International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 2 Feb 2022, pp: 1266-1278 www.ijaem.net ISSN: 2395-5252

Cloud detection based on Densely Connected Attention Networkfor remote sensing imagery

Chong Zhang [1]*, Zheng Sun[1], Bicheng Shi [2] and Jing Xu[1]
1 School of automation, Wuxi University, Wuxi 214105, China
2 Nanjing fenghuotiandi Communication Technology Co., Ltd, Nanjing 210000, China Corresponding Author:

Submitted: 10-02-2022 Revised: 22-02-2022 Accepted: 25-02-2022

ABSTRACT: Cloud detection of satellite cloud images is the key to the analysis and application of various remote sensing data. In the detection process based on deep learning, with the deepening of deep learning network, although it can effectively extract features, there are some problems in training, such as gradient disappearance, low training efficiency and difficult optimization. To solve these problems, this paper uses the densely connected convolutional neural network model to realize the cloud detection of multispectral satellite cloud images. In view of the low utilization efficiency of the densely connected network in the feature channel, it effectively combines the attention mechanism with the densely connected convolutional neural network to improve the utilization of the features of the model. In order to improve the performance of the model, this paper uses shared memory to reduce the display memory of the model in the training process, so that a deeper network structure can be used under the condition of limited computing resources, which greatly improves the performance of the network model; Combined with the latest AdaBound optimizer, the learning rate is dynamically trimmed, so that the model has both the fast convergence speed in the early stage of Adam training and the excellent performance in the later stage of SGD training. The experimental results show that this method has a good performance for the application of cloud image detection with complex spectral information.

KEYWORDS: Remote sensing ; Densely connection network ; Redundancy ; SGD ; DeepLearning

I. INTRODUCTION

Cloud cluster recognition with high accuracy is the key step of remote sensing satellite imagery analysis and the premise of analyzing 、 using remote sensing images. In the earlier days,

cloud detection methods can be divided into two categories: based on threshold and statistics. The method based on threshold is to recognize cloud pixels by using the high reflectance of clouds in three visible and near-infrared bands and the low temperature in thermal infrared bands. Ackerman et al. [1] used the Cloud Mask Algorithm in the product of generating cloud ,which comes from the data of medium resolution imaging spectrometer 1b, and Zhu et al. [2] used the classical f-mask (function of mask) algorithm for cloud detection of Landsat series satellite images. Because of its simplicity and fast computing speed, the method based on threshold method has been widely used in cloud detection for remote sensing imagery. Band selection and threshold selection depend on the spectral differences between cloud clusters and typical scenes. However, due to the different scenes and the complexity of cloud pose, it is usually difficult to identify accurately. Therefore, the method based on threshold usually has low accuracy and very unstable. In some areas with high reflectivity, it will be covered by snow surface, lakes and other areas. Moreover, it will fail to detect in some areas covered by thin clouds and broken clouds.

Another kind of cloud detection method is based on statistics, which is to calculate spectral and texture information from the original sample data, and then use pattern recognition methods, such as cluster analysis [3], support vector machines (SVM)
[4] and artificial neural network [5] to implement cloud detection. This method can automatically learn the spectral characteristics of clouds and typical scene types from the original sample data, and use a large number of samples to train the classifier. Richard [6] et al cut the remote sensing imagery into 16 × 16 pixel matrix as the input

sample, and the calculated spectrum, texture and other physical properties are input into the probability model. Liou et al. [7] extracted the texture features of cloud based on singular value decomposition (SVD) method, and classified the features using their own design drawing network. Compared with the method based on threshold, the cloud detection method based on statistics has higher accuracy and stability, and there are still some problems with detection of thin cloud and broken cloud.

In recent years, with the fast development and excellent feature extraction ability, more and more researchers are paying attention to deep learning. In the task of cloud detection for remote sensing imagery, the method based on DNN can be divided into pixel block classification and pixel level classification. In the view of pixel blocks, Shendrik [8] et al. proposed a multi network integration strategy for multi label classification of high-resolution satellite sub scenes with cloud, cloud shadow and land cover. Liu [9] et al. trained Deep Convolutional Neural Network (DCNN) to learn the characteristics of super pixels, and generated a cloud forming probability map by identifying the super pixels of the input image. In 2014, Long [10] and others firstly proposed Full Convolutional Networks (FCN) network for image pixel level segmentation. That was the first time that DCNNs model has been successfully applied to the task of image segmentation and has become the cornerstone of semantic segmentation. In 2015, Ronneberger [11] and others proposed a U-shaped network structure. U-net can obtain spatial location information and context information at the same time. It has the characteristics of simple network and excellent effect in semantic segmentation. Subsequently, many works [12,13,14] has been done for the improvement of u-net, which was applied to the field of remote sensing. Yin et al. [15] proposed a multi-level feature connected convolutional neural network to realize cloud detection in Landsat 7 and Landsat 8 images on gee. Zhan [16] et al. only used RGB images with three visible spectral bands and used the idea of FCN to segment clouds and snow. In order to make full use of the physical characteristics of clouds in different spectral bands, Dreonner [13] and others used u-net architecture to detect clouds in multispectral remote sensing images, which effectively improved the accuracy of cloud detection, but also brought additional computing difficulties for multispectral imagey model. Chai [17] et al. proposed an improved version of adaptive SegNet model. The convolution layer is used to extract different levels of spatial and spectral features, and the deconvolution layer produces detailed segmentation results. In terms of using spatial detail features, a pool index method is proposed to improve the accuracy of cloud and cloud shadow detection for remote sensing images. However, SegNet's ability to obtain global semantic information is insufficient, resulting in false detection. Chen [18] et al. proposed hole convolution in DeepLab network to increase the receptive field of the model and improve the accuracy of the segmentation model. Yan [19] et al. proposed a Pyramid Pooling Module (PPM) to obtain global context information, which enhances the ability to distinguish segmented objects. Li et al.

[20] introduced the multi-scale feature fusion module to effectively master the multi-scale information and global information. However, there are inevitable problems of cloud classification errors and long model training time. The cloud mask is extracted directly from the thumbnail of the remote sensing imagery (the preview image). Due to the loss of resolution and spectrum information, the cloud detection of the preview image has more challenges. Yang et al. [21] used a CNN based on cloud detection network to process the preview image of remote sensing imagery. The network used Feature Pyramid Module (FPM) to extract multi-scale context information. Although the above methods have achieved good results in the field of semantic segmentation, there are still unclear segmentation visualization results, and there are a lot of noise around the boundary of the segmented object.

At present, more and more people use the network model , so the accuracy is higher than before. The performance of densely connected network is the most prominent, but while pursuing the acme utilization of features, there are a large number of problems that generate features and occupy the memory of the model. Although the densely connected network improves the performance of network in spatial dimension, the densely connected network directly combines all the generated features , resulting in the annihilation of some useful feature information and the redundancy of some useless feature information. And the Stochastic Gradient Descent (SGD) algorithm used in densely connected networks has the same scaling of each gradient when updating parameters, resulting in slow training speed and poor effect

when the training data is very uneven. In order to solve this problem, this paper introduces the attention mechanism to optimize the resource utilization between each feature channel. In view of

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 2 Feb 2022, pp: 1266-1278 www.ijaem.net ISSN: 2395-5252

the fact that the densely connected neural network will produces a large number of middle layer features and occupies a large amount of video memory in the training process, the method of shared storage is introduced to solve the problem that the model occupies a large amount of video memory, so that deeper models will be trained to have better performance, and the learning rate can be dynamically cut in combination with the latest AdaBound optimizer, The model not only has the fast convergence speed in the early stage of Adam training, but also has the excellent performance in the later stage of SGD training. Experimental results show that the proposed method can realize fast and accurate segmentation of sexual cloud images.

II. METHODOLOGY

Convolutional neural network mainly relies on convolution operations, using the idea of local receptive fields to fuse spatial and channel information, so as to extracts effective features. However, it is difficult to learn a network with good performance. The attention module introduced in this paper does not introduce a new spatial dimension, but uses a new brand called "feature recalibration" method to integrate channels of features. "Feature recalibration" means that the network model determines the importance of each future's channel through learning, and enhances or suppresses the features of different importance. Essentially, it allows the feature channel to perform adaptive calibration.

Squeeze-and -Excitation module

The output of the convolutional layer does not consider the dependence of each channel. The goal of the Squeeze-and Excitation module (SE module) is to enable the model to selectively enhance features with a large amount of information, so as to make full use of useful features and suppress useless features.

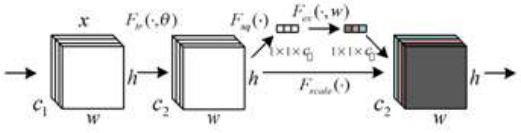


Figure 1 Squeeze-and- Excitation module

As shown in Figure 1, it is a schematic diagram of the Squeeze-and -Excitation module. The input is X , the number of channels is C_1 , and the number of channels becomes c_2 after a series of convolution operations. The difference from the traditional CNN operation is that the SE module recalibrates the characteristics of the front module can reconstruct any input information, and the input and output formulas are as follows:

$$F_{tr}: X \to U, X \in R^{W' \times H' \times C'}, U \in R^{W \times H \times C} \quad (1)$$

where F_{tr} represents the standardconvolution operator, and its formula is as follows (v_c represents the convolution kernel, represents the s input):

$$u_c = v_c * X = \sum_{s=1}^{C'} V_c^s * X^s \quad (2)$$

The result U is the second three-dimensional matrix on the left side of the module in Figure 1, which is a feature map of $H * W$ size, and U_c represents the first c matrix in U , and the subscript c represents the channel, followed by the Squeeze operation, the formula is as follows:

$$Z_c = F_{sq}(u_c) = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} u_c(i, j) \quad (3)$$

Therefore, the above formula converts the end through three operations [16]. The SE network input of $W \times H$ into the output of 1×1 , which corresponds to the operation of F_{sq} in Figure 1. After the operation of F_{sq} , the result is equivalent to indicating the value distribution of c feature maps, that is, the whole information. After the Squeeze operation (Squeeze), the next is the Excitation operation (Excitation), the formula is as follows:

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 2 Feb 2022, pp: 1266-1278 www.ijaem.net ISSN: 2395-5252

$$s = F_{ex}(z, W) = \sigma(W_2 \delta(W_1 z)),$$

(4)

where δ is the ReLU activation function, $W_1 \in R^{\frac{C}{r} \times C}$ and $W_2 \in R^{C \times \frac{C}{r}}$, In order to limit the complexity of the model and auxiliary generalization, two fully connected layers are introduced, that is, a dimensionality reduction layer with a parameter W_1, and then go through the ReLU activation function, followed by an ascending layer with a parameter W_2, and finally get s, s represents the weights of c feature maps, its dimension is $1 \times 1 \times c$, c represents the number of channels, and this weight is obtained from the learning of all the previous fully connected layers and nonlinear layers, so end-to-end training can be carried out. The main function of the two fully connected is to fuse the information of each channel. s is obtained by training, and then combined with U, the formula is as follows :

$$\tilde{X}_c = F_{scale}(u_c, s_c) = s_c \cdot u_c,$$

(5)

Model structure of SE-DenseNet

The SE-DenseNet in this article is a combination of SE module and DenseNet's dense connection module (Dense block) and transition module (Transition block). The input size is 28×28, then the size is reduced to 7×7 through the convolutional layer and the pooling layer of the initialization module. After that, the size remains the same, which is good for the continuous dense splicing of feature maps. And the SE module will re-calibrate the features in the densely connected module and the transition module, giving greater weight to the effective feature map and reducing the weight of the redundant feature map. The specific model diagram is shown in Figure 2.

where U_c is a two-dimensional matrix, and s_c is the weight. The above formula is to multiply each value in the matrix by s_c, corresponding to the operation of F_{scale} in Figure 1, which essentially improves feature discrimination by introducing dynamic characteristics conditioned on input.

For the SE module, firstly, the signal of each channel of the output feature is compressed to obtain the global spatial information as the channel description, and the statistical information of each channel is obtained by average pooling. Secondly, to examine the degree of dependence of each channel, there are two standards for implementing the function. One is functional flexibility (the core must be able to learn nonlinear interactions between channels). Second, non-exclusive relationships must be learned. In order to meet the above conditions, the SE module uses the Sigmoid activation function, where δ is the ReLU function.

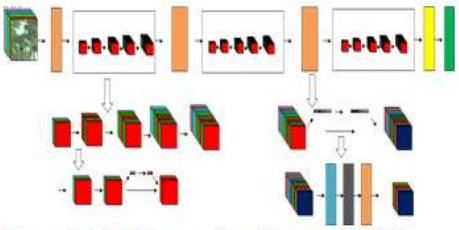


Figure 2 Cloud image detection model ofSE-DenseNet

Improved model parameter comparison

A. Comparison of different pooling methods and activation functions

In this section, we will mainly use experimental comparison to find the optimal improveed parameters. Table 1 shows that the 121-layer SE-DenseNet keeps other parameters unchanged and uses different pooling methods to compare the optimal verification accuracy of the maximum pooling and average pooling in the cloud map data set. Table 1 can be found in the model in the improvement, using average pooling as a pooling operation is better than maximum pooling.

Table 1 The effect of different pooling methods on the cloud map data set

Pooling method	Optimal verification accuracy
Max pooling	92.57%
Average pooling	93.29%

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 2 Feb 2022, pp: 1266-1278 www.ijaem.net ISSN: 2395-5252

Table 2 shows the effect of using different activation functions in the cloud image data set in SE-DenseNet121. Using the Sigmoid function as the activation function is better than the ReLU and Tanh activation functions.

Table 2 The effect of different activation functions on the cloud map data set

Activation function	Optimal verification accuracy
ReLU	92.07%
Tanh	92.27%
Sigmoid	93.29%

B. Experimental comparison of different decay rates The hyperparameter decay rate can be used to balance the accuracy and computational complexity. A smaller decay rate means greater computational consumption, but the accuracy rate will be higher. This time the improved algorithm adopts . It is worth noticng that the number of input channels must more than 16, otherwise the number of channels divided by 16 will become 0, and the model will not allow 0. Table 3 is the effect comparison of SE-DenseNet121 cloud image data set under different attenuation rates .

Table 3 The effect of different attenuation rates r on cloud data sets

Attenuation rate r	Optimal verification accuracy
4	93.41%
8	93.36%
16	93.29%

Considering the structure of the densely connected convolutional network, how to determine the parameters of the combination of the SE module and the densely connected network can be analyzed through the above experimental results. When SE module uses average pooling, Sigmoid function, and attenuation rate of 16, SE-densenet121 gets the best training results.

Video memory optimization

As a brand model, DenseNet make use of its own features so that can effectively improve the performance of the model, but it is inevitable that there will be a large number of intermediate features during the training process, which will cause the problem of large memory usage. In order to make deeper network training under the GPU environment, and to further improve the network performance, we can use shared storage to solve the above problem .

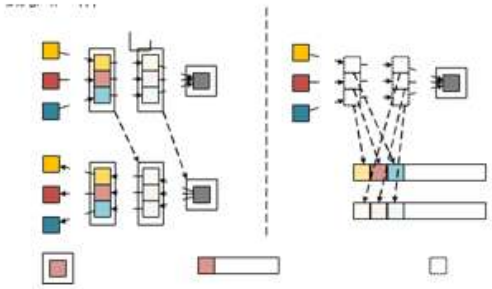


Figure 3 Comparison of training video memory storage methods

As shown in Figure 3, the left is the storage mechanism of DenseNet, the right is the improved DenseNet storage mechanism, the left is the input of a convolutional layer of a Dense block, including the output feature maps of all convolutional layers, and the new feature map is generated by concatenate operation, then as input features for convolution after normalization. However, the storage space will be reopened when concatenate and normalization operations are performed. Not only that, but in Pytorch, the features generated by backpropagation will also open up new storage space. The improved DenseNet stores the features of the middle layer in a temporary cache by pre-allocating shared storage space and pointers, further reducing the amount of storage.

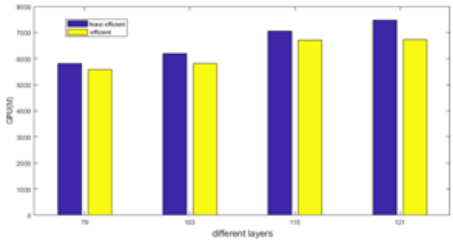


Figure 4 Comparison of video memory usage of different layers and different storage methods

In order to further verify the improvement of the shared memory space and pointers on the model's video memory occupancy, the model's video memory occupancy of the 79-layer, 103-layer, 115-layer, and 121-layer models are compared. From Figure 4, it can be found that the more layers it has, the better effect it will show. When the number of layers is 121, the optimization effect of the video memory is the most obvious, so training through shared storage space and pointers has very good practical value.

Model optimizer

With the continuous development of deep learning, many optimizers have been proposed.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 2 Feb 2022, pp: 1266-1278 www.ijaem.net ISSN: 2395-5252

This section mainly introduces the most representative model optimizer: SGD and Adam, points out their shortcomings, introduces the latest improved optimizer, and analyzes the adaptive algorithm combined with dynamic learning rate (Adaptive Gradient Methods with Dynamic Bound of Learning Rate, AdaBound), and applies it to the cloud detection model in this section to observe the performance on the cloud map data set.

SGD optimizer

The first-order optimization algorithm has made great progress in optimizing neural algorithms. As the most important algorithm among them, SGD has achieved good results in many applications. However, it is difficult to choose a suitable learning rate for SGD. When the data or features are sparse, the learning rate needs to be larger. When the data and features are more common, the learning rate needs to be smaller. The specific algorithm flow of SGD is shown in Table 4.

Table 4 SGD algorithm

Require : Learning rate ε_k
Require : Initial parameters θ
while do Stop criterion is not met do
Take a small batch of $\{x^1,\ldots,x^m\}$ samples from the t set m , Where the $x^{(i)}$ corresponding target is $y^{(i)}$
Compute gradient estimate: $\hat{g} \leftarrow +\frac{1}{m}\nabla_\theta \sum_i L\left(f(x^{(i)};\theta),\right.$
App update $\theta \leftarrow \theta - \varepsilon \hat{g}$:
end while

Adam optimizer

Adam is also a very popular algorithm in deep learning. It can replace the first-order optimization algorithm of the traditional stochastic gradient descent process. It is an algorithm that performs one-step optimization on the random objective function. It is relatively easy to implement and has high calculations. Efficiency and low memory requirements have a relatively good effect on solving large-scale data and parameter problems. And it can solve the hyperparameter problem intuitively, and the model can be optimized only by adjusting the parameters in a small amount. The specific algorithm flow is shown in Table 5.

Table 5 Adam algorithm

Require : Step size (recommended default: 0.001)
Require : The exponential decay rate of the moment estimation, and ρ_1 and ρ_2 are in the interval of $[0,1)$.
Require :Small constant for numerical stability δ
Require :Initial parameters θ
Initialize first and second moment variables s = 0 , r = 0
Initialization time step t = 0
while stopping criteria are not met *do*
Take a small batch of $\{x^1,\ldots,x^m\}$ samples from the training set m , , the $x^{(i)}$ corresponding target is $y^{(i)}$
Compute gradient : $g \leftarrow \frac{1}{m}\nabla_\theta \sum_i L(f(x^i;\theta),y^i)$; $t \leftarrow t+1$
Update the biased first moment estimate: $s \leftarrow \rho_1 s + (1-\rho_1) g$
Update the biased second moment estimate: $r \leftarrow \rho_2 r + (1-\rho_2) g \odot g$
Correct the deviation of the first and second moments: $\hat{s} \leftarrow \frac{s}{1-\rho_1^t}$; $\hat{r} \leftarrow \frac{r}{1-\rho_2^t}$
Calculate and apply updates: $\Delta\theta = -\varepsilon \frac{\hat{s}}{\sqrt{\hat{r}}+\delta}$; $\theta \leftarrow \theta + \Delta\theta$
end while

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 2 Feb 2022, pp: 1266-1278 www.ijaem.net ISSN: 2395-5252

AdaBound optimizer

SGD is currently the mainstream optimization algorithm for training models. However, However, since SGD updates the parameters, the size of the gradient scaling size is the same for each dimension ,which results in slow training speed .At the same time, the network model does not get good performance when the training data is extremely unevenly distributed. In order to solve the above problems, many adaptive methods such as Adam have emerged. Because Adam has a fast convergence rate, it has become a more commonly used optimization algorithm. However, optimizers like Adam have a faster performance in convergence speed than other optimizers in the early training stage, but later on the test set will be stagnant, and the final performance will be inferior to SGD.

In order to solve the above problems , the best result is the advantages of Adam's early rapid convergence can be combined with the advantages of good performance at the end of SGD. Based on the above ideas, through the dynamic adjustment of the learning rate, using AdaBound, this method is mainly inspired by the gradient clipping technology, the difference is that the learning rate is not gradient at this time, and the learning rate clipping operation is as follows:

$$Clip(\alpha / \sqrt{V_t}, \eta_l, \eta_u) \quad (6)$$

In the above formula, Clip limits the range of the learning rate between the lower bound η_l and upper η_u bound , SGD and Adam respectively correspond to the special cases : the learning rate a^* of SGD can be regarded as $\eta_l = \eta_u = a^*$; Adam can be regarded as $\eta_l = 0$, $\eta_u = \infty$, and the other values are in between .If two related functions t are used to replace the fixed value as the upper and lower bounds, which $\eta_l(t)$ converge from 0 to α^* and $\eta_u(t)$ converge from ∞ to α^* , the transition from Adam to SGD will be successfully realized. The specific process is shown in Table 6. Through the above settings, the influence of the upper and lower bounds on the learning rate can be reduced in the early training stage, making the algorithm closer to Adam. With the growth of time, the clipping interval becomes tighter and the learning rate of the model gradually becomes stable, and the performance that at the end of training also gets closer to SGD.

Table 6 Ada Bound algorithm

input: $x_1 \in F$, Initial step α , $\{\beta_{1t}\}_{t=1}^{T}$, β_2 , Lower bound function η_l , Upper bound function η_u

1: set up $m_0 = 0, v_0 = 0$
2: *for* $t = 1$ *to* T *do*
3: $g_t = \nabla f_t(x_t)$
4: $m_t = \beta_{1t} m_{t-1} + (1-\beta_{1t}) g_t$
5: $v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$ and $V_t = diag(v_t)$
6: $\hat{\eta}_t = Clip(\alpha / \sqrt{V_t}, \eta_l(t), \eta_u(t))$ and $\eta_t = \hat{\eta}_t / \sqrt{t}$
7: $x_{t+1} = \prod_{F, diag(\eta_t^{-1})} (x_t - \eta_t \odot m_t)$
8: *end for*

Comparison of related experiments

Experiment introduction

In order to verify the improved effect of the model in this article, this section adds an experimental comparison between the improved SE-DenseNet and M-DenseNet. The data comes from the China Resources Satellite Application Center, which is one of the three major satellite application centers in my country. The collection of cloud images mainly comes from HJ-1A/1B satellite images. Meteorological experts select 9600 thick cloud samples, thin cloud samples, and cloudless samples from the original cloud image as the data set, of which 8000 samples are taken as the training set. The remaining samples of the class sample are used as the test set, and the pixel size of each sample is 28×28 .

Comparison of simulation results

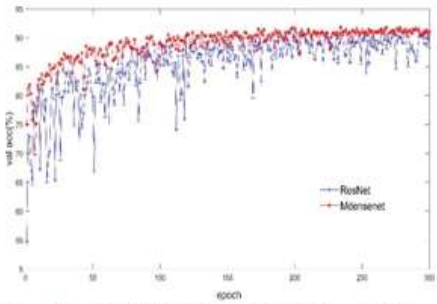


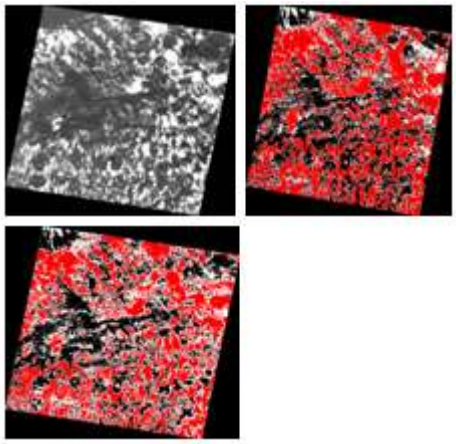
Figure 5 verification accuracy curve of M-DenseNet and RESNET

This simulation experiment uses a 121-layer multi-dimensional densely connected convolutional neural network (M-densenet) and a 152-layer deep residual convolutional neural network (ResNet) for comparison. Deep residual

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 2 Feb 2022, pp: 1266-1278 www.ijaem.net ISSN: 2395-5252

networks only rely on the most complex features of the network, while multi-dimensional densely connected convolutional neural networks utilize not only the most complex features, but also shallow features. Figure 5 shows the validation accuracy trends of deep residual convolutional neural networks and multi-dimensional densely connected convolutional neural networks on the Cloud dataset. In the first 50 iterations, the multidimensional densely connected network converges significantly faster than the convolutional neural network. And in the first 75 iterations, the validation accuracy of the multi-dimensional densely connected convolutional neural network is already close to 90%, while the validation accuracy of the deep residual convolutional neural network is close to 90% after
150 iterations. Densely connected convolutional neural networks significantly better than deep residual convolutional neural networks.

In order to enhance the contrast of the experiment, the cropped multispectral satellite cloud images are put into the trained model to be predicted, and then each satellite image is spliced in the form of pixels, so as to realize the overall prediction of satellite cloud images. The performance of the model can be evaluated intuitively. This simulation uses the point cloud experiment effect comparison chart. As shown in Fig. 6, no cloud is marked in black, thick cloud marked in red, and the thin cloud marked in white. From Figure 6, we can find that the deep residual network model has over-detection of thick cloud and false detection of thin cloud in the upper left corner, while the multi-dimensional densely connected convolutional neural network in this paper can bettercomplete different types of cloud image recognition, but mistakes are still inevitable. The false detection of thin cloud, shown in the upper left corner, illustrates that the multi-dimensional densely connected convolutional neural network still needs improvement.



(a)Original satallite cloud map (b) ResNet (c) M-densenet

Figure 6 Comparison of cloud detection effects of different methods

This simulation experiment mainly compares the improved 121-layer SE-DenseNet cloud detection model with the 121-layer M-DenseNet cloud detection model and the 152-layer ResNet cloud detection model. It can be seen from Figure 7 that the curve of SE-DenseNet is smoother than that of M-DenseNet, indicating that the features of the attention mechanism can be recalibrated by explicitly modeling the dependencies of the feature channels, and the importance of each feature channel can be learned by the SE module . The weights of features are recalibrated according to the importance of different feature channels to further optimize the performance of the model. From Figure 5.1, it can be found that compared with M-DenseNet, SE-DenseNet model has been well improved in terms of convergence speed and accuracy.

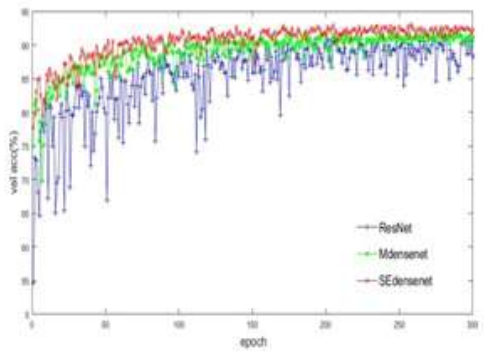
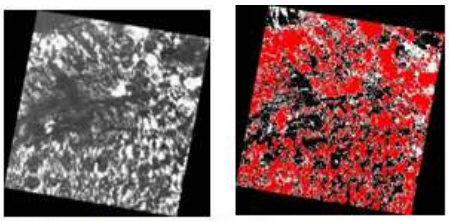


Figure 7 Verification accuracy curve of different algorithms

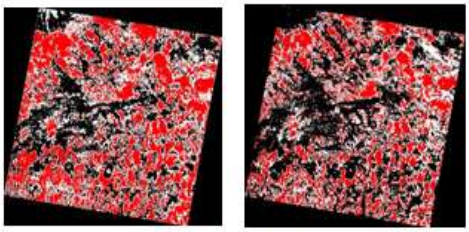
International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 2 Feb 2022, pp: 1266-1278 www.ijaem.net ISSN: 2395-5252

This simulation experiment mainly compares the 121-layer SE-DenseNet cloud detection model improved in this section, and compares the training results with the 121-layer M-DenseNet cloud detection model and the 152-layer ResNet cloud detection model. It can be seen from Figure 7 that the curve of SE- DenseNet is smoother than that of M-DenseNet, indicating that the attention mechanism can re-calibrate features by explicitly modeling the dependency of feature channels, and learn the importance of each feature channel through the SE module, and the weight of the feature can be re-calibrated according to the importance of different feature channels, and the performance of the model can be further optimized. From Figure 5.1, it can be found that SE-DenseNet is compared with M-DenseNet, whether it is the convergence speed of the model or the accuracy of the model relative to the original. The models have been well improved.

In order to further verify the improvement effect and compare the generalization effect of the model, this section also adds a comparison of cloud image detection results. As can be seen from Figure 8, compared to M-densenet, SE-densenet's over- and false-detection problems have been improved to a certain extent. Thus, it is proved that the SE-densenet model has better performance.



(a) Original satellite cloud image(b)ResNet



(c)M-densenet (d)SE-densenet

Figure 8 Comparison of cloud image detection effects of different algorithms

To further analyze the performance advantage of 121-layer SE-densenet over other networks, Figure 9 shows the accuracy curves of SE-densenet, LeNet, 19-layer VGG, and 152-layer ResNet. As can be seen from Figure 9, VGG and SE-densenet converge faster, but after 100 iterations, the accuracy of ResNet152 starts to catch up with VGG. The performance of the LeNet curve is poor and there is no faster convergence. Finally, the performance is also worse than other networks, while SE-densenet has the fastest convergence speed and better final performance, and the curve is smoother than other models, which further verifies that SE-densenet has better optimization performance and better performance. The convergence speed is fast, and the feature learning ability is stronger.

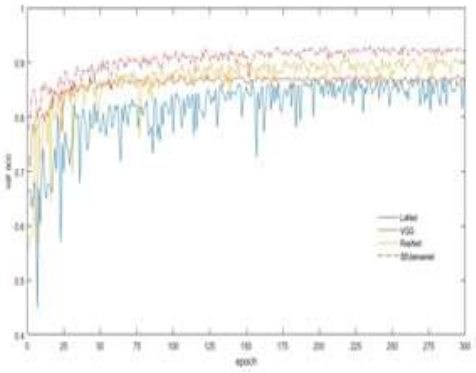


Figure 9 accuracy curve of each model verification

Precision and Recall are two metrics widely used in information retrieval and statistical classification. They are mainly used to evaluate the quality of results. Precision refers to how many samples the model judges to be positive. is the real positive sample; the recall rate refers to how many positive samples are judged as positive samples by the model. Let the set of positive samples output by the model is , and the set of real positive samples is .

Thick cloud			Thin cloud			No clou
recall	f1-score	precision	recall	f1-score	precision	recall
0.534	0.611	0.588	0.759	0.663	0.676	0.662
0.856	0.876	0.893	0.833	0.862	0.835	0.930
0.887	0.867	0.882	0.867	0.874	0.881	0.854
0.915	0.905	0.915	0.899	0.907	0.914	0.906
0.925	0.921	0.908	0.930	0.919	0.917	0.925

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 2 Feb 2022, pp: 1266-1278 www.ijaem.net ISSN: 2395-5252

$$Precision(A,B)=\frac{|A\cap B|}{|A|}, \quad (7)$$

$$Recall(A,B)=\frac{|A\cap B|}{|B|}. \quad (8)$$

When there is a conflict between the precision and recall indicators, it is necessary to consider these two indicators comprehensively. The most common method is F-Measure, F-Measure is the weighted harmonic mean of precision and recall, that is

$$F_\beta=(1+\beta^2)\frac{precision\times recall}{\beta^2\times(precision+recall)} \quad (9)$$

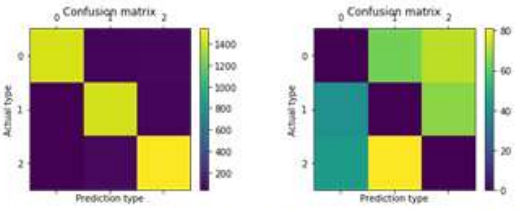
At the time when $\beta=1$, it was called the F_1 score, which was a common index for comprehensive evaluation. When the value of F_1 was large, it indicated the effectiveness of the experimental method.

As shown in Table 7 below, it is the index analysis results of each model on the test training set. From the table, it can be found that except for Thin Cloud, which differs from ResNet by 0.7% in precision, Thick Cloud differs from LeNet by 0.5% in recall rate. The f1-score as an indicator of precision and recall is better than other models, which further illustrates the excellent performance of SE-DenseNet in various sample classification.

Table 7 shows the index analysis results of each model on the test training set. From the table, it can be found that Thick cloud is 0.5% worse than LeNet in recall, Thin cloud is 0.7% worse than ResNet in precision. f1-score is superior to other models both in precision and recall, which further illustrates that SE-DenseNet has achieved good performance in various sample classifications.

Table 7 Index results of each model in the training data set

In the field of machine learning, the confusion matrix can be used as an error matrix to intuitively evaluate and supervise the performance of the algorithm. The size of the confusion matrix is a $n\times n$ square matrix representing the number of classes, where the size of the matrix is that each row represents an instance of the true class, and each column represents an instance of the predicted class. Confusion matrices can be used to visually analyze which classes are misclassified by the model.

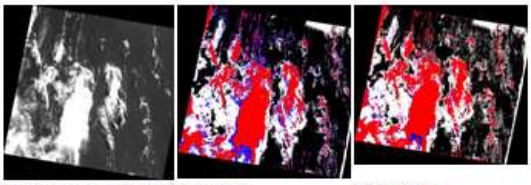


(a) Confusion Matrix I (b) Confusion Matrix II

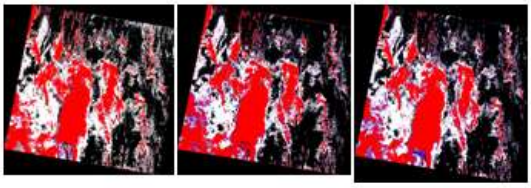
Figure 10 Confusion Matrix

As shown in Figure 10(a), the classification of each category by Se densenet is shown, where 0 is the label of cloud free, 1 is the label of thin cloud and 2 is the label of thick cloud. As shown in Figure 10(b), in order to visually observe the types misjudged by the model and set the number of correct classifications to zero, it can be found that the classification of thick cloud and thin cloud is the most prone to problems for the model, which is also due to the extremely complex and similar spectral characteristics of thick cloud and thin cloud.

Figure 10(a) is the classification of each category by SE-DenseNet, where 0 is the label of no cloud, 1 is the label of thin cloud, and 2 is the label of thick cloud. As shown in Fig. 10(b), in order to visually observe the types of model misclassifications and set the number of correct classifications to zero, it is found that the classification of thick and thin cloud is the most problematic model, which caused by extremely complexity and similarity in spectral characteristics .



(a)Original satellite image (b)SVM (c)LeNet



(d)VGG19 (e)ResNet152 (f)SE-densenet121

Figure 11 Comparison of cloud detection effects

To further compare the generalization performance of the models, a slider of 28×28 is used to cut the graph. The model predicts the probability of each slider and classifies each slider,

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 2 Feb 2022, pp: 1266-1278 www.ijaem.net ISSN: 2395-5252

thick cloud is marked in red, thin cloud is marked in white, but the overlapping part is difficult to judge, so we use the probability difference between thick and thin cloud to determine the overlap. The formula is as follows:

$$|S_h - S_b| \le 0.12 \quad (10)$$

Among them, S_h represents the probability of thick cloud, S_b represents the probability of thin cloud, and the overlapping part of thick cloud and thin cloud can be roughly calculated by the above formula, so as to complete the effect diagram of cloud detection.

Figure 11 shows the effect of cloud detection under different methods. Figure 11(a) is the satellite cloud image, Figure 11(b) is the effect of the traditional SVM, and Figure 11(c) is the effect of the LeNet neural network. Figure 11(d) is the rendering of the VGG19 neural network, Figure 11(e) is the rendering of the ResNet152 neural network, and Figure 11(f) is the rendering of the SE-densenet121 neural network. The thick cloud area is marked in red, the thin cloud part is marked in white, the mixed part of thick and thin cloud is marked in blue, and the cloudless part is marked in black. The characteristic of the traditional SVM method is that it can do classification quickly, but when the grayscale of the target range is large, it will inevitably cause the loss of some detected targets, resulting in false detection. It can be clearly seen from Figure 11(b) that the SVM method is not suitable for edge detection. Some cloud-free areas in the upper right corner will be falsely detected as thin clouds, and some cases with only thin clouds will be falsely detected as a mixture of thin cloud and thick cloud.

There are still some difficulties in using LeNet to detect clouds with complex spectral features, and limitations in feature representation also in being. From Figure 11(c), it can be found that there are still some false detections in the upper right edge region. From Figure 11(d), it can be found that although the error detection of thin clouds in the upper right corner is significantly improved, the detection effect of the boundary between thick cloud and thin cloud is very common. ResNet solves the gradient problem by increasing the residuals in the network. It can be seen from Figure 11(e) that the detection effect of ResNet is better than that of traditional SVM, LeNet, and VGG19, and is close to the detection effect of Figure 11(f), but there are still some weak areas that are wrongly detected as thick clouds. . The SE-DenseNet in Figure 11(f) has a better detection effect.

In order to verify the generality of the algorithm, in addition to the experimental comparison of the sheet cloud in Figure 12, the experimental results of the block cloud in Figure 12 are also analyzed. In the case of false detection of the boundary area, the detection effect of the mixed area of thick and thin clouds is also general, and a large number of thick and thin mixed clouds will be over-detected, and the SVM is more serious than the convolutional neural network. In Figure 12(d), there is a thick cloud over-checked in the upper left corner, and a thick cloud in the lower right corner will be missed to some extent. Figure 12(e) is slightly worse than Figure 12(f) for boundary detection between thick and thin clouds. Therefore, the method in this paper achieves a better cloud detection effect.

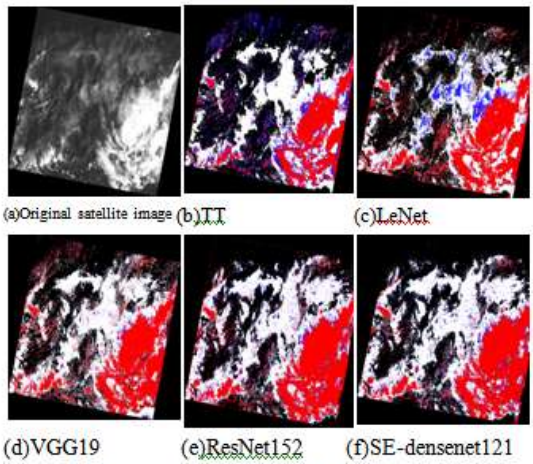
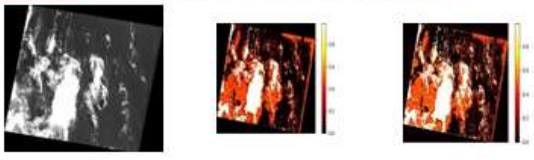


Figure 12 Comparison of cloud detection effects

In order to further analyze the cloud image detection effect, the prediction results are displayed in the form of a probability heat map, which is conducive to further analysis of the cloud image detection effect. In the cloud map heat map, the thick cloud is set to 1, no cloud is 0, and the color in the figure changes from black to white, and the chroma bar also changes from 0 to 1 accordingly.



International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 2 Feb 2022, pp: 1266-1278 www.ijaem.net ISSN: 2395-5252

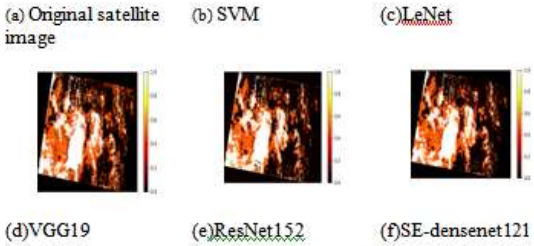


Figure 13 Comparison of cloud detection probability heat maps

From the above Figures 13(b), 13(c), 13(d), it can be seen that the detection effect of the right edge position is poor, and the Figures 13(f) is more delicate than that of Figures 13(e), thus further verifying the superiority of SE-DenseNet.

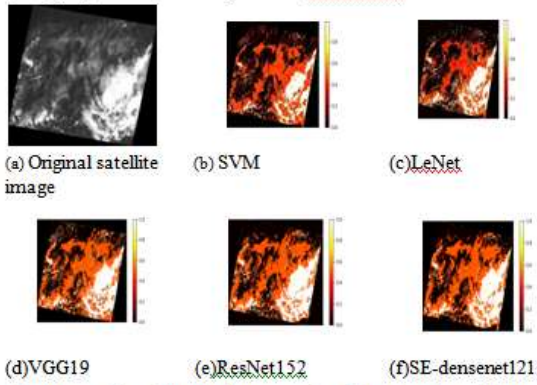


Figure 14 Comparison of cloud detection probability heat maps

From Figures 14(b), 14(c), 14(d) above, it can be found that the upper left corner will erroneously detect no clouds as thin clouds and thick clouds, while Figures 14(b) and 14(c) have thick cloud missed inspection. Figure 14 (e) is close to Figure 14 (f), but Figure 14 (f) is more suitable for edge detection. The heat map of SE-DenseNet121 in Figure 14 (f) is more detailed and plentiful, which proves the versatility of SE-DenseNet121. Compared with other deep learning methods, densenet121 algorithm has better performance.

III. CONCLUSION AND DISCUSSION

This paper mainly aims at Densenet's lack of feature channel dimension optimization, which greatly reduces the efficiency of feature extraction of the network model. Therefore, the SE module is combined with DenseNet to analyze the parameters of the SE module, and compare related experiments to determine the optimal algorithm parameters. In view of the situation that the video memory is heavily occupied during the training process, the shared memory is used, which effectively improves the utilization of the video memory. Finally, comparing the SGD to Adam optimization algorithms, pointing out shortcomings, AdaBound is applied to SE-DenseNet, and the improved SE-DenseNet is compared with the original model in the cloud image experiment to further verify the improvement of model performance.

REFERENCE

[1]. Xu B, Park T, Yan K. Analysis of global LAI/FPAR products from VIIRS and MODIS sensors for spatio-temporal consistency and uncertainty from 2012–2016[J]. Forests, 2018, 9(2): 73.

[2]. Zhu Z, Woodcock C E. Automated cloud, cloud shadow, and snow detection in multitemporal Landsat data: An algorithm designed specifically for monitoring land cover change[J]. Remote Sensing of Environment, 2014, 152: 217-234.

[3]. Lei Song, Min Xia, Junlan Jin, Ming Qian, Yonghong Zhang, SUACDNet: Attentional change detection network based on siamese U-shaped structure, International Journal of Applied Earth Observation and Geoinformation, Volume 105, 2021, 102597.

[4]. Min Xia, Yi Qu, Haifeng Lin, PADANet: parallel asymmetric double attention network for clouds and its shadow detection, Journal of Applied Remote Sensing. 15(4), 046512,2021.

[5]. Yi Qu, Min Xia, Yonghong Zhang, Strip pooling channel spatial attention network for the segmentation of cloud and cloud shadow. Computers and Geosciences, 157:104940, 2021.

[6]. Bankert, Richard L. Cloud Classification of AVHRR Imagery in Maritime Regions Using a Probabilistic Neural Network[J]. J.appl.meteorol, 1994, 33(8):909-918.

[7]. Liou R J , Azimi-Sadjadi M R , Reinke D L. Detection and classification of cloud data from geostationary satellite using artificial neural networks[C]// IEEE World Congress on IEEE International Conference on Neural Networks. IEEE, 1994.

[8]. Shendryk Y, Rist Y, Ticehurst C. Deep learning for multi-modal classification of cloud, shadow and land cover scenes in PlanetScope and Sentinel-2 imagery[J]. ISPRS Journal ofPhotogrammetry and Remote Sensing, 2019, 157: 124-136.

[9]. Liu H, Zeng D, Tian Q. Super-pixel cloud detection using hierarchical fusion CNN[C]//2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM). IEEE, 2018: 1-6.
[10]. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 3431-3440.
[11]. Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation[C]//International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015: 234-241.
[12]. Zhang J, Li X, Li L. Lightweight U-Net for cloud detection of visible and thermal infrared remote sensing images[J]. Optical and Quantum Electronics, 2020, 52(9): 1-14.
[13]. Drönner J, Korfhage N, Egli S. Fast cloud segmentation using convolutional neural networks[J]. Remote Sensing, 2018, 10(11): 1782.
[14]. Guo Y, Cao X, Liu B. Cloud Detection for Satellite Imagery Using Attention-Based U-Net Convolutional Neural Network[J]. Symmetry, 2020, 12(6): 1056. Maximum F1-Score Discriminative Training Criterion for Automatic Mispronunciation Detection[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2015, 23(4):787-797.
[15]. Yin Z, Ling F, Foody G M. Cloud detection in Landsat-8 imagery in Google Earth Engine based on a deep neural network[J]. arXiv preprint arXiv:2006.10358, 2020.
[16]. Zhan Y, Wang J, Shi J. Distinguishing cloud and snow in satellite images via deep convolutional network[J]. IEEE geoscience and remote sensing letters, 2017, 14(10): 1785-1789.
[17]. Chai D, Newsam S, Zhang H K. Cloud and cloud shadow detection in Landsat imagery based on deep convolutional neural networks[J]. Remote sensing of environment, 2019, 225: 307-316.
[18]. Chen L C, Papandreou G, Kokkinos I. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 40(4): 834-848.
[19]. Yan Z, Yan M, Sun H. Cloud and cloud shadow detection using multilevel feature fused segmentation network[J]. IEEE Geoscience and Remote Sensing Letters, 2018, 15(10): 1600-1604.
[20]. Li Z, Shen H, Cheng Q. Deep learning based cloud detection for medium and high resolution remote sensing images of different sensors[J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2019, 150: 197-212.
[21]. Yang J, Guo J, Yue H. CDnet: CNN-based cloud detection for remote sensing imagery[J]. IEEE Transactions on Geoscience and Remote Sensing, 2019, 57(8): 6195-6211.