ClouDet: A Dilated Separable CNN-Based Cloud Detection Framework for Remote Sensing Imagery

Hongwei Guo, Hongyang Bai, and Weiwei Qin

***Abstract*—Cloud detection is one of the essential procedures in optical remote sensing image processing because clouds are widely distributed in remote sensing images and cause a lot of challenges, such as climate research and object detection. In this article, a lightweight deep-learning-based framework is proposed to detect cloud in remote sensing imagery. First, a multiple features fusion strategy is designed to extract learnable manual features and convolution features from visible and near-infrared bands. Then, a lightweight fully convolutional neural network (ClouDet) with a microarchitecture named dilated separable convolutional module is used to extract the multiscale contextual information and gradually recovers segmentation results with the same size as input image, which is more effective for large-scale cloud detection with larger receptive field, less parameters, and lower compute complexity. Third, context pooling is designed to amend the possible misjudgments. Visual and quantitative comparison experiments are conducted on several public cloud detection datasets, which indicates that our proposed method can accurately detect clouds under different conditions, which is more effective and accurate than the compared state-of-the-art methods.**

***Index Terms*—Cloud detection, cloud segmentation, convolutional neural network (CNN), deep learning, remote sensing.**

I. Introduction

OPTICAL remote sensing imagery has become one of the most valuable data sources for monitoring changes in the ecological environment, land types, and human's impact on the surface. However, clouds cover more than 50% of the earth's surface, of which about 55% of the land and 72% of the area over the ocean is covered by clouds [1]. On the one hand, cloud coverage is an important factor for climate research, which can make valuable predictions about weather and climate change [2], [3]. On the other hand, for other practical applications, it is hard to extract useful information from the images covered by heavy clouds, which may limit the use of remote sensing images and reduces the utilization of remote sensing data. Moreover, the bright effects of clouds and the darkening effects of cloud shadows affect a variety of data analysis, causing problems in many remote sensing activities. Simultaneously, transferring useless remote sensing images with too much clouds to ground station brings an expensive cost of labor, storage, and computational resources. Crucially, almost all the processing of optical remote sensing imagery requires pixel-scale cloud detection thus cloud detection has become one of the most essential procedures in the analysis and preprocessing of optical remote sensing imagery.

In the past 20 years, a considerable amount of cloud detection methods have been developed, which could be divided into two categories: threshold-based methods [4]–[8] and machine-learning-based methods [9]–[12]. Threshold-based methods in traditional modeling are widely used for cloud detection because of their high accuracy and reliable robustness, which are designed for different sensors to detect clouds in remote sensing imagery by selecting appropriate thresholds of spectral reflectance or brightness temperature via specific channels. In recent years, several threshold-based methods have grown up around the theme of threshold-based methods. For instance, the automatic cloud cover assessment algorithm was designed for the cloud cover assessment of Landsat-7 imagery [4]. Zhu *et al.* [5] introduced the function of mask (Fmask) algorithm, which adopted a decision tree to separate the potential pixels from noncloud region based on multiple threshold functions. An improved version of Fmask was introduced in [6], which made a series of improvements in the Fmask algorithm for Landsat 4–7 and proved that the cirrus band is useful for cloud detection. In considering of that most previous haze/cloud detection methods for Landsat imagery cannot adequately suppress land surface information, Chen *et al.* [7] proposed an iterative haze optimized transformation for improving haze detection, which could effectively remove the land surface information. Different from the method of cloud detection for a single image, Sun *et al.* [8] used multitemporal airborne visible/infrared imaging spectrometer data with 224 bands at visible to SWIR wavelengths for detecting clouds. However, it is hard to determine the suitable thresholds to detect cloud accurately for the complex scenes and multiple types of cloud.

Most machine-leaning-based methods are on the basis of handcrafted features with classifiers such as support vector machine (SVM) [9], K-Nearest Neighbors (KNN) [10], and principal component analysis. These methods usually need handcrafted features such as texture features [11] and morphological features [12] as the input of above classifiers. For example, Li *et al.* [13] adopted an SVM classifier using the brightness and texture features to detect cloud. Yuan and Hu [14] developed a cloud detection method based on object classification

Manuscript received April 9, 2021; revised June 22, 2021, August 5, 2021, and September 5, 2021; accepted September 15, 2021. Date of publication September 21, 2021; date of current version October 8, 2021. This work was supported by the National Natural Science Foundation of China under Grant U2031138. (*Corresponding author: Hongyang Bai.*)

Hongwei Guo and Hongyang Bai are with the School of Energy and Power Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: njustghw@njust.edu.cn; hongyang@njust.edu.cn).

Weiwei Qin is with the Xi'an Research Institute of High-Tech, Xi'an 710025, China (e-mail: qww_1982@163.com).

Digital Object Identifier 10.1109/JSTARS.2021.3114171

using dense SIFT features extracted by the bag-of-words model. However, although the combination of handcrafted features and machine-learning-based classifiers deliver considerable performance, the handcrafted features are usually designed for special scenes and images from special sensors, which make them not robust.

Recently, deep-learning-based methods have been applied to many image processing applications and achieved good performance. Researchers have shown an increased interest in cloud detection using deep-learning. Xie *et al.* [15] proposed a deep convolutional neural network (CNN) with two branches is designed to predict these superpixels as thick cloud, thin cloud, or noncloud. Chen *et al.* [16] developed an end-to-end three-dimensional (3-D)-CNN method for cloud and cloud shadow detection with four band-combination images as the input imagery. Francis *et al.* [17] introduced a fully convolutional network architecture to detect cloud, known as U-net proposed by Ronneberger *et al.* [18], which fuses the shallowest and deepest layers of the network, thus routing low-level visible content to its deepest layers. Shao *et al.* [19] proposed a multiscale features CNN to learn the multiscale global features of input images, which consist of visible, near-infrared, short-wave, cirrus, and thermal infrared bands of Landsat 8 imagery. Mohajerani and Saeedi [20] trained a fully convolutional network with both local and global features from the entire scene for end-to-end pixel-level labeling of the satellite images. And to identify the cloud regions in aerial or satellite images accurately in the presence of snow and haze, an improvement version has been developed with filtered Jaccard loss in [21]. Chen *et al.* [22] applied an adaptive simple linear iterative clustering method to obtain high-quality superpixels and detect clouds by extracting multiscale features from each superpixel. Then, to improve the utilization of high-resolution satellite data, Chen *et al.* [23] presented a CNN architecture for cloud detection, which can use multisource data (content, texture, and spectral) as an input of the unified framework. In fact, most methods simply rely on the convolution kernel to extract the features in the image. To detect cloud mask using thumbnails, Yang *et al.* [24] propose a cloud detection neural network (CDnet) with a feature pyramid module and a boundary refinement block. In view of that the cloud–snow coexistence makes it difficult to detect clouds in remote sensing imagery, Guo *et al.* [25] proposed an improved version of CDnet based on adaptive feature fusing model and high-level semantic information guidance flows, which achieved accurate detection performance on the ZY-3 satellite thumbnail dataset. Recently, Jeppesen *et al.* [26] have been introduced RS-Net to detect clouds in Landsat 8 images, which is trained with both automatically (Fmask) and manually generated ground truth images of two public datasets. The experiments have shown that results obtained by weights trained with Fmask outperform the Fmask direct results. Current experimental results have delivered outstanding performance compared to traditional methods in cloud detection. However, most of the CNN-based cloud detection methods are merely developed on the basis of the semantic segmentation methods, such as FCN [27], SegNet [28], and Deeplab [29], and ignore the characteristics of cloud. For example, clouds tend to distribute in large scale and varies in shape, which require larger receptive field.

Some studies have shown that it can effectively improve the performance of CNN models by fusing CNN and handcrafted features. In the medical field, Li *et al.* [30] proposed a fusion algorithm that combines handcrafted features into the features learned at the output layer of a 3-D deep CNN, including intensity features, geometric features, and texture features. The experiment result has shown that the fusion algorithm takes full advantage of the handcrafted features and the highest level CNN features. Ragab and Attallah [31] proposed a novel CAD system called FUSI-CAD based on the fusion of multiple different CNN architectures with three handcrafted features including statistical features and textural analysis features, which has been proved to be reliable and accurate. For remote sensing image analysis, handcrafted features have always played an important role. Some researchers made attempts to combine deep CNN and handcrafted features. Sun *et al.* [32] integrated multilevel semantic features extracted by bag-of-visual-words model and CNN model to enhance the ability of capturing multiscale land objects. Zhao *et al.* [33] proposed a fractional Gabor convolutional network to extract multiscale, multidirectional, and semantic change features, which yield robust feature extraction against semantic changes. The above research article have shown that the fusion of handcrafted features and CNN features can help improve the performance of the CNN model.

In view of this, we proposed a novel method to identify cloud regions and separate them from noncloud regions in optical remote sensing imagery. Our proposed method is an end-to-end fully CNN, which detect clouds in pixel-scale. This network named ClouDet consists of microarchitecture named dilated separable convolutional module, multiple feature generation layer, and context pooling layer. First, inspired by the progress made in the traditional cloud detection methods, we design a multiple features fusion strategy by choosing several handcrafted features, such as Gabor, mean value, and Laplacian features, to generate new bands and then they are fused with original bands to form the multiple features input of the network. Then, we take each pixel of combination images as the basic research unit and construct ClouDet based on a series of dilated separable convolutional modules. With the equipment of dilated separable convolutional module, ClouDet has larger receptive field, less parameters, lower compute complexity, and could identify clouds accurately. Next, the high-level semantic information in different scales produced by feature learning would integrate with corresponding low-level spatial information in the process of classification. Finally, a context pooling layer is proposed to amend the possible misjudgments according to the context information.

The contributions of this article are as follows.

1) Proposing ClouDet, a light-weighted CNN-based cloud detection framework for remote sensing imagery, which provides a solution for efficient and accurate cloud detection tasks on embedded platforms, such as satellites.
2) We proposed a multiple features fusion strategy, which gives the network richer features that could hardly learn by convolutional operation.

TABLE I
INTRODUCTION OF EACH BAND OF LANDSAT 8

Band Num	Band	Wavelength (um)	Res. (m)
1	Coast	0.435-0.451	30
2	Blue	0.452-0.512	30
3	Green	0.533-0.590	30
4	Red	0.636-0.673	30
5	Near-infrared	0.851-0.879	30
6	Shortwave Infrared 1	1.566-1.651	30
7	Shortwave Infrared 2	2.107-2.294	30
8	Panchromatic	0.503-0.676	15
9	Cirrus	1.363-1.384	30
10	TIRS 1	10.60-11.19	100
11	TIRS 2	11.50-12.51	100

3) We proposed a dilated separable convolutional module to minimize model size and generate features with lager receptive field.
4) We designed a context pooling strategy to amend the possible misjudgments according to the context information.

The rest of this article is organized as follows. Section II introduces the proposed ClouDet framework. The experimental results and discussion are presented in Section III. Finally, Section IV concludes this article.

II. PROPOSED METHOD

In this section, the proposed methodology for addressing the problem of cloud detection is described. First, a brief explanation of the data provided by Landsat 8 is given. Next, the proposed cloud detection framework, multiple features fusion strategy, dilated separable convolutional module, and context pooling strategy are described, respectively.

A. Landsat 8

The Landsat 8 program is the eighth program of land satellites in the National Aeronautics and Space Administration (NASA). The program is jointly operated by NASA and the U.S. Geological Survey, and its goal is to make long-term observations of the ground. The Landsat 8 satellite provides rich and reliable data for people to better understand the earth and use resources. Landsat 8 carries two main payloads: OLI (operational land imager) and TIRS (thermal infrared sensor). OLI includes eight bands with a spatial resolution of 30 m and a panchromatic band with a spatial resolution of 15 m. OLI includes all the bands of the ETM + sensor. At the same time, in order to prevent atmospheric absorption characteristics, OLI readjusted the fifth and eighth bands. The fifth band excludes the water vapor absorption characteristics at 0.825 μm, and the eighth band of the panchromatic band has a narrower range, which can better distinguish the vegetation information. In addition, two new bands have also been added: the coast band is mainly used for coastal zone observations, and the cirrus band includes strong absorption characteristics of water vapor. Table I presents the relevant information of each band of Landsat 8. However, many platforms on satellites are only equipped with visible and near-infrared bands, such as ZY-3 [34] and GF-2 [35], which provides limited feature information to detect clouds in remote sensing imagery and make the task more difficult. In order to cover the cloud detection tasks on most of optical satellite platforms, only four spectral bands—Band 2 to Band 5 are used in this article.

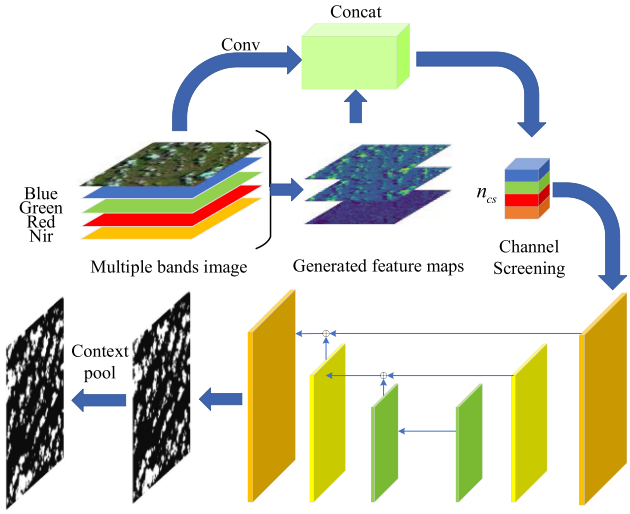


Fig. 1. Workflow of ClouDet in detail. Concat and Context pool mean concatenation and context pooling layers, respectively.

B. ClouDet

Previous state-of-the-art deep CNN have achieved significant performance in a wide range of computer vision tasks, such as image classification, saliency detection, object recognition, and semantic segmentation. Most of the CNN-based methods are developed on the basis of the semantic segmentation methods, such as FCN, which can conduct intensive prediction without fully connected layers. This structure can adapt to the image of any size. However, thin cloud and highlight objects on the ground tend to have similar characteristics, which make it a challenging task to separate them from each other merely with convolutional filters. Inspired by the effectiveness of handcrafted-based methods and the principle of FCN, we proposed ClouDet to extract the multiscale global features to identify clouds in optical remote sensing images.

Fig. 1 illustrates the overall framework of our proposed ClouDet method. We choose RGB and near-infrared bands to form the initial input remote sensing image. In view of the efficiency of handcrafted features, two kinds of edge features (Gabor and Laplacian) and mean value are chosen as the supplementary features, which are introduced in Section C. We designed three feature generated layers to generate the feature maps with filters of 3×3 pixels. The three kinds of feature maps are concatenated with the convolutional feature maps to form the final input image of the segmentation model. And a subnetwork n_{cs} based on SE block [36] is designed to link the final input image and the segmentation model, which could determine the importance of image features in different bands through gradient backpropagation and provide reference for the choice of handcrafted features in this framework. The segmentation model is based on FCN and similar to the UNet, which consists of two

arms, the encoder and the decoder. The convolutional structure of these arms is described in Section II-F. The encoder is designed to extract features from the scene, and expand the receptive field by changing the size of the feature maps at the same time. Meanwhile, the decoder takes these features and reprojects them to create the output mask. Residual connections are used to fuse low-level spatial information and high-level semantic information, which could provide more detailed information of clouds so as to realize the cloud detection in pixel level more accurately. The final layer of decoder adopts a SoftMax activation to identify each pixel as ground or cloud. At last, in view of that clouds are usually continuous in remote sensing images, a context-based cloud detection result correction strategy is proposed to amend the possible misjudgments and then the final output cloud mask could be created.

C. Multiple Features Fusion Strategy

The origin input image of the framework is the combination of bands 2–5 of Landsat 8. According to the observations of optical remote sensing images, clouds and high albedo terrains (bright urban areas, for example) tend to have similar features, which makes it a challenging task to identify the classification of each pixel. With the development of cloud detection technology in remote sensing, several handcrafted features are designed to identify cloud with machine-learning-based classifier, such as SVM and KNN. The main process of the above method is to identify each image block by the generated handcrafted features, which is similar to the process of CNN. Inspired by excellent performance of handcrafted features in cloud detection, we choose three handcrafted features (Gabor, Laplacian, and mean value) generated from the origin input image to create new feature maps, which are then concatenated with other convolutional feature maps to form the final input image of the segmentation model. The handcrafted features adopted in this article are described as follows.

1) Gabor: Gabor wavelet [37] was invented by Dennis Gabor, which uses complex function as the basis of Fourier transform in information theory applications. Gabor wavelet is very similar to the visual stimulus response of simple cells in the human visual system. It has good characteristics in extracting the local space and frequency-domain information of the target. Gabor wavelet is sensitive to the edge of the image, and can provide good direction selection and scale selection characteristics, in addition, it is not sensitive to changes in illumination, and can provide good adaptability to changes of illumination. The above characteristics make Gabor wavelet widely used in visual information understanding. The Gabor wavelet is defined as follows:

$$\begin{cases} g\left(x,y;\lambda,\theta,\psi,\sigma,\gamma\right)=\exp\left(-\frac{x'^2+\gamma^2y'^2}{2\sigma^2}\right)\cos\left(2\pi\frac{x'}{\lambda}+\psi\right) \\ x'=x\cos\theta+y\sin\theta \\ y'=-x\sin\theta+y\cos\theta \end{cases} \tag{1}$$

where λ represents the wavelength parameter of the cosine function in the Gabor kernel function, θ is the direction of parallel bands in the Gabor filter kernel, and ψ is the phase parameter of the cosine function in the Gabor kernel function, which is between -180° and 180°. γ is the spatial aspect ratio, which determines the ellipticity of the shape of the Gabor kernel function. σ represents the standard deviation of the Gaussian factor of the Gabor function. In this article, we develop an approach by utilizing Gabor filters to modulate the learned convolution filters. With the help of gradient backpropagation, the parameters of the Gabor function are learnable. Gabor convolutional layer has four parameters: output channel, input channel, height of convolution kernel, and width of convolution kernel, which can be embedded in CNNs like standard convolution layer.

2) Mean Value: The mean value refers to the arithmetic mean of the pixel values of all the pixels in an image block, which is calculated as the following equation:

$$\bar{f}=\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{f(i,j)}{M\cdot N} \tag{2}$$

where $\bar{f}$ is the mean value of the image block, *M* and *N* are the width and height of the image block, respectively, and f is the pixel value of each point.

3) Laplacian: The high-frequency component of the image corresponds to detailed information such as edge texture, and the low-frequency component of the image corresponds to the background. The amount of information in the image is mainly reflected in the edge texture. We define the edge as the boundary of the region where the gray scale changes sharply in the image. The change of the image gray scale can be reflected by the gradient of the image gray distribution. Laplacian operator is widely used in edge detection because of its high efficiency. In digital images, the Laplacian operator is defined as

$$\begin{aligned} &\nabla^2 f\left(x,y\right) \\ &\quad=f\left(x-1,y-1\right)+f\left(x-1,y\right)+f\left(x-1,y+1\right) \\ &\qquad+f\left(x,y+1\right)-8f\left(x,y\right)+f(x,y-1) \\ &\qquad+f\left(x+1,y+1\right)+f\left(x+1,y\right)+f\left(x+1,y-1\right) \end{aligned} \tag{3}$$

where $\nabla^2 f(x,y)$ is the generated edge texture image and $f(x,y)$ is the origin image.

The three kinds of handcrafted feature maps are shown in Fig. 2, which have distinct spectral characteristics and can be easily separated from the background.

D. Channel Screening

The multiple features fusion strategy designed in Section C concatenates three kinds of handcrafted features with convolutional features in channelwise. However, the channel relationships modeled by directly concatenates are inherently implicit and local. We expect the learning of multiple features to be enhanced by explicitly modeling channel interdependencies, so that the network is able to increase its sensitivity to informative features which can be exploited by subsequent transformations.

In order to address the issue of uncorrelated information between channels, we first consider the connection between the channels in feature maps. Each of the handcrafted feature

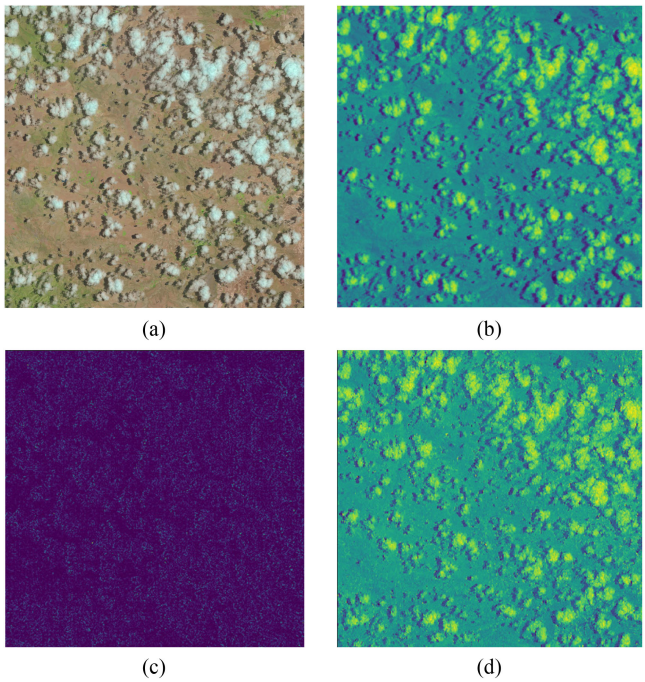


Fig. 2. Three kinds of handcrafted feature maps. (a) Origin image. (b) Mean value map. (c) Laplacian feature map. (d) Gabor feature map.

operators and convolutional filters deals with a local receptive field, which consequently make each unit of the transformation output unable to exploit contextual information outside of this region. The current attention mechanism mainly obtains channel attention through global average pooling, which is realized based on the global features obtained by global average pooling, such as the SE block proposed in [38]. Global average pooling pays more attention to the overall information, while global max pooling is easily affected by extreme values. So, global average pooling is commonly used to extract global features. However, the characteristic of global maximum pooling does not certainly cause negative effects in cloud detection tasks. When the object occupies a larger proportion in the feature map, global average pooling can better capture the features of object. For scenes with few clouds or thin clouds, global average pooling may pay more attention to irrelevant features that account for a larger proportion, while global max pooling may make it easier to capture object features. To build the connection between the channels in feature maps, we first intend to quantify the global spatial information of each channel through global average pooling and global max pooling. The global average pooling is defined as

$$F=\frac{\sum_{x=1}^{M}\sum_{y=1}^{N} f(x,y)}{M\cdot N} \tag{4}$$

where F is the mean value of the image block, M and N are the width and height of the image block, respectively, and f is the pixel value of each point. The global max pooling is defined as

$$F=\max(f(x,y))x=\{1,2,\ldots,M\},y=\{1,2,\ldots,N\} \tag{5}$$

where F is mean value of the image block, M and N are width and height of the image block, respectively, and f is the pixel value of each point.

Then, to fully capture channelwise dependencies, we design a bottleneck structure based on SE block. Specifically, we follow global average pooling and global max pooling with a convolutional layer of $1\times1\times\frac{C}{r}$, convolutional layer of $1\times1\times C$, and a ReLU, which could effectively capture the nonlinear interaction between channels. Then, we concatenate the feature maps from the two branches. Finally, a sigmoid activation is adopted to output the rescaling factor of the feature maps. With the redefinition of the importance of the feature map by the rescaling factor, the valuable information in the feature map can be effectively highlighted. A diagram illustrating the structure of channel screening module is shown in Fig. 3.

E. Dilated Separable Convolution

Clouds are usually continuous and occupy a large proportion in remote sensing images, which brings an urgent need to develop an accurate and robust method with larger receptive field. Inspired by large receptive field of the dilated convolution in [39] and high model compression ratio of the depthwise convolution in MobileNet V2 [40], we propose an efficient microarchitecture named dilated separable convolution based on atrous separable convolution proposed in deeplabV3+ [41], as shown in Fig. 4(d). Unlike the atrous separation convolution, in order to prevent the negative effects of continuous hole separation convolution, such as grid effect and irrelevant long-distance features, we added another branch with standard convolution. The channel dimension in dilated separable convolutional module is expanded first and then reduced on the purpose of achieving higher memory efficiency. We define the dilated separable convolutional module as follows. A dilated separable convolutional module is composed of an expand convolution layer with only 1×1 filters, a dilated separable convolution layer with kernels of different dilated rate, a standard convolution layer, and a squeeze convolution layer with only 1×1 filters. The standard convolution layer, atrous separable convolution layer, and squeeze convolution layer have the same number of kernels, which is half the number of kernels in the expand convolution layer. The dilated separable convolutional kernels with different dilated rate are shown in Fig. 4(a)–(c) respectively.

F. Segmentation Model

The network architecture is illustrated in Table II, which is divided into several stages according to the size of output feature maps. The input sizes and output sizes are set to 384×384 pixels. The segmentation model consists of feature generated module, downsampling module, upsampling module, dilated separable convolutional module, and a series of bottlenecks from ResNets [42], which has a single main branch and extensions with convolutional filters that separate from it, and then merge back with an elementwise addition. The feature generated module is designed as the initial block of ClouDet, which consists of convolutional layers, three kinds of hand-crafted feature layers. We place batch normalization [43] and ReLU [44] between all convolutions. The

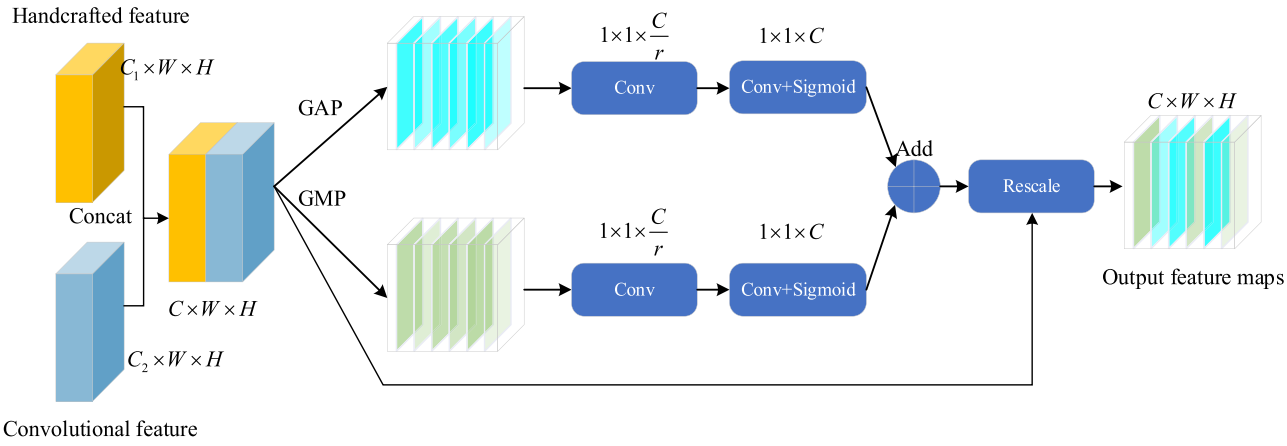


Fig. 3. Workflow of channel screening module. GAP, GMP, Conv, and Concat mean global average pooling, global max pooling, convolution, and concatenation, respectively.

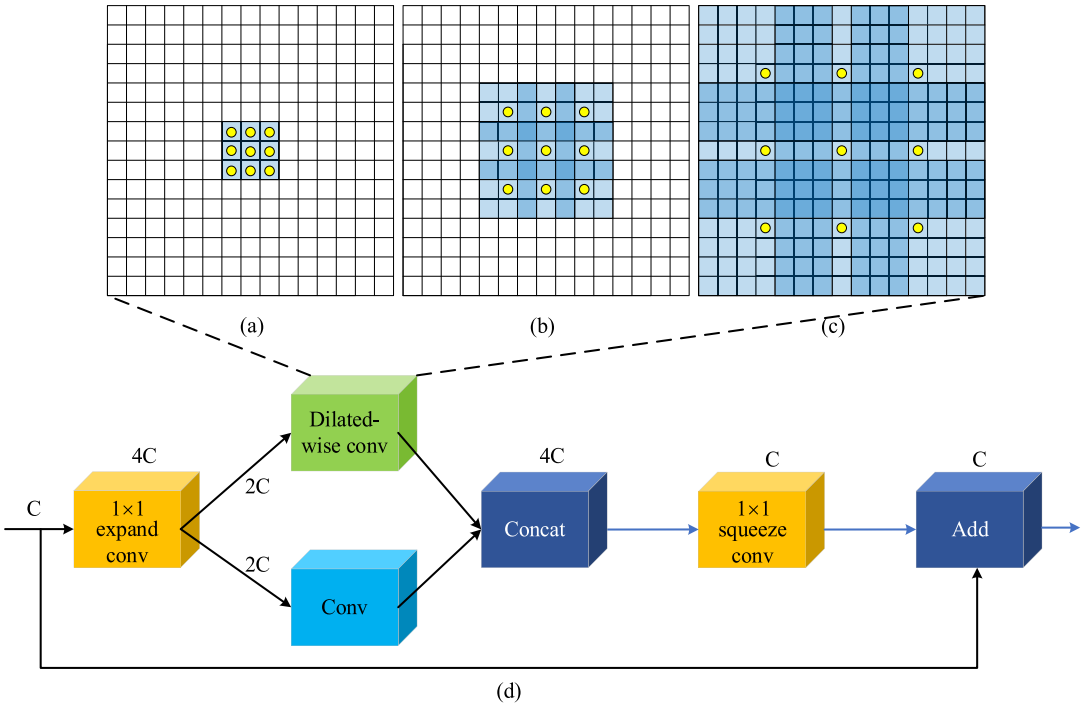


Fig. 4. Dilated separable convolutional module designed in this article with different dilated rate. (a) Dilated kernel of rate = 1. (b) Dilated kernel of rate = 2. (c) Dilated kernel of rate = 4. (d) Dilated separable convolutional module. C is the number of the feature maps.

downsampling module is similar to the bottleneck in ResNet with a 3×3 convolutional layer with stride 2. And on the contrary, the upsampling module is with a deconvolutional layer of stride 2.

The feature generated module contains a single block, which is introduced above. Stage 1 consists of a downsampling module and four bottleneck blocks. Stage 2 and stage 3 have almost the same structure with nine convolutional modules, with the exception that the first module of stage 2 is set to downsampling module. Considering that we used dilated convolution in the model, in order to eliminate the effect of the grid effect, we set the dilated rate to 1, 2, 5, and 9 in order. Stages 1–3 form the encoder and stages 4–5 are the decoder. As the last part of ClouDet, the full convolutional layer has C feature maps, which have the same size of input image.

G. Context Pooling

Since clouds and high albedo terrains (bright urban areas, for example) tend to have similar features, which increases the possibility of false positive, while clouds are generally distributed continuously in remote sensing images, rather than standalone deemed as discrete points, we believe that the category information of surrounding pixels has a certain reference value to the current pixel classification results. To address the problem, we designed a context pooling strategy to amend the possible

TABLE II
ARCHITECTURE OF CLOUDET. OUTPUT SIZES ARE GIVEN FOR AN EXAMPLE INPUT OF 384×384 PIXELS

Name	Type	Output size
Input		4×384×384
Feature generated		32×192×192
Downsampling1	downsampling	16×96×96
4×Bottleneck		16×96×96
Downsampling2	downsampling	16×48×48
DS conv1	Dilated rate = 1	16×48×48
DS conv2	Dilated rate = 2	16×48×48
DS conv3	Dilated rate = 5	16×48×48
DS conv4	Dilated rate = 9	16×48×48
DS conv5	Dilated rate = 1	16×48×48
DS conv6	Dilated rate = 2	16×48×48
DS conv7	Dilated rate = 5	16×48×48
DS conv8	Dilated rate = 9	16×48×48
Repeat stage 2, without downsampling module		
Upsampling1	upsampling	64×96×96
2×Bottleneck		64×96×96
Upsampling2	upsampling	64×192×192
Bottleneck		64×192×192
fullconv		C×384×384

DS conv = Dilated separable convolution.

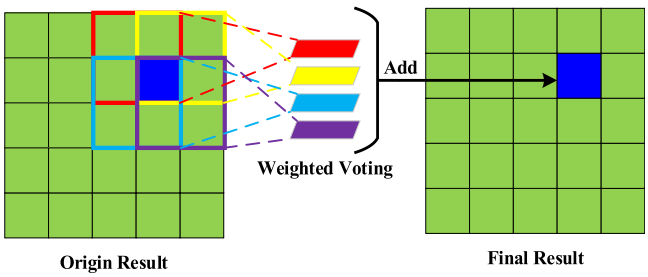


Fig. 5. Diagram of context-based pooling.

misjudgments without building a mathematical model or modify existing framework. In view of that clouds are continues and rarely exist alone in a single pixel, we use the classification results of neighboring pixels to correct the classification results of the current pixel. As shown in Fig. 5, each pixel is surrounded by eight pixels and they could form four rectangular regions of 2×2 pixels under the condition that the pixel is the intersection. The original result is a binary image, where the pixel value marked as cloud is 1, otherwise it is 0. The pixels in every 2×2 rectangular region determine the type of the region as cloud or ground by weighted voting. Then, the generated results are adopted to predict the final result by weighted voting again. The kernel size of context pooling is set to 3 and stride is set to 1. And the context pooling filter could be defined as

$$\begin{aligned}\nabla^2 f(x,y) &= f(x-1,y-1) + 2f(x-1,y) \\ &\quad + f(x-1,y+1) \\ &\quad + 2f(x,y+1) + 4f(x,y) + 2f(x,y-1) \\ &\quad + f(x+1,y+1) + 2f(x+1,y) \\ &\quad + f(x+1,y-1) \end{aligned} \tag{6}$$

where $\nabla^2 f(x,y)$ is the generated edge texture image and $f(x,y)$ is the origin image.

III. EXPERIMENTAL RESULTS

In this section, we briefly introduce the datasets used in this article, and then provide evaluation metrics. At last, the visual and numerical results over these datasets are reported and discussed.

A. Datasets

1) 38-Cloud Dataset: 38-Cloud dataset [20] consists of 38 Landsat 8 scenes with 4 bands (blue, green, red, and near-infrared) mainly selected from North America, which is divided into training set with 18 scenes and testing set with 20 scenes. The ground truths of these scenes are annotated manually in pixel-level. All the 38 scenes are cropped into 384×384 pixels. There are 8400 patches in the training set and 9201 patches in the testing set.

2) 95-Cloud Dataset: 95-Cloud dataset [21] is an improvement and supplement to 38-Cloud Dataset. A total of 57 new Landsat 8 scenes are selected and added to the training set of 38-Cloud Dataset. In order to simulate all the situations as much as possible, images in 95-Cloud are selected to cover many land cover types such as soil, vegetation, urban areas, snow, ice, water, haze, and different types of cloud patterns. And the cloud coverage in 95-Cloud Dataset is kept at about 50%. Same as 38-Cloud Dataset, images in 95-Cloud Dataset are divided into pieces of 384×384 pixels. The training set of 95-Cloud Dataset consists of 34 701 images and the testing set consists of 9201 images.

3) SPARCS Dataset: The SPARCS dataset [45] consists of 80 patches extracted from the Landsat 8 scenes, and the size of each patch is 1000×1000. The scenes in the dataset are manually annotated. Each pixel is classified as one of "cloud," "shadow," "snow/ice," "water," "land," and "flood." We combine all classes except "cloud" under the "clear" category to generate a binary mask for each patch.

B. Evaluation Criteria

We adopted several widely used measures to quantitatively evaluate the performance of our cloud detection method, including precision, recall, Jaccard index [20], specificity, and overall accuracy

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

$$\text{Jaccard Index} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{9}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{10}$$

$$\text{Overall Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{11}$$

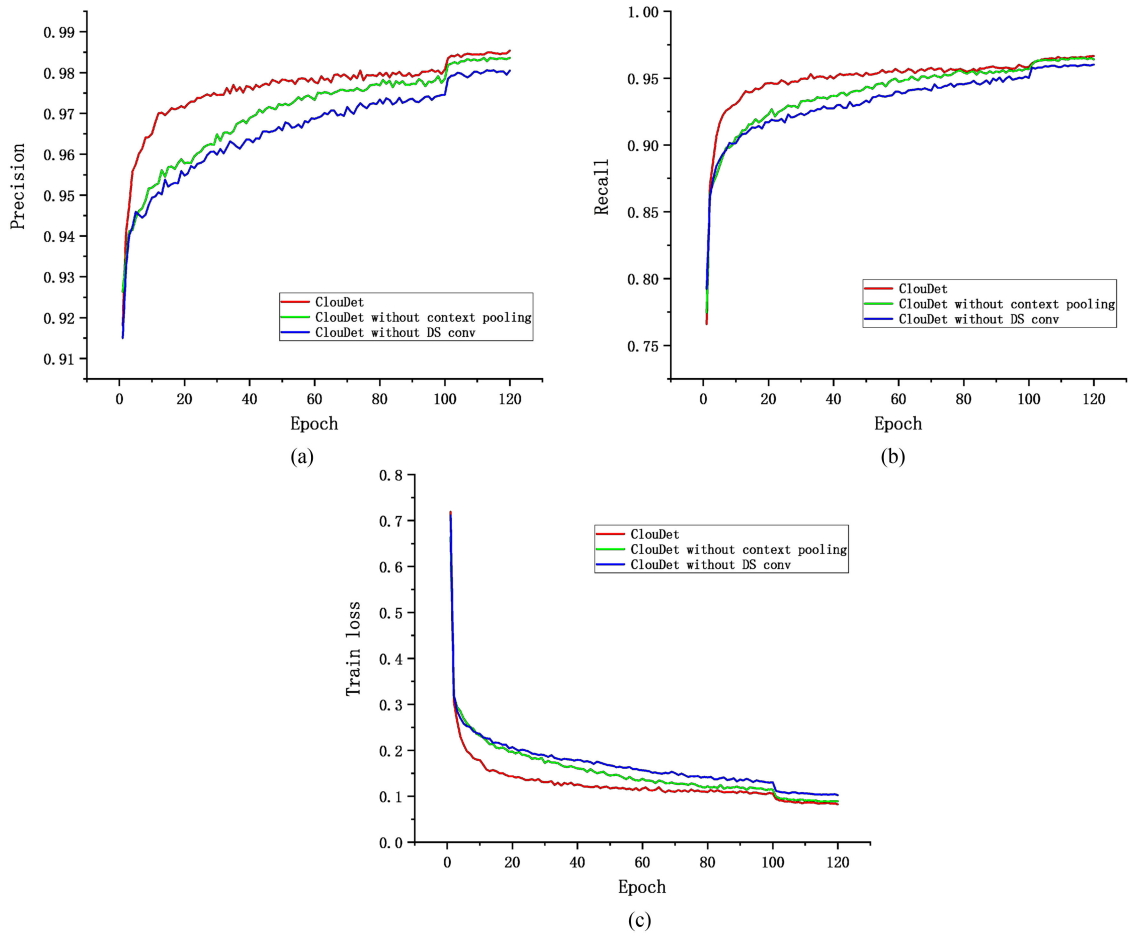


Fig. 6. Evolution of the loss function, precision, and recall on 38-Cloud Dataset and 95-Cloud Dataset. (a) Evolution of the precision rate. (b) Evolution of the recall rate. (c) Evolution of the loss function.

where TP , TN, FP, and FN are the numbers of true positive, true negative, false positive, and false negative pixels in each test set scene, respectively.

C. Baseline Methods

We compared the proposed algorithm ClouDet, with the state-of-the-art methods, including FCN [27], deeplabv3+ [41], Cloud-Net+ [21], BiSeNetV1[46], and Fmask [6]. We tested ClouDet and other methods mentioned above on the test dataset for comparison.

D. Implementation Details

We implemented our network on the open source Pytorch [47] framework and executed on a 64-bit Ubuntu 16.04 computer with 11 GB memory GeForce GTX1080Ti GPU and Intel Core i7-7700K CPU. The comparison models were implemented in their original environments without any additions.

The proposed ClouDet was trained on 38-Cloud Dataset, 95-Cloud Dataset, and SPARCS dataset successively, with an initial learning rate of 0.0005. The network was trained by Adam optimization algorithm [48] with a batch of eight images, and the weight decay and momentum are 0.1 and 0.9, respectively.

In order to validate the effectiveness of the network, we trained and tested the performance of ClouDet under different network configurations on training and validation sets. We compared the performances of ClouDet and models without context pooling and without dilated separable convolution to demonstrate the benefits of the addition of the dilated separable convolutional module and context pooling strategy in the encoder–decoder module. ClouDet without context pooling denotes the network without the context pooling layer, where we generate the output from the last convolutional layer in the decoder module. In the network of ClouDet without dilated convolution, we replace the dilatedwise convolution with convolution from each dilated separable convolutional module shown in Fig. 4. Except for the changes in the model structure, all the other parameter settings of ClouDet and the compared models were the same in the training and testing stages. The experiment al results are shown in Figs. 6 and 7 from which we can see that both the dilated separable convolutional module and context pooling strategy in ClouDet can help to improve the cloud detection performance.

E. Experimental Result

To determine the final neural network structure, we regard stage 2 in Table II as a superimposable component called block.

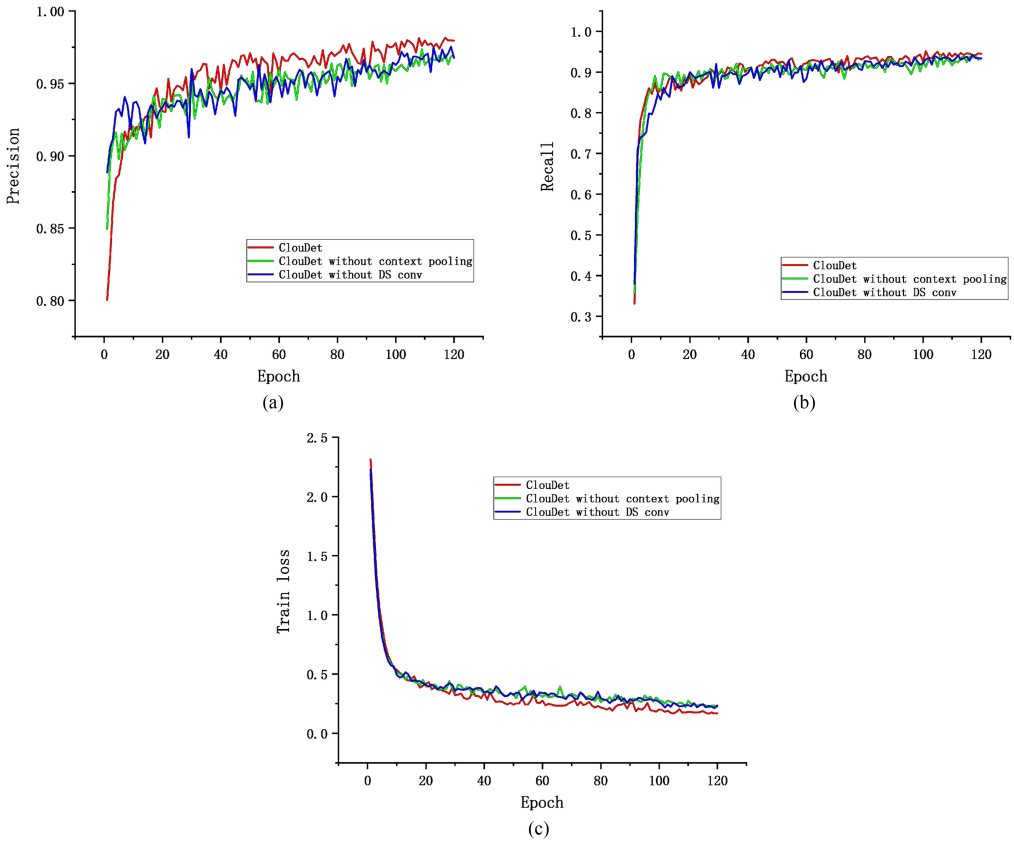


Fig. 7. Evolution of the loss function, precision, and recall on SPARCS dataset. (a) Evolution of the precision rate. (b) Evolution of the recall rate. (c) Evolution of the loss function.

TABLE III
PERFORMANCE COMPARISON OF MODELS WITH DIFFERENT DEPTHS

Model	GFLOPs	Parameters (M)	Model size (MB)	Performance over 38-cloud and 95-cloud dataset (in %)		
				Jaccard index	Precision	Recall
Cloudet-block1	1.42	0.22	2.9	88.85	94.51	96.68
Cloudet-block2	1.58	0.30	3.9	90.93	97.08	93.49
Cloudet-block3	1.63	0.39	4.8	90.82	97.47	93.01

By superimposing different numbers of blocks, we designed a series of neural network models with different depths. The performance of the above model is evaluated on 38-Cloud dataset and 95-Cloud dataset, and the results are given in Table III. Obviously, it can be seen from Table III that neural networks are difficult to achieve optimal performance with too shallow or deep structure. Simultaneously, a deeper network structure would generate more parameters and a larger amount of calculation, which limits the efficient cloud detection tasks on the embedded platform. Therefore, we adopt cloudet-block2 as the final neural network structure named ClouDet, as given in Table II.

Then, we test the performance of ClouDet on 38-Cloud dataset and 95-Cloud dataset to demonstrate efficient and accurate performance for practical applications. All our models, training, testing, and performance evaluation scripts were using the Pytorch framework, with cuDNN backend. To compare the results, we use precision, recall, Jaccard index, specificity, and overall accuracy metrics.

The results, as given in Table IV, indicate that the efficiency of ClouDet is evident with a comparison of number of floating-point operations and parameters used by different models. What stands out in the table is that ClouDet has extremely few parameters, that the required space is only 3.9 MB, which is 11.7, 18.1, 83.4, and 111.3 times smaller than BiSeNetV1, FCN-8s, deeplabv3+, and Cloud-Net+, respectively. The extremely few parameters make it possible to fit the whole network in an extremely fast on-chip memory in embedded development board. And the number of floating-point operations for ClouDet is 1.58G, which is 3.8, 29.6, 23.7, and 10.8 times smaller than BiSeNetV1, FCN-8, deeplabv3+, and Cloud-Net+, respectively. The extremely small number of floating-point operations enables ClouDet to process data faster in embedded development board. According to the inference time test in Table VI, the inference efficiency of ClouDet is better than most comparison methods.

TABLE IV
RESOURCE USAGE OF CLOUDET. GFLOPS ARE ESTIMATED FOR AN INPUT OF $3 \times 384 \times 384$

Model	Backbone	GFLOPs	Parameters	Model size	FPS on GTX 1080Ti
FCN-8s [27]	VGG16	48.35	18.64 M	74.6 MB	76.3
Deeplabv3+ [37]	ResNet50	39.02	40.87 M	329.2 MB	7.0
BiSeNetV1[46]	ResNet18	7.62	12.71 M	49.7 MB	220
Cloud-Net+ [21]	No	18.7	34.74 M	438.1 MB	0.37
ClouDet	No	1.58	0.3 M	3.9 MB	97.1

TABLE V
NUMERICAL PERFORMANCE OVER 38-CLOUD AND 95-CLOUD DATASET (IN %)

Model	Jaccard Index	Precision	Recall	Specificity	Overall Accuracy
FCN-8s	85.03	96.15	88.02	98.34	95.05
Fmask	85.91	88.65	96.52	94.20	94.94
Deeplabv3+	87.11	92.62	93.61	96.92	94.20
BiSeNetV1	85.60	89.83	95.10	96.79	95.70
Cloud-Net+	88.90	97.33	91.12	98.83	96.36
ClouDet	90.93	97.08	93.49	98.59	96.91

Table V provides the numerical results of the proposed ClouDet trained on 38-Cloud and 95-Cloud training set and evaluated on the testing set. We also compare ClouDet with other state-of-the-art methods on this dataset. The numerical results present that the proposed ClouDet captures more valuable features out of the input images and, therefore, the generated cloud masks from ClouDet are more similar to the manually extracted ground truths in different scenes. As Table V indicates, the precision of ClouDet is better than that of FCN-8s, deeplabV3+, BiSeNetV1, and Fmask by 0.93%, 4.46%, 7.25%, and 8.43%, respectively. And the recall rate of ClouDet is better than that of FCN-8s and Cloud-Net+ by 5.47% and 2.37%, respectively, which is slightly behind Fmask, BiSeNetV1, and deeplabV3+. The Jaccard index represents the degree of overlap between the generated cloud mask and the ground truth. The Jaccard index of ClouDet is 5.9%, 5.02%, 3.82%, 5.33%, and 2.03% higher than FCN-8s, Fmask, deeplabV3+, BiSeNetV1, and Cloud-Net+, respectively, which demonstrates that the generated cloud masks of ClouDet are more similar to the ground truth. Additionally, the overall accuracy obtained by ClouDet is higher than all comparison methods. Overall, these results indicate that the combination of the proposed ClouDet with feature generated strategy and context pooling strategy delivers superior performance than other methods.

The visualization results on 38-Cloud-Dataset and 95-Cloud-Dataset are shown in Fig. 8, respectively. Each pixel is classified as one of "true positive," "true negative," "false positive," and "false negative," which are represented by white, black, yellow, and red, respectively. The images used for the test have different proportions of cloud cover, and contain multiple types of land cover, such as land, ice, and snow. On the basis of dilated separable convolutional module and context pooling strategy, ClouDet effectively suppress the grid effect and single pixel misrecognition, but also lose some detailed information at the same time. In general, compared with other models, the proposed ClouDet can effectively detect clouds in different scenes.

In order to verify the influence of different handcrafted features adopted in the feature generated module on the accuracy of the model, we conducted a comparative analysis on the Jaccard index, precision, and recall of the model under different conditions, as given in Table VI. It can be seen from the data in Table VI that the Gabor feature and Laplacian features contribute the most to the Jaccard index and recall rate. The combination of Gabor features and Laplacian features improves the model's predictive ability at the boundary of clouds and land. And the addition of the mean value feature enables our model to judge pixel categories more accurate. In general, the three features have significantly improved the performance of the model in cloud detection tasks.

In addition, in order to further test the adaptability of the algorithm and verify whether the method is suitable for cloud detection in different climate environments, we evaluated the performance of ClouDet on the SPARCS dataset and compared it with other deep-learning-based benchmark methods. Table VII provides the numerical results on SPARCS dataset. As Table VII indicates, compared to other benchmark methods, ClouDet could achieve significantly better performance. The pixel-level visual results are shown in Fig. 9. Each pixel is classified as one of "true positive," "true negative," "false positive," and "false negative," which are represented by white, black, yellow, and red, respectively. From the pixel-level visual results in Fig. 9, it is apparent that ClouDet could achieve superior performance and lower false alarm rate under different terrain and climate environments. In addition, ClouDet could effectively detect small objects, and separate clouds and shadow regions.

IV. CONCLUSION

In this article, a lightweight deep-learning-based framework named ClouDet is proposed for efficient cloud detection in remote sensing imagery. Our main goal is to effectively make use of the scarce computing and storage resources on the embedded platform and accurately identify the cloud in remote sensing imagery. The proposed ClouDet benefits from the proposed dilated separable convolution module, which has extremely few parameters, low computational complexity, and small model size. Another more significant findings to emerge from this article is that ClouDet could achieve lager receptive

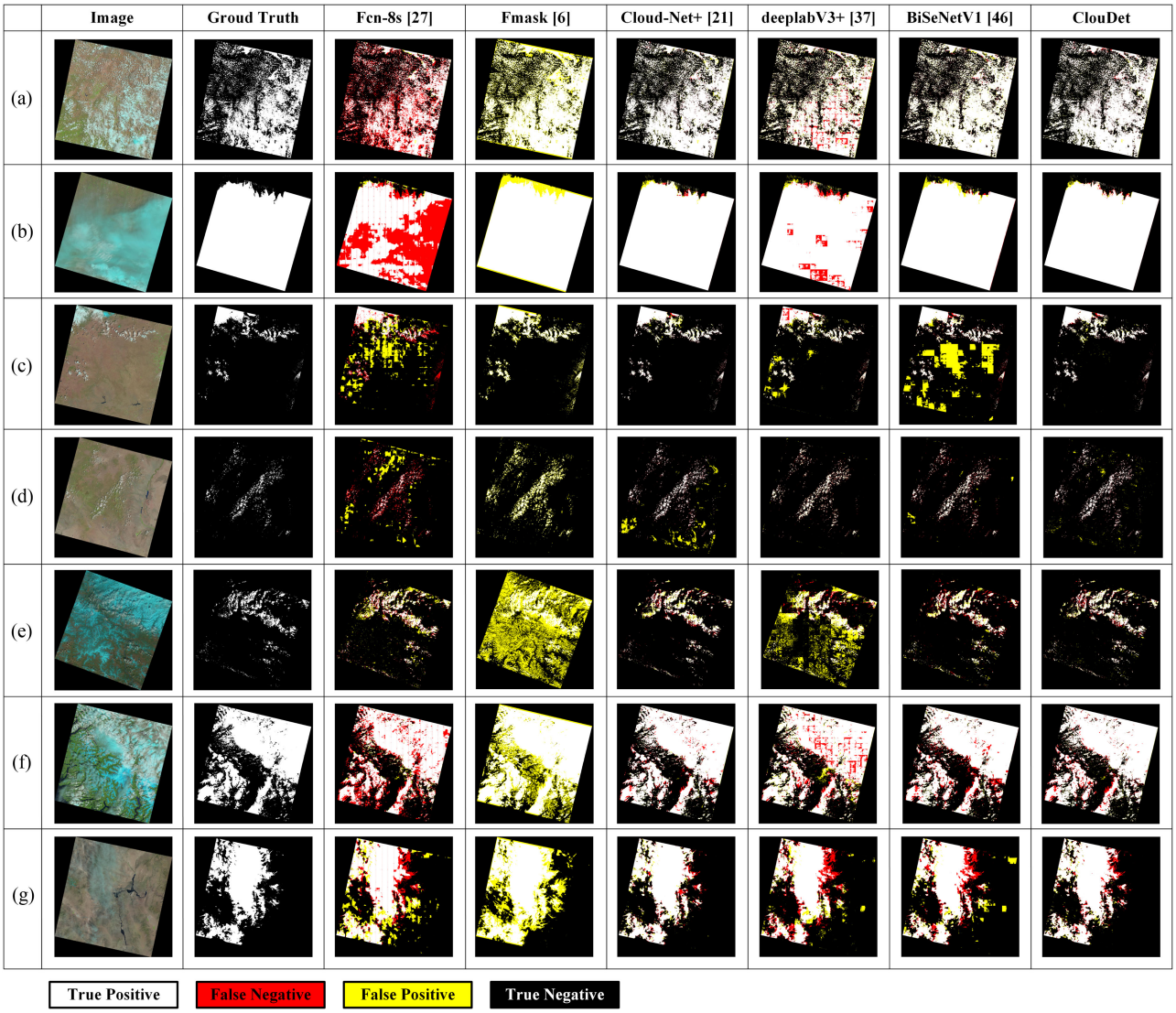


Fig. 8. Some visual examples of the results over 38-Cloud and 95-Cloud datasets.

TABLE VI
COMPARATIVE ANALYSIS OF DIFFERENT HANDCRAFTED FEATURES OF THE MODEL OVER 38-CLOUD AND 95-CLOUD DATASETS (IN %)

Model	Gabor	Laplacian	Mean Value	Jaccard Index	Precision	Recall
ClouDet	×	×	×	88.76	95.50	92.63
	✓	×	×	89.83	96.63	92.73
	×	✓	×	89.68	96.78	92.44
	×	×	✓	89.66	97.18	92.05
	✓	✓	×	89.85	96.88	92.53
	✓	×	✓	90.67	96.98	93.30
	×	✓	✓	90.60	96.59	93.59
	✓	✓	✓	90.93	97.08	93.49

TABLE VII
NUMERICAL PERFORMANCE OVER SPARCS DATASET (IN %)

Model	Jaccard Index	Precision	Recall	Specificity	Overall Accuracy
FCN-8s	77.30	89.33	85.16	97.57	95.17
Deeplabv3+	82.94	90.84	90.55	97.79	96.38
BiSeNetV1	82.67	92.44	88.66	97.72	96.23
Cloud-Net+	84.66	92.90	90.51	97.41	95.53
ClouDet	85.46	93.65	90.71	98.45	96.89

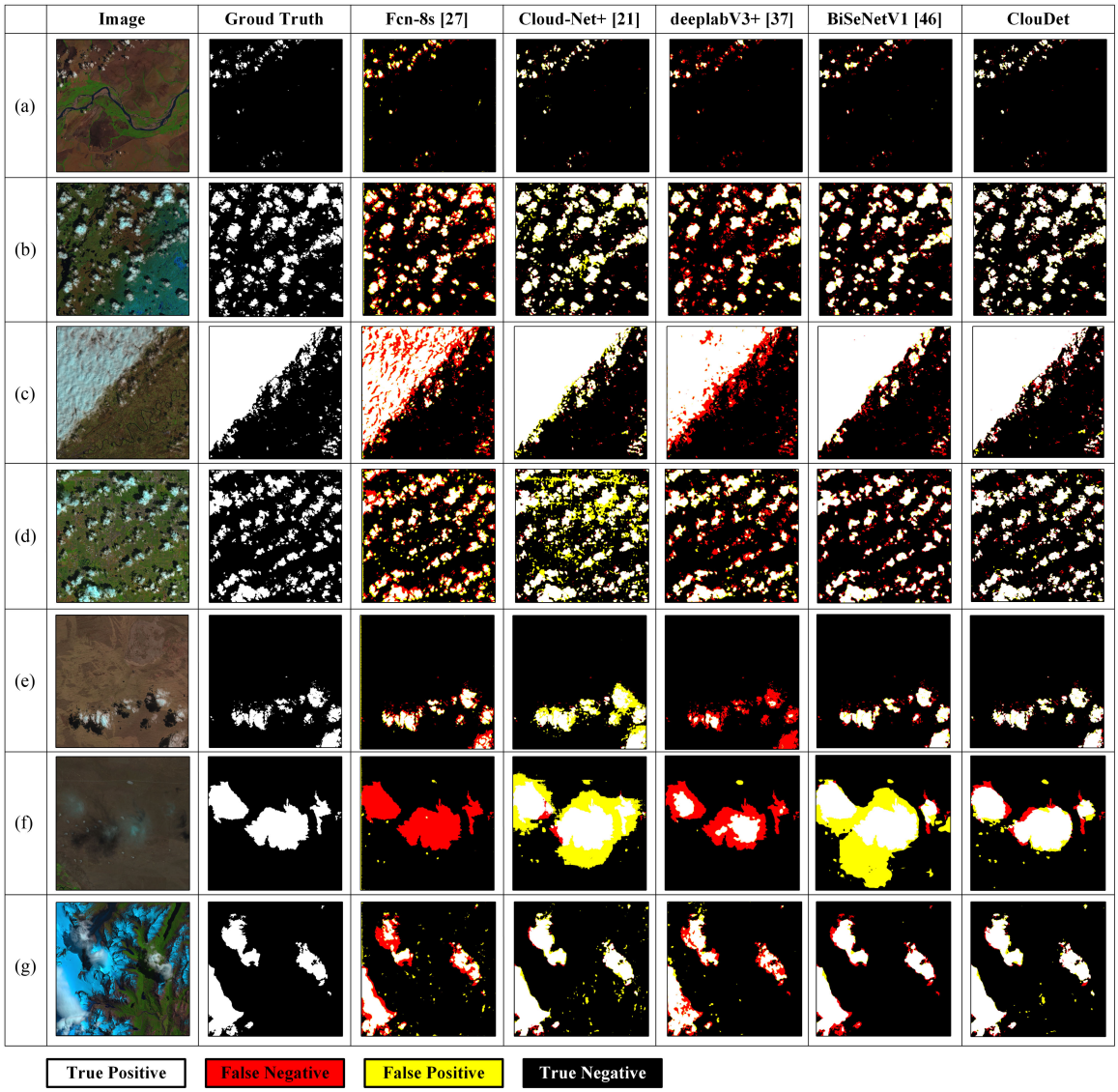


Fig. 9. Some visual examples of the results over SPARCS datasets.

field, significant higher precision, and Jaccard index on the basis of dilated convolution, multiple features generated strategy, and context pooling strategy. These results demonstrate that the proposed ClouDet could be well suited for accurate and efficient cloud detection on embedded platforms. Our further research might focus on determine the effectiveness of other handcrafted features and examine more closely the possibility of combining handcrafted features with CNNs for cloud detection.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for the constructive suggestions.

REFERENCES

[1] M. D. King, S. Platnick, W. P. Menzel, S. A. Ackerman, and P. A. Hubanks, "Spatial and temporal distribution of clouds observed by MODIS onboard the terra and aqua satellites," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 7, pp. 3826–3852, Jul. 2013.

[2] R. Stöckli *et al.*, "Cloud detection with historical geostationary satellite sensors for climate applications," *Remote Sens.*, vol. 11, no. 9, pp. 1052–1063, May 2019.

[3] X. Liu *et al.*, "Spectrally dependent CLARREO infrared spectrometer calibration requirement for climate change detection," *J. Climate*, vol. 30, no. 11, pp. 3979–3998, 2017.

[4] R. R. Irish *et al.*, "Characterization of the landsat-7 ETM+ automated cloud-cover assessment (ACCA) algorithm," *Photogramm. Eng. Remote Sens.*, vol. 72, no. 10, pp. 1179–1188, 2006.

[5] Z. Zhu and C. E. Woodcock, "Object-based cloud and cloud shadow detection in landsat imagery," *Remote Sens. Environ.*, vol. 118, pp. 83–94, 2012.

[6] Z. Zhu, S. Wang, and C. E. Woodcock, "Improvement and expansion of the fmask algorithm: Cloud, cloud shadow, and snow detection for landsats 4–7, 8, and sentinel 2 images," *Remote Sens. Environ.*, vol. 159, pp. 269–277, 2015.

[7] S. Chen, X. Chen, J. Chen, P. Jia, X. Cao, and C. Liu, "An iterative haze optimized transformation for automatic cloud/haze detection of landsat imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 5, pp. 2682–2694, May 2016.

[8] L. Sun *et al.*, "A cloud detection algorithm-generating method for remote sensing data at visible to short-wave infrared wavelengths," *ISPRS J. Photogramm. Remote Sens.*, vol. 124, pp. 70–88, 2017.

[9] P. P. Joshi, R. H. Wynne, and V. A. Thomas, "Cloud detection algorithm using SVM with SWIR2 and tasseled cap applied to landsat 8," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 82, pp. 101898–101908, 2019.

[10] D. U. Ufuk, C. Demirpolat, and M. F. Demirci, "Fast cloud detection using low-frequency components of satellite imagery," in *Proc. 25th Signal Process. Commun. Appl. Conf.*, 2017, pp. 1–4.

[11] H. Y. Cheng and C. C. Yu, "Block-based cloud classification with statistical features and distribution of local texture features," *Atmos. Meas. Techn.*, vol. 8, no. 3, pp. 1173–1182, 2015.

[12] A. Fisher, "Cloud and cloud-shadow detection in SPOT5 HRG imagery with automated morphological feature extraction," *Remote Sens.*, vol. 6, no. 1, pp. 776–800, 2014.

[13] P. Li *et al.*, "A cloud image detection method based on SVM vector machine," *Neurocomputing*, vol. 169, pp. 34–42, 2015.

[14] Y. Yuan and X. Hu, "Bag-of-words and object-based classification for cloud extraction from satellite imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 8, pp. 4197–4205, Aug. 2015.

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

[16] Y. Chen *et al.*, "Cloud and cloud shadow detection based on multiscale 3D-CNN for high resolution multispectral imagery," *IEEE Access*, vol. 8, pp. 16505–16516, 2020.

[17] A. Francis, P. Sidiropoulos, and J. P. Muller, "CloudFCN: Accurate and robust cloud detection for satellite imagery with deep learning," *Remote Sens.*, vol. 11, no. 19, pp. 2312–2324, 2019.

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

[19] Z. Shao, Y. Pan, C. Diao, and J. Cai, "Cloud detection in remote sensing images based on multiscale features-convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 4062–4076, Jun. 2019.

[20] S. Mohajerani and P. Saeedi, "Cloud-net: An end-to-end cloud detection algorithm for landsat 8 imagery," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 1029–1032.

[21] S. Mohajerani and P. Saeedi, "Cloud-Net+: A cloud segmentation CNN for landsat 8 remote sensing imagery optimized with filtered jaccard loss function," *arXiv:2001.08768.*

[22] Y. Chen *et al.*, "Multilevel cloud detection for high-resolution remote sensing imagery using multiple convolutional neural networks," *Int. J. Geo-Inf.*, vol. 7, no. 5, pp. 181–197, 2018.

[23] Y. Chen, L. Tang, X. Yang, R. Fan, M. Bilal, and Q. Li, "Thick clouds removal from multitemporal ZY-3 satellite images using deep learning," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 143–153, Dec. 2019.

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

[25] J. Guo, J. Yang, H. Yue, H. Tan, C. Hou, and K. Li, "CDnetV2: CNN-based cloud detection for remote sensing imagery with cloud-snow coexistence," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 700–713, Jan. 2021.

[26] J. H. Jeppesen *et al.*, "A cloud detection algorithm for satellite imagery based on deep learning," *Remote Sens. Environ.*, vol. 229, pp. 247–259, 2019.

[27] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

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

[29] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[30] S. Li *et al.*, "Predicting lung nodule malignancies by combining deep convolutional neural network and handcrafted features," *Phys. Med. Biol.*, vol. 64, no. 17, pp. 175012–175028, 2019.

[31] D. A. Ragab and O. Attallah, "FUSI-CAD: Coronavirus (COVID-19) diagnosis based on the fusion of CNNs and handcrafted features," *Peer J. Comput. Sci.*, vol. 6, pp. e306–e336, 2020.

[32] X. Sun, Q. Zhu, and Q. Qin, "A multi-level convolution pyramid semantic fusion framework for high-resolution remote sensing image scene classification and annotation," *IEEE Access*, vol. 9, pp. 18195–18208, 2021.

[33] X. Zhao, R. Tao, W. Li, W. Philips, and W. Liao, "Fractional gabor convolutional network formultisource remote sensing data classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 99, pp. 1–18, 2021.

[34] H. Luo *et al.*, "Land cover extraction from high resolution ZY-3 satellite imagery using ontology-based method," *ISPRS Int. J. Geo-Inf.*, vol. 5, no. 3, pp. 31–47, 2016.

[35] J. Zhou *et al.*, "Comparison of GF2 and SPOT6 imagery on canopy cover estimating in northern subtropics forest in China," *Forests*, vol. 11, no. 4, pp. 407–421, 2020.

[36] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.

[37] S. Arivazhagan, L. Ganesan, and S. P. Priyal, "Texture classification using gabor wavelets based rotation invariant features," *Pattern Recognit. Lett.*, vol. 27, no. 16, pp. 1976–1982, 2006.

[38] J. Hu *et al.*, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Aug. 2020.

[39] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv:1511.07122*.

[40] M. Sandler *et al.*, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.

[41] L.-C. Chen *et al.*, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 833–851.

[42] K. He *et al.*, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[43] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Machine Learn.*, 2015, pp. 448–456.

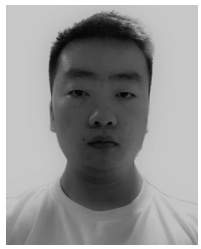
[44] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.

[45] M. J. Hughes and D. J. Hayes, "Automated detection of cloud and cloud shadow in single-date landsat imagery using neural networks and spatial post-processing," *Remote Sens.*, vol. 6, no. 6, pp. 4907–4926, 2014.

[46] C. Yu *et al.*, "BiSeNet: Bilateral segmentation network for real-time semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 325–341.

[47] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.

[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*.



Hongwei Guo received the B.E. degree in weapon system and launch engineering from the Nanjing University of Science and Technology, Nanjing, China in 2016, where he is currently working toward the Ph.D. degree in weapon science and technology with the School of Energy and Power Engineering.

His research interests include deep learning, image processing, and computer version.

Hongyang Bai received the Ph.D. degree in weapon science and technology from the Nanjing University of Science and Technology, Nanjing, China, in 2012.

He is currently an Associate Professor with the Nanjing University of Science and Technology. His research interests include integrated navigation, terminal guidance technology, and precisely target recognition for missiles and rockets.

Weiwei Qin received the Ph.D. degree in control science and engineering from the National University of Defense Technology, Changsha, China, in 2012.

He has been an Associate Professor and a Doctoral Supervisor with the Xi'an Research Institute of High-Tech, Xi'an, China, since 2018. His research interests include model predictive control, aircraft guide and control, and deep learning theory and methods.