Remote Sensing of Environment 250 (2020) 112045

Contents lists available at ScienceDirect

Remote Sensing of Environment

journal homepage: www.elsevier.com/locate/rse

Accurate cloud detection in high-resolution remote sensing imagery by weakly supervised deep learning

Yansheng Li[a], Wei Chen[a], Yongjun Zhang[a,*], Chao Tao[b], Rui Xiao[a], Yihua Tan[c,*]

[a] School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China
[b] School of Geosciences and Info-Physics, Central South University, Changsha 410083, China
[c] School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China

ARTICLE INFO

Keywords:
Cloud detection
Weakly supervised deep learning
Global convolutional pooling (GCP)
Local pooling pruning (LPP)
High-resolution remote sensing (RS) imagery

ABSTRACT

Cloud cover is a common and inevitable phenomenon that often hinders the usability of optical remote sensing (RS) image data and further interferes with continuous cartography based on RS image interpretation. In the literature, the off-the-shelf cloud detection methods either require various hand-crafted features or utilize data-driven features using deep networks. Overall, deep networks achieve much better performance than traditional methods using hand-crafted features. However, the current deep networks used for cloud detection depend on massive pixel-level annotation labels, which require a great deal of manual annotation labor. To reduce the labor needed for annotating the pixel-level labels, this paper proposes a weakly supervised deep learning-based cloud detection (WDCD) method using block-level labels indicating only the presence or the absence of cloud in one RS image block. In the training phase, a new global convolutional pooling (GCP) operation is proposed to enhance the ability of the feature map to represent useful information (e.g., spatial variance). In the testing phase, the trained deep networks are modified to generate the cloud activation map (CAM) via the local pooling pruning (LPP) strategy, which prunes the local pooling layers of the deep networks that are trained in the training phase to improve the quality (e.g., spatial resolution) of CAM. One large RS image is cropped into multiple overlapping blocks by a sliding window, and then the CAM of each block is generated by the modified deep networks. Based on the correspondence between the image blocks and CAMs, multiple corresponding CAMs are collected to mosaic the CAM of the large image. By segmenting the CAM using a statistical threshold against a clear-sky surface, the pixel-level cloud mask of the testing image can be obtained. To verify the effectiveness of our proposed WDCD method, we collected a new global dataset, for which the training dataset contains over 200,000 RS image blocks with block-level labels from 622 large GaoFen-1 images from all over the world; the validation dataset contains 5 large GaoFen-1 images with pixel-level annotation labels, and the testing dataset contains 25 large GaoFen-1 and ZiYuan-3 images with pixel-level annotation labels. Even under the extremely weak supervision, our proposed WDCD method could achieve excellent cloud detection performance with an overall accuracy (OA) as high as 96.66%. Extensive experiments demonstrated that our proposed WDCD method obviously outperforms the state-of-the-art methods. The collected datasets have been made publicly available online (https://github.com/weichenrs/WDCD).

1. Introduction

With the rapid development of remote sensing (RS) technology, RS images have been widely utilized in various applications. Compared with the active observation techniques (e.g., Synthetic Aperture Radar), optical RS imagery has remarkable advantages including a low price and a clear record of detailed information about the observed objects. Nevertheless, optical RS imagery is often degenerated because of cloud cover. As derived from the MODIS cloud mask, the global cloud fraction is approximately 67% (King et al., 2013). Generally, cloud detection in the optical RS imagery has two main application requirements, including the online and offline modes. The first online demand originates from the on-board communication processor module (Shan et al., 2009; Tan et al., 2016). To save the network bandwidth and storage space, the on-board processor needs to rapidly detect the cloud cover in the RS imagery and selectively transmit the fresh RS images based on the cloud cover rates. The second offline demand comes from the ground systems (Schmitt et al., 2019; Xu et al., 2019; Zhang et al.,

* Corresponding authors.
E-mail addresses: yansheng.li@whu.edu.cn (Y. Li), zhangyj@whu.edu.cn (Y. Zhang), yhtan@hust.edu.cn (Y. Tan).

https://doi.org/10.1016/j.rse.2020.112045
Received 30 December 2019; Received in revised form 6 August 2020; Accepted 10 August 2020
Available online 21 August 2020
0034-4257/

2019). As the preprocessing step of producing the wide-range RS imagery without clouds, cloud detection and removal can provide data support for continuous cartography and dynamic monitoring. Driven by various applications, cloud detection in RS imagery attracts extensive research interest. Although numerous methods have been proposed, off-the-shelf cloud detection methods tend to have limited performance and weak universality. Hence, cloud detection in the RS imagery is still facing challenges, and it is worthwhile to devote much effort to investigating this topic.

So far, many cloud detection methods are mainly designed for low-resolution RS imagery (e.g., MODIS (Ishida et al., 2018)) and medium-resolution RS imagery (e.g., Landsat (Zhu et al., 2015; Chai et al., 2019; Qiu et al., 2017; Qiu et al., 2019)). The images generally consist of many spectral bands that are beneficial to improvement of the cloud detection accuracy (Huang et al., 2010). Because of the growing number of high-resolution RS satellites that have been launched, multispectral RS imagery with four spectral bands has become increasingly prevalent. Compared with low-resolution and medium-resolution RS imagery, high-resolution RS multispectral imagery has higher spatial resolution but fewer spectral bands. As pointed out in (Li et al., 2017), the limited spectral information increases the ambiguity and confusion between cloud and the underlying surface, which makes it more difficult for cloud detection in the high-resolution RS imagery with only four spectral bands. Accordingly, it becomes very urgent to exploit the cloud detection technique for high-resolution RS imagery.

In recent years, different kinds of cloud detection methods have been proposed based on either hand-crafted features or deep learning. The deep learning-based cloud detection methods obviously outperform the methods based on hand-crafted features (Francis et al., 2019), but their superior performance highly depends on massive pixel-level cloud masks. Considering that different kinds of satellite imagery often have a large variance in terms of the spectrum and spatial resolution, a deep learning-based cloud detection method needs a corresponding pixel-level annotation dataset for each kind of satellite imagery, which requires a great deal of manual annotation labor. From this perspective, it is of great significance to explore the advanced deep learning-based cloud detection method to save annotation labor.

It is well known that scene-level/block-level labels are much easier to collect than pixel-level annotations of images. With the aid of global pooling operations, such as global average pooling (GAP), researchers (Zhou et al., 2014; Zhou et al., 2016a; Zhou et al., 2018) in the computer vision community have shown that deep networks trained with only scene-level/block-level labels are informative of object locations and can even be used for semantic segmentation. However, due to the use of local pooling layers and the inherent defects of global pooling operations (Li et al., 2018a), there is a lack of capability to obtain detailed information on objects with the existing methods, which is quite important for accurately detecting the cloud boundary in cloud detection tasks. Generally, the potential of weakly supervised deep learning has not been completely exploited to address cloud detection, and it deserves further exploration.

In this paper, we leverage only block-level supervision to train deep networks for pixel-level cloud detection. With the consideration that the accurate detection of clouds requires more useful information, we propose a new global pooling operation called global convolutional pooling (GCP) in the training phase, which learns channel-independent convolutional weights to enhance the ability of the feature map to represent useful information (e.g., spatial variance). Furthermore, we propose a novel local pooling pruning (LPP) strategy for use in the testing phase during generation of the cloud activation map (CAM), which is used to generate the final cloud mask. By pruning the local pooling layers in the trained deep networks, the quality (e.g., spatial resolution) of the CAM is improved, and the classification performance of the deep networks remains stable. Then, the final cloud mask of one RS image can be obtained through segmenting the CAM with a statistical threshold against a clear-sky surface, which can be calculated by using negative samples in the training dataset.

It is noted that thin cloud and thick cloud are treated in the same way in this paper, which means our proposed method strictly detects all the clouds together regardless of whether they are thin or thick. However, the thin or thick cloud can be distinguished if needed using some methods such as hand-crafted features or indexes (Li et al., 2017). With regard to the shadow around the cloud, it can also be discriminated after the accurate detection of the cloud. For instance, according to the solar elevation angle and solar azimuth angle from the image metadata, the location of the shadow can be easily inferenced according to the detected region of the cloud, which will be displayed in our future work.

Considering that there do not exist any qualified datasets that can be used to evaluate the weakly supervised deep learning-based cloud detection (WDCD) technique, we collected a new global dataset based on the multispectral imagery from the GaoFen-1 satellite. Specifically, the GaoFen-1 satellite includes two integrated cameras with 8-m spatial resolution and 4-day temporal resolution. Each camera has four multispectral bands, spanning the visible to the near-infrared spectral regions. The collected global dataset includes the training dataset, the validation dataset and the testing dataset. The training dataset consists of more than 200,000 image blocks cropped from 622 large GaoFen-1 images distributed around the world, and each image block has a binary label indicating if the block contains a cloud or not. The validation dataset consists of 5 large GaoFen-1 images with manually annotated pixel-level cloud masks, which are generally used to tune the hyperparameters. The testing dataset is composed of 25 large RS images with manually annotated pixel-level cloud masks, with 19 large images from the GaoFen-1 satellite and 6 large images from the ZiYuan-3 satellite. As displayed in Tables 1 and 2, the GaoFen-1 imagery has a similar bandwidth and spatial resolution setting as the ZiYuan-3 imagery. Since the GaoFen-1 imagery is one kind of typical high-resolution RS imagery, both the collected dataset and the proposed method possess a good generality. It is worth mentioning that the testing dataset is qualified for use to evaluate the cloud detection performance for multisource RS data.

In the experimental setting, we train deep networks under the supervision of RS image blocks but pursue the pixel-level cloud detection with coarse labels that only indicate whether an image block contains a cloud or not. Even under this extreme setting, our proposed method still yields promising results and outperforms the existing methods (Simonyan and Zisserman, 2014; Zhang and Xiao, 2014; Zhou et al., 2016a; Li et al., 2018a; Zou et al., 2019). The main contributions of this paper can be summarized as follows:

- This paper proposes a new WDCD method that trains the deep networks under block-level supervision for pixel-level cloud detection in high-resolution RS imagery.
- This paper proposes a novel global pooling operation (i.e., GCP). Compared with the existing global pooling operation (e.g., GAP), GCP can learn channel-independent convolutional weights to enhance the ability of the feature map to represent useful information (e.g., spatial variance). Additionally, the deep networks can be trained in an end-to-end manner.
- This paper proposes a new LPP strategy to generate the CAM. The LPP strategy dramatically enhances the quality (e.g., spatial

Table 1
The bandwidth information of the used satellite imagery.

Multi-spectral imagery	GaoFen-1/ZiYuan-3
Band 1 (Blue/um)	0.45–0.52
Band 2 (Green/um)	0.52–0.59
Band 3 (Red/um)	0.63–0.69
Band 4 (Near-Infrared/um)	0.77–0.89

Table 2
The spatial resolution information of the used satellite imagery.

Multi-spectral imagery	GaoFen-1	ZiYuan-3
Spatial resolution (m)	8	5.8

resolution) of the generated CAM.

- Last but not least, this paper collects a new global RS image dataset for weakly supervised cloud detection. Specifically, the training dataset contains over 200,000 image blocks with block-level binary labels (i.e., containing cloud or not), and the validation and testing datasets contain 30 large RS images with pixel-level cloud masks.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 specifically displays the collected dataset. Section 4 introduces our proposed WDCD method. Section 5 reports the experimental results. Section 6 gives a discussion of the experimental details. Finally, Section 7 gives the conclusion of this paper.

2. Related work

In this section, we briefly review the most relevant works in the literature that include weakly supervised deep learning and cloud detection in RS imagery.

2.1. Weakly supervised deep learning

To alleviate the labor of bounding box annotations, pioneers in computer vision exploit scene-level or image-level tags as weak supervision for localizing objects in images or scenes. More specifically, multi-instance learning was combined with deep convolutional features (Pinheiro and Collobert, 2015; Pathak et al., 2015; Cinbis et al., 2017; Wang et al., 2019b) to localize objects. Similarly, region proposal-based methods using weak supervision (Bilen and Vedaldi, 2016; Tang et al., 2017; Tang et al., 2018b) have been proposed to solve object detection. Based on the observation that the action depicted in the image/video can provide strong cues about the location of the associated object, Yang et al. (2019) and Singh and Lee (2019) leveraged the action labels to improve the performance of weakly supervised object detection. Most recently, the idea of weak supervision has also been widely explored in semantic segmentation (Kolesnikow and Lampert, 2016; Wei et al., 2016; Chen et al., 2018; Tang et al., 2018a) and saliency detection (Wang et al., 2017; Hsu et al., 2019). Moreover, Wei et al. (2018) revisited dilated convolution and proposed to leverage multiple convolutional blocks with different dilated rates to generate dense object localization maps. Gao et al. (2018) proposed count-guided weakly supervised localization that uses the per-class object count as a new form of supervision to improve weakly supervised localization. Wan et al. (2018) proposed a min-entropy latent model to address weakly supervised object detection, which combined recurrent learning with region proposals. Zhang et al. (2019) applied the idea of adversarial learning, which learns two parallel-classifiers, to leverage complementary object regions for classification and finally generate integral object localization together. In general, these methods were originally designed for natural images and fall short of detecting detailed information. Thus, they cannot be directly used for RS images because they have insufficient capability to handle the challenges in RS images, which contain complex backgrounds and densely distributed objects with arbitrary orientations (Li et al., 2018a).

2.2. Cloud detection in RS imagery

Cloud detection is a common issue in the RS domain. Traditional cloud detection algorithms can be divided into two categories: hand-crafted feature-based approaches and deep learning-based approaches. The hand-crafted feature-based approaches detect clouds either using a constant or adaptive threshold in different spectral bands derived from the physical characteristics of clouds, such as spectrum, texture, temperature and elevation (Wilson and Oreopoulos, 2013; Oishi et al., 2018; Wang et al., 2019a), or using manual features to train classifiers (e.g., decision trees (Hollstein et al., 2016), fuzzy models (Shao et al., 2017) and support vector machines (Ishida et al., 2018)). Researchers (Zhu et al., 2015; Zhou et al., 2016b; Li et al., 2017; Qiu et al., 2017; Qiu et al., 2019) also combine several methods together and use multifeatures for cloud detection. Although existing methods have achieved promising results under their specific experimental settings, these approaches possess limited generalization ability to some extent. There still exists much space to improve the performance.

Motivated by the great progress of deep learning (Krizhevsky et al., 2012; LeCun et al., 2015; Li et al., 2018b; Li et al., 2018c; Tan et al., 2018; Tao et al., 2019a; Tao et al., 2019b; Li et al., 2020), researchers in the RS community set up to develop the deep learning-based cloud detection approaches. By formulating cloud detection as a semantic segmentation problem, they modified several popular networks designed for semantic segmentation such as SegNet (Chai et al., 2019), U-Net (Wieland et al., 2019; Jeppesen et al., 2019), FCN (Mohajerani and Saeedi, 2019; Shao et al., 2019), DeepLab (Segal-Rozenhaimer et al., 2020) for cloud detection. Segal-Rozenhaimer et al. (2020) utilizes the domain adversarial neural networks to perform the cloud detection across multisource satellite imagery. Nevertheless, the performance of these deep learning-based methods highly depends on the number of training samples and the accuracy of their labels. Recently, Zou et al. (2019) formulated cloud detection as a mixed energy separation process between the foreground and background of images. Specifically, the generative adversarial framework was adopted to conduct weakly supervised matting of a cloud image by incorporating the physics behind it. It is worth noting that this work does not depend on pixel-level annotation. As a whole, this kind of work based on weak supervision is still in the embryonic stage. However, it requires massive efforts to improve the cloud detection performance by fully exploiting the weak supervision condition.

3. Dataset description

In this section, we detail the description of the dataset, which is specifically collected for evaluating cloud detection via weakly supervised deep learning.

3.1. The training dataset with block-level labels

In recent years, researchers (Chai et al., 2019; Shao et al., 2019; Jeppesen et al., 2019) have tended to regard cloud detection as a semantic segmentation problem, which uses pixel-level labels to indicate whether each pixel contains cloud or not. Given that there are no existing datasets for weakly supervised cloud detection methods such as our WDCD method, we created a large-scale block-level global dataset for it using the GaoFen-1 Level-1A imagery (Digital Number). The training dataset consists of 206,384 image blocks with their binary labels, which denote whether the block contains cloud or not. Each image block has a size of 250 × 250 and 4 spectral bands. First, image blocks are randomly cropped from 622 large GaoFen-1 Level-1A images distributed all over the world. With the aid of visual interpretation, domain experts label the blocks without any cloud pixels as negative samples and annotate the blocks with a cloud cover rate over 25% as positive samples for training stability. In total, 51,596 blocks are manually collected. Then, rotation is performed for each image block at angles of 90, 180 and 270 degrees so that the number of image blocks becomes four times that of the original number and is increased to 206,384. There are in total 109,312 negative samples and 97,072 positive samples.

As seen in Fig. 1, the negative samples do not contain any cloud and

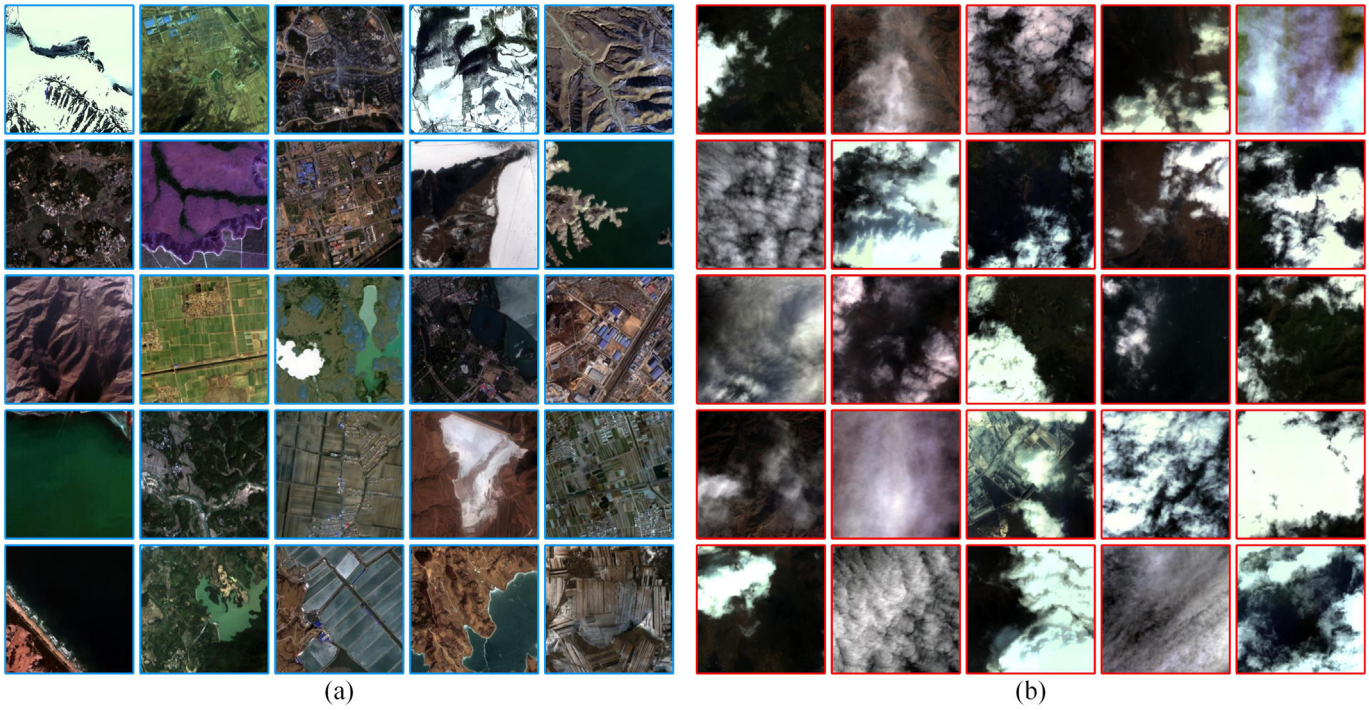

Fig. 1. Visual example of the training dataset. (a) Shows the negative samples (i.e., image blocks do not contain any cloud). (b) Illustrates positive samples (i.e., image blocks are covered by over 25% cloud).

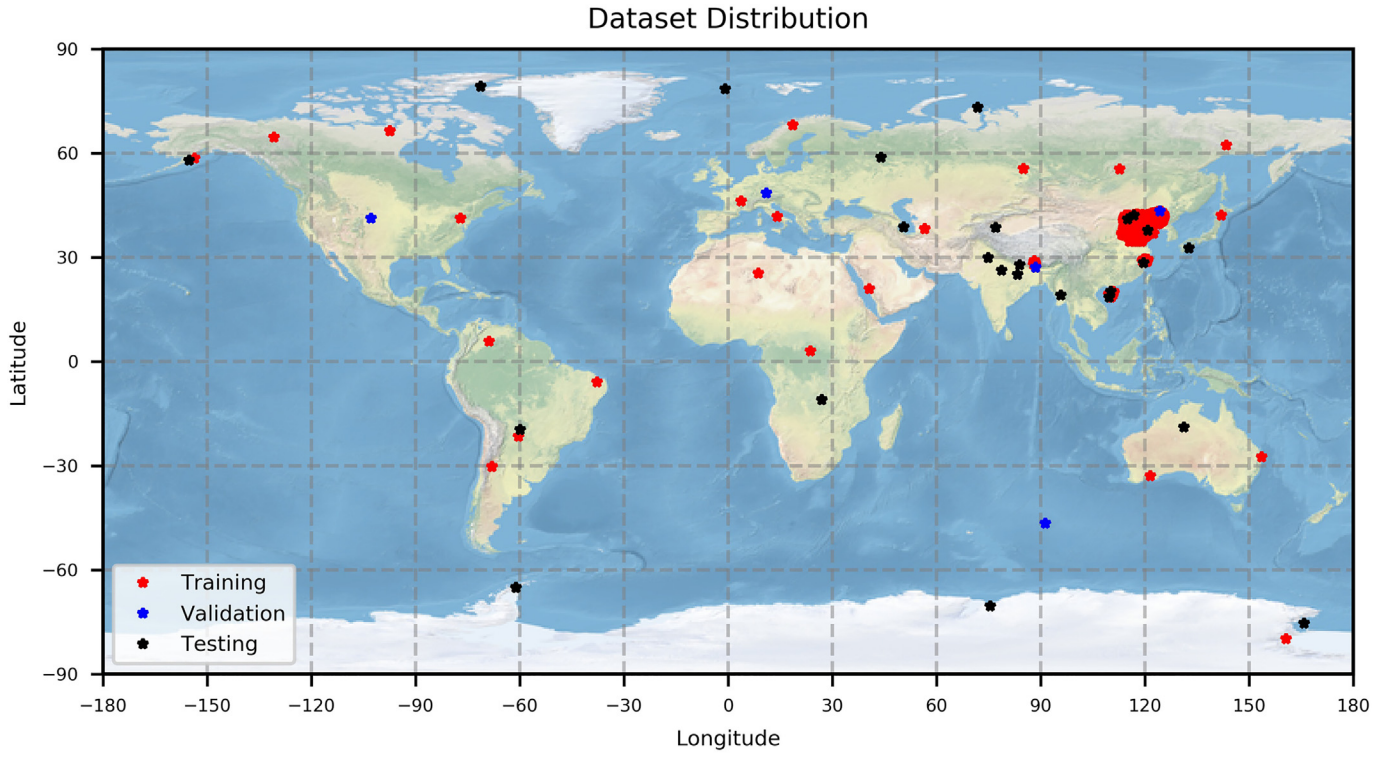

Fig. 2. The distribution of samples from the training, validation, and testing datasets. To avoid overfitting, the images from the training, validation, and testing datasets are not overlapped with each other.

include different land cover types such as ice, snow, bare land, vegetation, water, building, and farmland. The positive samples are covered by at least 25% cloud and include various cases in which image blocks are covered by clouds with different shapes, volumes, and underlying surfaces. The global distribution characteristic of samples from the training dataset is visually depicted in Fig. 2.

3.2. The validation and testing dataset with pixel-level labels

To determine the hyperparameters of the method and evaluate the cloud detection performance, we manually annotate 30 large RS images with pixel-level labels. In this work, both thin and thick clouds are considered, and both of them are carefully labeled by domain experts.

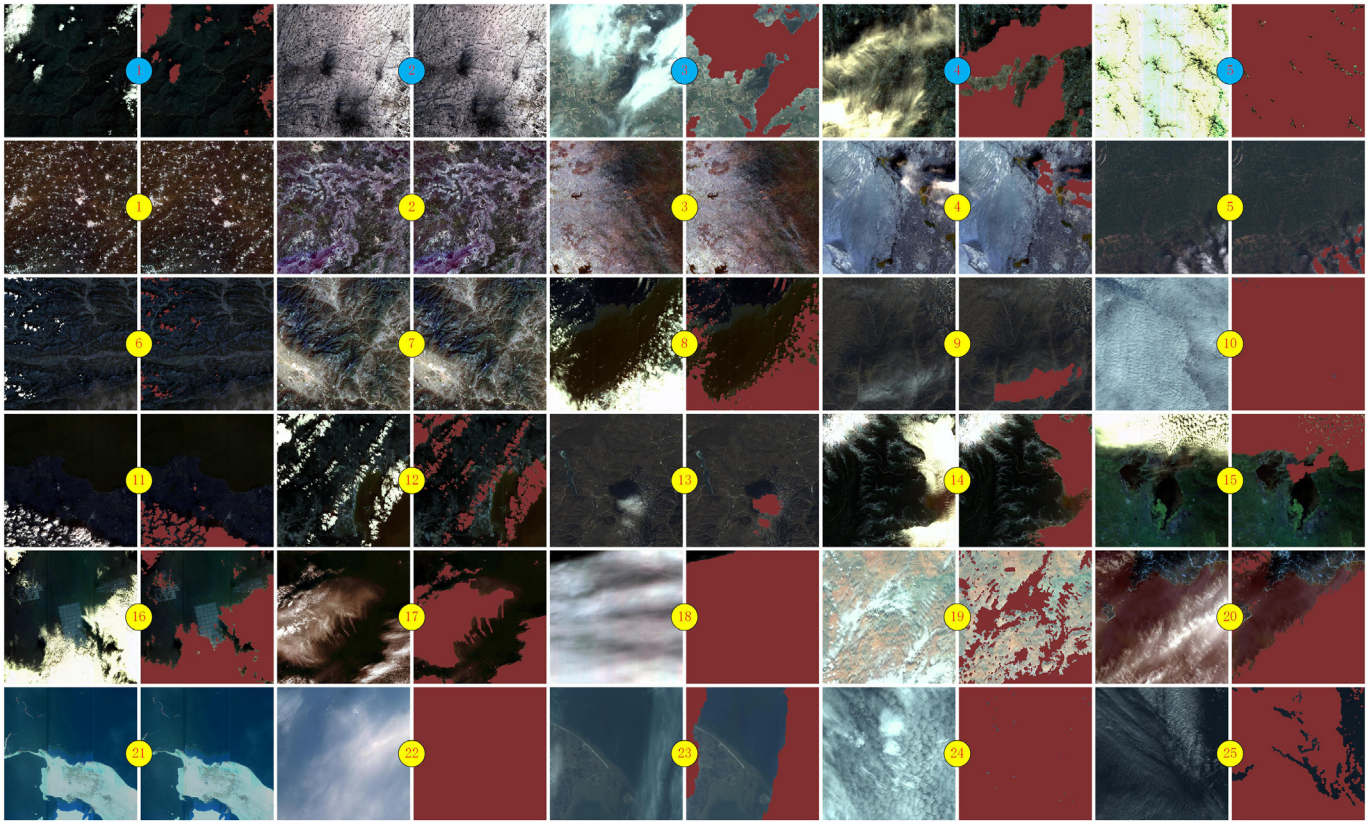

Fig. 3. The validation and testing datasets. The figure shows 30 large RS images and their corresponding pixel-level annotations. Each image is followed by its annotations where the red-color regions denote the cloud masks. The pairs in the first row with blue labels indicate the validation dataset, and those in the remaining four rows with yellow labels come from the testing dataset. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

As shown in Fig. 3, five of the images are from different regions and are used as validation to tune the hyperparameters, while the other twenty-five globally distributed images are used for testing and evaluating the overall performance. It is noted that all the pixel-level labels are only used for evaluating but not for training. Specifically, all images in the validation dataset (i.e., the 1st to the 5th validation images) are from the GaoFen-1 satellite. The 1st to the 6th testing images are from the ZiYuan-3 satellite, while the others are from the GaoFen-1 satellite. As seen in Tables 1 and 2, ZiYuan-3 and GaoFen-1 have the same bandwidth setting but different spatial resolutions. It is worth mentioning that 6 large ZiYuan-3 images (i.e., the 1th to the 6th testing images) are added to verify the generalization ability of our proposed WDCD approach. We chose these large testing images according to the variety in terms of cloud covers, land surfaces, and geographical locations. The images in the testing dataset contain several typical land cover types such as agriculture, water, grassland, bare, buildings, and snow/ice. Moreover, different kinds of cloud covers are included (e.g., 100% cloud cover, clear-sky, cumulus clouds, stratus cloud, cirrus cloud, and mixed). The distribution of validation and testing datasets is also depicted in Fig. 2. As seen, these images observed in various locations distributed all over the world show good universality. All datasets have been made available online (https://github.com/weichenrs/WDCD).

4. Methodology

In this section, we give the details of our proposed WDCD method. Section 4.1 gives the structure of our deep networks and the learning process in the training phase. The method used to perform the pixel-level cloud detection using the trained deep networks is introduced in Section 4.2.

4.1. Learning deep networks under block-level supervision

To clarify our WDCD method, we first depict the whole framework of our proposed deep networks based on GCP and intuitively illustrate the superiority of GCP. Then, we give the implementation details for learning the deep networks.

With the block-level labels, it is easy to build discriminative deep networks (e.g., VGG (Simonyan and Zisserman, 2014)) to classify the image blocks as cloud or non-cloud without the ability to perform object localization and segmentation. The development of GAP showed that block-level supervision can be used for object localization (Zhou et al., 2016a), but the localization accuracy needs to be improved due to the weak connectivity yielded by the GAP. Li et al. (2018a) improved the GAP method by utilizing a two-stage-learning (TSL) method for object localization, whereas the networks cannot be learned in an end-to-end way. To overcome the aforementioned limitation, we proposed the WDCD framework based on GCP.

As depicted in Fig. 4, during the training phase, the architecture of our deep networks is similar to that of the common convolutional neural networks (CNN) used for image recognition, where the CNN is composed of local convolutional (Conv) operations and local pooling (LP) operations. Under block-level supervision, normal CNNs are designed for block-level classification tasks; however, we employ it to perform pixel-level cloud detection under block-level supervision by leveraging the intermediate feature maps of the convolutional layers. For this reason, we replaced the GAP or fully connected layer with our proposed GCP layer, in order to promote the representation ability of the feature map. As displayed in Fig. 4, the feature map is performed with a spatialwise convolution of each channel with the GCP layer, by which the spatial variance will be well represented after several iterations of back-propagation.

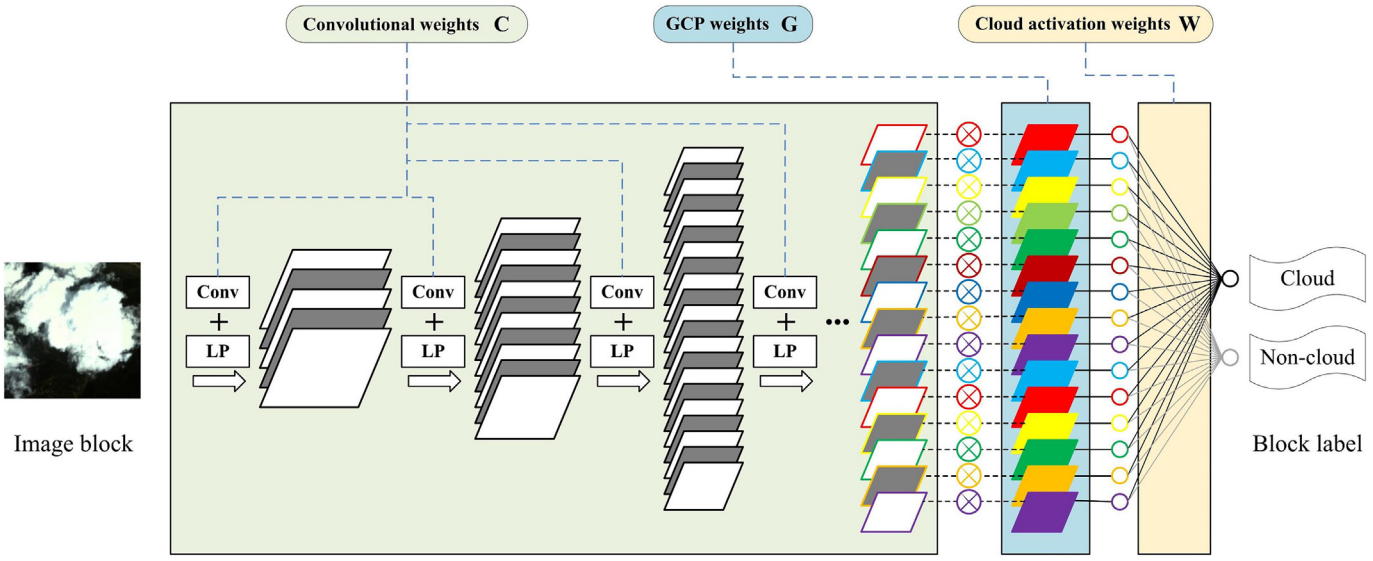

Fig. 4. The architecture of our adopted deep networks.

As mentioned in Section 1, GAP (Zhou et al., 2016a) is a popular global pooling operation. Due to the usage of the global pooling operation, there is only a very weak connectivity between the block label and feature map outputted by the last convolutional layer. To facilitate understanding, we give a toy example to show the weak connectivity that GAP yields and the drawback of this weak connectivity in Fig. 5(a) and (b). During the forward propagation, the spatial units of each channel in the feature map outputted by the last convolutional layer are aggregated into one single unit in the aggregation feature vector as illustrated in Fig. 5(a). Accordingly, the gradient value of each unit in the aggregation feature vector is equally divided into the spatial units of each corresponding channel in the feature map in the backward gradient propagation illustrated in Fig. 5(b). However, this process impairs perceiving the spatial variance of each channel.

Considering that GAP can impair perceiving the spatial variance of each channel, we propose a novel global pooling operation named GCP. As seen in Fig. 5(c) and (d), the GCP layer trains d channel-independent convolutional kernels where d denotes the number of channels, which will be further discussed in Section 6.1. Unlike the GAP operation shown in Fig. 5(a) and (b), the GCP layer learns channel-independent convolutional weights, and the feature map is used to perform spatialwise convolution of each channel with the learned GCP weights. After the iterative forward and backward propagations during training, we consider that the GCP layer owns the capability to exploit the important and useful information (e.g., spatial variance) of the feature map intuitively.

Outwardly, our proposed GCP is similar to the Depthwise Separable Convolution (DSC) (Chollet, 2017) in its computation form. However, there are some substantial differences between them. First, the convolutional kernel sizes and outputs are different. The kernel size of GCP is the same as the input feature map, and the output of GCP is a k-dimensional feature vector. According to DSC, its kernel size is smaller than the input feature map, while the output of DSC is still a feature map. Second, their purposes are totally different. We adopt GCP, which brings learnable parameters to capture the spatial variance of the feature map, whereas the DSC is used to reduce the number of parameters. The last but the most important difference is that GCP can serve as a variant of global pooling (e.g., global average pooling), which performs spatialwise convolution with each channel of the feature map to obtain the global features.

Let $\{(b_n, y_n)|\, n = 1, 2, \cdots, N\}$ denote the training cloud dataset. More specifically, N is the number of image blocks in the training dataset, b_n stands for the n-th image block, and y_n denotes its label (i.e., $y_n = [1, 0]$ indicates that the given image block contains cloud and $y_n = [0, 1]$ means that the given image block does not contain any cloud).

Let $\mathbf{\Psi} = \{\mathbf{C}, \mathbf{G}, \mathbf{W}\}$ denote all of the weights of the deep networks, where $\mathbf{C}$ stands for the weights of the hierarchical convolutional layers, $\mathbf{G}$ denotes the weights of the GCP layer, and $\mathbf{W}$ stands for the cloud activation weights. For a given image block b_n, it is sent to the deep networks and outputs the feature map ${f_n}^k$ as Eq. (1).

$$f_n^k = \varphi^k(b_n; \mathbf{C}) \tag{1}$$

where ${f_n}^k$ denotes the k-th channel of the last convolutional layer's output feature map, φ denotes the representation of convolution, pooling, activation computation in the deep networks. By global convolutional pooling ${f_n}^k$ per channel, we calculate the activation value of ${f_n}^k$ at each channel as depicted in Eq. (2).

$$O_n^k = f_n^k \otimes G^k \tag{2}$$

where ${O_n}^k$ denotes the activation value of ${f_n}^k$ at the k-th channel, $G^k \in \mathbf{G}$ stands for the weights of the GCP layer at the k-th channel, $\otimes$ denotes the spatialwise convolution channel by channel.

In this experimental setup, each training image block has its binary label indicating if the block contains cloud or not. Therefore, the softmax-based cross-entropy loss function is taken to learn the networks $\mathbf{\Psi} = \{\mathbf{C}, \mathbf{G}, \mathbf{W}\}$ and model the connectivity between the global convolutional result and the block label, which is specified by Eq. (3).

$$\begin{aligned}\min_{\mathbf{\Psi}=\{\mathbf{C},\mathbf{G},\mathbf{W}\}} \boldsymbol{J} &= -\sum_{n=1}^{N}\sum_{c=1}^{2} y_n^c \times \log\left(\frac{\exp\left(\sum_{k=1}^{d} W_k^c \times O_n^k + W_0^c\right)}{\sum_{c=1}^{2}\exp\left(\sum_{k=1}^{d} W_k^c \times O_n^k + W_0^c\right)}\right) \\ &= -\sum_{n=1}^{N}\sum_{c=1}^{2} y_n^c \times \log\left(\frac{\exp\left(\sum_{k=1}^{d} W_k^c \times ((\varphi^k(b_n;\mathbf{C})) \otimes G^k) + W_0^c\right)}{\sum_{c=1}^{2}\exp\left(\sum_{k=1}^{d} W_k^c \times ((\varphi^k(b_n;\mathbf{C})) \otimes G^k) + W_0^c\right)}\right)\end{aligned} \tag{3}$$

where ${W_k}^1 \in \mathbf{W}$ denotes the cloud activation weights, which indicate the contribution of f^k for cloud.

By optimizing the function in Eq. (3), the convolutional weights $\mathbf{C}$, the GCP weights $\mathbf{G}$ and the cloud activation weights $\mathbf{W}$ are learned simultaneously. In Section 4.2, we will introduce how to conduct cloud detection using the learned deep networks, whose parameters are composed of the convolutional weights $\mathbf{C}$, the GCP weights $\mathbf{G}$ and the

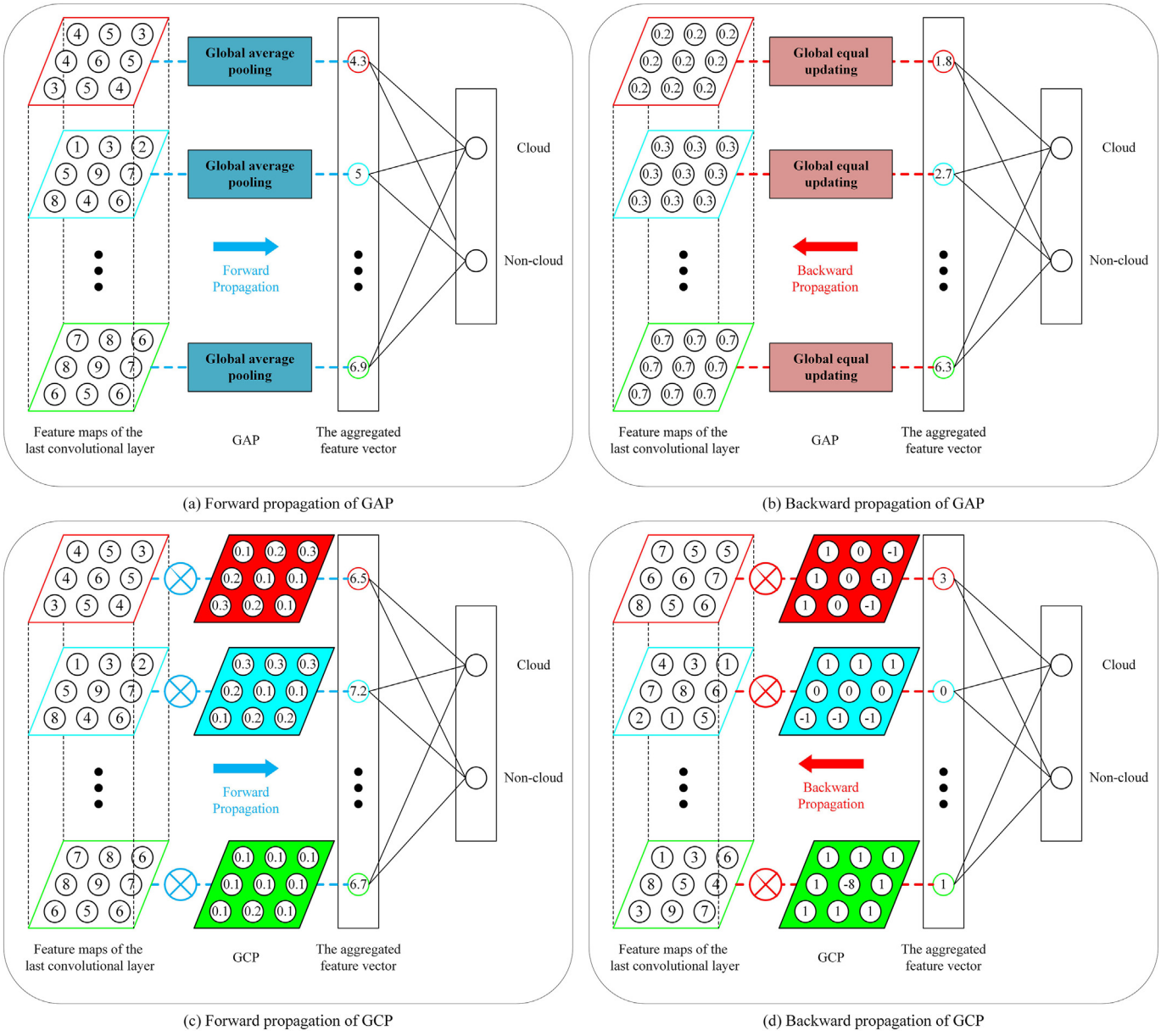

Fig. 5. The toy example of forward and backward propagation of GAP and GCP. (a) The forward propagation of GAP. (b) The backward propagation of GAP. (c) The forward propagation of GCP. (d) The backward propagation of GCP.

cloud activation weights **W**.

4.2. Pixel-level cloud detection using the trained deep networks

Section 4.2.1 discusses the effects of local pooling layers and introduces the local pooling operation pruning (LPP) strategy, which we use to modify the trained deep networks. In Section 4.2.2, we introduce how to automatically generate the CAM of one large RS image using the learned deep networks in Section 4.1. In addition, we give a brief summary of our proposed WDCD approach in Section 4.2.3.

4.2.1. Cloud activation maps for blocks via local pooling pruning

Local pooling operation is one common and essential part of convolutional neural networks. Typically, the local pooling operation is utilized to reduce the cost of memory and computation, enlarge the receptive field, and provide the translation invariance. However, the use of local pooling is an infinitely strong prior that each unit should be invariant to small translations. Local pooling is only useful when the assumptions made by the prior are reasonably accurate. If a task relies on preserving precise spatial information, then using local pooling on all features can increase the training error (Goodfellow et al., 2016). Furthermore, Ruderman et al. (2018) find that pooling layers are neither necessary nor sufficient for achieving the optimal form of deformation stability for natural image classification. Additionally, pooling confers too much deformation stability for image classification at initialization, and during training, networks have to learn to counteract this inductive bias (Ruderman et al., 2018).

It is well known that deep CNNs (DCNNs) are generally composed of multiple convolutional layers and local pooling layers. In addition, the convolutional layers contain learnable weights, but local pooling layers do not contain any weights and aim to pursue the shift-invariance and rotation-invariance by decreasing the size of feature maps. Due to the homogenization characteristic of cloud, shift and rotation are not the critical factors in the cloud detection task. To a certain degree, the kinds of cloud samples naturally cover the shift-rotation-invariance cases when the volume of various training samples is large enough. Hence, it

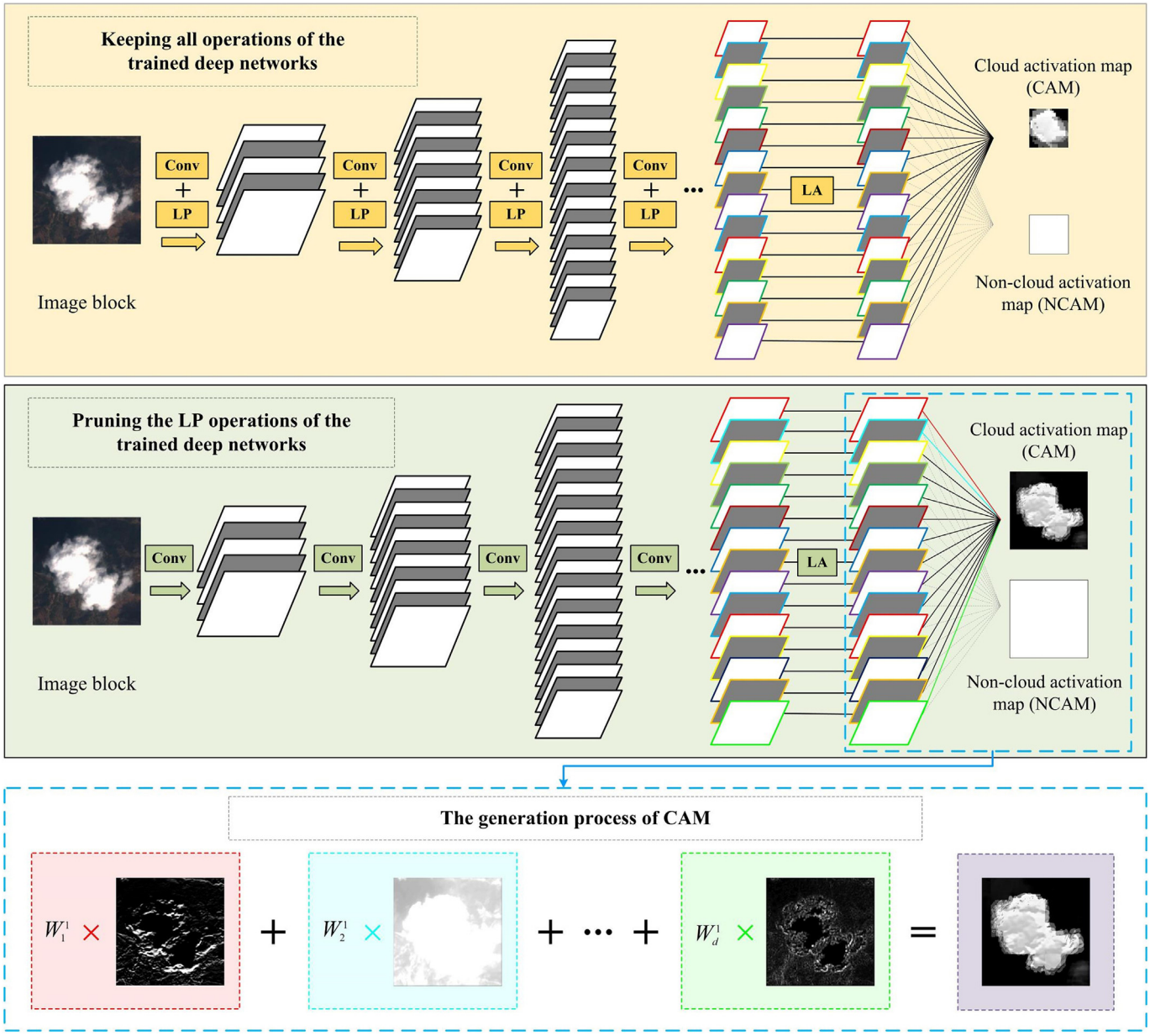

Fig. 6. The difference between deep networks with and without local pooling pruning (LPP). It is noted that we do not generate NCAM in this figure.

seems to be reasonable to use the convolutional layers as feature extractors and prune the local pooling layers in the testing phase to pursue the higher resolution of the feature maps. In addition, this statement is also verified in the experimental section.

Based on the aforementioned theory and the consideration that the cloud detection task requires preserving precise spatial information, we prune the local pooling layers from our cloud detection networks when generating the CAM. and this operation is named as LPP. Extensive experiments show that the LPP operation enhances the spatial resolution of the output CAM with the performance of networks remaining stable. Fig. 6 depicts the difference between deep networks with and without LPP when generating the CAM. The spatial resolution of CAM increases significantly from 20 × 20 to 230 × 230 when we adopt the LPP strategy. It is noted that, whether the LPP operation is utilized or not, the spatial resolution of the generated CAM will be resized to 250 × 250 so that the CAM will correspond to the size of the input image block.

Fig. 7 shows the comparison of the results generated by our method with and without LPP. To better illustrate the superiority of LPP, we used the synthetic image of the results. As depicted, the LPP dramatically enhanced the quality (i.e., spatial resolution) of the CAM, which is the key to detecting small and densely distributed objects. That is, the LPP operation can be leveraged in DCNN-based detection tasks that highly depend on the high spatial resolution of the output feature map, such as small-object detection, UAV image object detection and so on.

Given one image block b, the feature map f of the last convolutional layer can be calculated by Eq. (1) based on the convolutional weights $\mathbf{C}$, then the f is used to compute the activation value at each channel with the GCP weights $\mathbf{G}$. Due to the LPP operation, the feature map has a larger size than the GCP weights so that the GCP weights are resized to the same size as the feature maps before computation. After that we adopt a linear adjustment (LA) operation by Eq. (4).

$$T^k = \frac{\delta(f^k)}{\tau(f^k)} \times f^k \tag{4}$$

where T^k is the modified feature map of the k-th channel; $\delta(f^k) = f^k \otimes G^k$ denotes the activation value of the k-th channel of the last convolutional layer with the aid of the GCP weights $\mathbf{G}$; $\tau(f^k)$ stands

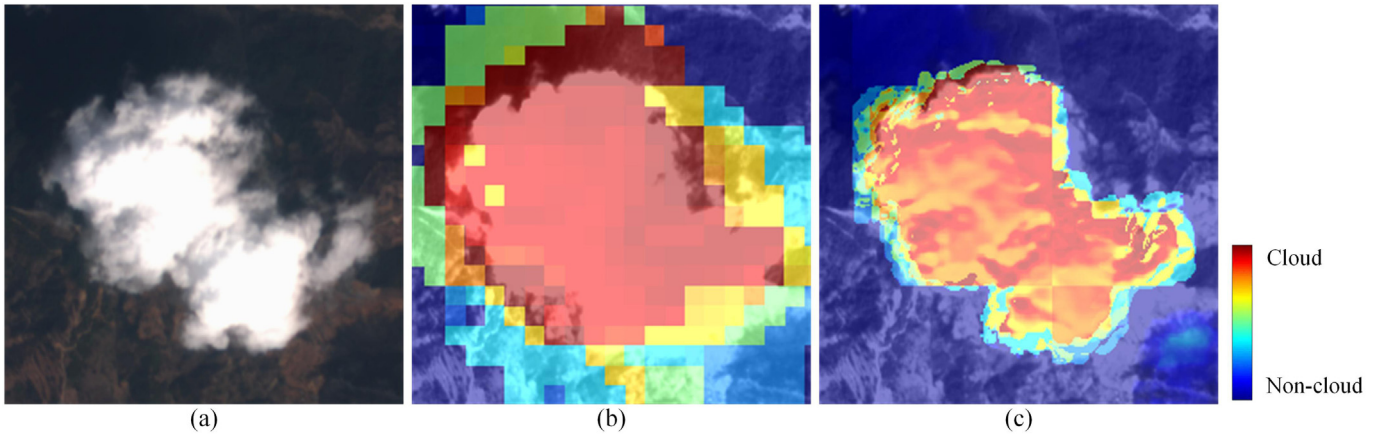

Fig. 7. The details of CAM computed with and without LPP. (a) Shows an input image block. (b) Shows the CAM of the block without LPP. (c) Illustrates the CAM of the block. It is noted that (b) and (c) depict the color scale from blue (the low cloud probability) to red (the high cloud probability). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

for a statistic value such as the average or median of f^k.

The LA operation is adopted to adjust the value of f to the appropriate range. First, we multiply the f with its activation values, After that the result will be divided by the average of f.

Furthermore, we calculate the CAM M^b of the block b by Eq. (5).

$$M^b = \sum_{k=1}^{d} W_k^1 \times T^k = \sum_{k=1}^{d} W_k^1 \times \frac{\delta(f^k)}{\tau(f^k)} \times f^k \tag{5}$$

where $W_k{}^1$, $k = 1, 2, \cdots, d$ stands for the cloud activation weight.

4.2.2. Cloud activation maps for extended scenes using sliding windows

Given one large RS image I, we obtain a set of overlapped blocks $\{b_1, b_2, \cdots, b_m\}$ by sliding windows from left to right and top to bottom. We calculate the CAM of each block from $\{b_1, b_2, \cdots, b_m\}$ using Eq. (5). To increase the detection efficiency, we first use the trained deep networks in Section 4.1 to classify the image blocks as cloud or non-cloud, and only those blocks categorized as cloud are used to compute the CAMs. Through mosaicking the block-level CAMs where the overlapped regions are fused by the average voting, the CAM M^I of the image I can be calculated by Eq. (6).

$$M^I = \text{Mosaic}(M^{b_1}, M^{b_2}, \cdots, M^{b_m}) \tag{6}$$

To facilitate clarification, the specific process for generating the CAM of one large RS image is visually shown in Fig. 8.

4.2.3. Generating the cloud mask by segmenting the cloud activation map

With the high-quality CAM, the binary cloud mask can be determined by a simple threshold segmentation algorithm. To restrain omission errors of thin clouds and small clouds, we calculated the threshold against a clear-sky surface by Eq. (7).

$$\hbar = \mu + k \times \sigma \tag{7}$$

where k is an empirical constant, μ denotes the average of the CAMs of all negative samples in the training dataset, σ stands for the standard deviation of the CAMs of all negative samples in the training dataset, and $\hbar$ denotes the threshold against clear-sky surface.

Based on the threshold $\hbar$ in Eq. (7), the binary cloud mask S^I of the image I is calculated by Eq. (8).

$$S^I(i,j) = \begin{cases} 255, & if\ M^I(i,j) \geq \hbar \\ 0, & if\ M^I(i,j) < \hbar \end{cases} \tag{8}$$

Given one RS image, the visual results of the segmenting process including the intermediate CAM and the final binary mask are visually depicted in Fig. 9.

To facilitate understanding of the proposed method, we briefly summarize the training and testing phases of our proposed WDCD approach in Fig. 10.

5. Experimental results

Section 5.1 first introduces the experimental setup of this paper. From the prediction perspective, Section 5.2 uses the user's accuracy-producer's accuracy (UA-PA) curves (Wang et al., 2017) and true positive rate-false positive rate (TPR-FPR) curves (Hanley, 1989) to

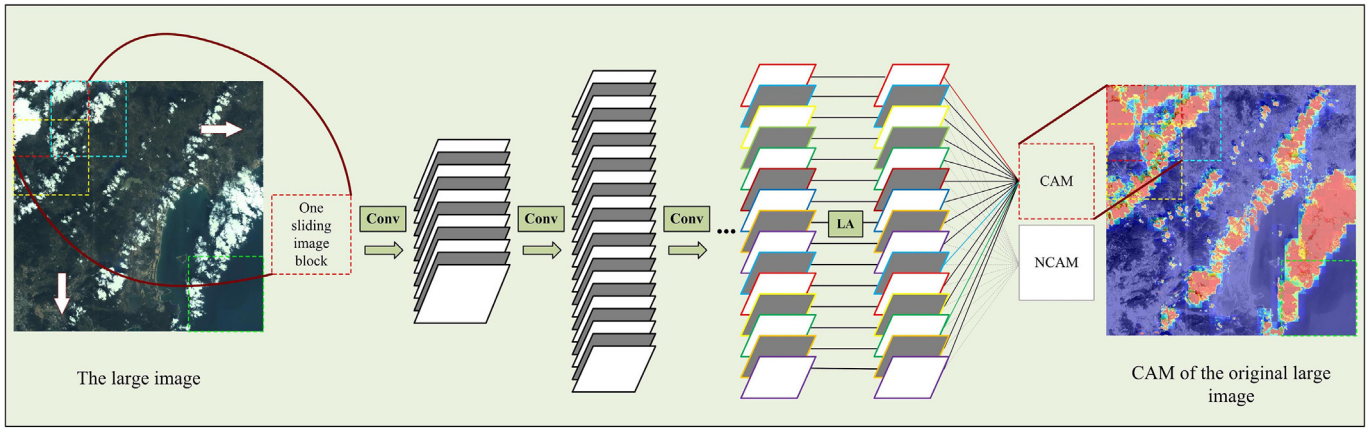

Fig. 8. The process of computing the cloud activation maps (CAM) of one large RS image.

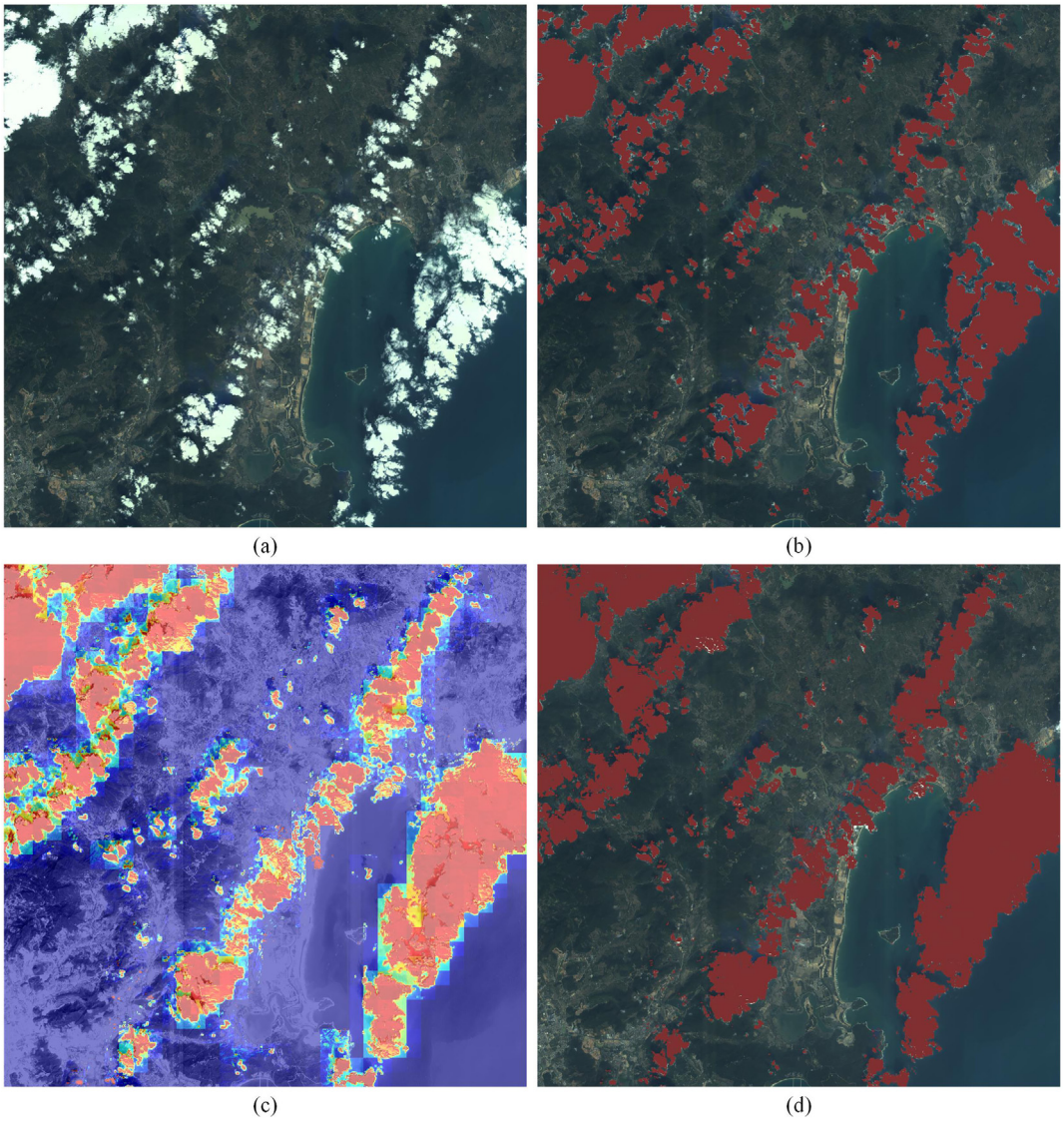

Fig. 9. The segmentation of CAM. (a) The original image, (b) The corresponding ground truth of (a), (c) The computed CAM of (a), (d) The final cloud mask computed by segmenting the CAM (c).

evaluate the cloud detection performance of the CAMs. Moreover, Section 5.3 reports quantitative detection results of our method as well as some baselines via several comprehensive metrics for evaluation of the cloud masks.

5.1. Experimental setup

In this section, we display the implementation details of our proposed WDCD method. In our implementation, we refer to the architecture of the VGG-16 net (Simonyan and Zisserman, 2014) and modify it based on our task. The details of the network structure are shown in Table 3. The size of Conv + ReLU means $s_k \times s_k \times n_{cin} \times n_{cout}$, where s_k denotes the size of convolutional kernel, n_{cin} and n_{cout} stands for the numbers of input and output channel. and the size of Local Pooling and Global Convolutional Pooling denotes $s_p \times s_p$, where s_p means the stride of the pooling window, while the size of Fully Connected stands for $n_{cin} \times n_{cout}$ which are the numbers of input and output channel. In the whole deep networks, the strides of all convolutional layers are 1, and zero padding is not used.

As far as the general parameters, we set them according to the results of our experiments and analysis. We employed the Adam optimizer (Kingma and Ba, 2015) with the default parameter setting except a learning rate that is decayed with iteration. The initial learning rate is set to 0.0001. After each iterative epoch, the learning rate will be multiplied by the learning rate decay which is empirically set to 0.9. The max iterative time is set to 10. During the training phase, the inputs of deep networks are image blocks with the size of 250 × 250, while the testing phase requires generating the cloud mask for one large image. Thus, the sliding window strategy is utilized to compute the CAM. The sliding window size was set to 250 by 250, and the sliding step was set to 125. The overlapped regions are fused by the average voting. After computing the CAM, we choose the appropriate parameter k to segment the CAM and generate the cloud detection mask.

All approaches including our proposed approach and other baselines are implemented by PyTorch and conducted on a Dell station with 8 Intel Core i7-9700 k processors, 32 GB of RAM, and the NVIDIA GeForce RTX 2080Ti.

5.2. Performance evaluation on cloud activation map

To directly verify the cloud detection performance of the CAM, this section adopts the pixel-level metrics in the saliency evaluation task

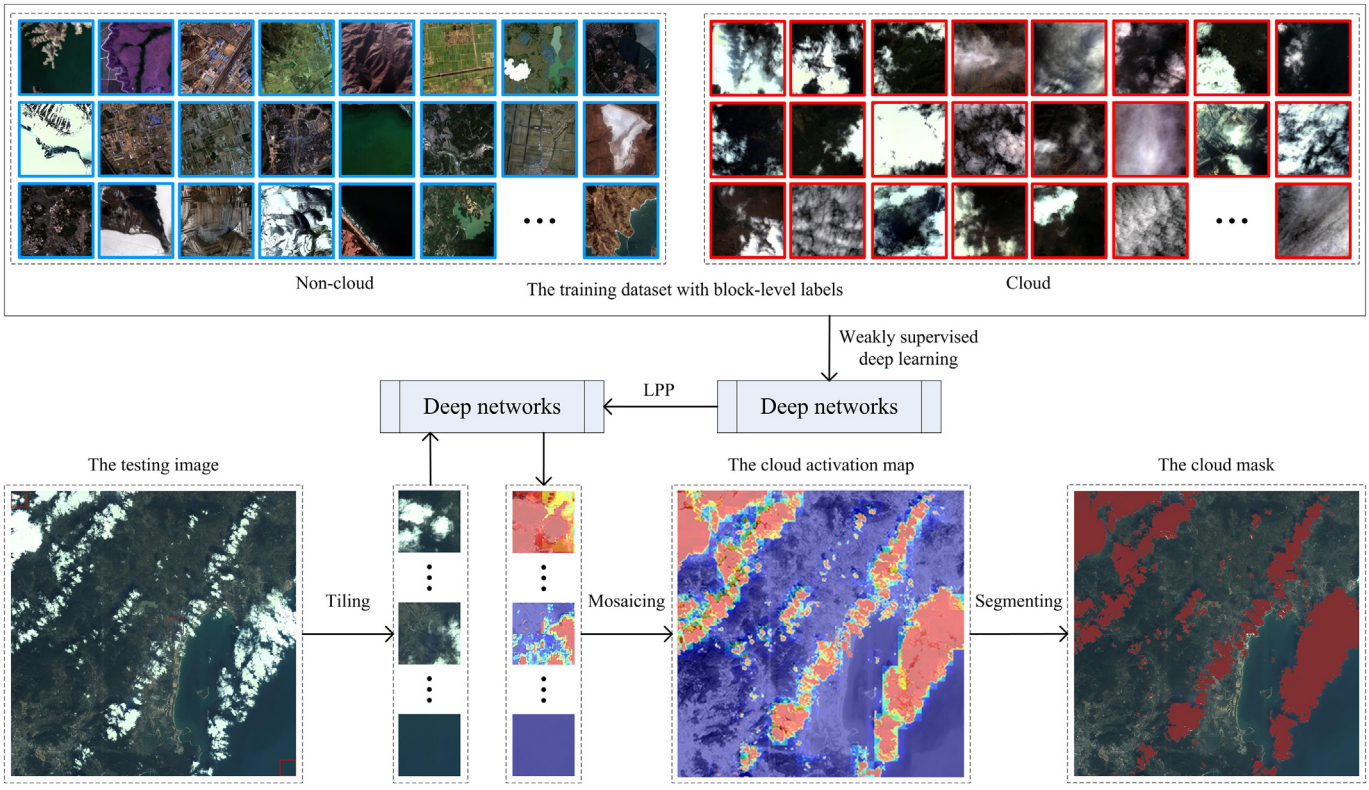

Fig. 10. The flowchart of our proposed WDCD approach.

Table 3
The architecture of our proposed deep networks.

Layers	Size
Conv1 + ReLU	$3\times3\times4\times64$
Conv2 + ReLU	$3\times3\times64\times64$
Local Pooling Layer (Pruned in the testing phase)	2×2
Conv3 + ReLU	$3\times3\times64\times128$
Conv4 + ReLU	$3\times3\times128\times128$
Local Pooling Layer (Pruned in the testing phase)	2×2
Conv5 + ReLU	$3\times3\times128\times256$
Conv6 + ReLU	$3\times3\times256\times256$
Conv7 + ReLU	$3\times3\times256\times256$
Local Pooling Layer (Pruned in the testing phase)	2×2
Conv8 + ReLU	$3\times3\times256\times512$
Conv9 + ReLU	$3\times3\times512\times512$
Conv10 + ReLU	$3\times3\times512\times1024$
Global convolutional pooling layer	20×20
Fully connected layer	1024×2

(Wang et al., 2017), which calculates the similarity between the estimated map and the ground truth map. Section 5.2.1 introduces the evaluation metrics, Section 5.2.2 quantitatively explains the characteristic of LPP, and Section 5.2.3 gives the quantitative comparison result with some competitive baselines.

5.2.1. Evaluation measures

By segmenting the CAM at different thresholds, we calculate the pixel-level evaluation measures including user's accuracy (UA), producer's accuracy (PA), true positive rate (TPR), and false positive rate (FPR) values by comparing the segmented CAM with the ground truth maps mentioned in Section 3.2. Additionally, the UA-PA curves (Wang et al., 2017) and the TPR-FPR curves (Hanley, 1989) are taken to evaluate the cloud detection performance of the CAM. These metrics are calculated by:

$$\mathrm{UA} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{9}$$

$$\mathrm{PA} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{10}$$

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{11}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{12}$$

where TP denotes the number of pixels whose ground truths (GT) and predictions are both positive, that is, categorized as cloud. TN stands for the number of pixels whose GTs and predictions are both negative (i.e., categorized as non-cloud). FN denotes the number of pixels whose GTs are positive while predictions are negative. Finally, FP indicates the number of pixels whose GTs are negative while predictions are positive.

5.2.2. Analysis of the local pooling pruning strategy

We specifically verify the effects of our proposed LPP operation by modifying several baselines of cloud detection with the LPP operation and comparing the cloud detection performance of the pairs such as CAM with GAP (Zhou et al., 2016a) and CAM with GAP + LPP, CAM with TSL (Li et al., 2018a, 2018b, 2018c) and CAM with TSL + LPP, our proposed CAM with GCP, and our proposed CAM with GCP + LPP. As depicted in Fig. 11, the results of methods with LPP achieve considerable improvements over those without LPP.

As aforementioned, in CAM with GCP + LPP, the local pooling layers are kept in the training phase but pruned in the testing phase. Furthermore, we verify the performance of our proposed deep learning model without local pooling layers in the training phase. In other words, this is a thorough solution of LPP where local pooling layers are pruned in both training and testing phases, termed as CAM with GCP + LPP*. Benefiting from the larger size of the GCP layer, which increases from 20×20 to 230×230, CAM with GCP + LPP* naturally owns better capability in exploiting the spatial variance than CAM with

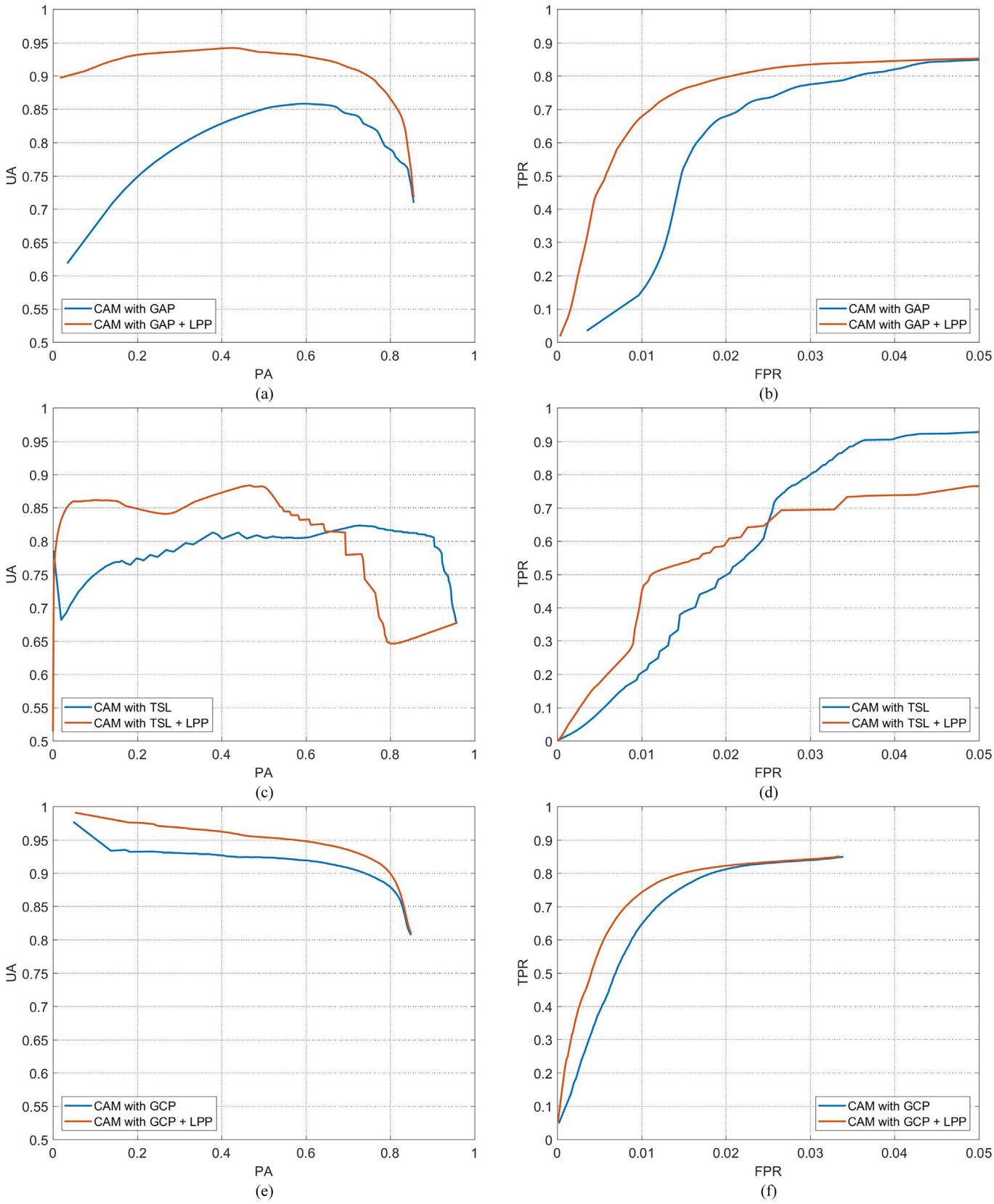

Fig. 11. The quantitative curves of different CAM generators with LPP or without it. (a), (c) and (e) show the UA-PA curves of CAM generators with LPP or without LPP. (b), (d) and (f) show the TPR-FPR curves of CAM generators with LPP or without LPP.

GCP + LPP. It can be seen from Fig. 12 that CAM with GCP + LPP* performs better than CAM with GCP + LPP. However, since the local pooling layers are pruned in the training phase, the intermediate feature maps in deep networks are much larger than they used to be. That costs much more memory occupation in the training phases. Furthermore, the training time of CAM with GCP + LPP* is almost 10

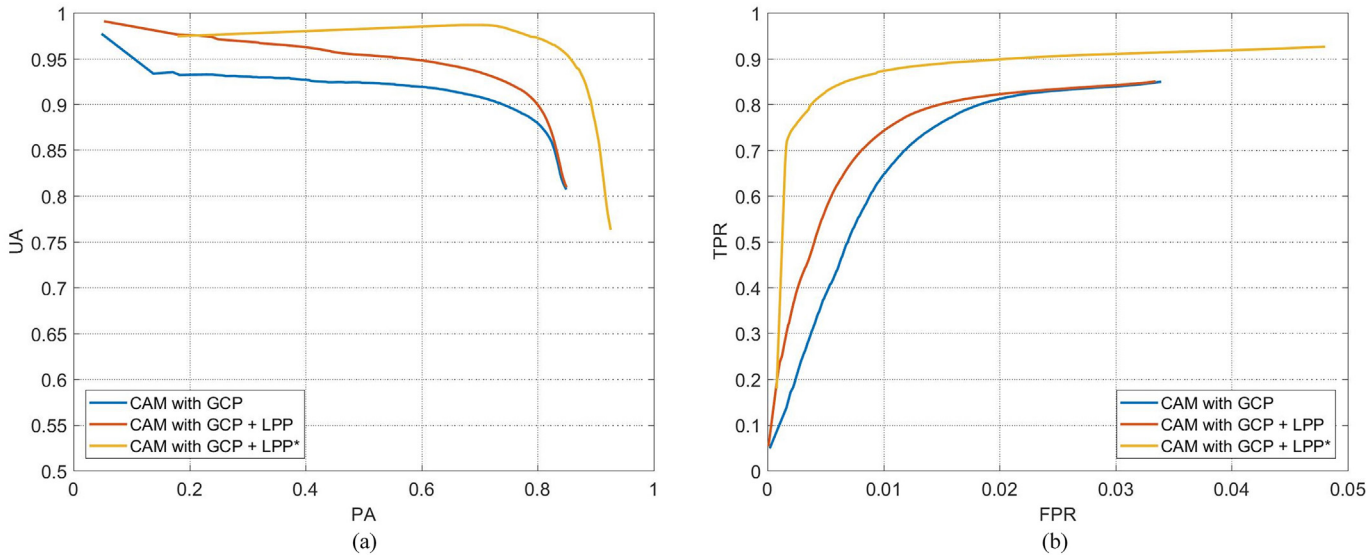

Fig. 12. The quantitative curves of GCP variants. (a), (c) and (e) show the UA-PA curves. (b), (d) and (f) depict the TPR-FPR curves.

times more than that of CAM with GCP + LPP. Actually, in the testing phase, CAM with GCP + LPP* also costs much more time than CAM with GCP + LPP*, which will be specifically discussed in the following section.

As CAM with GCP and CAM with GCP + LPP adopt the same training process, their running time equals each other. In addition, as shown in Table 4, CAM with GCP + LPP* is the most time-consuming. As a whole, CAM with GCP + LPP is recommended when the computational efficiency is a critical indicator. Of course, one can apply the CAM with GCP + LPP* to pursue better performance when the running time is allowed in the specific task.

5.2.3. Comparison with the state-of-the-art methods

To verify the superiority of our proposed method, we compare our proposed method with the state-of-the-art methods. More specifically, the baselines include the recently proposed weakly supervised deep learning-based cloud detection method (Zou et al., 2019). In addition, we also reimplement several recent object detection methods based on weakly supervised deep learning in the computer vision and RS domains as the baselines.

More specifically, the Generative Adversarial Training for Weakly Supervised Cloud Matting (GCM) (Zou et al., 2019) formulates cloud detection as a mixed energy separation process between foreground and background images. Their model consists of three networks, a cloud generator G, a cloud discriminator D, and a cloud matting network F, where G and D aim to generate realistic and physically meaningful cloud images by adversarial training, and F learns to predict the cloud reflectance and attenuation. The predicted cloud reflectance is used to compute the CAM. The network structure of CAM with GAP (Zhou et al., 2016a, 2016b) is quite similar to the VGG-16 net (Simonyan and Zisserman, 2014), while CAM with GAP replaces the fully connected layer with a global average pooling layer. After the training stage, CAM with GAP utilizes the activation weights to combine the feature map and to generate the CAM, which is the same with our proposed method. Li et al. (2018a, 2018b, 2018c) designs a two-stage-learning (TSL) method called CAM with TSL based on CAM with GAP, which trains the convolutional weights and the cloud activation weights in two different stages and then computes the CAM in the same way as CAM with GAP. In addition, we report the results of our proposed WDCD method under three variants including CAM with GCP, CAM with GCP + LPP, and CAM with GCP + LPP*.

It is worth noting that all the methods including the aforementioned baselines and our proposed CAM with GCP + LPP receive the input of image blocks cropped from the large image and compute the CAM of each image block separately. Finally, the CAM of the whole large image is generated via the sliding window approach. Fig. 13 shows the visualized results of the CAMs. As illustrated in Fig. 13, our proposed CAM with GCP can intuitively outperform the baselines. Without any further training cost, our proposed CAM with GCP + LPP can obtain further improved results. In addition, our proposed CAM with GCP + LPP* can obtain the best prediction performance.

In Fig. 14, we report the UA-PA curves and TPR-FPR curves of our proposed method and the baselines. As shown in Fig. 14, our proposed CAM outperforms the baselines by a large margin.

Table 4
Running time of methods with different training strategies.

Methods	GCP	GCP + LPP	GCP + LPP*
Running time (hour)	13.5	13.5	140.2

5.3. Performance evaluation on cloud mask detection

In the following, Section 5.3.1 introduces the metrics for evaluating binary cloud masks, and Section 5.3.2 summarizes the quantitative comparison result with the state-of-the-art methods.

5.3.1. Evaluation measures

Different from the pixel-level measures in Section 5.2, this section uses several comprehensive metrics including overall accuracy (OA), and F_1_score. These metrics are calculated by:

$$OA = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

$$F_1\_score = \frac{2 \times UA \times PA}{UA + PA} \tag{14}$$

The definitions of TP, TN, FP, FN and the calculations of UA and PA have been listed in Section 5.2.1.

5.3.2. Comparison with the state-of-the-art methods

In this section, we further evaluate the quality of binary cloud detection masks, which are generated by the following methods: GCM (Zou et al., 2019), progressive refinement scheme (PRS) (Zhang and Xiao, 2014), classification and assignment (CAA) (Simonyan and Zisserman, 2014), CAM with GAP (Zhou et al., 2016a), CAM with TSL

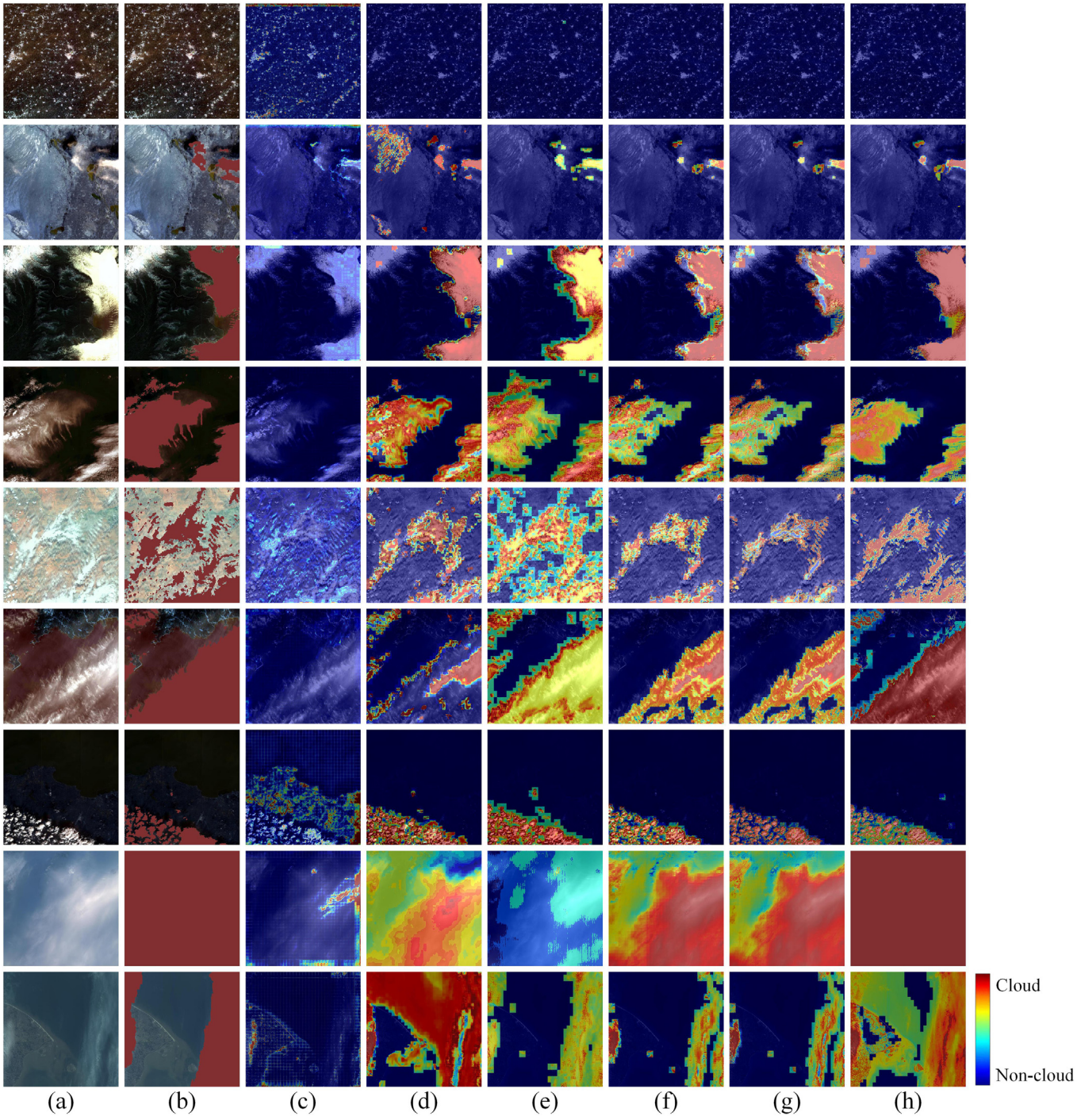

Fig. 13. The CAMs of testing images. (a) The original large testing images, (b) The corresponding ground truths of (a), (c) The GCM, (d) The CAMs with GAP, (e) The CAMs with TSL, (f) Our proposed CAMs with GCP, (g) Our proposed CAMs with GCP + LPP, (h) Our proposed CAMs with GCP + LPP*. (c) to (h) Depict the color scale from blue (low cloud probability) to red (high cloud probability). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(Li et al., 2018a), our proposed CAM with GCP, our proposed CAM with GCP + LPP, and our proposed CAM with GCP + LPP*.

More specifically, PRS (Zhang and Xiao, 2014) constructs a significance map, which highlights the difference between cloud regions and non-cloud regions. Based on the significance map and the proposed optimal threshold setting, it obtains a coarse cloud detection result, which classifies the input aerial photograph into the candidate cloud regions and non-cloud regions. To accurately detect the cloud regions from the candidate cloud regions, it then constructs a robust detail map derived from a multiscale bilateral decomposition to remove non-cloud regions from the candidate cloud regions. Finally, a guided feathering is performed to achieve the final cloud detection result, which detects semitransparent cloud pixels around the boundaries of cloud regions. CAA (Simonyan and Zisserman, 2014) uses DCNN to classify the image blocks as containing cloud or not. Taking the block as a basic unit, CAA can only predict the coarse cloud region, which has an obvious saw-tooth effect. Fig. 15 depicts the visualized results of the cloud masks by different methods. In addition, Table 5 reports the quantitative

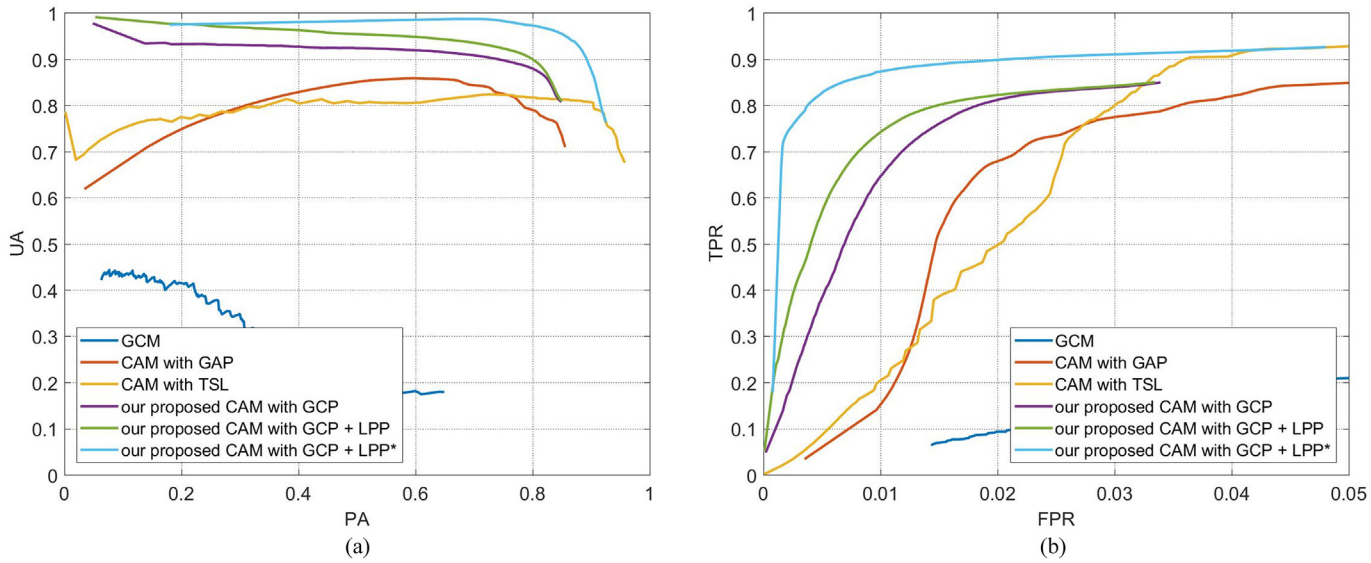

Fig. 14. The quantitative curves of our proposed method and the existing methods. (a) Shows the UA-PA curves. (b) Illustrates the TPR-FPR curves.

comparison results. As depicted in Table 5, our proposed WDCD method with several variants obviously outperforms the state-of-the-art methods. In addition, our proposed WDCD method based on CAM with GCP + LPP performs better than our proposed WDCD method based on CAM with GCP, which verifies the superiority of the presented LPP strategy. Our proposed WDCD method based on CAM with GCP + LPP* can achieve the best performance among all of the variants.

Furthermore, we also discuss the running time of various methods in the online stage (i.e., the testing phase). Table 6 reports the average running time of different methods in generating the cloud mask for one large multispectral image whose size is approximately 4548 × 4544. As shown in Table 6, the running time of our proposed CAM with GCP is slightly longer than that of PRS, CAA, CAM with GAP, and CAM with TSL and much shorter than that of GCM. As the local pooling layers are pruned, the computational complexity of our proposed CAM with GCP + LPP is naturally increased. As depicted in Table 6, our proposed CAM with GCP + LPP costs much more time than our proposed CAM with GCP, and our proposed CAM with GCP + LPP* costs much more time in generating cloud masks than other methods including CAM with GCP + LPP. Hence, researchers are suggested to select the appropriate solution from CAM with GCP + LPP and CAM with GCP + LPP* based on their specific demands.

6. Discussion

In the following, Section 6.1 provides the sensitivity analysis of the critical parameters based on the validation dataset used in our experiments. Section 6.2 shows the limitations of the work in this paper and gives suggestions on how to further improve the method in the future.

6.1. Sensitivity analysis of critical parameters

In this section, we specifically analyzed the sensitivity of the critical parameters based on the validation dataset. The critical parameters include the depth d of the global convolutional pooling (GCP) layer and the segmentation threshold parameter k.

We first trained the deep networks with d set to 512. To save computational time, when evaluating the performance of deep network under other d, we transfer the deep networks with d set to 512 and finetune the deep networks with the new depth of the last convolutional layer. Specifically, with other layers fixed, we finetune the last convolutional layer with the new depth, GCP layer and the fully connected layer. With the depth of the last convolutional layer d (whose depth should be the same as the GCP layer) set to 256, 1024 and 2048, we calculate the CAMs via different deep networks with different d. Furthermore, based on CAMs on the validation dataset, we calculate the UA-PA curves (Wang et al., 2017) and the TPR-FPR curves (Hanley, 1989). As depicted in Fig. 16, our proposed method is not very sensitive to d, and the depth of 1024 (i.e., the yellow line) shows a little superiority to the others. Hence, d is empirically set to 1024 in our implementation.

With the depth d of GCP layer fixed to 1024, we further analyze the sensitivity of the threshold parameter k. To comprehensively measure the performances of cloud masks under different values of k, we choose metrics for classification including OA and F_1_score. The average values of these evaluation metrics on the validation dataset are utilized to analyze the cloud detection performance under different k. As illustrated in Table 7, the best performance is obtained when $k = 0.6$. It is noted that the performance would be improved if we further tune the parameters d and k. We do not do so because the process is quite time-consuming. It can be evaluated in future work when more computational resources are available.

6.2. Limitations and future perspectives

Since the distributions of shadows are often around the cloud boundary, it is very hard to find an image block covered with shadows only, which results in a lack of training samples and an inability to detect shadows. Nevertheless, based on the cloud detection results, the existing cloud and shadow detection method (Li et al., 2017) can be used to solve the shadow detection problem. As shown in our experiments, it is worth noting that the existence of shadow does not influence the cloud detection performance of our proposed WDCD method.

Although our proposed WDCD method only requires block-level binary labels to address cloud detection, the performance of the method highly depends on the classification accuracy of the backbone deep networks in the training phase. Moreover, the ability of the feature map to represent useful information also matters significantly. In fact, compared with the VGG-16 net (Simonyan and Zisserman, 2014) used in our implementation, more advanced backbone deep networks have been proposed. For example, ResNet (He et al., 2016), which applies a deep residual network with a greatly increased depth to easily achieve higher accuracy than previous networks; and DenseNet (Huang et al., 2017), which connects each layer to every other layer in a feed-forward fashion to strengthen feature propagation and encourage feature reuse. Such backbone deep networks possess better abilities in terms of feature

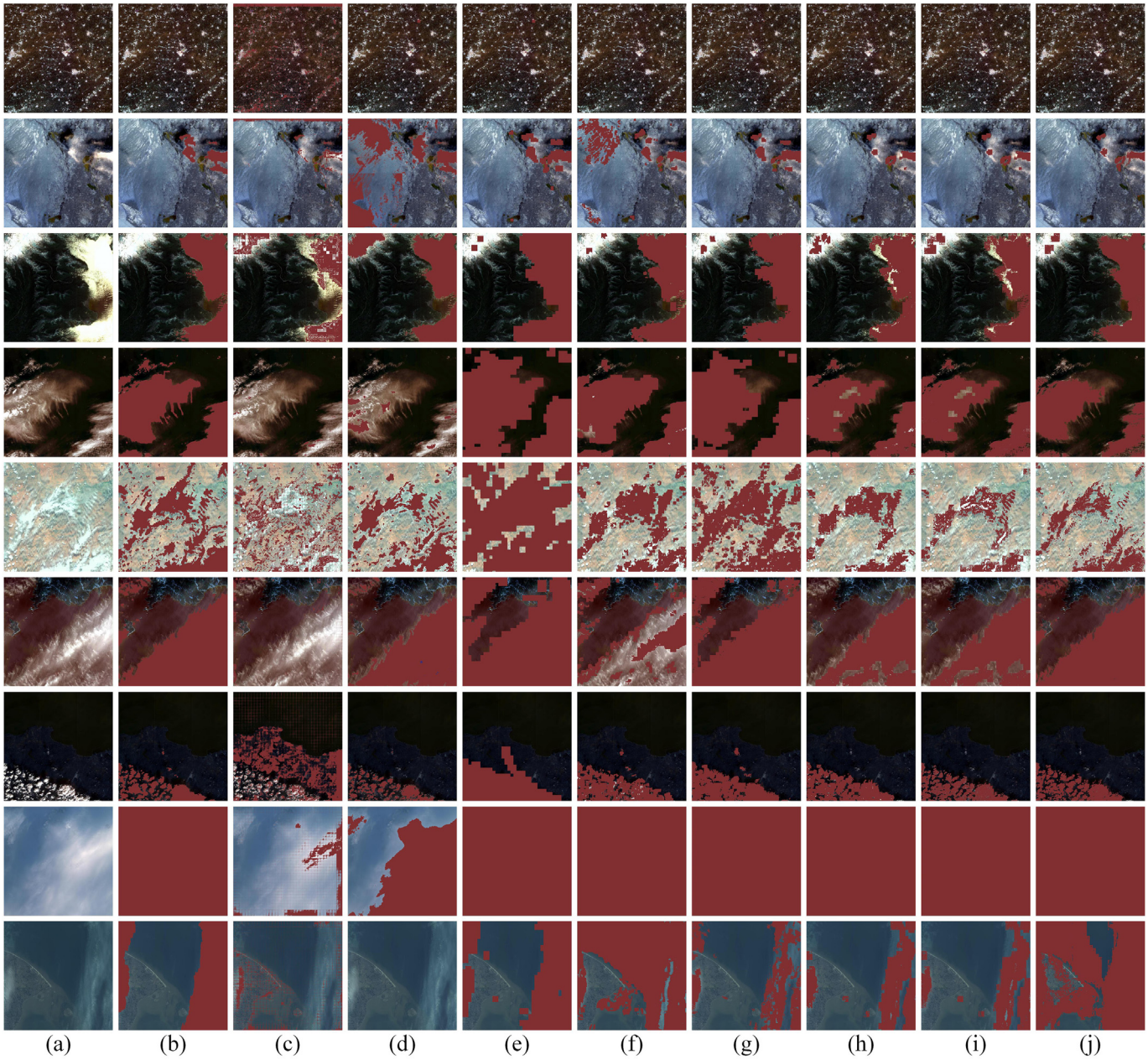

Fig. 15. The cloud masks of testing images where the red color regions stand for the cloud masks. (a) The original large testing images, (b) the corresponding GT of (a), (c) the cloud masks of GCM, (d) the cloud masks of PRS, (e) the cloud masks of CAA, (f) the cloud masks of CAM with GAP, (g) the cloud masks of CAM with TSL, (h) the cloud masks of our proposed WDCD method based on CAM with GCP, (i) the cloud masks of our proposed WDCD method based on CAM with GCP + LPP, (j) the cloud masks of our proposed WDCD method based on CAM with GCP + LPP*. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5
Comparison results between our proposed method and the existing methods.

Methods	OA	F_1_score
GCM (Zou et al., 2019)	0.7906	0.3267
PRS (Zhang and Xiao, 2014)	0.8789	0.6137
CAA (Simonyan and Zisserman, 2014)	0.9172	0.7701
CAM with GAP (Zhou et al., 2016a)	0.9410	0.7961
CAM with TSL (Li et al., 2018a)	0.9335	0.8034
Our proposed WDCD method based on CAM with GCP	0.9569	0.8421
Our proposed WDCD method based on CAM with GCP + LPP	0.9596	0.8504
Our proposed WDCD method based on CAM with GCP + LPP*	**0.9666**	**0.8855**

Table 6
The average running time of methods in generating the cloud mask.

Methods	Running time (Second)
GCM (Zou et al., 2019)	32.6
PRS (Zhang and Xiao, 2014)	22.1
CAA (Simonyan and Zisserman, 2014)	24.2
CAM with GAP (Zhou et al., 2016a)	24.4
CAM with TSL (Li et al., 2018a)	24.3
Our proposed WDCD method based on CAM with GCP	25.7
Our proposed WDCD method based on CAM with GCP + LPP	91.2
Our proposed WDCD method based on CAM with GCP + LPP*	145.0

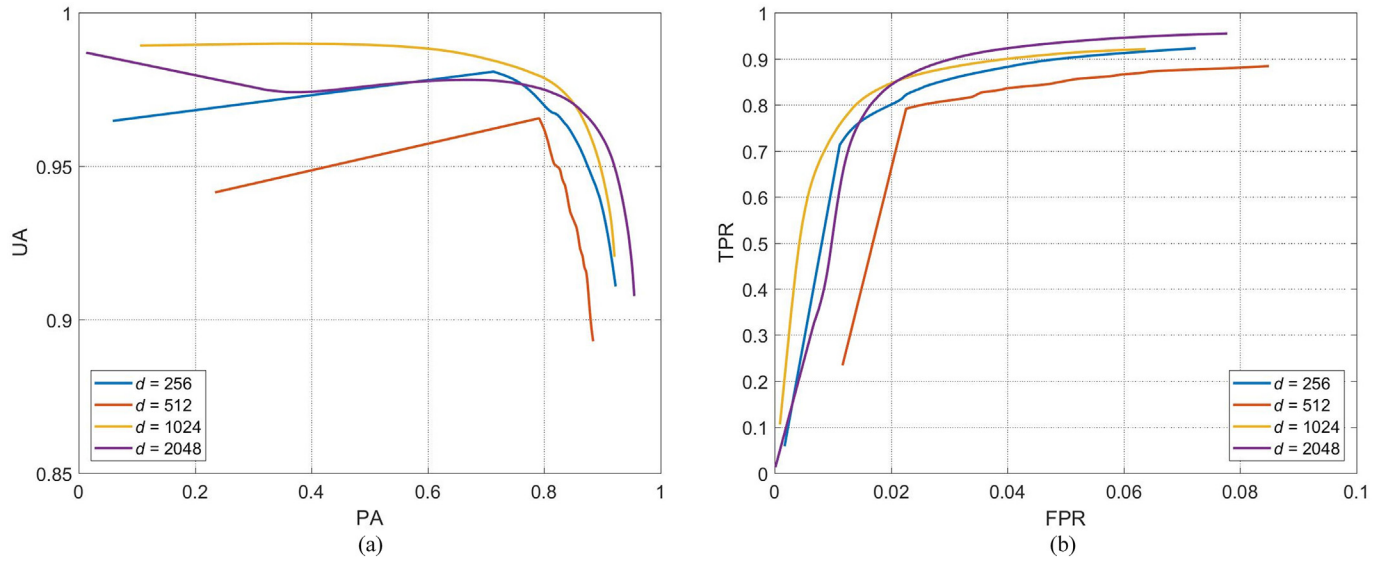

Fig. 16. The quantitative curves of our proposed method under different d (i.e., depths of GCP). (a) Shows the UA-PA curves and (b) illustrates the TPR-FPR curves.

Table 7
Performance of our proposed WDCD method under different k.

	$k = 0.2$	$k = 0.4$	$k = 0.6$	$k = 0.8$	$k = 1.0$
OA	0.9327	0.9331	**0.9344**	0.9333	0.9325
F1_score	0.9219	0.9223	**0.9234**	0.9224	0.9216

representation and may be employed to enhance the performance of our proposed WDCD method in future work. Additionally, it is worthwhile to further boost cloud detection performance under different conditions (e.g., different cloud types and different underlying terrains).

7. Conclusion

This paper proposes a new learning framework that can train deep networks with only block-level binary labels, which indicates whether the image block contains cloud or not, and the trained deep networks can detect the pixel-level cloud mask. To improve the ability of the feature map to represent the spatial context and textural and semantic information, we propose a new global pooling operation called GCP, which can learn the channel-independent convolutional weights of each channel of the feature map. After the iterative backward propagations, the feature map possesses the ability to represent the region of the cloud, which is used to compute the CAM. Furthermore, we propose the LPP to improve the quality and spatial resolution of the feature map, which is used to compute the CAM. After adaptively segmenting the CAM, the pixel-level cloud mask is obtained. Even under this extremely weak supervision, the proposed WDCD approach achieves promising cloud detection results and outperforms the state-of-the-art approaches. We released a new cloud detection dataset, which may benefit the rapid advance of the cloud detection direction. As a whole, the cloud detection results can be utilized in many tasks such as shadow detection (Li et al., 2017) and cloud removal (Schmitt et al., 2019), and further support continuous cartography and wide-range environmental evaluation. In future work, we will exploit more advanced deep network architectures to improve the performance of our proposed WDCD method under different conditions and explore the joint detection of cloud and shadow.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Key Research and Development Program of China under grant 2018YFB0505003; the National Natural Science Foundation of China under grant 41971284; the China Postdoctoral Science Foundation under grant 2016M590716 and 2017T100581; the Hubei Provincial Natural Science Foundation of China under grant 2018CFB501; and the Fundamental Research Funds for the Central Universities under grant 2042020kf0218.

References

Bilen, H., Vedaldi, A., 2016. Weakly supervised deep detection networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2846–2854.
Chai, D., Newsam, S., Zhang, H.K., Qiu, Y., Huang, J., 2019. Cloud and cloud shadow detection in Landsat imagery based on deep convolutional neural networks. Remote Sens. Environ. 225, 307–316.
Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A., 2018. DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. 40, 834–848.
Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1251–1258.
Cinbis, R., Verbeek, J., Schmid, C., 2017. Weakly supervised object localization with multi-fold multiple instance learning. IEEE Trans. Pattern Anal. Mach. Intell. 39, 189–203.
Francis, A., Sidiropoulos, P., Muller, J.P., 2019. CloudFCN: accurate and robust cloud detection for satellite imagery with deep learning. Remote Sens. 11 (19), 2312.
Gao, M., Li, A., Yu, R., Morariu, V.I., Davis, L.S., 2018. C-wsl: Count-guided weakly supervised localization. In: Proceedings of the European Conference on Computer Vision, pp. 152–168.
Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT press.
Hanley, J.A., 1989. Receiver operating characteristic (ROC) methodology: the state of the art. Crit. Rev. Diagn. Imaging 29 (3), 307–335.
He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 770–778.
Hollstein, A., Segl, K., Guanter, L., Brell, M., Enesco, M., 2016. Ready-to-use methods for the detection of clouds, cirrus, snow, shadow, water and clear sky pixels in Sentinel-2 MSI images. Remote Sens. 8 (8), 666.
Hsu, K.J., Lin, Y.Y., Chuang, Y.Y., 2019. Weakly supervised salient object detection by learning a classifier-driven map generator. IEEE Trans. Image Process. 28, 5435–5449.
Huang, C., Thomas, N., Goward, S.N., Masek, J.G., Zhu, Z., Townshend, J.R., Vogelmann,

J.E., 2010. Automated masking of cloud and cloud shadow for forest change analysis using Landsat images. Int. J. Remote Sens. 31 (20), 5449–5464.
Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708.
Ishida, H., Oishi, Y., Morita, K., Moriwaki, K., Nakajima, T.Y., 2018. Development of a support vector machine based cloud detection method for MODIS with the adjustability to various conditions. Remote Sens. Environ. 205, 390–407.
Jeppesen, J.H., Jacobsen, R.H., Inceoglu, F., Toftegaard, T.S., 2019. A cloud detection algorithm for satellite imagery based on deep learning. Remote Sens. Environ. 229, 247–259.
King, M.D., Platnick, S., Menzel, W.P., Ackerman, S.A., Hubanks, P.A., 2013. Spatial and temporal distribution of clouds observed by MODIS onboard the Terra and Aqua satellites. IEEE Trans. Geosci. Remote Sens. 51 (7), 3826–3852.
Kingma, D.P., Ba, J.L., 2015. Adam: A method for stochastic optimization. In: Proceedings of International Conference on Learning Representations, pp. 1–13.
Kolesnikow, A., Lampert, C., 2016. Seed, expand and constrain: three principles for weakly-supervised image segmentation. In: Proceedings of the 14th European Conference on Computer Vision. Springer, pp. 695–711.
Krizhevsky, A., Sutskever, I., Hinton, G., 2012. Imagenet classification with deep convolutional neural networks. In: Proceedings of the 26th Annual Conference on Neural Information Processing Systems, pp. 1097–1105.
LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature. 521, 436–444.
Li, Z., Shen, H., Li, H., Xia, G., Gamba, P., Zhang, L., 2017. Multi-feature combined cloud and cloud shadow detection in GaoFen-1 wide field of view imagery. Remote Sens. Environ. 191, 342–358.
Li, Y., Zhang, Y., Huang, X., Yuille, A.L., 2018a. Deep networks under scene-level supervision for multi-class geospatial object detection from remote sensing images. ISPRS J. Photogramm. Remote Sens. 146, 182–196.
Li, Y., Zhang, Y., Huang, X., Zhu, H., Ma, J., 2018b. Large-scale remote sensing image retrieval by deep hashing neural networks. IEEE Trans. Geosci. Remote Sens. 56 (2), 950–965.
Li, Y., Zhang, Y., Huang, X., Ma, J., 2018c. Learning source-invariant deep hashing convolutional neural networks for cross-source remote sensing image retrieval. IEEE Trans. Geosci. Remote Sens. 56 (11), 6521–6536.
Li, Y., Zhang, Y., Zhu, Z., 2020. Error-tolerant deep learning for remote sensing image scene classification. In: IEEE Transactions on Cybernetics, (in press).
Mohajerani, S., Saeedi, P., 2019. Cloud-net: an end-to-end cloud detection algorithm for Landsat 8 imagery. arXiv arXiv: 1901.10077.
Oishi, Y., Ishida, H., Nakamura, R., 2018. A new Landsat 8 cloud discrimination algorithm using thresholding tests. Int. J. Remote Sens. 39, 9113–9133.
Pathak, D., Krahenbuhl, P., Darrell, T., 2015. Constrained convolutional neural networks for weakly supervised segmentation. In: Proceedings of IEEE International Conference on Computer Vision, pp. 1796–1804.
Pinheiro, P., Collobert, R., 2015. From image-level to pixel-level labeling with convolutional networks. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1713–1721.
Qiu, S., He, B., Zhu, Z., Liao, Z., Quan, X., 2017. Improving Fmask cloud and cloud shadow detection in mountainous area for Landsats 4–8 images. Remote Sens. Environ. 199, 107–119.
Qiu, S., Zhu, Z., He, B., 2019. Fmask 4.0: improved cloud and cloud shadow detection in landsats 4-8 and Sentinel-2 imagery. Remote Sens. Environ. 231, 111205.
Ruderman, A., Rabinowitz, N.C., Morcos, A.S., 2018. Pooling is neither necessary nor sufficient for appropriate deformation stability in CNNs. arXiv arXiv: 1804.04438.
Schmitt, M., Hughes, L., Qiu, C., Zhu, X., 2019. Aggregating cloud-free Sentinel-2 images with Google earth engine. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci. 145–152.
Segal-Rozenhaimer, M., Li, A., Das, K., Chirayath, V., 2020. Cloud detection algorithm for multi-modal satellite imagery using convolutional neural-networks (CNN). Remote Sens. Environ. 237, 111446.
Shan, N., Zheng, T.Y., Wang, Z.S., 2009. Onboard real-time cloud detection using reconfigurable FPGAs for remote sensing. In: Proceedings of International Conference on Geoinformatics, pp. 1–5.
Shao, Z., Deng, J., Wang, L., Fan, Y., Sumari, N., Cheng, Q., 2017. Fuzzy autoencode based cloud detection for remote sensing imagery. Remote Sens. 9, 311.
Shao, Z., Pan, Y., Diao, C., Cai, J., 2019. Cloud detection in remote sensing images based on multiscale features-convolutional neural network. IEEE Trans. Geosci. Remote Sens. 57, 4062–4076.
Simonyan, K., Zisserman, A., 2014. Very Deep Convolutional Networks for Large-scale Image Recognition. arXiv preprint. arXiv. arXiv: 1409.1556.
Singh, K.K., Lee, Y.J., 2019. You reap what you sow: using videos to generate high precision object proposals for weakly-supervised object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9414–9422.
Tan, Y., Qi, J., Ren, F., 2016. Real-time cloud detection in high resolution images using maximum response filter and principle component analysis. In: Proceedings of IEEE International Geoscience and Remote Sensing Symposium, pp. 6537–6540.
Tan, Y., Xiong, S., Li, Y., 2018. Automatic extractionof built-up areas from panchromatic and multispectral remote sensing images using double-stream deep convolutional neural networks. IEEE J. Select. Top. Appl. Earth Observ. Remote Sensing. 11 (11), 3988–4004.
Tang, P., Wang, X., Huang, Z., Bai, X., Liu, W., 2017. Deep patch learning for weakly supervised object classification and discovery. Pattern Recogn. 71, 446–459.
Tang, M., Djelouah, A., Perazzi, F., Boykov, Y., Schroers, C., 2018a. Normalized cut loss for weakly-supervised cnn segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1818–1827.
Tang, P., Wang, X., Wang, A., Yan, Y., Liu, W., Huang, J., Yuille, A., 2018b. Weakly supervised region proposal network and object detection. In: Proceedings of the European Conference on Computer Vision, pp. 352–368.
Tao, C., Mi, L., Li, Y., Qi, J., Xiao, Y., Zhang, J., 2019a. Scene context-driven vehicle detection in high-resolution aerial images. IEEE Trans. Geosci. Remote Sensing. 57 (10), 7339–7351.
Tao, C., Qi, J., Li, Y., Wang, H., Li, H., 2019b. Spatial information inference net: road extraction using road-specific contextual information. ISPRS J. Photogramm. Remote Sens. 158, 155–166.
Wan, F., Wei, P., Jiao, J., Han, Z., Ye, Q., 2018. Min-entropy latent model for weakly supervised object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1297–1306.
Wang, L., Lu, H., Wang, Y., Feng, M., 2017. Learning to detect salient objects with image-level supervision. In: Proceedings of the 2017 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 136–145.
Wang, J., Liu, C., Yao, B., Min, M., Letu, H., Yin, Y., Yung, Y.L., 2019a. A multilayer cloud detection algorithm for the Suomi-NPP visible infrared imager radiometer suite (VIIRS). Remote Sens. Environ. 227, 1–11.
Wang, L., Li, Q., Zhou, Y., 2019b. Multiple-instance discriminant analysis for weakly supervised segment annotation. IEEE Trans. Image Process. 28, 5716–5728.
Wei, Y., Liang, X., Chen, Y., Shen, X., Cheng, M., Feng, J., Zhao, Y., Yan, S., 2016. Stc: a simple to complex framework for weakly-supervised semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39, 2314–2320.
Wei, Y., Xiao, H., Shi, H., Jie, Z., Feng, J., Huang, T.S., 2018. Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7268–7277.
Wieland, M., Li, Y., Martinis, S., 2019. Multi-sensor cloud and cloud shadow segmentation with a convolutional neural network. Remote Sens. Environ. 230, 111203.
Wilson, M.J., Oreopoulos, L., 2013. Enhancing a simple MODIS cloud mask algorithm for the Landsat data continuity mission. IEEE Trans. Geosci. Remote Sens. 51, 723–731.
Xu, M., Jia, X., Pickering, M., Jia, S., 2019. Thin cloud removal from optical remote sensing images using the noise-adjusted principal components transform. ISPRS J. Photogramm. Remote Sens. 149, 215–225.
Yang, Z., Mahajan, D., Ghadiyaram, D., Nevatia, R., Ramanathan, V., 2019. Activity driven weakly supervised object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2917–2926.
Zhang, Q., Xiao, C., 2014. Cloud detection of RGB color aerial photographs by progressive refinement scheme. IEEE Trans. Geosci. Remote Sens. 52 (11), 7264–7275.
Zhang, Y., Wen, F., Gao, Z., Ling, X., 2019. A coarse-to-fine framework for cloud removal in remote sensing image sequence. IEEE Trans. Geosci. Remote Sens. 57, 5963–5974.
Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A., 2014. Object detectors emerge in deep scene cnns. arXiv. arXiv: 1412.6856.
Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A., 2016a. Learning deep features for discriminative localization. In: Proceedings of the 2016 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2921–2929.
Zhou, G., Zhou, X., Yue, T., Liu, Y., 2016b. An optional threshold with SVM cloud detection algorithm and DSP implementation. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch. 41, 771–777.
Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A., 2018. Places: a 10 million image database for scene recognition. IEEE Trans. Pattern Anal. Mach. Intell. 40, 1452–1464.
Zhu, Z., Wang, S., Woodcock, C.E., 2015. Improvement and expansion of the Fmask algorithm: cloud, cloud shadow, and snow detection for Landsats 4–7, 8, and sentinel 2 images. Remote Sens. Environ. 159, 269–277.
Zou, Z., Li, W., Shi, T., Shi, Z., Ye, J., 2019. Generative adversarial training for weakly supervised cloud matting. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 201–210.