




Faster and Lighter Meteorological Satellite Image Classification by a Lightweight Channel-Dilation-Concatenation Net

Shuyao Shang , Jinglin Zhang , Xing Wang, Xinghua Wang, Yuanjun Li, and Yuanjiang Li 

Abstract—With the development of satellite photography, meteorologists are inclined to rely on methods for the automatic and efficient classification of weather images. However, many popular networks require numerous parameters and a lengthy inference time, making them unsuitable for real-time classification tasks. To solve these problems, a lightweight convolutional network termed the channel-dilation-concatenation network (CDC-net) is constructed for meteorological satellite image classification. When extracting features, CDC-net utilizes depth-wise convolution rather than standard convolution. Additionally, a FeatureCopy operation was employed instead of a half-convolution operation. CDC-net extracts high-dimensional features and contains a local importance-based pooling layer, reducing the network's depth, the number of network parameters and inference time. Based on these techniques, the CDC-net achieves an accuracy of 93.56% on the large-scale satellite cloud image database for meteorological research, with a graphics processing unit (GPU) inference time of 3.261 ms and 1.12 million parameters. Because many weather images reveal multiple weather patterns, multiple labels are necessary. Therefore, we propose a prediction method and conduct experiments on multilabel data. Experiments on single-label and multilabel meteorological satellite image datasets demonstrate the superiority of the CDC-net over other structures. Thus, the proposed CDC-net can provide a faster and lighter solution in meteorological satellite image classification.

Index Terms—Convolutional neural network (CNN), deep learning, lightweight, remote sensing, scene classification.

Manuscript received 21 November 2022; revised 10 January 2023 and 25 January 2023; accepted 28 January 2023. Date of publication 10 February 2023; date of current version 2 March 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB4500602, in part by the Key Research and Development Program of Jiangsu Province under Grant BE2021093, and in part by Taishan Scholars Program, Major Basic Research Projects of Shandong Province under Grant ZR2022ZD32. (Corresponding author: Xing Wang.)

Shuyao Shang is with the Department of Mechatronics and Information Engineering, Shandong University, Weihai 264209, China (e-mail: 202000800098@mail.sdu.edu.cn).

Jinglin Zhang is with the Department of Control Science and Engineering, Shandong University, Jinan 250061, China, also with the Department of Information Science and Engineering, Linyi University, Linyi 276000, China, and also with the Shandong Research Institute of Industrial Technology, Jinan 250100, China (e-mail: jinglin.zhang@sdu.edu.cn).

Xing Wang, Xinghua Wang, and Yuanjun Li are with the School of Automobile and Traffic Engineering, Jiangsu University of Technology, Changzhou 213001, China (e-mail: wangxing@lyu.edu.cn; 210854002052@lyu.edu.cn; liyuanjun@jsut.edu.cn).

Yuanjiang Li is with the Department of Marine, Jiangsu University of Science and Technology, Zhenjiang 212003, China (e-mail: liyuanjiang@just.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2023.3243915

I. INTRODUCTION

THE observation and identification of the weather on earth have been critical and challenging problems for a long time. Because of the independent nature of satellite remote-sensing imaging and its capacity for continuous observation, satellite cloud maps compensate for the lack of conventional detection data. However, even though considerable cloud data are acquired, its analysis and application have remained at the manual qualitative analysis stage, which is a resource-intensive and highly subjective approach. Therefore, how to automatically and effectively extract and identify patterns from hyperspectral image data obtained from satellites has become the hotspot of meteorological and remote-sensing image research. With the development of deep learning, convolutional neural networks (CNNs) have gradually become the standard practice for image classification in remote-sensing research. However, the current famous CNNs have deficiencies in classification speed for satellite weather images, which are rectified and optimized in this study.

First, we obtained satellite images as hyperspectral multi-channel images rather than RGB images, which are commonly used in deep-learning studies. In most current methods, features are extracted from multiple image channels, such as through the principal component analysis method [1] and the Fourier transform method [2]. However, these methods lead to a long inference time and too many parameters in the network, resulting in low classification speed and high demand for device storage space. In satellite images, many channels are highly similar or contain uninformative data. Therefore, we decided to combine the three channels most effectively representing various meteorological features to form an RGB image. It facilitated the visualization of the data and effectively reduced the network inference time and the number of parameters.

Space-grade computing systems that capture and process remote-sensing images must be real-time and lightweight. Because space-grade applications require high real-time processing power, the embedded network structures must exhibit rapid processing ability. In addition, because of the safety concerns in space propulsion systems and the adverse effects of single-particle inversions, the computer storage space and the number of embedded network parameters must not be manageable. For instance, Curiosity's computer system has only 256 MB of memory.

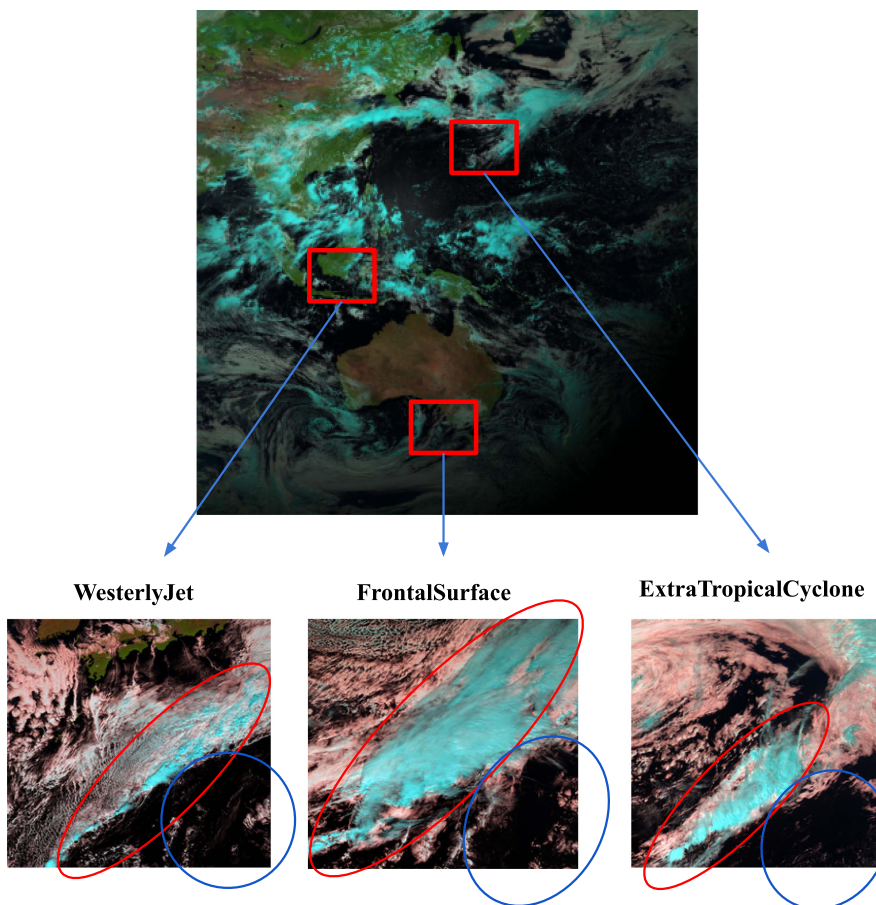


Fig. 1. Part of the different categories of meteorological images obtained from a satellite image. Patterns framed by the red and blue circles are similar, and these similar patterns are likely to pose classification difficulties.

Some of the common CNNs, such as ResNet [3] and DenseNet [4], use a deep network structure to achieve high accuracy, resulting in numerous network parameters and a slow inference speed. To develop CNNs that can be easily applied in spacecraft to process remote-sensing images, we designed the channel-dilation-concatenation network (CDC-net) in which depth-wise (DW) convolution is employed and proposed a FeatureCopy operation to reduce the number of convolutional operations required during feature extraction. CDC-net extracts high-dimensional features. The max-pooling layer is replaced by a local importance-based pooling (LIP) layer, which lowers the network's depth. These operations effectively reduce the number of network parameters and the inference time.

Fig. 1 demonstrates that there are complex and diverse patterns in meteorological satellite images, and different types of meteorological satellite images share similar patterns, revealing that CNNs with high classification accuracy is required. However, due to their lightweight nature, famous lightweight networks, such as MobileNet and ShuffleNet, are not sufficiently accurate. Therefore, to strike a balance between accuracy,

inference speed, and the number of parameters, we proposed the CDC-block extraction method with a channel-dilation, feature-extraction, and channel-squeeze structure, and introduced an LIP layer to ensure high classification accuracy.

Because of the complexity of meteorological images, images are often given multiple labels rather than being classified into one category. Only a few studies have investigated multilabel meteorological satellite image classification. Thus, we proposed a method for multilabel classification and conducted network performance experiments on a multilabel dataset. The results indicated that CDC-net could provide higher classification performance than other commonly used lightweight networks.

In summary, the contributions of this study are as follows.

- 1) To make the convolutional network more lightweight, we transformed the original hyperspectral images into RGB images via a direct channel selection operation.
- 2) To reduce the number of parameters and inference time of the network, we used DW convolution and designed a FeatureCopy operation. We also proposed a channel-dilation structure to guarantee high accuracy and adopted an attention-mechanism-based LIP layer.

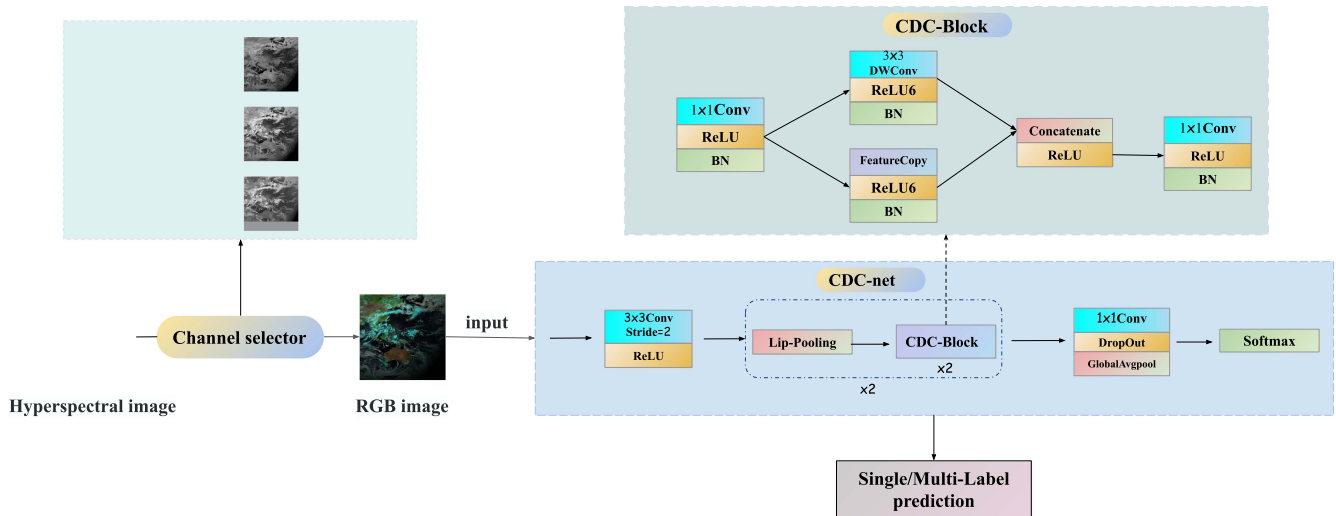


Fig. 2. Meteorological satellite images are first converted into RGB images, which are used as the input of CDC-net to obtain the category prediction results.

- 3) We created a classification method for multilabel images and conducted multilabel image classification experiments on CDC-net. The results demonstrated that CDC-net exhibited the highest multilabel classification performance.

The rest of this article is organized as follows. Section II discusses related studies. Section III focuses on the composition of the CDC-net and the CDC-block and introduces the DW convolution, FeatureCopy operation, LIP layer, and method for training multilabel data. Section IV describes the large-scale satellite cloud image database for meteorological research (LSCIDMR) in detail and provides multilabel model prediction results and details of the single-label and multilabel data experiments. It also presents a comparison between different network parameter numbers and inference times and highlights the visual analysis results. Section V gives more discussion about CDC-net. Finally, Section VI concludes this article.

II. RELATED WORK

Iandola et al. [5] presented SqueezeNet, a lightweight CNN in which the fire module was used for parameter compression, with the input layer first dimensionally compressed by a squeeze layer (1×1 convolution) and then dimensionally expanded by an expansion layer (a mixture of 1×1 and 3×3 convolution). Howard et al. [6] proposed MobileNetV1, in which deeply separable convolution was employed, which is a combination of DW convolution and 1×1 convolution.

Sandler et al. [7] proposed MobileNetV2, in which an inverted residual with linear bottleneck cells was used. The idea of channel expansion in CDC-net is also based on the inverted residual block. In MobileNetV3 introduced by Howard et al. [8], an additional squeeze-and-excitation (SE) layer was included and a combination of AutoML techniques with manual fine-tuning was used to obtain a lightweight network. Zhang et al. [9] designed ShuffleNetV1, in which a channel shuffle operation is employed to improve the performance of group convolution. Five guidelines for the design of lightweight networks

were brought forward by Ma et al. [10], and they put forward ShuffleNetV2 based on these guidelines, offering a fine balance between accuracy and speed.

To realize hyperspectral image classification, a global context spatial attention deep learning network with a global self-attentive mechanism module was designed by Chen et al. [11] for image classification. This global attention mechanism can significantly increase the classification accuracy of the network. Fan et al. [12] introduced a multiscale learning and attention enhancement network to range data fusion classification in an end-to-end manner, simplifying the network structure and making network training more efficient. Zhang et al. [13] proposed a multimodal attention-aware CNN which used an attention mechanism to enhance the classification performance of light detection and ranging data. Tu et al. [14] designed a global-local hierarchical weighted fusion architecture to do hyperspectral image classification, effectively integrating spectral and spatial features to improve classification accuracy.

Neural networks also participate in the analysis of weather remote-sensing data. For instance, Huang et al. [15] proposed multimodal spatiotemporal networks for processing hyperspectral weather images and forecasted the trajectory and intensity of tropical cyclones. Bai et al. [16] created a feature-extraction balanced network termed Rainformer to perform precipitation nowcasting. Hang et al. [17] invented an unsupervised feature learning model which utilized multimodal data to extract features without any label information. This strategy can explore semantic information and intrinsic structure information. Bai et al. [18] integrated images with meteorological elements and used such various modalities for clouds and weather systems to do satellite image classification tasks. Zhang et al. [19] built a ground-based cloud dataset and proposed a new CNN model called CloudNet for accurate ground-based meteorological cloud classification. Hang et al. [20] constructed a multiscale progressive segmentation network that cascaded three subnetworks for gradually segmenting objects into small-scale, large-scale, and other scales, which effectively alleviated the

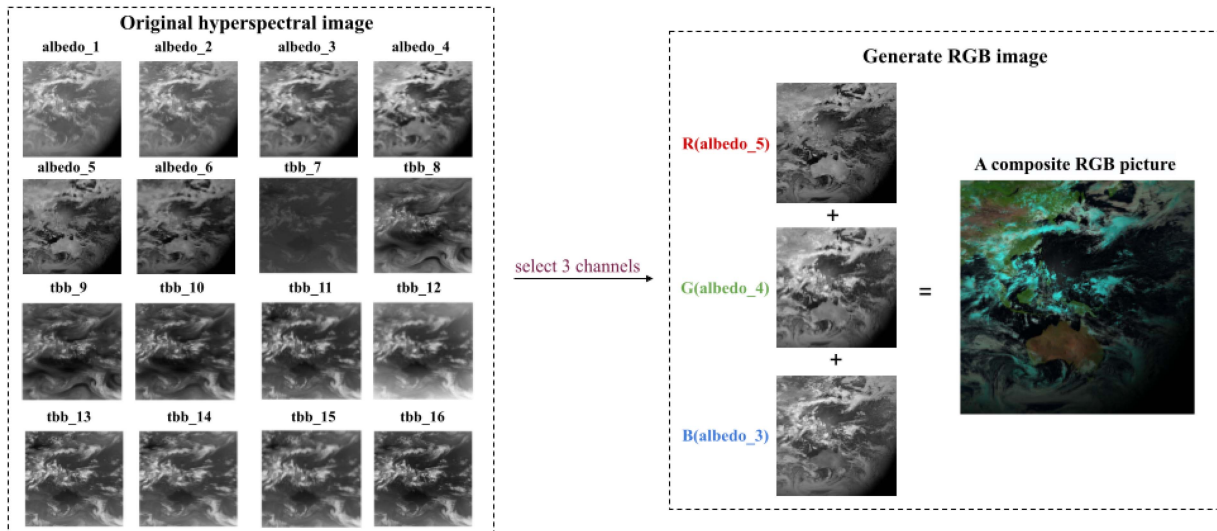


Fig. 3. Channel selection process. The raw data were obtained from a 16-channel hyperspectral image, which was subjected to a channel selection operation to obtain an RGB image.

limited learning capacity of each CNN. Hang et al. [21] designed a spectral super-resolution network by taking advantage of spectral correlation and projection property of hyperspectral imagery. This network contained a decomposition subnetwork and a self-supervised subnetwork to reconstruct hyperspectral imagery.

III. METHODS

Our proposed classification method is based on an end-to-end architecture. For a given meteorological satellite image, the three channels that best characterize the information in the image are selected and combined into an RGB image, which is then fed into a trained CNN to obtain classification results directly. The process is illustrated in Fig. 2.

A. Channel Selector

Generally, the primary classification method is based on multichannel hyperspectral images obtained from remote-sensing satellites. However, because the image information is similar among some channels or unhelpful for the classification, the network trains numerous redundant parameters. Therefore, in this study, we propose a channel selector operation in which only the three channels that optimally characterize the image information are selected and combined into RGB images.

In the LSCIDMR dataset, the raw satellite image data contains 16 channels. According to the official guidelines provided by the Meteorological Satellite Center of the Japan Meteorological Agency, we match the albedo_5, albedo_4, and albedo_3 channels to the R, G, and B channels, respectively, which helps to synthesize color images suitable for extracting satellite image meteorological features. This process is presented in Fig. 3.

For snow and ice-covered areas and clouds, the albedo_3 and albedo_4 channels have high reflectivity. The land appears dark, and the ocean appears the darkest. Cloud reflection depends on the optical thickness and density of cloud particles. Low clouds

and land and sea surfaces can be observed through thin, high clouds. Clouds can be distinguished by their texture, as stratus clouds have a smooth texture, and convective clouds have a rough texture. Because of the high reflectivity of chlorophyll in plants, the distribution of vegetation can be determined. In addition, high clouds composed of ice particles, snow, or ice can be observed in the albedo_4 and albedo_3 channels, despite their low reflectance in the albedo_5 channel. This information is reflected in RGB images, in which low clouds with high reflectivity (water clouds) appear white-gray, and vegetation appears green.

The reflectance properties of the albedo_5 channel are related to the phase and size of the cloud particles. Large clouds and ice particles have low reflectivity, whereas high clouds composed of ice particles, snow or ice, and sea ice have darker colors. In summary, the three channels, albedo_5, albedo_4, and albedo_3, can be used to characterize image information. The channel selection operation filters out the rest of the channel information, and these three channels are combined into an RGB image.

B. Channel-Dilation-Concatenation Network

The main structure of the CDC-net is elaborated in Fig. 4. The main body of the network is divided into two parts: 1) an extracted feature network; 2) a classification network. The extracted feature network firstly uses a 3×3 convolutional layer with a step size of 2 for downsampling, followed by a pooling layer for further downsampling, two CDC-blocks (see Section III-E) for feature extraction, a pooling layer for further downsampling, and finally two additional CDC-blocks for feature extraction. Subsequently, the feature map is gradually up-dimensioned by two CDC-blocks for feature extraction. In the classification network, the bulky fully connected (FC) layer is dropped, and a 1×1 convolutional layer is used to reduce the number of feature map dimensions such that it matches the number of categories (10 in this case) and to obtain the feature

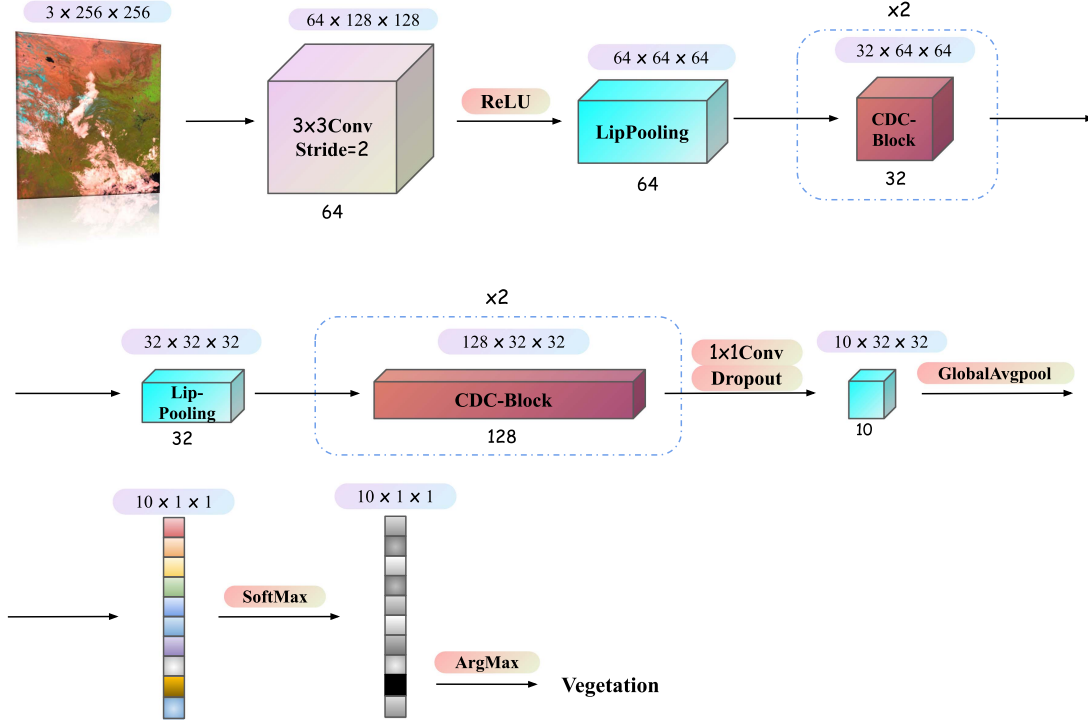


Fig. 4. General structure of the CDC-net. Size of the feature map is expressed as $channel \times W \times H$. The number below the feature map indicates its number of channels. For the input RGB images, a convolutional layer with a step size of 2 is used for feature extraction and downsampling, and then CDC-blocks and LIP-pooling layers are used to extract features. Finally, a 1x1 convolution layer and a global average pooling layer are used to obtain a 10-dimensional vector, and the prediction results are obtained by SoftMax operation.

map $T \in R^{10 \times W \times H}$. A global average pooling layer with no parameters is then used to obtain the output vector $v \in R^{10}$, which can be expressed as follows:

$$v_m = \frac{\sum_{i=0}^{W-1} \sum_{j=0}^{H-1} T_{m,i,j}}{W \times H} \quad (0 \leq m < 10). \quad (1)$$

If the FC layer is used, the network parameter ratios become

$$\frac{1 \times 1 \times 512 \times 10}{32 \times 32 \times 512 \times 10} = 0.09\% \quad (2)$$

and the calculated volume ratios are converted to

$$\frac{32 \times 32 \times 512 \times 10 + 32 \times 32 \times 10}{32 \times 32 \times 512 \times 10} \approx 1. \quad (3)$$

Therefore, the global average pooling layer used for classification has almost zero parameters compared with the commonly used FC layer. Thus, it does not affect the computational requirements. Lin et al. [22] suggested that global pooling layers have a regularization-like effect. According to the ablation experiments reported in Section IV-E, the global pooling layer classification accuracy is higher than that of FC layer classification.

Once the vector v has been obtained, its value needs to be transformed into a probability distribution for training purposes, which is achieved by using a *SoftMax* operator as follows:

$$p_i = \text{SoftMax}(v_i) = \frac{e^{v_i}}{\sum_{k=0}^9 e^{v_k}}. \quad (4)$$

The output p_i is the probability of the i th category predicted by the network, where $\sum_{i=0}^9 p_i = 1$.

For the loss function, we select a commonly used cross-entropy loss function. Let the true label be $\hat{y} \in R^{10}$, let i be the true category with $\hat{y}_i = 1$, and let the remaining values be 0. Here, the probability distribution of the network prediction is $p \in R^{10}$ and the cross-entropy loss function can be written as

$$\text{loss}(p, \hat{y}) = - \sum_{k=0}^9 \hat{y}_k \times \log(p_k) = -\log(p_i). \quad (5)$$

From the previous calculation, we can get

$$\log(p_i) = -\log\left(\frac{e^{v_i}}{\sum_{k=0}^9 e^{v_k}}\right) = -v_i + \log\left(\sum_{k=0}^9 e^{v_k}\right). \quad (6)$$

Therefore, the loss function can be expressed as

$$\text{loss}(p, \hat{y}) = \log\left(\sum_{k=0}^9 e^{v_k}\right) - v_i. \quad (7)$$

C. DW Convolution

DW convolution, first used in the Alexnet [23], is the group convolution with the number of groups equal to the number of channels (i.e., each convolution kernel is responsible for feature extraction from one channel).

Let the size of the input feature map be $C \times W \times H$, where C , W and H are the number of channels, width, and height of the feature map, respectively. The DW convolution kernels used in this study are all of the size 3×3 . By the definition of DW

TABLE I
DIFFERENT NETWORKS WITH CORRESPONDING GPU INFERENCE TIME FOR
ONE IMAGE (SEE SECTION IV-B FOR THE HARDWARE ENVIRONMENT)

Network	GPU Inference time(ms)
AlexNet	1.461
ResNet18	3.421
Vgg16	8.605
SqueezeNet1_0	3.284
MobileNetv3_small	6.696
ShuffleNetv2_x0.5	8.137
CDC-net	3.261

convolution, the number of convolution kernel channels is 1, and the number of kernels is C . Let the input feature map be $X \in R^{C \times W \times H}$ and the convolution kernel be $K \in R^{C \times 1 \times 3 \times 3}$, then the output feature map $Y \in R^{C \times W \times H}$ is provided by

$$Y_{m,a,b} = \sum_{\substack{1 \leq i \leq 3, 1 \leq j \leq 3}} K_{m,0,i,j} \times X_{m,a+i-1,b+j-1} \quad (0 \leq m < C, 0 \leq a). \quad (8)$$

The number of convolution kernel parameters for DW convolution is $C \times 1 \times 3 \times 3 = 9C$. However, if a normal convolution kernel is used, the number of parameters is $C \times C \times 3 \times 3 = 9C^2$, with the following parameter ratio:

$$\frac{9C^2}{9C} = C. \quad (9)$$

Therefore, DW convolution reduces the number of parameters by a factor of C .

According to the aforementioned equation, DW convolution has a computational complexity of $3 \times 3 \times C \times W \times H = 9CWH$, whereas ordinary convolution has a computational complexity of $3 \times 3 \times C \times C \times W \times H = 9C^2WH$, with a ratio of

$$\frac{9C^2WH}{9CWH} = C. \quad (10)$$

Therefore, DW convolution can reduce computational complexity by a factor of C .

DW convolution can learn convolutional kernels with strong correlation in a structured manner. When the number of parameters in the network is reduced in this manner, overfitting becomes difficult. This provides regularization-like effects to obtain a more accurate and efficient network. However, Wu et al. [24] noted that DW convolution is unsuitable for graphics processing unit (GPU) computation because the DW convolution kernel reuse rate is much lower, and the in-memory substitution rate is higher compared with those in standard convolution. In addition, because DW convolution is performed for each channel, the operation matrix of each convolution is extremely small, and obtaining complete parallelism with such a small matrix is not easy. As shown in Table I, the MobileNet and ShuffleNet families employing DW convolution do not have favorable GPU inference speeds. Therefore, only one DW convolution is assigned to each block in CDC-net, and the initial downsampling layer is a standard convolution layer.

D. FeatureCopy

In lightweight networks, operations with small parameters and fast inference are critical, but extracting high-dimensional features inevitably increases the number of parameters and computational requirements. Han et al. [25] found that many convolution kernels are similar in high-dimensional convolutional operations, which implies high parameter redundancy.

Therefore, to reduce the number of redundant parameters, some cheap transformation operations can replace some of the convolution operations.

In this study, we redesigned the high-dimensional feature extraction method by adding a FeatureCopy operation to replace half of the channel's high-dimensional DW convolution operations with a feature map copy, as illustrated in Fig. 5.

Let the high-dimensional feature map be $X \in R^{C \times M \times N}$ and the output of the transformation be $X' \in R^{C \times M \times N}$, in which the FeatureCopy operation is

$$X' = X. \quad (11)$$

Let the input feature map be $X \in R^{C_{in} \times M \times N}$. If 1×1 convolution is directly used to ascend to dimension C_{exp} and then 3×3 DW convolution is used to extract the features, the number of parameters required becomes

$$C_{in} \times C_{exp} + 3 \times 3 \times C_{exp}. \quad (12)$$

Here, the required amount of computation is

$$W \times H \times C_{in} \times C_{exp} + W \times H \times 3 \times 3 \times C_{exp}. \quad (13)$$

If the FeatureCopy method is used instead of half of the DW convolution, the number of required parameters is altered to

$$C_{in} \times \frac{C_{exp}}{2} + 3 \times 3 \times \frac{C_{exp}}{2}. \quad (14)$$

Here, the amount of computation required is

$$W \times H \times C_{in} \times \frac{C_{exp}}{2} + W \times H \times 3 \times 3 \times \frac{C_{exp}}{2}. \quad (15)$$

Hence, the ratio of the number of parameters is

$$\frac{C_{in} \times \frac{C_{exp}}{2} + 3 \times 3 \times \frac{C_{exp}}{2}}{C_{in} \times C_{exp} + 3 \times 3 \times C_{exp}} = 50\% \quad (16)$$

and the ratio of computational complexity is calculated as

$$\frac{W \times H \times C_{in} \times \frac{C_{exp}}{2} + W \times H \times 3 \times 3 \times \frac{C_{exp}}{2}}{W \times H \times C_{in} \times C_{exp} + W \times H \times 3 \times 3 \times C_{exp}} = 50\%. \quad (17)$$

Therefore, instead of half of the DW convolution, the FeatureCopy operation can reduce the number of parameters and computational requirements by half, simplifying the network.

E. Channel-Dilation-Concatenation-Block

A network block with a down-dimension, a feature-extraction, and an up-dimension structure is used in numerous deep CNNs, such as ResNet [3] and ResNeXt [26]. This block is based on the a priori assumption that picture features are organized as low-dimensional streams in a high-dimensional space. In small networks, obtaining sufficient decoding parameters for

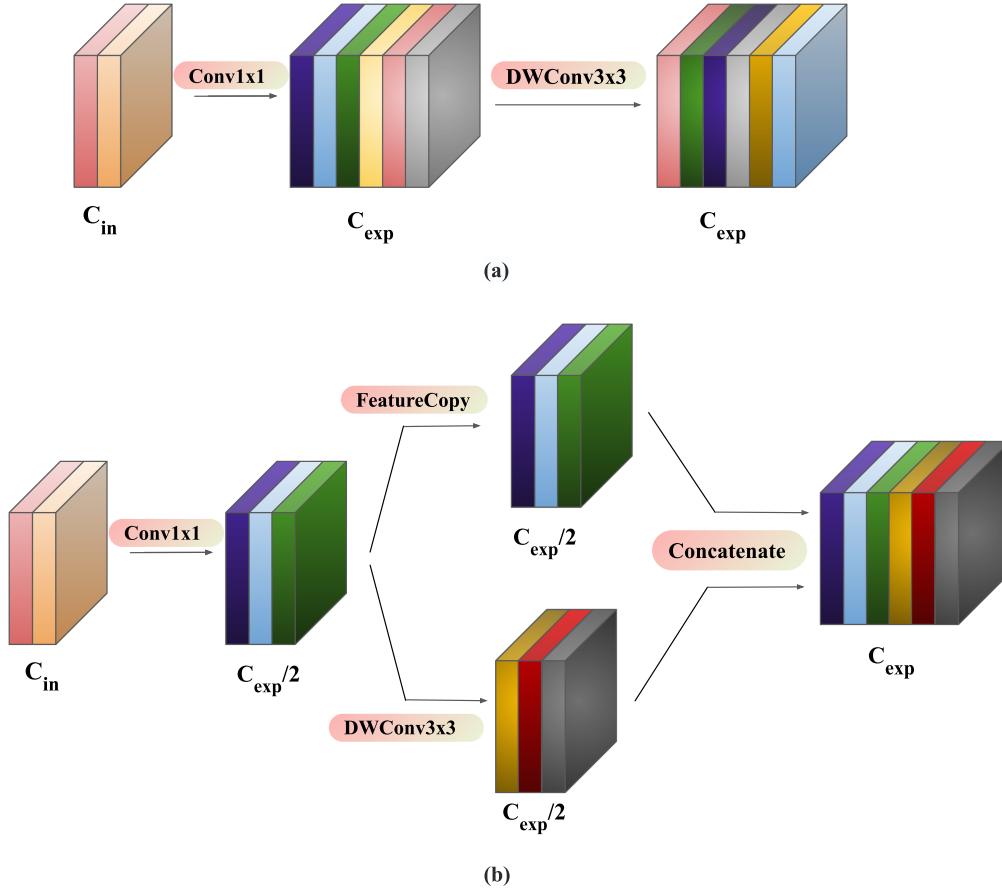


Fig. 5. (a) Original high-dimensional extraction of features and (b) high-dimensional extraction of features with FeatureCopy. Instead of increasing the dimension and then operating on the high-dimensional feature map, FeatureCopy only needs to increase the dimension to half of the original one, copy the feature map at the same time, and perform feature extraction on the feature map. Finally, it concatenates the two feature maps in the channel dimension.

such compression (encoding process) may be difficult, resulting in insufficient network fit. In addition, as mentioned in SqueezeNeXt [27], the lower dimensionality means lower computational performance and requires more layer operations concentrated in the higher-dimensional blocks. Therefore, in this study, we use a structure involving up-dimensioning, feature extraction in higher-dimensional space, and down-dimensioning. However, the utilization of 1×1 convolution to ascend to higher dimensions and the extraction of features in extremely high dimensions inevitably increase the number of parameters and computational requirements, which disobeys lightweight network principles. Therefore, we use the FeatureCopy operation outlined in Section III-D and utilize 1×1 convolution at the end to reduce the dimension to the specified output dimension. The CDC-block structure is illustrated in Fig. 6.

The CDC-block forward propagation algorithm is displayed in Algorithm 1.

F. LIP-Pooling

Downsampling is essential for CNNs, and pooling layers are typically a favorable choice. However, whether average or max pooling layer cannot prevent image information loss.

Algorithm 1: CDC-Block Forward Propagation.

Input: The input feature map $X \in R^{C_{in} \times M \times N}$, the dimension C_{exp} to which it wishes to rise, the final output dimension C_{out}

Output: The output feature map $\tilde{X} \in R^{C_{out} \times M \times N}$.

- 1: $X^{exp} = 1 \times 1 Conv_{C_{in}, C_{\frac{exp}{2}(X)}}$
 - 2: // Perform a dimensional lift
 - 3: $X^{exp} = BN(ReLU(X^{exp}))$
 - 4: $X' = 3 \times 3 DWConv_{C_{\frac{exp}{2}}, C_{\frac{exp}{2}(X^{exp})}}$
 - 5: // Extract features with 3×3 DW convolution
 - 6: $X' = BN(ReLU6(X'))$
 - 7: $X'' = FeatureCopy(X^{exp})$
 - 8: // Copy operation on Channel dimension
 - 9: $X'' = BN(ReLU6(X''))$
 - 10: $X^{cat} = ChannelConcatenate(X', X'')$
 - 11: // Concatenate on Channel dimension
 - 12: $X^{cat} = ReLU(X^{cat})$
 - 13: $\tilde{X} = 1 \times 1 Conv_{C_{exp}, C_{out}}(X^{cat})$
 - 14: // Downscale to the required channel
 - 15: $\tilde{X} = BN(ReLU(\tilde{X}))$
 - 16: **return** \tilde{X}
-

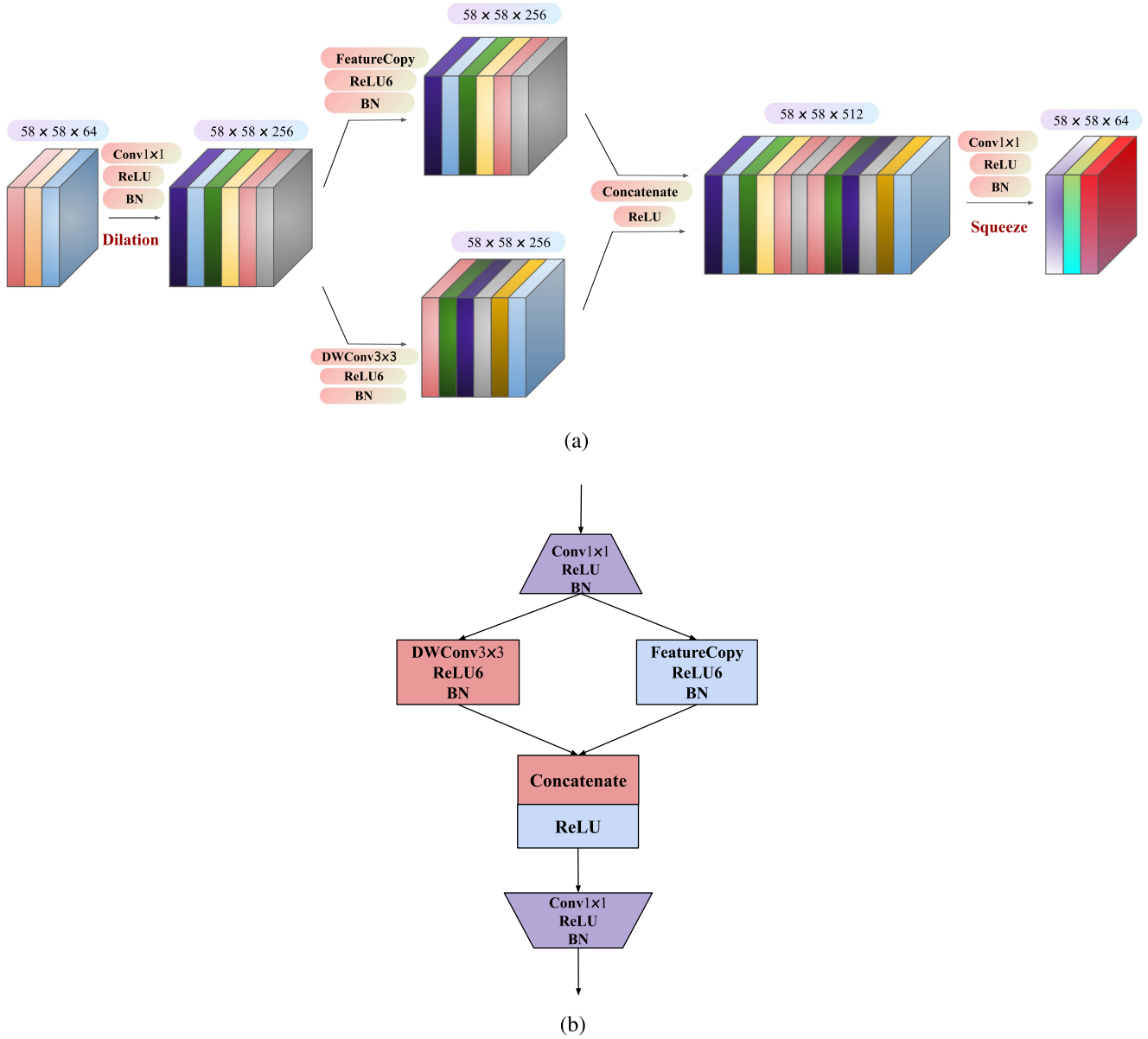


Fig. 6. CDC-block architecture. (a) CDC-block. (b) Simplified CDC-block structure. The 3x3 convolution for extracting information is a DW convolution, and a FeatureCopy operation is used instead of half of the DW convolution operation.

In large networks, such information loss can be compensated by depth. However, in lightweight networks with fewer parameters, an inappropriate pooling strategy may cause the network to lose details, thus hindering the learning process and ultimately leading to a suboptimal model. Therefore, we use LIP [28] to increase the network's accuracy (see Section IV-E for the ablation experiments).

The process of LIP follows the idea of an attention mechanism in which pooling is treated as a weighted sum of each window. Suppose the input is $X \in R^{C \times W \times H}$ and the learned attention weights $W = F(X) \in R^{C \times W \times H}$. To learn the attention weights, the logarithm of the weights $g(X) = \log(W) = \log(F(X))$ should first be determined [in forward propagation, $W = F(X) = \exp(g(X))$].

Let $(|\Delta x|, |\Delta y|) = (3, 3)$, consider a step size of 2, the output $Y \in R^{C \times \frac{W}{2} \times \frac{H}{2}}$ after the LIP pooling layer, let the pooling mapping be

$$[x : x + \Delta x, y : y + \Delta y] \mapsto (\tilde{x}, \tilde{y}). \quad (18)$$

Here, the LIP pooling is calculated as follows:

$$Y_{\tilde{x}, \tilde{y}} = \sum_{x \leq i < x + \Delta x, y \leq j < y + \Delta y} \frac{(W \odot X)_{i,j}}{W_{i,j}}. \quad (19)$$

As indicated by the calculation process illustrated in Fig. 7, to learn the g function, the following structure is employed: A 1×1 convolutional layer for up-dimensioning, a 3×3 convolutional

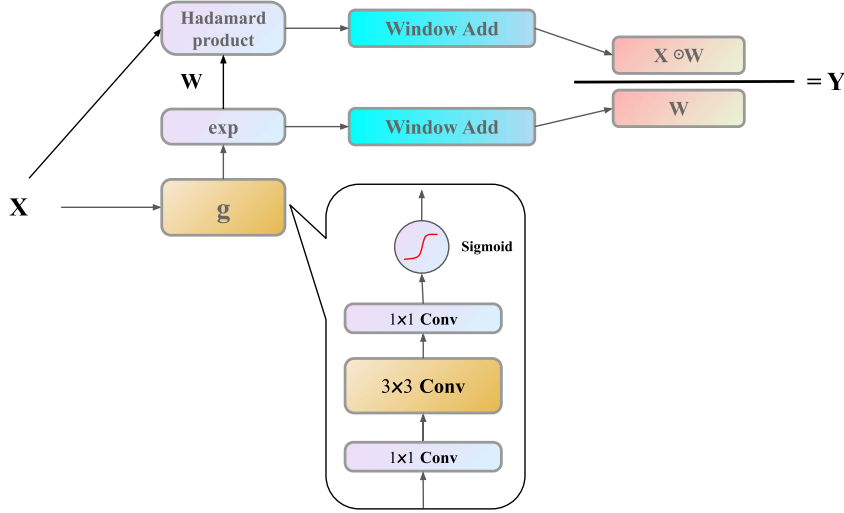


Fig. 7. Schematic of the local importance-based pooling algorithm. The attention parameters are learned from three convolutional layers and then calculated by weighted pooling.

Algorithm 2: Lip-Pooling Forward Propagation.

Input: The input feature map $X \in R^{C \times M \times N}$.
Output: The output feature map $Y \in R^{C \times \frac{M}{2} \times \frac{N}{2}}$.
1: $W = 1 \times 1 \text{Conv}_{C_{in}, 2 \times C_{in}}(X)$
2: // Up-dimension
3: $W = 3 \times 3 \text{Conv}_{2 \times C_{in}, 2 \times C_{in}}(W)$
4: // Using 3×3 convolution to extract weights
5: $W = 1 \times 1 \text{Conv}_{2 \times C_{in}, C_{in}}(W)$
6: // Downscaling to original dimensional values
7: $W = \exp(\sigma(W))$
8: // Transform to get the attention weights
9: **for** $[x : x + \Delta x, y : y + \Delta y] \in \Omega$ **do**
10: $Y_{x,y} = \sum_{x \leq i < x + \Delta x, y \leq j < y + \Delta y} \frac{(W \odot X)_{i,j}}{W_{i,j}}$
11: **endfor**
12: **return** Y

layer for high-dimensional feature extraction, a 1×1 convolutional layer for down-dimensioning, and finally a sigmoid activation function for attention weight mapping between 0 and 1.

The specific algorithm used for the LIP layer is displayed in Algorithm 2.

G. Multilabel Training

Multilabel training does not change the model. Rather, it requires only the transformation of the final SoftMax layer (to obtain the maximum probability label) into a sigmoid layer (for activation with a fixed threshold). Let the final network output value be $v \in R^{10}$ and the output of single-label training be

$$\text{argmax}(\text{SoftMax}(v)). \quad (20)$$

Here, the multilabel training output is

$$\{i | \sigma(v_i) > \alpha\} \quad (21)$$

where α is the activation threshold, with the value of 0.5.

To select the loss function, we treat each label as a binary classification. Here, each input sample corresponds to more than one label, and each label corresponds to a binary classification. Therefore, we select a binary cross-entropy (BCE) loss function. Let the network prediction and the actual label value be $X \in R^{N \times M}$ and $\hat{X} \in R^{N \times M}$, where N is the batch size and M is the total number of categories. Then, the loss function is

$$\text{BCELoss}(X, \hat{X}) = \sum_{0 \leq i < N} \frac{\text{loss}(X_i, \hat{X}_i)}{N} \quad (22)$$

where

$$\text{loss}(X_i, \hat{X}_i) = \sum_{0 \leq j < M} \frac{l(X_{i,j}, \hat{X}_{i,j})}{M} \quad (23)$$

and

$$l(X_{i,j}, \hat{X}_{i,j}) = - \left[\hat{X}_{i,j} \times \log(X_{i,j}) + (1 - \hat{X}_{i,j}) \times \log(1 - X_{i,j}) \right]. \quad (24)$$

Therefore, the BCE loss function is expressed as

$$\text{BCELoss}(X, \hat{X}) = - \frac{1}{NM} \sum_{0 \leq i < N} \sum_{0 \leq j < M} \left[\hat{X}_{i,j} \times \log(X_{i,j}) + (1 - \hat{X}_{i,j}) \times \log(1 - X_{i,j}) \right]. \quad (25)$$

IV. EXPERIMENTS

A. LSCIDMR Dataset

The LSCIDMR dataset built by Bai et al. [29] is the first publicly available benchmark database of satellite cloud images for meteorological research. It can be treated as a crucial guide for using deep-learning methods in satellite meteorological image classification. The dataset contains satellite images classified by meteorological experts into 11 categories (including a label-less category).

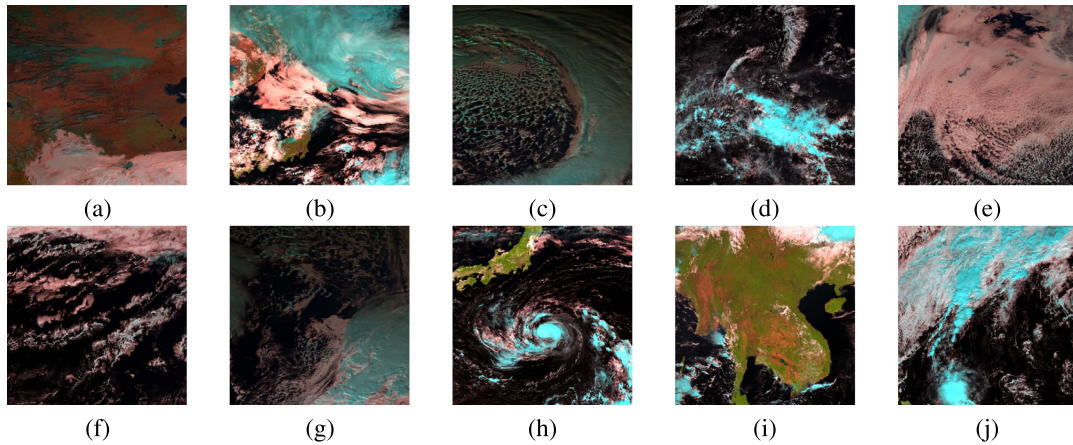


Fig. 8. Samples of single-label data in ten classes. (a) Desert. (b) Extratropical cyclone. (c) Frontal surface. (d) High ice cloud. (e) Low water cloud. (f) Ocean. (g) Snow. (h) Tropical cyclone. (i) Vegetation. (j) Westerly jet.

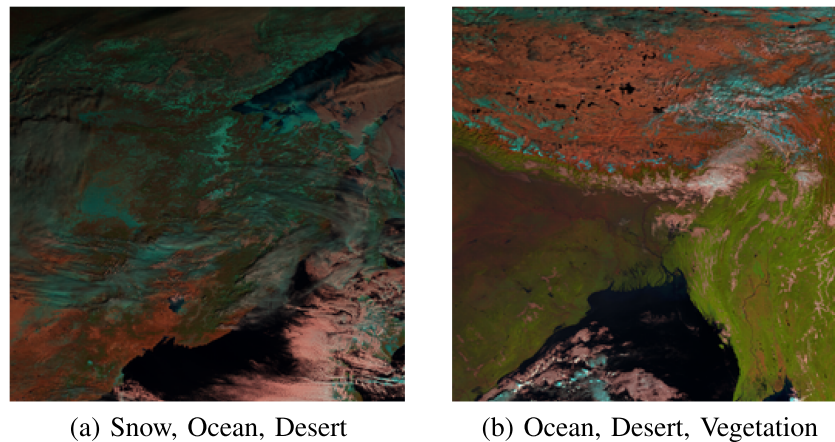


Fig. 9. Samples of multilabel data in several categories. The captions indicate the categories.

Based on the method outlined in Section III-A, Bai et al. generate 104 390 high-resolution images (256×256) by two annotation strategies: 1) LSCIDMR-s with single-label annotation; 2) LSCIDMR-m with multilabel annotation. By manually annotating the labels, Bai et al. obtained 414 221 multilabel data and 40 625 single-label data for the following experiments. Example images are displayed in Figs. 8 and 9.

B. Experimental Configuration

We use the LSCIDMR dataset for training, in which 20% is set as the test set. The model training is performed in an Ubuntu environment by a single NVIDIA 3090 GPU. The model is initialized using Xavier uniform initialization with the Adam optimizer, the initial learning rate is set to 0.001, and the learning rate decay strategy is used. The batch size used for training is 128, and 30 epochs are trained.

Table II lists the experimental parameters.

C. Single-Label Data Experiments

1) *Comparison of Classification Performance on Single-Label Data:* The accuracy obtained in the experiments is listed

TABLE II
EXPERIMENTAL ENVIRONMENTAL PARAMETERS

Experiment	Parameters
Dataset	LSCIDMR-S/LSCIDMR-M[29]
Environment	Ubuntu
Framework	Pytorch 1.11.0
CPU	Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz
GPU	NVIDIA GeForce RTX 3090
Memory	16GB
Initialization	Xavier uniform[30]
Optimal	Adam[31](beta1=0.9, beta2=0.999, eps=1e-8)
Learning rate	Initial 0.001, cosine function decay[32]
Batch Size	128
Epoch	30
Testing ratio	0.2

in Table III, and their test accuracy during the training procedure is shown in Fig. 10.

CDC-net is superior to the other three commonly used lightweight networks in inference time, and provides an extra 3% to 6% accuracy enhancement, indicating the effectiveness of CDC-net in balancing accuracy and lightweight.

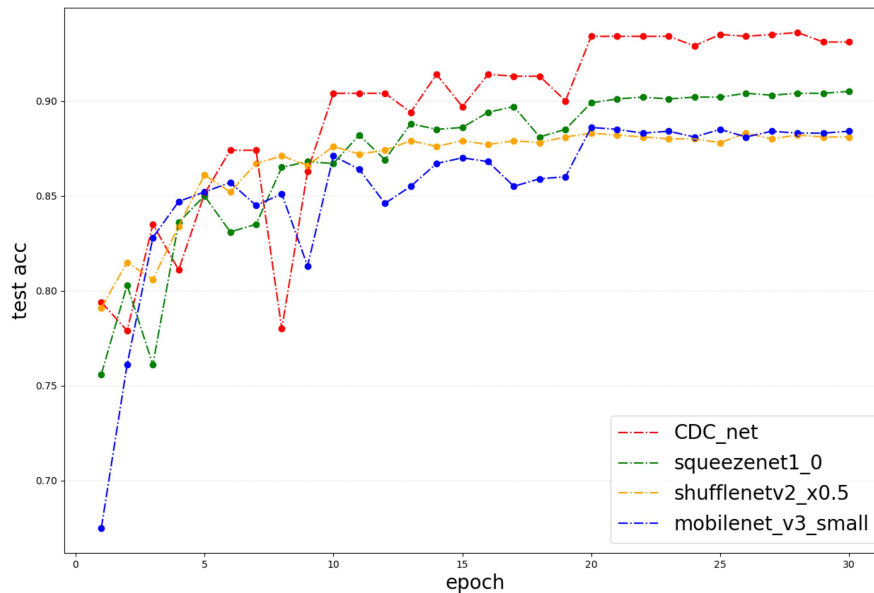


Fig. 10. Different networks' test accuracy during the training procedure for single-label datasets.

TABLE III
ACCURACY OF CDC-NET AND COMMONLY USED LIGHTWEIGHT NETWORKS

Network	Accuracy
shufflenet_v2_x0.5	87.83%
mobilenet_v3_small	88.10%
squeezenet1_0	90.42%
CDC-net	93.56%

The bold entities indicate the best result of the comparison methods.

The training curve indicates that CDC-net rapidly converges at the beginning of the training process. The accuracy remains consistently higher than the other networks, confirming that CDC-net has excellent convergence speed. Because of its favorable generalization ability, CDC-net does not reach a bottleneck until approximately the 20th epoch, while other networks become stuck at approximately the 15th epoch.

2) *Confusion Matrix*: To further investigate the classification capability of CDC-Net, we plotted the classification confusion matrix of ShuffleNetv2, MobileNetv3, SqueezeNet1_0, and CDC-net, as shown in Fig. 11.

As observed in the confusion matrix, the desert, snow, ocean, and vegetation categories are all accurately identified because the samples for these categories have distinctive features.

However, FrontalSurface is the most challenging category to be distinguished and is often confused with Ex-tropicalCyclone. The reason is that the sample size for FrontalSurface is only 1.56%, but 12.27% for ExtropicalCyclone. Such sample imbalance will make our classification model tend to predict samples with more data.

According to the confusion matrices of the four networks, CDC-net is the most accurate one, and in the most challenging categories, it gains improvement of 20% to 25%.

D. Multilabel Data Experiments

1) *Evaluation Approach*: We assume that the dataset comprises N images. We denote \hat{L}_k to be the set that contains all the predicted label(s) for the K th sample, and denote L_k to be the set that contains all the label(s) for the K th sample. The multilabel model can be evaluated using the following three measures.

a) *F1-score*: Precision is the percentage of correctly predicted labels relative to all predicted labels, which is calculated as follows:

$$Precision = \frac{1}{N} \sum_{k=1}^N \left(\frac{||L_k \cap \hat{L}_k||}{||\hat{L}_k||} \right). \quad (26)$$

Recall is the percentage of correctly predicted labels relative to all true labels, which is calculated as follows:

$$Recall = \frac{1}{N} \sum_{k=1}^N \left(\frac{||L_k \cap \hat{L}_k||}{||L_k||} \right). \quad (27)$$

Because of the mutual exclusivity of precision and recall in some cases, it is impossible to use these two methods directly to evaluate network performance. However, it is feasible to take their harmonic mean to denote the F1 score

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall}. \quad (28)$$

b) *Accuracy*: Accuracy is the percentage of correctly predicted tags relative to the total tags, which is calculated as follows:

$$Accuracy = \frac{1}{N} \sum_{k=1}^N \left(\frac{||L_k \cap \hat{L}_k||}{||L_k \cup \hat{L}_k||} \right). \quad (29)$$

c) *AbsoluteTrue*: AbsoluteTrue is the most stringent metric and the optimal indicator of a network's multilabel classification performance. For the k th image, the AbsoluteTrue value

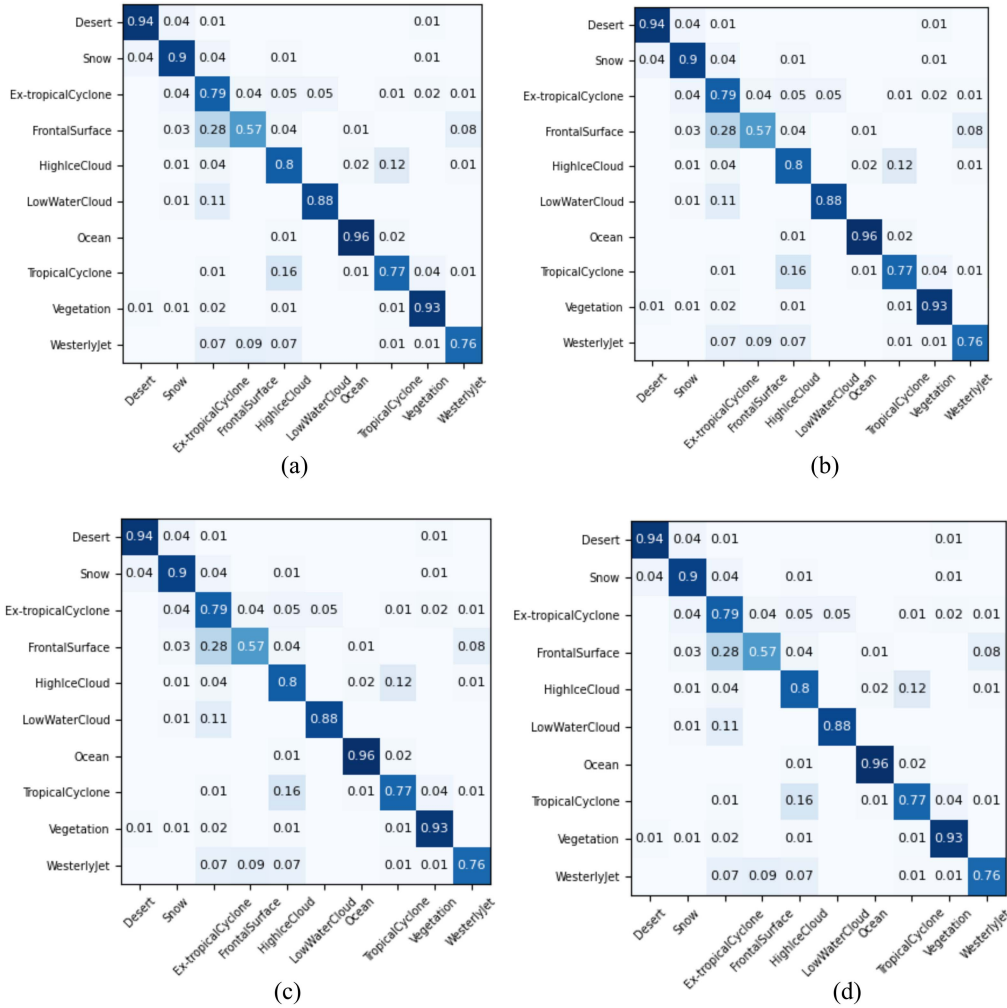


Fig. 11. Comparison of the the confusion matrices between CDC-net and other commonly used lightweight networks: (a) ShuffleNetV2_x0.5. (b) MobileNetV3_small. (c) SqueezeNet1_0. (d) CDC-net.

TABLE IV

COMPARISON OF DIFFERENT EVALUATION METRICS BETWEEN COMMONLY USED LIGHTWEIGHT NETWORKS AND CDC-NET WITH MULTILABEL DATASETS

Network	F1-score(%)	Accuracy(%)	AbsoluteTrue(%)
squeezenet1_0	94.98	90.33	68.43
shufflenet_v2_x0.5	96.23	93.01	75.84
mobilenet_v3_small	96.59	93.42	76.95
CDC-net	97.17	94.21	79.50

The bold entities indicate the best result of the comparison methods.

is 1 if and only if $L_k = \hat{L}_k$; otherwise, it is 0. The AbsoluteTrue value is calculated as follows:

$$AbsoluteTrue = \frac{1}{N} \sum_{k=1}^N \mathbb{I}(L_k = \hat{L}_k) \quad (30)$$

where \mathbb{I} is the indicator function (i.e., 1 when $L_k = \hat{L}_k$ and 0 otherwise).

2) *Comparison of Classification Performance on Multilabel Data:* Table IV and Fig. 12 present the results of the experiments conducted on the performance of various networks with multilabel datasets.

TABLE V

COMPARISON OF THE PERFORMANCE OF CDC-NET AND ALEXNET

Network	F1-score(%)	Accuracy(%)	AbsoluteTrue(%)
Alexnet	97.05	93.59	77.67
CDC-net	97.17	94.21	79.50

The bold entities indicate the best result of the comparison methods.

As shown in Fig. 12, CDC-net has the highest performance on the three evaluation metrics (i.e., F1 score, accuracy, and AbsoluteTrue), indicating its effectiveness in multilabel tasks.

By contrast, SqueezeNet performs worst. Its reason lies in its adoption of the channel squeeze operation causes a large amount of feature information to be lost, adversely affecting its performance in multilabel classification.

To investigate the effect of the number of parameters on multiclassification tasks, we tested AlexNet [23] with up to 57 million parameters. The results are presented in Table V.

Although AlexNet, with a sufficient number of parameters, achieves satisfactory results in the multi-classification task, CDC-net has higher accuracy with only 2% of its parameters, again confirming the effectiveness of CDC-net.

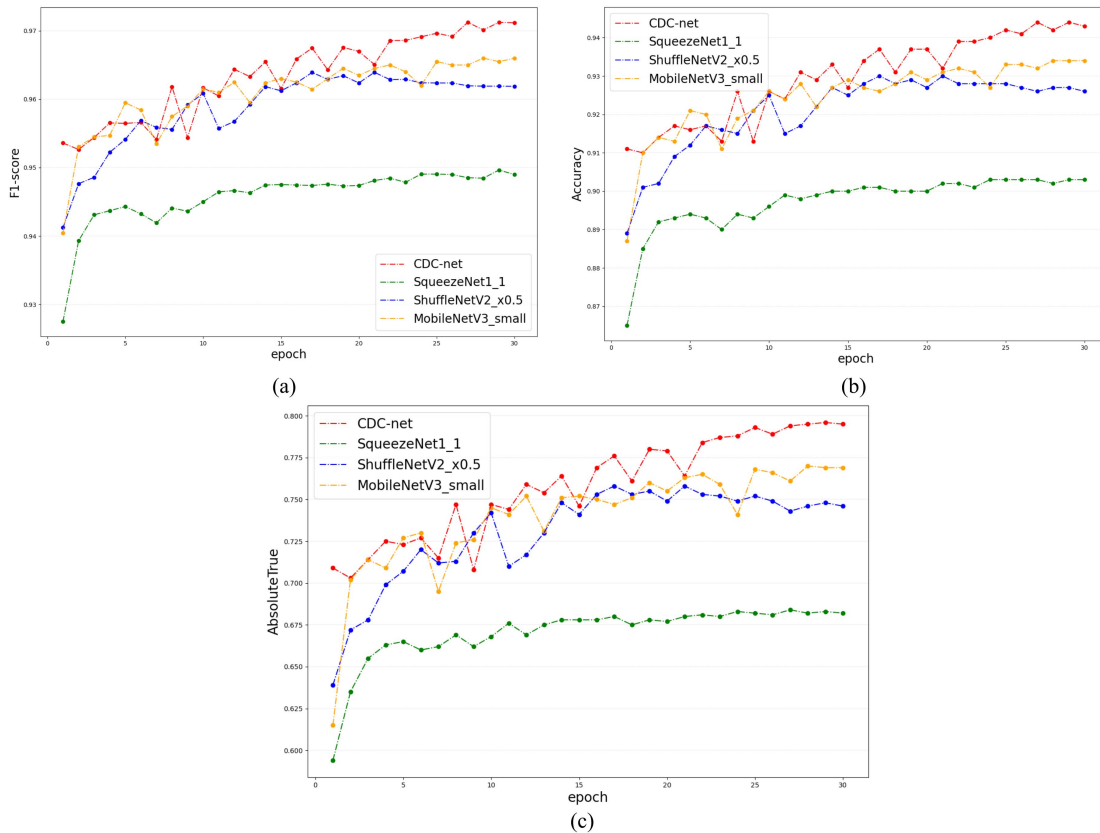


Fig. 12. Different networks' evaluation metrics during the training procedure for multilabel datasets. (a) F1-score. (b) Accuracy. (c) AbsoluteTrue.

TABLE VI

COMPARISON OF CDC-NET WITH OTHER COMMONLY USED LIGHTWEIGHT NETWORKS REGARDING INFERENCE SPEED AND THE NUMBER OF PARAMETERS (SEE SECTION IV-B FOR THE HARDWARE ENVIRONMENT)

Network	GPU Inference Time (ms)	Number of Parameters (M)
shufflenet_v2_x0.5	8.137	1.24
mobilenet_v3_small	6.696	2.54
squeezenet1_0	3.284	1.25
CDC-net	3.261	1.12

The bold entities indicate the best result of the comparison methods.

E. Comparison of Network Parameters

We compared the GPU inference times and model parameters for three commonly used lightweight networks with CDC-net, as shown in Table VI.

CDC-net was discovered to have the fewest parameters and shortest inference time.

The reason lies in that it uses DW convolution and Feature-Copy operations, it has lower depth and uses fewer blocks than the other networks.

F. Ablation Experiments

The results of the ablation experiments are presented in Table VII. Groups 1 and 3 revealed that a global average pooling layer considerably improved the network's accuracy. Therefore,

TABLE VII

RESULTS OF ABLATION EXPERIMENT, WHERE A TICK INDICATES THAT THIS METHOD WAS USED

Group/Acc(%)	GlobalAvgpool	DWConv	Lip-pool
1/84.61		✓	
2/92.19	✓		
3/92.17	✓	✓	
4/93.56	✓	✓	✓

a global average pooling layer was used instead of a FC layer. As shown in Groups 2 and 3, ordinary convolution did not considerably improve the performance. Therefore, DW convolution was used for the purpose of lightweight. Finally, as shown in Groups 3 and 4, LIP layers lead to a relatively significant increase in accuracy.

G. Visual Analysis

Class activation mapping (CAM) [33] is a critical tool in analyzing networks and their performances. In this study, we performed global average pooling on the CDC-net's last feature map, calculated each channel's mean value, and mapped each value to the corresponding values of all classes through FC layers.

Then, the gradient of class output with the highest probability of network prediction relative to the last feature map was calculated and visualized on the original map.

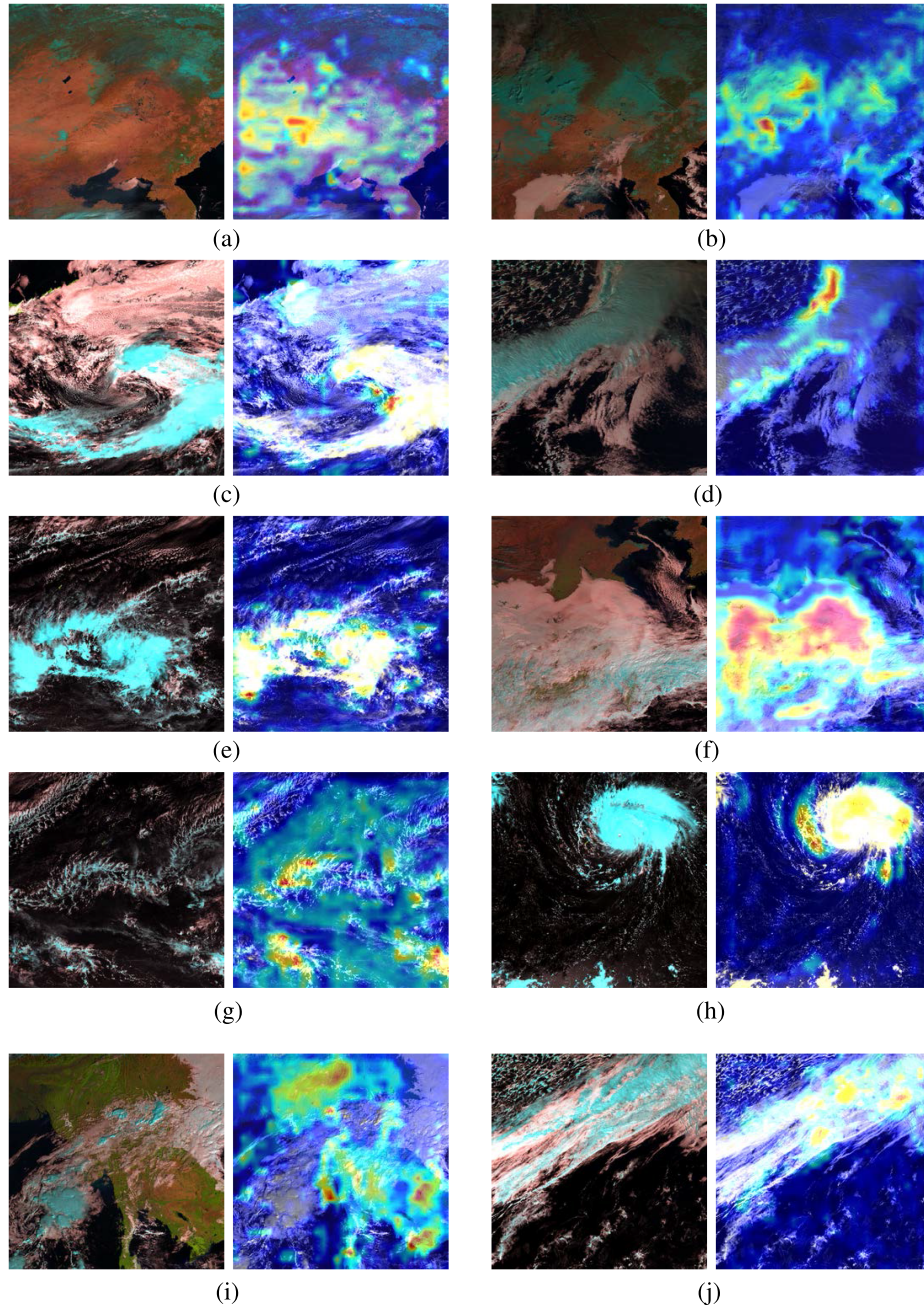


Fig. 13. CAM visualization results. The image shown on the left is the original image, whereas that shown on the right is the result of the CAM visualization. (a) Desert. (b) Snow. (c) Extra-tropicalCyclone. (d) FrontalSurface. (e) HighIceCloud. (f) LowWaterCloud. (g) Ocean. (h) TropicalCyclone. (i) Vegetation. (j) WesterlyJet.

Subsequently, all ten categories were selected for visualization and analysis, and the strongly highlighted parts represent what the network is more concerned about. The results are presented in Fig. 13. CDC-net correctly focuses on the true category regions and pays less attention to the remaining irrelevant regions, providing a high classification performance.

V. DISCUSSION

In this section, another criterion is utilized to evaluate the performances of CDC-net and several lightweight networks.

Besides, the application in object detection and semantic segmentation with our CDC-net will be discussed below.

Tan et al. [34] used reinforcement learning for network architecture search, and their reward was defined as

$$\underset{m}{\text{maximize}} ACC(m) \times \left[\frac{LAT(m)}{T} \right]^{-0.07} \quad (31)$$

where m is the given network, $ACC(m)$ denotes its accuracy, $LAT(m)$ denotes the inference latency, and T is the target latency which is additionally defined. It can be seen that the larger the reward for different networks m , the better the balance

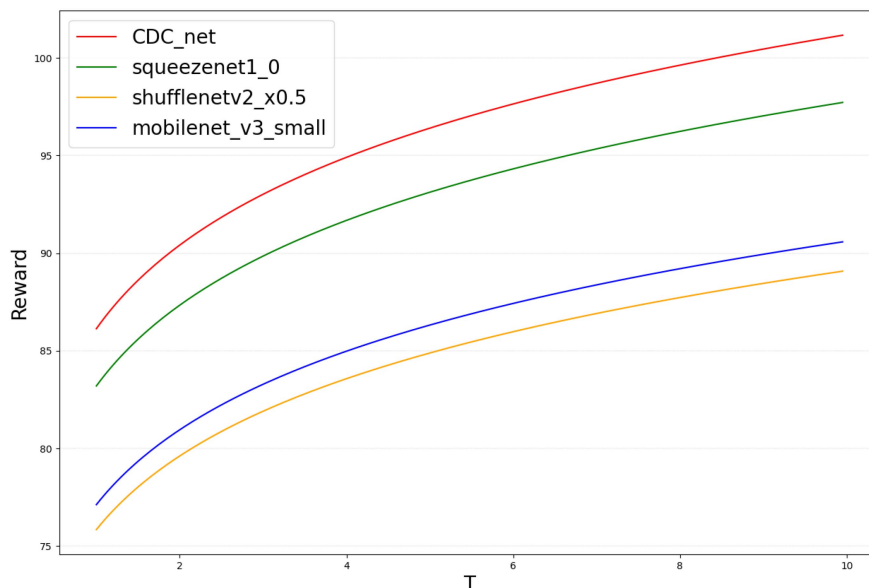


Fig. 14. Curve of Reward versus T value. It can be seen that the curve of our CDC-net is higher than that of other lightweight networks.

between precision and accuracy. For different values of T , we plot the T values against the reward (Fig. 14). It can be seen that the curve of our CDC-net is much higher than that of other well-known lightweight networks, which demonstrates the superior performance of the CDC-net over other famous lightweight networks.

A lightweight backbone is essential for target detection or semantic segmentation tasks requiring high real-time performance. Our CDC-net meets the lightweight requirement and can generate feature maps of different sizes during downsampling, which can easily realize the feature fusion required in target detection or semantic segmentation. In addition, our proposed CDC-block can also replace the convolutional blocks in target detection or semantic segmentation networks, speeding up their inference and reducing their number of parameters for easy deployment in end devices.

VI. CONCLUSION

This article presents a lightweight network termed CDC-net for faster and lighter meteorological satellite image classification. It helps process remotely sensed images and solve the problem of real-time single-label and multilabel meteorological satellite image classification. A lightweight convolutional network module termed CDC-block is designed to compose CDC-net, which extracts features in a high-dimensional space and utilizes the FeatureCopy operation. Meanwhile, to further improve the network accuracy, we introduce the LIP pooling layer based on the attention mechanism. The results indicated that the accuracy of CDC-net is 3% to 6% higher than that of the commonly used lightweight networks in single-label classification. When it comes to multilabel classification, CDC-net also performs best among all the networks. Similarly, in inference speed, CDC-net is the fastest, with only 1.12 million parameters, which is lower than other lightweight networks. These results

reveal that CDC-net can be embedded in a spacecraft with little memory to process remotely sensed images and perform real-time classification of weather satellite images. It can also be served as a backbone network for real-time target detection and semantic segmentation models of satellite remote-sensing images.

ACKNOWLEDGMENT

We gratefully thank the creators of the LSCIDMR dataset and the server support from Shandong University, Linyi University, and Jiangsu University of Science and Technology.

REFERENCES

- [1] F. Liu, J. Fu, Q. Wang, and R. Zhao, "Tensor dictionary self-taught learning classification method for hyperspectral image," *Remote Sens.*, vol. 14, no. 17, 2022, Art. no. 4373.
- [2] J. Xi, O. K. Ersoy, M. Cong, C. Zhao, W. Qu, and T. Wu, "Wide and deep Fourier neural network for hyperspectral remote sensing image classification," *Remote Sens.*, vol. 14, no. 12, 2022, Art. no. 2931.
- [3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 770–778.
- [4] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 2017, pp. 2261–2269.
- [5] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "SqueezeNet: Alexnet-level accuracy with 50x fewer parameters and < 1 mb model size," *CoRR*, vol. abs/1602.07360, 2016, [Online]. Available: <http://arxiv.org/abs/1602.07360>
- [6] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," Apr. 2017, arXiv:1704.04861.
- [7] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 4510–4520.
- [8] A. Howard et al., "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Seoul, South Korea, 2019, pp. 1314–1324.
- [9] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 6848–6856.

- [10] N. Ma, X. Zhang, H. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. 15th Eur. Conf. Comput. Vis.*, Munich, Germany, Springer, 2018, pp. 122–138.
- [11] W. Chen, S. Ouyang, W. Tong, X. Li, X. Zheng, and L. Wang, "GCSANet: A global context spatial attention deep learning network for remote sensing scene classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 1150–1162, Jan. 2022, doi: [10.1109/JSTARS.2022.3141826](https://doi.org/10.1109/JSTARS.2022.3141826).
- [12] Y. Fan et al., "MSLAENet: Multi-scale learning and attention enhancement network for fusion classification of hyperspectral and LiDAR data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 10041–10054, Nov. 2022, doi: [10.1109/JSTARS.2022.3221098](https://doi.org/10.1109/JSTARS.2022.3221098).
- [13] H. Zhang, J. Yao, L. Ni, L. Gao, and M. Huang, "Multimodal attention-aware convolutional neural networks for classification of hyperspectral and LiDAR data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, early access, pp. 1–10, 2022, doi: [10.1109/JSTARS.2022.3187730](https://doi.org/10.1109/JSTARS.2022.3187730).
- [14] B. Tu, W. He, W. He, X. Ou, and A. Plaza, "Hyperspectral classification via global-local hierarchical weighting fusion network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 184–200, 2022, doi: [10.1109/JSTARS.2021.3133009](https://doi.org/10.1109/JSTARS.2021.3133009).
- [15] C. Huang, C. Bai, S. Chan, and J. Zhang, "MMSTN: A multi-modal spatial-temporal network for tropical cyclone short-term prediction," *Geophysical Res. Lett.*, vol. 49, no. 4, 2022, Art. no. e2021GL096898.
- [16] C. Bai, F. Sun, J. Zhang, Y. Song, and S. Chen, "Rainformer: Features extraction balanced network for radar-based precipitation nowcasting," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, Mar. 2022, doi: [10.1109/LGRS.2022.3162882](https://doi.org/10.1109/LGRS.2022.3162882).
- [17] R. Hang, X. Qian, and Q. Liu, "Cross-modality contrastive learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–12, Jul. 2022, doi: [10.1109/TGRS.2022.3188529](https://doi.org/10.1109/TGRS.2022.3188529).
- [18] C. Bai, D. Zhao, M. Zhang, and J. Zhang, "Multimodal information fusion for weather systems and clouds identification from satellite images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 7333–7345, Aug. 2022.
- [19] J. Zhang, P. Liu, F. Zhang, and Q. Song, "CloudNet: Ground-based cloud classification with deep convolutional neural network," *Geophysical Res. Lett.*, vol. 45, no. 16, pp. 8665–8672, 2018.
- [20] R. Hang, P. Yang, F. Zhou, and Q. Liu, "Multiscale progressive segmentation network for high-resolution remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–12, Sep. 2022, doi: [10.1109/TGRS.2022.3207551](https://doi.org/10.1109/TGRS.2022.3207551).
- [21] R. Hang, Q. Liu, and Z. Li, "Spectral super-resolution network guided by intrinsic properties of hyperspectral imagery," *IEEE Trans. Image Process.*, vol. 30, pp. 7256–7265, Aug. 2021, doi: [10.1109/TIP.2021.3104177](https://doi.org/10.1109/TIP.2021.3104177).
- [22] M. Lin, Q. Chen, and S. Yan, "Network in network," in *Proc. 2nd Int. Conf. Learn. Representations*, Banff, AB, Canada, Apr. 14–16, 2014. [Online]. Available: <http://arxiv.org/abs/1312.4400>
- [23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, Nevada, United States, 2012, pp. 1106–1114. [Online]. Available: <https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html>
- [24] B. Wu et al., "Shift: A zero flop, zero parameter alternative to spatial convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 9127–9135.
- [25] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More features from cheap operations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, 2020, pp. 1577–1586.
- [26] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 2017, pp. 5987–5995, doi: [10.1109/CVPR.2017.634](https://doi.org/10.1109/CVPR.2017.634).
- [27] A. Gholami et al., "SqueezeNext: Hardware-aware neural network design," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Salt Lake City, UT, USA, 2018, pp. 1638–1647.
- [28] Z. Gao, L. Wang, and G. Wu, "LIP: Local importance-based pooling," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Seoul, South Korea, 2019, pp. 3354–3363, doi: [10.1109/ICCV.2019.00345](https://doi.org/10.1109/ICCV.2019.00345).
- [29] C. Bai, M. Zhang, J. Zhang, J. Zheng, and S. Chen, "LSCIDMR: Large-scale satellite cloud image database for meteorological research," *IEEE Trans. Cybern.*, vol. 52, no. 11, pp. 12538–12550, Nov. 2022.
- [30] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, Sardinia, Italy, 2010, pp. 249–256. [Online]. Available: <http://proceedings.mlr.press/v9/glorot10a.html>
- [31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, San Diego, CA, USA, May 7–9, 2015. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [32] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *Proc. 5th Int. Conf. Learn. Representations*, Toulon, France, Apr. 24–26, 2017. [Online]. Available: <https://openreview.net/forum?id=Skq89Scxx>
- [33] B. Zhou, A. Khosla, Á. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 2921–2929, doi: [10.1109/CVPR.2016.319](https://doi.org/10.1109/CVPR.2016.319).
- [34] M. Tan et al., "MnasNet: Platform-aware neural architecture search for mobile," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, 2019, pp. 2820–2828.



Shuyao Shang received the undergraduate degree from the School of Mechanical and Information Engineering, Shandong University, Weihai, China.

His research interests include computer vision and pattern recognition.



Jinglin Zhang received the B.E. degree in communication engineering from the South Central University for Nationalities, Wuhan, China, the M.E. degree in electronic circuit and system from Shanghai University, Shanghai, China, and the Ph.D. degree in electronics and communication engineering from the National Institute of Applied Sciences, Rennes, France, in 2007, 2010, and 2013, respectively.

He is currently a Professor with the School of Control Science and Engineering, Shandong University, Jinan, China. His research interests include computer vision and interdisciplinary research with pattern recognition and atmospheric science.



Xing Wang received the B.E. degree from Linyi Normal University, Linyi, China, and the Ph.D. degree in computer application technology from Northeastern University, Shenyang, China, in 2006 and 2011, respectively.

He is currently a Professor with the School of Information Science and Engineering, Linyi University, Linyi, China. His research interests include image processing and knowledge graph.



Xinghua Wang received the B.E. degree in computer science and technology from the Qilu Institute of Technology, Jinan, China, in 2020. He is currently working toward the master's degree in computer science and technology with the College of Information Science and Technology, Linyi University, Linyi, China.

His research interest includes image processing.



Yuanjun Li received the B.E. and M.E. degrees in transportation engineering from Hohai University, Nanjing, China, in 2015 and 2018, respectively. She is currently working toward the Ph.D. degree in naval architecture and ocean engineering with Ocean College, Jiangsu University of Science and Technology, Zhenjiang, China.

Her research interests include remote sensing and big data prediction.



Yuanjiang Li received the M.S.E. degree in signal and information processing from the Jiangsu University of Science and Technology, Zhenjiang, China, and the Ph.D. degree in information and communication engineering from the Nanjing University of Science and Technology, Nanjing, China, in 2006 and 2013, respectively.

Since 2006, he has been with the Faculty of School of Electronic Information, Jiangsu University of Science and Technology. His current research interests include big data, computer vision, high-performance computing, interdisciplinary research with pattern recognition, and fault diagnosis.