ISPRS Journal of Photogrammetry and Remote Sensing 209 (2024) 66–84

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

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

Remote sensing image cloud detection using a shallow convolutional neural network

Dengfeng Chai [a,*], Jingfeng Huang [b,c], Minghui Wu [d], Xiaoping Yang [a], Ruisheng Wang [e]

[a] Key Laboratory of Geoscience Big Data and Deep Resource of Zhejiang Province, School of Earth Sciences, Zhejiang University, School of Earth Sciences, Zhejiang University, Hangzhou, 310027, China
[b] Institute of Applied Remote Sensing and Information Technology, Zhejiang University, Hangzhou 310058, China
[c] Key Laboratory of Agricultural Remote Sensing and Information Systems, Zhejiang University, Hangzhou 310058, China
[d] School of Computer and Computing Science, Zhejiang University City College, Hangzhou, 310015, China
[e] Department of Geomatics Engineering, University of Calgary, Calgary, Canada

ARTICLE INFO

Keywords:
Cloud detection
Semantic segmentation
Shallow convolutional neural network (SCNN)
Landsat
Gaofen

ABSTRACT

The state-of-the-art methods for cloud detection are dominated by deep convolutional neural networks (DCNNs). However, it is very expensive to train DCNNs for cloud detection and the trained DCNNs do not always perform well as expected. This paper proposes a shallow CNN (SCNN) by removing pooling/unpooling layers and normalization layers in DCNNs, retaining only three convolutional layers, and equipping them with 3 filters of $1 \times 1, 1 \times 1, 3 \times 3$ in spatial dimensions. It demonstrates that the three convolutional layers are sufficient for cloud detection. Since the label output by the SCNN for a pixel depends on a 3×3 patch around this pixel, the SCNN can be trained using some thousands 3×3 patches together with ground truth of their center pixels. It is very cheap to train a SCNN using some thousands 3×3 patches and to provide ground truth of their center pixels. Despite of its low cost, SCNN training is stabler than DCNN training, and the trained SCNN outperforms the representative state-of-the-art DCNNs for cloud detection. The same resolution of original image, feature maps and final label map assures that details are not lost as by pooling/unpooling in DCNNs. The border artifacts suffering from deep convolutional and pooling/unpooling layers are minimized by 3 convolutional layers with $1\times1, 1\times1, 3\times3$ filters. Incoherent patches suffering from patch-by-patch segmentation and batch normalization are eliminated by SCNN without normalization layers. Extensive experiments based on the L7 Irish, L8 Biome and GF1 WHU datasets are carried out to evaluate the proposed method and compare with state-of-the-art methods. The proposed SCNN promises to deal with images from any other sensors. The code and data will be open for further research. [1]

1. Introduction

Cloud detection is an important topic in remote sensing image analysis (Li et al., 2022a). The annual mean cloud cover reaches 40% in Landsat images (Ju and Roy, 2008). The cloudy pixels do not support terrestrial remote sensing as they are contaminated by clouds. To exclude the cloudy pixels from various terrestrial remote sensing, it is necessary to distinguish them in advance. For example, a pixel-wise cloud mask distinguishing cloudy pixels from clear ones is released in the Quality Assessment band of Landsat Level 1 product.

The pixel-wise cloud mask available in Landsat Level 1 product is generated by FMask algorithm (Zhu and Woodcock, 2012), which distinguishes each cloudy pixel by checking its radiance from thermal bands, reflectance from reflective bands and some derived indexes. Since FMask algorithm is an unsupervised algorithm, it cannot learn image features from training data to classify cloudy and clear pixels, and consequently impairs performance on such difficult scenes as snow scenes. Deep convolutional neural networks (DCNNs) have been introduced to learn distinguishing features for cloud detection, however, their practical applications in remote sensing image cloud detection are prevented by their costs and performances (Chai et al., 2019; Mateo-García et al., 2020; Skakun et al., 2022).

It consumes a lot of computation resources to train DCNNs and to segment remote sensing images using DCNNs. It takes some days to train a DCNN using 96 Landsat images distributed globally over the

* Corresponding author.
E-mail addresses: chaidf@zju.edu.cn (D. Chai), hjf@zju.edu.cn (J. Huang), mhwu@zucc.edu.cn (M. Wu), xpyang@zju.edu.cn (X. Yang), ruiswang@ucalgary.ca (R. Wang).

[1] https://github.com/dfchai/SCNN

https://doi.org/10.1016/j.isprsjprs.2024.01.026
Received 7 June 2022; Received in revised form 17 January 2024; Accepted 23 January 2024
Available online 6 February 2024

earth even though GPU is employed for parallel computation according to the efficiency reported in Chai et al. (2019). More importantly, it is expensive to annotate enough ground truths to train a DCNN since tens or hundreds images with pixel-wise ground truths are required by DCNN training. This issue is addressed by three alternative solutions. First, cloud masks resulting from the Fmask algorithm are chosen to serve as ground truths for Landsat images (Jeppesen et al., 2019). Since the cloud masks from Fmask are not as precise as those by manual annotation, the model trained using the former is significantly outperformed by the one trained using the latter. Second, a weak annotation is employed to provide ground truth. For example, Li et al. 2020 label each patch instead of each pixel as clear or cloudy. However, their annotation is still laborious as they annotated 200,000 image patches to train a deep cloud detection model (Li et al., 2020). Third, a DCNN for images captured by a target sensor, e.g., Sentinel-2, is adapted from a DCNN trained using training samples captured by a different source sensor, e.g., World-View-2 (Segal-Rozenhaimer et al., 2020). However, the ground truths of World-View-2 images are still required and the adapted DCNN for Sentinel-2 images does not perform as well as the DCNN for World-View-2 images.

The high costs of ground truth annotation and computation consumption do not assure DCNNs good performances. Although some DCNNs perform well on some public datasets, no one has been delivered to practical applications as these datasets are very small with respect to all remote sensing images (López-Puigdollers et al., 2021; Li et al., 2022a). As illustrated in Fig. 1, a remote sensing image is partitioned into a set of patches and segmented patch by patch due to memory limitation. Due to batch normalization (Ioffe and Szegedy, 2015; Badrinarayanan et al., 2017), a seamless and coherent label map for an entire remote sensing image is not guaranteed by patch composition. As illustrated by segmentation via a DCNN in Fig. 1, the feature maps are down-sampled progressively so that global features in a large area of input image can be extracted by a filter in the latter layers. However, some details lost in down-sampling cannot be totally recovered by up-sampling. Therefore, the quality of segmentation is impaired by down/up sampling (pooling/unpooling). Since global features are extracted from a large area, the label of each pixel depends on a large window centering at this pixel. Therefore, filling (black) pixels involve in global feature extraction from patches across image area and filling area, and usually produce artifacts at borders of image area.

To reduce computation consumption, some light-weight networks have been developed for cloud detection in the past two years (Yao et al., 2021; Li et al., 2022b). However, they do not alleviate image annotation as they are still trained using image patches with pixel-wise ground truths. On the other hand, multi-layer perception neural network (MLP) (Lee et al., 1990; Tian et al., 1999; Scaramuzza et al., 2012; Hughes and Hayes, 2014) is trained using a set of individual pixels. However, they require some handcrafted features of each sample (pixel) as input and extract the handcrafted features by some extractors independent of the MLP. Since the handcrafted features are not extracted by the MLP itself, the MLP training is not end-to-end.

This paper proposes a shallow convolutional neural network (SCNN) consisting three convolutional layers to transform an image to its label map. Remote sensing image cloud detection using a SCNN is very cheap and achieves good performances. SCNN training is supervised by a set of 3×3 patches and ground truth of their center pixels. Such supervision is much weaker than a set of images with pixel-wise ground truths. It consumes limited computation resources both in SCNN training and in image segmentation using SCNN. Cloud detection in remote sensing images using SCNN can be carried out using either GPU or CPU. The SCNN removes batch normalization layers, and guarantees seamless and coherent label map for patch-by-patch segmentation and patch composition. The SCNN removes pooling/unpooling layers, and maintains details in the whole procedure of segmentation. The SCNN adopts 3 filters of $1\times1, 1\times1, 3\times3$, and therefore reduce the border width to one pixel such that the border artifacts are minimized. Moreover, the proposed method promises to deal with images from Landsat 7, Landsat 8, Gaofen-1 and the other satellite.

The rest of this paper is organized as follows. Related work is reviewed in Section 2, cloud detection using a SCNN is described in Section 3, and it is compared with cloud detection using DCNNs in 4, experiments including evaluations and comparisons are presented in Section 5, the discussion and conclusion are presented in Sections 6 and 7, respectively.

2. Related work

Since cloud detection is achieved by semantic segmentation in this paper, cloud detection and semantic segmentation are briefly reviewed as follows.

2.1. Cloud detection

Overall, there are unsupervised and supervised methods for cloud detection in remote sensing image.

Unsupervised methods are based on either a single image or multiple images of the same scene captured at different time. The former is based on the expectation that the values of multiple (original and derived) bands of cloudy pixel fall in some specific ranges (Hollingsworth et al., 1996; Irish et al., 2006; Roy et al., 2010; Scaramuzza et al., 2012; Vermote et al., 2016; Qiu et al., 2017; Li et al., 2017; Sun et al., 2018). Particularly, Fmask algorithm and its improved versions are state-of-the-art algorithms (Zhu and Woodcock, 2012; Qiu et al., 2019), and the CFMask algorithm is employed to generate cloud masks for Landsat Level 1 product. Since they detect clouds based on the test of spectral values, they are called as spectral test methods (Foga et al., 2017). The latter is based on the observation that clouds always move. They detect pixels changed at different time as cloudy pixels (Wang et al., 1999; Hagolle et al., 2010; Jin et al., 2013; Zhu and Woodcock, 2014; Frantz et al., 2015; Zhu and Helmer, 2018).

Supervised methods usually formulate cloud detection as pixel classification problem. A classifier assign each pixel one class from some given classes, e.g., clear, cloud and cloud shadow. The classifier is supervised by a set of training pixels together with their ground truth. That is, the classifier is trained such that the training pixels are classified as close as possible to their ground truths, and then used to classify unlabeled pixels (Ricciardelli et al., 2008; Amato et al., 2008). The employed classifiers include support vector machine (SVM) (Lee et al., 2004), random forest (Wei et al., 2020), multi-layer perception neural network (MLP) (Lee et al., 1990; Tian et al., 1999; Scaramuzza et al., 2012; Hughes and Hayes, 2014) and etc. However, the features input to the above classifiers are handcrafted.

Pixel classification is extended to superpixel (a set of similar and connected pixels in a local area) classification to classify superpixels instead of pixels. DCNNs are introduced to exploit the spectral and spatial features in an image patch for cloud detection (Xie et al., 2017; Zi et al., 2018; Wei et al., 2020). Their neurons are arranged in 3 dimensions of height, width and depth to naturally represent pixels in an input image patch and its feature maps. A DCNN convolves an input image patch step by step to output class scores of different classes and consequently classifies it as clear or cloudy patch. Since the features are extracted from the input image by the DCNN itself instead of some independent handcrafted extractors, the classifier (DCNN) together with the features used in classification are completely supervised by the input images together with their ground truths. In this regard, DCNN training is end-to-end. However, contexts among superpixels are not modeled as they are classified independently.

Semantic segmentation is an alternative formulation for pixel classification. By removing the fully connected layer, dedicated DCNNs outputting dense vectors of class scores are developed to classify all pixels simultaneously instead of independently. Benefiting from the open datasets and GPU acceleration, remote sensing image semantic

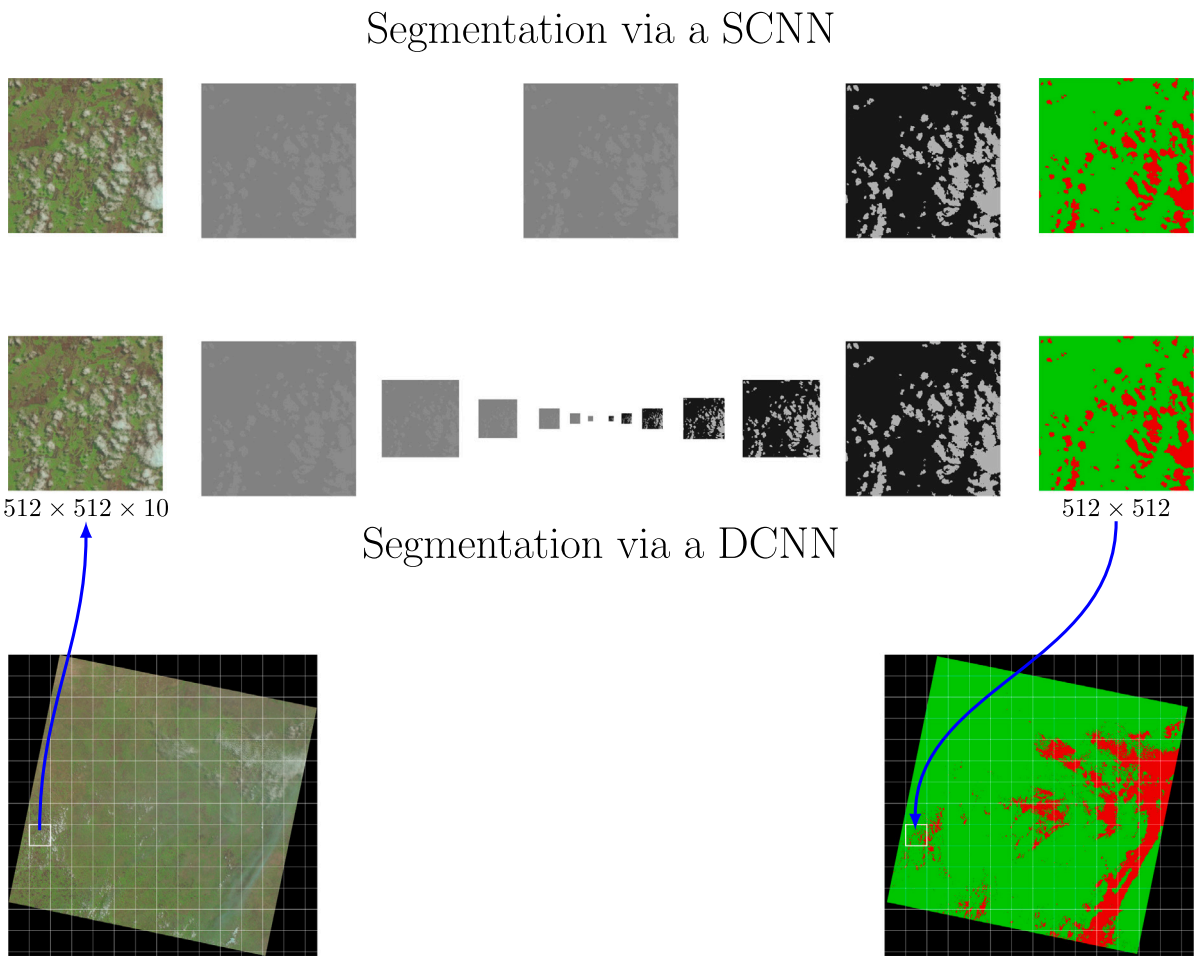

Fig. 1. SCNN-based segmentation vs. DCNN-based segmentation. Remote sensing image cloud detection is achieved via patch-by-patch segmentation, which is based on either a DCNN or a SCNN. The feature maps are down/up sampled by pooling/unpooling layers in a typical DCNN. In contrast, no down/up sampling is involved in the feature extraction via the proposed SCNN. There are many layers (e.g., 13 convolutional layers and 13 deconvolution layers) in a DCNN but only 3 convolutional layers in the proposed SCNN. In practice, a 2048 × 2048 patch can be convolved by a SCNN within one GPU.

segmentation via DCNNs advanced significantly in past decade. Particularly, semantic segmentation via DCNNs became a basic formulation for cloud detection (Chai et al., 2019). It dominates current researches for cloud detection in remote sensing image as most people believe in the power of DCNNs (Li et al., 2019; Jeppesen et al., 2019; Mohajerani and Saeedi, 2019; Mateo-García et al., 2020; Guo et al., 2020; López-Puigdollers et al., 2021; Skakun et al., 2022). The existing researches were conducted on the open datasets and their performances were evaluated against these datasets. When the trained DCNN is adapted from a source (e.g., World-View-2) to a target sensor (e.g., Sentinel-2), its performance is impaired (Segal-Rozenhaimer et al., 2020). When the trained DCNN is applied to deal with images from the same sensor but outside the datasets for training, no reports on their performances are available. Therefore, it is unclear the trained DCNN is general enough to any new images.

2.2. Semantic segmentation

Semantic segmentation is proposed to assign each pixel a label from a set of labels representing a set of semantic classes to support image understanding. It is achieved via pixel classification using a classifier and some image features.

In the early days, Random Forest (Shotton et al., 2008), Boosting (Shotton et al., 2009), Support Vector Machine (Fulkerson et al., 2009) are dominant classifiers and handcrafted features are dominant features. Pixel classification is carried out independently for each pixel in an image. To classify a pixel, its class scores are calculated by a classifier using some handcrafted features (e.g., color, texture and location) extracted from a local area around this pixel using some traditional feature extractors. To correct noisy labels resulted from independent pixel classification, Conditional Random Fields (CRFs) are employed to enforce the consistent constraints between neighboring pixels (Ladickỳ et al., 2010).

Deep Convolutional Neural Networks (DCNNs) are dominant model for semantic segmentation in the past decade (Long et al., 2015; Noh et al., 2015). They were adapted from AlexNet (Krizhevsky et al., 2012), VGG net (Simonyan and Zisserman, 2014) and GoogLeNet (Szegedy et al., 2015), which were developed for entire image classification. The fully connected layers outputting a single vector class scores for the entire image are replaced with convolutional ones outputting dense class score maps. Deconvolution and dilated (atrous) convolution are introduced to increase the spatial resolution of dense score maps. The former is employed in U-net (Ronneberger et al., 2015), SegNet (Badrinarayanan et al., 2017), and etc. to up-sample the feature maps. The latter is employed in DeepLab (Chen et al., 2018), PSPnet (Zhao et al., 2017), and etc. to down-sample and up-sample feature maps using filters with holes between the nonzero filter taps as in Holschneider et al. (1990). Both the DCNNs and hierarchical features are learned from a set of images and their label maps (ground truths) in an end-to-end manner. They significantly improve segmentation accuracy over traditional methods on semantic segmentation benchmarks (Everingham et al., 2015) and also in remote sensing tasks (Chai et al., 2020).

Cloud detection has been formulated as semantic segmentation (Chai et al., 2019; Li et al., 2019; Jeppesen et al., 2019; Mohajerani and Saeedi, 2019; Mateo-García et al., 2020; Guo et al., 2020; López-Puigdollers et al., 2021; Skakun et al., 2022). It is expensive to provide massive ground truths to train a DCNN, and it is unclear the trained DCNN is general enough to support practical applications.

3. Methods

Cloud detection is formulated as a semantic segmentation problem dedicated to two semantic classes, i.e., cloudy and clear pixels. Unlike the existing methods based on DCNNs, semantic segmentation is achieved via convolving the input image through a SCNN consisting three convolutional layers, which are designed to extract features, calculate scores, and incorporate contexts from neighboring pixels.

3.1. SCNN architecture

The SCNN consists of three layers, which are described as follows:

Input layer stores a $H \times W \times C$ volume, where H, W, C are the number of rows, columns and channels. This paper employs $C = 4$, $C = 7$ and $C = 10$ for Gaofen-1 WFV, Landsat 7 and Landsat 8 images.

Conv layer convolves the input layer to extract 64 features for each pixel by 64 filters of size $1 \times 1 \times C$. As indicated by the filter size, 1 row, 1 column and C channels are involved in each convolution for one pixel. The filter is local in spatial dimensions since it covers a single pixel, but it is global in spectral dimension as it covers all channels.

ReLU layer applies a rectified linear function to each output of the above Conv layer and generates an activation.

Conv layer convolves the above activations by 2 filters of size $1 \times 1 \times 64$ to derive a confidence of cloudy pixel and a confidence of clear pixel respectively.

Conv layer convolves the above confidence by 2 filters of size $3 \times 3 \times 2$ to smooth the confidence map.

3.2. Cloud detection using SCNN

Cloud detection using SCNN is achieved via the prediction based on 3 convolutions illustrated in Fig. 2. The first convolution is carried out as follows:

$$f_k^1 = \sum_{c=1}^{C} \omega_{k,c}^1 f_c^0 + \delta_k^1, k = 1, \ldots, 64, \tag{1}$$

where, $\omega_{k,c}^1$ and δ_k^1 represent the convolutional filter weights and biases respectively, $[f_1^0, \ldots, f_C^0]$ is a vector of pixel values. There are 640 (448, 256) weights and 64 biases when $C = 10$ (7, 4). A rectified linear function is applied to the above results to calculate the activations:

$$f_k^2 = max(0, f_k^1), k = 1, \ldots, 64. \tag{2}$$

The second convolution is carried out as follows:

$$f_{k'}^3 = \sum_{k=1}^{64} \omega_{k',k}^2 f_k^2 + \delta_{k'}^2, k' = 1, 2 \tag{3}$$

where, $\omega_{k',k}^2$ and $\delta_{k'}^2$ are 128 weights and 2 biases in this conv layer. The third convolution is carried out as:

$$f_{k\varepsilon}^4 = \sum_{i=-1}^{1} \sum_{j=-1}^{1} \sum_{k'=1}^{2} \omega_{k\varepsilon,i,j,k'}^3 f_{k'}^3 + \delta_{k\varepsilon}^3, k\varepsilon = 1, 2 \tag{4}$$

where, $\omega_{k\varepsilon,i,j,k'}^3$ and $\delta_{k\varepsilon}^3$ are 36 weights and 2 biases in this conv layer (see Fig. 2).

The score map is generated by calculating the class scores for each pixel using a softmax function as:

$$s_1 = \frac{\exp(f_1^4)}{\exp(f_1^4) + \exp(f_2^4)}, \tag{5}$$

$$s_2 = \frac{\exp(f_2^4)}{\exp(f_1^4) + \exp(f_2^4)}, \tag{6}$$

Each pixel is classified as clear pixel (or cloudy pixel) when $s_1 > s_2$ (or $s_2 > s_1$).

3.3. SCNN training supervised by a set of individual pixels

Since the first, second and third convolution covers a 1×1, 1×1 and 3×3 window in spatial dimensions respectively, the output for a pixel depends only on a 3×3 patch around the pixel. Therefore, only a 3×3 patch around a target pixel is used by the SCNN to predict the class label to be compared with its ground truth. As illustrated by Fig. 3, the SCNN training is supervised by a set of samples, each of which corresponds to one annotated pixel. It consists a pair of 3×3 patch and the ground truth label of the center pixel.

Since the SCNN consists only 3 convolutional layers with 872 (or 680, 488) free parameters, the SCNN training can be supervised by some thousands pixels with ground truths. Such limited training samples are assembled into one batch as illustrated in Fig. 3 to be convolved at one time.

Let $(P, G) = \{(P_n, G_n), n = 1, \ldots, N\}$ denotes N training samples, where P, G denote the set of pixels and ground truths respectively, P_n, G_n denote a pair of pixel and its ground truth respectively. A 3×3 patch around each sample P_n is fetched from the original image and all patches are assembled into one batch denoted as:

$$P_{n,i,j,c}, n = 1, \ldots, N, i = -1, 0, 1, j = -1, 0, 1, c = 1, \ldots, C. \tag{7}$$

This 4-dimensional volume is forwarded through the three layers of SCNN as in the previous subsection to output two scores for all samples:

$$F_{n,k\varepsilon}, n = 1, \ldots, N, k\varepsilon = 1, 2. \tag{8}$$

The convolutional filters are trained such that the output scores meet the ground truths as close as possible. As all samples are processed at one time, SCNN training is very efficient.

As the training samples can be selected arbitrary, no pixel-wise annotation is required anymore. Therefore, as listed in Fig. 4, point annotation and brush annotation are proposed to be alternatives for pixel-wise annotation, which is employed to provide ground truths for current semantic segmentation models for cloud detection. Therefore, the cost of annotation is reduced significantly as pixel-wise annotation for tens full images is replaced with annotation of some thousands individual pixels using either point annotation or brush annotation.

3.4. Loss function and its optimization

The convolutional filters are trained via minimizing an objective function that is the sum of loss functions over all training samples. It is implemented as sum over the dimension for batch illustrated in Fig. 3.

$$L(P, G; \Theta) = -\sum_{n=1}^{N} l(P_n, G_n; \Theta) = -\sum_{n=1}^{N} \log\left(\frac{\exp(F_{n,G_n})}{\exp(F_{n,0}) + \exp(F_{n,1})}\right), \tag{9}$$

where, Θ is the set of parameters, and the loss function for one sample is the cross-entropy loss between predicted score and ground truth label.

The above loss function is minimized by the backpropagation algorithm consisting of a forward pass and a backward pass (LeCun et al.,

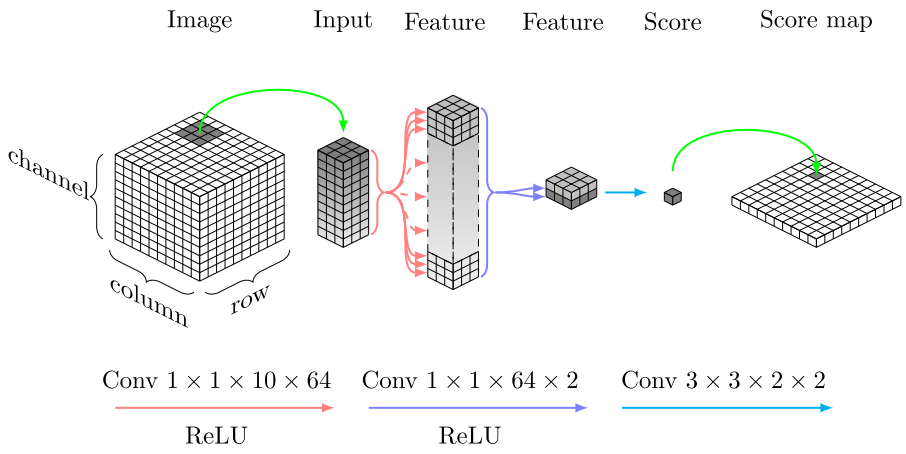

Fig. 2. Forward pass based on shallow convolutional neural network. A 3 × 3 patch is convolved by the neural network to generate a label indicating either cloud or clear for the center pixel. The shallow network consists of only three layers: an input layer, a hidden layer and an output layer.

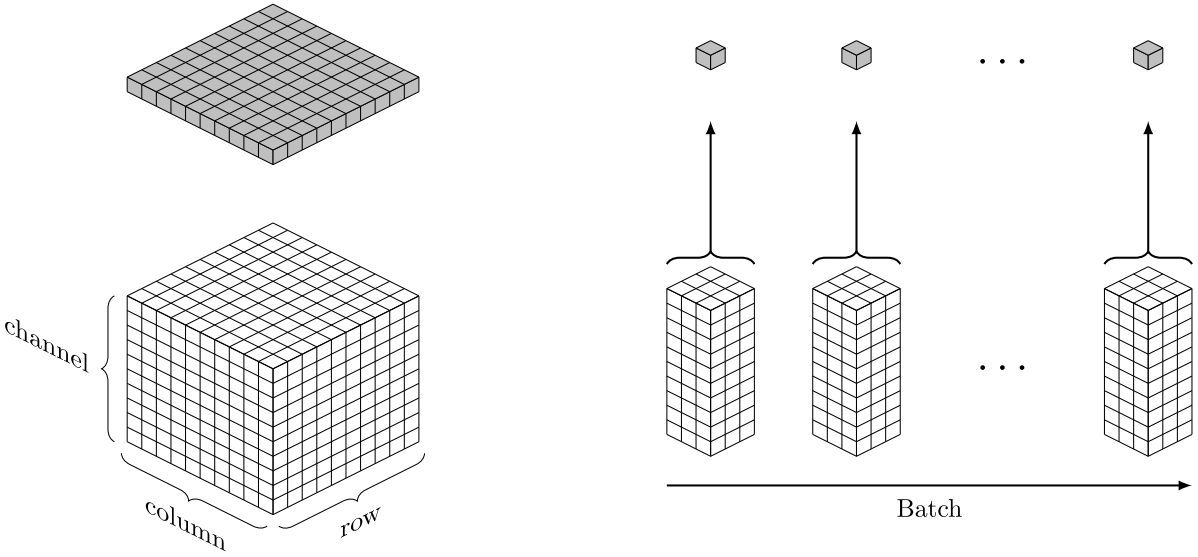

Fig. 3. SCNN training based on a set of individual pixels. The left figure illustrates a pair of image and its ground truth used to train a DCNN. The right figure illustrates a set of training samples used to train a SCNN, where, each training sample consists of a pair of original pixel and its ground truth. Moreover, each original pixel is replaced by a 3 × 3 patch around this pixel, which is fetched from the original image in the procedure of pixel annotation. Each 3 × 3 patch is convolved by the neural network to generate the label to be compared with the ground truth of the center pixel. All training samples are assembled into one batch to be processed as a whole.

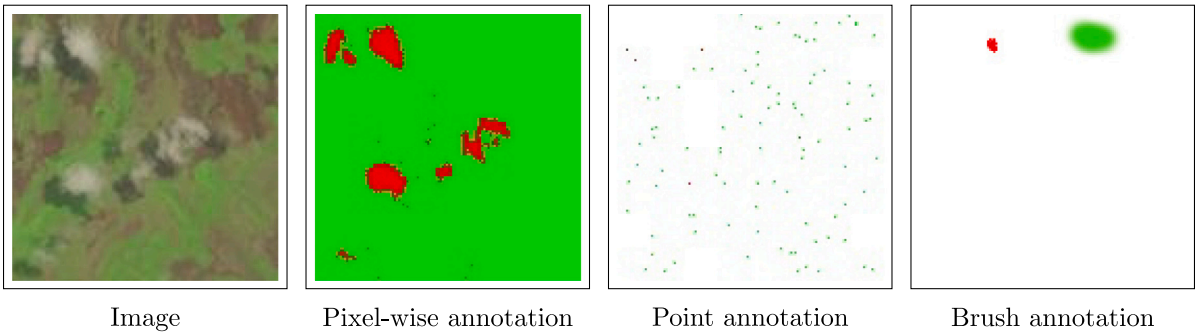

Fig. 4. Different kinds of annotations. Pixel-wise annotation is replaced by point annotation or brush annotation to provide samples for SCNN training.

Table 1
Comparison of SCNN with representative DCNNs.

Network Model	SCNN shallow	FCNN intermediate	CloudNet deep	MSegNet deep	UNet deep
Convolution	✓	✓	✓	✓	✓
Pooling/unpooling	✗	✓	✓	✓	✓
Batch normalization	✗	✓	✓	✓	✓

1998). In each pass, each parameter is updated using the RMSProp schema (Tieleman and Hinton, 2012):

$$\gamma = \beta * \gamma + (1 - \beta) * d\theta^2, \tag{10}$$

$$\theta = \theta - \alpha * d\theta / (\sqrt{\gamma}), \tag{11}$$

where, θ is a parameter in Θ, α and β are the learning rate and decay rate, $\alpha = 0.0001$ and $\beta = 0.995$ are employed in this paper, $\gamma = 0$ is set initially and updated iteratively. Moreover, dropout is employed to activate a neuron with a probability $p = 0.5$ (Srivastava et al., 2014).

4. SCNN versus DCNN

This section compares the SCNN with representative DCNNs including CloudNet (Mohajerani and Saeedi, 2019), FCNN (López-Puigdollers et al., 2021), MSegNet (Chai et al., 2019) and UNet (Ronneberger et al., 2015). CloudNet, FCNN and MSegNet are proposed for cloud detection in remote sensing images, and UNet serves a basic model of many approaches dedicated to cloud detection. Although many other DCNNs are developed for cloud detection, they share the same characteristics with the representatives.

4.1. Comparison of networks

While CloudNet, MSegNet and UNet are DCNNs, FCNN is regarded as an intermediate between SCNN and DCNNs since it has more layers than SCNN but fewer layers than DCNNs. As illustrated by the brief comparison in Table 1, the SCNN removes pooling/unpooling layers and batch normalization layers, only retains convolutional layers. Moreover, the layers and their depths are also significantly reduced in the SCNN. Therefore, the SCNN has much fewer parameters than the other networks. In one word, the SCNN is much simpler and lighter than the other CNNs.

4.2. Comparison of network training

Due to their different complexities, different supervisions are employed to train the deep and shallow networks, and the trained networks perform differently.

Since there are many convolutional layers and pooling/unpooling layers, the output by DCNNs for a pixel depends on a large patch, e.g. a 212 × 212 patch in MSegNet. Moreover, the huge number of free parameters need be trained with supervision of a huge number of pixels. Taking these two factors into account, the network training is supervised by patches. A patch is input to a DCNN to predict the class labels of all pixels in the entire patch to be compared with their ground truths. Therefore, pixel-wise annotation is employed to generate ground truths for DCNN training.

As described in Section 3.3, the output by the SCNN for a pixel depends only on a 3 × 3 patch around the pixel, only a 3 × 3 patch around a target pixel is used by the SCNN to predict the class label of the target pixel. Since only some hundreds free parameters need be trained, the SCNN training can be supervised by some thousands samples, each of which consists a 3 × 3 patch and a label of center pixel. Therefore, as listed in Fig. 4, point annotation and brush annotation can be employed to annotate individual pixels for SCNN training. The laborious pixel-wise annotation for the DCNNs is significantly alleviated as it is easy to label some thousands pixels to train the SCNN using either point annotation or brush annotation.

Moreover, SCNN training and SCNN-based segmentation is much efficient and less costly than DCNN training and DCNN-based segmentation since most layers of DCNNs are removed and only three convolutional layers are retained in SCNN.

Despite its low costs, SCNN training is much stabler than DCNN training as illustrated by Figs. 13 and 14 in Section 5.7.2.

4.3. Comparison of remote sensing image cloud detection

The first issue concerns the spatial resolutions in the feature maps and label maps by DCNNs and SCNN. In the convolutions by DCNNs, the spatial resolutions of intermediate feature maps are first reduced by pooling layers and then increased by unpooling layers. Since the spatial resolution is reduced in pooling, some details unavoidably lose in pooling. The lost details cannot be totally recovered even though the spatial resolution is increased in unpooling. Some solutions have been proposed to address this issue. Badrinarayanan et al. (2017) proposed a solution by using the indexes received from max-pooling to perform unpooling. However, only the indexes corresponding to maximal value are recorded, the other details cannot be recovered and may impair segmentation quality. Feature pyramid network is an alternative solution, which exploits the multi-scale hierarchy of DCNNs to build feature pyramids (Lin et al., 2017). Although the output may be improved by the fusion of multi-scale feature maps, the multi-scale feature maps are generated in a similar way, and therefore the details in a low-level feature map still lose in a high-level feature map. In the convolutions by SCNN, the spatial resolutions of intermediate feature maps are fixed and equal to the resolution of input image as there is no pooling/unpooling layer in the SCNN. The details are completely forwarded from input to output.

The second issue concerns the coherence between neighboring patches. Since remote sensing image is too large to be processed at one time, it needs to be processed patch by patch as shown in Fig. 1. As illustrated in Fig. 5, neighboring patches may have different means and variances even though the pixel values vary smoothly across the neighboring patches. When a pair of neighboring pixels across different patches are normalized using different means and variances in batch normalization (Ioffe and Szegedy, 2015; Badrinarayanan et al., 2017), they may be labeled differently even though they have similar or same values. Consequently, incoherent patches in the final label map are produced by patch-by-patch segmentation via the DCNNs with batch normalization layers. Such artifacts can be repaired to some extent by partitioning the images into overlapping patches and averaging the overlapped patches in the existing literature, but cannot be eliminated completely (Marmanis et al., 2018). Since the SCNN removes batch normalization layers, it does not produce incoherent patches anymore.

The third issue concerns the border artifacts. The global features for a pixel are extracted by DCNNs from a large patch around this pixel, e.g. a 212 × 212 patch by MSegNet. For a pixel at borders of image area, its global features are extracted from a large patch covering filling (black) area. Such incorrect features may result in incorrect label, which is an artifact in the label map. In contrast, the label of a pixel depends only on a 3 × 3 patch, and therefore the border width is only one pixel, and consequently, the border artifacts by SCNN are ignorable.

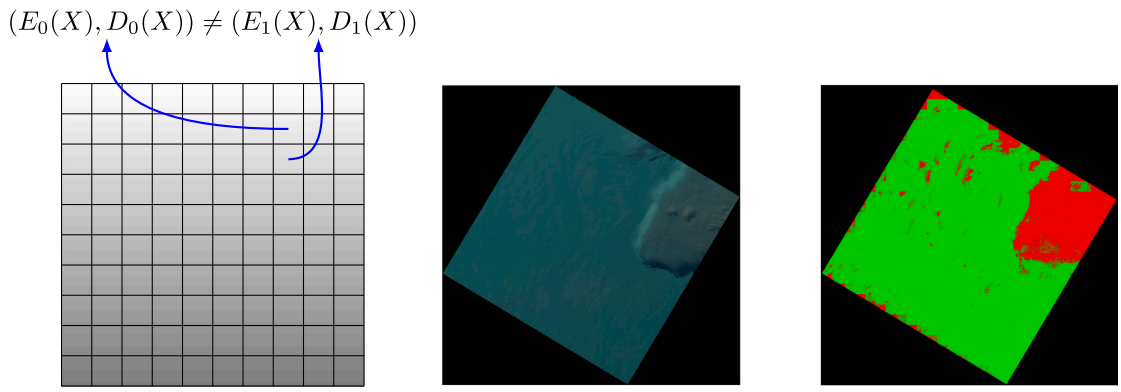

Fig. 5. Incoherent seams and grids induced by patch-by-patch segmentation based on batch normalization. Left figure illustrates that neighboring patches may have different means and variances even though pixel value varies smoothly across the neighboring patches. Incoherent seams and grids appear in the final label map (right) of a Landsat image (middle) when the patches are normalized by different means and variances.

5. Experimental results

This section illustrates cloud detection using the SCNN and compares with four state-of-the-art DCNNs based on both qualitative and quantitative evaluations. Spectral test methods are employed as baselines for comparisons.

5.1. Data description

5.1.1. L7 Irish and L8 Biome cloud cover assessment validation datasets

Evaluations and comparisons of cloud detection methods are carried out using Landsat 7 and Landsat 8 cloud cover assessment validation data[2,3] released by the U.S. Geological Survey (USGS) Earth Resources Observation and Science (EROS) Center. The Landsat 7 scenes were selected by Irish et al. (2006) and referred as **L7 Irish** dataset. The Landsat 8 scenes were selected by Foga et al. (2017) and referred as **L8 Biome** dataset.

Following Foga et al. (2017), 96 L8 Biome scenes are employed in our experiments. As illustrated in Fig. 6, they are evenly distributed over eight biomes: barren, forest, grass/crops, shrubland, snow/ice, urban, wetlands and water. There are 12 scenes for each biome. However, 93 instead of 102 L7 Irish scenes are included in our experiments since the projections in Landsat images (Section 5.1.2) and their ground truth cloud masks are different for the 9 scenes in polar south. As depicted in Fig. 6, the 93 scenes are evenly distributed over eight latitude zones: austral, boreal, mid latitude north, mid latitude south, polar north, sub tropical north, sub tropical south and tropical.

As is standard in the schema for deep learning, each dataset is randomly partitioned into a training set, a validation set and a test set, which are around 60%, 10% and 30% of the whole dataset. For L7 Irish, there are 57, 9 and 27 scenes in the training set, validation set and test set as illustrated by the left figure of Fig. 7. For L8 Biome, there are 56, 8 and 32 scenes in the training set, validation set and test set as illustrated by the middle figure of Fig. 7. The cloud detection models for Landsat 7 and 8 images are trained, validated and tested using the training, validation and test images of L7 Irish and L8 Biome datasets respectively.

A pixel-wise cloud mask is available for each scene in the L7 Irish and L8 Biome cloud cover assessment validation datasets. The cloud masks were produced by analysts at the USGS EROS Center based on manual interpretation of the corresponding Landsat images. The manually labeled cloud masks are employed to serve as ground truths for training and testing even though cloud masks labeled by different people may differ about 7% for the same scene (Scaramuzza et al., 2012).

Only 100 pixels randomly selected from each image (point annotation) are employed to train a SCNN for cloud detection even though the ground truths for all pixels are available. In contrast, all pixels are input to train a DCNN for cloud detection. Moreover, a pixel is treated as either cloudy pixel or clear pixel even though it is labeled as cloud, thin cloud or clear (cloud shadow is further distinguished from clear in some scenes). More specifically, thin cloud is treated as cloud, and cloud shadow is treated as clear.

5.1.2. Landsat 7 and Landsat 8 Collection 1 data preprocessing

The Collection 1 data corresponding to the above 93 L7 Irish scenes and 96 L8 Biome scenes were downloaded from the USGS EROS Center archive.[4]

The Digital Number (DN) images in the Collection 1 data are converted to Top of Atmosphere (TOA) Reflectance and TOA Brightness Temperature (BT) images for cloud detection as TOA is a better choice for cloud detection (Zhu and Woodcock, 2012; Chai et al., 2019). The conversions of Landsat 7 and 8 DN images are conducted by Landsat Ecosystem Disturbance Adaptive Processing System (LEDAPS) Land Surface Reflectance Modules and Landsat 8 Surface Reflectance Code (LaSRC) respectively, which were downloaded from USGS-EROS at GitHub.[5]

Since the cloud bit of each pixel in the Quality Assessment (QA) band in Collection 1 data is set to be 1 and 0 according to the output of CFMask algorithm (Zhu and Woodcock, 2012; Foga et al., 2017), the QA image is used to derive a cloud mask to serve as a baseline for comparison.

5.1.3. GF1 WHU cloud validation dataset

To test the generality of the proposed method, evaluations and comparisons are also extended to **GF1 WHU** cloud validation dataset data[6] released by Wuhan University. This globally distributed dataset consists of 108 Level 2 A Gaofen-1 scenes of different land-covers (Li et al., 2017). A Gaofen-1 WFV image (4 bands spanning the visible to near-infrared of 16 m spatial resolution) and a manually labeled pixel-wise cloud mask are provided for each scene. 86 and 22 scenes are

[2] https://landsat.usgs.gov/landsat-7-cloud-cover-assessment-validation-data

[3] https://landsat.usgs.gov/landsat-8-cloud-cover-assessment-validation-data

[4] https://earthexplorer.usgs.gov/

[5] https://github.com/USGS-EROS/espa-surface-reflectance/tree/master

[6] http://sendimage.whu.edu.cn/en/mfc-validation-data

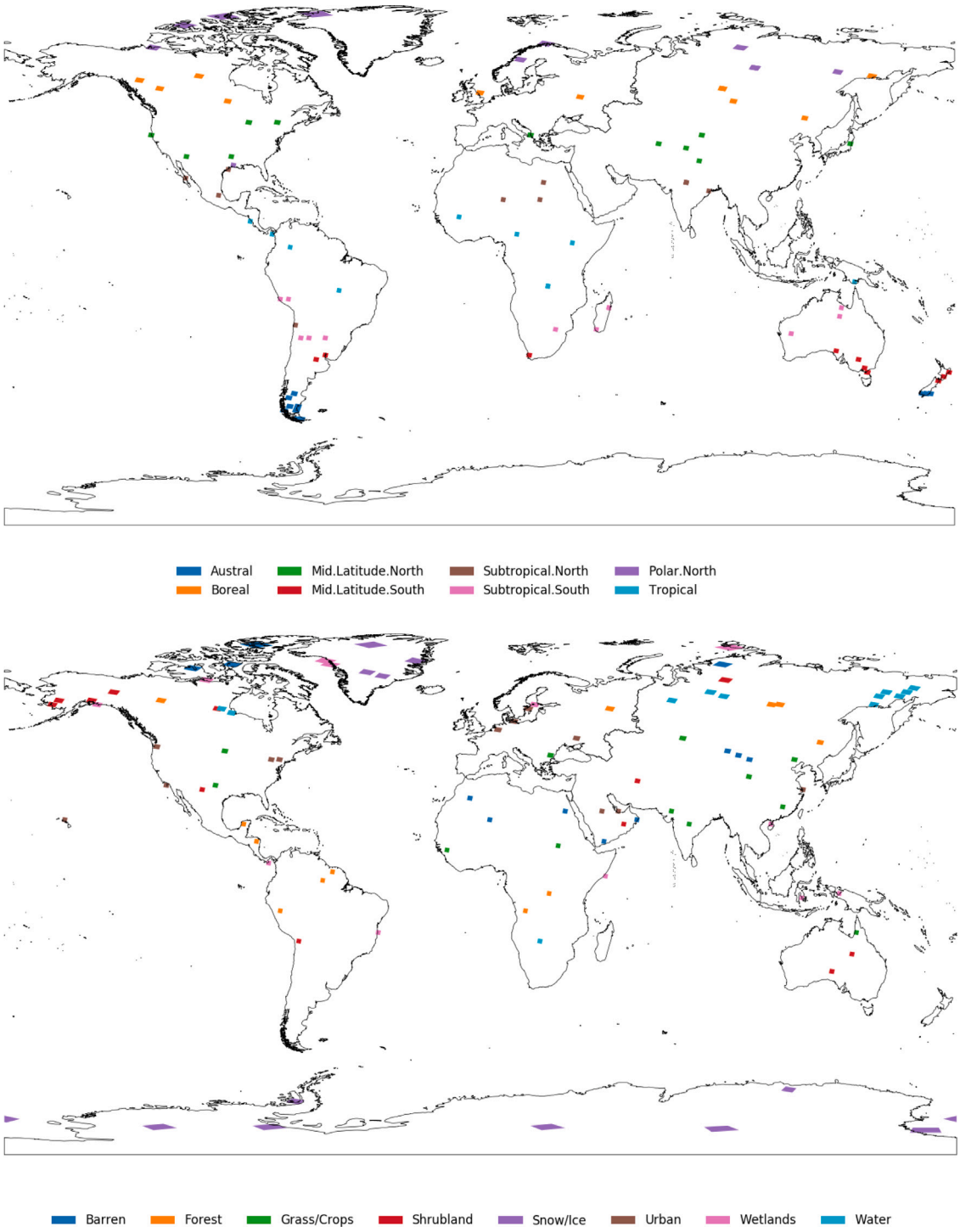

Fig. 6. Overview of the 93 unique L7 Irish(up) and 96 L8 Biome(bottom) scenes employed in the experiments. The L7 and L8 scenes are evenly distributed over eight latitude zones and eight biomes respectively.

grouped into the training set and test set in Li et al. (2019), 14 from the 86 training scenes are selected to constitute the validation set. The three sets are illustrated in the right figure of Fig. 7.

The available pixel-wise cloud masks are employed to serve as ground truths for cloud detection to support experiments on this dataset, which are similar to those on the L7 Irish and L8 Biome datasets. However, unlike the cloud detection in Landsat image, cloud detection in Gaofen-1 WFV image is based on DNs instead of TOA Reflectances and it is based on 4 bands. The baseline label maps for comparison is generated using multi-feature combined (MFC) algorithm (Li et al., 2017) developed by Wuhan University.[7]

[7] http://sendimage.whu.edu.cn/en/mfc

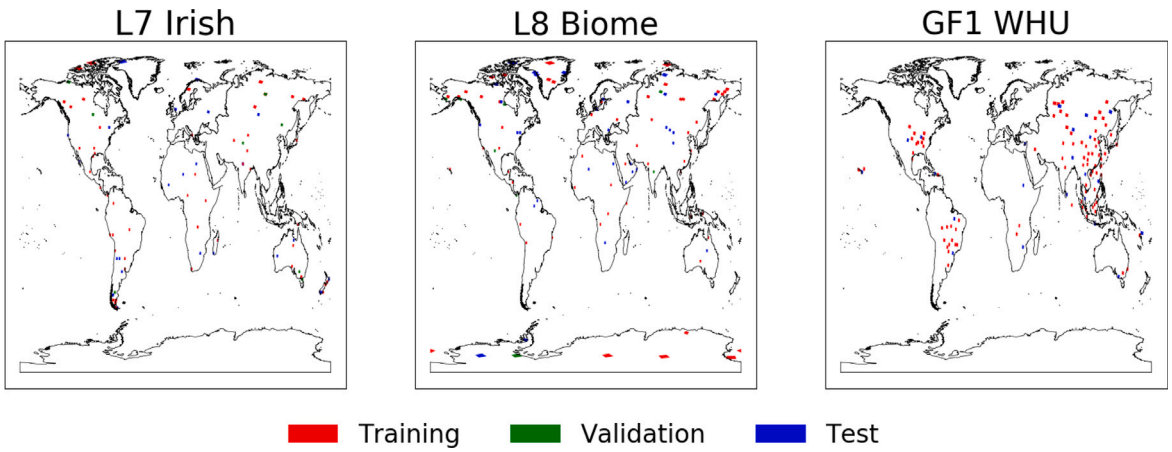

Fig. 7. Training, validation and test sets of the L7 Irish, L8 Biome and GF1 WHU datasets. Left, middle and right illustrate the scenes in L7 Irish, L8 Biome and GF1 WHU datasets grouped into the training set, validation set and test set respectively.

5.2. Quantitative metrics

By comparing the cloud masks detected by a method with their corresponding ground truths, 4 numbers are counted: N_{00} is the number of pixels being clear in both cloud mask and ground truth, N_{01} is the number of pixels being clear in cloud mask but cloudy in ground truth, N_{10} is the number of pixels being cloudy in cloud mask but clear in ground truth, N_{11} is the number of pixels being cloudy in both cloud mask and ground truth.

Then, the producer's and user's accuracies are calculated as:

$$A^P_{clear} = \frac{N_{00}}{N_{00} + N_{10}}, \tag{12}$$

$$A^P_{cloud} = \frac{N_{11}}{N_{01} + N_{11}}, \tag{13}$$

$$A^U_{clear} = \frac{N_{00}}{N_{00} + N_{01}}, \tag{14}$$

$$A^U_{cloud} = \frac{N_{11}}{N_{10} + N_{11}}. \tag{15}$$

The producer's and user's accuracies are the complements of omission and commission errors, respectively, which are alternatives for evaluation in the literature (Foga et al., 2017). The producer's and user's accuracies are also named as recall (completeness) and precision (correctness) in other literature (Chai et al., 2020)

Furthermore, the F_1 scores are calculated as follows:

$$F_{1,clear} = 2\frac{A^U_{clear} * A^P_{clear}}{A^U_{clear} + A^P_{clear}}, \tag{16}$$

$$F_{1,cloud} = 2\frac{A^U_{cloud} * A^P_{cloud}}{A^U_{cloud} + A^P_{cloud}}. \tag{17}$$

Finally, the overall accuracy is calculated as:

$$A^O = \frac{N_{00} + N_{11}}{N_{00} + N_{01} + N_{10} + N_{11}}. \tag{18}$$

5.3. State-of-the-art methods

Table 2 presents an overview of the characteristics of the state-of-the-art methods included for comparison. MUNet is an implementation of UNet (Ronneberger et al., 2015) based on the idea of MobileNets depthwise separable convolutions (Howard et al., 2017).

5.4. Experiment setup

The experiments are carried out on a desktop computer with an Intel Core i9-9820X CPU @3.30GHz*20 processor, 128 GB Memory and a NVDIA GeForce GTX 2080Ti/PCIe/SSE2 GPU.

The public available codes for the DCNNs are downloaded and integrated into our cloud detection platform. The superparameters are set as described in Section 3.4. For each DCNN to be compared, 3 models for Landsat 7 images, Landsat 8 images and Gaofen-1 WFV images are trained from scratch using 57 Landsat 7 images, 56 Landsat 8 images and 74 Gaofen-1 WFV images respectively. Due to GPU memory limitation, an entire Landsat image or Gaofen-1 image is segmented patch by patch, and all training images are partitioned into 256 × 256 patches, and those without filling pixels are fed to the training pipeline batch by batch. The batchsize is set to 8 for MSegNet, MUNet and CloudNet, and it is set to 16 for FCNN, which is lighter than the others.

The final label map for an entire image is generated by seaming all the label maps of patches. Considering the border effect of DCNNs, only the 128 × 128 center part of each 256 × 256 patch is retained to compose the entire label map. To this end, an entire label map is partitioned into 128 × 128 patches, and a 256 × 256 patch with the same center of each 128 × 128 patch is fetched from the original image and convolved to output the label map of this 128 × 128 patch. It means neighboring patches in the original image overlap 50% with each other. We do not average the values over overlapped patches but just discard the border parts.

The SCNN is trained using a set of individual pixels as described in Section 3.3. Specifically, 3 SCNNs for Landsat 7, Landsat 8 and Gaofen-1 WFV images are trained from scratch using 5700, 5600 and 7400 pixels by randomly selecting 100 pixels (except the filling pixels) from each of the 57 Landsat 7 images, 56 Landsat 8 images and 74 Gaofen-1 WFV images respectively. As presented in Table 3, their samples are also from different classes, and their numbers are proportional to their total numbers in all images. As described in Section 3.3, only a 3 × 3 patch around each training pixel is fed to the training pipeline. All 5700 (5600, 7400) 3 × 3 patches together are fed to train the SCNN at one time. To generate a label map for an entire remote sensing image, it is partition into 2046 × 2046 patches, and a 2048 × 2048 patch is fetched from original image to generate a label map for each patch as one pixel border is discarded. Only one is retained for overlapped patches.

5.5. Cloud detection based on SCNN

The models trained as in Section 5.4 are applied to detect clouds in all the testing images in L7 Irish, L8 Biome and GF1 WHU datasets respectively for evaluation and comparison.

5.5.1. Qualitative evaluation

All Landsat images in the test set are segmented to detect the clouds as described in Section 3. Two L7 Irish examples are presented in Fig. 15, two L8 Biome examples are presented in Fig. 16, and two

Table 2
Comparison of SCNN with FMask, MFC and 5 DCNNs. Please refer Table 5 for quantitative comparison of efficiency.

Category	Method	Spatial feature	Supervision	Samples	Efficiency
Spectral test	Fmask MFC	local	unsupervised	no	
Deep learning	MSegNet MUNet FCNN CloudNet	global	supervised	tens images	slow
Shallow learning	SCNN	local	supervised	thousands pixels	fast

Table 3
Samples used in the experiments. The experiment illustrated in Fig. 11 is based on the SCNNs trained using randomly selected 20, 50, 100, 200, 500, 1000 pixels from each image respectively. The other experiments are based on the SCNNs trained using 100 pixels from each image. The samples are treated as either clear or cloudy pixels even though pixels covered by cloud shadow (CShadow) and thin cloud (TCloud) are distinguished from clear and cloudy pixels respectively.

	L7 Irish				L8 Biome				GF1 WHU		
	CShadow	Clear	TCloud	Cloud	CShadow	Clear	TCloud	Cloud	CShadow	Clear	Cloud
20	3	741	160	236	18	589	154	359	43	1052	385
50	11	1848	391	600	62	1444	400	894	136	2631	933
100	22	3705	791	1182	113	2899	821	1767	261	5292	1847
200	48	7350	1623	2379	241	5765	1678	3516	518	10 593	3689
500	125	18 447	4023	5905	590	14 388	4167	8855	1213	26 485	9302
1000	213	36 921	8045	11 821	1262	28 713	8364	17 661	2478	53 195	18 327

Table 4
Quantitative evaluation for SCNN on L7 Irish, L8 Biome and GF1 WHU datasets listed in Fig. 7. It reports A^O together with A^P, A^U, F_1 for both clear and cloud.

Dataset	Zone/Biome		Clear			Cloud		
		A^O	A^P	A^U	F_1	A^P	A^U	F_1
L7 Irish	Austral	88.87	86.94	76.29	81.27	89.61	94.69	92.08
	Boreal	97.06	98.78	96.53	97.64	94.31	97.96	96.10
	Mid.N.	93.90	97.04	89.02	92.86	91.74	97.82	94.68
	Mid.S.	97.19	98.96	95.94	97.43	95.13	98.75	96.91
	Polar.N.	83.55	81.99	84.55	83.25	85.11	82.61	83.84
	SubT.N.	98.16	99.44	98.65	99.04	69.09	84.38	75.97
	SubT.S.	95.35	99.02	95.10	97.02	83.51	96.34	89.47
	Tropical	89.72	86.48	42.76	57.23	90.00	98.72	94.16
	Mean	94.13	97.17	93.73	95.42	88.97	94.88	91.83
L8 Biome	Barren	93.46	90.81	97.89	94.22	97.23	88.19	92.49
	Forest	96.65	90.65	88.18	89.40	97.76	98.26	98.01
	Grass	94.40	93.75	99.51	96.54	97.67	75.56	85.20
	Shrubland	99.24	98.69	99.83	99.26	99.83	98.64	99.23
	Snow	86.50	78.41	91.65	84.51	93.67	83.04	88.03
	Urban	95.29	88.87	98.29	93.34	99.08	93.76	96.35
	Wetlands	93.72	93.90	98.36	96.07	92.92	77.14	84.30
	Water	97.52	96.65	99.29	97.95	98.90	94.92	96.87
	Mean	94.35	91.75	97.75	94.65	97.47	90.79	94.01
GF1 WHU	Mean	94.46	92.07	97.62	94.76	97.31	91.11	94.11

GF1 examples are presented in Fig. 17. The scenes in Fig. 15 are from tropical and middle latitude north zones respectively. The scenes in Fig. 16 are from urban and snow scenes respectively. The scenes in Fig. 17 are covered by water and snow respectively. They represent various cloud coverages as about 90%, 60%, 40%, 20%, 40% and 75% of the six images are covered by clouds. Besides, different cloud sizes and cloud shapes appear in these examples.

As demonstrated by these examples, the proposed SCNN successfully detects clouds, which match their ground truths well. Meanwhile, some isolated and noisy pixels can be observed in the results by SCNN since the segmentation is based mostly on spectral features. Some cloudy pixels are misclassified as clear pixels in first example of Fig. 15 and second example of Fig. 16, which is a scene of snow/ice and is hard to deal with.

5.5.2. Quantitative evaluation

Table 4 presents the quantitative evaluation in terms of eight latitude zones of L7 Irish dataset and eight biomes of L8 Biome dataset. The evaluation averaged on test set of GF1 WHU dataset is also presented in the table. The overall accuracies on L7, L8 and GF1 datasets are over 94%. The overall accuracy on all the zones and biomes are over 80%.

Since snow/ice is similar to cloud in the images, it is challenging to distinguish them based on spectral features, the performances on polar north zone (L7 Irish) and snow/ice scene (L8 Biome) are inferior to the other scenes. In the Landsat 7 images of polar north zone, the user's accuracy, producer's accuracy and F_1 score are below 90% but above 80% for both snow/ice pixels and cloudy pixels. In the Landsat 8 images of snow/ice biome, the producer's accuracy and F_1 score are above 78% for snow/ice pixels, the user's accuracy and F_1 score are above 83% for cloudy pixels. The performances on austral and tropical zones (L7 Irish) are also impaired by their clear pixels.

Considering the fact that cloud masks labeled by different people for the same scene may differ about 7% (Scaramuzza et al., 2012), the achieved accuracies are very high as the user's accuracy, producer's accuracy and F_1 score averaged over eight zones or eight biomes are over 90% for both clear pixels and cloudy pixels.

5.6. Ablation study

Fig. 8 illustrates the performances achieved by the SCNNs with different number of layers. As described in Section 3.1, the basic SCNN

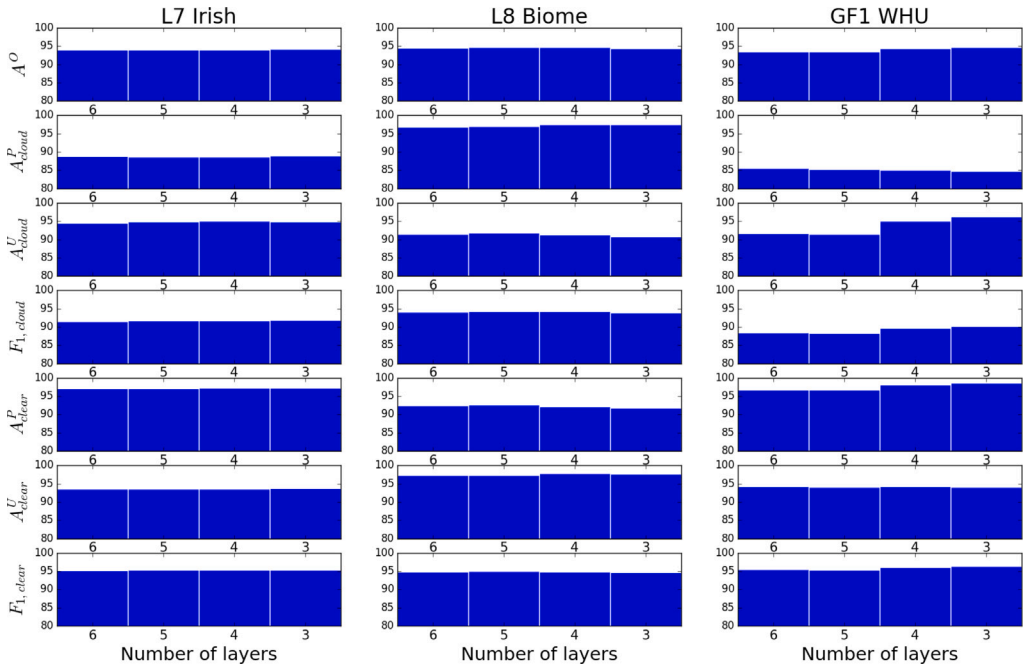

Fig. 8. Quantitative comparison for SCNN with different number of layers. The basic SCNN consists of 3 layers. Three versions of 4, 5 and 6 layers are constructed by adding 1, 2 and 3 feature extraction layers to the basic SCNN. From top to bottom, it reports A^O together with A^P_{cloud}, A^U_{cloud}, $F_{1,cloud}$ A^P_{clear}, A^U_{clear}, $F_{1,clear}$.

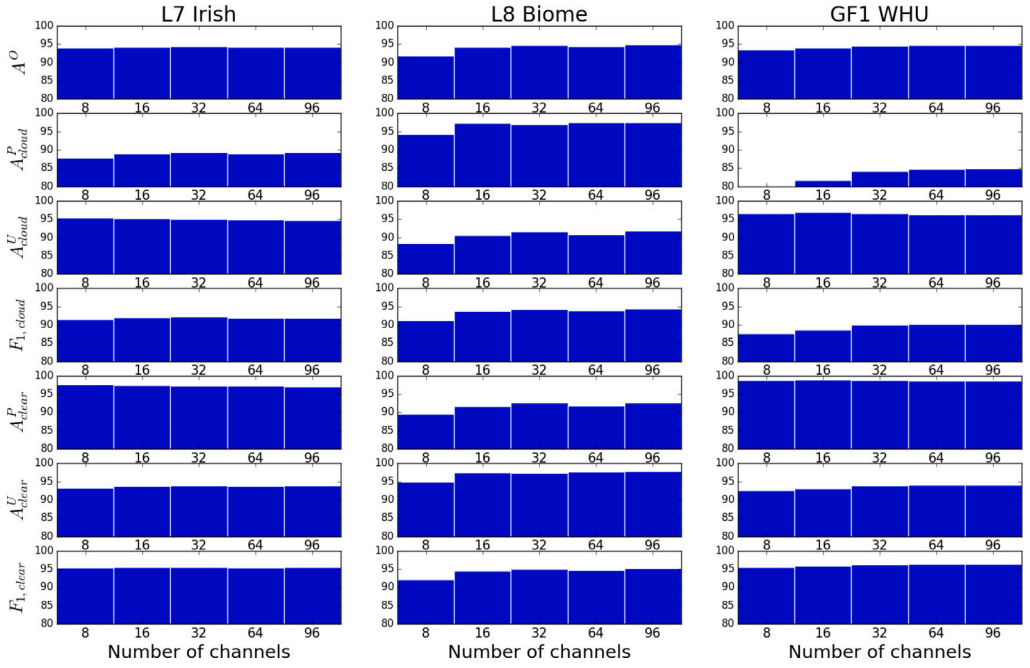

Fig. 9. Quantitative comparison for SCNN with feature extraction layer using 8, 16, 32, 64 and 96 kernels. From top to bottom, it reports A^O together with A^P_{cloud}, A^U_{cloud}, $F_{1,cloud}$ A^P_{clear}, A^U_{clear}, $F_{1,clear}$.

consists three layers, one of which is feature extraction layer. Three SCNNs consists of 4, 5 and 6 layers are constructed by adding 1, 2 and 3 feature extraction layers with 64 kernels to the basic SCNN. It is clear that the performance of network is stable with respect to the number of layers and is not improved by the extra layers. This motivate us to propose a shallow CNN for cloud detection. As supported by this comparison, one feature extraction layer is enough for cloud detection.

Fig. 9 illustrates the performances achieved by the SCNN whose feature extraction layer has different number of kernels. As described in Section 3.1, the proposed feature extraction layer consists 64 kernels. Four variants are developed by employing 8, 16, 32 and 96 kernels for the feature extraction layer respectively. Obviously, SCNN with 8 kernels is outperformed by that with more kernels. However, as indicated by this comparison, the performance is stable with respect to the rest number of kernels. SCNN with 64 kernels successfully extract features distinguishing cloudy pixels from clear pixels and slightly outperform SCNN with 16 and 32 kernels, its performance is almost the same with using 96 kernels. However, 64 kernels are preferred as they are lighter than 96 kernels.

Fig. 10 illustrates the performances achieved by the SCNN whose filter has different sizes. As described in Section 3.1, 2 filters of size $3 \times 3 \times 2$ are employed in the last convolutional layer to smooth the

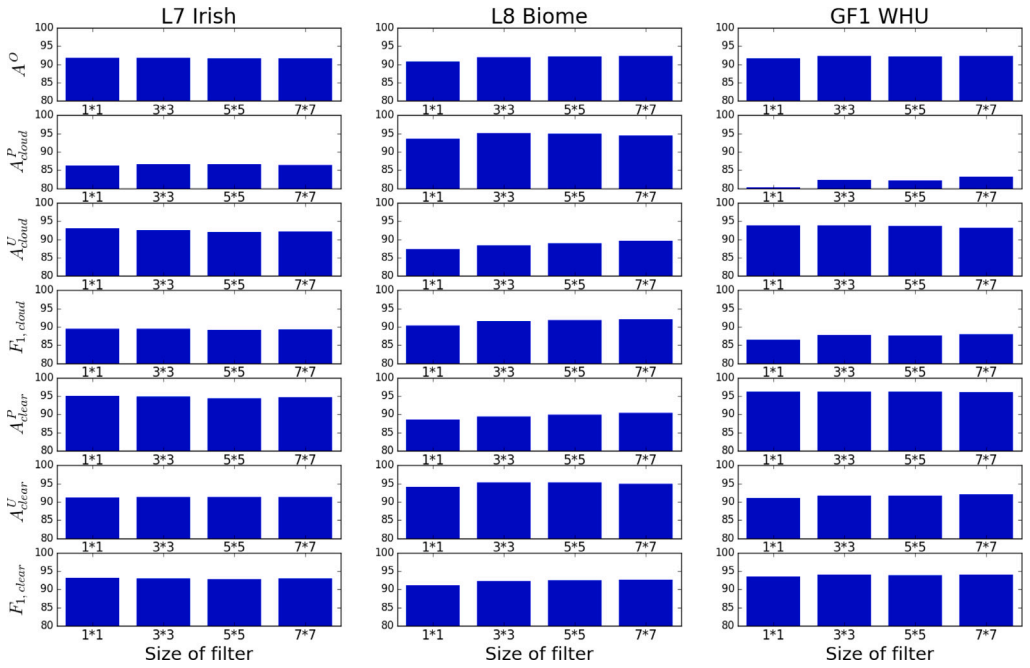

Fig. 10. Quantitative comparison for SCNN with filters of $1 \times 1, 3 \times 3, 5 \times 5, 7 \times 7$ respectively. From top to bottom, it reports A^O together with A^P_{cloud}, A^U_{cloud}, $F_{1,cloud}$ A^P_{clear}, A^U_{clear}, $F_{1,clear}$.

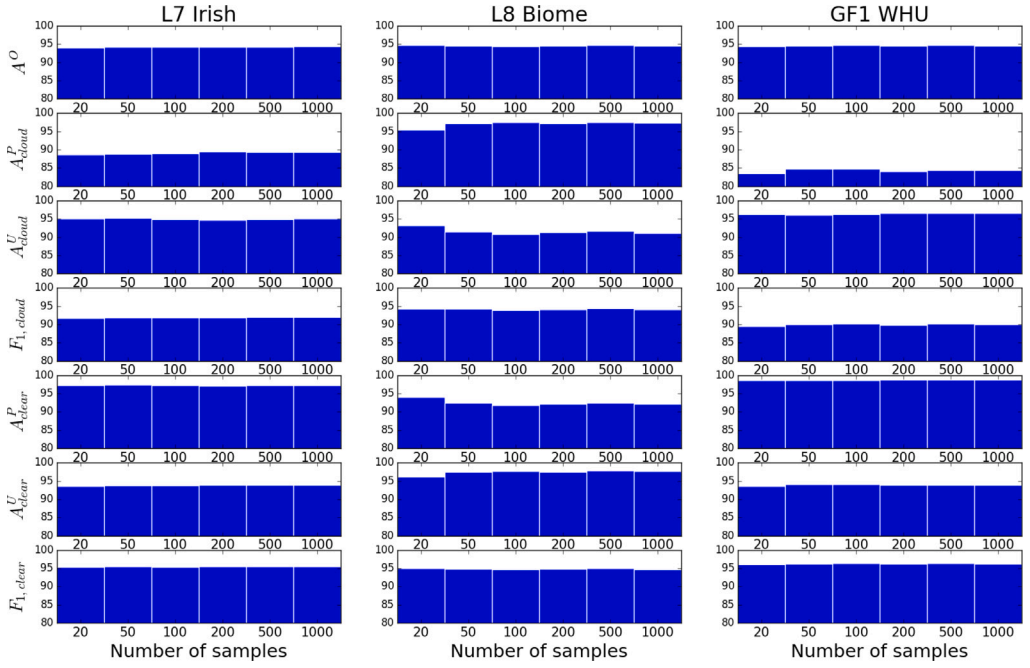

Fig. 11. Quantitative comparison for SCNN trained using 20, 50, 100, 200, 500, 1000 pixels respectively. From top to bottom, it reports A^O together with A^P_{cloud}, A^U_{cloud}, $F_{1,cloud}$ A^P_{clear}, A^U_{clear}, $F_{1,clear}$.

confidence map. They are replaced by filters of $1 \times 1 \times 2$, $5 \times 5 \times 2$ and $7 \times 7 \times 2$ respectively. Since the confidence map cannot be smoothed by filters of $1 \times 1 \times 2$, it is outperformed by the other filters. It ensures that smoothing improves performance. However, the performance is not further improved by larger filters. Therefore, $3 \times 3 \times 2$ are employed in the proposed SCNN.

Fig. 11 illustrates the performances achieved by the SCNN trained using different number of samples. As illustrated in Table 3, 20, 50, 100, 200, 500, 1000 pixels are randomly drawn from each image in the training set. Given 20 pixels in each image, there are 57*20 = 1140 (56*20 = 1120, 74*20 = 1480) training pixels, they are enough to train a SCNN with 872 (680, 488) parameters. As shown, they all achieve good performances, and their performances are stable with respect to the number of training samples. Their differences are ignorable in all three datasets. It is preferred to use fewer samples to save annotation time, however, more samples need not be excluded from training if they are already available for use.

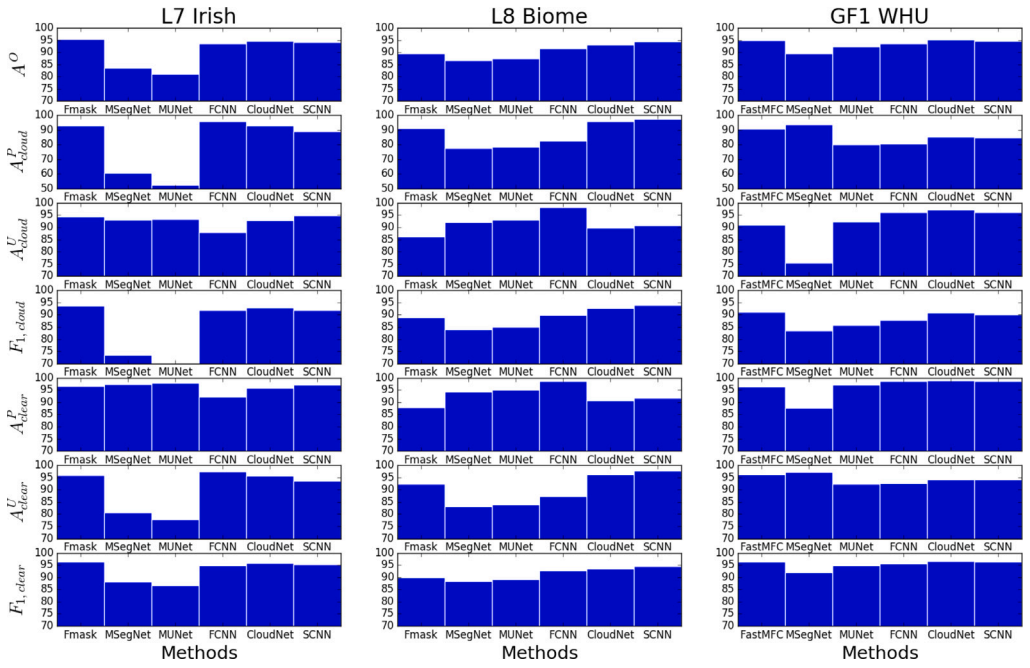

Fig. 12. Quantitative comparison of SCNN with DCNNs. From top to bottom, it reports A^O together with A^P_{cloud}, A^U_{cloud}, $F_{1,cloud}$ A^P_{clear}, A^U_{clear}, $F_{1,clear}$.

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

5.7.1. SCNN vs. Spectral test

The proposed SCNN is compared with Fmask algorithm (Zhu and Woodcock, 2012) and MFC algorithm (Li et al., 2017), which are developed for cloud detection in Landsat images and Gaofen-1 images respectively.

Both SCNN and Fmask algorithm extract local spatial features from TOA reflectance and TOA brightness temperature to detect cloudy pixels from clear pixels. SCNN distinguishes from Fmask algorithm in two ways. First, SCNN adopts three convolutions while Fmask employs some specific rules such as $B7 > 0.03$ and $BT < 27$ and $NDSI < 0.8$ and $NDVI < 0.8$ (Please refer (Zhu and Woodcock, 2012) for detailed descriptions). Second, the parameters of SCNN are learned with supervision while the thresholds for Fmask are calculated from the target Landsat image without supervision.

Two L7 Irish examples, two L8 Biome examples and two GF1 WHU examples are presented in Figs. 15, 16 and 17 respectively. In the second example of Fig. 16, Fmask algorithm fails to detect clouds in snow scene as it generates cloud masks that do not match their ground truths. Most pixels over snow/ice are incorrectly classified as cloudy pixels based on their high TOA values. In the second example of Fig. 17, some pixels of snow are detected as pixels of cloud by MFC algorithm. In contrast, clouds detected by SCNN match their ground truths even though they are not very accurate.

Overall, the general convolutions by supervised learning allow the SCNN to exploit more features to distinguish cloudy pixels from clear pixels, especially from pixels of snow/ice. They are better than the specific rules employed by the FMask and MFC algorithm. As shown in Fig. 12, SCNN significantly outperforms Fmask on L8 Biome datasets but performs a little bit inferior to Fmask on L7 Irish dataset. The Fmask performs well on L7 Irish dataset partly due to that snow/ice scenes in polar south zone are not included in the experiments.

5.7.2. SCNN vs. DCNN

SCNN is much simpler and lighter than DCNNs although both the SCNN and DCNNs are built upon convolutional neural network.

Fig. 13 reveals to some extent the difference between DCNN training and SCNN training by presenting both averaged accuracy on validation set and averaged loss on training set for every epoch in the procedure of training. Good convergence is achieved by SCNN training as its loss decreases smoothly and converges to a constant. Meanwhile, its averaged accuracy on validation set increases smoothly as the epochs increases. Its performance on validation and training set matches each other. In contrast, the accuracy curve of a DCNN may vary sharply as epoch increases even when the its loss reduces a little. Smooth loss curve means small difference in loss and small difference in accuracy on training set between neighboring epochs. The vibration in accuracy curve on validation set means large difference in accuracy on training set between neighboring epochs. The small and large difference indicate that DCNNs' performances on validation and training sets do not match. When the number of training parameters increases from hundreds to millions, the objective function gets much more complex and much harder to be optimized. It is much harder to train a DCNN with millions free parameters than a SCNN with hundreds parameters. It is well-known that DCNN training is very tricky.

The above observation is further convinced by their performances on test set presented in Fig. 14. 5 models of the last 5 epochs are saved in the procedure of model training. These models are applied to the images in the test set and their results are evaluated and reported in this figure. As illustrated by the figure, all the curves for SCNN are horizontal lines. It means that SCNN also performs stably on test set when the training converges. In contrast, DCNNs perform unstably on test set at the last 5 epochs as their curves vary sharply. It ensures that SCNN outperforms DCNNs as its metrics are higher than DCNNs' metrics. The curves of MSegNet and MUNet are significantly below the curves of SCNN, the curves of FCNN and CloudNet are slightly below the curves of SCNN. Averaged over the 5 epochs, its overall accuracy is 1.90%, 1.55% and 3.11% higher than that of FCNN on L7 Irish, L8 Biome and GF1 WHU respectively, 0.2%, 2.19% and 3.03% higher than that of CloudNet on the three datasets respectively.

The metrics on test set for the models at last epoch are also drawn as bars in Fig. 12. SCNN outperforms MSegNet and MUNet significantly. Particularly, its overall accuracy is more than 10% higher on L7 Irish. MSegNet and MUNet misclassify many cloudy pixels as clear pixels since their producer's accuracy are around 60%. SCNN outperforms FCNN slightly as its overall accuracy is 0.49%, 2.77% and 0.94% higher on L7 Irish, L8 Biome and GF1 WHU respectively. SCNN is competitive

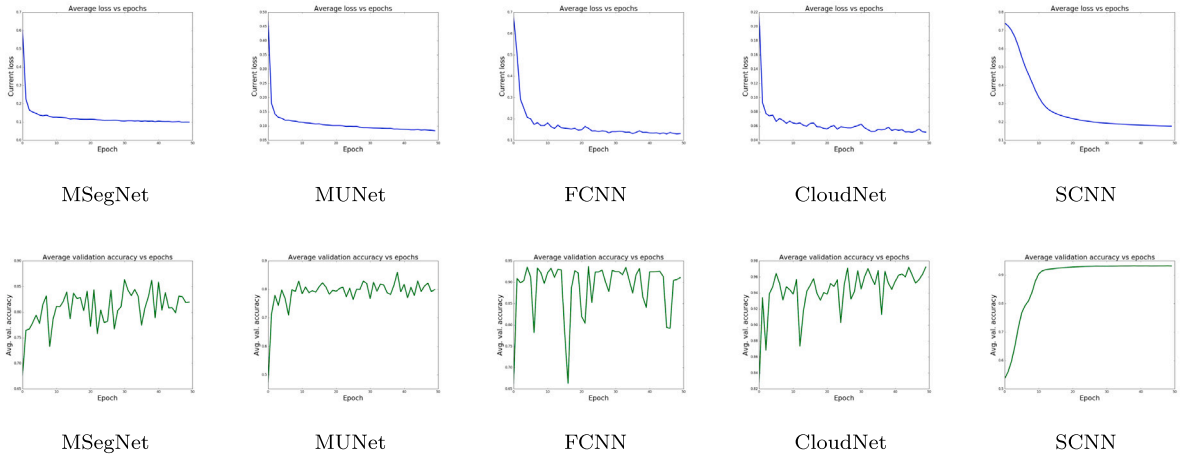

Fig. 13. Loss function and segmentation accuracy versus the number of epochs. Top row illustrates loss curves of DCNNs and SCNN during training. Bottom row illustrates accuracy curves of DCNNs and SCNN during training.

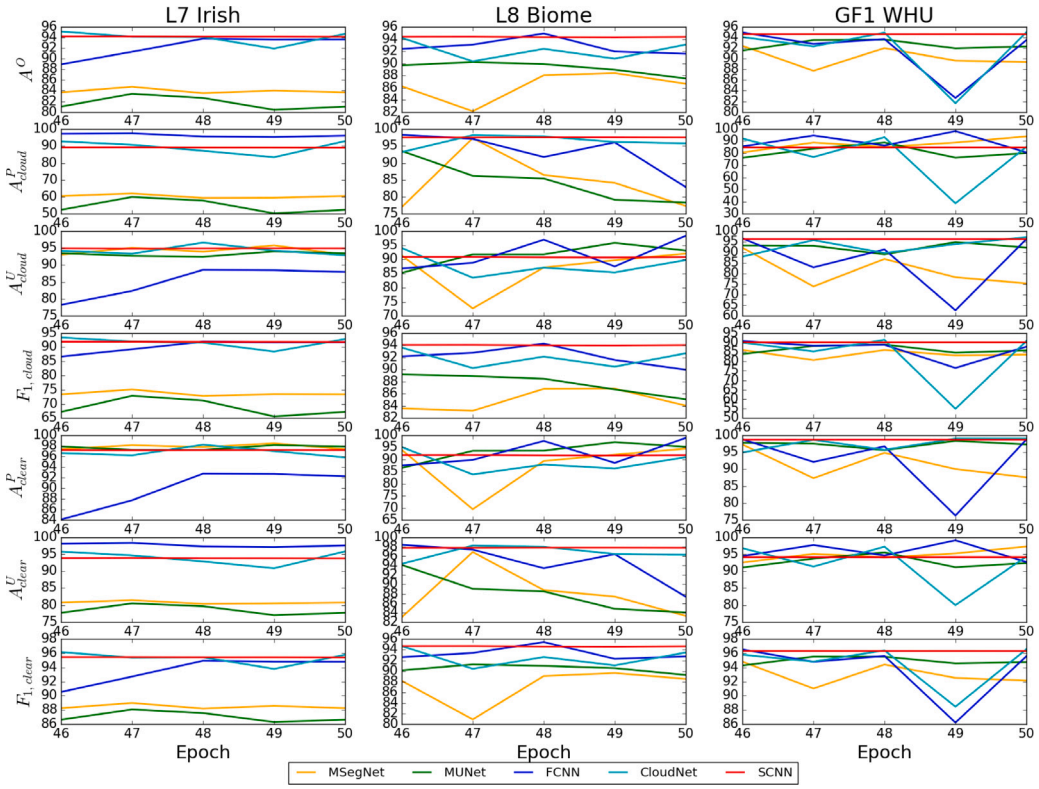

Fig. 14. Quantitative comparison of SCNN with DCNNs based on the 5 models saved at the end of last 5 epochs during model training. From top to bottom, it reports A^O together with A^P_{cloud}, A^U_{cloud}, $F_{1,cloud}$ A^P_{clear}, A^U_{clear}, $F_{1,clear}$.

with CloudNet as its overall accuracy is 1.25% higher on L8 Biome but 0.56% and 0.38% lower on L7 Irish and GF1 WHU respectively. CloudNet performs better than SCNN partly due to random variation, its performances vary at the last 5 epochs as demonstrated in Fig. 14. The accuracies by MSegNet are not as good as reported in Chai et al. (2019) due to three factors. First, 93 L7 Irish scenes and 98 L8 Biome scenes instead of 38 L7 Irish and 32 L8 Biome are employed in this experiment. MSegNet may not perform as expected when it deals with more new images. Second, the test set and training set do not share the same scenes as in the existing methods such as Chai et al. (2019). The training set and test set share the same Landsat image when it is split into a set of 512 × 512 patches since some are grouped into training and some are into test set. It is reasonable that lower accuracies are achieved when the training and test sets do not share the same scenes. Third, the evaluations are based on entire Landsat images instead of 512 × 512 patches. The accuracy may be impaired by some artifacts along the borders of image area as shown by the examples of Figs. 15, 16 and 17.

The DCNNs are expected to outperform the SCNN since hierarchical spatial and spectral features hidden in various images can be exploited by the DCNNs for cloud detection. However, they do not produce better cloud masks as depicted in Figs. 15, 16 and 17 due to the factors discussed in Section 4.3. First, DCNNs produce some artifacts along the borders between image area and filling area. Such artifacts can be easily observed in the label maps by MSegNet and MUNet in Fig. 15, along with the label maps by MSegNet, MUNet and CloudNet

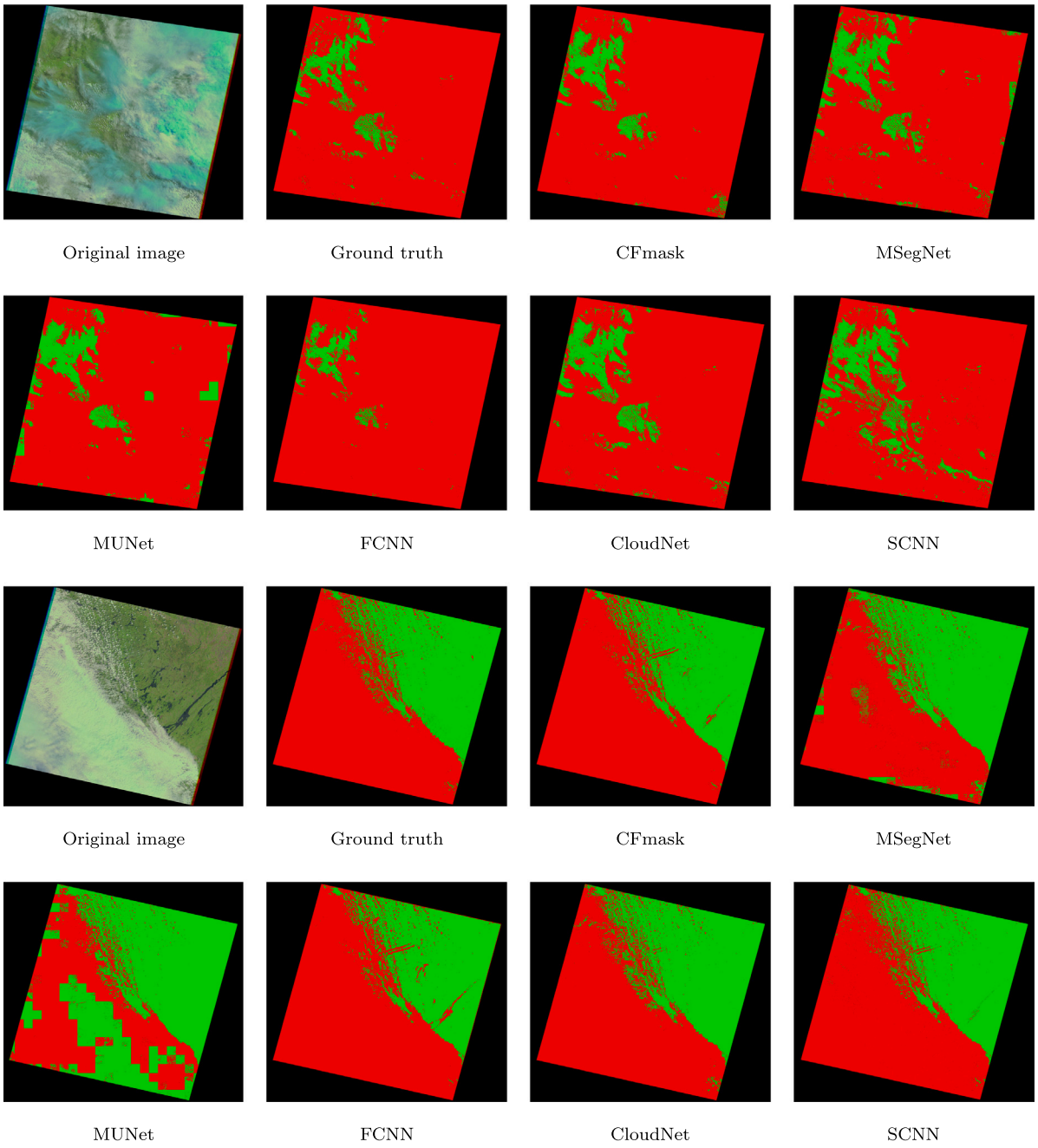

Fig. 15. Cloud detection of Landsat 7 image of a scene (Path 184, Row 055) in tropical zone and a scene (Path 016, Row 029) in middle latitude north zone. The true color composite of RGB bands, its ground truth, baseline result by CFmask algorithm, and the label maps by MSegNet, MUNet, FCNN, CloudNet and SCNN are presented. Cloudy pixels and clear pixels are illustrated as red and green in the label maps. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

in Fig. 17. Second, DCNNs produce obvious incoherence across neighboring patches. Such incoherent patches are clearly illustrated by the straight border across red and green area in the label maps by MSegNet and MUNet in Figs. 15, 16 and 17. Last, it is still hard for DCNNs to distinguish cloud from snow/ice even though they exploit hierarchical spatial and spectral features. In Fig. 16, MSegNet and MUNet classify cloud as snow/ice while as CloudNet classifies snow/ice as cloud. As demonstrated by the snow scene in Figs. 16 and 17, the SCNN exploit local spatial features to successfully distinguish clouds from snow/ice. The overall accuracies achieved by DCNNs are inferior to those by SCNN due to that global spatial features may not outperform local spatial features in cloud detection. As convinced by the performances of Fmask and MFC, cloud detection is dominated by spectral features. When the spectral features are coupled with local spatial features, better performances are achieved by the SCNN.

Table 5 reports the computation time of SCNN and DCNNs in both training and test stage. The time consumed to train the models are recorded for all five models and three datasets. It does not include the time used to evaluate the model on validation set at the end of each epoch in the training stage. The time used to segment a remote sensing image is averaged over the whole test set of each dataset. SCNN training is more than 1000 times faster than DCNN training and SCNN-based segmentation is many times faster than DCNN-based segmentation. Cloud detection using SCNN can also be achieved without the help of GPU acceleration. Without GPU, SCNN can still be trained in 10 min,

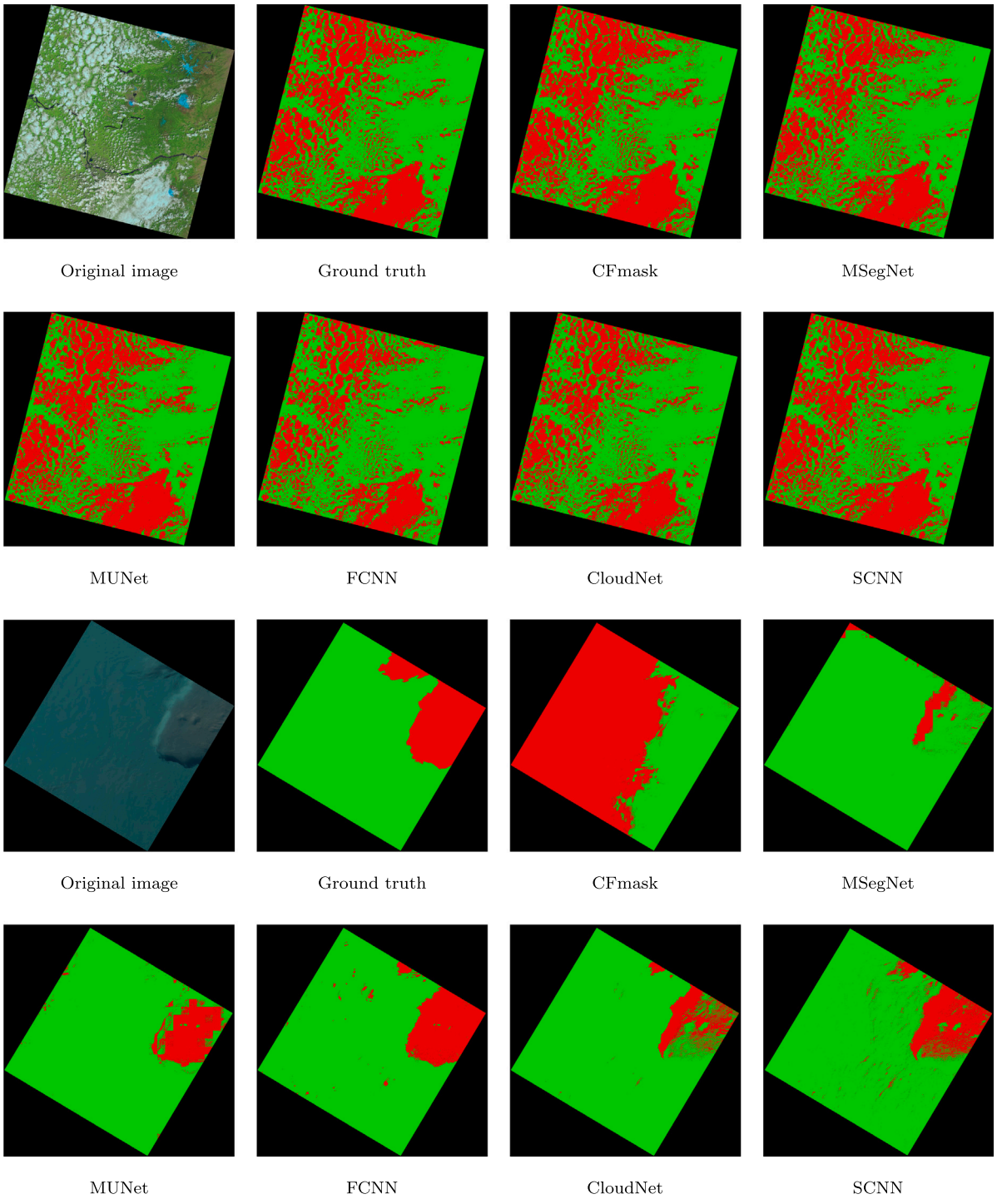

Fig. 16. Cloud detection of Landsat 8 image of an urban scene (Path 046, Row 028) and a snow/ice scene (Path 227, Row 119). The true color composite of RGB bands, its ground truth, baseline result by CFmask algorithm, and the label maps by MSegNet, MUNet, FCNN, CloudNet and SCNN are presented. Cloudy pixels and clear pixels are illustrated as red and green in the label maps.

a Landsat image and a Gaofen-1 WFV image can be segmented in 30 and 70 s respectively.

6. Discussion

This research is a subversive discovery as it reveals that shallow CNNs may outperform deep CNNs, which are regarded as strong tools for many problems and have been applied to computer vision, remote sensing and many other fields. This section discusses spectral vs. spatial features, local vs. global spatial features, shallow vs. deep CNNs for cloud detection and some potential tasks.

The first concerns spectral and spatial features. Both the spectral and spatial features are important for object detection and semantic segmentation. But for some specific problems such as cloud detection, spatial features are auxiliary. As a successful representative of cloud detection algorithm, FMask classifies each pixel using its spectral features. It is reasonable that the SCNN outperforms FMask algorithm since the fixed thresholds for pixel classification are replaced by nonlinear functions learned from real data. This is the key to the success of SCNNs.

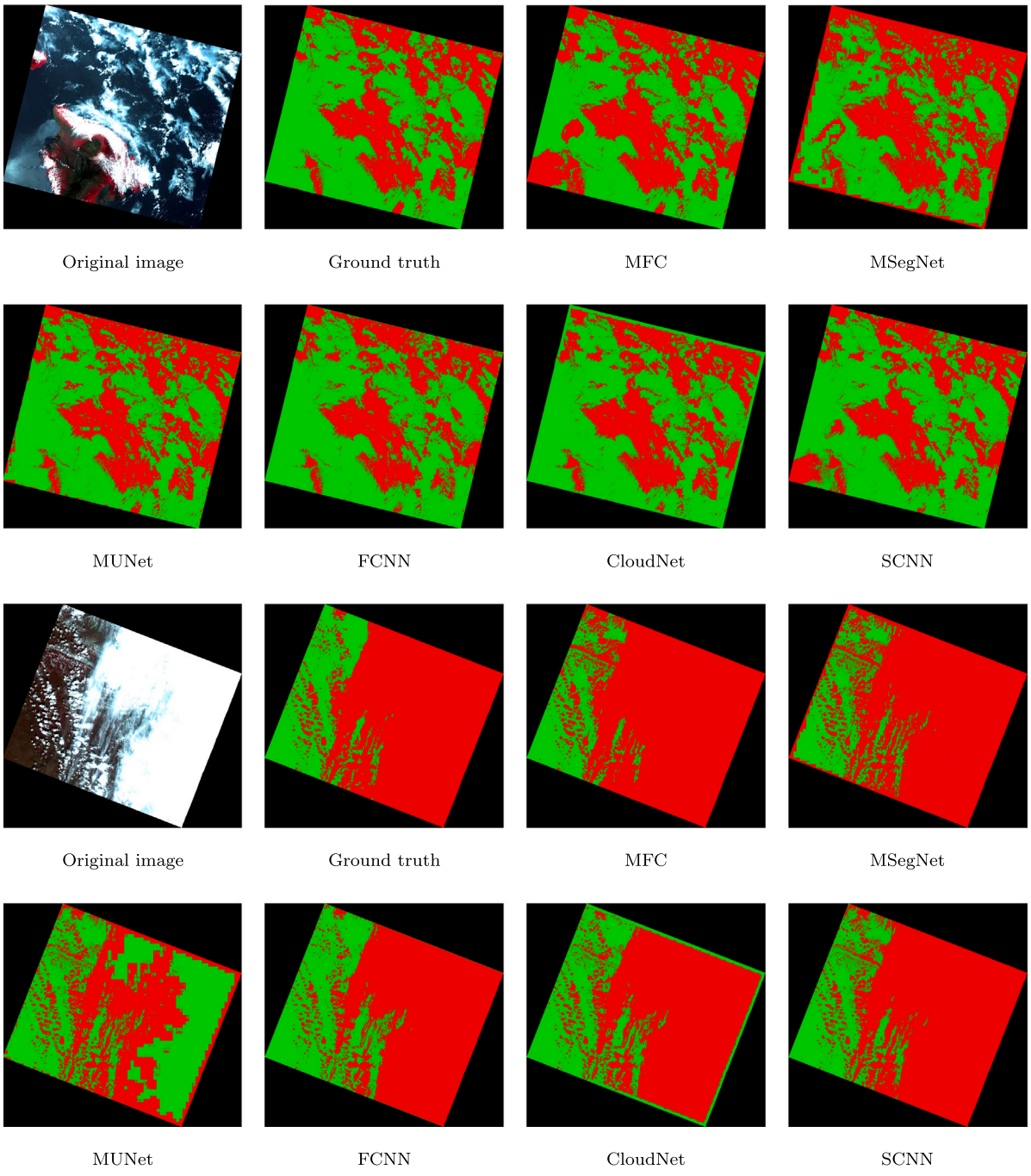

Fig. 17. Cloud detection of Gaofen-1 WFV image of a water scene (W155.2, N20.2) and a snow scene (E132.4, N53.2). The color composite of NIR-R-G bands, its ground truth, baseline result by MFC algorithm, and the label maps by MSegNet, MUNet, FCNN, CloudNet and SCNN are presented. Cloudy pixels and clear pixels are illustrated as red and green in the label maps.

Table 5
Computation time consumed by SCNN and DCNNs. It reports the time (seconds) consumed to train a model and the averaged time to segment a remote sensing image for all the three datasets. First, all experiments are carried out using a GPU. Then, the experiments for SCNN are also carried out without using a GPU.

	Network	Weight	L7 Irish		L8 Biome		GF1 WHU	
			Training	Segmenting	Training	Segmenting	Training	Segmenting
GPU	MSegNet	35M	72K	88	78K	101	390K	379
	MUNet	8M	84K	66	99K	78	495K	294
	CloudNet	36M	60K	45	69K	53	336K	146
	FCNN	96K	60K	19	75K	22	330K	64
	SCNN	680/872/488	59	7	60	14	63	17
CPU	SCNN	680/872/488	449	21	454	28	498	69

SCNNs have potential applications in many other tasks dominated by spectral features.

The second concerns local and global spatial features. Since there are many layers including pooling/unpooling layers in a DCNN, global spatial features can be extracted by a DCNN. In contrast, only local spatial features are extracted by the last convolutional layer of SCNN. Since local spatial features can be utilized to remove noisy pixels resulting from independent pixel classification, the accurate detection using spectral features can be improved by local spatial features. But global spatial features contribute little to cloud detection, which is dominated by spectral features. This is why SCNN outperforms DCNNs.

The third concerns parameter sensitivity. As presented in Section 5.6, the proposed SCNN performs stable with respect to different layer numbers, different filter numbers and different filter sizes. In contrast, the spectral test is very sensitive to the thresholds, which are based on empirical study in advance. Moreover, as revealed by previous studies of different DCNNs for cloud detection, their networks and configurations do have influence on cloud detection.

The fourth concerns advantages of SCNN over DCNNs. The first advantage concerns ground truth annotation, i.e., thousands pixels versus tens remote sensing images. The second advantage concerns computation resource consumption. All training samples for SCNN training are packed into one batch and fed to SCNN once for all while as tens remote sensing images for DCNN training are partitioned into small patches and packed into batches to be processed by DCNNs sequentially. Such difference leads to their different computation time reported in Table 5. The third advantage concerns the stability of model training, which are reflected by the variation of loss curve on training set and accuracy curve on validation set demonstrated in Fig. 13 . The fourth advantage concerns their different performances related with pooling/unpooling and normalization layers as illustrated by Figs. 12, 14, 15, 16 and 17.

Since vegetable detection can be formulated as semantic segmentation problem and it is dominated by spectral values, SCNNs are expected to be better than DCNNs for this problem. SCNN can be used to extract spectral features and local spatial features for problem other than semantic segmentation. SCNN is also applicable to object detection when the targets rely on spectral features and local spatial features. For example, night-time light can be detected using SCNN since it is also dominated by spectral features.

SCNN may also be useful to deal with some data other than images when no spatial features need to be extracted from the data. Such data are very common in the fields of sciences and engineering. Even when SCNNs and DCNNs perform equally in some problems, SCNN are still preferred since they save the cost of ground truth annotation and the time of model training.

7. Conclusion

This paper proposes a shallow CNN for cloud detection in remote sensing image. It promises practical applications since three practical issues concerning ground truth annotation, model training and segmentation performance are well addressed. First, it is much cheaper to annotate a set of pixels for SCNN training than pixel-wisely annotate a set of image for DCNN training. Second, it much faster to train a SCNN than DCNNs. Third, SCNN training is much stabler than DCNN training. Lastly, SCNN outperforms DCNNs or is competitive with DCNNs in cloud detection.

It is straightforward to extend the SCNN in two ways. First, it can be applied to detect clouds in images captured by any other satellites such as Sentinel-2. Such extensions are guaranteed by the fact that cloud detection is dominated by spectral features and convinced by the experiments on Landsat 7, Landsat 8 and Gaofen-1 WFV images. Second, it is also possible to detect some specific land covers that have distinguishing bands and values. For example, NDVI may be leveraged by the SCNN for vegetable detection.

Although clouds were successfully detected by the SCNN, cloud shadow detection is still under investigation. While cloudy pixels can be distinguished from clear pixels based on spectral features and local spatial features, cloud shadows cannot be identified without the contextual information from clouds since cloud shadows depend on clouds. Therefore, another SCNN is required to extract such spatial contextual relationships between clouds and their shadows.

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 Zhejiang Provincial Natural Science Foundation of China (No. LY22D010003), National Natural Science Foundation of China (No. 42230502), State Key Laboratory of Geo-Information Engineering and Key Laboratory of Surveying and Mapping Science and Geospatial Information Technology of MNR, CASM (No. 2021-03-02).

References

Amato, U., Antoniadis, A., Cuomo, V., Cutillo, L., Franzese, M., Murino, L., Serio, C., 2008. Statistical cloud detection from SEVIRI multispectral images. Remote Sens. Environ. 112 (3), 750–766.

Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39 (12), 2481–2495.

Chai, D., Newsam, S., Huang, J., 2020. Aerial image semantic segmentation using DCNN predicted distance maps. ISPRS J. Photogramm. Remote Sens. 161, 309–322. http://dx.doi.org/10.1016/j.isprsjprs.2020.01.023, URL https://www.sciencedirect.com/science/article/pii/S0924271620300290.

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. http://dx.doi.org/10.1016/j.rse.2019.03.007, URL http://www.sciencedirect.com/science/article/pii/S0034425719300987.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L., 2018. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans. Pattern Anal. Mach. Intell. 40 (4), 834–848.

Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A., 2015. The pascal visual object classes challenge: A retrospective. Int. J. Comput. Vis. 111 (1), 98–136.

Foga, S., Scaramuzza, P.L., Guo, S., Zhu, Z., Dilley, R.D., Beckmann, T., Schmidt, G.L., Dwyer, J.L., Hughes, M.J., Laue, B., 2017. Cloud detection algorithm comparison and validation for operational landsat data products. Remote Sens. Environ. 194, 379–390.

Frantz, D., Röder, A., Udelhoven, T., Schmidt, M., 2015. Enhancing the detectability of clouds and their shadows in multitemporal dryland Landsat imagery: extending fmask. IEEE Geosci. Remote Sens. Lett. 12 (6), 1242–1246.

Fulkerson, B., Vedaldi, A., Soatto, S., 2009. Class segmentation and object localization with superpixel neighborhoods. In: 2009 IEEE 12th International Conference on Computer Vision. IEEE, pp. 670–677.

Guo, J., Yang, J., Yue, H., Tan, H., Hou, C., Li, K., 2020. CDnetV2: CNN-based cloud detection for remote sensing imagery with cloud-snow coexistence. IEEE Trans. Geosci. Remote Sens. 59 (1), 700–713.

Hagolle, O., Huc, M., Pascual, D.V., Dedieu, G., 2010. A multi-temporal method for cloud detection, applied to FORMOSAT-2, VENμS, LANDSAT and SENTINEL-2 images. Remote Sens. Environ. 114 (8), 1747–1755.

Hollingsworth, B.V., Chen, L., Reichenbach, S.E., Irish, R.R., 1996. Automated cloud cover assessment for Landsat TM images. In: Imaging Spectrometry II. vol. 2819, International Society for Optics and Photonics, pp. 170–180.

Holschneider, M., Kronland-Martinet, R., Morlet, J., Tchamitchian, P., 1990. A real-time algorithm for signal analysis with the help of the wavelet transform. In: Wavelets. Springer, pp. 286–297.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861.

Hughes, M.J., Hayes, D.J., 2014. Automated detection of cloud and cloud shadow in single-date landsat imagery using neural networks and spatial post-processing. Remote Sens. 6 (6), 4907–4926.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Bach, F., Blei, D. (Eds.), Proceedings of the 32nd International Conference on Machine Learning. In: Proceedings of Machine Learning Research, vol. 37, PMLR, Lille, France, pp. 448–456, URL http://proceedings.mlr.press/v37/ioffe15.html.

Irish, R.R., Barker, J.L., Goward, S.N., Arvidson, T., 2006. Characterization of the Landsat-7 ETM+ automated cloud-cover assessment (ACCA) algorithm. Photogram. Eng. Remote Sens. 72 (10), 1179–1188.

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.

Jin, S., Homer, C., Yang, L., Xian, G., Fry, J., Danielson, P., Townsend, P.A., 2013. Automated cloud and shadow detection and filling using two-date landsat imagery in the USA. Int. J. Remote Sens. 34 (5), 1540–1560.

Ju, J., Roy, D.P., 2008. The availability of cloud-free Landsat ETM+ data over the conterminous United States and globally. Remote Sens. Environ. 112 (3), 1196–1211.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105.

Ladickỳ, L., Sturgess, P., Alahari, K., Russell, C., Torr, P.H., 2010. What, where and how many? combining object detectors and crfs. In: European Conference on Computer Vision. Springer, pp. 424–437.

LeCun, Y., Bottou, L., Orr, G.B., Müller, K.-R., 1998. Efficient backprop. In: Neural Networks: Tricks of the Trade. Springer, pp. 9–50.

Lee, Y., Wahba, G., Ackerman, S.A., 2004. Cloud classification of satellite radiance data by multicategory support vector machines. J. Atmos. Ocean. Technol. 21 (2), 159–169.

Lee, J., Weger, R.C., Sengupta, S.K., Welch, R.M., 1990. A neural network approach to cloud classification. IEEE Trans. Geosci. Remote Sens. 28 (5), 846–855.

Li, Y., Chen, W., Zhang, Y., Tao, C., Xiao, R., Tan, Y., 2020. Accurate cloud detection in high-resolution remote sensing imagery by weakly supervised deep learning. Remote Sens. Environ. 250, 112045.

Li, Z., Shen, H., Cheng, Q., Liu, Y., You, S., He, Z., 2019. Deep learning based cloud detection for medium and high resolution remote sensing images of different sensors. ISPRS J. Photogramm. Remote Sens. 150, 197–212.

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, Z., Shen, H., Weng, Q., Zhang, Y., Dou, P., Zhang, L., 2022a. Cloud and cloud shadow detection for optical satellite imagery: Features, algorithms, validation, and prospects. ISPRS J. Photogramm. Remote Sens. 188, 89–108.

Li, J., Wu, Z., Hu, Z., Jian, C., Luo, S., Mou, L., Zhu, X.X., Molinier, M., 2022b. A lightweight deep learning-based cloud detection method for sentinel-2A imagery fusing multiscale spectral and spatial features. IEEE Trans. Geosci. Remote Sens. 60, 1–19. http://dx.doi.org/10.1109/TGRS.2021.3069641.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 936–944. http://dx.doi.org/10.1109/CVPR.2017.106.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3431–3440.

López-Puigdollers, D., Mateo-García, G., Gómez-Chova, L., 2021. Benchmarking deep learning models for cloud detection in landsat-8 and sentinel-2 images. Remote Sens. 13 (5), 992.

Marmanis, D., Schindler, K., Wegner, J.D., Galliani, S., Datcu, M., Stilla, U., 2018. Classification with an edge: Improving semantic image segmentation with boundary detection. ISPRS J. Photogramm. Remote Sens. 135, 158–172.

Mateo-García, G., Laparra, V., López-Puigdollers, D., Gómez-Chova, L., 2020. Transferring deep learning models for cloud detection between Landsat-8 and Proba-V. ISPRS J. Photogramm. Remote Sens. 160, 1–17.

Mohajerani, S., Saeedi, P., 2019. Cloud-Net: An end-to-end cloud detection algorithm for Landsat 8 imagery. In: IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium. IEEE, pp. 1029–1032.

Noh, H., Hong, S., Han, B., 2015. Learning deconvolution network for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1520–1528.

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. http://dx.doi.org/10.1016/j.rse.2019.05.024, URL https://www.sciencedirect.com/science/article/pii/S0034425719302172.

Ricciardelli, E., Romano, F., Cuomo, V., 2008. Physical and statistical approaches for cloud identification using meteosat second generation-spinning enhanced visible and infrared imager data. Remote Sens. Environ. 112 (6), 2741–2760.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, pp. 234–241.

Roy, D.P., Ju, J., Kline, K., Scaramuzza, P.L., Kovalskyy, V., Hansen, M., Loveland, T.R., Vermote, E., Zhang, C., 2010. Web-enabled Landsat Data (WELD): Landsat ETM+ composited mosaics of the conterminous United States. Remote Sens. Environ. 114 (1), 35–49.

Scaramuzza, P.L., Bouchard, M.A., Dwyer, J.L., 2012. Development of the Landsat data continuity mission cloud-cover assessment algorithms. IEEE Trans. Geosci. Remote Sens. 50 (4), 1140–1154.

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.

Shotton, J., Johnson, M., Cipolla, R., 2008. Semantic texton forests for image categorization and segmentation. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 1–8.

Shotton, J., Winn, J., Rother, C., Criminisi, A., 2009. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. Int. J. Comput. Vis. 81 (1), 2–23.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Skakun, S., Wevers, J., Brockmann, C., Doxani, G., Aleksandrov, M., Batič, M., Frantz, D., Gascon, F., Gómez-Chova, L., Hagolle, O., et al., 2022. Cloud Mask Intercomparison eXercise (CMIX): An evaluation of cloud masking algorithms for Landsat 8 and Sentinel-2. Remote Sens. Environ. 274, 112990.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15 (1), 1929–1958.

Sun, L., Liu, X., Yang, Y., Chen, T., Wang, Q., Zhou, X., 2018. A cloud shadow detection method combined with cloud height iteration and spectral analysis for Landsat 8 OLI data. ISPRS J. Photogramm. Remote Sens. 138, 193–207.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–9.

Tian, B., Shaikh, M.A., Azimi-Sadjadi, M.R., Haar, T.H.V., Reinke, D.L., 1999. A study of cloud classification with neural networks using spectral and textural features. IEEE Trans. Neural Netw. 10 (1), 138–151.

Tieleman, T., Hinton, G., 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Netw. Mach. Learn. 4 (2), 26–31.

Vermote, E., Justice, C., Claverie, M., Franch, B., 2016. Preliminary analysis of the performance of the Landsat 8/OLI land surface reflectance product. Remote Sens. Environ. 185, 46–56.

Wang, B., Ono, A., Muramatsu, K., Fujiwara, N., 1999. Automated detection and removal of clouds and their shadows from Landsat TM images. IEICE Trans. Inf. Syst. 82 (2), 453–460.

Wei, J., Huang, W., Li, Z., Sun, L., Zhu, X., Yuan, Q., Liu, L., Cribb, M., 2020. Cloud detection for Landsat imagery by combining the random forest and superpixels extracted via energy-driven sampling segmentation approaches. Remote Sens. Environ. 248, 112005. http://dx.doi.org/10.1016/j.rse.2020.112005, URL https://www.sciencedirect.com/science/article/pii/S0034425720303758.

Xie, F., Shi, M., Shi, Z., Yin, J., Zhao, D., 2017. Multilevel cloud detection in remote sensing images based on deep learning. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 10 (8), 3631–3640.

Yao, X., Guo, Q., Li, A., 2021. Light-weight cloud detection network for optical remote sensing images with attention-based DeeplabV3+ architecture. Remote Sens. 13 (18), http://dx.doi.org/10.3390/rs13183617, URL https://www.mdpi.com/2072-4292/13/18/3617.

Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J., 2017. Pyramid scene parsing network. In: IEEE Conf. on Computer Vision and Pattern Recognition. CVPR, pp. 2881–2890.

Zhu, X., Helmer, E.H., 2018. An automatic method for screening clouds and cloud shadows in optical satellite image time series in cloudy regions. Remote Sens. Environ. 214, 135–153.

Zhu, Z., Woodcock, C.E., 2012. Object-based cloud and cloud shadow detection in Landsat imagery. Remote Sens. Environ. 118, 83–94.

Zhu, Z., Woodcock, C.E., 2014. Automated cloud, cloud shadow, and snow detection in multitemporal Landsat data: An algorithm designed specifically for monitoring land cover change. Remote Sens. Environ. 152, 217–234.

Zi, Y., Xie, F., Jiang, Z., 2018. A cloud detection method for landsat 8 images based on PCANet. Remote Sens. 10 (6), 877.