



Satellite data cloud detection using deep learning supported by hyperspectral data

Lin Sun, Xu Yang, Shangfeng Jia, Chen Jia, Quan Wang, Xinyan Liu, Jing Wei & Xueying Zhou

To cite this article: Lin Sun, Xu Yang, Shangfeng Jia, Chen Jia, Quan Wang, Xinyan Liu, Jing Wei & Xueying Zhou (2020) Satellite data cloud detection using deep learning supported by hyperspectral data, International Journal of Remote Sensing, 41:4, 1349-1371, DOI: [10.1080/01431161.2019.1667548](https://doi.org/10.1080/01431161.2019.1667548)

To link to this article: <https://doi.org/10.1080/01431161.2019.1667548>



© 2019 Informa UK Limited, trading as Taylor & Francis Group



Published online: 24 Sep 2019.



Submit your article to this journal [↗](#)



Article views: 733



View related articles [↗](#)



View Crossmark data [↗](#)



Satellite data cloud detection using deep learning supported by hyperspectral data

Lin Sun^a, Xu Yang^a, Shangfeng Jia^a, Chen Jia^a, Quan Wang^a, Xinyan Liu^a, Jing Wei^b and Xueying Zhou^c

^aGeomatics College, Shandong University of Science and Technology, Qingdao, Shandong, China; ^bCollege of Global Change and Earth System Science, Beijing Normal University, Beijing, China; ^cSchool of Remote Sensing and Information Engineering, Wuhan University, Wuhan, Hubei, China

ABSTRACT

Deep learning methods can play an important role in satellite data cloud detection. The number and quality of training samples directly affect the accuracy of cloud detection based on deep learning. Therefore, selecting a large number of representative and high-quality training samples is a key step in cloud detection based on deep learning. For different satellite data sources, choosing sufficient and high-quality training samples has become an important factor limiting the application of deep learning in cloud detection. This paper presents a fast method for obtaining high-quality learning samples, which can be used for cloud detection of different satellite data with deep learning methods. AVIRIS (Airborne Visible Infrared Imaging Spectrometer) data, which have 224 continuous bands in the spectral range from 400–2500 nm, are used to provide cloud detection samples for different types of satellite data. Through visual interpretation, a sufficient number of cloud and clear sky pixels are selected from the AVIRIS data to construct a hyperspectral data sample library, which is used to simulate different satellite data (such as data from the Moderate Resolution Imaging Spectroradiometer (MODIS) and Landsat Operational Land Imager (OLI) satellites) as training samples. This approach avoids selecting training samples for different satellite sensors. Based on the Keras deep learning framework platform, a backpropagation (BP) neural network is employed for cloud detection from Landsat 8 OLI, National Polar-orbiting Partnership (NPP) Visible Infrared Imaging Radiometer Suite (VIIRS) and Terra MODIS data. The results are compared with cloud coverage results interpreted via artificial vision. The results demonstrate that the algorithm achieves good cloud detection results for the above data, and the overall accuracy is greater than 90%.

ARTICLE HISTORY

Received 14 December 2018
Accepted 18 July 2019

1. Introduction

Statistical analysis shows that clouds cover approximately 67% of the Earth's surface (King et al. 2013). Optical sensors cannot effectively receive information from the surface due to cloud coverage, which greatly impacts the interpretation of ground objects and

the quantitative inversion of the physical parameters of the land, ocean and atmosphere (Harshvardhan, Randall, and Corsetti 1989; Jedlovec, Haines, and Lafontaine 2008; Li et al. 2017). Clouds are an important climatic and meteorological element that can help identify weather phenomena such as rainstorms and hurricanes and track and forecast meteorological changes. Therefore, the study of cloud detection methods has attracted the attention of remote sensing researchers.

Methods of cloud detection can be divided into two categories: threshold methods and statistical learning methods. Threshold methods use the high reflectivity of clouds in the visible and near-infrared bands and the low temperature in the thermal infrared band to detect clouds. Because of their simplicity and fast calculation speed, threshold methods have been widely used in cloud detection in remote sensing images, such as the APOLLO (AVHRR Processing scheme Over cLOUDs, Land, and Ocean) algorithm and the CLAVR (Clouds from the Advanced Very High Resolution Radiometer) algorithm used for AVHRR (Advanced Very High Resolution Radiometer) data cloud detection (Saunders and Kriebel 1988; Stowe et al. 1999), the MODIS (Moderate Resolution Imaging Spectroradiometer) cloud mask algorithm (Ackerman et al. 2015) used for the cloud product generated from MODIS 1B data (MODIS Characterization Support Team 2009), and the Fmask cloud mask algorithm used for Landsat-series data cloud detection (Zhu and Woodcock 2012, 2014). The band selection and threshold determination depend on analysis of the spectral differences between clouds and typical surfaces. However, due to the complexity of surface structures and cloud states, it is usually difficult to consider the influencing factors comprehensively. Therefore, threshold-based methods usually have low accuracy and are unstable, especially in some areas with high reflectivity, in addition to areas covered by thin and fragmentary clouds.

Based on sample data, the cloud detection method of statistical learning includes cluster analysis, support vector machines (SVMs), artificial neural networks (ANNs) and other pattern recognition methods to achieve cloud detection. Such methods can automatically learn the features of clouds and the typical land types directly from the input samples and train the cloud detection method using training samples. Richard (1994) uses a 16×16 -pixel matrix as the unit area and uses the calculated spectrum, texture and other physical features as the input to the probabilistic NN (PNN). AVHRR data are divided into 10 categories, including convolution clouds, cirrus clouds and other clear sky through training (Richard 1994). Liou, Azimi-Sadjadi, and Reinke et al. (1994) extracts the texture features of clouds based on the singular value decomposition (SVD) method and uses a self-organizing map network to classify the features and perform cloud detection.

Cloud detection methods based on statistical learning are based on spectral and texture information provided by sample data. Cloud detection is performed using pattern recognition methods such as clustering analysis, SVMs and ANNs. Generally, the accuracy and stability of cloud detection methods based on statistical learning are better than threshold method, but such methods are still limited by people's understanding of the spectral and texture differences of different objects and clouds.

Deep learning based on ANNs is currently widely used in satellite image cloud detection. A neural network (NN) usually involves more than two hidden layers; thus, it is a deep learning method. Like shallow NNs, deep NNs use features that are specifically learned from data rather than manual features designed based on specific

knowledge. Deep learning, as an alternative to NNs, has been rapidly developed due to advances in algorithms, high-performance computing and big data. It has been applied to computer vision, speech recognition and natural language processing. Additionally, there are many tasks in remote sensing data processing and information extraction that use deep learning, including target detection, land cover and land-use classification, and change detection (Papadomanolaki et al. 2016).

However, the application of deep learning to remote sensing still has many challenges. John Ball, Anderson, and Chan (2017) listed some of the unsolved challenges and opportunities; the first problem is the inadequacy of learning samples. Many deep learning systems have thousands or millions of parameters, and they require a large amount of training data (Sun et al. 2017). When using a deep learning algorithm for cloud detection with specific satellite data, a large number of representative training samples must be selected from the data, which requires expert knowledge and significant amounts of time, energy and money.

To reduce the work associated with sample identification in cloud detection using satellite data, this paper proposes to use hyperspectral AVIRIS (Airborne Visible Infrared Imaging Spectrometer) data with continuous spectral coverage in the spectral range between 360 and 2500 nm to identify samples. A large number of AVIRIS data are selected to mark training samples and construct a cloud detection pixel sample database based on the AVIRIS data. According to the spectral response function of the sensor from which the data are obtained (such as MODIS, Landsat Operational Land Imager (OLI), etc.), labelled samples from the AVIRIS data are simulated to mimic data from the sensor under consideration, which are then used as the training samples of the sensor for cloud detection in the deep learning method. Different sensors can use the same training samples; thus, users can focus on increasing the quality of the training samples using AVIRIS data. The number and quality of training samples can be greatly improved, and the time and cost required to create training samples can be greatly reduced.

Based on the Keras deep learning framework platform, a deep backpropagation (BP) NN for cloud detection is designed (Graziotin and Abrahamsson 2013). The multispectral sample data are input into the network, and multispectral sensor cloud detection rules based on spectral characteristics are obtained.

The technical flowchart is shown in Figure 1.

2. Construction of cloud detection training sample dataset based on AVIRIS

2.1. AVIRIS data introduction

AVIRIS is an instrument developed and owned by the National Aeronautics and Space Administration (NASA)/Jet Propulsion Laboratory (JPL). The instrument measures 224 continuous spectral bands of solar energy with a spectral range of 360–2500 nm (Green et al. 1998). AVIRIS is now an operational instrument with reliable radiation and spectral calibration. Typically, images with a 20 m pixel size are obtained from the NASA Er-2 aircraft at an altitude of 20 km. The ground width is approximately 12 km. AVIRIS can also obtain images of low-altitude aircraft with a spatial resolution of 1–4 m. AVIRIS radiation spectroscopy is used to identify, measure and monitor the composition of the

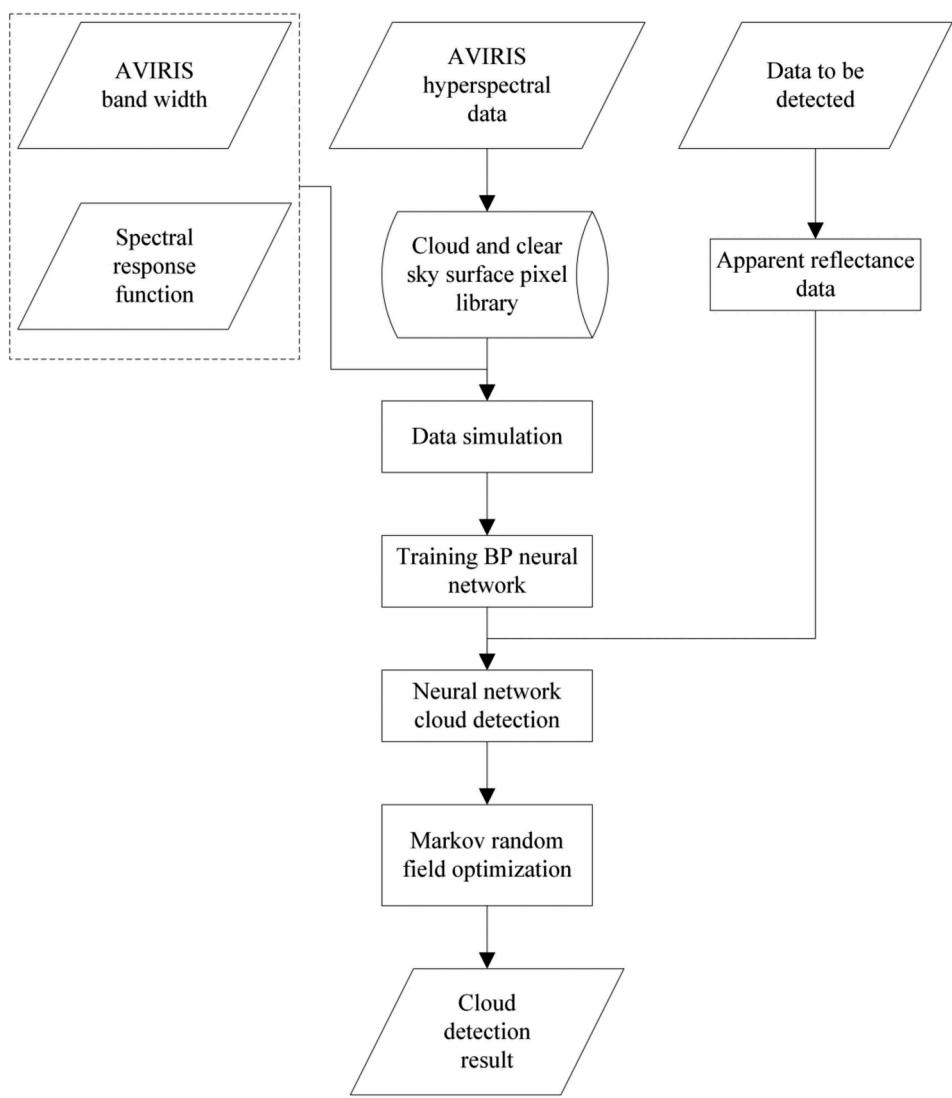


Figure 1. Technical flowchart.

Earth’s surface and atmosphere based on molecular absorption and particle scattering characteristics. AVIRIS has flown and acquired a large amount of data in North America, including Canada, southern Mexico and parts of the United States.

2.2. Establishment of the training sample dataset

To obtain a high-quality training sample dataset, 32 different AVIRIS images from different regions and at different times between 2007 and 2011 were selected. The data contain different types of land surfaces, such as vegetation, buildings, water bodies, and rocks, and clouds of different shapes, such as thin, cirrus, and fragmentary clouds. The selection of samples relies on manual visual interpretation experience. To improve

the accuracy of sample determination, false-colour synthetic images and cirrus bands are used to enhance the surface and cloud information. A total of 305,485 cloud pixels in 398 regions and 461,829 clear-sky pixel samples in 370 regions were selected.

Studies have shown that differences in the thickness and morphology of clouds can lead to large differences in spectral reflectance characteristics (Yuan and Hu 2015; Xu, Wong, and Clausi 2017). Considering the influence of the cloud height and sensor observation angle on variations in the cloud pixel reflectivity, thick clouds, thin clouds, fragmentary clouds and cloud edge regions are included in the cloud pixel sample dataset. In particular, the choice of thin clouds should consider the influence of different underlying surfaces on the cloud reflectance, and thus cloud pixels above vegetation, water bodies, towns, and bare land were selected.

Figure 2 is a typical sample of cloud pixels in a pixel sample dataset, including thick clouds, thin clouds, and fragmentary clouds. The reflection spectrum curves of the three cloud types show that the thick clouds have obviously higher reflectivity than the thin clouds and fragmentary clouds, which is because in the thick cloud occlusion area, the energy received by the optical sensor mainly comes from the reflection of the sun spectrum by clouds. Therefore, there is little difference in thick cloud reflectance on different underlying surfaces. Observing and comparing the thin cloud spectrum curves under different underlying surfaces reveals that the different types of surface cover have a greater influence on thin cloud reflectivity. In the thin cloud region, the energy received by the sensor mainly originates from the partial contribution superposition of the cloud layer and surface reflection. Water bodies have lower reflectivity in the visible

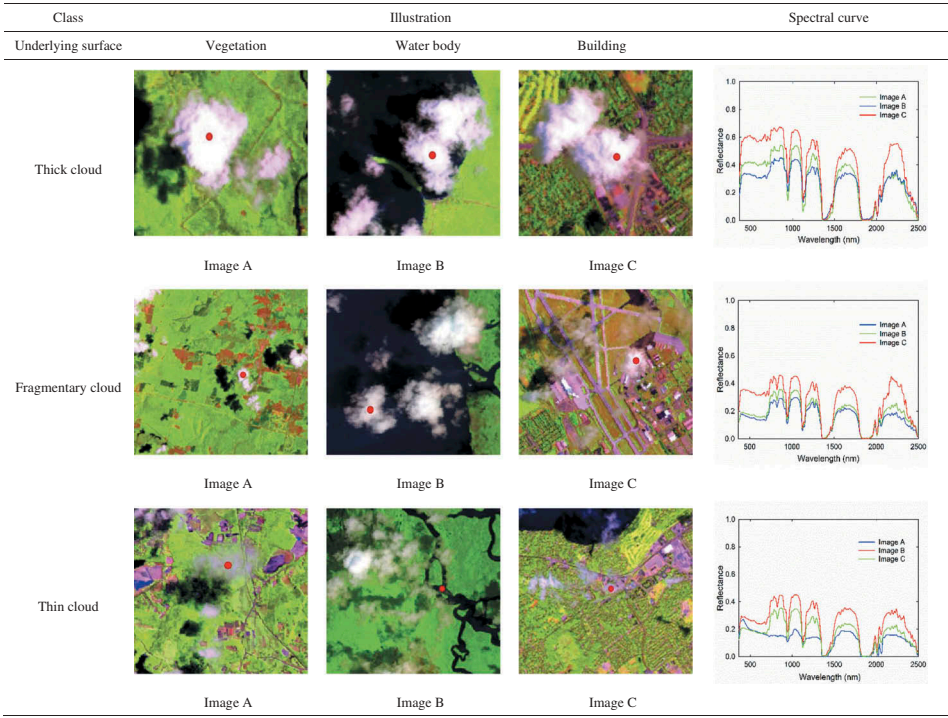


Figure 2. Example sample of cloud pixels.

and near-infrared bands than other typical surfaces, resulting in the thin cloud reflectivity above water being lower than the thin cloud reflectivities on other underlying surfaces.

The selection of the clear-sky pixel dataset mainly considers the difference in reflection characteristics caused by different underlying surface types, and therefore, it is necessary to select a sufficient number of samples, such as cultivated land, forest, grassland, bare land, water bodies, artificial surfaces, cloud shadows and other areas. The spectral reflectance characteristics of some underlying surfaces exhibit large differences over time, including differences in the vegetation growing area, and thus the clear-sky pixel sample dataset is based on multitemporal images. Due to the accumulation of urban populations, urban land surface utilization types are complex, including buildings, hardened roads, and vegetation cover. The spectral reflectance characteristics of these features are different, and the construction surfaces of different materials also exhibit large differences. When the dataset is used, the data are no longer subdivided and are considered to be urban areas. Cloud shadows are another major factor causing differences in surface reflection. The projections of clouds with different thicknesses on the surface is quite different. Combined with the difference in the underlying surface, the reflection characteristics of the cloud shadow area vary widely, and therefore, the clear-sky surface pixels also include a variety of cloud shadows. [Figure 3](#) is a typical area selected by clear-sky pixels and its reflection spectrum curve. Clear-sky pixels are divided into four classes: vegetation, water, town, and cloud shadow according to the type of surface coverage. Based on the reflection spectrum curve, the reflectivities of water and cloud shadow areas are low, and the reflection characteristics are also different due to the different characteristics of water depth and water suspension content. The differences in the types of underlying surface greatly influence the reflection spectrum of the cloud shadow area. The high reflectivity of the artificial surface has a certain similarity with the reflection characteristics of the cloud, which is the main reason for misjudgement.

2.3. Simulation of multispectral remote sensing data

To use the training samples of hyperspectral data to support the deep learning method to detect cloud pixels of multitype sensors, training samples of different remote sensing data must be simulated.

From the perspective of energy, optical remote sensing imaging is actually a sensor process that uses the photoelectric conversion device to quantify and sample received energy. In terms of wide-band multispectral sensor data, the energy of each band is a weighted synthesis of the energy over a range of wavelengths. Hyperspectral remote sensing data have many relatively narrow bands. By simulating the imaging process of optical sensors, multispectral image data can be obtained by weighted synthesis of hyperspectral bands over the wavelength coverage (Verhoef and Bach [2003](#); Knudsen [2005](#); Jia et al. [2014](#); Ting, Li, and Fei [2010](#)). [Figure 4](#) shows a spectral response function comparison of the red, green and blue bands of the AVIRIS and multispectral data. As can be observed from the figure, the band ranges of Landsat 8 OLI, VIIRS and MODIS

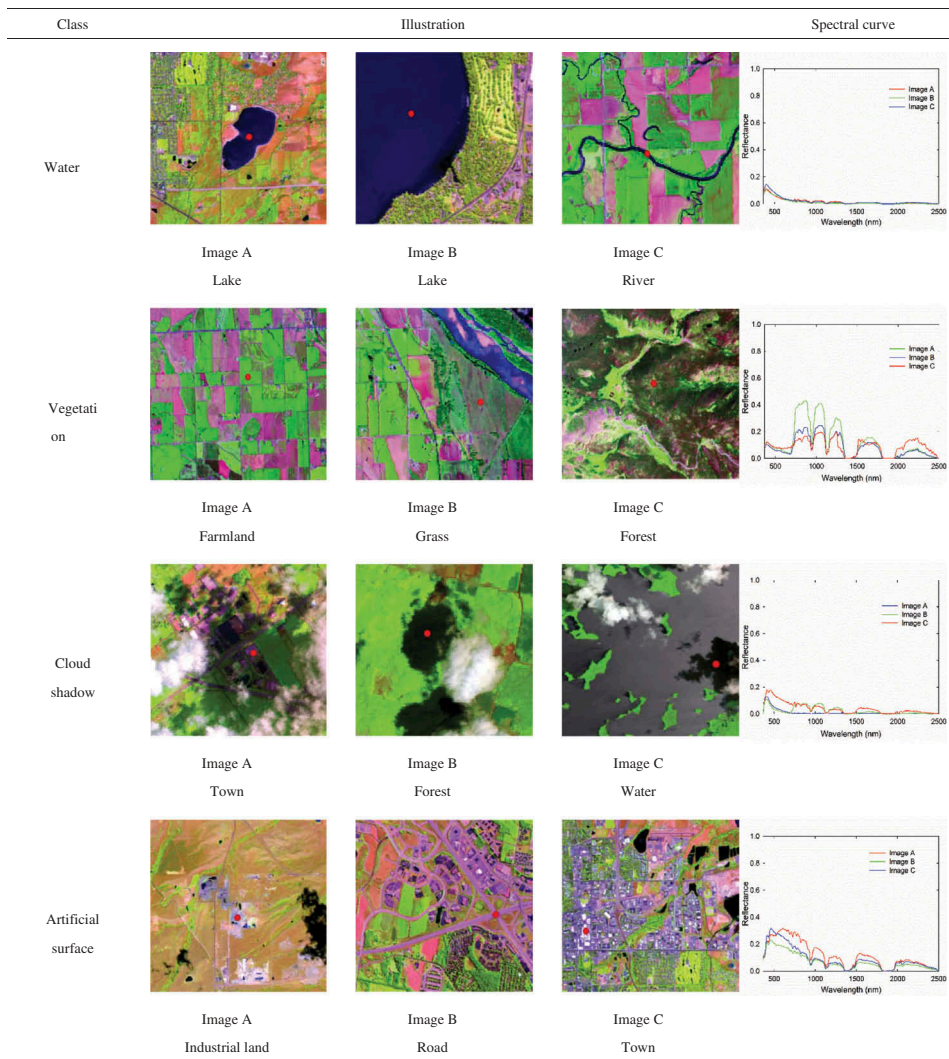


Figure 3. Typical examples of clear-sky pixels in the pixel dataset.

covers a continuous narrow band of multiple AVIRIS data, and therefore, multispectral data can be simulated using hyperspectral data.

The spectral response characteristics of the sensor to the target depend on the spectral range and response weight of the band. Based on the spectral response function of the multispectral sensor, AVIRIS apparent reflectivity and band spacing, multispectral wide-band data simulation can be performed using the following formula:

$$L_i^{\text{MSI}} = \frac{\sum_{j=1}^{N_{\text{HSI}}} \rho(\lambda_{i,j}) \Delta_j L_j^{\text{HSI}}}{\sum_{j=1}^{N_{\text{HSI}}} \rho(\lambda_{i,j}) \Delta_j} \quad i = 1, \dots, N_{\text{MSI}} \quad (1)$$

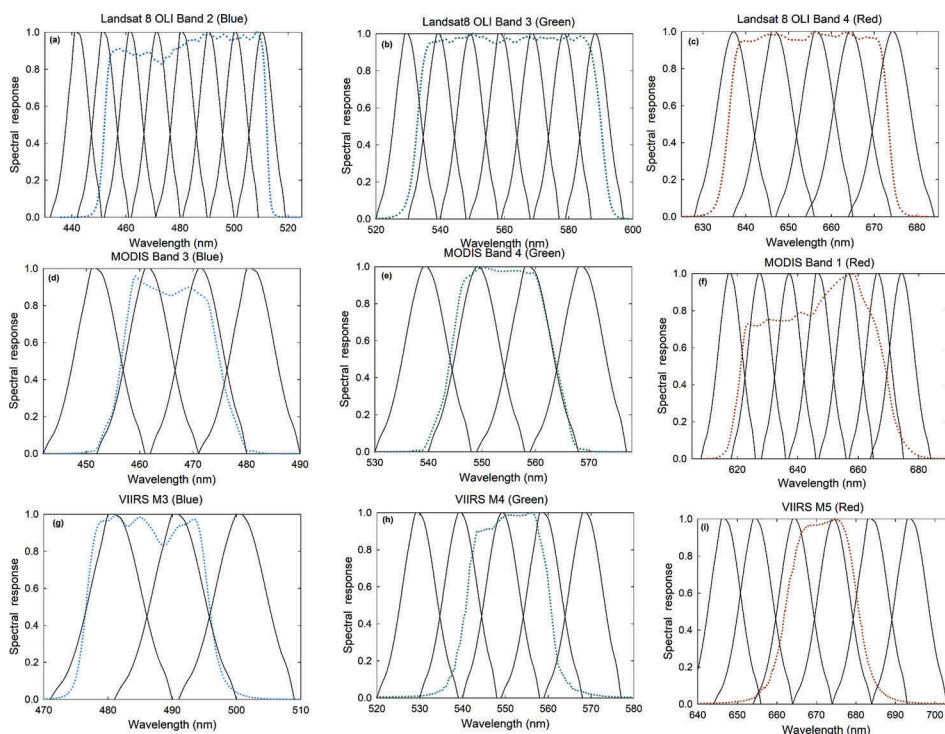


Figure 4. Spectral response function comparison of the visible spectrum between the AVIRIS and multispectral sensors.

where N_{HSI} is the number of hyperspectral sensor channels contained in the broad band spectrum to be simulated, N_{MSI} is the number of channels to be simulated for multispectral sensors, L_i^{MSI} is the apparent reflectance of multispectral data, L_j^{HSI} is the apparent reflectance of AVIRIS data, $\rho(\lambda_{i,j})$ is the spectral response of the multispectral sensor corresponding to the centre wavelength of one AVIRIS band, and Δ_j is the width of the AVIRIS data single band.

3. Deep learning theory and method

Cloud detection is actually a binary segmentation of remote sensing images that realizes the classification of cloud and clear-sky pixels. Deep learning has a good self-learning ability and fault tolerance, which allows for remote sensing image cloud detection. In this chapter, the theoretical knowledge of deep learning algorithms will be introduced.

3.1. BP algorithm

The BP algorithm was proposed by Paul Werbos (1974). BP is a pivotal algorithm in multilayer NN training and is the most commonly used and most effective method in deep learning networks. Figure 5 shows an NN structure consisting of an input layer, an output layer, and an implicit layer.

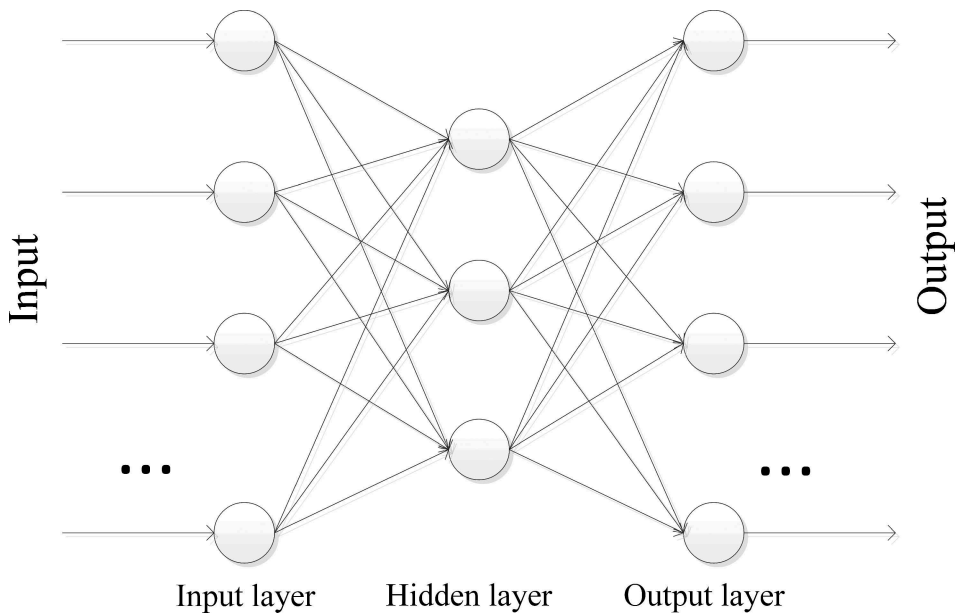


Figure 5. ANN model.

Let a set of training datasets of the NN be (X, Y) , where $X = (x_1, x_2 \dots x_i)^T$ is the set of input feature vector and i is the dimension of the feature vector and also the number of nodes of the input neuron. $Y = (y_1, y_2 \dots y_m)$ is the set of expected value of the output, that is, the goal of learning training. Scalar variable I_j is the j -th input value of the hidden layer node; scalar variable O_j is the j -th output value of the hidden layer node, and the j -th threshold is scalar variable θ_j . Scalar variable W_{ij} is the connection weight between the input layer and hidden layer, scalar variable I_m is the m -th input value of the output layer neurons, and the m -th threshold is scalar variable θ_m . Scalar variable $W_{j,m}$ is the connection weight between the hidden layer and output layer, and $f()$ is the activation function. The BP algorithm training NN includes two main steps of forward propagation and backward feedback:

3.1.1. Forward propagation

In an NN structure, a neuron has multiple inputs and one output, and each neuron input can be a linear combination of the output's upper neuron. Starting from the input layer, the input of the next neuron layer is calculated layer by layer until the output is obtained. This technique is the forward propagation process of the NN. The calculation process of a three-layer NN is as follows.

The input of the hidden layer node is as follows:

$$I_j = \sum_i W_{ij} x_i \quad (2)$$

The output of the hidden layer node is as follows:

$$O_j = f(I_j - \theta_j) = f\left(\sum_i W_{ij} x_i - \theta_j\right) \quad (3)$$

The output of the output layer node is as follows:

$$y_j = f(O_j - \theta_m) = f\left(\sum_j W_{ij}O_i - \theta_m\right) \quad (4)$$

3.1.2. Backward feedback

The BP process of the error is based on the gradient descent method. By correcting the connection weights of the neurons in the network and the node threshold, the specific calculation process is as follows.

The error of the output node is as follows:

$$E = \frac{\sum_m (y_m - O_m)^2}{2} \quad (5)$$

Calculate the output layer and hidden layer node errors separately.

The output layer is as follows:

$$\delta_m = \frac{\partial E}{\partial W_{j,m}} = \frac{\partial E}{\partial O_m} \frac{\partial O_m}{\partial W_{j,m}} \quad (6)$$

The hidden layer is as follows:

$$\delta_j = \frac{\partial E}{\partial W_{i,j}} = \frac{\partial E}{\partial O_j} \frac{\partial O_j}{\partial W_{i,j}} \quad (7)$$

Then, the connection weights and node thresholds are corrected based on the gradient descent method, where scalar variable k is the number of layers.

The weight of the hidden layer is as follows:

$$W_{ij}(k+1) = W_{ij}(k) + \Delta W_{ij} = W_{ij}(k) + \eta_j \sum \delta_m W_{j,m} f'(x) x_i \quad (8)$$

The weight of the output layer is as follows:

$$W_{j,m}(k+1) = W_{j,m}(k) + \Delta W_{j,m} = W_{j,m}(k) + \eta_m (y_m - O_m) f'(x) O_j \quad (9)$$

The threshold of the hidden layer is as follows:

$$\theta_j(k+1) = \theta_j(k) + \Delta \theta_j = \theta_j(k) + \eta_j \sum_m \delta_m W_{j,m} f'(x) \quad (10)$$

The threshold of the output layer is as follows:

$$\theta_m(k+1) = \theta_m(k) + \Delta \theta_m = \theta_m(k) + \eta_m (y_m - O_m) f'(x) \quad (11)$$

The forward propagation and backward feedback steps are iteratively performed until the network error reaches a predetermined target range or the number of trainings reaches the predetermined upper limit.

3.2. Logistic classifier

The logistic classifier is modelled by the Bernoulli distribution and is a two-class method. The logistic classifier can divide the data into 0 and 1 and is suitable for the output layer

in the cloud detection NN. The logistic classification process is simple and mainly includes linear summation, sigmoid function activation, error calculation, and parameter modification. The first two steps are used for judgement, and the last two steps are used for parameter correction. Suppose that there is a training sample set $X = \{X_1, X_2, \dots, X_m\}$, and the sample expects $Y = \{Y_1, Y_2, \dots, Y_m\}$, where $X_i = (x_1, x_2, \dots, x_n)^T$ is the feature vector of the sample; the sample expects that Y_m only belongs to category 0 or category 1. b is the offset for the n -dimensional weight vector \mathbf{w} .

3.2.1. Linear summation

Linear summation is the weighted summation of eigenvectors, where Z is a set of results of linear summation, and W is a set of n -dimensional weight vector \mathbf{w} :

$$Z = W^T \mathbf{X} + b \quad (12)$$

3.2.2. Calculate the activation function

The result of the linear summation Z range is $[-\infty, +\infty]$, and the category of sample X_i cannot be judged based on the Z value; thus, the Z value must be mapped between $[0, 1]$, which is the activation function. The sigmoid function is used here as the activation function, and the formula is as follows:

$$\delta(x) = \frac{1}{1 + e^{-z}} \quad (13)$$

Substituting the summation result into the activation function, the larger the value of Z , the closer $\delta(x)$ is to 1, and the other value is closer to 0. When $\delta(x)$ is greater than 0.5, the sample is judged to belong to class 1. If the value is less than 0.5, the sample is judged to be class 0. The accuracy of the logistic judgement depends on the weight vector \mathbf{w} and the offset amount b . Initially, the value of \mathbf{w} is randomly selected, and the value of b is 0, which must be iteratively modified to achieve a better classification effect.

3.2.3. The error calculation

The actual value of the sample and the expected output error are calculated using the loss function, where C is a scalar variable:

$$C = (\delta(x) - y)^2 / 2 \quad (14)$$

3.2.4. The parameter correction

Similar to the parameter correction method of the NN, the correction of the weight and offset here uses the gradient descent method to obtain the optimal solution via iteration:

The weight correction is as follows:

$$\mathbf{w} = \mathbf{w} - \eta \frac{\partial C}{\partial \mathbf{w}} \quad (15)$$

The offset correction is as follows:

$$b = b - \eta \frac{\partial C}{\partial b} \quad (16)$$

Where scalar variable η represents the learning rate.

4. Implementation of deep learning cloud detection algorithm based on keras

Keras can be backed by Computational network toolkit (CNTK), TensorFlow or Theano, has a highly modular advantage and supports existing common network structures; additionally, Keras is easy to expand. Based on the included NN layer, activation function, initialization, loss function, regularization and other modules, users can build their own networks simply and efficiently in Keras. Thus, a cloud detection BP NN is built in this paper based on Keras, with Theano as the background.

To reduce the influence of human subjective factors on the cloud detection algorithm as much as possible, there is no artificial input feature in this paper. Instead, the apparent reflectance of each band of multispectral data pixels is used as the feature vector, and the inductive ability of the NN is utilized to learn cloud detection rules. Studies have shown that a three-layer (including one hidden layer) NN can approximate any continuous function from n -dimensional to m -dimensional space (Vivone et al. 2014). It is generally believed that increasing the number of hidden layers can enable the network to learn deeper data features, reduce network errors, and improve accuracy, but increasing the number of hidden layers also complicates the network and increases training time. Considering the training efficiency, a deep BP NN with three hidden layers is used in this paper to construct a cloud detection network framework. Currently, there is no scientific, universal determination method for setting the network parameters. A cross-validation method is used in this paper to optimize network parameters, thus ensuring accuracy and improving the training efficiency. Parameter optimization mainly involves the number of neurons, batch values, dropout values, and momentum values.

4.1. Neuron number optimization

Generally, the capacity of the common NN increases with an increase in the number of hidden layer neurons in the network, and more complex NNs can express more complex mapping functions. However, more neurons can classify more complex data but may also result in overfitting of training data; that is, there is a strong classification ability for data noise. The model error during training is low, but the actual classification effect of the network is poor, and as the number of neurons increases, the network training time will also increase significantly. Currently, there is no definitive method to determine the number of neurons in the network. In this paper, we use the cross-validation method to determine the optimal number of nodes in the hidden layer through many experiments.

During the process of cross-validation, 20% of the sample data are randomly selected as verification data. Table 1 reports the variation in the data error with the number of neurons in the network training process using Landsat 8 OLI simulation data. The dropout value in the network is 0.5, the batch is 128, and the momentum is 0.9. The iteration of the network is stopped when the training error is less than 0.01 or the

Table 1. Verification error for different numbers of neurons.

Number of neurons	2	3	4	5	6	7	8	9	10	11
Error	0.16	0.147	0.13	0.105	0.081	0.064	0.051	0.039	0.03	0.029
Number of neurons	12	13	14	15	16	17	18	19	20	21
Error	0.029	0.027	0.023	0.024	0.024	0.02	0.019	0.024	0.029	0.026
Number of neurons	22	23	24	25	26	27	28	29	30	
Error	0.02	0.018	0.017	0.016	0.016	0.018	0.02	0.019	0.018	

number of iterations is greater than 1000. As shown in [Table 1](#), when the number of neurons changes from 2 to 10, the verification data error rate drops sharply. When the number of neurons in each hidden layer is greater than 10, the network error changes slightly, but the change is very minor, and thus in the optimization of other parameters, the number of neurons in the three hidden layers is taken to be 10.

4.2. Batch value optimization

The NN usually stops training when the loss error or the training number reaches a predetermined threshold. In NN training, the sample data are usually randomly divided into small datasets of size batches before each iteration and then input into the network as a small dataset to complete the training (Deng et al. 2009; Sato, Nishimura, and Yokoi 2015). There are three advantages to this method: training of small datasets synthesized by multiple samples can effectively suppress the sample noise influence; this method results in improved memory utilization efficiency, making full use of calculating resources, reducing the number of backward feedback required for all training samples to iterate once, and immediately improving the network training speed; as the batch increases, the direction of the gradient decreases and becomes more accurate, resulting in only minor training oscillation.

Notably, when the batch value is increased to a certain extent, the ability to suppress noise is no longer improved, and with an increasing batch value, the memory space is occupied, resulting in a decline in training efficiency. In this paper, different batch values are used for training. The number of hidden layer neurons in the network is 10, the dropout is 0.5, the momentum is 0.9, and when the training error is less than 0.01 or the number of iterations is greater than 1000, the network iteration is terminated. As reported in [Table 2](#), when the batch value changes from 100 to 600, the test error variation is small; after 600, the error slowly increases, and the network convergence speed also increases.

4.3. Dropout value optimization

When training the network model, the prediction results may be far from the training results, which is the model overfitting phenomenon. In practical applications, a network with overfitting cannot be used normally. Typically, during NN training, dropout is

Table 2. Verification error for different batch values.

Batch	100	200	300	400	500
Error	0.028	0.027	0.020	0.037	0.025
Batch	600	700	800	900	1000
Error	0.033	0.056	0.061	0.129	0.17

introduced to solve this problem. The principle of dropout is that in the training process of the NN, the hidden layer temporarily ignores the neurons from the network according to a certain probability, but these weights are saved in the network. When the next sample dataset is trained, all the neurons will be randomly enabled. As a network with n neurons, if you dropout with a probability of 0.5, then there will be $2n$ sparse networks (Srivastava et al. 2014), which can greatly increase the speed of the network training efficiency when other parameters are unchanged. The dropout mechanism makes it unnecessary for two neurons to be activated simultaneously in a single dropout network, reducing the complex coadaptation between neurons and improving the generalization ability of the model (Russakovsky et al. 2014; Szegedy, Liu, and Jia 2015). This method is simple to implement and is widely used because the method is effective in practical applications.

In this paper, the proportion of neurons in dropout is 0.5. In theory, the generalization ability of the NN is the strongest, and the effect is better (Wilson et al. 2017).

4.4. Momentum value optimization

The NN usually uses the gradient descent method to correct the weights, minimizes the loss function, and achieves the network training purpose. Stochastic gradient descent (SGD) is a popular method at present. The SGD method is combined with the batch processing method; that is, each time the weight is updated, the data of the batch number is used instead of the entire dataset (Besada-Portas et al. 2010). When there are more training data, the pressure of the computer can be reduced, and when the training data have greater redundancy, the convergence of the random gradient can increase the speed of the network convergence. Although the SGD method can improve the training efficiency to a certain extent, there is also a disadvantage. The network weight update is based on the current sample data of a batch. When the sample data fluctuate greatly, the weight update has a large fluctuation, which reduces the network training efficiency. To improve this situation, momentum is introduced into the NN model, which ensures the weight update direction inherits the previous sample direction; this increases the stability to some extent and can effectively solve the problem that the training model falls into local optimal. Momentum values are usually between 0.5 and 0.9. In this paper, momentum values are verified at intervals of 0.1. The results are reported in Table 3; it can be observed that the verification error varies little with the momentum.

5. Cloud detection optimization based on a markov random field

Markov random fields (MRFs) (Geman and Geman 1987) are based on the theory of probability and stochastic processes, and MRFs are one of the commonly used statistical learning models in the fields of computer vision, natural language processing, and pattern recognition. Remote sensing image is actually discrete sampling by a sensor of the real world. The image is

Table 3. Verification error for different momentum values.

Momentum	0.5	0.6	0.7	0.8	0.9
Error	0.036	0.035	0.03	0.035	0.04

represented in the form of a two-dimensional matrix. The attribute of the pixel is only related to the several pixels adjacent to the pixel. This correlation is very important for the image analysis. As a probabilistic graph model for modelling image space structure, the MRF model has few parameters and strong spatial constraints (Dosovitskiy, Fischery, and Ilg et al. 2015), which has incomparable advantages in remote sensing image classification. In this paper, the MRF model is used to optimize the cloud detection results of the BP NN.

Image segmentation based on the MRF has two main features. First, full use is made of the spatial correlation between image pixels, and this correlation is reflected in the form of energy in the model; second, the label assignment process of images is usually expressed as the minimizing global energy process. As a solution to the optimization problem of discrete sets, image segmentation cannot be solved by solving the analytic function. It is necessary to search for the optimal solution or a local optimal solution by means of an iterative optimization algorithm. In this paper, the optimization of cloud detection results based on an MRF is implemented based on the iterated conditional mode (ICM). The ICM was proposed by Besag (1986) in 1986 and can be used to calculate the maximum posterior probability when performing image segmentation. The ICM optimizes the target with the minimum global energy and iteratively updates the classification label to achieve local energy reduction and point-by-point estimation of the image to complete image segmentation. The algorithm is simple to calculate and has a fast convergence speed. However, before estimating the global energy parameters, ICM requires original classification label data as the initial value, and the final result has a great relationship with the initial segmentation result.

6. Results and analysis

To verify the accuracy of the proposed algorithm, cloud detection applications are conducted for multiple landscapes of Landsat 8, MODIS and NPP VIIRS in different regions from 2013 to 2017. The cloud coverage area is selected by visual interpretation, and the cloud detection accuracy of this method is evaluated according to four indicators: cloudy pixel correct rate (CCR), cloudy pixel omission error (COE), clear-sky pixel correct rate (SCR), and clear-sky pixel omission error (SOE).

6.1. Landsat 8 OLI cloud test results

Figure 6 shows the cloud detection results for Landsat 8 OLI images, which contain various types of land surface, including water (samples b, f), vegetation (samples a, d, h), artificial surface (samples b, c) and bare ground (samples e, f, g), part of each the image are covered by thick clouds, thin clouds, or fragmentary clouds. The images in the left column are the false-colour image composed of bands 5, 4 and 3 of the Landsat 8 OLI data. The images in the middle column are the results of cloud detection from Fmask method developed by Zhu and Woodcock (2012), and the right column are from the method in this paper. Cloudy pixels are marked white, and clear pixels are marked black.

Compared with Fmask method, the method presented in this paper shows higher accuracy in cloud detection over most surface types. For the traditional threshold method, cloud detection over bright surface and thin cloud detection over dark surface usually have low accuracy. Figure 6 shows that there are much large artificial surface areas (samples b, c) are misjudged as cloud pixels by the Fmask method for such kind of areas have high reflectance in

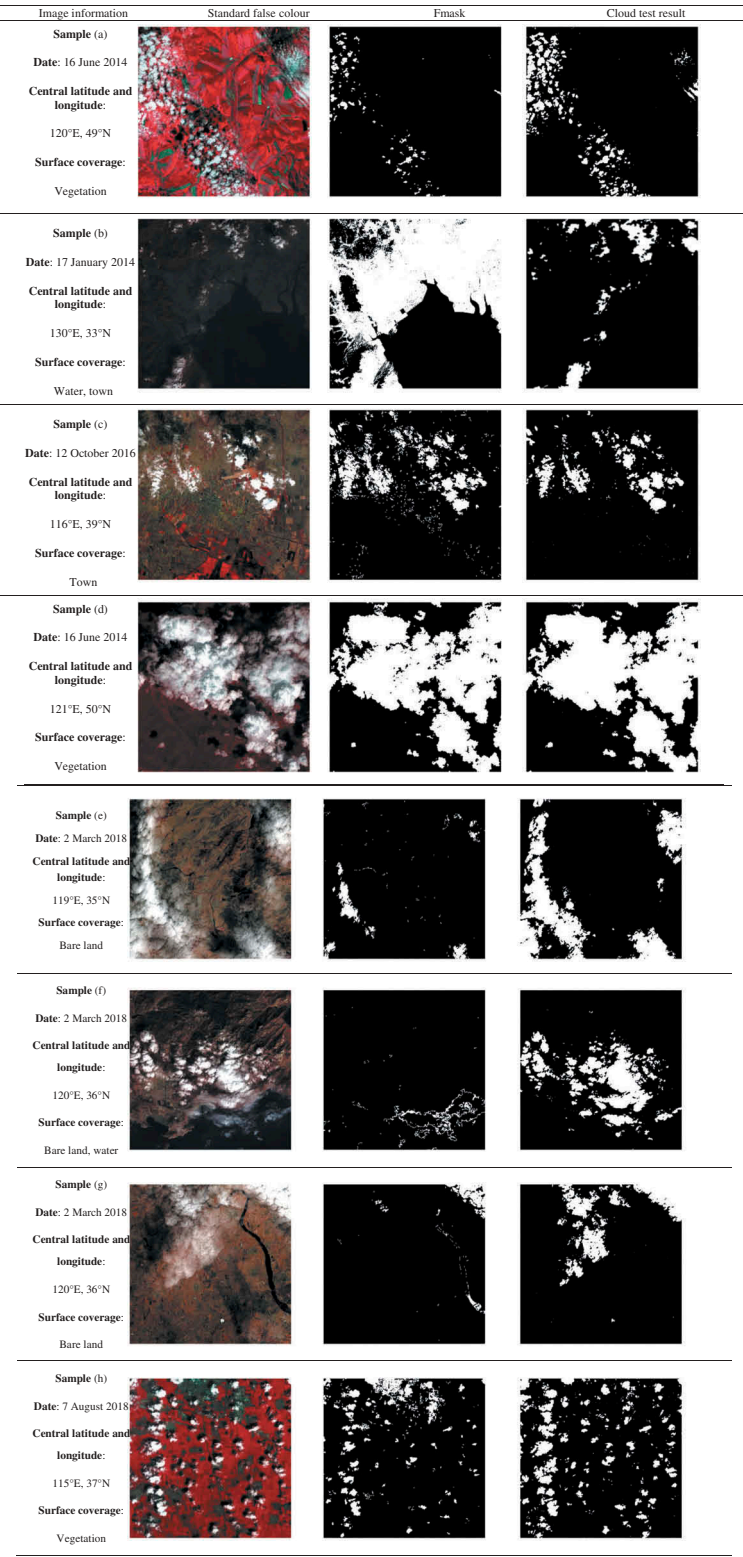


Figure 6. The cloud detection results for Landsat 8 OLI images. The image in the left column is the false colour image composed of bands 5, 4 and 3. The image in the middle column is the result of cloud detection from Fmask method, and the right column is from the method in this paper. Cloudy pixels are marked white, and clear pixels are marked black.




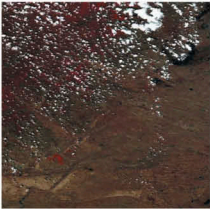

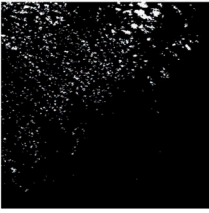
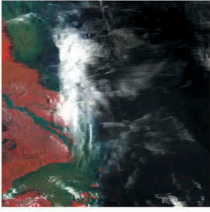

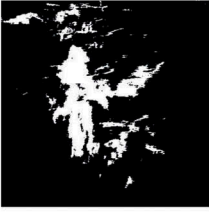
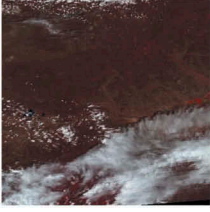


Image information	Standard false colour	VCCLO	Cloud test result
Sample (a)			
Date: 1 June 2015			
Central latitude and longitude:			
106°E, 40°N			
Surface coverage:			
Vegetation, bare land			
Sample (b)			
Date: 1 June 2015			
Central latitude and longitude:			
121°E, 44°N			
Surface coverage:			
Vegetation, bare land			
Sample (c)			
Date: 2 June 2015			
Central latitude and longitude:			
122°E, 31°N			
Surface coverage:			
Water, vegetation			
Sample (d)			
Date: 3 June 2015			
Central latitude and longitude:			
118°E, 43°N			
Surface coverage:			
Vegetation			

Figure 7. Cloud detection results of VIIRS images.

the visible and near infrared wavelength. Similar problems also occur in bare ground areas (samples e, f, g), which also show bright features in visible and near-infrared bands. In contrast, these problems have not been evident with the method proposed in this paper. The proposed method is also superior to Fmask cloud detection method in detecting thin clouds over bright surface. Most of the thin clouds that cover the water (samples b, f) have been omitted from the Fmask method, but they can be effectively identified by the method in this paper. For cloud detection over vegetation, both methods have high detection capability, and the detection of thick, thin and fragmented clouds shows high accuracy. (samples a, d, h).

6.2. VIIRS and MODIS data cloud detection results

The results of cloud detection using VIIRS data and MODIS data are shown in Figures 7 and 8. The left column is a false colour images of two types of images, for VIIRS is composed of bands

7, 5, and 4, and MODIS is composed of bands 2, 1, and 4. It can be seen that images with various land types, include vegetation, bare land, water, covered by thick, thin and fragmentary clouds are selected to verify the effectiveness of the method. The cloud products provided by NASA (National Aeronautics and Space Administration) are used for comparison. The middle columns of [Figures 7](#) and [8](#) are the VIIRS Cloud Cover Layers Environmental Data Record (EDR) data (VCCLO), and MODIS cloud mask product (MOD 35), respectively. The right columns of both [Figures 7](#) and [8](#) are the cloud detection results from the method of this paper. MODIS has 36 bands in visible to thermal infrared bands and VIIRS has 22 bands in the same wavelength range, reflectance difference and brightness temperature difference in multiple bands are used in the process of cloud products generation, so the cloud detection in most of the surface coverage areas can achieve high accuracy. However, due to the limitations of the threshold cloud detection method itself, there are still obvious cloud detection errors in some bright areas, as well as in the areas of thin cloud coverage, fragmentary cloud coverage or cloud edge area. As can be observed from [Figures 7](#) and [8](#), compared with the cloud detection results with the method proposed in this paper, in cloud detection products of VCCLO and MOD 35, there are obvious phenomena of thin cloud, broken cloud and cloud edge leaking over various surface types, such as in the areas of covered by bare land, water, vegetation, etc. The above problems have been significantly improved in the methods proposed in this paper.

6.3. Quantitative evaluation

To accurately evaluate the applicability of the algorithm for different sensor data, the results of the test are quantitatively analysed in this paper. The images in [Figure 6–8](#) are visually interpreted to select the cloud coverage area, which is compared with the cloud detection results of the algorithm, and the CCR, COE, SCR, and SOE values are calculated. There are four groups of classifications for each pixel in [Table 4](#).

The calculation formulas of the four detection accuracy indicators are as follows:

$$CCR = \frac{TP}{TP + FN} \tag{17}$$

$$SOE = \frac{FP}{FP + TN} \tag{18}$$

$$SCR = \frac{TN}{FP + TN} \tag{19}$$

$$COE = \frac{FN}{TP + FN} \tag{20}$$

CCR and SCR reflect the performances of cloud detection algorithms. COE indicates the ratio of cloud pixels that are misjudgements regarding the total number of cloud pixels,

Table 4. Contingency table.

	Recognition as cloud pixel in algorithm	Recognition as clear- sky pixel in algorithm
Cloud pixel in verify image	TP	FN
Clear-sky pixel in verify image	FP	TN

reflecting the algorithm's underestimation of the cloud. SOE indicates the ratio of the clear-sky pixels in the gap to the total number of clear-sky pixels, reflecting the degree of cloud overestimation by the algorithm.

Table 5 presents a quantitative evaluation of the results of three satellite data clouds in Figure 6–8. Based on the accuracy confusion matrix, the algorithm has high detection accuracy for Landsat 8 OLI, NPP VIIRS and MODIS. The recognition rate of cloud pixels is nearly greater than 0.9, and the false positive rate in clear-sky is also at a lower level. By separately analysing the accuracy indicators of each group of data, the shape of the cloud and type of underlying surface are known to have a certain influence on the accuracy of cloud detection. In the data with more thin clouds, the detection accuracy is lower overall, and the accuracy values for thick clouds and fragmentary clouds are greater. The type of underlying surface also has a certain impact on the results of cloud detection. The urban area has a higher false judgement rate of cloud pixels than high vegetation coverage areas, such as farmland and forests. Based on the accuracy verification, the cloud detection method proposed in this paper has achieved good detection results for Landsat 8 OLI, NPP VIIRS and MODIS.

7. Conclusion and discussion

Based on AVIRIS hyperspectral data, automatic cloud detection of multiple multispectral sensors is realized in this paper, and Landsat 8 OLI, Terra MODIS and NPP VIIRS data are used to verify the detection accuracy of the proposed algorithm. The results demonstrate that the cloud detection algorithm proposed in this paper has better detection results, and the overall accuracy of cloud detection is greater than 90%. The main research contributions of this paper are summarized as follows:

(1) Training samples are constructed based on AVIRIS data to support cloud detection with a deep learning method for multiple satellite data. Through visual interpretation, different types of cloud pixels and clear-sky pixels are manually selected from 32 different AVIRIS images. The cloud and clear-sky pixel dataset of the corresponding sensor are simulated using the spectral response function of the multispectral sensor

Table 5. Quantitative evaluation of cloud detection results (%).

Sensor	Table number	Number	CCR	SCR	SOE	COE
Landsat 8 OLI	Figure 6	a	0.920	0.981	0.019	0.080
		b	0.972	0.989	0.011	0.028
		c	0.970	0.967	0.033	0.030
		d	0.967	0.971	0.029	0.033
		e	0.929	0.988	0.012	0.071
		f	0.960	0.979	0.021	0.040
		g	0.911	0.991	0.009	0.089
		h	0.964	0.983	0.017	0.036
NPP VIIRS	Figure 7	a	0.901	0.894	0.106	0.099
		b	0.963	0.887	0.113	0.037
		c	0.884	0.910	0.090	0.116
		d	0.941	0.886	0.114	0.059
MODIS	Figure 8	a	0.912	0.939	0.061	0.088
		b	0.889	0.958	0.042	0.111
		c	0.929	0.944	0.056	0.071
		d	0.875	0.930	0.07	0.125
		e	0.948	0.945	0.055	0.052

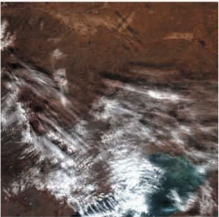
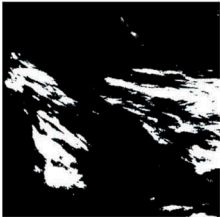
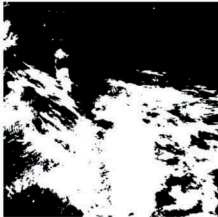
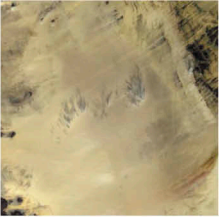

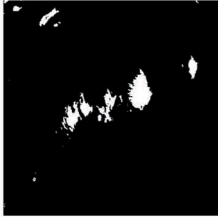
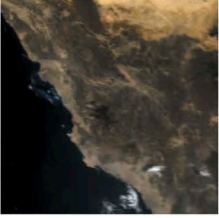
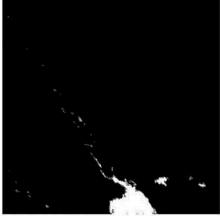

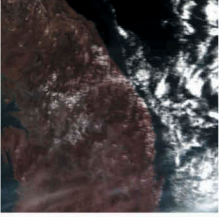

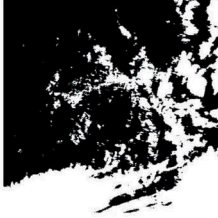
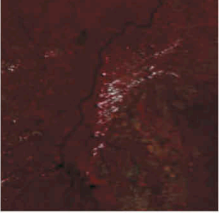

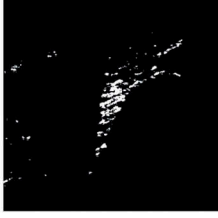
Image information	Standard false colour	MOD35	Cloud test result
Sample (a) Date: 25 December 2017 Central latitude and longitude : 123°E, 40°N Surface coverage : Bare land, water			
Sample (b) Date: 6 February 2016 Central latitude and longitude : 10°E, 20°N Surface coverage : Desert			
Sample (c) Date: 10 August 2016 Central latitude and longitude : 38°E, 25°N Surface coverage : Water, bare soil			
Sample (d) Date: 26 February 2017 Central latitude and longitude : 128°E, 37°N Surface coverage : Water, vegetation			
Sample (e) Date: 15 July 2017 Central latitude and longitude : 53°E, 10°N Surface coverage : Vegetation			

Figure 8. Cloud detection results for MODIS scenes.

and the band width of the hyperspectral sensor. This method greatly reduces the workload of training sample selection for cloud detection using deep learning methods with different satellite images. The method provides great convenience for fast application of deep learning algorithms in different satellite data cloud detection methods.

(2) A deep BP NN is built based on the Keras deep learning framework. The simulated reflectance values of the cloud and clear-sky pixel datasets of the multispectral sensor are used as the feature vectors of the BP NN and input into the network for training. The number of hidden layer nodes, batch value and momentum parameters of the NN are

optimized to improve the generalization ability of the network model, and the training time is reduced without reducing the training effect.

(3) The cloud detection results are optimized based on an MRF. The spatial constraint relationship of the pixels in the image is introduced into the cloud detection algorithm. In this paper, the MRF is implemented using the iterative conditional model algorithm to optimize the cloud detection results.

Although the method presented in this paper has shown good accuracy and stability in the preliminary cloud detection, the following problems still exist in this method, which needs to be gradually improved in the follow-up work. 1) Data used as learning samples for cloud detection with deep learning method are simulated using the same hyperspectral data, the accuracy of simulated data will directly affect the accuracy of cloud detection. Accurate evaluation of the difference between simulated data and real data is of great significance to understand the potential error of this method. Unfortunately, this work has not been carried out because airborne data are used to simulate satellite data in this paper, there are many difficulties in assessing actual radiation differences due to height differences and spatial matching of these two kinds of data. Thus, this paper is only to choose the best simulation method to get high credible simulated satellite data; 2) limited by the spatial and temporal distribution of current AVIRIS data, the existing training data do not include all possible surface types and cloud states, which will also introduce some errors and uncertainties into the cloud detection results when using the deep learning method.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

- Ackerman, S., and R. Frey. 2015. *MODIS Atmosphere L2 Cloud Mask Product*. USA: NASA MODIS Adaptive Processing System, Goddard Space Flight Center. doi:[10.5067/MODIS/MOD35_L2.006](https://doi.org/10.5067/MODIS/MOD35_L2.006).
- Ball, J. E., D. T. Anderson, and C. S. Chan. 2017. "A Comprehensive Survey of Deep Learning in Remote Sensing: Theories, Tools and Challenges for the Community." *Journal of Applied Remote Sensing* 11 (4): 042609. doi:[10.1117/1.JRS.11.042609](https://doi.org/10.1117/1.JRS.11.042609).
- Besada-Portas, E., S. M. Plis, J. M. D. L. Cruz, et al., 2010. "Adaptive Parallel/Serial Sampling Mechanisms for Particle Filtering in Dynamic Bayesian Networks." European Conference on Machine Learning and Knowledge Discovery in Databases, Barcelona, Spain. Springer-Verlag.
- Besag, J. 1986. "On the Statistical Analysis of Dirty Pictures." *Journal of the Royal Statistical Society* 48 (3): 259–302. doi:[10.1080/02664769300000059](https://doi.org/10.1080/02664769300000059).
- Deng, J., R. Socher, L. Fei-Fei, W. Dong, K. Li, and L.-J. Li, 2009. "ImageNet: A Large-scale Hierarchical Image Database," 2009 IEEE Conference on Computer Vision and Pattern Recognition(CVPR), Miami, FL, USA, pp. 248–255. doi:[10.1037/a0016184](https://doi.org/10.1037/a0016184)
- Dosovitskiy, A., P. Fischery, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox. 2015. "FlowNet: Learning Optical Flow with Convolutional Networks." IEEE International Conference on Computer Vision, Santiago, Chile, pp. 2758–2766.
- Geman, S., and D. Geman. 1987. "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images." *Readings in Computer Vision* 20 (5–6): 25–62. doi:[10.1080/02664769300000058](https://doi.org/10.1080/02664769300000058).

- Graziotin, D., and P. Abrahamsson. 2013. "A Web-based Modeling Tool for the SEMAT Essence Theory of Software Engineering." *Journal of Open Research Software* 1 (1): e4. doi:10.5334/jors.ad.
- Green, R. O., M. L. Eastwood, C. M. Sarture, T. G. Chrien, M. Aronsson, B. J. Chippendale, J. A. Faust, et al. 1998. "Imaging Spectroscopy and the Airborne Visible/infrared Imaging Spectrometer (AVIRIS)." *Remote Sensing of Environment* 65 (3): 227–248. doi:10.1016/S0034-4257(98)00064-9.
- Harshvardhan, D. A. Randall, and T. G. Corsetti. 1989. "Earth Radiation Budget and Cloudiness Simulations with a General Circulation Model." *Journal of the Atmospheric Sciences* 46 (13): 1922–1942. doi:10.1175/1520-0469(1989)046<1922:ERBACS>2.0.CO;2.
- Jedlovec, G. J., S. L. Haines, and F. J. Lafontaine. 2008. "Spatial and Temporal Varying Thresholds for Cloud Detection in Goes Imagery." *IEEE Transactions on Geoscience and Remote Sensing* 46 (6): 1705–1717. doi:10.1109/TGRS.2008.916208.
- Jia, G., H. Zhao, H. Shang, C. Lou, and C. Jiang. 2014. "Pixel-size-varying Method for Simulation of Remote Sensing Images." *Journal of Applied Remote Sensing* 8 (1): 083551. doi:10.1117/1.JRS.8.083551.
- King, M. D., S. Platnick, W. P. Menzel, S. A. Ackerman, and P. A. Hubanks. 2013. "Spatial and Temporal Distribution of Clouds Observed by Modis Onboard the Terra and Aqua Satellites." *IEEE Transactions on Geoscience and Remote Sensing* 51 (7): 3826–3852. doi:10.1109/TGRS.2012.2227333.
- Knudsen, T. 2005. "Technical Note: Pseudo Natural Colour Aerial Imagery for Urban and Suburban Mapping." *International Journal of Remote Sensing* 26 (12): 2689–2698. doi:10.1080/01431160500057020.
- Li, Z., H. Shen, H. Li, G. Xia, P. Gamba, and L. Zhang. 2017. "Multi-feature Combined Cloud and Cloud Shadow Detection in Gaofen-1 Wide Field of View Imagery." *Remote Sensing of Environment* 191: 342–358. doi:10.1016/j.rse.2017.01.026.
- Liou, R. J., M. R. Azimi-Sadjadi, D. L. Reinke, T. H. Vonder-Haar, and K. E. Eis. 1994. "Detection and Classification of Cloud Data from Geostationary Satellite Using Artificial Neural Networks." *IEEE World Congress on IEEE International Conference on Neural Networks* 7 (7): 4327–4332. doi:10.1109/ICNN.1994.374963.
- MODIS Characterization Support Team. 2009. *MODIS Level 1B Product User's Guide*. USA: NASA/Goddard Space Flight Center. doi:10.5067/MODIS/MOD021KM.006.
- Papadomanolaki, M., M. Vakalopoulou, S. Zagoruyko, and K. Karantza. 2016. "Benchmarking Deep Learning Frameworks for the Classification of Very High Resolution Satellite Multispectral Data." *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 3 (7): 12–19. doi:10.5194/isprs-annals-III-7-83-2016.
- Richard, L. B. 1994. "cloud Classification of AVHRR Imagery in Maritime Regions Using a Probabilistic Neural Network." *Journal of Applied Meteorology* 33 (8): 909–918. doi:10.1175/1520-0450(1994)033<0909:CCOAI>2.0.CO;2.
- Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, et al. 2014. "ImageNet Large Scale Visual Recognition Challenge." *International Journal of Computer Vision* 115 (3): 211–252. doi:10.1007/s11263-015-0816-y.
- Sato, I., H. Nishimura, and K. Yokoi. 2015. *APAC: Augmented PAttern Classification with Neural Networks*. Computer Science.
- Saunders, R. W., and K. T. Kriebel. 1988. "An Improved Method for Detecting Clear Sky and Cloudy Radiances from Avhrr Data." *International Journal of Remote Sensing* 9 (1): 123–150. doi:10.1080/01431168808954841.
- Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. 2014. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *Journal of Machine Learning Research* 15 (1): 1929–1958.
- Stowe, L. L., P. A. Davis, and P. E. McClain. 1999. "Scientific Basis and Initial Evaluation of the CLAVR-1 Global Clear/Cloud Classification Algorithm for Advanced Very High Resolution Radiometer." *Journal of Atmospheric and Oceanic Technology* 16: 656–681. doi:10.1175/15200426(1999)016<0656:SBAIEO>2.0.CO;2.

- Stowe, L. L., E. P. McClain, R. Carey, P. Pellegrino, G. G. Gutman, P. Davis, and S. Hart. 1991. "Global Distribution of Cloud Cover Derived from Noaa/avhrr Operational Satellite Data." *Advances in Space Research* 11 (3): 51–54. doi:[10.1016/0273-1177\(91\)90402-6](https://doi.org/10.1016/0273-1177(91)90402-6).
- Sun, L., X. Mi, J. Wei, J. Wang, X. Tian, H. Yu, and P. Gan. 2017. "A Cloud Detection Algorithm-generating Method for Remote Sensing Data at Visible to Short-wave Infrared Wavelengths." *ISPRS Journal of Photogrammetry and Remote Sensing* 124 :70–88. doi:[10.1016/j.isprsjprs.2016.12.005](https://doi.org/10.1016/j.isprsjprs.2016.12.005).
- Szegedy, C., W. Liu, and Y. Jia, 2015. "Going Deeper with Convolutions." 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Boston, MA, USA.
- Ting, Z., D. J. Li, and W. Fei. 2010. "Simulation of Image Multi-Spectrum Using Field Measured Endmember Spectrum." *Spectroscopy & Spectral Analysis* 30 (11): 2889–2893. doi:[10.3964/j.issn.1000-0593\(2010\)11-2889-05](https://doi.org/10.3964/j.issn.1000-0593(2010)11-2889-05).
- Verhoef, W., and H. Bach. 2003. "Simulation of Hyperspectral and Directional Radiance Images Using Coupled Biophysical and Atmospheric Radiative Transfer Models." *Remote Sensing of Environment* 87 (1): 23–41. doi:[10.1016/S0034-4257\(03\)00143-3](https://doi.org/10.1016/S0034-4257(03)00143-3).
- Vivone, G., P. Addesso, R. Conte, M. Longo, and R. Restaino. 2014. "A Class of Cloud Detection Algorithms Based on A Map-mrf Approach in Space and Time." *IEEE Transactions on Geoscience & Remote Sensing* 52 (8): 5100–5115. doi:[10.1109/TGRS.2013.2286834](https://doi.org/10.1109/TGRS.2013.2286834).
- Werbos, P. 1974. "Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Science." Ph.d.dissertation., Harvard University.
- Wilson, A. C., R. Roelofs, M. Stern, N. Srebro, and B. Recht. 2017. "The Marginal Value of Adaptive Gradient Methods in Machine Learning.,, 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.
- Xu, L., A. Wong, and D. A. Clausi. 2017. "A Novel Bayesian Spatial-temporal Random Field Model Applied to Cloud Detection from Remotely Sensed Imagery." *IEEE Transactions on Geoscience and Remote Sensing* 1–12. doi:[10.1109/TGRS.2017.2692264](https://doi.org/10.1109/TGRS.2017.2692264).
- Yuan, Y., and X. Hu. 2015. "Bag-of-words and Object-based Classification for Cloud Extraction from Satellite Imagery." *IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing* 8 (8): 4197–4205. doi:[10.1109/JSTARS.2015.2431676](https://doi.org/10.1109/JSTARS.2015.2431676).
- Zhu, Z., and C. E. Woodcock. 2012. "Object-based Cloud and Cloud Shadow Detection in Landsat Imagery." *Remote Sensing of Environment* 118 (6): 83–94. doi:[10.1016/j.rse.2011.10.028](https://doi.org/10.1016/j.rse.2011.10.028).
- Zhu, Z., and C. E. Woodcock. 2014. "Automated Cloud, Cloud Shadow, and Snow Detection in Multitemporal Landsat Data: An Algorithm Designed Specifically for Monitoring Land Cover Change." *Remote Sensing of Environment* 152: 217–234. doi:[10.1016/j.rse.2014.06.012](https://doi.org/10.1016/j.rse.2014.06.012).