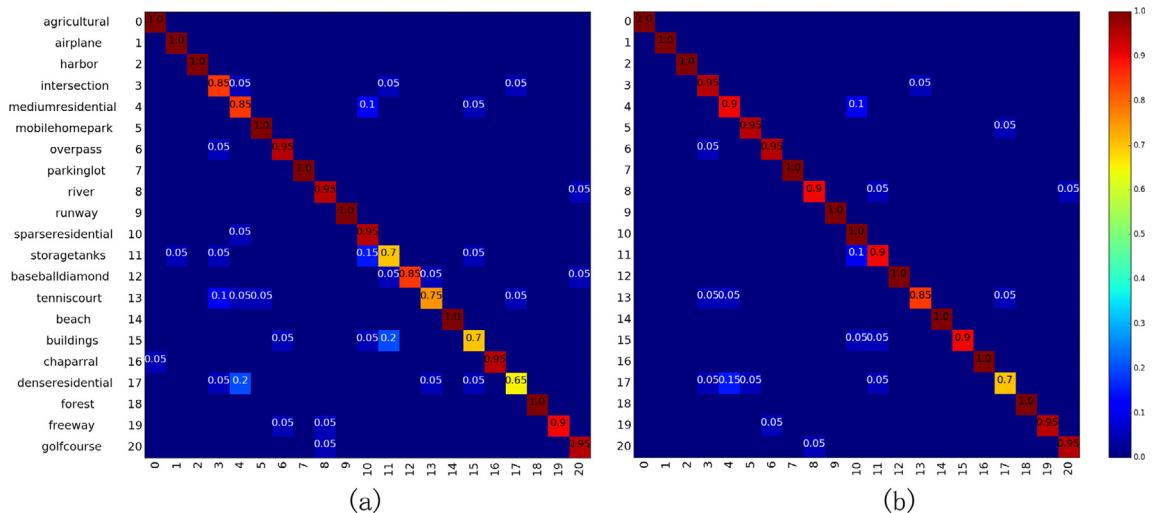


Fig. 8. The confusion matrices on the test set of the UCM dataset. (a) Confusion matrix information before learning the unlabeled set and (b) confusion matrix after learning the unlabeled set.

**Table 2**

The effect of the unlabeled sample ratio on accuracy for UCM dataset.

Method	Classification accuracy (%)				
	10%	20%	30%	40%	50%
Co-training	89.75 ± 1.27	91.62 ± 0.93	92.58 ± 0.78	93.42 ± 1.32	93.75 ± 1.42
SSGA-E	91.42 ± 0.95	92.68 ± 0.87	93.56 ± 1.42	94.21 ± 1.18	94.52 ± 1.38

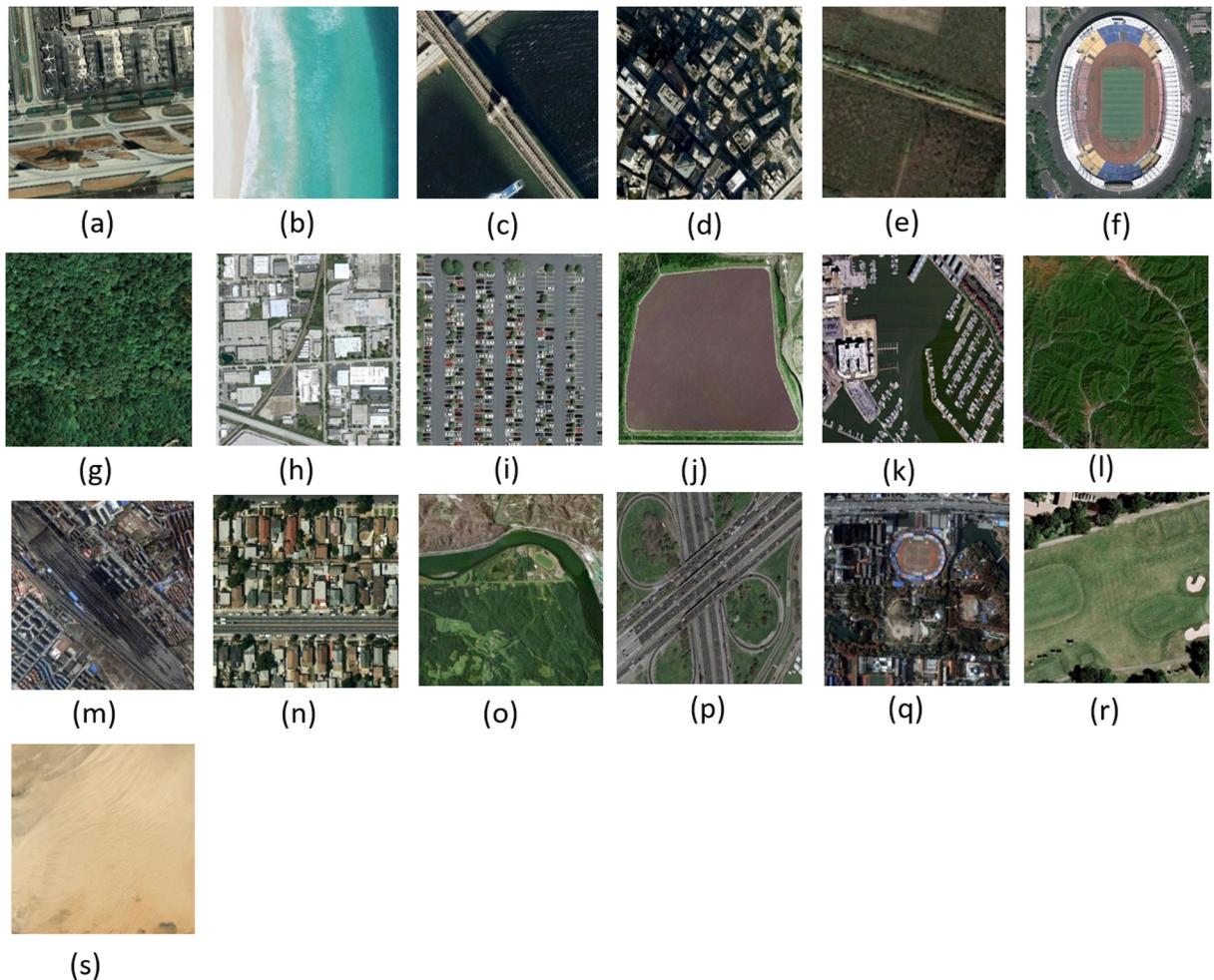
50%. This proves that as the number of unlabeled samples increases, the SSGA-E achieves better classification ability. (see Figs. 12 and 13).

#### 5.4. Experiment (3): NWPU-RESISC45 dataset

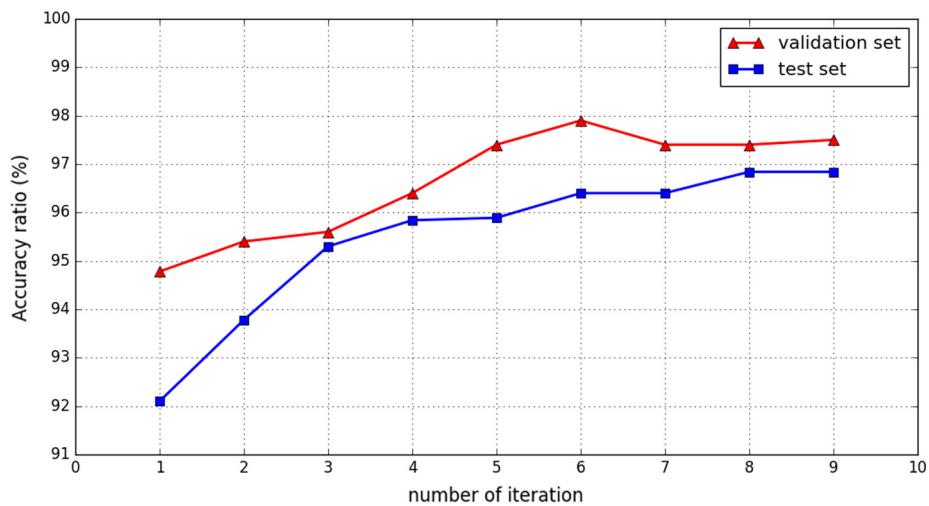
Then the proposed method is evaluated on two large scale datasets. The third dataset is the NWPU-RESISC45, which keeps 31,500 remote sensing images divided into 45 scene classes. Each scene class contains 700 images, each of which is set at 256 × 256 pixels in the RGB color space. The dataset was extracted by the specialists of the remote sensing field from Google Earth with spatial resolution ranges from about 30 m to 0.2 m per pixel. The NWPU-

RESISC45 is a large scale dataset of scene classes. It maintains a lot of variations in translation, viewpoint, object pose, spatial resolution, illumination, background, and occlusion. Accordingly, it contains high sufficiency between class similarity and within-class diversity, which determines that it is beneficial to develop and evaluate various data-driven algorithms. Due to the length of this paper, we only carried our the first experiment to evaluate the performance of SSGA-E on this dataset.

Due to the extensive scale of NWPU-RESISC45, we reduced the ratio of batch size to 10% of the unlabeled dataset. Correspondingly, number of the iterations increased. Figs. 15 and 14, and Table 5 show all experimental results. In Table 5, unlike the previous experiments where the classification accuracy easily reached



**Fig. 9.** The WHU-RS19 dataset. (a)–(s) Airport, beach, bridge, commercial area, farmland, football field, forest, industrial, parking lot, pond, port, mountain, railway, residential area, river, viaduct, meadow, desert.



**Fig. 10.** The classification accuracy results of the proposed method on the WHU-RS19 dataset.

95%, only the precision of the supervised method of ResNet-50 is higher than 90% with the labeled training set at 60%. The sufficient diversities and variations of this dataset is a challenge for the deep learning methods to achieve a desired accuracy. The proposed method ultimately reaches the second best classification result

on the test set with an average accuracy of 88.6%, which outperforms the total supervised methods, with the exception of ResNet and the semi-supervised methods. In Fig. 15, the curves of classification results are close to the previous experiments, and gradually ascends as the number of iteration rises. After the number of

**Table 3**

The classification results of all methods on the WHU-RS19 dataset.

Method name	Model name	Classification Accuracy (%)	
		10% training set	60% training set
Supervised learning	CaffeNet	85.91 ± 1.17	94.73 ± 1.88
	GoogLeNet	83.68 ± 2.44	93.41 ± 1.90
	VGG-F	87.63 ± 1.00	95.26 ± 2.11
	VGG-S	88.41 ± 1.85	95.65 ± 1.19
	VGG-M	87.10 ± 1.30	96.1 ± 1.27
	VGG-16	84.60 ± 1.12	95.13 ± 1.45
	VGG-19	85.00 ± 2.44	95.42 ± 1.00
Self-training	ResNet	<b>92.10 ± 2.06</b>	<b>98.00 ± 0.63</b>
	VGG-S	<b>90.62 ± 1.80</b>	
	ResNet	<b>94.68 ± 1.73</b>	
Co-training	ResNet&VGG-S	<b>96.41 ± 1.03</b>	
SSGA-E	ResNet&VGG-S&VGG-16	<b>96.84 ± 1.33</b>	

iteration steps reaches 10, the accuracy tends to remain stable. Finally, accuracy on the validation set reaches 88.95% and that on the test set reaches 88.6%. In Fig. 14, the 45 different scene classes form a large confusion matrix, wherein the problem of incorrectly labeling samples occurs more frequently. The pairs of confusion classes increases significantly. By comparing Fig. 14(a) and (b), the ratio of misclassification for almost all scene classes is reduced. It shows that the proposed method may be used on large scale datasets.

### 5.5. Experiment (4): Aerial image dataset

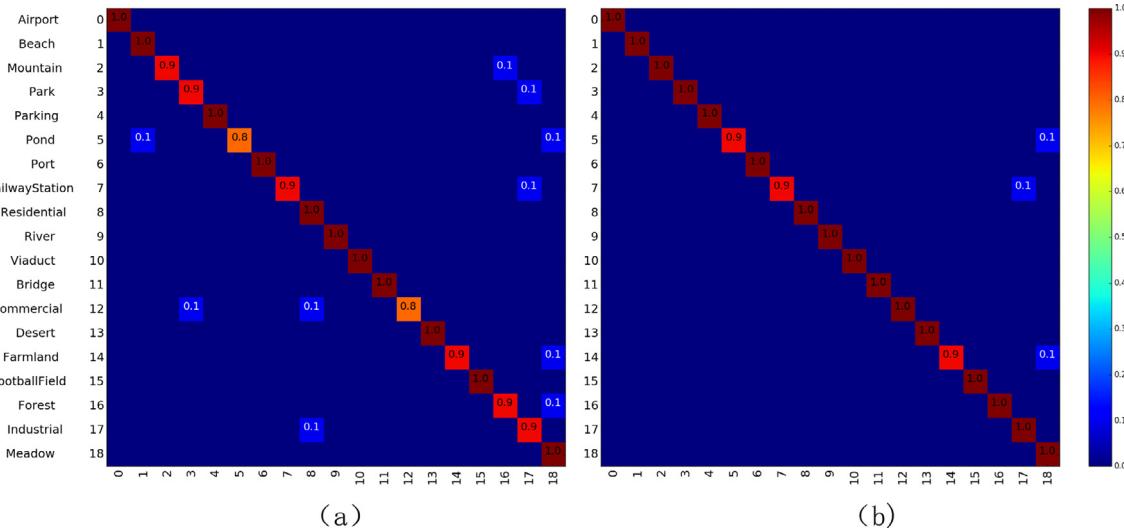
The last dataset is the Aerial Image Dataset (AID) (Xia et al., 2017), which is a large scale dataset for aerial scene classification.

It contains 10,000 annotated aerial images with a fixed resolution of 600 × 600 pixels within 30 classes. And number of samples per class is a lot various from 220 up to 420. As samples in AID are collected from different remote sensing sensors, the samples are multisource, which leads to more challenges than the single source images, such as UCM dataset. Another main difference is that AID has multiresolutions: the pixel resolution changes from 8 m to about 0.5 m, and the size of each images is fixed to cover a scene category with different resolutions.

Due to the scale of AID dataset, we kept the ratio of batch size to 10% of unlabeled dataset as the setting in NUPW-RESISC45. Figs. 16 and 17 show complete experimental results. In Table 6, the overall accuracies in AID are higher than the results in NUPW-RESISC45. Different from previous results for supervised methods, VGG-19 implements the worst precision of 79.2% with 10% labeled training set. ResNet achieves the best accuracy of 85.71% with 10% labeled set and 93.31% with 60% labeled set. The proposed SSGA-E outperforms other semi-supervised comparison algorithms and reaches the second best classification accuracy in all results. In Fig. 16, the precision curves of SSGA-E on validation set and test set gradually rise with the increase of iteration number. Finally, accuracy reaches 92.65% on the validation set and 91.35% on the test set. In Fig. 17, a large confusion matrix formed by 30 difference remote sensing scene categories shows the classification results for each class. By comparing Fig. 17(a) and (b), the results of the error classification is effectively reduced. The experimental results once again prove that our method can be used in large-scale datasets.

### 5.6. Sensitivity analysis and confidence discussion

We discuss the impacts of *batch\_size* and *C<sub>thresh</sub>* on the performance of the SSGA-E, as well as discuss the confidence level of generative samples in this section. In order to further analyze the



**Fig. 11.** The confusion matrices on the test set of the WHU-RS dataset. (a) Confusion matrix before learning the unlabeled set and (b) confusion matrix after learning the unlabeled set.

**Table 4**

The effect of the unlabeled sample ratio on accuracy for WHU-RS19 dataset.

Method	Classification accuracy (%)				
	10%	20%	30%	40%	50%
Co-training	93.68 ± 1.03	95.25 ± 1.59	95.83 ± 1.57	96.15 ± 0.82	96.41 ± 1.03
SSGA-E	94.76 ± 1.56	95.78 ± 1.12	96.31 ± 0.76	96.56 ± 0.85	96.84 ± 1.33

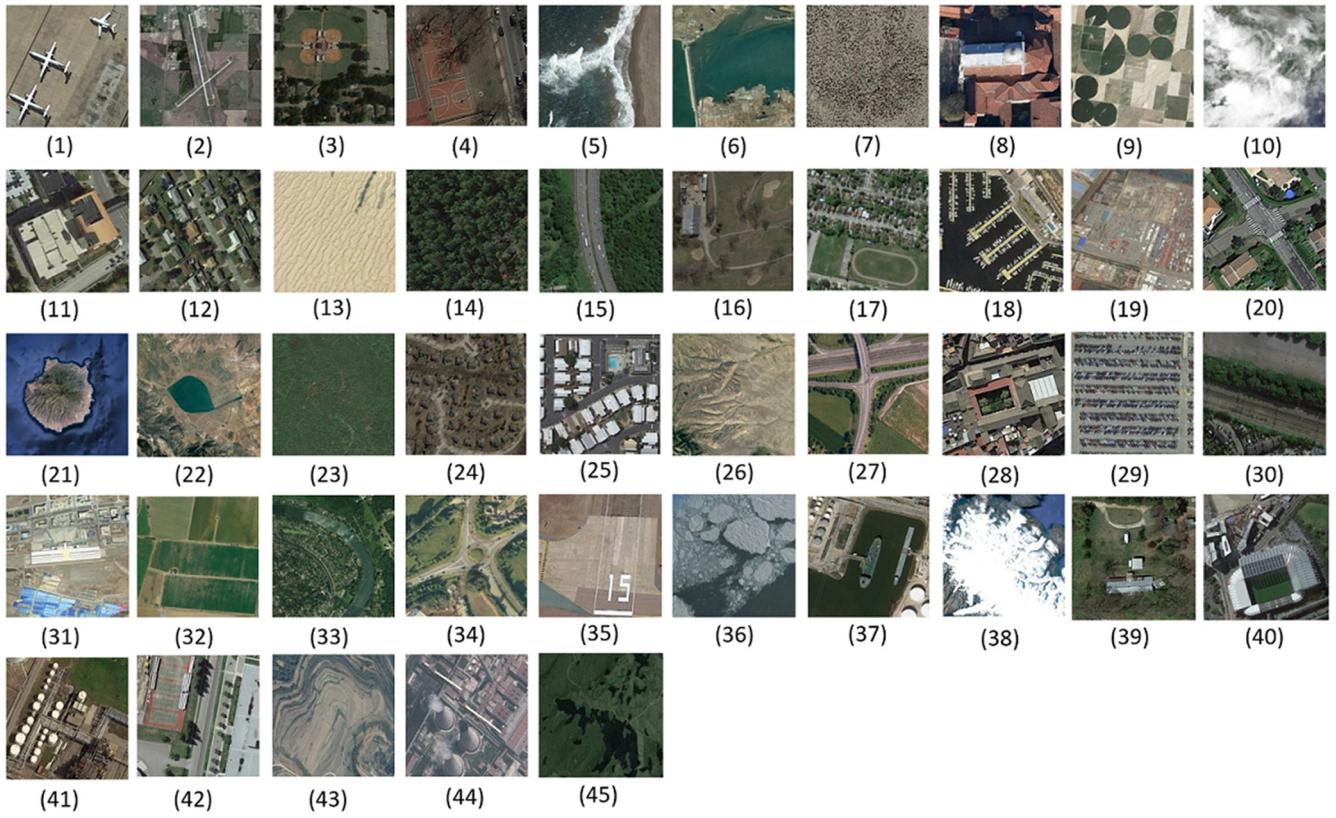
effect of discriminator on individual categories, additional experiment is also performed.

### 5.6.1. Batch size: *batch\_size*

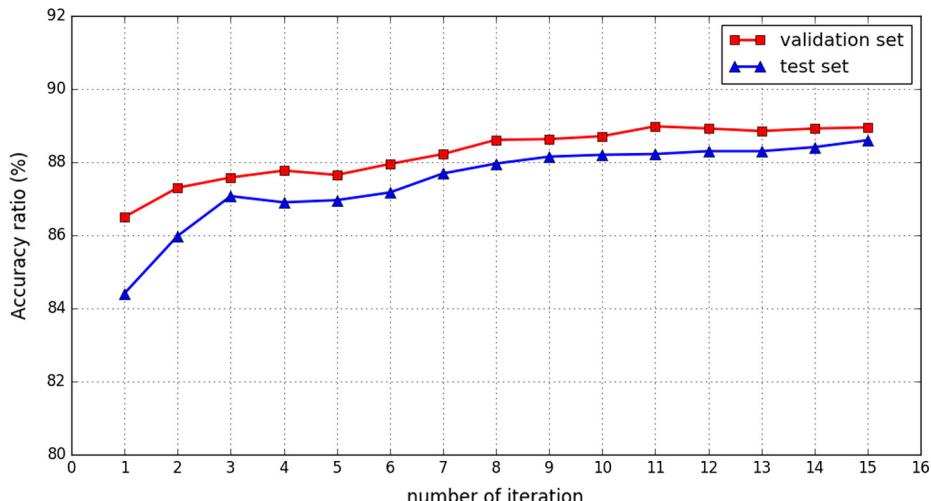
The batch size *batch\_size* is an important parameter in the SSFA-E algorithm, which is the ratio of unlabeled sets to be processed in each iterative round. Batch size can effect performance as well as computation time. To better evaluate the impact of *batch\_size*, we have analyzed its selection on the UCM dataset. All experimental results are the average of five rounds of calculations. The classifica-

tion accuracy results obtained with different *batch\_size* are shown in Fig. 18.

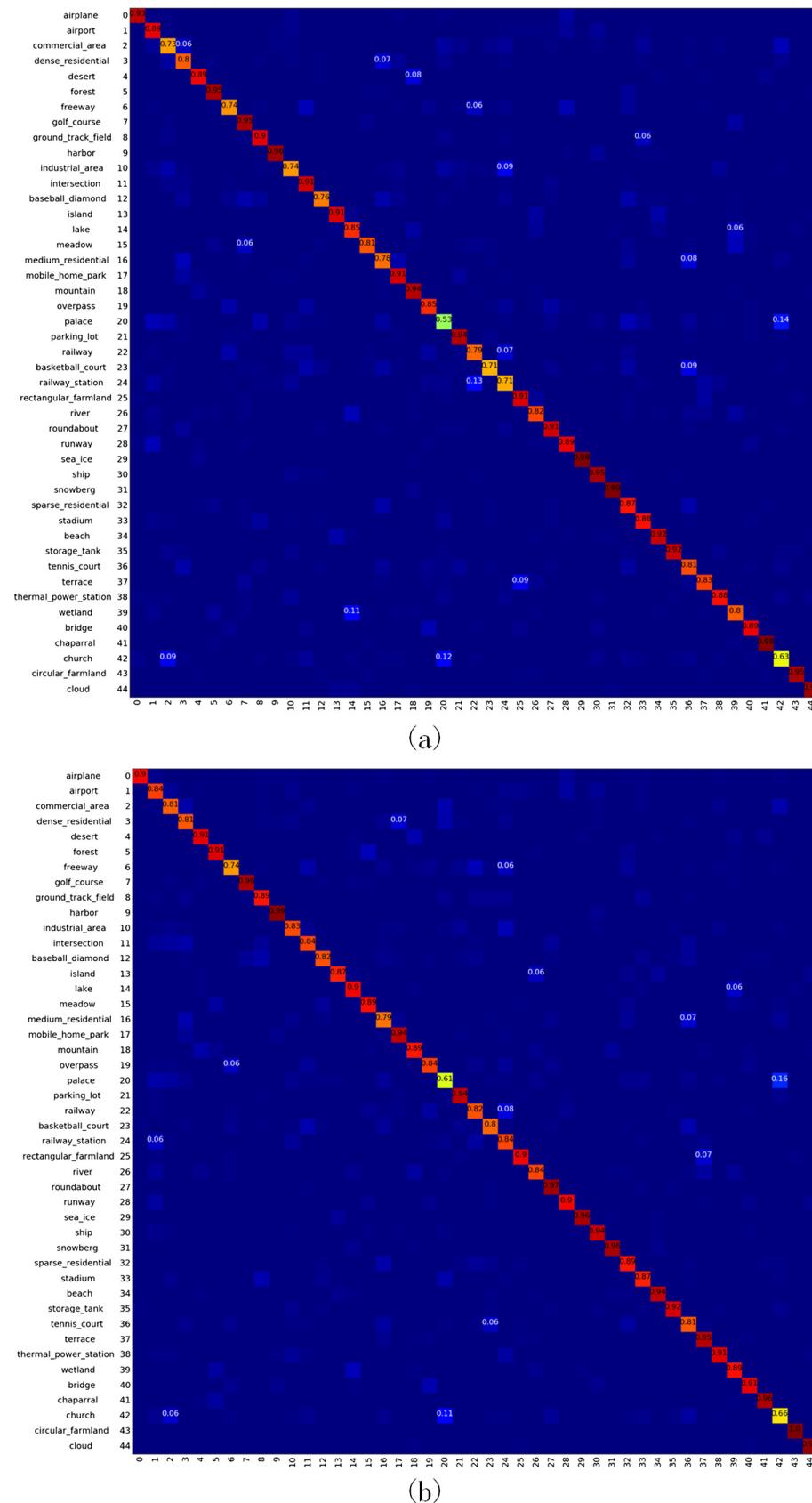
It can be observed from Fig. 18 that accuracy is the highest compared with the other values when the size is set to 5% and 10%. The algorithm with a *batch\_size* ratio of 20% researches a similar accuracy. The accuracy of the algorithm gradually decreases both on validation and test set as the ratio of *batch\_size* rises. When the *batch\_size* is set to 50% and 100%, the algorithm shows poor performance. Accordingly, a small value of *batch\_size* is beneficial for the SSGF to achieve a good performance, it results in an increase in the



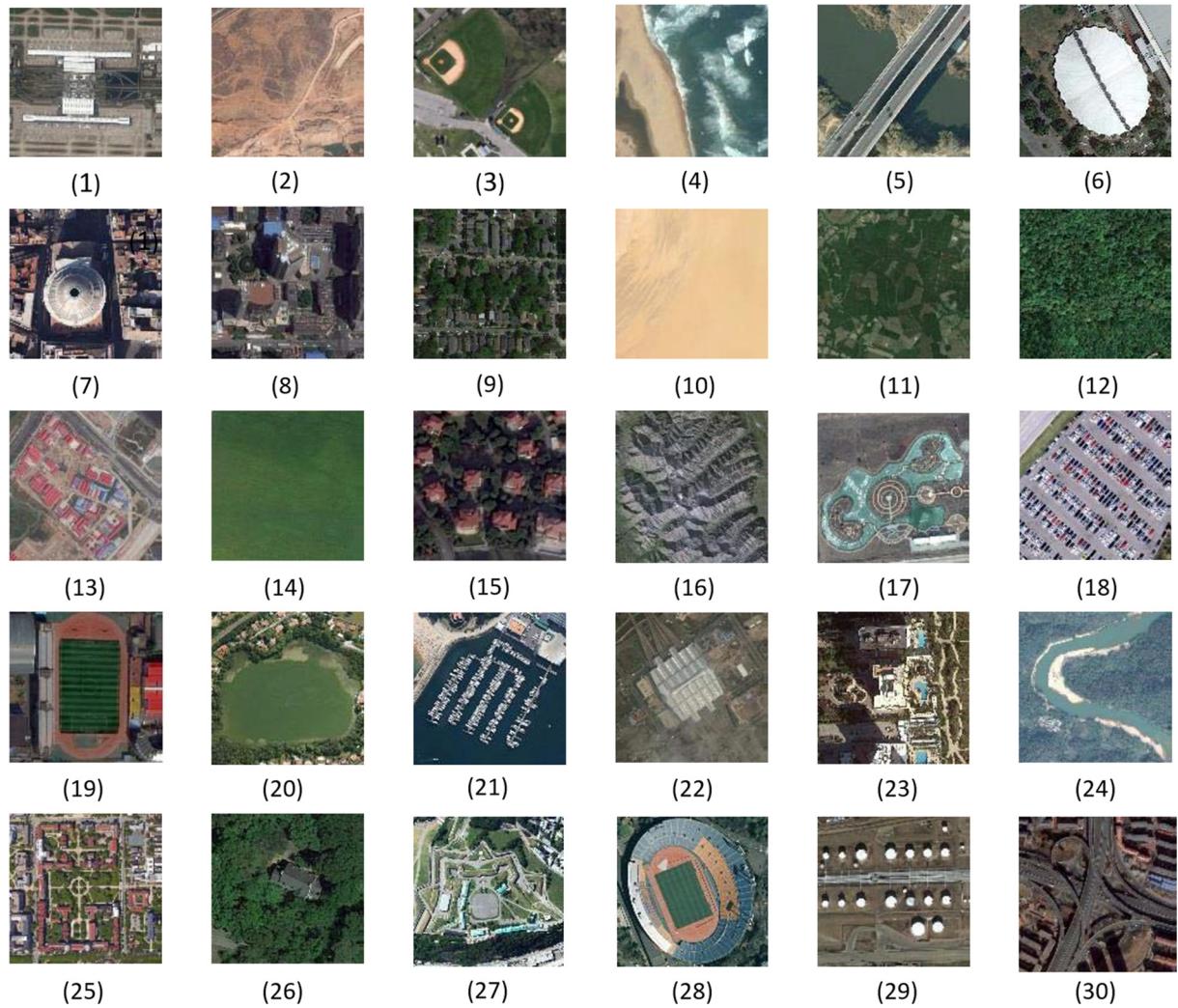
**Fig. 12.** Some examples from the NWPU-RESISC45 dataset. (1)–(45) Airplane, airport, baseball diamond, basketball court, beach, bridge, chaparral, church, circular farmland, cloud, commercial area, dense residential, desert, forest, freeway, golf course, ground track field, harbor, industrial area, 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, snowberg, sparse residential, stadium, storage tank, tennis court, terrace, thermal power station, wetland.



**Fig. 13.** The classification accuracy curves of the proposed SSGA-E respectively on the validation set and the test set for the NWPU-RESISC45 dataset.



**Fig. 14.** The confusion matrices on the test set of the NWPU-RESISC45 dataset. (a) Confusion matrix before learning unlabeled set and (b) confusion matrix after learning unlabeled set.



**Fig. 15.** Some examples from the AID dataset. (1)–(30) 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 tank, viaduct.

**Table 5**

The classification results of all methods on the NWPU-RESISC45 dataset.

Method name	Model name	Classification accuracy (%)	
		10% training set	60% training set
Supervised learning	CaffeNet	$77.62 \pm 0.72$	$84.57 \pm 0.63$
	GoogLeNet	$76.00 \pm 0.99$	$82.65 \pm 2.08$
	VGG-F	$79.40 \pm 0.80$	$86.33 \pm 0.47$
	VGG-S	$80.00 \pm 1.65$	$86.96 \pm 0.34$
	VGG-M	$79.24 \pm 0.48$	$86.73 \pm 0.79$
	VGG-16	$78.64 \pm 0.93$	$86.14 \pm 0.96$
	VGG-19	$78.49 \pm 1.14$	$85.24 \pm 0.11$
	ResNet	<b><math>84.30 \pm 0.60</math></b>	<b><math>90.2 \pm 0.2</math></b>
Self-training	VGG-S	<b><math>81.46 \pm 0.68</math></b>	
	ResNet	<b><math>85.82 \pm 1.30</math></b>	
Co-training	ResNet&VGG-S	<b><math>87.25 \pm 0.95</math></b>	
	SSGA-E	<b><math>88.6 \pm 0.31</math></b>	

number of iterative steps, which significantly introduces complex calculation times. According to the results, that the ratio of *batch\_size* is set as 10% or 20% is a reasonable tradeoff between calculation time and classification performance.

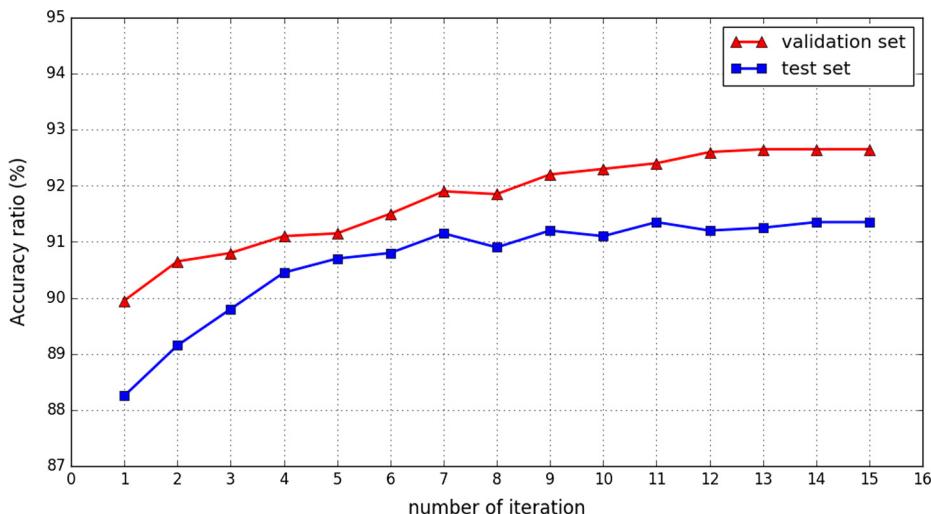
#### 5.6.2. Determination threshold of confusion classes: $C_{thres}$

In the extended algorithm, the determination threshold of confusion classes  $C_{thres}$  plays an important role in the final objective model to affect the final classification capability, which determines

**Table 6**

The classification results of all methods on the AID dataset.

Method name	Model name	Classification accuracy (%)	
		10% training set	60% training set
Supervised learning	CaffeNet	81.81 ± 1.26	88.6 ± 0.6
	GoogLeNet	79.67 ± 0.98	85.29 ± 1.04
	VGG-F	82.25 ± 0.95	89.28 ± 0.87
	VGG-S	83.09 ± 1.81	89.91 ± 0.39
	VGG-M	82.04 ± 0.84	89.4 ± 0.9
	VGG-16	79.69 ± 0.96	87.61 ± 0.69
	VGG-19	79.2 ± 0.75	87.3 ± 0.7
	ResNet	<b>85.71 ± 1.44</b>	<b>93.31 ± 1.67</b>
Self-training	VGG-S	<b>86.02 ± 0.93</b>	
	ResNet	<b>89.38 ± 0.87</b>	
Co-training	ResNet&VGG-S	<b>90.87 ± 1.08</b>	
SSGA-E	ResNet&VGG-S&VGG-16	<b>91.35 ± 0.83</b>	



**Fig. 16.** The classification accuracy curves of the proposed SSGA-E respectively on the validation set and the test set for the AID dataset.

the number of confusion classes. The extended algorithm with different ratio  $C_{thresh}$  is evaluated on the UCM dataset to analyze the performance impact of the parameter. As shown in Fig. 19, the SSGA-E achieves the best results when  $C_{thresh}$  is set to 0.05 both on the validation set and the test set. The accuracy on the test set is acceptable with  $C_{thresh}$  of 0.1. Additionally, as  $C_{thresh}$  instantly increases, accuracy tends to decrease and the tendency on the validation set is more evident than on the test set. When  $C_{thresh}$  is set to greater than 0.3, the accuracy results respectively on validation and test sets are close and reach the minimum value. Because there is hardly any scene class with a wrong classification ratio of more than 30%. Hence, the discriminative evaluator does not enhance classification of the confusion classes. According to the results, it is reasonable to set  $C_{thresh}$  between 0.05 and 0.15.

### 5.6.3. Confidence discussion of generative samples

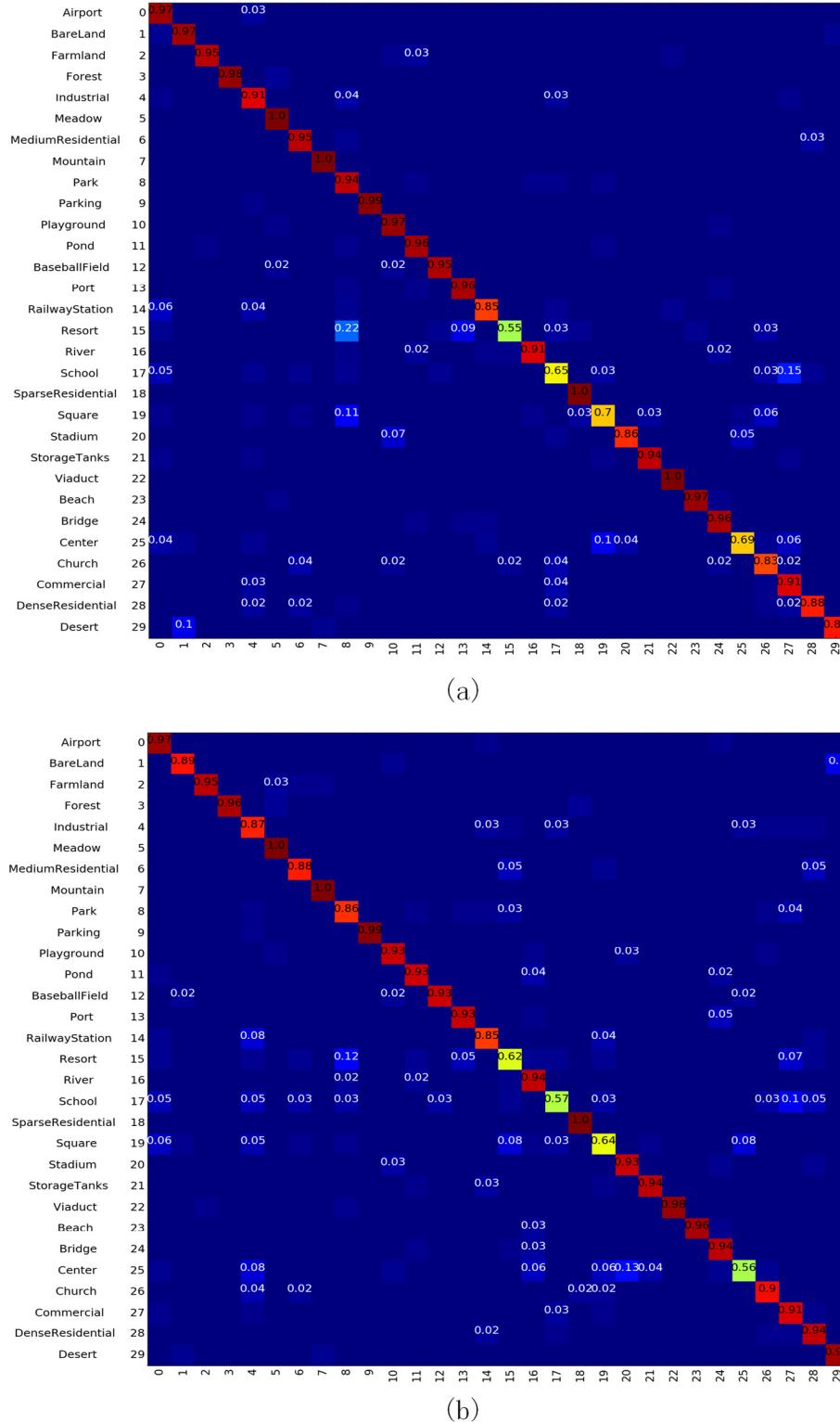
The confidence degree of generative samples of SSGA-E and the independent classifiers is discussed in this section, assuming that the original labels of the dataset are correct. Confidence is the ratio of correctly classified samples to all learned samples. An independent classifier with ResNet feature extractor is denoted as Res-evaluator, and the classifier with VGG-S is denoted as VGG-evaluator. The UCM dataset is used, and the dataset segmentation is the same as the setup in Section 5.2. After increasing the unlabeled ratio from 10% to 50% at 10% intervals, the confidence is compared in Fig. 20.

beled ratio from 10% to 50% at 10% intervals, the confidence is compared in Fig. 20.

The confidence of the generative samples of SSGA-E is at the highest level, 92.24% with 10% unlabeled training set and 94.4% with 50% unlabeled training set. For the independent evaluators, Res-evaluator achieves better confidence than VGG-evaluator. As the ratio of unlabeled samples increases, the confidence level of SSGA-E and independent evaluators have an upward tendency. The confidence difference of the two independent classifiers is relatively large, with less unlabeled training set. As the scale of unlabeled training set increases, the confidence of VGG-evaluator rapidly grows and the difference is greatly reduced.

### 5.6.4. The effect of the discriminator on individual classes

Since the discriminator adds additional constraints to the confusion classes, we further analyze the effect of the discriminator on the accuracies of individual classes. The co-training and SSGA-E methods are respectively performed on UCM dataset with the same setting. The final confusion matrices on test set are shown in Fig. 21. By comparison (a) and (b) in Fig. 21, the accuracies of mobilehomepark, river, denseresidential, storagetanks, freeway grow more than 10%. Furthermore, the accuracies of freeway, denseresidential achieve the maximum growth of 15%. The exper-



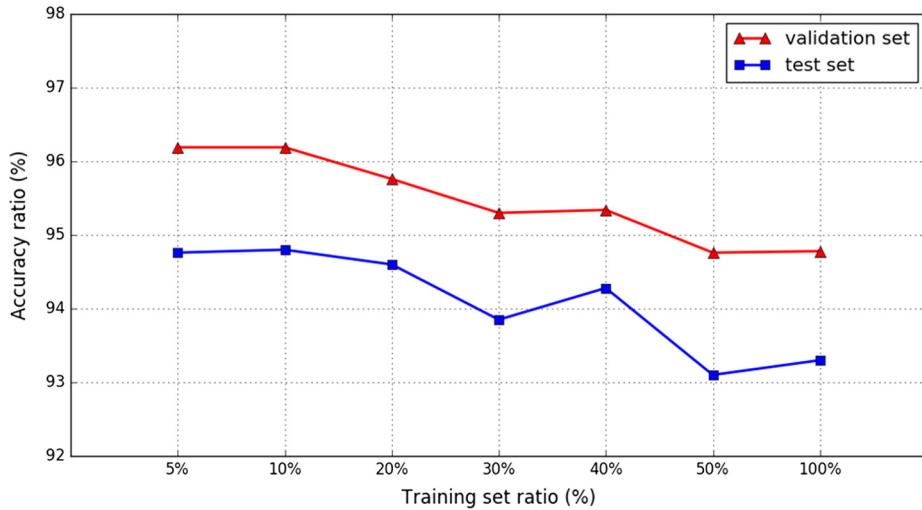
**Fig. 17.** The confusion matrices on the test set of the AID dataset. (a) Confusion matrix before learning unlabeled set and (b) confusion matrix after learning unlabeled set.

imental results prove that discriminator produces a positive effect on classes with a classification accuracy less than a satisfied value.

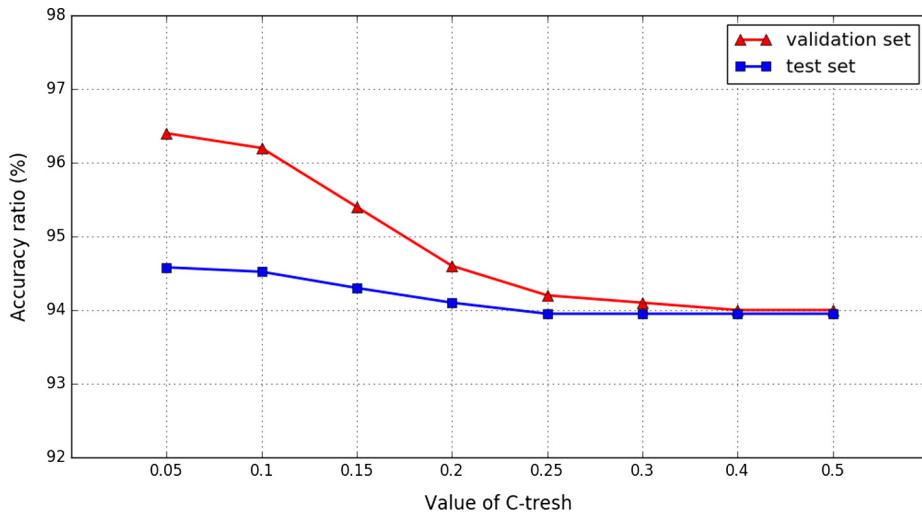
## 6. Conclusion and future work

In this paper, we focus on the problem of insufficient manually-labeled samples in the remote sensing to develop a semi-

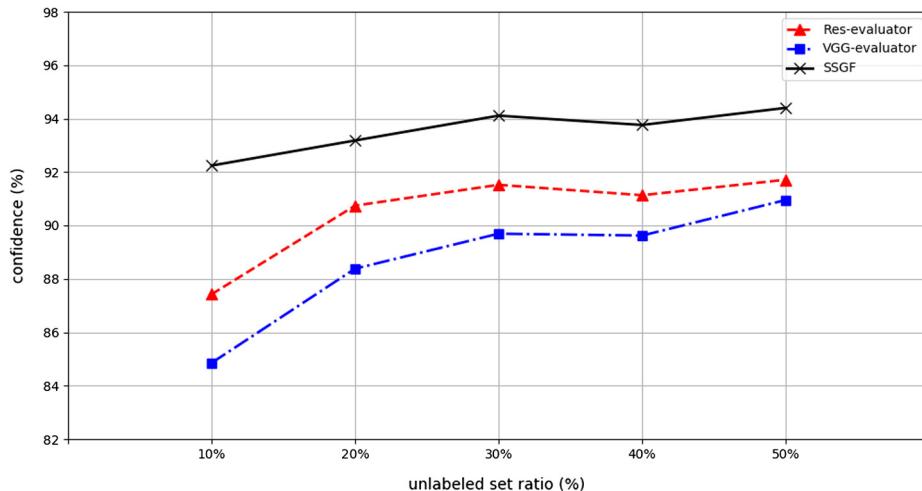
supervised generative framework. This goes along with a limited number of labeled samples and extensive unlabeled samples for reliable annotation datasets for HRRS scene classification. The proposed framework combines deep-learning-based features, the co-training-based self-label method and the discriminative evaluation method to complete the annotation task. The deep-learning-based features have powerful performance in representing the sufficient



**Fig. 18.** The accuracy curves with different *batch\_size* on the validation set and the test set.



**Fig. 19.** The accuracy curves with different values of  $C_{thres}$  on the validation set and the test set.



**Fig. 20.** Curves of the confidence level. The black curve is the confidence of the generative samples learned by SSGF. The red line is the confidence of the Res-evaluator. The blue is the confidence of the VGG-evaluator. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

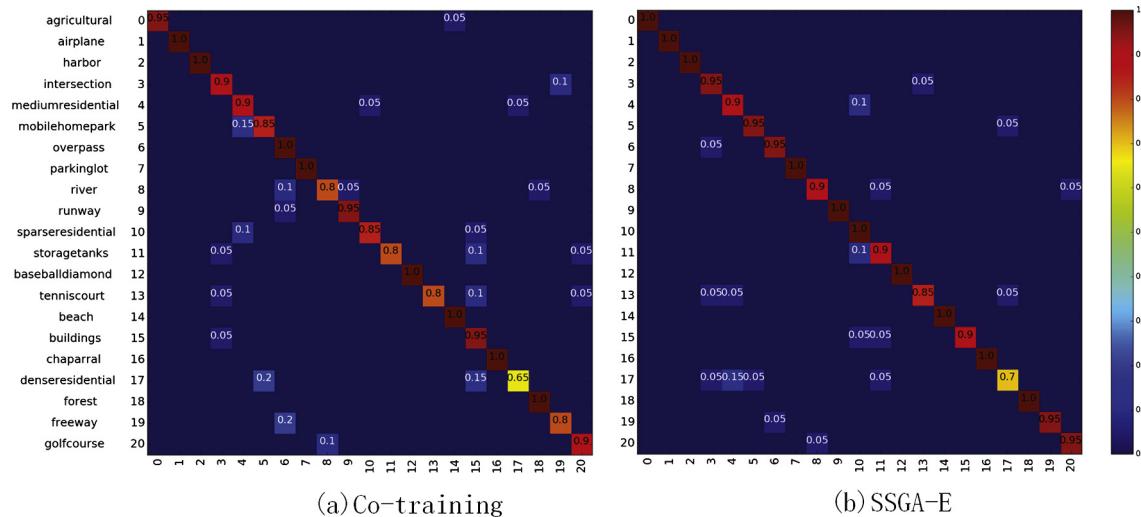


Fig. 21. Confusion matrices of co-training and SSGA-E on test set of UCM dataset. (a) is co-training method and (b) is SSGA-E.

varieties and diversities of HRRS images. The co-training evaluation makes full use of the low-confident samples to improve the generalizability of the model and reduces misclassification. Discriminative evaluation enhances classification of the confusion classes with similar visualize features. Then, we developed an extended algorithm by analyzing the CNNs with different architectures.

We have validated the proposed algorithm through exclusive experiments on four public HRRS datasets and achieves impressive experimental results with a small number of labeled instances. The extended algorithm outperforms than most of the fully-supervised methods and semi-supervised methods with the third best accuracy on the UCM dataset and the second best accuracy on the WHU-RS, the NWPU-RESISC45, and AID datasets. The impressive results demonstrate that the SSGF and the extended algorithm are effective in solving the problem of lacking an annotated dataset for HRRS image scene classification. They are able to learn effective information from the unlabeled samples to improve classification ability. Hence, by learning a sufficient number of unlabeled samples, an ideal model can be obtained, and the generative dataset can be used for the supervised learning scene classification.

In future studies, we plan to introduce active learning and transfer learning to further reduce the requirement of our SSGF for HRRS image scene classification and apply computer vision methods to enhance recognition ability.

## Acknowledgments

We gratefully acknowledge the editor, associate editor, and reviewers for their comments in helping us to improve this work. We also acknowledge the support of the National Natural Science Foundation of China (No. 41571413 and No. 41701429); the Fundamental Research Funds for the Central Universities, China University of Geosciences (Wuhan) (No. CUG170625); and NVIDIA Corporation for the donation of the Titan X GPU used in this research.

## References

- Blaschke, T., 2010. Object based image analysis for remote sensing. *ISPRS J. Photogram. Rem. Sens.* 65 (1), 2–16.
- Blaschke, T., Lang, S., Hay, G., 2008. Object-Based Image Analysis: Spatial Concepts for Knowledge-Driven Remote Sensing Applications. Springer Science & Business Media.
- Blum, A., Mitchell, T., 1998. Combining labeled and unlabeled data with co-training. In: Proceedings of the Eleventh Annual Conference on Computational Learning Theory. ACM, pp. 92–100.
- Cantalloube, H.M., Nahum, C.E., 2013. Airborne sar-efficient signal processing for very high resolution. *Proc. IEEE* 101 (3), 784–797.
- Castelluccio, M., Poggi, G., Sansone, C., Verdoliva, L. Land Use Classification in Remote Sensing Images by Convolutional Neural Networks <<http://arxiv.org/abs/1508.00092>> (accessed on 14 August 2015).
- Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A., 2014. Return of the devil in the details: Delving deep into convolutional nets. In: Proceedings of the British Machine Vision Conference.
- Cheng, G., Han, J., 2016. A survey on object detection in optical remote sensing images. *ISPRS J. Photogram. Rem. Sens.* 117, 11–28.
- Cheng, G., Zhou, P., Han, J., 2016. Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. *IEEE Trans. Geosci. Rem. Sens.* 54 (12), 7405–7415.
- Cheng, G., Han, J., Lu, X., 2017. Remote sensing image scene classification: benchmark and state of the art. *Proc. IEEE* 99, 1–19.
- Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, CVPR 2005, vol. 1. IEEE, pp. 886–893.
- Davenport, E., Cronin, B., 2000. Knowledge management: semantic drift or conceptual shift? *J. Educ. Lib. Inf. Sci.*, 294–306.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. Imagenet: a large-scale hierarchical image database. In: *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, CVPR 2009. IEEE, pp. 248–255.
- Dópez, I., Li, J., Marpu, P.R., Plaza, A., Dias, J.M.B., Benediktsson, J.A., 2013. Semisupervised self-learning for hyperspectral image classification. *IEEE Trans. Geosci. Rem. Sens.* 51 (7), 4032–4044.
- Dou, M., Chen, J., Chen, D., Chen, X., Deng, Z., Zhang, X., Xu, K., Wang, J., 2014. Modeling and simulation for natural disaster contingency planning driven by high-resolution remote sensing images. *Fut. Gener. Comp. Syst.* 37, 367–377.
- Fu, M., Yuan, Y., Lu, X., 2015. Unsupervised feature learning for scene classification of high resolution remote sensing image. In: *2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*. IEEE, pp. 206–210.
- Gamba, P., 2013. Human settlements: a global challenge for eo data processing and interpretation. *Proc. IEEE* 101 (3), 570–581.
- Gómez-Chova, L., Tuia, D., Moser, G., Camps-Valls, G., 2015. Multimodal classification of remote sensing images: a review and future directions. *Proc. IEEE* 103 (9), 1560–1584.
- Han, J., Zhang, D., Cheng, G., Guo, L., Ren, J., 2015. Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning. *IEEE Trans. Geosci. Rem. Sens.* 53 (6), 3325–3337.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778.
- Hinton, G.E., Osindero, S., Teh, Y.-W., 2006. A fast learning algorithm for deep belief nets. *Neural Comput.* 18 (7), 1527–1554.
- Hu, F., Xia, G.-S., Hu, J., Zhang, L., 2015. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Rem. Sens.* 7 (11), 14680–14707.
- Janssen, L.L., Middelkoop, H., 1992. Knowledge-based crop classification of a landsat thematic mapper image. *Int. J. Rem. Sens.* 13 (15), 2827–2837.
- Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T., 2014. Caffe: convolutional architecture for fast feature embedding.

- In: Proceedings of the 22nd ACM International Conference on Multimedia. ACM, pp. 675–678.
- Jolliffe, I., 2002. Principal Component Analysis. Wiley Online Library.
- Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105.
- Li, X., Wang, L., 2015. On the study of fusion techniques for bad geological remote sensing image. *J. Amb. Intell. Human. Comput.* 6 (1), 141–149.
- Li, M., Zhou, Z.-H., 2007. Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. *IEEE Trans. Syst., Man, Cybernet.-Part A: Syst. Hum.* 37 (6), 1088–1098.
- Li, J., Gamba, P., Plaza, A., 2014. A novel semi-supervised method for obtaining finer resolution urban extents exploiting coarser resolution maps. *IEEE J. Select. Top. Appl. Earth Observ. Rem. Sens.* 7 (10), 4276–4287.
- Li, X., Guo, Q., Lu, X., 2016. Spatiotemporal statistics for video quality assessment. *IEEE Trans. Image Process.* 25 (7), 3329–3342.
- Liu, X., Yuan, Y., Zheng, X., 2017. Jointly dictionary learning for change detection in multispectral imagery. *IEEE Trans. Cybernet.* 47 (4), 884–897.
- Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. *Int. J. Comp. Vis.* 60 (2), 91–110.
- Lu, X., Zheng, X., Yuan, Y., 2017. Remote sensing scene classification by unsupervised representation learning. *IEEE Trans. Geosci. Rem. Sens.*, 1–10.
- Lu, X., Zheng, X., Li, X., 2017. Latent semantic minimal hashing for image retrieval. *IEEE Trans. Image Process.* 26 (1), 355–368.
- Luo, X., Wang, M., Dai, G., Chen, X., 2017. A novel technique to compute the revisit time of satellites and its application in remote sensing satellite optimization design. *Int. J. Aerosp. Eng.* 2017.
- Ma, X., Wang, H., Wang, J., 2016. Semisupervised classification for hyperspectral image based on multi-decision labeling and deep feature learning. *ISPRS J. Photogram. Rem. Sens.* 120, 99–107.
- Narayanan, V., Desai, R., Choudhury, S., 2012. Bootstrapping Image Classification with Sample Evaluation, pp. 1–8.
- Odena, A., 2016. Semi-Supervised Learning with Generative Adversarial Networks, pp. 1–3, arXiv preprint arXiv:1606.01583.
- Oliva, A., Torralba, A., 2001. Modeling the shape of the scene: a holistic representation of the spatial envelope. *Int. J. Comp. Vis.* 42 (3), 145–175.
- Olshausen, B.A., Field, D.J., 1997. Sparse coding with an overcomplete basis set: a strategy employed by v1? *Vis. Res.* 37 (23), 3311–3325.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* 12, 2825–2830.
- Plaza, A., Plaza, J., Paz, A., Sanchez, S., 2011. Parallel hyperspectral image and signal processing [applications corner]. *IEEE Sig. Process. Magaz.* 28 (3), 119–126.
- Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y. Overfeat: Integrated Recognition, Localization and Detection Using Convolutional Networks, arXiv preprint arXiv:1312.6229.
- Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In: Proceedings of the International Conference on Learning Representations.
- Springenberg, J.T., 2015. Unsupervised and Semi-Supervised Learning with Categorical Generative Adversarial Networks, pp. 1–20 arXiv preprint arXiv: 1511.06390.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9.
- Triguero, I., García, S., Herrera, F., 2015. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowl. Inf. Syst.* 42 (2), 245–284.
- Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A., 2010. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* 11 (December), 3371–3408.
- Wang, Y., Xu, X., Zhao, H., Hua, Z., 2010. Semi-supervised learning based on nearest neighbor rule and cut edges. *Knowl.-Based Syst.* 23 (6), 547–554.
- Wang, L., Song, W., Liu, P., 2016. Link the remote sensing big data to the image features via wavelet transformation. *Clust. Comput.* 19 (2), 793–810.
- Wang, L., Zhang, J., Liu, P., Choo, K.-K.R., Huang, F., 2017. Spectral-spatial multi-feature-based deep learning for hyperspectral remote sensing image classification. *Soft Comput.* 21 (1), 213–221.
- Xia, G.-S., Yang, W., Delon, J., Gousseau, Y., Sun, H., Maître, H., 2010. Structural high-resolution satellite image indexing. In: ISPRS TC VII Symposium-100 Years ISPRS, vol. 38, pp. 298–303.
- Xia, G.-S., Hu, J., Hu, F., Shi, B., Bai, X., Zhong, Y., Zhang, L., Lu, X., 2017. Aid: a benchmark data set for performance evaluation of aerial scene classification. *IEEE Trans. Geosci. Rem. Sens.* 55 (7), 3965–3981.
- Yang, Y., Newsam, S., 2010. Bag-of-visual-words and spatial extensions for land-use classification. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, pp. 270–279.
- Yao, X., Han, J., Cheng, G., Qian, X., Guo, L., 2016. Semantic annotation of high-resolution satellite images via weakly supervised learning. *IEEE Trans. Geosci. Rem. Sens.* 54 (6), 3660–3671.
- Yarowsky, D., 1995. Unsupervised word sense disambiguation rivaling supervised methods. In: Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, pp. 189–196.
- Yuan, Y., Zheng, X., Lu, X., 2017. Discovering diverse subset for unsupervised hyperspectral band selection. *IEEE Trans. Image Process.* 26 (1), 51–64.
- Yu, Y., Guan, H., Zai, D., Ji, Z., 2016. Rotation-and-scale-invariant airplane detection in high-resolution satellite images based on deep-hough-forests. *ISPRS J. Photogram. Rem. Sens.* 112, 50–64.
- Zhang, F., Du, B., Zhang, L., 2015. Saliency-guided unsupervised feature learning for scene classification. *IEEE Trans. Geosci. Rem. Sens.* 53 (4), 2175–2184.
- Zhang, P., Gong, M., Su, L., Liu, J., Li, Z., 2016. Change detection based on deep feature representation and mapping transformation for multi-spatial-resolution remote sensing images. *ISPRS J. Photogram. Rem. Sens.* 116, 24–41.
- Zhao, W., Du, S., 2016. Learning multiscale and deep representations for classifying remotely sensed imagery. *ISPRS J. Photogram. Rem. Sens.* 113, 155–165.