symmetry

MDPI

Article

The Expansion Methods of Inception and Its Application

Cuiping Shi [1,2,*], Zhenquan Liu [2,3], Jiageng Qu [2] and Yuxin Deng [2]

[1] College of Information Engineering, Huzhou University, Huzhou 313000, China
[2] College of Communication and Electronic Engineering, Qiqihar University, Qiqihar 161000, China; 2023222354@nwnu.edu.cn (Z.L.); 2019132066@qqhru.edu.cn (J.Q.); 2019132228@qqhru.edu.cn (Y.D.)
[3] College of Physics and Electronic Engineering, Northwest Normal University, Lanzhou 730070, China
* Correspondence: shicuiping@qqhru.edu.cn

Abstract: In recent years, with the rapid development of deep learning technology, a large number of excellent convolutional neural networks (CNNs) have been proposed, many of which are based on improvements to classical methods. Based on the Inception family of methods, depthwise separable convolution was applied to Xception to achieve lightweighting, and Inception-ResNet introduces residual connections to accelerate model convergence. However, existing improvements for the Inception module often neglect further enhancement of its receptive field, while increasing the receptive field of CNNs has been widely studied and proven to be effective in improving classification performance. Motivated by this fact, three effective expansion modules are proposed in this paper. The first expansion module, Inception expand (Inception-e) module, is proposed to improve the classification accuracy by concatenating more and deeper convolutional branches. To reduce the number of parameters for Inception e, this paper proposes a second expansion module—Equivalent Inception-e (Eception) module, which is equivalent to Inception-e in terms of feature extraction capability, but which suppresses the growth of the parameter quantity brought by the expansion by effectively reducing the redundant convolutional layers; on the basis of Eception, this paper proposes a third expansion module—Lightweight Eception (Lception) module, which crosses depthwise convolution with ordinary convolution to further effectively reduce the number of parameters. The three proposed modules have been validated on the Cifar10 dataset. The experimental results show that all these extensions are effective in improving the classification accuracy of the models, and the most significant effect is the Lception module, where Lception (rank = 4) on the Cifar10 dataset improves the accuracy by 1.5% compared to the baseline model (Inception module A) by using only 0.15 M more parameters.

Keywords: Inception module; expansion; lightweight; classification; CNN

check for updates

Citation: Shi, C.; Liu, Z.; Qu, J.; Deng, Y. The Expansion Methods of Inception and Its Application. *Symmetry* **2024**, *16*, 494. https://doi.org/10.3390/sym16040494

Academic Editor: Theodore E. Simos

Received: 15 March 2024
Revised: 10 April 2024
Accepted: 14 April 2024
Published: 18 April 2024

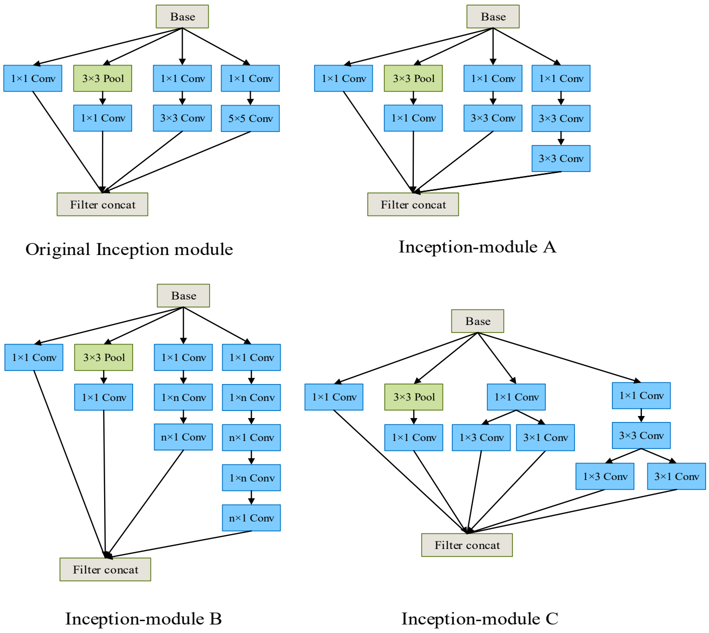
1. Introduction

Convolutional neural networks (CNNs) have experienced rapid development in the past decades. Currently, CNNs are widely used on many computer vision application tasks including facial expression recognition [1–4], Alzheimer's disease diagnosis [5], and so on. LeNet [6] marked the beginning of CNNs, which were early attempts but were limited in the computational resource. Over time, the emergence of AlexNet [7] pushed the breakthrough of CNNs, which realized the deep network by introducing the ReLU activation function and using two GPUs to share the computation. VGG [8] builds on this foundation by using more small-sized convolutional kernels to control the cost of the parameter amount and thus deepen the network in order to extract more representative global features. ResNet [9] introduces the residual connection to solve the gradient vanishing problem which makes the network deeper. All of the above models have no branching structure or the branches only act as residual connections, which makes them ineffective in extracting globally different features. Unlike them, the Inception family of networks consists of a series of modules with a multi-branch structure. By using convolution kernels of different sizes or using branches of different depths, the modules can extract features with varying global degrees.

In previous studies, researchers have mainly focused on building networks using classical Inception modules and fusing Inception modules with other methods, with relatively little exploration of network depth and width. Generally speaking, the deeper the network, the more global features can be extracted. The wider the network or the more branches it has, the richer the extracted features. Therefore, we would like to make a deep study in this area and explore how to further improve the Inception module to enhance its performance and its ability to extract features. Specifically, in this paper, we improve the performance of the Inception module by extending its depth and width. However, extending the depth and width of the module while incurring a huge parametric cost may suffer a high computational complexity in some tasks, which limits its application in resource-constrained environments, such as mobile devices or edge computing devices. Therefore, in this paper, the Inception module is optimized and improved, and the idea of lightweighting is also introduced to improve its computational efficiency. We propose three extension methods for progressive optimization. With these improvements, the performance of the Inception module is enhanced, which promotes its wide application in various tasks in resource-constrained environments.

1.1. The Models of the Inception Series

The first member of the Inception model family was the landmark GoogLeNet [10]. As a model proposed in the same year as VGG, it has deeper depth and achieved higher classification accuracy on the ImageNet dataset. As shown in Figure 1, the core module of GoogLeNet can be referred to as the Original Inception module, which uses pooling layers and multiple convolutional layers with different convolutional kernel sizes in parallel to obtain globally different features. C. Szegedy et al. [11] proposed to reduce the number of parameters by decomposing the convolutional layers in the original Inception module with larger convolutional kernel sizes into convolutional layers with smaller convolutional kernel sizes. This decomposition can be either symmetric or asymmetric, as in Figure 1; the decomposed structures are referred to in this paper as Inception module A (symmetric decomposition), Inception module B (asymmetric decomposition), and Inception module C (asymmetric decomposition). Another effective improvement of [11] is the application of the batch normalization [12] method. C. Szegedy et al. [13] further proposed Inception v4 and also proposed to accelerate the convergence of the model by introducing residual connections.



Original Inception module

Inception-module A

Inception-module B

Inception-module C

Figure 1. The modules of Inception.

In addition to the classical methods mentioned above, there are also some approaches to improve the Inception structure in recent years. X. Zhang et al. proposed a new module, Residuals Inception (RI) [14], in which each parallel branch of the RI module is replaced by three densely connected convolutional layers to the original structure, allowing the neural network to extract a richer set of features. M.Z. Alom et al. proposed an IRCNN [15], which combines CNNs and Recurrent Neural Networks (RNNs) to improve classification accuracy. L. Xie et al. [16] proposed the use of an optimized Inception module combined with a convolutional block attention module (CBAM) attention mechanism [17], and introduced residual connectivity to its structure to extract multi-scale features to improve classification accuracy. F. Chen et al. proposed a BeIn-v4 [18], which introduced the SKNet [19] attention mechanism to extract features of images more effectively to improve classification accuracy.

1.2. Lightweight CNNs

Lightweight CNNs significantly reduce the number of parameters while ensuring model accuracy. The main way to reduce the number of parameters is to use a convolutional approach that is as sparse as possible but does not affect the inter-feature information instead of the traditional approach with high-density connections. This idea is adopted in the mainstream depthwise separable convolution and grouped convolution, in which the grouped convolution groups the input tensor, then convolves it group-by-group and finally splices it to obtain the output. However, this method cannot realize communication among the convolution groups, which means it cannot extract the features of images effectively. To solve this problem, X. Zhang et al. proposed a tensor rearrangement method in ShuffleNet v1 [20], which realizes the communication between convolution groups by rearranging the output of grouped convolution. Another convolutional approach is the Depthwise Separable convolution proposed by F. Chollet et al. [21]. This method consists of two steps: 1. Extracting features channel-by-channel using depthwise convolution. 2. Implementing inter-feature communication using pointwise convolution.

There are also some works about the optimization of the structure of the lightweight network to improve the classification accuracy. GhostNet [22], proposed by K. Han et al., is a stack of some ghost modules in which each ghost module is generated by identity and depth-separable convolutional layers to generate the complete feature map. MobileNeXt [23], proposed by D. Zhou et al., is a replacement of the depthwise separable convolution in the MobileNet v2 [24] network with a sandglass module (which consists of 3 × 3 Depthwise Separable convolution, 1 × 1 convolution (squeeze), 1 × 1 convolution (recover), and 3 × 3 Depthwise Separable convolution) to improve the network performance. WeightNet [25], proposed by N. Ma et al., is a simple and efficient dynamic generative network, which applies the SENet [26] channel attention mechanism, i.e., the dynamic weight tensor is first obtained by global average pooling and a fully connected layer with sigmoid activation, and then the original tensor is weighted by the weight tensor. In contrast, EfficientNet [27], proposed by M. Tan, is improved by a neural network architecture search (NAS) in three aspects simultaneously—input resolution, network depth, and width—to improve the classification accuracy of the network model.

Some recent works on model lightweighting have been carried out mainly for the Vision Transformer (ViT) [28]. ViT has achieved impressive results in the field of computer vision. However, such models are often not better deployed on mobile devices due to their large model size and high latency; thus, a lightweight design for such models is necessary [29–35]. MobileViG [29] is the first hybrid CNN–GNN for vision tasks on mobile devices, which mainly proposes Sparse Visual Graph Attention (SVGA) for faster speed. FastViT [30] introduces a novel token mixing operator, RepMixer, which effectively reduces memory access costs. SwiftFormer [31] introduces a novel efficient additive attention mechanism that effectively replaces the quadratic matrix multiplication operations with linear element-wise multiplications.

In summary, researchers have made some improvements to Inception in terms of convolution kernel size, convolution method, and inter-stage connectivity, but no research has been conducted from the perspective of expanding the depth and width of the Inception module. In this paper, we first extend the depth and width of the Inception module from different ways and show that the proposed method can provide excellent classification performance. However, simply extending the depth and width of the module would involve a large number of parameters. Therefore, we further improve the structure and incorporate a lightweight network to propose a number of feasible extension methods. The main contributions of this work are:

1. A basic extension method, the Inception-e module, is proposed by us. Based on the Inception module A, firstly, the basic expansion method is raised, and the experimental results prove that increasing the depth and width of the Inception module A is beneficial to the improvement of the classification accuracy of the model, but the method is accompanied by a huge number of parameters.
2. To solve the problem of increasing the number of parameters due to extension, an equivalent extension method, the Eception module, is proposed by us, which has comparable perceptual field and feature extraction abilities to Inception-e. The Eception module improves the classification accuracy of the model while saving on the number of parameters.
3. A lightweight expansion method, Lception module, is proposed. On the basis of Eception module, inspired by the idea of a lightweight convolutional neural network, by cross-replacing the ordinary convolutional layers of the Eception module with depthwise convolutional layers, the weights of these layers are sparser, and thus reduce the number of parameters. The experimental results show that the Lception module can effectively improve the classification accuracy of the network with almost the same number of parameters.

The remaining part of this article is arranged as follows. In Section 2, the structures of Inception-e, Eception, and Lception are described in detail. In Section 3, the main focus is on the experiments and analyses we conducted, including the datasets used for the experiments, the validation of the three extended methods, the Grad-CAM visualization analysis, and the comparison with some other methods. The conclusions are provided in Section 4.

2. Methods

2.1. Basic Expansion Method—Inception-e

Paralleling more and deeper convolution branches on the basis of Inception module A can improve its ability to extract features, so that the model can provide higher classification accuracy. The extended structure is named Inception-e module.

The original Inception module uses convolutional layers with different kernel sizes to extract global features and fuses them, where the larger the kernel size is, the more global (more abstract) features are extracted. The convolutional layers effectively save parameters without reducing the structure's ability to extract features. We believe that concatenating convolutional branches with increasing depth steps on top of Inception-module A enables the structure to extract globally richer features. As shown in Figure 2, different depths of convolutional branches have different effects on feature extraction, with deeper branches capturing more global features and shallower branches extracting more detailed features.

As shown in Figure 2, parallel concatenating more branches with deeper depths allows the model to extract more representative and richer features. Specifically, shallower convolutional branches extract more detailed features such as facial texture, while deeper branches extract features that are more representative of facial features such as facial contours.

As shown in Figure 3, our proposed method only expands the core structure of Inception module A, while the other parts remain unchanged. The Inception-e method is to progressively concatenate deeper convolutional branches on top of the core structure. The nth rank expansion structure is named as Inception-e module (rank = n). Although this method improves the feature extraction ability of the structure and enables the model to obtain higher classification accuracy, it is accompanied by a huge parameter cost, and the number of parameters rises rapidly with the increase of rank.

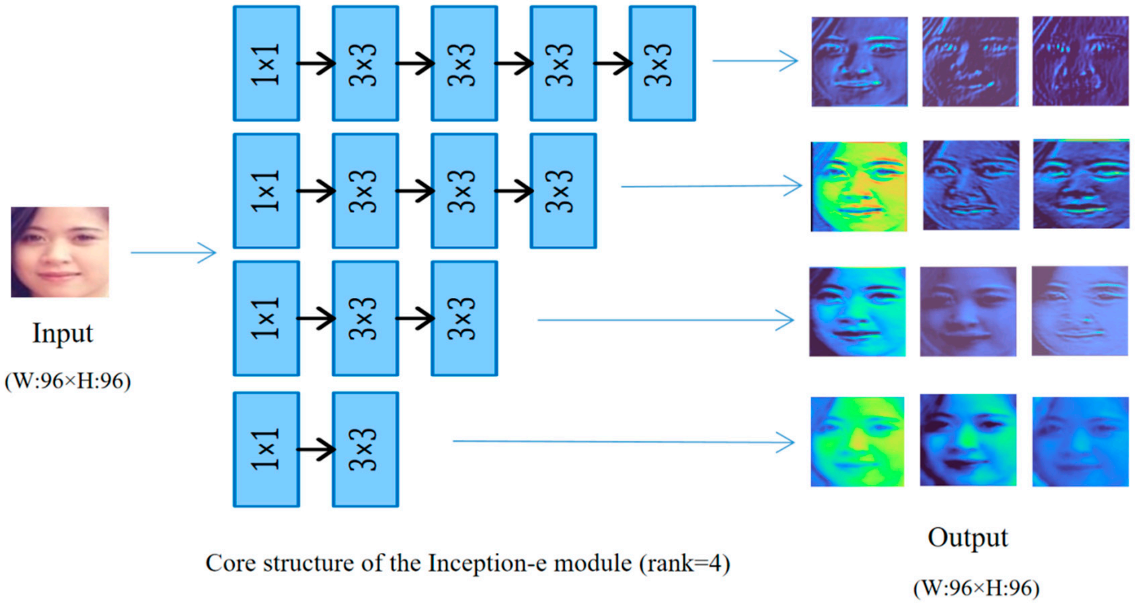


Figure 2. The relationship between the depth of the branch and the extracted features.

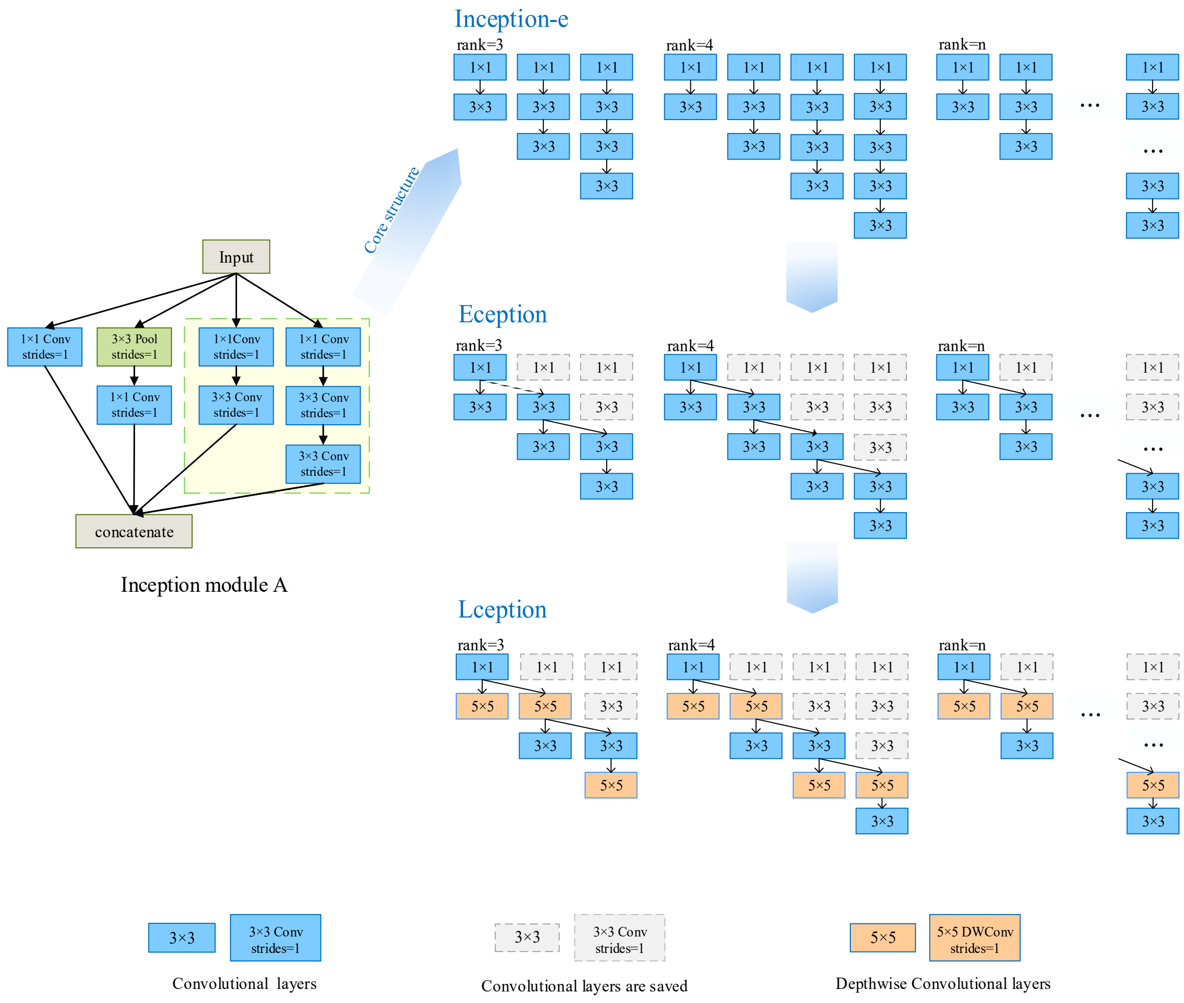


Figure 3. The schematic diagram of the Inception-e module, the Eception module, and the Lception module. The structure within the dashed box is the core structure of Inception module A.

2.2. Equivalent Expansion Method—Eception

In order to reduce the number of parameters brought by the extension, the Eception module is proposed in this paper. This structure is equivalent to the effect of Inception-e module for feature extraction, but saves a large number of parameters.

As shown in Figure 3, unlike Inception-e module, which takes the output tensor of the previous convolutional layer of the branches as input, the Eception module only keeps the last two convolutional layers of the branches and discards the rest of the convolutional layers, while the penultimate convolutional layer takes the output tensor of the previous convolutional layer of the adjacent branches as input.

This method effectively improves the efficiency of the convolutional layers and reduces the number of redundant convolutional layers, thus suppressing the spike in the number of parameters caused by the expansion. When the size of the output tensor of the core structure is $W \times H \times (C \times \text{rank})$, the number of 3 × 3 convolutional kernels of the Incption-e module is N_I, the number of 3 × 3 convolutional kernels of the Eception module is N_E, and the comparison of the relationship between the number of 3 × 3 convolutional kernels and rank of the two modules is shown in Table 1.

Table 1. Comparison of the relationship between the number of 3 × 3 convolutional kernels and the rank of the Inception-e module and the Eception module.

Rank	$\boldsymbol{N_I}$	$\boldsymbol{N_E}$	$\boldsymbol{\frac{N_E}{N_I}}$
2	$3C^2$	$3C^2$	1.00
3	$6C^2$	$5C^2$	0.83
4	$10C^2$	$7C^2$	0.70
5	$15C^2$	$9C^2$	0.60
…	…		…
n	$(\frac{n^2+n}{2})C^2$	$(2n-1)C^2$	$\frac{4n-2}{n^2+n}$

Obviously, when the rank is ≥3, the Eception module can effectively save the number of parameters compared to the Inception-e module, and the higher the rank, the larger the saving ratio.

When reducing the number of parameters, the change in the structure of the Eception module compared with the Inception-e module does not affect the feature extraction ability of the module. As shown in Figure 4, the receptive field of the input features are the same for both. When the input tensor is X_{input}, the output tensor X_{output} and $X^{'}{}_{output}$ of the core structure of the Inception-e module and the Eception module can be represented as

$$X_{output} = F_{rf=3}(X_{input})//F_{rf=5}(X_{input})//F_{rf=7}(X_{input}) \tag{1}$$

$$X'_{output} = F'_{rf=3}(X_{input})//F'_{rf=5}(X_{input})//F'_{rf=7}(X_{input}) \tag{2}$$

where F is the convolution operation, rf is the equivalent receptive field size for that operation, and // is the connection tensor. The receptive field can be calculated as

$$rf_l = rf_{l-1} + (k_l - 1) \times \mathrm{s}_\mathrm{l} \tag{3}$$

where rf_l denotes the receptive field size of the convolution at layer l, rf_{l-1} denotes the receptive field size of the convolution at layer $l-1$, k_l denotes the size of the convolution kernel at layer l (assuming the convolution kernel is symmetric), and s_l denotes the convolution step size at layer l.

Obviously, multiple convolutions with a small convolution kernel can convolve a receptive field equal to that of one convolution with a large-sized convolution kernel: e.g., $F_{s=7}(.)$ means that the operation is equivalent to the receptive field of a 7 × 7 convolutional layer. Clearly, the equivalent receptive field sizes of the convolutions experienced by the components of X_{output} and $X^{'}{}_{output}$ are the same, so the ability of the Eception module to extract features is approximately equivalent to that of the Inception-e module. The later experiments confirm the idea that the classification accuracy of the Eception module is very close to that of the Inception-e module.

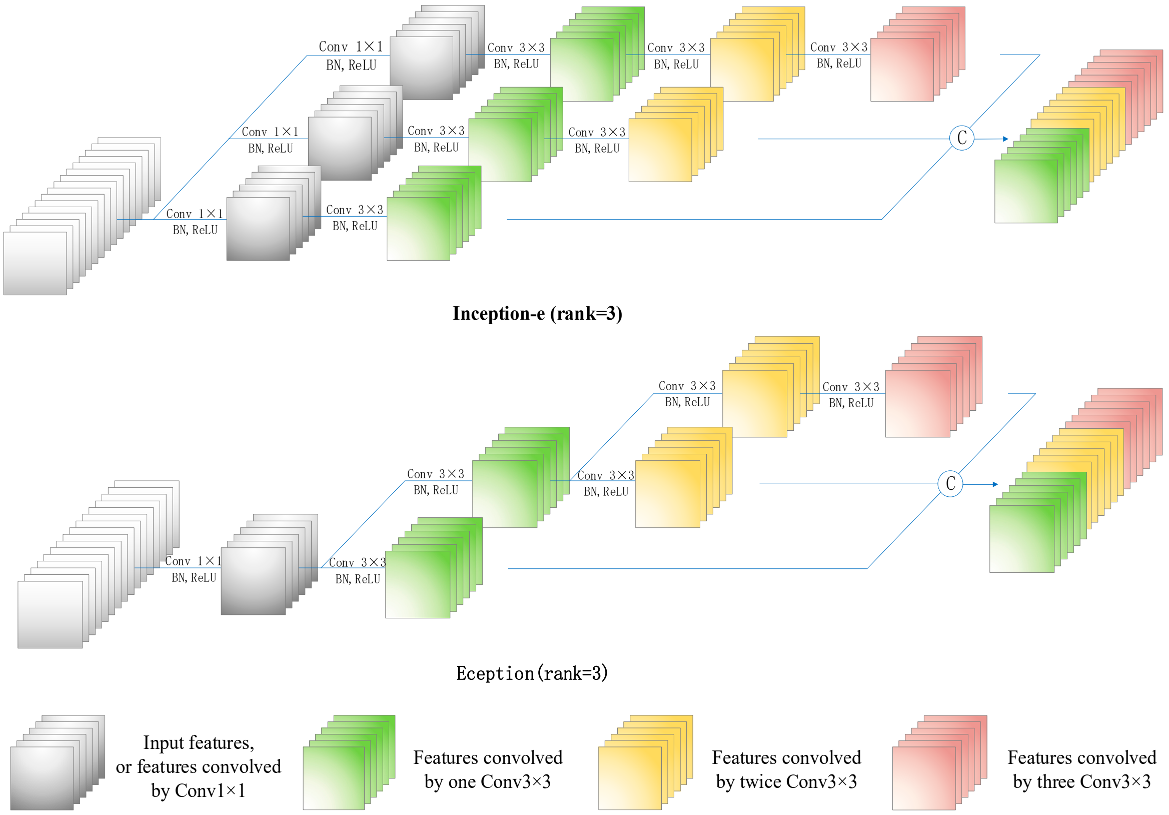

Conv 1×1
BN, ReLU
Conv 3×3
BN, ReLU
C
Inception-e (rank=3)
Eception(rank=3)
Input features, or features convolved by Conv1×1
Features convolved by one Conv3×3
Features convolved by twice Conv3×3
Features convolved by three Conv3×3

Figure 4. The overall comparison of features extracted by the Eception module and the Inception-e module.

2.3. *Lightweight Expansion Method—Lception*

Cross-replacing the ordinary convolution layers in the Eception module with depthwise convolution layers can further reduce the number of parameters of the structure, which is named the Lception module.

As in Figure 3, to obtain the Lception module, the ordinary convolutional layers of the Eception module are cross-replaced as depthwise convolutional layers with activation function h-swish [36] and convolutional kernels of size 5 × 5. The use of depthwise convolution allows the structure to further reduce the number of parameters on top of the Eception module. The introduction of h-swish instead of RELU as the activation function of the depthwise convolution layers can improve the model performance by effectively avoiding the neuron death phenomenon. In addition, since the weights of the depthwise convolutional layers are very sparse, using a larger convolutional kernel (such as a convolutional kernel of size 5 × 5) can effectively expand the perceptual field of the Lception module with only a very small parameter cost.

As shown in Figure 5, depthwise convolution, as a very sparse convolution method, must implement effective inter-channel feature communication to enable it to extract features effectively. Depthwise convolution achieves inter-channel feature communication through pointwise convolutional layers, while our approach uses depthwise convolutional layers crossed with ordinary convolutional layers to achieve inter-channel feature communication as well. Moreover, since we retain the ordinary convolutional layers with 5 × 5 convolutional kernels, this structure has a larger perceptual field than the depthwise convolution, and can extract more abstract and useful features for classification.

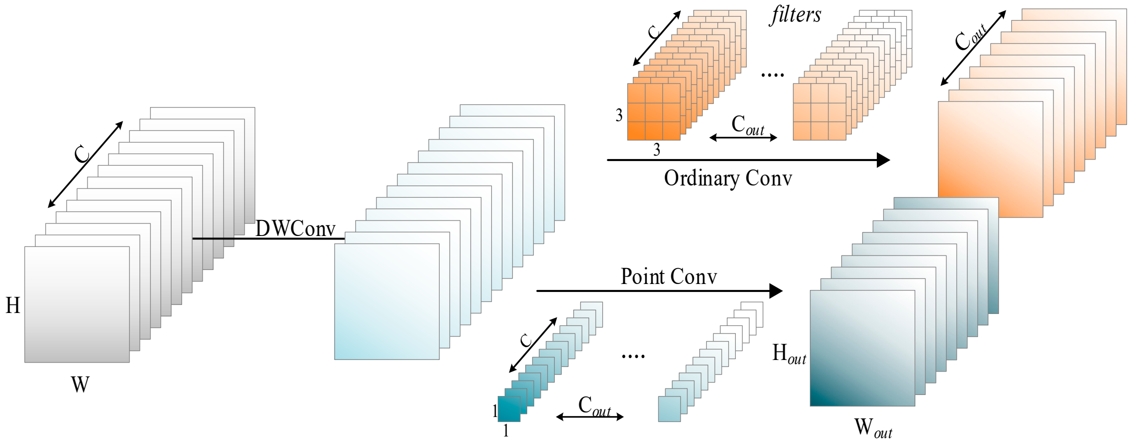


Figure 5. Comparison of the depthwise separable convolution with the proposed method.

3. Experiments and Analysis

3.1. Experimental Datasets

Four datasets are used for our experiments, including the benchmark dataset Cifar10, and three facial expression classification datasets RAF-DB [37], FER2013 [38], and FERplus (FER+) [39].

To verify the performance of the proposed model, the benchmark dataset Cifar10 is mainly used for the experiments. In the Cifar10 dataset, the training set includes 50,000 images and the test set includes 10,000 images, totaling 60,000 32 × 32 color images in 10 classes. To facilitate the experiments, both the training set and the test set images are enlarged to 96 × 96 size and normalized, and data enhancement is done only for the training set.

The FER2013 dataset consists of 35,887 grey scale images of size 48 × 48 in seven categories, including: 1—Surprise, 2—Fear, 3—Disgust, 4—Happiness, 5—Anger, 6—Neutral, 7—Sadness. It includes 28,709 images in the training set (set to be the training set), 3589 images in the public test set (set to be the validation set), and 3589 images in the private test set (set to be the test set). We normalize all images and only do data augmentation on the training set.

The FER+ dataset is relabeled from FER2013. When using the majority mode, it includes 25,045 images in the training set, 3191 images in the validation set, and 3137 images in the test set, totaling 31,373 48 × 48 grayscale images in eight categories. The FER+ set has one more category than FER2013 and RAD-DB set: 8—Contempt. We normalize all images and only do data augmentation on the training set.

The RAF-DB dataset, including single-label and double-label, totals 29,672 images. These are used in this paper, including 12,271 images in the training set and 3069 images in the test set, totaling 15,340 100 × 100 color images in seven categories. To facilitate the experiment, both the training and test sets were cropped to 96 × 96 size and normalized, and only the training set was data enhanced.

In Figure 6, some images of the FER+ dataset and RAF-DB dataset are given.

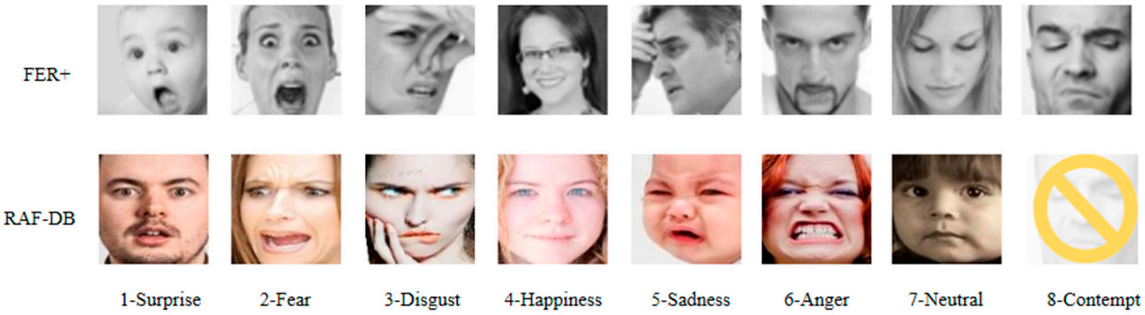


Figure 6. Some examples of the FER+ dataset and RAF-DB dataset.

3.2. The Network Models and Experimental Conditions

Table 2 shows the network models for the experiments using the Cifar10 dataset. On the classifier, the global pooling flattening tensor is firstly utilized, then Dropout regularization [40] is performed to discard 20% of the features, and finally, the classification is achieved using a full connection layer with an activation function of Softmax. We consider that due to the different sizes of input images, the number of modules used in the models of other datasets will also be adjusted accordingly, and other settings are similar to those shown in Table 1.

Table 2. The network model structures when using the Cifar10 dataset.

Experimental Structure in Cifar10			**Output Size**
	Input		96 × 96
Inception-e module channel = 32	Eception module channel = 32	Lception module channel = 32	96 × 96
	1/2 MaxPool		48 × 48
Inception-e module channel = 64	Eception module channel = 64	Lception module channel = 64	48 × 48
	1/2 MaxPool		24 × 24
Inception-e module channel = 96	Eception module channel = 96	Lception module channel = 96	24 × 24
	1/2 MaxPool		12 × 12
Inception-e module channel = 128	Eception module channel = 128	Lception module channel = 128	12 × 12
	1/2 MaxPool		6 × 6
Inception-e module channel = 160	Eception module channel = 160	Lception module channel = 160	6 × 6
	1/2 MaxPool		3 × 3
	Global Average Pool		1 × 1
	Dropout (0.2)		
	Fully-concatenate, softmax		10

For experimental conditions, on the Cifar10 dataset:

1. Optimizer: SGD [41]
2. Training batch: 32
3. Momentum: 0.9
4. Regularization: L2 regularization of the weights of all convolutional layers.

On the RAF-DB, FER+, FER2013 datasets:

No momentum acceleration is used. Considering the small dataset, the training epochs are extended. The rest of the conditions are the same as above.

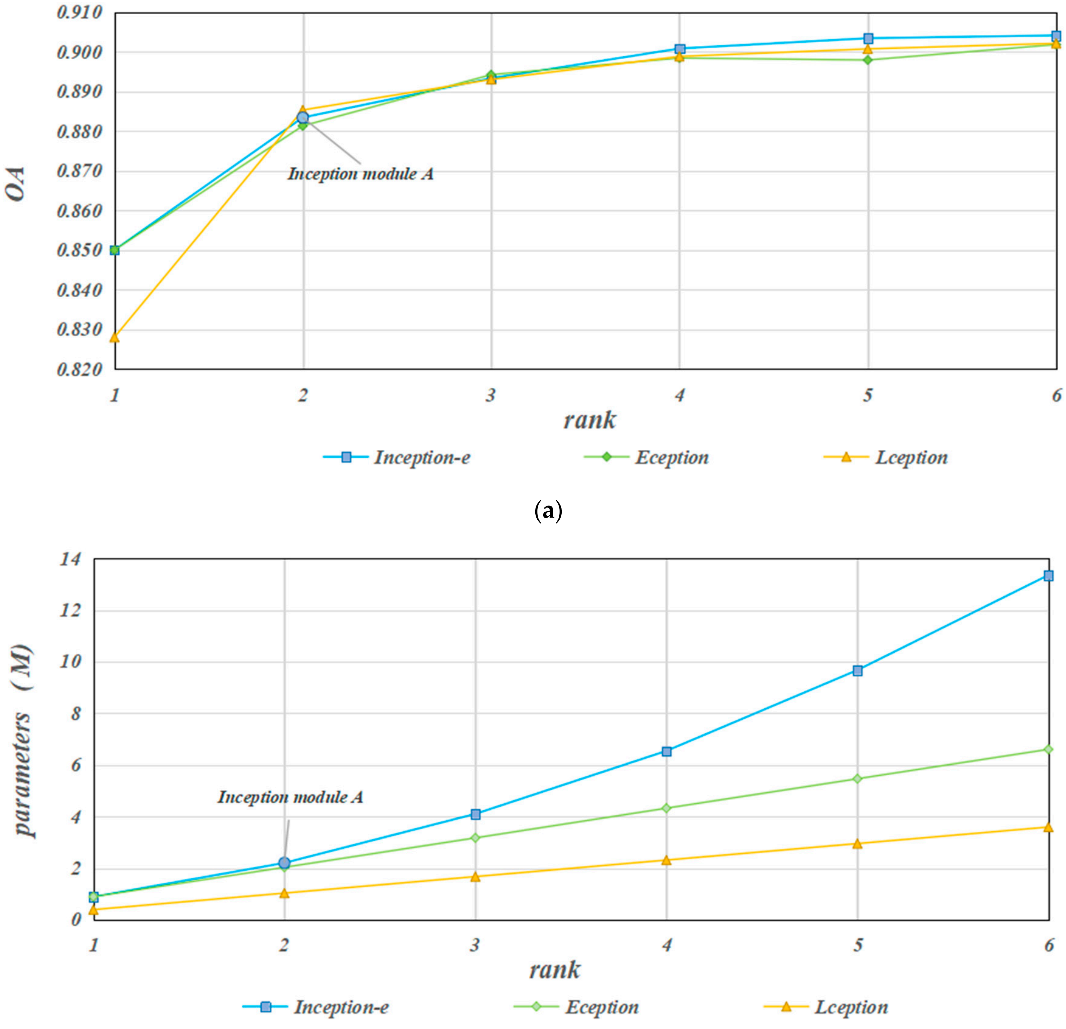
3.3. Validation of the Three Expansion Methods

Some experiments were conducted to validate the proposed three expansion methods. Overall Accuracy (OA) is adopted as a measure of the classification accuracy of the model, and OA can be expressed as:

$$OA = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

TP denotes the number of positive classes predicted as positive, FN denotes the number of positive classes predicted as negative, FP denotes the number of negative classes predicted as positive, and TN denotes the number of negative classes predicted as negative.

In this section, the Cifar10 dataset is used for the experiments. Figure 7 shows the experimental results of the classification accuracy, number of parameters, and time complexity of the proposed three methods under each rank.



(**a**)

(**b**)

Figure 7. Comparison of experimental results of the Inception-e module, the Eception module, and the Lception module. (**a**) Comparison of the classification accuracy of the three methods; (**b**) comparison of the parameter quantities of the three methods.

For the Inception-e module, as seen in Figure 7a, parallel concatenating more and deeper convolutional branches can effectively improve the classification accuracy of the model. Inception-e module (rank = 6) has the highest classification accuracy of 90.4%, which is a 2.1% improvement over Inception module A. As seen in Figure 7b, the classification accuracy can be effectively improved by the Inception-e, but this expansion is accompanied by a huge number of parameters. The larger the rank, the faster the number of parameters rises, even up to 13.4 M when rank = 6.

For the Eception module, as seen in Figure 7a, the accuracy of the Eception module is close to that of the Inception-e module, which indicating that the two modules are equivalent in their ability to extract features. At rank = 6, the Eception module achieves the highest classification accuracy of 90.2%, which is 1.8% higher than that of Inception module A and only 0.2% lower than that of Inception-e module (rank = 6). As seen in Figure 7b, the Eception module effectively reduces the number of parameters that added to the expansion compared to the Inception-e module. For example, the Eception module (rank = 6) has 6.6 M parameters, which is 49% less than that of the Inception-e module (rank = 6) of the same rank.

For the Lception module, as shown in Figure 7a, the classification accuracy of the Lception module approximates that of the other two methods, and the classification accuracy of the Lception module (rank = 6) reaches 90.2%, which is 1.9% higher than Inception module A, only 0.2% lower than that of the Inception-e module (rank = 6), and basically similar to that of the Eception module (rank = 6). The number of parameters of the Lception module (rank = 6) is 3.6 M, which is only 27% of Inception-e (rank = 6) and 55% of Eception (rank = 6). The classification accuracy of Lception (rank = 4) is 1.5% higher than that of Inception module A and pays only 0.15 M more parameters.

In summary, compared with the classical Inception module A, the proposed Inception-e, Eception, and Lception modules all have significant advantages in classification accuracy. By using more and deeper branches, the Inception-e module effectively improves the ability to extract features. The Eception module effectively reduces the number of parameters and achieves higher classification accuracy by improving the structure, while the Lception module greatly reduces the number of parameters and achieves a lighter network by replacing the traditional convolutional crossover in the Eception module with depthwise convolution. In addition, the Lception module makes more effective use of intra-regional correlation by increasing the perceptual field of convolutional kernels on the image, and obtains more discriminative image features, thus overcoming the degradation of classification performance caused by the reduction of the number of convolutional kernels due to the replacement of the depthwise convolution layer.

For the proposed three model structures, some experiments are conducted on the Cifar10 dataset. The experimental results show that, as an extended structure of Inception module A, all three methods effectively improve the classification performance compared to Inception module A. The proposed Lception module can obtain the classification accuracy that approximates that of the Inception-e module and the Eception module with the minimum number of parameters, which fully demonstrates the effectiveness of the methods.

3.4. Grad-CAM Visual Analysis

In order to more intuitively illustrate the impact of different expansion methods on network performance, some images of different classes are selected in RAF-DB and the Grad-CAM [42] method is utilized to visualize and analyze the four structures of Inception module A, the Inception-e module (rank = 6), the Eception module (rank = 6), and the Lception module (rank = 6). The experimental results are shown in Figure 8.

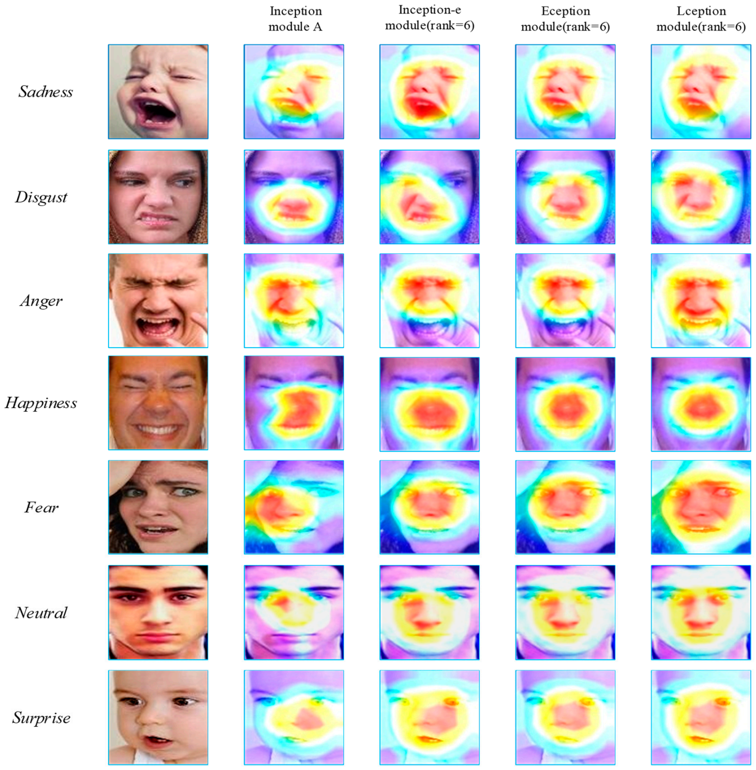


Figure 8. Heat maps of different categories of images in the RAF-DB dataset.

The color of the heat map reflects how much the network pays attention to the region, and the darker the color indicates that the neural network pays more attention to the region. As shown from these visualization results, all three of our proposed expansion modules focus on a larger range and more features than Inception module A. This further demonstrates the effectiveness of the proposed expansion method. This is because with more and deeper branching structures, the module is able to capture more global features while still retaining the ability to extract fine-grained features.

3.5. Comparison with More Methods

To further verify the performance of the proposed modules, we selected Lception module (rank = 4), Eception module (rank = 4), and Eception module (rank = 8) to compare with some classical networks, lightweight networks, and some facial expression classification methods, and the comparison results are shown in Table 3.

Table 3. Comparison of experimental results (using FER+ dataset).

	FER+	
	Parameters	**OA**
Lception module (rank = 4) ours	**1.3 M**	**86.9**
Eception module (rank = 4) ours	**2.3 M**	**87.3**
Eception module (rank = 8) ours	**4.8 M**	**87.6**
Original Inception module *	1.4 M	85.2
Inception module A *	1.2 M	86.3
VGG-11-GAP *	2.4 M	86.5
VGG-13-GAP *	3.5 M	85.0
VGG-13 [43]	9.4 M	84.4
VGG-19 [43]	20.0 M	84.4
ResNet18 [43]	11.2 M	84.8
MobilNet v1 [43]	1.1 M	83.5
MobilNet v2 [43]	2.2 M	81.4
ShuffleNet v1 [43]	0.9 M	84.1
ShuffleNet v2 [43]	1.3 M	80.4
ESRs [44]	-	87.2
SHCNN [45]	-	86.5
LER [43]	-	85.7
TFE-JL [46]	-	84.3

The * in Table 3 indicates that the structure was performed under the same experimental environment as the method proposed in this paper. VGG-GAP refers to the use of global average pooling instead of flatten operation to spread the tensor on top of classical VGG and uses only one layer of fully connected layer for classification. Some results are also cited from the literature [43] for a number of networks when using the FER+ dataset.

As shown in Table 3, the overall accuracy of Lception module (rank = 4) is 86.9, Eception module (rank = 4) is 87.3%, and Eception module (rank = 8) is 87.6% on the FER+ dataset. There are different degrees of improvement compared with original Inception module and Inception module A. Compared with other classical structures VGG-11-GAP, VGG-13-GAP, VGG-13, VGG-19, ResNet18, Lception module (rank = 4), Eception module (rank = 6) uses a smaller number of parameters to obtain higher classification accuracy. For example, on the FER+ dataset, the number of parameters of Lception module (rank = 4) is only 6.5% of that of VGG-19, but the accuracy is 2.5% higher than that of VGG-19, which is a significant advantage. Compared with the lightweight networks Mobilenet v1 [47] and Mobilenet v2, and Shufflenet v1 and Shufflenet v2 [48], the accuracy of the proposed Lception module (rank = 4) is increased by 2.8%~6.5% in the FER+ set with a slightly increased number of parameters.

The proposed Eception module and Lception module also show advantages over other neural network-based methods for facial expression classification. SHCNN [45] is a shallower neural network and the model achieves 86.5% accuracy on the FER+ dataset. The core method of [44] consists of ensembles with shared representations (ESRs) based on convolutional networks. The classification accuracy of the ESR-9 achieves 87.15% on the FER+ dataset. In [43], a lightweight emotion recognition (LER) model was proposed that combines densely connected convolutional layers and model compression techniques into a framework that eliminates redundant parameters, obtaining an accuracy of 85.67% on the FER+ dataset. The method in [46] obtained a classification accuracy of 84.29% on the FER+ dataset. The listed Eception (rank = 6) and Lception (rank = 4) can obtain a higher classification accuracy compared to the above methods.

As shown in Table 4, we also conducted some experiments using the FER2013 dataset. The literature [49] uses transfer learning to classify facial expressions. By fine-tuning classical convolutional neural networks, the feature extraction capability of large classical networks can be effectively utilized. The literature [50] proposes to judge the reliability of the current classification result using a multi-layer perceptron (MLP) classifier. If the result is unreliable, the given face image is used as a query condition to search for similar images. Then, another MLP is trained to predict the final emotion category by summarizing the classification output vectors of the query image and its retrieved similar images. The literature [51] presents a generic convolutional neural network model for real-time CNNs. The literature [52] proposes several differently structured subnets. These subnets are compact CNN models trained individually. The whole network is composed by assembling these subnets together. Compared to these models, our proposed method can obtain higher accuracy on the FER2013 dataset.

Table 4. Comparison of experimental results (using FER2013 dataset).

FER2013	
	OA
Lception module (rank = 4) ours	**68.4**
Eception module (rank = 4) ours	**68.5**
Eception module (rank = 8) ours	**67.9**
Alexnet [49]	66.7
GoogLenet [50]	64.6
GoogLenet + MLP [50]	65.8
mini-Xception [51]	66.0
Subnet3 [52]	62.4
Subnet Ensemble [52]	65.0

In summary, the comparison with Inception module A demonstrates that the proposed extension methods are effective in improving the model accuracy by paying only a smaller number of parameters. Compared with some classical structures, lightweight structures, and other mainstream methods, the proposed Eception module and Lception module can obtain higher classification accuracy. This is because the proposed modules parallelize more branches, allowing them to extract richer and more abstract features. However, the shape of the convolutional kernels used in these modules are symmetric, which makes the number of parameters in the modules tend to be larger than those using asymmetric convolutional kernels. This issue will be addressed in our next research work. The paper involves relevant code that will be uploaded at https://github.com/LIUZHENQUANS/EMI (accessed on 14 March 2024).

4. Conclusions

In this paper, we investigate the extension methods of the Inception module and propose a new idea of extending the network structure. By carefully designing the network structure, it can provide the improvement of classification performance with a large margin than that of the original network, and with fewer parameters.

Specifically, we first propose the Inception-e module, which improves the classification accuracy by concatenating more and deeper convolutional branches, and then propose the Eception module to solve the problem of excessive parameters due to the increase of depth and width. Then, the Lception module is designed based on the Eception module by cross-replacing the ordinary convolution in the Eception module with depthwise convolution. The experimental results show that the extended network structure can effectively improve the classification accuracy as well as reduce the number of network parameters. We also found that the convolution kernels used in the proposed methods are symmetric. Considering that models using asymmetric convolution kernels can obtain larger receptive field with the same number of parameters, further application of asymmetric convolution kernels based on the proposed method may be beneficial to improve the model performance, which is a possible direction for future related work.

It is worth noting that the method proposed in this paper has strong generalization ability. It can be applied not only to the Inception network, but also other similar classical network structures. In the future, we will continue to explore more effective network structures to improve the classification performance of networks.

Author Contributions: Conceptualization, C.S.; data curation, C.S., Z.L. and J.Q.; formal analysis, Y.D.; methodology, C.S.; software, Z.L.; validation, C.S., Z.L. and J.Q.; writing—original draft, J.Q.; writing—review and editing, C.S. and Y.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Natural Science Foundation of China (42271409), in part by the Heilongjiang Province Higher Education Teaching Reform Research Project (Grant number: SJGY20220403), and in part by the Education Science Research Project of Qiqihar University (Project Number: GJQTYB202216).

Data Availability Statement: The code of this study is openly available in [https://github.com/LIUZHENQUANS/EMI] (accessed on 14 March 2024).

Acknowledgments: We would like to thank the handling editor and the anonymous reviewers for their careful reading and helpful remarks.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Meena, G.; Mohbey, K.K.; Indian, A.; Khan, M.Z.; Kumar, S. Identifying emotions from facial expressions using a deep convolutional neural network-based approach. *Multimed. Tools Appl.* **2024**, *83*, 15711–15732. [CrossRef]
2. Febrian, R.; Halim, B.M.; Christina, M.; Ramdhan, D.; Chowanda, A. Facial expression recognition using bidirectional LSTM-CNN. *Procedia Comput. Sci.* **2023**, *216*, 39–47. [CrossRef]
3. Sajjad, M.; Ullah, F.U.; Ullah, M.; Christodoulou, G.; Cheikh, F.A.; Hijji, M.; Muhammad, K.; Rodrigues, J.J. A comprehensive survey on deep facial expression recognition: Challenges, applications, and future guidelines. *Alex. Eng. J.* **2023**, *68*, 817–840. [CrossRef]
4. Adyapady, R.R.; Annappa, B. A comprehensive review of facial expression recognition techniques. *Multimed. Syst.* **2023**, *29*, 73–103. [CrossRef]
5. Fouladi, S.; Safaei, A.A.; Mammone, N.; Ghaderi, F.; Ebadi, M.J. Efficient deep neural networks for classification of Alzheimer's disease and mild cognitive impairment from scalp EEG recordings. *Cogn. Comput.* **2022**, *14*, 1247–1268. [CrossRef]
6. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324.
7. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
8. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2015**, arXiv:1409.1556. [CrossRef]
9. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
10. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9. [CrossRef]
11. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2818–2826. [CrossRef]
12. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167. [CrossRef]

13. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, New York, NY, USA, 4–9 February 2017; pp. 4278–4284.
14. Zhang, X.; Huang, S.; Zhang, X.; Wang, W.; Wang, Q.; Yang, D. Residual Inception: A New Module Combining Modified Residual with Inception to Improve Network Performance. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 3039–3043. [CrossRef]
15. Alom, M.Z.; Hasan, M.; Yakopcic, C.; Taha, T.M. Inception recurrent convolutional neural network for object recognition. *arXiv* **2017**, arXiv:1704.07709. [CrossRef]
16. Xie, L.; Huang, C. A Residual Network of Water Scene Recognition Based on Optimized Inception Module and Convolutional Block Attention Module. In Proceedings of the 2019 6th International Conference on Systems and Informatics (ICSAI), Shanghai, China, 2–4 November 2019; pp. 1174–1178. [CrossRef]
17. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV 2018), Munich, Germany, 8–14 September 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 3–19.
18. Chen, F.; Wei, J.; Xue, B.; Zhang, M. Feature fusion and kernel selective in Inception-v4 network. *Appl. Soft Comput.* **2022**, *119*, 108582. [CrossRef]
19. Li, X.; Wang, W.; Hu, X.; Yang, J. Selective kernel networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 510–519.
20. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 6848–6856. [CrossRef]
21. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1800–1807. [CrossRef]
22. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. GhostNet: More Features From Cheap Operations. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 1577–1586. [CrossRef]
23. Zhou, D.; Hou, Q.; Chen, Y.; Feng, J.; Yan, S. Rethinking bottleneck structure for efficient mobile network design. In Proceedings of the European Conference Computer Vision (ECCV 2020), Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 680–697.
24. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted residuals and linear bottlenecks. *arXiv* **2019**, arXiv:1801.04381. [CrossRef]
25. Ma, N.; Zhang, X.; Huang, J.; Sun, J. Weightnet: Revisiting the design space of weight networks. In Proceedings of the European Conference on Computer Vision (ECCV 2020), Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 776–792.
26. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141. [CrossRef]
27. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the 36th International Conference on Machine Learning, PMLR, Los Angeles, CA, USA, 9–15 June 2019; pp. 6105–6114.
28. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16 × 16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
29. Munir, M.; Avery, W.; Marculescu, R. Mobilevig: Graph-based sparse attention for mobile vision applications. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 2211–2219.
30. Vasu, P.K.; Gabriel, J.; Zhu, J.; Tuzel, O.; Ranjan, A. FastViT: A fast hybrid vision transformer using structural reparameterization. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 5785–5795.
31. Shaker, A.; Maaz, M.; Rasheed, H.; Khan, S.; Yang, M.H.; Khan, F.S. SwiftFormer: Efficient additive attention for transformer-based real-time mobile vision applications. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 17425–17436.
32. Li, Y.; Yuan, G.; Wen, Y.; Hu, J.; Evangelidis, G.; Tulyakov, S.; Wang, Y.; Ren, J. Efficientformer: Vision transformers at mobilenet speed. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 12934–12949.
33. Li, Y.; Hu, J.; Wen, Y.; Evangelidis, G.; Salahi, K.; Wang, Y.; Tulyakov, S.; Ren, J. Rethinking vision transformers for mobilenet size and speed. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 16889–16900.
34. Yu, W.; Luo, M.; Zhou, P.; Si, C.; Zhou, Y.; Wang, X.; Feng, J.; Yan, S. Metaformer is actually what you need for vision. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–20 June 2022; pp. 10819–10829.
35. Wang, A.; Chen, H.; Lin, Z.; Pu, H.; Ding, G. Repvit: Revisiting mobile cnn from vit perspective. *arXiv* **2023**, arXiv:2307.09283.
36. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, South Korea, 27 October–2 November 2019; pp. 1314–1324.

37. Li, S.; Deng, W. Reliable crowdsourcing and deep locality-preserving learning for unconstrained facial expression recognition. *IEEE Trans. Image Process.* **2019**, *28*, 356–370. [CrossRef] [PubMed]
38. Goodfellow, I.J.; Erhan, D.; Carrier, P.L.; Courville, A.; Mirza, M.; Hamner, B.; Cukierski, W.; Tang, Y.; Thaler, D.; Lee, D.-H.; et al. Challenges in representation learning: A report on three machine learning contests. In Proceedings of the International Conference on Neural Information Processing, Daegu, Korea, 3–7 November 2013; pp. 117–124.
39. Barsoum, E.; Zhang, C.; Ferrer, C.C.; Zhang, Z. Training deep networks for facial expression recognition with crowd-sourced label distribution. In Proceedings of the 18th ACM International Conference on Multimodal Interaction, Tokyo, Japan, 25–29 October 2016; pp. 279–283. [CrossRef]
40. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
41. Bottou, L. Stochastic gradient descent tricks. In *Proceedings of the Neural Networks: Tricks of the Trade*; Montavon, G., Orr, G.B., Müller, K.R., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 421–436.
42. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 618–626. [CrossRef]
43. Zhao, G.; Yang, H.; Yu, M. Expression recognition method based on a lightweight convolutional neural network. *IEEE Access* **2020**, *8*, 38528–38537. [CrossRef]
44. Siqueira, H.; Magg, S.; Wermter, S. Efficient facial feature learning with wide ensemble-based convolutional neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 3 April 2020; No. 4. Volume 34, pp. 5800–5809. [CrossRef]
45. Miao, S.; Xu, H.; Han, Z.; Zhu, Y. Recognizing facial expressions using a shallow convolutional neural network. *IEEE Access* **2019**, *7*, 78000–78011. [CrossRef]
46. Li, M.; Xu, H.; Huang, X.; Song, Z.; Li, X.; Li, X. Facial expression recognition with identity and emotion joint learning. *IEEE Trans. Affect. Comput.* **2018**, *12*, 544–550. [CrossRef]
47. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861. [CrossRef]
48. Ma, N.; Zhang, X.; Zheng, H.-T.; Sun, J. ShuffleNet V2: Practical guidelines for efficient cnn architecture design. In *Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; Volume 11218, pp. 122–138. [CrossRef]
49. Shengtao, G.; Chao, X.; Bo, F. Facial expression recognition based on global and local feature fusion with CNNs. In Proceedings of the International Conference on Signal Processing, Communications and Computing (ICSPCC), Dalian, China, 20–23 September 2019; pp. 1–5.
50. Pham, T.T.D.; Kim, S.; Lu, Y.; Jung, S.-W.; Won, C.-S. Facial action units-based image retrieval for facial expression recognition. *IEEE Access* **2019**, *7*, 5200–5207. [CrossRef]
51. Arriaga, O.; Valdenegro-Toro, M.; Plöger, P. Real-time convolutional neural networks for emotion and gender classification. *arXiv* **2017**, arXiv:1710.07557. [CrossRef]
52. Liu, K.; Zhang, M.; Pan, Z. Facial expression recognition with CNN ensemble. In Proceedings of the International Conference on Cyberworlds (CW), Chongqing, China, 28–30 September 2016; pp. 163–166.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.