CloudA: A Ground-Based Cloud Classification Method with a Convolutional Neural Network

MIN WANG
School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing, China

SHUDAO ZHOU
College of Meteorology and Oceanography, National University of Defense Technology, Nanjing, China

ZHONG YANG
College of Intelligent Science and Control Engineering, Jinling Institute of Technology, Nanjing, China

ZHANHUA LIU
College of Meteorology and Oceanography, National University of Defense Technology, Nanjing, China

(Manuscript received 7 November 2019, in final form 9 April 2020)

ABSTRACT

Conventional classification methods are based on artificial experience to extract features, and each link is independent, which is a kind of “shallow learning.” As a result, the scope of the cloud category applied by this method is limited. In this paper, we propose a new convolutional neural network (CNN) with deep learning ability, called CloudA, for the ground-based cloud image recognition method. We use the Singapore Whole-Sky Imaging Categories (SWIMCAT) sample library and total-sky sample library to train and test CloudA. In particular, we visualize the cloud features captured by CloudA using the TensorBoard visualization method, and these features can help us to understand the process of ground-based cloud classification. We compare this method with other commonly used methods to explore the feasibility of using CloudA to classify ground-based cloud images, and the evaluation of a large number of experiments show that the average accuracy of this method is nearly 98.63% for ground-based cloud classification.

1. Introduction

Cloud classification is very important for weather forecasts since it directly determines weather activities such as precipitation, snow, hail, and lightning. According to the shape, structure, characteristics, and height, clouds can be divided into 3 groups (high, middle, and low clouds), 10 general types, and 29 categories. They are characterized by their many kinds, rapid changes, similarity, and easy integration with the sky background. Therefore, it is difficult to classify so many kinds of ground-based clouds with high accuracy.

In recent years, many researchers focus on feature extraction techniques for different cloud attributes. Singh and Glennen (2005) used cooccurrence matrices and kernels to extract many features and distinguish five different sky conditions that no single feature extraction method was best suited for recognizing all classes. The different classification results are obtained using different feature methods, and the overall recognition rates can mask the ability of the classifiers to recognize each class individually. Calbó and Sabburg (2008) used texture attributes and the Fourier transform of the visible channels of a camera to classify up to eight types of sky conditions with an accuracy of approximately 62%. Sun et al. (2009) proposed a classification method of whole cloud image based on the local binary pattern (LBP) operator with an average accuracy of 87.2% for five classes: status, cumulus, undulatus, cirrus, and clear sky. Heinle et al. (2010) proposed an automatic cloud classification algorithm called Heinle feature based on a set of statistical features that described the color (mean,

Corresponding author: Min Wang, yu0801@163.com

DOI: 10.1175/JTECH-D-19-0189.1
© 2020 American Meteorological Society. For information regarding reuse of this content and general copyright information, consult the AMS Copyright Policy (www.ametsoc.org/PUBSReuseLicenses).
standard deviation, skewness, and difference) and texture (energy, entropy, contrast, uniformity, and amount of clouds) of whole sky images. The success rate of this method for classifying seven types of clouds was approximately 75%. Kazantzidis et al. (2012) proposed a multicolor standard for sky images that attained an average performance of approximately 87% for seven cloud categories. Kliangsuwan and Heednacram (2015) used a new method based on the fast Fourier transform to extract cloud features. The automatic classification accuracy of the seven cloud types reached 90%. Wacker et al. (2015) used the measured longwave radiation as the auxiliary information for cloud classification. Compared with the information using only sky cameras, the accuracy increased by nearly 10%, and the average accuracy was between 80% and 90%. Dev et al. (2015) proposed a modified texton-based classification approach that integrates both color and texture information for improved cloud classification results, the average accuracy is nearly 95% on Singapore Whole-Sky Imaging Categories (SWIMCAT) dataset. Li et al. (2016) adopted a new cloud type recognition method, which analyzed an image as a group of patches instead of a group of pixels. The accuracy of this method for five kinds of sky conditions was 90%. Zheng et al. (2018, 2019) developed a mature dynamic and adaptive wind power classification program that incorporates a comprehensive consideration of energy, environmental risk and cost factors.

It can be seen that the traditional automatic cloud recognition method relies on manual design, strongly depends on the characteristics of professional knowledge, and data, and it is difficult to learn an effective classifier from a large number of data to fully mine the association between data with a limited scope of application and low recognition accuracy. Therefore, we need to find a method that can automatically learn different cloud features. The convolutional neural network (CNN) has achieved great success in large-scale image classification tasks. The CNN is a deep, feedforward artificial neural network that has the ability to perform in-depth learning. After in-depth learning, it is possible to express features that are difficult to express in general, fully mine the association between data, extract the global features and context information of images, and carry out statistical recognition of varied clouds to obtain more accurate recognition results. AlexNet is a classical convolutional neural network, which entered a variant of this model in the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC-2012) and achieved a winning top-give test error rate of 15.3% (Krizhevsky et al. 2012, 2017).

In this study, a classification method for ground-based clouds with convolutional neural networks is proposed. Two different sample datasets, the SWIMCAT sample library and the total-sky sample library, are trained and tested to obtain the classification accuracy of CNN for two datasets. At the same time, the process of image processing by convolutional neural networks is visualized and compared with other commonly used methods to explore the feasibility of using CNNs to classify ground-based visible cloud images.

2. Data

In this paper, the SWIMCAT sample library and the total-sky sample library are selected as experimental samples. The similarities of the two datasets are divided into five categories, each of which is typical sky features, and there are strong differences and differentiability between the categories. The difference is that the first dataset is cloud patches with relatively small pixels, which makes it easier to distinguish. The second sample library is actually observed cloud images, which is relatively more difficult to distinguish.

a. SWIMCAT sample library

The SWIMCAT sample library mainly uses images taken by a wide-angle high-resolution sky imaging system. The system is a ground-based all-sky imager. It mainly consists of a Canon EOS 600D camera body and a SIGMA circular fisheye lens. The field of view is 180°, and the image can be captured automatically on a regular basis. The SWIMCAT sample library contains 784 image patches of five cloud categories acquired by the ground-based all-sky imager. The patches are cut from the whole image. These five types of clouds include sky, pattern, thick-dark, thick-white, and veil clouds. All image patches are 125 × 125 pixels in PNG format, as shown in Fig. 1.
b. Total-sky sample library

The total-sky sample library contains 5000 all-sky images, which are obtained from total-sky imager (TSI), a sky camera located over 4000 m above sea level in Tibet. TSI mainly consists of a digital camera and a curved surface mirror, which are taken every 10 s. All images are stored in JPG format with a resolution of 800 × 800 pixels. The image is divided into five sky conditions: cirrus, cumulus, stratus, clear sky, and hybrid, as shown in Fig. 2.

3. Methodology

The main idea of this method is shown in Fig. 3. First, the CNN model is trained using cloud images, and then the output feature map of the convolution layer is de-sampled by the maximum pooling operation interpolated in the model, and the visual features are obtained. Finally, the support vector machine (SVM) classifier is used to classify the features and obtain the classification results.

a. Model architecture

Because there are many kinds of ground-based clouds, which can be similar and fast changing, we propose a network structure, named CloudA. As shown in Fig. 4, it contains four convolution layers, four pooling layers, and three full connection layers as the standard configuration of the convolution neural network. The design considerations are as follows. 1) Because the cloud image input in the SWIMCAT sample library is patchy and has no obvious large scale or has difficulty in identifying features, feature extraction is not very difficult, and the requirement for the convolution layer is not very high. 2) Compared with other CNN network structures, this structure is relatively simple and has less computational complexity. 3) A simple network structure can be used. It greatly reduces the number of parameters to be learned and avoids overfitting.

The detailed configuration of CloudA is shown in Table 1. After the ground-based visible cloud image is entered into the network as input, the first layer is a convolution layer with a kernel of 5 × 5 pixels and an output channel of 32. The convolution kernel of the second layer is 5 × 5 pixels, the output channel is 64, the convolution kernel of the third and fourth convolution layers is 3 × 3 pixels, and the output channel is 128. All convolution layers are equipped with rectified linear unit (ReLU) (2009). Among them, the first and second edges are filled with 2, the third and the fourth edges are filled with 1, and there is a maximum pooling layer behind each convolution layer. The next three layers are completely connected neural layers. The number of neurons in each layer is 1024, 512, and 5 in turn. According to experience, for small-scale datasets, a better SVM classifier is selected to train the model.

The setting of parameters of each layer takes into account the actual situation of experience and input data. The size of the receptive field is usually cardinal, such as 7 × 7, 5 × 5, and 3 × 3 pixels, and decreases in turn with the deepening of layers. The recommended step size of the 5 × 5 and 3 × 3 pixel convolution nuclei is 1. Filling is mainly used to extract edge information effectively, and the larger the receptive field, the more the filling value will usually increase. The number of layers of convolution nuclei is usually a power of 2. Considering the size and complexity of the input image, the more layers that exist, the more information will be extracted; the last layer of

![Fig. 3. The main idea of this method.](image-url)
the full connection layer is the number of categories that need to be classified.

b. Training method

The training process of the network is shown in Fig. 5. After the sample library is input into the deep network, it is randomly assigned to the training set and the verification set. After the pretreatment and the initialization of the network structure and parameters, the network will carry out forward calculation and backward propagation to update the weights. Finally, the generalization ability of the model is tested by the verification set. If the model parameters reach the standard, the training can be completed by saving the model parameters. Those who fail to meet the standards need further training.

In this process, the input data are computed layer by layer to generate the final output and compared with the correct answer. After calculating the error of the output, the error flows backward through the backpropagation in the network. Each time the backpropagation occurs, the model parameters are adjusted in an attempt to reduce errors, thereby improving the model. Generally, training can be regarded as an iterative process, which involves transferring input data several times until the model converges.

4. Experimental details

The SWIMCAT sample library has 125 × 125 pixels in all image patches; the total-sky sample library has 800 × 800 pixels, and too many parameters will make the network training time too long, so for the total-sky sample library, the image is normalized to 125 × 125 pixels by bilinear interpolation. We use the bilinear interpolation method to zoom the image. The expression of the method is as follows:

\[
f(x, y) = \frac{1}{(x_2 - x_1)(y_2 - y_1)} \left[ (x_2 - x)(y_2 - y)f(x_1, y_1) + (x - x_1)(y_2 - y)f(x_2, y_1) + (x_2 - x)(y - y_1)f(x_1, y_2) \\
+ (x - x_1)(y - y_1)f(x_2, y_2) \right],
\]

where \(f(x, y)\) is the gray value of the target pixel \((x, y)\), so that \(f(x_1, y_1), f(x_2, y_1), f(x_1, y_2), \) and \(f(x_2, y_2)\) are the gray values of the four pixels \((x_1, y_1), (x_2, y_1), (x_1, y_2), \) and \((x_2, y_2)\), which are approximately pixel \((x, y)\).

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Convolution kernel</th>
<th>Step length</th>
<th>Filling</th>
<th>Convolutional kernel layers</th>
<th>Number of neurons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv1</td>
<td>5 × 5</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>32</td>
</tr>
<tr>
<td>Max pooling</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conv2</td>
<td>5 × 5</td>
<td>1</td>
<td>2</td>
<td>6</td>
<td>64</td>
</tr>
<tr>
<td>Max pooling</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conv3</td>
<td>3 × 3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>128</td>
</tr>
<tr>
<td>Max pooling</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conv4</td>
<td>3 × 3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>128</td>
</tr>
<tr>
<td>Max pooling</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fc5</td>
<td></td>
<td></td>
<td></td>
<td>128</td>
<td>1024</td>
</tr>
<tr>
<td>Fc6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>512</td>
</tr>
<tr>
<td>Fc7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
</tr>
</tbody>
</table>
The categories in each dataset are divided into a training set and test set according to the ratio of 9:1, and the samples are randomly selected as training samples. In deep learning, the common proportion of training set and test set is 8:2 or 9:1, and the proportion of the test set can be reduced appropriately when the sample library is large. For example, 500 total-sky sample libraries and 78 SWIMCAT sample libraries are taken out to verify the work and explain the problem. If the sample library is larger, the proportion of the test set can be smaller.

The batch size is set to 64, and the training stops at 100 epochs. The batch size is usually a multiple of 32, such as 32, 64, and 128, and the epoch number is based on several experiments. In view of the sample size of the total-sky sample library being 5000, batch processing is too small to create many calculations; the SWIMCAT sample library is 784, so batch processing is too large, and it is easy to train insufficiently, so the compromise value is 64. In the training process, the accuracy and loss curve reached its peak value at 80 epochs at the latest time, so it was slightly extended to 100. The activation function and loss function are selected as ReLU and L2-loss, respectively.

We select Adam (Kingma and Ba 2015) as the optimizer, which uses both order momentum and second-order momentum to maintain the exponential decay mean of the past gradient $m_t$.
where $m_t$ and $v_t$ are deviation estimates at the first and second times of the gradient, respectively, and $\beta_1$ and $\beta_2$ are attenuation rates, which are close to $1$. The Adam update rule is as follows:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\tilde{v}_t} + \epsilon} \tilde{m}_t,$$

where $\theta$ is the weight, $\eta$ is the learning rate, and $\tilde{m}_t$ and $\tilde{v}_t$ are the bias correction estimates at the first time and the second time of the gradient, respectively.

To visualize the cloud features captured by the convolution neural network, the image generated by the output of the features learned from the convolution layers of the CNN using the TensorBoard visualization method is shown in Fig. 6. The first is two different cloud images, followed by four different convolution kernels in different convolution layers. For example, conv1–9 can be interpreted as the features of the ninth convolution kernel in the first convolution layer. By comparing the images, the following characteristics can be obtained. 1) For the same image, the information obtained by deep and shallow convolution is different. For example, in appearance, the four feature maps produced in the first image are different. In shallow layers such as conv1–9, the texture and light and dark information in the cloud body are basically retained, but in deep layers such as conv4–3, the light and dark information in the cloud body has been omitted, but the boundary between the cloud body and the background is clearer (the contrast between green and blue is stronger), and the edge wheel is more prominent. 2) For the same convolution layer, the information obtained varies with the input image. For example, conv1–9 and conv1–27 produce corresponding feature maps because of different input images, and the same convolution layer shows different features for different types of cloud images.

5. Results and discussion

In this paper, we use the CloudA network to test SWIMCAT and total-sky sample libraries 10 times. All the algorithms are implemented in PyCharm with the help of the TensorFlow framework. The experimental results and the average recognition rate are shown in Table 2. Using the SWIMCAT datasets, the proposed convolutional neural network is compared with the following methods: 1) Heinle feature, which mainly captures the color, edge, and texture information of sky and cloud images (Heinle et al. 2010); 2) LBP is a very effective method for cloud image representation of texture description (Sun et al. 2009), and experiments are carried out using $(P, R) = (8, 1), (16, 2), (24, 3)$, where $P$ is the number of sampling points on a circle with radius $R$, and only the results in $(P, R) = (16, 2)$ with the best performance are listed; 3) the texton-based method recently proposed (Dev et al. 2015); and 4) AlexNet is a famous CNN with five convolution layers and three full connection layers (Krizhevsky et al. 2012, 2017). To adapt to this classification task, we modify the last full connection layer to five nodes, each node represents a category in the image sample library. The experiment results are shown in Table 3. Because of the difference of cloud image datasets, the classification rate of this paper is different from that of above references.

1) Through 10 experiments, it can be concluded that the proposed CloudA achieves good accuracy on both datasets. As shown in Table 2, the average accuracy of the SWIMCAT datasets is 98.47%, the total-sky sample dataset is 98.83%, the accuracy of individual recognition is 100%, and the worst accuracy is 97%, which is not different from the average value and has a certain stability.

2) As shown in Table 2, comparing two kinds of datasets, the total-sky sample library has higher image resolution and more noise in the image, but the

<table>
<thead>
<tr>
<th>Methods</th>
<th>Average accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBP</td>
<td>93.47</td>
</tr>
<tr>
<td>Heinle feature</td>
<td>93.09</td>
</tr>
<tr>
<td>Texton-based method</td>
<td>95</td>
</tr>
<tr>
<td>AlexNet</td>
<td>96.17</td>
</tr>
<tr>
<td>CloudA</td>
<td>98.63</td>
</tr>
</tbody>
</table>
accuracy obtained in the experiment is comparable to that of the SWIMCAT dataset. It can be concluded that CloudA can learn features and effectively accomplish classification tasks, regardless of image resolution, if the training data are sufficient when the classification categories are comparable.

3) As shown in Table 3, the experimental results show that the performance of the proposed method is better than the Heinle feature and LBP algorithm in the sample dataset. CloudA has a great advantage over both of them and even shows a nearly 4% advantage over the recently proposed texton-based method. This shows that whether CloudA or AlexNet, CNN has obvious advantages in cloud feature representation than the traditional manual method.

4) Comparing the computational speed of the two datasets with CloudA, Fig. 7 shows the typical accuracy curve of the two datasets (SWIMCAT sample library curve is on the left, total-sky sample library curve is on the right). When the SWIMAT sample library is trained 200 times, the accuracy is relatively stable, and there is no large fluctuation, while the total-sky sample library reaches its peak value only after 400 training times, and then it is accompanied by small oscillation. It can be verified that the SWIMCAT sample library is relatively simple and easy to classify. In fact, only 300 training times are needed to ensure the classification effect, while the total-sky sample library requires more computing resources.

5) Regarding model learning and overfitting, CloudA undoubtedly achieves satisfactory classification accuracy, but the overfitting situation of the model needs to be determined. The network status corresponding to several common training L2-loss and test L2-loss curves can be seen in Table 4. We compared the experimental training loss (left panel in Fig. 8) with test loss (right panel in Fig. 8). The loss of training and test sets generally shows a downward trend. Although there are fluctuations, it eventually tends to converge. According to Table 4, it can be judged that the network is in good condition and has not been fitted.

6. Conclusions

To improve the accuracy and stability of the traditional classification method for ground-based clouds, a convolutional neural network structure called CloudA, which is trained from scratch, is proposed to solve the relatively small-scale ground-based visible cloud classification problem. The model can preprocess the original image directly without extracting features manually, thus realizing the intelligent classification of cloud images. Moreover, it can visualize the cloud features captured of CloudA. On the same sample library, CloudA’s classification accuracy is higher than the current popular

<table>
<thead>
<tr>
<th>Situation</th>
<th>Train loss</th>
<th>Test loss</th>
<th>Network condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Decline</td>
<td>Decline</td>
<td>Learning</td>
</tr>
<tr>
<td>2</td>
<td>Decline</td>
<td>Unchanged</td>
<td>Overfitting</td>
</tr>
<tr>
<td>3</td>
<td>Unchanged</td>
<td>Decline</td>
<td>Sample inventory problem</td>
</tr>
<tr>
<td>4</td>
<td>Unchanged</td>
<td>Unchanged</td>
<td>Learning encounters bottlenecks, which can reduce the learning rate</td>
</tr>
<tr>
<td>5</td>
<td>Rise</td>
<td>Rise</td>
<td>Improper network structure design or parameter setting</td>
</tr>
</tbody>
</table>
methods of extracting features from the traditional
manual texture descriptors.

Acknowledgments. This work is supported by the
National Natural Science Foundation of China (41775165
and 41775039). We thank American Journal Experts
(www.aje.cn) for English language editing. The author
declares that there is no conflict of interest regarding the
publication of this paper.

REFERENCES
Calbó, J., and J. Sabburg, 2008: Feature extraction from whole-sky
ground-based images for cloud-type recognition. J. Atmos. Oceanic
Dev, S., Y. H. Lee, and S. Winkler, 2015: Categorization of cloud
image patches using an improved texton-based approach.
IEEE Int. Conf. on Image Processing, Quebec City, QC,
Canada, IEEE, https://doi.org/10.1109/ICIP.2015.7350833.
Heinle, A., A. Macke, and A. Srivastav, 2010: Automatic cloud
classification of whole sky images. Atmos. Meas. Tech., 3,
Kazantzidis, A., P. Tzoumanikas, A. F. Bais, S. Fotopoulos, and
G. Economou, 2012: Cloud detection and classification with
the use of whole-sky ground-based images. Atmos. Res., 113,
Kingma, D., and J. Ba, 2015: Adam: A method for stochastic op-
timization. Int. Conf. for Learning Representations, San Diego,
CA, Computational and Biological Learning Society.
Kliangsawan, T., and A. Heednacram, 2015: Feature extract-
tion techniques for ground-based cloud type classification.
classification with deep convolutional neural networks. 26th
Conf. on Neural Information Processing Systems, Lake Tahoe,
NV, Neural Information Processing Systems Foundation,
1097–1105.
——, ——, and ——, 2017: ImageNet classification with deep
convolutional neural networks. Commun. ACM, 60 (6), 84–90,
https://doi.org/10.1145/3065386.
Li, Q., Z. Zhang, W. Lu, J. Yang, Y. Ma, and W. Yao, 2016: From
pixels to patches: A cloud classification method based on bag
Singh, M., and M. Glennen, 2005: Automated ground-based cloud
Sun, X., L. Liu, T. Gao, and S. Zhao, 2009: Classification of whole
sky infrared cloud image based on the LBP operator. Daqi
Ke xue Xue bao, 32, 490–497.
Wacker, S., and Coauthors, 2015: Cloud observations in Switzerland
using hemispherical sky cameras. J. Geophys. Res. Atmos., 120,
global offshore wind energy resources. Renewable Energy,
——, Y. Chen, C. Zhan, and Q. Wang, 2019: Source tracing of the swell
energy: A case study of the Pacific Ocean. IEEE Access, 7,
139 264–139 275, https://doi.org/10.1109/ACCESS.2019.2943903.