A Geographic Information-driven Method and A New Large Scale Dataset for Remote Sensing Cloud/Snow Detection

Xi Wu[a,b,c], Zhenwei Shi[a,b,c,*], Zhengxia Zou[d]

[a]*Image Processing Center, School of Astronautics, Beihang University, Beijing 100191, PR China*
[b]*Beijing Key Laboratory of Digital Media, Beihang University, Beijing 100191, PR China*
[c]*State Key Laboratory of Virtual Reality Technology and Systems, School of Astronautics, Beihang University, Beijing 100191, PR China*
[d]*Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, MI 48109, USA*

Abstract

Geographic information such as the altitude, latitude, and longitude are common but fundamental meta-records in remote sensing image products. In this paper, it is shown that such a group of records provides important priors for cloud and snow detection in remote sensing imagery. The intuition comes from some common geographical knowledge, where many of them are important but are often overlooked. For example, it is generally known that snow is less likely to exist in low-latitude or low-altitude areas, and clouds in different geographic may have various visual appearances. Previous cloud and snow detection methods simply ignore the use of such information, and perform detection solely based on the image data (band reflectance). Due to the neglect of such priors, most of these methods are difficult to obtain satisfactory performance in complex scenarios (e.g., cloud-snow coexistence). In this paper, a novel neural network called "Geographic Information-driven Network (GeoInfoNet)" is proposed for cloud and snow detection. In addition to the use of the image data, the model integrates the geographic information at both training and detection phases. A "geographic information encoder"

*Corresponding author: Zhenwei Shi.
Email addresses: xiwu1000@buaa.edu.cn (Xi Wu), shizhenwei@buaa.edu.cn (Zhenwei Shi), zzhengxi@umich.edu (Zhengxia Zou)

is specially designed, which encodes the altitude, latitude, and longitude of imagery to a set of auxiliary maps and then feeds them to the detection network. The proposed network can be trained in an end-to-end fashion with dense robust features extracted and fused. A new dataset called "Levir_CS" for cloud and snow detection is built, which contains 4,168 Gaofen-1 satellite images and corresponding geographical records, and is over 20x larger than other datasets in this field. On "Levir_CS", experiments show that the method achieves 90.74% intersection over union of cloud and 78.26% intersection over union of snow. It outperforms other state of the art cloud and snow detection methods with a large margin. Feature visualizations also show that the method learns some important priors which is close to the common sense. The proposed dataset and the code of GeoInfoNet are available in `https://github.com/permanentCH5/GeoInfoNet`.

Keywords: Geographic information, cloud and snow detection, deep convolutional neural networks, remote sensing image.

1. Introduction

The fast development of remote sensing technology in the past decades has helped people better understand the earth. Optical remote sensing technology, as an important branch of the remote sensing family, is of great significance to many applications, such as target detection [1, 2, 3, 4], scene classification [5], etc. However, the imaging process of remote sensing images is often disturbed by clouds and snow. Previous literature shows that cloud covers on average more than half of the earth's surface every day [6, 7, 8, 9]. In some high latitude regions, the ground may be also covered by snow and ice all year round. On one hand, both of the above factors will greatly affect the processing and analysis of remote sensing imagery, where the cloud can be a form of occlusion[10, 11] and the snow might increase the reflectance sharply. On the other hand, environmental studies like climate study [12] and ecological change analysis [13, 14] require cloud/snow masks but manually labeling the images is usually time-consuming and expensive [15]. Automatic cloud and snow detection provides an efficient way of producing pixel-wise cloud/snow masks and thus forms the basis of many remote sensing applications.

Geographic information such as the altitude, longitude, and latitude are important meta-records in remote sensing imagery products. Such a group of

records provides auxiliary and even crucial information for image processing and analysis tasks. In cloud and snow detection, it also provides important priors. For example, it is generally known that snow is less likely to exist in low-latitude or low-altitude areas, and clouds in different geographic may have various visual appearances. Figure 1 shows some cloud and snow sample images. Each image covers around 40 thousand square kilometers, and it represents differently in different locations around the Earth. In recent years, many deep learning cloud detection and snow detection methods have been proposed. Despite the efforts made and the great improvements in this field, previous methods, even the state of the art ones, still have limitations. One of the most serious flaws of the methods is that they simply ignore the use of geographic information when performing detection. That is to say, these deep learning methods are designed solely based on the use of the image data (band reflectance), while ignoring other essential priors, such as altitude and locates. In complex scenarios such as when the cloud and snow both appear, these methods usually have difficulty generating accurate cloud and snow masks.

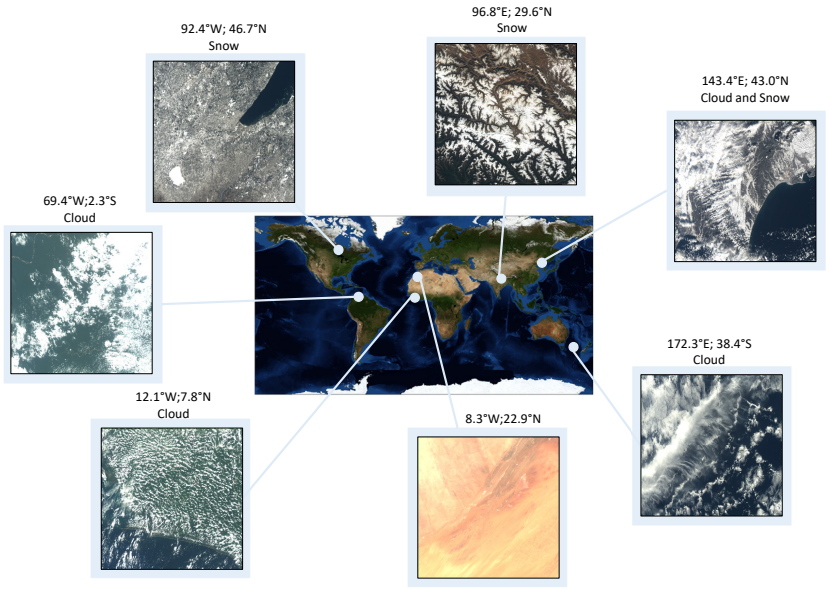

Figure 1: (Better viewed in color) Cloud and snow may represent great differences in different geographic environments. Base map credit: NASA Visible Earth.

In this paper, a novel deep learning based method is proposed for cloud

and snow detection. The method is called as Geographic Information-driven Neural Networks (GeoInfoNet). Different from the previous methods that simply focus on using image data (band reflectance) while ignoring geographic information, in the method, a "geographic information encoder" is designed, which encodes the altitude, latitude, and longitude of an image into a set of 2D maps. These maps are then integrated pixel-wisely to the detection networks and then train the whole detection model in an end-to-end fashion. It can be observed that the consistent improvement of the cloud and snow detection accuracy with the integration of the auxiliary information. The method outperforms other state of the art cloud and snow detection methods with a large margin. In addition to the new detection framework, a large dataset is also built for cloud and snow detection, which consists of 4,168 images of the Gaofen-1 satellite and is over 20 times larger than other datasets of this field. More importantly, the dataset contains the corresponding geographic information, including the longitude, latitude, and the high-resolution altitude map of each image. The contributions of this paper are summarized as follows:

1. Different from previous cloud and snow detection methods that are build based on band reflectance and simply ignore the geographic information of the imagery, a novel deep learning framework called "GeoInfoNet" is proposed which integrates the geographic information to the detection flow and learns the detection prior automatically. An encoder is designed to encode the auxiliary information such as altitude, longitude, and latitude into a set of 2D maps, which can be efficiently learned pixelwisely by the detection network in an end-to-end fashion.

2. Extensive studies on the feature visualizations are provided to show what prior knowledge the framework learns and how much the different parts contribute to the detection results.

3. A new dataset is built for cloud and snow detection, which is 20x larger than previous datasets of this task. More importantly, the geographic information along with each image is recorded in the dataset while such information is not included in previous datasets.

The following of this paper is organized as follows. In Section 2, related work to the method is introduced. In Section 3, the proposed method is introduced in detail. In Section 4, the details of the dataset Levir_CS are

given. In Section 5, extensive experiments on the method are conducted and the discussions are presented in Section 6. Finally, Section 7 concludes this paper.

2. Related Work

Efforts have been made for years to develop algorithms on automatic cloud and snow detection. Current methods mainly include 1) physical model based methods, 2) statistic model based methods, and 3) deep learning based methods. As for the discrimination between cloud and snow, generally, these methods are able to deal with it spectrally, spatially or temporally.

2.1. Physical Model Based Methods

The first line of the detection methods mainly focuses on the reflectance of a specific image band or the ratio between two bands. For cloud detection, the well-known method is Automatic Cloud Cover Assessment (ACCA) [16, 17], which is designed based on the 2nd-6th bands of the Landsat-7 ETM+ imagery. However, this method fails on detecting warm cirrus clouds and does not produce cloud shadow masks [18]. A method named Function of masks (Fmask) [18, 19] is therefore proposed and can be viewed as an extension of the ACCA. The Fmask takes more bands into consideration with more physical tests, such as the whiteness test, haze optimal transformation (HOT) test, and water test. In [20], a modified version of the Fmask called Mountainous Fmask (MFmask) is proposed for cloud detection in the mountainous area. In [21], 'Fmask4.0' improves cloud detection by analyzing the spectral variability probability.

There are also some other physical model based methods proposed recently [22, 23, 24, 25, 26, 27, 28]. In [22], similar to the ACCA, the information of single band, multiband, band ratio, and band difference are extracted to detect clouds on Landsat8, NPP-VIIRS and MODIS imagery. In [23], different indexes such as HOT, the relative difference (RD) and the shadow index (SI), are calculated for cloud and cloud shadow detection on Sentinel-2 imagery. In [24], cloud masks are generated by band reflectance relationships among visible and near-infrared bands of multi-temporal VENmS, FORMOSAT-2, Sentinel-2 and Landsat series imagery. Similarly, in [25, 26], multi-temporal information of spectral bands has been used in the cloud detection of HuanJing-1 satellite images. In [27], a modified ACCA algorithm is developed for discriminating the cloud from the clear background

by using the 2nd-4th bands of the GaoFen-1 WFV and HuanJing-1-CCD imagery. In [28], similar to the Fmask [18, 19], relationships between all bands of the GF1-WFV imagery are considered for cloud detection.

As for the snow detection, the most commonly used method is the Normalized Difference Snow Index (NDSI) [16, 17, 18, 24, 29, 30]. This method separates cloud and snow pixels by computing the ratio between the difference and the sum of the green band ($0.52\mu m - 0.60\mu m$) and the short-wave-infrared band ($1.55\mu m - 1.75\mu m$). The mechanics behind this method is that the snow is much more reflective in the green band than in the short-wave-infrared band [16, 17, 18, 24, 29, 30], therefore, a pixel with a higher NDSI indicates it is more likely to be covered by snow.

Despite the wide applications of NDSI, this method still has some limitations. In [19], the authors mentioned that NDSI values of the snow covered forest areas are much lower than the pure snow pixels. To overcome this problem, a modified Norwegian Linear Reflectance-to-Snow-Cover algorithm (NLR) [31] is proposed for generating better snow masks [19]. In [20], by using Digital Elevation Models (DEMs), the temperature-elevation relationship is established for better discrimination between the clouds and snow/ice pixels on the mountain area. In [14], by substituting the information of green band to the near-infrared band ($0.845\mu m - 0.885\mu m$), the normalized difference forest snow index (NDFSI) is proposed for better detecting snow pixels in the forest area.

By comparing to pre-defined thresholds, the above physical model based methods can efficiently generate cloud and snow masks of the input imagery. The advantage of these methods is that they do not require any pixel-wise labels or any training process. However, these methods may also heavily rely on the reflectance of the imagery band and the pre-defined thresholds, which lacks flexibility and robustness under complex scenarios. More importantly, these methods may also fail in situations where some spectral bands, especially some short-wave-infrared bands, are not provided in the imagery [28].

2.2. Statistic Model Based Methods

To overcome the above problems, some statistic model based methods are proposed which aim to design more representative image features, along with the use of machine learning techniques. Different from those physical model based methods that only consider band features, the statistical methods use also consider spatial image features such as edge and texture, which further explore the information behind the imagery. The statistic model

based methods [32, 33] usually deal with the cloud/snow detection under a classification paradigm. Some well-known classifiers such as support vector machine (SVM) [34] and random forest [35] are commonly used in this task.

The most frequent used image features for cloud and snow detection include the following types: brightness features, texture features, and local statistical features. Brightness features, i.e., the reflectance of the image bands, are the most commonly used features in statistical methods [32, 7]. As cloud and snow pixels are usually with high reflectance, brightness features are often the first to be considered to separate the cloud/snow apart from the background. Besides, there are also methods to convert the band values to other color spaces to enrich the brightness features. For example, in [36], the RGB image bands are converted to the Hue-Saturation-Intensity (HSI) color space. In [33], the band-differences, band-ratios, and other generalized indexes, are also computed. For the local statistical features and texture features [7, 32, 37, 38, 36], these features are usually computed by sliding windows across the whole image. By setting different window sizes, features can be extracted in multiple scales. In [7, 36], the mean pixel value and the variance within the window are used as the local statistical feature for cloud detection. Besides, the gradient features, the Gabor filter features [39, 40, 41], and the gray level co-occurrence matrix (GLCM) [42, 43] are also commonly used for cloud or snow detection [32, 7, 37, 36, 44, 45] .

2.3. Deep Learning Based Methods

In recent years, deep neural networks [46, 47, 48] have made great breakthroughs in many computer vision tasks such as image classification, object detection, etc. The deep neural networks also have greatly promoted the research of cloud and snow detection in the remote sensing field.

Some early attempts of this group of methods consider the cloud and snow detection task as a patch-by-patch image classification process (into three categories: "cloud", "snow" and "background") [49, 8, 50, 51, 52]. In [52], cloud detection is conducted on 33×33 image patches with a 2-layer convolutional neural network. In [50, 51, 49] , a pixel cluster method called simple linear iterative cluster (SLIC) [53] is first used to segment the image into a set of super-pixels, and then neural networks are used to classify each of these super-pixels. In [8], a modified SLIC method is proposed and a modified AlexNet [54] is applied to further predict whether these super-pixels are covered by thick or thin cloud. The above methods take advantage of the power of deep learning neural networks in image classification and

obtain higher accuracy than the statistic model based methods. However, the patch-based detection methods also have some limitations. The first limitation is that it fails when the image patch contains pixels from multiple classes and the second one is that the model only perceives locally and ignores the information from the surrounding patches.

To overcome these limitations, the fully convolutional networks (FCN-s) [55] are recently introduced for cloud and snow detection [15, 56, 57, 58, 59, 9, 60, 61, 62, 63]. This group of methods frames the cloud and snow detection as a pixel-wise semantic segmentation process. In [15], a VGG16-based [46] fully convolutional network is applied for cloud and snow detection. In [56, 57, 58], UNet [64] architecture is used to study on cloud and cloud shadow segmentation. In [59], a modified residual network with pyramid pooling modules is used for cloud and cloud shadow detection. In [9, 62], feature fusion of different layers are introduced to improve the details of the cloud detection result. In [60], multiple bands of the Landsat-8 imagery are used as the network input, where other methods usually take the RGB bands as their inputs. In [61], some network units are specifically designed for cloud detection, including the context exploitation, score map resolution preservation, and boundary refinement. In [63], cloud detection is processed in a deep matting framework where the images of cloud reflectance, attention mask and opacity can all be obtained.

Although these deep learning methods greatly improve the detection accuracy over traditional detection methods, the use of geographic information is not explored yet. This is one of the reasons the method is designed in this paper.

2.4. Discrimination between Cloud and Snow

In the process of cloud and snow detection, there are usually difficult cases where the remote sensing image contains both cloud and snow. Therefore, the discrimination between cloud and snow is very significant. Generally, the above-summarized cloud and snow detection methods can also be divided into three classes in another dimension: spectral methods, spatial methods and temporal methods.

In detail, many physical methods[16, 17, 18, 19, 20, 21, 28, 29, 30, 14] tend to separate cloud and snow in the spectral domain. These spectral methods analyze the relationships between the spectral bands of the remote sensing images to classify cloud and snow. Based on the spectral band relationships, many spectral filters can be established to choose cloud and snow pixels.

Although the filters of obtaining cloud masks are generally different among these methods, the approaches to generate snow masks are almost the same. Specifically, the core of the spectral snow detection methods is NDSI, which is an index utilizing the green band and the short-wave-infrared band. With the previously defined thresholds, these spectral methods are able to produce cloud and snow masks of the input remote sensing images.

The second part is spatial methods. These methods are generally the above mentioned statistical model based methods [32, 33, 7, 36, 37, 38, 44, 45] and deep learning based methods [49, 8, 50, 51, 52, 15, 56, 57, 58, 59, 9, 60, 61, 62, 63]. The spatial methods try to use the spatial information of the image by extracting mannual designed image features or deep learning network features. After image features are obtained, there are usually pre-trained classifiers or classifying network layers to discriminate cloud and snow according to these spatial features. To make the classifiers robust, enough training data are needed in these spatial methods. Therefore, spatial methods can also be viewed as data-driven methods.

A few methods try to discriminate cloud and snow by using multi-temporal information of the input images [24, 25, 26]. Among this type of methods, multi-temporal tests are introduced by using the blue band to detect the cloud pixels. The multi-temporal tests can utilize the time-series images and are very efficient to obtain cloud pixels. For snow detection, NDSI [24] or whiteness [25, 26] tests are used to extract snow pixels, which is similar to spectral methods. As these temporal methods need image series, they are not proper in the situations where the input is limited to only one remote sensing image scene.

It should be noted that the discrimination between cloud and snow is a challenging task mainly due to two aspects of reasons. One is that cloud and snow represent a high visual similarity in remote sensing images, especially in visual bands. For example, both of them have high reflectance in most bands. Also, they all have irregular shapes and appear in different scales in remote sensing images. Hence, many spectral tests can not distinguish cloud and snow very well and are heavily relied on pre-defined thresholds. These methods such as [28, 20, 21] may produce wrong-detection results that misclassify snow as cloud. The other reason is the imbalanced data distribution. Generally, it is more likely to see clouds in remote sensing images. On our planet, more than half of the surface is covered by clouds every day [6, 7, 8, 9], while snow is seldom witnessed in some regions, such as low-latitude or low-altitude areas. Therefore, the spatial methods will more

pay more attention to clouds rather than snow mainly because there are much more image data with clouds than snow. As a result, the area of snow may be more likely to be detected as clouds by using the spatial methods as the visual results shown in [61, 63]. Above all, the spectral and the spatial information may not be enough in the discrimination of cloud and snow, and using geographic information can be one potential solution.

Despite there are many challenges in the discrimination between cloud and snow, there are no public datasets for snow detection, which may limit the pace of the related research. The proposed Levir_CS dataset in this paper may contribute to this research field.

3. Methodology

In this section, a detailed description of the detection method is given and how the geographic information is encoded and integrated to the network.

3.1. An Overview of the GeoInfoNet

Figure 2 shows an overview of the method. The proposed GeoInfoNet is an end-to-end network that utilizes both the input image and a set of auxiliary maps. The auxiliary maps are produced by the Geographic Information Encoder, which will be introduced in Section 3.2. In GeoInfoNet, the network structure in DenseNet [48] is followed as the backbone network and extract multi-scale dense features from the input image and auxiliary maps separately. Then these features extracted from the two branches will be merged and used to produce the final cloud and snow masks. The two basic modules in the method, i.e., the "Dense Feature Extraction" and the "Dual Feature Concatenation", which form "Feature Extraction Networks", will be described in detail in Section 3.3.1 and Section 3.3.2.

3.2. Geographic Information Encoder

A geographic information encoder is designed to encode three types of meta-records along with the imagery, i.e., the longitude, latitude, and altitude, into a set of auxiliary maps. This module can be viewed as a preprocessing module of the proposed GeoInfoNet. These maps are generated to be the ones with the same spatial size of the input image but may have a different number of channels. Figure 3 shows the processing pipeline of the geographic information encoder.

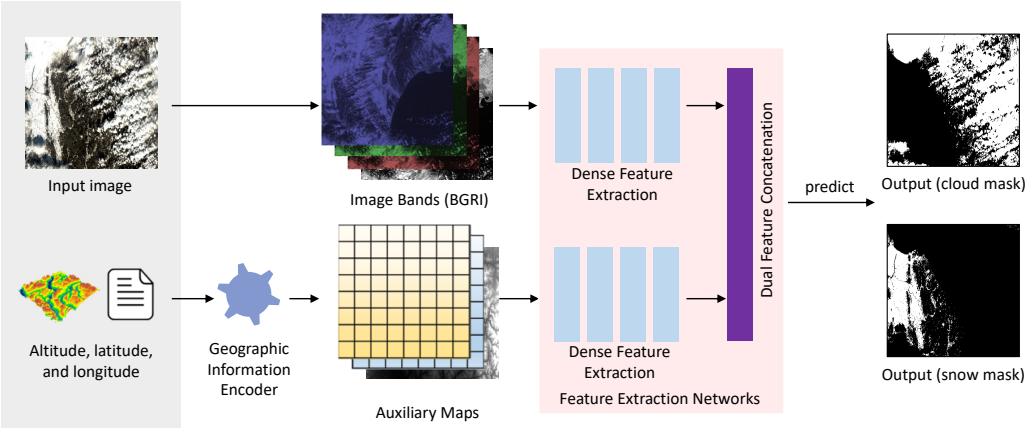

Figure 2: Overview of the proposed method. A new network called GeoInfoNet is proposed that makes use of both the image data and the auxiliary geographic information for cloud and snow detection. A Geographic Information Encoder is designed to encode this piece of information into a set of auxiliary maps. Features of both network-branches are extracted by "Feature Extraction Networks", which includes "Dense Feature Extraction" module and "Dual Feature Concatenation" module. The former module can extract representative features of each branch, while the latter module is designed to produce the refined feature representation, which is further used for generating cloud and snow masks.

Given a remote sensing image with the size of $h \times w$, firstly the longitude and latitude are recorded on its the top-left corner and bottom-right corner. Then the longitude map A_{Long} and the latitude map A_{Lat} are generated through an Affine transformation model [65, 66]. For a certain pixel in row y $(0 \leq y < h)$ and column x $(0 \leq x < w)$, the corresponding longitude $A_{\text{Long}}(y, x)$ and latitude $A_{\text{Lat}}(y, x)$ can be calculated as followings:

$$\begin{aligned} A_{\text{Long}}(y, x) &= A_{\text{Long}}(0, 0) + y \times r_{1,1} + x \times r_{1,2} \\ A_{\text{Lat}}(y, x) &= A_{\text{Lat}}(0, 0) + y \times r_{2,1} + x \times r_{2,2} \end{aligned} \tag{1}$$

where $A_{\text{Long}}(0, 0)$ and $A_{\text{Lat}}(0, 0)$ are the longitude and latitude value of the top-left image corner. $r_{1,1}$, $r_{1,2}$, $r_{2,1}$, and $r_{2,2}$ are the longitude/latitude resolution units on x and y directions, which can be obtained from the metafile of the imagery product or can be estimated from the coordinates of the four image corners and the center point.

In addition to the longitude and latitude, the altitude of the image is also encoded to another auxiliary map A_{Alt}. Given an image along with

its corresponding longitude/latitude information, the altitude map A_{Alt} can be generated by pixel-wisely wrapping the global Digital Elevation Models (DEMs) to the projection coordinates of this image. For most optical remote sensing imagery products, the image altitude information is not included in the metafile. In the paper, the used DEMs are created based on the data collected by the 2000 Shuttle Radar Topography Mission (SRTM) and the resolution is 3 arc seconds (spatial resolution: 90 meters). The data can be download from the following URL: `http://viewfinderpanoramas.org/dem3.html`.

The final encoded auxiliary maps A for each input image can be represented as a concatenation of the above three maps in the channel dimension as follows,

$$A = \mathrm{concat}(A_{\mathrm{Alt}}, A_{\mathrm{Long}}, A_{\mathrm{Lat}}), \tag{2}$$

where the dimension of the concated map A is $(h, w, 3)$.

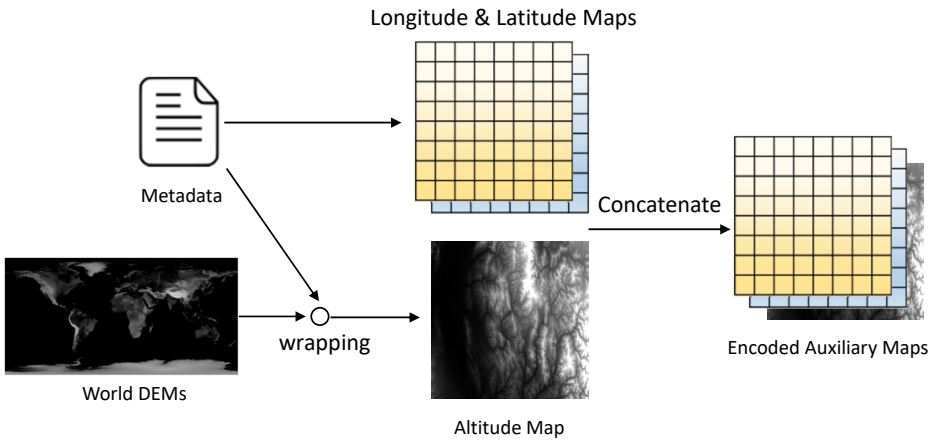

Figure 3: The processing pipeline of the Geographic Information Encoder. For an input image, the Longitude map A_{Long} and the Latitude map A_{Lat} are generated from the metadata based on the equation (1). The altitude map A_{Alt} of the input image is also generated by pixelwisely wrapping the world DEMs to the image projection coordinates. Finally the three maps are concatenated to produce the final encoded auxiliary maps.

3.3. Feature Extraction Networks

3.3.1. Dense Feature Extraction

In deep learning based cloud and snow detection methods, learning robust feature representations is crucial for the detection task. Since improving

backbone of the networks is not the focus of the paper, a well-known backbone called "DenseNet" [48] is simply used, which achieves state of the art results in a variety of tasks, as the backbone network to extract high quality features from the input data arrays.

The DenseNet consists multiple dense blocks. In each block, the feature from all preceding convolutional layers are concated together. Formally, the feature maps M_{l+1} of the $(l+1)^{th}$ layer can be calculated as follows,

$$M_{l+1} = \sigma(\text{concat}(M_l, M_{l-1}, ..., M_1)), \tag{3}$$

where $\sigma(\cdot)$ represents a non-linear transformation on the features. Figure 4 shows the process of the calculation of M_{l+1}.

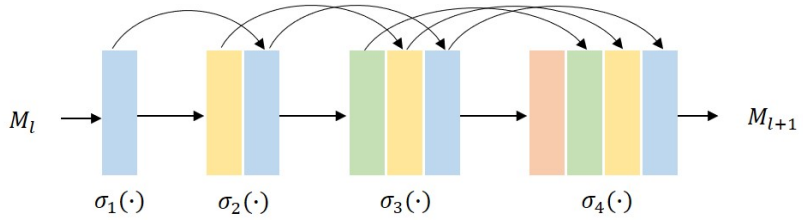

Figure 4: An illustration of a 4-layer dense block. Each convolution layer takes all preceding feature-maps as input.

From Eq. 3 , feature maps $M_1, M_2, ..., M_l$ are preserved in the process of calculating M_{l+1}. Considering the concatenation of feature maps is space consuming, the filter number of each convolutional layers u is set to a small number, say, $u = 32$, compared to that in a standard convolutional network, e.g. VGG [46] and ResNets [47]. In this case, the number of input feature maps in the $(l+1)^{th}$ layer will be $u_1 + 32 \times l$, where u_1 is the number of feature maps in the first layer and 32 is the filters in each layer, which also can be considered as the increasing rate. A small increasing rate not only regulates the number of features, which makes the feature extracting networks go relatively deep, but also equalizes the number of features added in each layer since the newly added information should be viewed as the same importance.

The non-linear transformation $\sigma(\cdot)$ in the networks consists of two types of operations, the normalization operation (batch normalization [67]), and the non-linear activation operation (rectified linear unit function [68]). It should be noted that 1×1 convolution can be placed before 3×3 ones, which

Layers	Layer Settings
Conv_0	7x7 conv, stride=1, #out_channels=64
Pool_0	3x3 max_pool, stride=2
Dense_block_1	6 bottlenecks, #out_channels=256
Transition_1	1x1 conv, 2x2 avg_pooling, stride=2, #out_channels=128
Dense_block_2	12 bottlenecks, #out_channels=512
Transition_2	1x1 conv, 2x2 avg_pool, stride=2, #out_channels=256
Dense_block_3	32 bottlenecks, #out_channels=1280
Transition_3	1x1 conv, 2x2 avg_pool, stride=2, #out_channels=640
Dense_block_4	32 bottlenecks, #out_channels=1664

Table 1: The configuration of the dense feature extraction module.

seems like a "bottleneck", and the settings are able to improve computational efficiency in [47, 48]. Therefore, following the idea of the "Bottleneck design", $\sigma(\cdot)$ is designed in the form of BN-ReLU-Conv(1×1)-BN-ReLU-Conv(3×3), where each Conv (1×1) outputs $4u$ feature maps.

In addition to the above dense connection module, some downsampling modules are also designed in the networks to reduce the size of feature maps spatially and increase the computational efficiency [9]. These modules are designed as transition blocks by following the configuration of BN-ReLU-Conv(1×1)-Pool(average,2×2), and are placed between dense blocks. The 1×1 convolution here outputs a half number of the input feature maps.

In [48], several different types of DenseNet configurations have been proposed, including DenseNet121, DenseNet169, DenseNet201 and DenseNet264. The number "X" in "DenseNetX" represents the number of convolution layers used in the classification network. In Dense Feature Extraction module, the configuration of DenseNet169 is adopted because of the balance of computation efficiency and the cost of GPU memory. The module takes in an input array which is first processed through an initial convolution layer ("Conv_0") and an initial pooling layer ("Pool_0"), then through four dense blocks and three transition blocks accordingly. Different from the settings in [48], the stride of "Conv_0" is set to 1 and remove the last classification layer for the tasks of cloud and snow detection. The configuration details of the Dense Feature Extraction module is listed in Table 1.

3.3.2. Dual Feature Concatenation

In the above feature extraction process, as the layers go deeper, the number of output feature maps becomes larger. The spatial resolution of the final output is down-sampled to 16x compared to that of the input, as shown in Table 1. To produce high-resolution cloud and snow masks, it is essential to increase the feature resolution by taking features. This can be done by merging the features from different blocks and generate fine-grained feature representations. The Dense Feature Concatenation module thus is designed according to this purpose. In this module, the initial features from both of the Blue-Green-Red-Infrared (BGRI) input image and the encoded auxiliary maps are used.

As shown in Figure 5, for either of the two branches of the networks (i.e., input image branch and the auxiliary maps branch), the spatial features from each feature block are firstly upsampled to the size of the input image by using bilinear interpolations. Then, the upsampled features are concatenated all together along their channel dimension. Before the concatenation, we also use 1x1 convolution to adjust the channel dimension of the features from each block so that they will have the same number of channels. The intuition behind this operation is that it is assumed that for all the blocks in the networks, the features should be viewed with the same significance in the cloud and snow detection tasks. The final concatenated features M from all blocks the two branches can be represented as follows,

$$M = \text{concat}(M_{\text{img},0}, ..., M_{\text{img},4}, M_{\text{aux},0}, ..., M_{\text{aux},4}), \tag{4}$$

where the subscripts "img" and "aux" refer to the features from the BGRI image branch and the auxiliary information branch, respectively. The subscripts "0∼4" refer to the upsampled features from the "Conv_0", "Dense_block_1", "Dense_block_2", "Dense_block_3", and "Dense_block_4" of the Dense Feature Extraction module.

3.3.3. Loss Settings

In the proposed GeoInfoNet, a prediction layer (a convolutional layer with 1×1 filters) is used to produce the pixel-wise score maps of different classes: background S_1, cloud S_2 and snow S_3. The output score maps are normalized by using a softmax function and convert the pixel scores $(-\infty, \infty)$ to probabilities [0,1]. The probability map P_t of the each class $t = \{1, 2, 3\}$

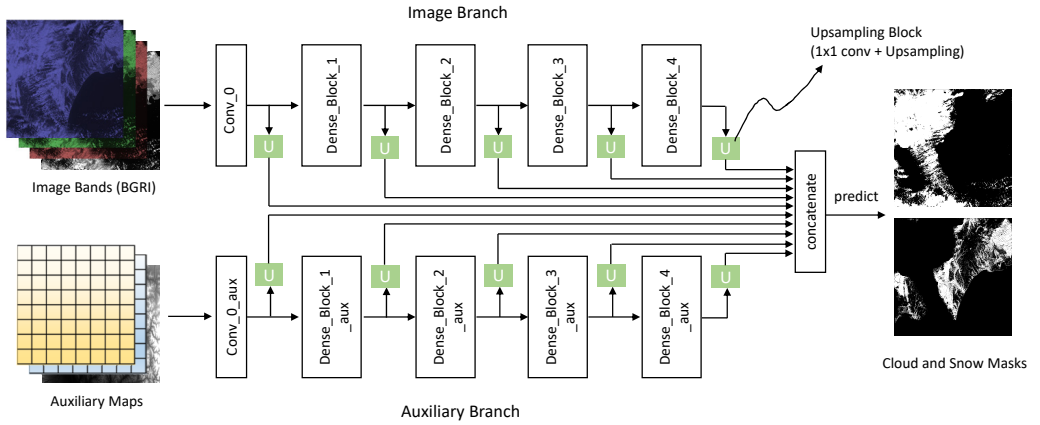

Figure 5: Details of the Dense Feature Extraction Module. The method takes in two types of the inputs simultaneously, i.e., the original RGBI bands of the input remote sensing image and the auxiliary maps that are encoded by the Geographic Information Encoder. In both of the two branches, the dense features are extracted by the "Convo_0" layer and the following four dense blocks. The features from different blocks are upsampled to the same size and adjust the number of their channels by 1x1 convolutions, and then concatenate all these features along their channel dimension. Finally, a prediction layer is used to produce the pixel-wise score maps of the cloud and snow.

can be expressed as follows,

$$P_t = \frac{\exp\left(S_t\right)}{\sum_{m=1}^{3} \exp\left(S_m\right)}. \tag{5}$$

As the detection of cloud and snow is essentially a pixel-wise classification process, the networks are trained by using a standard pixel-wise classification loss (a.k.a., the cross-entropy loss). Suppose $y_m \in \{0, 1\}$ represents the ground truth label of the class m. The loss function of each pixel is expressed as follows:

$$L = -\sum_{m=1}^{3} y_m \log(P_m). \tag{6}$$

Finally the average loss across all pixel from all images in the training set is computed as the final loss function.

4. Levir_CS: A New Large Scale Dataset for Cloud and Snow Detection

Dataset	Source	#Scenes	Snow	Geo Info
L7_Irish [69]	Landsat-7 ETM+	166	×	×
L8_Biome [70]	Landsat-8 OLI/TIRS	92	×	×
GF1_WHU [28]	Gaofen-1 WFV	108	×	×
Levir_CS (ours)	Gaofen-1 WFV	4,168	✓	✓

Table 2: A comparison between our dataset and the other public cloud detection datasets.

A large scale dataset called "Levir_CS" is built, where 'C' is for cloud and 'S' is for snow, respectively. As the name of the authors' laboratory is "LEarning, VIsion and Remote sensing laboratory", similar to [4], the name of this dataset is started with "Levir". Although there are already some public datasets on this topic released in the past, they are relatively small and do not contain geographic information. Besides, there are no previous public datasets for snow detection. Table 2 shows a comparison between our dataset and the other public cloud detection datasets[69, 70, 28]. Compared to other datasets listed in Table 2, the number of scenes in LEVIR_CS is over 20× larger than the other datasets, therefore, the proposed dataset is called "Large Scale". The proposed Levir_CS dataset is available at https://github.com/permanentCH5/GeoInfoNet/.

Gaofen-1 satellite (GF-1) is running at sun synchronous orbit, where the angle is 98.0506° and the average orbit height is 645km. The revisiting time is 4 days. The descending node is 10:30 am. The radiometric resolution GF-1 Wide Field of View sensor (GF-1 WFV) is 10 bit. A GF-1 WFV scene, each 211km wide by 192km long, has an Instantaneous Field Of View (IFOV) of 16 meters × 16 meters in all four bands. The spectral range is 450 nm to 890 nm. In detail, the spectral range of these bands are 450nm - 520nm (Blue Band or Band 1), 520nm - 590nm (Green Band or Band 2), 630nm - 690nm (Red Band or Band 3), 770nm - 890nm (Near Infra-Red Band or Band 4), respectively.

Our proposed Levir_CS consists of 4,168 GF-1 WFV scenes in total. These scenes are randomly divided into two sets, a training set with 3,068 scenes and a testing set with 1,100 scenes. The scenes in the dataset have a global distribution, as shown in Figure 6. They cover different types of

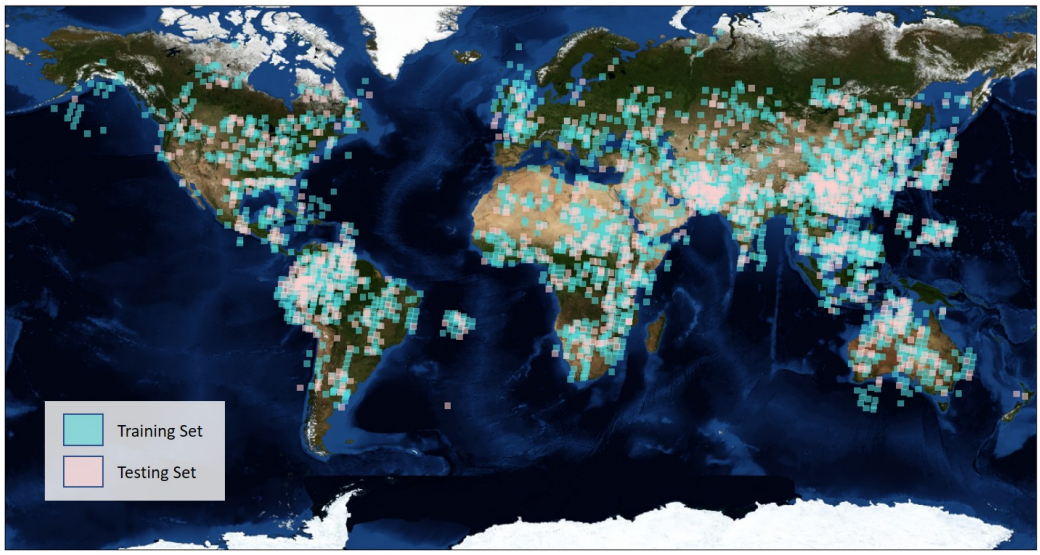

Figure 6: The global distribution of the images in the Levir_CS dataset. Levir_CS consists of 4,168 Gaofen-1 Wide Field of View (GF-1 WFV) images. All images are obtained from the China Centre for Resources Satellite Data and Application (CRESDA) `http://www.cresda.com/`. Base map credit: NASA Visible Earth.

ground features, such as plain, plateau, water, desert, ice, etc. There are also combinations of the above mentioned ground feature types. Figure 7 presents some sample scenes. Besides, as these scenes are in a global distribution, therefore, these scenes may contain different types of climate conditions, such as desert climate (see Figure 7(c)) or sea climate (see Figure 7(b,e)), which may help the related researches similar to [12]. All the scenes were acquired from May 2013 to February 2019 and were downloaded from `http://www.cresda.com/`.

In the proposed LEVIR_CS dataset, for each scene, the level-1A product data with the process of radiation calibration is used and the current data is not produced with systematic geometric correction. This is because in many practical cases, cloud and snow detection is required to be performed in this product level to save the time of geometric correction or for fast browsing. Dataset users are able to obtain according to the provided file of rational polynomial coefficients (RPC) to conduct systematic geometric correction if it is needed. Furthurmore, to reduce the processing time of each scene and to accelerate the learning process of the global information, similar to [71], the images in LEVIR_CS dataset are 10x downsampled. For each scene in

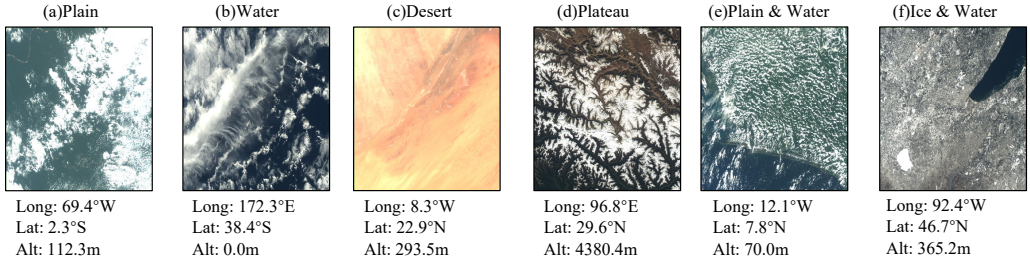

Figure 7: Some sample scenes with different types of ground features in the proposed Levir_CS dataset. On the top of each scene, the type of the ground feature is given. On the bottom of each scene, the longitude and latitude of the central point and the mean altitude of the image is presented.

LEVIR_CS dataset, the image size is 1320×1200 and the spatial resolution is $160m$. All the four bands are used. Therefore, the resolution of DEM ($90m$) is high enough in the altitude map generation. Therefore, SRTM data are chosen as the source of DEM.

In the proposed LEVIR_CS dataset, for each scene, the georeferenced multi-spectral image, the digital elevation model image and the corresponding ground truth image are all provided. The cartographic projection system used in the dataset is the World Geodetic System (WGS) and the latest version (WGS 84) is used. Through this cartographic projection system, for each scene, all the images can be registered through the geographic information. Therefore, climatic conditions do not relate to the generation of georeferenced images. For the generation of digital elevation model image, the average producing time is 45.62s per scene.

For all the images in the dataset, their pixel-wise label masks are manually labeled into three categories: "background" (labeled as 0), "cloud" (labeled as 127) and "snow" (labeled as 255). Similar to [28], the labeling process is finished in Adobe Photoshop. Blue, green and red bands of the original images are combined to compose a RGB image for manually labeling. To increase the labeling efficiency, similar to [72], a pre-segmentation is firstly performed by manually setting thresholds as the traditional physical methods such as [27] indicates. Then, a rough pixel classification on these pre-segmentation region is conducted. The boundary of cloud or snow area is usually fuzzy. Like the previous research [28, 62], these regions of the image are carefully labeled by using the brush tool (less than 10 pixels) or the lasso tool. For the thin cloud area, if the ground cover is invisible, then it

is labeled as clouds. For the shadow area, as it is very dark and the region is invisible, it is labed as the background. When labeling difficult area, the magnifying glass tool is used (more than 200% local area enlarged), which helps the labeling man to identify the exact class of the pixels. Cloud shadow detection is not the focus of this paper, therefore, this class is not labeled.

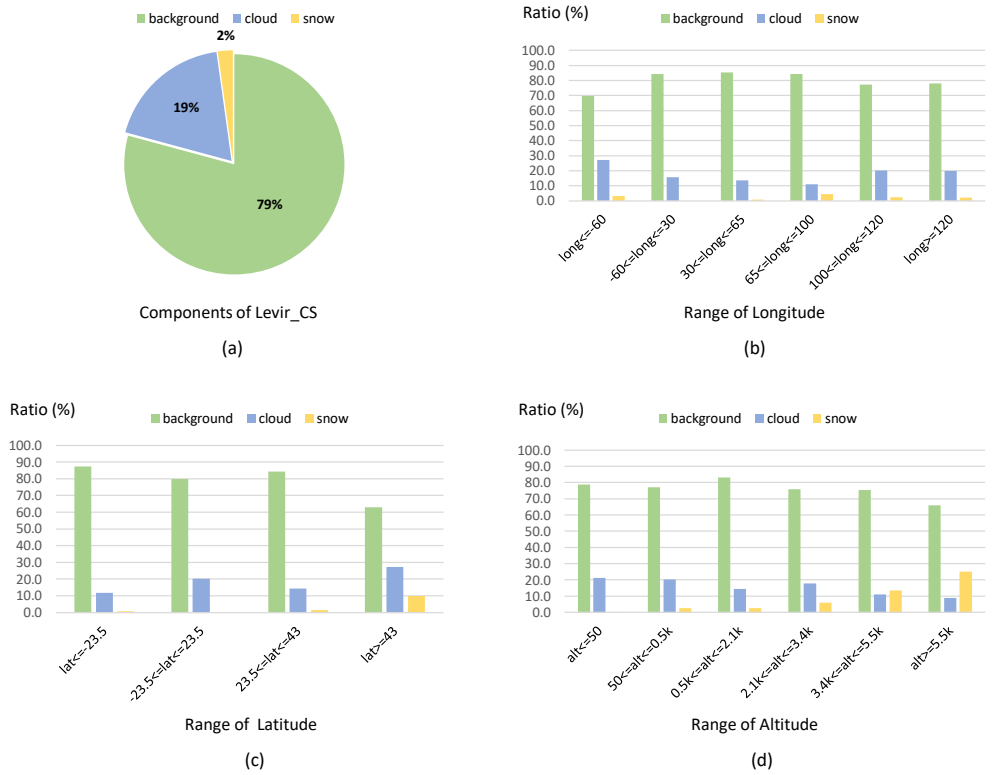

Figure 8: Statistics of the Levir_CS dataset from different views. (a) The pixel population of the three categories: the background occupies the most (79.2%), while the snow occupies the least (2.2%). The cloud pixels occupy 18.6% of the whole pixel population. (b)-(d) Label components from longitude, latitude and altitude views, respectively. It can be seen that the distributions of the three categories are very different in different areas.

Figure 8 shows the statistics of the Levir_CS dataset. The distribution of the label components is calculated from different views. As shown in Figure 8 (a), in Levir_CS, the background pixels occupy the most population (79.2%) while the snow occupies the least (2.2%). The cloud pixels occupy 18.6% of the whole pixel population. Figure 8 (b)-(d) show the label components from

the longitude, latitude and altitude views, respectively. From these figures, the following observations can be summarized:

- The pixel population of the three categories is very different in different locations.
- Clouds are common in different geolocations. For example, in North America, clouds appear in different kinds and forms[22].
- From the view of the longitude, it can be seen that the snow may be less likely to appear in the range of $-60^\circ \leq long \leq 30^\circ$ (Atlantic Ocean) (see Figure 7(c,e) for examples), while in the area of $65^\circ \leq long \leq 100^\circ$, it is easier to find snow cover (see Figure 7(d) for an example).
- From the view of the latitude, it can be seen that most of the snow appears in the high latitudes ($lat \geq 43^\circ$) (see Figure 7(f) for an example). In the United States, the number of snow days is higher in the high latitude regions, according to [73]. Besides, at high latitudes in the polar region, snow and ice does not melt in some seasons[29]. There is almost no snow covers the Equatorial regions ($-23.5^\circ \leq lat \leq 23.5^\circ$)(see Figure 7(a,c,e) for examples).
- From the view of the altitude, it can be seen that the cloud percentage is higher in the area where the altitude is less than 500 meters (see Figure 7(a,b,e) for examples) and the snow percentage gradually increases as the altitude increases (see Figure 7(d,f) for examples). Usually, the high altitude area is mountainous area, and snow cover is regularly changed in seasons here [14]. For the area with an altitude higher than 3400 meters, it is even easier to find snow than to find clouds (see Figure 7(d) for an example).

From the above statistics, it can be seen that using geographic information in cloud and snow detection is of importance.

5. Experimental Results and Analysis

In this section, extensively evaluation are made on the proposed method and compare it with other state of the art ones. First the implementation details are introduced and how the experiments are set up are described. Then, the controlled experiments are conducted on multiple aspects of the method.

Thirdly, qualitative and quantitative comparisons with other methods on Levir_CS are made. Finally, the transferability of the proposed GeoInfoNet by evaluating on other sensor data is tested on L8_Biome [70].

5.1. Experiment Setup and Implementation Details

In this paper, all the deep learning models tested are implemented with PyTorch 1.0 on Ubuntu 16.04 with an NVIDIA Geforce GTX 1080Ti GPU card. The networks are trained by using the stochastic gradient descent method with an initial learning rate of 0.001. The learning rate decay policy is set to "poly" as [9] did, and the power parameter is set to 0.9. The number of iteration, the l_2 weight decay, and the momentum are set to 2×10^5, 0.0001, and 0.9, respectively.

All backbone feature extraction models (including the Dense Feature Extraction module, VGG16 [46], ResNet101 [47], and DenseNet169 [48]) are pretrained on the Imagenet Dataset [74]. The weights of the convolutional layers in the other components of the networks are initialized by the "msra" method [75]. The number of feature maps provided by each additional convolution layer in the method is set to 64.

As informed in the above Section 4, the size of each image is 1320×1200 in both the training set and the test set. For the limitation of the GPU card memory, the training batch size is set to 4, and all the inputs are randomly cropped to 240 $\times$ 240 in the training phase. As for the testing phase, the input is croped to 600 $\times$ 600 patches for evaluation and then these patches are combined together.

In the training phase, to increase the diversity of the images, data augmentation is performed by randomly rotating the inputs for 0°, 90°, 180°, 270°. All the input bands are normalized to [0, 1]. In detail, for image bands, as the input images are 10-bit images, they are all divided by 1023. For the longitude bands, they are divided by 360 after added by 180. For the latitude bands, they are divided by 180 after added by 90. Finally, for the altitude bands, they are divided by 10000.

To evaluate the performance of different methods, three types of benchmark metrics are used in the experiments: *F1-Score* ($F1$), *Intersection-over-Union* (IoU), and *False Alarm Ratio* (FAR). These metrics are all widely used in cloud detection tasks [76, 9, 15, 59]. In detail, for cloud or snow, $F1$

is calculated as Eq.7 shows.

$$F1 = \frac{2p \cdot r}{p + r}, \tag{7}$$

where p and r are calculated as Eq.8 and Eq.9,

$$p = \frac{N_{correct}}{N_{correct} + N_{false-alarm}}, \tag{8}$$

$$r = \frac{N_{correct}}{N_{ground-truth}}, \tag{9}$$

where $N_{correct}$ is the number of pixels of correct detection, $N_{false-alarm}$ is the number of pixels of false alarms and $N_{ground-truth}$ is the number of pixels of the certain type in the groundtruth images, respectively. Similarly, IoU of cloud or snow is calculated as Eq.10 displays,

$$IoU = \frac{N_{correct}}{N_{ground-truth} + N_{false-alarm}}. \tag{10}$$

FAR is a type of benchmark to show the performance of all classes of detection (including cloud, snow and background), and it can be calculated as,

$$FAR = \frac{N_{wrong}}{N_{all}}, \tag{11}$$

where N_{wrong} is the number of pixels of wrong detection and N_{all} is the number of all pixels.

For the $F1$ and the IoU, a higher score indicates better and the results on the cloud and the snow are recorded accordingly. In the following statistics, these two figures are differently recorded in different classes. For the cloud, $F1\_c$ and IoU_c are used to represent the capacity of cloud detection, while for the snow, $F1\_s$ and IoU_s are used instead. For the FAR, a low score indicates better and the result on the whole test set of the Levir_CS is recorded.

Inference time is also evaluated per scene (image size: 1320×1200). In the testing phase, if the altitude map is used, it is directly loaded from the previous generated maps. This strategy is also used in the training phase. Therefore, it should be noted that the inference time does not include the time to create the corresponding altitude map. The time cost of making the

Network Structure	$F1\_c$	$F1\_s$	IoU_c	IoU_s	FAR	T
Only Image	94.37	83.10	89.34	71.08	2.58	2.24s
One Way	94.62	83.94	89.79	72.32	2.44	2.69s
Simple Concatenation	94.71	85.74	89.95	75.03	2.40	3.37s
Fusion	95.00	86.70	89.62	76.52	2.28	3.55s
Dual Feature Concatenation	**95.15**	**87.80**	**90.75**	**78.26**	**2.20**	3.45s

Table 3: Quantitative results of four possible network configurations in the method. The result of F1, IoU, and FAR are recorded (%). The subscript "c" and "s" refer to the class of "cloud" and "snow".

altitude map has been introduced in the above Section 4 (45.62s per scene) or other mentioned.

5.2. Controlled Experiments

In this subsection, three types of controlled experiments are conducted, which focus on the verification of different technical components of the method: the network structure, auxiliary components design, and the selection of backbone feature extractor. All these studies aim to find a reasonable design of the proposed method from different views.

5.2.1. Network Structure Design

It should be noticed that in addition to the network used in Figure 5, there are also other possible structures can be chosen. These chooses are all suitable for the cloud and snow detection tasks but may have different accuracy performance. Figure 9 presents four alternative choices on network configurations for the method.

Choice 1. In Figure 9 (a), a "Only Image" structure is adopted in the cloud and snow detection tasks. In this structure, only image bands are processed. The cloud and snow masks are predicted by using the concatenated dense features from only the information of the image. This structure can be viewed as a baseline network structure since many strategies can be applied in this structure to improve the performance.

Choice 2. In Figure 9 (b), a "One Way" structure is designed in a straightforward way, which concatenates all the bands together before extracting dense features. The image information and the auxiliary information are thus processed together in only one network branch.

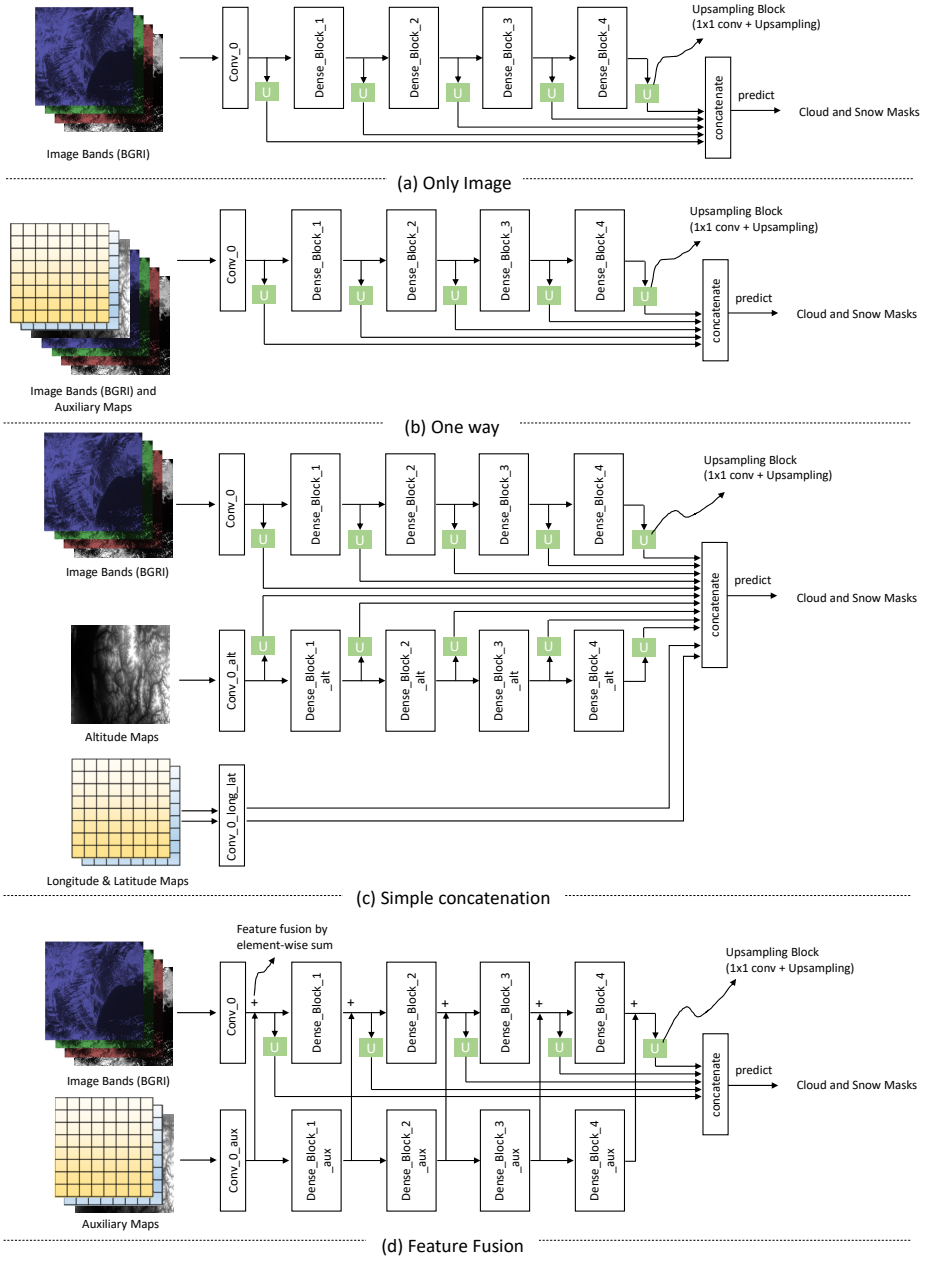

Figure 9: Other possible network structures in the method in addition to the one used in Figure 5: (a) Only Image. (b) One Way. (c) Simple Concatenation. (d) Feature Fusion (element-wise sum).

Longitutde	Latitude	Altitude	$F1\_c$	$F1\_s$	IoU_c	IoU_s	FAR	T
			94.37	83.10	89.34	71.08	2.58	2.24s
✓			95.10	84.39	90.65	73.00	2.24	3.34s
	✓		94.85	84.88	90.20	73.73	2.37	3.28s
		✓	94.85	85.04	90.21	73.97	2.40	3.31s
✓	✓		95.05	85.08	90.57	74.03	2.27	3.34s
✓		✓	95.01	85.95	90.49	75.36	2.28	3.34s
	✓	✓	94.47	85.47	89.53	74.63	2.54	3.27s
✓	✓	✓	**95.15**	**87.80**	**90.75**	**78.26**	**2.20**	3.45s

Table 4: Quantitative results of all possible combinations of auxiliary information in the cloud detection and snow detection tasks (%).

Choice 3. Another alternative design is "Simple Concatenation", which is shown in Figure 9 (c). The intuition behind this design is that the longitude and the latitude of the pixels do not need to be excessively processed because their values within each scene will not change too much. Therefore, it is supposed that only the image-like bands need to be fed into the Dense Feature Extraction module, and the maps of the longitude and latitude are only one-layer convoluted before the feature concatenation.

Choice 4. The last choice is "Feature Fusion", which is shown in Figure 9 (d). In this choice, the same network structure is used as the default settings, while the only differences are: 1) the "feature concatenation" is changed to the "element-wise sum" operation, and 2) the features from the auxiliary branch within each stage are fused to the BGRI image branch before the further procedures.

Table 3 shows the comparison results of the four network structures in the method (the default structure and the *Choice 1-4*). Slight differences can be observed in the results of the different settings and the proposed *Dual Feature Concatenation* shown in Figure 5 achieves the best in all metrics.

5.2.2. Ablation Studies on the Auxiliary Maps

In this experiment, ablation studies are conducted on three different types of geographic information: 1) longitude, 2) latitude, and 3) altitude. To show the importance of these auxiliary components, different combinations of them are evaluated. The network configurations for all the possible auxiliary information combinations are the same (described in the above Section 5.1), except for the number of input channel of the first convolution layer "Conv_0_aux". The order of auxiliary information is always: altitude, longitude

and latitude. For example, if only the information of latitude A_{lat} and altitude A_{alt} is used, the auxiliary map will be formed as $A = \text{concat}(A_{\text{Alt}}, A_{\text{Lat}})$. Table 4 shows the quantitative results of all the possible combinations of auxiliary information. The result indicates that the use of the auxiliary information is effective in the cloud and snow detection task. It can be seen that the integration of any of these auxiliary components brings noticeable improvement on the detection accuracy and the best result can be achieved when all the auxiliary components are integrated. In the cases that no altitude information is provided, adding the longitude map and the latitude map still can be beneficial for the cloud and snow detection.

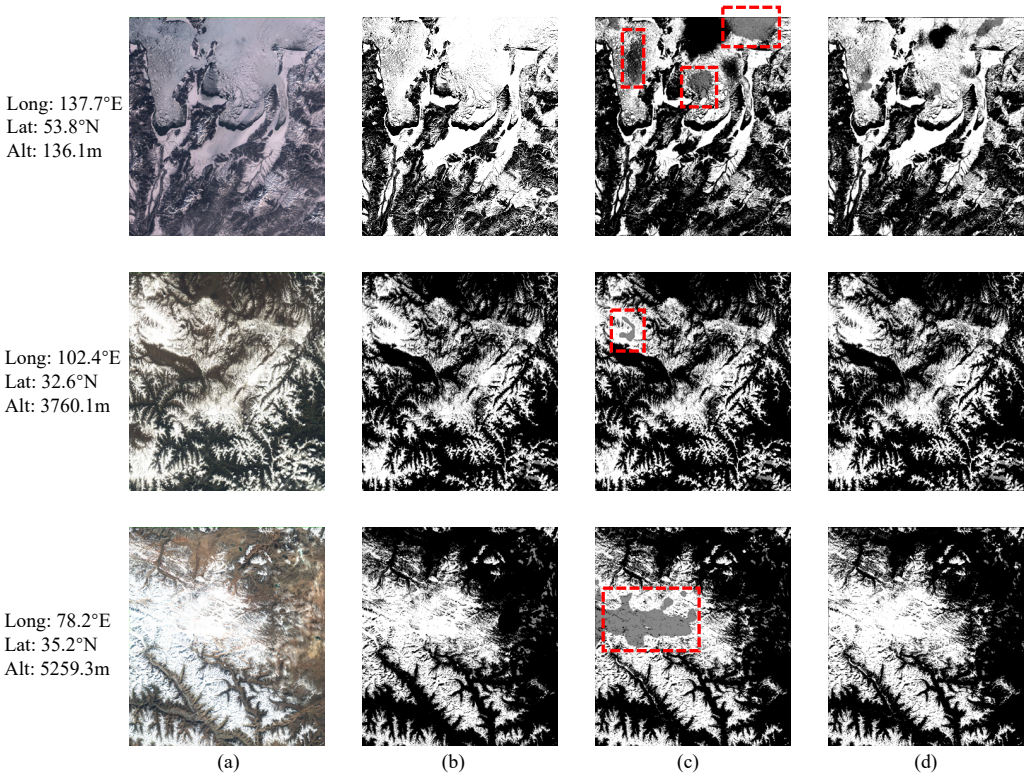

Figure 10: Results of the method w/ and w/o using auxiliary information. (a) Input image. (b) Ground truth label. (c) Detection results w/o auxiliary information. (d) Detection results w/ auxiliary information. In (b)-(d), the white, grey, and black pixels represent the snow, cloud, and background, respectively. Red boxes shows the false alarms of clouds which should be detected as snow, if auxiliary information is not used.

Besides, detailed comparisons are also made on detection results about

w/ and w/o auxiliary information. From Table 4, it can be noticed that with auxiliary information involved, the snow detection performance impressively improve. Figure 10 is an illustration of w/ and w/o using auxiliary information. If auxiliary information is adopted, the snow area which is wrongly recognized as cloud will be obviously reduced. It should be noted that snow detection is a very challenging task. As the first row of Figure 10 illustrates, even with the auxiliary information, the snow area can be wrongly detected as cloud. This is why the performance of snow detection is lower than that of cloud. From Table 5, it can be seen that using auxiliary information can improve cloud and snow detection performance on different geographical location ranges. Specially, the performance of cloud detection raises if using the auxiliary information. For the region where the altitude is higher than 5500m, the false alarms of clouds reduce as the last row of Figure 10 implies. The performance of snow detection is generally improved in different locations. In the low altitude regions, the snow detection performance increases with a large margin, which can help to reduce the missing alarms of snow as the first row of Figure 10 indicates. Therefore, the auxiliary geographic information helps the detection and does not hinder cloud or snow detection.

5.2.3. Evaluation on Network Backbone and Loss Function

For many deep learning based cloud and snow detection methods [15, 59, 9, 60, 61, 62], their backbone networks are built based on VGG [46] or ResNet [47]. Therefore, in this experiment, different network backbones are evaluated for the task.

Besides, considering that the huge difference in the number of pixels between different classes, whether class-balancing will be beneficial to the detection accuracy is also evaluated. In this case, the weighted softmax loss is used, which is similar to [77], as the loss function of pixel, which is expressed as follows:

$$L_{\text{weighted}} = -\sum_{m=1}^{3} \alpha_m y_m \log(P_m), \tag{12}$$

where α_m is the balancing weight for each class. α_m is set according to the pixel ratio of different classes, $\alpha_m = (n_m^{-1})/(\sum_{i=1}^{3} n_i^{-1})$, where n_m is the number of pixels belong to the class m in the dataset.

	w/o	w/	Δ	w/o	w/	Δ
	Cloud detection IoU_c			Snow detection IoU_s		
$long \leq -60°$	92.19	**93.93**	1.74	75.76	**83.91**	8.15
$-60° < long \leq 30°$[1]	88.13	**89.43**	1.30	-	-	-
$30° < long \leq 65°$	84.67	**86.87**	2.20	58.41	**60.59**	2.18
$65° < long \leq 100°$	83.43	**85.54**	2.11	75.18	**79.53**	4.35
$100° < long \leq 120°$	88.62	**89.86**	1.24	59.11	**68.42**	9.31
$long > 120°$	90.74	**91.08**	0.34	70.85	**79.17**	8.32
$lat \leq -23.5°$	86.29	**87.59**	1.30	76.96	**86.54**	9.58
$-23.5° < lat \leq 23.5°$[2]	92.07	**92.89**	0.82	-	-	-
$23.5° < lat \leq 43°$	89.04	**90.57**	1.53	75.58	**78.52**	2.94
$lat > 43°$	84.44	**86.98**	2.54	69.51	**77.96**	8.45
$alt \leq 50m$	91.07	**91.48**	0.41	68.01	**81.96**	13.95
$50m < alt \leq 500m$	88.60	**90.49**	1.89	72.25	**79.01**	6.76
$500m < alt \leq 2100m$	88.19	**89.88**	1.69	62.48	**72.43**	9.95
$2100m < alt \leq 3400m$	90.46	**93.23**	2.77	72.79	**81.04**	8.25
$3400m < alt \leq 5500m$	88.84	**90.27**	1.43	82.08	**83.22**	1.14
$alt > 5500m$	68.47	**84.60**	16.13	70.70	**77.44**	6.74

[1] On the test set of Levir_CS, there is little snow in this longitude range, therefore, the snow detection results of this range are not able to be collected.

[2] On the test set of Levir_CS, there is little snow in this latitude range, therefore, the snow detection results of this range are not able to be collected.

Table 5: Cloud and snow detection improvement on different locations.(%)

Backbones	Weighted	$F1\_c$	$F1\_s$	IoU_c	IoU_s	FAR	T
VGG16	×	77.49	39.88	63.25	24.91	9.30	3.31s
VGG16	√	88.48	57.10	79.33	39.95	5.20	3.31s
ResNet101	×	84.72	59.05	73.48	41.90	6.35	4.00s
ResNet101	√	90.67	68.71	82.94	52.33	4.45	4.00s
DenseNet169	×	**95.15**	**87.80**	**90.75**	**78.26**	**2.20**	3.45s
DenseNet169	√	93.98	85.18	88.65	74.20	2.81	3.45s

Table 6: Quantitative results of the method with different network backbones and weighted loss (%).

Table 6 shows the quantitative results of the method with different network backbones and losses. For the choice of the network backbones, the default design, i.e., the DenseNet169, obtains the best results. The accuracy increase is particularly significant for the snow detection task. As for the weighted loss, it can be observed that if the network backbone is VGG16 or ResNet101, the class-balancing can be useful. However, it does not help the detection if the network backbone is DenseNet169. As shown in Figure 11, if the weighted loss is used, there will be more false alarms in the detection result, especially for the snow pixels. The above observations indicate that the backbone of DenseNet169 can be a robust network backbone and suffer little impact of imbalanced data. If class-balancing techniques are applied, the detection results may get worse. Therefore, the Dense Feature Extraction produces robust image features where the imbalanced data problems can be alleviated. This is the reason why class-balancing is not used in the default loss function.

5.3. Comparison with Other Methods

In this section, the method is compared with four state of the art cloud and snow detection methods, the DCN [15], FECN [9], cloudFCN[78] and cloudUNet[57] . The DCN [15] is built based on the VGG16 backbone and produces the detection probability map from each level of the network. The final detection map of the DCN is obtained by summarizing all these probability maps. FECN [9] is also designed base on the VGG16 backbone. In this method, the features from all stages are concatenated for the final prediction. The cloudFCN[78] is a cloud detection method based on fully convolutional neural networks. It modifies the original framework of FCN[55] and can be

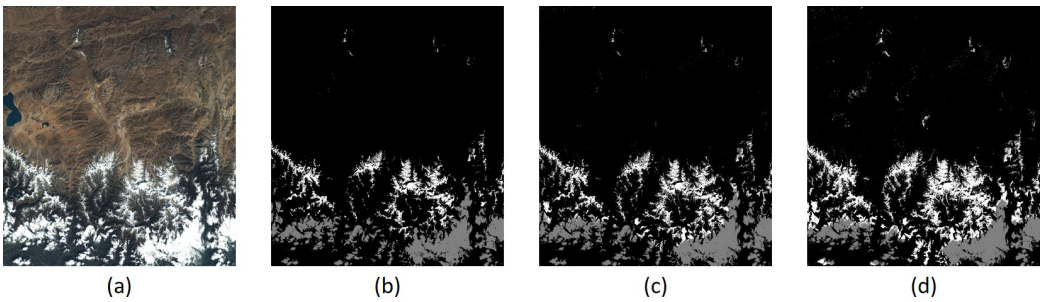

Figure 11: Results of the method w/ and w/o using class-balancing in the loss function. (a) Input image. (b) Ground truth label. (c) Detection results w/o class-balancing. (d) Detection results w/ class-balancing. In (b)-(d), the white, grey, and black pixels represent the snow, cloud, and background, respectively. More false alarms are observed in the detection results when the class-balancing is used, especially for the snow pixels.

successfully applied in cloud detection tasks. Similarly, cloudUNet[57] is a method based on UNet[64]. In this framework, shallow features are reused in the upsampling procedures. Note that FECN, cloudFCN and cloudUNet are originally proposed for cloud detection, and in the experiment, we extend them for both cloud and snow detection by modifying the number of output classes from 2 (cloud and background) to 3 (cloud, snow, and background).

Since the cloud and snow detection is essentially a semantic segmentation problem, the FCN-16s [55] and DeeplabV3+ [76], which are two well-known image segmentation methods in computer vision, are also compared. FCN-16s [55] uses the 16x downsampled feature maps for segmentation. Deeplab-V3+ [76] integrates both low-level features (4x downsampled) and high-level features (produced by spatial pyramid pooling at the end of the network encoder and are 16x downsampled), and performs 4x upsampling on the prediction result. For a fair comparison, the network backbones used in these two methods are both DenseNet169, which is the same as used in the method. Besides, a low-resolution version of the method: "GeoInfoNet4x" is also experimented, which only uses 4x and larger times downsampled features for segmentation, for the comparison of the feature scales. In addition, traditional statistic methods based on machine learning Scene Learning [7] and Coarse-to-Fine [36] are also evaluated. These methods are used for cloud detection and cannot finish the snow detection tasks.

Visual comparison results of different detection methods are shown in Figure 12. Table 7 shows their quantitative evaluation results. It can be

Methods	$F1\_c$	$F1\_s$	IoU_c	IoU_s	FAR	T
Scene Learning [7]	35.02	-	21.23	-	63.70	74.62s
Coarse-to-Fine [36]	57.63	-	40.48	-	15.37	342s
DCN [15]	90.64	59.08	82.88	41.92	4.05	1.03s
FECN [9]	89.72	60.25	81.36	43.12	4.33	2.17s
cloudFCN [78]	93.01	75.50	86.94	60.64	3.06	2.59s
cloudUNet [57]	94.03	82.54	88.73	70.27	2.65	2.00s
FCN-16s [55]	79.19	69.32	65.54	53.04	8.62	0.86s
DeeplabV3+ [76]	89.15	76.58	80.43	62.05	4.95	0.93s
Only Image (ours)	94.37	83.10	89.34	71.08	2.58	2.24s
GeoInfoNet4x (ours)	89.39	78.53	80.82	64.66	4.66	3.34s
GeoInfoNet (ours)	**95.15**	**87.80**	**90.74**	**78.26**	**2.20**	3.45s

Table 7: Quantitative comparisons of different methods on the Levir_CS dataset (%).

seen that the GeoInfoNet method generates more accurate cloud and snow masks than other methods. Even without using higher resolution features, the GeoInfoNet4x still achieves satisfying performance (especially the snow detection). Compared with deep learning methods, traditional methods based on machine learning may not be proper in cloud detection tasks on global region data. Although the DCN [15] and the FECN [9] do not perform very well in snow detection (probably due to the imbalanced classes as we analyzed in Section 5.2), they still outperforms the FCN-16s [55], the DeeplabV3+ [76], and even the GeoInfoNet4x in cloud detection. Based on the above observations, it can be concluded that for cloud and snow detection, utilizing all scales of features can obtain better detection results than those methods which only use lower resolution features.

5.4. Evaluation on Other Sensor Data

In this subsection, the proposed GeoInfoNet is evaluated on other sensor data to test the transferability of the method. L8_Biome [70] is chosen in this part. As L8_Biome [70] does not contain snow information masks, only evaluate only the performance on cloud detection is evaluated. In L8_Biome [70], images are randomly divided into the training set and the testing set. The number of scenes of the training set is 77 while that of the testing set is 18. To generate the corresponding digital elevation map, the original images is first transferred in the World Geodetic System (WGS84) by using the GDAL

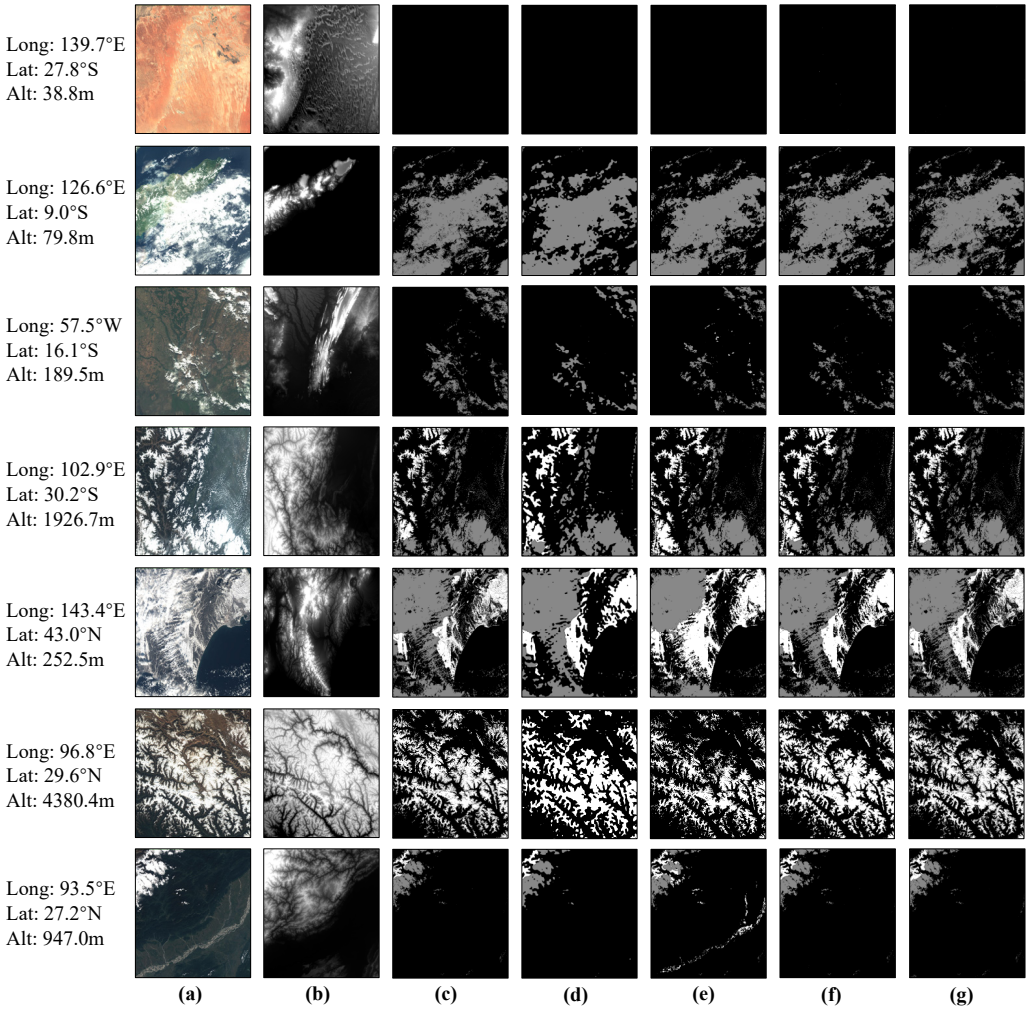

Figure 12: (Better viewed in color.) Visual comparison of different cloud and snow detection methods. (a) Input image. (b) The corresponding altitude map. (c) Ground truth label. (d)-(g) The detection results of DCN [15], DeeplabV3+ [76], GeoInfoNet4x (ours) and GeoInfoNet (ours), respectively. The white, grey, and black pixels represent the snow, cloud, and background. On the very left side of each row, the longitude and latitude of the central point and the mean altitude of the image is given.

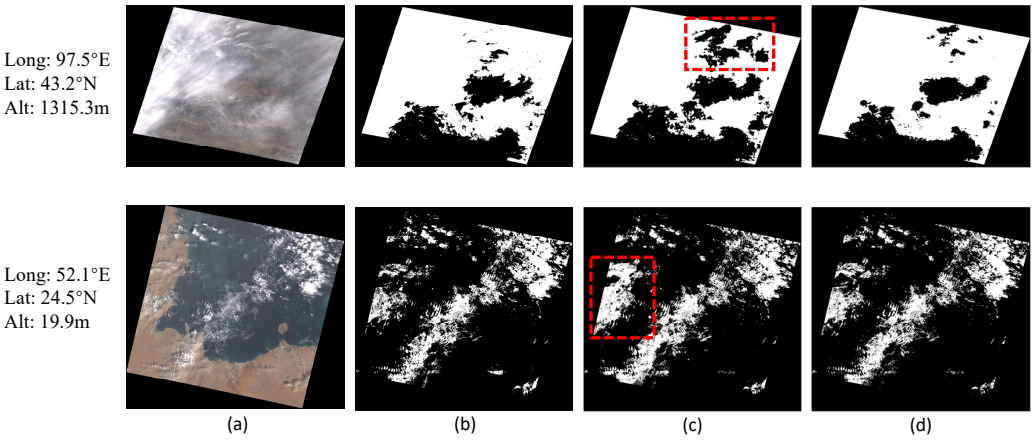

Figure 13: (Better viewed in color.) Visual evaluation results on L8_Biome [70]. (a) Input image. (b) Ground truth label. (c) The detection result of Only Image. (d) The detection result of GeoInfoNet. The white and black pixels represent the cloud (both thick and thin cloud) and the other catagories (including the clear region, cloud shadow and the filling region). Red boxes shows the wrong-predictions (first row: more missing area; second row: more false alarms), if auxiliary information is not used. On the very left side of each row, the longitude and latitude of the central point and the mean altitude of the image is given.

libray [65]. The mean area of the scenes in the test set is $55,126,569$ pixels (around $7425px \times 7425px$). The digital elevation map generation costs 158.11s time per scene on average. Similar experiment settings to those introduced in Section 5.1 are adopted here, except for the number of iterations is set to 8×10^5. Two network structures are evaluated, 1) the network with only image branch and 2) the proposed GeoInfoNet.

Figure 13 illustrates the visual comparison results of different detection methods. Table 8 shows the quantitative evaluation results. It can be seen that the proposed method GeoInfoNet can be well applied to Landsat8. Besides, these results also prove the effectiveness of adding geographic information for detecting cloud.

Networks	$F1\_c$	IoU_c	FAR	T
Only Image (ours)	95.17	90.79	2.79	118.82s
GeoInfoNet (ours)	**95.84**	**92.01**	**2.43**	188.00s

Table 8: Quantitative evaluation results on the test set of L8_Biome [70] (%).

6. Discussion

6.1. What Prior Information Does Our Method Learn?

In this section, an interesting question is raised: what kind of prior information does GeoInfoNet learn? A deep investigation has been made based on the method of Activation Maximization [79]. This method was originally proposed to visualize the learned convolutional filters of the network by optimizing the feature maps. For a typical input U and a fixed network parameter Θ, the input U can be optimized as follows,

$$U^* = argmax f_{i,j}(\Theta, U), s.t. ||U|| \leq \rho, \tag{13}$$

where $f_{i,j}(\cdot, \cdot)$ is an activation function of the input U and the network parameter Θ, given a convolution filter i from a given layer j in the network. Here, the parameters of GeoInfoNet are fixed, and the inputs of auxiliary branch A are optimized. For the details of optimization, Adam optimizer[80] is used with the learning rate is set to 0.1 and the number of iterations is set to 90. The activation function $f_{i,j}(\cdot, \cdot)$ is set as the second or third channel of the outputs of the final score layer, which represent the score of the cloud class and snow class respectively. Through activation maximization, the inputs of auxiliary branch A will be changed through different activations, which represents the prior auxiliary information learned by GeoInfoNet.

Therefore, all the images in the testing sets can be optimized and the mean values of the optimized auxiliary information map have been calculated. Figure 14, Figure 15 and Figure 16 illustrate the histograms of the mean values of the optimized auxiliary information maps in different location ranges. As the optimized values of the auxiliary maps have exceeded the real range values of geographic ranges, therefore, the activation maximization can only reflect the tendencies. From these figures, it can be seen that the method learns the prior knowledge that the cloud tends to appear in low altitude, low latitude and high longitude, while the trend for the snow is the opposite. This is consistent with the common sense because it is indeed easier to see snow in high altitudes. Besides, to some degree, it also grasp the data distribution shown in Figure 8.

6.2. Analysis of the Feature Importance

In the method, since the features from all levels from both the image branch and the auxiliary branch are used, it is necessary to analyze how

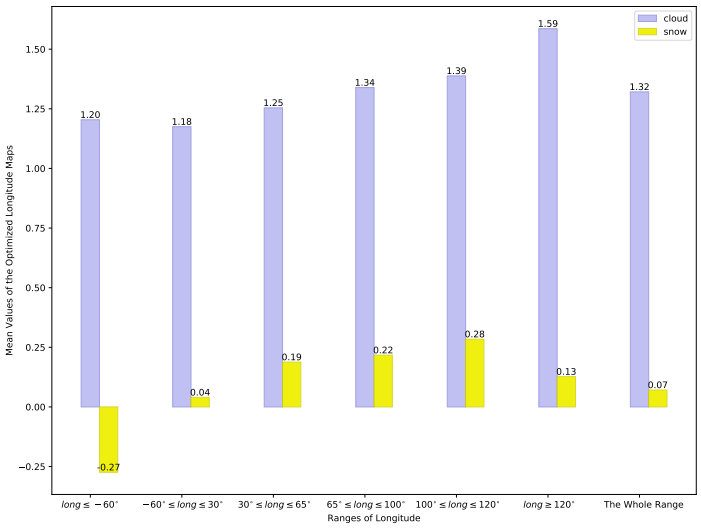

Figure 14: The histogram of activation maximization results of the longitude maps.

much these features of different levels and different branches contribute to the cloud and snow detection task.

In the proposed GeoInfoNet, the cloud and snow detection results are obtained according to the concatenated features M formed by different levels of features $M_{\text{img},0}, ..., M_{\text{img},4}, M_{\text{aux},0}, ..., M_{\text{aux},4}$ in both two branches (shown in Eq.4). Therefore, the gradient of the prediction score S_t of a specific class t (cloud or snow in the topic) is computed with respect to M, and then multiply this gradient on the feature map to produce the "importance map" G of each pixel location on this feature map. The "importance" can be expressed as follows:

$$G_t = \sum_{i=1}^{C} (\frac{\partial S_t}{\partial M} M)^{(i)}, \tag{14}$$

where i is the channel index and C is the number of channels in the feature map M. Here, the gradient information conveys the neuron importance weight, therefore, by multipling the gradient and the feature maps, the generated map G_t is a hot map which can show the most sensitive region of the

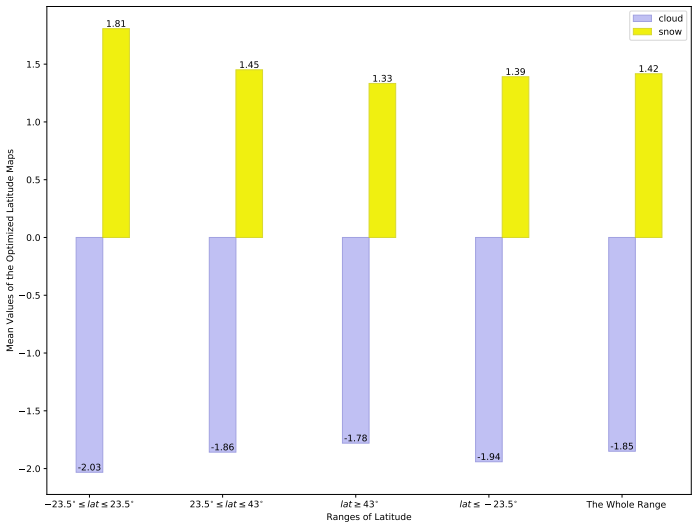

Figure 15: The histogram of activation maximization results of the latitude maps.

specific class t grasped by the network. Thay is why G_t is named "importance map". This process is similar to a well-known feature visualization method called Gradient Class Activation Map (Grad-CAM) [81]. The original Grad-CAM [81] uses ReLU operation for it only interests in pixels with positive response to the specific class t. With ReLU function operated, those pixels with negative response are filtered. This operation is removed here because the negative values may reflect the negative tendency to this class (t), which may help to reduce the false alarms to this class.

Since the Eq. 14 is linear, the importance of different channel splits can also be computed from the equation. For example, the importance of the features from the image branch and the auxiliary branch can be efficiently computed by accumulating the above scores over the corresponding feature channels. Suppose G_t^{img} and G_t^{aux} represent for the feature importance of the two branches for the class t, and thus $G_t = G_t^{\text{img}} + G_t^{\text{aux}}$. Figure 17 illustrates the importance maps target on the class of cloud and snow, which indicates that features from both two branches take effect in the computation of obtaining cloud and snow masks. Besides, the auxiliary information plays on an "auxiliary" role since the absolute values of G_c^{aux} and G_s^{aux} is much smaller

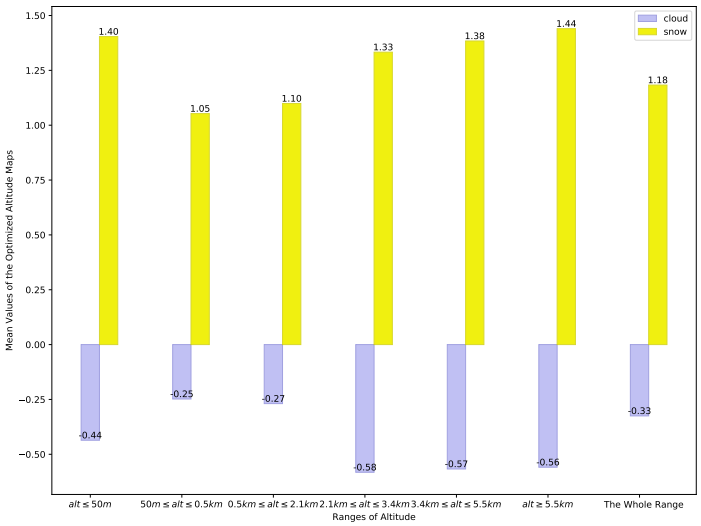

Figure 16: The histogram of activation maximization results of the altitude maps.

than G_c^{img} and G_s^{img}. Therefore, the proposed GeoInfoNet still mainly relies on the image information and the auxiliary information helps the network improve the cloud and snow detection performance.

Besides, by using this method, the feature importance of different feature levels can also be easily obtained. Therefore, all the images in the test set of Levir_CS are scanned and the importance of the different feature blocks in both branches is calculated. For each group of the features, an average importance score is computed, which is shown in Figure 18. From this figure, it can be seen that in most of the feature groups (except for "Dense1" and "Dense2"), the importance of the image branch larger than that of the auxiliary branch in both cloud and snow detection tasks. This observation indicates that the image branch dominates the detection but the geographic information still contributes to the results to some degree. It can also be seen that in both branches, the importance of the very first convolution layer "Conv_0" and the very last convolution group "Dense_Block_4" are both very high, which shows that low-level features with high resolution and high-level features with low resolution are both crucial for cloud and snow detection tasks. As a comparison, for the groups from the "Dense_Block_1"

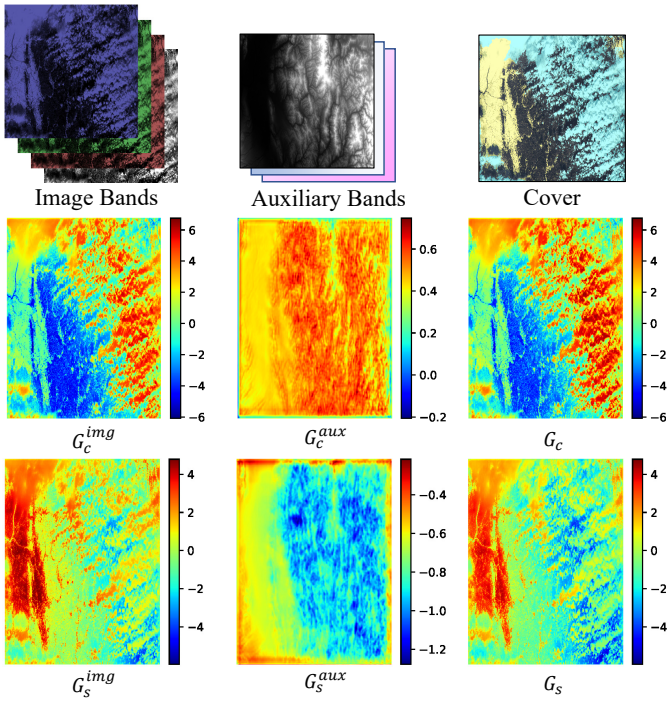

Figure 17: Visualization results of the feature importance from the two different branches by using Eq. 14. The feature importance of cloud and snow are shown in the 2nd row and 3rd row, respectively. The image on the top-right is a cover image that represents the detection result, where light-blue represent cloud pixels while light-yellow represents snow pixels.

to "Dense_Block_3", the feature importance of both branches and both tasks is very close. Therefore, in the middle levels, features of both branches contribute to almost the same degree to the results.

7. Conclusion and Future Works

In this paper, a novel cloud and snow detection method is proposed for remote sensing images named "Geographic Information-drive Neural Networks (GeoInfoNet)". Different from previous methods that simply perform detection solely based on the image data, the method integrates both the image and geographic information (altitude, latitude, and longitude) for training and detection. A large dataset for cloud and snow detection is also built, which contains 4,168 scenes and the corresponding geographic information.

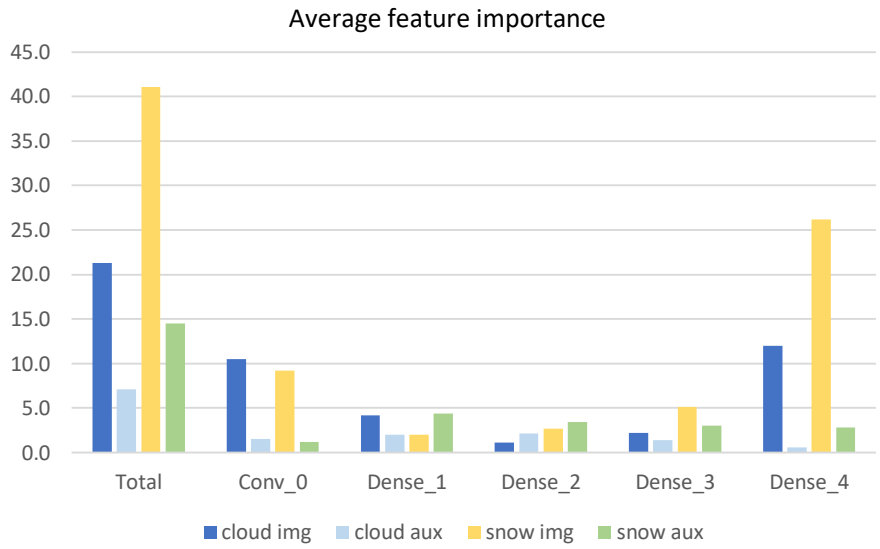

Figure 18: The importance of the features from different levels and different branches for cloud detection and snow detection tasks. For each group of features, an average importance score on the test set of Levir_CS is computed.

Extensive experiments verified the effectiveness of integrating geographic information for the cloud and snow detection tasks. The method outperforms other state-of-the-art methods with a large margin. Besides, the visualization is also presented to show what the method learns and how much the different parts of the network contribute to the detection tasks.

The future works include four parts. The first part is the improvement of the computational efficiency of the network. The second part of work is the integration of other types of geographic information (e.g., sun altitude angle, imaging time, temperature, etc). Third, cloud shadow will be focused in the future works, and this source of information will be integrated in the dataset. Finally, cloud and snow detection in time series (or multi-temporal cloud and snow detection) at specific locations will be investigated.

Acknowledgments

The work was supported by the National Key R&D Program of China under the Grant 2019YFC1510905, the National Natural Science Foundation of China under the Grant 61671037 and the Beijing Natural Science

Foundation under the Grant 4192034.

References

[1] Z. Zou, Z. Shi, Ship detection in spaceborne optical image with svd networks., IEEE Trans. Geosci. Remote Sens. 54 (10) (2016) 5832–5845.

[2] H. Lin, Z. Shi, Z. Zou, Maritime semantic labeling of optical remote sensing images with multi-scale fully convolutional network., Remote Sens. 9 (5) (2017) 480.

[3] H. Lin, Z. Shi, Z. Zou, Fully convolutional network with task partitioning for inshore ship detection in optical remote sensing images., IEEE Geosci. Remote Sens. Lett. 14 (10) (2017) 1665–1669.

[4] Z. Zou, Z. Shi, Random access memories: A new paradigm for target detection in high resolution aerial remote sensing images., IEEE Trans. Image Process. 27 (3) (2017) 1100–1111.

[5] T. Shi, Q. Xu, Z. Zou, Z. Shi, Automatic raft labeling for remote sensing images via dual-scale homogeneous convolutional neural network., IEEE Trans. Image Process. 10 (7) (2018) 1130.

[6] Q. Zhang, C. Xiao, Cloud detection of rgb color aerial photographs by progressive refinement scheme., IEEE Trans. Geosci. Remote Sens. 52 (11) (2014) 7264–7275.

[7] Z. An, Z. Shi, Scene learning for cloud detection on remote-sensing images., IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 8 (8) (2015) 4206–4222.

[8] F. Xie, M. Shi, Z. Shi, J. Yin, D. Zhao, Multilevel cloud detection in remote sensing images based on deep learning., IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 10 (8) (2017) 3631–3640.

[9] X. Wu, Z. Shi, Utilizing multilevel features for cloud detection on satellite imagery., Remote Sens. 10 (11) (2018) 1853.

[10] X. Li, H. Shen, L. Zhang, H. Zhang, Q. Yuan, G. Yang, Recovering quantitative remote sensing products contaminated by thick clouds and shadows using multitemporal dictionary learning, IEEE Trans. Geosci. Remote Sens. 52 (11) (2014) 7086–7098.

[11] X. Li, Y. Jing, H. Shen, L. Zhang, The recent developments in cloud removal approaches of modis snow cover product., Hydrol Earth Syst Sci 23 (5).

[12] J. Bi, J. H. Belle, Y. Wang, A. I. Lyapustin, A. Wildani, Y. Liu, Impacts of snow and cloud covers on satellite-derived pm2. 5 levels, Remote Sens. Environ. 221 (2019) 665–674.

[13] J. L. Campbell, M. J. Mitchell, P. M. Groffman, L. M. Christenson, J. P. Hardy, Winter in northeastern north america: a critical period for ecological processes, Front Ecol Environ. 3 (6) (2005) 314–322.

[14] X. Wang, J. Wang, T. Che, X. Huang, X. Hao, H. Li, Snow cover mapping for complex mountainous forested environments based on a multi-index technique, IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 11 (5) (2018) 1433–1441.

[15] Y. Zhan, J. Wang, J. Shi, G. Cheng, L. Yao, W. Sun, Distinguishing cloud and snow in satellite images via deep convolutional network, IEEE Geosci. Remote Sens. Lett. 14 (10) (2017) 1785–1789.

[16] R. R. Irish, Landsat 7 automatic cloud cover assessment, in: Algorithms for Multispectral, Hyperspectral, and Ultraspectral Imagery VI, Vol. 4049, International Society for Optics and Photonics, 2000, pp. 348–355.

[17] R. R. Irish, J. L. Barker, S. N. Goward, T. Arvidson, Characterization of the landsat-7 etm+ automated cloud-cover assessment (acca) algorithm, Photogramm. Eng. Remote Sens. 72 (10) (2006) 1179–1188.

[18] Z. Zhu, C. E. Woodcock, Object-based cloud and cloud shadow detection in landsat imagery, Remote Sens. Environ. 118 (2012) 83–94.

[19] Z. Zhu, C. E. Woodcock, Automated cloud, cloud shadow, and snow detection in multitemporal landsat data: An algorithm designed specifically for monitoring land cover change, Remote Sens. Environ. 152 (2014) 217–234.

[20] S. Qiu, B. He, Z. Zhu, Z. Liao, X. Quan, Improving fmask cloud and cloud shadow detection in mountainous area for landsats 4–8 images, Remote Sens. Environ. 199 (2017) 107–119.

[21] S. Qiu, Z. Zhu, B. He, Fmask 4.0: Improved cloud and cloud shadow detection in landsats 4–8 and sentinel-2 imagery, Remote Sens. Environ. 231 (2019) 111205.

[22] L. Sun, X. Mi, J. Wei, J. Wang, X. Tian, H. Yu, P. Gan, A cloud detection algorithm-generating method for remote sensing data at visible to short-wave infrared wavelengths, ISPRS J. Photogramm. Remote Sens. 124 (2017) 70–88.

[23] X. Zhu, E. H. Helmer, An automatic method for screening clouds and cloud shadows in optical satellite image time series in cloudy regions, Remote Sens. Environ. 214 (2018) 135–153.

[24] O. Hagolle, M. Huc, D. V. Pascual, G. Dedieu, A multi-temporal method for cloud detection, applied to formosat-2, venμs, landsat and sentinel-2 images, Remote Sens. Environ. 114 (8) (2010) 1747–1755.

[25] J. Bian, A. Li, H. Jin, W. Zhao, G. Lei, C. Huang, Multi-temporal cloud and snow detection algorithm for the hj-1a/b ccd imagery of china, in: 2014 IEEE Geoscience and Remote Sensing Symposium, IEEE, 2014, pp. 501–504.

[26] J. Bian, A. Li, Q. Liu, C. Huang, Cloud and snow discrimination for ccd images of hj-1a/b constellation based on spectral signature and spatio-temporal context, Remote Sens. 8 (1) (2016) 31.

[27] B. Zhong, W. Chen, S. Wu, L. Hu, X. Luo, Q. Liu, A cloud detection method based on relationship between objects of cloud and cloud-shadow for chinese moderate to high resolution satellite imagery, IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 10 (11) (2017) 4898–4908.

[28] Z. Li, H. Shen, H. Li, G. Xia, P. Gamba, L. Zhang, Multi-feature combined cloud and cloud shadow detection in gaofen-1 wide field of view imagery, Remote Sens. Environ. 191 (2017) 342–358.

[29] D. J. Selkowitz, R. R. Forster, An automated approach for mapping persistent ice and snow cover over high latitude regions, Remote Sens. 8 (1) (2016) 16.

[30] H. Choi, R. Bindschadler, Cloud detection in landsat imagery of ice sheets using shadow matching technique and automatic normalized difference snow index threshold value decision, Remote Sens. Environ. 91 (2) (2004) 237–242.

[31] T. Andersen, Operational snow mapping by satellites, in: Hydrological aspects of alpine and high mountain areas, Proceedings of the Exeter symposium, no. 138, 1982, pp. 149–154.

[32] P. Li, L. Dong, H. Xiao, M. Xu, A cloud image detection method based on svm vector machine, Neurocomputing 169 (2015) 34–42.

[33] A. Hollstein, K. Segl, L. Guanter, M. Brell, M. Enesco, Ready-to-use methods for the detection of clouds, cirrus, snow, shadow, water and clear sky pixels in sentinel-2 msi images, Remote Sens. 8 (8) (2016) 666.

[34] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.

[35] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.

[36] X. Kang, G. Gao, Q. Hao, S. Li, A coarse-to-fine method for cloud detection in remote sensing images, IEEE Geosci. Remote Sens. Lett. 16 (1) (2018) 110–114.

[37] C. Deng, Z. Li, W. Wang, S. Wang, L. Tang, A. C. Bovik, Cloud detection in satellite images based on natural scene statistics and gabor features, IEEE Geosci. Remote Sens. Lett. 16 (4) (2018) 608–612.

[38] A. N. Srivastava, J. Stroeve, Onboard detection of snow, ice, clouds and other geophysical processes using kernel methods, in: Proceedings of the ICML, Vol. 3, Citeseer, 2003.

[39] A. K. Jain, F. Farrokhnia, Unsupervised texture segmentation using gabor filters, in: 1990 IEEE international conference on systems, man, and cybernetics conference proceedings, IEEE, 1990, pp. 14–19.

[40] R. Mehrotra, K. R. Namuduri, N. Ranganathan, Gabor filter-based edge detection, Pattern Recognit. 25 (12) (1992) 1479–1494.

[41] T. P. Weldon, W. E. Higgins, D. F. Dunn, Efficient gabor filter design for texture segmentation, Pattern Recognit. 29 (12) (1996) 2005–2015.

[42] J. Munkres, Algorithms for the assignment and transportation problems, J. Soc. Ind. Appl. Math. 5 (1) (1957) 32–38.

[43] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin, Shape distributions, ACM Trans. Gr. 21 (4) (2002) 807–832.

[44] G. Chen, D. E, Support vector machines for cloud detection over ice-snow areas, Geo. Spat. Inf. Sci. 10 (2) (2007) 117–120.

[45] J. P. Musial, F. Hüsler, M. B. Sütterlin, C. Neuhaus, S. Wunderle, Probabilistic approach to cloud and snow detection on advanced very high resolution radiometer (avhrr) imagery, Atmos Meas Tech (AMT) 7 (3) (2014) 799–822.

[46] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556. Submission date: 10th April, 2015.

[47] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[48] G. Huang, Z. Liu, L. Van Der Maaten, K. Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.

[49] M. Shi, F. Xie, Y. Zi, J. Yin, Cloud detection of remote sensing images by deep learning, in: 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), IEEE, 2016, pp. 701–704.

[50] M. Le Goff, J.-Y. Tourneret, H. Wendt, M. Ortner, M. Spigai, Deep learning for cloud detection, in: ICPRS (8th International Conference of Pattern Recognition Systems), IET, 2017.

[51] Y. Zi, F. Xie, Z. Jiang, A cloud detection method for landsat 8 images based on pcanet, Remote Sens. 10 (6) (2018) 877.

[52] G. Mateo-García, L. Gómez-Chova, G. Camps-Valls, Convolutional neural networks for multispectral image cloud masking, in: 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), IEEE, 2017, pp. 2255–2258.

[53] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Süsstrunk, Slic superpixels compared to state-of-the-art superpixel methods, IEEE Trans Pattern Anal. Mach. Intell. 34 (11) (2012) 2274–2282.

[54] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[55] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.

[56] M. Wieland, Y. Li, S. Martinis, Multi-sensor cloud and cloud shadow segmentation with a convolutional neural network, Remote Sens Environ. 230 (2019) 111203.

[57] J. H. Jeppesen, R. H. Jacobsen, F. Inceoglu, T. S. Toftegaard, A cloud detection algorithm for satellite imagery based on deep learning, Remote Sens Environ. 229 (2019) 247–259.

[58] D. Chai, S. Newsam, H. K. Zhang, Y. Qiu, J. Huang, Cloud and cloud shadow detection in landsat imagery based on deep convolutional neural networks, Remote Sens Environ. 225 (2019) 307–316.

[59] Z. Yan, M. Yan, H. Sun, K. Fu, J. Hong, J. Sun, Y. Zhang, X. Sun, Cloud and cloud shadow detection using multilevel feature fused segmentation network, IEEE Geosci. Remote Sens. Lett. 15 (10) (2018) 1600–1604.

[60] Z. Shao, Y. Pan, C. Diao, J. Cai, Cloud detection in remote sensing images based on multiscale features-convolutional neural network, IEEE Trans. Geosci. Remote Sens. 57 (6) (2019) 4062–4076.

[61] J. Yang, J. Guo, H. Yue, Z. Liu, H. Hu, K. Li, Cdnet: Cnn-based cloud detection for remote sensing imagery, IEEE Trans. Geosci. Remote Sens. 57 (8) (2019) 6195–6211.

[62] Z. Li, H. Shen, Q. Cheng, Y. Liu, S. You, Z. He, Deep learning based cloud detection for medium and high resolution remote sensing images of different sensors, ISPRS J. Photogramm. Remote Sens. 150 (2019) 197–212.

[63] W. Li, Z. Zou, Z. Shi, Deep matting for cloud detection in remote sensing images, IEEE Trans. Geosci. Remote Sens. 58 (12) (2020) 8490–8502.

[64] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical image computing and computer-assisted intervention, Springer, 2015, pp. 234–241.

[65] F. Warmerdam, The geospatial data abstraction library, in: Open source approaches in spatial data handling, Springer, 2008, pp. 87–104.

[66] S. Zhao, T. Yu, Q. Meng, Q. Zhou, F. Wang, L. Wang, Y. Hu, Gdal-based extend arcgis engine's support for hdf file format, in: 2010 18th International Conference on Geoinformatics, IEEE, 2010, pp. 1–3.

[67] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv:1502.03167. Submission date: 2nd March, 2015.

[68] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, in: Proceedings of the fourteenth international conference on artificial intelligence and statistics, 2011, pp. 315–323.

[69] P. L. Scaramuzza, M. A. Bouchard, J. L. Dwyer, Development of the landsat data continuity mission cloud-cover assessment algorithms, IEEE Transactions on Geoscience and Remote Sensing 50 (4) (2011) 1140–1154.

[70] S. Foga, P. L. Scaramuzza, S. Guo, Z. Zhu, R. D. Dilley Jr, T. Beckmann, G. L. Schmidt, J. L. Dwyer, M. J. Hughes, B. Laue, Cloud detection algorithm comparison and validation for operational landsat data products, Remote sensing of environment 194 (2017) 379–390.

[71] Z. Zou, W. Li, T. Shi, Z. Shi, J. Ye, Generative adversarial training for weakly supervised cloud matting, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 201–210.

[72] J. Lu, Y. Wang, Y. Zhu, X. Ji, T. Xing, W. Li, A. Y. Zomaya, P_segnet and np_segnet: New neural network architectures for cloud recognition of remote sensing images, IEEE Access 7 (2019) 87323–87333.

[73] H. Tran, P. Nguyen, M. Ombadi, K.-l. Hsu, S. Sorooshian, X. Qing, A cloud-free modis snow cover dataset for the contiguous united states from 2000 to 2017, Scientific data 6 (2019) 180300.

[74] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.

[75] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1026–1034.

[76] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 801–818.

[77] S. Xie, Z. Tu, Holistically-nested edge detection, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1395–1403.

[78] A. Francis, P. Sidiropoulos, J.-P. Muller, Cloudfcn: Accurate and robust cloud detection for satellite imagery with deep learning, Remote Sens. 11 (19) (2019) 2312.

[79] D. Erhan, Y. Bengio, A. Courville, P. Vincent, Visualizing higher-layer features of a deep network, University of Montreal 1341 (3) (2009) 1.

[80] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980. Submission date: 30th Jan, 2017.

[81] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 618–626.