

Article

Succulent Plant Image Classification Based on Lightweight GoogLeNet with CBAM Attention Mechanism

Xingyu Tong [†], Zhihong Liang ^{*} and Fangrong Liu [†] 

College of Big Data and Intelligent Engineering, Southwest Forestry University, Kunming 650224, China; tongxingyu@swfu.edu.cn (X.T.); liufangrong@swfu.edu.cn (F.L.)

^{*} Correspondence: zhliang@swfu.edu.cn

[†] These authors contributed equally to this work.

Abstract: Aiming at the model overfitting problem caused by limited datasets and visual complexity in succulent plant classification tasks, this study proposes a GoogLeNet classification method based on light weighting and improving the Convolutional Block attention module (CBAM). Meanwhile, batch normalization (BN) operations are added after each convolutional layer to accelerate network convergence and improve model stability. In addition, the model's ability to extract key features is enhanced by integrating the channel and spatial attention mechanisms of the CBAM attention module. Experimental results show that the improved lightweight GoogLeNet achieves 99.4% classification accuracy on the validation set, effectively mitigates the overfitting problem, and maintains high computational efficiency in resource-constrained environments. The model parameters and computational complexity are significantly reduced by streamlining the Inception modules from nine to seven and introducing depth-separable convolution. To further validate the model robustness, this study extends the dataset via data augmentation methods, and the experiments show that the improved model still maintains stable performance in small dataset environments, demonstrating its advantages in data-scarce scenarios. This study provides an effective solution for the task of succulent plant classification, which has significant application value. Future research will focus on further optimization of the model structure to continuously improve the classification accuracy and robustness.



Academic Editor: Grigorios Beligiannis

Received: 1 March 2025

Revised: 25 March 2025

Accepted: 25 March 2025

Published: 28 March 2025

Citation: Tong, X.; Liang, Z.; Liu, F. Succulent Plant Image Classification Based on Lightweight GoogLeNet with CBAM Attention Mechanism. *Appl. Sci.* **2025**, *15*, 3730. <https://doi.org/10.3390/app15073730>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: succulents; light weighting; CBAM attention mechanism; GoogLeNet; classification and recognition

1. Introduction

In recent years, convolutional neural networks (CNNs), as a core method of deep learning technology, have achieved remarkable results in image classification, object detection, and other fields. CNNs extract multi-scale features of images through multiple layers of convolution and pooling operations and perform well in tasks such as plant classification [1–3]. In plant classification, researchers have widely explored strategies such as multi-scale feature extraction, transfer learning, and ensemble classifiers, which have significantly improved model performance. For example, Hu et al. [4] proposed the multi-scale fusion convolutional neural network (MSF-CNN), which achieved excellent performance on the MalayaKew Leaf [5] and LeafSnap [6] datasets, verifying the effectiveness of multi-scale features. Pereira et al. [7] used AlexNet transfer learning and image distortion techniques to improve recognition accuracy on the Flavia leaf dataset. Tan et al. [8] proposed the D-Leaf model, which combines classifiers such as support vector machines (SVMs), artificial neural networks (ANNs) and k-nearest neighbors (k-NNs), to

achieve more accurate plant species identification. Ukwoma et al. [9] combined VGG16, ResNet50, and YOLOV7 to achieve an accuracy of over 98% in classifying fruit images, further demonstrating the powerful potential of deep learning in plant image analysis. The above studies have laid a solid foundation for applying CNN in plant image recognition.

On this basis, in recent years, researchers have conducted in-depth explorations around the structural optimization and performance improvement of CNNs, with particularly groundbreaking progress made in 2024 and 2025. In 2024, Hasan et al. [10] proposed an image classification method based on CNNs, achieving an accuracy rate of 98% on the MNIST dataset, further demonstrating the powerful ability of CNNs to improve image classification performance. In 2025, Sun et al. [11] proposed the scalable quantum convolutional neural network (SQCNN) model, which uses quantum circuits to extract features in parallel, achieving a classification accuracy of 99.79% on the MNIST and Fashion MNIST datasets. This is significantly better than existing quantum neural network models and shows strong generalization capabilities. Meanwhile, the CA-MFE model proposed by Liu Yongmin et al. [12] combines deformable CNN with an attention mechanism to construct a multi-scale graph neural network (GNN), which effectively extracts multi-scale local and global features and significantly improves image classification performance. It also performs well on both the mini-ImageNet and tiered-ImageNet datasets. In addition, the CBAM-SqueezeNet model proposed by Zhao et al. [13] further optimizes the feature extraction module of the CNN by introducing channel and spatial attention mechanisms, achieving more accurate and efficient detection of target grasping by robotic arms. This model achieved a grasp detection accuracy of 94.8% and 96.4% on the Cornell Grasping Dataset and Jacquard Dataset, respectively, and a success rate of 93% in actual robotic grasping experiments. The inference speed reached 15ms, which fully balances model accuracy and speed. These research results show that CNN models that combine quantum characteristics, attention mechanisms, and multi-scale feature extraction have made significant breakthroughs in the past two years, further promoting the development of image classification and related applications.

Despite the many advances made by CNNs in the field of image analysis, they still face significant challenges in the task of succulent plant classification. Succulents come in a wide variety of species, with highly similar morphological characteristics, and are significantly affected by their growth environment [14,15]. There are more than 12,000 species of succulents known worldwide, belonging to about 80 families [16], but only about 100 species are sold as ornamental potted plants in the Chinese market. Due to the difficulty involved in identifying them with the naked eye, traditional manual classification methods are inefficient and rely on expert knowledge, making it difficult to meet large-scale automation needs. Therefore, the core challenge in the task of classifying succulents is how to improve the generalization ability of the model with a limited dataset.

To solve the above problems, researchers have proposed various effective strategies in recent years, including data augmentation, dropout methods, and label regularization. Data augmentation methods help models learn more robust features by expanding sample diversity and are a common means of alleviating overfitting. Methods such as CutMix [17] and RandAugment [18] significantly improve the adaptability of models in complex image tasks through image cropping, splicing, and transformation. Dropout-like methods such as MaxDropout [19] and DropBlock [20] reduce the model's dependence on specific weights and alleviate the overfitting of deep networks. Label regularization methods such as Label Smoothing [21] and JoCoR [22] have also achieved remarkable results in dealing with noisy labels and improving model stability. The above research provides an important technical reference for the task of image classification of succulents.

However, in the task of classifying succulents, the scarcity of data, small differences between classes, and large differences within classes significantly exacerbate the problem of overfitting the model. To alleviate this problem, researchers have proposed various effective strategies, among which data augmentation techniques have achieved remarkable results as an important means to improve the generalization ability of models. Feng et al. [23] introduced advanced enhancement methods such as Cutout and Mixup, combined with a variety of image transformation strategies, which improved the model accuracy from 67.35% to 78.68%, significantly improving the robustness of the model in complex scenarios. Similarly, Wen et al. [24] used data augmentation methods such as random flipping, rotation, and color jittering, combined with transfer learning strategies, to improve the model's test set accuracy from 59.48% to 96.90%, effectively alleviating the overfitting of the model. In addition, in recent years, researchers have gradually introduced attention mechanisms to improve the model's ability to focus on key features, thereby exhibiting excellent performance in fine-grained classification tasks [25,26]. While emphasizing significant regions of an image, the attention mechanism can effectively suppress redundant features, further improving the model's classification accuracy in complex images. To further address the problem of data scarcity, researchers have begun to explore few-shot learning methods to enable effective training of models on small datasets. The few-shot learning model combined with an attention mechanism aggregates multi-level fine-grained features through a multi-branch structure, achieving more accurate feature representation and classification performance, especially in the task of succulent plant classification with limited data resources [27,28]. The above studies show that combining methods such as data augmentation and attention mechanisms provides an effective way to solve the overfitting problem in succulent plant classification and improve the generalization performance of the model, opening up new research directions for the application of deep learning in the analysis of complex images.

In summary, this paper proposes a multi-succulent plant classification algorithm framework that combines deep learning, a lightweight nature, and an attention mechanism to address the shortcomings of current classification methods. The research object is a multi-succulent plant image dataset, covering multi-succulent plants from various families. The core of the research is to design a model based on a multi-branch structure, which aims to more accurately capture the key details of multi-succulent plants, enhance category distinction, and improve the generalization ability of the model. The innovations of this model mainly include the following two aspects:

(1) Spatial-channel attention mechanism (CBAM): The CBAM module (Convolutional Block Attention Module) enhances feature expression through a combination of channel and spatial attention. The channel attention module introduces global average pooling and max pooling to generate weights for different channels and multiply them with the original features to enhance the feature expression of key channels. Spatial attention combines the average pooling and maximum pooling results of the input features, generates a spatial attention map through a 7×7 convolution, and then multiplies it with the original features to ensure that the network can focus on important spatial locations. The optimization and integration of the CBAM module improves the model's feature extraction ability and further alleviates the classification difficulties caused by the similar morphological characteristics of succulents.

(2) Lightweight Inception Module: The lightweight Inception module replaces the original 5×5 convolution with two 3×3 convolutions. This design significantly reduces the computational complexity and parameters while maintaining the same receptive field. In addition, the number of convolution channels in the module is reduced to further reduce the amount of computation. Compared with the original Inception module, the lightweight

module integrates CBAM, which adds an attention mechanism to the output feature map after stitching, making the feature representation more accurate. These modifications make the lightweight Inception module not only have a lower computational cost, but also retain good feature extraction capabilities, making it suitable for application scenarios with limited computing resources.

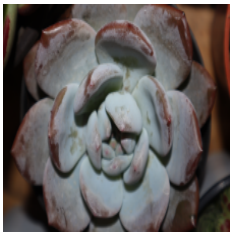
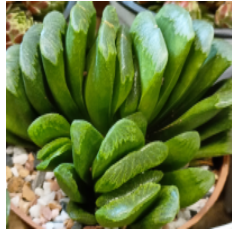
In this study, GoogLeNetCBAM is chosen as the model architecture, mainly based on its advantages in feature extraction, parameter optimization, and overfitting prevention. GoogLeNet adopts the Inception module with a powerful multi-scale feature extraction ability, which can extract rich image features under different sensory fields, especially in the case of succulent plants, considering the large number of species and high similarity between some categories, which helps the model capture more discriminative features and improve the classification accuracy. Meanwhile, the CBAM attention mechanism introduced by the model combines channel attention and spatial attention, which can effectively guide the model to focus on the key feature regions and inhibit the interference of irrelevant information, thus further improving the generalization ability of the model. To further alleviate the overfitting problem in small sample scenarios, we introduce various optimization strategies in the model training process. These include: data enhancement to enrich the diversity of data samples and enhance the model's adaptability to image transformations; the introduction of dropout ($p = 0.3$) in the fully connected layer to reduce neuron co-adaptation and improve the robustness of the model; the addition of weight decay (Weight Decay = 1×10^{-4}) in the Adam optimizer to limit the amplitude of the parameters and prevent overfitting caused by too-large model parameters; and the introduction of an early stopping strategy to prevent overfitting caused by too-large model parameters. The early stop strategy (patience = 15) is introduced to stop training when the accuracy of the validation set is not improved in consecutive rounds, to avoid the model from overfitting due to overtraining. In summary, the advantages of GoogLeNetCBAM architecture in feature extraction and attention mechanism, combined with a variety of optimization strategies, enable the model to improve the classification accuracy while significantly enhancing the stability and generalization ability, ensuring its excellent performance in small sample scenarios.

2. Materials and Methods

2.1. Succulent Data Collection

The dataset for this study comprises a total of 691 succulent plant images covering 10 categories. The samples were obtained from a succulent planting site in cooperation with Southwest Forestry University. The image data were captured by a Huawei Mate 30 cell phone (40 megapixels, f/1.8 wide-angle lens) at an angle of 90° and a height of 50 cm. Image data processing included two steps: first, the succulent images were cropped using OpenCV's Sobel edge detection method to ensure that the samples had a uniform image background; second, data enhancement was performed, including image flipping, resizing (224×224), and grayscale conversion to improve the model's generalization ability. Meanwhile, to ensure effective training and reasonable evaluation of the model, we divided the dataset into a training set and test set in a ratio of 9:1, where the training set images were used for the training of the model and parameter optimization, while the test set images were used for evaluating the generalization performance and classification accuracy of the model. Table 1 details the specific distribution of the 10 succulent plant categories and their corresponding image samples.

Table 1. Sample table of succulent plant categories and their corresponding images.

Sample	Sample	Sample	Sample	Sample
				
<i>Crassula obliqua</i> 'Gollum'	<i>Echeveria</i> <i>chihuahuensis</i>	<i>Crassula perforata</i>	<i>Haworthia cooperi</i> var. <i>truncata</i>	<i>Sedum burrito</i>
				
<i>Echeveria agavoides</i> 'Ebony'	<i>Senecio haworthii</i>	<i>Graptopervia</i> 'Opalina'	<i>Haworthia truncata</i>	<i>Sedum rubrotinctum</i>

In Table 1, we find that some categories have a high degree of similarity, which is likely to lead to overfitting of the model. Specifically, *Crassula perforata* and *Graptopervia* 'Opalina' have similar leaf colors and surface textures, while *Haworthia cooperi* var. *truncata* and *Haworthia truncata* also have strong morphological similarities. In addition, *Sedum burrito* and *Sedum rubrotinctum* further increase the risk of model confusion due to their spherical leaves and dense arrangement. To alleviate the overfitting problem caused by these category similarities, we have taken two optimization measures in the model training process: first, we have expanded the original image dataset by applying various data augmentation strategies. These data augmentation strategies have effectively improved the robustness of the model to image transformations. Second, we introduced the CBAM attention mechanism into the model structure. By fusing channel attention and spatial attention modules, the model is guided to focus on more discriminative feature regions in the image, thereby further improving the model's ability to distinguish between similar categories. To visually demonstrate the confusion that may occur when there are visually similar categories in the model, we have adjusted the structure of Table 1. The three columns at the end of Table 1 specifically compare examples of images from categories that are prone to confusion.

2.2. Experimental Environment

This experiment relies on a robust hardware configuration to ensure efficient training and inference of deep learning models. Specifically, the hardware setup includes an Intel(R) Core(TM) i9-12700 CPU (Intel Corporation, Santa Clara, CA, USA) and NVIDIA GeForce RTX 4090 Laptop GPU (NVIDIA Corporation, Santa Clara, CA, USA), which provide powerful computational capabilities for handling complex model training tasks. The software environment comprises the Windows 11 64-bit operating system, with Python 3.8 as the programming language and PyTorch 1.10 as the deep learning framework. Additionally, the experiment employs the JupyterLab platform for development and interaction, offering a flexible and intuitive environment for coding, debugging, and visualization. This configuration not only provides sufficient computational resources and a stable development environment for model training and testing but also ensures the efficiency and

reproducibility of the experiment, serving as a solid foundation for subsequent network structure optimization and model performance improvement.

In this study, the key hyperparameters of the model were carefully set to enhance performance and prevent overfitting. The Adam optimizer was used and the learning rate was set to 0.0003 to ensure that the model maintained stability while converging quickly. The batch size was set to 8, which helped to improve the model's generalization ability on small sample datasets. For the model architecture, GoogLeNetCBAM with auxiliary classifiers and weight initialization was chosen to enhance the feature extraction capability and accelerate convergence. In addition, a combination of weight decay (1×10^{-4}) and an early stopping strategy was used to further enhance the stability and generalization ability of the model.

2.3. Improved Lightweight GoogLeNet

2.3.1. CBAM

CBAM is an effective extension to the classical GoogLeNet architecture, aiming to enhance the model's ability to focus on important features by introducing an attention mechanism, thus improving its performance in image classification and target detection tasks. As shown in Figure 1, the CBAM module consists of two components, channel attention and spatial attention, which act sequentially on the feature map, from global information to local features for fine-grained weighting [29]. In these two stages, channel attention focuses on “which feature” is more important, while spatial attention focuses on “where” is more important. In channel attention, based on the global average pooling and maximum pooling, the attention weight of each channel is generated, and the channels are weighted to highlight the important channels. In spatial attention, based on the average pooling and maximum pooling of channels, a spatial attention map is generated to highlight important spatial locations.

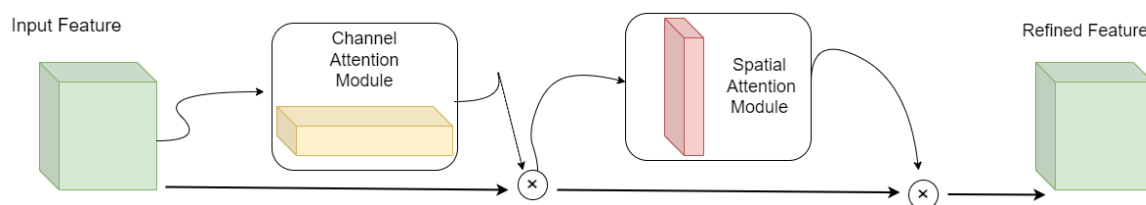


Figure 1. Structure of the CBAM attention mechanism.

Firstly, the channel attention mechanism captures the global information of each channel through global average pooling and max pooling, as shown in Equations (1) and (2). Global average pooling and global maximum pooling aggregate the input feature maps through spatial dimensions, as follows:

$$Avg_pool(X) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W X(i, j) \quad (1)$$

$$Max_pool(X) = \max X(i, j) \quad (2)$$

These two pooling approaches process each channel of the input feature map separately to generate two different global feature descriptions. Subsequently, these two features are computed and merged by a multilayer perceptron (MLP) to generate a set of attention weights, as shown in Equation (3).

$$M_c(X) = \sigma(FC2(ReLU(FC1(AVG_pool(X))))) + \sigma(FC2(ReLU(FC1(AVG_pool(X))))) \quad (3)$$

The updated feature map is obtained by weighting each channel of the original feature map and finally multiplying the channel's elements element-wise. This process allows the network to better focus on “which features” are more important [29]. The structure of channel attention is shown in Figure 2:

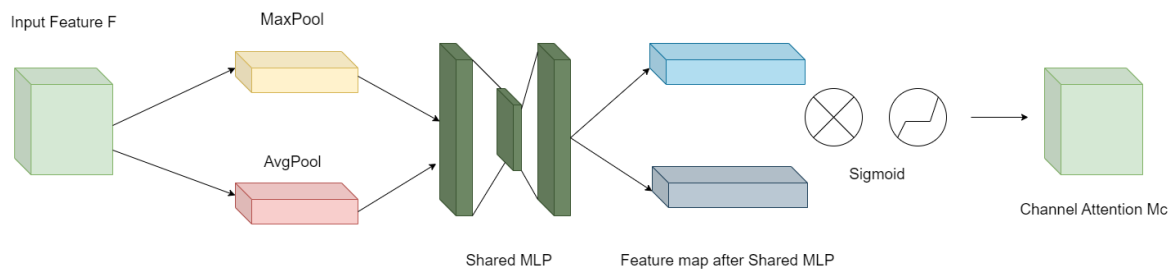


Figure 2. Structure of channel attention.

Next, the channel-weighted feature maps are input to the spatial attention mechanism. As shown in Equations (4) and (5), the spatial attention mechanism first performs average pooling and maximum pooling on the channel dimensions to obtain two 2D feature maps.

$$Avg(X) = \frac{1}{C} \sum_{c=1}^C X(C) \quad (4)$$

$$Max(X) = \max X(C) \quad (5)$$

Then, as shown in Equation (6), these two feature maps are convolved with a 7×7 kernel to generate a spatial attention map, and the spatial attention structure map is shown in Figure 3:

$$M_s(X) = \sigma(\text{Conv } 7 \times 7([Avg(X); Max(X)])) \quad (6)$$

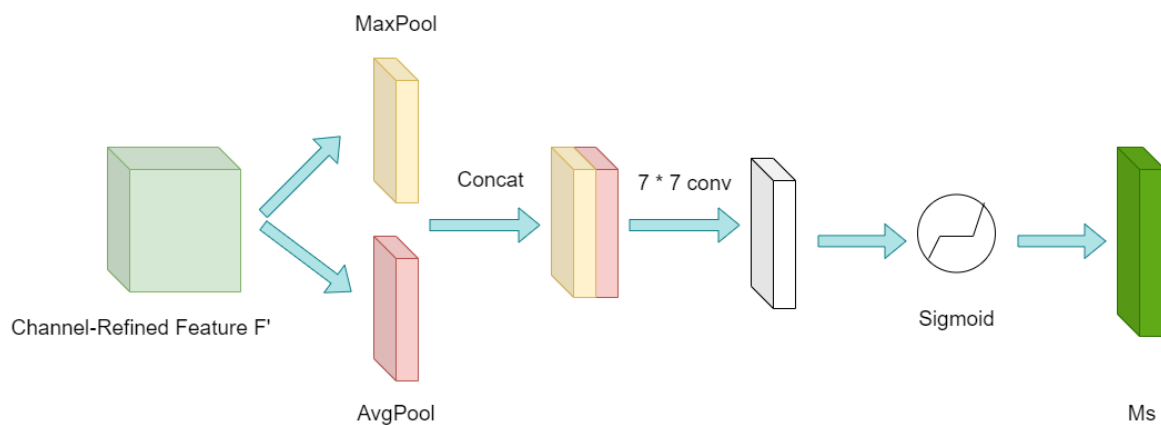


Figure 3. Structure of spatial attention.

This attention map is applied to the feature map after the channel update to complete the joint optimization of space and channel. The spatial attention map is weighted to emphasize key spatial regions in the feature map, allowing the model to more accurately identify “which locations” are critical to the task [29].

2.3.2. Lightweight Inception Module

In the improved GoogLeNet network, several optimizations are made to address the computational complexity and gradient vanishing problems of the original model. First, the nine Inception modules in the original GoogLeNet are reduced to seven lightweight

Inception modules. To maintain the efficiency and accuracy of the network, the improved Inception module introduces depth-separable convolution, which replaces the original complex convolution operation and reduces the number of parameters and computation. In addition, the new Inception module introduces a batch normalization (BN) operation after each convolutional layer to accelerate the convergence of the network and eliminates the Local Response Normalization (LRN) layer in the original GoogLeNet to avoid the computational overhead introduced by repeated normalization. This is shown in Figures 4 and 5.

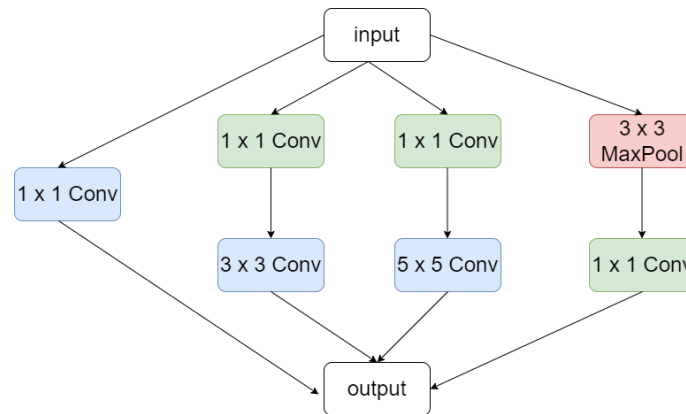


Figure 4. The original Inception module.

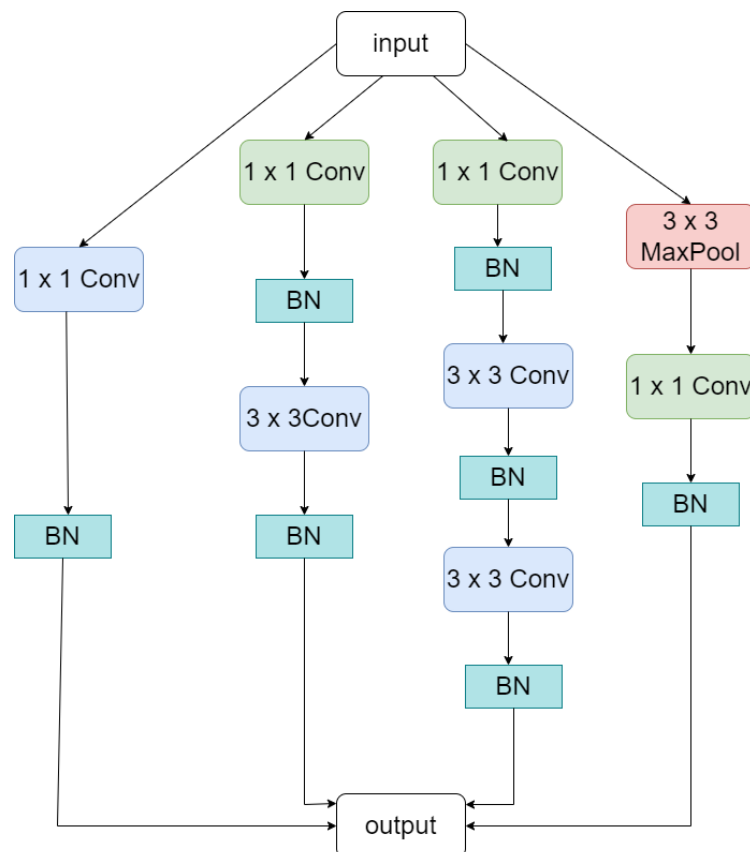


Figure 5. The improved Inception module.

Second, in the original model, two auxiliary classifiers were introduced after the 4th and 7th Inception modules to prevent gradient vanishing. The improved model reduces these two auxiliary classifiers to one and repositions it after the 5th Inception module. Also, the new auxiliary classifiers incorporate a batch normalization operation to further

improve the stability of training and prevent overfitting. With these optimized designs, the improved GoogLeNet model reduces the computational complexity while still effectively maintaining the model performance for resource-limited environments, as shown in Table 2.

Table 2. Network structure and parameter settings of the lightweight GoogLeNet model.

Type	Convolution Kernel Size	Stride	Output Size	1 × 1 Convolution (1)	Dimensionality -Decreasing	3 × 3 Convolution	1 × 1 Convolution (2)
conv	7 × 7	2	112 × 112 × 64	-	-	-	-
conv	1 × 1	1	56 × 56 × 64	-	-	-	-
conv	3 × 3	1	56 × 56 × 192	-	-	-	-
Inception 3a	1 × 1, 3 × 3, 5 × 5	- - -	28 × 28 × 256	64	96	128	32
Inception 3b	1 × 1, 3 × 3, 5 × 5	- - -	28 × 28 × 480	128	128	192	64
Inception 4a	1 × 1, 3 × 3, 5 × 5	- - -	14 × 14 × 512	192	96	208	64
Inception 4b	1 × 1, 3 × 3, 5 × 5	- - -	14 × 14 × 512	160	112	224	64
Inception 4c	1 × 1, 3 × 3, 5 × 5	- - -	14 × 14 × 512	128	128	256	64
Inception 4d	1 × 1, 3 × 3, 5 × 5	- - -	14 × 14 × 528	112	144	288	64
Inception 4e	1 × 1, 3 × 3, 5 × 5	- - -	14 × 14 × 832	256	160	320	128
Inception 5a	1 × 1, 3 × 3, 5 × 5	- - -	7 × 7 × 832	256	160	320	128
Inception 5b	1 × 1, 3 × 3, 5 × 5	- - -	7 × 7 × 1024	384	192	284	128

Table 2 Network structure and parameter settings of the lightweight GoogLeNet model (convolutional layers (conv): the initial part consists of three convolutional layers, 7 × 7, 1 × 1, and 3 × 3 convolutions, which are used for feature extraction. Inception module: the Inception module uses several convolutional kernels of different sizes (1 × 1, 3 × 3, and 5 × 5) in conjunction with pooling operations. The table contains the parameters of the convolution kernels for each layer and the corresponding output size. 1 × 1 convolution (1) and 1 × 1 convolution (2): 1 × 1 convolution is used to reduce the number of channels and to integrate the features.

The core operation of the Inception module is the convolution operation. The lightweight Inception module utilizes 1 × 1 convolution and 3 × 3 convolution instead of 5 × 5 convolution. The basic mathematical formula for convolution is as follows.

For the convolution operation, the value of the output feature map is computed from the weighted sum of the input feature map and the convolution kernel:

$$Y(i, j) = \sum_{m=0}^{K-1} \sum_{n=0}^{K-1} X(i+m, j+n) \cdot W(m, n) \quad (7)$$

where the convolution kernel size of K is 1×1 or 3×3 .

In the lightweight Inception module, replacing the 5×5 convolution with two consecutive 3×3 convolutions, the formula still follows the basic form of this convolution operation, but is divided into two 3×3 convolutions, respectively:

$$Y_1(i, j) = \sum_{m=0}^{K-1} \sum_{n=0}^{K-1} X(i+m, j+n) \cdot W_1(m, n) \quad (8)$$

$$Y_2(i, j) = \sum_{m=0}^{K-1} \sum_{n=0}^{K-1} X(i+m, j+n) \cdot W_2(m, n) \quad (9)$$

By replacing the 5×5 convolution with two 3×3 convolutions, it is possible to keep the same receptive field (5×5 coverage) but reduce the number of parameters and the amount of computation. Meanwhile, the convolutional layers in the Inception module generate a large number of computational operations whose complexity can be quantified by the number of floating point operations (FLOPs). For each convolutional layer, the FLOPs are calculated as:

$$FLOPs = 2 \times (C_{in} \times H_{out} \times W_{out} \times K \times K \times C_{out}) \quad (10)$$

where C_{in} is the number of input channels; C_{out} is the number of output channels, which also represents the height and width of the output feature map; and H_{out} and W_{out} are the sizes of the convolutional kernel. By replacing the 5×5 convolution with two 3×3 convolutions, the number of computations in the FLOPs formula is significantly reduced.

Batch normalization is introduced behind the convolutional layers in each Inception module to speed up model convergence and stabilize the training process. The mathematical formula for batch normalization is:

$$\hat{X}^{(K)} = \frac{X^{(K)} - \mu_{batch}}{\sqrt{\sigma_{batch}^2 + \epsilon}} \quad (11)$$

where $X^{(K)}$ is the k th value of the input feature map; μ_{batch} is the mean of a small batch of data; σ_{batch}^2 is the variance of the small batch; ϵ is a small constant that prevents division by zero; and $\hat{X}^{(K)}$ is the output after batch normalization. The purpose of batch normalization is to allow the data to have the same distribution (zero mean, unit variance) as it passes through each layer, thus stabilizing the training process of the network and accelerating convergence.

In the training process of the network, the Cross-Entropy Loss function (CELF) is used to measure the difference between the model output and the real labels with the formula:

$$Loss = - \sum_{i=1}^c y_i \log(p_i) \quad (12)$$

where c is the number of categories; y_i is a real label; and p_i is the probability value predicted by the model.

The losses of the auxiliary classifiers are also used in the training process, where the losses of the primary and auxiliary classifiers are proportionally combined into a total loss:

$$Total\ Loss = Loss_0 + 0.3 \times Loss_1 + 0.3 \times Loss_2 \quad (13)$$

where $Loss_0$ is the loss of the main classifier, and $Loss_1$ and $Loss_2$ are the losses of the two auxiliary classifiers.

3. Results

3.1. Contrastive Learning Succulent Image Classification Model

To further validate the advantages of the lightweight GoogLeNet model, this paper trains the succulent dataset on the Aexnet, VGG16, GoogLeNet, and lightweight GoogLeNet network models, respectively. The validation accuracy curves and loss curves of these network models in the training phase are shown in Figure 6.

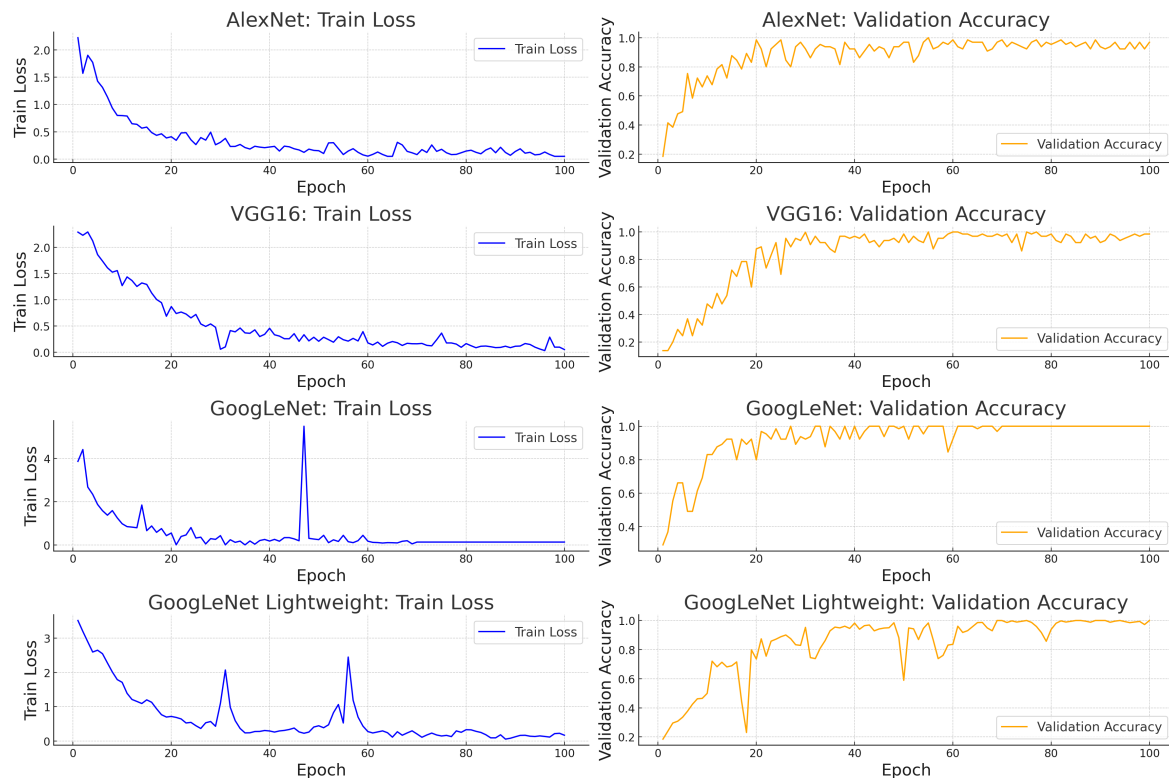


Figure 6. Validation accuracy and loss curves for each network model.

From Figure 6, we can see that there are obvious differences in the performance of the four models in terms of training loss and validation accuracy: AlexNet has a faster decline in training loss and a stable validation accuracy of about 90%, but the complexity of the model is low and there is limited room for improvement in the accuracy; VGG16 has a smooth decline in training loss and a gradual improvement in validation accuracy, but there are large fluctuations in the later stage, indicating that it is prone to overfitting and has a weaker generalization ability. The validation accuracy of GoogLeNet improves faster in the early stage, but there are large fluctuations in the later stage, which shows the overfitting phenomenon, and the model's performance on the validation set is not stable enough. In contrast, the lightweight GoogLeNet performs well, the training loss decreases rapidly, the validation accuracy increases steadily and finally exceeds 90%, and the fluctuation in the validation accuracy is small, which indicates that the model has good generalization ability. Introducing the lightweight module reduces the complexity of the model, solves the overfitting problem of the original GoogLeNet, and maintains high accuracy despite the limited resources. Overall, the lightweight GoogLeNet has the best overall performance among these four models, solving the overfitting problem and achieving excellent performance on the validation set.

3.2. Comparative Analysis of Data Enhancement Models

In order to verify the robustness of the improved model to the size of the dataset, we extended it by applying various data enhancement strategies to the original image dataset. Specific enhancement methods include: random horizontal flipping to simulate changes in left and right viewpoints, random rotation to generate image views with different angles, color dithering, which entails the random perturbation of brightness, contrast, saturation, and hue to simulate complex lighting conditions, and random cropping and scaling, which is an adjustment of image field of view ranges and enhances the learning ability of local features. Through these enhancement operations, five diverse enhanced images were generated for each original image. The enhanced data significantly increased the diversity in terms of spatial transformations, color distributions, and image contents, successfully simulating complex real-world application scenarios. These enhancement strategies not only effectively improved the generalization ability of the deep learning model, but also enhanced its robustness to image transformations. In addition, data augmentation effectively mitigates the overfitting risk of the model by expanding the diversity of the data under limited sample conditions, and provides the model with more comprehensive and diverse training samples. This enables the model to perform more stably and accurately under complex conditions. Eventually, the number of images in the original dataset was expanded from 691 to 3455 succulent images, providing rich data support for subsequent experiments.

In order to verify the robustness of the improved model to the dataset size, we conducted experiments on the original dataset of 691 images and the extended dataset of 3455 images after data enhancement (as shown in Figure 7). In the original dataset, the training set was 626 images and the test set was 65 images, and the final validation accuracy of the model reached 98.5%, with a training loss as low as 0.001, which demonstrates the efficient learning ability and excellent performance of the improved model under small dataset conditions. In the extended dataset after data enhancement, the training set contained 3180 images and the test set 342, and the validation accuracy was slightly improved to 99.4%, while the training loss was kept at 0.001 (as shown in Table 3). This comparison experiment verifies the efficient learning and stability of the improved model in a small dataset environment, and the effect of data augmentation was shown to be minimal due to saturation, although it did improve the accuracy. The robustness and effectiveness of the model have been fully demonstrated under the conditions of a small dataset with strong adaptive ability, especially in application scenarios where data scarcity is an issue.

Table 3. Comparison between the original dataset and the enhanced dataset.

Dataset	Number of Images	Training Set	Test Set	Train_Loss	Val_Accuracy (%)
Succulent dataset	691	626	65	0.001	98.5
Enhanced succulent dataset	3455	3180	342	0.001	99.4

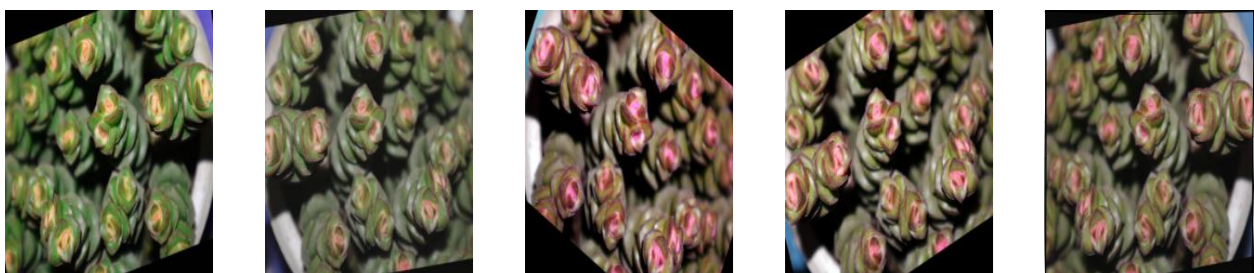


Figure 7. Data Enhancement Approach.

By comparing the experimental results on datasets of different sizes, it can be seen that the improved model is able to adapt effectively to both small and large datasets, and the validation accuracy is always maintained at a high level. Although the data enhancement technique further optimizes the generalization ability of the model by extending the data diversity, the experimental results on small-scale datasets show that the model is still able to achieve high performance stably under data-constrained conditions. This result fully demonstrates that the model is insensitive to the size of the dataset and also has good robustness under a limited sample size. In addition, the loss values of the model on both small and extended datasets in the experiments remain consistently as low as 0.001, which further supports that the optimization process of the model does not depend on changes in dataset size. This phenomenon indicates that the improved model has high efficiency in feature extraction and decision learning, and can still fully mine effective information and realize high-precision prediction in the case of fewer training samples.

In summary, the experimental results demonstrate that the improved model has a low dependence on dataset size and can achieve high validation accuracy and stable training loss using both small and large-scale datasets. This provides important support for the application of the model in data-constrained scenarios, and further illustrates the robustness and reliability of the model under dataset size variation.

3.3. Model Predictive Analytics

The purpose of this experiment was to evaluate the performance of the improved GoogLeNetCBAM model in the succulent plant image classification task, and to explore the effect of mixed-precision training on the model performance improvement. The dataset consists of 691 images of succulent plants, covering 10 categories. It was divided into a training set and a test set in a ratio of 9:1, where the training set contained 626 images and the test set contained 65 images. The experiment evaluated the classification accuracy and confidence of the model by visualizing and analyzing the test set.

In the experimental setup, the hardware environment was first checked for CUDA support to ensure that the model could run efficiently on the GPU. Image preprocessing was performed using a uniform process, including resizing the images to 224×224 pixels and applying normalization (0.5 mean and 0.5 standard deviation for the three RGB channels) to ensure format consistency in the input data. For modeling, we loaded the pre-trained GoogLeNetCBAM model. The model is based on the classic GoogLeNet architecture with the addition of the Convolutional Block Attention Module (CBAM), which improves the feature representation of the model by enhancing attention to important feature regions. The experiments turned off the auxiliary classifiers and focused on the prediction results of the main branch loaded with pre-trained weights. The output of the model was processed by Softmax to generate the probability distribution of each category, and the category with the highest probability was finally selected as the prediction result.

The experiment visually displayed the prediction results of the model and output the probability distribution of each category at the terminal, to analyze the classification performance of each model (as shown in Table 4). The values in Table 4 represent the Softmax output probabilities of each model when classifying succulents, which were used to measure the confidence of each model in each category. During the experiment, a random succulent image was selected from the validation set as a sample, and it was input into the AlexNet, VGG16, GoogLeNet, and lightweight GoogLeNet models for prediction. The output of each model was a vector of 10 values, each of which corresponded to the predicted probability of a category. A value of 1 indicates that the model correctly identified the category with the highest confidence; non-zero values for the incorrect category reflect that the model was confused by some similar category during prediction. In this way,

the recognition accuracy of each model in the succulent plant classification task and its ability to distinguish similar categories can be analyzed. The experimental results further show that the lightweight GoogLeNet model exhibits stronger discriminative ability in many cases, with higher probability values for the correct category and closer to zero probability values for other categories, which verifies its advantages in improving accuracy and anti-confusion. The design of this experiment effectively evaluates the performance of the lightweight GoogLeNet model in multi-class image classification tasks, providing a reference for further optimizing and improving the model's performance.

Table 4. Comparison of different networks for succulent plant classification.

Class	AlexNet	VGG16	GoogLeNet	Lightweight GoogLeNet
<i>Senecio haworthii</i>	3.05×10^{-8}	3.65×10^{-5}	3.87×10^{-10}	0.0
<i>Crassula perforata</i>	8.29×10^{-7}	1.66×10^{-4}	2.68×10^{-9}	2.38×10^{-7}
<i>Crassula obliqua</i> 'Gollum'	1.34×10^{-11}	6.56×10^{-6}	3.45×10^{-13}	0.0
<i>Echeveria chihuahuensis</i>	1.15×10^{-6}	2.74×10^{-4}	7.4×10^{-8}	4.26×10^{-5}
<i>Echeveria agavoides</i> 'Ebony'	5.29×10^{-10}	9.03×10^{-6}	4.6×10^{-11}	0.0
<i>Graptoveria</i> 'Opalina'	1.0	0.999	1.0	1.0
<i>Haworthia cooperi</i> var. <i>truncata</i>	2.98×10^{-6}	3.16×10^{-4}	1.78×10^{-8}	1.19×10^{-7}
<i>Haworthia truncata</i>	9.38×10^{-9}	8.83×10^{-5}	3.77×10^{-8}	5.96×10^{-8}
<i>Sedum burrito</i>	1.46×10^{-9}	1.16×10^{-5}	1.77×10^{-12}	0.0
<i>Sedum rubrotinctum</i>	5.27×10^{-9}	9.15×10^{-6}	1.29×10^{-11}	0.0

By comparing the performance of four models, AlexNet, VGG16, GoogLeNet, and lightweight GoogLeNet, in the succulent classification task, this experiment aims to highlight the advantages of lightweight GoogLeNet, especially in terms of classification accuracy and exclusion of incorrect categories. Lightweight GoogLeNet demonstrated excellent classification performance while maintaining a low computational overhead (as shown in Figure 8).

First, in the *Graptoveria* 'Opalina' category, the prediction probability of all the models was 1.0, which indicates that all the models successfully recognized the images in this category. However, of more comparative significance is the model performance in the other categories. In most categories, the lightweight GoogLeNet maintained comparable prediction accuracy to the original GoogLeNet, and performed particularly well when excluding the wrong categories. For example, in both the *Crassula obliqua* 'Gollum' and *Sedum burrito* categories, the lightweight GoogLeNet's prediction probability of 0.0 accurately excluded these categories, showing its excellent exclusion of non-target categories. This result is close to the GoogLeNet model, but the lightweight GoogLeNet requires significantly fewer computational resources, demonstrating its efficiency advantage. In contrast, AlexNet and VGG16 did not perform as well as lightweight GoogLeNet in predicting certain categories. e.g., in the *Haworthia truncata* category, lightweight GoogLeNet exhibited a prediction close to 0.0, indicating that it can accurately exclude this category, whereas the prediction probability of 8.83×10^{-5} for VGG16 shows a certain degree of categorization uncertainty. This indicates that lightweight GoogLeNet reduces the possibility of misclassification while maintaining high accuracy.

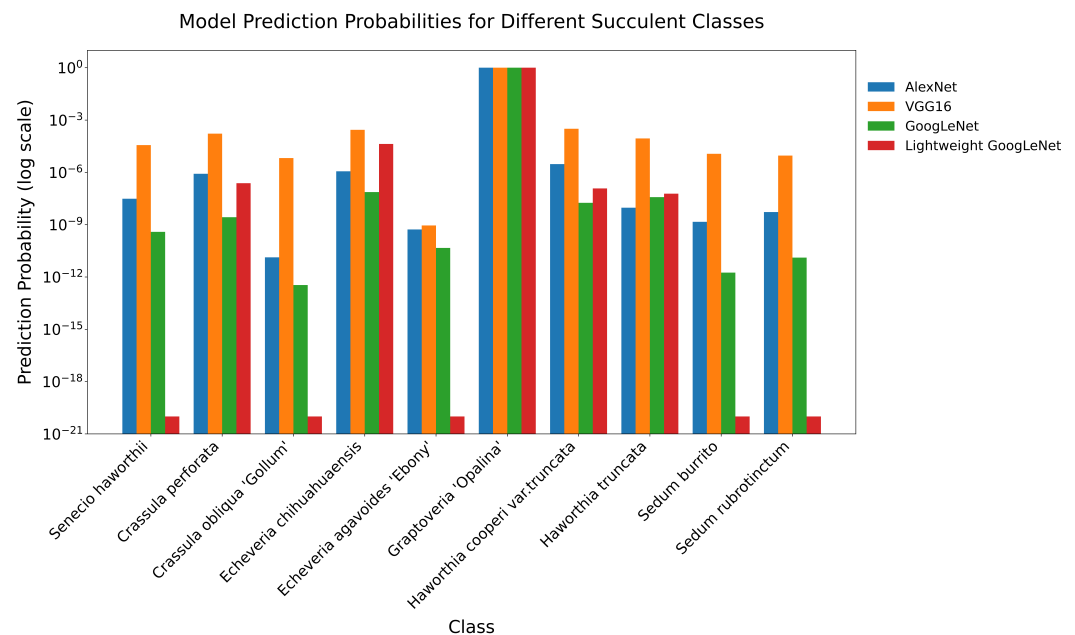


Figure 8. Validation accuracy and loss curves for each network Model.

Overall, the lightweight GoogLeNet not only maintains similar prediction performance to the original GoogLeNet in several categories, especially in the categories of *Senecio haworthii* and *Echeveria agavoides* 'Ebony', but its classification exclusion effect is much better than that of AlexNet and VGG16. Meanwhile, it significantly reduces computational costs, which makes the model more practical in real applications. Through this experimental comparison, lightweight GoogLeNet demonstrates the ability to optimize computational resources while maintaining model accuracy, making it suitable for deployment in resource-constrained environments.

Therefore, lightweight GoogLeNet not only retains the high-precision performance of GoogLeNet but also highlights its potential value in real-world applications through higher efficiency and accurate category exclusion. This advantage provides strong support for deployment in large-scale image classification tasks.

4. Discussion

Although the GoogLeNet classification and recognition method based on the lightweight and CBAM attention module proposed in this paper has achieved significant classification results, there is still room for further optimization and exploration. The following summarizes possible future improvement directions.

4.1. Parameter Tuning

In this study, the hyperparameter selection of the model mainly relied on experience and limited tuning due to limitations in training resources and time. Although good accuracy has been achieved in the experiments, there is still potential for further improvement in model performance. Future work can focus on the systematic and automated tuning of hyperparameters, such as using Bayesian optimization or evolutionary algorithms to explore the parameter space efficiently and thus improve the model's generalization ability on different datasets. Meanwhile, targeted tuning for task-specific datasets can help improve the model's performance in fine-grained classification tasks.

4.2. Data Enhancement

Data augmentation techniques have been widely used to enhance the performance of deep learning models, especially in fine-grained classification tasks where they play an important role. Data augmentation techniques were used in this study, but in the future, we can focus on exploring unsupervised data augmentation methods to enhance the generalization ability of the models. For example, Generative Adversarial Networks (GANs) can be employed to generate training samples with higher diversity, or data transformation strategies in self-supervised learning can be used to enhance data richness. In addition, reasonable data augmentation can help to solve data imbalance problems and further enhance the robustness of the model and its ability to adapt to complex scenarios.

4.3. Improvements in Attention Mechanisms

Currently, attention mechanisms are widely used in computer vision to highlight the model's attention to key features. In this paper, the CBAM attention module is used, but other more efficient attention mechanisms, such as Self-Attention or Vision Transformer (ViT), can be explored in the future to further enhance the quality of feature representation. Especially on resource-limited devices, finding an efficient attention module with low computational overhead is an important direction for future research. In addition, combining multiple attention mechanisms to achieve collaborative modeling of different feature layers may also improve the overall performance of the model.

4.4. Model Lightweighting

Some degree of model lightweighting was achieved in this study by reducing the number of Inception modules (from nine to seven) and introducing depth-separable convolution, as well as adding batch normalization (BN) operations after each convolutional layer. However, how to maintain the performance of a model while compressing it remains an important research question. In the future, knowledge distillation techniques can be introduced to efficiently migrate the knowledge from large models to lightweight models, thus improving the classification accuracy of the models while significantly reducing the number of parameters. In addition, techniques such as pruning and quantization can be explored to further reduce the demand for computational resources and make the model suitable for real-time applications on low-power devices.

In summary, this paper has enhanced the attention mechanism of GoogLeNet and made it more lightweight when deployed in classification recognition tasks, and achieved certain results. However, future research can further improve the accuracy and adaptability of the model through more refined parameter tuning, reasonable data enhancement strategies, innovative attention mechanism design, and more efficient model lightweight methods. It is hoped that this study can provide a useful reference for such future research.

5. Conclusions

In this paper, a GoogLeNet improvement method based on a lightweight and CBAM attention mechanism is proposed to cope with the dataset limitations and visual complexity problems in succulent plant classification tasks. By reducing the number of Inception modules and introducing depth-separable convolution and a CBAM attention module, the model effectively reduces computational complexity, while significantly improving classification accuracy and generalization ability. Experimental results show that the improved model achieves an accuracy of 98.5% on the validation set without data augmentation, and further improves to 99.4% when data augmentation techniques are applied. The improved model effectively alleviates the overfitting problem and demonstrates good computational efficiency and adaptability in resource-constrained environments. Despite the satisfactory

results achieved in this study, future research still needs to explore parameter tuning, data enhancement, and further lightweighting. By adopting automated hyperparameter optimization strategies, introducing unsupervised data enhancement methods, and leveraging knowledge distillation techniques, the model's performance is expected to be further improved. In summary, the proposed method not only provides a practical solution for the succulent plant classification task but also offers valuable insights and references for other fine-grained classification tasks, demonstrating significant application value and development potential.

Author Contributions: X.T.: Writing—original draft, Writing—review and editing, Making major revisions; Z.L.: Supervision, Writing—review and editing; F.L.: Writing—review and editing, Making major revisions, data curation. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Central-Guided Local Science and Technology Development Fund Project “China-Myanmar Cross-border Logistics and Trade Integration Service Platform” (Grant No. 202307AB110009) and Yunnan Provincial Major Science and Technology Project (Grant No. 202002AD080002).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

References

1. Chen, R.; Pan, L.; Li, C.; Zhou, Y.; Chen, A.; Beckman, E. An Improved Deep Fusion CNN for Image Recognition. *Comput. Mater. Contin.* **2020**, *65*, 1691–1706. Available online: <https://www.techscience.com/cmc/v65n2/39900> (accessed on 8 September 2024).
2. Fan, X.; Feng, X.; Dong, Y.; Hou, H. COVID-19 CT Image Recognition Algorithm Based on Transformer and CNN. *Displays* **2022**, *72*, 102150. [CrossRef] [PubMed]
3. Valarmathi, G.; Suganthi, S. U.; Subashini, V.; Janaki, R.; Sivasankari, R.; Dhanasekar, S. CNN Algorithm for Plant Classification in Deep Learning. *Mater. Today Proc.* **2021**, *46*, 3684–3689. [CrossRef]
4. Hu, J.; Chen, Z.; Yang, M.; Zhang, R.; Cui, Y. A Multi-Scale Fusion Convolutional Neural Network for Plant Leaf Recognition. *IEEE Signal Process. Lett.* **2018**, *25*, 853–857. [CrossRef]
5. Lee, S.H.; Chan, C.S.; Wilkin, P.; Remagnino, P. Deep-plant: Plant identification with convolutional neural networks. In Proceedings of the IEEE International Conference on Image Processing, Quebec City, QC, Canada, 27–30 September 2015; pp. 452–456. Available online: <https://dergipark.org.tr/en/pub/pajes/issue/65491/1014525> (accessed on 8 September 2024)
6. Kumar, N.; Belhumeur, P.N.; Biswas, A.; Jacobs, D.W.; Kress, W.J.; Lopez, I.C.; Soares, J.V. Leafsnap: A computer vision system for automatic plant species identification. In *Proceedings of the 12th European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 502–516. [CrossRef]
7. Pereira, C.S.; Morais, R.; Reis, M.J.C.S. Deep learning techniques for grape plant species identification in natural images. *Sensors* **2019**, *19*, 4850. [CrossRef] [PubMed]
8. Tan, J.W.; Chang, S.W.; Abdul-Kareem, S.; Yap, H.J.; Yong, K.T. Deep learning for plant species classification using leaf vein morphometric. *IEEE/Acm Trans. Comput. Biol. Bioinform.* **2020**, *17*, 82–90. [CrossRef] [PubMed]
9. Ukwuoma, C.C.; Zhiguang, Q.; Bin Heyat, M.B.; Ali, L.; Almaspoor, Z.; Monday, H.N. Recent advancements in fruit detection and classification using deep learning techniques. *Math. Probl. Eng.* **2022**, *2022*, 9210947. [CrossRef]
10. Hasan, M. A.; Bhargav, T.; Sandeep, V.; Reddy, V. S.; Ajay, R. Image Classification Using Convolutional Neural Networks. *Int. J. Mech. Eng. Res. Technol.* **2024**, *16*, 173–181.
11. Sun, Y.; Li, D.; Wang, Q.; Yuan, Y.; Hu, Z.; Hua, X.; Jiang, Y.; Zhu, Y.; Fu, Y. Scalable Quantum Convolutional Neural Network for Image Classification. *Phys. Stat. Mech. Its Appl.* **2025**, *657*, 130226. [CrossRef]
12. Liu, Y.; Xiao, F.; Zheng, X.; Deng, W.; Ma, H.; Su, X.; Wu, L. Integrating Deformable CNN and Attention Mechanism into Multi-scale Graph Neural Network for Few-shot Image Classification. *Sci. Rep.* **2025**, *15*, 1306.

13. Zhao, B.; Wu, C.; Zhang, X.; Sun, R.; Jiang, Y. Target Grasping Network Technology of Robot Manipulator Based on Attention Mechanism. *J. Jilin Univ.* **2024**, *54*, 3423–3432. [\[CrossRef\]](#)
14. Yang, J.; Hao, Z. Research progress on the characteristics and application value of succulent plants. *Mod. Hortic.* **2020**, *43*, 112–114.
15. Hu, Y. Investigation on the Application of Crassulaceae Succulent Plants. Master's Thesis, Central South University of Forestry and Technology, Changsha, China, 2013.
16. Xie, W. New scope and new classification of succulent plants. *China Flowers Bonsai* **2012**, *6*, 14–15. (In Chinese)
17. Yun, S.; Han, D.; Oh, S.J.; Chun, S.; Choe, J.; Yoo, Y. Cutmix: Regularization Strategy to Train Strong Classifiers with Localizable Features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6023–6032.
18. Cubuk, E.D.; Zoph, B.; Shlens, J.; Le, Q.V. Randaugment: Practical Automated Data Augmentation with a Reduced Search Space. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 13–19 June 2020; pp. 702–703.
19. Santos, C.F.G.D.; Roder, M.; Passos, L.A.; Papa, J.P. Maxdropoutv2: An Improved Method to Drop Out Neurons in Convolutional Neural Networks. In *Iberian Conference on Pattern Recognition and Image Analysis*; Springer International Publishing: Cham, Switzerland, 2022; pp. 271–282.
20. Dai, Z.; Chen, M.; Gu, X.; Zhu, S.; Tan, P. Batch Dropblock Network for Person Re-Identification and Beyond. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3691–3701.
21. Zhang, C. B.; Jiang, P.T.; Hou, Q.; Wei, Y.; Han, Q.; Li, Z.; Cheng, M.M. Delving Deep into Label Smoothing. *IEEE Trans. Image Process.* **2021**, *30*, 5984–5996. [\[CrossRef\]](#) [\[PubMed\]](#)
22. Li, W.; Dasarathy, G.; Berisha, V. Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics. *Proc. Mach. Learn. Res.* **2020**, *108*, 1453–1463. Available online: <https://proceedings.mlr.press/v108/li20e.html> (accessed on 8 September 2024).
23. Feng, Y.; Liang, S.; Tong, H. Research on Classification and Recognition Method of Succulent Plants Based on Contrastive Learning. *J. Henan Agric. Sci.* **2023**, *52*, 154–162. [\[CrossRef\]](#)
24. Wen, Y.; Zhang, S.-P.; Huang, M.; Hu, Z.; Tao, L. Small Sample Succulent Image Classification Based on Transfer Learning. In Proceedings of the 2024 3rd Asia Conference on Algorithms, Computing and Machine Learning, Shanghai, China, 22–24 March 2024; ACM: New York, NY, USA, 2024; pp. 424–429. [\[CrossRef\]](#)
25. Obeso, A. M.; Benois-Pineau, J.; Vázquez, M. S. G.; Acosta, A. Á. Visual vs Internal Attention Mechanisms in Deep Neural Networks for Image Classification and Object Detection. *Pattern Recognit.* **2022**, *123*, 108411. [\[CrossRef\]](#)
26. Ouyang, D.; He, S.; Zhang, G.; Luo, M.; Guo, H.; Zhan, J.; Huang, Z. Efficient Multi-Scale Attention Module with Cross-Spatial Learning. In Proceedings of the ICASSP 2023–2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2023; pp. 1–5. [\[CrossRef\]](#)
27. Gao, T.; Fisch, A.; Chen, D. Making Pre-Trained Language Models Better Few-Shot Learners. *arXiv* **2020**, arXiv:2012.15723. [\[CrossRef\]](#)
28. Wu, Q.; He, M.; Liu, Z.; Liu, Y. Multi-Scale Spatial–Spectral Residual Attention Network for Hyperspectral Image Classification. *Electronics* **2024**, *13*, 262. [\[CrossRef\]](#)
29. 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), Munich, Germany, 8–14 September 2018; pp. 3–19. Available online: <https://arxiv.org/abs/1807.06521> (accessed on 8 September 2024).

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.