Cloud Change Prediction System Based on Deep Learning

Dai Zheng(✉), Zhao Kanglian, and Li Wenfeng

NanJing University, 163 Xianlin Street, Qixia Distirct, Nanjing 210023, Jiangsu, China
1056318647@qq.com

Abstract. In satellite-to-ground laser communications, the laser beam is susceptible to the effects of atmospheric media when it passes through the atmosphere. The main reason is that the laser beam will be absorbed and scattered by the cloud when it passes through the cloud, causing the communication link to be blocked. In order to know the cloud cluster information around the laser beam in advance, this paper proposes a Cloud Prediction Network (CloudNet) model, which classifies first, then predict the cloud trajectory for the next 100 s by collecting clouds images over a ground station, so as to reasonably allocating the resources of the link and select the ground stations. The experimental results show that the prediction accuracy of the model is up to 81% under the condition of 5% error.

Keywords: Laser communication · Cloud Prediction Network · Cloud sequence prediction

1 Introduction

Laser communication is the main method of communication technology in future satellites-to-satellites and satellites-to-ground in space networks. Compared with traditional microwave communication technology, space laser communication takes the advantages of using laser as a carrier, which can realize large data transmission at high speed, and has high anti-interference and confidentiality [1]. Compared with inter-satellite laser links, satellite-to-ground laser links which need to pass through the atmosphere are subject to atmospheric turbulence, clouds, fog, haze and other atmospheric environments, which seriously attenuates the laser transmission signal. Therefore, it must be considered that atmospheric factors affect the transmission quality in the satellite-to-ground laser communication. It can provide important priori information on whether satellites and grounds are suitable for laser link construction and whether meeting expectations of link quality when learning about the status of cloud clusters over ground stations in advance so as to ensure uninterrupted satellite-to-ground laser communication. Therefore, in order to know the cloud coverage information around the laser communication link in advance, it is necessary to predict the change trend of the cloud over the ground station in advance.

Q. Wu et al. (Eds.): WiSATS 2020, LNICST 357, pp. 297–310, 2021.
https://doi.org/10.1007/978-3-030-69069-4_25

The extrapolation methods commonly used in radar image prediction mainly predict radar images on the premise that the system remains stable e.g. linear prediction methods, cross-correlation methods and single centroid methods which extrapolate to match the cloud image based on local feature matching and sequential cloud motion vector. However, linear prediction methods have been rarely used as most of the actual scenarios are non-linear problems. Reference [2] simply predict the cloud cluster position at the next moment based on the cross-correlation method by calculating the cross-correlation coefficient of two adjacent images. However, due to the inherent disadvantages of the cross-correlation method, the algorithm's forecasting ability rapidly decreases with time [3]. The monomer centroid method first needs to identify the monomers, and then scans the images at adjacent moments to perform the matching and tracking of the monomers. This type of algorithm is suitable for easily identified target monomers. For the application scenario of this paper, the cloud in the atmosphere will be deformed or disappear, and new clouds will be generated, so the cloud cannot be used as a single centroid model.

In the case that the traditional extrapolation methods have their own shortcomings, this paper proposes a solution, CloudNet, whose workflow is shown in Fig. 1, including a classification model and an unsupervised prediction model which both based on deep learning for predicting the trend of cloud clusters in the airspace through the laser link. First, a large number of continuous cloud images above the ground station were collected to make a cloud sequence dataset named CloudSequence. Then according to the characteristics of cloud cluster categories and total cloud cover in meteorology, the cloud sequence data set is divided into three categories: few-cloud, cloudy and stratiform clouds. During training, four neural network models need to be trained, namely a classification model and three prediction models corresponding to respective categories. During prediction, the cloud image sequence to be predicted is first classified by a classification model, and then the corresponding prediction model is used to predict the dynamic change trend of the cloud at the next moment according to the classification result.

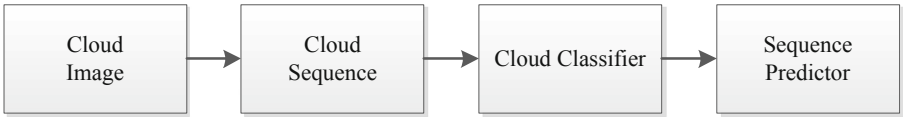


Fig. 1. CloudNet workflow

2 Classification of Cloud Sequences

2.1 Cloud Categories

Currently, clouds are generally divided into low, medium, high and direct expansion clouds as a international tradition. Based on this classification system in China, clouds are specifically divided into three families, ten genera, and 29 categories of clouds by combining observations and actual usage descriptions. Calbó et al. [4] defined 8 cloud

categories for different sky conditions while Heinle et al. [5] defined 7. These standards are formulated according to the needs of the scenario. For the use scenario in this article, the purpose of classification is to improve the period of time and accuracy of prediction so as to achieve uninterrupted transmission of the laser link. We attempt to divide the cloud sequences into two categories: few-cloud and cloudy according to the degree of cloud cover [6] by threshold of 1/3 due to the amount of cloud over the link will directly affect the prediction result. Furthermore, We treat stratiform clouds as a separate category and have the highest classification priority as the stratiform clouds are evenly curtain-like, with a large range and long duration among all categories of clouds, which are different from others. Based on the above characteristics, the cloud sequences are divided into three categories: few-cloud, cloudy and stratiform clouds.

2.2 Classification Model

Deep learning method is selected for its great performance in the field of computer vision while VGG16 model shows a graceful accuracy in the fields of image classification and object detection. Therefore, this paper extract features from cloud image sequences using VGG16 model whose structure is shown in Fig. 2.

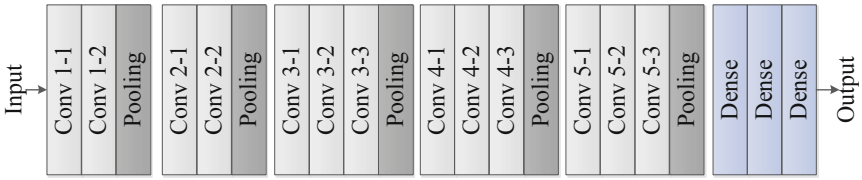


Fig. 2. VGG16 model structure

VGG16 has 13 shareable convolutional layers and three fully connected layers with very small (3×3) convolution filters, which shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16–19 weight layers [7]. This article uses pre-trained weight models on ImageNet for transfer learning, which means the trainable parameters of all layers except the last three fully connected layers setting to untrainable and only the parameters of the last three fully connected layers being adjusted.

3 Prediction Model of Cloud Sequences

It is necessary to build an internal model of the object and its motion model in Kinematics if you want to predict the trajectory of an object. But when it comes to the field of deep learning, all models are data-driven, that is, to predict the change trend of cloud clusters over ground stations of satellite-to-ground laser communication links, we do not need to model the changes and movements of cloud clusters, but trains model parameters through a large amount of continuous cloud image sequences.

Taking into account the temporal correlation of cloud sequences which actually are high-dimensional time series, this paper proposes a deeper PredNet model based on the Convolution Long-Short-Term Memory Network (ConvLSTM) as a prediction model. And we compared it with traditional stacked ConvLSTM model. We train these two network models with different types of cloud image sequences as input data to predict the change trend of cloud sequence at the next moment. And compare each other to determine which model better in actual scenario based on the actual testing accuracy.

3.1 Long Short-Term Memory Network

Long short-term memory (LSTM) is a special model of recurrent neural network (RNN) [8] that successfully solves the problem of gradient disappearance and explosion during the training process of back propagation through time (BPTT) of recurrent neural network [9] and can make full use of historical information, modeling the time dependence of signals. At present, LSTM networks show outstanding performance in many sequence prediction tasks, such as time series prediction, natural language processing [10] and so on.

The LSTM structure is shown in Fig. 3. It consists of an input layer, an output layer, and a hidden layer. Compared with the traditional RNN structure, the hidden layer of the LSTM is no longer an ordinary neural unit, but an LSTM unit with a special memory function [11]. Control gate and memory unit are defined in each LSTM unit, as shown in Fig. 4. The control gate includes input gate, forget gate and output gate, which are usually expressed by sigmoid or tanh functions, and the memory unit includes unit state (c_t) and output state (o_t).

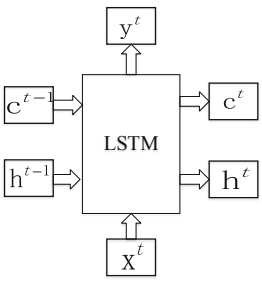


Fig. 3. LSTM structure

The specific working process of the LSTM unit is as follows: At time t, the LSTM input includes the input x_t at the current time, the last hidden state h_{t-1} and the last unit state c_{t-1} at the previous time, that is, retain the useful information in the cloud image sequence at the last moment. x_t and h_{t-1} get the current input unit state $\widetilde{c}_t$ through activation function *tanh*. The current unit state c_t is the sum of the last unit state c_{t-1} passing through the forget gate f_t and the current input unit state $\widetilde{c}_t$ passing through the input gate i_t. Finally, LSTM output h_t is calculated by the current unit state c_t through the activation function and output gate. The calculation formula [12] for each variable

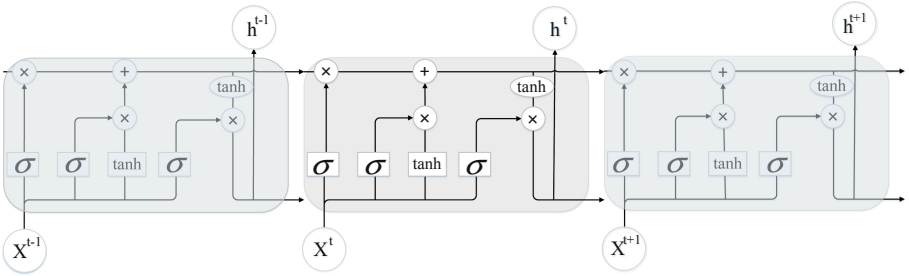


Fig. 4. Four interaction layers in LSTM

is as Eq. (1) to (5):

$$i_t = \sigma_g(W_{xi} \cdot x_t + W_{hi} \cdot h_{t-1} + W_{ci} \cdot c_{t-1} + b_i) \tag{1}$$

$$f_t = \sigma_g\left(W_{xf} \cdot x_t + W_{hf} \cdot h_{t-1} + W_{cf} \cdot c_{t-1} + b_f\right) \tag{2}$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \sigma_c(W_{xc} \cdot x_t + W_{hc} \cdot h_{t-1} + c) \tag{3}$$

$$o_t = \sigma_g(W_{xo} \cdot x_t + W_{ho} \cdot h_{t-1} + W_{co} \cdot c_t + b_o) \tag{4}$$

$$h_t = o_t \otimes \sigma_h(c_t) \tag{5}$$

Where: W_{xf}, W_{xi}, W_{xo}, W_{xc} are weight matrices connected to the input x_t while W_{hc}, W_{hi}, W_{ho}, W_{hf} are weight matrices connected to the last hidden state h_{t-1}. W_{ci}, W_{cf}, W_{co} are diagonal matrices connecting the current unit state c_t and the gate function. b_i, b_f, b_c, b_o are offset vectors. σ_g is the activation function, usually tanh or sigmoid function and '$\otimes$' denotes the Hadamard product.

One advantage of using the memory cell and gates to control information flow is that the gradient will be trapped in the cell (also known as constant error carousels [9]) and be prevented from vanishing too quickly, which is a critical problem for the vanilla RNN model [9, 13].

3.2 Convolution Long Short-Term Memory Network

Although the FC-LSTM layer has proven powerful for handling temporal correlation, it contains too much redundancy for spatial data. To address this problem, [14] propose an extension of LSTM called ConvLSTM which has convolutional structures in both the input-to-state and state-to-state transitions. The distinguishing feature of ConvLSTM is that all the inputs X_1, X_2..... X_t, cell outputs C_1, C_2,.....,C_t, hidden states H_1, H_2,.....,H_t, and gates i_t, f_t, o_t, of the ConvLSTM are 3D tensors whose last two dimensions are spatial dimensions (rows and columns). The ConvLSTM determines the future state of a certain cell in the grid by the inputs and past states of its local neighbors. This can easily be achieved by using a convolution operator in the state-to-state and

input-to-state transitions (see Fig. 5). The key equations of ConvLSTM are shown in Eq. (6) to (10) below, where '*' denotes the convolution operator and '⊗', as before, denotes the Hadamard product:

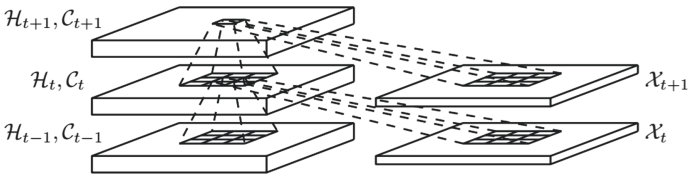


Fig. 5. Struture of ConvLSTM

$$i_t = \sigma_g(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} * C_{t-1} + b_i) \tag{6}$$

$$f_t = \sigma_g\left(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} * C_{t-1} + b_f\right) \tag{7}$$

$$c_t = f_t \otimes C_{t-1} + i_t \otimes \sigma_c(W_{xc} * X_t + W_{hc} * H_{t-1} + c) \tag{8}$$

$$o_t = \sigma_g(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} * C_t + b_o) \tag{9}$$

$$h_t = o_t \otimes \sigma_h(C_t) \tag{10}$$

This paper take advantage of ConvLSTM for multilayer stacking (see Fig. 6) which consists of two parts, an encoding network and a decoding network. The encoding ConvLSTM compresses the whole cloud image sequence into a hidden state tensor and the decoding ConvLSTM unfolds this hidden state to give the final prediction by upsampling.

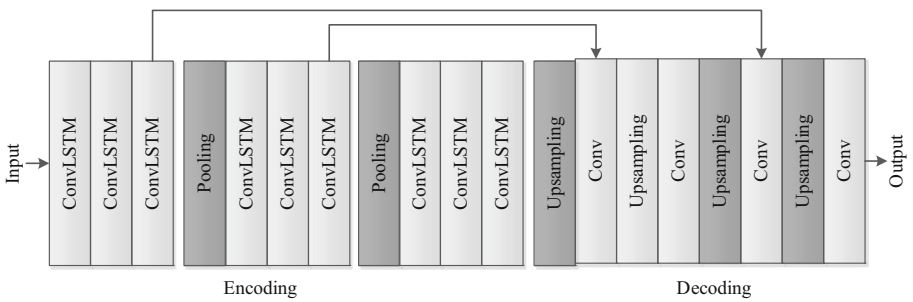


Fig. 6. Stacking ConvLSTM

3.3 Deeper PredNet

PredNet was originally used for short-term video prediction, and it is one of the state-of-art methods in the field of video prediction [15]. In this paper, we propose a deeper cloud sequences nowcasting model based on the PredNet architecture.

The Deeper PredNet model is improved based on the ConvLSTM model with increased stack layers from 4 to 6 and enlarged kernel size from 3 * 3 to 5 * 5 for larger state-to-state kernels are more suitable for capturing spatiotemporal correlations [14]. Each layer is composed of four basic modules: an input convolutional layer (A_l), a recurrent representation layer (R_l), a prediction layer ($\widehat{A_l}$), and an error representation (E_l). As Fig. 7 shows, these modules stacked in a certain relationship instead of simple linear stacking of ConvLSTM. The representation layer, R_l, is a recurrent convolutional network that can generate a prediction, $\widehat{A_l}$, of what the layer input, A_l, will be on the next frame. The network takes the difference between A_l and $\widehat{A_l}$ and outputs an error representation, E_l, which is split into separate rectified positive and negative error populations. The error, E_l, is then passed forward through a convolutional layer to become the input to the next layer (A_{l+1}). The recurrent prediction layer R_l receives a copy of the error signal E_l, along with top-down input from the representation layer of the next level of the network (R_{l+1}) [15].

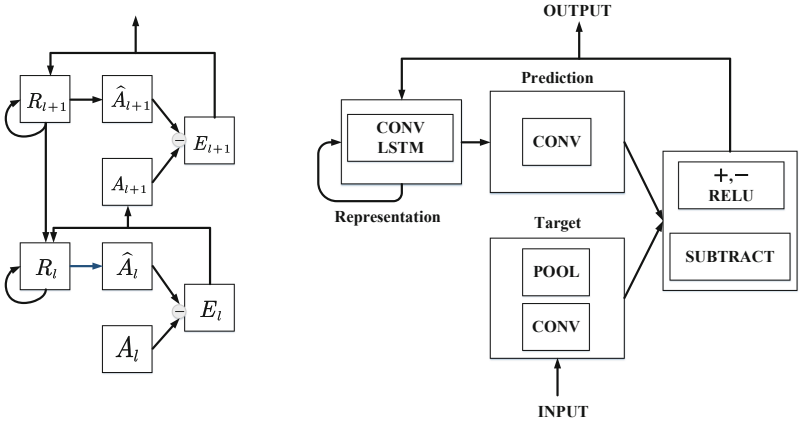


Fig. 7. Module structure of deeper PredNet

Although the model architecture is general with respect to the kinds of data with fixed-size, here we focus on cloud image sequences data x_t. The specific update rules are shown in Eq. (11) to (14). Different from the prednet, the error of each layer uses Huber loss function instead of L1 error [15]. The total loss function is defined as the weighted sum of the prediction errors at each layer and at each timestep, which is listed in Eq. (15), Where λ_t is the error weight at each time, while λ_l is the error weight at each layer, and n_l is the number of units in the lth layer.

The update sequence of each unit in the model is implemented according to the status update algorithm given in Fig. 8. The status is updated by two steps: first, the R_l^t, state is

updated by the top-down pass; then the A, A_l^t, and E_l^t states are updated by forward pass. R_l and E_l are initialized to 0 on the account of the nature of the convolutional network, indicating that the original predictions are spatially uniform.

$$A_l^t = \begin{cases} x_t, & if\ l = 0 \\ MAXPOOL\big(RELU\left(CONV\left(E_{l-1}^t\right)\right)\big), & l > 0 \end{cases} \tag{11}$$

$$\widehat{A}_l^t = RELU\left(CONV\left(R_l^t\right)\right) \tag{12}$$

$$E_l^t = \begin{cases} \frac{1}{2}(y - f(x))^2 & for\ |y - f(x)| \le \delta, \\ \delta|y - f(x)| - \frac{1}{2}\delta^2, & otherwise \end{cases} \tag{13}$$

$$R_l^t = CONVLSTM\left(E_l^{t-1}, R_l^{t-1}, UPSAMPLE\left(R_{l+1}^t\right)\right) \tag{14}$$

$$L_{train} = \sum\nolimits_t \lambda_t \sum\nolimits_l \frac{\lambda_l}{n_l} \sum\nolimits_{n_l} E_l^t \tag{15}$$

Algorithm 1 Calculation of PredNet state

Input: x_t

$A_0^t \leftarrow x_t$
$E_l^0, R_0 \leftarrow 0$
for $t = 1$ to T **do**
  for $l = L$ to 0 **do** ▶ Update R_l^t states
    if $l = L$ **then**
      $R_L^t = CONVLSTM(E_L^{t-1}, R_L^{t-1})$
    else
      $R_l^t = CONVLSTM(E_l^{t-1}, R_l^{t-1}, UPSAMPLE(R_{l+1}^t))$
    end if
  end for
  for $l = 0$ to L **do** ▶ Update $\hat{A}_l^t, A_l^t, E_l^t$ states
    if $l = 0$ **then**
      $\hat{A}_0^t = SATLU(RELU(CONV(R_0^t)))$
    else
      $\hat{A}_l^t = RELU(CONV(R_l^t))$
    end if
    $E_l^t = [RELU(A_l^t - \hat{A}_l^t); RELU(\hat{A}_l^t - A_l^t)]$
    if $l < L$ **then**
      $A_{l+1}^t = MAXPOOL(CONV(E_t^l))$
    end if
  end for
end for

Fig. 8. Algorithm of PredNet state calculation

4 Experiment

The overall model of the prediction system CloudNet structure is shown in Fig. 9. The cloud image sequence to be predicted is first classified by the classifier and then entered

into the predictor of the corresponding category. Our implementations of the models are in Python with the help of Keras2.2.4 using tensorflow backend. We run all the experiments on a computer with a single NVIDIA 2080Ti GPU.

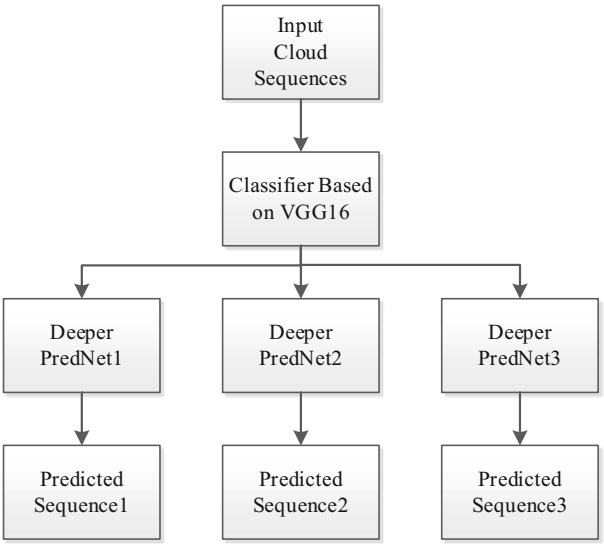


Fig. 9. CloudNet structure

4.1 DataSet

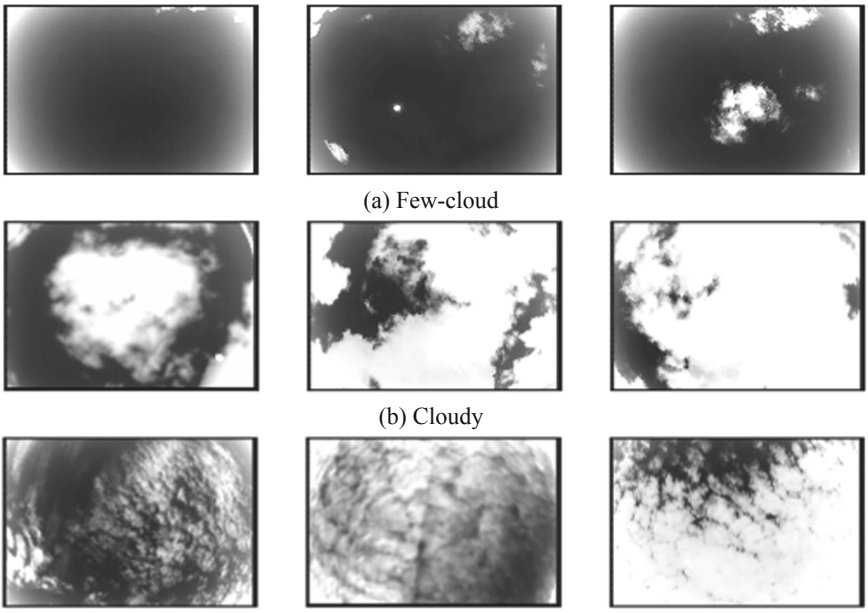
We collected the cloud image sequences over a specific area in Nanjing in March with the cloud infrared imager independently developed by our research laboratory. The infrared imager collected a total of 30291 cloud images as the basic dataset CloudImage by taking pictures every 5 s, all of which are grayscale images with a resolution of 720 × 480. 6,000 cloud image sequences with a time span of 100 s are sampled every 20 s on the CloudImage, and binary serialized after normalization and saved as cloud sequence dataset CloudSequence in hickle format.

4.2 Cloud Sequence Classification

We divided CloudSequence into three categories based on the principles of few-cloud, cloudy and stratiform clouds. As shown in Fig. 10, it is clear that the three categories of clouds have different characteristics. The classification accurancy with VGG16 model is up to 91% after the model being trained for several hours.

4.3 Cloud Sequence Prediction

It is shown in Fig. 11 and Fig. 12 respectively that the prediction results of using PredNet model and CloudNet model, where (a) to (c) are few-cloud, cloudy, and stratiform clouds



(a) Few-cloud

(b) Cloudy

(c) stratiform clouds

Fig. 10. Three cloud categories

image sequence respectively. In each image sequence, the first row is the ground truth of cloud image sequence, while the second row is predicted cloud image sequence based on the previous timestep. By comparing the predicted images, we can see that both two models can predict cloud image sequences. Under all three conditions, the prediction results of the PredNet model are significantly different from real images in the last several images. In comparison, the prediction result of CloudNet is more consistent with the ground truth.

Table 1, 2 and 3 respectively calculate the mean square error (MSE) and accuracy of the prediction results of PredNet and CloudNet on three conditions of few-cloud, cloudy and Stratiform clouds. In a few-cloud situation, the prediction accuracy will be relatively high for most of the picture is covered by the clear sky while a small part was occupied by cloud. In the case of cloudy, the accuracy of cloud motion prediction is more credible due to it has obvious movement visually and is growing and disappearing over time. Stratiform clouds are evenly distributed, with a large range and a long duration, making it easier for model to extract features, and as a result, the accuracy of prediction is quite acceptable. In actual use, all three cases will occur, so the data sets of the three cases will be used for model training, but the prediction accuracy of cloudy is relatively more credible. We train the model on three sets of data sets, but take cloudy prediction accuracy as the credible accuracy considering the three cases will occur in actual use.

According to the prediction results, CloudNet can predict the cloud status at the next moment more accurately than PredNet. The prediction average accuracy of the which is up to 81% with 100 s prediction interval and a tolerance of 5% error per pixel. Therefore,

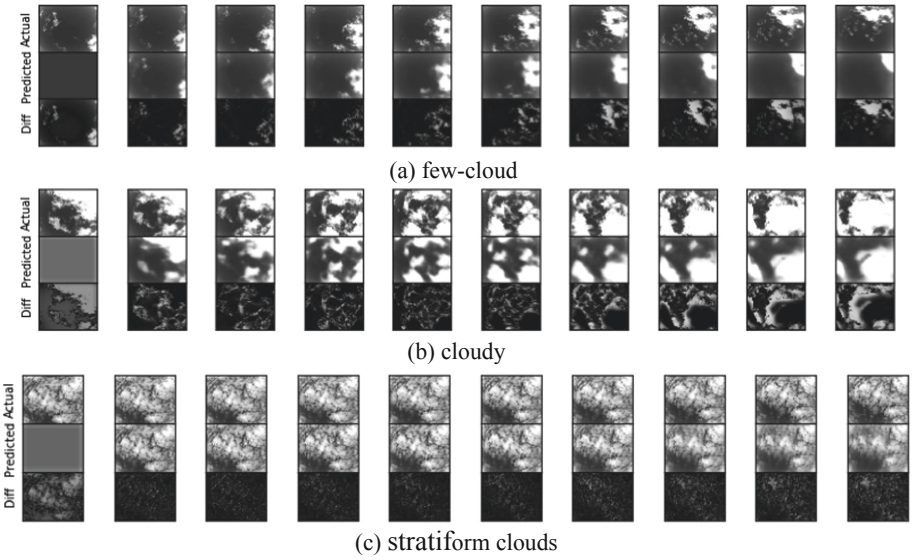


Fig. 11. Prediction results of PredNet

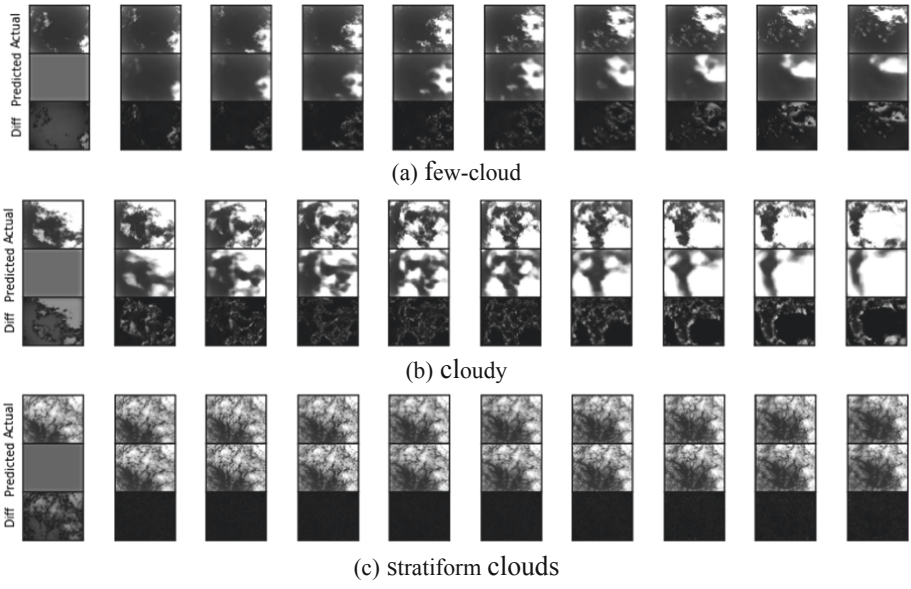


Fig. 12. Prediction results of CloudNet

we use the CloudNet model that is superior to the other model as the first choice in the scenario of cloud change trend prediction on the satellite-to-ground laser communication link.

Table 1. Few-cloud result

Model	PredNet	CloudNet
Average MSE	0.021499	0.014128
Last 5 frame MSE	0.035378	0.035097
Average accuracy	79.8133%	88.7363%
Last 5 frame accuracy	73.5179%	81.4399%
Last frame accuracy	71.2973%	75.8605%

Table 2. Cloudy result

Model	PredNet	CloudNet
Average MSE	0.022660	0.017616
Last 5 frame MSE	0.054533	0.028306
Average accuracy	76.7533%	81. 7590%
Last 5 frame accuracy	66.7648%	79. 8004%
Last frame accuracy	57.3893%	75. 7755%

Table 3. Cloudy result

Model	PredNet	CloudNet
Average MSE	0.003645	0.000412
Last 5 frame MSE	0.007160	0.001200
Average accuracy	90.3664%	96.4508%
Last 5 frame accuracy	82.7626%	88.6943%
Last frame accuracy	77.3012%	81.6838%

In order to compare with models in the field of video prediction, this paper uses the Hong Kong Observatory's public dataset HKO-7 to evaluate the performance of the model. The prediction results of the model on HKO-7 are shown in the Fig. 13, and the accuracy comparison results are shown in the Table 4. CloudNet has a good performance on satellite cloud images prediction although the it is specifically used to predict infrared cloud images collected from the ground.

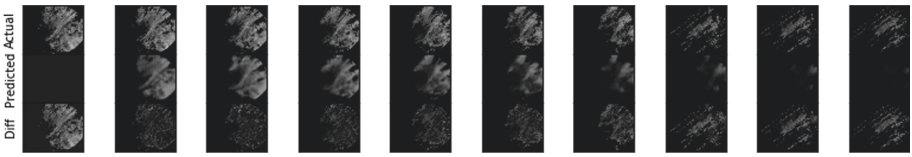


Fig. 13. Prediction result of CloudNet on HKO-7

Table 4. HKO-7 result

Model	CloudNet
Average MSE	0.005941
Last 5 frame MSE	0.009553
Average accuracy	86.9808%
Last 5 frame accuracy	86.0788%
Last frame accuracy	84.6634%

5 Conclusion and Future Work

This paper takes the influence of atmospheric cloud clusters on satellite-ground laser links as a starting point, and we have successfully applied the deep learning to the challenging cloud change trend nowcasting problem. We made two cloud datasets, CloudImage and CloudSequence, and proposed a cloud change trend prediction system CloudNet model based on the extension of PredNet to tackle the problem. By incorporating classification model and prediction model into the system, we build an end-to-end trainable model suitable for different weather conditions over the ground station. For future work, we will make a more detailed classification of cloud categories according to Meteorology. Besides, more research will be done on how to improve prediction time such as taking all-sky cloud image sequences as dataset for it has a longer time span.

References

1. Arnon, S., Kopeika, N.S.: Laser satellite communication network-vibration effect and possible solutions. Proc. IEEE **85**(10), 1646–1661 (1997). https://doi.org/10.1109/5.640772
2. Evans, K.F., Wiscombe, W.J.: An algorithm for generating stochastic cloud fields from radar profile statistics. Atmos. Res. **72**(1–4), 263–289 (2004). https://doi.org/10.1016/j.atmosres.2004.03.016
3. Hamill, T.M., Nehrkorn, T.: A short-term cloud forecast scheme using cross correlations. Weather Forecast. **8**(4), 401–411 (2002). https://doi.org/10.1175/1520-0434(1993)008%3c0401:astcfs%3e2.0.co;2
4. Calbó, J., Sabburg, J.: Feature extraction from whole-sky ground-based images for cloud-type recognition. J. Atmos. Ocean. Technol. **25**(1), 3–14 (2008). https://doi.org/10.1175/2007JTECHA959.1
5. Heinle, A., Macke, A., Srivastav, A.: Automatic cloud classification of whole sky images. Atmos. Meas. Tech. **3**(3), 557–567 (2010). https://doi.org/10.5194/amt-3-557-2010

6. Wood, R., Bretherton, C.S.: On the relationship between stratiform low cloud cover and lower-tropospheric stability. J. Clim. **19**(24), 6425–6432 (2006). https://doi.org/10.1175/JCLI3988.1
7. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, p. 14 (2015)
8. Karpathy, A., Johnson, J., Fei-fei, L.: V Isualizing and, p. 11 (2015)
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735
10. Narasimhan, K., Kulkarni, T.D., Barzilay, R.: Language understanding for text-based games using deep reinforcement learning. In: Conference Proceedings - EMNLP 2015 Conference on Empirical Methods In Natural Language Processing, p. 11 (2015). https://doi.org/10.18653/v1/d15-1001
11. Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to forget: continual prediction with LSTM. Neural Comput. **12**(10), 2451–2471 (2000). https://doi.org/10.1162/089976600300015015
12. Yao, K., Peng, B., Zhang, Y., Yu, D., Zweig, G., Shi, Y.: Spoken language understanding using long short-term memory neural networks. In: 2014 IEEE Workshop Spoken Language Technology, SLT 2014 - Proceedings, no. October, pp. 189–194 (2014). https://doi.org/10.1109/SLT.2014.7078572
13. Pascanu, R., Tour, D., Mikolov, T., Tour, D.: On the difficulty of training recurrent neural networks, no. 2
14. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.C.: Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: Advances in Neural Information Processing Systems, vol. 2015-January, pp. 802–810 (2015)
15. Lotter, W., Kreiman, G., Cox, D.: Deep predictive coding networks for video prediction and unsupervised learning. In: 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings, p. 18 (2019)