GA-GhostNet: A Lightweight CNN Model for Identifying Pests and Diseases Using a Gated Multi-Scale Coordinate Attention Mechanism

Yu Xiao, Jie Wu*, Chi Ma

Abstract—**Plant diseases and pests represent significant threats to agricultural yields, underscoring the need for precise and prompt identification of pathogens and pests. In this paper, we propose a lightweight convolutional neural network model called Gated Asymmetrical GhostNet (GA-GhostNet), specifically designed for the automatic identification of plant diseases and pests. The model incorporates a Gated Multi-scale Coordinate Attention (GM-CA) module to filter out noise and irrelevant information in the image, while capturing location information of diseases and pests at different scales. The model also utilizes an improved feature extraction module called Asymmetrical Ghost (AG) module to enhance robustness to image flipping, as well as improve feature extraction capabilities. Additionally, the CutMix data augmentation method is employed to improve generalization ability. In extensive experimental evaluations on the IP102 pest dataset, GA-GhostNet achieved impressive results with 63.73% MRec, 67.37% MPre, 65.12% MF1, and 71.90% Acc. Moreover, through transfer learning on the Jute pest, Embrapa disease, and Apple disease datasets, the model outperforms numerous lightweight models, attaining accuracies of 99.89%, 96.97%, and 95.17%, respectively, while having only 3.73 million parameters. GA-GhostNet demonstrates high accuracy and efficiency in identifying plant diseases and pests.**

Index Terms—**Plant disease identification, Pest identification, convolutional neural network, lightweight model, feature extraction, Attention mechanism.**

I. Introduction

PLANT diseases and invasive insects result in significant annual economic losses of approximately US$220 billion and US$70 billion, respectively [1]. With the advent of deep learning, the application of neural networks for pest and disease identification has become a feasible solution. However, conventional image classification CNN models (e.g., ResNet [2] and AlexNet [3]) often impose substantial memory and computing requirements, making mobile deployment challenging. Consequently, lightweight neural network models emerge as a more viable option for this task.

Lightweight neural network architectures for this task can be classified into two primary types: those based solely on Convolutional Neural Networks (CNNs) and those that combine CNNs and Transformers. In comparison, lightweight CNN models entail fewer dot product operations. In recent years, existing lightweight CNNs have been extensively applied to pest and disease identification with remarkable success. However, their performance varies across datasets, as exemplified by GhostNet [4], which integrates Ghost modules and Squeeze-Excitation (SE) modules. Thakur et al. [5] reported an accuracy of 96.18% using GhostNet on the PlantVillage dataset but only 43.33% on the Rice dataset. This inconsistency indicates that the model encountered challenges in effectively capturing disease location information and lacked sufficient robustness to image flipping and rotation.

To overcome these limitations, we propose Gated Asymmetrical GhostNet (GA-GhostNet), a lightweight CNN specifically designed for plant pest and disease identification. With just 3.73M parameters, our model achieves state-of-the-art performance on four public datasets. The main contributions of our work are as follows:

1) We propose a Gated Multi-scale Coordinate Attention (GM-CA) module, which incorporates gated mechanisms and coordinate attention [6]. GM-CA can effectively filter irrelevant information and noise while identifying pest and disease locations across multiple spatial scales. This allows it to surpass GhostNet's SE modules [7] in locating disease and pest regions for improved accuracy.

2) We propose an enhanced feature extraction module called Asymmetrical Ghost (AG) based on Asymmetric Convolution Blocks [8]. AG enhances robustness against image flipping while simultaneously improving feature extraction. During inference, AG utilizes convolution kernel fusion to avoid additional computation and parameter overhead.

3) Transfer learning is leveraged to transfer pest dataset training parameters to multiple disease datasets, thereby boosting disease identification accuracy.

4) Online data augmentation methods are utilized to enhance the model's generalization ability.

II. Related Work

In recent years, deep learning techniques have demonstrated remarkable performance in pest and disease identification tasks. Earlier works predominantly relied on conventional CNN architectures. Mohanty et al. [9] conducted a comparison between [3] and GoogleNet [10], revealing that GoogleNet, when applied with transfer learning, attained an accuracy of 99.35% on the PlantVillage dataset. This underscores the efficacy of neural networks in addressing

Manuscript received November 26, 2023; revised May 7, 2024. This work is supported by Foundation of Guangdong Educational committee under Grant the No.2022ZDZX4052, 2021ZDJS082, No.2019KQNCX148.

Yu Xiao is a postgraduate student of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China (email: xy1198522811@163.com).

Jie Wu is an associate professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China (*corresponding author to provide email: wujieaa@163.com).

Chi Ma is an associate professor of School of Computer Science and Engineering, Huizhou University, Huizhou 516007, China (email: machi@hzu.edu.cn).

this particular task. Picon et al. [11] proposed three CNN models utilizing ResNet [2] as the backbone network, which achieved 98% accuracy on a dataset covering 17 diseases. Cheng et al. [12] developed a pest identification method using deep residual learning, achieving 98.67% accuracy on 10 pest classes for agricultural applications, and can be applied to practical agricultural pest control tasks. Thenmozhi et al. [13] applied a deep CNN model to three public pest datasets, surpassing the performance of VGG-16 [14] and ResNet. To mitigate overfitting, they employed data augmentation methods involving rotation and translation. Mique Jr et al. [15] proposed a CNN model to assist farmers in identifying pests and diseases. Their approach involved preprocessing the collected images and subsequently utilizing them for model training, resulting in an accuracy of 90.9% on the test dataset.

Recent studies have increasingly explored the incorporation of attention mechanisms to enhance image recognition tasks. Lin et al. [16] proposed a graph pyramid attention CNN (GPA-Net) which obtained 99% on cassava leaves, 97% on the AI Challenger dataset, and 56.9% on IP102 pests. Zhao et al. [17] combined ResNet-50 and squeeze-and-excitation blocks [7], resulting in improved accuracy for tomato disease identification, reaching 96.81%, which is 4.25% higher than ResNet-50 alone. This demonstrates the benefits of attention for extracting complex disease features.

Despite their strong performance on specific datasets, these approaches are often hindered by their large model sizes, rendering them unsuitable for deployment on mobile devices. Lightweight CNNs address this limitation while maintaining strong feature extraction capabilities. Bao et al. [18] constructed SimpleNet using convolutions and inverted residuals with CBAM blocks [19], achieving 94.1% on wheat ear diseases. Adedoja et al. [20] proposed NASNet-Mobile CNN model plant disease diagnostic system achieved an accuracy rate of 99.31%. Chen et al. [21] proposed DFCANET for corn diseases, incorporating Coordinate Attention (CA) to outperform SE by 1.52%. By capturing spatial and cross-channel information, CA accurately localizes disease regions. DFCANET achieved 98.47% accuracy, surpassing MobileNetV2 [22], MobileNetV3 [23], and ShuffleNetV2 [24]. Guan et al. [25] designed an EfficientNetV2-based [26] model called Dise-Efficient, attaining 99.8% on PlantVillage and 64.4% on IP102. Thakur et al. [5] proposed VGG-ICNN integrating VGG16 and Inception-v7 to handle multi-scale objects. They evaluated VGG-ICNN on five publicly available datasets, where it achieved an accuracy of 99.16% on PlantVillage. Their experiments demonstrated that lightweight networks such as ShuffleNetV2 [24] exhibit inconsistent performance across datasets. For instance, ShuffleNetV2 obtained 97.96% on PlantVillage but only 77.78% on the Maize dataset [27]. In contrast, VGG-ICNN showed greater robustness across datasets, achieving an accuracy of 91.36% on the Maize dataset. This indicates that VGG-ICNN offers improved generalization capabilities compared to existing lightweight networks.

III. Materials and Methods

A. Dataset

Four datasets were utilized in this work:

The IP102 Pest Dataset [28]: A large publicly available dataset containing 75,222 images across 102 pest categories. The images follow a natural long-tailed distribution.

The Jute Pest Dataset [29]: Contains 17 pest categories and 6,209 images. This dataset first collects the image information for 13 categories using a Python open-source library and then converts the grayscale images to RGB images. Finally, it combines them with another publicly available dataset that contains four types of pest classes.

The Embrapa Disease Dataset [30]: It includes image information on 93 plant disease categories spanning 18 crop types. Although the original dataset showed an imbalance in the number of images per category, data augmentation was strategically utilized to generate a total of 46,376 images, ensuring a more balanced representation.

The Apple Disease Dataset [31]: Released on Kaggle and contains images across 4 apple disease categories - healthy, rusty, scab, and multiple diseases. The dataset has 3,642 total images, of which 1,822 have labels and the rest are unlabeled. Only the labeled images were used in this work.

In the IP102 dataset, the training, validation, and test sets comprise 45,095, 7,508, and 22,619 images, respectively. We adopted the same split for our experiments. For the Jute, Embrapa, and Apple datasets, the ratios of training, validation, and test sets are 70:15:15, 64:16:20, and 64:16:20, respectively.

B. GA-GhostNet Architecture

The proposed GA-GhostNet is a lightweight CNN model designed for pest and disease recognition. As depicted in Figure 1, GA-GhostNet comprises multiple core modules stacked sequentially to construct a deep neural network architecture.

The fundamental building blocks of GA-GhostNet are the Gated Asymmetrical Ghost bottleneck (GAG-bneck) modules, which will be elaborated in III-D. Each GAG-bneck module contains two key components: (1) The AG module for feature extraction (III-E); (2) The GM-CA module for attention (III-F). By stacking these GAG-bneck modules in increasing depth, GA-GhostNet is able to learn hierarchical feature representations from images.

The main module parameters of GA-GhostNet are provided in Table I, where #exp denotes the expansion size, controlling how much the input channels are expanded in the GAG-bneck. #out denotes the output feature map size of each module. GM-CA refers to the proposed Gated Multi-scale Coordinate Attention mechanism. Stride denotes the step size of the convolutional kernels used in each module.

In addition, we emply the CutMix data augmentation method (III-C) during training to enhance the model's generalization capability.

Lastly, we adopt transfer learning (III-G) to initialize the model parameters by transferring knowledge from pre-trained pest recognition models. Subsequently, we fine-tune the model on plant disease datasets, resulting in improved performance compared to training the model from scratch.

C. Data Augmentation Module

CutMix [32] is an online data augmentation method that effectively addresses the issue of erasing crucial feature

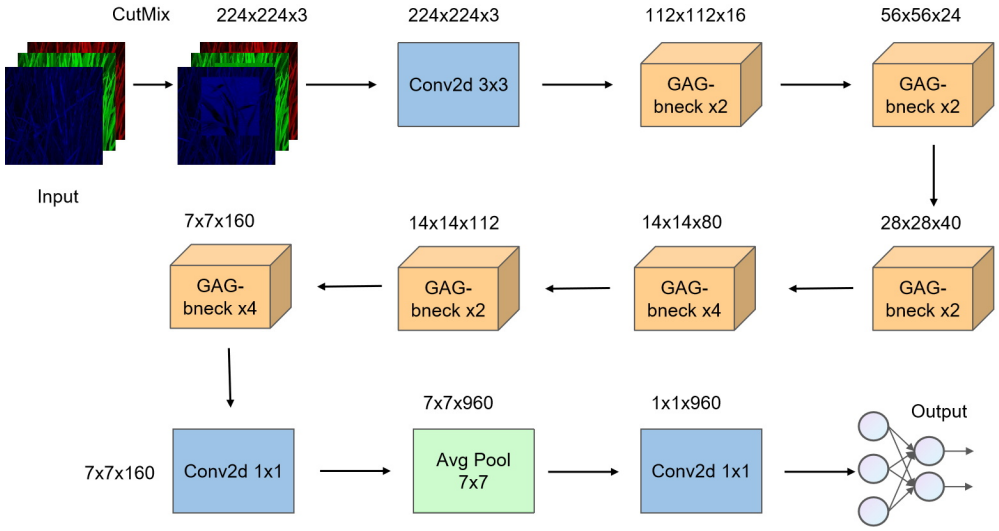


Fig. 1: GA-GhostNet structure

TABLE I: GA-GhostNet Network Architecture

Input	Operator	#exp	#out	GM-CA	Stride
224x224x3	Conv2d 3x3	-	16	-	2
112x112x16	GAG-bneck	16	16	-	1
112x112x16	GAG-bneck	48	24	-	2
56x56x24	GAG-bneck	72	24	-	1
56x56x24	GAG-bneck	72	40	1	2
28x28x40	GAG-bneck	120	40	1	1
28x28x40	GAG-bneck	240	80	-	2
14x14x80	GAG-bneck	200	80	-	1
14x14x80	GAG-bneck	184	80	-	1
14x14x80	GAG-bneck	184	80	-	1
14x14x80	GAG-bneck	480	112	1	1
14x14x112	GAG-bneck	672	112	1	1
14x14x112	GAG-bneck	672	160	1	2
7x7x160	GAG-bneck	960	160	-	1
7x7x160	GAG-bneck	960	160	1	1
7x7x160	GAG-bneck	960	160	-	1
7x7x160	GAG-bneck	960	160	1	1
7x7x160	Conv2d 1x1	-	960	-	1
7x7x960	AvgPool 7x7	-	-	-	-
1x1x960	Conv2d 1x1	-	1280	-	1
1x1x1280	FC	-	Num class	-	-

regions, which can occur with methods like Cutout [33] and RandomErasing [34]. CutMix combines two training data$(x_A, y_A)$and (x_B, y_B) into new training data$(\tilde{x}, \tilde{y})$, and the combination method is as follows

$$\begin{aligned} \tilde{x} &= \mathbf{M} \odot x_A + (\mathbf{1} - \mathbf{M}) \odot x_B \\ \tilde{y} &= \lambda y_A + (1 - \lambda) y_B \end{aligned} \tag{1}$$

where $\mathbf{M} \in \{0,1\}^{W \times H}$ is a binary mask that indicates the positions to remove and fill from two images, 1 represents a binary mask with all elements being 1, $\odot$ is element-wise multiplication. follows a Beta distribution: $\lambda \sim \text{Beta}(\alpha, \alpha)$, and α is set to 1 in the experiment, that is λ is sampled from a uniform distribution of $(0, 1)$. Before combining the training data, the CutMix algorithm retrieves a portion of the image by cropping a bounding box B = (r_x, r_y, r_w, r_h), indicating the cropping regions on x_A and x_B. The region B in x_A is removed and filled with the patch cropped from B in x_B. The bounding box coordinates are uniformly sampled according to the following

$$\begin{aligned} r_x &\sim \text{Unif}(0, W), \quad r_w = W\sqrt{1-\lambda} \\ r_y &\sim \text{Unif}(0, H), \quad r_h = H\sqrt{1-\lambda} \end{aligned} \tag{2}$$

The cropping region ratio $\frac{r_w r_h}{WH} = 1 - \lambda$ is satisfied. For the cropping region, the binary mask $\mathbf{M} \in \{0,1\}^{W \times H}$ is determined by filling 0 inside the bounding box B, otherwise 1.

D. Gated Asymmetrical Ghost Bottleneck

The proposed GAG-bneck module draws inspiration from the Ghost bottleneck module utilized in the GhostNet architecture. As depicted in Figure 2(a), the original Ghost bottleneck comprises two paired Ghost modules as well as a SE module, which are responsible for expanding and compressing the feature channels. Our GAG-bneck replaces the Ghost modules and SE module with the proposed the AG module and GM-CA module respectively, to enhance feature extraction and attention modeling.

As depicted in Figure 2(b), the GAG-bneck consists of two AG modules with the GM-CA module integrated in between them. The AG modules effectively replace the Ghost modules, while the GM-CA module takes the place of the SE module. Specifically, when stride=1, the first AG module expands the input channels, while the second AG module reduces the channels to match the shortcut path dimensionality, followed by element-wise addition with the shortcut connection. For stride=2, a downsampling layer is inserted between the two AG modules to halve the spatial dimensions. The shortcut connection also uses a downsampling layer to match dimensions.

By replacing the Ghost and SE modules with AG and GM-CA modules, GAG-bneck enhances the model's feature extraction and attention capabilities. Stacking GAG-bneck modules empowers GA-GhostNet to acquire robust multi-scale representations, facilitating precise localization for pest and disease recognition tasks.

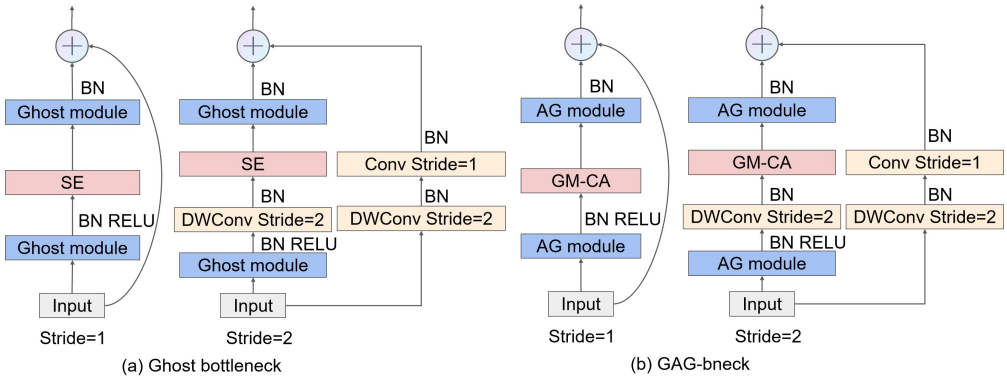


Fig. 2: Ghost bottleneck and GAG-bneck network structure

E. AG Module

Inspired by ACB [8] and Ghost module, the AG module enhances the model's robustness against image rotation and flipping transformations. The network structure of AG module is depicted in Figure 3. The AG module begins by employing a 1x1 standard convolution to reduce the channel count of the input image. Subsequently, it utilizes the Asymmetric Group Convolution Block (AGCB) to expand the feature maps, and finally concatenates different feature maps to form a new output. Specifically, the AGCB comprises three branches, each consisting of depthwise convolutions with kernel sizes of 3x3, 1x3, and 3x1, respectively. Although of different sizes, these kernels can be fused into one during inference using the additivity of convolution. The convolution kernel fusion is as follows

$$I * K^{(1)} + I * K^{(2)} = I * \left(K^{(1)} \oplus K^{(2)}\right) \tag{3}$$

where I represents the input feature map, and $K^{(1)}$ and $K^{(2)}$ be two convolution kernels of compatible sizes, $\oplus$ is the element-wise addition of the kernel parameters at the corresponding positions. This means convolving the input I with $K^{(1)}$ and $K^{(2)}$ separately then adding the results is equivalent to convolving I directly with the fused kernel $K^{(1)} \oplus K^{(2)}$. The input I may be cropped or padded accordingly. Consequently, during inference, the AG module utilizes the fused convolution kernel. This enhances feature extraction abilities relative to the original asymmetric branches, but without increasing computational costs [8].

F. GM-CA

GM-CA is an attention module that integrates a gated mechanism and the CA module to effectively capture spatial information in images. As shown in Figure 4, GM-CA consists of GM-W and GM-H submodules, detailed in Figure 5(a) and 5(b). Specifically, given the input x_c of the GM-CA module, two spatial pooling kernels of size $(H, 1)$ and $(1, W)$ are applied along the horizontal and vertical axes respectively to encode each channel's information. After decomposing the input along the horizontal dimension, the resulting feature map with height h can be expressed as

$$z_c^h(h) = \frac{1}{W} \sum_{0 \le i < W} x_c(h, i) \tag{4}$$

After decomposed in the vertical direction, the channel output with a width of w can be expressed as

$$z_c^w(w) = \frac{1}{H} \sum_{0 \le j < H} x_c(j, w) \tag{5}$$

After extracted the features in both spatial directions, the features are aggregated and then fed into the GM-W module. The output can be expressed as

$$\begin{aligned} f_{11} &= \mathrm{F}_{11}\left(\left[z^h, z^w\right]\right)\left(\delta\left(\mathrm{F}_{11}\left(\left[z^h, z^w\right]\right)\right)\right) \\ f_{13} &= \mathrm{F}_{13}\left(\left[z^h, z^w\right]\right)\left(\delta\left(\mathrm{F}_{13}\left(\left[z^h, z^w\right]\right)\right)\right) \\ f_{15} &= \mathrm{F}_{15}\left(\left[z^h, z^w\right]\right)\left(\delta\left(\mathrm{F}_{15}\left(\left[z^h, z^w\right]\right)\right)\right) \\ f_1 &= f_{11} + f_{13} + f_{15} \end{aligned} \tag{6}$$

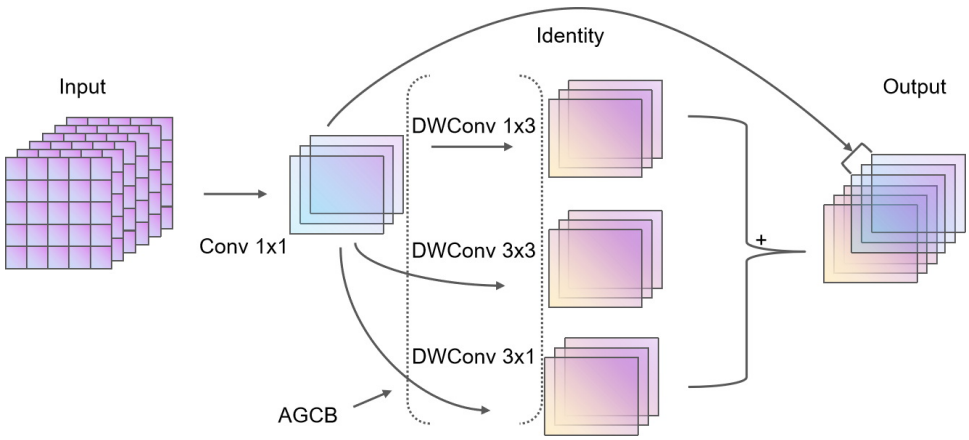


Fig. 3: AG Module

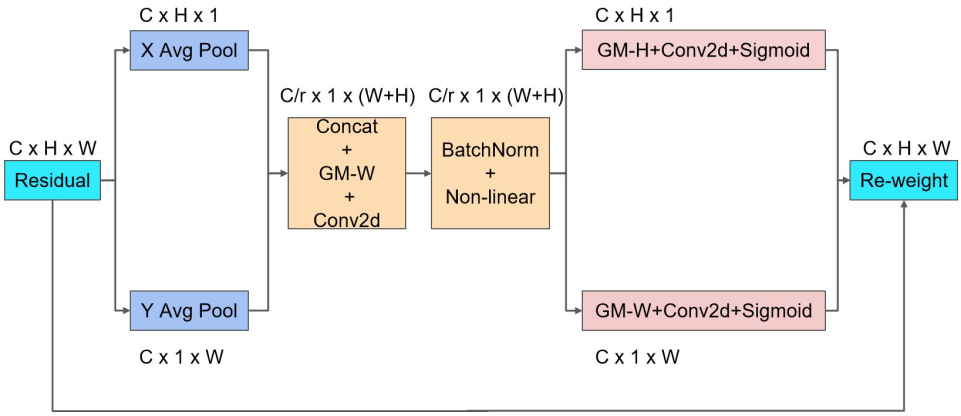


Fig. 4: GM-CA Module

where $[\cdot,\cdot]$ represents the concatenation operation along the spatial dimension, δ is the sigmoid function, F_{11}, F_{13}, and F_{15} are the depthwise convolutional transformation functions of 1x1, 1x3 and 1x5 respectively.

Then the result f_1 is sent to the 1x1 convolution transformation function F_1, and the output obtained is expressed as

$$f = \delta\left(\mathrm{F}_1\left(f_1\right)\right) \tag{7}$$

where $f \in \mathbb{R}^{C/r\times(H+W)}$ is the intermediate feature map that encodes the spatial information of the horizontal and vertical directions, δ is the nonlinear activation function. Here r is the compression ratio that controls the block size, which is the same as the SE block.

Then f is split along the spatial dimension into two independent tensors $f^h \in \mathbb{R}^{C/r\times H}$ and $f^w \in \mathbb{R}^{C/r\times W}$. f^h and f^w are sent to GM-H and GM-W respectively, and the output obtained is expressed as

$$\begin{aligned}
f^{h11} &= \mathrm{F}_{h1}\left(f^h\right)\left(\delta\left(\mathrm{F}_{h1}\left(f^h\right)\right)\right)\\
f^{h31} &= \mathrm{F}_{h3}\left(f^h\right)\left(\delta\left(\mathrm{F}_{h3}\left(f^h\right)\right)\right)\\
f^{h51} &= \mathrm{F}_{h5}\left(f^h\right)\left(\delta\left(\mathrm{F}_{h5}\left(f^h\right)\right)\right)\\
f^{w11} &= \mathrm{F}_{w1}\left(f^{\mathrm{w}}\right)\left(\delta\left(\mathrm{F}_{w1}\left(f^{\mathrm{w}}\right)\right)\right)\\
f^{w13} &= \mathrm{F}_{w3}\left(f^{\mathrm{w}}\right)\left(\delta\left(\mathrm{F}_{\mathrm{w3}}\left(f^{\mathrm{w}}\right)\right)\right)\\
f^{w15} &= \mathrm{F}_{w5}\left(f^{\mathrm{w}}\right)\left(\delta\left(\mathrm{F}_{\mathrm{w5}}\left(f^{\mathrm{w}}\right)\right)\right)\\
f^{h1} &= f^{h11} + f^{h31} + f^{h51}\\
f^{w1} &= f^{w11} + f^{w13} + f^{w15}
\end{aligned} \tag{8}$$

where δ denotes the sigmoid function, and F_{h1}, F_{h3}, and F_{h5} represent 1x1, 3x1 and 5x1 depthwise convolutional transformation functions, respectively. Similarly, F_{w1}, F_{w3}, and F_{w5} denote 1x1, 1x3 and 1x5 depthwise convolutional transformation functions, respectively.

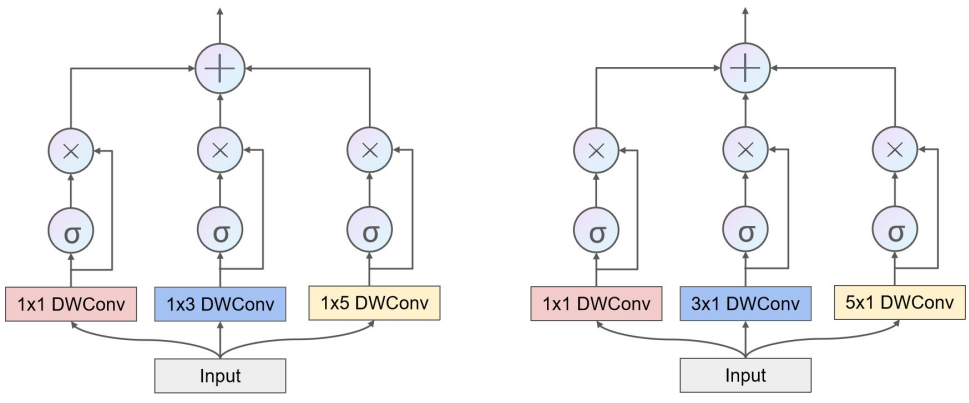
Then use two 1x1 convolutions, F_h and F_w, are applied to transform f^{h1} and f^{w1} into tensors with the same number of channels as the input x_c, and the two outputs are expressed as

$$\begin{aligned}
g^h &= \sigma\left(\mathrm{F}_h\left(f^{h1}\right)\right)\\
g^w &= \sigma\left(\mathrm{F}_w\left(f^{w1}\right)\right)
\end{aligned} \tag{9}$$

where σ is the sigmoid function. Finally, the output is expressed as

$$y_c(i,j) = x_c(i,j) \times g_c^h(i) \times g_c^w(j) \tag{10}$$

where x_c multiplies g^h and g^w as the attention weights in two spatial directions. In contrast to the CA module, the proposed GM-CA module not only emphasizes spatial information in specific rows and columns but also captures multi-scale spatial information. As described previously, horizontal and vertical attention are simultaneously applied to the input tensor. Each element in the two attention maps indicates whether the object of interest exists in the corresponding rows and columns, or spans multiple rows and columns.



(b) GM-W (b) GM-H

Fig. 5: GM-W and GM-H Modules

(a) IP102 (b) Apple

Fig. 6: Images from the IP102 dataset and the Apple dataset

G. Transfer Learning

The technique of transfer learning leverages knowledge gained from solved previous tasks to facilitate learning on new, related tasks [35]. As depicted in Figure 6(a) and (b), there are visual similarities between pest images in the IP102 dataset and disease images in the Apple dataset. Leveraging these resemblances, model parameters pre-trained on the IP102 dataset can serve as valuable initialization for the Jute, Embrapa, and Apple disease datasets.

IV. Results

A. Experimental Environment and Settings

Experiments were conducted using a Tesla P100 GPU with Python 3.10 and PyTorch 1.10. The loss function was cross-entropy loss optimized via AdamW. The initial learning rate was 0.0004 with a batch size of 32. For the IP102 and Embrapa datasets, models were trained for 50 epochs. For the Apple and Jute datasets, 30 epochs were used.

B. Data Preprocessing and Evaluation Indicators

During data preprocessing, input images were initially subjected to random cropping, resulting in a uniform size of 224x224 pixels. Subsequently, images were then randomly flipped and normalized by dividing each pixel by 255. Finally, the images were standardized using the mean and standard deviation values calculated for each color channel based on the IP102 dataset. This normalization and standardization accelerated model convergence.

To evaluate model performance, we utilized macro-average precision (MPre), macro-average recall (MRec), macro-average F1 score (MF1), and accuracy (Acc). Macro-averaging computes the metric independently for each class and takes the average, giving equal weight to all classes.

C. Comparative Experiments with Different Models

To substantiate the superiority of the proposed model, GA-GhostNet was comparatively evaluated against established lightweight CNN models and CNN+Transformer models on four publicly available datasets. Table II shows the experimental results on the IP102 dataset, where GA-GhostNet achieved the highest performance of 63.73%, 67.37%, 65.12%, and 71.90% for MRec, MPre, MF1, and Acc, respectively. GA-GhostNet exhibits lower FLOPs compared to other lightweight CNN models, such as MobileNetV2 [22],

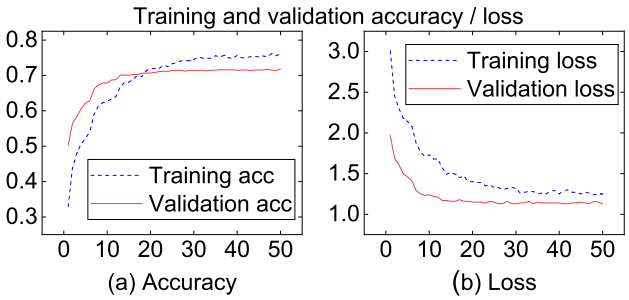


Fig. 7: Performance of GA-GhostNet on the IP102 dataset

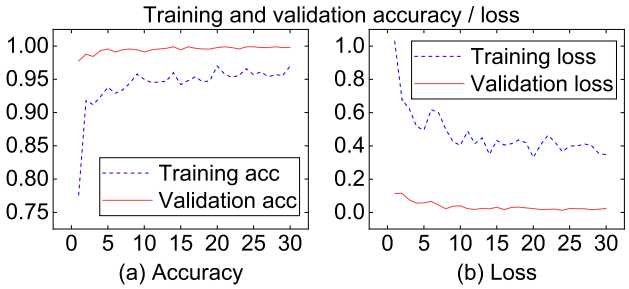


Fig. 8: Performance of GA-GhostNet on the Jute dataset

MobileNetV3 [23], MixNet_l [36], and EfficientNet_b1 [37]. While GA-GhostNet possesses a marginally higher number of parameters than MobileNetV2, its accuracy substantially surpasses that of MobileNetV2. In contrast, EffcientNet_b1 and MixNet_l had larger parameter sizes but were 2.86% and 2.27%, 1.06% and 0.83% lower than GA-GhostNet on MF1 and Acc, respectively. This performance gap can be attributed to the use of SE modules in EfficientNet_b1 and MixNet_l, which cannot capture image position information. Furthermore, when compared to MobileVit_s [38] and EdgeNext_small [39], both of which are CNN+Transformer models, GA-GhostNet achieved a remarkable 1.88% and 0.58% higher Acc while maintaining significantly reduced FLOPs.

Table III shows the results on the Jute dataset, where GA-GhostNet achieved 99.89% on all metrics, demonstrating its strong performance on pest datasets. Table IV shows the results on the Embrapa disease dataset. GA-GhostNet achieved the highest scores of 94.18% MRec, 96.36% MPre, 94.98% MF1, and 96.97% Acc. In contrast, MobileVit_s, EdgeNext_small, and EfficientNet_b1 exhibit significantly lower performance. Specifically, they fall behind by 5.91%,

TABLE II: IP102 Dataset Comparison Experiment

Method	Params	FLOPs	Mrec	Mpre	MF1	Acc
MobileNetV2	2.35M	299.69M	59.65%	64.01%	60.77%	69.56%
MobileNetV3	4.18M	215.36M	63.69%	65.77%	64.42%	71.33%
MixNet_l	5.89M	553.95M	62.35%	64.42%	62.85%	71.07%
MobileVit_s	4.99M	1420.27M	62.19%	64.51%	62.69%	70.12%
EdgeNext_small	5.31M	959.63M	63.01%	66.20%	64.05%	71.33%
EfficientNet_b1	6.40M	622.71M	61.68%	64.86%	62.26%	70.84%
GA-GhostNet	3.73M	168.40M	63.73%	67.37%	65.12%	71.90%

TABLE III: Jute Dataset Comparison Experiment

Method	Params	FLOPs	Mrec	Mpre	MF1	Acc
MobileNetV2	2.21M	299.58M	98.00%	98.00%	98.00%	98.00%
MobileNetV3	4.14M	215.24M	98.70%	98.58%	98.63%	98.78%
MixNet_l	5.76M	553.82M	99.21%	99.07%	99.13%	99.11%
MobileVit_s	4.94M	1420.22M	99.67%	99.65%	99.66%	99.66%
EdgeNext_small	5.28M	959.61M	99.60%	99.58%	99.59%	99.56%
EfficientNet_b1	6.29M	622.60M	99.30%	99.24%	99.25%	99.21%
GA-GhostNet	3.62M	168.29M	99.89%	99.89%	99.89%	99.89%

TABLE IV: Embrapa Dataset Comparison Experiment

Method	Params	FLOPs	Mrec	Mpre	MF1	Acc
MobileNetV2	2.34M	299.68M	91.12%	94.97%	92.14%	95.54%
MobileNetV3	4.17M	215.35M	91.51%	94.75%	92.61%	95.96%
MixNet_l	5.88M	553.94M	91.55%	95.54%	92.79%	96.23%
MobileVit_s	4.98M	1420.26M	88.27%	93.23%	89.59%	95.22%
EdgeNext_small	5.30M	959.62M	88.44%	92.75%	89.86%	95.27%
EfficientNet_b1	6.38M	622.70M	88.11%	93.84%	89.76%	95.28%
GA-GhostNet	3.72M	168.39M	94.18%	96.36%	94.98%	96.97%

TABLE V: Apple Dataset Comparison Experiment

Method	Params	FLOPs	Mrec	Mpre	MF1	Acc
MobileNetV2	2.19M	299.56M	72.63%	92.79%	72.10%	90.62%
MobileNetV3	4.13M	215.22M	84.95%	89.96%	86.81%	94.32%
MixNet_l	5.74M	553.80M	84.32%	90.71%	86.52%	93.46%
MobileVit_s	4.93M	1420.21M	83.84%	92.25%	86.52%	94.31%
EdgeNext_small	5.27M	959.60M	87.13%	88.55%	87.13%	94.03%
EfficientNet_b1	6.27M	622.58M	80.94%	88.53%	83.40%	90.90%
GA-GhostNet	3.60M	168.27M	88.01%	90.40%	89.06%	95.17%

5.74%, and 6.07% on MRec, and 5.39%, 5.12%, and 5.22% on MF1, respectively. This indicates these three models performed poorly on the disease dataset. Table V shows the Apple disease dataset results. GA-GhostNet achieved the best results, significantly outperforming EfficientNet_b1 and MobileNetv2 across all metrics. Specifically, EfficientNet_b1 recorded only a 0.28% higher accuracy than MobileNetv2, but underperformed GA-GhostNet by 5.66% and 4.27% on MF1 and accuracy. These findings indicate that larger CNN models struggled more with the relatively small dataset.

Figure 7-10 illustrate the training and validation performance over epochs. As shown in Figure 8(a) and 9(a), the training Acc on the Embrapa and Jute datasets was lower than the validation Acc, exhibiting an underfitting phenomenon. This can be attributed to the complex augmented data generated by CutMix, which has the potential to confuse the model during training. Despite the underfitting issue, GA-GhostNet still demonstrates effective disease and

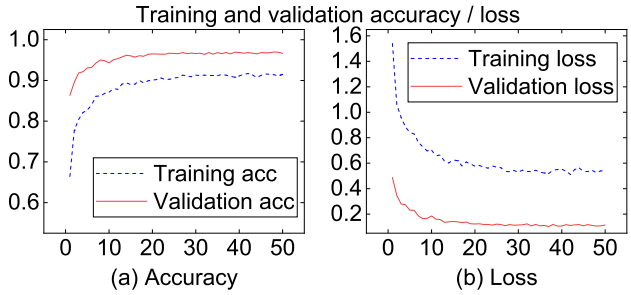


Fig. 9: Performance of GA-GhostNet on the Embrapa dataset

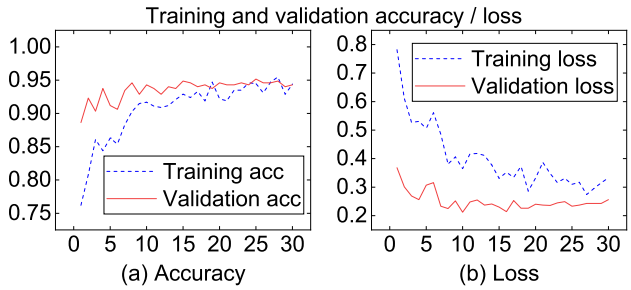


Fig. 10: Performance of GA-GhostNet on the Apple dataset

pest identification capabilities across all four datasets. The comparative experiments demonstrate the superiority of GA-GhostNet over lightweight CNNs and CNN+Transformers for pest and disease recognition across diverse datasets.

D. The Results of the Experiment were Compared with Previous Studies

GA-GhostNet was compared to other models on the IP102, Jute, Embrapa, and Apple datasets. The results are shown in Table VI.

TABLE VI: Accuracy Results Comparison with Previous Research

Dataset	Model	Params	Acc
IP102	Ayan et al. [40]	>23M	67.13%
	Z et al. [41]	18.9M	71.60%
	Setiawan et al. [42]	4.2M	71.32%
	Albattah et al. [43]	7.08M	68.74%
	GA-GhostNet	3.73M	71.90%
Jute	Thakur et al. [29]	18.35M	99.00%
	GA-GhostNet	3.62M	99.89%
Embrapa	Zhao et al. [44]	6.71M	88.48%
	Thakur et al. [5]	6M	93.66%
	GA-GhostNet	3.72M	96.97%
Apple	Zhao et al. [44]	6.71M	88.71%
	Thakur et al. [5]	6M	94.24%
	GA-GhostNet	3.60M	95.17%

On the IP102 dataset, GA-GhostNet achieves an Acc of 71.90%, surpassing the lightweight model proposed by Albattah et al. by a margin of 0.58%. This demonstrates the superior performance of the proposed model compared to previous lightweight architectures in pest recognition. On the Jute dataset, GA-GhostNet attained near-perfect Acc of 99.89%, further evidencing its capabilities on pest classification tasks. On the Embrapa and Apple plant disease datasets, GA-GhostNet significantly outperforms the VGG-ICNN model proposed by Thakur et al., achieving Acc improvements of 3.31% and 0.93%, respectively.

These comprehensive benchmark comparisons demonstrate the superior performance of GA-GhostNet in both pest and disease recognition tasks across a diverse range of datasets. The consistently high accuracy shows the benefits of its lightweight design, multi-scale spatial attention mechanism, robust feature extraction, and transfer learning approach.

E. The Impact of Different Data Augmentations on the Results

To determine an appropriate data augmentation method for pest identification, various data augmentation methods were assessed on the IP102 dataset using GhostNet. The results are shown in Table VII. CutMix combined with RandomHorizontalFlip (RHF) achieved the best performance of 62.72% MRec, 66.89% MPre, 64.02% MF1, and 71.43% Acc. In contrast, Cutout and RandomErasing exhibited poor performance, possibly due to their tendency to occlude crucial pest features. Unlike these techniques, CutMix operates by replacing image regions with patches extracted from other training examples. This approach preserves the validity of the image content and prevents the occlusion of important features. However, Mixup [45] interpolates the two graphs proportionally to mix the samples, resulting in an unnatural blending of image features. This limitation leads to a slightly lower accuracy compared to CutMix.

TABLE VII: Results of Different Data Augmentation Methods

Method	Mrec	Mpre	MF1	Acc
Without	54.52%	58.44%	55.49%	64.24%
RHF	61.87%	66.74%	63.47%	70.98%
RHF+Cutout	61.77%	65.69%	62.91%	70.85%
RHF+RandomErasing	62.12%	66.34%	63.42%	71.09%
RHF+Mixup	62.02%	66.83%	63.57%	71.25%
RHF+CutMix	62.72%	66.89%	64.02%	71.43%

F. Ablation Experiments

Table VIII compares the performance of different attention mechanisms. The CA module achieved 0.27% and 1.23% higher Acc than the SE and the CBAM modules. This is because CA can capture both spatial information and cross-channel information. However, CA only considers single-scale spatial relationships along each row and column. Especially in earlier layers, the convolution kernels have small receptive fields. This makes it difficult for CA to capture the overall positional information of larger objects. Single convolutions have limited receptive fields and may learn

TABLE VIII: Comparison of the Results of Different Attention Mechanisms

Method	Params	FLOPs	Mrec	Mpre	MF1	Acc
SE	4.03M	154.32M	61.87%	66.74%	63.47%	70.98%
CBAM	4.03M	156.03M	62.27%	66.37%	63.80%	70.12%
CA	3.66M	166.52M	63.09%	66.33%	64.21%	71.25%
GM-CA	3.73M	168.40M	63.49%	67.07%	64.82%	71.51%

features that lack rich contextual information, which hinders the detection of multi-scale targets [46].

In contrast, the proposed GM-CA module possesses the ability to effectively filter out irrelevant information and noise through its gating units. This capability allows GM-CA to capture multi-scale spatial relationships, thereby enabling precise localization of both large and small objects at each stage of the network. Consequently, GM-CA achieved notable accuracy improvements of 0.53%, 1.39%, and 0.26% compared to the SE, CBAM, and CA modules, respectively.

Table IX shows the ablation study results for different modules. GhostNet served as the baseline model, and its performance was significantly enhanced by employing the CutMix data augmentation method. Substituting the default SE module in GhostNet with the GM-CA module increased Acc by 0.35%. Substituting the AG module for the Ghost module led to an Acc gain of 0.12%. This is because the 1x3 convolution kernel in AGCB can extract the same features at identical spatial locations even after flipping the input image. Consequently, horizontal kernels such as 1x3 enhance the model's robustness against image flipping, while the vertical 3x1 kernel offers similar advantages against image rotation. The Ghost module of GhostNet generates numerous feature maps through simple linear operations (DWConv3x3), which effectively reduces the number of parameters. However, these feature maps lack robustness to image flipping and rotation, as 3x3 kernels alter the extracted features when images are flipped [8]. In contrast, the AG module enhances feature extraction without introducing additional inference-time parameters by merging the asymmetric convolution branches.

TABLE IX: Ablation Experiments on IP102 Dataset

Method	Mrec	Mpre	MF1	Acc
Baseline	61.87%	66.74%	63.47%	70.98%
CutMix	62.72%	66.89%	64.02%	71.43%
CutMix+GM-CA	63.55%	67.36%	64.91%	71.78%
CutMix++GM-CA+AG module	63.73%	67.37%	65.12%	71.90%

V. Conclusion

A novel lightweight CNN model, GA-GhostNet, is proposed for effective disease and pest identification, with a parameter size of only 3.73M. Additionally, experiments are conducted to evaluate the impact of various data augmentation techniques on identification accuracy. GA-GhostNet has the following characteristics: GM-CA can locate the feature regions of diseases and pests, the AG module can enhance the feature extraction ability, and it does not increase the extra computation in inference. On the IP102 dataset, GA-GhostNet achieves the highest Acc of 71.90%, and also performs well on the other three metrics of Mrec, Mpre, and MF1, with 63.73%, 67.37%, and 65.12%, respectively. By leveraging transfer learning on the Jute, Embrapa, and Apple datasets, GA-GhostNet achieves near-perfect Acc levels of 99.89%, 96.97%, and 95.17%, respectively. These results demonstrate the superiority of GA-GhostNet compared to existing lightweight networks in pest and disease recognition tasks.

References

[1] P. S. Thakur, P. Khanna, T. Sheorey, and A. Ojha, "Explainable vision transformer enabled convolutional neural network for plant disease identification: Plantxvit," *arXiv preprint arXiv:2207.07919*, 2022.

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

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[4] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "Ghostnet: More features from cheap operations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1580–1589.

[5] P. S. Thakur, T. Sheorey, and A. Ojha, "Vgg-icnn: A lightweight cnn model for crop disease identification," *Multimedia Tools and Applications*, vol. 82, no. 1, pp. 497–520, 2023.

[6] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 713–13 722.

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

[8] X. Ding, Y. Guo, G. Ding, and J. Han, "Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1911–1920.

[9] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, p. 1419, 2016.

[10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[11] A. Picon, M. Seitz, A. Alvarez-Gila, P. Mohnke, A. Ortiz-Barredo, and J. Echazarra, "Crop conditional convolutional neural networks for massive multi-crop plant disease classification over cell phone acquired images taken on real field conditions," *Computers and Electronics in Agriculture*, vol. 167, p. 105093, 2019.

[12] X. Cheng, Y. Zhang, Y. Chen, Y. Wu, and Y. Yue, "Pest identification via deep residual learning in complex background," *Computers and Electronics in Agriculture*, vol. 141, pp. 351–356, 2017.

[13] K. Thenmozhi and U. S. Reddy, "Crop pest classification based on deep convolutional neural network and transfer learning," *Computers and Electronics in Agriculture*, vol. 164, p. 104906, 2019.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[15] E. L. Mique Jr and T. D. Palaoag, "Rice pest and disease detection using convolutional neural network," in *Proceedings of the 1st International Conference on Information Science and Systems*, 2018, pp. 147–151.

[16] S. Lin, Y. Xiu, J. Kong, C. Yang, and C. Zhao, "An effective pyramid neural network based on graph-related attentions structure for fine-grained disease and pest identification in intelligent agriculture," *Agriculture*, vol. 13, no. 3, p. 567, 2023.

[17] S. Zhao, Y. Peng, J. Liu, and S. Wu, "Tomato leaf disease diagnosis based on improved convolution neural network by attention module," *Agriculture*, vol. 11, no. 7, p. 651, 2021.

[18] W. Bao, X. Yang, D. Liang, G. Hu, and X. Yang, "Lightweight convolutional neural network model for field wheat ear disease identification," *Computers and Electronics in Agriculture*, vol. 189, p. 106367, 2021.

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

[20] A. O. Adedoja, P. A. Owolawi, T. Mapayi, and C. Tu, "Intelligent mobile plant disease diagnostic system using nasnet-mobile deep learning," *IAENG International Journal of Computer Science*, vol. 49, no. 1, pp. 216–231, 2022.

[21] Y. Chen, X. Chen, J. Lin, R. Pan, T. Cao, J. Cai, D. Yu, T. Cernava, and X. Zhang, "Dfcanet: A novel lightweight convolutional neural network model for corn disease identification," *Agriculture*, vol. 12, no. 12, p. 2047, 2022.

[22] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[23] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1314–1324.

[24] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 116–131.

[25] H. Guan, C. Fu, G. Zhang, K. Li, P. Wang, and Z. Zhu, "A lightweight model for efficient identification of plant diseases and pests based on deep learning," *Frontiers in Plant Science*, vol. 14, 2023.

[26] M. Tan and Q. Le, "Efficientnetv2: Smaller models and faster training," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 096–10 106.

[27] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, "Using deep transfer learning for image-based plant disease identification," *Computers and Electronics in Agriculture*, vol. 173, p. 105393, 2020.

[28] X. Wu, C. Zhan, Y.-K. Lai, M.-M. Cheng, and J. Yang, "Ip102: A large-scale benchmark dataset for insect pest recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8787–8796.

[29] M. S. H. Talukder, M. R. Chowdhury, M. S. U. Sourav, A. Al Rakin, S. A. Shuvo, R. B. Sulaiman, M. S. Nipun, M. Islam, M. R. Islam, M. A. Islam *et al.*, "Jutepestdetect: An intelligent approach for jute pest identification using fine-tuned transfer learning," *Smart Agricultural Technology*, vol. 5, p. 100279, 2023.

[30] J. G. A. Barbedo, L. V. Koenigkan, B. A. Halfeld-Vieira, R. V. Costa, K. L. Nechet, C. V. Godoy, M. L. Junior, F. R. A. Patricio, V. Talamini, L. G. Chitarra *et al.*, "Annotated plant pathology databases for image-based detection and recognition of diseases," *IEEE Latin America Transactions*, vol. 16, no. 6, pp. 1749–1757, 2018.

[31] R. Thapa, N. Snavely, S. Belongie, and A. Khan, "The plant pathology 2020 challenge dataset to classify foliar disease of apples," *arXiv preprint arXiv:2004.11958*, 2020.

[32] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6023–6032.

[33] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.

[34] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 13 001–13 008.

[35] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[36] M. Tan and Q. V. Le, "Mixconv: Mixed depthwise convolutional kernels," *arXiv preprint arXiv:1907.09595*, 2019.

[37] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114.

[38] S. Mehta and M. Rastegari, "Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer," *arXiv preprint arXiv:2110.02178*, 2021.

[39] M. Maaz, A. Shaker, H. Cholakkal, S. Khan, S. W. Zamir, R. M. Anwer, and F. Shahbaz Khan, "Edgenext: efficiently amalgamated cnn-transformer architecture for mobile vision applications," in *European Conference on Computer Vision*. Springer, 2022, pp. 3–20.

[40] E. Ayan, H. Erbay, and F. Varçın, "Crop pest classification with a genetic algorithm-based weighted ensemble of deep convolutional neural networks," *Computers and Electronics in Agriculture*, vol. 179, p. 105809, 2020.

[41] Z. Hechen, W. Huang, and Y. Zhao, "Vit-lsla: Vision transformer with light self-limited-attention," *arXiv preprint arXiv:2210.17115*, 2022.

[42] A. Setiawan, N. Yudistira, and R. C. Wihandika, "Large scale pest classification using efficient convolutional neural network with augmentation and regularizers," *Computers and Electronics in Agriculture*, vol. 200, p. 107204, 2022.

[43] W. Albattah, M. Masood, A. Javed, M. Nawaz, and S. Albahli, "Custom cornernet: a drone-based improved deep learning technique for large-scale multiclass pest localization and classification," *Complex & Intelligent Systems*, vol. 9, no. 2, pp. 1299–1316, 2023.

[44] Y. Zhao, C. Sun, X. Xu, and J. Chen, "Ric-net: A plant disease classification model based on the fusion of inception and residual structure and embedded attention mechanism," *Computers and Electronics in Agriculture*, vol. 193, p. 106644, 2022.

[45] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[46] L. Zhang, J. Dai, H. Lu, Y. He, and G. Wang, "A bi-directional message passing model for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1741–1750.