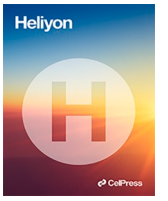
Heliyon 10 (2024) e33447

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon



Research article

A lightweight CNN model for pepper leaf disease recognition in a human palm background

Youyao Fu [a,b,*], Linsheng Guo [c], Fang Huang [a]

[a] School of Electronic & Information Engineering, Taizhou University, Taizhou, 318000, China
[b] Nuclear Technology Application Engineering Research Center of the Ministry of Education, Nanchang, 330013, China
[c] School of Geophysics and Measurement-Control Technology, East China University of Technology, Nanchang, 330013, China

ARTICLE INFO

Keywords:
Disease recognition
Pepper leaf disease
Human palm background
GGM-VGG16
Ghost module
Application

ABSTRACT

The identification of pepper leaf diseases is crucial for ensuring the safety and quality of pepper yield. However, existing methods heavily rely on manual diagnosis, resulting in inefficiencies and inaccuracies. In this study, we propose a lightweight convolutional neural network (CNN) model for recognizing pepper leaf diseases and subsequently develop an application based on this model. To begin with, we acquired various images depicting healthy leaves as well as leaves affected by viral diseases, brown spots, and leaf mold. It is noteworthy that these images were captured against a background of human palms, which is commonly encountered in field conditions. The proposed CNN model adopts the GGM-VGG16 architecture, incorporating Ghost modules, global average pooling, and multi-scale convolution. Following training with the collected image dataset, the model was deployed on a mobile terminal, where an application for pepper leaf disease recognition was developed using Android Studio. Experimental results indicate that the proposed model achieved 100 % accuracy on images with a human palm background, while also demonstrating satisfactory performance on images with other backgrounds, achieving an accuracy of 87.38 %. Furthermore, the developed application has a compact size of only 12.84 MB and exhibits robust performance in recognizing pepper leaf diseases.

1. Introduction

China stands as the predominant producer and consumer of pepper worldwide, boasting a pepper planting area that accounts for approximately 40 % of the global total [1]. Nevertheless, shifts in the global climate and soil environment have precipitated a surge in the occurrence of pepper diseases, exerting a substantial influence on both the yield and quality of pepper. Pepper leaves exhibit a pronounced susceptibility to diseases, with the majority of pepper ailments primarily afflicting the foliage. Presently, the identification of pepper leaf diseases heavily hinges on the subjective expertise of plant protection experts or growers, presenting formidable challenges in achieving the desired levels of efficiency and accuracy [2].

To address this challenge, researchers have proposed several conventional machine learning approaches for the detection of pepper diseases. Kerim et al. [3] utilized a spectral reflectometer to capture the reflected light from pepper leaves, subsequently applying the K-nearest neighbor method for the classification of two types of pepper leaf diseases. Meanwhile, Schor et al. [4] devised a methodology for pepper disease identification grounded on principal component analysis, yielding accuracy rates of 95 % and 90 % for

* Corresponding author. School of Electronic & Information Engineering, Taizhou University, Taizhou 318000 China.
E-mail addresses: fuyouyao828@126.com (Y. Fu), 474101480@qq.com (L. Guo), tiny4414@126.com (F. Huang).

https://doi.org/10.1016/j.heliyon.2024.e33447
Received 26 December 2023; Received in revised form 20 June 2024; Accepted 21 June 2024
Available online 22 June 2024

powdery mildew and viral infections, respectively. Despite the encouraging outcomes demonstrated by these conventional machine learning techniques, their feature extraction procedures often entail labor-intensive and inconsistent processes, largely attributable to the requirement of manual intervention [5–8].

In recent years, Convolutional Neural Networks (CNNs) have garnered widespread adoption in the domain of plant disease recognition owing to their exceptional feature extraction and learning capabilities [9–12]. Liang et al. [9] devised a recognition model for diagnosing rice blast using CNN and subsequently conducted comparative analyses with other models. The comparative results illustrated the superior recognition prowess of their model, boasting an accuracy rate exceeding 95 % when juxtaposed with alternative approaches. Meanwhile, Shin et al. [11] devised six convolutional neural network models aimed at identifying powdery mildew and persistent fungal infections in strawberries, achieving an accuracy of 98.11 % utilizing the ResNet50-based model. Additionally, Zeng et al. [12] proposed a population multi-scale attention network-based model for rubber disease recognition, yielding an accuracy of 98.06 %. Furthermore, Yadav et al. [13] introduced a CNN model tailored specifically for detecting peach leaf bacterial disease, achieving an impressive accuracy rate of 98.75 %. Yu et al. [14] employed an enhanced ResNet18-based model to detect soybean diseases, achieving an accuracy of 98.49 %. Liu et al. [15] introduced a CNN model utilizing Kiwi-Inception architectures and a dense connectivity approach for detecting kiwifruit leaf diseases. This novel design surpassed GoogLeNet by 2.29 % and ResNet-20 by 9.51 %, achieving an overall accuracy of 98.54 %. Thangaraj et al. [16] proposed an enhanced algorithm derived from Xception to distinguish between tomato leaf diseases, achieving an accuracy rate of 99.55 %. Zhao et al. [17] presented a CNN model incorporating an enhanced convolutional block attention module for diagnosing diseases in tomatoes, potatoes, and corn. The model achieved accuracies of 95.20 %, 99.43 %, and 98.44 % for tomatoes, potatoes, and maize, respectively. Chakraborty et al. [18] investigated four deep-learning models trained on the PlantVillage dataset to recognize potato leaf diseases. They found that among these models, VGG16 attained the highest accuracy rate of 97.89 %. Zeng et al. [19] proposed an SKPSNet-50 model to accurately recognize corn leaf diseases in natural scene images. The experimental results indicated that the proposed model achieved commendable recognition outcomes with an average accuracy of 92.9 %. Dong et al. [20] devised a modified AlexNet model tailored specifically for identifying pests and strawberry diseases. The model outperformed other models, achieving a recognition accuracy of 97.35 %

Previous research has showcased the efficacy of Convolutional Neural Networks (CNNs) in plant disease recognition. However, the intricate network architectures of these models have limited their usability on mobile devices. Addressing this challenge, scholars have devised lightweight CNNs tailored specifically for plant disease identification. Pandi et al. [21] introduced a DCNN (Diluted Convolutional Neural Network) model integrated with a Global Average Pooling (GAP) function for discerning four rice diseases. This model is formulated by replacing the convolution kernel of a conventional CNN with a diluted convolution kernel along with the incorporation of a Global Average Pooling (GAP) function. Experimental results showcased that the proposed model enhanced the training accuracy by 5.49 % compared to the traditional CNN model. Bao et al. [22] introduced the SimpleNet model, leveraging deep separable convolution for wheat ear disease detection, achieving an impressive accuracy of 94.1 % on the test dataset. In a similar vein, Tang et al. (2020) proposed a lightweight model based on ShuffleNet for grape disease identification, boasting an accuracy of 99.14 %. Kamal et al. [23] utilized a modified version of MobileNet to detect plant diseases in the PlantVillage dataset, achieving an outstanding recognition accuracy of 98.34 %. Barman et al. [24] conducted staged detection of citrus leaf diseases employing MobileNet and SSCNN, ensuring both recognition accuracy and efficiency of a lightweight model. Zeng et al. [25] applied the lightweight LDSNet to identify corn disease images in a public dataset, attaining a recognition accuracy of 95.4 %, surpassing other classical lightweight models. Chen et al. [26] employed the MobileNetV2 model for rice disease recognition in complex backgrounds, yielding an average accuracy of 98.48 %. Lastly, Zhong et al. [5] devised a lightweight model explicitly tailored for tomato leaf disease recognition. The model integrates a deep convolution network, Phish module, and light residual module, showcasing promising recognition outcomes on public datasets. Bi et al. [27] applied the MobileNet model to precisely identify two types of apple diseases, with the test results showcasing recognition performance comparable to traditional convolutional network models. Sharma et al. [28] introduced the versatile DLMC-Net model specifically tailored for the detection of multiple plant diseases. The model's lightweight architecture is achieved through the utilization of depth-separable convolution. The findings indicate that the proposed model outperforms all considered models, achieving accuracies of 92.34 %, 99.50 %, 93.56 %, and 96.56 % for cucumber, grape, citrus, and tomato, respectively. Naik et al. [29] devised a lightweight convolutional neural network model (SECNN) based on squeezing and excitation. The model underwent initial testing using the Chilli leaf dataset, demonstrating accuracies of 98.63 % and 99.12 %, respectively.

In summary, the studies mentioned above have implemented CNN models for plant disease recognition systems to classify plant images. These images are categorized into two types based on their background: natural background images and pure background images. Natural background images are captured under plants' natural growth conditions, often resulting in complex backgrounds characterized by high levels of uncertainty. Conversely, pure background images are captured against a stable backdrop, such as a tabletop or the ground, facilitating easier and more reliable plant disease identification. However, prior research heavily relied on public datasets consisting of individualistic images with pure backgrounds, hindering replication in the field and leading to their insufficient generalization performance. Furthermore, most current lightweight models deviate from the traditional convolutional structure, resulting in poor model performance. To tackle these challenges, we propose a lightweight architecture named GGM-VGG16 for identifying pepper leaf diseases with a human palm background. The key contributions and innovations are elaborated upon below.

(1) We compiled a database of pepper leaf images featuring human palms as the background, utilizing these images to train a model for recognizing pepper leaf diseases. It's worth noting that the human palm background represents the most reproducible backdrop in real-world settings, ensuring the model's robust generalization performance.
(2) We devised a lightweight version of the VGG16 model, termed GGM-VGG16, tailored for the identification of pepper leaf diseases. This model's foundational architecture is derived from the classical VGG16 network, guaranteeing robust feature

extraction capabilities. In the initial layer of the VGG16 network, we integrated a multi-scale convolution operation to capture features across various scales. Additionally, we replaced the conventional convolution module and fully connected layer of VGG16 with a ghost module and a global average pooling layer, thereby reducing the network's parameters.

(3) We developed a mobile application for recognizing pepper leaf diseases based on the proposed GGM-VGG16 model. Employing TensorFlow Lite, we deployed the GGM-VGG16 model on mobile devices, and an application interface was developed using Android Studio. With a modest size of only 12.84 MB, the application can be conveniently installed on farmers' mobile phones, facilitating effortless identification of pepper leaf diseases.

(4) Extensive experimentation on diverse datasets of pepper leaf diseases has demonstrated the superior performance of GGM-VGG16 compared to other prevalent lightweight CNN models. Moreover, even when benchmarked against state-of-the-art disease recognition methods proposed in recent years for images with pure backgrounds, GGM-VGG16 remains highly competitive.

The paper is structured as follows: Section 2 offers a comprehensive overview of the materials and methods employed in this study, encompassing aspects such as data collection, labelling procedures, and the architectural framework of the model. Section 3 delineates the experimental outcomes and subsequent discussion, comprising insights into the experimental setup, performance metrics, and a comparative examination of the results. Section 4 delves into the elaboration of an application tailored for the recognition of pepper leaf diseases. Section 5 encapsulates the concluding remarks. Finally, Section 6 outlines the directions for our future research work.

2. Materials and methods

2.1. Image datasets

2.1.1. Data acquisition

Brown spot, leaf mold, and viral diseases are three prevalent leaf ailments affecting peppers, exerting notable effects on their

Fig. 1. Several pepper leaf images with human palm background. The images in (a), (b), (c), and (d) respectively depict pepper leaves with brown spot, healthy, leaf mold, and viral disease.

growth, development, and yield. Brown spot arises from bacterial or fungal infections, manifesting as expanding brown spots on the leaves. Leaf mold, attributed to fungal infections, manifests as gray to brown patches on leaf surfaces. Conversely, viral diseases stem from viral infections, commonly presenting symptoms such as leaf yellowing, deformation, and curling.

This study is dedicated to classifying the aforementioned three leaf diseases of peppers along with healthy pepper leaves. The image data utilized in this study were gathered from the experimental fields of Nanchang Academy of Agricultural Sciences, encompassing four image categories: brown spot, healthy leaves, leaf mold, and viral diseases. Throughout the image collection process, various mobile camera devices were employed to ensure sample diversity. Each shooting device was equipped with automatic focus, automatic white balance, and set to a 1:1 photo ratio, with original image resolutions exceeding 2000*2000 pixels. Additionally, images were captured under various lighting conditions, including indoor and outdoor environments, as well as daytime and nighttime, to enhance the adaptability of the image data. It is noteworthy that we consistently employed human palm as the background for all captured images, which is the most easily replicable background in field conditions. For this study, we gathered a total of 1262 images of pepper leaves with palm backgrounds, encompassing 376 images of brown spot, 204 images of leaf mold, 331 images of viral diseases, and 351 images of healthy leaves. Significantly, all these images underwent verification by agricultural experts, and sample images from this dataset are depicted in Fig. 1(a–d).

To assess the robustness of our model, we collected two additional datasets of pepper leaf images. One dataset comprises pepper leaf images with other people's palms as backgrounds, mainly palms of elderly individuals and children, totaling 603 images. The other dataset consists of pepper leaf images with non-palm backgrounds, totaling 2928 images. Non-palm backgrounds include tabletops, floors, and plastic film. For computational efficiency and streamlined image processing, all images were uniformly resized to 224 × 224 pixels. Sample images selected from these datasets are illustrated in Fig. 2.

2.1.2. Image augmentation

The efficacy of CNN model training heavily depends on the quantity and quality of image data. Consequently, dataset expansion has emerged as a prominent research area for enhancing model performance. Currently, two widely adopted methods for augmenting image data exist. The first method involves utilizing generative adversarial networks (GAN), which employ adversarial learning techniques to generate realistic images [30]. However, training GAN networks can be time-consuming and often encounter issues such as mode collapse, leading to the production of homogeneous and non-diverse samples [31]. Alternatively, the second method involves generating image data through various image processing techniques, such as geometric transformations and image fusion. Bearing this

Fig. 2. Several pepper leaf images with other backgrounds.

in mind, this paper adopts the latter approach and expands the pepper leaf image dataset with a human palm background using three operations: (1) rotation around the centroid of the image pixel matrix, (2) image mirroring along the horizontal and vertical axes, and (3) image fusion to adjust exposure and contrast through an API. Consequently, the original dataset was expanded to include 9038 images of pepper leaves. Fig. 3(a–d) showcases several examples of image augmentation.

2.2. GGM-VGG16

In fields like pepper disease recognition, the utilization of lightweight models has become increasingly crucial. This necessity arises from the limited computational resources of data collection devices commonly employed in pepper cultivation, such as smartphones or basic cameras. Hence, lightweight models are indispensable for achieving real-time disease recognition. Nevertheless, certain lightweight models currently in use require enhancement in their feature extraction capabilities. This is essential to ensure the effective capture of the intricate features associated with pepper leaf diseases.

To tackle this issue, we introduced the VGG network [32], renowned for its robust feature extraction capabilities achieved by stacking deep convolutional layers with small-sized kernels. This approach effectively extracts high-level features from images. Compared to some existing lightweight models, the VGG network demonstrates stronger expressive power and superior generalization performance. It can more accurately capture the features of pepper leaf diseases. In response, we developed a lightweight model, GGM-VGG16, for pepper leaf disease recognition, as depicted in Fig. 4. GGM-VGG16 adopts the backbone network structure of VGG16 but integrates a multi-scale convolutional structure in the first layer to extract pepper leaf disease features at various scales. The remaining convolutional layers are replaced with Ghost modules, effectively reducing the model's parameters. Additionally, the three fully connected layers in VGG16 are substituted with a global average pooling layer to further reduce the model's parameters.

2.2.1. Ghost module

Convolution layers typically comprise multiple convolutions, resulting in an extensive number of network parameters and significant computational expenses. Fortunately, convolution layer outputs often contain redundant features, which can be inexpensively generated using simple transformations without convolution. Building on this concept, Han et al. [33] proposed the Ghost module, a novel feature extraction module that utilizes simple linear operations to generate features, resulting in significant reduction of the

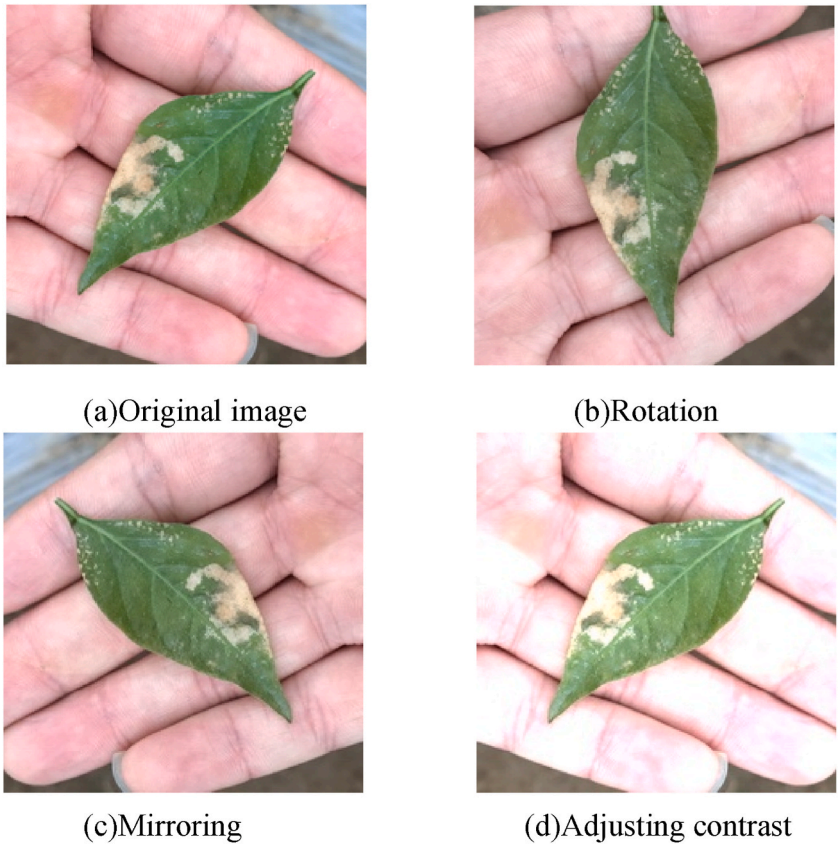


Fig. 3. Several examples of image enhancement. (a), (b), (c), and (d) represent the original image, the rotated image, the mirrored image, and the contrast-adjusted image, respectively.

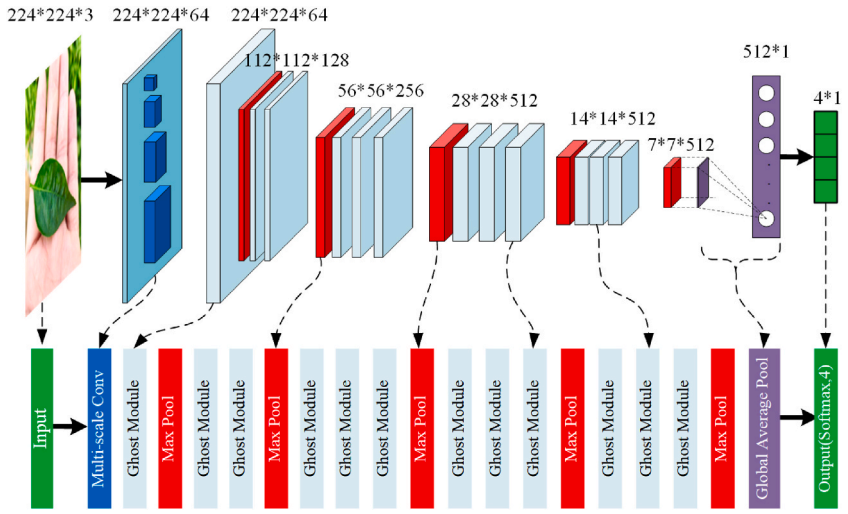


Fig. 4. The structure of the GGM-VGG16.

number of network parameters. Ghost module is a lightweight convolutional module with fewer parameters and computational complexity. Compared to traditional convolutional modules, the Ghost module can more effectively extract features and helps reduce the computational burden of the model, making it suitable for resource-constrained mobile devices. In this paper, the Ghost module is used to replace the convolution layer in VGG16. Fig. 5(a) and (b) illustrates the feature extraction processes of the convolutional layer and Ghost module, respectively. As depicted, the feature extraction of the convolutional layer involves a single stage employing only one convolution operation, while the Ghost module's feature extraction comprises two stages involving three operations: convolution, identity mapping, and linear transformation. The Ghost module's feature extraction process consists of the following steps.

Stage 1. Extracting partial features from the input using convolution. Suppose the input $\boldsymbol{X}\in R^{n\times h\times w}$, where n denotes the input channel number, w and h represent the input data's width and height, respectively, and the feature extraction process is shown in Equation (1).

$$Y = X * \mathrm{f} \tag{1}$$

Where * denotes the ordinary convolution operation. $\boldsymbol{Y}$ outputs the feature map, which has a channel number of m and a height and width of h' and w', respectively. $\boldsymbol{f}$ is the convolution filter, and the size of the convolution kernel is $l \times l$.

Stage 2. Mapping features to the output and generating new features using linear transformations. Precisely, the feature maps

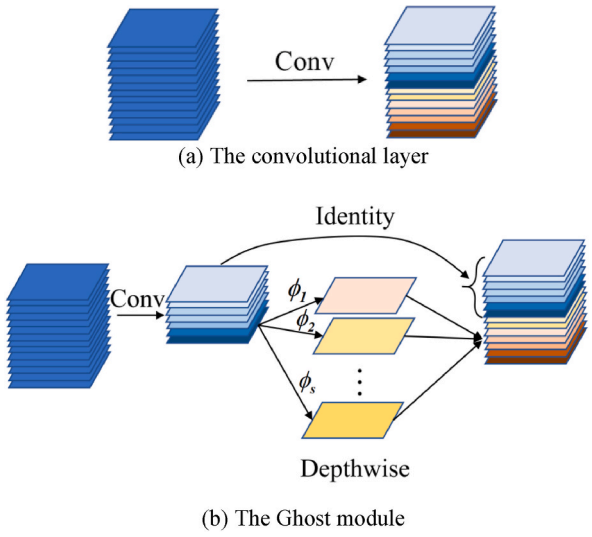


Fig. 5. The feature extraction process of the convolutional layer and Ghost module. (a) and (b) represent the feature extraction processes of the ordinary convolution and the Ghost module, respectively.

extracted in stage 1 are mapped to the output using identity mapping. Meanwhile, these extracted feature maps are also linearly transformed s times to generate new feature maps, which are then combined with the mapped feature maps to form the final output. The linear transformation operation is shown in Equation (2).

$$y_{ij} = \Phi_{i,j}(y_i) \quad \forall \quad i = 1, ..., m, j = 1, ..., s \tag{2}$$

where $\boldsymbol{y_i}$ represents the ith feature in $\boldsymbol{Y}$; $\boldsymbol{y_{ij}}$ denotes the feature map obtained after the jth linear transformation of $\boldsymbol{y_i}$; $\boldsymbol{\Phi}_{i,j}$ denotes the jth linear operation performed on $\boldsymbol{y_i}$; Note that the linear operation $\boldsymbol{\Phi}$ operates on each channel, and its computational cost is much smaller than that of ordinary convolution.

2.2.2. Global average pooling

The VGG16 architecture is characterized by three fully connected layers at its tail end. However, these layers contain an excessive number of neurons, resulting in high memory usage and an increased risk of overfitting. To address this issue, a feasible solution is to replace the fully connected layers with a global average pooling operation [34]. By computing the average value of each feature map, the global average pooling operation effectively reduces the number of model parameters, thus decreasing the risk of overfitting and improving the model's generalization ability. This operation not only reduces computational load without compromising model performance but also makes the model more suitable for deployment in resource-constrained environments, such as mobile devices. Moreover, global average pooling integrates global spatial information from the feature map, enhancing robustness against spatial transformations in images. Therefore, by substituting fully connected layers with global average pooling, we can maintain excellent model performance while also making the model more lightweight and practical for deployment. The computation of global average pooling is illustrated in Equation (3).

$$y = \frac{1}{m * n}\left[\sum_{i=0}^{m}\sum_{j=0}^{n} x_{ij}\right] \tag{3}$$

where $m*n$ represents the feature map size, $\boldsymbol{x}$ denotes the feature value, and $\boldsymbol{y}$ represents the output value of the global average pooling.

2.2.3. Multi-scale convolution layer

In convolutional neural networks, the size of the convolutional kernel significantly influences feature extraction. Larger convolutional kernels are adept at capturing coarse-grained features like edges and contours in images, whereas smaller ones excel at extracting fine-grained features such as texture. Notably, pepper brown spot disease manifests as small textured spots, while pepper leaf mold disease is characterized by contour features like leaf deformation and folding. To tackle this challenge, this paper introduces a multi-scale convolutional layer into the first layer of the VGG model. By employing multi-scale convolution in the initial layer, the model can effectively capture both detailed and global information in images, thereby enhancing its capability to identify various types of pepper leaf diseases. Specifically, the first layer of VGG16 is structured with four convolutional kernels of different scales (1x1, 3x3, 5x5, and 7x7), each comprising 16 convolutional kernels.

3. Experiments results and discussion

3.1. Experiment setup

All experiments were conducted on a computer platform running Windows 10, equipped with an NVIDIA RTX 2060 SUPER graphics card and an Intel Core i7-10700F processor. TensorFlow 2.0 served as the deep learning framework, and PyCharm was employed for software compilation. The augmented image dataset underwent partitioning into training, validation, and test sets in a 7:2:1 ratio. The VGG, MobileNet, SqueezeNet, and GGM-VGG16 models were trained using the training set. The initial learning rate was set to 0.001, progressively decreasing by 0.9 after each epoch. The number of epochs and batch size were configured as 80 and 32, respectively. Cross-entropy functioned as the loss function, while the Adam Optimizer was applied for optimizing the update parameters. Furthermore, a batch normalization operation [35] was executed on the input of each layer to mitigate gradient disappearance. Ultimately, the trained models underwent evaluation using the test set and other background datasets.

3.2. Performance metrics

In this study, five evaluation parameters—memory requirement (MR), forward propagation rate (FPR), accuracy, precision, and recall—are utilized as performance metrics to assess the model's effectiveness. MR quantifies the model's size, with a smaller value facilitating efficient operation on mobile devices. FPR denotes the time required for model inference, reflecting the model's recognition speed. Accuracy represents the ratio of correctly classified samples to the total sample count. Precision signifies the proportion of truly positive samples among those predicted as positive by the classifier. Recall indicates the proportion of successfully predicted positive samples among all truly positive samples. The formulas for calculating accuracy, precision, and recall are presented as Equation (4), Equation (5), and Equation (6), respectively.

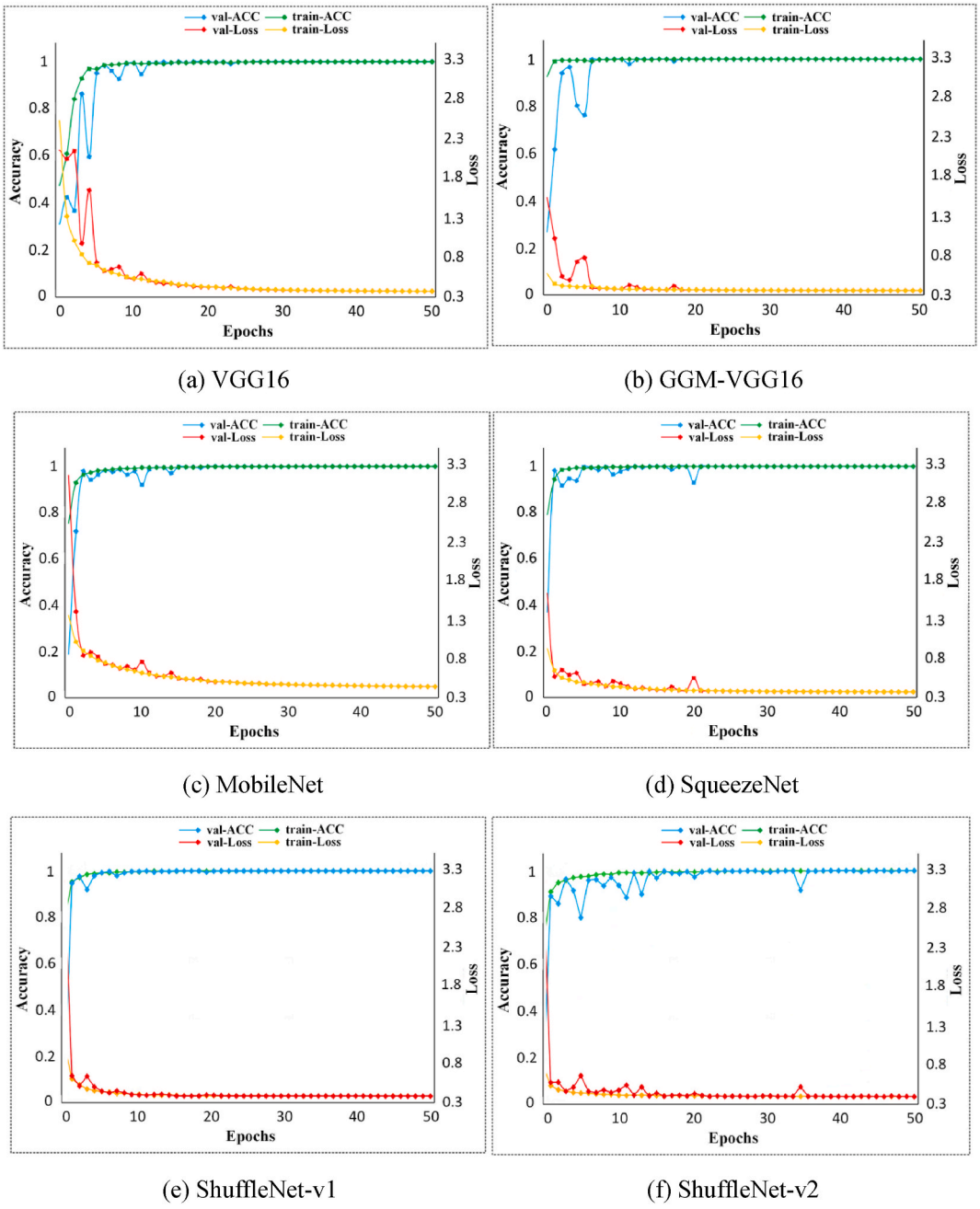


Fig. 6. The training curves of different models. (a), (b), (c), (d), (e), and (f) represent the training curves of VGG16, GGM-VGG16, MobileNet, SqueezeNet, ShuffleNet-v1, and ShuffleNet-v2, respectively.

$$Accuray = \frac{TP + TN}{TP + FP + TN + FN} \times 100\% \tag{4}$$

$$Pr\ ecision = \frac{TP}{TP + FP} \times 100\% \tag{5}$$

$$\text{Re}call = \frac{TP}{TP + FN} \times 100\% \tag{6}$$

where TP, FP, TN, and FN are the number of true-positive samples, false-positive samples, true-negative samples, and false-negative samples, respectively.

3.3. Results and discussion

3.3.1. Training results for different models

In this section, we conduct a comparative analysis of the training performance of the proposed model against other models. Fig. 6 (a–f) illustrates the training curves of several models, including VGG16, GGM-VGG16, MobileNet, SqueezeNet, ShuffleNet-v1, and ShuffleNet-v2. The figure reveals that all models attained 100 % accuracy on both the training and validation datasets. Furthermore, they exhibited minimal fluctuations during the training process and converged swiftly, indicating stable training without overfitting across all models. Specifically, in comparison to the VGG16 model, all lightweight models demonstrated superior training performance characterized by negligible fluctuations and accelerated convergence rates. This can be attributed chiefly to the simplified network architectures and efficient computational methods employed by lightweight models. Notably, both the GGM-VGG16 and ShuffleNet models exhibited remarkably rapid convergence, achieving convergence after approximately seven iterations. The expedited convergence of the GGM-VGG16 model can be primarily attributed to the utilization of batch normalization, whereas the ShuffleNet model's swift convergence is credited to its incorporation of channel shuffling and grouped convolutions. In summary, the proposed GGM-VGG16 model yields outstanding training results and showcases optimal convergence speeds.

3.3.2. Test results of different models on the test set

The performance of all aforementioned trained models was evaluated using the test set. Table 1 presents the corresponding results. Each model exhibited excellent recognition accuracy, achieving a rate of 100 %. However, some models displayed inferior performance in terms of model size and recognition speed. Particularly noteworthy was the original VGG model, notable for its substantial memory requirement of 745 MB and a relatively slow false positive rate (FPR) of 177 ms. Fortunately, the GGM-VGG16 model achieved a significant reduction in both FPR and memory requirements (MR), reducing them to only 97 ms and 11.5 MB, respectively. This enhancement surpasses the capabilities of the other lightweight models. In conclusion, the GGM-VGG16 model demonstrates superior recognition performance for pepper diseases, coupled with a smaller size and faster recognition speed compared to the other models, making it the optimal choice for mobile applications.

3.3.3. Test results of different models on other people's palm background images

Following the training of the models mentioned earlier, this study employs them to identify datasets featuring backgrounds of individuals' palms, aiming to assess the models' resilience to varying backgrounds. Typically, model training data may not encompass all possible backgrounds, leading to potential challenges when encountering backgrounds different from those in the training data during practical applications. In such cases, the model may face disruptions due to background variations, resulting in inaccurate or incomplete feature extraction and thereby affecting recognition performance. However, a model's ability to accurately identify images with diverse backgrounds indicates robust generalization performance. To underscore the impact of background variations in the test data, this study employs datasets featuring backgrounds of elderly and children's palms to evaluate the models, with the test results presented in Table 2.

As depicted in the table, all models demonstrate favorable recognition rates on datasets with backgrounds of individuals' palms, achieving accuracies, precisions, and recalls exceeding 96 %. Particularly noteworthy is the GGM-VGG16 model, which achieves 100 % recognition accuracy, showcasing remarkable resilience to palm backgrounds. This resilience can be attributed to several factors: firstly, the model was trained using palm backgrounds from diverse individuals, reducing its reliance on specific backgrounds; secondly, the overall differences in palm backgrounds are not substantial enough to significantly impact the model's performance. In

Table 1
The test results of different models on the test set.

Model	Total Number of Parameters	MR/MB	FPR/ms	Accuracy/%
VGG16	65087556	745	177 ms	100
GGM-VGG16	**973348**	**11.5**	**97 ms**	**100**
SqueezeNet	1258828	14.7	109 ms	100
MobileNet	3211076	37.2	121 ms	100
ShuffleNet-v1	1045030	12.7	101 ms	100
ShuffleNet-v2	2336224	27.5	115 ms	100

Table 2
The test results on the dataset with other people's palm backgrounds.

Model	MR/MB	Number of samples correctly recognized				Accuracy/%
		brown spot(144)	Healthy(152)	Leaf mold(163)	viral disease(144)	
VGG16	745	139	151	162	143	98.67
GGM-VGG16	**11.5**	**144**	**152**	**163**	**144**	**100**
SqueezeNet	14.7	128	152	163	138	96.35
MobileNet	37.2	142	151	159	143	98.67
ShuffleNet-v1	12.7	141	149	157	140	97.35
ShuffleNet-v2	27.5	140	151	158	142	98.01

Note: the numbers in parentheses are the total number of samples for each disease.

summary, the proposed model in this study exhibits exceptionally robust performance against palm backgrounds. Considering the ease of obtaining human palm background images in field settings, employing them for identifying pepper leaf diseases proves to be a prudent decision.

3.3.4. Test results of different models on non-palm background images

To further validate the model's effectiveness on diverse background images, we conducted tests using various non-palm background images. Tables 3 and 4 present the test results of different models, including GGM-VGG16, VGG16, SqueezeNet, MobileNet, ShuffleNet-v1, and ShuffleNet-v2. As depicted in Table 3, the recognition performance of all models significantly lags behind that of palm background data, particularly evident in VGG, SqueezeNet, MobileNet, ShuffleNet-v1, and ShuffleNet-v2, which achieved accuracies of only 76.89 %, 72.50 %, 67.78 %, 70.55 %, and 74.76 %, respectively. This discrepancy primarily arises from the disparity between non-palm and palm backgrounds, impacting the models' feature extraction process. In contrast, the GGM-VGG16 model demonstrates acceptable recognition results, with an average accuracy of 87.38 %, indicating a certain level of resilience to non-palm background data. This resilience can be mainly attributed to the adoption of the traditional VGG16 network structure and multi-scale architecture in GGM-VGG16. The VGG16 network possesses superior feature extraction capabilities compared to the lightweight models mentioned above, while the multi-scale architecture facilitates more effective extraction of both detailed and global features.

Furthermore, as indicated in Table 4, all models exhibit subpar performance in recognizing leaves affected by brown spot disease and healthy leaves, with significantly lower recall rates compared to the other two types of diseased leaves. This discrepancy may stem from the subtle features of healthy leaves, which are susceptible to interference from background characteristics. Additionally, the disease characteristics of brown spot disease share similarities with certain background noise features, such as desktop spots and film stains, making them prone to misclassification in the presence of such backgrounds. Notably, all models demonstrate the highest precision in identifying healthy leaves, likely due to the extraction of background noise characteristics similar to disease features, leading to misclassification of healthy leaves as other diseased types.

3.3.5. Comparison with the previous state-of-the-art studies

This section provides a comparative analysis of the proposed method with several state-of-the-art lightweight plant disease detection methods based on pure background images. The comparison results are presented in Table 5. Various researchers have effectively utilized lightweight models for plant disease identification in publicly available datasets, yielding improved recognition outcomes. Kamal et al. [23] employed a modified version of MobileNet on the PlantVillage dataset, achieving a recognition accuracy of 98.34 % for 10 plant diseases with a model size of only 13 M. Sharma et al. [28] introduced the multifunctional DLMC-Net, which implements a lightweight architecture using depth-separated convolution, yielding an average accuracy of 95.49 % across different plant diseases, with a model size of 25.6 M. Thakur et al. [36] developed the lightweight network VGG-ICNN, demonstrating outstanding performance on the PlantVillage dataset with a recognition accuracy of 99.16 % and a model size of 23.2 M. Yang et al. [37] proposed the novel lightweight high-precision network DGLNet for rice leaf disease identification, achieving a recognition accuracy of 99.82 % on a dataset comprising 38 plant diseases, with a model size of only 13.5 M. Baser et al. [38] proposed an enhanced CNN model for diagnosing ten tomato plant leaf diseases, achieving an average recognition accuracy of 98.19 % and a model size of 12.0 M. Naik et al. [29] curated their own chili leaf dataset and designed a lightweight convolutional neural network model (SECNN)

Table 3
The accuracy of different models on the dataset with non-palm backgrounds.

Model	MR/MB	Number of samples correctly recognized				Accuracy/%
		brown spot(763)	Healthy(723)	Leaf mold(725)	viral disease(717)	
VGG16	745	573	389	602	715	76.89
GGM-VGG16	**11.5**	**640**	**572**	**675**	**703**	**87.38**
SqueezeNet	14.7	446	370	647	686	72.50
MobileNet	37.2	447	263	720	579	67.78
ShuffleNet-v1	12.7	508	342	627	589	70.56
ShuffleNet-v2	27.5	502	363	664	660	74.76

Note: the numbers in parentheses are the total number of samples for each disease.

Table 4
Precision and recall of different models on the dataset with non-palm backgrounds.

Model	Metrics	brown spot	Healthy	Leaf mold	viral disease
VGG16	Precision	85.14 %	92.84 %	77.97 %%	67.20 %
	Recall	75.1 %	53.80 %	83.03 %	99.72 %
GGM-VGG16	**Precision**	**87.67 %**	**96.62 %**	**87.10 %**	**83.39 %**
	Recall	**83.88 %**	**79.11 %**	**93.10 %**	**98.05 %**
SqueezeNet	Precision	74.83 %	92.5 %	70.56 %	66.93 %
	Recall	58.45 %	51.18 %	89.24 %	95.69 %
MobileNet	Precision	71.29 %	86.80 %	64.29 %	65.20 %
	Recall	58.58 %	36.38 %	99.31 %	80.75 %
ShuffleNet-v1	Precision	77.20 %	87.24 %	65.52 %	64.09 %
	Recall	66.58 %	47.30 %	86.48 %	82.15 %
ShuffleNet-v2	Precision	83.39 %	90.30 %	70.34 %	67.35 %
	Recall	65.79 %	50.21 %	91.59 %	92.05 %

Table 5
Comparison results with the previous state-of-the-art studies.

References	CNN models	Nnmber. of classes	Accuracy/%	MR/MB
Kamal et al., 2019	MobileNet	10	98.34	
Sharma et al., 2023	DLMC-Net	4	95.49	25.6
Thakur et al. , 2023	VGG-ICNN	38	99.16	23.2
Yang et al., 2023	DGLNet	38	99.82	13.5
Baser et al., 2023	TomConv	10	98.19	12.0
Naik et al., 2022	SECNN	6	98.63	5.4
Proposed approach	**GGM-VGG16**	**4**	**100**	**11.5**

based on squeezing and excitation, achieving an accuracy of 98.63 % on the chili leaf dataset, with a model size of only 5.4 M.

Through comparative analysis, it becomes apparent that the lightweight model proposed in this paper achieves optimal disease recognition accuracy compared to other lightweight models, while maintaining a compact model size. Additionally, the utilization of highly diverse graphical backgrounds in previous studies poses challenges in replicating such backgrounds in real-world settings, affecting the generalization performance of the models. In contrast, this study employs image data with palm backgrounds, readily obtainable in field conditions, resulting in improved generalization performance compared to other models. Therefore, the method proposed in this paper, which involves training the GGM-VGG16 model using palm background images, provides a more practical approach for identifying other plant diseases, which can be further explored in subsequent research on plant disease identification.

3.3.6. Ablation study on the model's performance

In this section, we conduct an ablation study to assess the effectiveness of each module. Our methodology entails using the original VGG16 model as a baseline and integrating three fundamental modules: ghost, global average pooling, and multi-scale convolution. To conduct the experiment, we adopt a sequential approach, incorporating one module at a time into the baseline model, followed by training and testing. The results of the ablation experiments are presented in Table 6. From the table, it is evident that both Global-VGG16 and Ghost-VGG16 exhibit comparable recognition accuracy to that of the original VGG16 across multiple datasets. Notably, they achieve this with smaller model sizes, particularly Global-VGG16. This validates the impact of the ghost module and global average pooling in reducing the model size. Additionally, GGM-VGG16 demonstrates a notable enhancement in recognition accuracy across different datasets compared to Ghost-VGG16, Global-VGG16, and GG-VGG16. This suggests that multi-scale convolution plays a significant role in augmenting model performance.

Further analysis reveals that the ghost module is typically utilized solely as a substitute for convolutional layers in feature extraction networks, thereby limiting its effectiveness in reducing the model size. Consequently, Ghost-VGG16 retains a substantial model size of 587 M. In contrast, GGM-VGG16 boasts a significantly smaller model size of 11.5 M, primarily attributed to the

Table 6
The results of the ablation experiments.

Model	Ghost	Global average pool	Multi-scale convolution	MR/MB	Accuracy/%		
					Dataset1	Dataset2	Dataset3
VGG16				745	100	98.67	76.89
Ghost-VGG16	✓			587	100	99.67	77.97
Global-VGG16		✓		42.4	100	100	79.13
GG-VGG16	✓	✓		11.2	100	99.83	80.28
GGM-VGG16	✓	✓	✓	**11.5**	**100**	**100**	**87.38**

Note: Dataset1(The test set); Dataset2(The dataset with other people's palm backgrounds); Dataset3(The dataset with non-palm backgrounds).

integration of a global average pooling layer instead of the conventional fully connected layer. Additionally, the strategic placement of the multi-scale convolution operation in the initial layer of GGM-VGG16 notably enhances the model's adaptability to various diseases. Consequently, GGM-VGG16 achieves outstanding recognition accuracy for each specific disease, surpassing other models by a considerable margin. In summary, each module proposed in this paper significantly contributes to optimizing the model's overall performance.

3.3.7. Visualization analysis and discussion

Fig. 7(a and b) depicts the feature maps extracted by convolutional kernels of two distinct scales during training. As depicted in the figure, the 1x1 size convolutional kernel adeptly captures the intricate spot-related features of pepper leaf disease, whereas the 7x7 size convolutional kernel effectively captures the overarching shape features of the leaf. This underscores the efficacy of multi-scale convolution operations in extracting disease features across various scales.

Furthermore, we present visualized features extracted by the GGM-VGG16 model from various types of diseased leaf images and healthy leaf images, as depicted in Fig. 8. Upon inspection of the visualized features across different image categories, it becomes evident that the GGM-VGG16 model adeptly captures key characteristics such as shape, color, and texture information. This observation underscores the model's potential and efficacy in the domain of leaf disease recognition.

We conducted a detailed analysis of the visual features extracted by the GGM-VGG16 model when processing leaf disease images captured under non-palm backgrounds. The visual analysis, illustrated in Fig. 9, reveals that the GGM-VGG16 model tends to capture certain background features alongside disease-related features. Particularly in complex backgrounds with noise, the model exhibits heightened susceptibility to background interference, potentially resulting in misclassifications. This underscores the importance of addressing background interference issues in practical model applications. In future endeavors, we plan to mitigate this challenge by implementing additional data preprocessing steps or refining the model architecture.

4. Development of a pepper leaf disease recognition application

Deploying deep learning models on mobile devices and other platforms holds significant practical implications. Historically, deep learning models were predominantly executed on high-performance servers, posing challenges for implementation on mobile devices. Fortunately, Google has developed TensorFlow Lite, a versatile tool facilitating the deployment of deep learning models across various devices, including mobile terminals, embedded systems, and IoT devices. This approach eliminates the necessity for network connectivity and substantially enhances the efficiency of utilizing deep learning models. Nonetheless, given the limited memory resources on mobile devices, optimizing deep learning models to be as compact and efficient as possible is imperative. It is noteworthy that the GGM-VGG16 model proposed in this paper demonstrates high efficiency in disease recognition and boasts a compact size, measuring only 11.5 M, making it highly suitable for deployment on mobile terminals. Consequently, this paper utilizes the TensorFlow Lite tool to deploy the GGM-VGG16 model on mobile terminals, with the deployment process outlined as follows.

(1) Model preparation. Save the model file after model training and use the TensorFlow lite converter to convert the model to TFlite format.

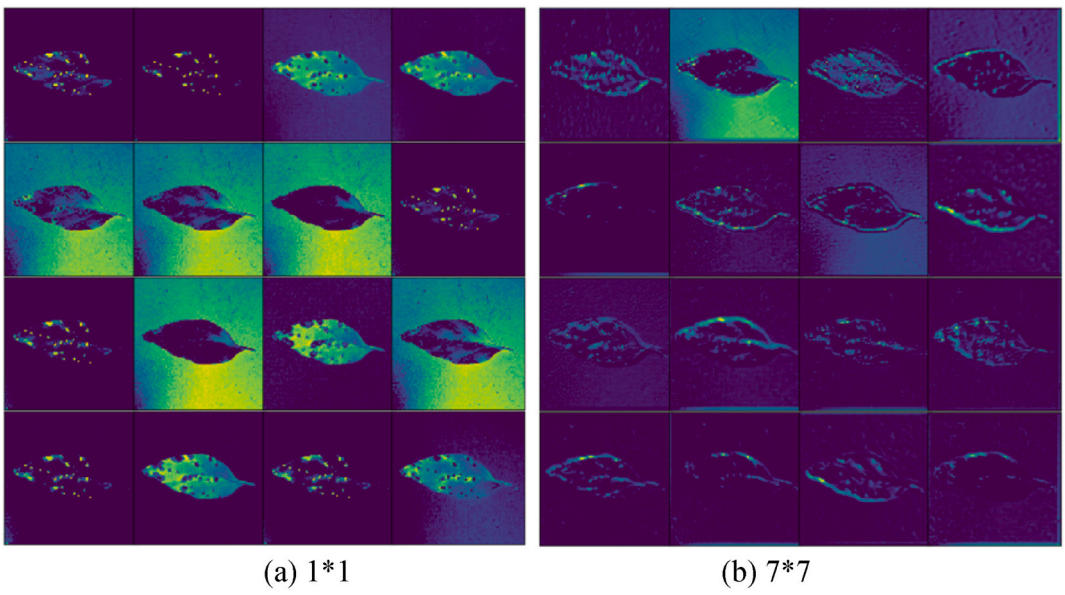


Fig. 7. Visualization feature maps after convolution with different scales. (a) and (b) represent feature maps at the 1x1 scale and the 7x7 scale, respectively.

Fig. 8. Visualization feature maps of different types of pepper leaves.

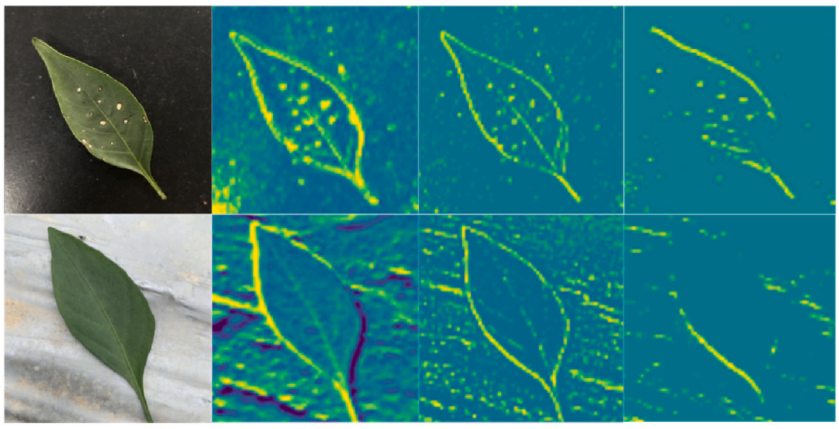


Fig. 9. Visualization feature maps of pepper leaves under non-palm backgrounds.

(2) Software configuration. Install the corresponding SDK version and configure the corresponding version of the Build Gradle.
(3) Import the model. Put the prepared TFlite model and the corresponding tag txt file into App/src/main/assets directory.
(4) Compile and load. Compile the APK file and install the application on the mobile device.

Following the outlined procedures, we have successfully deployed the GGM-VGG16 model to mobile terminals. Subsequently, utilizing Android Studio, we developed an intelligent application for identifying pepper leaf diseases. With a compact size of only 12.84 MB, the application is easily installable on any mobile terminal, significantly enhancing its accessibility for farmers conducting disease identification in the field. The recognition process of the application is depicted in Fig. 10. Initially, the input image undergoes resizing to dimensions of $224 \times 224 \times 3$, followed by normalization of its pixel values. Subsequently, the model executes an inference

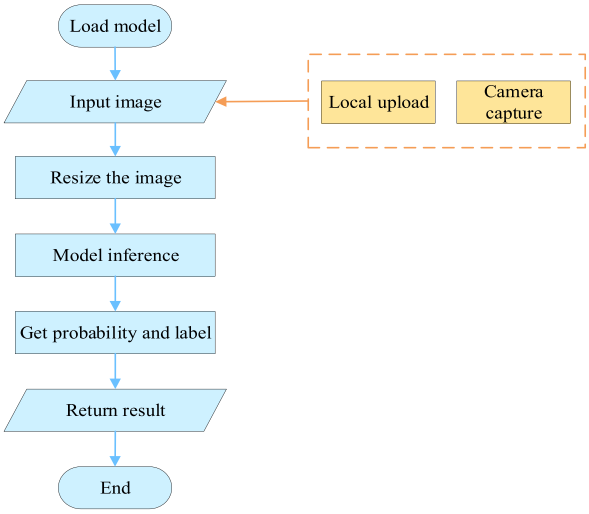


Fig. 10. The recognition process of the application.

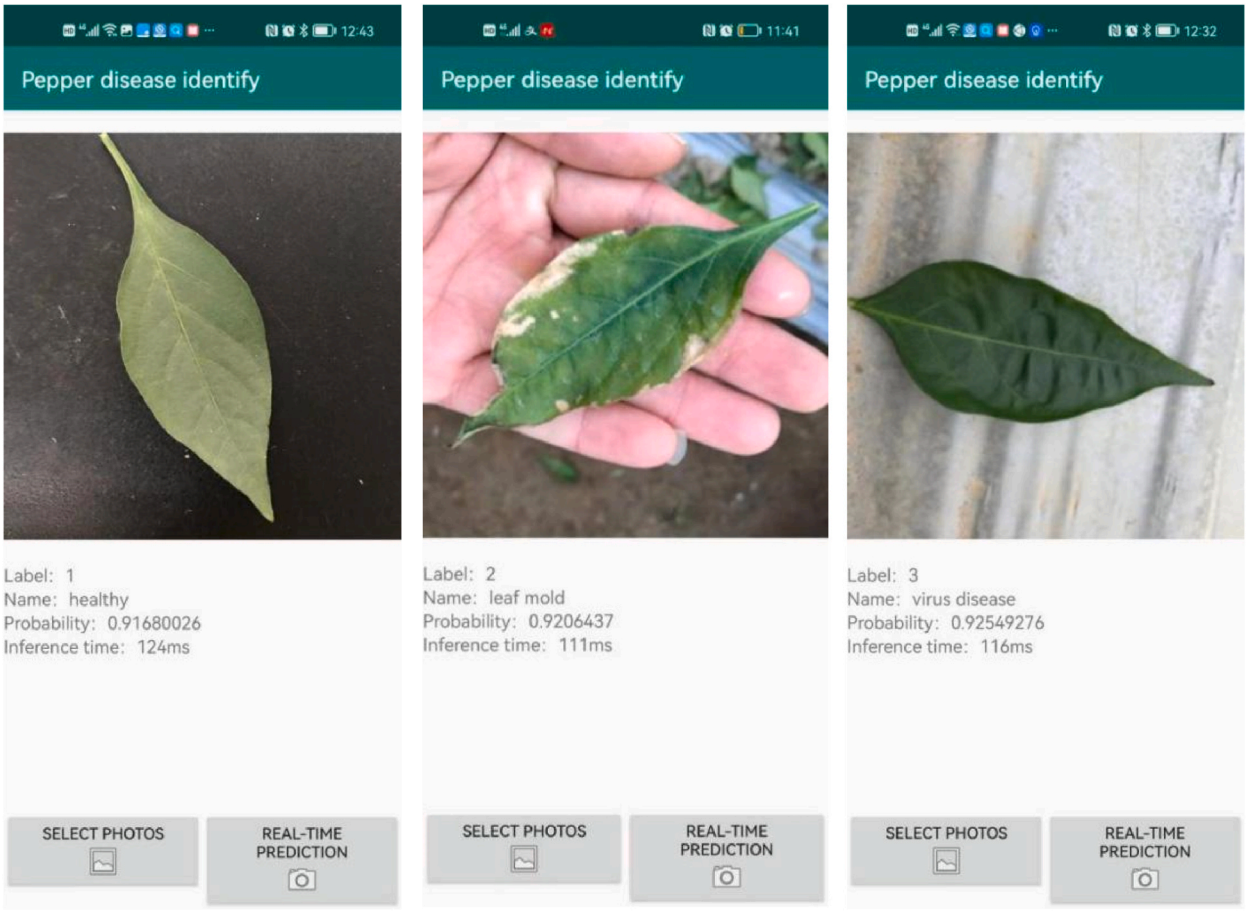


Fig. 11. Several test results of the application.

operation, predicting the maximum probability along with its corresponding label and the category of the pepper leaf disease. Importantly, the application provides users with the flexibility to select images from their photo albums or capture real-time images using the camera for pepper leaf disease identification.

We performed tests on the application, and in Fig. 11, various examples of recognition results obtained through the application are presented. The test results illustrate the application's capability to accurately identify various types of pepper leaf diseases while maintaining an exceptionally rapid recognition rate of less than 200 ms. This suggests that the application can achieve swift and efficient identification of pepper leaf diseases.

5. Conclusions

This study introduces a lightweight convolutional neural network (CNN) model for identifying pepper leaf diseases. Initially, we compiled a dataset of pepper leaf images with palm backgrounds, encompassing healthy leaves as well as those infected with leaf mold, viral diseases, and brown spot. Subsequently, we trained a lightweight model named GGM-VGG16 based on the collected dataset. Experimental results demonstrate the excellent performance of this model in recognizing pepper leaf diseases, coupled with its compact model size of only 11.5 M. Moreover, the model achieves recognition accuracies of 100 % and 87.38 % on datasets with backgrounds of other people's palms and non-palm backgrounds, respectively, indicating its robust generalization capability in practical scenarios. Following this, we deployed the GGM-VGG16 model onto mobile devices using TensorFlow Lite and successfully developed an Android Studio application for identifying pepper leaf diseases. With a file size of merely 12.84 M, the application can be effortlessly installed on mobile devices, greatly facilitating pepper farmers in monitoring and diagnosing leaf diseases. Importantly, the method proposed in this paper, based on training the GGM-VGG16 model using palm background images, can also be applied to the identification of other plant diseases, thereby enhancing the robustness of disease recognition across various plant diseases.

6. Future work

Further research endeavors will significantly advance the field. Moving forward, we will prioritize the following tasks. Firstly, we will delve into more intricate and sophisticated model architectures to further augment the model's performance and resilience. This may entail integrating additional deep learning architectures, attention mechanisms, or adaptive techniques to enhance the model's disease recognition capabilities across diverse environmental backgrounds. Secondly, we will emphasize the quality and diversity of the data to enhance the model's generalization capacity. Exploring varied data sources, including datasets from different geographical regions or various types of pepper leaf disease images, will bolster the model's adaptability and robustness. Lastly, we will explore datasets pertaining to other crops or plant diseases and endeavor to apply our model to validate its versatility across different diseases. Through model fine-tuning and techniques such as transfer learning, we aim to enhance the model's adaptability and generalization across various plant diseases.

Data availability statement

Data will be made available on request.

CRediT authorship contribution statement

Youyao Fu: Writing – review & editing, Validation, Supervision, Methodology, Funding acquisition, Formal analysis, Conceptualization. **Linsheng Guo:** Writing – original draft, Visualization, Methodology, Investigation, Data curation. **Fang Huang:** Writing – review & editing, Validation, Resources, Formal analysis, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Natural Science Foundation of Jiangxi Province(No. 20202BABL214032), Taizhou Science and Technology Plan Project(No. 22nya18), and the Nuclear Technology Application Engineering Research Center of the Ministry of Education open-funded projects(No. HJSJYB2021-10). Moreover, the authors thank Qingyou Huang and Weichai Chen from the Nanchang Academy of Agricultural Sciences and some undergraduates from the East China University of Technology for their help in collecting data.

References

[1] X.H. Liu, C. Li, S.P. Wu, L.S. Wang, First report of Carpesium abrotanoides L. as a natural host plant for pepper vein yellows virus in China, J. Plant Pathol. 102 (2020) 1369.
[2] A.H. Islam, P. Schreinemachers, S. Kumar, Farmers' knowledge, perceptions and management of pepperpepper anthracnose disease in Bangladesh, Crop Protect. 133 (2020) 105139.
[3] K. Karadağ, M.E. Tenekeci, R. Taşaltın, A. Bilgili, Detection of pepper fusarium disease using machine learning algorithms based on spectral reflectance, Sustain. Comput.-Infor. 28 (2020) 100299.

[4] N. Schor, S. Berman, A. Dombrovsky, et al., Development of a robotic detection system for greenhouse pepper plant diseases, Precis. Agric. 18 (2017) 394–409.
[5] G. Li, S.L. Wu, H.Z. Cai, et al., IncepTCN : a new deep temporal convolutional network combined with dictionary learning for strong cultural noise elimination of controlled source electr-omagnetic data, Geophysics 88 (4) (2023) E107–E122.
[6] Y. Zhong, Z.H. Teng, M.J. Tong, LightMixer: a novel lightweight convolutional neural network for tomato disease detection, Front. Plant Sci. 13 (2023) 1166296.
[7] S.N.H. Bukhari, A. Jain, E. Haq, A. Mehbodniya, J. Webber, Ensemble machine learning model to predict SARS-CoV-2 T-Cell epitopes as potential vaccine targets, Diagnostics 11 (11) (2021) 1990.
[8] S.N.H. Bukhari, A. Jain, E. Haq, A. Mehbodniya, J. Webber, Machine learning techniques for the prediction of B-Cell and T-Cell epitopes as potential vaccine targets with a specific focus on SARS-CoV-2 pathogen: a review, Pathogens 11 (2) (2022) 146.
[9] W. Liang, H. Zhang, G. Zhang, H. Cao, Rice blast disease recognition using a deep convolutional neural network, Sci. Rep. 9 (1) (2019) 2869.
[10] S. Pudumalar, S. Muthuramalingam, Hydra: an ensemble deep learning recognition model for plant diseases, Journal of Engineering Research (2023), https://doi.org/10.1016/j.jer.2023.09.033.
[11] J. Shin, Y.K. Chang, B. Heung, et al., A deep learning approach for RGB image-based powdery mildew disease detection on strawberry leaves, Comput. Electron. Agric. 183 (3) (2021) 106042.
[12] T.W. Zeng, C.M. Li, B. Zhang, R.R. Wang, et al., Rubber leaf disease recognition based on improved deep convolutional neural networks with a cross-scale attention mechanism, Front. Plant Sci. 13 (2022) 829479.
[13] S. Yadav, N. Sengar, A. Singh, A. Singh, M.K. Dutta, Identification of disease using deep learning and evaluation of bacteriosis in peach leaf, Ecol. Inf. 61 (2021) 101247.
[14] M. Yu, X.D. Ma, H.O. Guan, M. Liu, T. Zhang, A recognition method of soybean leaf diseases based on an improved deep learning model, Front. Plant Sci. 13 (2022) 878834.
[15] B. Liu, Z. Ding, Y. Zhang, D. He, J. He, Kiwifruit leaf disease recognition using improved deep convolutional neural networks, in: 2020 IEEE 44th Annual Computers, Software, and Applications Conference, 2020, pp. 1267–1272.
[16] R. Thangaraj, S. Anandamurugan, V.K. Kaliappan, Automated tomato leaf disease classification using transfer learning-based deep convolution neural network, J. Plant Dis. Prot. 128 (1) (2021) 73–86.
[17] Y. Zhao, C. Sun, X. Xu, J.G. Chen, RIC-Net: a plant disease classification model based on the fusion of Inception and residual structure and embedded attention mechanism, Comput. Electron. Agric. 193 (2022) 106644.
[18] K.K. Chakraborty, R. Mukherjee, C. Chakroborty, K. Bora, Automated recognition of optical image based potato leaf blight diseases using deep learning, Physiol. Mol. Plant Pathol. 117 (2021) 101781.
[19] W. Zeng, H. Li, G. Hu, D. Liang, Identification of maize leaf diseases by using the SKPSNet-50 convolutional neural network model, Sustain. Comput.-Infor. 35 (2022) 100695.
[20] C. Dong, Z.W. Zhang, J. Yue, L. Zhou, Automatic recognition of strawberry diseases and pests using convolutional neural network, Smart Agric. Tech. 1 (2021) 100009.
[21] S.S. Pandi, A. Senthilselvi, J. Gitanjali, K. ArivuSelvan, J. Gopal, J. Vellingiri, Rice plant disease classification using dilated convolutional neural network with global average pooling, Ecol. Model. 474 (2022) 110166.
[22] W.X. Bao, X.H. Yang, D. Liang, G.S. Hu, X.J. Yang, Lightweight convolutional neural network model for field wheat eardisease identification, Comput. Electron. Agric. 189 (2021) 106367.
[23] K. Kamal, Z. Yin, M. Wu, Z. Wu, Depthwise separable convolution architectures for plant disease classifification, Comput. Electron. Agric. 165 (2019) 104948.
[24] U. Barman, R.D. Choudhury, D. Sahu, G.G. Barman, Comparison of convolution neural networks for smartphone image based real time classifification of citrus leaf disease, Comput. Electron. Agric. 177 (2020) 105661.
[25] W. Zeng, H. Li, G. Hu, D. Liang, Lightweight dense-scale network (LDSNet) for corn leaf disease identifification, Comput. Electron. Agric. 197 (2022) 106943.
[26] J. Chen, D. Zhang, A. Zeb, Y.A. Nanehkaran, Identifification of rice plant diseases using lightweight attention networks, Expert Syst. Appl. 169 (2021) 114514.
[27] C. Bi, J. Wang, Y. Duan, B. Fu, J.-R. Kang, Y. Shi, MobileNet based apple leaf diseases identifification, Mobile Network. Appl. 25 (2022) 1–9.
[28] V. Sharma, A.K. Tripathi, H. Mittal, Dlmc-net: deeper lightweight multi-class classification model for plant leaf disease detection, Ecol. Inf. 75 (2023) 102025.
[29] B.N. Naik, R. Malmathanraj, P. Palanisamy, Detection and classification of chilli leaf disease using a squeeze-and-excitation-based CNN model, Ecol. Inf. 69 (2022) 101663.
[30] I.J. Goodfellow, J.P. Abadie, M. Mirza, B. Xu, et al., Generative Adversarial Networks, 2014 arXiv:1406.2661vol. 1.
[31] A. Srivastava, L. Valkov, C. Russell, et al., VEEGAN: Reducing Mode Collapse in Gans Using Implicit Variational Learning, 2017 arXiv:1705.07761.
[32] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014 arXiv:1409.1556.
[33] K. Han, Y. Wang, Q. Tian, et al., GhostNet: More Features from Cheap Operations, 2020 arXiv:1911.11907vol. 2.
[34] M. Lin, Q. Chen, S. Yan, Network in Network, 2014 arXiv.1312.4400.
[35] S. Loffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, 2015, pp. 448–456.
[36] P.S. Thakur, T. Sheorey, A. Ojha, Vgg-icnn: a lightweight cnn model for crop disease identification, Multimed. Tool. Appl. 82 (2023) 497–520.
[37] Y. Yang, G. Jiao, J.H. Liu, W.C. Zhao, J.H. Zheng, A lightweight rice disease identification network based on attention mechanism and dynamic convolution, Ecol. Inf. 78 (2023) 102320.
[38] P. Baser, J.R. Saini, K. Kotecha, Tomconv: an improved cnn model for diagnosis of diseases in tomato plant leaves, Procedia Comput. Sci. 218 (2023) 1825–1833.