Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 20(8): 13900–13920.
DOI: 10.3934/mbe.2023619
Received: 23 May 2023
Revised: 04 June 2023
Accepted: 06 June 2023
Published: 19 June 2023

Research article

Flower image classification based on an improved lightweight neural network with multi-scale feature fusion and attention mechanism

Zhigao Zeng[1,2], Cheng Huang[1,2], Wenqiu Zhu[1,2], Zhiqiang Wen[1,2] and Xinpan Yuan[1,2,*]

[1] School of Computer Science, Hunan University of Technology, Zhuzhou, Hunan 412007, China

[2] Hunan Key Laboratory of Intelligent Information Perception and Processing Technology, Zhuzhou, Hunan 412007, China

* **Correspondence:** Email: xpyuan@hut.edu.cn.

Abstract: In order to solve the problem that deep learning-based flower image classification methods lose more feature information in the early feature extraction process, and the model takes up more storage space, a new lightweight neural network model based on multi-scale feature fusion and attention mechanism is proposed in this paper. First, the AlexNet model is chosen as the basic framework. Second, a multi-scale feature fusion module (MFFM) is used to replace the shallow single-scale convolution. MFFM, which contains three depthwise separable convolution branches with different sizes, can fuse features with different scales and reduce the feature loss caused by single-scale convolution. Third, two layers of improved Inception module are first added to enhance the extraction of deep features, and a layer of hybrid attention module is added to strengthen the focus of the model on key information at a later stage. Finally, the flower image classification is completed using a combination of global average pooling and fully connected layers. The experimental results demonstrate that our lightweight model has fewer parameters, takes up less storage space and has higher classification accuracy than the baseline model, which helps to achieve more accurate flower image recognition on mobile devices.

Keywords: flower image classification; multi-scale feature fusion; depthwise separable convolution; attention mechanism

1. Introduction

Flower image classification is a branch of fine-grained image classification. Unlike coarse-grained image classification, flower image classification is more challenging because different species of floral images are relatively similar and are susceptible to lighting and distortion [1]. In addition, some flower images have leaves in the background, while others have grass in the background, and the difference in

background further increases the difficulty of the flower image classification task. Manual classification is not only costly but also prone to misclassification. Therefore, there is a need to develop efficient and accurate methods for classifying floral images with the aid of computers.

Conventional flower image classification methods are often done by manually extracting specific features or fusing multiple types of features. For example, Nilsback and Zisserman [2] extracted four different features of the target image for classification using a multi-core support vector machine (SVM) combined with weighted linear kernels to improve the flower image classification performance. Fernando et al. [3] proposed a feature fusion method based on logistic regression model to fused the color and shape features of the flower images for the classification of flower images. Angelova [4] segmented the target image, extracted the histogram of directional gradient (HOG) features at four scales of the image, then encoded the features using local constrained linear coding (LLC) and classified them using SVM. Zawbba et al. [5] first segmented flower images from the original images, then extracted the image features using both Scale Invariant Feature Transform (SIFT) and Sgmentation-based Fractal Texture Analysis (SFTA), and finally completed the classification using an SVM classifier. Inthiyaz et al. [6] proposed a level set algorithm that fuses three features of flowers, color, texture and shape, to segment images, which achieved good results on public data sets and was very helpful for subsequent classification tasks. The feature selection of the above algorithms mainly relies on the experience of researchers. For different images, the ability to select appropriate features will directly affect the accuracy of classification, so the generalization ability of the algorithm of classification will be affected.

It is well known that deep learning has been widely used in various fields in recent years because it can extract features automatically with high accuracy. The concept of deep learning was introduced by Hinton [7] in 2006. Deep learning can improve the accuracy of classification or prediction by building deep neural network models and large amounts of training data to learn significant features. Among the many neural network models, the convolutional neural network (CNN) has attracted the attention of many researchers due to its excellent achievements in image processing. The famous models AlexNet [8], VGG [9] and GoogLeNet [10] are all CNNs. Since image processing is subject to uncertainty, some researchers have started to investigate tools based on fuzzy logic, with good results [11–13]. CNN also performs well in flower image classification. For example, Liu et al. [14] combined saliency map and luminance map for flower images to perform region selection of the images and used CNN to extract features from the selected regions, and then used a softmax classifier to classify flower images with an accuracy of 84.0% on the Oxford 102 Flowers data set. Cao et al. [15] introduced a visual attention mechanism in the residual module and proposed an improved residual network model to improve the accuracy of flower classification. Xia et al. [16] used the pre-trained Inception-v3 model for flower image classification and improved the accuracy of flower classification greatly. Qin et al. [17] added the inverse residual module to the Inception-v3 model and then improved the classification accuracy of fine-grained images by inputting raw images of different sizes. Simon et al. [18] first used neural activation maps to locate key regions of fine-grained images and then extracted image features from the key regions for final classification. Cıbuk et al. [19] first extracted image features using AlexNet and VGG16 and combined them. They then selected more efficient features by the Max-Relevance and Min-Redundancy (mRMR) feature selection algorithm and finally used SVM for classification. Bae et al. [20] proposed an improved multimodal convolutional neural network (M-CNN) for flower image classification by first learning features of text data through a text

CNN, then learning image features through an image CNN and finally inputting text features and image features into the classifier through a concatenated CNN to complete the classification. Pang et al. [21] combined Feature Pyramid Network (FPN) with Bilinear-CNN (B-CNN) and proposed a Bilinear Pyramid Network (BPN), which uses up-sampling operations to unify the feature dimensions of different network layers and then fuses them by bilinear pooling with good results. Liu et al. [22] used FPN and a channel attention mechanism to locate key regions of fine-grained images and then increased the weight of key regions through a spatial attention mechanism. Guan et al. [23] proposed a channel cumulative attention mechanism that uses the Cusum function to obtain hierarchical channel attention with a clear bias and incorporates it into a ResNet model, achieving very good classification results on four fine-grained image data sets.

Although all of the above CNN-based approaches have yielded promising results for flower image classification, they still have four main shortcomings: 1) Classification accuracy is not high enough. Flower images are fine-grained images with the visual characteristic that the inter-class variance is smaller than the intra-class variance, causing the training results of the model to be different from the expectations. 2) The single convolution used in the first layer causes excessive feature loss during the process of feature exaction. Because the current CNNs use a single-scale convolutional kernel in the initial feature extraction process, they cannot extract the multi-scale features of the original image. Some models, such as GoogLeNet and Inception-v3, only start to extract multi-scale features in the middle and late stages, so these models cannot reduce the feature loss generated in the first layer. 3) There are not enough training samples. At present, there is a lack of large-scale public data sets like ImageNet for flower image classification, but the training of a CNN requires a lot of data. Although data augmentation can be used to alleviate the problem, it cannot fundamentally solve the problem of the lack of data sets. The lack of sufficient training samples affects the performance of the CNN models which are used for flower classification. 4) The storage space taken up by the model is relatively large. At present, the demand for flower recognition on mobile devices is gradually increasing, and the model is too large, making it difficult to deploy to mobile devices, which also makes the flower recognition on mobile devices not accurate enough.

To address the above shortcomings, this paper proposes a Multi-scale Feature Fusion Module (MFFM), an Improved Inception Module (IIM) and a Hybrid Attention Module (HAM) for CNN. We use AlexNet as the baseline model, replace the first convolutional layer with MFFM and add two layers of IIM and a layer of HAM in the later stage to obtain a lightweight CNN model (FHNet). The experimental results of flower image classification show that FHNet has the advantages of fewer model parameters, higher classification accuracy and better generalization ability than AlexNet and other classical models. It is demonstrated that the lightweight model proposed in this paper can help achieve high-accuracy flower image classification on mobile devices.

The contributions of this paper are summarized as follows:

1). We propose a Multi-scale Feature Fusion Module (MFFM), which contains three depthwise separable convolution branches with different sizes and can fuse features with different scales and reduce the feature loss caused by single-scale convolution.

2). We propose an Improved Inception Module (IIM) to enhance the extraction of deep features by the neural network.

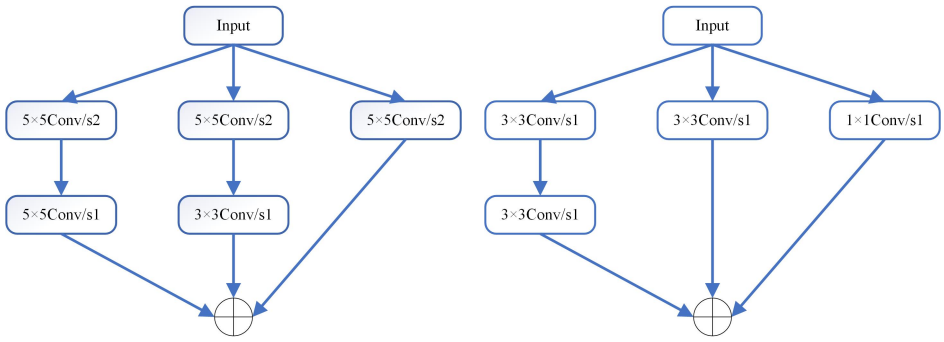
3). We propose a Hybrid Attention Module (HAM) that enhances the model's attention to key features in floral images by fusing spatial attention with channel attention.

4). We propose a novel lightweight flower image classification model (FHNet), which is based on AlexNet and incorporates MFFM, IIM and HAM. Experimental results on three flower image data sets demonstrate the superiority of FHNet.

2. Proposed module and lightweight model

2.1. Multi-scale feature fusion module

AlexNet uses a large kernel with the size of 11 × 11 in the first convolutional layer. The large kernel is undoubtedly inappropriate for the classification of fine-grained images, such as floral images, as shown by the experimental results in Figure 9. This is because the convolution kernel with large size will cause the model to fail to recognize the subtle differences in different flower images, and the accuracy of flower image classification will be affected. Therefore, this paper proposes MFFM, which can improve the ability of the model to identify tiny differences between flower images and reduce the loss of feature information. There are two types of MFFM, described as Fusion1 and Fusion2, respectively, and their structures are shown as Figure 1.



(a) Fusion1 module (b) Fusion2 module

Figure 1. Two types of MFFM.

Shown in Figure 1, MFFM has three branches of different scales. To improve the nonlinear capability of the module and reduce the loss of feature information caused by the large-scale convolution kernel, we replace the large-scale convolution with two layers of small-scale convolutions. The "s" in Figure 1 indicates the stride of the convolution kernel. To keep the output size consistent across the different branches in the Fusion1 module and to reduce the computational effort of the model, we set the stride of the first layer of convolution in the module to 2. To reduce the parameters of the module, we use depthwise separable convolution [24] instead of standard convolution. The standard convolution is decomposed by the depthwise separable convolution into two parts: channel-wise convolution and point-wise convolution. First, a single convolution kernel is used for each channel to perform the convolution operation, and then the output of the channel-wise convolution is combined by the 1 × 1 point-wise convolution. In addition, considering that different branches have different effects on the final classification accuracy of the model, they are given different weights. The weight W^i of each branch is calculated from the lowest loss value L^i obtained

after 200 epochs of individual training. The calculation is as follows:

$$W^i = \frac{\frac{1}{L^i}}{\sum_{i=1}^{3} \frac{1}{L^i}}. \tag{2.1}$$

The output of MFFM is defined as

$$\boldsymbol{Y} = \sum_{i=1}^{3} W^i \otimes \boldsymbol{F}_i\left(\boldsymbol{X}, \left\{\mathrm{Conv}_j\right\}\right), \tag{2.2}$$

where $\boldsymbol{X}$ is the input, $\otimes$ represents element multiplication, and $\boldsymbol{F}_i\left(\boldsymbol{X}, \left\{\mathrm{Conv}_j\right\}\right)$ denotes the feature mapping of the i-th branch which will be learned. Take the third branch as an example: $\boldsymbol{F}_3 = \sigma(\beta(\mathrm{Conv}_1(\boldsymbol{X})))$, in which β represents the batch normalization (BN) [25], and σ denotes the activation function ReLU. The output of the three branches is multiplied by their respective weights and then summed up to give the final output $\boldsymbol{Y}$.

2.2. *Improved Inception module*

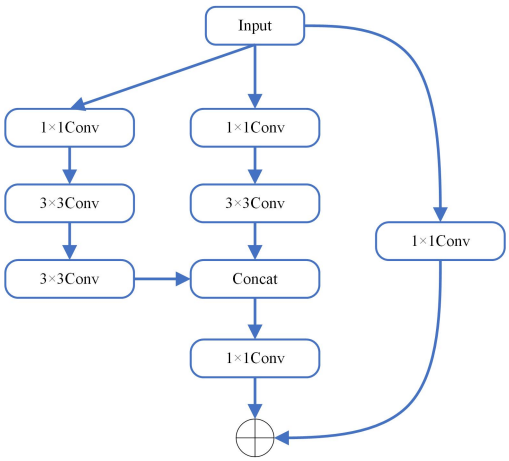


Figure 2. An overview of IIM.

The Inception module first appeared in GoogLeNet. Before that, most CNNs were stacking convolutional layers to increase the depth of the network in the hope of getting good enough results. The Inception module used in GoogLeNet is a highly representative work that increases the width of the network without increasing its depth, and it can extract sufficient image features. Later, Szegedy et al. [26] provided Inception-v2 and Inception-v3 by introducing the ideas of batch normalization and factorization. After He et al. [27] demonstrated the validity of shortcut connections, Szegedy et al. combined the Inception module with the Residual module to put forward an Inception-ResNet module [28]. The Inception-ResNet module removes 1×1 convolution from residual connections, so it is necessary for the Inception-ResNet-v1 model to use other modules to complete the dimensional transformation. To better extract the deep image features and reduce the computational parameters of the module, this study incorporated the depthwise separable convolution and the residual connection

containing 1×1 convolution into the Inception module, generating the IIM shown as Figure 2. Unlike MFFM, the depthwise separable convolution in IIM performs point-wise convolution first and then channel-wise convolution.

2.3. *Hybrid attention module*

The attention mechanism mimics the human visual mechanism, which focuses on the crucial features of an image and reduces the impact of irrelevant features of the image. The attention model was initially used in machine translation and has now become an important concept in deep learning [29]. Many scholars have introduced attention mechanisms in neural networks to facilitate their research on the topic of image processing. Hu et al. [30] proposed a new Squeeze-and-Excitation module that computes channel attention using global average pooling (GAP) to compress the feature map, and they won first place in the image classification project of the ImageNet competition in 2017. Woo et al. [31] devoted a convolutional block attention module (CBAM), which computes the channel attention and spatial attention of the input feature map successively through a serial architecture, and CBAM got better results than the results from using only channel attention. For the flower image classification problem, the attention mechanism can effectively suppress the effects of problems such as fewer training samples and minor differences between different flowers, and this can improve the accuracy of classification. Specifically, this paper proposes a HAM by combining an improved CBAM structure with residual connections.

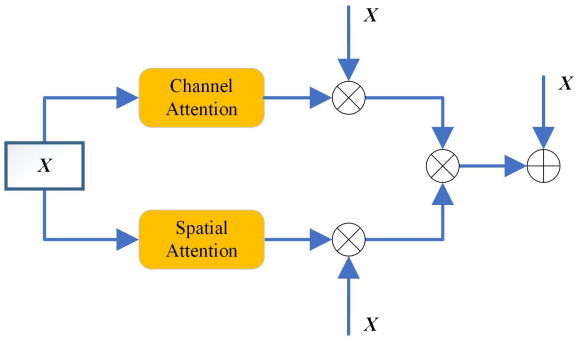


Figure 3. An overview of HAM.

Shown as Figure 3, the HAM connects the channel attention branch with the spatial attention branch in parallel. The input matrix $\boldsymbol{X}$ will learn the corresponding attention after being processed by the two attention branches. The final output of HAM is obtained by mixing the outputs of the two branches by elemental multiplication and then adding them to $\boldsymbol{X}$. The computational process of HAM is defined as

$$\boldsymbol{Y} = (\boldsymbol{W}_{\mathrm{c}} \otimes \boldsymbol{X}) \otimes (\boldsymbol{W}_{\mathrm{s}} \otimes \boldsymbol{X}) + \boldsymbol{X}, \tag{2.3}$$

where $\boldsymbol{X}$, $\boldsymbol{Y}$, $\otimes$, $\boldsymbol{W}_{\mathrm{c}}$, $\boldsymbol{W}_{\mathrm{s}}$ represent the input, output, element multiplication, channel weight matrix and spatial weight matrix, respectively. The details of the two attention branches will be described next.

2.3.1. Improved channel attention branch

To increase the weight of critical channels is the goal of channel attention. CBAM calculates channel attention in a relatively simple way, which is not suitable for direct application to flower

image classification. Inspired by the cross-channel pooling approach proposed by Goodfellow et al. [32], as shown in Figure 4, the improved channel attention branch fuses the average pooling and maximum pooling results by Eqs (2.4) and (2.5), thus making fuller use of the information from the image features. Then, a compression activation operation is used to calibrate the response of the filter, where r is the compression rate. Finally, the sigmoid function assigns values to each channel to obtain the channel weight matrix $\boldsymbol{W}_{\mathrm{c}}$ using Eq (2.6). The above process is expressed as

$$\boldsymbol{X}_1=\mathrm{AvgPool}\left(\boldsymbol{X}\right)\otimes\mathrm{MaxPool}\left(\boldsymbol{X}\right), \tag{2.4}$$

$$\boldsymbol{X}_2=\mathrm{AvgPool}\left(\boldsymbol{X}\right)\oplus\mathrm{MaxPool}\left(\boldsymbol{X}\right), \tag{2.5}$$

$$\boldsymbol{W}_{\mathrm{c}}=\delta\left(\mathrm{Conv}\left(\sigma\left(\beta\left(\mathrm{Conv}\left(\boldsymbol{X}_1;\boldsymbol{X}_2\right)\right)\right)\right)\right), \tag{2.6}$$

where β, σ, δ represent BN, ReLU, sigmoid function, respectively.

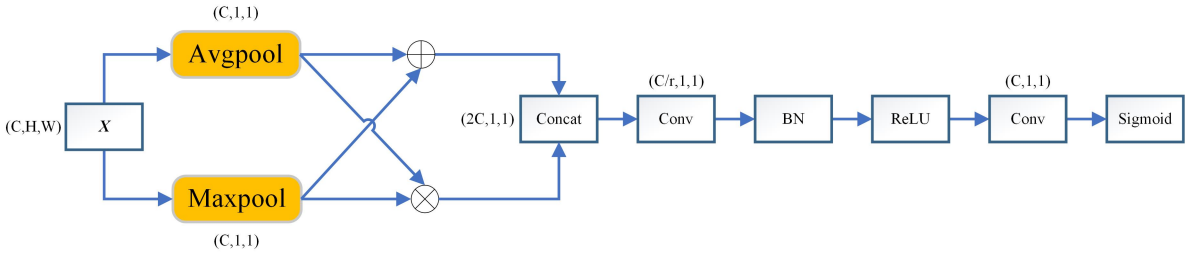


Figure 4. Improved channel attention branch.

2.3.2. Improved spatial attention branch

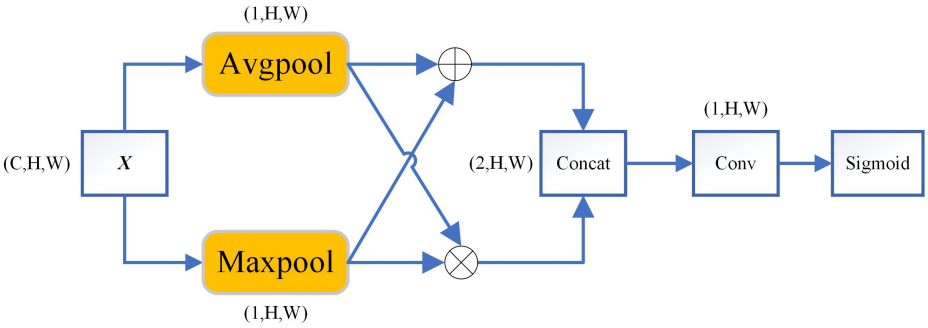


Figure 5. Improved spatial attention branch.

To increase the weight of vital feature maps is the goal of spatial attention. As shown in Figure 5, the improved spatial attention branch also uses Eqs (2.4) and (2.5) to process the pooled features and concatenates the results to generate a two-channel feature map. Unlike the channel attention branch, the spatial attention branch uses pooling operations on the channel axis. Finally, the spatial weight matrix $\boldsymbol{W}_{\mathrm{s}}$ is obtained by a basis convolution and a sigmoid function, the process of which can be expressed as follows Eq (2.7):

$$\boldsymbol{W}_{\mathrm{s}}=\delta\left(\mathrm{Conv}_{5\times5}\left(\boldsymbol{X}_1;\boldsymbol{X}_2\right)\right), \tag{2.7}$$

where δ represents the sigmoid function, and $\text{Conv}_{5\times5}$ denotes the 5×5 convolution.

2.4. Improved lightweight model

We use AlexNet as the baseline model, which is the winner of the ImageNet competition in 2012. Compared with LeNet-5 [33], AlexNet has a deeper network structure and is capable of learning higher-dimensional image features. Compared to VGG and ResNet, AlexNet has fewer convolutional layers, and therefore the improved model takes up less storage space. Based on the previous work, we made improvements to AlexNet and obtained an improved lightweight model named FHNet.

Shown as Table 1, FHNet uses the Fusion1 module in the first convolution layer, and the second to fifth convolution layers are all standard convolution layers. After the fifth convolution layer, two layers of IIM and one layer of HAM are added. Like most models, FHNet also uses the BN layer to accelerate the network training and uses max pooling between the different convolution layers to compress the spatial dimension of the feature maps. Finally, the classification of flower images is achieved using GAP and fully connected layers.

Table 1. FHNet model structure.

Layer	Type	Input size
Conv1	Fusion1	$3 \times 224 \times 224$
MaxPool	Pool 3×3	$64 \times 112 \times 112$
Conv2	Conv 5×5	$64 \times 56 \times 56$
MaxPool	Pool 3×3	$128 \times 56 \times 56$
Conv3	Conv 3×3	$128 \times 28 \times 28$
Conv4	Conv 1×1	$192 \times 28 \times 28$
MaxPool	Pool 3×3	$192 \times 28 \times 28$
Conv5	Conv 1×1	$192 \times 14 \times 14$
Conv6	IIM	$128 \times 14 \times 14$
Conv7	IIM	$256 \times 14 \times 14$
MaxPool	Pool 3×3	$512 \times 14 \times 14$
Attention	HAM	$512 \times 7 \times 7$
AvgPool	Pool 7×7	$512 \times 7 \times 7$
FC	Classifier	$512 \times 1 \times 1$

2.5. Model training process

Due to the use of the Fusion1 module, the training process of FHNet needs to be operated in the following steps:

Step 1. Pre-process data. We randomly crop the input flower images to a size of $3 \times 224 \times 224$ and perform the normalization operation for the images.

Step 2. Load data. When loading the training set, the Shuffle function is used to disrupt the order of the flower images to avoid over-fitting.

Step 3. Retain one of the branches in the Fusion1 module and block the remaining two branches.

Step 4. Feed the data into FHNet and record the lowest loss value for this branch after 200 iterations.

Step 5. Repeat steps 3 and 4 twice to record the loss values corresponding to the other two branches.

Step 6. Calculate the weights of each branch by Eq (2.1).
Step 7. Update the output Eq (2.2) of the Fusion1 module based on the weights.
Step 8. Input the flower images into the full FHNet model to complete the training.

3. Experiments

3.1. Experiment data

The experimental data sets are the Oxford 17 Flower [34] and Oxford 102 Flower [2] data sets. The former contains 17 species of flowers commonly found in Britain, and there are 80 images in each category. The latter includes 8191 flowers images in 102 categories, and there are 40 to 258 flowers images in each category. In addition, we select images of five rare flowers from the Plant Photo Bank of China (http://ppbc.iplant.cn/) and construct a small-scale data set named China 5 Flower, in which there are 150 images of each flower. Some of these images are shown in Figure 6.

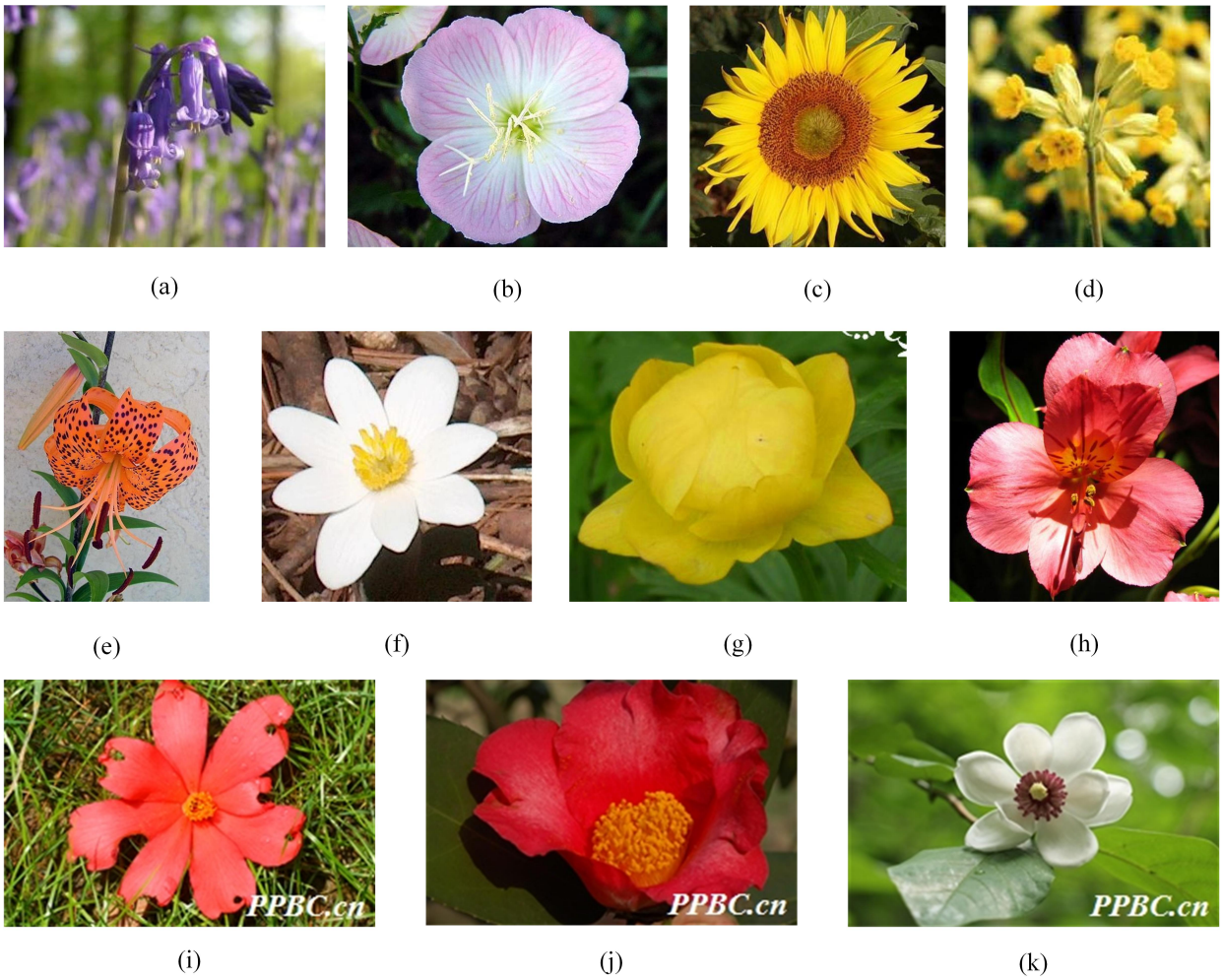


Figure 6. Some images in the flower data set, where a-d belong to Oxford 17 Flower, (e)–(h) to Oxford 102 Flower and (i)–(k) to the Plant Photo Bank of China (http://ppbc.iplant.cn/).

Since the number of images per category in the Oxford 17 Flower data set is too little, we use methods of data augmentation to expand the number of flower images. Data augmentation can make the model more robust and effectively mitigate over-fitting. Specifically, for each original image, we rotate it by 90°, flip it symmetrically, reduce the brightness by half and add Gaussian noise, respectively.

3.2. Experimental environment and hyper-parameter selection

A computer with Windows operating system is used for the experiments, the GPU is RTX 3060, the deep learning framework is PyTorch-1.8.1, and the programming language is Python.

In order to select more suitable hyper-parameters, we investigate the effect of different hyper-parameters on classification accuracy in the Oxford 102 data set. The learning rate starts with 0.001, and the program is stopped at 200 iterations. The variation of batch size and the learning rate are set as follows:

1) The batch size is set to 4, 8, 12 and 16, respectively. The learning rate is multiplied by 0.8 for every 30 iterations. The experimental results are shown in Figure 7(a).

2) The batch size is set to 4, 8, 12 and 16, respectively. The learning rate is multiplied by 0.5 for every 60 iterations. The experimental results are shown in Figure 7(b).

3) The batch size is set to 4, 8, 12 and 16, respectively. The learning rate is multiplied by 0.1 for every 100 iterations. The experimental results are shown in Figure 8(a).

The experimental results in Figures 7 and 8 and Table 2 show that the accuracy with a batch size of 4 is the lowest when the learning rate is multiplied by 0.8 every 30 iterations or by 0.5 every 60 iterations, and the accuracies of other batch sizes are basically the same. When the learning rate is multiplied by 0.1 every 100 iterations, the highest accuracy is achieved with a batch size of 8. Figure 8(b) shows the accuracy variation curves when the batch size is fixed to 8 and the learning rate is varied in different ways. It is clear that the highest classification accuracy is achieved when the learning rate is multiplied by 0.1 every 100 iterations. In summary, the hyper-parameters in this paper are set as follows: The batch size is set to 8, the learning rate is multiplied by 0.1 for every 100 iterations, the training process uses the Adam optimizer, and the loss function is selected as cross-entropy.

Table 2. Comparison of classification accuracies under different hyper-parameters.

Scheduler	4	8	12	16
s = 30/g = 0.8	93.5	96.3	95.5	96.0
s = 60/g = 0.5	95.2	96.1	95.9	96.3
s = 100/g = 0.1	95.7	97.2	95.9	95.7

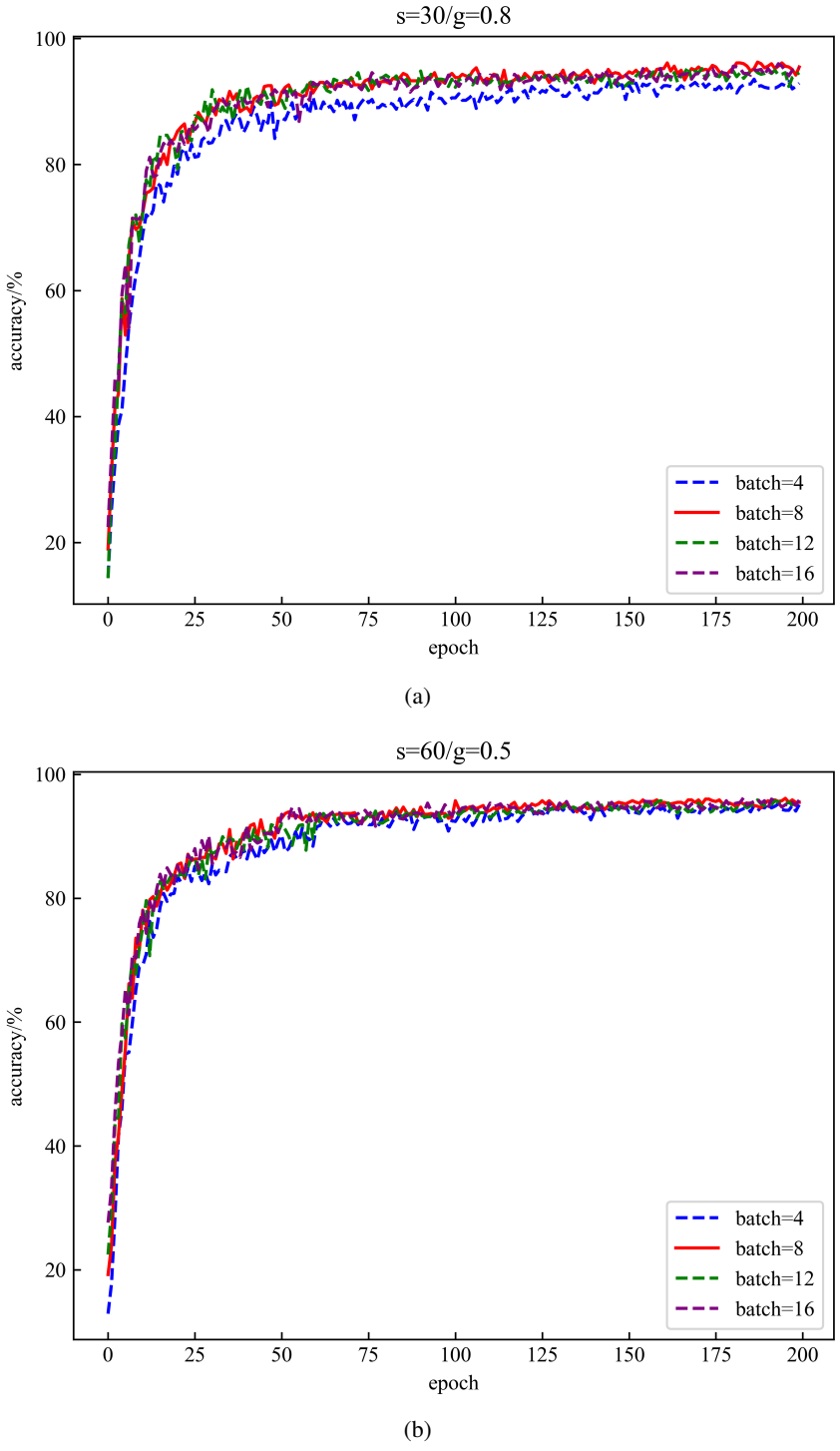


Figure 7. Accuracy variation curves of different hyper-parameters.

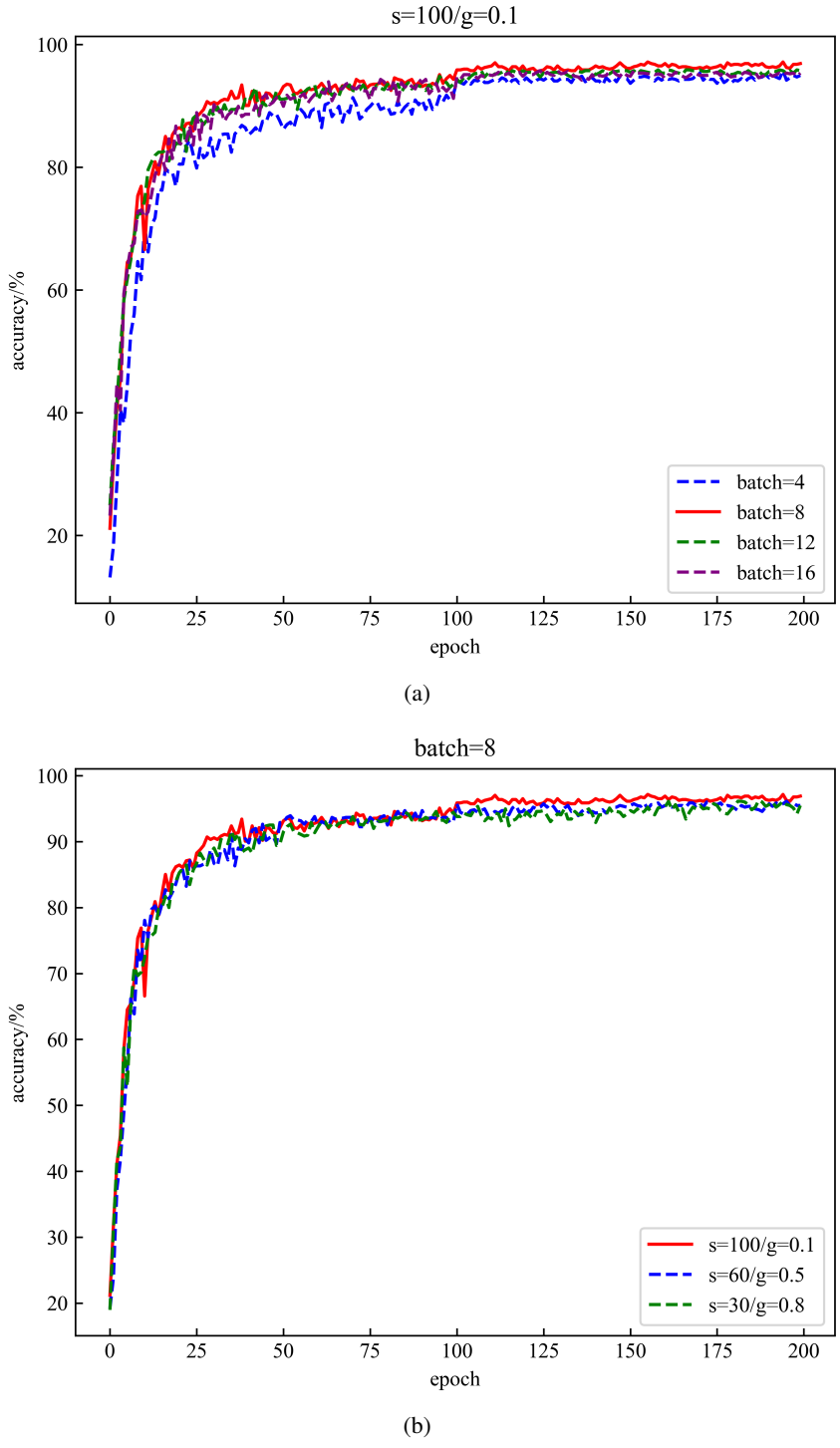


Figure 8. Accuracy change curves of different hyper-parameters.

3.3. Experimental results and analysis

In order to validate the performance of FHNet, Table 3 shows the experimental results of FHNet and other classical models on China 5 Flower. The evaluation metrics include Parameters (M), FLOPs (Floating-point Operations/G), Model size (MB), and Top-1 accuracy (%). FLOPs are used to measure the complexity of the model. The model size is the storage space occupied by the model weights after the training is completed. The experimental results obtained by some classical network models or other researchers on the Oxford 17 Flower and Oxford 102 Flower data sets are compared with the results of our model, shown in Tables 4 and 5. We use ten-fold cross-validation to make the experimental results more convincing.

Table 3. The results of different classification methods on the China 5 Flower data set.

Method	Parameters/M	FLOPs/G	Model size/MB	Top-1 accuracy/%
AlexNet [8]	61.10	0.72	217	93.3
VGG16 [9]	138.36	15.50	512	96.0
GoogLeNet [10]	6.62	1.51	39.4	96.7
MobileNet-v2 [24]	3.50	0.32	8.81	96.3
Inception-v3 [26]	23.83	2.85	93.2	94.7
ResNet34 [27]	21.80	3.67	81.3	97.3
FHNet	1.99	1.19	3.56	98.7

Table 4. The results of different classification methods on the Oxford 17 Flower data set.

Method	Top-1 accuracy/%
AlexNet [8]	89.3
VGG16 [9]	90.6
GoogLeNet [10]	86.6
MobileNet-v2 [24]	96.3
Nilsback and Zisserman [34]	88.3
ResNet34 [27]	93.9
Fernando [3]	93.0
Cao [15]	85.7
Xia [16]	95.0
SMA-Net [22]	97.3
FHNet	97.8

Table 5. The results of different classification methods on the Oxford 102 Flower data set.

Method	Top-1 accuracy/%
AlexNet [8]	81.5
VGG16 [9]	90.1
GoogLeNet [10]	79.6
MobileNet-v2 [24]	95.7
Nilsback and Zisserman [2]	72.8
ResNet34 [27]	93.7
Angelova [4]	80.7
Liu [14]	84.0
Xia [16]	94.0
Qin [17]	96.6
M-CNN [20]	93.7
BPN [21]	94.2
CCA-ResNet [23]	97.0
FHNet	97.2

The experimental results in Table 3 show that FHNet has much fewer parameters than AlexNet. The number of FLOPs for FHNet is slightly higher than AlexNet and MobileNet-v2 but lower than the other models. The model size of FHNet is much smaller than AlexNet and smaller than the classic lightweight model MobileNet-v2, which means that FHNet can be deployed on mobile devices or embedded devices. In addition, the classification accuracy of FHNet on the China 5 Flower data set is also higher than that of other models, indicating that it has an excellent performance in identifying rare Chinese flowers. Compared with AlexNet, FHNet performs slightly worse in terms of FLOPs but performs much better in terms of parameters, model size and Top-1 accuracy. The experimental results in Tables 4 and 5 show that FHNet achieves a Top-1 accuracy of 97.8% and 97.2% on the Oxford 17 and Oxford 102 flower data sets, respectively, outperforming other models or methods. Overall, FHNet has the advantages of fewer model parameters, high classification accuracy and small storage space occupation, and its overall performance is superior for meeting the needs of flower recognition on mobile devices.

3.4. Ablation studies

We conducted ablation studies on the Oxford 102 Flower data set to investigate the components of FHNet.

To investigate the contribution of MFFM, the first convolutional layer of FHNet is set to 11×11 Conv and the Fusion1 module, respectively. Shown as Figure 9, the accuracy of FHNet using the Fusion1 module is significantly higher, indicating that the large size of the convolutional kernel used in the original AlexNet model caused excessive feature loss in the feature extraction process of the flower images, which in turn affected the final classification accuracy. In contrast, MFFM effectively reduces the feature loss caused by single-scale convolution and improves model performance through the weighted fusion of depthwise separable convolutional branches with different scales.

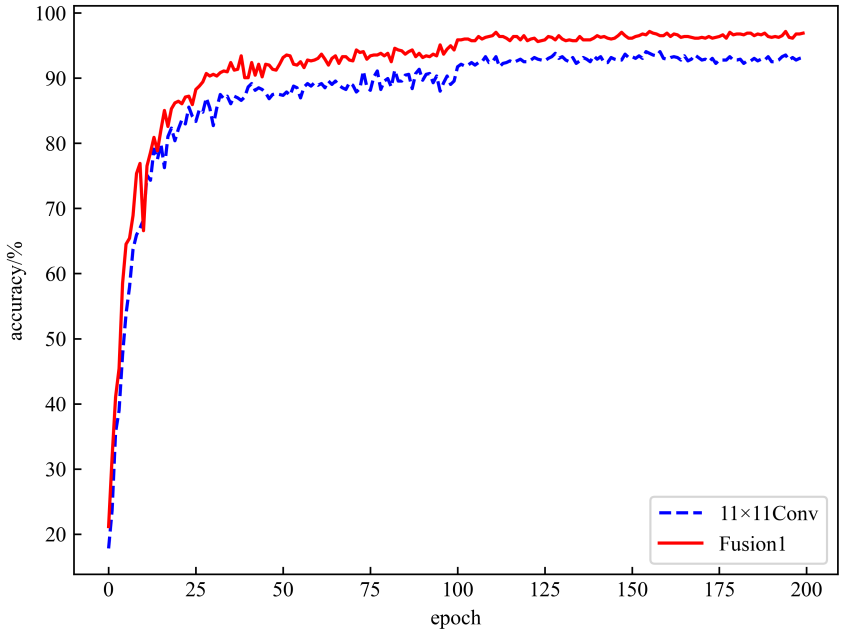


Figure 9. Accuracy change curve of FHNet with and without Fusion1.

The accuracy change curve in Figure 10 shows that when we replace the IIM in FHNet with 3×3 Conv, the classification accuracy of the model drops significantly, indicating that the IIM can enhance the model for deep feature extraction.

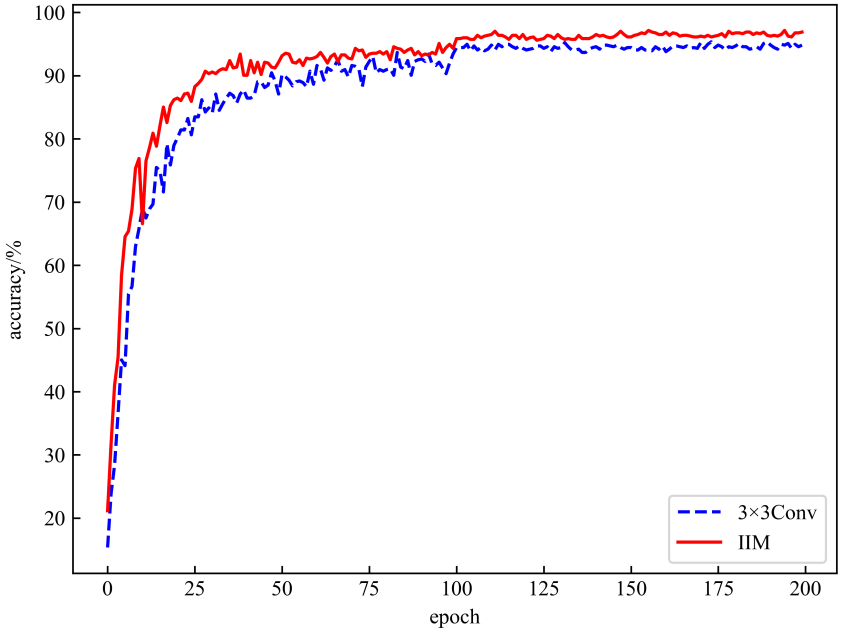


Figure 10. Accuracy change curve of FHNet with and without IIM.

By training FHNet without HAM and comparing it with the full model, we investigate the effect of HAM. The experimental results in Figure 11 show that the classification accuracy of FHNet with HAM is higher, which demonstrates that HAM can improve the model's attention to critical features.

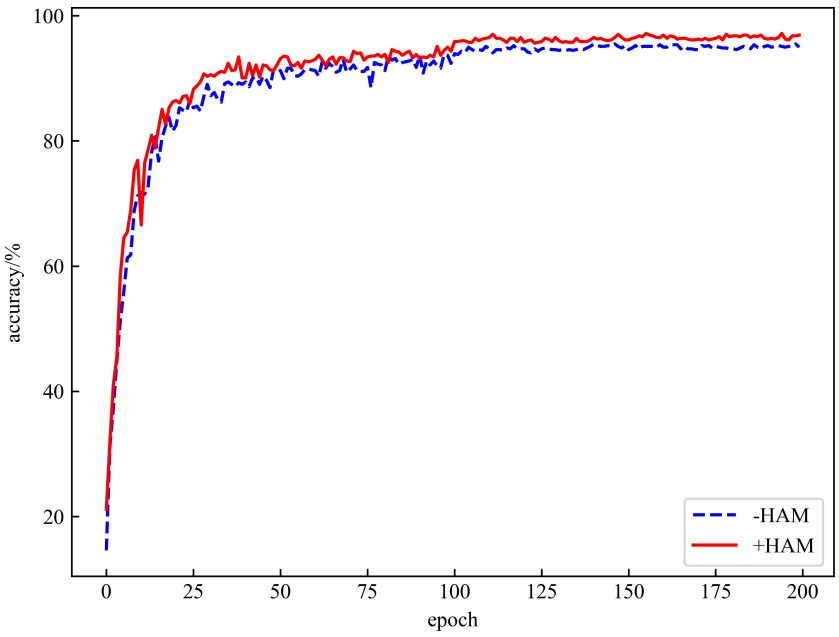


Figure 11. Accuracy change curve of FHNet with and without HAM.

We set up the first three convolutional layers of FHNet according to experiments a, b and c in Table 6 to test the effect of the number of MFFM in the model on the classification accuracy. Regrettably, the experimental results in Figure 12 show that stacking the number of MFFM does not further improve the classification accuracy. We will further analyze the results of this experiment through feature visualization.

Table 6. Model structure settings for MFFM with different number of layers.

Experiment	Conv1	Conv2	Conv3
a	Fusion1	Conv 5×5	Conv 3×3
b	Fusion1	Fusion2	Conv 3×3
c	Fusion1	Fusion2	Fusion2

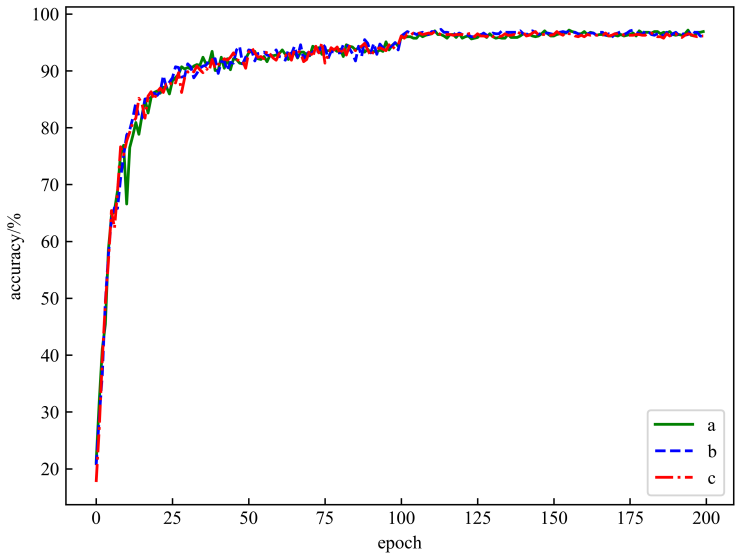


Figure 12. Accuracy variation curves for different numbers of MFFM in FHNet.

Figure 13 shows the feature visualization results of buttercup, daisy and windflower. As can be seen from the results in Figure 13, the Conv1 layer mainly extracts high-resolution features such as texture and color, while MFFM has a good enhancement effect on the extraction of such features. Conv2 and Conv3 mainly extract low-resolution features such as contours and shapes. For low-resolution features, the feature loss due to single-scale convolution is relatively minor. Therefore, increasing the number of MFFM cannot further improve the accuracy of classification.

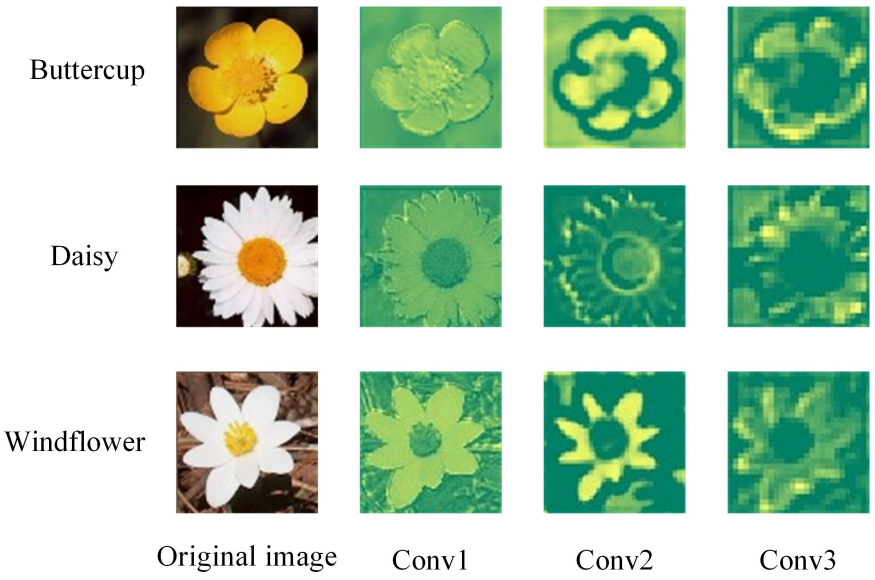


Figure 13. Feature visualization results of three kinds of flowers, with the original image from the Oxford 17 Flower data set.

4. Conclusions

In this paper, we propose a lightweight deep neural network model (FHNet) based on multi-scale feature fusion and attention mechanism for flower image classification. By using MFFM in the early stage, FHNet can extract more adequate image features. By adding IIM and HAM at a later stage, FHNet can strengthen the focus on critical features. The experimental results show that FHNet achieves fairly good classification results on three flower image data sets, demonstrating the model's applicability to the flower image classification problem. The lightweight nature of FHNet facilitates the deployment of the model to mobile or embedded devices, thus meeting the need for flower recognition on mobile devices. In addition, we found during the research that MFFM is not effective for the medium-term feature extraction process. We will try other feature enhancement methods in the future to further improve the performance of FHNet on flower image classification.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This study is supported by the Major Project for New Generation of AI (2018AAA0100400), the Scientific Research Fund of Hunan Provincial Education Department, China (21A0350, 21C0439) and the National Natural Science Foundation of Hunan Province, China (2022JJ50051, 2021JJ50058, 2020JJ6088, 2022JJ30231).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. H. Hiary, H. Saadeh, M. Saadeh, M. Yaqub, Flower classification using deep convolutional neural networks, *IET Comput. Vision*, **12** (2018), 855–862. https://doi.org/10.1049/iet-cvi.2017.0155
2. M. E. Nilsback, A. Zisserman, Automated flower classification over a large number of classes, in *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, (2008), 722–729. https://doi.org/10.1109/ICVGIP.2008.47
3. B. Fernando, E. Fromont, D. Muselet, M. Sebban, Discriminative feature fusion for image classification, in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, (2012), 3434–3441. https://doi.org/10.1109/CVPR.2012.6248084
4. A. Angelova, S. Zhu, Efficient object detection and segmentation for fine-grained recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2013), 811–818.

5. H. M. Zawbaa, M. Abbass, S. H. Basha, M. Hazman, A. E. Hassenian, An automatic flower classification approach using machine learning algorithms, in *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, (2014), 895–901. https://doi.org/10.1109/ICACCI.2014.6968612
6. S. Inthiyaz, B. Madhav, P. Kishore, Flower segmentation with level sets evolution controlled by colour, texture and shape features, *Cogent Eng.*, **4** (2017). https://doi.org/10.1080/23311916.2017.1323572
7. G. E. Hinton, R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, *Science*, **313** (2006), 504–507. https://doi.org/10.1126/science.1127647
8. A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks, *Commun. ACM*, **60** (2017), 84–90. https://doi.org/10.1145/3065386
9. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, *arXiv preprint*, (2014), arXiv:1409.1556. https://doi.org/10.48550/arXiv.1409.1556
10. C. Szegedy, W. Liu, Y. Q. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., Going deeper with convolutions, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2015), 1–9.
11. F. Marzialetti, S. Giulio, M. Malavasi, M. G. Sperandii, A. T. R. Acosta, M. L. Carranza, Capturing coastal dune natural vegetation types using a phenology-based mapping approach: The potential of sentinel-2, *Remote Sens.*, **11** (2019), 1506. https://doi.org/10.3390/rs11121506
12. M. Ragab, A. E. Abouelregal, H. F. AlShaibi, R. A. Mansouri, Heat transfer in biological spherical tissues during hyperthermia of magnetoma, *Biology*, **10** (2021), 1259. https://doi.org/10.3390/biology10121259
13. M. Versaci, G. Angiulli, P. Crucitti, D. D. Carlo, F. Laganà, D. Pellicanò, et al., A fuzzy similarity-based approach to classify numerically simulated and experimentally detected carbon fiber-reinforced polymer plate defects, *Sensors*, **22** (2022), 4232. https://doi.org/10.3390/s22114232
14. Y. Y. Liu, F. Tang, D. W. Zhou, Y. P. Meng, W. M. Dong, Flower classification via convolutional neural network, in *2016 IEEE International Conference on Functional-Structural Plant Growth Modeling, Simulation, Visualization and Applications (FSPMA)*, (2016), 110–116. https://doi.org/10.1109/FSPMA.2016.7818296
15. S. Cao, B. Song, Visual attentional-driven deep learning method for flower recognition, *Math. Biosci. Eng.*, **18** (2021), 1981–1991.
16. X. L. Xia, C. Xu, B. Nan, Inception-v3 for flower classification, in *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, (2017), 783–787. https://doi.org/10.1109/ICIVC.2017.7984661
17. J. H. Qin, W. Y. Pan, X. X. Xiang, Y. Tan, G. M. Hou, A biological image classification method based on improved CNN, *Ecol. Inf.*, **58** (2020), 101093. https://doi.org/10.1016/j.ecoinf.2020.101093
18. M. Simon, E. Rodner, Neural activation constellations: Unsupervised part model discovery with convolutional networks, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, (2015), 1143–1151.

19. M. Cıbuk, U. Budak, Y. Guo, M. C. Ince, A. Sengur, Efficient deep features selections and classification for flower species recognition, *Measurement*, **137** (2019), 7–13. https://doi.org/10.1016/j.measurement.2019.01.041

20. K. Bae, J. Park, J. Lee, Y. Lee, C. Lim, Flower classification with modified multimodal convolutional neural networks, *Expert Syst. Appl.*, **159** (2020), 113455. https://doi.org/10.1016/j.eswa.2020.113455

21. C. Pang, W. H. Wang, R. S. Lan, Z. Shi, X. N. Luo, Bilinear pyramid network for flower species categorization, *Multimedia Tools Appl.*, **80** (2021), 215–225. https://doi.org/10.1007/s11042-020-09679-8

22. C. Liu, L. Huang, Z. Q. Wei, W. F. Zhang, Subtler mixed attention network on fine-grained image classification, *Appl. Intell.*, **51** (2021), 7903–7916. https://doi.org/10.1007/s10489-021-02280-y

23. X. Guan, G. Q. Wang, X. Xu, Y. Bin, Learning hierarchal channel attention for fine-grained visual classification, in *Proceedings of the 29th ACM International Conference on Multimedia*, (2021), 5011–5019. https://doi.org/10.1145/3474085.3475184

24. M. Sandler, A. Howard, M. L. Zhu, A. Zhmoginov, L. C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2018), 4510–4520.

25. S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in *Proceedings of the 32nd International Conference on Machine Learning*, (2015), 448–456.

26. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016), 2818–2826.

27. K. M. He, X. Y. Zhang, S. Q. Ren, J. Sun, Deep residual learning for image recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016), 770–778.

28. C. Szegedy, S. Ioffe, V. Vanhoucke, A. A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in *Proceedings of the AAAI Conference on Artificial Intelligence*, (2017), 4278–4284. https://doi.org/10.1609/aaai.v31i1.11231

29. S. Chaudhari, V. Mithal, G. Polatkan, R. Ramanath, An attentive survey of attention models, *ACM Trans. Intell. Syst. Technol.*, **12** (2021), 1–32. https://doi.org/10.1145/3465055

30. J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2018), 7132–7141.

31. S. Woo, J. Park, J. Y. Lee, I. S. Kweon, Cbam: Convolutional block attention module, in *Proceedings of the European Conference on Computer Vision (ECCV)*, (2018), 3–19.

32. I. Goodfellow, D. W. Farley, M. Mirza, A. Courville, Y. Bengio, Maxout networks, in *Proceedings of the 30th International Conference on Machine Learning*, (2013), 1319–1327.

33. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, *Proc. IEEE*, **86** (1998), 2278–2324. https://doi.org/10.1109/5.726791

34. M. E. Nilsback, A. Zisserman, A visual vocabulary for flower classification, in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, (2006), 1447–1454. https://doi.org/10.1109/CVPR.2006.42