applied sciences

MDPI

Article

Cloudformer: Supplementary Aggregation Feature and Mask-Classification Network for Cloud Detection

Zheng Zhang, Zhiwei Xu, Chang'an Liu, Qing Tian and Yanping Wang *

School of Information Science and Technology, North China University of Technology, Beijing 100144, China; zhangzheng@ncut.edu.cn (Z.Z.); xuzhiwei98@mail.ncut.edu.cn (Z.X.); furk0416@mail.ncut.edu.cn (C.L.); tianqing@ncut.edu.cn (Q.T.)

* Correspondence: wangyp@ncut.edu.cn

Abstract: Cloud detection is an important step in the processing of optical satellite remote-sensing data. In recent years, deep learning methods have achieved excellent results in cloud detection tasks. However, most of the current models have difficulties to accurately classify similar objects (e.g., clouds and snow) and to accurately detect clouds that occupy a few pixels in an image. To solve these problems, a cloud-detection framework (Cloudformer) combining CNN and Transformer is being proposed to achieve high-precision cloud detection in optical remote-sensing images. The framework achieves accurate detection of thin and small clouds using a pyramidal structure encoder. It also achieves accurate classification of similar objects using a dual-path decoder structure of CNN and Transformer, reducing the rate of missed detections and false alarms. In addition, since the Transformer model lacks the perception of location information, an asynchronous position-encoding method is being proposed to enhance the position information of the data entering the Transformer module and to optimize the detection results. Cloudformer is experimented on two datasets, AIR-CD and 38-Cloud, and the results show that it has state-of-the-art performance.

Keywords: cloud detection; mask classification; remote-sensing images; transformer

Citation: Zhang, Z.; Xu, Z.; Liu, C.; Tian, Q.; Wang, Y. Cloudformer: Supplementary Aggregation Feature and Mask-Classification Network for Cloud Detection. *Appl. Sci.* **2022**, *12*, 3221. https://doi.org/10.3390/app12073221

Academic Editors: Qizhi Xu, Jin Zheng and Feng Gao

Received: 25 December 2021
Accepted: 11 March 2022
Published: 22 March 2022

1. Introduction

Recently, space technology has been developing continuously [1]. The performance of remote-sensing satellites has been greatly improved and the remote-sensing data service system based on spatial information has been formed [2]. However, according to statistics, more than 65% of the Earth's surface is covered by clouds [3], most of the optical remote-sensing data are affected, and its analysis value is greatly reduced. As the "first step" of remote-sensing image processing tasks [4], cloud detection is an important means to assist researchers in evaluating the quality of remote-sensing data and in speeding up remote-sensing data processing. In addition, the cloud detection task is more challenging because of the diversity of the cloud layer itself and the complexity of the background in the remote-sensing data. Therefore, it is of great significance to study high-precision and high-university cloud detection methods.

At present, the research on cloud detection of optical remote-sensing image data is being widely carried out, and a large number of methods have emerged [5]. There are three main types of cloud detection methods: (1) traditional image processing methods based on thresholding and texture analysis, (2) statistical learning methods based on hand-designed features of physical attributes [6], and (3) deep learning methods based on deep convolutional neural networks (CNN).

FMask [7–9] is a well-known cloud detection method based on thresholds. This method supports multiple Landsat series satellites and has been applied for some scenarios. However, taking the Fmask3.2 algorithm as an example, it is necessary to use the atmospheric apparent reflectance image of the multispectral band as the main input, and at the same time use the atmospheric reflectance image of the cirrus bands as the auxiliary

inputs to get good results. This limits the university of the FMask method. Although statistical-based processing methods have been designed on the basis of learning, the feature extraction methods of the models are still mainly designed manually. For example, Xu et al. propose a model that uses multiple spectra to extract temporal and spatial features and put the features of a Bayesian probability model for classification. Handcrafted traditional image processing methods mostly require manual designs of parameters such as thresholds, which will cause the quality of the model to be highly dependent on the designer's professional knowledge and professional experience. When the model requires multiple spectral band data, this will further reduce the versatility of the model.

Cloud detection methods based on convolutional neural networks (CNN) [10] have achieved advanced results for a variety of scenes [11], and the idea of semantic segmentation of images by pixel classification has joined the current mainstream. The CNN-based method has made important breakthroughs in cloud detection with its excellent local feature extraction capabilities. For example, Cloud-Net+ [12] is a fully convolutional neural network model that focuses on using Landsat8 satellite data. It achieved state-of-the-art (SOTA) on the 38-Cloud public dataset. The deep learning method has the ability of automatic feature extraction and automatic learning, which can still guarantee good results when only a small amount of spectral band data is used as input. Consequently, the cloud detection method based on deep learning has strong versatility [13]. However, these methods still have some limitations. The receptive field of CNN is limited by the size of the convolutional kernel, leading to the shortcomings of CNN-based methods in extracting global features.

With the development of computer vision technology, the Transformer [14] network, which itself is used in the direction of natural language processing, began to be introduced into the field of computer vision and demonstrated excellent performance. By dividing the image of multiple patches, the image can be smoothly sent to the Transformer. Using Transformer's attention mechanism to establish connections between different patches can commendably capture the global characteristics of the image. Recently, many works have begun to try to apply the mask-classification to semantic segmentation tasks. The learnable queries of the Transformer can well integrate mask-classification into the model. From another point of view, the Transformer model requires a lot of calculation and cannot effectively capture the regional features. At the same time, the Transformer model is difficult to converge on the training process.

For cloud detection, the physical characteristics of the cloud make detection more difficult. For example, a small cloud that occupies a small number of pixels has a greater probability of being ignored in the feature extraction process. When the cloud cover is thin, the combination of the cloud and the background makes the model wrong. In some scenes where clouds and snow coexist, the cloud pixels are visually very similar to the snow pixels, causing the model to incorrectly classify snow as clouds.

We try to use the latest achievements in the field of semantic segmentation to solve the above problems, as well as the new cloud detection method (Cloudformer) that we propose. Cloudformer integrates CNN and Transformer. Cloudformer uses convolutional structure to initially extract features, and then uses Transformer's excellent Global feature extraction capabilities to help the model better predict the spatial information of the cloud. The local features extracted by CNN will help the architecture distinguish similar objects (such as clouds and snow). In addition, since the Transformer cannot capture the sequential relationship of the input sequence, it is necessary introduce positional information in the calculation in most cases. For this issue, we propose an asynchronous position encoding to enhance the position information of the data entering the Transformer module and to optimize the detection results.

Most importantly, the main contributions of this work are as follows:

1. We have designed a set of Spatial pyramid structure encoder (SPS encoder), which can efficiently to extract features from remote-sensing images and provide a basis for the work of the entire architecture.

2. To better integrate the deep features and shallow features, and to maximize the potential relationship between them, we designed a pixel-level decoder structure (SupA decoder) with a supplementary aggregation layer.
3. To protect the position information about the feature map, we propose a position asynchronous coding method. It enlarges the position information on the feature map as much as possible, allowing the features extracted from the pixel-level encoder to enter the Transformer module in a more reasonable way.

The rest of this article is organized as follows. In Section 2, recent works in the field of semantic segmentation and cloud detection are introduced. In Section 3, we will show our framework in detail. Section 4 discusses the results of the experiment. Finally, Section 5 summarizes our work.

2. Related Works

Cloud detection has been noticed recently because of the development of optical remote-sensing technology. In this section, we will introduce some works related to our model.

2.1. Cloud Detection with CNN-Based

Thanks to the excellent performance of convolution in computer vision [15], the deep learning method has been widely used in remote sensing. For example, Biserka Petrovska et al. used deep learning methods to extract features from remote-sensing images and perform classification [16], and their experiments achieved good results. Jacob H et al. designed a cloud detection method named RS-Net [17], which is a framework based on U-net. RS-Net shows SOTA performance, especially on smaller satellites with limited multi-spectral capabilities. Meanwhile, Zhen Feng Shao et al. proposed a MF-CNN [18] model based on multi-scale feature extraction. This method takes the combined spectral information as input of MF-CNN to enhance the model's ability to detect thin clouds. Li et al. introduced a weakly supervised deep-learning-based cloud detection method abbreviation WDCD [9]. WDCD uses the block-level labels to indicate whether there is cloud of the image block, thus reducing the workload of image annotation.

Mohajerani et al. proposed an End-to-End Algorithm Cloud-Net [19] that consists of a fully convolutional network. This method not only achieves good precision performance, but also seldom requires complex data preprocessing. Ding et al. designed a method based on Fully Convolutional Neural Networks named CM-CNN for FY-3D MERSI [20]. This model has good performance while only using mid-infrared and long-infrared band data. Kai Zheng et al. proposed an Encoder–Decoder Deep Convolutional Neural Network [21]. This network is used to perform cloud and snow segmentation.

However, most of these methods do not pay attention to the feature extraction of small clouds. This will lead to frequent omissions when a large number of broken clouds appear in remote-sensing images.

2.2. Transformer-Based Computer Vision Method

In 2017, Google proposed the Transformer framework [14]. The method abandoned the traditional CNN and RNN, the further constructs a model based on the self-attention mechanism. Transformer has demonstrated powerful performance in Natural Language Processing (NLP) years ago. Recently, it also began to emerge in computer vision.

Alexey Dosoviskiy et al. proposed the Vision Transformer (ViT) in 2020 [22]. This was an attempt made in Computer Vision. ViT was used for Image Classification and obtained a satisfactory performance. This work proves the feasibility of Transformer in Computer Vision.

A few months after that, more than 100 Transformer-based methods have been proposed by researchers [23]. This phenomenon shows us the vitality and attraction of Transformer. For example, the DETR [24] proposed by Facebook has made Transformer a breakthrough in Object Detection. DETR not only abandons Non-Maximum Suppression

(NMS), but it also detects object by set prediction and object query. These designs made it the very first End-to-End Transformer detector.

Recently, Swin Transformer [25] and CSwin Transformer [26] made unnecessary the discussion over whether Transformer or CNN were better. These methods introduced the concept of "window". At the same time, all components in the model began to integrate the idea of CNN. It also brings some inspiration to the method proposed in this paper.

2.3. Mask Classification

In an instance-level segmentation task, mask classification is very common. For example, Mask-R CNN [27] is an excellent instance segmentation algorithm first proposed by He et al. in 2017. However, per-pixel classification is still the mainstream in semantic-level segmentation tasks. Bowen Cheng et al. proposed the MaskFormer [28], which applied mask classification to semantic-level segmentation tasks. Meanwhile, its performance exceeded that of per-pixel classification baselines.

Skillfully applying learnable query embeddings is the key to the Transformer combined with mask classification. Researchers are committed to using queries to directly predict masks. The main ideas are as follows: to replace object queries with mask embeddings and then to perform classification and mask prediction directly through multiple embeddings from Transformer. Several methods based on Transformer with mask embeddings have been proposed, such as Max-Deeplab [29] and Segmenter [30].

Because the background of a remote-sensing image is complex, and the object characteristics are similar, we hope to design a framework that combines the advantages of CNN, Self-attention, and Transformer with mask embeddings. The details will be described in the next section.

3. Methodology

Cloudformer is a semantic segmentation framework that incorporates the mask prediction branch. On the one hand, we constructed the pixel-level branch through traditional convolution. In this branch, we extracted multi-scale features and establishing the dependency relationships between deep semantic features and shallow spatial features was the main goal. On the other hand, the mask-level branch built by the Transformer decoder will make the final predictions based on the pixel-level branch.

The structure of the proposed Cloudformer is shown in Figure 1. It contains four key modules: (1) the multi-scale extraction module Spatial Pyramid Structure Encoder (SPS Encoder), (2) the Supplementary Aggregation Decoder (SupA Decoder), (3) the Transformer module, and (4) mask segmentation.

3.1. Overview

First, we apply the Cloudformer to a remote image-sensing cloud detection task. To avoid cloud feature loss consisting only of a few pixels, features will be extracted from different scale spaces by the Spatial Pyramid module. Then the feature representation with the deep branch and shallow branch becomes complementary, and there is no communication between them. To enhance communication, we proposed a Supplementary Aggregation Layer. In addition, the deep semantic features that get by the SPS Encoder will be sent to the Transformer module. Calculations are made concerning the N embeddings through the self-attention mechanism. These embeddings will help the framework make more accurate predictions. In mask segmentation, the binary mask prediction diagram is obtained by using the mask embeddings of the N embeddings under MLP mapping; the class probability prediction is carried out using class predictions, and the output prediction is obtained after combination.

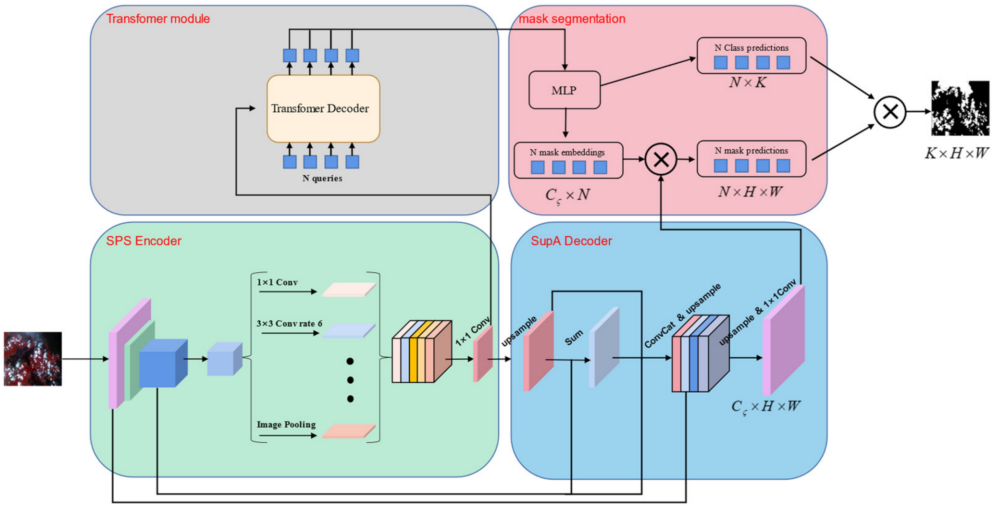


Figure 1. There are four key components in the framework: SPS Encoder, SupA Decoder, Transformer module, and mask segmentation. The remote-sensing data are input into the SPS encoder to extract the feature of the cloud through convolution. Then the feature is sent to the SupA decoder for pixel-level decoding. The feature will also be sent to the Transformer module to use the attention mechanism for further calculations to obtain mask information and classification embeddings of information. Finally, the two parts of information are combined in the mask segmentation to obtain the prediction result.

3.2. SPS Encoder

To preserve the spatial characteristics of the clouds and make the framework ignore as little as possible the small-sized clouds that occupy fewer pixels, a structure that contains only a few down-samplings was designed, as shown in the Figure 2. The image of size $H \times W$ is first fed into the encoder, and the feature maps are obtained by convolution and pooling. Then, the feature maps are fed into the pyramid structure. The three dilated convolutions with different expansion rates are used to extract features and improve the global receptive field while maintaining the resolution. In addition, the pyramid structure extracts features from multiple-scale spaces to enhance the expressive information of the clouds and enhance the mining of the hidden relationships between pixels. To ensure that the characteristic signal can densely cover more areas, we concatenate the features extracted from differently scaled spaces. Next, the SupA encoder further extracts the information and adjusts the number of channels by convolution. From this framework, a set of feature maps ℓ_{deep} containing rich information can be obtained as:

$$\ell_{deep} \in \mathbb{R}^{C_{\ell} \times \frac{H}{S^n} \times \frac{W}{S^n}} \tag{1}$$

where C_{ℓ} is the number of channels, S is stride, and n means the down-sampling frequency. These feature maps will be the basis of pixel-level analysis and mask-level analysis of the whole framework.

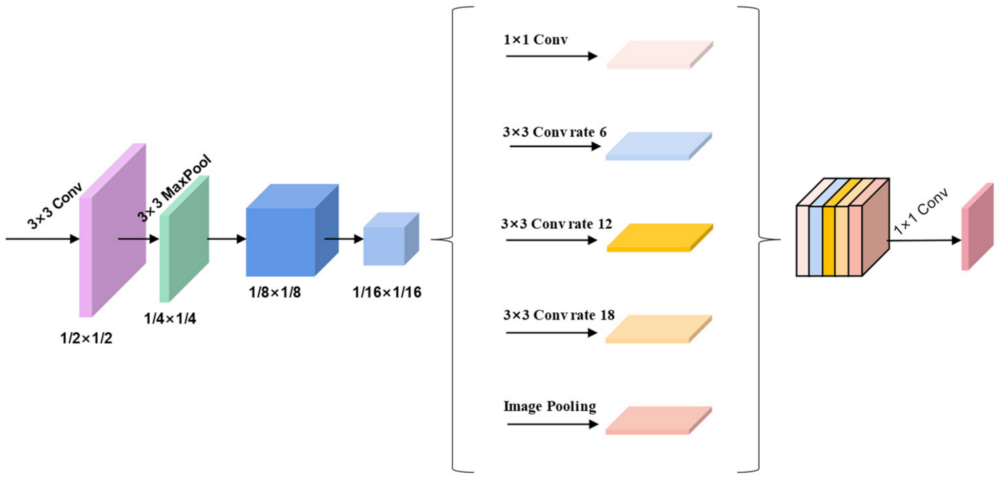


Figure 2. The structure of the Spatial Pyramid Structure Encoder.

3.3. SupA Decoder

As a pixel-level decoding structure, the feature maps need to be restored to the same size. Although the deep features have rich semantic information, their spatial information has been lost, which is inevitable and irreversible. Therefore, a large amount of spatial information will be lost if only deep features are used for decoding. Referring to U-Net [31], we use shallow features containing rich spatial information to help SupA decoder recover more information when decoding. However, the simple concatenate operation disregards the richness between the two features. To overcome this problem, the computation link of corresponding elements is introduced into the module.

As shown in Figure 3, first we sum up the feature $\ell_{shallow}$ of size $H/8 \times W/8$ and the deep feature ℓ_{deep} after an up-sampling operation for element-by-element aggregation to obtain ℓ_s. Then we concatenate three feature maps. In addition, the context information is supplemented by features ℓ with the size of $H/2 \times W/2$. We refer to BiseNet v2 [32] for feature aggregation and use a more common operation of fusing deep and shallow features in equal proportion. The specific process is as follows:

$$\ell_{decoder} = \text{Concat}[f\left(\ell_{deep}\right), f(\ell_{shallow}), f(\ell_s), \ell] \tag{2}$$

$$[\ell_s]_{i,j} = \alpha_{i,j} + \beta_{i,j} (\alpha \in \ell_{deep}, \beta \in \ell_{shallow}) \tag{3}$$

where f is bilinear up-sampling. Finally, we adjust the dimensionality of $\ell_{decoder}$ by 1×1 convolution and reshape it to $\ell_{pixel} \in \mathbb{R}^{C_\ell \times H \times W}$ by bilinear up-sampling. C_ξ is the same embedding dimension in the mask-level branch.

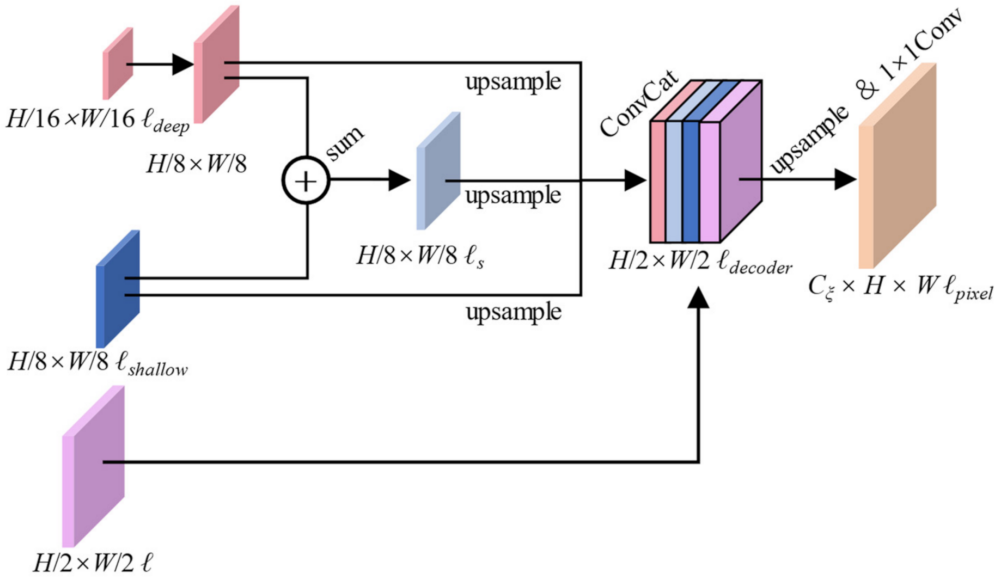


Figure 3. The structure of the Supplementary Aggregation Decoder.

3.4. Transformer Module

The Transformer module is a major component of the mask-level branch. We use a standard Transformer decoder to obtain the N embeddings output used for mask prediction and class prediction. Converting the feature maps ℓ to token embeddings is a prerequisite for feeding them into the Transformer decoder. Dimensionality reduction will lead to the loss of element position information. To preserve this information, we introduced a positional encoding method.

As shown on the left of Figure 4, we created a tensor with the same dimensions as the feature map ℓ_{deep}, and established a Cartesian coordinate system with the first element in the lower left corner of the tensor as the origin. The value of the element in the tensor is the sum of the horizontal and vertical coordinates of the current element. We easily find a lot of similarities in the position coding of the elements, although they are not close to each other. To this end, we expand the horizontal coordinates of the elements to twice their original size and the vertical coordinates to three times their original size. Such a change brings two advantages: (1) it enhances distance signal between elements, and (2) increases the sparsity of the position coding value.

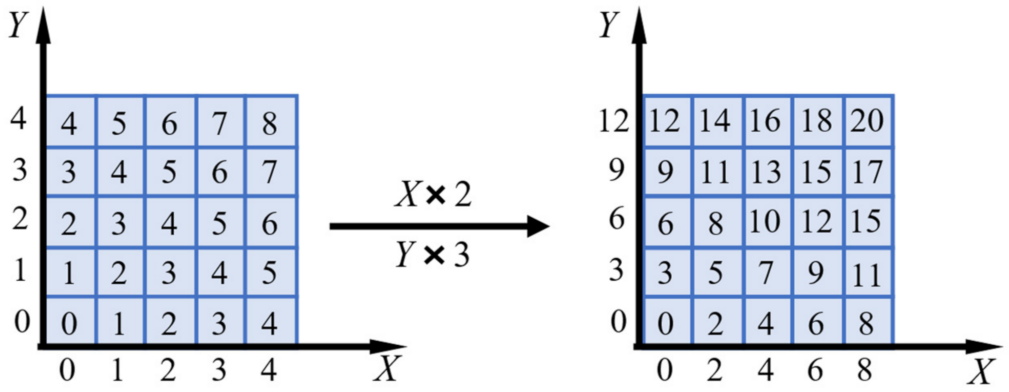


Figure 4. Establish a coordinate system based on the elements in the lower left corner, and expand the abscissa to twice the original size, and the ordinate to triple the original size. Such an encoding method can amplify the position information between elements and reduce the encoding repetition rate.

As shown on the right of Figure 5, this method produces a more reasonable position-coding map.

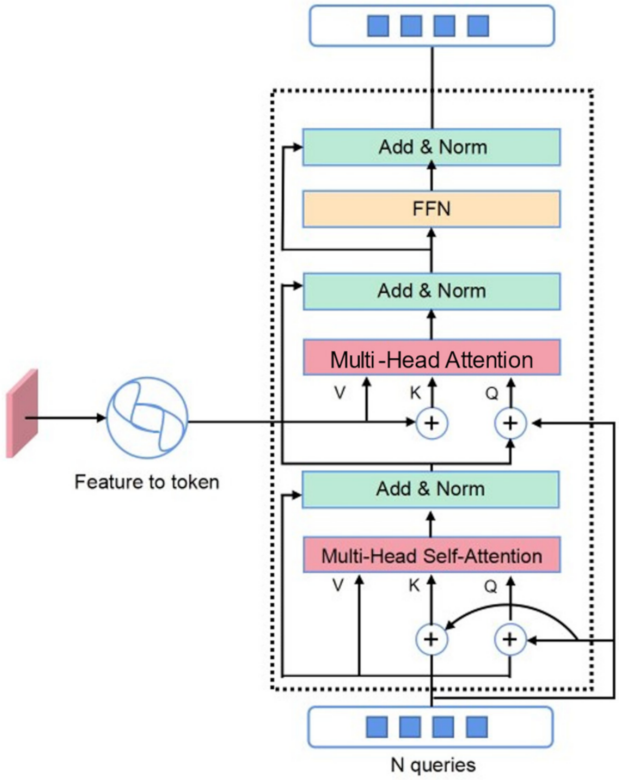


Figure 5. The Transformer module consists of a standard Transformer decoder and a feature converter. To strengthen the position information between the features, the feature converter uses an asynchronous position code.

Afterwards, we reshape the position-coding map with the same method as feature maps ℓ_{deep} to obtain a position embedding. Finally, position embeddings and token embeddings are added and sent into the Transformer component to calculate the N outputs mentioned above in parallel with the N learnable queries.

3.5. Mask Segmentation

As shown in the upper right of Figure 1, the N embeddings output from the Transformer module are inputs into the mask segmentation module. We obtain two sets of information through the Multilayer Perceptron (MLP) structure: (1) N class predictions, and (2) N mask embeddings.

The former set of information is obtained by a linear classifier and SoftMax activation. Per-segment queries in class predictions produce $K + 1$ prediction probabilities $\{P_i \mid P_i \in \Delta^{K+1}\}$. It contains K categories and a "no object" category. Therefore, in the cloud detection task, $\ell_{class} \in \mathbb{R}^{N\times(K+1)}$, where $K = 2$ (cloud and background). This classification information is obtained by fusing deep semantic features through a multi-head attention mechanism. Therefore, it combines semantic features and context information more effectively than the pixel-by-pixel-based approach and can help distinguish objects with similar features (such as clouds and snow).

For the latter set of information, a Multilayer Perceptron (MLP) will convert the input N queries into N mask embeddings. The length of pre-segment embeddings is C_ζ. A large amount of spatial information is contained in the mask embeddings. We embed N mask embeddings ℓ_{mask} at the mask-level into the output of the pixel-level branch through dot

products and then apply sigmoid activation to obtain the binary spatial mask prediction. This process can be defined as:

$$M = Sigmoid({\ell_{mask}}^{\mathrm{T}} \cdot \ell_{pixel}) M \in \mathbb{R}^{N \times H \times W} \quad (4)$$

where T means transpose.

Finally, class prediction and mask prediction are combined to obtain the standard semantic segmentation task output. We refer to DETR [24]; a focal loss and a dice loss are used as binary spatial mask prediction. For class prediction, cross entropy classification loss is the best choice. The overall loss function can be written as:

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{mask} \quad (5)$$

$$\mathcal{L}_{mask} = \lambda_{focal} \mathcal{L}_{focal} + \lambda_{dice} \mathcal{L}_{dice} \quad (6)$$

where λ_{focal}, and λ_{dice} balance the weight of two losses as hyper-parameters. We set the hyper-parameters to λ_{focal} = 20.0 and λ_{dice} = 1.0.

4. Experiment

In this section, we conduct a detailed evaluation of Cloudformer on the AIR-CD and 38-Cloud datasets. Specifically, we first briefly introduce the dataset processing and experimental details, and then we show the ablation experiments on the main components of Cloudformer. Finally, the overall performance of the model is compared and analyzed.

4.1. Evaluation Criteria and Data Processing

On the one hand, to evaluate the actual overall performance of various methods of cloud detection tasks, we selected four widely used indicators, namely Mean Intersection with Union (MIoU) [33], Frequency weighted Intersection over Union (FwIoU) [34], Mean Accuracy (MAcc) [35], and Pixel Accuracy (PAcc) [36]. The selected index will evaluate the accuracy of the model from multiple angles. The larger the value, the higher the accuracy. In the ablation experiment, our purpose was to illustrate the effectiveness of the module, so we only selected MIoU, MAcc, and PAcc for evaluation. In the tables of this article, for the MIoU, FwIoU, MAcc and PAcc we bolded the highest values, while for the time is targeted the lowest value.

On the other hand, we choose two public remote image-sensing cloud detection datasets to test the generalization ability of the model in different scenarios. The AIR-CD dataset contains multiple remote-sensing images of 7300 × 6908 obtained by GF-2 satellites [37]. We divide it into 640 × 640 RGB images as the training set of the network and add random noise, such as rotation and horizontal flipping at a specific angle, before sending it to the network to enhance the training data. The 38-Cloud dataset contains multiple remote-sensing images 7601 × 7761 in size obtained by LandSat8 satellites. When inputting into Cloudformer, we discarded the data on the near-infrared band, and only stacked the visible bands, to perform the image analysis. Overlapping segmentation constructed a dataset containing 3249 640 × 640 RGB images. In the experiments in this section, the training set and the test set are constructed at a ratio of 8 to 2.

4.2. Compare Models and Experimental Settings

This section compares Cloudformer with several recent representative cloud detection models. On the AIR-CD dataset, we chose DABNet [38] and CDNet [39], which perform well on this dataset, and the classic semantic segmentation network DeepLabv3+ [40] for comparison. On the 38-Cloud dataset, we selected the Cloud-Net [19] and Cloud-Net+ [12] series networks proposed by the contributors of the dataset, and the classic U-net [31] network for comparison. To ensure fairness, ResNet-50 [41] was set as the backbone network of all segmented networks.

In the experiment, each model is trained on two constructed datasets, using the same division method to ensure that the training set and test set of the comparison model are the same, and that it only detects clouds, not cloud shadows. During the training process, we uniformly set the batch size to 8, the initial learning rate is set to 1×10^{-6}, and then the learning rate approximates an exponential decay, and a total of 160,000 batches are trained. The Adam algorithm is used to optimize the model. All experiments in this article are implemented using NVIDIA RTX2080TI GPU under the Pytorch framework.

4.3. Ablation Experiment

To verify the performance of Cloudformer, we conducted ablation experiments on a supplementary aggregation decoder (SupA Decoder), asynchronous position coding, and mask embeddings queries.

4.3.1. Effect of SupA Decoder

Figure 5 shows the structure of the decoder we designed. We use the deeplabv3+ model as the baseline to perform ablation experiments on the SupA Decoder. To avoid interference from other factors, the experiment chose ResNet-50 as the backbone and added several design schemes of the SupA Decoder in the experiment. Therefore, the data in the table are that the deeplabv3+ decoder only combines the decoding structure of the 1/8 size feature map, and aggregates the decoding structure of the 1/8 and 1/2 feature maps, and the final SupA Decoder.

As Table 1 shows, the decoding structure that simultaneously aggregates the 1/8 and 1/2 feature maps is similar in accuracy to the final SupA Decoder, but the speed of inference is greatly reduced. Therefore, the final SupA Decoder is a better choice for comprehensive accuracy and speed.

Table 1. Pixel-level decoder ablation experiment results.

Method	MIoU (%)	MAcc (%)	PAcc (%)	Reasoning Time (s)
baseline	85.76	86.11	95.23	**0.3360**
+SupA (1/8)	86.32	87.26	95.77	0.3523
+SupA (1/8 + 1/2)	**87.81**	89.13	96.55	0.5107
+SupA (1/8) + 1/2	87.79	**89.41**	**96.58**	0.3610

4.3.2. Effect of Asynchronous Position Coding

In this experiment, we keep the other parts of the model unchanged, and only change the operation when the pixel-level decoder extracts the feature map and sends it to the Transformer module. We compared the effects of not adding position coding, synchronous long position coding, and asynchronous position coding. Table 2 shows the model with asynchronous position coding added and the performance is slightly improved. This is because asynchronous position coding improves the difference in coding and enlarges the position information in the coding, reducing the loss of hidden information when resizing the feature map.

Table 2. Asynchronous position coding ablation experiment results.

Method	MIoU (%)	MAcc (%)	PAcc (%)
Baseline	95.89	96.79	98.23
+Synchronization	96.18	97.19	98.77
+Asynchronous	**96.47**	**97.78**	**98.89**

4.3.3. Effect of Mask Embeddings Queries

To verify the effect of mask classification in cloud-detection tasks, we refer to the method of Kirillov et al., and compare the structure of PerPixelBaseline with Cloudformer.

In this experiment, the structure of PerPixelBaseline is shown in Figure 6. The figure shows that Cloudformer has better performance for thin cloud detection. The learnable queries in the Transformer module can use features from a more comprehensive perspective to achieve more accurate classification. From the data point of view, Cloudformer also performs better than PerPixelBaseline. The specific data are shown in Table 3.

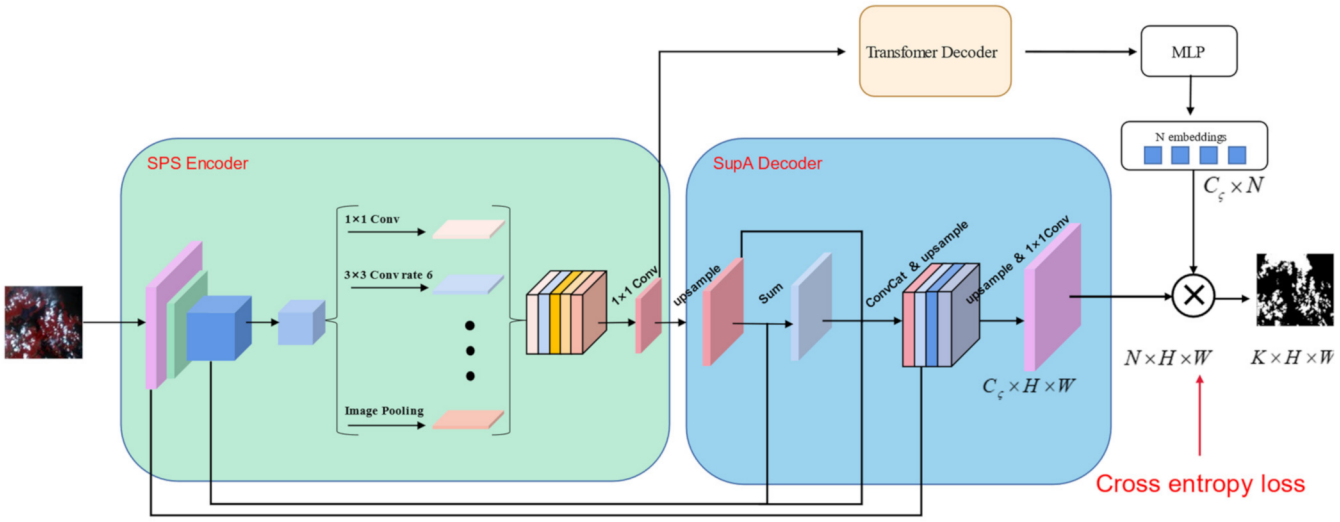
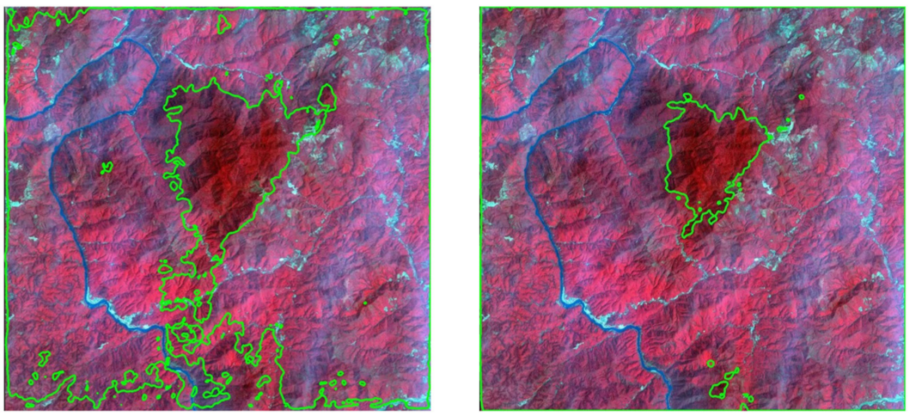


Figure 6. In PerPixelBaseline, the structure of the pixel-level branch is maintained, and the Transformer module is also retained, but, finally, a traditional cross entropy loss is used for per-pixel classification.

Table 3. Mask embeddings queries ablation experiment results.

Method	MIoU (%)	MAcc (%)	PAcc (%)
PerPixelBaseline	92.31	97.28	98.32
Cloudformer	**96.56**	**98.29**	**99.07**

At the same time, we can see from Figure 7 that the mask classification method is better than the per-pixel classification method for thin cloud detection.



(**a**) (**b**)

Figure 7. Comparison of mask classification method and per-pixel classification method on AIR-CD dataset. (**a**) Per-pixel classification. (**b**) Mask classification.

4.4. Comparison with State-of-the-Art Methods

4.4.1. AIR-CD Dataset

Table 4 and Figure 8 demonstrate the performance of Cloudformer and other methods on the AIR-CD dataset [38]. Judging from the evaluation results, Cloudformer has an accuracy that surpasses other cloud-detection methods. At the same time, because of the accurate annotation of the AIR-CD dataset, each model demonstrates good performance on this dataset. Table 4 shows that Cloudformer has surpassed other methods in the four selected indicators. The other models in this experiment are all classic CNN-based methods. Cloudformer adds the Transformer module based on CNN. With Transformer's strong global attention, the model can use features from different angles, so that Cloudformer has a good performance in accuracy and, especially in the detection of thin clouds and small clouds, Cloudformer has a huge advantage. As well, the design of mask classification helps this model improve the clarity of cloud boundaries in cloud-detection tasks.

Table 4. Comparison of different cloud-detection methods on AIR-CD.

Method	MIoU (%)	FwIoU (%)	MAcc (%)	PAcc (%)
Deeplabv3+ [40]	89.76	90.02	94.62	95.08
CDNet [39]	91.83	93.04	97.45	96.72
DABNet [38]	92.08	93.25	97.69	98.43
Cloudformer (ours)	**96.56**	**98.17**	**98.29**	**99.07**

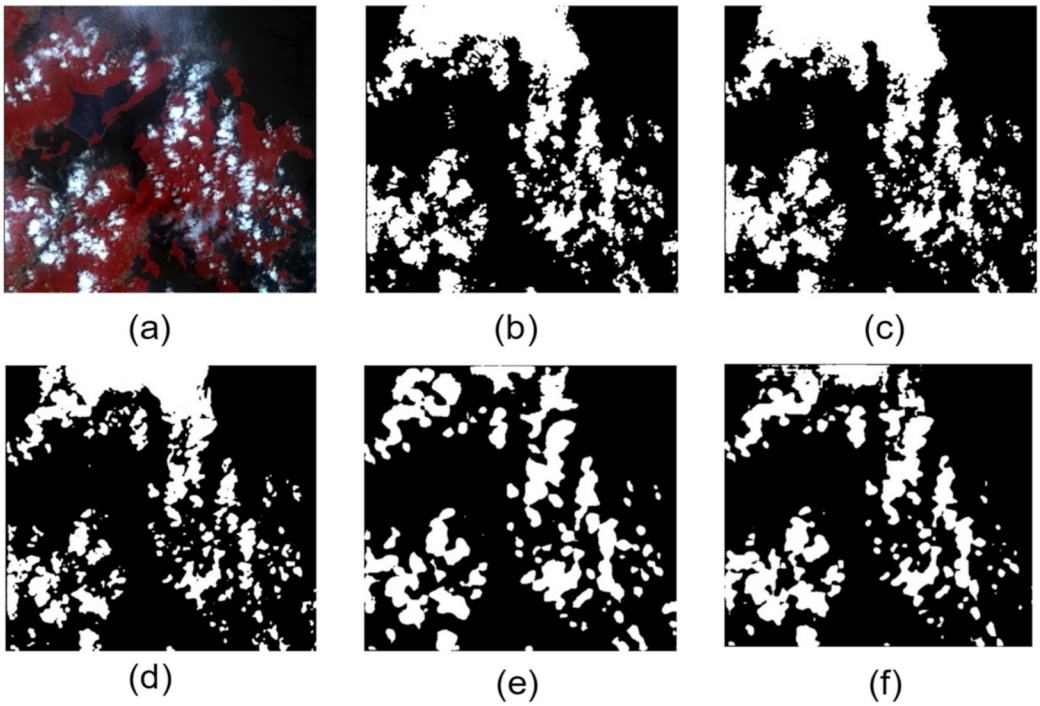


Figure 8. Comparison of different cloud-detection methods on the AIR-CD dataset. (**a**) Image. (**b**) Ground truth. (**c**) Cloudformer. (**d**) DABNet. (**e**) Deeplabv3+. (**f**) CDNet.

4.4.2. 38-Cloud Dataset

Table 5 and Figure 8 show the performance of our model and other methods on the 38-cloud dataset. From the data in Table 5, we can see that Cloudformer has obtained similar results to Cloud-Net+ on the 38-Cloud dataset. However, Cloudformer only inputs the visible light spectrum band data, and Cloud-Net+ inputs the near-infrared spectrum data at the same time as the visible light spectrum band data. This means that Cloudformer has higher versatility and can be applied to remote-sensing satellites equipped with only a

few spectral band sensors. Figure 9 shows the visual mask of the detection results of each model. Visually, the detection results of Cloudformer are excellent.

Table 5. Comparison of different cloud-detection methods on 38-Cloud.

Method	MIoU (%)	FwIoU (%)	MAcc (%)	PAcc (%)
U-Net [31]	85.21	87.52	95.05	96.15
Cloud-Net [19]	87.32	88.26	95.86	97.60
Cloud-Net+ [12]	88.85	90.23	**96.35**	97.39
Cloudformer (ours)	**90.71**	**92.31**	96.33	**97.89**

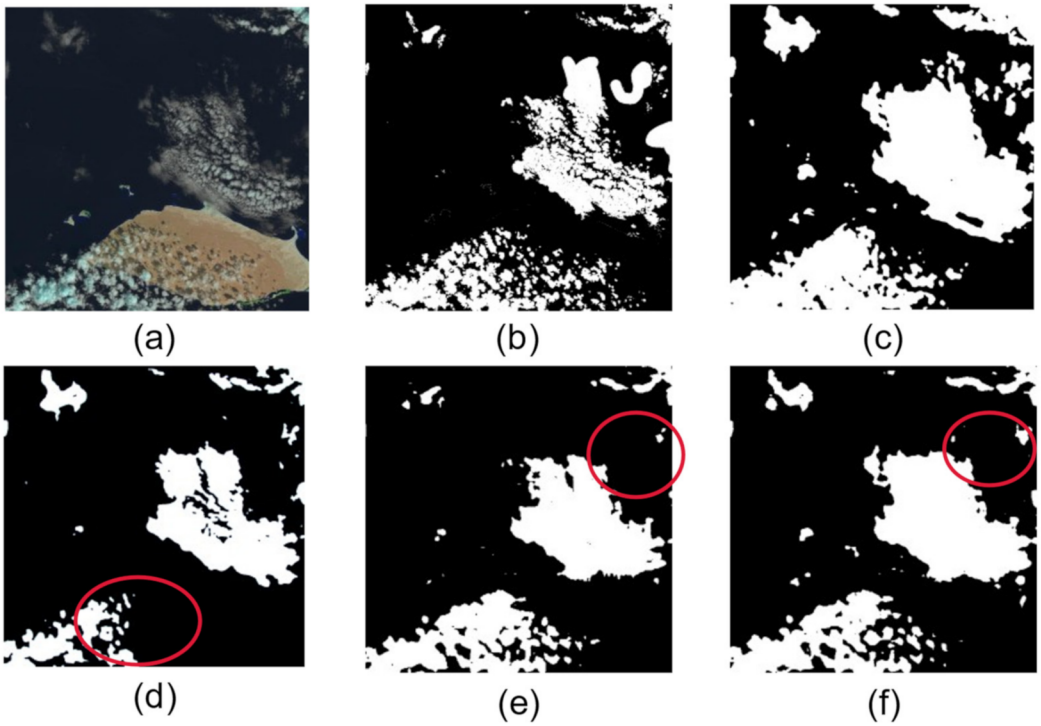


Figure 9. Comparison of different cloud detection methods on 38-Cloud dataset. (**a**) Image. (**b**) Ground truth. (**c**) Cloudformer. (**d**) Cloud-Net+. (**e**) Cloud-Net. (**f**) U-Net.

4.5. Versatility

To prove the versatility of the method for semantic segmentation tasks, we tested our method on a public dataset, ADE20K [42]. The dataset contains more than 25,000 pictures and 150 categories, is rich in scenes, and can comprehensively reflect the performance of the model in each scene. Cloudformer achieved 47.7% of MIoU in the mission. In Table 6, compared with the current methods [43,44] that perform well on the ADE20K dataset, the performance of Cloudformer is similar. It is sufficient to show that Cloudformer is versatile enough and can also be applied to segmentation tasks in other scenarios.

Table 6. Comparison of different semantic segmentation methods on ADE20K.

Method	MIoU (%)	Year
VIT-B [43]	48.1	2021
Cloudformer	47.7	-
ResNeSt-269 [44]	47.6	2020

4.6. Limitation

Cloudformer can achieve good accuracy in cloud detection tasks. Meanwhile, due to the overall complexity of the model, there is still room for improvement in inference speed, for example, the reasoning for a remote-sensing image with a size of 1024 × 1024. The inference running time of Cloudformer is 34% longer than that of DABNet under the same hardware conditions. This makes Cloudformer not the first choose in scenarios where there are strict requirements for inference speed. On the premise of ensuring accuracy, further improving the inference speed will be our most important next step. At the same time, in other scenarios, the accuracy of Cloudformer still has a lot of room for improvement. Next, we will optimize the model to further improve the overall performance of the method.

5. Conclusions

This paper proposes a cloud detection method for high-resolution remote-sensing images. Compared with the various detection methods currently released, Cloudformer has higher accuracy and stronger versatility. In general, our method discards the idea of pixel-by-pixel classification and uses multiple modules to integrate global and local information for feature extraction and processing. We designed SPS Encoder and SupA Decoder, which can extract richer features from the model. At the same time, we use asynchronous position coding, so that the feature map retains more position information when it enters the Transformer decoder. Experiments based on two public cloud-detection datasets show that Cloudformer is superior to other current cloud-detection methods. Experiments based on the ADE20K dataset also prove that our method is sufficiently versatile. In the future, we will focus on improving the reasoning speed of the model.

Author Contributions: Conceptualization, Z.Z., Z.X. and Q.T.; methodology, Z.Z. and Z.X.; software, Z.X.; validation, Q.T. and Z.X.; formal analysis, Z.X. and C.L.; investigation, Z.Z.; resources, Z.Z. and Z.X.; data curation, Z.Z. and Z.X.; visualization, Z.X.; writing-original draft preparation, Z.X.; writing-review and editing, Z.Z.; supervision, Y.W.; project administration, Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by the National Key R&D Program of China (No. 2018YFC1505100, No.2018YFC1505103) and by the Fundamental Research Fund of Beijing Municipal Education Commission and by North China University of Technology Research Start-up Funds.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Boulila, W.; Sellami, M.; Driss, M.; Al-Sarem, M.; Safaei, M.; Ghaleb, F.A. RS-DCNN: A novel distributed convolutional-neural-networks based-approach for big remote-sensing image classification. *Comput. Electron. Agric.* **2021**, *182*, 106014. [CrossRef]
2. Li, W.; He, C.; Fang, J.; Zheng, J.; Fu, H.; Yu, L. Semantic Segmentation-Based Building Footprint Extraction Using Very High-Resolution Satellite Images and Multi-Source GIS Data. *Remote Sens.* **2019**, *11*, 403. [CrossRef]
3. Hagolle, O.; Huc, M.; Pascual, D.V.; Dedieu, G. A multi-temporal method for cloud detection, applied to FORMOSAT-2, VENS, LANDSAT and SENTINEL-2 images. *Remote Sens. Environ.* **2010**, *114*, 1747–1755. [CrossRef]
4. Liu, Y.; Key, J.R.; Frey, R.A.; Ackerman, S.A.; Menzel, W.P. Nighttime polar cloud detection with MODIS. *Remote Sens. Environ.* **2004**, *92*, 181–194. [CrossRef]
5. Chen, Y.; Fan, R.; Bilal, M.; Yang, X.; Wang, J.; Li, W. Multilevel cloud detection for high-resolution remote sensing imagery using multiple convolutional neural networks. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 181. [CrossRef]
6. Zi, Y.; Xie, F.; Jiang, Z. A cloud detection method for landsat 8 images based on pcanet. *Remote Sens.* **2018**, *10*, 877. [CrossRef]
7. Qiu, S.; Zhu, Z.; He, B. Fmask 4.0: Improved cloud and cloud shadow detection in Landsats 4–8 and Sentinel-2 imagery. *Remote Sens. Environ.* **2019**, *231*, 111205. [CrossRef]
8. Zhu, Z.; Wang, S.; Woodcock, C.E. 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.* **2015**, *159*, 269–277. [CrossRef]

9. Li, Y.; Chen, W.; Zhang, Y.; Tao, C.; Xiao, R.; Tan, Y. Accurate cloud detection in high-resolution remote sensing imagery by weakly supervised deep learning. *Remote Sens. Environ.* **2020**, *250*, 112045. [CrossRef]
10. Drönner, J.; Korfhage, N.; Egli, S.; Mühling, M.; Thies, B.; Bendix, J.; Freisleben, B.; Seeger, B. Fast Cloud Segmentation Using Convolutional Neural Networks. *Remote Sens.* **2018**, *10*, 1782. [CrossRef]
11. Gao, Q.; Lim, S.; Jia, X. Hyperspectral image classification using convolutional neural networks and multiple feature learning. *Remote Sens.* **2018**, *10*, 299. [CrossRef]
12. Mohajerani, S.; Saeedi, P. Cloud and cloud shadow segmentation for remote sensing imagery via filtered jaccard loss function and parametric augmentation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 4254–4266. [CrossRef]
13. Foga, S.; Scaramuzza, P.L.; Guo, S.; Zhu, Z.; Dilley, R.D.; Beckmann, T.; Schmidt, G.L.; Dwyer, J.L.; Joseph Hughes, M.; Laue, B. Cloud detection algorithm comparison and validation for operational landsat data products. *Remote Sens. Environ.* **2017**, *194*, 379–390. [CrossRef]
14. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
15. Tang, T.; Zhou, S.; Deng, Z.; Zou, H.; Lei, L. Vehicle detection in aerial images based on region convolutional neural networks and hard negative example mining. *Sensors* **2017**, *17*, 336. [CrossRef]
16. Petrovska, B.; Zdravevski, E.; Lameski, P.; Corizzo, R.; Štajduhar, I.; Lerga, J. Deep Learning for Feature Extraction in Remote Sensing: A Case-Study of Aerial Scene Classification. *Sensors* **2020**, *20*, 3906. [CrossRef]
17. Jeppesen, J.H.; Jacobsen, R.H.; Inceoglu, F.; Toftegaard, T.S. A cloud detection algorithm for satellite imagery based on deep learning. *Remote Sens. Environ.* **2019**, *229*, 247–259. [CrossRef]
18. Shao, Z.; Pan, Y.; Diao, C.; Cai, J. Cloud detection in remote sensing images based on multiscale features-convolutional neural network. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 4062–4076. [CrossRef]
19. Mohajerani, S.; Saeedi, P. Cloud-net: An end-to-end cloud detection algorithm for landsat 8 imagery. In Proceedings of the 2019 IEEE International Geoscience and Remote Sensing Symposium, IGARSS, Yokohama, Japan, 28 July–2 August 2019; pp. 1029–1032.
20. Ding, Y.; Hu, X.; He, Y.; Liu, M.; Wang, S. Cloud detection algorithm using advanced fully convolutional neural networks in FY3D-MERSI imagery. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*; Peng, Y., Liu, Q., Lu, H., Sun, Z., Liu, C., Chen, X., Zha, H., Yang, J., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 615–625.
21. Zheng, K.; Li, J.; Ding, L.; Yang, J.; Zhang, X.; Zhang, X. Cloud and Snow Segmentation in Satellite Images Using an Encoder–Decoder Deep Convolutional Neural Networks. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 462. [CrossRef]
22. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2021**, arXiv:201011929.
23. Liu, Y.; Zhang, Y.; Wang, Y.; Hou, F.; Yuan, J.; Tian, J.; Zhang, Y.; Shi, Z.; Fan, J.; He, Z. A survey of visual transformers. *arXiv* **2021**, arXiv:211106091.
24. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with Transformers. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 213–229.
25. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021.
26. Dong, X.; Bao, J.; Chen, D.; Zhang, W.; Yu, N.; Yuan, L.; Chen, D.; Guo, B. CSWin Transformer: A General Vision Transformer Backbone with Cross-Shaped Windows. *arXiv* **2021**, arXiv:210700652.
27. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–27 October 2017; pp. 2961–2969.
28. Cheng, B.; Schwing, A.G.; Kirillov, A. Per-pixel classification is not all you need for semantic segmentation. *arXiv* **2021**, arXiv:210706278.
29. Wang, H.; Zhu, Y.; Adam, H.; Yuille, A.; Chen, L.-C. MaX-DeepLab: End-to-End Panoptic Segmentation with Mask Transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 5463–5474.
30. Strudel, R.; Garcia, R.; Laptev, I.; Schmid, C. Segmenter: Transformer for semantic segmentation. *arXiv* **2021**, arXiv:210505633.
31. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In Proceedings of the 18th International Conference, Munich, Germany, 5–9 October 2015; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241.
32. Yu, C.; Gao, C.; Wang, J.; Yu, G.; Shen, C.; Sang, N. BiSeNet V2: Bilateral Network with Guided Aggregation for Real-time Semantic Segmentation. *arXiv* **2020**, arXiv:200402147. [CrossRef]
33. Yuheng, S.; Hao, Y. Image segmentation algorithms overview. *arXiv* **2017**, arXiv:170702051.
34. Artacho, B.; Savakis, A. Waterfall atrous spatial pooling architecture for efficient semantic segmentation. *Sensors* **2019**, *19*, 5361. [CrossRef] [PubMed]
35. Thoma, M. A survey of semantic segmentation. *arXiv* **2016**, arXiv:160206541.

36. Lateef, F.; Ruichek, Y. Survey on semantic segmentation using deep learning techniques. *Neurocomputing* **2019**, *338*, 321–348. [CrossRef]
37. Qin, Y.; Wu, Y.; Li, B.; Gao, S.; Liu, M.; Zhan, Y. Semantic segmentation of building roof in dense urban environment with deep convolutional neural network: A case study using GF2 VHR imagery in China. *Sensors* **2019**, *19*, 1164. [CrossRef]
38. He, Q.; Sun, X.; Yan, Z.; Fu, K. DABNet: Deformable contextual and boundary-weighted network for cloud detection in remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–16. [CrossRef]
39. Yang, J.; Guo, J.; Yue, H.; Liu, Z.; Hu, H.; Li, K. CDnet: CNN-based cloud detection for remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6195–6211. [CrossRef]
40. Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the Computer Vision—ECCV 2018: 15th European Conference, Munich, Germany, 8–14 September 2018; pp. 801–818.
41. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
42. Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; Torralba, A. Scene Parsing Through ADE20K Dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July; 2017; pp. 633–641.
43. He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; Girshick, R. Masked autoencoders are scalable vision learners. *arXiv* **2021**, arXiv:211106377.
44. Zhang, H.; Wu, C.; Zhang, Z.; Zhu, Y.; Lin, H.; Zhang, Z.; Sun, Y.; He, T.; Mueller, J.; Manmatha, R.; et al. ResNeSt: Split-Attention Networks. *arXiv* **2020**, arXiv:200408955.