Lightweight U-Net for cloud detection of visible and thermal infrared remote sensing images

Jiaqiang Zhang[1,3,4] · Xiaoyan Li[1,2,3,4] · Liyuan Li[1,3,4] · Pengcheng Sun[3,5] · Xiaofeng Su[1,3] · Tingliang Hu[1,3] · Fansheng Chen[1,2,3]

Abstract

Accurate and rapid cloud detection is exceedingly significant for improving the downlink efficiency of on-orbit data, especially for the microsatellites with limited power and computational ability. However, the inference speed and large model limit the potential of on-orbit implementation of deep-learning-based cloud detection method. In view of the above problems, this paper proposes a lightweight network based on depthwise separable convolutions to reduce the size of model and computational cost of pixel-wise cloud detection methods. The network achieves lightweight end-to-end cloud detection through extracting feature maps from the images to generate the mask with the obtained maps. For the visible and thermal infrared bands of the Landsat 8 cloud cover assessment validation dataset, the experimental results show that the pixel accuracy of the proposed method for cloud detection is higher than 90%, the inference speed is about 5 times faster than that of U-Net, and the model parameters and floating-point operations are reduced to 12.4% and 12.8% of U-Net, respectively.

Keywords Fully convolutional network · Depthwise separable convolution · Cloud detection · Semantic segmentation · Lightweight network

1 Introduction

According to the results of the International Satellite Cloud Climatology Project (ISCCP), clouds cover two-thirds of the land surface on earth (Rossow and Schiffer 1991), and high cloud coverage will reduce the accuracy and application of remote sensing data, resulting

Jiaqiang Zhang and Xiaoyan Li have contributed equally to this work.

✉ Xiaoyan Li
xiaoyanLee2018@163.com

✉ Xiaofeng Su
fishsu@mail.sitp.ac.cn

✉ Fansheng Chen
cfs@mail.sitp.ac.cn

Extended author information available on the last page of the article

in changes in the texture and spectral information of remote sensing images and affecting the radiation correction, geometric calibration and distortion correction (Li et al. 2019). Since the time window for satellite data downlink transmission is only 5 to 10 min, cloud detection preprocessing of the data before transmission can improve the quality and efficiency of the remote sensing images (Williams et al. 2002). With the continuous development of remote sensing technology, miniaturization of satellites has limited the computing power and power consumption of electronic systems. Therefore, a lightweight cloud detection model with low computing power requirements is necessary.

Spectral analysis method is widely used in cloud detection. Radiance and reflectance of cloud in visible, short-wave infrared, thermal infrared and other wave bands are used for cloud detection through threshold method. Irish et al. (2006) proposed a cloud detection algorithm called Automatic Cloud Cover Assessment (ACCA) for Landsat 7 images, which used cloud reflectance in different wavebands to implement cloud detection. Li and Li (2011) converted the pseudo-color composite images of bands 1, 6, and 26 of the MODIS images into HSV, and used the H channel as threshold to obtain cloud detection results. Zhu and Woodcock (2012) used the physical properties of the cloud to obtain potential cloud pixels, and then combined the normalized temperature, spectral variability, and brightness probabilities to further determine the potential cloud pixels as cloud or non-cloud. It is hard to detect clouds using limited spectral information of small satellites (Li et al. 2017). Therefore, spectral analysis method cannot show satisfactory performance on small satellites.

Since 2016, deep learning has been applied to cloud detection research (Mateo-García et al. 2017). Shi et al. (2016) combined superpixels with convolutional neural networks to complete cloud detection of Quickbird data. Zi et al. (2018) used the SLIC algorithm to extract superpixels from Landsat 8 band 6, 3, 2 images, then input the superpixel patch into a double-branch PCANet to determine whether the superpixel patch was cloud or non-cloud, and finally applied a conditional random field for post-processing. Chen et al. (2018) applied multi-convolutional neural networks to classify the superpixels generated by the adaptive SLIC algorithm to complete cloud detection. Although deep learning networks combined with superpixels can obtain a good accuracy for cloud detection, the methods mentioned above are difficult to be implemented on-orbit and cannot complete pixel-wise cloud detection.

Semantic segmentation was introduced into cloud detection research in 2018 and became a hot spot in current research. Wieland et al. (2019) applied U-Net (Ronneberger et al. 2015) to the data of Landsat and Sentinel 2 to complete the pixel-wise segmentation of cloud and cloud shadow, with an accuracy of 89%. Gao et al. (2019) applied the neural network based on dilated convolution to complete the end-to-end cloud detection of ZY-3 image. Li et al. (2019) proposed the MSCFF network, which can detect clouds on images of different sensors with a resolution of 0.5 to 50 m. Although the cloud detection methods mentioned above can complete pixel-wise end-to-end cloud detection with a good result, the size and the computational cost of the networks cannot meet the demands of small satellites with limited computing power and power consumption.

In this paper, we propose a lightweight network based on U-Net to reduce the size of model and computational cost of the pixel-wise cloud detection method and improve the portability of the cloud detection method on embedded platforms. The proposed method improves the pixel accuracy of the visible bands on the Landsat 8 cloud cover assessment (L8 CCA) validation dataset, and greatly reduces the size of the network, computing power requirements and inference time. Further experiments validated that the method is also suitable for cloud detection in the thermal infrared band.

2 Methodology

2.1 U-Net based on depthwise separable convolutions

As shown in Fig. 1, the process of the proposed method consists of two parts: training and testing. In this paper, the L8 CCA data set is divided into training set, validation set, and test set. In the training phase, the images with the corresponding ground truth of the training set and the validation set are input into the network, and the model is trained by a backpropagation (BP) algorithm to obtain the parameters with minimal loss. In the testing phase, the model outputs the probability tensor of the test set images, and the cloud detection result is obtained by the ARGMAX function.

The network used in this paper is a fully convolutional network based on an encoding–decoding architecture improved by depthwise separable convolutions (Howard et al. 2017), as shown in Fig. 2. The input data is processed by encoders based on depthwise separable convolution to generate high-dimensional feature maps of different scales; then, the decoders continuously upsample the feature maps and concatenate the feature maps of encoders and decoders with same scale; finally, the network outputs a tensor containing the classification probability of each pixel, and the class with the highest probability is used as the label of the pixel.

U-Net (Ronneberger et al. 2015) is originally used for segmentation of biomedical images. With the characteristics of simple model, easy to train, and more suitable for small data sets, U-Net is currently widely used in medical diagnosis, remote sensing and other fields. The convolutional layers in U-Net are calculated using standard convolutions which are the most important component of convolutional neural networks. The operation of standard convolution can be divided into two parts. First, the input feature maps are convolution filtered, and then the convolution filtered results are combined into output feature maps. Figure 3a shows the process of standard convolutions. Suppose that a feature map with a size of $H \times W$ and a channel number of M is input into a standard convolution, and after the calculation of N filters with a kernel size of $K \times K$, a feature map with a size of $H \times W$ and a channel number of N is output. Then the parameters of standard convolution are
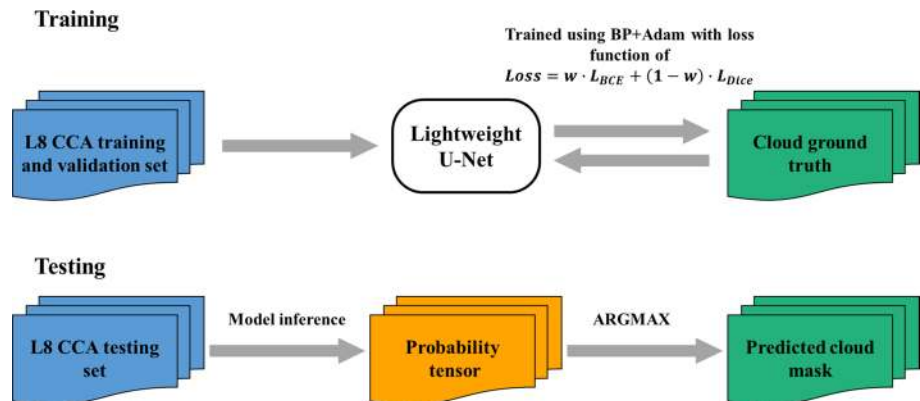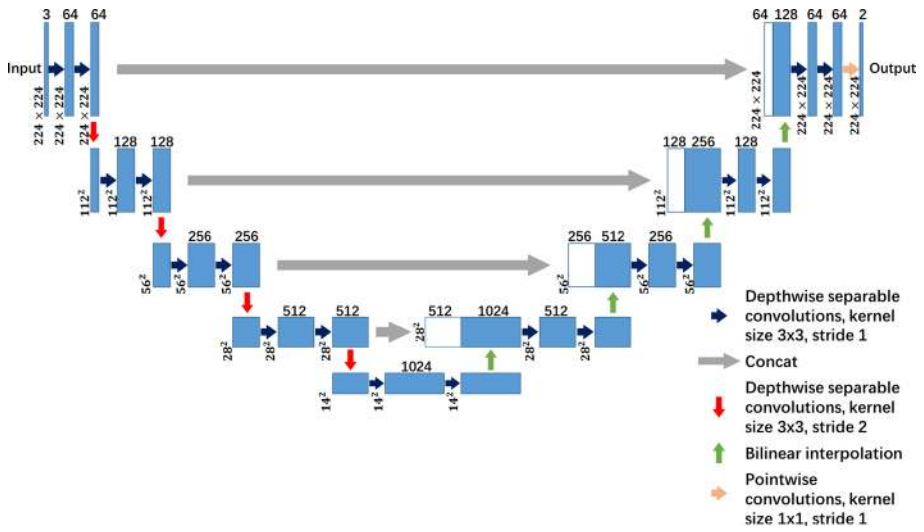

Fig. 1 Flowchart of the proposed method

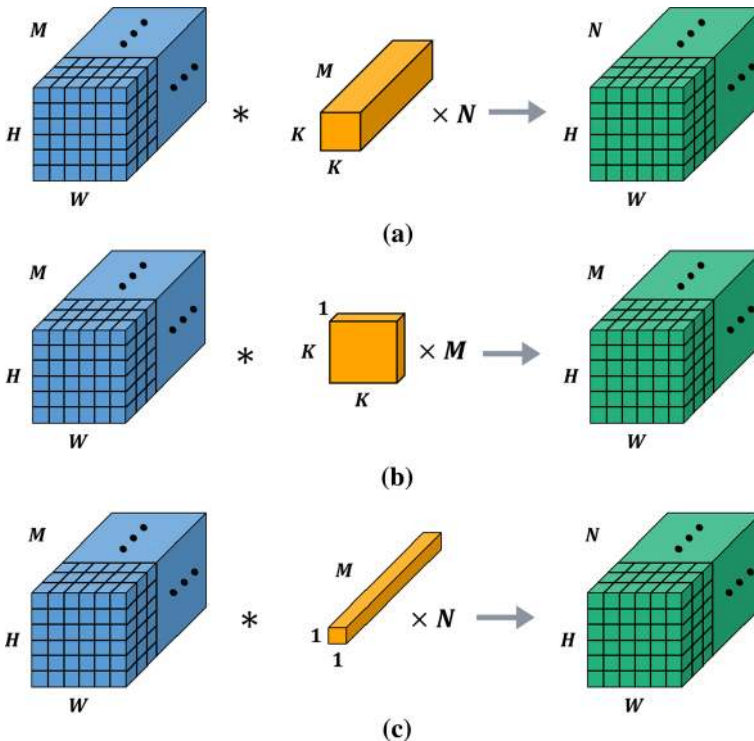**Fig. 2** Architecture of U-Net based on depthwise separable convolution

Fig. 3 Process of convolutions. **a** Standard convolution; **b** depthwise convolution; **c** pointwise convolution

$$K \cdot K \cdot M \cdot N \tag{1}$$

The computational cost is

$$H \cdot W \cdot K \cdot K \cdot M \cdot N \tag{2}$$

Depthwise separable convolution divides standard convolution into depthwise convolution and pointwise convolution to complete filtering and channel combination respectively (Howard et al. 2017). As shown in Fig. 3b, depthwise convolution with a $K \times K$ kernel processes the input feature map of size $H \times W \times M$ into a feature map of M channels, then the parameters of depthwise convolution is

$$K \cdot K \cdot M \tag{3}$$

The computational cost is

$$H \cdot W \cdot K \cdot K \cdot M \tag{4}$$

The parameters and cost are $1/N$ of the standard convolution.

Figure 3c shows the process of pointwise convolutions. The pointwise convolution is a standard convolution with a kernel size of 1×1, which is mainly used to change the dimension of the output feature map of the depthwise convolution in order to complete the combination operation of the standard convolution. The parameters of pointwise convolution are

$$M \cdot N \tag{5}$$

The computational cost is

$$H \cdot W \cdot M \cdot N \tag{6}$$

The computational cost of depthwise separable convolution is

$$H \cdot W \cdot K \cdot K \cdot M + H \cdot W \cdot M \cdot N \tag{7}$$

which is the sum of depthwise convolution and pointwise convolution. According to the computational cost of standard convolution shown in Eq. (2), the relationship between the computational cost of the depthwise separable convolution and the standard convolution is:

$$\frac{H \cdot W \cdot K \cdot K \cdot M + H \cdot W \cdot M \cdot N}{H \cdot W \cdot K \cdot K \cdot M \cdot N} = \frac{1}{N} + \frac{1}{K^2} \tag{8}$$

The size of the convolution kernel used in this article is 3×3, therefore the computational cost of the depthwise separable convolution is about one-ninth to one-eighth of the standard convolution. By breaking the interaction between the number of output channels and the size of the kernel with depthwise separable convolution, the parameters and computational cost are greatly reduced.

In order to prevent the gradient disappearing during training, a batch normalization layer and a ReLU are added after each depthwise convolution and pointwise convolution. The stride of the depthwise convolution can be 1 or 2, and the stride of the pointwise convolution is fixed at 1.

The optimization of U-Net in this article includes two aspects: replacement of standard convolutions and modification of sampling method. U-Net is composed of four encoders and four decoders between which are the corresponding skip connections. The network includes 18 layers of 3×3-kernel-sized convolution operations, and there is much room for

improvement in the size of model and computational cost. As mentioned above, depthwise separable convolutions can reduce the parameters and computational cost of standard convolution, so replacing the standard convolution with depthwise separable convolutions can reduce the size of model and floating-point operations(FLOPs) of U-Net. Sampling methods include upsampling and downsampling. According to Springenberg et al. (2014), max-pooling does not always improve the performance of deep neural networks, and the use of convolutions with stride instead of pooling helps reduce the information loss of feature maps caused by downsampling. Therefore, this paper replaces the downsampling method of the encoders in U-Net from the max-pooling layer to a depthwise separable convolution with a kernel size of 3×3 and a stride of 2. The upsampling method uses bilinear interpolation which is easier to train than the deconvolution in U-Net. The comparison between the proposed method and U-Net is shown in Table 1. The parameters of this method are 3.9 Million, and the FLOPs are 11,002.5 Million, which are 12.4% and 12.8% of U-Net, respectively.

The proposed network is still an encoder-decoder architecture, consisting of four encoders and four decoders. The encoders consist of two depthwise separable convolutions with kernel size of 3×3 and stride of 1 and one depthwise separable convolutions with kernel size of 3×3 and a stride of 2. The decoders consist of one layer of bilinear interpolation and two depthwise separable convolutions with kernel size of 3×3 and stride of 1. The skip connections between encoders and decoders enable the network to combine the high-level and low-level semantic features extracted from the image to improve the segmentation effect (Long et al. 2015). The structure of the proposed method is shown in Fig. 2, and the specific parameters are shown in Table 2.

2.2 Metrics

Metrics used in this paper are pixel accuracy (PA), mean pixel accuracy (mPA) and mean intersection over union (mIoU) (Garcia-Garcia et al. 2017). PA is the ratio of correctly classified pixels and all pixels. mPA is the mean value of PA of all classes. mIoU is the mean value of IoU of all classes, where IoU is the ratio of the intersection and union of the prediction and the ground truth. The metrics can be solved by the confusion matrix shown in Table 3, where TP, FP, FN, and TN stand for True Positive, False Positive, False Negative, and True Negative.

Therefore, the formulas of PA, mPA, and mIoU are as follow:

$$PA = \frac{TP + TN}{TP + FN + FP + TN} \tag{9}$$

$$mPA = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{FP + TN}\right) \tag{10}$$

$$mIoU = \frac{1}{2}\left(\frac{TP}{TP + FN + FP} + \frac{TN}{FP + FN + TN}\right) \tag{11}$$

Table 1 Algorithm efficiency comparison of proposed method and U-Net

	Parameters/million	FLOPs/million
U-Net	31.4	85,683.9
Proposed method	3.9	11,002.5

Table 2 Parameters of U-Net based on depthwise separable convolution

Layer name	Input size	Information of blocks	Output size
Encoder1	$224 \times 224 \times 3$	$\begin{bmatrix} ConvDwSep(3 \times 3, s1) \\ ConvDwSep(3 \times 3, s1) \\ ConvDwSep(3 \times 3, s2) \end{bmatrix}$	$112 \times 112 \times 64$
Encoder2	$112 \times 112 \times 64$	$\begin{bmatrix} ConvDwSep(3 \times 3, s1) \\ ConvDwSep(3 \times 3, s1) \\ ConvDwSep(3 \times 3, s2) \end{bmatrix}$	$56 \times 56 \times 128$
Encoder3	$56 \times 56 \times 128$	$\begin{bmatrix} ConvDwSep(3 \times 3, s1) \\ ConvDwSep(3 \times 3, s1) \\ ConvDwSep(3 \times 3, s2) \end{bmatrix}$	$28 \times 28 \times 256$
Encoder4	$28 \times 28 \times 256$	$\begin{bmatrix} ConvDwSep(3 \times 3, s1) \\ ConvDwSep(3 \times 3, s1) \\ ConvDwSep(3 \times 3, s2) \end{bmatrix}$	$14 \times 14 \times 512$
Bridge	$14 \times 14 \times 512$	$\begin{bmatrix} ConvDwSep(3 \times 3, s1) \\ ConvDwSep(3 \times 3, s1) \end{bmatrix}$	$14 \times 14 \times 1024$
Decoder1	$14 \times 14 \times 1024$	$\begin{bmatrix} UpSampling, Concat \\ ConvDwSep(3 \times 3, s1) \\ ConvDwSep(3 \times 3, s1) \end{bmatrix}$	$28 \times 28 \times 256$
Decoder2	$28 \times 28 \times 256$	$\begin{bmatrix} UpSampling, Concat \\ ConvDwSep(3 \times 3, s1) \\ ConvDwSep(3 \times 3, s1) \end{bmatrix}$	$56 \times 56 \times 128$
Decoder3	$56 \times 56 \times 128$	$\begin{bmatrix} UpSampling, Concat \\ ConvDwSep(3 \times 3, s1) \\ ConvDwSep(3 \times 3, s1) \end{bmatrix}$	$112 \times 112 \times 64$
Decoder4	$112 \times 112 \times 64$	$\begin{bmatrix} UpSampling, Concat \\ ConvDwSep(3 \times 3, s1) \\ ConvDwSep(3 \times 3, s1) \end{bmatrix}$	$224 \times 224 \times 64$
Output	$224 \times 224 \times 64$	$Conv(1 \times 1)$	$224 \times 224 \times 2$

Table 3 Confusion matrix

Ground truth	Predicted class	
	Cloud	Non cloud
Cloud	TP	FN
Non cloud	FP	TN

Table 4 Experimental environment

CPU	Memory	GPU	Operating system
E5-2620 v4	32 GB	GeForce 1080 Ti with 11 GB memory	Ubuntu 16.04

3 Experimental results and discussion

3.1 Experimental images and parameter setting

In this paper, the visible bands 4, 3, and 2 of the Landsat 8 Cloud Cover Assignment (L8 CCA) dataset (Foga et al. 2017) are used as RGB channels to merge true color images. 72 images from Landsat 8 CCA were selected as the data set and processed into 28,800 patches with size of 224×224, 24,000 of which were used as the training set, and 2400 of which were used as the validation set. 6 images of 4480×4480 size were used as test set. The same process is performed on band 11 in the L8 CCA data set to verify the applicability of this method in the thermal infrared band.

The experimental environment of this study is shown in Table 4. All methods are implemented using Python. The deep learning framework, PyTorch (Paszke et al. 2017), is used for model training and testing. The training epoch is 70. Adam (Kingma and Ba 2014) is used as optimizer. Batch size is set to 16. The initial value of the learning rate is 0.001, and it is updated every 10 epochs with a decrease factor of 0.5. As shown in the Eq. (12), the loss function is the weighted average of the Dice loss (Milletari et al. 2016) and the Binary Cross Entropy (Zhang and Sabuncu 2018). The weight of the Binary Cross Entropy is 0.8. Dice loss and Binary Cross Entropy are shown in Eqs. (13) and (14), respectively, where P is the prediction result and T is the ground truth.

$$L = w \cdot L_{Bce} + (1 - w) \cdot L_{Dice} \tag{12}$$

$$L_{Dice} = 1 - \frac{2|P \cap T|}{|P| + |T|} \tag{13}$$

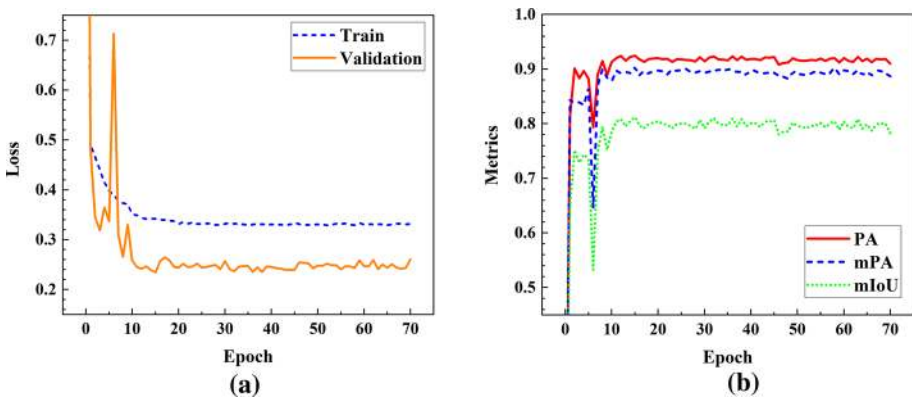$$L_{Bce} = -\sum_i T_i \ln(P_i) + (1 - T_i) \ln(1 - P_i) \tag{14}$$

Fig. 4 Training procedure. **a** Loss change curves versus the number of epochs; **b** metrics change curves versus the number of epochs
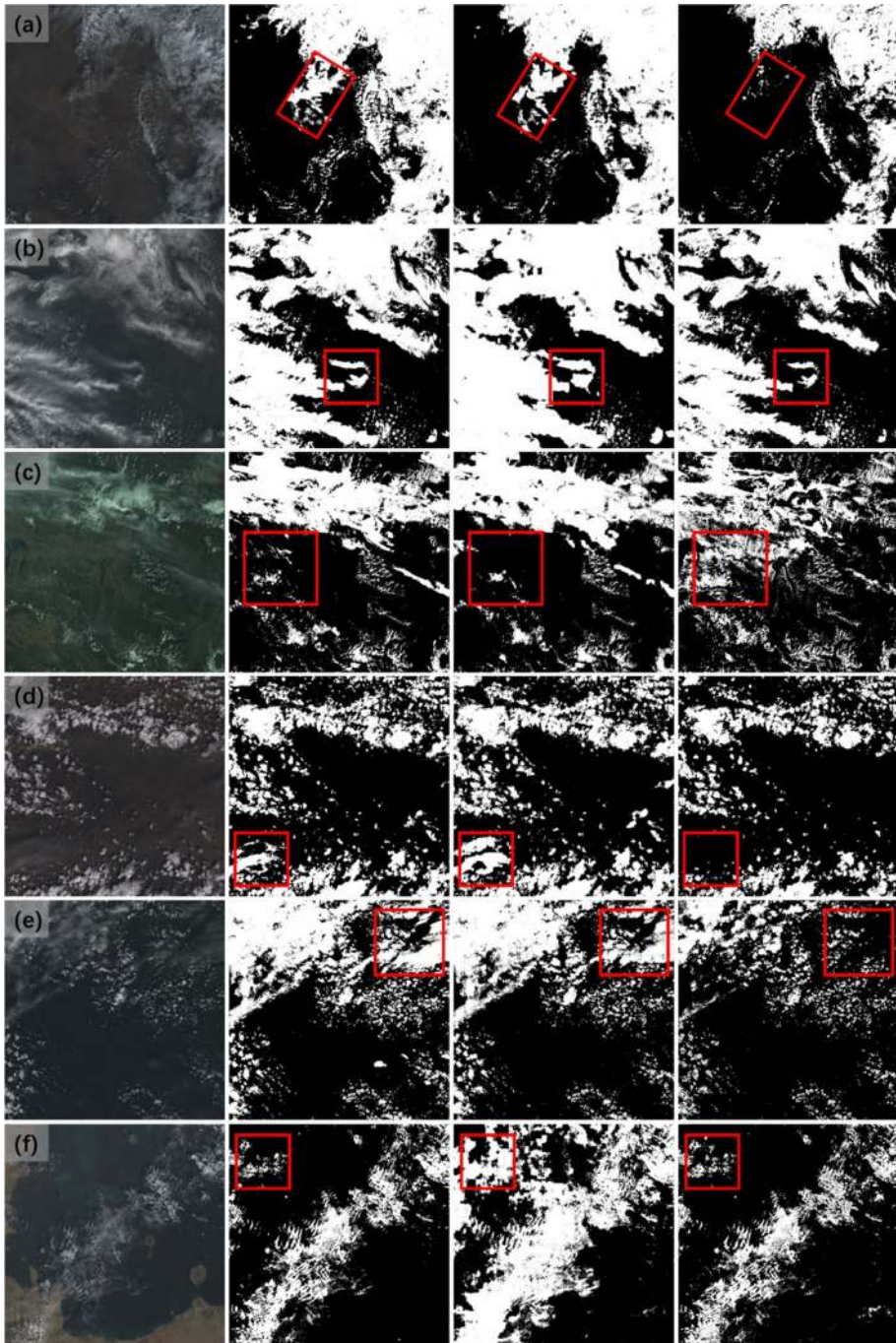
Fig. 5 Examples of cloud detection in band 2, 3, and 4. From top to bottom, the examples are from scenes of LC81590362014051LGN00, LC80310202013223LGN00, LC81320352013243LGN00, LC81390292014135LGN00, LC80630152013207LGN00 and LC81620432014072LGN00. From left to right are the true-color input images, the ground truth, the results of our method, and the results of U-Net

Fig. 6 Examples of cloud detection in thermal infrared. From top to bottom, the examples are from ►
scenes of LC81590362014051LGN00, LC80310202013223LGN00, LC81320352013243LGN00,
LC81390292014135LGN00, LC80630152013207LGN00 and LC81620432014072LGN00. From left to
right are the Band 11 input images, the ground truth, and the results of our method

Table 5 Results comparison of proposed method and U-Net in band 2, 3, and 4

Methods	PA/%	mPA/%	mIoU/%	Inference time/(s/patch)
U-Net	88.17	83.86	75.44	10.25
Proposed method	90.54	91.36	81.53	2.18

Table 6 Detail results of test set in band 2, 3, and 4

Scene ID	Detection algorithm	PA/%	mPA/%	mIoU/%
LC81590362014051LGN00	U-Net	83.34	80.47	69.21
	Proposed	95.00	94.46	90.20
LC80310202013223LGN00	U-Net	92.58	92.87	86.19
	Proposed	88.40	87.85	78.81
LC81320352013243LGN00	U-Net	82.51	78.47	65.92
	Proposed	93.21	92.06	85.18
LC81390292014135LGN00	U-Net	92.73	86.98	82.15
	Proposed	94.50	95.06	87.53
LC80630152013207LGN00	U-Net	82.30	73.15	62.71
	Proposed	94.81	92.81	88.69
LC81620432014072LGN00	U-Net	95.54	91.24	86.45
	Proposed	77.35	85.91	58.79

3.2 Experimental results and analysis

The loss and metrics changes of proposed method within 70 epochs are shown in Fig. 4, where Fig. 4a is the loss change of the network on the training set and the validation set, and Fig. 4b is the metrics change on validation set. It can be seen from the curve in the figure that the loss and metrics fluctuate greatly before epoch 20, and gradually smooth after epoch 30. The loss gradually decreases and stabilizes with the increase of the epoch, and the metrics gradually increases and stabilizes with the increase of the epoch. This change trend shows that the model gradually converges. The model of epoch 40 with the highest metrics in the validation set after convergence is chosen to test the network.
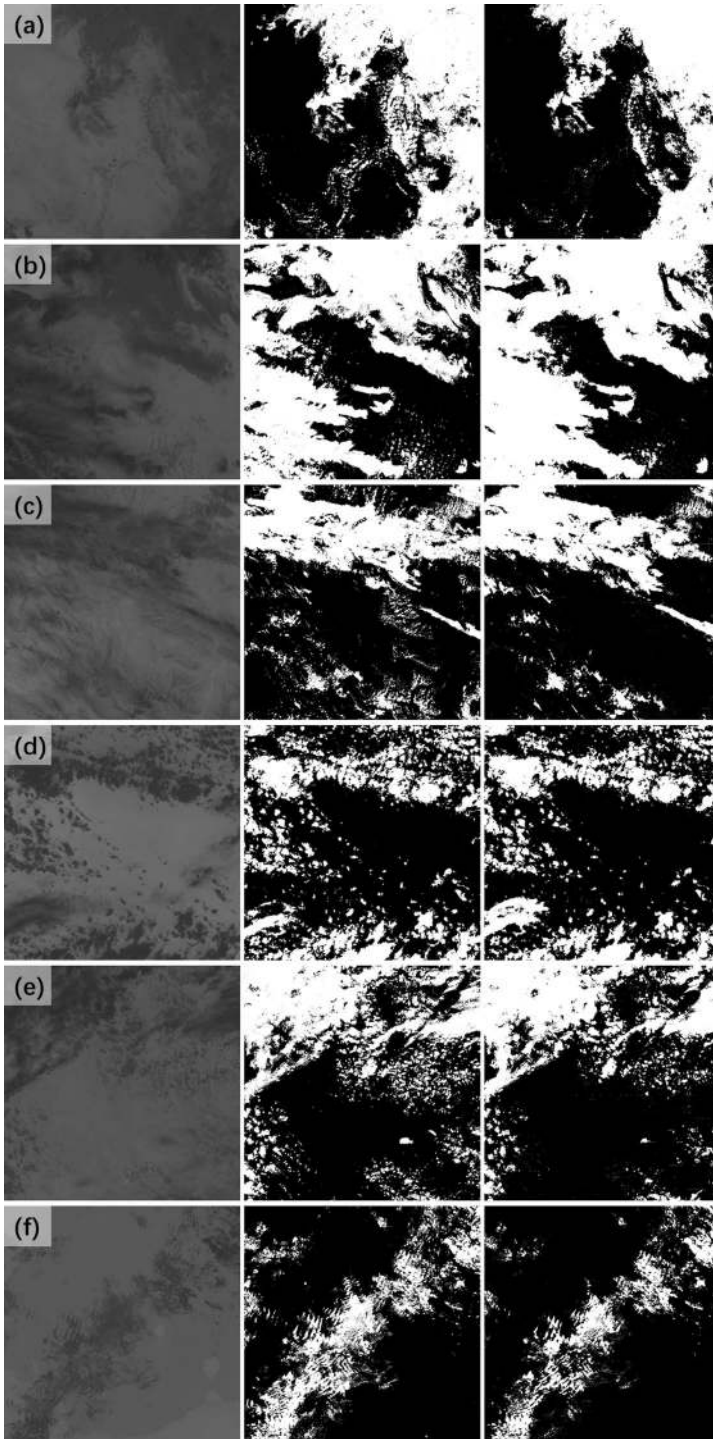
Table 7 Detail results of test set in band 11

Scene ID	PA/%	mPA/%	mIoU/%
LC80630152013207LGN00	91.69	88.86	82.42
LC80310202013223LGN00	88.86	88.37	79.62
LC81320352013243LGN00	89.15	85.09	76.66
LC81390292014135LGN00	93.74	91.90	85.43
LC81590362014051LGN00	91.51	90.39	83.75
LC81620432014072LGN00	91.00	79.56	73.07
Average	90.99	87.36	80.16

Figure 5 shows the cloud detection results of the proposed method and U-Net. It can be seen that the overall results of the proposed method are better than U-Net. In the boxed areas in Fig. 5a, d, e, the detection results of the proposed method are more complete, while U-Net does not detect clouds in these areas. In the boxed area of Fig. 5c, the proposed method misses some broken clouds, and U-Net has a larger range of misclassification. The result of U-Net in the thin cloud box area in Fig. 5b is better than the proposed method. Although the result of proposed method completely covers the true value in that area, the non-cloud pixels around the cloud are misclassified as cloud. The proposed method has a large area of misclassification in the box and the surrounding area of Fig. 5f. This problem may be due to the existence of underwater reefs in this area in the original image. Some features of these reefs are similar to thin clouds. The proposed method is more sensitive than U-Net in thin cloud area, therefore misclassifies these reefs as thin cloud.

To objectively evaluate the cloud detection results of the proposed method, the trained model is used to infer the test set images. Table 5 shows the average results of the test set and its comparison with the U-Net results, where the inference time is the time it takes the algorithm to complete the cloud detection of a patch of 224×224 on the CPU. Table 6 shows the detailed results of the images in the test set. The experimental results show that the overall pixel accuracy of proposed method is 90.54%, which is 2.37% higher than U-Net's 88.17%; the mPA is 91.36%, which is 7.5% higher than U-Net; the mIoU is 81.53%, which is 6.09% higher than U-Net. The inference time of proposed method is 2.18 s/patch, and the inference speed is about 5 times faster than that of U-Net.

In order to verify the applicability of this method in the thermal infrared band, we apply this method to the band 11 of L8 CCA. Figure 6 shows the cloud detection results of the proposed method for band 11, and Table 7 shows the detailed results of the images in the test set. Experimental results show that for thermal infrared images, the pixel accuracy of proposed method can reach 90.99%. However, because the thermal infrared band of Landsat 8 has a resolution of 100 meters, and the input data is a single-channel image, the thermal infrared image has less information than the visible image. The mPA and mIoU of this method drop to 87.36% and 80.16%, respectively.

4 Conclusion

In this paper, a lightweight deep learning network for cloud detection of visible and thermal infrared remote sensing images is proposed. Depthwise separable convolutions is applied to compress the model of U-Net and accelerate the inference speed of the neural network. The proposed method is verified on Landsat 8 remote sensing images. Experimental results show that the proposed method can efficiently detect clouds on visible and thermal infrared images with high inference speed and low computational cost. Compared with U-Net, the cloud detection results of the proposed network in visible bands are greatly improved; And although the mPA and mIoU decreased in the thermal infrared images, the PA still remained at a high level. The parameters and FLOPs of the network are reduced to 12.4% and 12.8% of U-Net, respectively, and the inference speed increases by nearly 5 times. In the future, the pruning, quantization, and knowledge distillation will be adopted to improve the effectiveness and further compress the model.

References

Chen, Y., Fan, R., Bilal, M., Yang, X., Wang, J., Li, W.: Multilevel cloud detection for high-resolution remote sensing imagery using multiple convolutional neural networks. ISPRS Int. J. Geo Inf. **7**(5), 181 (2018). https://doi.org/10.3390/ijgi7050181

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

Gao, L., Song, W.D., Tan, H., Liu, Y.: Cloud detection based on multi-scale dilation convolutional neural network for ZY-3 satellite remote sensing imagery. Acta Opt. Sin. **39**(1), 0104002 (2019). https://doi.org/10.3788/AOS201939.0104002

Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Garcia-Rodriguez, J.: A review on deep learning techniques applied to semantic segmentation (2017) arXiv:1704.06857

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

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

Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2014). arXiv preprint arXiv:1412.6980

Li, W., Li, D.: The cloud detection study of MODIS based on HSV color clustering algorithm. In: 2011 International Workshop on Multi-Platform/Multi-Sensor Remote Sensing and Mapping. IEEE, pp. 1–4 (2011)

Li, Z., Shen, H., Li, H., Xia, G., Gamba, P., Zhang, L.: Multi-feature combined cloud and cloud shadow detection in GaoFen-1 wide field of view imagery. Remote Sens. Environ. **191**, 342–358 (2017)

Li, X., Su, X., Hu, Z., Yang, L., Zhang, L., Chen, F.: Improved distortion correction method and applications for large aperture infrared tracking cameras. Infrared Phys. Technol. **98**, 82–88 (2019)

Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440 (2015)

Mateo-García, G., Gómez-Chova, L., Camps-Valls, G.: Convolutional neural networks for multispectral image cloud masking. In: 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, pp. 2255–2258 (2017)

Milletari, F., Navab, N., Ahmadi, S.A.: V-net: fully convolutional neural networks for volumetric medical image segmentation. In: 2016 Fourth International Conference on 3D Vision (3DV). IEEE, pp. 565–571 (2016)

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. 31st conference on neural information processing systems. Long Beach, CA, USA (2017)

Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, pp. 234–241 (2015)

Rossow, W.B., Schiffer, R.A.: ISCCP cloud data products. Bull. Am. Meteorol. Soc. **72**, 2–20 (1991)

Shi, M., Xie, F., Zi, Y., Yin, J.: Cloud detection of remote sensing images by deep learning. In: 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, pp. 701–704 (2016)

Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: the all convolutional net (2014). arXiv preprint arXiv:1412.6806

Wieland, M., Li, Y., Martinis, S.: Multi-sensor cloud and cloud shadow segmentation with a convolutional neural network. Remote Sens. Environ. **230**, 111203 (2019). https://doi.org/10.1016/j.rse.2019.05.022

Williams, J.A., Dawood, A.S., Visser, S.J.: FPGA-based cloud detection for real-time onboard remote sensing. In: 2002 IEEE International Conference on Field-Programmable Technology, 2002. (FPT). Proceedings. IEEE, pp. 110–116 (2002)

Zhang, Z., Sabuncu, M.: Generalized cross entropy loss for training deep neural networks with noisy labels. In Advances in Neural Information Processing Systems, pp. 8778–8788 (2018)

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

Zi, Y., Xie, F., Jiang, Z.: A cloud detection method for Landsat 8 images based on PCANet. Remote Sens.ing **10**(6), 877 (2018). https://doi.org/10.3390/rs10060877

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Jiaqiang Zhang[1,3,4] · Xiaoyan Li[1,2,3,4] · Liyuan Li[1,3,4] · Pengcheng Sun[3,5] · Xiaofeng Su[1,3] · Tingliang Hu[1,3] · Fansheng Chen[1,2,3]

1   Key Laboratory of Intelligent Infrared Perception, Chinese Academy of Sciences, Shanghai 200083, China

2   Hangzhou Institute for Advanced Study, University of Chinese Academy of Sciences, Hangzhou 310024, China

3   Shanghai Institute of Technical Physics, Chinese Academy of Sciences, Shanghai 200083, China

4   University of Chinese Academy of Sciences, Beijing 100049, China

5   Key Laboratory of Specialty Fiber Optics and Optical Access Networks, Joint International Research Laboratory of Specialty Fiber Optics and Advanced Communication, Shanghai Institute of Advanced Communication and Data Science, Shanghai University, Shanghai 200444, China