LigCDnet:Remote Sensing Image Cloud Detection Based on Lightweight Framework

Baotong Su[1,2] and Wenguang Zheng[1,2](✉)

[1] School of Computer Science and Engineering, Tianjin University of Technology, TianJin 300384, China
subaotong@stud.tjut.edu.cn, wenguangz@tjut.edu.cn

[2] Tianjin Key Laboratory of Intelligence Computing and Novel Software Technology, Tianjin University of Technology, TianJin 300384, China

Abstract. Cloud contamination is inevitable in remote sensing images, resulting in a large number of images that cannot be applied in various fields. Therefore, cloud detection is one of the important tasks in remote sensing image preprocessing, aimed at removing images obstructed by clouds. Most existing methods are mostly based on CNN and feature a complex network structure, requiring a significant amount of computational resources, making it challenging to deploy them in practical applications. To tackle this problem, we propose a lightweight cloud detection framework (LigCDnet) with a lightweight feature extraction module (LFEM), a channel attention module (CAM), and a lightweight feature pyramid module (LFPM). The LFEM serves as the backbone of the network to capture rich spatial and contextual information; the CAM adaptively adjusts the channel weights of the feature maps; and the LFPM extracts cloud features at multiple scales. The effectiveness of our approach is evaluated on two public datasets, GF-1 and LandSat8. Extensive experiments have demonstrated that the proposed LigCDnet achieves state-of-the-art detection accuracy while significantly reducing computational burden and having a smaller model size.

Keywords: Remote sensing images · Cloud detection · Lightweight Framework · GF-1 · LandSat8

1 Introduction

With the rapid development of remote sensing technology, optical remote sensing images have been extensively used in various fields such as agriculture engineering, geographical survey, military reconnaissance, natural disaster prediction, and environmental pollution monitoring [16]. However, cloud occlusion is an inevitable challenge in satellite imagery due to the extensive cloud cover that spans over 60% of the Earth's surface area [14]. The cloud cover obstructs the satellite sensor's ability to obtain a clear view of the Earth's surface, making

S. Rudinac et al. (Eds.): MMM 2024, LNCS 14555, pp. 437–450, 2024.
https://doi.org/10.1007/978-3-031-53308-2_32

many image analysis tasks difficult, such as remote sensing image classification and segmentation [29], image matching [8], etc. Therefore, it is necessary to quickly and accurately detect cloud cover in order to enhance the availability of remote sensing images.

Over the years, researchers have conducted in-depth studies on cloud detection algorithms in remote sensing imagery and have proposed numerous algorithms. These methods can be broadly categorized into two types: threshold-based methods and machine learning-based methods. Threshold-based methods rely on the physical characteristics of clouds and set appropriate thresholds based on these characteristics to classify pixels in an image into cloud and non-cloud categories. ISCCP [21] cloud mask algorithm utilized the fact that cloud and clear scenes differ in the amount of radiance variability they exhibit in space and time to detect clouds. Cihlar and Howarth [7] proposed a method that can identify clouds with different opacities as well as cloud shadows present in composite materials, effectively eliminating the impact of cloud contamination in AVHRR synthetic images on land. Huang et al. [12] used clear forest pixels as a reference to define cloud boundaries and separate clouds from clear surfaces in a spectral-temperature space. However, these methods lack a universal threshold and do not consider the structure and texture of clouds when dealing with complex scenes, resulting in low robustness. The principle of machine learning-based methods for cloud detection is to extract features from remote sensing images as input and then train a classification model by comparing these features with labeled samples. An and Shi [2] designed a scene learning-based cloud detection algorithm, this algorithm utilizes the color features, texture features, and structural features of the image. Li et al. [13] extracted brightness features, texture features, and average gray-level co-occurrence matrix features [10] from the image, they then used these features to train a support vector machine (SVM) [25] classifier. Shi et al. [23] proposed a ground-based cloud detection method using graph model built upon super-pixels [1] to integrate multiple sources of information. However, these methods extract shallow features from images through statistical means such as mean, maximum, minimum, variance, etc., which do not effectively comprehend the images, leading to a decrease in detection accuracy.

In recent years, convolutional neural networks (CNNs) have allowed the field of computer vision to grow rapidly with their powerful feature extraction capabilities. CNN-based approaches can improve the model's understanding of images by stacking convolutional layers and yield superior performance in target detection, image classification, and semantic segmentation [24,33]. Yang et al. [30] utilize thumbnail images to extract cloud masks, extracting multi-scale contextual information without losing resolution. Wu and Xu [28] present cross-supervised learning for cloud detection to address the issue of insufficient labeled cloudy images. However, most existing deep learning methods feature complex network structures and high computational resource requirements. In practical applications, the deployed devices often lack significant computational power and storage space. Therefore, this limits the applicability of these methods.

To solve the above problem, we design a lightweight cloud detection framework called LigCDnet, which achieves excellent performance with very few parameters. The contributions of this paper are briefly summarized as follows:

1. We propose a lightweight cloud detection framework based on the U-shaped architecture, called LigCDnet. It achieves state-of-the-art detection accuracy compared to existing cloud detection algorithms while having an extremely small number of parameters, only 2.39M.
2. We utilize a lightweight feature extraction module (LFEM) to capture spatial and contextual information and design a channel attention module (CAM) to adjust the impact of different channels on cloud detection performance. Additionally, we propose a lightweight feature pyramid module (LFPM) to extract cloud features at different scales.

2 Proposed Method

2.1 Overall Framework

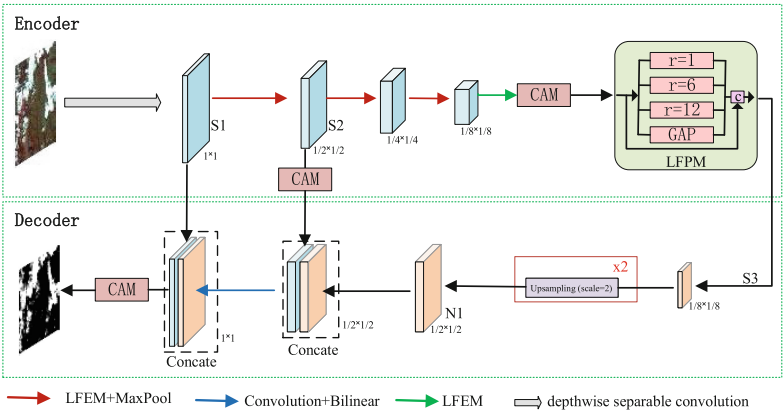

Fig. 1. Framework of the proposed LigCDnet.

We use a U-shaped encoder-decoder structure as the framework for our cloud detection network model, as shown in Fig. 1. High-level features contain rich semantic information, low-level features contain rich spatial information [32]. The abundant spatial information plays a crucial role in generating cloud masks. Therefore, in the encoder part, we performed three downsampling operations to preserve rich spatial information in the feature map. We utilize the lightweight feature extraction module (LFEM) that maximizes the extraction of cloud features while minimizing the increase in parameters. To enhance the feature maps channels that are favorable for cloud segmentation, designing a channel attention module (CAM) to adjust the channel weights. To better understand clouds

of different scales, we propose the lightweight feature pyramid module (LFPM). In the decoder part, we gradually restore the resolution of the feature maps through upsampling and compensate for the spatial information lost during the encoding stage by connecting them with the feature maps in the encoder using skip connections.

Given a remote-sensing image I as input, the feature map $S1$ is first generated in the encoder by depthwise separable convolution operations. Depthwise separable convolution consists of depthwise convolution and pointwise convolution. that is

$$S1 = H_{conv_{dep}}\big(H_{conv_{poi}}(I)\big) \tag{1}$$

where $H_{conv_{dep}}(\cdot)$ represents depthwise convolution operation, and $H_{conv_{poi}}(\cdot)$ denotes pointwise convolution operation, $S1$ has the same size as the input image.

To reduce the computational complexity while extracting cloud information, we use a downsampling unit, that is

$$F_{downsampling} = MaxPool\big(H_{LFEM}(S)\big) \tag{2}$$

where $MaxPool(\cdot)$ is max pooling operation, and $H_{LFEM}(\cdot)$ denotes the lightweight feature extraction module. Then, the feature maps $S2, S3$ are generated by consecutive downsampling unit, that is

$$S2 = F_{downsampling}(S1) \tag{3}$$

$$S3 = H_{LFPM}\Big(H_{CAM}\Big(H_{LFEM}\big(F^2_{downsampling}(S2)\big)\Big)\Big) \tag{4}$$

where $F^2_{downsampling}(\cdot)$ means that $F_{downsampling}$ is executed two times, $H_{CAM}(\cdot)$ represents channel attention module, H_{LFPM} denotes lightweight feature pyramid module. $S2$ is $1/2 \times 1/2$ size of the input image, $S3$ is $1/8 \times 1/8$ size of the input image.

Due to the small spatial resolution of the feature map $S3$ generated in the encoder, it leads to problems such as information loss, insufficient contextual information, and blurred boundaries. In the decoder part, we restore the feature map to the same resolution as the input image by gradually upsampling it, the upsampling is limited to $2\times$. To reduce computational complexity, we employ a simple bilinear interpolation to directly upsample $S3$ twice, resulting in the generation of feature map $N1$, $N1$ is $1/2 \times 1/2$ size of the input image. And introduce the upsampling unit, that is

$$F_{upsampling} = bilinear\big(H^2_{conv_3}(I)\big) \tag{5}$$

where $bilinear(\cdot)$ denotes bilinear interpolation operation, $H_{conv_3}(\cdot)$ represents a convolution operation with a convolution kernel size of 3, and the predicted cloud detection result I_O can be described as

$$I_O = H_{CAM}\Big(concat\Big(S1, F_{upsampling}\Big(concat\big(H_{CAM}(S2), N1\big)\Big)\Big)\Big) \tag{6}$$

where *concate* denotes the concatenate operation, I_O has the same size as the input image.

2.2 Lightweight Feature Extraction Module

In recent years, the main trend in improving the network's understanding of complex scenes has been the development of deeper and more complex networks. However, these networks require a significant amount of computational cost both during training and inference phases. To address this challenge, a plethora of lightweight network frameworks have been proposed. For instance, in MobileNet [22], depthwise separable convolution is employed, which consists of depthwise convolution and pointwise convolution. Depthwise convolution operates independently on each channel of the input feature map, pointwise convolution integrates information between different channels to enhance the network's representational capacity. In LEDNet [26], channel splitting and shuffle operations are applied to each residual block. Channel splitting divides the channels of the feature map into multiple groups, allowing the network to independently extract different types of features. Shuffle operations enable information exchange between different channel groups. Inspired by the MobileNet, we designed the lightweight feature extraction module shown in Fig. 2. The lightweight feature extraction module consists of two pointwise convolutions and one depthwise convolution. For a feature map with C channels, the first step is to increase its dimensionality to $2C$ by employing a pointwise convolution. The different channels of feature maps can be seen as the network's response to various characteristics of the data, enabling the network to understand the data from different perspectives. Subsequently, a depthwise convolution is utilized to capture spatial features of the cloud and extract local information for each channel. Lastly, another pointwise convolution is employed to reduce the dimensionality back to C while integrating features across the channels of the feature map.

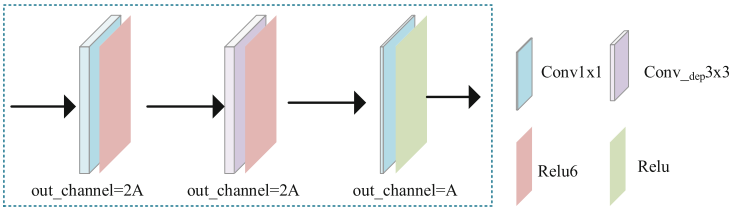

Fig. 2. Structure of LFEM.

For a 3×3 standard convolution, with an input feature map of $[H \times W \times C]$, output channel set to $2C$, and convolution layer depth set to 3, the number of parameters of the module is $3{\times}3{\times}C{\times}2C{+}3{\times}3{\times}2C{\times}2C{+}3{\times}3{\times}2C{\times}2C = 90C$. And the number of parameters of our lightweight feature extraction module is $1 \times 1 \times C \times 2C + 3 \times 3 \times 2C + 1 \times 1 \times 2C \times 2C = 24C$. With the same depth of convolution layers, the number of standard convolutional parameters is 3.74 times higher than ours, and the LFEM module greatly reduces the number of parameters while increasing the inference speed and computational efficiency.

The lightweight feature extraction module can be stated as

$$H_{\mathrm{LFEM}} = H_{conv_{poi}}\left(H_{\mathrm{conv_{dep}}}\left(H_{conv_{poi}}(I)\right)\right) \tag{7}$$

2.3 Channel Attention Module

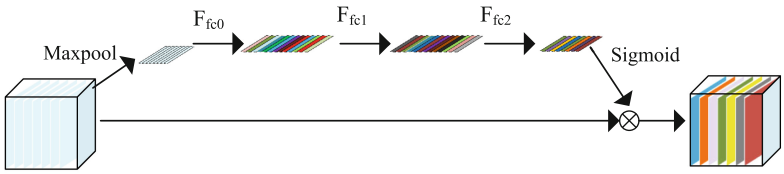

Fig. 3. Structure of CAM.

In computer vision tasks, which often rely on convolutional operations to extract features from images, different channels of the feature map play different important roles for the task in the process of network learning. Therefore, we designed a channel attention module and introduced a learnable weight vector to enable the network to automatically learn the importance of different channels in the task. It allows the network to adjust the weights of each channel, enhancing the dependency on important channels and reducing the dependency on unimportant channels, as shown in Fig. 3. First, we apply max-pooling operations to each channel of the feature map to generate initial channel weight vectors. Then, these vectors are fed into three layers of linear units to let the network learn the importance of different channels. Subsequently, the weight vectors are normalized using the sigmoid function, and finally, the weight vectors are element-wise multiplied with their corresponding channels. By adjusting the channels of the feature map, the network can utilize the information between channels more effectively, thereby improving the performance of the task. The channel attention module can be stated as

$$H_{\mathrm{LFEM}} = F_{fc2}\left(F_{fc1}\left(F_{fc0}\left(MaxPool(I)\right)\right)\right) \otimes I \tag{8}$$

where F_{fc} denotes Linear layer operation, $\otimes$ is element-wise multiplication operation.

2.4 Lightweight Feature Pyramid Module

Clouds have diverse morphologies, and accurately segmenting clouds of different sizes is a fundamental challenge for cloud detection algorithms. Capturing multi-scale cloud features and establishing contextual information can effectively

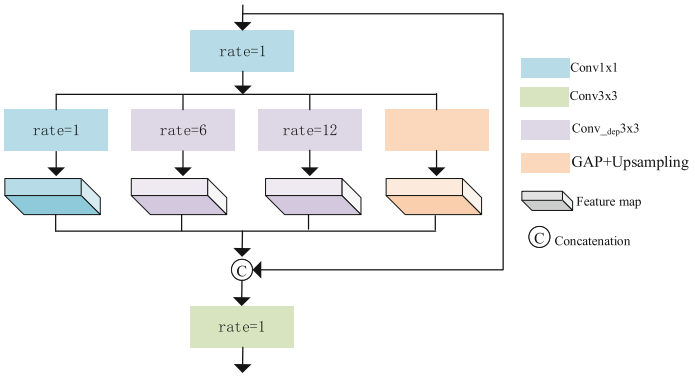

Fig. 4. Structure of LFPM.

enable the network to learn feature differences between cloud regions and backgrounds. Inspired by the ASPP [5] model, we propose a lightweight pyramid module, as shown in Fig. 4. The number of channels of the feature map is first adjusted by a pointwise convolution. Dilated convolution [31], also known as atrous convolution, can significantly expand the receptive field of convolutional neural networks. Combining multiple dilated convolutions with different sampling rates in parallel effectively captures multi-scale contextual information. Therefore, we use parallel dilated convolutions with dilation rates of 1, 6, and 12. To reduce computational complexity, the dilated convolutions are replaced with deepwise convolutions while keeping the dilation rates unchanged. Additionally, a global average pooling (GAP) layer is introduced to extract global contextual information. Subsequently, the features captured by the four parallel branches are concatenated along the channel dimension. To facilitate feature reuse and mitigate the gradient vanishing problem, short connections are introduced. The Lightweight Feature Pyramid Module can be stated as

$$H_{\text{LFPM}} = concate(I, H_{conv_{poi}}(I), H_{conv_{dep-r6}}(I), H_{conv_{dep-r12}}(I), bilinear(H_{GAP}(I))) \tag{9}$$

where $H_{conv_{dep-r}}(\cdot)$ is depthwise convolution with dilate rate r, H_{GAP} denotes global average pooling operation.

3 Experimental Results

3.1 Dataset and Experimental Setup

Dataset. We chose two widely used datasets, GF-1 remote sensing images and datasets of CloudSat8, to validate the effectiveness of our method, using only their visible channels. The GF-1 remote sensing images includes 108 GF-1 Wide Field of View (WFV) level-2A scenes and its reference cloud and cloud shadow

masks. 86 of the images are used for training and 22 images are used for testing [15]. The CloudSat8 dataset contains 18 images of size 1000 × 1000 for training and 20 same-size images for the test [19]. We crop the above original high pixel image into 512 × 512 × 3 sub-images for training and testing.

Evaluation Metrics. In order to measure the performance of the model comprehensively, we used six widely used quantitative metrics, including JaccardIndex, Precision, Recall, F1-score, and overall accuracy (OA), mean intersection over union (MIoU). These metrics are defined as follows:

$$JaccardIndex = \frac{TP}{(TP + FN + FP)} \tag{10}$$

$$Precision = \frac{TP}{(TP + FP)} \tag{11}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{12}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{(Precision + Recall)} \tag{13}$$

$$OverallAccuracy = \frac{TP + TN}{(TP + TN + FP + FN)} \tag{14}$$

$$MIoU = \frac{1}{k} \sum_{i=1}^{k} \frac{n_{ii}}{\sum_{j=1}^{k} n_{ij} + \sum_{j=1}^{k} n_{ji} - n_{ii}} \tag{15}$$

where TP, TN, FP, and FN are the total number of true-positive, true-negative, false-positive, and false-negative pixels, respectively. The k represents the number of categories, n_{ii} represents the count of correctly predicted pixels, and nij represents the count of pixels where the true value is i and they were predicted as j.

Parameter Settings. Our model is implemented using the Pytorch framework [20], with the training step running on Ubuntu 22.04 and an RTX3090 GPU. Using the Stochastic Gradient Descent (SGD) [3] algorithm for optimization with an initial learning rate of 2×10^{-4}, decay strategy "poly" [4], batch size of 4, momentum of 0.9. All CNN-based methods are trained using the same configuration and settings without the need for pre-training.

3.2 Comparative Experiments

Comparative Methods. This paper compares a machine learning-based cloud detection method: SVM [11], and also compares six state-of-the-art deep learning-based cloud detection algorithms: FCN-8 [17], DeeplabV3+ [6], CDNetV1 [30], CDnetV2 [9], LWCDnet [18], Boundarynet [27]. Among these, LWCDnet is a lightweight cloud detection method.

Table 1. Quantitative comparisons with other cloud detection methods on the GF-1 test set. Cloud extraction accuracy (%)

Method	Jaccard index	precision	recall	F1-score	OA	MIoU
SVM	62.01	81.17	71.62	73.61	89.20	72.72
FCN-8	72.94	80.20	86.96	83.15	92.23	79.66
deeplabV3+	77.82	83.95	89.78	87.14	94.14	83.59
CDnetV1	80.77	86.45	91.34	89.22	94.65	85.85
CDnetV2	76.80	82.86	89.78	86.33	93.86	82.9
LWCDnet	75.57	80.99	87.69	83.81	93.03	82.61
Boundarynet	83.14	**90.68**	89.90	90.60	**95.88**	87.68
LigCDnet	**84.29**	90.42	**92.18**	**91.11**	**95.88**	**88.35**

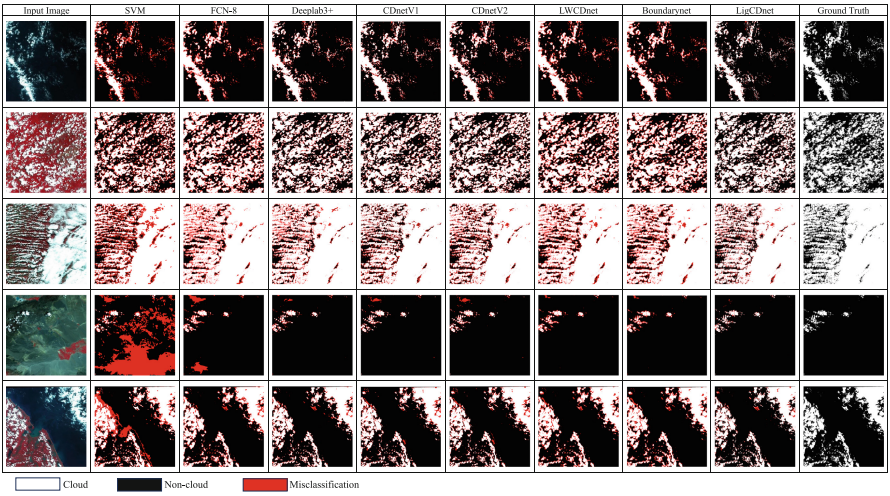

Fig. 5. Visual comparisons of different cloud detection methods on GF-1 dataset.

Cloud Detection Results on GF-1 Dataset: Table 1 reports the results of different cloud detection methods in the GF-1 dataset. From the results, our proposed LigCDnet outperforms most of them. Compared to the SVM machine learning method, deep learning methods have significant advantages in various metrics. FCN-8, in terms of Jaccard index, recall, and F1-score, shows an average improvement of 12% over SVM. Compared to cloud detection methods, the boundarynet achieves slightly higher precision, with the same score as ours on OA. However, in terms of Jaccard index and MIoU, our method is higher than it by 1.15% and 0.67%, respectively. Compared to the lightweight method LWCDnet, our proposed method outperforms LWCDnet by 7.3% and 5.74% in terms of F1-score and MIoU metrics, respectively. Fig. 5 shows a visual comparison of five typical examples of cloud segmentation methods in the GF-1 dataset, with

a variety of cloud cover and backgrounds. For clarity, we use white to represent correctly labeled cloud pixels and black to represent non-cloud pixels. Red markings indicate misclassified pixels. From a visual perspective, SVM's performance is the poorest; it only extracts the physical features of the image and does not fully comprehend the context of the image. CDnetV2 tends to misclassify bright objects as clouds. Overall, our LigCDnet performs the best.

Table 2. Quantitative comparisons with other cloud detection methods on the LandSat8 test set. Cloud extraction accuracy (%)

Method	Jaccard index	precision	recall	F1-score	OA	MIoU
SVM	72.38	86.72	77.16	80.74	85.10	65.12
FCN-8	79.09	85.19	83.81	87.58	93.09	74.00
deeplabV3+	81.30	86.61	87.38	87.26	92.06	76.74
CDnetV1	84.83	90.72	86.52	87.74	94.61	81.91
CDnetV2	79.68	86.89	84.83	86.48	93.64	79.07
LWCDnet	82.09	85.48	86.78	88.42	93.44	76.20
Boundarynet	83.71	89.25	87.28	90.20	94.64	80.55
LigCDnet	**88.02**	**93.99**	**90.16**	**92.25**	**95.00**	**84.39**

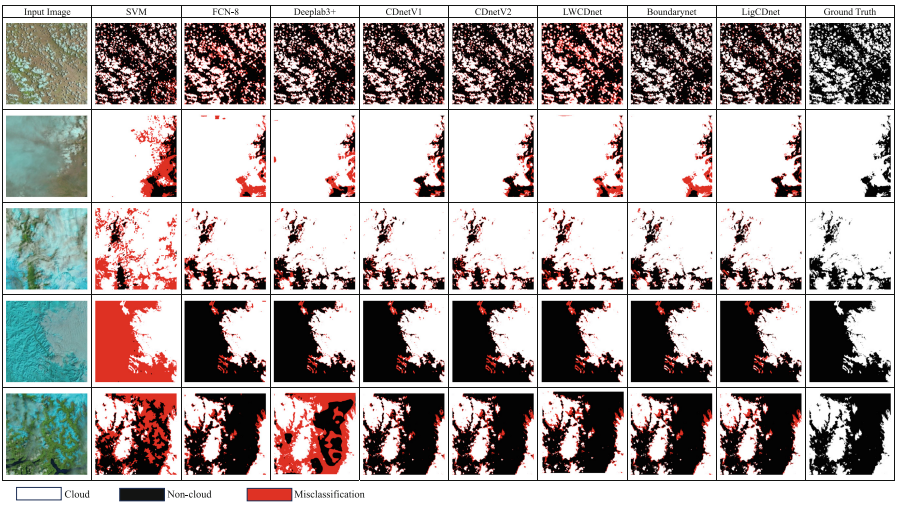

Fig. 6. Visual comparisons of different cloud detection methods on LandSat8 dataset.

Cloud Detection Results on LandSat8 Dataset: Table 2 reports the results of different cloud detection methods on the LandSat8 dataset. From the results, it can be observed that our proposed LigCDnet achieves better performance, especially in terms of Jaccard index and MIoU. While LigCDnet's OA is only 0.39% higher than CDnetV1, there is a significant improvement of 3.19% in MIoU. Compared to the lightweight network LWCDnet, our proposed network still demonstrates clear advantages, with a 3.38% higher recall and an 8.19% higher MIoU score. Figure 6 illustrates five examples from the LandSat 8 dataset, these examples encompass various backgrounds, such as situations where thin clouds and cloud ice coexist. From the visual results, it is evident that SVM performs poorly in handling scenarios where ice and snow coexist. DeeplabV3+ and LWCDnet also exhibit significant errors when dealing with scenes containing thin clouds. In contrast, our method demonstrates the best overall performance in handling all complex scenarios. It has fewer false positives (highlighted in red) compared to other methods.

Computational Complexity Analysis: In Table 3, we utilized floating point operations (FLOPs) and the number of trainable parameters to assess the computational complexity of these networks. Due to the results of the efficiency evaluation being directly proportional to the input image size, the FLOPs results were computed from input images sized at $224 \times 224 \times 3$. From the table, it can be observed that our proposed network has the fewest parameters. Although our proposed method has 7.69% higher GFLOPs compared to the lightweight model LWCDnet, we demonstrate significant advantages in both quantitative and qualitative analyses on various datasets. This is because we employ the Channel Attention Module (CAM) multiple times to adjust the weights of feature map channels, and we have designed a Lightweight Feature Pyramid Module (LFPM) to capture features of multi-scale clouds.

Table 3. Computational Complexity Analysis Based on CNN Method

Method	GFLOPs (224×224)	Params (M)
FCN-8	26.57	32.9
deeplabV3+	33.19	39.75
CDnetV1	59.94	47.50
CDnetV2	14.27	67.08
LWCDnet	**3.10**	2.55
Boundarynet	97.62	88.87
LigCDnet	10.79	**2.39**

3.3 Ablation Study

The LigCDnet proposed by us consists of three modules, namely the lightweight feature extraction module (LFEM), the channel attention module (CAM) and the lightweight feature pyramid module (LFPM). To investigate the performance of different components in the network, we conducted an ablation analysis on the GF-1 dataset. Table 4 provides detailed quantitative results.

From the results, LigCDnet demonstrates the best performance. Decreasing any of the blocks results in a certain degree of degradation in network performance. Removing CAM results in a deterioration of the metrics, indicating that CAM adjusts the weights of different channels in the feature maps, allowing channels favorable for the detection task to play a major role. Without LFPM, the metrics show a decrease, which suggests that LFPM can capture cloud features at different scales. Overall, these three modules play important roles in the cloud detection task.

Table 4. Ablation study on the GF-1 dataset by our LigCDnet with different modules

Method	Jaccard index	precision	recall	F1-score	OA	MIoU
LFEM	82.11	88.66	90.64	89.74	95.41	86.76
LFEM+CAM	82.85	88.31	91.88	90.23	95.70	87.40
LFEM+LFPM	83.15	89.44	91.54	90.07	95.76	87.66
LigCDnet	**84.29**	**90.42**	**92.18**	**91.11**	**95.88**	**88.35**

4 Conclusions

This article proposes a lightweight method (LigCDnet) for cloud detection. Compared with existing cloud detection models, LigCDnet achieves the best detection accuracy with a minimal number of parameters. In LigCDnet, we extensively extract multi-scale contextual features and further enhance segmentation accuracy by adjusting the channel weights of the feature maps. In the encoder, LFEM effectively extracts the semantic information of clouds, while CAM enhances feature map channels beneficial for the detection task and suppresses feature map channels that interfere with segmentation accuracy. Due to the diverse morphology of clouds, LFPM efficiently captures contextual features at different scales. In the decoder, the feature maps are gradually restored to the size of the input image through skip connections. Extensive experiments have been conducted on GF-1 and LandSat8 datasets, and the results show that LigCDnet can achieve excellent performance while reducing computational effort.

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Anal. Mach. Intell. **34**(11), 2274–2282 (2012)
2. An, Z., Shi, Z.: Scene learning for cloud detection on remote-sensing images. IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. **8**(8), 4206–4222 (2015)
3. Bottou, L.: Stochastic gradient descent tricks. In: Montavon, G., Orr, G.B., Müller, K.-R. (eds.) Neural Networks: Tricks of the Trade. LNCS, vol. 7700, pp. 421–436. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35289-8_25
4. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 834–848 (2017)
5. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
6. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 833–851. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_49
7. Cihlar, J., Howarth, J.: Detection and removal of cloud contamination from AVHRR images. IEEE Trans. Geosci. Remote Sens. **32**(3), 583–589 (1994)
8. Guo, J.h., Yang, F., Tan, H., Wang, J.x., Liu, Z.h.: Image matching using structural similarity and geometric constraint approaches on remote sensing images. J. Appl. Remote Sens. **10**(4), 045007–045007 (2016)
9. Guo, J., Yang, J., Yue, H., Tan, H., Hou, C., Li, K.: Cdnetv2: CNN-based cloud detection for remote sensing imagery with cloud-snow coexistence. IEEE Trans. Geosci. Remote Sens. **59**(1), 700–713 (2020)
10. Hafizah, W.M., Supriyanto, E., Yunus, J.: Feature extraction of kidney ultrasound images based on intensity histogram and gray level co-occurrence matrix. In: 2012 Sixth Asia Modelling Symposium, pp. 115–120. IEEE (2012)
11. Hao, Q., Zheng, W., Xiao, Y.: Fusion information multi-view classification method for remote sensing cloud detection. Appl. Sci. **12**(14), 7295 (2022)
12. Huang, C., et al.: Automated masking of cloud and cloud shadow for forest change analysis using landsat images. Int. J. Remote Sens. **31**(20), 5449–5464 (2010)
13. Li, P., Dong, L., Xiao, H., Xu, M.: A cloud image detection method based on SVM vector machine. Neurocomputing **169**, 34–42 (2015)
14. Li, Y., Yu, R., Xu, Y., Zhang, X.: Spatial distribution and seasonal variation of cloud over china based on ISCCP data and surface observations. J. Meteorol. Soc. Jpn. Ser. II **82**(2), 761–773 (2004)
15. 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)
16. Long, J., Shi, Z., Tang, W., Zhang, C.: Single remote sensing image dehazing. IEEE Geosci. Remote Sens. Lett. **11**(1), 59–63 (2013)
17. 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)
18. Luo, C., et al.: LWCDnet: a lightweight network for efficient cloud detection in remote sensing images. IEEE Trans. Geosci. Remote Sens. **60**, 1–16 (2022)

19. Mohajerani, S., Saeedi, P.: Cloud-net: an end-to-end cloud detection algorithm for landsat 8 imagery. In: IGARSS 2019–2019 IEEE International Geoscience and Remote Sensing Symposium, pp. 1029–1032. IEEE (2019)
20. Paszke, A., et al.: Pytorch: an imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
21. Rossow, W.B., Garder, L.C.: Cloud detection using satellite measurements of infrared and visible radiances for ISCCP. J. Clim. **6**(12), 2341–2369 (1993)
22. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv 2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4510–4520 (2018)
23. Shi, C., Wang, Y., Wang, C., Xiao, B.: Ground-based cloud detection using graph model built upon superpixels. IEEE Geosci. Remote Sens. Lett. **14**(5), 719–723 (2017)
24. Sun, L., et al.: A cloud detection algorithm-generating method for remote sensing data at visible to short-wave infrared wavelengths. ISPRS J. Photogramm. Remote. Sens. **124**, 70–88 (2017)
25. Suthaharan, S., Suthaharan, S.: Support vector machine. Machine learning models and algorithms for big data classification: thinking with examples for effective learning, pp. 207–235 (2016)
26. Wang, Y., et al.: Lednet: a lightweight encoder-decoder network for real-time semantic segmentation. In: 2019 IEEE International Conference on Image Processing (ICIP), pp. 1860–1864. IEEE (2019)
27. Wu, K., Xu, Z., Lyu, X., Ren, P.: Cloud detection with boundary nets. ISPRS J. Photogramm. Remote. Sens. **186**, 218–231 (2022)
28. Wu, K., Xu, Z., Lyu, X., Ren, P.: Cross-supervised learning for cloud detection. GISci. Remote Sens. **60**(1), 2147298 (2023)
29. Yang, F., Guo, J., Tan, H., Wang, J.: Automated extraction of urban water bodies from zy-3 multi-spectral imagery. Water **9**(2), 144 (2017)
30. Yang, J., Guo, J., Yue, H., Liu, Z., Hu, H., Li, K.: CDnet: CNN-based cloud detection for remote sensing imagery. IEEE Trans. Geosci. Remote Sens. **57**(8), 6195–6211 (2019)
31. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015)
32. Zhang, Z., Zhang, X., Peng, C., Xue, X., Sun, J.: ExFuse: enhancing feature fusion for semantic segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11214, pp. 273–288. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01249-6_17
33. Zhou, P., Han, J., Cheng, G., Zhang, B.: Learning compact and discriminative stacked autoencoder for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. **57**(7), 4823–4833 (2019)