Scene Classification via a Gradient Boosting Random Convolutional Network Framework

Article *in* IEEE Transactions on Geoscience and Remote Sensing · October 2015

DOI: 10.1109/TGRS.2015.2488681

3 authors, including:

Fan Zhang
Wuhan University
8 PUBLICATIONS **204** CITATIONS
SEE PROFILE

Liangpei Zhang
Wuhan University
524 PUBLICATIONS **7,865** CITATIONS
SEE PROFILE

Scene Classification via a Gradient Boosting Random Convolutional Network Framework

Fan Zhang, Bo Du, *Senior Member, IEEE*, and Liangpei Zhang, *Senior Member, IEEE*

***Abstract*—Due to the recent advances in satellite sensors, a large amount of high-resolution remote sensing images is now being obtained each day. How to automatically recognize and analyze scenes from these satellite images effectively and efficiently has become a big challenge in the remote sensing field. Recently, a lot of work in scene classification has been proposed, focusing on deep neural networks, which learn hierarchical internal feature representations from image data sets and produce state-of-the-art performance. However, most methods, including the traditional shallow methods and deep neural networks, only concentrate on training a single model. Meanwhile, neural network ensembles have proved to be a powerful and practical tool for a number of different predictive tasks. Can we find a way to combine different deep neural networks effectively and efficiently for scene classification? In this paper, we propose a gradient boosting random convolutional network (GBRCN) framework for scene classification, which can effectively combine many deep neural networks. As far as we know, this is the first time that a deep ensemble framework has been proposed for scene classification. Moreover, in the experiments, the proposed method was applied to two challenging high-resolution data sets: 1) the UC Merced data set containing 21 different aerial scene categories with a submeter resolution and 2) a Sydney data set containing eight land-use categories with a 1.0-m spatial resolution. The proposed GBRCN framework outperformed the state-of-the-art methods with the UC Merced data set, including the traditional single convolutional network approach. For the Sydney data set, the proposed method again obtained the best accuracy, demonstrating that the proposed framework can provide more accurate classification results than the state-of-the-art methods.**

***Index Terms*—Convolutional networks (CNets), gradient boosting machine (GBM), scene classification.**

I. Introduction

RECENT years have witnessed a boom in the numbers of different satellite sensors, which now provide us with a large volume of very high resolution (VHR) remote sensing images. How to automatically recognize and analyze scenes from these VHR images effectively and efficiently has drawn great interest in the remote sensing field [1]. However, due to the complex composition and the large number of land-cover types, it is still a difficult task.

There have been many attempts over the years to recognize and analyze scenes from VHR images effectively and efficiently. The majority of the early approaches, such as the bag-of-visual-words (BoVW) based scene classification method [2], simply count the occurrences of the low-level features [e.g., color histogram, local binary pattern, and scale-invariant feature transform (SIFT)] in an image and assign a semantic label to the image according to its content. Although the BoVW approach is highly efficient, a large semantic gap between the low-level features and the high-level semantic meanings still exists, which severely limits the ability to handle complex image scenes. The traditional BoVW-based methods are mainly based on low-level features [2]–[5], and low-level features cannot precisely represent the semantics of the images because of the so-called semantic gap. To bridge this gap, some authors have applied latent Dirichlet allocation [6]–[8], a hierarchical probabilistic model, for the unsupervised learning of the topic allocation, based on the visual word distribution. Under these models, each image is represented as a finite mixture of an intermediate set of topics, which are expected to summarize the scene information [9]. These approaches extract the low-level feature representations from the image and build mid-level features to overcome the semantic gap, but they still ignore the high-level features and the hierarchical feature representations of the image.

Recently, a lot of work in scene classification has been focused on the learning of hierarchical internal feature representations from image data sets [10], [11]. These methods are called "deep learning" methods. Good internal feature representations are hierarchical. In an image, pixels are assembled into edgelets, edgelets are assembled into motifs, motifs are assembled into parts, parts are assembled into objects, and finally, objects are assembled into scenes [10]. This suggests that recognizing and analyzing scenes from VHR images should involve multiple trainable feature extraction stages stacked on top of each other, and we should learn the hierarchical internal feature representations from the image. Convolutional networks (CNets) are a popular deep learning method and have recently enjoyed great success in large-scale visual recognition [11].

To train a neural network model, the model weight is a nonconvex optimization problem with many local minima. The randomness from different initial points tends to differentiate the errors of the networks. Ensembles of networks have been proved to be able to alleviate the problem [12]. An earlier

Manuscript received May 28, 2015; revised August 30, 2015; accepted September 30, 2015. Date of publication October 28, 2015; date of current version February 24, 2016. This work was supported in part by the National Basic Research Program of China (973 Program) under Grants 2011CB707105 and 2012CB719905, by the National Natural Science Foundation of China under Grants 61471274 and 41431175, and by the Natural Science Foundation of Hubei Province under Grant 2014CFB193.

F. Zhang and L. Zhang are with the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, Wuhan 430079, China (e-mail: rszhang@whu.edu.cn; zlp62@whu.edu.cn).

B. Du is with the School of Computer Science, Wuhan University, Wuhan 430072, China (e-mail: remoteking@whu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2015.2488681

overview of different ways of combining neural networks was reported in [13]. We can also consider an ensemble as building a set of "strong" models, which can be further combined to produce a better prediction. The most common ensemble techniques, such as random forests, rely on simple averaging of the models in the ensemble. However, how to combine the classifiers effectively and efficiently is a critical problem. Recently, a boosting framework called gradient boosting machine (GBM) has shown considerable promise in a wide range of practical applications. GBM is a useful practical tool for different predictive tasks, and it can consistently provide more accurate results than the conventional single machine learning models [14]. During the last few years, many empirical studies have been published, which use decision trees as the base learners for GBM [15], [16]. All of these studies have reported impressive improvements in the generalization behavior, suggesting that GBM is a powerful machine learning technique.

Although much useful work has been done, most of the studies have focused on decision trees, and no related research into CNets has been reported in the literature. For the first time, this paper optimally incorporates GBM into a CNet framework. Moreover, due to the extreme flexibility of GBM, we introduce a new multiclass soft-max loss function to combine the GBM with CNet framework. However, because of the limitation of the training data in the satellite image and the high computation complexity of the ensemble CNet framework, we also develop a novel random CNet (RCNet), which can reuse the weight which is learned in each CNet model and reduce the computation complexity.

In this paper, we systematically consider the following questions. Are "strong" models like CNets suitable for an ensemble framework? Does the GBM framework work well for CNets? Due to the flexibility of GBM and the architecture of CNets, how should we choose the loss function for scene classification? Because of the high computational complexity of CNets, is there a way to reuse the weight which is learned in each CNet model and reduce the number of parameters?

The major contributions of this paper are as follows.

1) We propose a gradient boosting random convolutional network (GBRCN) framework for scene classification.
2) We introduce a multiclass soft-max loss function into the GBRCN framework.
3) We propose a novel base learner, the RCNet, which can reuse the weight which is learned in each CNet model and reduce the number of parameters, so that the feature extraction procedure is more effective.

The rest of this paper is organized as follows. In Section II, we briefly review the related works about CNets and introduce the ensemble method for neural networks. In Section III, we describe the GBRCN framework in detail. The details of the experiments and the results are presented in Section IV. Finally, Section V concludes this paper with a discussion of the results and our ideas for future work.

II. Related Work

In this section, we briefly introduce the classical deep learning method, the CNet, and the ensemble method for neural networks.

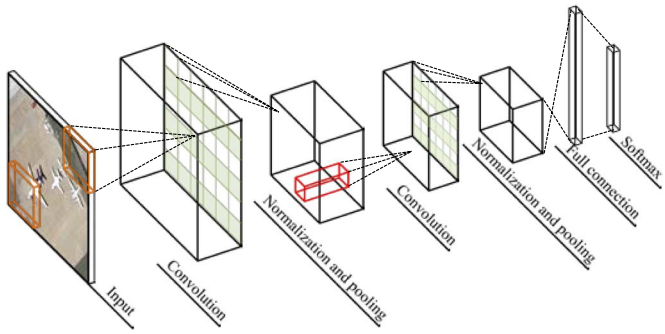


Fig. 1. Typical CNet architecture with two feature extraction stages.

The CNet [17], [18] is a trainable multilayer architecture composed of multiple feature extraction stages. Each stage consists of three layers: 1) a convolutional layer; 2) a nonlinearity layer; and 3) a pooling layer. The architecture of a CNet is designed to take advantage of the 2-D structure of the input image. A typical CNet is composed of one, two, or three of such feature extraction stages, followed by one or more traditional fully connected layers and a final classifier layer, as shown in Fig. 1. Each layer type is described in the following.

Convolutional Layer: The input to the convolutional layer is a 3-D array with r 2-D feature maps of size $m \times n$. Each component is denoted as $x^i_{m,n}$, and each feature map is denoted as x^i. The output is also a 3-D array $m_1 \times n_1 \times k$, composed of k feature maps of size $m_1 \times n_1$. The convolutional layer has k trainable filters of size $l \times l \times q$, also called the filter bank W, which connect the input feature map to the output feature map. The convolutional layer computes the output feature map $z^s = \sum_i^q W_i^s * x^i + b_s$, where $*$ is the 2-D discrete convolution operator and b is a trainable bias parameter.

Nonlinearity Layer: In a traditional CNet, this layer simply consists of a pointwise nonlinearity function applied to each component in the feature map. The nonlinearity layer computes the output feature map $a^s = f(z^s)$, $f(\cdot)$ is commonly chosen to be a rectified linear unit (ReLU), and $f(x) = \max(0, x)$.

Pooling Layer: The pooling layer involves executing a max operation over the activations within a small spatial region G of each feature map: $p^s_G = \max_{i \in G} a^s_i$. To be more precise, a pooling layer can be thought of as consisting of a grid of pooling units spaced s pixels apart, each summarizing a small spatial region of size $p^* p$ centered at the location of the pooling unit.

After the multiple feature extraction stages, the entire network is trained with back-propagation [19] of a supervised loss function such as the classic least-squares output, and the target output y is represented as a $1 - of - K$ vector, where K is the number of outputs and L is the number of layers

$$J(\theta) = \sum_{i=1}^{N} \left(\frac{1}{2} \left\| h(x_i, \theta) - y \right\|^2 \right) + \lambda \sum_{l}^{L} \text{sum} \left(\left\| \theta^{(l)} \right\|^2 \right) \tag{1}$$

where $h(x_i, \theta)$ denotes the CNet function, θ is the learned parameter, λ is a regularization term (also called a weight decay term), N is the number of training samples, and l indexes the layer number. Our goal is to minimize $J(\theta)$ as a function of θ.

To train the CNet, we apply stochastic gradient descent with back-propagation to optimize the function [20].

Considered as a single model, the CNet approach has recently achieved great success and obtained a lot of state-of-the-art results in image understanding. In order to increase the generalization performance, an ensemble of a number of neural networks can significantly improve the generalization ability of a neural network system. The common approaches to an ensemble are plurality voting or majority voting [12] for classification tasks and simple averaging [21] or weighted averaging [22] for regression tasks. In [20] and [23], the authors showed that training a multiple CNet model and simple averaging of the models in the ensemble could improve the performance. In [24], Alvaro *et al.* proposed an artificial neural network ensemble model to estimate global solar radiation from Meteosat satellite images. Meher *et al.* also evaluated different combining criteria for the multispectral remote sensing image classification task using an ensemble of adaptive rule-based granular neural networks [25]. In [22], Perrone *et al.* also presented a weighted averaging scheme for neural networks and showed that the averaging could improve the performance. In [26], Schwenk proposed adaptive boosting of a multilayer network for character recognition, which also obtained a significant improvement. The adaptive boosting algorithm is a specific model under GBM that uses a simple exponential loss function, which limits the performance ability. However, to date, no studies have been reported on how to build a general ensemble framework with CNets using the GBM concept.

III. GBRCN

In this section, the novel modified RCNet and its corresponding GBRCN framework are proposed. The motivation behind our method is to design an ensemble framework for the RCNets and to combine this with a more flexible loss function that is suitable for the RCNet base function.

A. GBRCN

Compared to the traditional machine learning algorithms, the principal difference of the boosting methods is that optimization is undertaken in the function space. That is, we parameterize the function estimate $\hat{f}$ in the additive functional form

$$\hat{f}(x) = \hat{f}^M(x) = \sum_{t=0}^{M} \hat{f}_t(x) \tag{2}$$

where M is the number of functions, $\hat{f}_0$ is the initial estimated function, and $\{\hat{f}_t(x)\}_1^M$ are the function increments, also called "boosts."

In order to distinguish it from the overall ensemble function estimates $\hat{f}(x)$, we denote the base-learner function as $h(x, \theta)$, in which we use the RCNet as our base learner, which is described in the next section.

Given a set of N training examples of the form $(x, y)_{i=1}^N$, the goal of supervised learning is to reconstruct the unknown functional dependence $x \xrightarrow{f} y$ with our estimation $\hat{f}(x)$ and to minimize a specified loss function $\Psi(y, f)$, where $x = (x_1, \ldots, x_d)$ refers to the d-dimensional input variables and y refers to the corresponding label of the response variable

$$\hat{f}(x) = y \tag{3}$$

$$\hat{f}(x) = \arg\min_{f(x)} \Psi\left(y, \hat{f}(x)\right). \tag{4}$$

Compared to a conventional machine learning algorithm, we calculate the function increment according to a "greedy stagewise" approach with the base learner. For the function estimate, the optimization rule is therefore defined as

$$\hat{f}_t \leftarrow \hat{f}_{t-1} + \rho_t h(x, \theta_t) \tag{5}$$

$$(\rho_t, \theta_t) = \arg\min_{\rho,\theta} \sum_{i=1}^{N} \Psi\left(y, \hat{f}_{t-1} + \rho h(x_i, \theta)\right). \tag{6}$$

In practice, given a specific loss function $\Psi(y, f)$ and a base-learner function $h(x, \theta)$, the solution to the parameter estimation can be difficult to obtain. In this paper, we use a multiclass soft-max loss function and the RCNet as the base learner, and we look for the solution to the function increment in the function space.

B. Soft-Max Loss Function for Classification

To use the GBRCN for the scene classification task, we have to provide the specific choice of loss function and base learner. In this section, we describe a multiclass soft-max loss function combined with the RCNet for the scene classification. The loss function is

$$\Psi\left(y, \hat{f}(x)\right) = -\sum_{k=1}^{K} y_k \log\ p_k(x) \tag{7}$$

where the target label y is represented as a $1 - of - K$ vector, where K is the number of classes, $\hat{f}(x)$ is the overall ensemble function estimate, and

$$p_k(x) = \frac{\exp\left(f_k(x)\right)}{\sum_{l=1}^{K} \exp\left(f_l(x)\right)}. \tag{8}$$

In practice, given the soft-max loss function and the RCNet base learner, the solution to (6) can be difficult to obtain. To deal with this, it is proposed to choose a base-learner function $h(x, \theta_t)$ to be the most parallel to the negative gradient $g_t(x)$ along the observed data

$$g_t(x) = \left[\frac{\partial \Psi\left(y, f(x)\right)}{\partial f(x)}\right]_{f(x)=\hat{f}_{t-1}(x)}. \tag{9}$$

Instead of looking for the general solution in the function space, this permits the replacement of a potentially very difficult optimization task with a classic least-squares minimization

$$\theta_t = \arg\min_{\theta} \sum_{i=1}^{N} \left[-g_t(x_i) + h(x_i, \theta)\right]^2. \tag{10}$$

Substituting (8) into (7) and taking the first derivatives, we have

$$-g_t(x) = -\left[\frac{\partial \Psi\left(y, f(x)\right)}{\partial f(x)}\right]_{f(x)=\hat{f}_{t-1}(x)} = y - p(x). \tag{11}$$

Given the approximate negative gradient function $h(x, \theta_t)$, the multiplier ρ_t in (6) is calculated as follows:

$$\rho_{t,k} = \arg\min_{\rho} \sum_{i=1}^{N} \sum_{k=1}^{K} \Psi\left(y_k, \hat{f}_{t-1,k} + \rho_{t,k} h_k(x_i, \theta)\right). \quad (12)$$

This formula has no closed-form solution. We apply stochastic gradient descent to find the multiplier ρ_t. The update rule is described as follows.

For simplicity, we only consider the situation with a single data point. Substituting (8) into (12) and taking the first derivatives, we have (13)–(15), shown at the bottom of the page, where $1(c = k)$ is the indicator function. This function takes a value of 1 when $c = k$, and 0 otherwise.

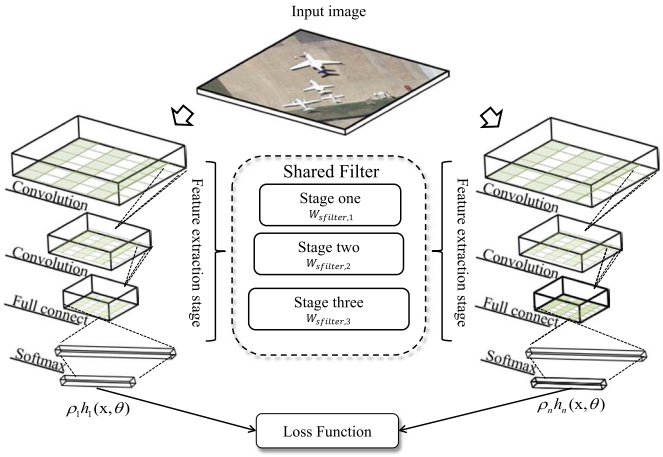


Fig. 2. Overall network architecture of the GBRCN framework.

C. Base Learner: RCNet

We now describe the RCNet base learner for the function estimate. As mentioned before, the CNet [17], [18] is a trainable multilayer architecture composed of multiple feature extraction stages, followed by one or more traditional fully connected layers and a final classifier layer.

Due to the specific architecture of the CNet, we can view each feature extraction stage as a feature learning process from the low-level features to the high-level feature concepts [27]. Recently, using a "pretrained" CNet and sharing the parameters between the feature extraction stages in different CNets have been attracting a lot of attention [28]. How to reuse or share the parameters learned from the base learner in the GBM framework, thereby reducing the number of parameters, has become a critical problem.

The simplest way is to share all of the parameters between the feature extraction stages in different CNet functions. However, this will result in overfitting. Inspired by the randomness in the random forest method, we have developed a random version of CNet called RCNet, which randomly shares some of the parameters with a filter bank, as shown in Fig. 2.

$$\Psi\left(y, \hat{f}(x)\right) = -\sum_{k=1}^{K} 1(y_k = c) \log\ p_k(x) = -\log\ p_c(x) \quad (13)$$

$$\nabla\rho_{t,k} = \frac{\partial}{\partial \rho_{t,k}} - \log p_c(x) \quad (14)$$

$$= -\frac{1}{p_c(x)} \frac{\partial}{\partial \rho_{t,k}} \left(\frac{\exp\left(f_{t-1,c}(x) + \rho_{t,c} h_c(x)\right)}{\sum_{k=1}^{k} \exp\left(f_{t-1,k}(x) + \rho_{t,k} h_k(x)\right)} \right)$$

$$= -\frac{1}{p_c(x)} \cdot \left(\frac{\frac{\exp(f_{t-1,c}(x) + \rho_{t,c} h_c(x))}{\partial \rho_{t,k}}}{\sum_{k=1}^{k} \exp\left(f_{t-1,k}(x) + \rho_{t,k} h_k(x)\right)} - \frac{\exp(f_{t-1,c}(x) + \rho_{t,c} h_c(x)) \cdot \frac{\sum_{k=1}^{k} \exp(f_{t-1,k}(x) + \rho_{t,k} h_k(x))}{\partial \rho_{t,k}}}{\left(\sum_{k=1}^{k} \exp\left(f_{t-1,k}(x) + \rho_{t,k} h_k(x)\right)\right)^2} \right)$$

$$= -\frac{1}{p_c(x)} \cdot \left(\frac{1(c = k) \cdot \exp\left(f_{t-1,c}(x) + \rho_{t,c} h_c(x)\right)}{\sum_{k=1}^{k} \exp\left(f_{t-1,k}(x) + \rho_{t,k} h_k(x)\right)} \cdot h_c(x) \right.$$

$$\left. - \frac{\exp(f_{t-1,c}(x) + \rho_{t,c} h_c(x)) \cdot \exp(f_{t-1,k}(x) + \rho_{t,k} h_k(x))}{\left(\sum_{k=1}^{k} \exp\left(f_{t-1,k}(x) + \rho_{t,k} h_k(x)\right)\right)^2} \cdot h_k(x) \right)$$

$$= -\frac{1}{p_c(x)} \cdot (1(c = k) \cdot p_c(x) \cdot h_c(x) - p_c(x) \cdot p_k(x) \cdot h_k(x))$$

$$= -\left(1(c = k) \cdot h_c(x) - p_k(x) \cdot h_k(x)\right) \rho_{t,k} \leftarrow \rho_{t-1,k} - \nabla\rho \quad (15)$$

Fig. 3. Example images associated with the 21 land-use categories in the UC Merced data set: 1) agricultural, 2) airplane, 3) baseballdiamond, 4) beach, 5) buildings, 6) chaparral, 7) denseresidential, 8) forest, 9) freeway, 10) golfcourse, 11) harbor, 12) intersection, 13) mediumresidential, 14) mobilehomepark, 15) overpass, 16) parkinglot, 17) river, 18) runway, 19) sparseresidential, 20) storagetanks, and 21) tenniscourt.

During each RCNet function, we randomly sample a W_{filter} from the shared filter bank W_{sfilter} to construct the feature extraction stage of the RCNet function and simultaneously update the filter parameter in the RCNet function and shared filter bank. The filter size of the shared filter bank is larger than the filter in the RCNet function, which means that the different RCNets share some of the filter parameters, thereby decreasing the number of parameters in the whole framework. The randomness also maintains the diversity between the different RCNet base learners.

D. Hyperparameters

The GBM framework has two important hyperparameters. The most important hyperparameter is the number of base learners M or "boosting iterations" in the ensemble. However, a larger value of M increases the model complexity and can lead to overfitting. Another hyperparameter is the regularization parameter called "shrinkage."

The simplest form of regularization through shrinkage is direct proportional shrinkage [29], [30]. In this case, the effect of the shrinkage is directly defined as the parameter $\lambda \in (0, 1]$. The regularization is applied to the final step in the gradient boosting algorithm

$$\hat{f}_t \leftarrow \hat{f}_{t-1} + \lambda \rho_t h(x, \theta_t). \tag{16}$$

It is a common pattern that, the smaller the value of parameter λ, and therefore the lower the shrinkage, the better the generalization [14]. In this paper, we set the parameter as $\lambda = 1/(M + 1)$. Clearly, choosing a smaller value of λ will increase the number of iterations M, which is required for convergence to a similar empirical loss minimum when using a larger value of λ [29].

E. Procedure of the GBRCN

The detailed procedure of the proposed GBRCN is described in Algorithm 1. One RCNet base-learner function is achieved from steps 3 to 5, and the output of the base-learner function and multiplier is added to the previous function estimate. We repeat this process until the M RCNet base-learner functions and multipliers are achieved.

Algorithm 1 Gradient Boosting Random Convolutional Network

Inputs:
-input data $(x, y)_i^N$
-number of iterations M
-shrinkage parameter $\lambda = 1/(M + 1)$
Algorithm:
1. Initialize $f_0(x) = 0, k = 1, \ldots, K$
2. **for** $t = 1$ *to* M **do**
3. compute the negative gradient $g_t(x)$
4. fit an RCNet base-learner function

$$h(x, \theta_t) = \sum_{i}^{N_c} \left(\frac{1}{2} \| h(x, \theta_t) - g_t(x) t \|^2 \right) + \lambda \sum_{l}^{L} sum\left(\left\| \theta^{(l)} \right\|^2 \right)$$

5. find the best gradient descent multiplier ρ_t

$$\rho_{t,k} = \arg\min_{\rho} \sum_{i=1}^{N} \sum_{k=1}^{K} \Psi \left(y_k, \hat{f}_{t-1,k} + \rho_{t,k} h_k(x_i, \theta) \right)$$

6. update the function estimate:

$$\hat{f}_t \leftarrow \hat{f}_{t-1} + \lambda \rho_t h(x, \theta_t)$$

7. **end for**

IV. Experiments and Analysis

In this section, we first describe the data sets used for the experiments and the parameter settings of the proposed method. The results obtained for the scene classification are then discussed.

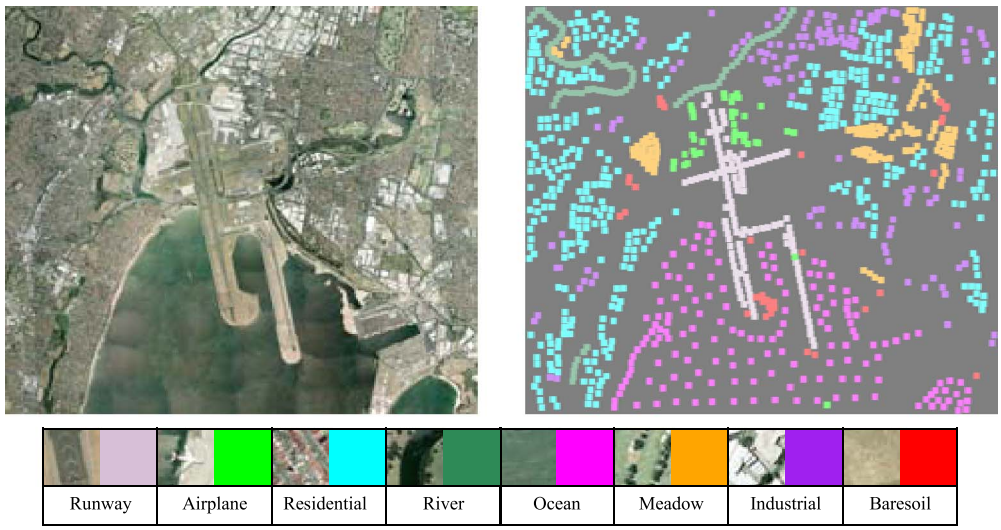


Fig. 4. (Left) Whole image for image annotation. (Right) Image ground truth. (Bottom) Example images associated with the eight land-use categories from the image: 1) runway, 2) airplane, 3) residential, 4) river, 5) ocean, 6) meadow, 7) industrial, and 8) baresoil.

A. Description of the Data Sets and Experimental Settings

Two different image data sets were used in the experiments. The first data set chosen for investigation was the well-known UC Merced data set [4]. Fig. 3 shows a few example images representing various aerial scenes that are included in this data set. The images have a resolution of 1 ft per pixel and are of 256×256 pixels. In our experiment, we resized the image to 128×128 pixels. The data set contains 21 challenging scene categories, with 100 image samples per class. The data set represents highly overlapping classes, such as denseresidential, mediumresidential, and sparseresidential, which mainly differ in the density of structures. Following the experimental setup in [1], we randomly selected 80% of the samples from each class for training and set the remaining images for testing.

The other image data set was constructed from a large satellite image, which was acquired from Google Earth, of Sydney, Australia. The spatial resolution of this image was about 1.0 m. The large image to be annotated was of 7849×9073 pixels, as shown in Fig. 4. There were eight classes of training images: residential, airplane, meadow, rivers, ocean, industrial, baresoil, and runway. Fig. 4 shows some examples of such images. This data set consists of not only the eight defined classes but also some other classes that have not been learned, such as the bridges and the main roads. We manually labeled part of the image to obtain a subregion image data set, in which each subregion was of the size of 128×128, whereby each subimage was supposed to contain a certain scene. The training set for each class contained 25 samples of the labeled images for each class, and the remaining images were used for testing, as shown in Table I.

For the UC Merced data set, we trained the CNet and RCNet functions using stochastic gradient descent, with a batch size of 64, a momentum of 0.9, a weight decay of 0.0005, and a learning rate of 0.01. We trained each RCNet for roughly 500 cycles with the whole training set.

For the Sydney data set, we trained the CNet and RCNet functions using stochastic gradient descent, with a batch size of 32, a momentum of 0.9, a weight decay of 0.0005, and a learning rate of 0.01. We trained each RCNet for roughly 800 cycles with the whole training set.

TABLE I
TRAINING AND TEST SAMPLES FOR THE SYDNEY DATA SET

Class		Samples	
No	Name	Training	Test
1	Runway	25	97
2	Airplane	25	16
3	Residential	25	381
4	River	25	59
5	Ocean	25	133
6	Meadow	25	102
7	Industrial	25	101
8	Baresoil	25	9
Total		200	898

These experiments were run on a personal computer with a single Intel core i7 CPU, an NVIDIA Titan GPU, and 6-GB memory. The operating system was Windows 7, and the implementation environment was under MATLAB 2014a with a CUDA kernel.

B. Experimental Results With the UC Merced Data Set

To measure the scene classification performance with the UC Merced data set, we first compared the classification accuracies with a single CNet function of different sizes of feature extraction architecture. The different CNet architectures evaluated in this paper are outlined in Table II, one per column. Table III shows the classification performance with the different sizes of feature extraction architecture. The convolutional layer parameters are denoted as "conv(receptive field size)-(number of features)." The results show that using three feature extraction stages achieved the best classification accuracy. The CNet could benefit from increasing the depth of the network moderately. Recent study by He *et al.* [31] also suggests that CNet can extract a more efficient high-level feature from the deeper network than the shallow network. However, with the

TABLE II
DIFFERENT CNET ARCHITECTURES

CNet configurations						
CNet-A	CNet-B	CNet-C	CNet-D	RCNet-A	RCNet-B	RCNet-C
Input image						
Conv3*3-96	Conv3*3-96	Conv3*3-128	Conv3*3-128	Conv3*3-96(128)	Conv3*3-96(128)	Conv3*3-128(160)
Maxpool						
Conv3*3-128	Conv3*3-128	Conv3*3-192	Conv3*3-192	Conv3*3-128(160)	Conv3*3-128(160)	Conv3*3-192(256)
Maxpool						
	Conv3*3-128	Conv3*3-192	Conv3*3-192 Conv3*3-192		Conv3*3-128(160)	Conv3*3-192(256)
LRN-Maxpool						
Full-connect-420						
Full-connect-10						
Soft-max						

TABLE III
COMPARISON OF THE DIFFERENT CNET ARCHITECTURES WITH THE UC MERCED DATA SET

Method	CNet-A-UCM	CNet-B-UCM	CNet-C-UCM	CNet-D-UCM
Accuracy	87.62%	90.00%	91.67%	90.95%

TABLE IV
COMPARISON OF THE DIFFERENT RCNET ARCHITECTURES WITH THE UC MERCED DATA SET

Architecture	RCNet-A-UCM			RCNet-B-UCM			RCNet-C-UCM		
Base-learner number	6	9	12	6	9	12	6	9	12
Average	89.05%	88.33%	88.33%	90.72%	91.67%	91.67%	92.01%	92.01%	92.39%
GBRCN	89.29%	89.29%	89.05%	92.39%	93.09%	92.86%	93.09%	94.05%	94.53%

TABLE V
COMPARISON OF THE DIFFERENT BASE-LEARNER FUNCTIONS WITH THE UC MERCED DATA SET

Architecture	RCNet-C-UCM		
Base-learner number	6	9	12
Average	92.01%	92.01%	92.39%
GBCN	93.09%	93.57%	93.57%
GBRCN	93.09%	94.05%	94.53%

increasing depth of the network to the four-layer CNet-D-UCM, the classification accuracy oscillated slightly where the deep architecture has more parameters to train, and the limitation of the training samples restricted the performance of the deep architecture. What is more, why "deeper is not better" is still an open question, which requires investigations to better understand the relationship between deep architectures and data.

We then compared RCNets with different architectures and base-learner numbers, as shown in Table II. For the RCNets, we used the same architecture as CNet-A, CNet-B, and CNet-C, and we denoted the size of the shared filter bank as "conv(receptive field size)-(number of features)-(size of shared filter bank)." Table IV shows the classification performance with the proposed GBRCN method with different numbers of base-learner functions. From Table IV, we can observe the same result that RCNets could benefit from increasing the depth of the network moderately. We also compared the proposed method to simple averaging of CNets. In all cases, the proposed GBRCN framework improved the generalization performance of the different network architectures. Compared to the simple averaging of CNets, the proposed method RCNet-C-UCM improved the performance from 92.39% to 94.53%, which is significant.

We also evaluated the impact of different base-learner functions. As shown in Table V, we first compared the performance of our GBM framework with traditional CNets base-learner functions (GBCN) to simple average multiple CNets. The result revealed that GBM is an efficient ensemble framework compared to the simple average method. Then, we compared the performance of our GBM framework with RCNet base-learner functions (GBRCN) and CNet base-learner functions (GBCN). From Table V, we can observe that the GBRCN achieves the best classification accuracy, which confirms our purpose that RCNet can efficiently reuse the weight and improve the generalization performance. Based on the results, we can find that the GBM framework is useful in improving the classification performance. However, the huge size of the deep network and the large number of base-learner functions

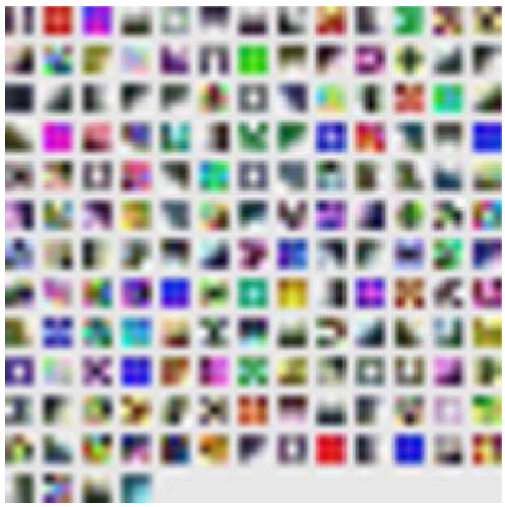


Fig. 5. One hundred sixty convolutional kernels of size $3 \times 3 \times 3$ learned by the first convolutional layer with the UCM data set and RCNet-I-C. All of the kernels are upsampled by bilinear interpolation to $10 * 10$ for better visualization.

TABLE VI
COMPARISON WITH THE PREVIOUS REPORTED ACCURACIES WITH THE UC MERCED DATA SET

Method	SPMK [3]	SSC [1]	SSAE[11]	RCNet-C-UCM
Accuracy	74%	81.67%	82.72%	94.53%

TABLE VII
COMPARISON OF THE DIFFERENT CNET ARCHITECTURES WITH THE SYDNEY DATA SET

Method	CNet-A-SYD	CNet-B-SYD	CNet-C-SYD	CNet-D-SYD
Accuracy	93.43%	94.99%	94.99%	94.33%

TABLE VIII
COMPARISON OF THE DIFFERENT RCNET ARCHITECTURES WITH THE SYDNEY DATA SET

Architecture	RCNet-A-SYD			RCNet-B-SYD			RCNet-C-SYD		
Base-learner number	6	9	12	6	9	12	6	9	12
Average	95.87%	96.88%	96.88%	95.87%	96.33%	96.11%	96.00%	96.00%	96.11%
GBRCN	98.11%	98.33%	98.45%	98.67%	98.78%	98.78%	98.56%	98.67%	98.67%

may bring the overfitting problem. The novel RCNet which can reuse the weight and reduce the computation complexity also contributed to the performance improvement.

For visualization of the learned feature extractors, we show the layer-1 shared feature filters generated from RCNet-C-UCM in Fig. 5. Here, it can be seen that the proposed method tends to learn edges and textural feature extractors.

Finally, we also compared the scene classification performance of the proposed approach with the spatial pyramid matching kernel (SPMK) [3], the SIFT + sparse coding (SSC) approach described in [1], and the saliency-guided sparse autoencoder (SSAE) method in [32]. We compared the reported classification performances with the challenging UC Merced data set. Of the four strategies that we compared, the GBRCN framework produced the best performance, as shown in Table VI.

C. Experimental Results With the Sydney Data Set

For the Sydney data set, we also compared the classification accuracies with a single CNet function of different sizes of the feature extraction architecture and filter sizes, as shown in Table VII. The experimental analysis shows that the architectures of CNet-B-SYD and CNet-C-SYD produced the best accuracy with a single CNet in this data set. We can get the same result as in the UC Merced data set that CNet could benefit from increasing the depth of the network moderately. However, for the Sydney data set which is simpler and only has eight classes and less training sample, CNet-B is enough to produce the best accuracy.

We then compared RCNets with different architectures. For the RCNets, we used the same architectures that were used with the UCM data set. Table VIII shows the classification performance of the proposed GBRCN method with different numbers of base-learner functions. A similar conclusion can be made for the Sydney data set: The proposed GBRCN framework improved, in all cases, the generalization performance of the different network architectures. Compared to the simple averaging of CNets, the RCNet-B-SYD method produced the best result. We also show the layer-1 shared feature filter generated from RCNet-B-SYD in Fig. 6.

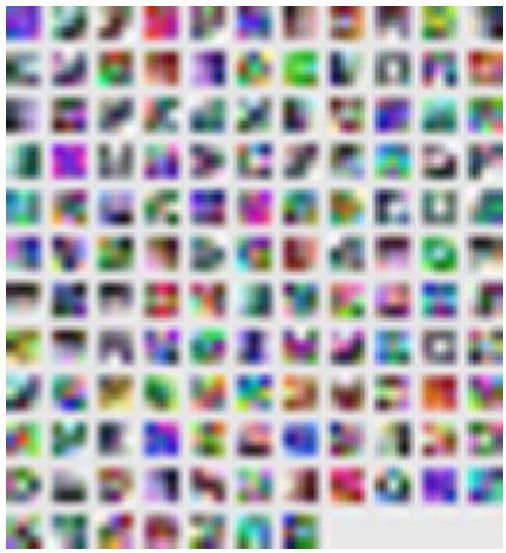


Fig. 6. One hundred twenty-eight convolutional kernels of size $3 \times 3 \times 3$ learned by the first convolutional layer with the Sydney data set and RCNet-I-B. All of the kernels are upsampled by bilinear interpolation to $10 * 10$ for better visualization.

TABLE IX
COMPARISON OF THE DIFFERENT BASE-LEARNER FUNCTION WITH THE SYDNEY DATA SET

Architecture	RCNet-B-SYD		
Base-learner number	6	9	12
Average	95.87%	96.33%	96.11%
GBCN	98.45%	98.45%	98.11%
GBRCN	98.67%	98.78%	98.78%

We also evaluated the impact of base-learner functions. As shown in Table IX, we compared the performance of our GBM framework with CNet base-learner functions (GBCN). From Table IX, we can observe the same phenomenon that GBCN is an efficient ensemble framework compared to simple average multiple CNets. However, compared to GBRCN, when using a large base-learner number, GBCN gets a slight decrease. The result revealed that, when training such a huge size ensemble method, it is efficient and necessary to share the parameter and reuse the weight between the different base-learner functions like RCNet base-learner functions. Furthermore, the result also

TABLE X
OVERALL ACCURACY FOR THE DIFFERENT METHODS WITH THE SYDNEY DATA SET

Method	SPMK	SSC	SSAE	RCNet-B-SYD
Accuracy	89.67%	91.33%	92.20%	98.78%

shows that the GBM framework and RCNet base-learner functions contribute to the performance improvement equally.

Finally, we compared the final classification accuracies for RCNet-B-SYD and the traditional methods. Table X shows the average overall accuracies for the four methods. The results confirm that using the GBRCN framework is an efficient way to increase the classification accuracy.

V. CONCLUSION

In this paper, we have proposed a GBRCN framework for scene classification, which combines many single deep neural networks, and we have developed a novel modified version of the CNet called the RCNet, which is suitable for the GBM framework. The proposed framework gave a state-of-the-art performance with two challenging high-resolution data sets. The experiments showed the following: 1) the RCNet is a useful and accurate base-learner function to construct the neural network ensemble and 2) the novel GBRCN framework can generate more accurate results than the traditional methods. Future work may focus on application of GBRCN on hyperspectral applications [33].

REFERENCES

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

[2] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 13–16, 2003, vol. 2, pp. 1470–1477.

[3] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 2006, vol. 2, pp. 2169–2178.

[4] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. ACM GIS*, 2010, pp. 270–279.

[5] F. Hu *et al.*, "Unsupervised feature learning via spectral clustering of multidimensional patches for remotely sensed scene classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 5, pp. 2015–2030, May 2015.

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

[7] M. Lienou, H. Maitre, and M. Datcu, "Semantic annotation of satellite images using latent Dirichlet allocation," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 1, pp. 28–32, Jan. 2010.

[8] Y. Zhong, Q. Zhu, and L. Zhang, "Scene classification based on the multifeature fusion probabilistic topic model for high spatial resolution remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 11, pp. 6207–6222, Nov. 2015.

[9] N. Rasiwasia and N. Vasconcelos, "Latent Dirichlet allocation models for image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2665–2679, Nov. 2013.

[10] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proc. IEEE ISCAS*, Jun. 2010, pp. 253–256.

[11] P. Sermanet *et al.*, "OverFeat: Integrated recognition, localization and detection using convolutional networks," in *Proc. ICLR*, 2014, pp. 1–16.

[12] L. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 10, pp. 993–1001, Oct. 1990.

[13] M. P. Perrone, "Putting it all together: Methods for combining neural networks," in *Proc. Adv. Neural Inf.*, 1994, vol. 6, pp. 1188–1189.

[14] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Front. Neurorobot.*, vol. 7, p. 21, 2013.

[15] A. Bissacco, M. H. Yang, and S. Soatto, "Fast human pose estimation using appearance and motion via multi-dimensional boosting regression," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, 2007, pp. 1–8.

[16] R. Johnson and T. Zhang, "Learning nonlinear functions using regularized greedy forest," in *Proc. IEEE Trans. Pattern Anal. Mach. Intell.*, 2014, pp. 942–954.

[17] Y. LeCun *et al.*, "Handwritten digit recognition with a back-propagation network," in *Proc. NIPS*, 1990, pp. 396–404.

[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[19] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[20] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf.*, 2012, vol. 25, pp. 1106–1114.

[21] D. W. Opitz and J. W. Shavlik, "Actively searching for an effective neural network ensemble," *Connect. Sci.*, vol. 8, no. 3/4, pp. 337–353, 1996.

[22] M. P. Perrone and L. N. Cooper, "When networks disagree: Ensemble methods for hybrid neural networks," in *Neural Networks for Speech and Image Processing*, R. J. Mammone, Ed. London, U.K.: Chapman & Hall, 1993.

[23] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. CVPR*, 2015, p. 1.

[24] A. Linares-Rodriguez, J. A. Ruiz-Arias, D. Pozo-Vazquez, and J. Tovar-Pescador, "An artificial neural network ensemble model for estimating global solar radiation from Meteosat satellite images," *Energy*, vol. 61, pp. 636–645, Nov. 2013.

[25] S. K. Meher and D. A. Kumar, "Ensemble of adaptive rule-based granular neural network classifiers for multispectral remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 5, pp. 2222–2231, May 2015.

[26] H. Schwenk and Y. Bengio, "Boosting neural networks," *Neural Comput.*, vol. 12, pp. 1869–1887, 2000.

[27] M. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. ECCV*, 2013, pp. 818–833.

[28] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. CVPR*, 2013, pp. 1–25.

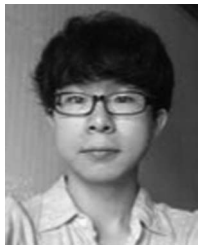
[29] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.

[30] L. Zhang *et al.*, "Ensemble manifold regularized sparse low-rank approximation for multiview feature embedding," *Pattern Recognit.*, vol. 48, no. 10, pp. 3102–3112, 2015.

[31] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proc. IEEE CVPR*, 2015, pp. 5353–5360.

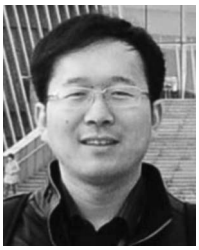
[32] F. Zhang, B. Du, and L. Zhang, "Saliency-guided unsupervised feature learning for scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 2175–2184, Apr. 2015.

[33] B. Du and L. Zhang, "A discriminative metric learning based anomaly detection method," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 11, pp. 6844–6857, Nov. 2014.



Fan Zhang received the B.S. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2012, where he is currently working toward the Ph.D. degree in the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing.

His research interests include high-resolution image and hyperspectral image classification, machine learning, and computation vision in remote sensing applications.



Bo Du (M'10–SM'15) received the B.S. degree and the Ph.D. degree in photogrammetry and remote sensing from the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, Wuhan, China, in 2005 and 2010, respectively.

He is currently an Associate Professor with the School of Computer, Wuhan University. He has more than 40 research papers published in the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING (TGRS), IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), IEEE JOURNAL OF SELECTED TOPICS IN EARTH OBSERVATIONS AND APPLIED REMOTE SENSING (JSTARS), IEEE GEOSCIENCE AND REMOTE SENSING LETTERS (GRSL), etc. His major research interests include pattern recognition, hyperspectral image processing, and signal processing.

Dr. Du received the best reviewer awards from the IEEE Geoscience and Remote Sensing Society (GRSS) for his service to JSTARS in 2011 and ACM rising star awards for his academic progress in 2015. He was the Session Chair for the 4th IEEE GRSS Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS). He also serves as a reviewer of 20 Science Citation Index (SCI) magazines including IEEE TGRS, TIP, JSTARS, and GRSL.

Liangpei Zhang (M'06–SM'08) received the B.S. degree in physics from Hunan Normal University, Changsha, China, in 1982, the M.S. degree in optics from Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, China, in 1988, and the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 1998.

He is currently the Head of the Remote Sensing Division, State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University. He is also a "Chang-Jiang Scholar" Chair Professor appointed by the Ministry of Education of China. He is currently a Principal Scientist with the China State Key Basic Research Project (2011–2016) appointed by the Ministry of National Science and Technology of China to lead the remote sensing program in China. He has more than 450 research papers and five books. He is the holder of 15 patents. His research interests include hyperspectral remote sensing, high-resolution remote sensing, image processing, and artificial intelligence.

Dr. Zhang is the Founding Chair of the IEEE Geoscience and Remote Sensing Society (GRSS) Wuhan Chapter. He is a Fellow of the Institution of Engineering and Technology (IET), an executive member (Board of Governor) of the China National Committee of International Geosphere-Biosphere Programme, an executive member of the China Society of Image and Graphics, etc. He regularly serves as a Cochair of the series SPIE Conferences on Multispectral Image Processing and Pattern Recognition, Conference on Asia Remote Sensing, and many other conferences. He edits several conference proceedings, issues, and geoinformatics symposiums. He also serves as an Associate Editor of the *International Journal of Ambient Computing and Intelligence*, *International Journal of Image and Graphics*, *International Journal of Digital Multimedia Broadcasting*, *Journal of Geo-spatial Information Science*, and *Journal of Remote Sensing*, and the Guest Editor of the *Journal of Applied Remote Sensing* and *Journal of Sensors*. He is currently serving as an Associate Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. He received the best reviewer awards from the IEEE Geoscience and Remote Sensing Society (GRSS) for his service to the IEEE JOURNAL OF SELECTED TOPICS IN EARTH OBSERVATIONS AND APPLIED REMOTE SENSING (JSTARS) in 2012 and the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS (GRSL) in 2014. He was the General Chair for the 4th IEEE GRSS Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS) and the Guest Editor of JSTARS. His research teams won the top three prizes of the IEEE GRSS 2014 Data Fusion Contest, and his students have been selected as the winners or finalists of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS) student paper contest in recent years. He was a recipient of the 2010 Best Paper Boeing Award and the 2013 Best Paper ERDAS Award from the American Society of Photogrammetry and Remote Sensing (ASPRS).