https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

CloudMViT: Cloud Classification Using Ground-Based Remote Sensing Imagery and a Lightweight Hybrid Architecture

Wei Xu[12], Ningning Wu[1], Lin Feng[1]

[1]School of Electronic and Information Engineering, Nanjing University of Information Science & Technology, Nanjing 210044, China
[2]Jiangsu Key Laboratory of Meteorological Observation and Information Processing, Nanjing University of Information Science & Technology, Nanjing 210044, China

Correspondence to: Wei Xu, Professor, Ph.D.(xw@nuist.edu.cn)

Abstract. Ground-based remote sensing cloud image data can be used to analyze regional cloud type variation trends, thereby predicting future water resource supply capacity. However, existing cloud classification methods based on ground-based remote sensing imagery often suffer from limited recognition accuracy due to insufficient fine-grained feature extraction, and their large model parameter counts hinder deployment on embedded terminals. To address these issues, this study proposes CloudMViT, a lightweight hybrid network architecture fusing a dual-pooling channel attention module and cross-scale self-attention, which enhances both local and global feature representation of cloud images while optimizing computational efficiency. Specifically, the model suppresses sky background interference and strengthens cloud edge features via the dual-pooling channel attention module that combines global average pooling (GAP) and global max pooling (GMP); captures cross-channel detailed features (e.g., cirrus fibril structures and stratocumulus shadows) using depthwise separable convolution and a decoupling mechanism; and further reduces model parameters by introducing CloudGhost cascade compression technology through linear feature redundancy elimination.Experiments on the World Meteorological Organization (WMO)-compliant HBMCD (10 standard cloud genera) and GCD (7 sky conditions) datasets demonstrate that CloudMViT achieves classification accuracies of 98.81% and 95.13%, respectively, significantly outperforming lightweight models such as MobileViT and EfficientNet. Ablation experiments validate the effectiveness of the dual-pooling channel attention module (improving accuracy by 5.31%) and the CloudGhost module (increasing inference speed by 50%). When deployed on the RK3588 embedded platform, the INT8-quantized CloudMViT enables real-time inference, maintaining an accuracy of 94.79% with only 0.47 MB of memory occupation. The proposed cloud classification method and hardware acceleration strategy provide a feasible solution for the development of portable ground-based cloud observation and classification devices.

1 Introduction

Clouds are hydrometeors composed of tiny particles of liquid water, ice, or both, suspended in the atmosphere and typically not in contact with the ground. Approximately 68% of the Earth's surface is covered by clouds, making them one of the key factors influencing global climate change. Different cloud types are closely linked to precipitation intensity: cumulonimbus

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

clouds usually indicate heavy precipitation, thunderstorms, or even hail, while stratocumulus clouds bring light to moderate precipitation.

The World Meteorological Organization (WMO) classifies clouds into 10 basic genera (e.g., cumulus, cumulonimbus, stratocumulus, stratus) based on their shape and structure, and has published the latest *International Cloud Atlas*. As cloud density gradually decreases, the boundary between clouds and the sky becomes blurred, and clouds exhibit complex textural features—posing significant challenges for ground-based cloud image classification.

All-sky imagers, as ground-based cloud observation devices, provide real-time sky cloud imagery. Traditional cloud classification methods mainly rely on Fourier transforms and statistical texture data of cloud images, with classification accuracies typically below 70%. Furthermore, the cloud types distinguishable by these methods do not fully cover the WMO's classification standards.

In 2017, Ye et al. proposed DeepCloud, the first CNN-based cloud classification algorithm, which infers global categories from local cloud features. By combining vector encoding and discriminative local pattern mining techniques, the algorithm improved the average classification accuracy to 86.7%. However, it requires aggregating a large number of local features, consuming over 1 TB of memory during data clustering in shallow convolutional layers, resulting in high computational complexity. Additionally, the small dataset size means the algorithm's generalization ability has not been fully validated.

In 2018, Zhang et al. constructed the Cirrus Cumulus Stratus Nimbus (CCSN) dataset and proposed CloudNet, a novel cloud classification algorithm. Through data augmentation and stochastic gradient descent strategies, the precision, recall, and F1-score for 11 cloud categories ranged from 86% to 96%, with an overall average accuracy of 88%. Nevertheless, the algorithm struggles to distinguish fine-grained features between altostratus and stratocumulus, cirrostratus and stratocumulus, as well as stratocumulus and stratus.

In 2019, Jia et al. built the high-quality HBMCD dataset and fine-tuned the DenseNet201 model using transfer learning.Leveraging a unique dense connection mechanism, the model achieves cloud feature reuse, reduces the loss of key cloud information, and ultimately improves the classification accuracy to 96%. However, the network model exhibits high parameter complexity with a depth of 201 layers, making it challenging to embed in mobile observation devices.

In 2020, to reduce the computational load during training and inference, Wang et al. proposed CloudA, a lightweight algorithm with a simple structure consisting of only four convolutional layers, four pooling layers, and three fully connected layers. It achieved an accuracy exceeding 95% on the SWIMCAT dataset. Nevertheless, the dataset contains fewer than 800 samples, failing to comply with the World Meteorological Organization (WMO) cloud classification standards.

In 2021, Durrani et al. first built GCD, a new benchmark dataset containing 19,000 cloud images, and investigated classification algorithms based on graph neural networks and attention mechanism techniques. The GCD dataset lays a solid foundation for the validation of innovative cloud image classification algorithms.

In 2022, Li et al. combined the advantages of CNNs and Transformers to simultaneously extract local and global cloud features, aiming to improve the accuracy of ground-based cloud classification. However, during feature fusion, the model suffers from either the loss of cloud boundary information or feature redundancy.

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

In 2023, Liang et al. proposed MMST, a novel cloud classification model that employs the Transformer architecture for global cloud feature modeling and integrates multi-modal information (temperature, humidity, air pressure, wind speed) to further enhance classification performance. The overall classification accuracy reaches 91.30%, but the model performs poorly on mixed clouds, with precision, recall, and F1-score of only 79.32%, 75.28%, and 77.51%, respectively. In the same year, Li et al. proposed a neural network-based classification method for cloud layer type identification from ground-based cloud images, achieving accurate discrimination of different cloud layers and types. However, it fails to address the redundancy issue during feature fusion. Long et al., on the other hand, focused on the task of thick cloud removal from multi-temporal remote sensing images and proposed the Bishift Networks architecture, which provides new insights for the development of cloud-related processing technologies but is not designed for ground-based cloud image classification scenarios. Fan et al. systematically reviewed the latest advances in ground-based sky image methods in this field, pointing out that fast and accurate cloud classification is a key bottleneck for improving prediction accuracy.

In 2024, Shi et al. proposed an improved CloudRVE ground-based cloud classification method. It adopts small-sized convolutional kernels, adds residual branches to enhance the extraction capability of local detailed features, and improves model inference speed through structural reparameterization technology. However, this model suffers from a low recall rate and an excessively large parameter count of 105.17 M. In the same year, Guzel et al. performed cloud image classification using transfer learning based on the pre-trained Xception model, which reduced model training time and computational resources but failed to provide specific resource consumption metrics.

In summary, current ground-based remote sensing cloud classification algorithms mainly face three key challenges: (1) insufficient fine-grained feature extraction leading to low recall rates for certain cloud genera; (2) excessive model parameters hindering deployment on mobile meteorological observation devices; (3) experimental datasets that do not comply with the World Meteorological Organization (WMO) cloud classification standards. To address these issues, this study proposes an algorithm with excellent classification performance and a small parameter count, leveraging attention mechanisms for fine-grained feature enhancement and lightweight compression optimization strategies for network structures. This work aims to promote the intelligent development of ground-based remote sensing cloud observation methods and devices.

2 Methodology

2.1 ECA-DP: Dual-Pooling Channel Attention for Edge Enhancement in Cloud Imagery with Sky Background Interference

Global Max Pooling (GMP) focuses on the most strongly activated regions in feature maps, retaining maximum values to highlight distinct boundary features in cloud images and preventing edge information from being averaged out. When the brightness of the sky background in cloud images changes, GMP helps suppress irrelevant background information, thereby improving the accuracy of cloud classification.

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

To capture global cloud image information while emphasizing salient local features of different cloud types, we propose an Efficient Channel Attention with Dual Pooling (ECA-DP) mechanism that combines Global Average Pooling (GAP) and Global Max Pooling (GMP), with its structure illustrated in Fig. 1. Attention mechanisms enable neural networks to distinguish key points and dynamically adjust attention distribution. The dual-pooling strategy extracts cloud image features in parallel, enhancing the complementarity between global and local information.

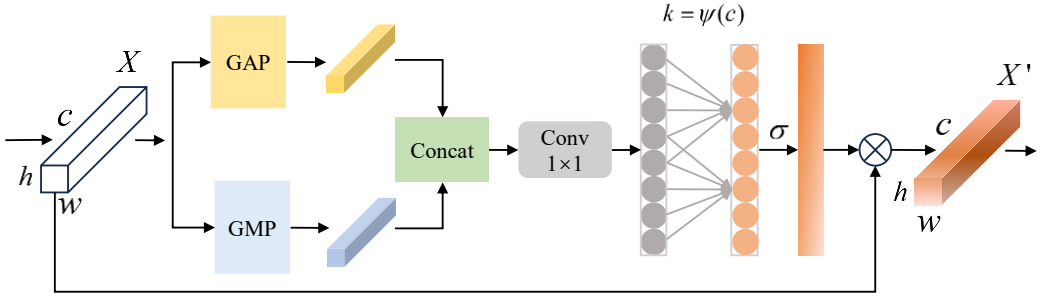

Fig. 1 ECA-DP structure

In Fig. 1, X denotes the input cloud feature map. First, global average pooling (GAP) and global max pooling (GMP) are performed in parallel on the input tensor X along the channel dimension. The resulting features are then concatenated, and a 1×1 convolutional kernel is used to adjust the number of channels, yielding a global feature vector of the cloud image along the channel dimension with a size of 1×1×c (where c represents the number of channels). Next, a 1D convolution (Conv) is applied to capture correlations between local channels, with the size of this 1D convolutional kernel adaptively adjustable. Finally, an activation function k is used to calculate channel-wise weights for the cloud feature map, and these attention weights σ are multiplied by the original input feature map to generate the final output cloud feature map.

The calculations for Global Average Pooling (GAP), Global Max Pooling (GMP), and channel attention weights are as follows:

$$\gamma_{\text{gap}} = \frac{1}{wh}\sum_{i=1,j=1}^{w,h} X_{ij},\ X \in R^{w\times h\times c}, \tag{1}$$

$$\gamma_{\text{gmp}} = \max\left(\sum_{i=1,j=1}^{w,h} X_{ij}\right),\ X \in R^{w\times h\times c}, \tag{2}$$

$$\eta = \sigma\left(V_k \gamma\right),\ V_k \in R^{c\times c}, \tag{3}$$

where γ_{gap} denotes the result of global average pooling, and γ_{gmp} denotes the result of global max pooling; w, h, and c represent the width, height, and number of channels of the input feature map X, respectively; i and j denote the spatial positions on channel c; γ represents the feature after channel adjustment; V_k is the 1D convolutional kernel; σ is the sigmoid function; and η represents the transformed attention weight.

In the ECA-DP structure, a 1D convolutional kernel is used to replace traditional fully connected layers, which avoids parameter redundancy and the loss of feature channel information. Compared with standard convolution, it has the advantages

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

of lower computational and parameter overhead. There exists a non-linear mapping relationship ϕ between the size k of the 1D convolutional kernel and the channel dimension c, expressed as Eq. (4).

$$c = \phi(k) = 2^{(r*k-b)}, \tag{4}$$

where b and r are set to 1 and 2, respectively.

To handle the complex textural features in ground-based remote sensing cloud images more flexibly, given the number of channels c, the size k of the 1D convolutional kernel can be adaptively adjusted via a function, as expressed in Eq. (5). Here, $|t|_{odd}$ odd denotes the odd integer closest to t.

$$k = \psi(c) = \left| \frac{\log_2(c)}{r} + \frac{b}{r} \right|_{odd}, \tag{5}$$

By averaging all values in the feature map, Global Average Pooling (GAP) generates features with global characteristics, which helps to capture the overall structural information of clouds.

2.2 Scale-Aware Self-Attention: Long-Range Feature Correlation for Cumulonimbus Anvils and Cirrus Fibrils

The fibrous structure of cirrus clouds usually exhibits spatially discontinuous distribution, and the spacing between fibrils often exceeds the local receptive field of traditional CNNs. The locality limitation of CNNs leads to severely insufficient capability for extracting features related to cloud system interactions across the entire image.Cumulonimbus clouds contain important cross-scale features such as convective towers and anvils. Convective towers occupy a small spatial extent in pixels, while anvil tops span distant regions in the image, resulting in a huge scale difference between them. CNNs tend to ignore the cirrus-like tendril structures at the edges of anvil clouds, causing scale-sensitivity imbalance and further reducing the discrimination accuracy of cumulonimbus.The gaps between stratocumulus clusters are typically around 10 pixels, which represent critical classification features. However, conventional single-scale CNNs easily cause irreversible loss of such microstructural information.

The inductive bias of CNNs fails to effectively establish cross-scale nonlinear relationships. Therefore, it is necessary to introduce a global attention mechanism and construct a hybrid architecture with CNNs to fully extract effective local and global features of cloud types. The core idea of Transformer is self-attention. Self-attention can compute the correlation degree between feature vectors and assign different attention weights to spatial information at different levels according to the computed scores. It can effectively establish long-range global dependencies among feature vectors, thereby improving the global feature extraction capability of the network. Transformers excel at capturing global information in complex scenes, but their network structure usually involves a large number of parameters and high computational cost, and requires massive data for training to achieve satisfactory performance.For cloud types with significant multi-scale characteristics such as cumulus, self-attention can simultaneously analyze and extract small-scale thermal bubble structures and large-scale vertically

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

developed features through the multi-head parallel processing mechanism, thus improving recognition accuracy. It reduces the quadratic complexity of traditional self-attention to linear complexity, decreasing network parameters and computational overhead.

Therefore, to capture global features of cloud images and reduce the number of parameters, self-attention is integrated into the network of the classification algorithm. The image processing pipeline of self-attention is shown in Fig. 2.The input in Fig. 2(a) is the cloud feature map obtained after preprocessing with convolutional kernels. First, the feature map is divided into multiple non-overlapping patches of 2×2 window size with a stride of 2; the operation of this step is illustrated in Fig. 2(b). Then, the width and height dimensions of each patch are merged to obtain the encoded vector. Subsequently, the encoded vectors are fed into the self-attention module for global correlation calculation. Finally, a feature map with enhanced attention to key texture regions is output.

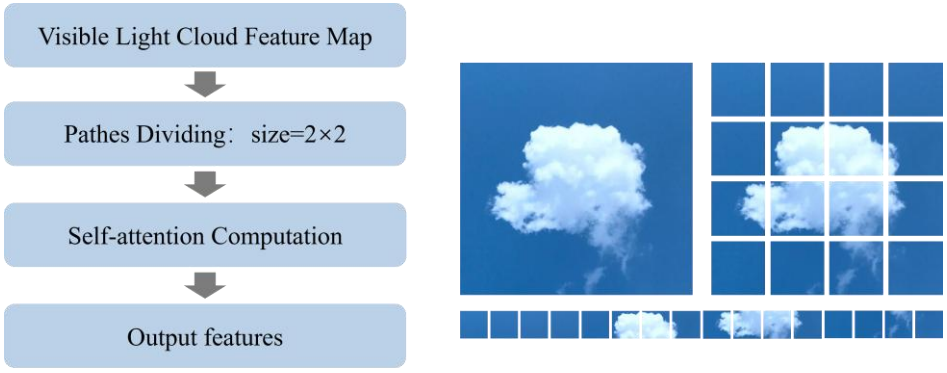

Fig.2 Self-Attention processing flow (a) Self-attention processing(b) Feature map division

The self-attention calculation is given in Eq. (6), where Q, K, and V are trainable weight matrices constructed from the encoded vectors.Q is the query vector, used to search for relevant information;K is the key vector, matched with the query to compute the correlation;V is the value vector, storing the actual information content;and d_k denotes the dimension of the hidden layer.

$$Attention(Q,K,V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V\,, \tag{6}$$

Figure 3 illustrates the computational process of self-attention for capturing global information from cloud images. $x_i, i \in (1,2,\ldots,n)$ denotes the bottom-layer input cloud feature map, and W^Q, W^K, W^V are the parameter matrices.Taking vector x_1 as an example, the trained parameter matrices W^Q, W^K and W^V with $x_i, i \in (1,2,\ldots,n)$ are first multiplied with $x_i, i \in (1,2,\ldots,n)$ respectively to obtain their corresponding q_i, k_i and v_i, $i \in (1,2,\ldots,n)$ .Next, the correlation between each current node and other nodes is calculated by performing dot-product operations between $q_i, i \in (1,2,\ldots,n)$ and $k_j, j \in (1,2,\ldots,n)$ of other vectors.Specifically, q_1 is dotted with $k_j, j \in (1,2,\ldots,n)$ to obtain the correlation coefficients

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

$a_{1i}, i \in (1,2,\ldots,n)$.A larger dot-product result indicates a stronger correlation between the two features.Subsequently, the obtained correlation coefficients are fed into the *Softmax* function for normalization, yielding the weight distribution parameters $b_{1i}, i \in (1,2,\ldots,n)$.Finally, these weights are multiplied by the corresponding $v_j, j \in (1,2,\ldots,n)$ and then summed to obtain the new feature vector representation Z_1.The calculation for other vectors follows the same procedure.Through the self-attention scores, a new feature representation $Z = (Z_1, Z_2, Z_3, \ldots Z_i), i \in (1,2,\ldots n)$ that incorporates the attention weights for global cloud features can be obtained.

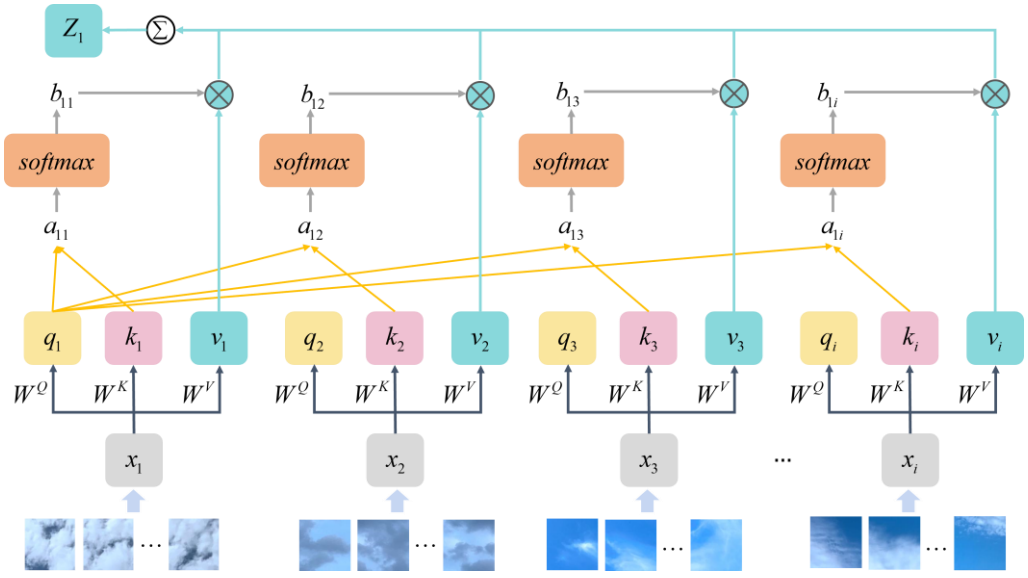

Fig. 3 Self-attention structure for capturing global information from cloud images

2.3 DW-PW Collaborative Feature Decoupling: Cross-Channel Enhancement for Cirrus Edges and Stratocumulus Shadows

Depthwise separable convolution decomposes the overall computation into two separate steps: depthwise convolution (DW) and pointwise convolution (PW), as illustrated in Fig. 4.For the input cloud feature map, DW performs convolution independently on each channel, focusing on local and detailed variations of cloud features within each channel, and extracts multi-scale features from cloud images using convolutional kernels of different sizes.For some cloud types with highly similar visual morphology and structure, such as cirrus and cirrocumulus, DW can extract subtle edge differences from their similar textures, thereby improving the accuracy of the cloud classification algorithm.

DW can extract local features channel by channel and perform simple superposition, but cannot correlate features across different channels.In contrast, PW can fuse cloud features along the channel dimension. It employs 1×1 convolutional kernels to perform convolutions channel-wise, and captures correlations of cloud features between different channels in the image without changing the spatial resolution.For example, stratus and stratocumulus usually exhibit a uniform layered structure with

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

small variations in color and texture and lack distinct edges.However, stratocumulus is thicker than stratus in structure and shows more obvious shadow variations in certain specific channels, which can be effectively captured by PW.Therefore, PW is beneficial for processing layered cloud types, enhancing the discriminability to reduce misclassification.

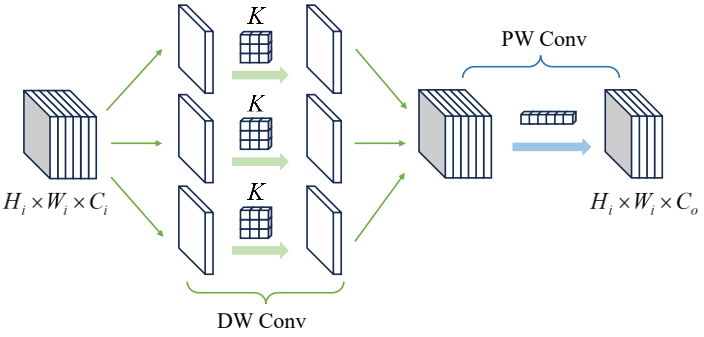

Fig.4 Depthwise separable convolution for processing cloud image features

By fully extracting the local and channel features of cloud images through DW and PW, richer feature representations are output, thereby improving the classification accuracy of the algorithm.

2.4 DSC-CloudGhost Cascade Compression: Co-Optimization of FLOPs and Parameters

Compared with standard conventional convolution, the greatest advantage of depthwise separable convolution lies in its ability to bring significant improvements in the computational efficiency of the algorithm model.

If the size of the input cloud feature map is $H_i \times W_i \times C_i$, the size of the output feature map is $H_i \times W_i \times C_o$, the size of the depthwise convolution kernel is $K \times K$, and the stride is 1, the computational complexity C_{st} and parameter count P_{st} of the standard convolution are respectively:

$$C_{st} = H_i \times W_i \times C_i \times K \times K \times C_o, \tag{7}$$

$$P_{st} = K \times K \times C_i \times C_o, \tag{8}$$

The computational complexity C_{dp} and parameter count P_{dp} of the depthwise separable convolution are respectively:

$$C_{dp} = H_i \times W_i \times C_i \times \left(K^2 + C_o\right), \tag{9}$$

$$P_{dp} = K \times K \times C_i + 1 \times 1 \times C_i \times C_o, \tag{10}$$

The ratios of computational cost F_C and parameter count F_P between depthwise separable convolution and standard convolution are given by:

$$F_C = \frac{C_{dp}}{C_{st}} = \frac{1}{C_o} + \frac{1}{K^2}, \tag{11}$$

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

$$F_P = \frac{P_{dp}}{P_{st}} = \frac{1}{C_o} + \frac{1}{K^2}, \tag{12}$$

Therefore, in the design of the cloud image classification model, when the number of output channels C_o is relatively large, applying depthwise separable convolution can reduce both the computational cost and the number of parameters to a fraction $1/K^2$ of the original ones. Replacing standard convolution with depthwise separable convolution achieves network lightweighting.

Although pointwise convolution can flexibly adjust the number of output channels of cloud feature maps and control the depth and width of the network, it still introduces considerable computational cost and parameters under high-dimensional inputs, increasing the memory requirement and computational burden of the network. To further reduce network parameters and computations, we propose to replace traditional pointwise convolution with CloudGhost to achieve feature dimensionality reduction and compression, whose structure is shown in Fig. 5. A small number of pointwise convolutions are used to generate part of the ground-based remote sensing cloud image features; then, similar feature maps of the same size are generated via linear operations. Finally, the two parts of features are concatenated and fused to obtain the output cloud image features.

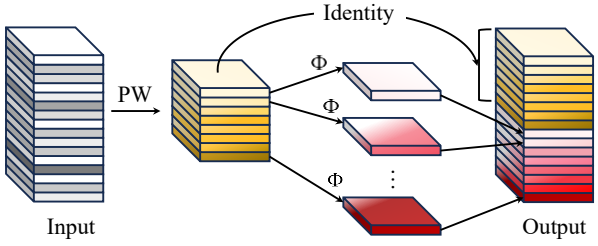

Fig. 5 Efficient feature dimensionality reduction module

When the proportion of original features in the output cloud features is 1/2, the computation of traditional PW convolution is:

$$P_{pw} = C_i \times C_o, \tag{13}$$

The computational complexity P_{cg} induced by the CloudGhost module is:

$$P_{cg} = C_i \times \frac{C_o}{2} + K \times K \times \frac{C_o}{2}, \tag{14}$$

The parameter compression ratio R can be obtained as:

$$R = \frac{P_{pw}}{P_{cg}} = \frac{2C_i}{C_i + K^2} \approx 2, \tag{15}$$

Where, given that the dimensions of the cloud feature maps remain unchanged, the linear computation employs depthwise convolution with kernel size K, input channels C_i, and output channels C_o, where, assuming the size of the cloud feature map

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

remains unchanged, linear computation is performed using a depthwise convolution of size K, with an input channel C_i and an output channel C_o. In this case, $C_i \gg K^2$.

As derived above, the proposed CloudGhost module achieves an *R*-fold reduction in parameters compared to conventional pointwise convolutions, significantly compressing the parameter count for cloud feature extraction. Crucially, it maintains the complete spatial structure of cloud features while enabling channel-dimension adjustment of feature maps. Moreover, by reusing existing features, it effectively increases the network's capacity without additional parameters. This architecture proves particularly advantageous for capturing fine-grained features in ground-based remote sensing cloud imagery, thereby enhancing classification performance.

2.5 CloudMViT Edge Computing Framework: Lightweight Cloud Classification Implementation with Integrated Multi-Feature Enhancement Modules

A lightweight visual network model named CloudMViT was designed for ground-based remote sensing cloud image classification. Its specific structure is illustrated in Fig. 6, which mainly consists of three steps.

Step 1: Extraction of local detailed features from cloud imagesFirst, pointwise convolution is adopted to extract shallow cloud image features. Without losing resolution, it can not only capture fine boundary features of remote sensing cloud images but also expand the number of channels of the feature map, enabling cross-channel information interaction with fewer weight parameters. Then, depthwise convolution with a kernel size of 3×3 is used to extract global features, followed by channel compression via linear CloudGhost. Finally, the feature map processed by the ECA-DP module on the shortcut branch is fused with the main branch to obtain the final feature map output. To avoid overfitting of the model, a Batch Normalization (BN) layer is added after each convolution operation. A nonlinear activation function is incorporated to increase the nonlinearity of the network and enhance its ability to express complex features.

Step 2: Extraction of global features from cloud imagesFirst, a convolutional layer with a kernel size of 1×1 is used to expand the channels of the input feature map. Then, Feature_Unfold is utilized to flatten the feature vectors, which are fed into the Self-Attention module to calculate attention scores and obtain globally relevant weight values. The Feature_Unfold module is responsible for restoring the feature maps with different attention allocations. Its local features and global features are concatenated (Concat) along the channel dimension. After concatenation, the features are fed into the lightweight convolutional CloudGhost module to adjust the channel dimension. Finally, the output is obtained by fusing the shortcut branch with the original input features.

Step 3: Enhancement of local detailed features from cloud imagesThe same steps as Step 1 are repeated to improve the classification performance of the network.

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

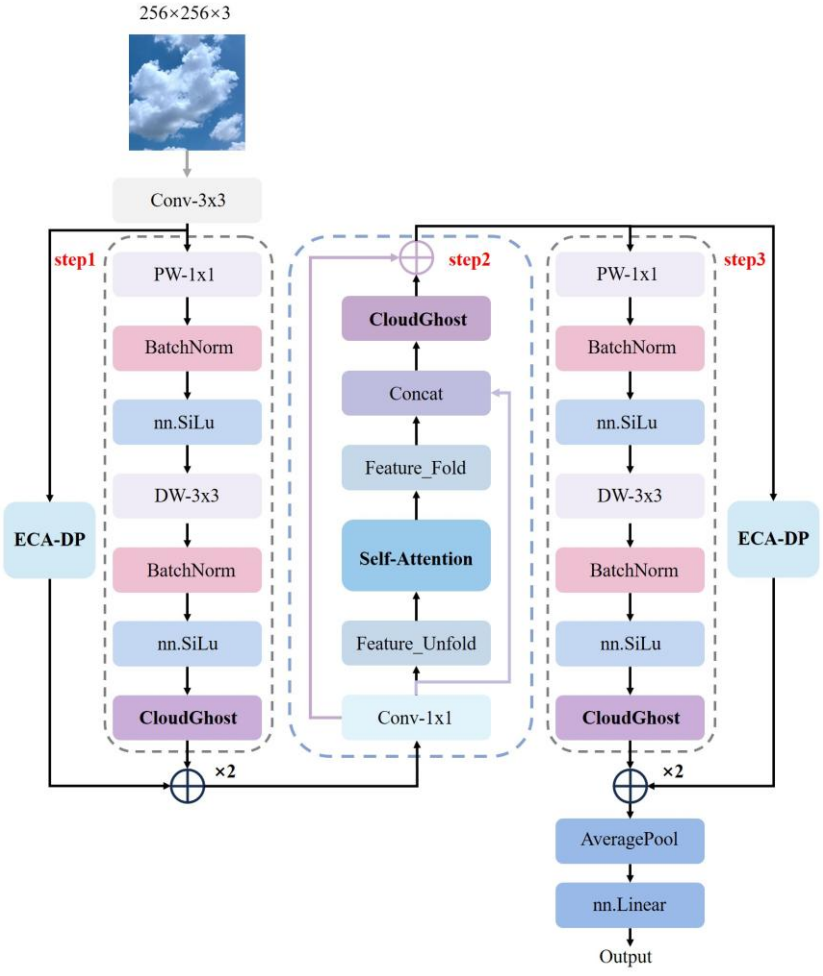

Fig. 6 Architecture of the CloudMViT network

The CloudMViT algorithm model was deployed on the embedded platform RK3588. This process mainly includes three steps: model quantization, model conversion, and NPU deployment, as illustrated in Fig. 7.

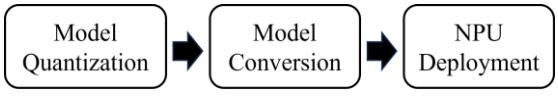

Fig. 7 Model Deployment Process of CloudMViT on RK3588

Model quantization converts weights and activation values in the model from high-bit floating-point data to low-bit fixed-point data, thereby reducing the model's memory footprint and computational intensity. To fully utilize the computing power of the NPU and achieve collaborative optimization of algorithm performance and hardware efficiency, the INT8 quantization technique was adopted to optimize the CloudMViT algorithm model.

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

Model conversion transforms the model format generated by the PyTorch framework into the RKNN format, facilitating high-performance inference computation on the RKNPU platform. Rockchip provides RKNN-Toolkit2, a development kit for model conversion, which can directly convert models in other formats into RKNN models.

NPU deployment involves running the model for inference computation on RK3588. The RKNPU driver is used to invoke the NPU to accelerate the inference computation of algorithms in the model and improve data processing efficiency. RKNPU is an on-board development component that provides cross-platform programming interfaces based on C/C++, enabling the practical deployment of RKNN models.

The rknn.eval_memory API can be used on the RK3588 platform to obtain the actual memory usage of the model during operation. This API enables a comprehensive evaluation of the memory occupied by weight parameters, intermediate tensors, and the entire model in the model. Table 1 lists the specific memory resource usage of the cloud image classification model running on the NPU before and after quantization.

Without quantization, the weight parameters occupy 0.412 MB of memory; after INT8 quantization, the memory usage is halved to only 0.234 MB. The overall memory of the model is reduced from 12.95 MB to 7.18 MB, a reduction of approximately 44%. The remote sensing cloud image classification model has low memory usage on RK3588, and quantization can effectively reduce the consumption of storage resources.

Table 1 Memory Occupancy Evaluation Results

Quantization Method	Weight Memory (MB)	Intermediate Tensor Memory (MB)	Total Model Memory Occupancy (MB)
No quantization	0.412	5.37	12.95
INT8 quantization	0.234	2.88	7.18

The inference time on CPU and NPU for each layer, total computation time, and time consumption ratio of the model running on RK3588 are listed in Table 2. The data in the table represent the time consumption of each layer of the model without quantization. The NPU undertakes most of the inference computation, while the CPU is only responsible for part of the lightweight computations, including operators such as Gather, Transpose, and ReduceSum. Among all operator layers, exSwish consumes the longest time, accounting for 25.28%, followed by the normalization layer exLayerNorm, the reshape layer, and the convolutional layer Conv, accounting for 21.71%, 20.64%, and 11.29%, respectively. The computation time of other layers is relatively short, and almost all of them are executed on the NPU, thus the NPU greatly improves the computational efficiency.

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

Table 2 Time Consumption Results for Each Model Layer

Operator Type	CPU Time Cost (us)	NPU Time Cost (us)	Total Time Cost (us)	Time Consumption Ratio (%)
exSwish	0	17714	17714	25.28
exLayerNorm	0	15217	15217	21.71
Reshape	0	14461	14461	20.64
Gather	4251	0	4251	6.07
Conv	0	7909	7909	11.29
Transpose	2636	0	2636	3.76
ReduceSum	1519	0	1519	2.17
Concat	0	1306	1306	1.86
Mul	0	1053	1053	1.5
BatchNormalization	0	862	862	1.23
ConvTranspose	0	682	682	0.97
ConvAdd	0	669	669	0.95
Split	0	544	544	0.78
Add	0	536	536	0.76
Relu	0	301	301	0.43
Expand	0	268	268	0.38
exSoftmax	136	0	136	0.19
InputOperator	9	0	9	0.01
OutputOperator	4	0	4	0

Table 3 compares the total computation time and frame rate of the ground-based remote sensing cloud image classification model before and after quantization. After INT8 quantization of the model, the time consumed to process a single cloud image is 0.031 s, and the FPS reaches 21.58. Compared with the unquantized model, the inference speed and frame rate are both greatly improved.

Table 3 Inference Time and Frame Rate of the Model

Quantization Method	Model Inference Time (s)	Frame Rate (FPS)
No quantization	0.071	12.36
INT8 quantization	0.031	21.58

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

3 Datasets and Experimental Setup

3.1 Datasets and Preprocessing

Two large-scale, high-resolution ground-based remote sensing cloud image datasets with the most complete cloud categories were adopted in the experiments, namely the HBMCD and GCD datasets.

HBMCD was collected by an all-sky imager at the Huayun Shengda meteorological station of the China Meteorological Administration. According to the cloud classification standard of the World Meteorological Organization (WMO), HBMCD contains 10 standard cloud genera and 1 cloud-free class, with a total of 25,118 images. The resolution is 1358×1358 pixels, and the data format is JPEG. It is currently the largest dataset that conforms to the WMO classification standard.

GCD is a ground-based remote sensing cloud dataset that includes 7 sky conditions, with a total of 12,000 images at a resolution of 512×512 pixels. The dataset was collected from 9 provinces in China from 2019 to 2020, including Tianjin, Anhui, Sichuan, Gansu, Shandong, Hebei, Liaoning, Jiangsu, and Hainan. All cloud images were annotated collaboratively by meteorologists and researchers specializing in ground-based cloud observation.

Some sample cloud images of the datasets are shown in Fig. 8. To improve the generalization ability of the algorithm, the same preprocessing strategy was applied to both datasets. Four data augmentation techniques, including image rotation, contrast enhancement, Gaussian noise addition, and gamma correction, were used to increase data diversity. The cloud images were further preprocessed by image resizing, data conversion, and normalization to reduce the impact of inconsistent data distribution on model performance. The training and test sets were split at a ratio of 4:1, with the detailed division listed in Table 4 and Table 5.

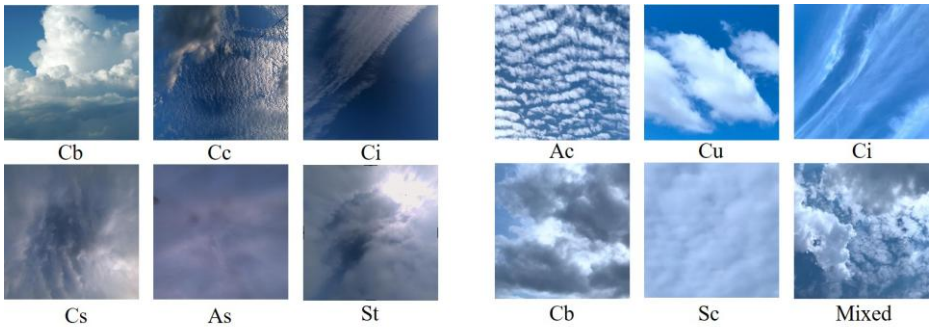

Fig.8 Different types of cloud images

Table 4 HBMCD dataset and experimental split

Cloud Category	Abbreviation	Total Samples	Train Set	Test Set

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

Altocumulus	Ac	11280	9024	2256
Altostratus	As	12920	10336	2584
Cumulonimbus	Cb	6960	5568	1392
Cirrocumulus	Cc	6510	5208	1302
Cirrus	Ci	15375	12300	3075
Cirrostratus	Cs	13190	10552	2638
Cumulus	Cu	15500	12400	3100
No Cloud	No	19020	15216	3804
Nimbostratus	Ns	6180	4944	1236
Stratocumulus	Sc	10740	8592	2148
Stratus	St	7915	6332	1583
Total		125590	100472	25118

Table 5 GCD dataset and experimental split

Cloud Category	Abbreviation	Total Samples	Train Set	Test Set
Cumulus	Cu	7750	6200	1550
Altocumulus	Ac	7250	5800	1450
Cirrus	Ci	4420	3536	884
Clear sky	Cl	12570	10056	2514
Stratocumulus	Sc	6480	5184	1296
Cumulonimbus	Cb	18050	14440	3610
Mixed	Mi	3480	2784	696
Total		60000	48000	12000

3.2 Experimental Setup

The model training and testing were conducted in a Python 3.8 compilation environment, with an NVIDIA GeForce RTX 3080 GPU. PyTorch 1.10.1 was adopted as the deep learning framework. The initial learning rate was set to 0.001, the total number of iterations was 100, and the batch size was set to 256. The Adam optimizer was used to optimize all trainable parameters in the network. The cross entropy loss (CEL) was employed, with its calculation formula given in Eq. (16). To accelerate the training speed, a step-wise learning rate decay strategy was adopted, where the learning rate was reduced to half of its original value every 35 epochs.

$$Loss = -\frac{1}{N}\sum_{i=1}^{N}[y_i \ln(p_i) + (1-y_i)\ln(1-p_i)], \tag{16}$$

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

Where N is the number of classes, i denotes the dimension of the prediction vector, y_i is the ground-truth label of the sample, and p_i is the predicted probability of the sample label, ranging from 0 to 1.

3.3 Evaluation Metrics

To objectively reflect the classification performance of the CloudMViT model on the ground-based remote sensing cloud image datasets, the classification performance of the model for each cloud type is evaluated using three metrics: Precision, Recall, and F1-score. The formulas for these metrics are given as follows:

$$Precision = \frac{TP}{TP + FP}, \tag{17}$$

$$Recall = \frac{TP}{TP + FN}, \tag{18}$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{19}$$

The overall classification performance of the model is evaluated using Accuracy, macro-precision P_{macro}, macro-recall R_{macro}, and macro-F1 F_{1macro}. The formulas are given as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{20}$$

$$P_{macro} = \frac{1}{n}\sum_{i=1}^{n} P(i), \tag{21}$$

$$R_{macro} = \frac{1}{n}\sum_{i=1}^{n} R(i), \tag{22}$$

$$F1_{macro} = \frac{1}{n}\sum_{i=1}^{n} F_1(i), \tag{23}$$

Where n denotes the number of classes, TP (True Positive) denotes the number of true positives, FP (False Positive) indicates the count of negative samples incorrectly predicted as positive, FN (False Negative) corresponds to the number of false negatives, and TN (True Negative) signifies the number of true negatives.

4 Experimental Results and Analysis

4.1 Confusion Matrices

To analyze the experimental results more clearly, confusion matrices were generated based on the final test results, as illustrated in Fig. 9 and Fig. 10. The horizontal axis represents the cloud class labels predicted by the network, and the vertical axis represents the ground-truth cloud labels. The diagonal values indicate the number of correctly classified samples for each

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

category, while the off-diagonal elements represent the number of misclassified samples. A darker color corresponds to a larger number of correctly predicted samples.

It can be observed from Fig. 9 that the model achieves satisfactory overall performance on the HBMCD dataset. The cloud-free category (No) obtains the best classification accuracy, whereas cumulonimbus (Cb) and cirrocumulus (Cc) exhibit relatively high error rates. The main misclassifications are that cumulus (Cu) is easily misclassified as cumulonimbus (Cb), and altocumulus (Ac) is often misclassified as cirrocumulus (Cc). The reason is that cumulus and cumulonimbus show similar visual structures at low altitudes. As weather conditions change, cumulus may develop rapidly into cumulonimbus, and the ambiguous cloud features during this transition easily lead to confusion between the two types. In addition, both altocumulus and cirrocumulus have layered structures and are usually distributed in waves or patches. Variations in illumination and different observation angles can make them visually similar, which increases the difficulty of classification.

It can be seen from Fig. 10 that the model performs well in classifying cumulonimbus (Cb), followed by clear sky (Cl). The model performs relatively poorly on cirrus (Ci) and mixed cloud (Mi). Most cirrus (Ci) samples are misclassified as altocumulus (Ac) and mixed cloud (Mi). Misclassified samples of mixed cloud (Mi) are mainly distributed in cumulus (Cu), altocumulus (Ac), and cumulonimbus (Cb). During the construction of the GCD dataset, images containing two or more cloud types are labeled as mixed cloud. Therefore, the features of mixed cloud images are generally more complex and ambiguous than those of single-type cloud images, making it difficult for the model to distinguish them accurately from other cloud categories, thus resulting in misclassification.

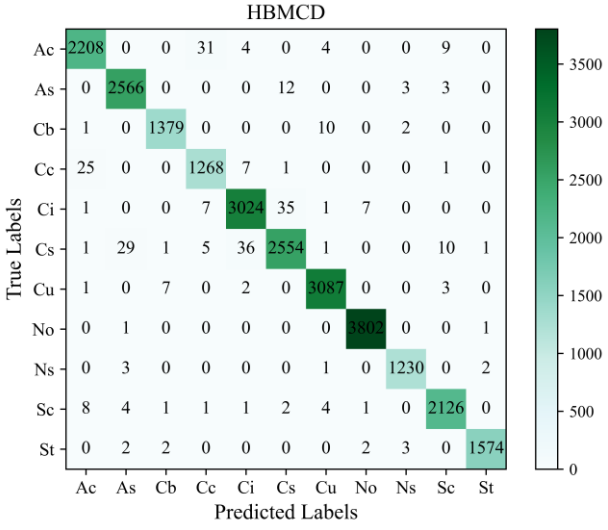

Fig.9 Confusion matrix of CloudMViT on the HBMCD dataset

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

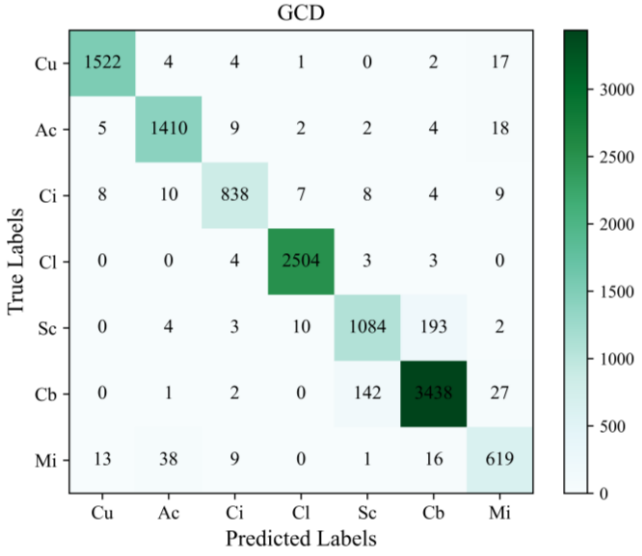

Fig.10 Confusion matrix of CloudMViT on the GCD dataset

Tables 6 and 7 present the detailed evaluation metric results of the CloudMViT model on each category in the HBMCD and GCD datasets, respectively. It can be observed that CloudMViT achieves values of precision, recall, and F1-score above 96% for most cloud types in the HBMCD dataset. In particular, the recognition accuracies of cloud-free (No), nimbostratus (Ns), and stratus (St) are close to 100%. For the GCD dataset, all metrics for clear sky (Cl) reach up to 99%, and the classification performance for most other cloud genera also exceeds 94%.

Table 6 Evaluation metrics on the HBMCD dataset

Abbreviation	*Precision* (%)	*Recall* (%)	F_1 (%)
Ac	98.35	97.87	98.11
As	98.5	99.3	98.9
Cb	99.21	99.07	99.14
Cc	96.65	97.39	97.02
Ci	98.37	98.34	98.36
Cs	98.08	96.82	97.44
Cu	99.32	99.58	99.45
No	99.74	99.95	99.84
Ns	99.35	99.51	99.43
Sc	98.79	98.98	98.88
St	99.75	99.43	99.59

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

Table 7 Evaluation metrics on the GCD dataset

Abbreviation	*Precision* (%)	*Recall* (%)	F_1 (%)
Cu	98.32	98.19	98.26
Ac	96.11	97.24	96.67
Ci	96.43	94.8	95.61
Cl	99.21	99.6	99.4
Sc	87.42	83.64	85.49
Cb	93.93	95.24	94.58
Mi	89.45	88.94	89.19

Table 8 summarizes the overall performance evaluation metrics of the model. The experimental data demonstrate that CloudMViT achieves an accuracy of 98.81% on the HBMCD dataset, with macro-average precision, macro-recall, and macro-F1 all exceeding 98.7%, indicating that CloudMViT exhibits excellent classification and generalization performance. Under the same experimental settings, CloudMViT still achieves satisfactory classification performance on the GCD dataset, with an overall accuracy of 95.13% for the 7 cloud classes. The macro-average precision, macro-recall, and macro-F1 are 94.41%, 93.95%, and 94.17%, respectively.

Table 8 Overall evaluation metrics

Dataset	P_{macro} (%)	R_{macro} (%)	$F1_{macro}$ (%)	*Accuracy* (%)
HBMCD	98.74	98.75	98.74	98.81
GCD	94.41	93.95	94.17	95.13

In summary, the proposed CloudMViT algorithm demonstrates outstanding recognition capability and satisfactory generalization performance on the HBMCD dataset. When the algorithm is transferred to the GCD dataset for training and validation, its classification accuracy exceeds 95%, which effectively proves that CloudMViT possesses strong feature adaptability and robustness. It can maintain stable discriminative performance under diverse meteorological conditions and adapt to various application scenarios.

4.2 Ablation Study and Visualization Analysis

(1) Performance Validation of Sub-modules

To investigate the classification performance of the proposed algorithm in its original structure and different improvement stages, ablation experiments were conducted. The detailed evaluation metrics are presented in Table 9. MobileViT was adopted as the baseline network, and its improved version was denoted as MobileViT-S. CloudGhost is a linear compression module,

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

ECA is a traditional attention module, and ECA-DP is an improved attention module with a dual-pooling strategy. CloudMViT is the final network obtained by combining CloudGhost and ECA-DP. For the convenience of expression, the algorithm models MobileViT, MobileViT-S, MobileViT-S+ECA, MobileViT-S+ECA-DP, MobileViT-S+CloudGhost, and CloudMViT are sequentially labeled as Method ① to Method ⑥, respectively.

It can be observed from Table 9 that the performance of the network model at each improvement stage is improved compared with the previous stage. On the HBMCD dataset, by comparing Method ② and Method ④, the accuracy of MobileViT-S is increased from 89.83% to 95.14% (an increase of 5.31%) after adding the designed ECA-DP module. Meanwhile, the precision, recall, and F1-score are improved by 5.07%, 4.49%, and 5.86%, respectively. In contrast, when comparing Method ② and Method ③, adding the traditional ECA module to MobileViT-S only improves the accuracy by 2.44%. On the GCD dataset, comparing Method ② and Method ④, the accuracy is increased from 89.30% to 94.34% (an increase of 5.04%), with the precision, recall, and F1-score improved by 5.75%, 3.68%, and 3.56%, respectively. In comparison, the F1-score is only improved by 1.5% and the recall remains almost unchanged when comparing Method ② and Method ③. These results fully verify that the ECA-DP module can significantly enhance the noise suppression and channel feature extraction capabilities in cloud images. In addition, a comparison between Method ② and Method ⑤ reveals that the integrated CloudGhost module also contributes to the improvement of model performance, with all metrics increased by more than 2% on the HBMCD dataset and exhibiting a similar improvement effect on the GCD dataset.

Table 9 Impact of different modules on cloud image classification performance

Dataset	Method	*Accuracy* (%)	P_{macro} (%)	R_{macro} (%)	$F1_{macro}$ (%)
HBMCD	① MobileViT	90.52	90.39	89.54	89.68
	② MobileViT-S	89.83	89.89	89.83	89.73
	③ MobileViT-S+ECA	92.27	92.14	91.49	92.75
	④MobileViT-S+ECA-DP	95.14	94.96	94.32	95.59
	⑤MobileViT-S+CloudGhost	93.78	92.78	93.85	93.21
	⑥ CloudMViT	**98.81**	**98.74**	**98.75**	**98.74**
GCD	① MobileViT	90.09	88.69	89.51	89.4
	② MobileViT-S	89.3	87.67	89.24	89.02
	③ MobileViT-S+ECA	92.84	91.05	90.19	90.52
	④MobileViT-S+ECA-DP	94.34	93.42	92.92	92.58
	⑤MobileViT-S+CloudGhost	93.59	92.79	91.12	91.84
	⑥ CloudMViT	**95.13**	**94.41**	**93.95**	**94.17**

(2) Analysis of Sub-module Complexity and Inference Speed

The effects of ECA-DP and CloudGhost on the computational efficiency and forward inference of the network are comprehensively evaluated on the HBMCD dataset. Table 10 shows the number of parameters, floating-point operations, and inference time on the test set for different improved network structures.By comparing Method ② and Method ④, it can be

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

seen that after adding the ECA-DP module, the number of parameters of the network remains unchanged, the floating-point operations only increase by 0.13 M, and the inference time does not increase. This indicates that the ECA-DP module introduces no extra parameters or extra computational burden to the network.By comparing Method ② and Method ⑤, it can be found that after integrating CloudGhost, the number of parameters decreases from 0.98 M to 0.25 M, and the floating-point operations are reduced by 52.73 M. The inference time is reduced from 62.43 s to 35.18 s, and the inference speed is improved by nearly 50%. This demonstrates that the CloudGhost module can significantly reduce the parameters, computational cost, and inference time of the network.

Table 10 Complexity and Inference Time of Different Models

Evaluation Metric	Model ①	Model ②	Model ③	Model ④	Model ⑤	**Model ⑥**
Params/M	1.17	0.98	0.98	0.98	0.25	**0.14**
FLOPS/M	503.93	515.54	515.67	515.67	462.81	**318.01**
Inference Time/s	49.55	62.43	62.85	61.35	35.18	**30.14**

(3) Feature Visualization Analysis

To intuitively compare the impact of each algorithm module on the overall network performance, feature maps from the same intermediate layer of each algorithm model are extracted and visualized. The feature maps highlight the important regions of the predicted images, where brighter regions indicate higher importance. Fig. 11 and Fig. 12 show the feature maps of three different ground-based remote sensing cloud images from the HBMCD and GCD datasets, respectively. By comparing the feature extraction results, it can be observed that CloudMViT exhibits the best capability to extract fine-grained texture features of ground-based remote sensing clouds.

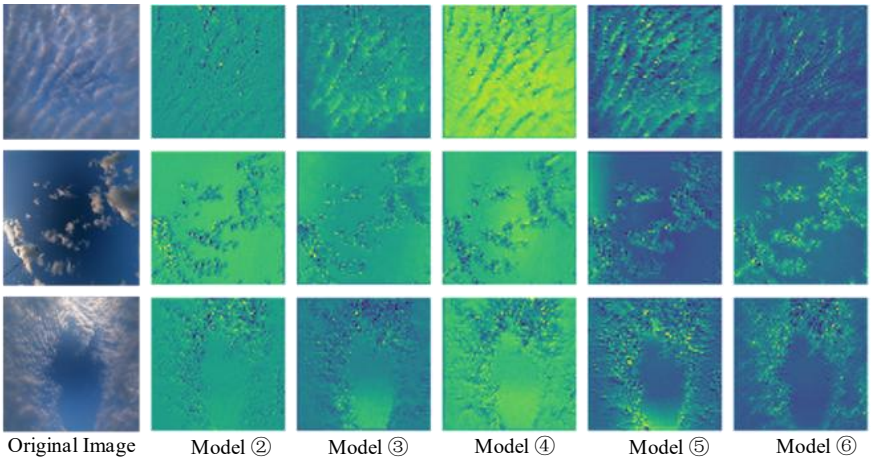

Fig.11 Visualization of intermediate layer feature maps in different networks on the HBMCD

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

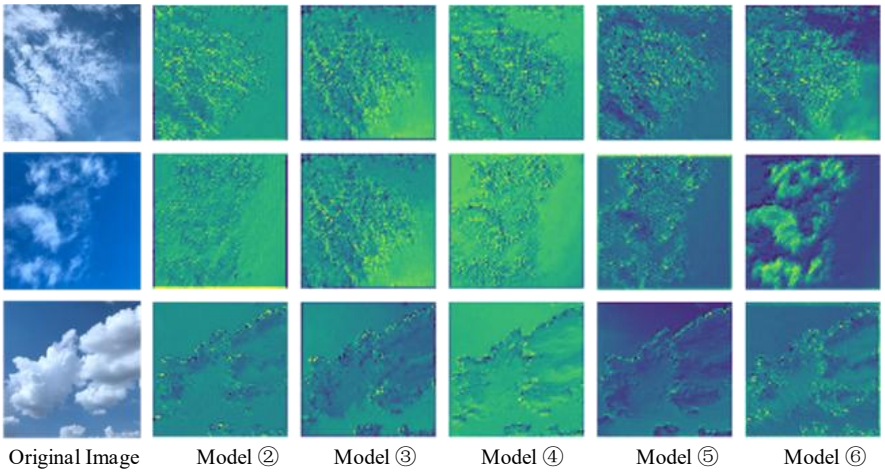

Fig.12 Visualization of intermediate layer feature maps in different networks on the GCD

4.3 Comparison with Different Lightweight Classification Methods

To verify the performance of the proposed algorithm, the designed CloudMViT model is compared with seven typical lightweight classification models: EfficientNet, MobileNet, ShuffleNe, MobileViT, ResNet18, SqueezeNet, and DenseNet121.

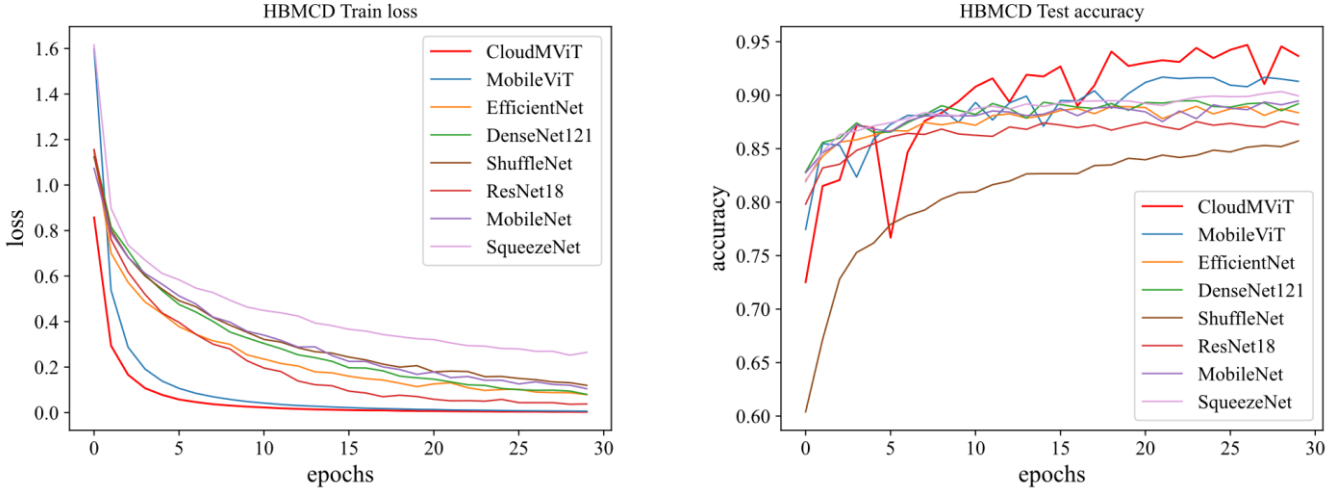

Fig.13 Loss and accuracy curves of different methods on the HBMCD dataset

Figures 13 and 14 show the curves of training loss and accuracy on the HBMCD and GCD datasets, respectively. Taking Fig. 13 as an example, during the training process, the loss value of CloudMViT decreases to approximately 0.1 at the 10th epoch, indicating that the model enters the convergence stage. The accuracy exceeds 90% at the 18th epoch and shows an upward trend with the increase of iterations. In contrast, the loss values of other methods decrease rapidly in the first 4 epochs but slow

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

down in the later stage.The accuracy of these models reaches a training bottleneck after reaching 88%, with no significant improvement thereafter. CloudMViT achieves the fastest loss convergence speed and the highest classification accuracy.

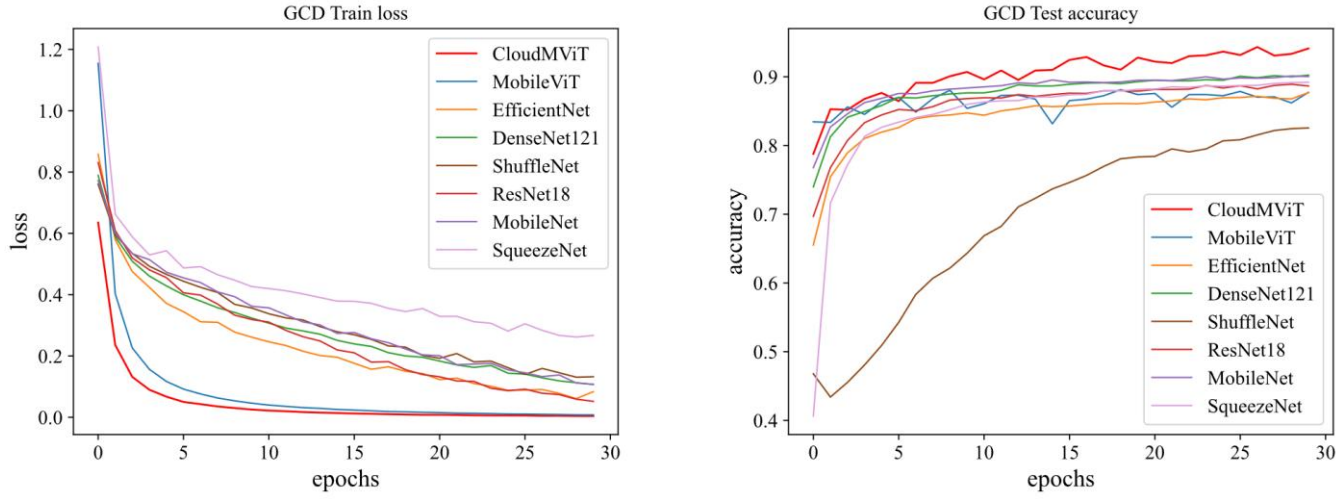

Fig.14 Loss and accuracy curves of different methods on the GCD dataset

Table 11 shows the comparison of parameters, FLOPs, memory usage, accuracy, and inference time among different lightweight classification algorithms.Among them, MobileNet, SqueezeNet, and ShuffleNet have FLOPs of no more than 0.5 G and short forward inference time, but their classification accuracy is relatively low.DenseNet121 and ResNet18 achieve high cloud image recognition accuracy, but their parameter counts are as high as 6.97 M and 11.18 M, with FLOPs of 3.78 G and 2.38 G, respectively.The overall memory requirements of the models reach 268.51 MB and 99.46 MB, which are very unfriendly for resource-constrained edge deployment scenarios.Compared with other models, MobileViT and SqueezeNet perform well in terms of parameters and computational cost.However, MobileViT has a long inference time and high memory consumption during inference, up to 267.31 MB.The reason is that MobileViT combines CNN and Transformer, whose computation requires storing a large number of tensor features, thus increasing memory usage.In addition, its architecture is more complex than traditional CNNs, leading to longer inference time.SqueezeNet has few network layers and a simple structure, so its FLOPs, parameters, and inference time are relatively low.However, it performs poorly in remote sensing cloud image classification, with a classification accuracy of only 88.84%.This is because the model focuses excessively on structural lightweight design, resulting in insufficient feature extraction capability and thus unsatisfactory classification performance.

Based on the above results, the proposed CloudMViT has only 0.14 M parameters, which is one-fifth of the parameters of SqueezeNet—the smallest baseline model.Its FLOPs are 0.32 G, and it achieves the highest classification accuracy of 98.81% compared with all other networks.The parameter memory usage of CloudMViT is 0.47 MB, which is the lowest among all classic models.Furthermore, CloudMViT achieves the shortest inference time on the overall test set, at only 30.14 s.

Table 11 Performance of different lightweight models on the HBMCD

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

Network	Parameters (M)	FLOPs (G)	Parameter Memory (MB)	Model Memory (MB)	Accuracy (%)	Inference Time (s)
EfficientNet	4.05	0.51	21.15	162.83	89.25	45.82
DenseNet121	6.97	3.78	31.92	268.51	90.83	76.92
ResNet18	11.18	2.38	46.76	99.46	90.94	38.26
MobileNet	2.23	0.39	14.02	154.37	89.73	34.34
ShuffleNet	4.35	0.19	9.11	50.66	87.14	34.02
MobileViT	1.17	1.01	19.78	267.31	91.27	49.55
SqueezeNet	0.74	0.46	4.94	32.99	88.84	32.97
CloudMViT	**0.14**	**0.32**	**0.47**	**103.33**	**98.81**	**30.14**

4.4 Edge-device Performance Validation

Connect the TYPE-C0 interface on the RK3588 to the USB port of a virtual machine on the computer. Open the Ubuntu system, and connect the Fuzhou Rockchip Rockusb Device in the removable devices of the virtual machine. Launch the terminal to enable the compiler, generate executable files and model files using the cross-compilation toolchain, and deploy the CloudMViT model to the RK3588 for system testing.

A total of 5028 images were randomly selected from the HBMCD dataset and tested sequentially on the RK3588. By collating the test results, the recall rate for each cloud category and the overall classification accuracy were obtained, as shown in Table 12. During inference on the RK3588, the model achieved a recall rate of over 90% for all cloud categories except cirrus clouds (which had a relatively low recall rate). Among them, the recognition rate for the cloud-free category was the highest, with a recall rate of 97.19%. The overall accuracy of the model was 94.79%, which was 4.02% lower than the performance on the PC side. INT8 quantization converts floating-point numbers in the model to 8-bit integers, a process that tends to lose some details and thus leads to a slight decrease in prediction accuracy.

Table 12 Performance Test Results of the Model on RK3588

Abbreviation	Recall (%)	Accuracy (%)
Ac	95.12	94.79
As	96.34	
Cb	95.91	
Cc	87.06	
Ci	93.45	
Cs	92.76	
Cu	96.21	
No	97.19	

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

Ns	95.81
Sc	93.02
St	95.98

5 Conclusions

The proposed CloudMViT algorithm enhances the fine-grained feature extraction and global attention for different remote sensing cloud types through improved channel attention and self-attention mechanisms. A lightweight deep separable convolution and a linear feature redundancy compression strategy are adopted to collaboratively optimize the network architecture, which significantly reduces the network parameters and computational complexity. Classification accuracies of 98.81% and 95.13% are achieved on the two high-quality datasets, respectively.

The designed CloudMViT model has only 0.14 M parameters, a computational cost of 0.32 GFLOPs, and a memory usage of 0.47 MB. Compared with the state-of-the-art lightweight classification schemes, it saves more than 80% of storage resources, and the inference time for a single remote sensing cloud image is only 0.006 s. The deployment of CloudMViT for ground-based remote sensing cloud recognition is implemented on the RK3588 hardware platform. With a certain degree of accuracy loss, the overall recognition accuracy of CloudMViT still reaches 94.79%.

The algorithm solves the problem that traditional methods cannot simultaneously capture the fine local structures and the global overall morphology of clouds. However, under the influence of other background clouds or noise, the subtle differences between altocumulus and cirrus clouds are difficult to distinguish, requiring more diverse and large-scale data samples for model training. In future work, the network can be further optimized by fusing multi-spectral data or combining multi-modal cloud information to improve the performance of ground-based remote sensing cloud classification models.

Author contributions

W.X. conceived the study, designed the methodology, performed the formal analysis, developed the software and figures, wrote and revised the manuscript, supervised the research, and administered the project.

NW. conducted the investigation and data curation, and contributed to the methodology.

LF. participated in the methodology design and assisted with data curation.

All authors contributed to the discussion and revision of the paper and approved the final manuscript.

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

Financial support

This research has been supported by the Joint Fund Project of the National Natural Science Foundation of China (Grant No. U2268217), the Scientific Research and Development Plan of China State Railway Group Co., Ltd. (Grant No. K2024T001), and the Fund Project of China Academy of Railway Sciences Group Co., Ltd. (Grant Nos. 2025YJ095, 2025YJ106).

References

Calbo, J., Long, C. N., González, J. A., Augustine, J., and McComiskey, A.: The thin border between cloud and aerosol: sensitivity of several ground-based observation techniques, *Atmos. Res.*, 196, 248–260, 2017.

Yann, F., Bijan, N., Stefan, W., Niklas, B., Rudolph, T., Marcel, H., Pascal, K., Z. L., F., and Robert, P. P.: Applying self-supervised learning for semantic cloud segmentation of all-sky images, *Atmos. Meas. Tech.*, 15, 797–809, 2022.

Li, Z., Rosenfeld, D., and Fan, J.: Aerosols and their impact on radiation, clouds, precipitation, and severe weather events, *Oxford Research Encyclopedia of Environmental Science*, 2017.

Wang, M., Zhuang, Z., Wang, K., Zhou, S., and Liu, Z.: Intelligent classification of ground-based visible cloud images using a transfer convolutional neural network and fine-tuning, *Opt. Express*, 29, 41176–41190, 2021.

Calbó, J., and Sabburg, J.: Feature Extraction from Whole-Sky Ground-Based Images for Cloud-Type Recognition, J. *Atmos. Ocean. Technol.*, 25, 3, 2008.

Heinle, A., Macke, A., and Srivastav, A.: Automatic cloud classification of whole sky images, *Atmos. Meas. Tech.*, 3, 2010.

Sun, X., Liu, L., and Zhao, S.: Whole Sky Infrared Remote Sensing of Cloud, *Procedia Earth Planet. Sci.*, 2, 278–283, 2011.

Zhuo, W., Cao, Z., and Xiao, Y.: Cloud Classification of Ground-Based Images Using Texture–Structure Features, J. *Atmos. Ocean. Technol.*, 31, 79–92, 2014.

Cheng, H. Y., and Yu, C. C.: Block-based cloud classification with statistical features and distribution of local texture features, *Atmos. Meas. Tech.*, 8, 3, 1173–1182, 2015.

Ye, L., Cao, Z., and Xiao, Y.: DeepCloud: Ground-Based Cloud Image Categorization Using Deep Convolutional Features, *IEEE Trans. Geosci. Remote Sens.*, PP, 1–12, 2017.

Zhang, J., Liu, P., Zhang, F., and Song, Q.: CloudNet: Ground-Based Cloud Classification With Deep Convolutional Neural Network, *Geophys. Res. Lett.*, 45, 8665–8672, 2018.

Fang, C., Jia, K., Liu, P., and Zhang, L.: Research on cloud recognition technology based on transfer learning, 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), IEEE, 791–796, 2019.

Wang, M., Zhou, S., Yang, Z., and Liu, Z.: Clouda: A ground-based cloud classification method with a convolutional neural network, *J. Atmos. Ocean. Technol.*, 37, 1661–1668, 2020.

Liu, S., Duan, L., Zhang, Z., Cao, X., and Durrani, T. S.: Ground-based remote sensing cloud classification via context graph attention network, *IEEE Trans. Geosci. Remote Sens.*, 60, 1–11, 2021.

https://doi.org/10.5194/egusphere-2026-1512
Preprint. Discussion started: 11 May 2026

Li, X., Qiu, B., Cao, G., Wu, C., and Zhang, L.: A novel method for ground-based cloud image classification using transformer, *Remote Sens.*, 14, 3978, 2022.

Liang, W., Tingting, Z., Yiren, G., and Chao, N.: MMST: A Multi-Modal Ground-Based Cloud Image Classification Method, *Sensors*, 23, 2023.

Li, Z., Kong, H., and Wong, C. S.: Neural Network-Based Identification of Cloud Types from Ground-Based Images of Cloud Layers, *Appl. Sci.*, 13, 2023.

Long, C., Li, X., Jing, Y., and Shen, H.: Bishift Networks for Thick Cloud Removal with Multitemporal Remote Sensing Images, *Int. J. Intell. Syst.*, 2023, 1–17, 2023.

Fan, L., Yao, Z., and Jianxue, W.: Recent advances in intra-hour solar forecasting: A review of ground-based sky image methods, *Int. J. Forecast.*, 39, 244–265, 2023.

Shi, C., Han, L., Zhang, K., Xiang, H., Li, X., Su, Z., and Zheng, X.: Improved RepVGG ground-based cloud image classification with attention convolution, *Atmos. Meas. Tech.*, 17, 979–997, 2024.

Guzel, M., Kalkan, M., Bostanci, E., Acici, K., and Asuroglu, T.: Cloud type classification using deep learning with cloud images, *PeerJ Comput. Sci.*, 10, e1779, 2024.

Mehta, S., and Rastegari, M.: Separable self-attention for mobile vision transformers, *Trans. Mach. Learn. Res.*, https://doi.org/10.48550/arXiv.2206.02680, 2023.

Han, K., Wang, Y., Xu, C., Guo, J., Xu, C., Wu, E., and Tian, Q.: GhostNets on heterogeneous devices via cheap operations, *Int. J. Comput. Vis.*, 130, 1050–1069, https://doi.org/10.1007/s11263-022-01575-y, 2022.

Tan, M., and Le, Q. V.: EfficientNet: rethinking model scaling for convolutional neural networks, in: Proceedings of the 36th International Conference on Machine Learning, PMLR, 6105–6114, 2019.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L. C.: Inverted residuals and linear bottlenecks: mobile networks for classification, detection and segmentation, *IEEE Trans. Pattern Anal. Mach. Intell.*, 42, 3017–3032, https://doi.org/10.1109/TPAMI.2019.2952529, 2020.

Ma, N., Zhang, X., Zheng, H. T., and Sun, J.: ShuffleNet V2: practical guidelines for efficient CNN architecture design, *IEEE Trans. Pattern Anal. Mach. Intell.*, 42, 1968–1983, https://doi.org/10.1109/TPAMI.2019.2903360, 2020.

Mehta, S., and Rastegari, M.: MobileViT: light-weight, general-purpose, and mobile-friendly vision transformer, in: Proceedings of the International Conference on Learning Representations, 2022.

Shelhamer, E., Long, J., and Darrell, T.: Fully convolutional networks for semantic segmentation, *IEEE Trans. Pattern Anal. Mach. Intell.*, 39, 640–651, https://doi.org/10.1109/TPAMI.2016.2572683, 2017.

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., and Keutzer, K.: SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size, in: Proceedings of the International Conference on Learning Representations, 2017.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q.: Convolutional networks with dense connectivity, *IEEE Trans. Pattern Anal. Mach. Intell.*, 41, 1628–1641, https://doi.org/10.1109/TPAMI.2019.2918284, 2019.