Hindawi
Journal of Sensors
Volume 2023, Article ID 9835050, 1 page
https://doi.org/10.1155/2023/9835050

Retraction

Retracted: Channel-Wise Correlation Calibrates Attention Module for Convolutional Neural Networks

Journal of Sensors

Received 19 December 2023; Accepted 19 December 2023; Published 20 December 2023

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] Z. Lu, Y. Dong, J. Li, Z. Lu, P. He, and H. Ru, "Channel-Wise Correlation Calibrates Attention Module for Convolutional Neural Networks," *Journal of Sensors*, vol. 2022, Article ID 2000170, 10 pages, 2022.

Hindawi
Journal of Sensors
Volume 2022, Article ID 2000170, 10 pages
https://doi.org/10.1155/2022/2000170

Research Article

Channel-Wise Correlation Calibrates Attention Module for Convolutional Neural Networks

Ziqiang Lu, Yanwu Dong, Jie Li, Ziying Lu, Pengjie He, and Haibo Ru

State Grid UHV Transmission Co. of SEPC, Taiyuan, Shanxi, China 030000

Correspondence should be addressed to Ziqiang Lu; luziqiang41616@163.com

Received 20 July 2022; Accepted 15 September 2022; Published 3 October 2022

Academic Editor: Yuan Li

It is well known in image recognition that global features represent the overall and have the ability to generalize an entire object, while local features can reflect the details, both of which are important for extracting more discriminative features. Recent research has shown that the performance of convolutional neural networks can be improved by introducing an attention module. In this paper, we propose a simple and effective channel attention module named layer feature that meets channel attention module (LC module, LCM), which combines the layer global information with channel dependence to calibrate the correlation between channel features and then adaptively recalibrates channel-wise feature responses. Compared with the traditional channel attention methods, the LC module utilizes the most significant information that needs to be focused on in the overall features to refine the channel relationship. Through empirical studies on CIFAR-10, CIFAR-100, and mini-ImageNet, this work proved its superiority compared to other attention modules in different DCNNs. Furthermore, we performed the two-dimensional visualization of the feature map through the class activation map and intuitively analyzed the effectiveness of the model.

1. Introduction

The deep convolutional networks (DCNNs) are fundamental for visual cognition tasks, including image [1], classification [1, 2], object detection [3, 4], target tracking [5], action recognition [6], and semantic segmentation [7]. Designing a more efficient network architecture and producing refined features is essential to further improve the performance of DCNNs and promote the development of visual cognition tasks.

Over the past few years, several DCNNs have been proposed with a state-of-the-art performance at that time, from AlexNet [8] and VGG [9] networks only stacking convolutional layers to GoogleNet [10–13] using a multiscale convolution kernel on a single convolutional layer and even ResNet [1] adding skip connection mechanism to propagate information to deeper layers of networks, and the accuracy and depth of CNNs are constantly improving. To ensure maximum information flow between layers, Dense Net [2] connects all layers directly with each other. Each layer gets inputs from all previous layers and passes on its own feature map to all subsequent layers.

No matter what kind of network, DCNNs autonomously extract features gradually from low-level local features to high-level global features through a stack of convolutional operators. The high-level global feature vector takes the entire image into consideration, which reflects the overall attributes or specific parts in the image. In contrast, each dimension low-level local feature vector corresponds to only one kind of feature on the image, focusing on extracting detailed features like edges and noise in the image.

The combination of local and global features has been explored by a few studies [10–15]. In the convolutional neural network, if global features are incorporated into the features of each convolutional layer, the features extracted by the convolutional layer will be more refined, which will help improve the performance of the model. GoogleNet [10–13] is a typical example; it has a simple concatenation that is designed to aggregate multiscale information from different convolutional kernels inside the "inception" building block. In face recognition, many studies also use both global and local features to represent faces and use global features to describe the overall attributes of the face for rough matching [14, 16], such as skin color, contours, and the distribution of facial organs. Local features are used to describe the detailed changes of the human face for detailed confirmation, such as the characteristics of facial organs and some strange features of the face [14, 16] (moles, scars, dimples, etc.). Recently, the attention mechanism module, plugged into the networks, has received increasing attention, achieving much better performance than traditional networks in various vision tasks [17–24]. Among them are a lot of works on the channel attention mechanism [17, 19] which are used to refine channel local representation, and its performance is getting better and better. Exploring the integration of layer global features and channel local attention mechanism is one of the effective methods to improve the accuracy of modeling channel dependence and refining features.

In this paper, motivated by attention mechanism and global-local fusion mechanism, we propose a simple channel attention module, called layer feature, that satisfies the channel attention module(LC module, LCM), which not only realizes the channel attention in each convolutional layer but also incorporates the global feature of each layer. The layer global feature represents the most significant information extracted by each convolutional layer and reflects that should be most concerned about in each layer. The LC module first combines the significant feature with channel local features to calibrate the weight of channels and then adaptively recalibrates channel-wise feature responses through the attention operation.

The proposed LCM is lightweight, and it adds only small parameters and computational cost, which can be conveniently inserted at any location in the deep convolutional neural networks. In order to explore the optimal feature fusion strategy, this paper proposes two LC modules, named LC module 1 and LC module 2. Both LC modules consist of a quadruplet of operators: capture channel local features, capture layer global features, combine layer feature with channel dependence, and reweight as shown in Figure 1. In the first and second operation, LC module 1 and LC module 2 are implemented in the same way. Given the input feature maps, the first operation aggregates the feature maps across spatial dimensions weight and height in the feature to produce a channel-local descriptor. One descriptor represents a kind of local feature. The first operation is the foundation of the second. In the second operation, channel local features of the convolutional layer are used to calculate the layer features by pooling operation across channel dimensions. As we all know, the implementation of pooling operation is simple and less parameter. In the third operation, the implementation of LC module 1 and LC module 2 are slightly different. In LC module 1, channel local features utilize softmax layer to adaptively obtain layer information and then adopt activation operation to learn relationship between the features, while in LC module 2, the local features first express the channel feature correlation and then uses the layer information to modify the channel correlation. One is feature-to-feature refinement, and the other is salient feature-to-weight refinement. At last, the feature maps are reweighted to generate the output. We plugged two LC modules into the DenseNet and ResNet and proved the performance on CIFAR-10 and CIFAR-100. Experiments have found that in DenseNet and ResNet, LC module 2 obtains higher classification accuracy. And we determined LC module 2 as the final module and renamed it as LC module. Then on the CIFAR-10, CIFAR-100, and mini-ImageNet datasets, this paper verifies its effectiveness by comparing with the current excellent attention methods in several kinds of DCNNs and different depths. Through massive experiments, our method obtained competitive results. At last, we apply the Score-CAM to different networks to intuitive analysis on mini-ImageNet validation set. The visualization of feature map further proves the effectiveness of the module.

In summary, our contributions are as follows: (1) we propose a simple and effective channel attention module named layer feature that meets channel attention module (LC module, LCM), which can focus on the overall features to refine the channel relationship. (2) Comprehensive experiments with different DCNNs on widely used classification datasets (CIFAR-10, CIFAR-100, and mini-ImageNet) demonstrate the superior performance of the proposed method.

2. Related Word

Deep convolutional neural networks are currently the most commonly used method for image classification. Given images as input, by simulating the human visual system, the deep convolutional neural network autonomously extracts features gradually from low-level detailed features to high-level semantic features and finally generating global image representations connected with softmax layer for classification [1, 8, 9].

With a large number of deep convolutional neural networks proposed, from AlexNet [8], VGG [9], GoogleNet [10–13], to ResNet [1], RoR [25] and DenseNet [2], deep convolutional neural networks are relatively mature. In order to better characterize the complex boundaries of thousands of classes in a very high-dimensional space, a feasible approach is to further explore the feature extraction of the model for enhancing modeling capability of convolutional neural networks, so that the deep convolutional neural network can extract more and finer image features. Vision attention mechanism is one of the effective measures.

The vision attention mechanism is a unique signal processing mechanism of human vision. When looking at an image, humans quickly scan the global image to obtain important areas and suppress other useless information. Inspired by visual attention mechanisms, more and more studies have introduced attention mechanisms to neural networks to improve performance. Wang et al. [18] proposed the residual attention network. The network adds a soft mask branch on the basis of the original residual block. The residual attention network refines the feature map and improves the learning ability of the network by utilizing multiple attention modules. Hu et al. [17] proposed a squeeze-and-excitation (SE) module to adaptively recalibrate channel-wise feature responses, which consists of a squeeze operation and an excitation operation. The squeeze operation aggregates the feature maps across spatial dimensions to produce a channel descriptor embedding the global distribution of channel-wise feature. Then, the purpose of the

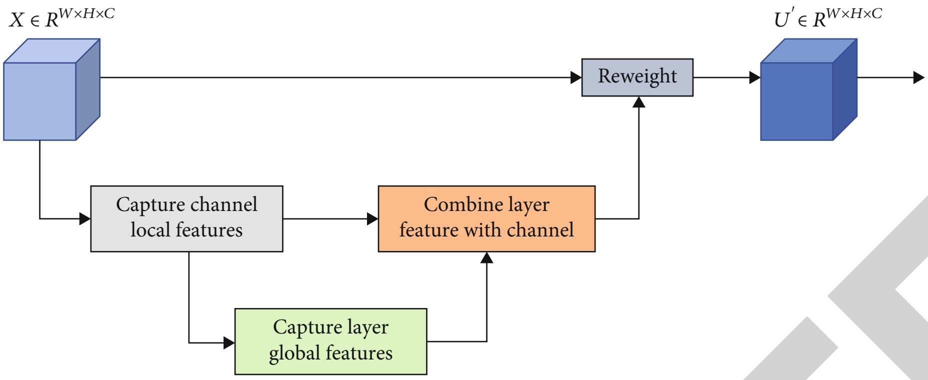
$X \in R^{W\times H\times C}$
Reweight
$U' \in R^{W\times H\times C}$
Capture channel local features
Combine layer feature with channel
Capture layer global features

Figure 1: The layer feature that meets channel attention module (LC module, LCM). LC module consists of a triplet of operators: capture channel local features, capture layer global features, and combine layer feature with channel dependence. The LC module first combines the significant feature with channel local features to calibrate the weight of channels and then adaptively recalibrates channel-wise feature responses through the attention operation.

excitation operation is to fully capture the channel-wise dependencies. Furthermore, inspired by the SE module, the convolutional block attention module (CBAM) [19] emphasizes meaningful features in two dimensions: channel and spatial axes. Zhang et al. [15] combined ResNet or RoR models with LSTM units to effectively improve the accuracy of age estimation by extracting age-sensitive local regions.

Recently, in order to learn higher-order representations for enhancing nonlinear modeling capability, attention mechanism adopting second-order pooling has received more and more attentions, achieving much better performance than first-order methods in various vision tasks. Both convolutional and recurrent operations deal with a local neighborhood in space or time. Wang et al. [20] presented nonlocal block (NL block) to capture long-range dependencies using deep neural networks. In a nonlocal block, each location in the feature map is connected with all other locations through self-adaptively predicted attention maps. This situation leads to high computational complexity and huge number of GPU memory. In order to capture long-range dependencies more efficiently and effectively, Huang et al. [21] proposed criss-cross network (CCNet), in which each pixel first obtains the contextual information of its surrounding pixels on the criss-cross path through a criss-cross attention module and then the long-range dependencies are obtained from all pixels by taking a further recurrent operation. By fusing NL block and SE block, Cao et al. [22] proposed GCNet to effectively model the global context, achieving better performance than both NL block and SE block on major benchmarks for various recognition tasks. To capture the global feature dependencies in the spatial and channel dimensions, Fu et al. [26] proposed dual attention network (DANet). The position attention module learns the spatial interdependencies of features and channel attention module models channel interdependencies. Chen et al. [23] proposed the double attention block. The double attention block captures long-range feature interdependencies in two steps, where the first step collects features from the entire space into a compact set through second-order attention pooling and the second step adaptively selects and distributes features to each location. Gao et al. [24] proposed a novel global second-order pooling (GSoP) capturing global second-order statistics along channel dimension or position dimension, which can be easily inserted into existing deep neural networks conveniently with low computational complexity and less number of GPU memory.

3. Method

In order to achieve channel attention including the layer information, we illustrate two modules named layer feature that meets channel attention module 1 (LC module 1, LCM1) and layer feature that meets channel attention module 2 (LC module 2, LCM2). Note that the two LC modules can be conveniently inserted at any location in a deep convolutional neural network. And they all modify the channel local feature through fusing layer information. The difference is that the channel local feature is adaptively integrated with the layer global feature, and then, they together capture the correlation between channel features in LC module 1, while the LC module 2 first expresses the channel feature correlation and then uses the layer feature to modify the channel correlation. That is to say, one is feature-to-feature refinement, and the other is salient feature-to-weight refinement. We explain two modules in detail.

3.1. LC Module 1. Figure 2 shows the diagram of the layer feature that meets channel attention module 1. Obviously, this module is simple. We implement the module via three operators—capture channel local features, capture layer global features, and combine layer feature with channel dependence. For any given feature maps, $X \in R^{W\times H\times C}$. By default, the feature maps first conduct transformations $F : X \longrightarrow U$, composed with 3×3 convolutions, both batch normalization (BN) and a rectified linear unit (ReLU) function in sequence. Note that $U \in R^{W'\times H'\times C'}$.

3.1.1. Capture Channel Local Features. Inspired by SE module, we capture the channel local features by squeeze operation. In $U \in R^{W'\times H'\times C'}$, H' and W' are the space height and width and C' is the number of channels. We aggregate the feature maps across the spatial dimensions (height and

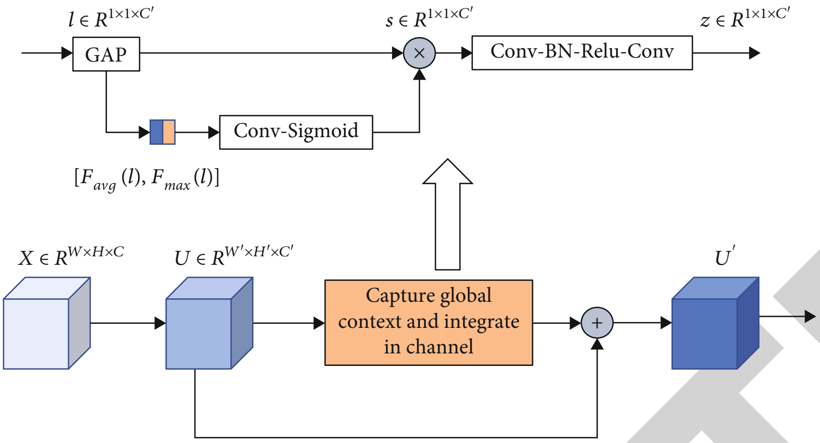
$l \in R^{1\times1\times C'}$
GAP
$s \in R^{1\times1\times C'}$
×
Conv-BN-Relu-Conv
$z \in R^{1\times1\times C'}$
Conv-Sigmoid
$[F_{avg}(l), F_{max}(l)]$
$X \in R^{W\times H\times C}$
$U \in R^{W'\times H'\times C'}$
Capture global context and integrate in channel
+
U'

Figure 2: Layer feature that meets channel attention module 1 (LC module 1).

weight) and then convert each two-dimensional feature map to a channel descriptor by using global average pooling. Specifically, this j^{th} element is computed by shrinking U by spatial dimensions $W \times H$:

$$l_j = F_{gp}(U_j) = \frac{1}{H' \times W'} \sum_{i=1}^{H'} \sum_{K=1}^{W'} U_j(i,k), \quad (1)$$

where $l = [l_1, l_2, \cdots l_{C'}]$, [] denotes concatenate operation and $l \in R^{C' \times 1 \times 1}$, and $F_{gp}(\cdot)$ denotes global average pooling operation.

3.1.2. Capture Layer Global Features. Global features can reflect the overall change of the image. Capturing the layer significant feature that need to be concerned will help the network to understand the connotation of the layer. Based on the output features of each convolutional layer, we use pooling across the channel dimensions to capture the layer information. We argue that max pooling collects important unique object features and average pooling gathers universal features between channels. Thus, to capture the layer global features, we first apply both average pooling and max pooling operations simultaneously along the channel axis and connect them. It is effective to apply pooling operations along the channel axis. Then on the concatenated feature descriptor, a convolution layer is applied to generate a layer descriptor that represents the overall distribution of the convolutional layer. We will describe the detailed operation below.

For $l = [l_1, l_2, \cdots l_{C'}]$, we first aggregate channel information by using two pooling operations, generating two layer global feature descriptor: $F_{avg}(l) \in R^{1\times1\times1}$ and $F_{\max}(l) \in R^{1\times1\times1}$. They are then concatenated and convolved with standard convolutional layers to generate our layer descriptors.

$$g = \delta\left(f_{\text{conv}}\left(\left[F_{\text{avg}}(l), F_{\max}(l)\right]\right)\right), \quad (2)$$

where f_{conv} represents a convolution operation with the filter size of 3×3, δ represents the sigmoid function, and $g \in R^{1\times1\times1}$. We can find that the parameters of this layer are only $2 \times 1 \times 1 \times 1$. The module is extremely lightweight.

3.1.3. Combine Layer Feature with Channel Dependence. We fuse the channel local features with the layer information, adopting layer feature to achieve the purpose of modifying channel local features, so as refine features. The basic idea is adaptively carrying global information into different channel local features. To achieve this goal, we first apply a softmax operator on the channel local features:

$$l'_j = \frac{e^{l_j}}{\sum_{K=1}^{C'} e^{l_k}}, \quad (3)$$

where $l_j \in l \in R$. Note that $l'_j \in (0,1)$, which denotes the weight-gathering global information. Then, we fuse information from global feature branch and local feature branch via a multiplication operation:

$$s_j = l'_j \times g, \quad (4)$$

where s = $[s_1, s_2, ,\cdots, s_{C'}]$, which adaptively fuse the channel local features with layer significant information, and $s \in R^{C'\times1\times1}$.

Further, in order to recalibrate feature responses, we create a compact feature $z \in R^{C'\times1\times1}$ to refine the input feature map U. This is achieved by a composite function of consecutive operations: convolution (Conv, 3×3), batch normalization (BN), followed by a rectified linear unit (ReLU), and then convolution (Conv):

$$z = w_2(\sigma(B(\omega_1 s))), \quad (5)$$

where σ is the ReLU function, B represents the batch normalization, $\omega_1 \in R^{(C'/r)\times C'}$, and $\omega_2 \in R^{C'\times C'/r}$. To limit the complexity of the model, we use a bottleneck, where Conv first adopts a dimensionality reduction with parameters w_1 and a dimensionality reduction ratio r and then adopts a dimensionality-increasing Conv layer with parameters w_2.

3.1.4. Reweight. The output of combining layer feature with channel dependence represents channel-wise dependence. The continuous multiplication operation will make the output feature very small, which is not conducive to the back propagation of the gradient. In order to complete the refinement of input features, the final output of the block is obtained by adding the feature z with transformation output U for adaptive feature refinement:

$$U' = U + z. \tag{6}$$

The LC module 1 adaptively carries global information into different channel local features and then models channel-wise feature dependencies. It achieves feature-to-feature refinement.

3.2. LC Module 2. Figure 3 shows the diagram of the layer feature that meets channel attention module 2 (LC module 2). Obviously, the module is simple. In LC module 2, the output feature first passed through two streams to capture channel relationship and layer global feature. Both processes are applied in parallel, and the output of the two streams is added and normalized with the sigmoid function. The layer global information is used to refine channel relationship. The following describes the details of our LC module 2. Same as LC module 1, given feature maps first conduct transformations. Note that $U \in R^{W' \times H' \times C'}$.

3.2.1. Capture Channel Local Features. The output feature maps are passed through global average pooling operation. This operation aggregates the feature maps across the spatial dimensions and converts each two-dimensional feature map into a channel descriptor. The above shares the same implementation with LC module 1.

3.2.2. Capture Layer Global Features. The global feature represents the overall performance features of the layer and the essential features that need to be concerned, which is essential for the network to correctly understand the high-level semantics of the image. This part shares the same implementation with the LC module 1. The feature characterizes the most significant information in the layer, which is complementary to the channel relationship.

3.2.3. Combine Layer Feature with Channel Dependence. The fusion method of LC module 2 is different from LC module 1. Layer information refine the channel relationship in LC module 2 where layer information recalibrates channel feature in LC module 1. l_k is the descriptor of a channel. We first utilize channel descriptors to calculate the interdependencies between channels. For capturing channel-wise dependencies information, two full-connected layers have been commonly adopted so far. Hu et al. adopt it in their attention module to learn a nonmutually exclusive relationship. Woo et al. applied a multilayer perceptron (MLP) with one hidden layer. And in the middle layer, they all use the ReLU function as follows: $\text{ReLU}(Wl) = \max\,(W, l, 0)$. However, the ReLU function has obvious defects: expected mean is not 0 and convergence is slow. In order to ease optimization, we adopt Conv-BN-ReLU-Conv operations to capture channel dependencies. Meanwhile, in order to reduce parameter overhead, the filter of first 1×1 Conv is set to $R^{(C/r) \times 1 \times 1}$, where r is the reduction ratio. This is followed by batch normalization operation. In short, the process is calculated by

$$u = W_2 \text{ReLU}(\text{BN}(W_1 l)), \tag{7}$$

where W_1 and W_2 are the Conv weight, $W_1 \in R^{(C/r) \times C}$, and $W_2 \in R^{C \times C/r}$.

In order to utilize layer global significant information to refine local channel relationship, the output of the two streams is added and normalized with the sigmoid function. It performs dynamic channel-wise feature recalibration. The process can be computed as

$$s = \text{sigmoid}(u + g). \tag{8}$$

3.2.4. Reweight. Finally, the final output is obtained by rescaling the U with the attention weight s, which is calculated by

$$U' = U \cdot s, \tag{9}$$

where refers to the channel-wise multiplication.

Since our LC module 2 is extremely lightweight, it can be applied in multiple layers with only a slight increase in parameter and computational cost. Our module not only calculates the relationship between local channels but also incorporates the most significant global layer information. The modeling method allows global layer information that need to be concerned to refine local channel-wise relationship at each layer. On the other hand, when strengthening local important information and weakening local noise information, we consider globally significant information, which makes the strengthening or weakening of local channel information more accurate and produces strong discriminative features.

4. Experiments and Results

In this section, we will present the results of the experiment and the details of our implementation. First, we show the validity of LC module 1 (LCM1) and LC module 2 (LCM2). Then, we present the robustness and effectiveness of LCM based on DCNNS of different types and different depths on CIFAR datasets and mini-ImageNet. The implementation is in Pytorch.

4.1. Experiment Setups

4.1.1. Datasets. This paper conducts experiments on three different datasets for image classification tasks. The three datasets include CIFAR-10, CIFAR-100, and mini-ImageNet.

CIFAR: the CIFAR dataset [27] is the most commonly used datasets for image classification task and consists of 32×32 pixels colored natural scene images. The dataset

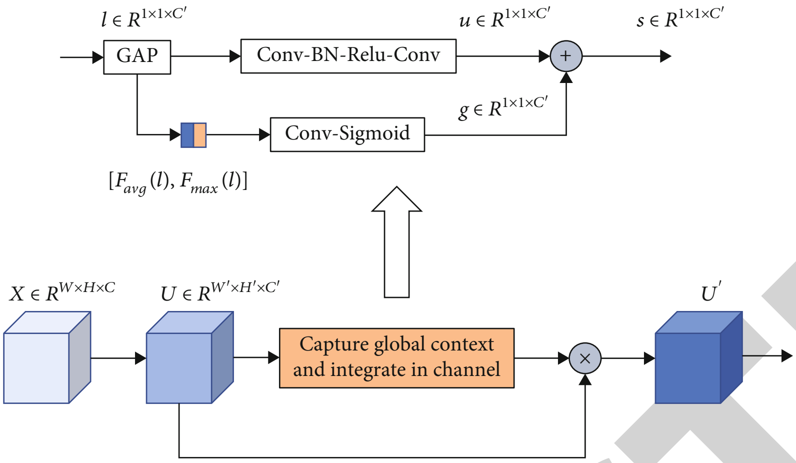
$l \in R^{1\times1\times C'}$
GAP
Conv-BN-Relu-Conv
$u \in R^{1\times1\times C'}$
+
$s \in R^{1\times1\times C'}$
$g \in R^{1\times1\times C'}$
Conv-Sigmoid
$[F_{avg}(l), F_{max}(l)]$
$X \in R^{W\times H\times C}$
$U \in R^{W'\times H'\times C'}$
Capture global context and integrate in channel
×
U'

Figure 3: Layer feature that meets channel attention module 2 (LC module 2).

Table 1: Test accuracy (%) on CIFAR-10/CIFAR-100 by DenseNet.

Model	Para (M)	GFloat	CIFAR-10	CIFAR-100
DenseNet-40	1.059	0.293	94.67	74.69
DenseNet-40+LCM1	1.065	0.293	95.26	75.61
DenseNet-40+LCM2	1.065	0.293	95.32	75.79
DenseNet-64	2.830	0.761	95.20	77.52
DenseNet-64+LCM1	2.840	0.761	95.84	78.22
DenseNet-64+LCM2	2.840	0.761	95.84	78.24
DenseNet-100	7.084	1.875	95.66	78.76
DenseNet-100 +LCM1	7.100	1.875	95.97	79.12
DenseNet-100 +LCM2	7.100	1.875	96.01	79.44

Table 2: Test accuracy (%) on CIFAR-10/ CIFAR-100 by ResNet.

Model	Para (M)	GFloat	CIFAR-10	CIFAR-100
ResNet-20	0.270	0.041	92.26	68.34
ResNet-20+LCM1	0.278	0.041	92.77	69.40
ResNet-20+LCM2	0.278	0.041	92.70	69.53
ResNet-32	0.464	0.070	93.30	70.67
ResNet-32+LCM1	0.478	0.070	93.90	70.88
ResNet-32+LCM2	0.478	0.070	94.12	71.75

includes a training set and a test set, where the training set contains 50,000 images and the test set contains 10,000 images. CIFAR-10 images are drawn from 10 classes, and the CIFAR-100 images are drawn from 100 classes. Compared with CIFAR-10, the CIFAR-100 dataset contains more categories. But the number of images per category in the CIFAR-100 dataset is relatively small. Overfitting is easy to occur in image classification tasks, which is more challenging. We employ a standard data augmentation scheme in our experiments. First, zero-pad the image with 4 pixels on each side, and then, randomly crop it to generate 32×32 images again. Finally, mirror half of the images horizontally. In terms of data preprocessing, we preprocess the dataset by subtracting the mean and dividing the standard deviation.

Mini-ImageNet: the original ImageNet dataset [28] is a popular large-scale benchmark for training deep neural networks. It contains 1.28 million training images and 50 k validation images from 1000 classes. Since the cost of performing experiments on the ImageNet dataset might be prohibitive, we do not have enough resources to train the ImageNet dataset. We select the mini-ImageNet [29] to evaluate the networks. Mini-ImageNet contains 100 classes and the image size is the same as ImageNet. In the training set, each class contains 500 images. And in the test set, each class contains 100 images. This dataset is more complex than CIFAR but fits in memory on modern machines, making it very convenient for the image classification task. We first train networks on the training set and then report the top-1 and top-5 errors on the test set. Image preprocessing and data augmentation methods are identical to ImageNet. We use the same data augmentation method [1] when training images. Meanwhile, we apply a single crop with a size 224×224 in the test.

4.1.2. Network Use and Training Strategies. In this paper, we use three different networks in the experiment for comparison. The three networks are DenseNet [2], ResNet [1], and ResNeXt [30].

On the CIFAR dataset, we use Stochastic Gradient Descent (SGD) to train the network for 300 epochs, and set a mini-batch size for per epoch to 64. We use a weight decay of 1e-4 with a Nesterov momentum of 0.9. The learning rate starts at 0.1 and is divided by 10 during 50% and 75% of the training procedure.

On mini-ImageNet, we set 100 epochs when training the models and set the batch size to 64 for per epoch. The learning rate is initially set to 0.1 and reduced by a factor of 10 at epoch 30 and 60. All experiments are implemented on Pytorch 1.0 with one NVIDIA GeForce GTX TITAN X Pascal GPU.

Table 3: Test accuracy (%) on different models on CIFAR-10.

Model	Para (M)	GFloat	CIFAR-10
DenseNet-100	7.084	1.875	95.66
DenseNet-100+SE	7.113	1.875	95.74
DenseNet-100+nonlocal	7.084	1.875	95.90
DenseNet-100+CBAM	7.121	1.879	94.53
DenseNet-100+GC	7.117	1.875	95.93
DenseNet-100+LCM	7.100	1.875	96.01
ResNet-50	0.756	0.113	94.14
ResNet-50+SE	0.778	0.113	94.34
ResNet-50+CBAM	0.781	0.114	94.25
ResNet-50+nonlocal	0.758	0.114	94.18
ResNet-50+GC	0.778	0.113	94.10
ResNet-50+LCM	0.778	0.113	94.64

Table 4: Test accuracy (%) on different models on CIFAR-100.

Model	Para (M)	GFloat	CIFAR-10
DenseNet-100	7.084	1.875	78.76
DenseNet-100+SE	7.113	1.875	79.41
DenseNet-100+nonlocal	7.084	1.875	79.22
DenseNet-100+CBAM	7.121	1.879	75.86
DenseNet-100+GC	7.117	1.875	79.14
DenseNet-100+LCM	7.100	1.875	79.44
ResNet-50	0.756	0.113	72.25
ResNet-50+SE	0.778	0.113	73.04
ResNet-50+CBAM	0.781	0.114	72.95
ResNet-50+nonlocal	0.758	0.114	72.82
ResNet-50+GC	0.778	0.113	71.85
ResNet-50+LCM	0.778	0.113	73.87

4.2. Compare the Effectiveness of LCM1 and LCM2. In order to verify the validity of LC module 1 (LCM1) and LC module 2 (LCM2), we inserted two modules into ResNet and DenseNet and conducted experiments on the CIFAR dataset. The results are shown in Tables 1 and 2. This part chooses 40-layer, 64-layer and 100-layer DenseNet. The DenseNet used in CIFAR experiments has three dense blocks with equal numbers of layers. Within each dense block, all the convolutional layers use filters with kernel size 3 × 3, and each side of the inputs is zero-padded by one pixel to keep the feature map size fixed. In the three dense blocks, the feature map sizes are 32 × 32, 16 × 16, and 8 × 8, respectively. The DenseNet+LCM1 means DenseNet inserted LCM1. And DenseNet+LCM2 means DenseNet inserted LCM2. And in ResNet

s, we present experiments in 20 layers and 32 layers. The ResNet has three blocks with equal numbers of residual block, and all the residual blocks use filters with kernel size 3 × 3. Similar to DenseNet, the feature map sizes in the three blocks are 32 × 32, 16 × 16, and 8 × 8, respectively.

In Tables 1 and 2, it can be found that no matter whether LCM1 or LCM2, the parameters and calculations are very small. And parameters and calculations in LCM1 and LCM2 are almost the same. In Table 1, the classification results of DenseNet inserted LCM1 or LCM2 are better than DenseNet on CIFAR-10 and CIFAR-100 datasets in different depths. And no matter what depth or datasets, compared with DenseNet inserted LCM1, the classification accuracy of DenseNet inserted LCM2 is slightly higher or almost the same. Table 2 shows the results on ResNet. We can see that ResNet with LCM1 or LCM2 outperforms ResNet in different depths. It is proved that the global information extracted from each convolution layer can refine local channel-wise relationship at each layer. It is shown that ResNet with LCM2 gets higher accuracy almost no matter whether it is on CIFAR-10 or CIFAR-100, except for the 20-layer ResNet, the results of embedding LCM1 and embedding LCM2 are almost similar on CIFAR-10. In general, LCM2 is more effective. The reason is that in LCM2, the global feature not only corrects the channel correlation but also introduces bias in the channel attention module. The introduction of bias increases the flexibility and fitting ability of the neural network. We identified LCM2 as the most effective module, renamed as layer feature that meet channel attention module (LCM), merging the global layer information with channel correlation feature and implementing channel attention. Next, we will explore the effectiveness of layer feature that meets channel attention module on different models and datasets.

Table 5: Test accuracy (%) on different models on mini-ImageNet.

Model	Para (M)	GFloat	Top1	Top5
DenseNet-121	8.081	2.898	82.45	95.17
DenseNet-121+SE	8.113	2.898	82.61	95.17
DenseNet-121+nonlocal	8.084	2.899	82.19	95.09
DenseNet-121+CBAM	8.119	2.902	80.60	94.56
DenseNet-121+GC	8.113	2.899	79.47	93.94
DenseNet-121+LCM	8.112	2.898	83.05	95.23
ResNet-50	23.71	4.132	80.54	94.60
ResNet-50+SE	26.24	4.137	81.46	94.94
ResNet-50+CBAM	26.25	4.143	82.33	95.15
ResNet-50+nonlocal	25.82	4.544	80.23	94.23
ResNet-50+GC	26.24	4.138	77.06	92.89
ResNet-50+LCM	24.97	4.133	81.94	95.09
ResNeXt-50 32 × 4d	23.19	4.287	81.54	94.76
ResNeXt-50 32 × 4d + SE	25.72	4.292	82.40	95.06
ResNeXt-50 32 × 4d + CBAM	25.72	4.298	82.56	95.31
ResNeXt-50 32 × 4d + nonlocal	25.29	4.699	81.13	94.53
ResNeXt-50 32 × 4d + GC	33.19	4.300	76.18	92.26
ResNeXt-50 32 × 4d + LCM	23.81	4.287	82.72	95.25

To further demonstrate the effectiveness of layer feature that meets channel attention module (LCM), we performed CIFAR and mini-ImageNet classification experiments to rigorously evaluate LCM. And we evaluate in various network architectures including DenseNet, ResNet, and ResNeXt

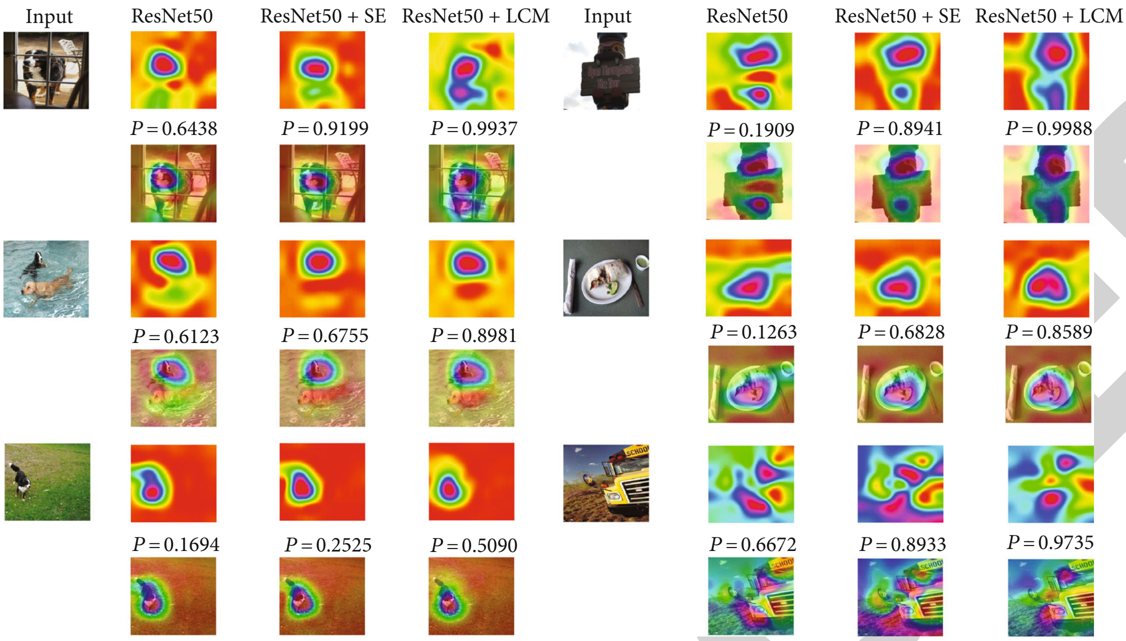

Figure 4: Score-CAM visualization results. We compare the visualization results of LC module-integrated network (ResNet50+LCM) with the results of the baseline (ResNet50) and SE-integrated network (ResNet50+SE). The Score-CAM visualization is calculated for the final convolutional outputs. On the middle of each input image, the ground-truth label is shown. And P represents the softmax score of each network for the ground-truth class. On the right side of the input image, the first line represents score-weighted class activation heatmap and the second line represents score-weighted class activation heatmap on image. In the two color bars, the corresponding degree gradually decreases from top to bottom.

and compare our LCM with SE module, nonlocal module, CBAM, and GC module. They are currently well-known attention modules.

4.3. Experiments on Different Networks and Different Datasets. Tables 3 and 4 shows the test accuracy on CIFAR-10 and CIFAR-100, respectively. We evaluate our module on 100-layer DenseNet and 50-layer ResNet. Because the accuracy of the CIFAR-10 dataset is already very high, there is little room for improvement. On CIFAR-10, the test accuracy inserted LCM has slightly improved no matter on ResNet or DenseNet. And the networks with LCM outperform all the other attention modules significantly. And we can see that the accuracy of ResNet inserted LCM has been greatly improved, compared with other attention modules. Table 4 shows the test accuracy on CIFAR-100. This part of experimental results demonstrates that the LCM can generalize well on various models in the small-scale dataset.

Then, we prove the effectiveness on high-resolution dataset. In this part, we choose mini-ImageNet and conduct experiments on 121-layer DenseNet, 50-layer ResNet, and 50-layer 32 × 4 ResNeXt. The results are shown in Table 5. The 121-layer DenseNet has four dense blocks with equal numbers of layers. Within each dense block, all the convolutional layers use filters with kernel size 1 × 1 and 3 × 3, and each side of the inputs is zero-padded by one pixel to keep the feature map size fixed. And each 3 × 3 convolutional layer produces 12 feature maps. The reduction ratio of r is 1. The feature map sizes in four dense blocks are 56 × 56, 28 × 28, 14 × 14, and 7 × 7, respectively. The network uses 1 × 1 convolution and uses 2 × 2 average pooling between two contiguous dense blocks. A global average pooling is performed at the end of the last dense block, followed by a softmax classifier, which is attached. The 50-layer ResNet and 50-layer 32 × 4 ResNeXt also have four blocks with equal numbers of residual block. Within each block, all the residual block use filters with kernel size 1 × 1, 3 × 3, and 1 × 1. The reduction ratio of r is 32.

First, we compare our attention LCM against the standard architecture. Whether on DenseNet, ResNet, or ResNeXt, baseline models with LCM have better results, demonstrating that the LC module has good generalization ability to various models in the large-scale dataset. Furthermore, on the basis of SE, one of the most powerful channel attention methods, the LC module is adopted to build the model, which improves the accuracy of the model. It means that our proposed method is powerful. This shows the effectiveness of the layer global feature that includes information that needs to be concerned in each layer. Compared to SE, the LC module utilizes the most significant information that needs to be focused on in the overall features to refine the channel relationship. The feature is more discriminative. The result of our LC module is slightly lower than CBAM in ResNet. The CBAM sequentially infers attention maps through two independent dimensions, which include channel and spatial. Our module only considers channel attention. And from overall view, compared with SE module, nonlocal module, CBAM, and GC module on 121-layer DenseNet, 50-layer ResNet, and 50-layer 32×4 ResNeXt, our modules perform better in mini-ImageNet dataset. This suggests that the proposed LCM can achieve higher robustness and better generalization in the image classification.

4.4. Network Visualization with Score-CAM. On the mini-ImageNet validation set images, we apply Score-CAM [31] to different networks for qualitative analysis. Score-CAM is a recently proposed visualization approach that uses gradients to calculate the importance of the spatial locations in convolutional layers. We try to see how well this network exploits the features through observing the regions that the network has considered as important for predicting a class. We compare the visualization results of ResNet-50+LCM with baseline (ResNet50) and ResNet-50+SE. Figure 4 shows the visualization results. And the figure also shows the softmax scores of the target class. In Figure 4, we show the score-weighted class activation heatmap and the score-weighted class activation heatmap on image. We can clearly see that the masks of the ResNet-50+LCM cover the target object regions better than other methods. Obviously, in the first example, the heatmap with our LCM covers the target object regions more comprehensive. And in the last example, the masks of the ResNet-50+SE cover a lot of background regions, but our ResNet-50+LCM covers more target object regions. That is, the ResNet-50+LCM can make use of the information in target object regions.

5. Conclusions

In this study, we proposed a new channel attention module named layer feature that meets channel attention module (LC module, LCM), which utilizes layer global information that needs to be focused on to refine local channel-wise relationship at each layer. On the one hand, when strengthening local important information and weakening local noise information, the module considers globally significant information, which makes the strengthening or weakening of local channel information more accurate and produces strong discriminative features. On the other hand, it can be conveniently inserted anywhere in a deep convolutional neural network and imposes only a slight increase in parameter and computational cost. And whether on large datasets or small datasets, the module has better performance in image classification tasks.

Data Availability

The datasets generated and analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the Science and Technology Project of State Grid Shanxi Electric Power Company (No. 52051K21000B).

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, 2016.

[2] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2261–2269, Honolulu, HI, USA, 2017.

[3] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: towards real-time object detection with region proposal networks,” in *Advances in Neural Information Processing Systems*, pp. 91–99, Curran Associates, Inc., 2015.

[4] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988, Venice, Italy, 2017.

[5] A. He, C. Luo, X. Tian, and W. Zeng, “A twofold siamese network for real-time object tracking,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018.

[6] S. Sudhakaran, S. Escalera, and O. Lanz, “Gate-shift networks for video action recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020.

[7] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: a deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[9] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, https://arxiv.org/abs/1409.1556.

[10] C. Szegedy, W. Liu, Y. Jia et al., “Going deeper with convolutions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, Boston, MA, USA, 2015.

[11] S. Ioffe and C. Szegedy, “Batch normalization: accelerating deep network training by reducing internal covariate shift,” in *in International Conference on Machine Learning*, pp. 448–456, Lille, France, 2015.

[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, Las Vegas, NV, USA, 2016.

[13] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in *Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, California USA, 2017.

[14] A. R. Chadha, P. P. Vaidya, and M. M. Roja, "Face recognition using discrete cosine transform for global and local features," in *2011 International Conference on Recent Advancements in Electrical, Electronics and Control Engineering*, pp. 502–505, Sivakasi, India, 2011.

[15] K. Zhang, N. Liu, X. Yuan, X. Guo, C. Gao, and Z. Zhao, "Fine-grained age estimation in the wild with attention LSTM networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 9, 2019.

[16] Z. Liu and C. Liu, "Fusion of color, local spatial and global frequency information for face recognition," *Pattern Recognition*, vol. 43, no. 8, pp. 2882–2890, 2010.

[17] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, Salt Lake City, UT, USA, 2018.

[18] F. Wang, M. Jiang, C. Qian et al., "Residual attention network for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6450–6458, Honolulu, HI, USA, 2017.

[19] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: convolutional block attention module," in *European Conference on Computer Vision*, pp. 3–19, Munich, Germany, 2018.

[20] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803, Salt Lake City, 2018.

[21] Z. Huang, X. Wang, L. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross attention for semantic segmentation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 603–612, Seoul, Korea, 2019.

[22] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "Gcnet: non-local networks meet squeeze-excitation networks and beyond," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 1971–1980, Seoul, Korea, 2019.

[23] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng, "Aˆ2-nets: double attention networks," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, H. Cesa-Bianchi, and R. Garnett, Eds., vol. 31, pp. 352–361, Curran Associates, Inc., 2018.

[24] Z. Gao, J. Xie, Q. Wang, and P. Li, "Global second-order pooling convolutional networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3019–3028, Long Beach California, 2019.

[25] K. Zhang, M. Sun, T. X. Han, X. Yuan, L. Guo, and T. Liu, "Residual networks of residual networks: multilevel residual networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 6, pp. 1303–1314, 2018.

[26] J. Fu, J. Liu, H. Tian et al., "Dual attention network for scene segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3141–3149, Long Beach California, 2019.

[27] A. Krizhevsky and G. Hinton, *Learning multiple layers of features from tiny images*, Citeseer, 2009.

[28] O. Russakovsky, J. Deng, H. Su et al., "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[29] O. Vinyals, C. Blundell, T. Lillicrap, and D. Wierstra, "Matching networks for one shot learning," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[30] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5987–5995, Honolulu, HI, USA, 2017.

[31] H. Wang, Z. Wang, M. Du et al., "Score-cam: score-weighted visual explanations for convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020.