

# Multilevel Cloud Detection in Remote Sensing Images Based on Deep Learning

Fengying Xie, Mengyun Shi, Zhenwei Shi, *Member, IEEE*, Jihao Yin, *Senior Member, IEEE*, and Danpei Zhao

**Abstract**—Cloud detection is one of the important tasks for remote sensing image processing. In this paper, a novel multilevel cloud detection method based on deep learning is proposed for remote sensing images. First, the simple linear iterative clustering (SLIC) method is improved to segment the image into good quality superpixels. Then, a deep convolutional neural network (CNN) with two branches is designed to extract the multiscale features from each superpixel and predict the superpixel as one of three classes including thick cloud, thin cloud, and noncloud. Finally, the predictions of all the superpixels in the image yield the cloud detection result. In the proposed cloud detection framework, the improved SLIC method can obtain accurate cloud boundaries by optimizing initial cluster centers, designing dynamic distance measure, and expanding search space. Moreover, different from traditional cloud detection methods that cannot achieve multilevel detection of cloud, the designed deep CNN model can not only detect cloud but also distinguish thin cloud from thick cloud. Experimental results indicate that the proposed method can detect cloud with higher accuracy and robustness than compared methods.

**Index Terms**—Cloud detection, convolutional neural network (CNN), deep learning, remote sensing images, superpixel.

## I. INTRODUCTION

WITH the rapid development of remote sensing technology, remote sensing images are widely applied to various fields such as military target recognition, environment monitoring, meteorology, mineral development, and geographical mapping. However, 50% of the Earth's surface is covered by cloud at any time [1]. Cloud often appears and covers objects on the surface in remote sensing images, which makes much difficulty to many image analysis tasks and causes inaccurate analysis results [2], [3]. Hence, detecting and removing cloud regions are highly necessary to improve the availability of remote sensing images.

Many cloud detection methods have been proposed. These methods can be roughly classified into two main categories: threshold-based ones and machine-learning-based ones. They

Manuscript received June 29, 2016; revised November 2, 2016; accepted March 16, 2017. This work was supported by the National Natural Science Foundation of China and the Chinese Academy of Sciences Joint Fund of Astronomy under Grant 61471016 and Grant U1331108. (*Corresponding author: Fengying Xie.*)

The authors are with the Beijing Key Laboratory of Digital Media and the Image Processing Center, School of Astronautics, Beihang University, Beijing 100191, China (e-mail: xfy\_73@buaa.edu.cn; fox.smy@gmail.com; shizhenwei@buaa.edu.cn; yjh@buaa.edu.cn; zhaodanpeibuaa@163.com).

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/JSTARS.2017.2686488

extract a variety of manual features pixel by pixel, followed by setting a threshold or learning a binary classifier to determine whether this pixel belongs to cloud area or not. Cihlar and Howarth [4] used the value of the temporal normalized difference vegetation index profile to detect cloud-contaminated pixels and optionally replaced these with interpolated values in NOAA/AVHRR images. Girolamo and Davies [5] used a two-step direct threshold technique to separate clear and cloudy pixels in degraded AVIRIS data. Jedlovec *et al.* [6] used the bispectral composite threshold technique with the 20-day composites of two channels' imagery for cloud detection. The threshold methods have rapid computing speed, but the structure and texture of cloud are neglected and the detection results rely on the sensors. Machine learning methods with more robustness gain popularity. In [7], Le Hégarat-Masclé and André detected cloud by using the Markov random field framework to formalize the physical properties of cloud. Rossi *et al.* [8] extracted the features of cloud by using singular value decomposition and identified the cloud cover by support vector machine (SVM) in Quickbird images. Li *et al.* [9] trained an SVM classifier using the brightness and texture features to detect cloud. Yuan and Hu [10] proposed a cloud detection method based on object classification using image features extracted by the bag-of-words model. Zhang and Xiao [11] developed a progressive refinement scheme based on the color properties derived from observations and statistical results to extract cloud regions from color aerial photographs. An and Shi [12] proposed a supervised approach based on the scene-learning scheme and designed a cloud detector to detect cloud on the whole image scenes. Generally, machine learning methods yield more accurate cloud detection results than threshold ones.

In remote sensing images, ground objects are diverse and the thickness and forms of cloud are varied, and there are many easily confusing objects such as white buildings, planes, and snow. Therefore, cloud detection in remote sensing images is quite challenging. Although many works have been developed for cloud detection, most of the existing methods cannot work well on thin cloud. Thin cloud is semitransparent and the contrast between cloud and background is weak; hence, it is easily missed. In addition, the multilevel cloud detection cannot be achieved by the existing methods. The detected cloud region can be removed through image reconstruction/restoration technology [13]. Generally, the thick cloud regions are reconstructed by a sequence of temporal optical images, and the thin cloud is removed through image enhancement techniques like Retinex [14]. Therefore, the removal schemes for them are very

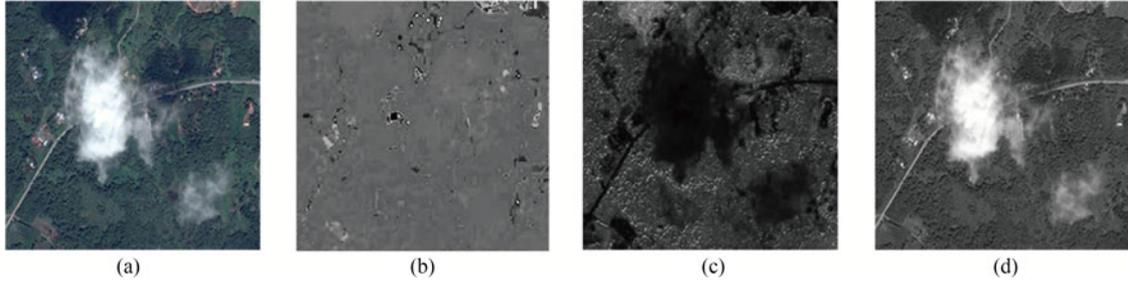


Fig. 1. HSI color space of an image. (a) Original RGB color image. (b) Hue component image. (c) Saturation component image. (d) Intensity component image.

different. However, the existing cloud detection methods cannot distinguish thin cloud from thick cloud, which is important for automatic cloud removal and other image analysis tasks.

Deep learning methods can mine high-level features and have improved effectiveness of many computer vision tasks. In this paper, a novel cloud detection method based on deep learning is proposed for remote sensing images. The simple linear iterative clustering (SLIC) method is first improved to segment the image into superpixels with accurate image boundaries. Then, a deep convolutional neural network (CNN) with two branches is designed to predict these superpixels as thick cloud, thin cloud, or noncloud. And the predictions of all the superpixels in the image yield the final cloud detection result.

In summary, the proposed method has the following three main contributions.

1) A multilevel cloud detection framework is proposed. The proposed framework combines the superpixel method with deep learning to detect cloud and distinguish thin cloud from thick cloud.

2) The SLIC algorithm is improved through optimizing initial cluster centers, designing dynamic distance measure, and expanding search space. With the improved SLIC algorithm, accuracy cloud boundaries are obtained.

3) A new CNN architecture with two branches is designed, which can extract multiscale features and detect thick cloud and thin cloud effectively even in complex circumstance.

The remainder of this paper is organized as follows. Section II describes the superpixel segmentation. In Section III, cloud detection using the designed deep CNN is introduced. Section IV presents and analyzes experimental results. Finally, Section V gives the conclusion.

## II. SUPERPIXEL SEGMENTATION

Superpixel algorithms [15] group pixels into perceptually meaningful atomic regions, which can be used to replace the rigid structure of the pixel grid and greatly reduce the complexity of subsequent image processing tasks such as depth estimation [16], segmentation [17], body model estimation [18], and object localization [19]. In this paper, the image is segmented into superpixels, which are used as basic units to detect cloud. As a widely used superpixel method, the SLIC algorithm [15] can output good quality superpixels that are compact and roughly equally sized. However, because SLIC obtains initial cluster centers through dividing the image into several equal-size grids

and its search space is limited to a local region, the produced superpixels cannot adhere to weak cloud boundaries well and the smooth thick cloud region will be over segmented. In this paper, the SLIC algorithm is improved from three aspects: optimizing initial cluster centers, designing dynamic distance measure, and expanding search space. By using the improved SLIC method, better quality superpixels can be generated for remote sensing images.

### A. Color Space Transformation

The color of cloud is white and bright, with high intensity and low saturation. Similar to [11] and [12], the color space transformation to the hue, saturation, intensity (HSI) [20], [21] color model is first performed. For 1 pixel, the transformation from RGB to HSI color model is defined as

$$H = \begin{cases} \theta, & B \leq G \\ 360 - \theta, & B > G \end{cases} \quad (1)$$

$$S = 1 - \frac{3 \times \min(R, G, B)}{R + G + B} \quad (2)$$

$$I = \frac{R + G + B}{3} \quad (3)$$

$$\theta = \cos^{-1} \left\{ \frac{[(R - G) + (R - B)]/2}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right\} \quad (4)$$

where  $R$ ,  $G$ , and  $B$  are the values of red, green, and blue channels of input image, and  $H$ ,  $S$ , and  $I$  are the values of hue, saturation, and intensity components in the HSI space, respectively. Fig. 1 shows HSI color space of an example image. It can be seen that the cloud region is salient in  $S$  and  $I$  components. Therefore,  $S$  and  $I$  components are used in our improved superpixel method.

### B. Initial Cluster Center Optimization

SLIC obtains initial cluster centers by using equal-size grids, which does not take account of image content. In [22], a graph-based superpixel approach GS04 was developed, which can generate big superpixels in smooth regions and small superpixels in complex texture regions, but their shapes are very irregular. In this paper, we used the GS04 method to generate initial cluster centers.

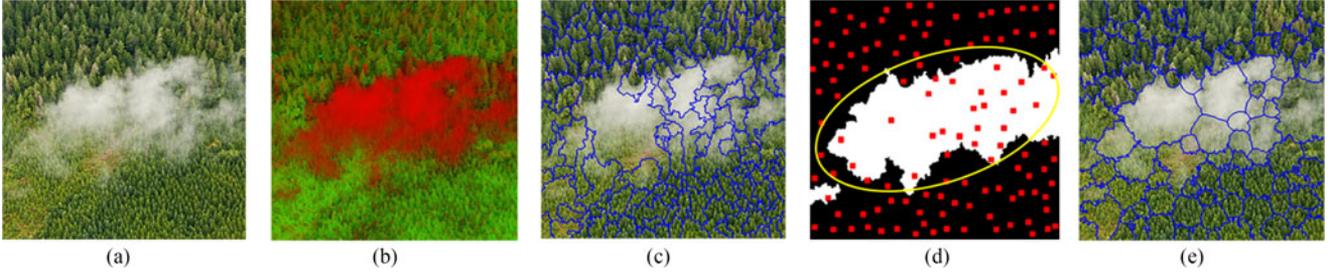


Fig. 2. Instance of initial cluster center generation. (a) Original RGB image. (b) Transformed image with S and I channels. (c) Subregions obtained by GS04. (d) Initial cluster centers. (e) Final superpixel result.



Fig. 3. Flowchart of our improved SLIC.

We define the dissimilarity measure between the  $i$ th and  $j$ th pixels as

$$\omega(p_i, p_j) = \sqrt{(S_i - S_j)^2 + (I_i - I_j)^2} \quad (5)$$

where  $p_i$  is the  $i$ th pixel, and  $S$  and  $I$  are two values of saturation and intensity components in the HSI space.

In the initial graph, each pixel denotes a component. Let  $\text{Int}(C)$  be the internal difference of component  $C$ , and  $\text{Dif}(C_1, C_2)$  be the difference between two neighboring components  $C_1$  and  $C_2$ :

$$\text{Int}(C) = \max_{p_i \in C, p_j \in C} \omega(p_i, p_j) \quad (6)$$

$$\text{Dif}(C_1, C_2) = \min_{p_i \in C_1, p_j \in C_2} \omega(p_i, p_j). \quad (7)$$

According to [22], the pairwise comparison predicate  $D(C_1, C_2)$  is defined as

$$D(C_1, C_2) = \begin{cases} \text{true}, & \text{Dif}(C_1, C_2) > \text{MInt}(C_1, C_2) \\ \text{false}, & \text{otherwise} \end{cases} \quad (8)$$

$$\text{MInt}(C_1, C_2) = \min(\text{Int}(C_1) + \tau(C_1), \text{Int}(C_2) + \tau(C_2)) \quad (9)$$

where  $\tau(C) = k/|C|$  is the threshold function,  $|C|$  denotes the size of  $C$ , and  $k$  is a constant parameter; in this paper,  $k = 50$ .

The merge process is started with the initial graph. Then, final subregions are gradually produced by using the region comparison predicate  $D(C_1, C_2)$  to merging similar components in loop steps.

Fig. 2 is an instance of initial cluster center generation, where Fig. 2(c) is subregions generated by carrying out the GS04 method on the image in the HSI space. It can be seen that these produced subregions adhere well to image boundaries, but their sizes and shapes are very irregular. We filter out small subregions with a threshold 500, and geometric center pixels of the remained subregions are taken as initial cluster centers [see the red dots in Fig. 2(d)]. It can be seen that initial cluster

centers are sparse in the smooth cloud region and dense in the background region with a complex texture.

### C. Iterative Clustering

Next, iteration clustering is carried out in the 4-D space  $[S, I, x, y]^T$ , where  $[S, I]^T$  are the values of saturation and intensity components in the HSI space and  $[X, Y]^T$  are the position coordinates of a pixel.

In the assignment step, each pixel  $p_i$  is associated with the nearest cluster center through comparing it with all cluster centers by using distance measure  $D$ , which will be discussed in Section II-D. In [15], the search for the nearest cluster center is limited to a local region in order to speed up the algorithm, which constrains the flexibility of superpixels in boundary and size. Therefore, we search the entire image to determine the nearest cluster center for the clustering pixel in this paper.

Once each pixel has been associated with the nearest cluster center, an update step will adjust the cluster centers to be the mean  $[S, I, x, y]^T$  vector of all pixels belonging to the cluster. The  $L_2$  norm is used to compute a residual error  $E$  between the new cluster center locations and previous cluster center locations. The assignment and update steps will be repeated iteratively until the error converges.

### D. Distance Measure Design

Generally, a weighted distance measure combining color and spatial proximity is needed in superpixel segmentation. In this paper, the distance measure between the  $i$ th pixel and the  $j$ th cluster center  $c_j$  is defined as

$$D = D_C + \frac{\alpha}{\text{Size}} D_S \quad (10)$$

$$D_C = \sqrt{(S_i - S_{c_j})^2 + (I_i - I_{c_j})^2} \quad (11)$$

$$D_S = \sqrt{(x_i - x_{c_j})^2 + (y_i - y_{c_j})^2} \quad (12)$$

where  $\alpha$  is a constant parameter and  $\text{Size}$  is the area of the  $j$ th cluster in current loop.

The parameter  $\alpha$  is used to control the relative importance between color similarity and spatial proximity. A bigger  $\alpha$  gives more relative importance to spatial distance than color proximity, which can generate more compact superpixels. When  $\alpha$  is small, the resulting superpixels will adhere more tightly to

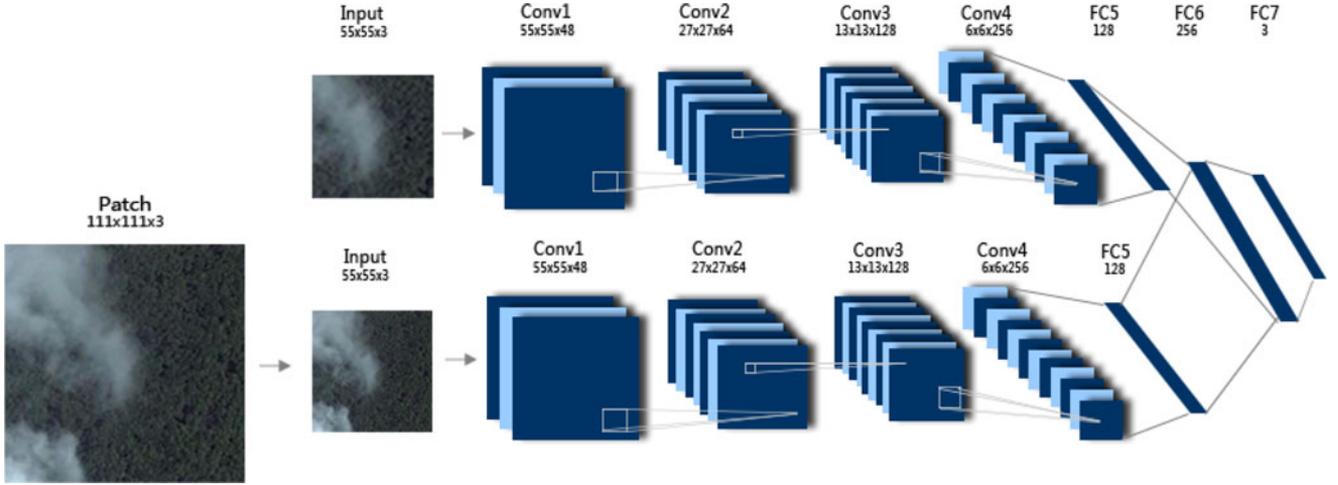


Fig. 4. Architecture of our designed CNN.

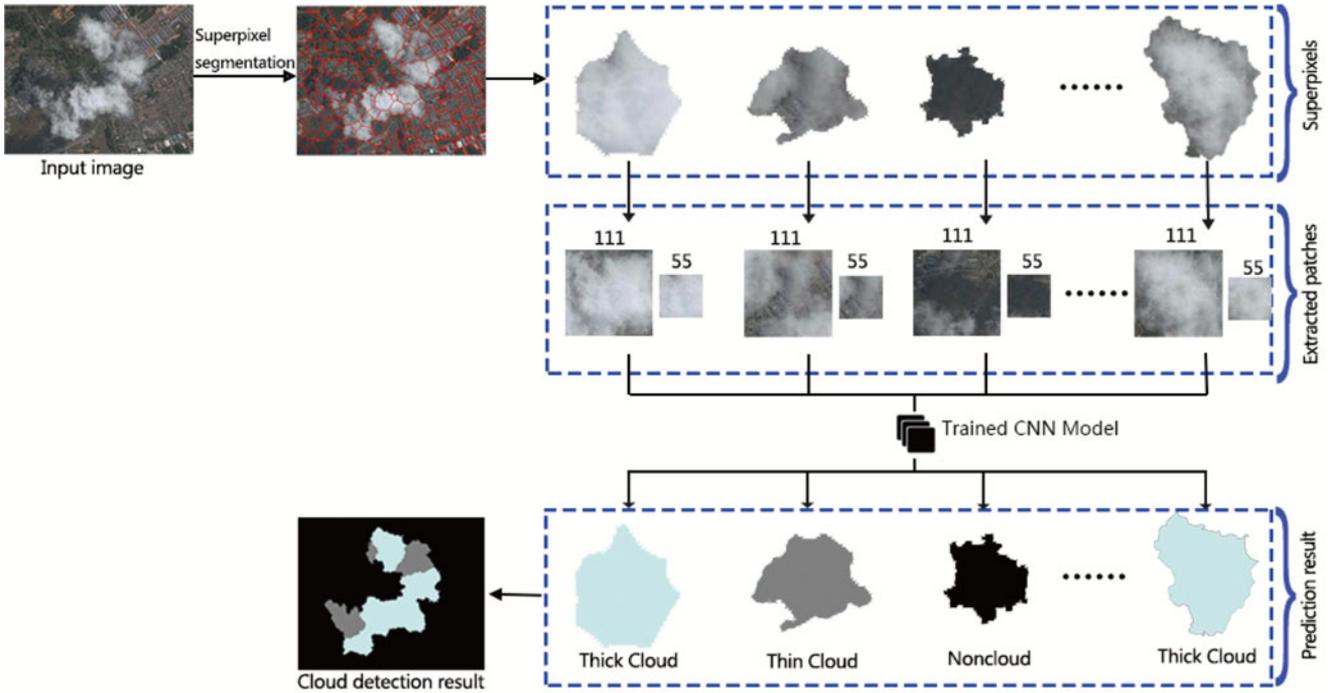


Fig. 5. Processing chain of cloud detection in the proposed framework.

image boundaries, but have less regular shape. In this paper,  $\alpha$  is set to 8000. As for parameter *Size*, different from original SLIC [15] in which *Size* is a constant parameter, *Size* is changed with iterative clustering and used for adjusting the influence of the spatial distance  $D_S$  on the clustering pixel when searching for nearest distance center. When the compared cluster center belongs to a large cluster region, *Size* can weaken the influence of  $D_S$ , which can ensure that the pixel far from its cluster center is segmented into the correct cluster, and avoid small superpixels in large homogenous regions (over-segmentation). And when the compared cluster center belongs to a small cluster region, the influence of  $D_S$  is enhanced, which can reduce the

probability of big superpixels occurring in complex texture regions (undersegmentation). Therefore, the designed distance measure can dynamically balance the relative importance between color proximity and spatial proximity in each iteration to generate good quality superpixels.

Fig. 3 shows the flowchart of our superpixel method. The initial cluster centers are first obtained through the GS04 method using S and I component values of the HSI color space, and then, the assignment and update steps are repeated iteratively in the 4-D space  $[S, I, x, y]^T$  using the designed dynamic distance measure  $D$  to generate the final superpixels. Fig. 2(e) shows the superpixel segmentation result generated from Fig. 2(a) by our

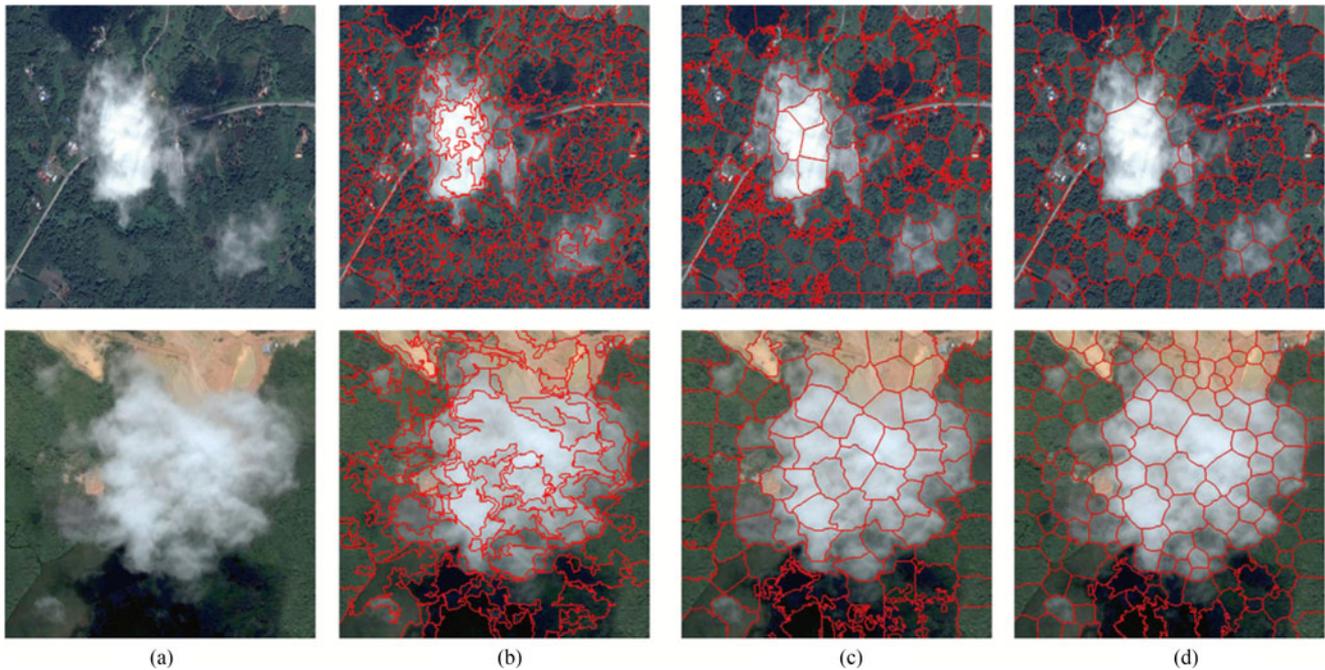


Fig. 6. Superpixel segmentation instance. (a) Original image. (b) GS04 [22]. (c) SLIC [15]. (d) Our improved SLIC.

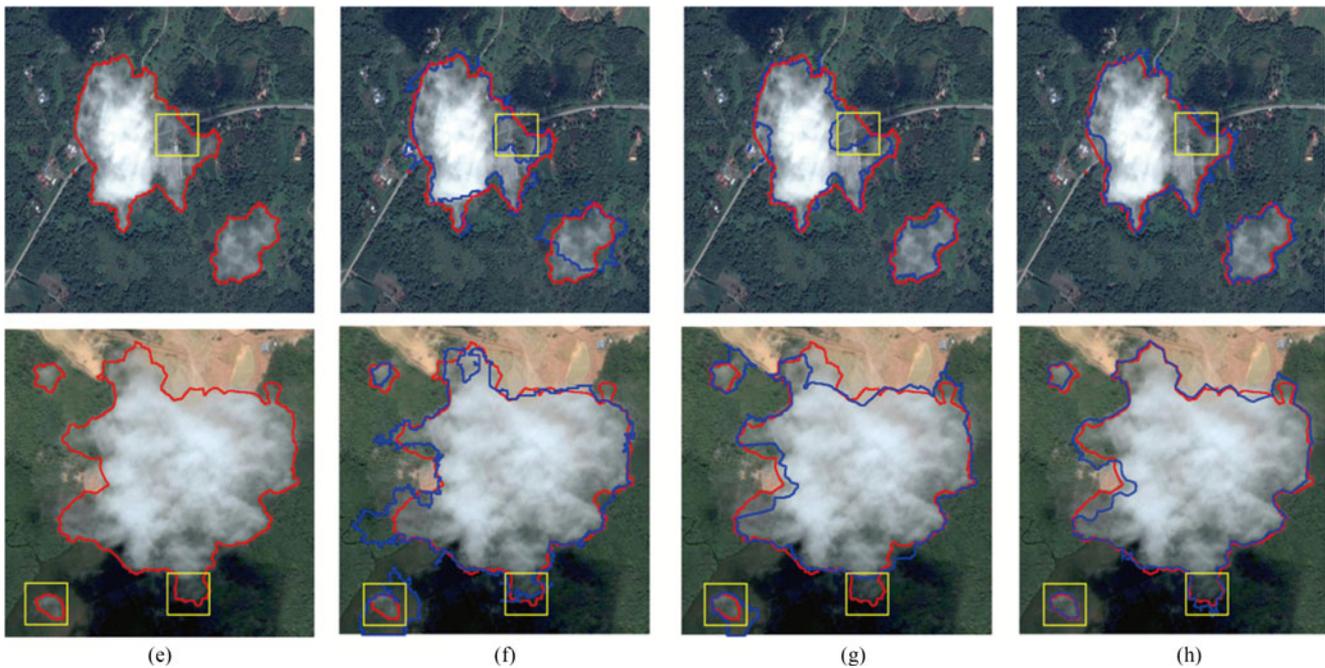


Fig. 7. Cloud detection results for the two images in Fig. 6 using different superpixel methods. (a) Original image. (b) GS04 [22]. (c) SLIC [15]. (d) Our improved SLIC.

method. It can be seen that the resulting superpixels are compact and adhere to cloud boundaries very well.

### III. CLOUD DETECTION

As a deep learning method, a CNN has improved the performance dramatically for a wide range of computer vision tasks

such as image classification [23], saliency detection [24], object detection [25], and superresolution [26]. In [23], a single-branch CNN is designed for image classification, which cannot extract multiscale features of interesting objects. In [24], a double-branch CNN is employed to model saliency of objects in images, which only solve binary classification problem and cannot be used for multi-level cloud detection. In this paper, a deep

TABLE I  
STATISTICS OF DIFFERENT SUPERPIXEL METHODS

Methods	Precision <sub>c</sub>	Recall <sub>c</sub>	Precision <sub>t</sub>	Recall <sub>t</sub>	Precision	Recall
GS04	<b>0.8972</b>	0.8230	0.4039	0.4516	0.8895	0.8567
SLIC	0.8610	0.8470	0.4887	0.5195	0.8944	0.9052
Proposed	0.8759	<b>0.8947</b>	<b>0.5547</b>	<b>0.6141</b>	<b>0.8945</b>	<b>0.9356</b>

CNN with two branches is designed to mine multiscale features to classify superpixels (image patches) as one of three classes including thick cloud, thin cloud, and noncloud.

#### A. Designed CNN Architecture

As shown in Fig. 4, the overall architecture of the designed CNN contains two branches with four convolutional layers and one fully connected layer, and two branches are followed by two fully connected layers. For convolutional layers, the size of the feature map is defined as width  $\times$  height  $\times$  depth, where the first two dimensions describe the spatial size and the depth defines the number of channels. Consider CONV# as a convolutional layer, LRN# as a local response normalization, MAXP# as a max pooling layer, ReLU# as a nonlinear rectified linear unit function [27], FC# as a fully connected layer, and Dropout# as a dropout layer [28]. The structure of the network can be described concisely by the sizes of feature maps at each layer. Both branches have the same structure, which can be described as

CONV1( $55 \times 55 \times 48$ )  $\rightarrow$  LRN1  $\rightarrow$  MAXP1  $\rightarrow$   
 CONV2( $27 \times 27 \times 64$ )  $\rightarrow$  RELU2  $\rightarrow$  LRN2  $\rightarrow$   
 MAXP2  $\rightarrow$  CONV3( $13 \times 13 \times 128$ )  $\rightarrow$   
 CONV4( $6 \times 6 \times 256$ )  $\rightarrow$  MAXP4  $\rightarrow$   
 FC5(128)  $\rightarrow$  RELU5  $\rightarrow$  Dropout5.

In the two branches, the size of filters is  $5 \times 5$  in CONV1 and CONV2 and  $3 \times 3$  in CONV3 and CONV4, and the max pooling layers perform max pooling over  $3 \times 3$  spatial neighborhoods with a stride of 2 pixels on the output of the convolution layers.

The two branches are followed by two fully connected layers. In this paper, the output of the last fully connected layer indicate the probabilities of the input patch, which belongs to cloud, thin cloud, and noncloud, respectively. It means that the unit number of the output layer is 3. Therefore, the structure of the last two layers can be described as

FC6(256)  $\rightarrow$  RELU6  $\rightarrow$  Dropout6  $\rightarrow$  FC7(3).

#### B. Training and Detection

In the training stage, a couple of patches with size of  $55 \times 55$  and  $111 \times 111$  centered at each trained pixel are extracted and their three-channel RGB values are inputted into the designed CNN, where the small patch is for the first branch and the big one, resized to  $55 \times 55$  before inputted into the CNN, is for the second branch. Through the two branches of the designed CNN, the high-level features are extracted at two scales. Through training, a CNN classifier with three class predictions is generated for cloud detection.

TABLE II  
STATISTICS OF DIFFERENT CNNs

Methods	Precision <sub>c</sub>	Recall <sub>c</sub>	Precision <sub>t</sub>	Recall <sub>t</sub>	Precision	Recall
AlexNet	0.8393	0.8956	0.5262	0.4168	0.8978	0.8893
Our CNN (55)	0.8759	0.8947	0.5547	0.6141	0.8945	0.9356
Our CNN (111)	0.8733	0.8571	0.5111	0.5828	0.9005	0.9185
Our CNN (55 and 111)	<b>0.9026</b>	<b>0.9253</b>	<b>0.6379</b>	<b>0.6672</b>	<b>0.9039</b>	<b>0.9454</b>

In this paper, the weights in each layer were initialized from a zero-mean Gaussian distribution with standard deviation 0.01 and the neuron biases in the second and fourth convolutional layers as well as in the fully connected hidden layers were initialized with the constant 0.1. We used stochastic gradient descent with a minibatch size of 256. The weight decay and momentum were set to 0.0001 and 0.9, respectively. The learning rate started from 0.03 and was divided by 10 when the error plateaus, and the models were trained for up to  $50 \times 800$  iterations.

In the detection stage, superpixels are first obtained from the testing image using the improved superpixel method described in Section II. For each superpixel, a couple of image patches ( $55 \times 55$  and  $111 \times 111$ ) centered at its geometric center pixel are extracted and inputted into the trained CNN model, and thus, the class of the superpixel is predicted. Merging the predictions of all superpixels in the testing image, the final cloud detection result is obtained. The procedure of cloud detection is illustrated in Fig. 5.

## IV. EXPERIMENT RESULTS

The proposed algorithm is implemented by using C++ on the PC with Intel CPU i7-4790k at 4 GHz and IGABYTE GV-N970WF3OC-4GD GPU, and the designed deep network is implemented through the software library Caffe [29]. A total of 81 experimental images with sizes ranging from  $500 \times 400$  to  $800 \times 600$  are obtained from the satellite Quickbird (<http://glcf.umd.edu/data/quickbird/>) with the spatial resolution about 2.44–2.88 m, Flickr.com, and the Google map. Of these experimental images, 38 images are used for training and 43 for test. The ground truth of these images is obtained manually. Centering at each pixel in training images, we extract a couple of patches with size of  $55 \times 55$  and  $111 \times 111$ , respectively. In this way, 204 000 couples of patches are obtained from the training set, where the number of cloud, thin cloud, and noncloud patches are 26 000, 22 000, and 156 000, respectively. For a testing image, superpixels are first obtained by the improved superpixel method, and a couple of image patches are then extracted from each superpixel and inputted into the trained CNN model to predict the class of this superpixel, and the final cloud detection result of the testing image is achieved by using the predictions of all its superpixels.

#### A. Effectiveness of the Improved Superpixel Method

In the proposed cloud detection framework, the SLIC method is used to cluster the image into small subregions, which is

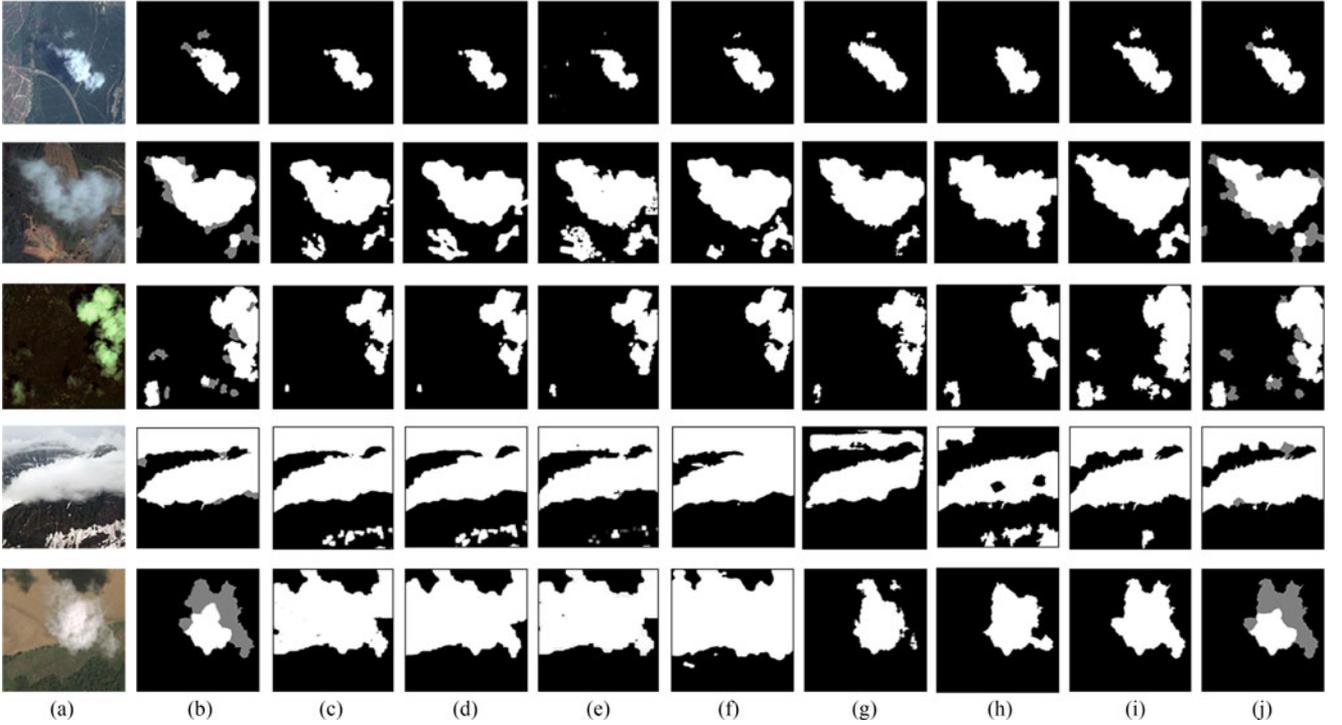


Fig. 8. Visual comparisons of different cloud detection methods. (a) Original image. (b) Ground truth. (c) K-means. (d) Mean-shift. (e) Chan-Vese. (f) Method of [11]. (g) Method of [12]. (h) Original method of [24]. (i) Improved method of [24]. (j) Our proposed framework.

TABLE III  
DETECTION PERFORMANCE OF DIFFERENT METHODS FOR THICK  
CLOUD AND THIN CLOUD

Methods	Recall <sub>c</sub>	Recall <sub>t</sub>
K-means	0.8753	0.2161
Mean-shift	0.8836	0.2478
Chan-Vese	0.8861	0.3474
Method of [11]	0.9091	0.3559
Method of [12]	0.8981	0.4725
Original method of [24]	0.8820	0.5632
Improved method of [24]	0.9774	0.8117
Proposed method	<b>0.9780</b>	<b>0.8467</b>

improved through optimizing initial cluster centers, designing dynamic distance measure, and expanding search space. In order to verify the effectiveness of the improved SLIC method, we compare it with SLIC [15] and GS04 [22].

Fig. 6 shows two superpixel segmentation instances, where the cloud is thick and smooth in the first row, and thin and unsmooth in the second row. It can be seen that SLIC and our improved method can obtain more compact and regular superpixels than the GS04 method. While compared with SLIC, since our designed distance measure can dynamically adjust the relative importance between color proximity and spatial proximity in each iteration and the search of the nearest cluster center is in the entire image, our superpixel method can not only avoid oversegmentation in large homogenous regions but also suppress noise superpixels.

We use cloud detection results to evaluate the three superpixel methods. Fig. 7 shows the cloud detection results for the two

images in Fig. 6 using the same single branch CNN structure with  $55 \times 55$  input (see the first branch of Fig. 4) combined with different superpixel methods, where the red line represents ground truth and the blue line is the cloud detection result. It is obvious that all methods can detect the most of the cloud. However, for the weak cloud boundaries and isolated cloud regions, our improved superpixel method can achieve more accurate results (see the yellow boxes in Fig. 7) because our method can obtain suitable initial cluster centers in thin cloud region and the size of the superpixel can adaptively change with image content, and the produced superpixels are more easy to adhere to weak cloud boundaries.

Two metrics precision and recall are used to evaluate the performance of cloud detection using different superpixel methods, which are defined as follows:

$$\text{precision} = \frac{CP}{DP} \quad (13)$$

$$\text{recall} = \frac{CP}{GN} \quad (14)$$

where CP is the number of pixels correctly detected as cloud, DP is the number of pixels detected as cloud, and GN is the number of cloud pixels in ground truth. A better cloud detection method has higher precision and recall. By using 43 test images, we calculate the average precision and recall for different superpixel methods. Table I shows statistical results, where Precision<sub>c</sub> and Recall<sub>c</sub> are results of thick cloud, Precision<sub>t</sub> and Recall<sub>t</sub> represent results of thin cloud, and Precision and Recall are for the entire cloud region (including thick cloud and thin cloud).

Obviously, our improved superpixel method has better precision and recall values than compared methods.

### B. Performance of Different CNN Architectures

In this paper, a double-branch CNN is designed to detect thick cloud and thin cloud, which can mine the features of cloud at two scales. We compare our double-branch CNN with three single-branch CNNs including Alexnet [23] and our two single-branch CNNs with  $55 \times 55$  input (the first branch in Fig. 4) and  $111 \times 111$  input (the second branch in Fig. 4), respectively. Table II shows statistical results for different CNN architectures combined with our superpixel method on the test set. It can be seen that our double-branch CNN has the best metric values whatever for thick cloud, thin cloud, or entire cloud. Therefore, our double-branch CNN has more effective cloud detection results compared with three single-branch CNNs.

### C. Comparison With Other Cloud Detection Methods

In this paper, the image is first segmented into superpixels using our improved SLIC method, and the classes of these superpixels are then predicted using the designed CNN model to obtain the cloud detection result. The proposed framework can detect two-level cloud including thick cloud and thin cloud at the same time. We compare our cloud detection framework with K-means [30], Mean-shift [31], Chan-Vese [32], [33], and two cloud detection methods in [11] and [12].

In [24], a double-branch CNN is used to model saliency of objects in images. The input of the upper branch is a superpixel-centered large context window including the full image to extract the global context features, and the lower branch focuses on a local context to refine the prediction. In this paper, the method of [24] is also compared. In addition, in order to verify the effectiveness of the CNN architecture of [24] for cloud detection, we improved the method of [24] by replacing the inputs of the two branches using our image patches with the size of  $111 \times 111$  and  $55 \times 55$ , respectively, and the results are given.

Fig. 8 shows some example results using different cloud detection methods, where Fig. 8(b) shows ground truth (white region represents thick cloud and gray region is thin cloud). Because these compared methods cannot differentiate thin cloud from thick cloud, results by them only have white regions, which denote the detected entire cloud. The first row in Fig. 8 is a simple case. The cloud is thick and the contrast between cloud and background is strong, and all methods can obtain satisfactory detection result. From the second row to the last row, tested images are complex, in which there are bright background, snow (see the forth row), thin, and even semitransparent cloud. For these complex situations, the last two methods based on the CNN have better detection results.

In order to evaluate the detection performance of these methods for thick cloud and thin cloud, recall values are calculated on the 43 test images for thick cloud and thin cloud, respectively. Statistic results are given in Table III. Because the compared methods cannot distinguish thin cloud from thick cloud, the metric precision is not given here. The seventh row is the statistic results of the method of [24], as can be seen that the

TABLE IV  
STATISTICS OF DIFFERENT CLOUD DETECTION METHODS

Methods	RR	ER	FAR	RER
K-means	0.6924	0.0862	0.0165	8.0279
Mean-shift	0.7087	0.0869	0.0212	8.1578
Chan-Vese	0.7388	0.0900	0.0254	8.2075
Method of [11]	0.7568	0.0623	<b>0.0135</b>	12.1415
Method of [12]	0.7787	0.0613	0.0165	12.4978
Original method of [24]	0.8032	0.0796	0.0199	10.0873
Improved method of [24]	0.9322	0.0396	0.0221	23.5285
Proposed method	<b>0.9454</b>	<b>0.0330</b>	0.0189	<b>28.6067</b>

CNN-based method has obvious superiority in thin cloud than the five non-CNN-based methods. And when we replace the original inputs of the CNN using our multiscale image patches, the detection results are improved greatly for both thick cloud and thin cloud, see the eighth row, which means that our multiscale features are more effective for cloud detection than the multicontext features of [24]. However, the improved method of [24] still cannot distinguish thin cloud from thick cloud. On the contrary, our method with highest recall values can distinguish thin cloud from thick cloud. Therefore, our method is superior to the compared methods.

We evaluate the algorithm performance for the entire cloud detection. Here, four metrics are used including the right rate (RR), the error rate (ER), the false alarm rate (FAR), and the ratio of RR to ER (RER). RR is the same with metric recall, and other three metrics are defined as

$$ER = \frac{CN + NC}{TN} \quad (15)$$

$$FAR = \frac{NC}{GN} \quad (16)$$

$$RER = \frac{RR}{ER} \quad (17)$$

where CN is the number of cloud pixels detected as noncloud pixels, NC is the number of noncloud pixels detected as cloud pixels, TN is the total number of pixels in the input image, and GN is the number of cloud pixels in ground truth.

A good cloud detection method has high values of RR and RER and low values of ER and FAR. Table IV presents the average values of four metrics for the 43 test images. It can be seen that our method has the best values in RR, ER, and RER except for FAR. Especially RER metric, our method is higher at least 5.1 than compared methods. Considering that our method has the best detection accuracy and can achieve multilevel detection of cloud, our method outperforms other compared methods greatly.

### D. Time Complexity

In this paper, a two-branch CNN is combined with the superpixel method to detect cloud. The size of feature maps in the designed CNN is small, which reduces the computation complexity. In addition, more runtime is saved through predicting

TABLE V  
RUNTIME STATISTICS OF CLOUD DETECTION UNDER DIFFERENT SITUATIONS  
ON TEST SET

Situations	Superpixel segmentation (s)	CNN prediction (s)	Total time (s)
Situation 1	0	474	474
Situation 2	4.62	5.26	9.88
Situation 3	4.62	0.26	4.88

the class of the superpixels to achieve the final cloud detection result. In [24], a two-branch CNN is used to model saliency of objects in images, which has more layers and bigger maps than our proposed CNN architecture. In [12], each pixel is predicted to detect cloud, which is time consuming. Table V illustrates the runtime for three situations on the 43 test images, including the following:

*Situation 1:* remove superpixel step from our framework and predict each pixel in the image to obtain the final cloud detection result.

*Situation 2:* combine the CNN of [24] with our proposed superpixel method to detect cloud.

*Situation 3:* our final cloud detection framework.

It can be seen from Table V that our final detection framework has the fastest speed with 4.83 s per image. Therefore, through superpixel preprocessing and size reduction of feature maps of the CNN, the runtime is saved greatly.

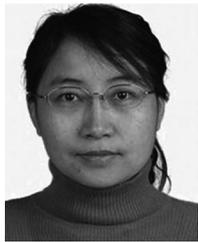
## V. CONCLUSION

Cloud often appears and covers the objects on the surface in remote sensing images, which will make much difficulty to target detection, object recognition, and other tasks. Usually, the treatment for thin cloud is different from thick cloud in some image analysis tasks, such as cloud removal and target detection. Therefore, it is useful to distinguish thin cloud from thick cloud in cloud detection. However, the multilevel detection of cloud is still not addressed in spite of the fact that many cloud detection methods have been proposed. In this paper, a novel multilevel cloud detection method based on the deep CNN is proposed for remote sensing images. The image is first segmented into superpixels using the improved SLIC method. A couple of image patches are then extracted from each superpixel and inputted to the designed deep CNN to predict the class of this superpixel as thick cloud, thin cloud, or noncloud. For an image, the final cloud detection result is obtained using the predictions of all its superpixels. In the proposed cloud detection framework, the SLIC method is improved through optimizing initial cluster centers, designing dynamic distance measure, and expanding search space, which can obtain accurate cloud boundaries. Moreover, the CNN with two branches is designed to extract multiscale features, which can detect thick cloud and thin cloud effectively even in complex circumstance. Qualitative and quantitative analysis are implemented in the experimental stage. Experimental results demonstrate that our proposed framework can achieve multilevel cloud detection, and the result is more accurate and robust than that of the compared methods.

## REFERENCES

- [1] G. W. Paltridge and C. M. R. Platt, *Radiative Processes in Meteorology and Climatology*. New York, NY, USA: Elsevier, 1976.
- [2] R. Saunders, "An automated scheme for the removal of cloud contamination from AVHRR radiances over western europe," *Int. J. Remote Sens.*, vol. 7, no. 7, pp. 867–886, 1986.
- [3] R. W. Saunders and K. T. Kriebel, "An improved method for detecting clear sky and cloudy radiances from AVHRR data," *Int. J. Remote Sens.*, vol. 9, no. 1, pp. 123–150, 1988.
- [4] J. Cihlar and J. Howarth, "Detection and removal of cloud contamination from AVHRR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 3, pp. 583–589, May 1994.
- [5] L. D. Girolamo and R. Davies, "The image navigation cloud mask for the multiangle imaging spectroradiometer (MISR)," *J. Atmos. Ocean. Technol.*, vol. 12, no. 6, pp. 1215–1228, 1995.
- [6] G. J. Jedlovec, S. L. Haines, and F. J. LaFontaine, "Spatial and temporal varying thresholds for cloud detection in goes imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 6, pp. 1705–1717, Jun. 2008.
- [7] S. Le Hégarat-Masclé and C. André, "Use of Markov random fields for automatic cloud/shadow detection on high resolution optical images," *ISPRS J. Photogramm. Remote Sens.*, vol. 64, no. 4, pp. 351–366, 2009.
- [8] R. Rossi, R. Basili, F. Del Frate, M. Luciani, and F. Mesiano, "Techniques based on support vector machines for cloud detection on quickbird satellite imagery," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2011, pp. 515–518.
- [9] P. Li, L. Dong, H. Xiao, and M. Xu, "A cloud image detection method based on SVM vector machine," *Neurocomputing*, vol. 169, pp. 34–42, 2015.
- [10] Y. Yuan and X. Hu, "Bag-of-words and object-based classification for cloud extraction from satellite imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 8, pp. 4197–4205, Aug. 2015.
- [11] Q. Zhang and C. Xiao, "Cloud detection of RGB color aerial photographs by progressive refinement scheme," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 11, pp. 7264–7275, Nov. 2014.
- [12] Z. An and Z. Shi, "Scene learning for cloud detection on remote-sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 8, pp. 4206–4222, Aug. 2015.
- [13] F. Melgani, G. Mercier, L. Lorenzi, and E. Pasolli, "Recent methods for reconstructing missing data in multispectral satellite imagery," in *Applications+ Practical Conceptualization+ Mathematics= Fruitful Innovation*. New York, NY, USA: Springer, 2016, pp. 221–234.
- [14] H. Zhu and G. Wan, "Local contrast preserving technique for the removal of thin cloud in aerial image," *Optik-Int. J. Light Electron Opt.*, vol. 127, no. 2, pp. 742–747, 2016.
- [15] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.
- [16] C. L. Zitnick and S. B. Kang, "Stereo for image-based rendering using image over-segmentation," *Int. J. Comput. Vis.*, vol. 75, no. 1, pp. 49–65, 2007.
- [17] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum, "Lazy snapping," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 303–308, 2004.
- [18] G. Mori, "Guiding model search using segmentation," in *Proc. 10th IEEE Int. Conf. Comput. Vision*, 2005, vol. 2, pp. 1417–1423.
- [19] B. Fulkerson *et al.*, "Class segmentation and object localization with superpixel neighborhoods," in *Proc. IEEE 12th Int. Conf. Comput. Vision*, 2009, vol. 9, pp. 670–677.
- [20] E. Welch, R. Moorhead, and J. Owens, "Image processing using the HSI color space," in *Proc. IEEE Southeastcon*, 1991, pp. 722–725.
- [21] A. R. Weeks and G. E. Hague, "Color segmentation in the HSI color space using the k-means algorithm," *Proc. SPIE*, vol. 3026, pp. 143–154, 1997.
- [22] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vision*, vol. 59, no. 2, pp. 167–181, 2004.
- [23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [24] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 1265–1274.
- [25] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2014, pp. 580–587.
- [26] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Eur. Conf. Comput. Vision*, 2014, pp. 184–199.

- [27] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 807–814.
- [28] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [29] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.
- [30] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, Jul. 2002.
- [31] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
- [32] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 266–277, Feb. 2001.
- [33] L. A. Vese and T. F. Chan, "A multiphase level set framework for image segmentation using the Mumford and Shah model," *Int. J. Comput. Vision*, vol. 50, no. 3, pp. 271–293, 2002.



**Fengying Xie** received the Ph.D. degree in pattern recognition and intelligent system from Beihang University, Beijing, China, in 2009.

She was a Visiting Scholar with the Laboratory for Image and Video Engineering, University of Texas at Austin, from 2010 to 2011. She is currently a Professor with the School of Astronautics, Beihang University. Her research interests include biomedical image processing, remote sensing image understanding and application, image quality assessment, and pattern recognition.



**Mengyun Shi** received the B.Eng. degree in information management and information system from Beijing Forestry University, Beijing, China, in 2014. He is a postgraduate majored in pattern recognition and intelligent system with the School of astronautics, Beihang University, Beijing.

His research interests include remote sensing image processing, image segmentation, and deep learning.



**Zhenwei Shi** (M'13) received the Ph.D. degree in mathematics from Dalian University of Technology, Dalian, China, in 2005.

He was a Postdoctoral Researcher with the Department of Automation, Tsinghua University, Beijing, China, from 2005 to 2007. He was a Visiting Scholar with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, USA, from 2013 to 2014. He is currently a Professor and the Dean of the Image Processing Center, School of Astronautics, Beihang

University, Beijing. He has authored or coauthored more than 100 scientific papers in refereed journals and proceedings, including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON NEURAL NETWORKS, the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, and the IEEE Conference on Computer Vision and Pattern Recognition. His current research interests include remote sensing image processing and analysis, computer vision, pattern recognition, and machine learning.



**Jihao Yin** (M'10–SM'15) received the Ph.D. degree in computer science from Beijing Institute of Technology, Beijing, China, in 2007.

Meanwhile, he joined Beihang University, Beijing, where he is currently an Associate Professor with the Image Processing Center, School of Astronautics. As a Principle Investigator, his research works have been supported by several key research funds in China including the National Natural Science Fund of China, National High Technology Research and Development Program of China, Research Fund for the Doctoral Program of Higher Education of China, and Aerospace Research Fund of China. He currently directs the Space Information and Remote Sensing Laboratory, School of Astronautics, Beihang University. His research interests include image processing, computer vision, and remote sensing applications.



**Danpei Zhao** received the Ph.D. degree in optical engineering from Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Beijing, China, in 2006.

She is an Associate Professor at Beihang University, Beijing, where he is the Vice Director of the Center of Image Processing. From 2006 to 2008, she was with Beihang University for postdoctoral research. She was a Visiting Scholar with the Department of Computer Science, Rutgers, the State University of New Jersey, USA, from 2014 to 2015. Her research

interests include saliency detection and its application in remote sensing target detection, remote sensing image understanding, target detection, tracking, and recognition.

Dr. Zhao serves as a standing member of the Executive Council of the Beijing Society of Image and Graphics.