Pattern Recognition 129 (2022) 108713

Contents lists available at ScienceDirect

Pattern Recognition

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

ECDNet: A bilateral lightweight cloud detection network for remote sensing images

Chen Luo, Shanshan Feng, Xutao Li, Yunming Ye*, Baoquan Zhang, Zhihao Chen, YingLing Quan

Department of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China

ARTICLE INFO

Article history:
Received 29 September 2021
Revised 4 March 2022
Accepted 13 April 2022
Available online 14 April 2022

Keywords:
Lightweight network
Efficient cloud detection
Dual-branch architecture

ABSTRACT

Cloud detection is one of the critical tasks in remote sensing image pre-processing and it has attracted extensive research interest. In recent years, deep neural networks based cloud detection methods have surpassed the traditional methods (threshold-based methods and conventional machine learning-based methods). However, current approaches mainly focus on improving detection accuracy. The computation complexity and large model size are ignored. To tackle this problem, we propose a lightweight deep learning cloud detection model: Efficient Cloud Detection Network (ECDNet). This model is based on the encoder-decoder structure. In the encoder, a two-path architecture is proposed to extract the spatial and semantic information concurrently. One pathway is the detail branch. It is designed to capture low-level detail spatial features with only a few parameters. The other pathway is the semantic branch, which is mainly for capturing context features. In the semantic branch, a proposed dense pyramid module (DPM) is designed for multi-scale contextual information extraction. The number of parameters and calculations in DPM is greatly reduced by features reusing. Besides, a FusionBlock is developed to merge these two kinds of information. Then the extreme lightweight decoder recovers the cloud mask to the same scale as the input image step by step. To improve performance, boost loss is introduced without inference cost increment. We evaluate the proposed method on two public datasets: LandSat8 and MODIS. Extensive experiments demonstrate that the proposed ECDNet achieves comparable accuracy as the state-of-art cloud detection methods, and meantime has a much smaller model size and less computation burden.

1. Introduction

Cloud detection is an essential step in remote sensing. Two-thirds of our earth's surface is covered by clouds [38]. When the earth's surface is the objective of satellite images, cloud will be treated as noise and should be removed in pre-processing phase. This phase ensures the high quality of various remote sensing applications, such as land cover classification [38], environment observation [17], and vegetation engineering [30]. However, the strict bandwidth constraints for downlink transmission conflict with transferring high-resolution hyperspectral images from satellites [8]. Recently, the trend of deploying remote sensing applications directly on-board satellite attracts more and more attention [12]. Therefore, the cloud detection method, which can be implemented and executed on satellites, is in great demand.

In practice, to tackle the cloud detection problem, the traditional methods can be divided into two groups, i.e., threshold-based methods and machine learning-based methods. The threshold-based approaches utilize the spectral and wavelength difference between cloud and other objects for cloud identification [40]. However, the process of fetching an appropriate threshold is computationally expensive. In addition, for multi-spectral satellite imagery, which only has four bands (red, green, blue, and near-infrared), the threshold method is usually not robust enough. To overcome the limitation of threshold-based approaches, researchers adopted machine learning techniques in cloud detection, e.g., support vector machines [22], random forest [7]. Unfortunately, their performance highly relies on manual-crafted features. And these features usually contain insufficient distinguishable information. Therefore, it is difficult to discriminate cloud from other objects with them especially in complicated cases.

* Corresponding author.
E-mail address: yeyunming@hit.edu.cn (Y. Ye).

https://doi.org/10.1016/j.patcog.2022.108713
0031-3203/

Recently, with the rise of deep learning and its success in computer vision [27] and remote sensing image processing [13], cloud detection methods based on convolutional neural network (CNN) have been widely investigated [20,24,31]. In these methods, features are extracted via CNN automatically. And the derived methods have achieved significant improvement in performance. However, these CNN-based models [31] tend to require a great number of parameters to achieve satisfactory performance. And the satellite, as a space device, has limitations in onboard resources like storage, computation, and power. Therefore, it's impractical to directly deploy the existing CNN-based cloud detection methods onboard.

Since semantic segmentation is similar to cloud detection, e.g., they both associate each pixel of an image with a class label [27], another possible solution is deploying lightweight models for semantic segmentation tasks in onboard cloud detection. In recent decades, researchers have achieved significant improvement [10,27,28,39] in lightweight semantic segmentation for Natural Scene Images (NSIs). However, there is an obvious gap between natural scene images and Remote Sensing Images (RSIs). First, the RSIs have the characteristics that the feature variance of the object is bigger in intra-class and smaller in inter-class than that of NSIs. For example, because of the influence of background, the features of the thin cloud are different from the features of the cloud. Besides, snow and ice cloud have quite similar features in most RSI bands [31]. Second, the object boundary, especially the cloud boundary is unclear, e.g., the thin cloud situation [6]. Therefore, it is unreliable to directly apply the semantic segmentation methods.

Recently, some researchers have paid attention to efficient cloud detection methods. In [8], a CNN-based model is proposed for nanosatellites to select eligible images to transmit to the ground. Even though the network structure is designed with low power consumption and low latency in inference, its accuracy is low. In [16], a lightweight network for cloud detection is designed on Sentinel-2A images. In this model, the number of parameters is reduced by using depthwise separable convolution and sharing kernel between channels in feature extraction blocks. However, kernel sharing cannot reduce the computation complexity, and hence the computation burden still exists. In conclusion, there still exist challenges in onboard cloud detection.

We observe that the cloud pixels are usually not isolated from others. Even though the thin cloud pixel has different features from the cloud and its spectral feature is heavily influenced by background, when it's surrounded by cloud pixel, it's of high possibility to be cloud. Therefore, context information of multi-scale is crucial for cloud detection. And the first challenge is how to make full use of such multi-scale context information. Since the current lightweight cloud detection method only focuses on minimizing the number of parameters, to obtain both effective and efficient onboard cloud detection, the second challenge is reducing the computation complexity as well.

Specifically, we propose the Efficient Cloud Detection Net (ECDNet), a new lightweight encoder-decoder neural network architecture to achieve comparable performance with generic state-of-art cloud detection methods. Inspired by Yu et al. [34], the encoder part is designed as a two-pathway architecture. One pathway is the detail branch. To keep enough spatial information, it is designed with shallow layers and wide channels. To decrease parameter amounts and computation cost, the lightweight module GhostModule [9] is adopted in the detail branch. The other pathway is the semantic branch. Different from the detail branch, in the semantic branch, as much as multi-scale semantic context information is captured. To have a large receptive field and meanwhile keep the network lightweight, we propose a dense pyramid module (DPM). In this module, each layer has fewer channels, and the features of layers are reused via dense connection to decrease computation cost. Inside each layer, a feature pyramid module is designed to extract and concatenate features with different receptive fields. The proposed semantic branch consists of a stemblock and two DPMs. It incorporates a large context without parameter amount increment. Then the outputs of the detail branch and semantic branch are fused via our proposed fusion module (FM) to gain more comprehensive feature maps. At last, a lightweight decoder takes intermediate results of the encoder to compensate for lost spatial features in downsampling. And it resizes feature maps to the original size of input step by step.

The main contributions of our work can be summarized as follows:

a) We propose a neural network: ECDNet. It consists of a lightweight two-pathway encoder and an extremely lightweight decoder.
b) In the encoder, the dense pyramid module (DPM) is designed to have large and diverse receptive fields in feature extraction.
c) In the encoder, the fusion module (FM) is developed to fuse detail and semantic information more efficiently.
d) The experiment results on LandSat8 and MODIS demonstrate that ECDNet can achieve state-of-the-art performance.

2. Related work

2.1. Cloud detection methods

The most straightforward way to distinguish cloud from other objects is by utilizing differences in their spectral characteristics to calculate thresholds. Fmask [40], which used the Landsat Top of Atmosphere (TOA) reflectance and Brightness Temperature (BT) of Landsat images, produced a probability mask as the threshold for cloud detection. Wei et al. proposed an algorithm to dynamically determine a proper threshold [29]. And the database used in this algorithm is constructed with MODIS surface reflectance products. Recently, in Li et al. [14], the authors focused on improving the accuracy of cloud detection in images with special scenes, such as bright land surface and severe haze. They proposed an automatic cloud detection method based on thresholds to tackle this problem. Considering these methods are not robust enough in complex scenes, researchers have taken more attention to utilizing machine learning algorithms for cloud detection. In [1], a novel approach for cloud detection using machine learning and multi-feature fusion is proposed. It analyzed typical spectral, textural, and other feature differences between clouds and backgrounds comparatively. The typical machine learning method like Random Forest (RF) [7] has also been adopted in cloud detection. However, the machine learning-based algorithms take the manual-crafted feature as input. And the performance heavily relies on inputs.

In recent years, deep learning-based methods, especially those built on the convolutional neural network (CNN), are proven to be able to process diverse image information concurrently in an automatic way. The information includes texture, color, shape, and correlation between spatial information. And the methods based on deep learning in image processing and remote sensing applications [19] have achieved outstanding performance. Specifically, to tackle cloud detection problem, in Mohajerani and Saeedi [20] an end-to-end Fully Convolutional Network (FCN) is trained on multiple patches of LandSat8 images. To detect both thin cloud and thick cloud in complex background scenes, a novel CNN model is proposed in Shi et al. [24]. For more complex scenes in multi-spectrum satellite images, e.g., cloud and snow co-existence situation, a fully convolutional network [36] is designed to learn deep patterns for better performance. However, they require a great number of parameters to achieve satisfactory performance. Directly

applying these models on satellites, where the computation resource is usually limited, is impractical.

2.2. Lightweight semantic segmentation methods

The lightweight semantic segmentation methods [15] focus on associating each pixel of an optical image with a class label. Meanwhile, the models have fewer parameters and need less time in training and inference steps.

In the semantic segmentation task, many models improved the performance by increasing the receptive field. With a large receptive field, much semantic information can be captured. Instead of using a normal convolution with a large kernel size, the receptive field can be enlarged by using the atrous convolution. It enlarges the receptive field by setting the dilation rate larger than 1. Besides, the number of parameters in the model doesn't increase. The atrous convolution is first introduced in DeepLab [3] for semantic segmentation task and it has achieved great performance improvement. In [32], it shows that when the model is equipped with combined atrous convolutions and each of them has different dilation rates, the model gains the capability to capture multi-scale object information. In [2], the atrous convolutions with different dilation rates were used to build a pyramid dilated module. However, the "gridding issue" [26] caused by dilated convolution leads to an inevitable accuracy decrease in performance.

To make the model more efficient, depthwise separable convolution is employed in model [5]. It is combined with a depthwise convolution and a pointwise convolution. In the depthwise convolution, each filter channel only works at one input channel, and the number of output features channels is the same as that of input. In pointwise convolution, 1×1 kernel is used to fuse the output channel features from depthwise convolution and increase the image depth. In [37], the block in ResNet is replaced by the bottleneck built on depthwise separable convolution. The number of parameters in this reduces significantly, besides, it is proven that the segmentation accuracy is improved. In the deeplab v3+ model [3] convolution operations are replaced by depthwise separable convolution operation and the modified Xception model is proven more efficient. Even though the depthwise separable convolution has much fewer parameters than standard convolution, the 1×1 convolution operation in pointwise convolution is computation consuming [37].

3. Method

3.1. Overview of efficient cloud detection network (ECDNet)

In Fig. 1 (Page 9), the Efficient Cloud Detection Network (ECDNet) is depicted, which is based on an encoder-decoder architecture.

In the encoder, to capture the spatial and multi-scale context information separately, a two-path architecture is adopted. It ensures that the diversity of features is taken into account to strengthen the expressive ability of the feature map. The detail branch is designed as a 3-stage ResNet module with shallow layers to extract detail spatial features. And the semantic branch has, in contrast, 3-stages with deeper layers to extract multi-scale long-range context features. The first stage in the semantic branch is a StemBlock. It can help extract the diverse feature and downsample the feature maps with the combination of a normal convolution and an average pooling operation. Both of them have stride equals 2. Then, 2 DPMs are stacked sequentially in the following stages for context feature extraction with much fewer parameters and less computation complexity. In the encoder, the FusionBlock is designed as a two-path architecture to merge the output feature maps from detail and semantic branches in an efficient way. In FusionBlock, in addition to an element-wise addition operation, the contextual information is processed via the sigmoid activation function to guide the response of the detail branch feature map.

In the decoder, the output feature map of the encoder is restored to the size of the input image step by step. At each restore stage, the output feature maps from FusionBlocks are concatenated to compensate for the lost low-level features. In detail, the output feature map of the encoder is the output of the third stage FusionBlock. It is first up-sampled with scale-factor 2. Then the output is concatenated with the output of the second FusionBlock. The feature map of concatenating operation is then further up-sampled with scale-factor 2. Finally, the output of upsampling is concatenated with the output of the first stage FusionBlock. The results are resized to the same size as the input image.

Here, the main loss is calculated between the inference result and the ground truth. Inspired by Bisenet-V2 [33], in addition to the main loss, two boost loss values are added to supervise the training. These two loss values L_{s2} and L_{s3} are calculated against the output of the 2nd and 3rd stages of the semantic branch separately to enhance the feature representation by supervising the feature extraction in diverse feature levels. Before the loss calculation, outputs of stages are first scaled to the size of the input image in SegHead. Finally, these two losses are weighted by parameters α and β. The total loss can be formulated as:

$$L = L_{main} + \alpha \times L_{s2} + \beta \times L_{s3}, \tag{1}$$

here, L_{main} is the main loss, L_{s2} and L_{s3} represents the additional loss from the 2nd and 3rd stage in semantic branch separately. The boost loss is deployed in training phase and discarded in inference phase. For each loss calculation, cross entropy loss function is adopted:

$$L_{CrossEntropy} = -\sum_{c=1}^{C}(p_c \times logq_c), \tag{2}$$

where p_c denotes the reference cloud mask and q_c is the predicted mask. There are totally C classed, here C equals 2: cloud and non-cloud.

3.2. Detail branch

In the detail branch, to keep as much spatial information as possible, the network is designed to have shallow layers and high channel capacity in feature maps. Especially, the proposed detail information extraction branch has three stages: 1st stage contains a downsampling block and a GhostModule, each of 2nd and 3rd stage contains a downsampling block and a GhostLayer. And the GhostLayer consists of two stacked GhostModules with a residual connection [9]. To extract diverse features and downsample the feature map, in the downsampling block, two parallel manners are adopted: normal convolution with kernel size 3×3 and a max-pooling, both of them have stride equals 2. Then results from two branches are concatenated for strengthening feature expression ability. In GhostLayer, the GhostModule utilizes the pointwise convolution and depthwise convolution to generate features. The computation complexity is reduced by using such computationally cheaper operators. Table 1 (Page 11) states the layers in detail branch.

3.3. Semantic branch

3.3.1. Stemblock

To preserve diverse features and enhance the feature expression ability, as shown in Fig. 2 (Page 12) we use the StemBlock as in Szegedy et al. [25]. It consists of a normal convolution with kernel

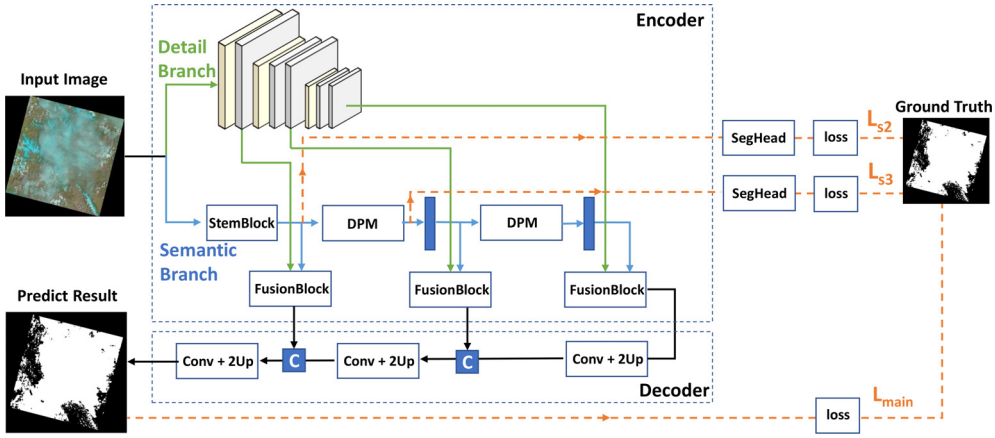

Fig. 1. Efficient Cloud Detection Network (ECDNet). It's an encoder-decoder architecture with training loss component. The encoder part is a two-way architecture, green lines represent the information flow in detail branch and blue lines represent that in semantic branch. In FusionBlock they are merged and then compensated to the decoder. The decoder part takes the extracted feature maps from encoder and here "C" presents the concatenate operation. In training phase, besides the main loss, two boost loss L_{s2} and L_{s3} are calculated against the intermediate outputs of semantic branch (orange dash line). L_{s2} and L_{s3} are discarded in inference phase. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Structure of detail branch, which is a three-stage architecture.

Stage	Operator	Output size	Channel	Stride
image	–	384 × 384	4	–
S1	Downsampler	192 × 192	32	2
	GhostModule	192 × 192	32	1
S2	Downsampler	96 × 96	64	2
	GhostLayer	96 × 96	64	1
S3	Downsampler	48 × 48	128	2
	GhostLayer	48 × 49	128	1

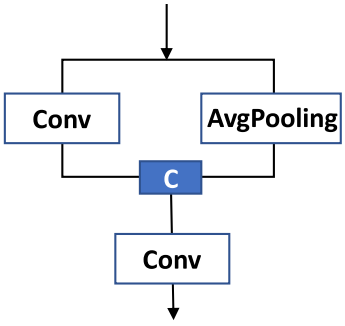

Fig. 2. The proposed StemBlock. It consists of a normal convolution with kernel size 3 × 3 and stride 2 and an average pooling operation. C is the concatenate operation.

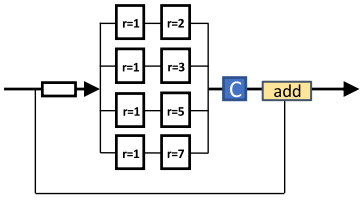

Fig. 3. The pyramid architecture consisted of 4 parallel path. In each the group convolution is with dilation rates of 2, 3, 5, 7. Then results are concatenated and added with an input skip connection.

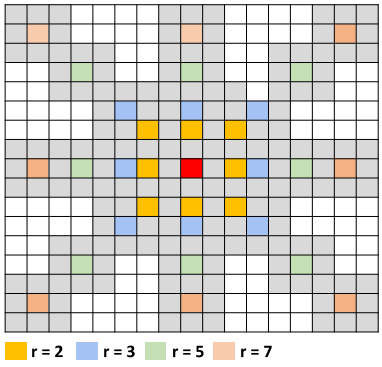

Fig. 4. Taking the dilation rates of prime number. Here, red pixel is the center. By taking sequential prime numbers, no "gridding issue" happens and receptive field is enlarged by concatenating the results. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

size 3 × 3 and an average pooling operation. Their outputs are concatenated as the output of StemBlock. The utilization of two downsampling methods helps reduce computation and meanwhile extract diverse information. In StemBlock, the feature map is downsampled with stride equals 2.

3.3.2. Dense pyramid module (DPM)

Since semantic information is critical for cloud detection, and diverse receptive field is significant for effective context feature extraction, we propose a novel module: dense pyramid module (DPM) to extract multi-scale semantic features in an extremely lightweight way. In DPM, the pyramid architectures are organized in dense connections to reuse the features. In this way, the number of parameters and computation complexity of the model are greatly reduced by feature reusing.

Pyramid architecture The Pyramid Architecture is shown in Fig. 3 (Page 13), it is designed to gain the feature from diverse receptive fields. The input feature map is first projected from high-dimensional to low-dimensional, and each parallel path takes a feature map with low channel capacity as input. To extract multi-scale features with a small number of parameters, in each path, two group convolution is stacked with dilation rates of 1 and r. Here to alleviate the impact of the "gridding issue", we adopted 4 following prime numbers: 2, 3, 5, 7 in each path, as in Fig. 4 (Page 13). And then output feature maps, which contain multi-scale context features, are concatenated. To strengthen feature propagation and encourage feature reusing, a skip connection of the input feature is added to the final pyramid features.

Pyramid module in dense connection In DPM, the Pyramid Architecture is designed to be organized in dense connection to reuse the feature by creating short paths from early layers to later layers. Besides, the feature propagation is strengthened with short

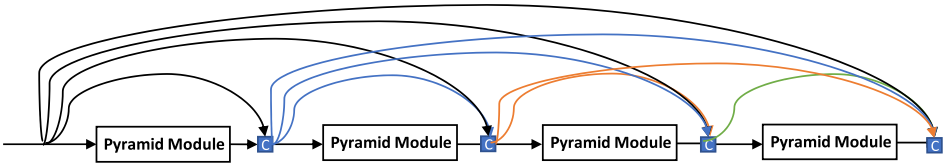

Fig. 5. In dense pyramid module (DPM), the pyramid module is connected in dense connection for feature reusing.

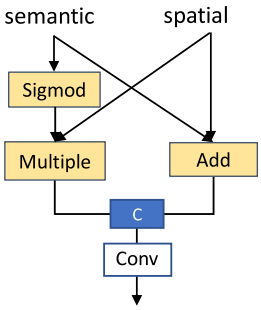

Fig. 6. The proposed Fusion Block, consisting of two paths. Sigmoid is the sigmoid activation function, Multiple is the element-wise multiple operation, Add is the element-wise addition operation.

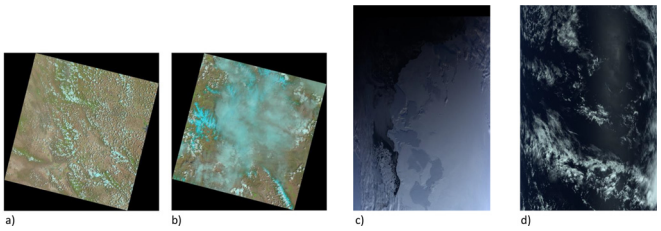

Fig. 7. Samples of several false color images. a and b are from LandSat8; c and d are from MODIS.

connections. The dense connection has been proven to achieve improved performance with much fewer parameters. In our proposed DPM, 4 pyramid modules are organized in dense connection as in Fig. 5 (Page 14). Hence, the reused features with diverse receptive fields make the derivative feature map more expressive with fewer parameters and computation cost.

3.4. Fusion block

The feature maps extracted from the detail branch and semantic branch can be merged differently, for example, the most common way is element-wise addition or submission. However, the outputs of two branches represent the different levels of features, when simply combing them, the characteristics of each output feature are ignored. It causes a degradation in performance and makes the optimization harder. Based on such analysis, we design a two-way fusion block to fuse the diverse feature map efficiently. As shown in Fig. 6 (Page 15), in the left way, the contextual information from the semantic branch is deployed to guide the response of the detail branch feature map. In the right way, the feature maps of semantic branch and detail branch are simply element-wise added. Finally, features extracted from two branches are concatenated.

4. Experimental results and analysis

4.1. Datasets and experiments setup

To evaluate the proposed ECDnet, we compare ECDnet with other state-of-art cloud detection and semantic segmentation approaches on two remote sensing images (RSI) datasets: LandSat8 and MODIS. Then we conduct ablation experiments. The models are compared in performance and efficient aspect.

4.1.1. Datasets

LandSat8, which is firstly released in Mohajerani and Saeedi [20], is a remote sensing dataset for cloud detection. It's originally named as 38-Cloud dataset. In LandSat8, each image is of size 1000 × 1000 and has 4 corresponding spectral channels: red (band 4), green (band 3), blue (band 2), and near-infrared (band 5). The 38 LandSat8 images are divided for training (18 pics) and test (20 pics). For feeding in the model, images are cropped into patches of 384 × 384. Sequentially, the training set contains 8400 cropped patches as training samples, and in the test set, there are 9201 patches of the same size. As shown in Fig. 7 (Page 15), (a) and (b) are two sample images in the LandSat8 dataset, their background is easy to distinguish from the cloud. However, there are plenty of areas covered with thin clouds.

Another used dataset is from Moderate-Resolution Imaging Spectroradiometer (MODIS), which is an important earth observation system in NASA. The dataset contains the images from the MODIS Level 1B product. And there are mainly four scenes in the MODIS dataset: ocean, land, land and ocean, and polar glaciers. Since the cloud, ice and ocean have similar characteristics in RSI bands, cloud detection in such co-existence scenarios is challenging. In the data pre-processing step, we remove some channels. In those channels, cloud can not be separated from other objects. Besides, some problematic images have been removed. The final dataset consists of 1422 remote sensing images, and each image has ten selected informative channels (band 1, 3, 4, 18, 20, 23, 28, 29, 31, and 32) from the original 36 bands. Images are separated into a training set with 1192 samples, a validation set with 80 samples, and a test set with 150 samples. After cropping them into overlapping patches with size 512 × 512, the training set contains 17,880 images, the validation set contains 1200 images, and the test set contains 2250 images. Fig. 7 (Page 15) (c) and (d) are samples from MODIS dataset. The scenes of Polar (c) and ocean (d) account for a large part of them, which makes cloud detection very difficult. MODIS cloud detection dataset is available at https://github.com/xiachangxue/MODIS-Dataset-for-Cloud-Detection.

4.1.2. Experimental setup

The evaluated networks equipped with our proposed modules are all implemented with the Pytorch framework and optimized by the Adam optimizer. The training step runs on Ubuntu 16.05 with two RTX 3090 GPUs in 100 epochs. The learning rate starts from 0.01 and decays with the policy that from epoch 36 the learning rate decays to 0.008, from epoch 65 decays to 0.005, and from epoch 85 decays to 0.003. All other methods are trained with the same configuration and settings.

4.1.3. Evaluation metrics

For each pixel, the prediction result and ground truth have two classes: cloud and non-cloud. The performance is measured via metrics widely used in semantic segmentation including Jaccard Index, Precision, Recall, MIoU, F1-score, and Overall Accuracy.

Table 2
Comparison with cloud detection methods and semantic segmentation methods on LandSat8 Dataset. The last line shows the difference between our ECDNet and the highest value of other methods.

Category	Methods	Jaccard	Precision	Recall	MIoU	Overall	F1-Score
Cloud Detection Methods	FCN [18]	83.2	**95.3**	86.7	87.7	94.4	90.8
	U-Net [21]	85.0	93.2	90.6	88.9	94.9	91.9
	Deeplab [3]	84.7	92.5	90.8	88.6	94.7	91.6
	RS-Net [11]	85.8	93.4	91.4	89.5	95.2	92.4
	MF-CNN [23]	87.0	95.1	91.0	90.4	95.6	93.0
	MFGNet [35]	85.2	93.3	95.6	89.3	95.3	94.4
	CDFM [16]	87.4	94.4	96.2	91.0	96.0	95.3
Semantic Segmentation Methods	MobileNet [10]	85.5	93.5	95.4	89.5	95.3	94.4
	ICNet [39]	85.4	93.0	95.5	89.4	95.3	94.2
	DFAnet [15]	85.4	92.1	93.1	86.3	93.7	92.6
	LEDNet [27]	85.4	93.5	95.4	89.5	95.3	94.4
	HANet [4]	87.7	94.0	96.3	91.2	**96.1**	95.1
	BiseNet-V2 [33]	**87.9**	94.6	**96.4**	**91.3**	96.0	**95.5**
	ECDNet	**88.3**	95.1	96.0	**91.6**	**96.2**	**95.5**
	+/− diff	+0.4	−0.2	−0.4	+0.3	+0.1	+0.0

The bold values show the best values in comparison.

These metrics are defined as follows:

$$JaccardIndex = \frac{TP}{TP + FN + FP}, \tag{3}$$

$$Precision = \frac{TP}{TP + FP}, \tag{4}$$

$$Recall = \frac{TP}{TP + FN}, \tag{5}$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{6}$$

$$OverallAccuracy = \frac{TP + TN}{TP + TN + FN + FP}, \tag{7}$$

$$MIoU = \frac{1}{k+1} \sum_{i=0}^{k} \frac{p_{ii}}{\sum_{i=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}}, \tag{8}$$

here, TP, TN, FP, and FN are the total number of true positive, true negative, false positive, and false negative pixels, respectively. The Jaccard Index and Mean Intersection over Union (MIoU) are two widely adopted metrics for measuring the performance in image segmentation tasks. MIoU combines the accuracies of both cloud pixels and non-cloud pixels in the calculation.

Besides, to measure the efficiency of models, in LandSat8, we collected the metrics including GFLOPs (Giga floating-point operations per second), the number of model parameters, time spent for inference per image, model size, and memory occupied by model.

4.2. Performance evaluation

As the baseline, the comparison methods include cloud detection models: FCN [18], UNet [21], Deeplab [3], RS-Net [11], MF-CNN [23], MFGnet [35], CDFM [16] and lightweight semantic segmentation methods: MobileNet [10], ICNet [39], DFAnet [15], LEDNet [27], HANet [4], BisenetV2 [33].

4.2.1. Cloud detection results on LandSat8 dataset

Table 2 (Page 18) reports results of different cloud detection methods and lightweight semantic segmentation methods on the LandSat8 dataset. From the results, in comparison with cloud detection methods, our proposed ECDNet outperforms most of them. For metrics Jaccard, MIoU, Overall, and F1-Score, the proposed ECDNet achieves a higher score than other methods. And FCN obtains a slightly higher score only in Precision, however, its recall value is much worse than that of our methods. And in comparison with the Bisenet-V2, the Recall score of our ECDNet is slightly decreased by 0.4, since more convolution operations and wider hidden feature maps are adopted in the Bisenet-V2. The results show that the performance of our proposed ECDNet can achieve the best metric scores of state-of-the-art cloud detection methods. In comparison with lightweight semantic segmentation methods, our ECDNet is only inferior to Bisenet-V2 with a tiny gap. For all metrics, our model can achieve satisfactory performance.

Fig. 8 (Page 19) shows the visual comparison of cloud segmentation methods on three examples from the LandSat8 dataset. The examples include a diverse background, e.g., thin cloud and cloud-ice coexisting cases. From the results, its significant that ECDNet obtains comparable performance in normal cases. In addition, it performs even better in some complicated situations like thin cloud. Fig. 9 (Page 20) shows the visual results of cloud segmentation methods, our proposed model can accurately identify the cloud in diverse backgrounds. Its obvious that the proposed methodology can achieve better or comparable performance in cloud detection.

4.2.2. Cloud detection results on MODIS dataset

Table 3 (Page 21) reports the results of different cloud detection and semantic segmentation methods on MODIS dataset data. Compared to the cloud detection methods, our ECDNet achieves better or similar performance. Compared to MF-CNN and CDFM which have the highest value in MIoU, our method ECDNet has a 0.8% decrease in MIoU. Especially, ECDNet performs best in all metrics in comparison with other lightweight semantic segmentation methods. It's caused by that, in MODIS dataset, the special scenes, such as oceans and Polar ice regions have much more requirements for cloud boundary processing in cloud detection methods. However, to keep the parameter and size of the model at a low level the lightweight model usually adopted the dilate revolution or multi-resolution, they all miss some detail information in the process. The best available trade-off between accuracy and efficiency is the target in the lightweight model.

Fig. 10 (Page 21) shows the visual comparison of cloud detection methods with our ECDNet on three examples from MODIS dataset. The false-color images show that the cloud in imagery in MODIS has more detail information than that in LandSat8. However, in such a situation, our ECDNet has comparable performance in most normal cases. Fig. 11 (Page 22) shows the visual results of semantic segmentation methods, our proposed model has much better or even comparable results in scenes.

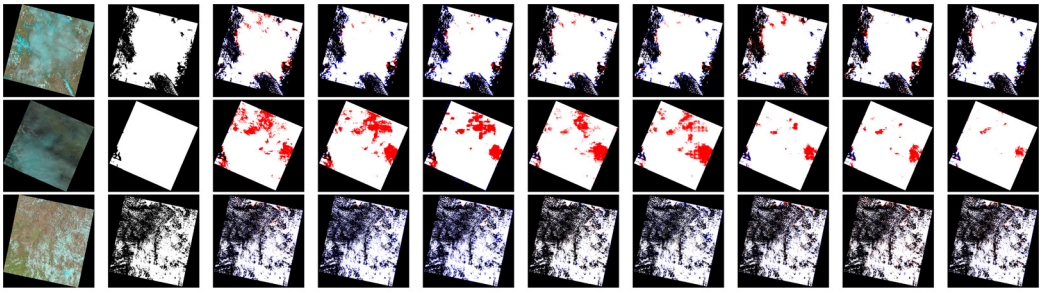

Fig. 8. The visual comparisons of different cloud detection methods in the scene of three examples from LandSat8 dataset. White area represents cloud and black area represents non-cloud. Besides, red area represents false positive detection and blue area represents false negative detection. (a) denotes the false-color remote sensing image; (b) is the ground-truth; (c) Result of FCN [18]; (d) Result of U-Net [21]; (e) Result of DeepLab [3]; (f)Result of RS-Net [11]; (g) Result of MF-CNN [23]; (h) Result of MFGNet [35]; (i) Result of CDFM [16]; (j) Result of our proposed ECDNet. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

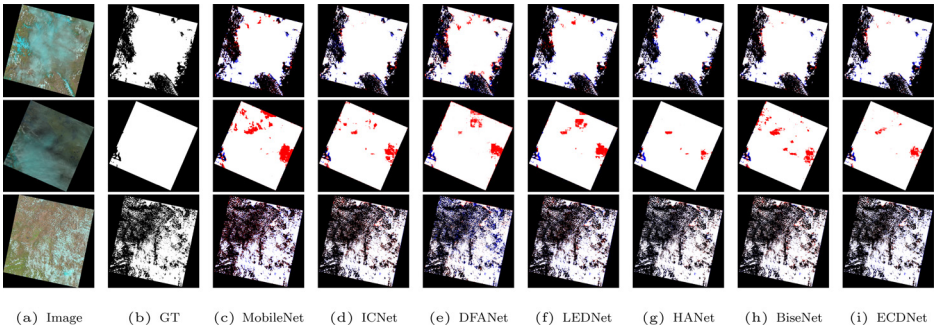

Fig. 9. The visual comparisons of different lightweight semantic segmentation methods in the scene of three examples from LandSat8 dataset. White area represents cloud and black area represents non-cloud. Besides, red area represents false positive detection and blue area represents false negative detection. (a) denotes the false-color remote sensing image; (b) is the ground-truth; (c) Result of MobileNet [10]; (d) Result of ICNet [39]; e) Result of DFAnet [15]; (f) Result of LEDNet [27]; (g) Result of HANet [4]; (h) Result of BiseNet-v2 [33] (i) Result of our proposed ECDNet. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
Comparision with cloud detection and lightweight semantic segmentation methods on MODIS dataset. The last line shows the difference between our ECDNet and the highest value of other methods.

Category	Methods	Jaccard	Precision	Recall	MIoU	Overall	F1-Score
Cloud Detection Methods	FCN [18]	89.4	92.3	91.6	86.3	93.1	92.0
	UNet [21]	89.7	92.6	91.5	86.8	93.2	92.1
	Deeplab [3]	86.9	91.0	91.9	82.8	91.1	91.5
	RS-Net [11]	89.7	**95.2**	92.5	86.6	93.3	**93.8**
	MF-CNN [23]	90.0	92.4	**93.5**	**86.9**	**93.4**	93.0
	MFGNet [35]	87.7	90.5	**93.5**	83.5	91.6	92.0
	CDFM [16]	**89.9**	93.2	92.7	**86.9**	**93.4**	92.9
Semantic Segmentation Methods	MobileNet [10]	82.2	87.3	86.5	76.9	87.7	81.4
	ICNet [39]	85.2	87.5	88.3	79.8	89.6	87.9
	DFAnet [15]	81.3	85.9	84.6	74.5	86.5	85.3
	LEDNet [27]	85.8	89.8	88.7	80.6	90.1	89.3
	HANet [4]	87.8	91.5	91.4	84.3	92.0	87.7
	BiseNet-V2 [33]	88.7	92.2	92.0	85.3	92.5	92.1
	ECDNet	89.3	92.7	92.4	86.2	93.0	92.5
	+/− diff	−0.6	−2.5	−1.1	−0.7	−0.4	−1.3

The bold values show the best values in comparison.

4.3. Efficiency evaluation

The efficiency of the model is evaluated based on LandSat8. For other datasets, the metrics values can be calculated with the proportion between its input image size and that of LandSat8. The summarized results are stated in Table 4 (Page 23) and Fig. 12 (Page 23). Here, all metrics are collected in the test phase. As shown in the table, our proposed ECDNet has the least parameters and operations times. Fewer parameters meet the storage limitation on satellites. And fewer operations times means less power consumption is needed for inference. Because of the limited power onboard, to implement a model in satellites, the power consumption of the model should be as less as possible. Here the time/image is the average inference time with frame size 384 × 384. Data and parameters load time is not taken into account, and the employed GPU is NVIDIA Corporation Device 2204 with 24G storage. Our proposed ECDNet is of the smallest model size. Besides, it occupies less memory than all compared cloud detection methods and most lightweight semantic segmentation methods. The time spent per image of our ECDNet is 12 milliseconds longer than the fast inference model MobileNet [10] by comparing with the semantic segmentation methods and shorter than all

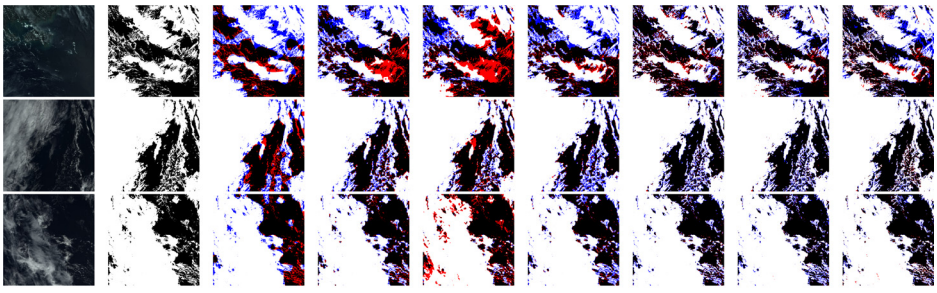

Fig. 10. The visual comparisons of different cloud detection methods in the scene of three examples from MODIS dataset. White area represents cloud and black area represents non-cloud. Besides, red area represents false positive detection and blue area represents false negative detection. a) denotes the false-color remote sensing image; b) is the ground-truth; c) Result of FCN [18]; d) Result of U-Net [21]; e) Result of DeepLab [3]; f)Result of RS-Net [11]; g) Result of MF-CNN [23]; h) Result of MFGNet [35]; i) Result of CDFM [16]; j) Result of our proposed ECDNet. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

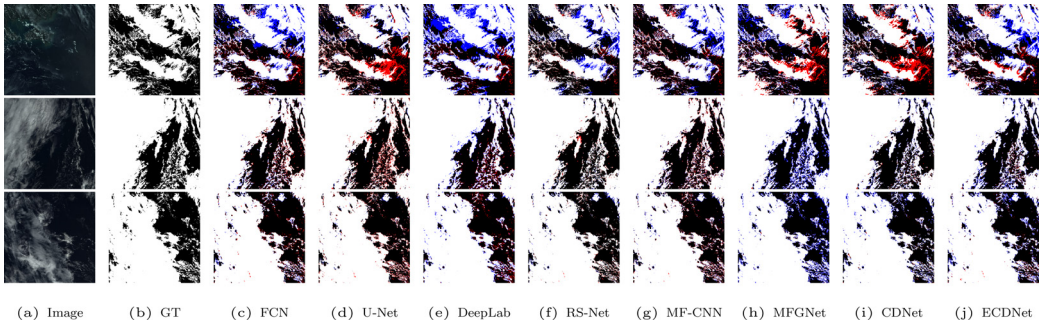

Fig. 11. The visual comparisons of different lightweight semantic segmentation methods in the scene of three examples from MODIS dataset. White area represents cloud and black area represents non-cloud. Besides, red area represents false positive detection and blue area represents false negative detection. (a) denotes the false-color remote sensing image; (b) is the ground-truth; (c) Result of MobileNet [10]; d) Result of ICNet [39]; (e) Result of DFAnet [15]; f) Result of LEDNet [27]; (g) Result of HANet [4]; (h) Result of BiseNet-v2 [33] i) Result of our proposed ECDNet. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4
Efficiency comparsion on LandSat8 dataset. The last show shows the differences between our method and the best scores.

Category	Methods	GFLOPs	Params (10^6)	Time/image (ms)	Model size (M)	Occupied Memory (M)
Cloud Detection Methods	FCN [18]	45.30	18.64	22.8	143	409.78
	UNet [21]	147.50	34.53	27.5	102	1163.25
	Deeplab [3]	50.00	59.34	38.0	454	607.02
	RS-Net [11]	92.40	7.85	25.0	60	498.90
	MF-CNN [23]	70.57	17.41	29.1	133	412.85
	MFGNet [35]	68.65	7.83	25.1	90	1613.01
	CDFM [16]	5.61	0.73	21.6	9.0	500.00
Semantic Segmentation Methods	MobileNet [10]	1.72	4.5	**8.4**	38	169.6
	ICNet [39]	1.88	4.98	20.9	58	**79.57**
	DFAnet [15]	1.01	2.17	43.1	26	124.45
	LEDNet [27]	3.56	0.92	30.0	11	213.17
	HANet [4]	39.13	40.96	28.1	470	436.53
	BiseNet-V2 [33]	7.25	3.47	12.1	42	175.3
	ECDNet	**0.47**	**0.087**	20.1	**2.1**	167.9
	best/ours	1/2.14	1/8.4	1/0.42	1/4.3	1/0.47

The bold values show the best values in comparison.

cloud detection methods. As stated in Zhang et al. [37], the inference time is closely related to parallelism. And the degree of parallelism is reduced by network fragmentation, therefore, the "multi-path" structure is unfriendly for devices with strong parallel computing powers like GPU. In MobileNet [10], each block consists of 2 or 3 convolution operations. And the block in BiseNet-V2 [33] consists of sequential convolution operations. However, our ECDNet has the pyramid module which contains 4 parallel convolution operations and dense pyramid modules (DPM) where feature maps are concatenated step by step. Therefore, ECDNet is more fragmental and needs more time in inference. In conclusion, our method achieves state-of-art performance in cloud detection and meanwhile is much more efficient than methods with comparable accuracy.

4.4. Study on hyper-parameters

In addition to the main loss, we introduce two boost losses. These boost losses are adopted in the training phase to improve the segmentation accuracy and discarded in the inference phase to avoid the increment of computation complexity. We calculate boost losses L_{s2} and L_{s3} against the outputs of the 2nd and 3rd stages of the semantic branch to supervise the feature extraction and segmentation in diverse feature levels. To balance the effect of

Table 5
Ablation study includes: number of stacked pyramid architectures in DPM (5(a)), number of adopted DPM layers (5(b)), dilation rates adopted in pyramid architecture in DPM (5(c)), and fusion methods for detail and context feature (5(d)).

Model	MIoU(%)	GFlops
(a) The ablation study of the number of stacked pyramid architectures (Fig. 3 (Page 13)) in the dense pyramid module (DPM).		
Pyramid-3	91.3	**0.46**
Pyramid-4	**91.6**	0.47
Pyramid-5	91.2	0.54
Model	MIoU(%)	GFlops
(b) The ablation study of the number of stacked dense pyramid modules (DPM) in semantic branch.		
DPM-1	91.1	**0.45**
DPM-2	**91.6**	0.47
DPM-3	90.2	0.51
Model	MIoU(%)	GFlops
(c) The ablation study of dilation rate: (2,3,5,7), (1,2,3,4), (1,6,12,18) utilized in pyramid architecture in DPM.		
Dilation rate (2,3,5,7)	**91.6**	**0.47**
Dilation rate (1,2,3,4)	91.0	**0.47**
Dilation rate (1,6,12,18)	91.4	**0.47**
Model	MIoU(%)	GFlops
(d) The ablation study of the fusion methods.		
FusionBlock	**91.6**	0.47
Addition	91.2	**0.43**
Multiplication	91.0	**0.43**

The bold values show the best values in comparison.

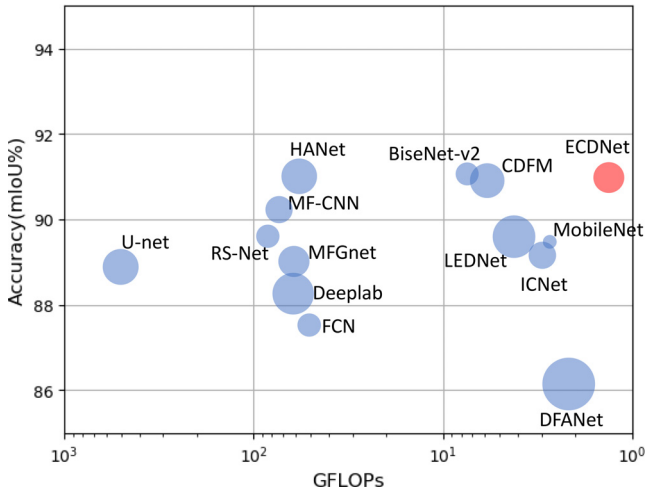

Fig. 12. GFLOPs, mIoU performance and FPS on LandSat8 data set. X-axis is the GFLOPs, Y-axis is the Accuracy presented by mIoU and the bubble size represents the inference time per image, when the bubble is smaller, the shorter time the method used in inference. Here our method is compared with cloud detection methods: FCN [18], U-net [21], Deeplab [3], RS-Net [11], MF-CNN [23], MFGnet [35] and real-time semantic segmenation methods, including MobileNet [10], ICNet [39], LEDNet [27], DFANet [15] and HANet [4].

main loss and boost losses, we introduce two hyper-parameters α and β. Fig. 13 (Page 25) shows the results of the study on these two hyper-parameters. The *X*-axis represents the value of α and *Y*-axis represents the value of β, and *Z*-axis shows the accuracy represented by MIoU of each run. For α and β, we select the values (0.1, 0.3, 0.5, 0.7, 0.9) and run tests on the combination of each two of them. From the results, when the α is 0.5 and β is 0.5 the proposed ECDNet has the best segmentation performance with MIoU 91.6.

4.5. Ablation studies

To evaluate the effectiveness and efficiency of ECDNet and the components included in it, we conduct our ablation studies on the LandSat8 dataset. The models are trained on their training set, tested on the test set, and the performance is evaluated by MIoU and GFlops.

Stacked pyramid architectures in DPM In the dense pyramid modules (DPM), the pyramid architectures (Fig. 3 (Page 13)) are stacked via the dense connection. The proposed ECDNet adopts DPM which consists of 4 pyramid architectures. To figure out the most proper number of pyramid architectures, we conduct the ablation study by varying the numbers 3, 4, and 5. In Table 5(a) (Page 26), the results show that when the number of pyramid architectures is 4, the model achieves the best performance with 91.6% MIoU. Besides, the calculation complexity increases by 0.01 GFlops in comparison with the model that adopts 3 stacked pyramid architectures in DPM.

DPM layers The dense pyramid modules (DPM) are stacked in the semantic branch for context feature extraction in our ECDNet. We adopt 1, 2, and 3 stacked DPMs in experiments to find out the most proper number. Table 5(b) (Page 26) shows the results by adopting 1, 2 and 3 DPMs. We can see that the 2 DPMs stacked model is the most effective, it outperforms the model adopted 1 DPM by 1% in MIoU with 0.02 increment in GFlops and the model adopted 3 DPM by 0.4% with 0.04 decrement in GFlops. Since more DPMs cannot guarantee improved performance, we stop the number of DPMs to 3. Finally, we use 2 DPMs to build the semantic branch.

Dilation rates in pyramid architecture Fig. 3 (Page 13) shows the pyramid architecture we adopted in the proposed DPM. The dilation rates in the model are consecutive prime numbers (2,3,5,7). To evaluate them, we conduct experiments by setting them to (1,2,3,4) and (1,6,12,18). In Table 5(c) (Page 26), different dilation rates are compared in effectiveness and efficiency. From the results, the dilation rates of (2,3,5,7) outperform (1,2,3,4) and (1,6,12,18) with 0.2% in MIoU and with these three dilation rates, the model has the same calculation complexity.

Fusion method for detail and context feature In ECDNet, we proposed the FusionBlock to merge the detail and context feature in a more efficient way. In Table 5(d) (Page 26), we compare the FusionBlock with two simple feature fusion methods: pixel-wise addition and multiplication. The FusionBlock has 0.2% and 0.3% MIoU improvement over Addition and Multiplication, with 0.04 GFlops increment. We can see that by adopting the FusionBlock, the cloud detection effectiveness can be improved with less calculation increment.

Paths in model The ECDNet is an encoder-decoder structure, the encoder part is a model of two-path architecture: semantic branch and detail branch. Besides, an additional loss is added in the train-

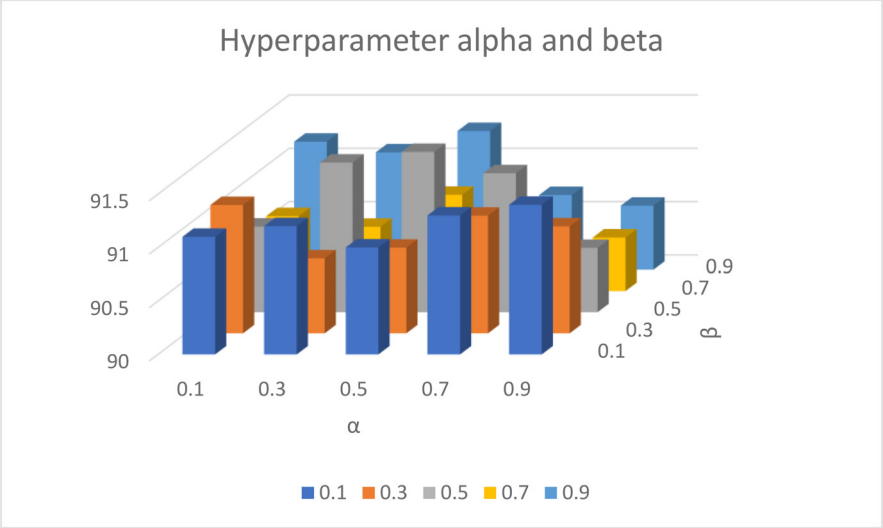

Fig. 13. Study on hyper-parameters α and β of boost loss values.

Table 6
The performance of network with combination of individual path, includes detail branch, semantic branch, decoder and the additional loss path (L_{s2} and L_{s3}) in training phase.

Detail branch	Semantic branch	Decoder	Main loss	Loss L_{s2}	Loss L_{s3}	MIoU(%)	GFlops
✓		✓	✓			90.1	0.34
	✓	✓	✓			87.7	**0.12**
✓	✓		✓			90.1	0.47
✓	✓	✓	✓			90.7	0.47
✓	✓	✓	✓	✓		91.1	0.47
✓	✓	✓	✓		✓	91.2	0.47
✓	✓	✓	✓	✓	✓	**91.6**	0.47

The bold values show the best values in comparison.

ing phase. To evaluate the individual path, we conduct experiments on the combination of them. Table 6 (Page 27) illustrates the experiment results. The results show that only the detail branch or semantic branch can achieve 89.4% or 87.6% in MIoU. When combing them, the MIoU is increased to 89.7%, and the computation complexity is the addition of two branches. By adding the decoder to the model, the model can achieve 89.7% MIoU, in addition, the increment in GFlops is little, which means that the decoder contributes lots to effectiveness with merely burden increment to computation. The 4th and 5th row shows the results of adding addition loss L_{s2} and L_{s3}. With the additional loss, the MIoU increases by 0.4% and 0.5%. From the observation that every path included in ECDNet contributes to the effectiveness of cloud detection and is of high efficiency.

5. Conclusion and future work

In this paper, we propose a lightweight method based on a deep convolutional neural network for cloud detection: ECDNet. It aims at tackling the onboard cloud detection problem on satellites. By considering that the satellites have limitations on computation, storage, and power resource, ECDNet is designed to be an extremely lightweight model with less performance degradation in comparison with existing state-of-art methods. The method is based on encoder-decoder architecture. In the encoder, lightweight two-pathway architecture is designed to parse the detail and semantic feature separately. In the decoder, the lost high-level feature in the encoder is compensated and the feature map is restored to the size of origin input step by step. In experiments we use datesets LandSat8 and MODIS. Here, the newly used dataset MODIS consists of 1422 remote sensing images with complex scenes. Each multi-spectral RSI in MODIS contains 10 channels from visible, near-infrared, and infrared bands. The results on both datasets demonstrate that ECDNet can achieve the best trade-off between performance and resource usage.

Although the ECDNet has satisfactory results in the lightweight cloud detection task, it is meaningful to extend the current model to multi-classes classification in the future. Different from the utilized LandSat8 and MODIS datasets, the multi-classes cloud detection targets on the dataset contain classification tags: cloud, thin cloud, cloud shadow, and snow. And in cloud detection, the recognition of snow/ice is difficult since they have a similar spectral feature to cloud and should be paid particular attention. Besides, the investigation on the non-local relationship between pixels will be conducted to further enforce the feature of cloud and suppress the influence of noise. However, the large memory consumption caused by relationship affinity matrix should be considered in lightweight model.

Acknowledgments

This work was supported by the Shenzhen Science and Technology Program under Grant No. JCYJ20210324120208022 and Grant No. JCYJ20200109113014456.

References

[1] T. Bai, D. Li, K. Sun, Y. Chen, W. Li, Cloud detection for high-resolution satellite imagery using machine learning and multi-feature fusion, Remote Sens. 8 (2016) 715.
[2] L. C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking atrous convolution for semantic image segmentation, arXiv preprint arXiv:1706.05587(2017).

[3] L.C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 801–818.
[4] S. Choi, J.T. Kim, J. Choo, Cars can't fly up in the sky: improving urban-scene segmentation via height-driven attention networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9373–9383.
[5] F. Chollet, Xception: deep learning with depthwise separable convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1251–1258.
[6] L. Di Girolamo, R. Davies, Cloud fraction errors caused by finite resolution measurements, J. Geophys. Res. 102 (1997) 1739–1756.
[7] N. Ghasemian, M. Akhoondzadeh, Introducing two random forest based methods for cloud detection in remote sensing images, Adv. Space Res. 62 (2018) 288–303.
[8] G. Giuffrida, L. Diana, F. de Gioia, G. Benelli, G. Meoni, M. Donati, L. Fanucci, Cloudscout: a deep neural network for on-board cloud detection on hyperspectral images, Remote Sens. 12 (2020) 2205.
[9] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, C. Xu, Ghostnet: more features from cheap operations, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1580–1589.
[10] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: efficient convolutional neural networks for mobile vision applications, arXiv preprint arXiv:1704.04861(2017).
[11] J.H. Jeppesen, R.H. Jacobsen, F. Inceoglu, T.S. Toftegaard, A cloud detection algorithm for satellite imagery based on deep learning, Remote Sens. Environ. 229 (2019) 247–259.
[12] V. Kothari, E. Liberis, N.D. Lane, The final frontier: deep learning in space, in: Proceedings of the 21st International Workshop on Mobile Computing Systems and Applications, 2020, pp. 45–49.
[13] K. Nogueira, O.A. Penatti, J.A.D. Santos, Towards better exploiting convolutional neural networks for remote sensing scene classification, Pattern Recognit. 61 (2017) 539–556.
[14] C. Li, J. Ma, P. Yang, Z. Li, Detection of cloud cover using dynamic thresholds and radiative transfer models from the polarization satellite image, J. Quant. Spectrosc. Radiat. Transf. 222 (2019) 196–214.
[15] H. Li, P. Xiong, H. Fan, J. Sun, Dfanet: deep feature aggregation for real-time semantic segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9522–9531.
[16] J. Li, Z. Wu, Z. Hu, C. Jian, S. Luo, L. Mou, X.X. Zhu, M. Molinier, A lightweight deep learning-based cloud detection method for sentinel-2a imagery fusing multiscale spectral and spatial features, IEEE Trans. Geosci. Remote Sens. (2021) 1–19.
[17] Z. Li, Y. Zhang, J. Shao, B. Li, J. Hong, D. Liu, D. Li, P. Wei, W. Li, L. Li, et al., Remote sensing of atmospheric particulate mass of dry PM2.5 near the ground: method validation using ground-based measurements, Remote Sens. Environ. 173 (2016) 59–68.
[18] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.
[19] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, B.A. Johnson, Deep learning in remote sensing applications: ameta-analysis and review, ISPRS J. Photogramm. Remote Sens. 152 (2019) 166–177.
[20] S. Mohajerani, P. Saeedi, 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, 2019, pp. 1029–1032.
[21] O. Ronneberger, P. Fischer, T. Brox, U-net: convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2015, pp. 234–241.
[22] R. Rossi, R. Basili, F. Del Frate, M. Luciani, F. Mesiano, Techniques based on support vector machines for cloud detection on quickbird satellite imagery, in: 2011 IEEE International Geoscience and Remote Sensing Symposium, IEEE, 2011, pp. 515–518.
[23] Z. Shao, Y. Pan, C. Diao, J. Cai, Cloud detection in remote sensing images based on multiscale features-convolutional neural network, IEEE Trans. Geosci. Remote Sens. 57 (2019) 4062–4076.
[24] M. Shi, F. Xie, Y. Zi, J. Yin, Cloud detection of remote sensing images by deep learning, in: 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), IEEE, 2016, pp. 701–704.
[25] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2017.
[26] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, G. Cottrell, Understanding convolution for semantic segmentation, in: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2018, pp. 1451–1460.
[27] Y. Wang, Q. Zhou, J. Liu, J. Xiong, G. Gao, X. Wu, L.J. Latecki, Lednet: a lightweight encoder-decoder network for real-time semantic segmentation, in: 2019 IEEE International Conference on Image Processing (ICIP), IEEE, 2019, pp. 1860–1864.
[28] M. Oršić, S. Šegvić, Efficient semantic segmentation with pyramidal fusion, Pattern Recognit. 110 (2021) 107611.
[29] J. Wei, L. Sun, C. Jia, Y. Yang, X. Zhou, P. Gan, S. Jia, F. Liu, R. Li, Dynamic threshold cloud detection algorithms for modis and landsat 8 data, in: 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), IEEE, 2016, pp. 566–569.
[30] J. Xue, B. Su, Significant remote sensing vegetation indices: a review of developments and applications, J. Sens. (2017) 17.
[31] J. Yang, J. Guo, H. Yue, Z. Liu, H. Hu, K. Li, Cdnet: CNN-based cloud detection for remote sensing imagery, IEEE Trans. Geosci. Remote Sens. 57 (2019) 6195–6211.
[32] M. Yang, K. Yu, C. Zhang, Z. Li, K. Yang, Denseaspp for semantic segmentation in street scenes, in: Proceedings of the IEEE Conference On Computer Vision and Pattern Recognition, 2018, pp. 3684–3692.
[33] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, N. Sang, Bisenet v2: bilateral network with guided aggregation for real-time semantic segmentation, arXiv preprint arXiv:2004.02147(2020).
[34] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, N. Sang, Bisenet: bilateral segmentation network for real-time semantic segmentation, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 325–341.
[35] J. Yu, Y. Li, X. Zheng, Y. Zhong, P. He, An effective cloud detection method for gaofen-5 images via deep learning, Remote Sens. 12 (2020) 2106.
[36] Y. Zhan, J. Wang, J. Shi, G. Cheng, L. Yao, W. Sun, Distinguishing cloud and snow in satellite images via deep convolutional network, IEEE Geosci. Remote Sens. Lett. 14 (2017) 1785–1789.
[37] X. Zhang, X. Zhou, M. Lin, J. Sun, Shufflenet: an extremely efficient convolutional neural network for mobile devices, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6848–6856.
[38] Y. Zhang, W.B. Rossow, A.A. Lacis, V. Oinas, M.I. Mishchenko, Calculation of radiative fluxes from the surface to top of atmosphere based on ISCCP and other global data sets: refinements of the radiative transfer model and the input data, J. Geophys. Res. 109 (2004) 19.
[39] H. Zhao, X. Qi, X. Shen, J. Shi, J. Jia, Icnet for real-time semantic segmentation on high-resolution images, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 405–420.
[40] Z. Zhu, C.E. Woodcock, Object-based cloud and cloud shadow detection in landsat imagery, Remote Sens. Environ. 118 (2012) 83–94.

Chen Luo is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. She received the B.Sc. degree from Xidian University, China in 2011, and received M.Sc. degree from Hannover University, Germany in 2014. Her research interests include deep learning, remote sensing and computer vision.

Shanshan Feng is currently an Associate Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. He received the Ph.D. degree in Computer Science from Nanyang Technological University, Singapre, in 2017. His research interests include sequential data mining and social network analysis.

Xutao Li is currently an Associate Professor with the Department of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. He received the Ph.D. and Master degrees in Computer Science from Harbin Institute of Technology in 2013 and 2009, and the Bachelor from Lanzhou University of Technology in 2007. His research interests include data mining, machine learning, graph mining, and social network analysis, especially tensor-based learning and mining algorithms.

Yunming Ye is currently a Professor with the Department of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. He received the Ph.D. degree in Computer Science from Shanghai Jiao Tong University, Shanghai, China, in 2004. His research interests include data mining, text mining, and ensemble learning algorithms.

Baoquan Zhang is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. He received the B.S. degree from the Harbin Institute of Technology, Weihai, China, in 2015, and the M.S. degree from the Harbin Institute of Technology, China, in 2017. His current research interests include meta learning, few-shot learning, and machine learning.

Zhihao Chen is currently pursuing the M.Sc degree with the Department of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. He received the B.S. degree from the Harbin Institute of Technology, China, in 2020. His research interests include machine learning, remote sensing and computer vision.

Yingling Quan is currently pursuing the M.Sc degree with the Department of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. She received the B.Sc. degree from University of Science and Technology of China, China in 2021. Her research interests include machine learning, remote sensing and computer vision.