International Journal of Geo-Information

MDPI

Article

Multilevel Cloud Detection for High-Resolution Remote Sensing Imagery Using Multiple Convolutional Neural Networks

Yang Chen [1,2,*,†], Rongshuang Fan [2], Muhammad Bilal [3], Xiucheng Yang [4], Jingxue Wang [1,†] and Wei Li [5]

1 School of Geomatics, Liaoning Technical University, Fuxin 123000, China; xiaoxue1861@163.com
2 Chinese Academy of Surveying and Mapping, Beijing 100830, China; fanrsh@casm.ac.cn
3 School of Marine Sciences, Nanjing University of Information Science and Technology, Nanjing 210044, China; muhammad.bilal@connect.polyu.hk
4 ICube Laboratory, University of Strasbourg, 67000 Strasbourg, France; xiuchengyang@163.com
5 Mining Engineering Institute, Heilongjiang University of Science and Technology, Harbin 150027, China; iamink@163.com
* Correspondence: chenyang1017@126.com; Tel.: +86-153-5020-5816
† These authors contributed equally to this work.

Received: 5 April 2018; Accepted: 7 May 2018; Published: 9 May 2018

Abstract: In high-resolution image data, multilevel cloud detection is a key task for remote sensing data processing. Generally, it is difficult to obtain high accuracy for multilevel cloud detection when using satellite imagery which only contains visible and near-infrared spectral bands. So, multilevel cloud detection for high-resolution remote sensing imagery is challenging. In this paper, a new multilevel cloud detection technique is proposed based on the multiple convolutional neural networks for high-resolution remote sensing imagery. In order to avoid input the entire image into the network for cloud detection, the adaptive simple linear iterative clustering (A-SCLI) algorithm was applied to the segmentation of the satellite image to obtain good-quality superpixels. After that, a new multiple convolutional neural networks (MCNNs) architecture is designed to extract multiscale features from each superpixel, and the superpixels are marked as thin cloud, thick cloud, cloud shadow, and non-cloud. The results suggest that the proposed method can detect multilevel clouds and obtain a high accuracy for high-resolution remote sensing imagery.

Keywords: multiple convolutional neural networks; cloud detection; superpixel; high-resolution remote sensing imagery

1. Introduction

With the advancement in remote sensing technology, high-resolution satellite imagery is widely used in various fields such as resource surveying, environmental monitoring, and geographical mapping [1]. However, according to the International Satellite Cloud Climatology Project-Flux Data (ISCCP-FD), the global annual mean cloud cover is approximately 66% [2]. Cloud often appears and covers objects on the surface in high-resolution remote sensing images, which not only leads to missing information and spectral distortion, but also can affect the processing of remote sensing imagery [3]. Therefore, cloud detection in high-resolution satellite imagery is of great significance.

Up to now, a series of cloud detection methods have been proposed [4], and these methods can be divided into two categories: (i) threshold-based methods and (ii) machine learning-based methods. The threshold-based methods are practical and fast in calculation, so are widely used in practical applications. Threshold-based methods, including the International Satellite Cloud Climatology

Project (ISCC) [5], the NOAA Cloud Advanced Very High Resolution Radiometer (CLAVR) [6], and the MODIS cloud mask method [7], have established a series of direct thresholds using the apparent reflectance or brightness temperatures through single or multiple channels for different satellite imageries. Zhang et al. [8] obtained a coarse cloud detection result relying on the significance map and the proposed optimal threshold setting.

Many other methods have also used thermal infrared (TIR) bands for cloud detection, but high-resolution images have a lack of thermal infrared (TIR) bands. So, the threshold-based methods are generally difficult to apply to the high-resolution images due to the limited spectral resolution.

However, machine learning algorithms such as support vector machine (SVM) have been widely used in image classification [9]. Machine learning-based methods extract a range of manual characteristics pixel by pixel, followed by learning a binary classifier to determine whether this pixel belongs to cloud area or not [10]. Hughes et al. [11] developed a neural network approach to detect clouds in Landsat images using spectral and spatial information. Başeski and Cenaras [12] trained an SVM classifier using texture characteristics to detect cloud in RGB images. These methods are manually designed features which rely on prior knowledge and have difficulty in accurately representing the cloud features in the complex environment [13]. In high-resolution imagery, ground objects are complex and shapes of the cloud are varied, and many other objects such as ice, white buildings, snow, and so forth can cause confusion. Therefore, cloud detection in high-resolution images is a challenging task. By summarizing the existing cloud detection methods, it can be found that most of the methods are only concerned with the cloud-covered area. However, a few other methods showed interest in the identification of cloud types. Thus, multilevel cloud detection is of significance for thin cloud removal and image analysis tasks.

Deep learning is the learning process of simulating the human brain, which automatically extracts high-level features from the low-level features of the input image [14,15]. The deep convolutional neural network (CNN), which is one of the deep learning methods, has its unique advantage, especially for processing visual-related problems [16].

Generally, it is difficult to obtain good results for multilevel cloud detection when using imagery which only includes visible and near-infrared spectral bands. In this study, we propose multilevel cloud detection (thin cloud, thick cloud, and cloud shadow) from Chinese high-resolution satellite imagery based on MCNNs. In order to avoid the entire image into the network for cloud detection, first, we extend simple linear iterative clustering (SLIC) [17] to generate superpixels, and the superpixel is taken as the basic unit of cloud detection. We propose a novel MCNNs architecture consisting of three different patch-based CNN models. Each different patch-based CNN replaces fully connected layers with global self-adaptive pooling (GSAP). The results indicate that the MCNNs architecture performs well in typical land-cover types, and it can also accurately detect multilevel cloud using only the four optical bands.

The major contributions of this paper are:

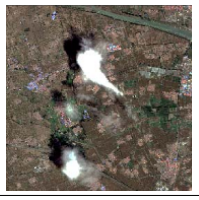
(1) In order to reduce the loss of image features during the process of pooling, we propose self-adaptive pooling (SAP).
(2) A novel MCNNs architecture is designed for multilevel cloud detection.
(3) Adaptive simple linear iterative clustering (A-SLIC) algorithm is proposed through affinity propagation clustering and expanding the searching space. The A-SLIC algorithm was applied to obtain segmentation of the image into good-quality superpixels.

2. Datasets

In this study, three categories of different spatial resolutions satellite imagery, GaoFen-1 (GF-1), GaoFen-2 (GF-2), and ZiYun-3 (ZY-3), were used for multilevel cloud detection. The information of Chinese high-resolution satellite imagery is given in Table 1. Experimental imageries contain various cloud types such as small thin cloud, medium-sized thin cloud, large thick cloud, cloud shadow, and so

forth. These satellite imageries contain many underlying surface environments such as building, sand, ice, sea, vegetation, snow, and so forth. In general, it is difficult to distinguish between the thin cloud and the thick cloud. In addition, it also very difficult to distinguish snow and white buildings from cloud pixels based on their spectral features.

Table 1. Summary of the dataset used in this study.

Image Name	Image Size (Pixels)	Spatial Resolution (m)	Cloud Types	Surface Types	True Color Multispectral Image
ZY-3	2900 × 3000	5.8	medium thin cloud; medium thick cloud; cloud shadow	water; mountain; bare rock	
ZY-3	3000 × 3000	5.8	small thin cloud; medium thick cloud; cloud shadow	building; river; city road	
GF-1	2100 × 2399	8	non-cloud	lake; mountain; bare rock; ice; snow	
GF-2	3000 × 3000	4	medium thick cloud; small thin cloud; cloud shadow	vegetation; building; road; lake	

Our training dataset is collected from ten ZY-3 multispectral images, seven Gaofeng-1 multispectral images, and nine Gaofeng-2 multispectral images. Our testing dataset is collected from two ZY-3 multispectral images, one Gaofeng-1 multispectral image, and one Gaofeng-2 multispectral image.

3. Methods

In this section, the proposed framework used for cloud detection is shown in Figure 1. First, the A-SCLI algorithm is proposed for application to the segmentation of the remote sensing image, which can obtain adjacent SLIC regions. The A-SCLI algorithm is used to enhance CNN outputs. Second, the MCNNs architecture and learning framework are discussed.

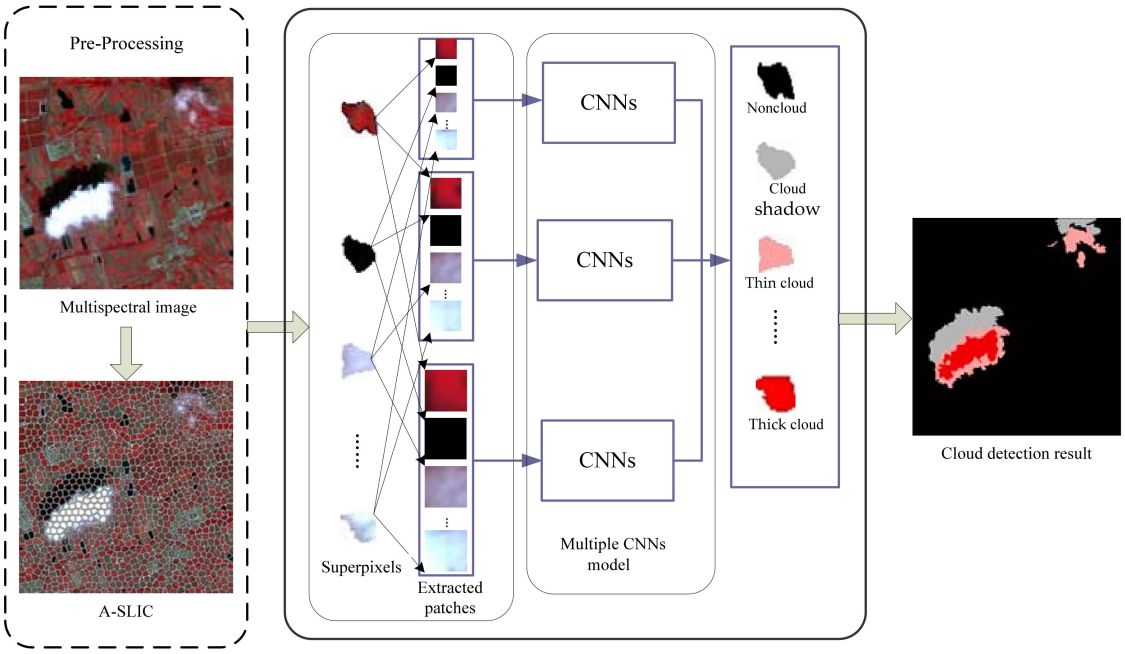


Figure 1. Proposed cloud detection framework. A-SLIC: adaptive simple linear iterative clustering; MCNNs: multiple convolutional neural networks.

3.1. Preprocessing

For cloud detection in high-resolution remote sensing images, many cloud detection methods are based on the pixel level [18], and a few cloud detection methods are based on the superpixel level [19]. In this study, the MCNNs model is adopted to extract multiscale features from high-resolution satellite imagery. Therefore, if pixels have been used as the basic unit of cloud detection, then the efficiency of the cloud detection method is very low. The term superpixel refers to the adjacent image blocks with similar color and brightness characteristics [20,21]. It groups the pixels based on the similarities of features and obtains the redundant information of the image, which greatly reduces the complexity of subsequent image processing tasks.

As a widely used superpixel algorithm [17], the SLIC algorithm can output good-quality superpixels which are compact and roughly equally sized, but some problems still exist such as the fact that the number of separations should be designed artificially and the ultrapixel edges are divided vaguely. As SLIC obtains initial cluster centers through dividing the image into several equally sized grids and the search space is limited to a local region, the produced superpixels cannot adhere to weak cloud boundaries well and the cloud will be oversegmented [22]. In this paper, the SLIC algorithm has improved from affinity propagation clustering and expanding the searching space.

Generally, the color of the cloud is white, with low reflectivity and high saturation. Similarly to the RGB color model, the color space transformation to the hue, saturation, and intensity (HSI) color model is first performed [23]. The transformation from RGB to the HSI color model is expressed as follows:

$$H = \begin{cases} \theta & B \le G \\ 360 - \theta & B > G \end{cases} \tag{1}$$

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

$$I = \frac{R + G + B}{3} \tag{3}$$

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

where R, G, and B are the values of band one, band two, and band three channels of input for the remote sensing image. H, S, and I are the values of hue, saturation, and intensity components in the HSI space.

Figure 2 shows an example of the HSI color space. Figure 2a is an original RGB color image. Figure 2b–d shows an intensity component image, hue component image, and saturation component image, respectively. It can be seen that the cloud region is prominent in I and S components. So, I and S components are used in our proposed SLIC method.

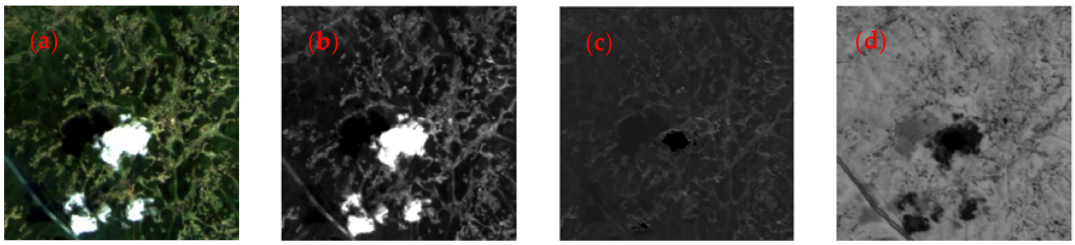


Figure 2. (**a**) Original RGB color image; (**b**) Intensity component image; (**c**) Hue component image; (**d**) Saturation component image.

Generally, a weighted similarity measure combining color and spatial proximity is needed in the simple linear iterative clustering algorithm [24]. In this paper, the similarity measure between the ith pixel and jth cluster center c_j is expressed as follows:

$$d(i,j) = d_c + \frac{\alpha}{S} d_{xy} \tag{5}$$

$$d_c = \sqrt{(I_i - I_{C_j})^2 + (S_i - S_{C_j})^2} \tag{6}$$

$$d_{xy} = \sqrt{(x_i - x_{c_j})^2 + (y_i - y_{c_j})^2} \tag{7}$$

where d_c is ith pixel and jth pixel color difference d_c, d_{xy} is ith pixel and jth pixel space distance, and S is the area of the jth cluster in the current loop. The α parameter is used to control the relative importance of color similarity and spatial proximity.

The attraction function reflects the possibility of the jth pixel attracting the ith pixel as its cluster [25]. The attraction function is expressed as:

$$\alpha(i,j) = s(i,j) - \max_{j' \neq j} \{\beta(i,j') + s(i,j')\} \tag{8}$$

where $s(i,j) = -d(i,j)$ is the similarity between the ith pixel and the jth pixel and $s(i,j') = -d(i,j')$ is the similarity between the ith pixel and the non-jth pixel.

The iterative relationship of the attraction function is expressed as:

$$\alpha^t(i,j) = s(i,j) - \max_{j' \neq j} \left\{ \beta^{t-1}(i,j') + s(i,j') \right\} \tag{9}$$

where t is the number of iterations.

The attribution function reflects the possibility that the ith pixel attracts the jth pixel as its cluster [26]. The attribution function is expressed as:

$$\beta(i,j) = \begin{cases} \min\limits_{i \neq j} \left\{ 0, \alpha(j,j) + \sum\limits_{i' \neq i,j} \max[0, \alpha(i',j)] \right\} & i \neq j \\ \sum\limits_{i' \neq j} \max[0, \alpha(i',j)] & i = j \end{cases} \tag{10}$$

The iterative relationship of the attribution function is expressed as:

$$\beta^t(i,j) = \begin{cases} \min\limits_{i \neq j} \left\{ 0, \alpha^{t-1}(j,j) + \sum\limits_{i' \neq i,j} \max[0, \alpha^{t-1}(i',j)] \right\} & i \neq j \\ \sum\limits_{i' \neq j} \max[0, \alpha^{t-1}(i',j)] & i = j \end{cases} \tag{11}$$

where t is the number of iterations.

Using both attraction and attribution functions, two types of messages are continuously transmitted to possible clustering centers to increase their likelihood of becoming cluster centers. So, the larger the sum of $\alpha(i,j)$ and $\beta(i,j)$, the more likely the jth pixel is a cluster center. In this case, the greater the probability that the ith pixel belongs to this class, the more likely that the point is updated as a new cluster center. In order to reduce the computation complexity, this paper firstly divided the segmentation images, and $\alpha(i,j)$ and $\beta(i,j)$ were calculated in the local area. In this study, the main processes of the A-SLIC algorithm are as follows:

- **Step 1.** For an image containing M pixels, the size of the predivided region in this algorithm is N, and the number of regions is n. Each predivided area is labeled as η. $\alpha(i,j)$ and $\beta(i,j)$ is defined as zero, and t is defined as one.
- **Step 2.** HIS transformation is performed on the image of the marked area. In the ηth region, according to Equation (5), the similarity between two pixels is calculated in turn.
- **Step 3.** According to Equations (9) and (11), the sum of $\beta^t(i,j)$ and $\alpha^t(i,j)$ is calculated and the iteration begins.
- **Step 4.** If $\beta^t(i,j)$ and $\alpha^t(i,j)$ no longer change or reach the maximum number of iterations, the iteration is terminated. The point where the sum of $\beta^t(i,j)$ and $\alpha^t(i,j)$ is maximum is regarded as the cluster center R_i^η.
- **Step 5.** Repeat steps 3 to 4 until the entire image is traversed, and adaptively determine the number of superpixels ($R' = \sum\limits_{\eta=1}^{n} W_\eta$). Finally, complete the superpixel segmentation.

The results of A-SLIC and SLIC segmentation are shown in Figure 3c,d. The experimental results indicate that the superpixels segmented by the A-SLIC algorithm were compact and regular shape and adhered well to the cloud region boundaries.

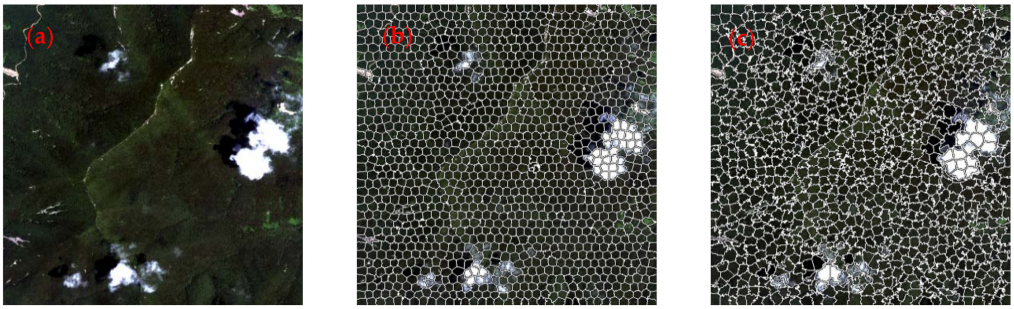


Figure 3. Segmentation results. (**a**) Original image; (**b**) A-SLIC segmentation results; (**c**) SLIC segmentation results.

3.2. Proposed Convolutional Neural Network Architecture

CNN architecture has become a hot topic in the field of deep learning [27]. The CNN architecture provides a new method for remote sensing image high-level feature extraction. Generally, the CNN architecture consists of the input layer, convolution layer, pooling layer, full connection layer, and output layer, as shown in Figure 4.

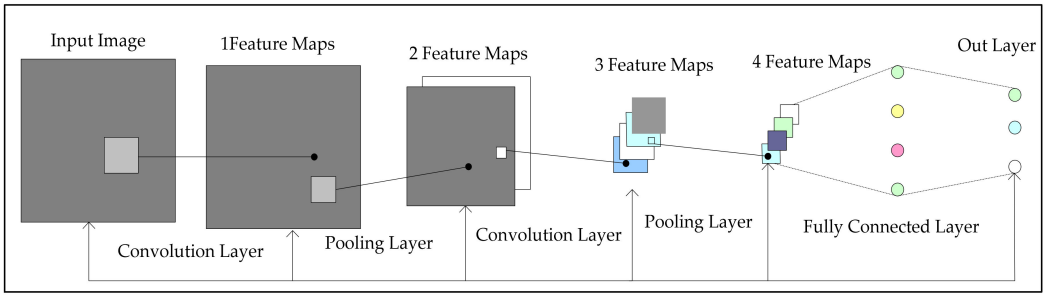


Figure 4. The standard architecture of the CNN.

The input image is convoluted in the convolutional and filtering layers. Generally, convolutional and filtering layers require an activation function to connect [28]. We use $\mathbf{G}_i$ to represent the feature map of the ith layer of the convolutional neural network. The convolution process can be described as:

$$\mathbf{G}_i = f(\mathbf{G}_{i-1} \otimes \mathbf{W}_i + \mathbf{b}_i) \tag{12}$$

where $\mathbf{W}_i$ represents the weight feature vector of the ith convolution kernel, the operation symbol $\otimes$ represents a convolution operation of the ith layer of the image and the $i-1$th layer of the image, and $\mathbf{b}_i$ is the offset vector. Finally, the feature map $\mathbf{G}_i$ of the ith layer is obtained by a linear activation function $f(\bullet)$.

There are two kinds of activation functions: one is a linear activation function and the other is a nonlinear activation function. There are three common nonlinear activation functions: hyperbolic function, sigmoid, and soft plus [29]. The hyperbolic function is a variant of the sigmoid function. The range of the hyperbolic function is $[-1, 1]$, and the range of the sigmoid function is $[0, 1]$. The activation state of the linear correction function and biological neurons after stimulation is relatively close. The linear correction function is commonly used as the activation function of convolution neural networks because of its sparsity and simple calculation [30]. In this paper, we use $f(x) = \max(0, x)$ as an activation function.

After the convolution layer is the pooling layer, and the convolution layer and the pooling layer are linked by an activation function. There are two main models of pooling layer: one is the max pooling model as shown in Equation (13) and the other is an average pooling model as shown in Equation (14).

The feature map obtained by the convolution layer is $\mathbf{G}_{ij}$, the size of the pooling area is $c \times c$, the pooling step length is c, and $\mathbf{b}_i$ is the offset. The max pooling model can be expressed as:

$$\mathbf{F}_{ij} = \max_{i=1,j=1}^{c}(\mathbf{G}_{ij}) + \mathbf{b}_i \tag{13}$$

The average pooling model can be expressed as:

$$\mathbf{F}_{ij} = \frac{1}{c^2}(\sum_{i=1}^{c}\sum_{j=1}^{c}\mathbf{G}_{ij}) + \mathbf{b}_i \tag{14}$$

where $\max_{i=1,j=1}^{c}(\mathbf{G}_{ij})$ represents the max element from the feature map $\mathbf{G}$ in the pooled region of size $c \times c$.

Due to the complexity of the objects in high-resolution images, the traditional pooling model cannot extract the image features very well. Therefore, this research takes two kinds of pooling areas in the pooling layer, as shown in Figure 5. The blank space indicates that the pixel value is 0 the shaded area is composed of different pixel values, and a represents the maximum value area. The features of the whole feature map are mainly concentrated at A as shown in Figure 5a. If pooling is done with the average pooling model, the features of the entire feature map will be weakened. The features of the feature map are mainly distributed in A, B, and C, as shown in Figure 5b. In the case of the unknown relationship between A, B, and C, the features of the entire feature map will be weakened by using the maximum pooling model. This will eventually affect the detection accuracy of the cloud in remote sensing images.

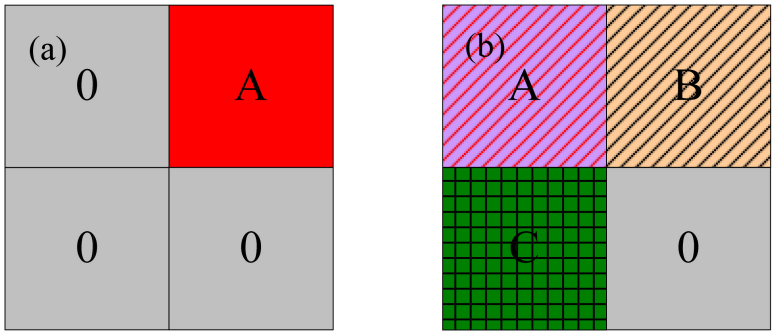


Figure 5. Different pooling areas. (**a**) one Feature mapping; (**b**) other feature mapping.

In order to reduce the loss of image features during the process of pooling, this paper presents a SAP according to the principle of interpolation, based on the maximum pool model and the average model. The model can adaptively adjust the pooling process through the pooling factors u in the complex pooled area. The expression is:

$$\mathbf{F}_{ij} = \frac{u}{c^2}\left(\sum_{i=1}^{c}\sum_{j=1}^{c}\mathbf{G}_{ij}\right) + (1-u)\max_{i=1,j=1}^{c}\left(\mathbf{G}_{ij}\right) + \mathbf{b}_i \tag{15}$$

where u indicates the pooling factor. The role of u is to dynamically optimize the traditional pooling model based on different pooled areas. The expression is:

$$u = \frac{a(b_{\max} - a)}{b_{\max}^2} \tag{16}$$

where a is the average of all elements, except for the max element in the pooled area, and $b_{\max}$ is the max element in the pooled area. The range of u is [0, 1]. The model takes into account advantages of both the max pooling model and the average model. According to the characteristics of different pooling regions, the adaptive optimization model can be used to extract the features of the map as much as possible, so as to improve the removal accuracy of the convolution neural network.

Figure 6b–d shows three feature maps from the different pooling models. It can be seen that the feature map obtained from the adaptive pooling model has obvious features, while the max pooling model and the average pooling model weaken the thin cloud and cloud shadow features.

Figure 7 shows the proposed CNN architecture. In this study, we designed a multiple CNN model, which use three different-sized patches (128 × 128, 64 × 64, and 32 × 32) as the input data to extract features from remote sensing imagery. The output is a 1024-dimensional vector, which is reshaped into four 16 × 16 channels (thin cloud, thick cloud, cloud shadow, and background).

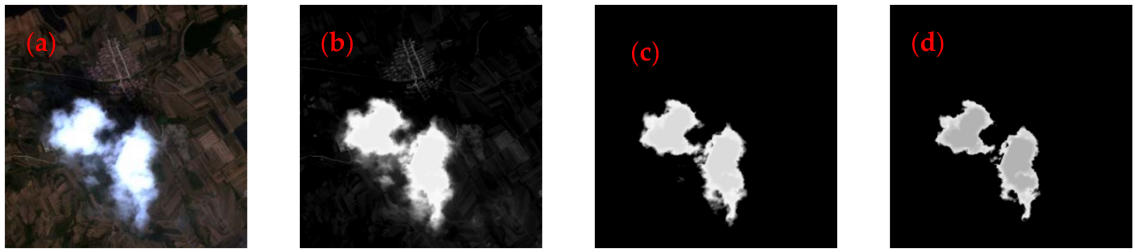


Figure 6. The feature map from the different pooling models. (**a**) Original image; (**b**) the feature map obtained from the self-adaptive pooling (SAP) model; (**c**) the feature map obtained from the average pooling model; (**d**) the feature map obtained from the max pooling model.

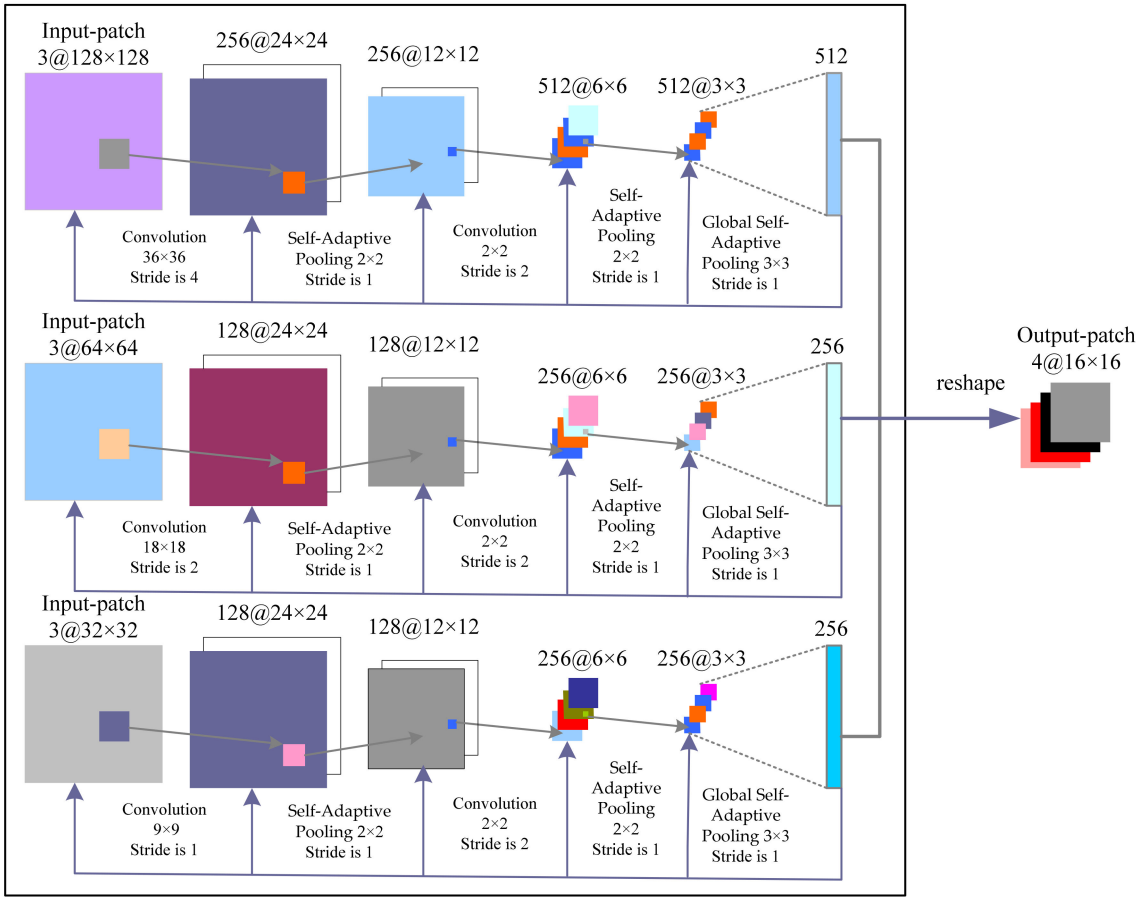


Figure 7. The architecture of our designed multiple CNN.

We propose a MCNNs architecture consisting of three different patch-based CNN models. A fully connected layer causes overfitting because of parameters [31,32]. GSAP simply self-adapts the feature maps where similar results are expected in a patch. So, each different patch-based CNN works by replacing fully connected layers with GSAP. Each different patch-based CNN contains two convolution layers, two self-adaptive pooling, and one global self-adaptive pooling.

3.3. Accuracy Assessment Method

The ground truths of multilevel cloud areas were manually extracted. We evaluate the algorithm performance for multilevel cloud detection. So, five metrics are used, including the overall accuracy (OA), the kappa, the edge overall accuracy (EOA), the edge omission error (EOE), and edge commission error (ECE). This paper designed the evaluation algorithm as follows: (i) firstly, obtain the boundary of

the cloud by artificial visual interpretation; (ii) the morphological expansion is performed in the cloud boundary obtained in step 1 to create a buffer zone centered on the boundary line and having a radius of four pixels; and (iii) finally, the pixels in the buffer area are judged. Suppose that the total number of pixels in the buffer area is N, the number of correctly classified cloud pixels is N_R, the number of missing pixels is N_O, and the number of false alarm pixels is N_c. Then, EOA, EOE, and ECE are defined as:

$$EOA = \frac{N_R}{N} \times 100\%,\ EOE = \frac{N_O}{N} \times 100\%,\ \ ECE = \frac{N_C}{N} \times 100\% \tag{17}$$

The OA and kappa are defined as [33,34]:

$$OA = \frac{TN + TP}{T} \times 100\% \tag{18}$$

$$Kappa = \frac{T \times (TN + TP) - [(TP + FP) \times (TP + FN) + (FN + TN) \times (FN + TN)]}{T \times T - [(TP + FP) \times (TP + FN) + (FN + TN) \times (FN + TN)]} \tag{19}$$

where, T is the total number of pixels in the experimental remote sensing image and TP, FN, FP, and TN are the pixels categorized by comparing the extracted cloud pixels with the ground truth reference:

- TP: true positives, i.e., the number of correct extractions;
- FN: false negatives, i.e., the number of cloud pixels not detected;
- FP: false positives, i.e., the number of incorrect extractions;
- TN: true negatives, i.e., the number of non-cloud pixels that were correctly rejected.

4. Experiments and Discussion

The proposed algorithm is implemented by using Python on a PC with CPU Intel(R) Xeon(R) E5-2630 and GPU NVidia Tesla M40 12 G memory, and the designed SAPCNN is implemented through the software library Tensor flow. The multiclass training dataset of 30,000 couples of patches is obtained from the training set, where the number of thin cloud, thick cloud, and cloud shadow events and patches are 9000, 6000, 9000, and 6000, respectively. For testing a remote sensing image, first, superpixels are obtained by the adaptive simple linear iterative clustering algorithm. Then, three different-sized patches (128 × 128, 64 × 64, and 32 × 32) centered at its geometric center pixel are extracted from each superpixel and inputted into the trained multiple CNN model to predict the class of this superpixel. Finally, the multilevel cloud detection resulting from the testing of the remote sensing image is achieved by using the predictions of all its superpixels. In this paper, we initialized the weights in each layer with a random number drawn from a zero-mean Gaussian distribution with standard deviation of 0.01. The learning rate started from 0.005 and was divided by 10 when the error reached a plateau and with initial bias set to the constant of 0.1.

4.1. Impact of the Superpixel Segmentation on the Performance of Multilevel Cloud Detection

In order to verify the effectiveness of the A-SLIC method, we compared the cloud detection accuracy using A-SLIC + MCNNs, SLIC + MCNNs, and Pixel + MCNNs.

Figure 8b,c shows some superpixel segmentation results using different superpixel segmentation methods. Visual inspection of Figure 8a,b indicated that our improved SLIC method and SLIC can obtain compact superpixels, but our improved method can obtain more regular superpixels than the SLIC method. Our A-SLIC method can not only void oversegmentation in large homogeneous regions, but can also obtain regular superpixels.

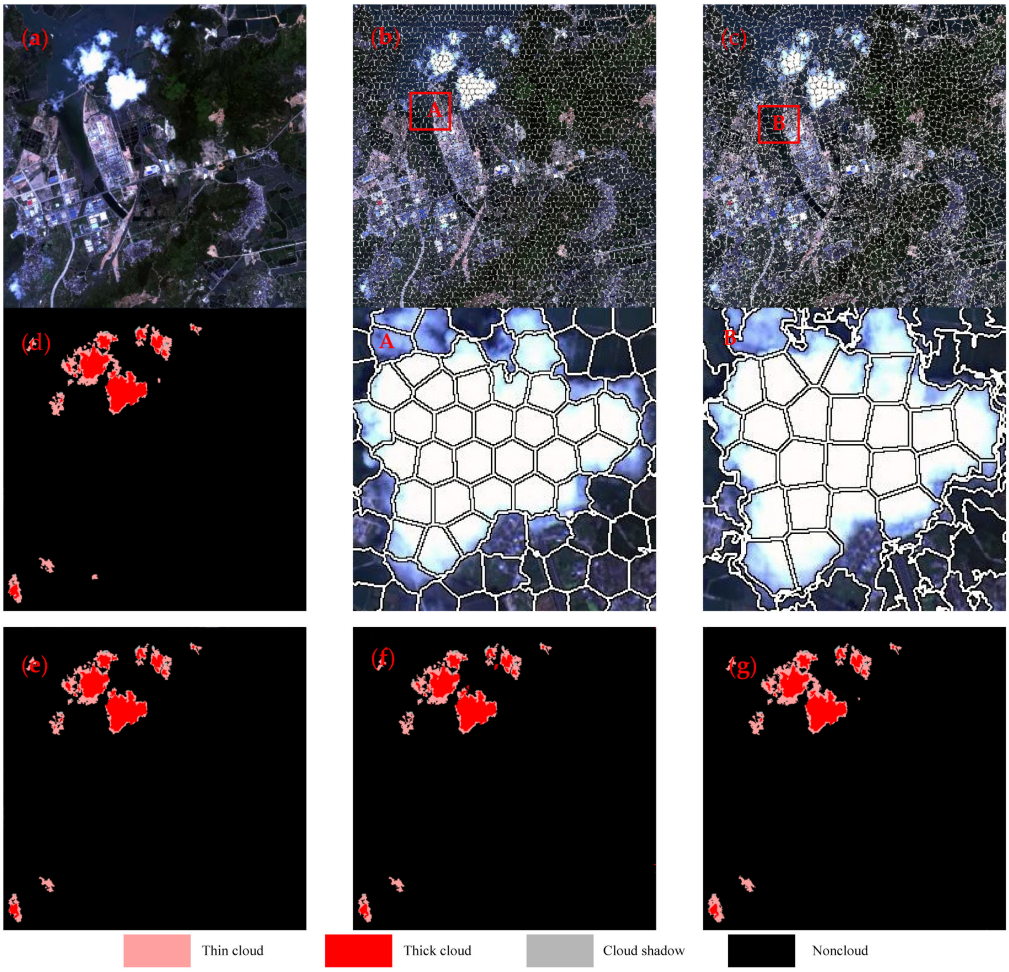


Figure 8. Multilevel cloud detection results using different superpixel segmentation methods: (**a**–**c**) Original image, A-SLIC segmentation map, and SLIC segmentation map, respectively; A and B are the magnified area corresponding to the red line regions of the segmentation map. (**d**–**g**) Ground truth reference, A-SLIC + MCNNs, SLIC + MCNNs, and Pixel + MCNNs, respectively.

From Figure 8b,c, it is obvious that all methods can extract most of the clouds; but for the blurry cloud boundaries and thin cloud regions, our improved superpixel method can achieve more accurate results because of our method through leading affinity propagation clustering and expanding the searching space, and the produced superpixels are easier to adhere to blurry cloud boundaries.

Eight metrics (OA, kappa, EOA, EOE, ECE, superpixel segmentation times, MCNNs prediction times, and total time) are used to evaluate the performance of entire cloud detection using different superpixel segmentation methods. Table 2 shows the statistical results.

Table 2. Statistics of different superpixel algorithms.

Parameter	A-SLIC + MCNNs	SLIC + MCNNs	Pixel + MCNNs
OA (%)	98.27	94.34	92.14
Kappa (%)	92.34	88.31	87.31
EOA (%)	97.36	93.13	90.38
EOE (%)	0.94	2.61	4.24
ECE (%)	1.70	4.26	5.38
Superpixel segmentation (s)	6.81	5.63	0
MCNNs prediction (s)	2.37	5.91	481
Total time (s)	9.18	11.54	481

From Table 2, it can be seen that the proposed method produced in this paper yields the best OA and EOA, and its EOE was lower than that of the other methods. Because the superpixel segmentation method introduced the idea of affinity propagation, adaptively determining the number of superpixels and the center of clustering, the superpixel can contain the cloud boundary well. From Table 1, our cloud detection architecture has the fastest speed, with 9.18 s per remote sensing image. So, superpixel preprocessing can effectively improve the cloud detection accuracy and efficiency.

4.2. Comparison Between Different CNN Architectures

In this paper, the MCNNs is designed to extract cloud and mine multi-scale features of the cloud. We compared the cloud detection accuracy using our proposed approach, and SAP + MCNNs, max pooling (MP) + MCNNs, and average pooling (AP) + MCNNs approaches

Figure 9 shows the multilevel detection results of images containing different underlying surfaces with different methods. From Figure 9c–f, it can be seen that the multilevel cloud detection can be achieved by the proposed MCNNs algorithm. However, the traditional pooling (max pooling and average pooling) + MCNNs method mistakenly highlighted some thin cloud as non-cloud (in the white box). In addition, the proposed method has shown better performance than SAP + MCNNs for multilevel cloud detection because MCNNs are integrated with A-SLIC segmentation in the preprocessing stage to improve the performance of the MCNNs.

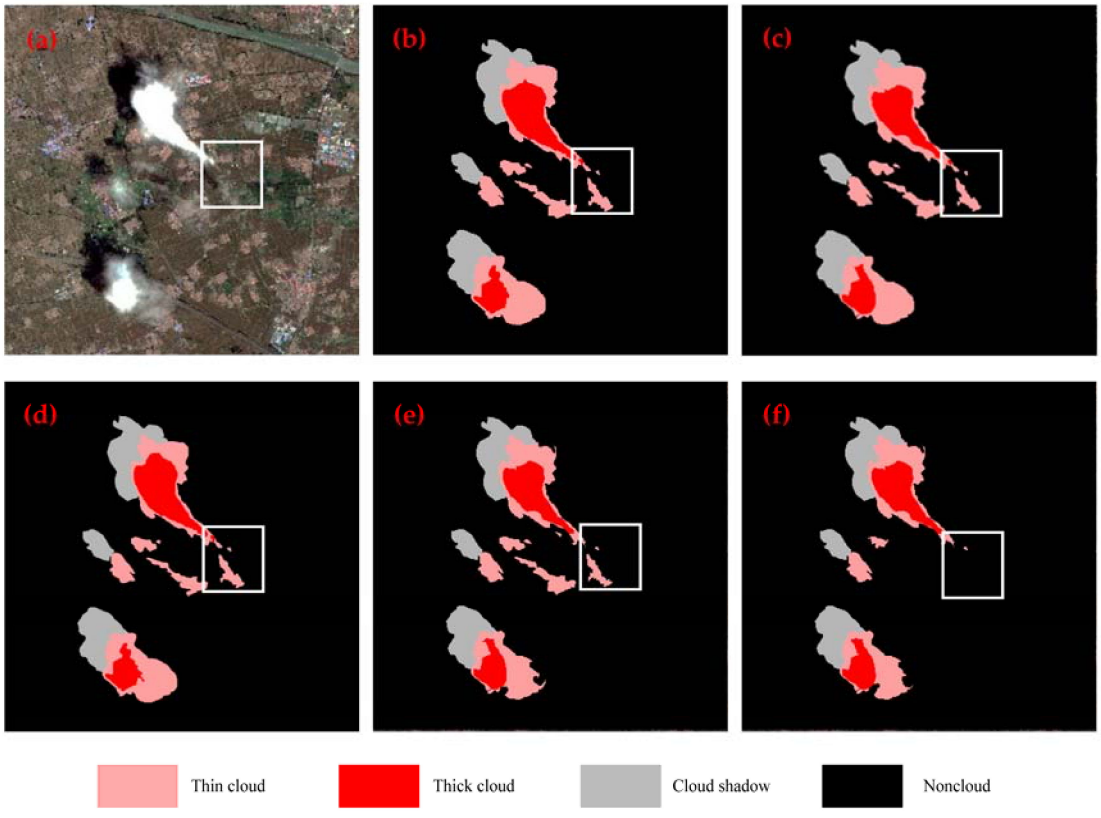


Figure 9. Multilevel cloud detection results using different CNN architectures: (**a**) original image; (**b**) ground-truth image; (**c–f**) our proposed approach, SAP + MCNNs, MP + MCNNs, and AP + MCNNs, respectively.

Five metrics (OA, kappa, EOA, EOE, and ECE) are used to evaluate the performance of entire cloud detection using different CNN architectures. Table 3 shows the statistical results.

The overall accuracy and kappa of the proposed approach was more than 95% (Table 3), and the overall edge accuracy was more than 97.37%, indicating that self-adaptive pooling and superpixel

combinations are effective in multilevel cloud detection. The overall accuracy of AP + MCNNs was the lowest, which indicates that more thin cloud regions were misjudged as non-cloud regions than the others. Our results demonstrate that SAP + MCNNs is more effective at extracting cloud features compared with two traditional pooling MCNNs and detects thin and thick cloud effectively on different underlying surfaces.

Table 3. Statistics of different CNN architectures.

Parameter	Proposed Approach	SAP + MCNNs	MP + MCNNs	AP + MCNNs
OA (%)	98.64	96.17	89.07	84.13
Kappa (%)	95.27	88.34	89.34	87.81
EOA (%)	97.37	94.01	87.34	82.41
EOE (%)	1.02	2.28	4.81	8.17
ECE (%)	1.61	3.71	7.85	9.42

4.3. Comparison with Other Methods

In order to verify the effectiveness of the proposed method, we compared the proposed cloud detection architecture with SVM [12], neural network [11], and K-means [32] approaches.

Non-cloud and cloud images are considered in this experiment, as illustrated in Table 1. We display only two groups of cloud detection results with different methods. Figure 10 shows some example cloud detection results using different methods. The first row in Figure 10a is a variety of tested cloud images covering various underlying surface environments, in which there is a bright background, mountain, bare rock, thin cloud, thick cloud, and/or cloud shadow. The second row in Figure 10a is a non-cloud image, in which there is a bright background, mountain, snow, and ice. The comparative results are shown in Figure 10c–f; it can be seen that our method in this paper produces the best results, and especially, can distinguish the thin cloud from the thick cloud. As can be seen from the second row in Figure 10d–f, some ice was identified as cloud by the SVM, neural network, and K-means.

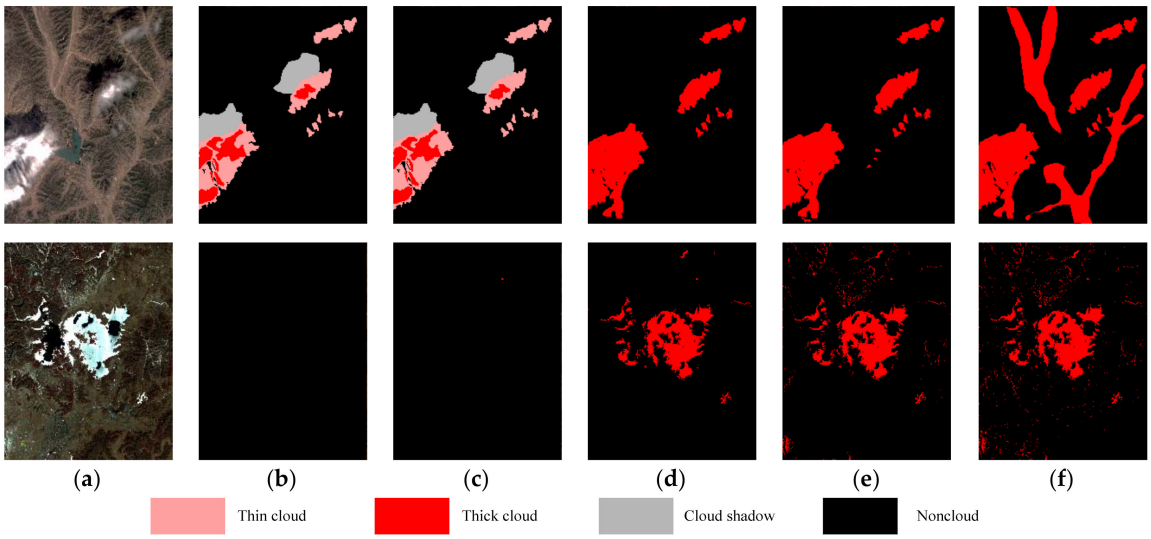


Figure 10. Cloud detection results using different methods. (**a**) Original image; (**b**) ground-truth image; (**c**) our proposed method; (**d**) SVM; (**e**) neural network; (**f**) K-means.

Five metrics (OA, kappa, EOA, EOE, and ECE) are used to evaluate the performance of entire cloud detection using different cloud detection methods. A good cloud detection algorithm has high values of OA and EOA and low values of EOE and ECE. Table 4 presents the average values of four

metrics for the five test images. The metric precision is not given here because the compared methods cannot be separated from thin cloud, thick cloud, and cloud shadow.

Table 4. Statistics of different cloud detection algorithms.

Parameter	Proposed Approach	SVM	Neural Network	K-Means
OA (%)	98.53	81.34	78.07	65.27
Kappa (%)	94.37	78.34	70.34	60.74
EOA (%)	96.17	79.51	76.39	62.37
EOE (%)	1.14	8.12	10.39	16.18
ECE (%)	2.69	12.37	13.22	21.45

From Table 3, it can be seen that the proposed method has high values of OA and kappa and low values of EOE and ECE. The compared methods misjudged the bright background (snow, ice, bare rock, and so on) for cloud pixels, and were also weak in detecting the thin cloud pixels. So, their average overall accuracies were lower than that of the proposed method. The results show that the proposed method has good accuracy and can achieve the multilevel detection of cloud.

5. Conclusions

Generally, it is difficult to obtain good results for multilevel cloud detection when using high-resolution remote sensing imagery which only includes visible and near-infrared spectral bands. This paper presents a cloud detection for high-resolution remote sensing imagery using and improved convolutional neural network model. The advantages of the proposed CNN model is that it can automatically extract multi-scale features. It is based on patch-based MCNNs, which consists of three different patch-based CNN models; each different patch-based CNN contains two convolution layers, two self-adaptive pooling, and one global self-adaptive pooling.

In our cloud detection architecture, the SLIC method was improved through affinity propagation clustering and expanding the searching space. The A-SLIC method was applied to segment the image into adjacent superpixels, which were used to enhance CNN outputs. The experiments proved that the proposed method can achieve multilevel cloud detection and obtained the best cloud detection accuracy compared to other methods. In a future study, our research will consider automatic training sample selection methods, design powerful MCNNs, and apply multisource remote sensing images for multilevel cloud detection.

Author Contributions: Y.C. is responsible for the research design, experiments and analysis, and drafting of the manuscript. J.W. made valuable suggestions to improve the quality of the paper. X.Y. and R.F. designed the model. W.L. collected the dataset. M.B. reviewed the paper. All authors reviewed the manuscript.

Acknowledgments: This work was supported by the National Key R&D Program of China (No. 2016YFC0803100), the National Natural Science Foundation of China (No. 41101452), and the Doctoral Program Foundation of Institutions of Higher Education of China (No. 20112121120003).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zortea, M.; De Martino, M.; Serpico, S. A SVM Ensemble Approach for Spectral-Contextual Classification of Optical High Spatial Resolution Imagery. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Barcelona, Spain, 23–28 July 2007; pp. 1489–1492.
2. Zhang, Y.; Rossow, W.B.; Lacis, A.A. Calculation of radiative fluxes from the surface to top of atmosphere based on ISCCP and other global data sets. *J. Geophys. Res.* **2004**, *109*, 1121–1125. [CrossRef]
3. Xu, X.; Guo, Y.; Wang, Z. Cloud image detection based on Markov Random Field. *Chin. J. Electron.* **2012**, *29*, 262–270. [CrossRef]
4. Qing, Z.; Chunxia, X. Cloud detection of rgb color aerial photographs by progressive refinement scheme. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 7264–7275. [CrossRef]

5. Lee, K.-Y.; Lin, C.-H. Cloud detection of optical satellite images using support vector machine. In Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Prague, Czech Republic, 12–19 July 2016; pp. 289–293.
6. Marais, I.V.; Du Preez, J.A.; Steyn, W.H. An optimal image transform for threshold-based cloud detection using heteroscedastic discriminant analysis. *Int. J. Remote Sens.* **2011**, *32*, 1713–1729. [CrossRef]
7. Li, Q.; Lu, W.; Yang, J.; Wang, J.Z. Thin cloud detection of all-sky images using markov random fields. *IEEE Geosci. Remote Sens. Lett.* **2012**, *9*, 417–421. [CrossRef]
8. Shao, Z.; Hou, J.; Jiang, M.; Zhou, X. Cloud detection in landsat imagery for antarctic region using multispectral thresholds. *SPIE Asia-Pac. Remote Sens. Int. Soc. Opt. Photonics* **2014**. [CrossRef]
9. Wu, W.; Luo, J.; Hu, X.; Yang, H.; Yang, Y. A Thin-Cloud Mask Method for Remote Sensing Images Based on Sparse Dark Pixel Region Detection. *Remote Sens.* **2018**, *10*, 617. [CrossRef]
10. Bai, T.; Li, D.R.; Sun, K.M.; Chen, Y.P.; Li, W.Z. Cloud detection for high-resolution satellite imagery using machine learning and multi-feature fusion. *Remote Sens.* **2016**, *8*, 715. [CrossRef]
11. Wang, H.; He, Y.; Guan, H. Application support vector machines in cloud detection using EOS/MODIS. In Proceedings of the Remote Sensing Applications for Aviation Weather Hazard Detection and Decision Support, San Diego, CA, USA, 25 August 2008.
12. Base ski, E.; Cenaras, C. Texture color based cloud detection. In Proceedings of the 2015 7th International Conference on Recent Advances in Space Technologies (RAST), Istanbul, Turkey, 16–19 June 2015.
13. Alireza, T.; Fabio, D.F.; Cristina, C.; Stefania, V. Neural networks and support vector machine algorithms for automatic cloud classification of whole-sky ground-based images. *IEEE Trans. Geosci. Remote Sens.* **2015**, *12*, 666–670.
14. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *37*, 1904–1916. [CrossRef] [PubMed]
15. Yang, L.; MacEachren, A.M.; Mitra, P.; Onorati, T. Visually-Enabled Active Deep Learning for (Geo) Text and Image Classification: A Review. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 65. [CrossRef]
16. Sherrah, J. Fully Convolutional Networks for Dense Semantic Labelling of High-Resolution Aerial Imagery. *arXiv*, **2016**. [CrossRef]
17. Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; Süsstrunk, S. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2274–2282. [CrossRef] [PubMed]
18. Csillik, O. Fast Segmentation and Classification of Very High Resolution Remote Sensing Data Using SLIC Superpixels. *Remote Sens.* **2017**, *9*, 243. [CrossRef]
19. Huang, X.; Zhang, L. An SVM ensemble approach combining spectral, structural, and semantic features for the classification of high-resolution remotely sensed imagery. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 257–272. [CrossRef]
20. Guangyun, Z.; Xiuping, J.; Jiankun, H. Superpixel-based graphical model for remote sensing image mapping. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 5861–5871.
21. Li, H.; Shi, Y.; Zhang, B.; Wang, Y. Superpixel-Based Feature for Aerial Image Scene Recognition. *Sensors* **2018**, *18*, 156. [CrossRef] [PubMed]
22. Hagos, Y.B.; Minh, V.H.; Khawaldeh, S.; Pervaiz, U.; Aleef, T.A. Fast PET Scan Tumor Segmentation Using Superpixels, Principal Component Analysis and K-Means Clustering. *Methods Protoc.* **2018**, *1*, 7. [CrossRef]
23. Zollhöfer, M.; Izadi, S.; Rehmann, C.; Zach, C.; Fisher, M.; Wu, C.; Fitzgibbon, A.; Loop, C.; Theobalt, C.; Stamminger, M. Real-time non-rigid reconstruction using an RGB-D camera. *ACM Trans. Graph.* **2014**, *33*, 156. [CrossRef]
24. Fouad, S.; Randell, D.; Galton, A.; Mehanna, H.; Landini, G. Epithelium and Stroma Identification in Histopathological Images Using Unsupervised and Semi-Supervised Superpixel-Based Segmentation. *J. Imaging* **2017**, *3*, 61. [CrossRef]
25. Yang, J.; Yang, G. Modified Convolutional Neural Network Based on Dropout and the Stochastic Gradient Descent Optimizer. *Algorithms* **2018**, *11*, 28. [CrossRef]
26. Chen, F.; Ren, R.; Van de Voorde, T.; Xu, W.; Zhou, G.; Zhou, Y. Fast Automatic Airport Detection in Remote Sensing Images Using Convolutional Neural Networks. *Remote Sens.* **2018**, *10*, 443. [CrossRef]
27. Pouliot, D.; Latifovic, R.; Pasher, J.; Duffe, J. Landsat Super-Resolution Enhancement Using Convolution Neural Networks and Sentinel-2 for Training. *Remote Sens.* **2018**, *10*, 394. [CrossRef]

28. Scarpa, G.; Gargiulo, M.; Mazza, A.; Gaetano, R. A CNN-Based Fusion Method for Feature Extraction from Sentinel Data. *Remote Sens.* **2018**, *10*, 236. [CrossRef]
29. Cai, Z.; Fan, Q.; Feris, R.; Vasconcelos, N. A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection. In Proceedings of the IEEE European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 354–370.
30. Hu, F.; Xia, G.S.; Hu, J.; Zhang, L. Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery. *Remote Sens.* **2015**, *7*, 14680–14707. [CrossRef]
31. Chen, Y.; Fan, R.; Yang, X.; Wang, J.; Latif, A. Extraction of Urban Water Bodies from High-Resolution Remote-Sensing Imagery Using Deep Learning. *Water* **2018**, *10*, 585. [CrossRef]
32. Weatherill, G.; Burton, P.W. Delineation of shallow seismic source zones using K-means cluster analysis, with application to the Aegean region. *Geophys. J. Int.* **2009**, *176*, 565–588. [CrossRef]
33. Pontius, R.G., Jr.; Millones, M. Death to Kappa: Birth of quantity disagreement and allocation disagreement for accuracy assessment. *Int. J. Remote Sens.* **2011**, *32*, 4407–4429. [CrossRef]
34. Stein, A.; Aryal, J.; Gort, G. Use of the Bradley-Terry model to quantify association in remotely sensed images. IEEE Trans. *Geosci. Remote Sens.* **2005**, *43*, 852–856. [CrossRef]