Using Minimum Component and CNN for Satellite Remote Sensing Image Cloud Detection

Hailin Sun, Li Li, Mai Xu, *Senior Member, IEEE*, Qinpeng Li, and Zheng Huang

***Abstract*— Cloud detection is an important part of remote sensing (RS) image preprocessing. For earth observation tasks, the reliability of RS images will be judged based on the presence of clouds. A large number of cloud detection methods have been developed. There are two difficulties for cloud detection. First, it is hard to detect thin clouds and ragged clouds. Second, clouds are hard to distinguish from photometrically similar regions, such as snow. The rise of deep learning has brought new methods to address the above problems. In this letter, we propose a novel end-to-end neural network that detects clouds without additional manual work. Furthermore, we develop an RGB minimum component transformation mechanism that is useful for discriminating clouds from snow. Moreover, we have made our data set public to help others for further research. Our method increases the precision of cloud detection to 93.73%.**

***Index Terms*— Cloud detection, convolutional neural networks (CNNs), image segmentation, remote sensing (RS) image processing.**

I. Introduction

REMOTE sensing (RS) imagery plays a significant role in many fields, such as resource exploration, environment protection, military reconnaissance and agricultural engineering [1]. However, more than 50% of the earth's surface is covered by clouds at any time [2]. For all ground observation and target detection tasks, the presence of clouds can greatly reduce image quality and interfere with subsequent analysis and utilization [3], [4]. Therefore, cloud detection has become an extremely important and indispensable part of RS image preprocessing.

In recent years, many cloud detection methods from different perspectives have been proposed. Early RS satellites used microwave imaging, and researchers detected clouds through the difference between land, ocean, and cloud's reflections of electromagnetic waves of specific bands. Frey *et al.* [5] used 7.2-μm water vapor absorption band and surface temperature maps of land and ocean to reduce the dependence on ocean brightness temperature and to detect clouds in the Moderate Resolution Imaging Spectroradiometer (MODIS) data.

As optical data become more accessible, more algorithms based on optical images have been proposed. Xie *et al.* [6] and Shi *et al.* [7] used a combination of simple linear iterative clustering (SLIC) and convolutional neural networks (CNNs) to detect clouds. Xie *et al.* [6] transformed images from RGB color space to hue, saturation, and intensity (HSI) color space and then used the saturation and intensity channels to perform improved SLIC. Then, the generated super pixels were classified by CNNs. Both Xie *et al.* [6] and Shi *et al.* [7] neglected the distinction between clouds and snow. Tuia *et al.* [8] used image time series as input to the recurrent neural network (RNN) and obtained the difference between cloudy images and noncloud images of the same region at different times. This method requires a lot of memory and needs to collect data at different times in the same area. Li *et al.* [9] developed a method that integrates multiple features. They first implemented threshold segmentation based on spectral features. After mask refinement based on guided filtering, they used geometric features in combination with the texture features to improve the cloud detection results and produced the final cloud mask. The steps are cumbersome. Yan *et al.* [10] proposed a fully convolutional network called multilevel feature fused segmentation network (MFFSNet). They designed a special multilevel feature fused structure to combine semantic information with spatial information from different levels. Deng *et al.* [11] combined natural scene statistic (NSS) and Gabor features to detect clouds. They first applied SLIC to segment the image into ambiguous super pixels and then used the NSS features to classify the ambiguous super pixels into possible thick clouds, thin clouds, and nonclouds and finally used one support vector machine (SVM) to distinguish clouds from snow. Lu *et al.* [12] modified the symmetric structure based on SegNet [13] and proposed two networks called P_SegNet and NP_SegNet. They found that symmetric networks performed better in cloud detection. However, their networks have difficulty converging in the absence of a residual block (RB). Yang *et al.* [1] proposed CDnet for cloud detection. They used feature pyramid (FP) module to extract the multiscale contextual information without the loss of resolution and coverage and used boundary refinement (BR) block to refine object boundaries. Their network consisted of one ResNet-50 network, three FP modules, and six BR blocks. There is a large number of parameters in the entire network.

Manuscript received March 3, 2020; revised June 9, 2020; accepted August 1, 2020. Date of publication August 18, 2020; date of current version November 24, 2021. This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 61922009, Grant 61876013, and Grant 61573037. *(Corresponding author: Li Li.)*

Hailin Sun, Li Li, and Zheng Huang are with the School of Electronic Engineering, Beihang University, Beijing 100191, China (e-mail: lili2005@buaa.edu.cn).

Mai Xu is with the School of Electronic Engineering, Beihang University, Beijing 100191, China, and also with the Hangzhou Innovation Institute, Beihang University, Hangzhou 310051, China.

Qinpeng Li is with the General System Department, China Center for Resources Satellite Data and Application, Beijing 100094, China.

Color versions of one or more of the figures in this letter are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2020.3014358

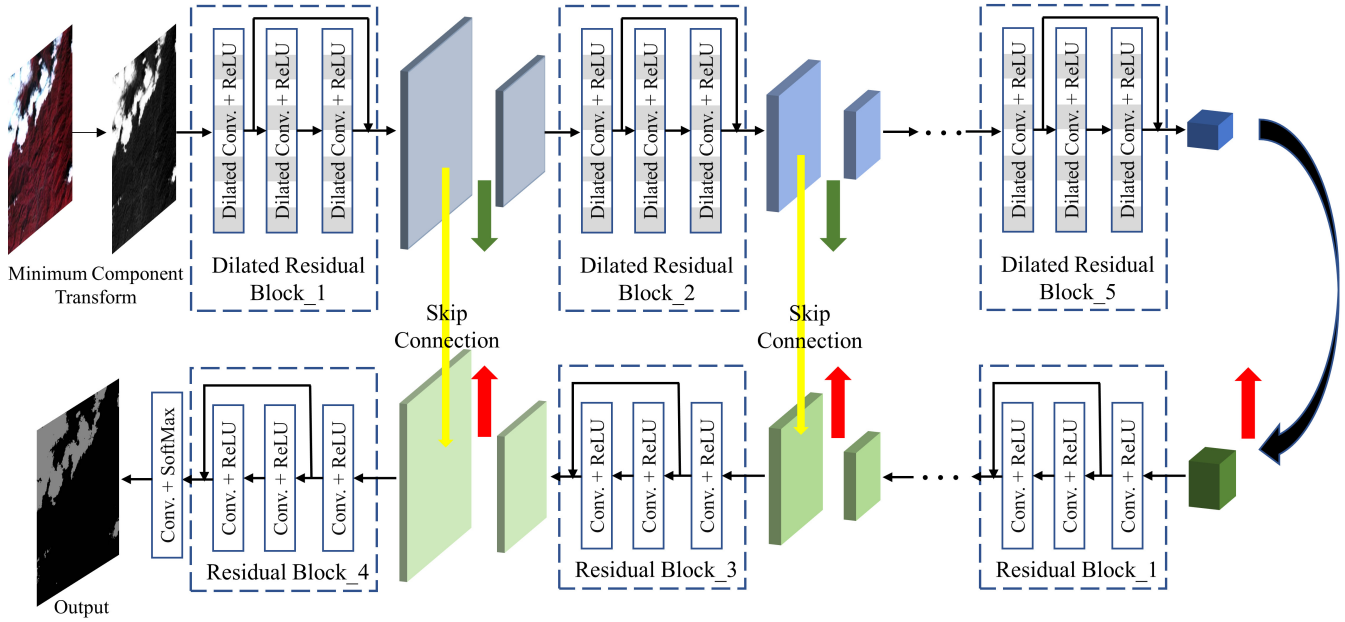

Fig. 1. Our cloud detection CNN. Green down arrows and red up arrows correspond to max pooling and upsampling, respectively. Yellow arrows indicate the copying and concatenation of the corresponding map features.

In this letter, we propose a novel end-to-end network with RBs and skip connection to detect cloud and develop a minimum component (MC) transformation mechanism for RGB images that is useful for discriminating clouds from snow. The remainder of the letter is organized as follows. In Section II, we introduce our network architecture and the MC transformation method. Section III describes the experimental details and results. We provide the conclusion in Section IV.

II. Proposed Method

In this part, we will illustrate our network architecture (see Fig. 1) in detail. Then, we discuss the MC transformation for RGB images.

A. Encoder and Decoder

During the encoding process, we use five dilated RBs (DRBs), each of which has three dilated convolution layers. Early dilated convolutions have larger receptive fields than normal convolutions and will extract more texture features. In the whole network, we keep the convolution kernel sizes at 3×3. Rectified Linear Unit (ReLU) [14] is used as the activation function throughout the CNN. In DRB 1, the number of kernels of each convolution layer is 128. This number reaches 256 in DRB 2 and 3. Then, in DRB 4 and 5, we double the number of convolution kernels to 512. The input size of our network is 256×256. After each max pooling (green down arrows), the size of the feature maps will be reduced by half.

In the decoding stage, we have four normal RBs, each of which has three normal convolution layers with a kernel size of 3×3. According to the result in [12], a symmetric architecture can achieve a higher accuracy in cloud detection, which is why we try to maintain the same number of channels in the decoder as in the encoder. We copy and concatenate the map features at the ends of each yellow arrow. Upsampling (red arrows) will double the feature map size, so the size of the output is restored to 256×256. The output layer has three channels: white for snow, gray for clouds, and black for background areas.

B. MC Transform

For the convolution calculation of the neural network, each layer is calculated as follows:

$$\text{output}_j = \sum_{i \in \text{InputChannel}} \text{input}_i \otimes \text{kernel}_j. \quad (1)$$

For example, we give the input tensor size (row, col, channel) and the convolution kernel size (height, width, and number). With zero padding, the size of the output tensor is (row-height+1, col-width+1, number). The "channel" in the input dimension disappeared

$$\text{output}_{i,j,k} = \sum_{p \in \text{channel}} \sum_{m=1}^{\text{height}} \sum_{n=1}^{\text{width}} (\text{input}_{m+i,n+j,p} \text{kernel}_{m,n,k}). \quad (2)$$

As the number of network layers increases, the gradient will vanish during the backward propagation process. In other words, when the gradient is transferred from the output layer back to the input layer, the gradient amplitude will be greatly attenuated. The weight of the input layer will only change slightly.

For cloud detection tasks, clouds have high brightness and whiteness, high intensity and low saturation. Xie *et al.* [6] transformed RGB images to HSI color space and produced a good result. Using the high intensity and low saturation characteristics of clouds, we propose a new transformation

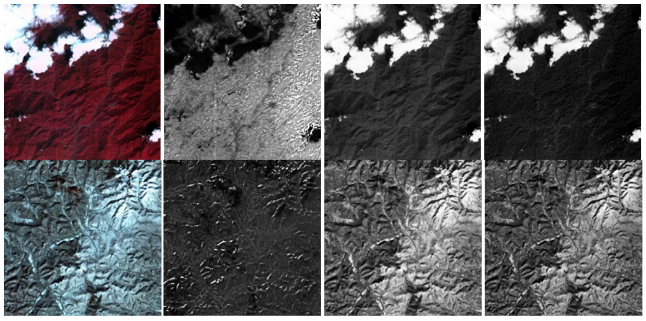

Fig. 2. Difference between RGB images, gray images, HSI images, and MC images. (a) RGB images. (b) Saturation component of the HSI color space. (c) Intensity component of the HSI color space. (d) MC of the RGB images.

method called MC transformation [See (3)]

$$\mathrm{MC} = \min(\mathrm{R}, \mathrm{G}, \mathrm{B}) \tag{3}$$

$$\mathrm{MC}\begin{pmatrix} \mathrm{R} \\ \mathrm{G} \\ \mathrm{B} \end{pmatrix}\begin{bmatrix} 255 & 255 & 0 & 255 \\ 255 & 0 & 255 & 255 \\ 0 & 255 & 255 & 255 \end{bmatrix} = [\,0 \quad 0 \quad 0 \quad 255\,]. \tag{4}$$

We use the MC of the three (RGB) channels as input to the CNN. Fig. 2 shows the difference between RGB images, gray images, HSI images, and MC images. Equation (4) shows a simple example of MC transformation. If a region has any low value in the RGB channels, it will become quite dark after MC transformation.

This method can effectively address the interference of high-brightness areas. As the cloud pixels are pure white, MC transformation can maintain high recognition while reducing the gray level of other areas. The improvement on the results will be shown in Section II-C.

C. Focal Loss

Lin *et al.* [15] proposed focal loss for dense object detection to address the problem that the accuracy of one-stage methods in object detection is not as high as that of two-stage methods. Focal loss deals with the imbalance of positive and negative samples and the predominance of simple samples in the gradient. For cloud detection tasks, thick clouds and large clouds can be easily detected, while thin clouds and snow are difficult to identify. Using focal loss [see (5)] instead of traditional cross entropy [see (6)] can effectively solve the above problem. Focal loss uses hyperparameters α and γ to improve CrossEntropy. α is the multiplier decay factor and γ is the exponential decay factor. p_t is the classification probability output by the network

$$\mathrm{FocalLoss}(p_t) = -\alpha(1-p_t)^{\gamma}\log(p_t) \tag{5}$$

$$\mathrm{CrossEntropy}(p_t) = -\log(p_t). \tag{6}$$

Compared to the cross entropy, the focal loss uses a weighting coefficient $(1-p_t)^{\gamma}$. This coefficient reduces the weights of simple samples, that is, samples with large p_t. This drawback becomes more pronounced when γ is large. As shown in Fig. 3(a), for a part with a large prediction probability, i.e., the simple samples (such as $p_t > 0.6$), the focal loss will reduce its proportion in the loss function, thereby focusing the computing power on the classification of difficult samples. As illustrated in Fig. 3(b), when γ is large, the weights of simple samples are reduced, while the weights of complex samples are increased. However, a larger γ is not always better. The neural network will not converge when γ is too large, which we guess is because of the large proportion of complex samples. We use grid search to find out the optimal solution of α and γ, which is shown in Table I.

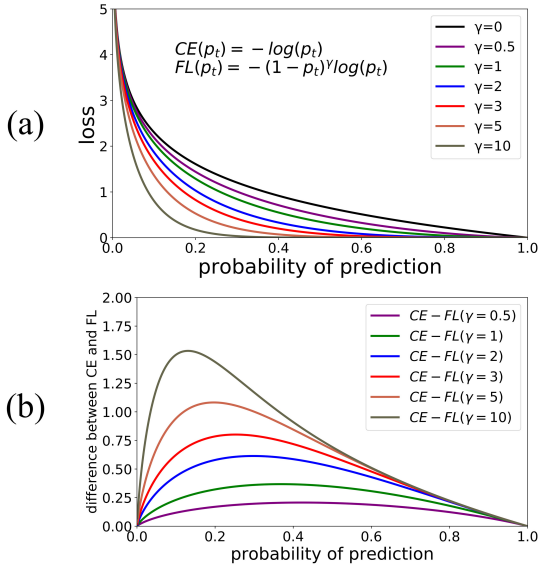

Fig. 3. (a) Focal loss for different γ values. (b) Difference between the cross entropy and focal loss of different γ values.

III. Experimental Results

A. Data Set

We use a total of 2443 RGB images[1] taken by GF series satellites (obtained from China Center for Resources Satellite Data and Application [http://www.cresda.com/CN/]). Among these 2443 pictures, 1000 contain clouds, 1000 contain snow, and 443 contain clouds and snow. For the first 2000 images, we use threshold segmentation to generate masks. For the other 443 images we use label-me [16] for pixel-level labeling. We perform the same data augmentation procedures as with U-Net [17], such as horizontal or vertical translation, flipping, and random noise interference, to improve the complexity and reliability of the model. In addition, we also perform random brightness changes to make the algorithm more robust in discriminating clouds from snow. The data set includes clouds of various categories, including thin, thick, and ragged clouds (see Fig. 4), that commonly occur in high-resolution satellite images. The size of the original images is approximately 3000×2000, and we uniformly scale them to 256×256, which is the input size of our neural network. We use fivefold cross validation to improve the universality and generalizability of the algorithm. For each independent model, we use

[1]We make most of the data available at https://bhpan.buaa.edu.cn:443/link/DDC7765A5A049E0F9A0DAD0E9F7692C5

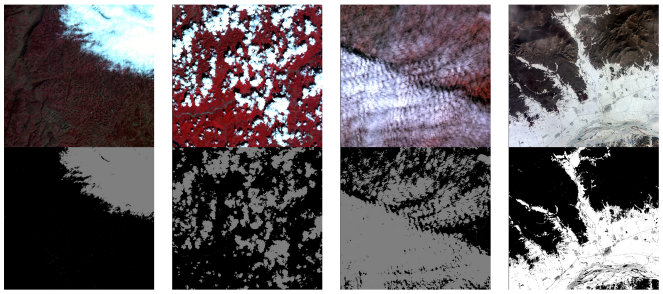

Fig. 4. Some examples from the data set. The images in the second row are the corresponding masks; the clouds are gray, and snow is white. (a) Big and thick cloud. (b) Ragged cloud. (c) Thin cloud. (d) Snow and ice.

TABLE I
EFFECT OF DIFFERENT FOCAL LOSS COEFFICIENTS ON THE CLOUD DETECTION PRECISION

γ \ α	0.25	0.5	0.75
2	0.8490	0.8834	0.8585
3	0.8439	**0.9373**	0.9035
4	0.8520	0.8662	0.8543
5	0.8516	0.8850	0.8690
10	0.8337	0.8836	0.7353

1954 images for training and 489 images for testing. The final result is the average of results of five models.

B. Setting

All images are used for fivefold cross validation. Our neural network is implemented with the Keras [18] framework. Adam [19] is used to optimize our CNN, where the learning rate is set to 0.0001. We set the batch size as 8 and the maximum training round as 2k. In addition, we use L2 regularization with a penalty factor of 0.001 to avoid overfitting. Since in cloud detection, it is a major difficulty to recognize snow and thin clouds, we use the focal loss [15] [see (5)] as our network loss function instead of the cross entropy [see (6)], which can put more attention and computing power on complex example mining. Therefore, our total loss function is

$$l = -\alpha(1 - p_t)^{\gamma} \log(p_t) + l_2\|\omega\| \tag{7}$$

α and γ are the hyperparameters of focal loss. l_2 is the coefficient of regularization. p_t is the ouput of the CNN, and ω are the network weights. In our experiments, we will compare our results with those of [6] and [7] and the P_SegNet of Lu *et al.* [12], which are state-of-the-art methods in cloud detection.

Because the focal loss contains two coefficients, we compare the effect of these two parameters on the detection accuracy. As shown in Table I, when $\alpha = 0.5$ and $\gamma = 3$, the detection precision obtained by the focal loss is maximized. As seen from Table I, the use of the focal loss does improve the detection accuracy of snow.

C. Cloud Detection Accuracy

The number of parameters in neural networks affects the training and inference speeds. The inference speed directly determines the application performance of the neural network algorithm. We compared the number of parameters of the proposed networks with the three networks mentioned above. The comparison results are shown in Table II. Shi's and Xie's methods have millions of parameters, mainly because they still use fully connected layers in the last steps for classification. Both Lu's method and the proposed method use a fully convolutional network. Compared to Lu's CNN, our CNN has approximately 4k more parameters, which we think are worthwhile for the performance improvement.

TABLE II
NUMBER OF NETWORKS PARAMETERS

Layer type	Shi *et al.*	Xie *et al.*	Lu *et al.*	Proposed
Conv.	4464	2976	79104	82944
Fully connected	1179904	2426368	**0**	**0**
Total	1.18M	2.43M	**79k**	83k

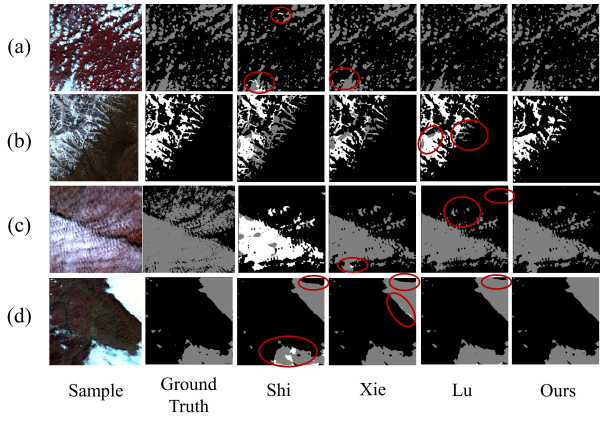

Fig. 5. Visual comparison of the detection results. White area: snow. Gray area: cloud. Black area: background. (a) Ragged cloud. (b) Snow. (c) Thin cloud. (d) Thick cloud.

Fig. 5 illustrates the results of the above compared methods in the cloud detection task. The first column is the RS image used for testing. The second column is the ground truth, with gray corresponding to clouds, black corresponding to background and white corresponding to snow. The remaining four columns are the test results of the compared algorithms. Fig. 5(a) shows the detection of ragged clouds. The results of the four networks show that CNNs have satisfactory performance in extracting features. However, Shi's and Xie's methods mistake some small clouds for snow, while the detection accuracies of Lu's and our methods are higher. Fig. 5(b) corresponds to snow detection. Shi's and Xie's methods perform poorly on the edge and sharp areas of snow, which may be due to the use of more fully connected neurons and fewer convolution kernels, the quantity of which is not sufficient for extracting features. Fig. 5(c) shows the detection of thin clouds. Shi's method misjudges the whole thin cloud as snow, and Xie's method misjudges only a small part. Lu's and our method have certain discriminating ability for ordinary thin clouds. However, all of the methods omitted thinner clouds. Fig. 5(d) is the result of thick cloud detection. All the

TABLE III
METRICS OF THE COMPARED METHODS

Methods	Cloud			Snow		
	Precision	*Recall*	*F1-score*	*Precision*	*Recall*	*F1-score*
[6]	0.8987	0.8813	0.8899	0.6935	0.8023	0.7439
[7]	0.9059	0.8814	0.8934	0.7141	0.8185	0.7627
[12]	0.9133	0.9020	0.9076	0.7577	0.8289	0.7917
Our CNN	0.9112	0.9024	0.9067	0.7230	0.8565	0.7841
Our CNN+MC	0.9369	0.9345	0.9356	0.7961	0.8547	0.8244
Our CNN+MC+FL	**0.9373**	**0.9370**	**0.9371**	**0.8113**	**0.8668**	**0.8381**

compared methods perform fairly well, but Shi's, Xie's, and Lu's methods missed the edge area, especially the top edge of sample images, while our method is better than the other methods

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{8}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

$$F_1 = 2\frac{\text{Precision Recall}}{\text{Precision} + \text{Recall}}. \tag{10}$$

Table III shows the metrics of the detection results. For TP, TN, FP, and FN, TP is the ratio of the number of *true positive* samples detected to the number of all *positive* samples, and TN is the ratio of the number of *true negative* samples detected to the number of all *negative* samples, and so on. The *Precision* emphasizes the detection accuracy. The *Recall* focuses on the sensitivity to the detected objects. The *F1-score* is the harmonic average of *Precision* and *Recall*. Our proposed network has a high detection precision for clouds and a slightly lower detection precision for snow than Lu's method. After using the MC method, we achieve a 7.31% improvement in the snow detection precision and a 2.57% improvement in the cloud detection precision. With the focal loss replacing the cross entropy as the loss function, the detection precision of snow increased by 1.52%, and the recall increased by 1.21%.

IV. CONCLUSION

In this letter, we created a publicly available data set with thousands of satellite images for cloud detection. In addition, we propose a fully convolutional network for RS image cloud detection. Based on prior knowledge of clouds and snow, we propose a MC transformation based on RGB channel values. This mechanism can improve the detection accuracy of clouds and snow. In addition, we found a set of focal loss parameter pairs that can further improve performance. The experimental results demonstrate the effectiveness of our method.

In future work, we will try to prune and compress the existing cloud detection neural network, which is also the main goal of current neural network development. A more streamlined network means a faster inference speed, which is important.

REFERENCES

[1] J. Yang, J. Guo, H. Yue, Z. Liu, H. Hu, and K. Li, "CDnet: CNN-based cloud detection for remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 6195–6211, Aug. 2019.

[2] G. W. Paltridge and C. M. R. Platt, *Radiative Processes in Meteorology and Climatology*. New York, NY, USA: Elsevier, 1976.

[3] R. W. Saunders, "An automated scheme for the removal of cloud contamination from AVHRR radiances over western Europe," *Int. J. Remote Sens.*, vol. 7, no. 7, pp. 867–886, Jul. 1986.

[4] R. W. Saunders and K. T. Kriebel, "An improved method for detecting clear sky and cloudy radiances from AVHRR data," *Int. J. Remote Sens.*, vol. 9, no. 1, pp. 123–150, Jan. 1988.

[5] R. A. Frey *et al.*, "Cloud detection with MODIS. Part I: Improvements in the MODIS cloud mask for collection 5," *J. Atmos. Ocean. Technol.*, vol. 25, no. 7, pp. 1057–1072, Jul. 2008.

[6] F. Xie, M. Shi, Z. Shi, J. Yin, and D. Zhao, "Multilevel cloud detection in remote sensing images based on deep learning," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 8, pp. 3631–3640, Aug. 2017.

[7] M. Shi, F. Xie, Y. Zi, and J. Yin, "Cloud detection of remote sensing images by deep learning," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 701–704.

[8] D. Tuia, B. Kellenberger, A. Perez-Suey, and G. Camps-Valls, "A deep network approach to multitemporal cloud detection," in *Proc. IGARSS - IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2018, pp. 4351–4354.

[9] Z. Li, H. Shen, H. Li, G. Xia, P. Gamba, and L. Zhang, "Multi-feature combined cloud and cloud shadow detection in GaoFen-1 wide field of view imagery," *Remote Sens. Environ.*, vol. 191, pp. 342–358, Mar. 2017.

[10] Z. Yan *et al.*, "Cloud and cloud shadow detection using multilevel feature fused segmentation network," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 10, pp. 1600–1604, Oct. 2018.

[11] C. Deng, Z. Li, W. Wang, S. Wang, L. Tang, and A. C. Bovik, "Cloud detection in satellite images based on natural scene statistics and Gabor features," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 4, pp. 608–612, Apr. 2019.

[12] J. Lu *et al.*, "P_Segnet and NP_Segnet: New neural network architectures for cloud recognition of remote sensing images," *IEEE Access*, vol. 7, pp. 87323–87333, 2019.

[13] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for scene segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[14] G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines Vinod Nair," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.

[15] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.

[16] K. Wada. (2016). *labelme: Image Polygonal Annotation with Python*. [Online]. Available: https://github.com/wkentaro/labelme

[17] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Intervent*, 2015, pp. 234–241.

[18] F. Chollet. (2015). *Keras*. [Online]. Available: https://keras.io

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980