Article

The Lightweight Deep Learning Model in Sunflower Disease Identification: A Comparative Study

Liqian Zhang [1,2,*] and Xiao Wu [1]

[1] College of Computer and Information Engineering, Inner Mongolia Agriculture University, Hohhot 010018, China; 17686508532@163.com

[2] Inner Mongolia Autonomous Region Key Laboratory of Big Data Research and Application of Agriculture and Animal Husbandry, Hohhot 010018, China

* Correspondence: zhanglq@imau.edu.cn

Abstract: (1) With the development of artificial intelligence, people expect to use modern information technology to solve the critical problems encountered in agriculture. How to identify sunflower diseases as early and quickly as possible and take corresponding measures has become a key issue for increasing crop production and farmers' income. Sunflowers, as an important oil crop, are vulnerable to infections by various diseases, such as downy mildew, leaf scar, gray mold, etc. (2) In order to select a better lightweight model that can be embedded into mobile devices or embedded devices for sunflower disease detection, we compared five lightweight deep learning models in this study, including SqueezeNet, ShuffleNetV2, MnasNet-A1, MobileNetV3-Small, and EfficientNetV2-Small. The dataset used to train and test the models included 1892 images. These images were divided into four categories, namely, downy mildew, gray mold, leaf scar, and fresh leaves. (3) By evaluating the accuracy, precision, recall, and F1 score of each model, we found that EfficeintNetV2-Small exhibited the highest performance with an accuracy of 90.19%. Whereas the other models, SqueezeNet, ShuffleNetV2, MnasNet-A1, and MobileNetV3-Small, achieved accuracies of 84.08%, 79.31%, 88.59%, and 84.08%, respectively. To address the problem of poor generalization ability of models caused by small datasets, we adopted the transfer learning technique. After doing that, the recognition accuracies of the five models, SqueezeNet, ShuffleNetV2, MnasNet-A1, MobileNetV3-Small, and EfficeintNetV2-Small, reached 96.02%, 95.23%, 94.96%, 96.92%, and 99.20%, respectively. The accuracies of these five models were improved by 14.2%, 20%, 7.2%, 15.2%, and 10%. Based on the comparative results, we found EfficeintNetV2-Small was an optimal choice for sunflower disease identification due to its high detection accuracy.

Keywords: artificial intelligence; image classification; lightweight deep learning model; sunflower disease identification; transfer learning

check for updates

Academic Editor: Andrea Prati

Received: 17 January 2025
Revised: 14 February 2025
Accepted: 15 February 2025
Published: 17 February 2025

Citation: Zhang, L.; Wu, X. The Lightweight Deep Learning Model in Sunflower Disease Identification: A Comparative Study. *Appl. Sci.* **2025**, *15*, 2104. https://doi.org/10.3390/app15042104

1. Introduction

With the continuous advancement of artificial intelligence technology, deep learning algorithms, such as Convolutional Neural Networks (CNN), have made significant progress in the field of image recognition. Many CNN models have emerged, such as AlexNet [1], VGGNet [2], ResNet [3], SENet [4], DenseNet [5], and so on. Meanwhile, the continuous development of information technology has been propelling the progress of precision agriculture, furnishing robust technical backing for the identification of crop diseases. Accurately identifying diseases and taking corresponding measures can greatly reduce the yield reduction caused by crop diseases, thus increasing farmers' incomes. Furthermore, it can also reduce the use of pesticides and achieve green and sustainable development [6].

Sunflower, as a widely known oilseed crop plant, holds significant economic value and social significance globally [7]. Statistical data show that the global production of sunflower seeds in 2023/2024 was approximately 55 million metric tons [8]. In 2022/2023 the global production of sunflower seeds was approximately 53.4 million metric tons. In 2021/2022, it was approximately 46.3 million metric tons.

From the trend of changes in global sunflower production in the world, it can be seen that it is showing an upward trend. This puts forward higher requirements for the accurate identification of plant diseases [9]. The diseases that affect the yield and quality of sunflowers mainly include downy mildew [10], gray mold [11], sclerotinia disease [12], leaf scar disease [13], and sunflower rust disease [14,15]. Therefore, early and accurate identification of sunflower disease types followed by precise control measures can greatly reduce the impact of these diseases.

The traditional approach to identify crop diseases primarily depends on farmers' long-term and meticulous observations, combined with their accumulated experience. This traditional method relies excessively on expert experience, so the accuracy of the diagnosis cannot be guaranteed. In precision agriculture, deep learning and Internet of Things (IoT) technologies continue to deeply integrate with agriculture [16]. In the pursuit of agricultural efficiency and productivity, people anticipate harnessing state-of-the-art information technologies to preemptively and precisely identify crop diseases in a timely manner. Concurrently, farmers harbor the expectation of effortlessly and autonomously diagnosing crop ailments. They envision achieving this by simply capturing images of the affected areas of their crops using their mobile devices or other embedded systems, which would revolutionize the way they manage and protect their agricultural yields. In the field of crop identification, images of leaves, stems, and other parts of crops are used to identify the types of diseases and the severity. Some studies on crop disease research are shown in Table 1.

Table 1. Studies on crop disease identification using deep learning method.

Author	Target	Models	Work Details	Results
Hua Yang, et al., 2024 [17]	Rice leaf disease	DHLC-DETR	Proposed a dense higher-level composition feature pyramid network (DHLC-FPN) and integrated it into the detection transformer (DETR) algorithm.	The proposed model achieved an average accuracy of 97.44% on the IDADP rice disease identification.
Francis et al., 2019 [18]	Tomato	CNN model	The proposed model comprises four convolutional layers, followed by equivalent pooling layers. The model also uses a sigmoid activation function and two dense layers that are fully coupled.	The system's output demonstrates an impressively high accuracy rate of 87%.
Yi Zhong and MengJun Tong, 2023 [19]	Sunflower leaf disease	TeenyNet	Designed a module to extract multi-frequency multi-scale features and proposed lightweight dual-fusion attention and multi-branching structure to identify sunflower disease.	TeenyNet obtains the highest accuracy of 98.94%.

Table 1. *Cont.*

Author	Target	Models	Work Details	Results
Kanaparthi, et al., 2023 [20]	Chili	Squeeze-Net	The study explored the Squeeze-Net architecture and the impact of the different CNN optimizers and other training parameters on the identification accuracy.	The research results indicate that the Squeeze-Net CNN architecture can achieve a 100% accuracy rate in classifying chili peppers when using the ADAM and RMSPROP optimizers.
Islam, M. M. et al., 2023 [21]	Crop disease	ResNet-50	Developing a smart web application using ResNet50 to identify crop disease.	Research shows that the accuracy of the ResNet-50 model is 98.98%.
Li Ma, et al., 2023 [22]	Maize leaf disease	Yolov5n	Adding CA attention module and STR to YOLOv5n.	The average recognition accuracy of the algorithm model can reach 95.2%.
Sandeep Kumar, et al., 2022 [23]	Cotton disease	CNN	Developing iOS app to make the disease predication.	The model accuracy is around 90%.
Si Chen, et al., 2021 [24]	Sunflower leaf disease	YoloV4	Obtain three effective feature layers of MobileNet, and replace the feature layer of YoloV4 backbone.	The improved model has a certain improvement in evaluation of precision, recall, and F1.
A. Sirohi and A. Malik, 2021 [25]	Sunflower diseases	A hybrid model	Use the stacking ensemble learning technique and combine two models, which are VGG16 and MobileNet.	The proposed model gave 89.2% accuracy on their dataset.
Jaweria Kainat, et al., 2021 [26]	Cucumber leaf disease	Fine KNN	Using the feature extraction of HOG, LBP, and color features.	The best accuracy is 94.60%.
Y. Zhong and M. Zhao, 2020 [27]	Apple leaf disease	DenseNet-121	Three methods of regression, multi-label classification, and focus loss function were proposed.	The accuracy of the three methods is 93.51%, 93.31%, and 93.71%.
Prabira Kumar Sethy, et al., 2020 [28]	Rice leaf disease	11 CNN models	11 CNN models in transfer learning approach and in deep feature plus (SVM) were applied to rice disease classification.	The deep feature of ResNet50 plus SVM performs better with an F1 score of 98.38% in rice disease identification.

From Table 1, it can be seen that many researchers have applied deep learning methods to the disease identification of some crops or fruits, such as cotton, maize, rice, chili, tomato, cucumber, apple, sunflower, etc. In addition to the studies listed in Table 1, in terms of sunflower disease identification, Promila Ghosh et al. [29] proposed a hybrid model using transfer learning (TL) and CNN to detect sunflower diseases. The study utilized 467 images, including images of gray mold, downy mildew, leaf scar, and healthy leaves. The experiments demonstrated that the proposed hybrid model achieved the best results in terms of precision, recall, F1 score, and accuracy. T.A. Sathi et al. [30] developed a deep learning approach called SunNet for detecting sunflower diseases. The study investigated various disease classification techniques on 1428 sunflower leaf images and found that the best-performing classifier overall was a ResNet50 classifier, which achieved an average accuracy of 97.88%. Yunis Gulzar et al. [31] compared the performance of five models in the identification of sunflower diseases, such as AlexNet, VGG16, InceptionV3, MobileNetV3, and EfficientNetB3, and finally found that EfficientNetB3 had the highest accuracy of 97.9%.

Guowei Dai et al. [32] proposed a DFN-PSAN method based on YOLOv5, which achieved an accuracy of 94.47% on the sunflower dataset.

The image information of the dataset used in the previously mentioned studies is shown in Table 2.

Table 2. The details of images used in some references.

References	Image Bands	Image Size	Number of Images
[17]	3	3000 × 2000	2400
[18]	3	64 × 64	3663
[19]	3	224 × 224	1668
[20]	3	128 × 128	160
[21]	3	224 × 224	10,000
[22]	3	640 × 640	4353
[23]	3	256 × 256	825
[24]	3	416 × 416	2030
[25]	3	224 × 224	Not mentioned
[26]	3	640 × 480	1262
[27]	3	128 × 128	2462
[28]	3	300 × 300	5932
[29]	3	150 × 150	467
[30]	3	224 × 224	1428
[31]	3	512 × 512	1892
[32]	3	256 × 256	3784

From the analysis of the aforementioned literature, it can be observed that most implementations of deep learning for crop disease recognition are currently confined to the laboratory research phase. Some models always have a large number of parameters and require high hardware requirements, making them unsuitable for application on mobile devices to achieve real-time field image collection and recognition. In order to deploy deep learning models on mobile terminals, researchers have been making models more lightweight in aspects such as reducing model depth, decreasing model parameters, and lowering the hardware requirements of models. These kinds of model are called lightweight deep learning models. Many lightweight models are proposed, such as SqueezeNet [33], ShuffleNetV1 [34], ShuffleNetV2 [35], MnasNet-A1 [36], MobileNetV1 [37], MobileNetV2 [38], MobileNetV3 [39], EfficientNetV1 [40], EfficientNetV2 [41], etc. These lightweight models make it possible to run neural network models in mobile terminals or embedded devices.

Researchers have indeed conducted relatively few studies on the field of identification of sunflower diseases, and even fewer have been put into practical application. In order to select an efficient lightweight model that can be employed in a mobile system, we compared the performance of five lightweight models, which were SqueezeNet, ShuffleNetV2, MnasNet-A1, MobileNetV3-Small, and EfficientNetV2-Small, using the same image dataset. These five models were representative ones in the development process of lightweight neural networks. This study can provide an application basis for the identification of agricultural crop diseases.

Our main contributions in this study include the following:

- We pointed out the limitations of traditional crop disease identification methods and summarized the applications of deep learning models in various crop diseases identified in recent years.
- We conducted an in-depth discussion on the architectures and characteristics of five currently mainstream lightweight models, which were SqueezeNet, ShuffleNetV2, MnasNet-A1, MobileNetV3-Small, and EfficientNetV2-Small.

- We applied these five models to the sunflower disease identification and conducted a comparative analysis in terms of recognition accuracy, recall, precision, and F1 score. We found that the EfficientNetV2-Small model had excellent performance on recognizing sunflower disease.
- In order to overcome the shortage of small dataset, we improved the identification accuracy of sunflower disease by using pre-trained parameters on ImageNet, and the optimal model parameters were obtained, which could be applied in mobile devices or embedded equipment in future work.

2. Materials and Methods

In order to train deep learning models, we collected the sunflower disease images as a dataset and split the dataset into a training set and a testing set in a ratio of 8:2. Before training the models, we performed normalization on the images to enhance model performance. After training models and obtaining the optimal weight parameters of the models, we proceeded with model testing. The models were evaluated using precision, recall, accuracy, and F1 score. The definitions of these metrics will be presented in Section 2.3. The methodology used in this paper is illustrated in Figure 1.

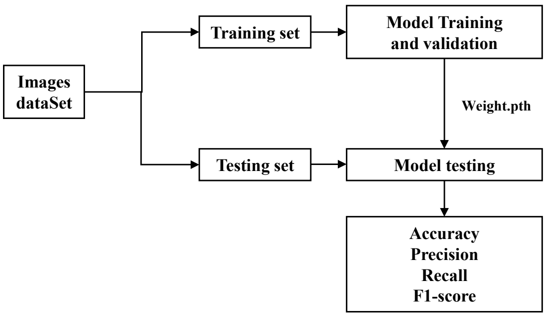


Figure 1. The process used in this study.

2.1. Image Dataset

In this study, we utilized a public sunflower image dataset [13], which contained 467 original images and 1868 augmented images. These images included four categories, such as gray mold, downy mildew, leaf scar, and fresh leaves. In this dataset, augmentation was achieved through rotation, scaling, and shearing. The augmented 1868 images were used for model training and testing. The distributions of all images are shown in Table 3.

Table 3. The distribution of dataset images.

Name of Classes	Number of Original Images	Number of Augmented Images
Downy mildew	72	398
Gray mold	120	470
Leaf scar	141	509
Fresh leaves	134	491
Total	467	1868

Downy mildew disease can cause chlorotic spots on the leaves of sunflower, with a white velvety mold layer on the underside of the leaves. Then the leaves turn brown and scorched. The infected plants become stunted, and the flower heads become deformed.

Gray mold disease mainly affects the flower heads of sunflowers. In the early stages of infection, the flower heads exhibit water-soaked decay. When the humidity is high, gray mold layers grow, leading to the rotting of the flower heads and failure to bear seed.

Leaf scar disease is characterized by the appearance of circular or nearly circular scar-like lesions on the leaves. The edges of these lesions are typically dark brown, while the centers are grayish-white, and there may also be the presence of a gray mold layer. Although this disease does not directly cause the death of sunflowers, it reduces the photosynthesis capability of the leaves, thereby affecting the yield. Four types of image samples are displayed in Figure 2.

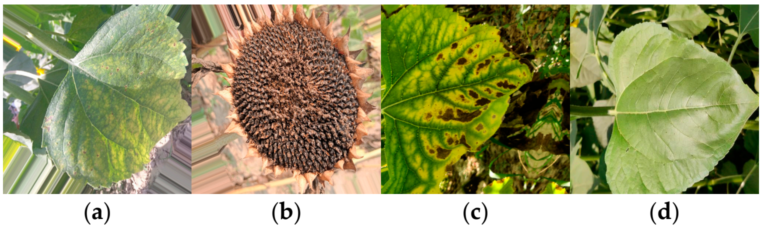


Figure 2. The samples of four classes: (**a**) downy mildew, (**b**) gray mold, (**c**) leaf scar, and (**d**) a fresh leaf.

2.2. Lightweight Deep Learning Models

As deep learning progresses, the parameter size and computational complexity of models continue to increase, which makes it difficult to run these models on mobile devices or embedded systems. Consequently, some researchers have begun to focus on reducing the number of model parameters and computational load, while maintaining model performance. The design intention of lightweight models was to address the deployment challenges faced by large models in resource-constrained equipment.

In this comparative study, we selected five lightweight deep learning models, including SqueezeNet [33], ShuffleNetV2 [35], MnasNet-A1 [36], MobileNetV3-Small [39], and EfficientNetV2-Small [41]. These models were trained on the same image dataset to compare their performances.

2.2.1. SqueezeNet

In 2016, SqueezeNet [33] was proposed by researchers of UC Berkeley and Stanford University. The structure of this model is shown in Figure 3.

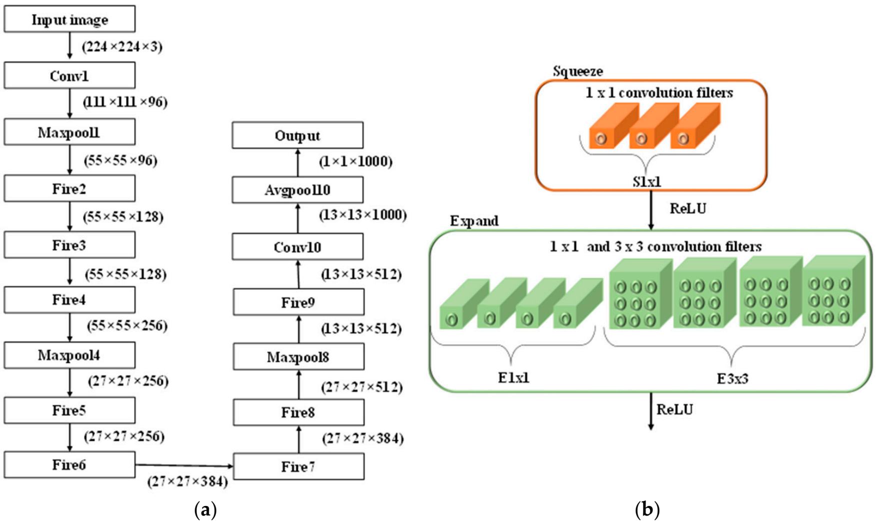


Figure 3. The structures of SqueezeNet and Fire Module [33]: (**a**) SqueezeNet structure. Conv, convolution layer; Fire, fire module; Maxpool, maximum polling layer; (**b**) Fire Module, Squeeze: squeeze layer, S1 × 1: the number of 1 × 1 convolution filters in squeeze layer, ReLU: activation function is ReLU, Expand: expand layer, E1 × 1: the number of 1 × 1 convolution filters in expand layer, E3 × 3: the number of 3 × 3 convolution filters in expand layer.

From Figure 3a, we find that main innovation of SqueezeNet lies in the introduction of the Fire Module [33], which consists of a Squeeze layer and an Expand layer. In Figure 3b, the squeeze layer uses 1 × 1 convolution kernels to reduce the depth of the feature maps, while the expand layer uses 1 × 1 and 3 × 3 convolution kernels to increase the depth of the feature maps. This kind of design effectively reduces the number of model parameters while maintaining high accuracy. According to the research results, this model achieves the same level of accuracy as AlexNet [1] on ImageNet, but with 50 times fewer parameters.

2.2.2. ShuffleNet

ShuffleNetV1 [34] is a CNN architecture with extremely high computational efficiency. It is specifically designed for mobile devices with very limited computing power, and its architecture is shown in Figure 4. This model employs two new operations, namely Pointwise Group Convolution and Channel Shuffle, which greatly reduce the computational cost while maintaining accuracy. On ARM-based mobile devices, ShuffleNetV1 achieves approximately 13 times the actual acceleration compared to AlexNet [1] while maintaining comparable accuracy.

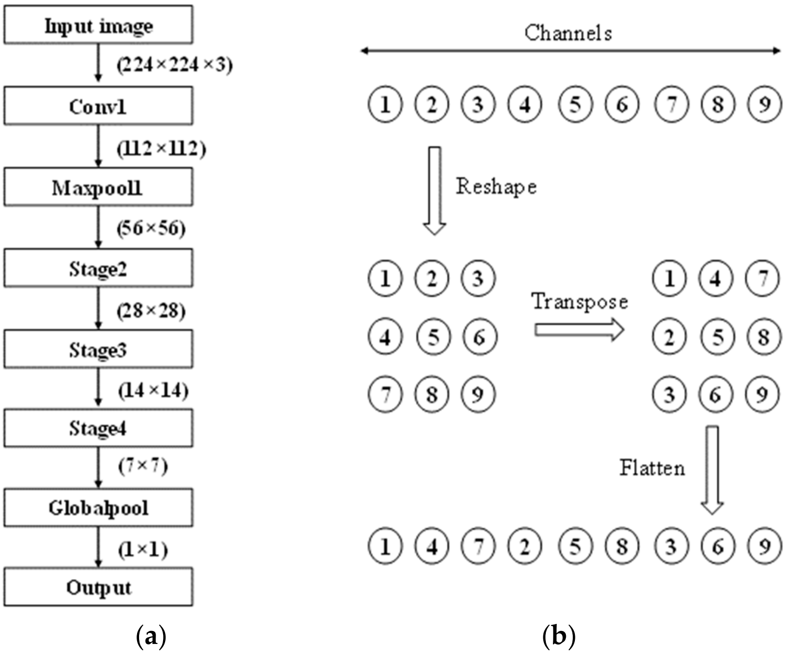
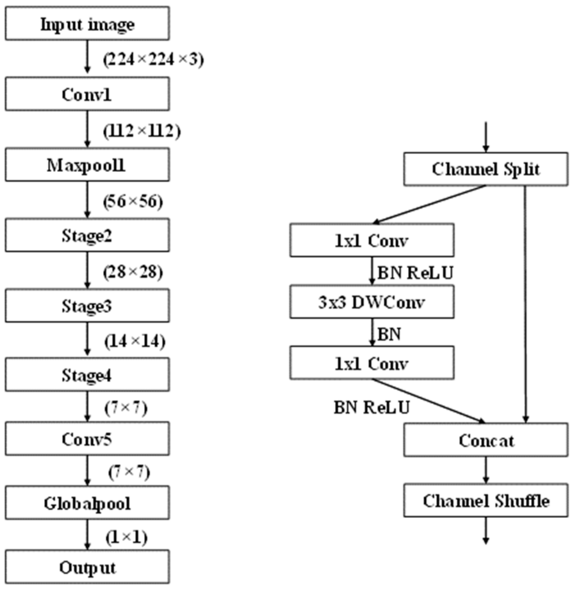


Figure 4. ShuffleNetV1 [34] architecture and the process of channel shuffle: (**a**) the architecture of ShuffleNetV1, Conv: convolution layer, Maxpool: maximum pooling layer; Stage, stage block; Globalpool, global pooling layer. (**b**) An example of channel shuffle with 9 channels.

Compared with ShuffleNetV1, ShuffleNetV2 [35] further optimized the channel shuffle operation. To reduce the memory access cost (MAC), ShuffleNetV2 abandoned the grouped convolutions used in ShuffleNetV1, especially the 1 × 1 grouped convolutions. Meanwhile, ShuffleNetV2 introduced the channel split operation, which divided the input channels into two branches. One branch performed an identity mapping to keep the number of input and output channels unchanged, while the other branch executed multi-layer convolutions to ensure that the number of input and output channels remained equal. The architecture of ShuffleNetV2 [35] and channel split operation are shown in Figure 5.



(**a**) (**b**)

Figure 5. ShuffleNetV2 [35] architecture and the process of channel split: (**a**) the architecture of ShuffleNetV2. Conv, convolution layer; Maxpool, maximum polling layer; Stage, stage block; Globalpool, global pooling layer. (**b**) An example of channel split. 1 × 1 Conv, convolution with 1 × 1 kennel size; 3 × 3 DWConv, deep-wise convolution; BN, batch normalization; ReLU, activation function is ReLU; Concat, tensor concat.

2.2.3. MnasNet

MnasNet [36] is also a lightweight CNN architecture designed for mobile and edge devices, which was proposed by the Google team. There are two relatively important innovations in the MnasNet network. The first one is the application of a multi-objective function, which takes into account both the accuracy of the model and the speed of model fitting. The second one is neural architecture search (NAS), which aims to find a network structure that balances inference time and accuracy through NAS. The architecture of MnasNet-A1 [36] is shown in Figure 6.

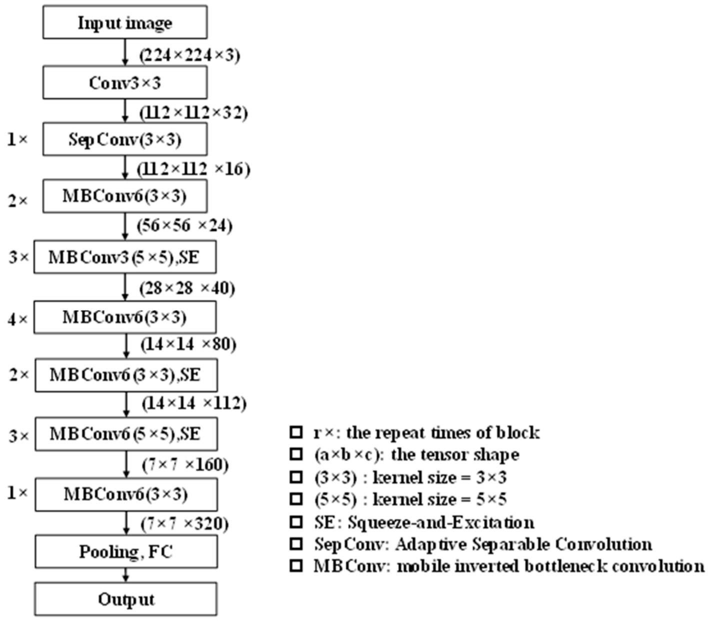


Figure 6. The architecture of MnasNet-A1.

The network mainly consists of seven blocks. Among them, “SepConv” represents depth-wise separable convolution. “MBConv6” represents the inverted residual structure with an expansion factor of 6. “SE” represents the block using an attention mechanism, and (3 × 3) or (5 × 5) represents performing convolution using convolution kernels with a kernel size of 3 × 3 or 5 × 5. The “×” in front of each block indicates the number of times that this block needs to be repeated.

2.2.4. MobileNet

MobileNetV1 [37] was proposed by the Google team. It mainly uses depth-wise separable convolutions to replace standard convolution models, thereby reducing the number of parameters. MobileNetV2 [38] introduced the inverted residual structure with a linear bottleneck on the basis of MobileNetV1. In 2019, the Google team proposed the MobileNetV3 [39] architecture again. It adopted neural architecture search (NAS) to obtain the optimal parameters and modified the bottleneck structure by introducing the Hard-Swish function and the SE module, thus improving the accuracy of the model. The architecture of MobileNetV3-Small [39] is shown in Figure 7.

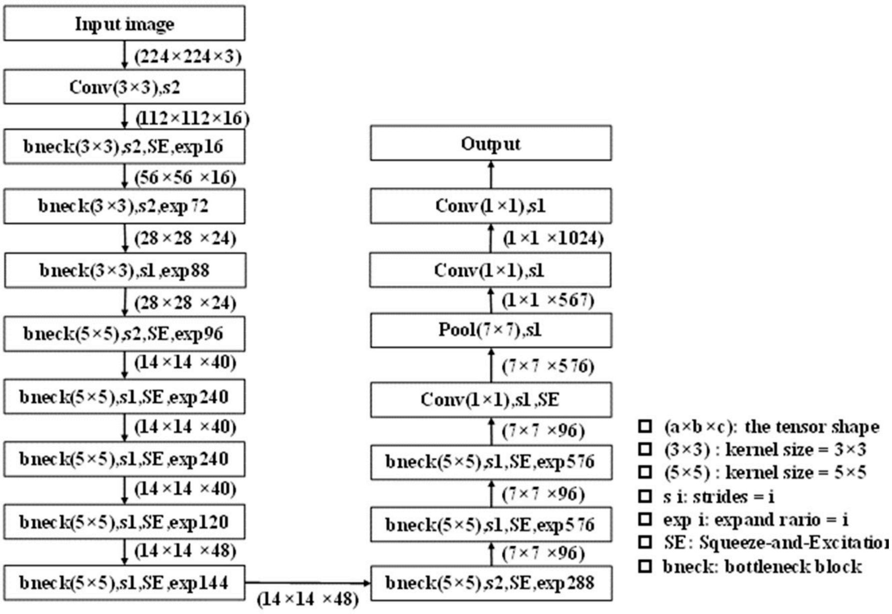


Figure 7. The architecture of MobileNetV3-Small.

2.2.5. EfficientNet

In 2020, Google launched the EfficientNet [40] model. This structure simultaneously changes the width (that is, increases the number of convolution kernels), the depth of the network (that is, uses more layer structures), and the resolution of the input image, hoping to find a balanced network width, depth, and resolution to improve the performance of the network. From EfficientNetB0 to EfficientNetB7, the top-1 accuracy on ImageNet can reach 76.3%, 78.8%, 79.8%, 81.1%, 82.6%, 83.3%, 84.0%, and 84.3%.

EfficientNetV2 [41] was proposed by Google Research and the Brain Team. In this model, a new operator called Fused-MBConv was used, then NAS (neural architecture search) and scaling were combined to optimize the training speed, model accuracy, and parameter size. The EfficientNetV2 model is much faster in training than EfficientNetV1 and is 6.8 times smaller in size, achieving a top-1 accuracy of up to 87.3% on ImageNet. The architecture of EfficientNetV2-Small [41] is shown in Figure 8.

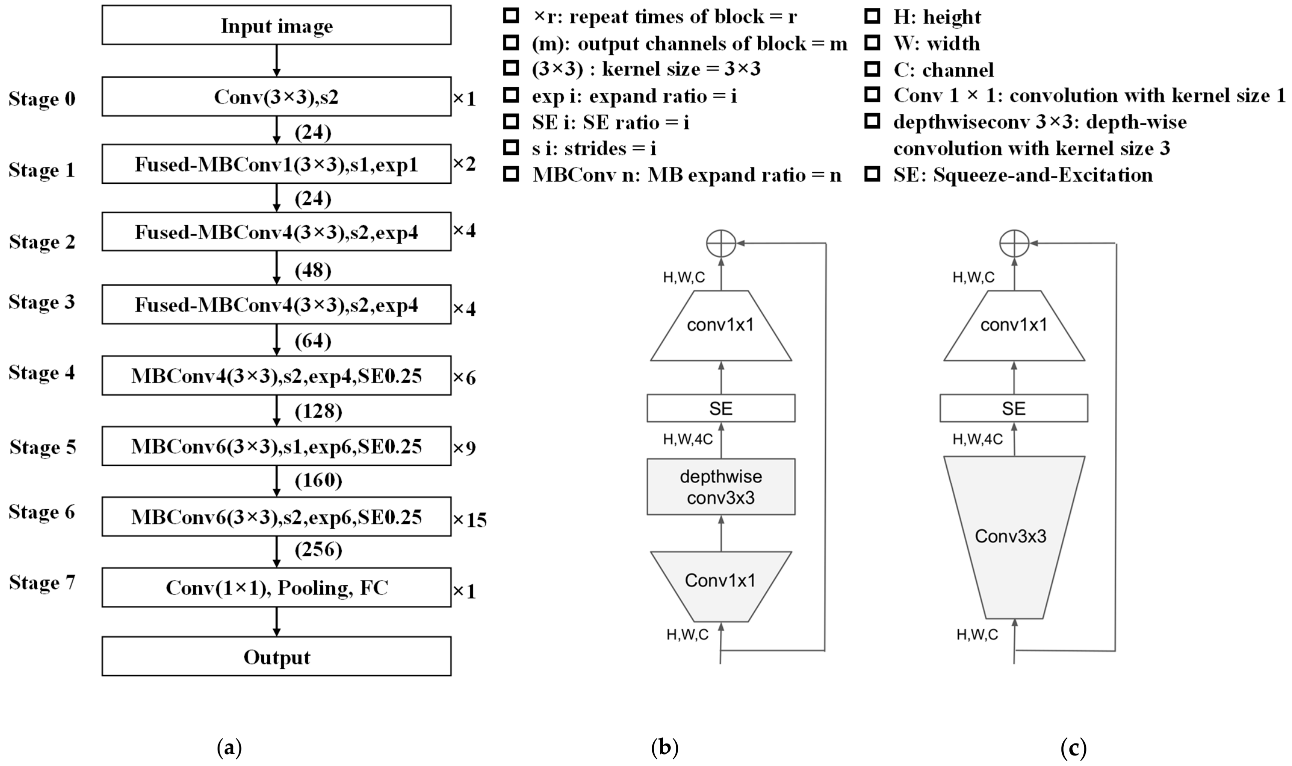

Input image
Stage 0
Conv(3×3),s2
×1
(24)
Stage 1
Fused-MBConv1(3×3),s1,exp1
×2
(24)
Stage 2
Fused-MBConv4(3×3),s2,exp4
×4
(48)
Stage 3
Fused-MBConv4(3×3),s2,exp4
×4
(64)
Stage 4
MBConv4(3×3),s2,exp4,SE0.25
×6
(128)
Stage 5
MBConv6(3×3),s1,exp6,SE0.25
×9
(160)
Stage 6
MBConv6(3×3),s2,exp6,SE0.25
×15
(256)
Stage 7
Conv(1×1), Pooling, FC
×1
Output
□ ×r: repeat times of block = r
□ (m): output channels of block = m
□ (3×3) : kernel size = 3×3
□ exp i: expand ratio = i
□ SE i: SE ratio = i
□ s i: strides = i
□ MBConv n: MB expand ratio = n
□ H: height
□ W: width
□ C: channel
□ Conv 1 × 1: convolution with kernel size 1
□ depthwiseconv 3×3: depth-wise convolution with kernel size 3
□ SE: Squeeze-and-Excitation
H,W,C
conv1x1
SE
H,W,4C
depthwise conv3x3
Conv1x1
H,W,C
H,W,C
conv1x1
SE
H,W,4C
Conv3x3
H,W,C
(a)
(b)
(c)

Figure 8. The architecture of EfficientNetV2-Small and the modules: (**a**) the architecture of EfficientNetV2-Small, (**b**) the MBConv module, and (**c**) the Fused-MBConv module.

The five lightweight deep learning models analyzed above have far fewer parameters than traditional deep learning models, yet they have achieved high accuracy on ImageNet, as shown in Table 4.

Table 4. The performance of five lightweight deep learning models on ImageNet: # of Params, number of trainable parameters; Top-1 Acc, the top-1 accuracy on the ImageNet validation set.

Model Name	# of Params (M)	Top-1 Acc (%)
SqueezeNet [33]	4.76	57.5
ShuffleNetV2 (1x) [35]	8.69	69.4
MnasNet-A1 [36]	3.9	75.2
MobileNetV3-Small [39]	2.9	67.4
EfficientNetV2-Small [41]	22	83.9

2.3. *Experiments Settings and Performance Evaluation Methods*

In this study, our main objective was to find an optimized model among these lightweight models that could classify and identify sunflower disease images effectively. When comparing various models, it was necessary to use the same hyper-parameters, software platform, and evaluation methods. Therefore, we chose to run the program on the GPU server V100S with the CUDA 12.1. The version of the Python environment was 3.10. The python Torch version was 2.1, and the Torch Vision version was 0.16. The hyper-parameters of models were set according to Table 5.

Table 5. Hyper-parameter setting of models.

Parameters	Epochs	Learning Rate	Batch Size
Value	100	0.001	8

To measure the performance of the model in classifying sunflower disease images, we selected accuracy, precision, recall, and F1 score as metrics. During the model testing, we could obtain the results of the classification. In a binary classification task, if the true label of a sample was positive and the model also predicted it as positive, then this sample belonged to a true positive (TP). If the true label of a sample was negative and the model also predicted it as negative, then this sample belonged to a true negative (TN). If the true label of a sample was positive but the model incorrectly predicted it as negative, then this sample belonged to a false negative (FN). If the true label of a sample was negative but the model incorrectly predicted it as positive, then this sample belonged to a false positive (FP). In a multi-class classification task, the concepts of TP, TN, FN, and FP were similar to those in a binary classification problem, but they needed to be calculated separately for each class.

In our task, there were four classes in the dataset, so after model testing, each sample received a recognition result. These results could be entered in a table to form a confusion matrix as shown in Table 6.

Table 6. The definition of confusion matrix.

		True Label			
		Downy Mildew	**Fresh Leaf**	**Gray Mold**	**Leaf Scars**
Predicted label	Downy mildew	C_{11}	C_{21}	C_{31}	C_{41}
	Fresh leaf	C_{12}	C_{22}	C_{32}	C_{42}
	Gray mold	C_{13}	C_{23}	C_{33}	C_{43}
	Leaf scars	C_{14}	C_{24}	C_{34}	C_{44}

In Table 6, each element C_{ij} represents the number of samples whose true label is class i but are predicted as class j. From the confusion matrix, the values of TP_i, TN_i, FN_i, and FP_i for class i could be extracted as shown in Equations (1)–(4).

$$TP_i = C_{ii} \tag{1}$$

$$TN_i = \sum_{j \neq i} C_{ij} \tag{2}$$

$$FP_i = \sum_{j \neq i} C_{ji} \tag{3}$$

$$FN_i = \left(\sum_{i=1}^{4} \sum_{j=1}^{4} C_{ij}\right) - TP_i - TN_i - FP_i \tag{4}$$

The model accuracy refers to the ratio of the number of samples that are accurately identified to the total number of samples, and the formula is shown as Equation (5). For each class, precision, also known as the precision rate, represents the proportion of samples that are actually positive among those predicted as positive, and the formula is shown as Equation (6). Recall, also known as the recall rate, represents the proportion of samples that are predicted as positive among those that are actually positive, and the formula is shown as Equation (7). The F1 score is the harmonic mean of precision and recall, and the calculation method is shown as Equation (8). The i in Equations (6)–(8) represents the ith class.

$$\text{Accuracy} = \frac{\sum_{i=1}^{4} TP_i}{\sum_{i=1}^{4} (TP_i + FP_i + TN_i + FN_i)} \tag{5}$$

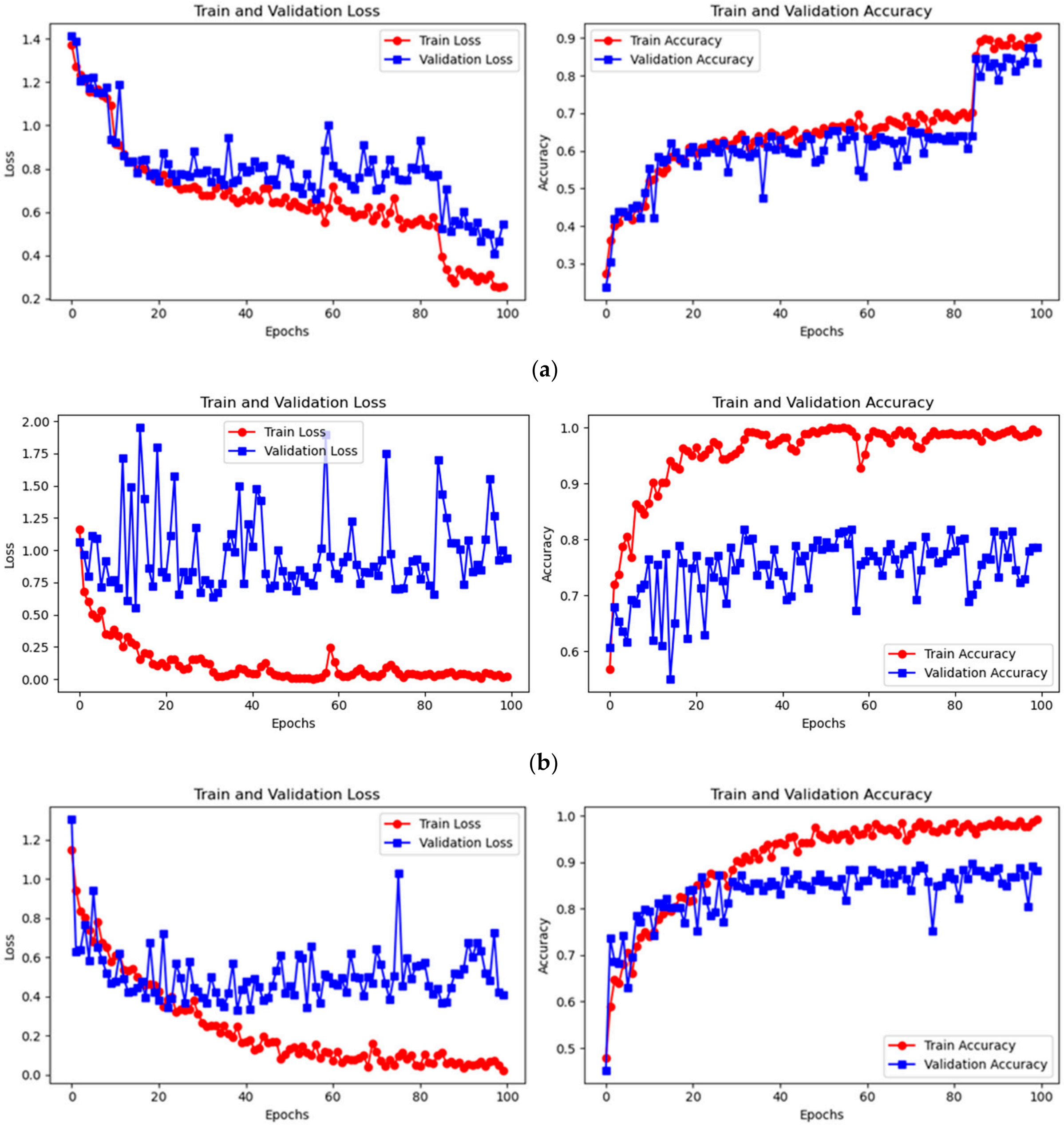
$$\text{Precision}_i = \frac{TP_i}{TP_i + FP_i} \tag{6}$$

$$\text{Recall}_i = \frac{TP_i}{TP_i + FN_i} \tag{7}$$

$$\mathrm{F1-score_i} = 2 \times \frac{\mathrm{Recall_i} \times \mathrm{Precision_i}}{\mathrm{Recall_i} + \mathrm{Precision_i}} \tag{8}$$

3. Results

In this section, we will present the performance evaluation results of five models, which were SqueezeNet, ShuffleNetV2, MnasNet-A1, MobileNetV3-Small, and EfficientNetV2-Small, in the classification and recognition of sunflower disease images. During model training, we tracked the convergence of the models and drew convergence graphs from four aspects: training accuracy, validation accuracy, training loss, and validation loss. In the experiment, we set some parameters for training the model according to Table 5. The training process data are shown in Figure 9.

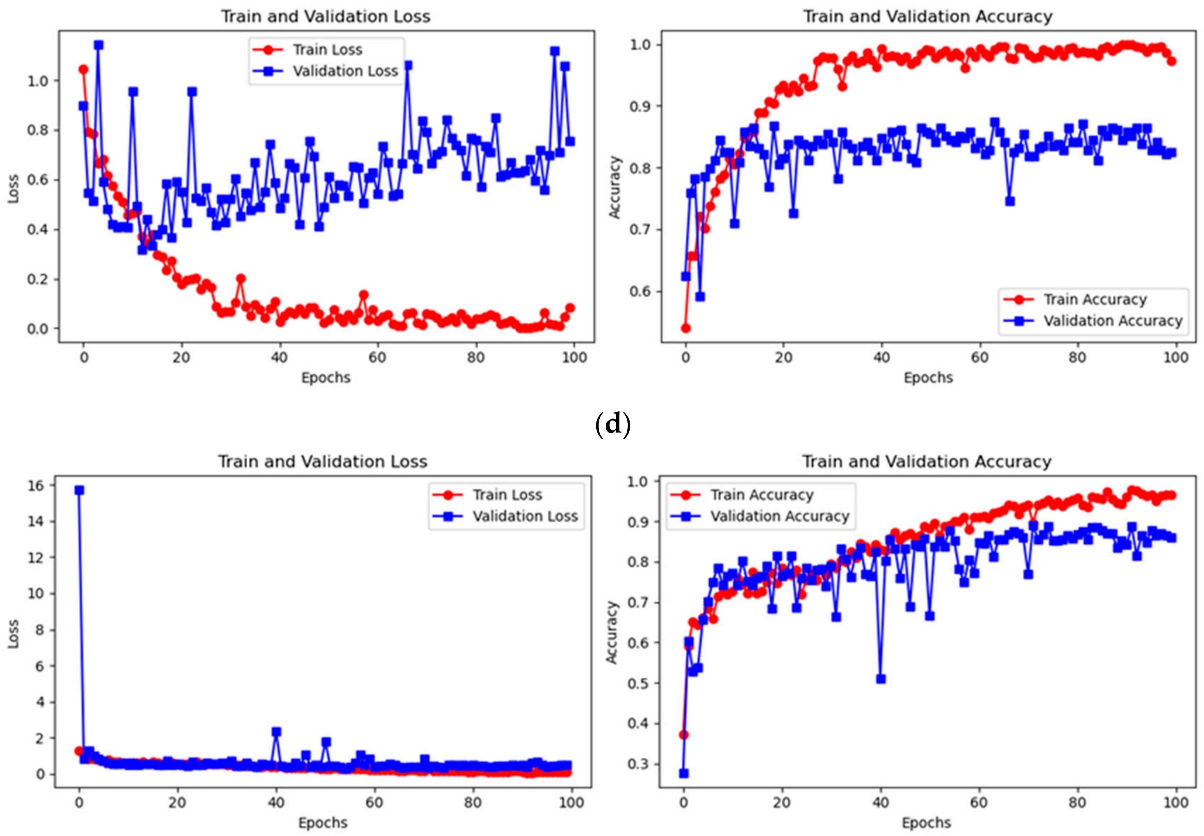


(**a**)

(**b**)

(**c**)

Figure 9. *Cont.*



(**d**)

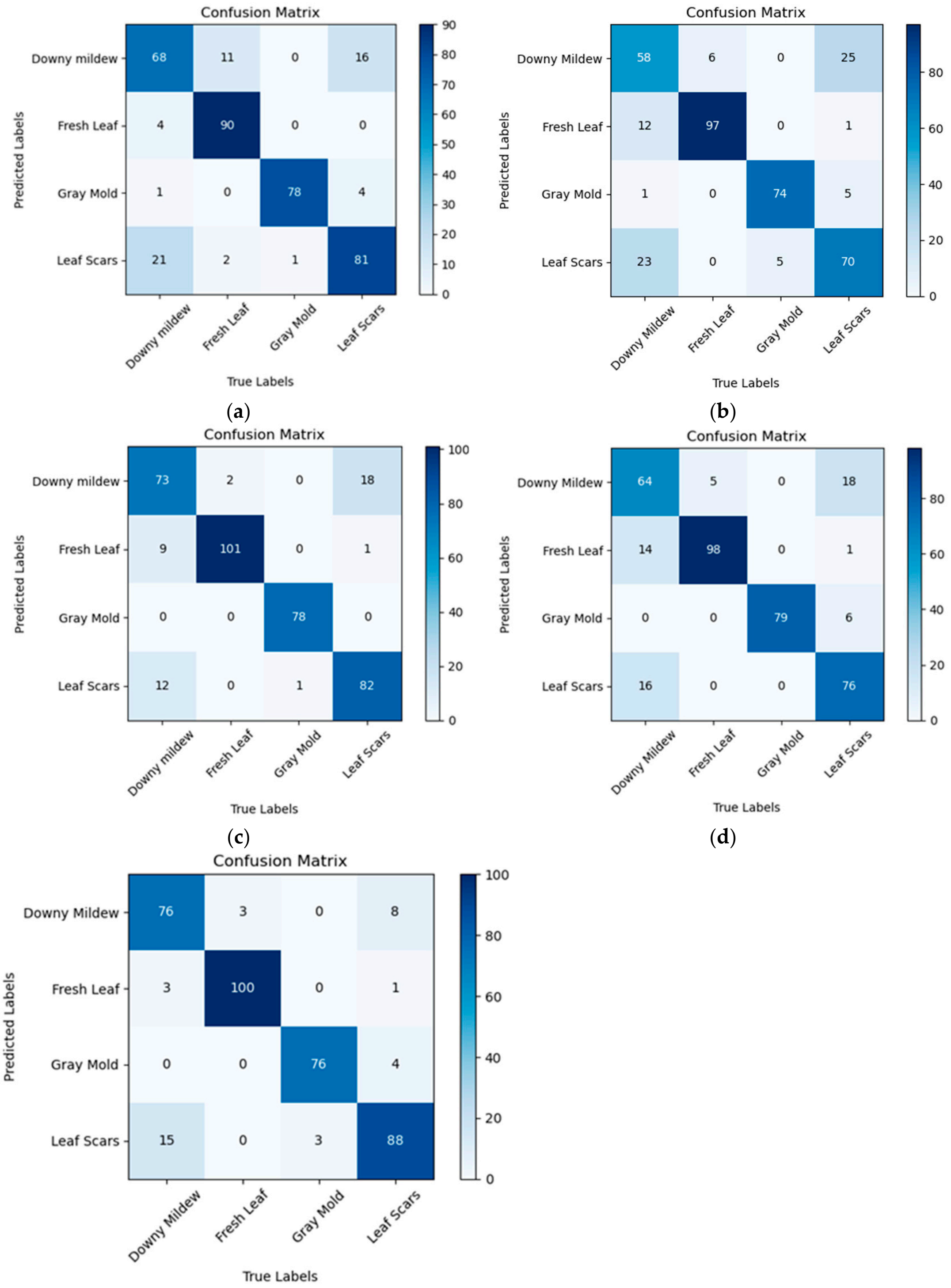
(**e**)

Figure 9. The convergence process of five models. The number of images in overall dataset was 1868. We split this dataset into training set and testing data with a ratio of 8:2. Then the training set was used to train models. The ratio of training images and validation images was 8:2, too: (**a**) SqueezeNet, (**b**) ShuffleNetV2, (**c**) MnasNet-A1, (**d**) MobileNetV3-Small, and (**e**) EfficientNetV2-Small.

In Figure 9, we present the changes of the training accuracy, validation accuracy, training loss, and validation loss of these five models. It can be seen that the training results of these five models were different.

Figure 9a shows that on the sunflower dataset, the SqueezeNet model had a relatively slow convergence speed. During epoch 20 to 80, the training and validation accuracy changed very little and did not exceed 70%. After 100 epochs, the training and validation accuracies of the SqueezeNet model did not exceed 90%. Figure 9b shows that on the sunflower dataset, the training accuracy of the ShuffleNetV2 model was much higher than the validation accuracy. The training loss of ShuffleNetV2 was less than the validation loss. After 100 epochs, the training and validation accuracies of the model exceeded 90%, but there was a certain gap between the training accuracy and the validation accuracy. Figure 9c,d show that on the sunflower dataset, there was a relatively large gap between the training accuracy and the validation accuracy of MnasNet-A1 and MobileNetV3-Small models. The training accuracy is nearly 99%, but the validation accuracy failed to reach 90%, and both losses were quite large. Models experienced overfitting. Figure 9e shows a good performance for EfficientNetV2-Small. The training accuracy was nearly 99%, and the validation accuracy was nearly 90%. The gap between the training accuracy and validation accuracy was lower than that of ShuffleNetV2, MnasNet-A1, and MibileNetV3-Small. Both the training and validation losses of EfficientNetV3-Small were less than 0.1.

To further verify the performances of these five models, we used the weight parameters corresponding to the highest training accuracy during the model training to conduct tests on the test set and obtained the confusion matrix, as shown in Figure 10.



(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 10. The confusion matrix of five models: the x axis is the true label, and the y axis is the predicted label. (**a**) SqueezeNet, (**b**) ShuffleNetV2, (**c**) MnasNet-A1, (**d**) MobileNetV3-Small, and (**e**) EfficientNetV2-Small.

Based on the confusion matrixes of each model, the identification accuracies of the five models, namely, SqueezeNet, ShuffleNetV2, MnasNet-A1, MobileNetV3-Small, and

EfficientNetV2-Small, on the test set could be calculated as 84.08%, 79.31%, 88.59%, 84.08%, and 90.19%, respectively. The accuracy, precision, recall, and F1 score of each category of images are presented in Table 7.

Table 7. The precision, recall, F1 score, and accuracy of each class of five models.

Models	Class	Precision	Recall	F1 Score	Accuracy (%)
SqueezeNet	Downy mildew	0.715	0.723	0.719	72.34
	Fresh leaf	0.957	0.874	0.914	87.38
	Gray mold	0.940	0.987	0.963	98.73
	Leaf scars	0.771	0.802	0.786	80.20
ShuffleNetV2	Downy mildew	0.734	0.617	0.67	60.70
	Fresh leaf	0.882	0.942	0.911	94.17
	Gray mold	0.925	0.937	0.931	93.67
	Leaf scars	0.714	0.693	0.703	69.31
MnasNet-A1	Downy mildew	0.785	0.777	0.781	77.66
	Fresh leaf	0.910	0.981	0.944	98.06
	Gray mold	1.0	0.987	0.993	98.73
	Leaf scars	0.863	0.812	0.837	81.19
MobileNetV3-Small	Downy mildew	0.736	0.681	0.707	68.09
	Fresh leaf	0.867	0.951	0.907	95.15
	Gray mold	0.929	1.0	0.963	100
	Leaf scars	0.826	0.752	0.787	75.25
EfficientNetV2-Small	Downy mildew	0.874	0.809	0.840	80.85
	Fresh leaf	0.962	0.971	0.966	97.09
	Gray mold	0.950	0.962	0.956	96.20
	Leaf scars	0.854	0.871	0.862	87.13

In our sunflower disease image classification research, the dataset was relatively small, with only 1892 images. Transfer learning [31] becomes crucial in this context. Transfer learning is based on the principle that knowledge learned in one domain can be transferred to another related domain. By leveraging pre-trained models on large-scale datasets like ImageNet, we could take advantage of the general patterns and features they learned. These pre-trained models were exposed to a vast number of images, enabling them to capture diverse visual concepts such as edges, textures, and object shapes. For our small-scale sunflower dataset, transfer learning allowed the model to quickly adapt to the specific characteristics of sunflower diseases, which significantly improved the model's generalization ability. Without transfer learning, training a model from scratch on our limited data would likely lead to overfitting and poor performance. To adapt to the four-class classification of sunflower disease images, we performed the following steps for model adjustment:

- First, instantiate the model.
- Modify the parameters of the classification layer of the instantiated model to match the four-class sunflower disease image classification task.
- Read the pre-trained weights provided by PyTorch official for the models on ImageNet, and also read the existing model weights.
- Screen out the parameters whose keys and shapes match between the pre-trained model and the new classification layer setup. These are the parameters that will be updated to better suit our specific task.
- Screen out the parameters whose keys and shapes match between the pre-trained model and the new classification layer setup.
- Update the model weights using the selected parameters.

- Load the updated model weights back into the model. Then, conduct the model training process with our sunflower disease image dataset.

The overall performance of five models in test dataset before and after incorporating transfer learning is given in Table 8. The average precision, recall, F1 score, and accuracy in Table 8 are the arithmetic mean of the precision, recall, F1 score, and accuracy obtained for each class of images in the test set.

Table 8. The average precision, recall, F1 score, and accuracy of four classes of image identification in the test dataset (including 377 images) of five models before and after transfer learning.

	Models	Average Precision	Average Recall	Average F1-Score	Average Accuracy (%)
Before transfer learning	SqueezeNet	0.846	0.847	0.846	84.08
	ShuffleNetV2	0.793	0.797	0.795	79.31
	MnasNet-A1	0.890	0.889	0.889	88.59
	MobileNetV3-Small	0.840	0.846	0.841	84.08
	EfficientNetV2-Small	0.904	0.903	0.903	90.19
After transfer learning	SqueezeNet	0.962	0.961	0.961	96.02
	ShuffleNetV2	0.955	0.954	0.954	95.23
	MnasNet-A1	0.953	0.951	0.952	94.96
	MobileNetV3-Small	0.969	0.970	0.969	96.82
	EfficientNetV2-Small	0.992	0.992	0.992	99.20

According to the results given in Table 8, after incorporating transfer learning, EfficientNetV2-Small achieved the highest test accuracy of 99.20% among the five models. MnasNet-A1 achieved the lowest classification accuracy of 94.96%. Compared with the accuracy before incorporating transfer learning, the accuracies of these five models were improved by 14.2%, 20%, 7.2%, 15.2%, and 10%.

4. Discussion

In this study, we delved into the characteristics and architectures of five lightweight models and compared their performance in terms of accuracies, precisions, recalls, and F1 scores on the sunflower disease image dataset. The evaluated models included SqueezeNet, ShuffleNetV2, MnasNet-A1, MobileNetV3-Small, and EfficientNetV2-Small. Initially, these models achieved sunflower disease identification accuracies of 84.08%, 79.31%, 88.59%, 84.08%, and 90.19%, respectively, before incorporating transfer learning. To address the challenge of a relatively small dataset and enhance the accuracy of each model, we employed pre-training models and the transfer learning method. Consequently, the accuracies of these models improved by 14.2%, 20%, 7.2%, 15.2%, and 10%, reaching 96.02%, 95.23%, 94.96%, 96.82%, and 99.20%. Upon comparing the performances of the five models, we found that EfficientNetV2-Small outperformed the other four models in terms of precision, recall, F1 score, and accuracy.

Yi Zhong and MengJun Tong [19] designed a module to extract multi-frequency multi-scale features and proposed lightweight dual-fusion attention and multi-branching structure to identify sunflower disease. This method named TeenyNet obtained the highest accuracy of 98.94%. SunNet [30] is a deep learning approach for detecting sunflower diseases. The study investigated various disease classification techniques on 1428 sunflower leaf images and found that the best-performing classifier overall was a ResNet50 classifier, which achieved an average accuracy of 97.88%. Yunis Gulzar et al. [31] compared the performance of five models in the identification of sunflower diseases, such as AlexNet, VGG16, InceptionV3, MobileNetV3, and EfficientNetB3, and finally found that Efficient-

NetB3 had the highest accuracy of 97.9%. The lightweight model DFN-PSAN [32] achieved an accuracy of 94.47% on the sunflower dataset. The dataset sizes used in these methods are shown in Table 9.

Table 9. The dataset used in TeenyNet [19], VGG19+CNN [29], SunNet [30], EfficientNetB3 [31], DFN-PSAN [32], and ours.

Method	# of Total Images	The Ratio of Training Set to Test Set
TeenyNet [19]	1668	90:10
VGG19+CNN [29]	467	80:20
SunNet [30]	1428	90:10
EfficientNetB3 [31]	1892	85:15
DFN-PSAN [32]	1892	90:10
EfficientNetV2-Small (ours)	1892	80:20

Compared with TeenyNet [19], VGG19+CNN [29], SunNet [30], EfficientNetB3 [31], and DFN-PSAN [32], which are also used for detecting sunflower disease, EfficientNetV2-Small (in our study) also achieved the best accuracy. The comparison graph is shown in Figure 11.

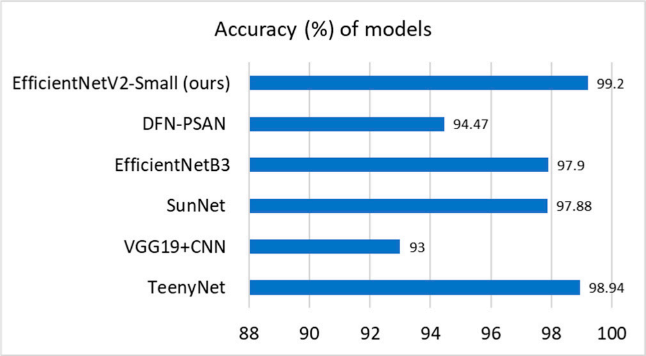


Figure 11. The accuracy of EfficientNetV2-Small (in our study), TeenyNet [19], VGG19+CNN [29], SunNet [30], EfficientNetB3 [31], and DFN-PSAN [32].

From Figure 11, we found the accuracy of EfficientNetV2-Small reached the highest accuracy of 99.2% in these models.

There were several reasons why EfficientNetV2-Small achieved a relatively high recognition accuracy. First, EfficientNetV2-Small employed the method of compound scaling. By uniformly adjusting the model's depth, width, and image resolution, it achieved the optimal balance between performance and efficiency. This scaling method allowed the model to leverage the benefits of increased capacity while maintaining computational efficiency, which was crucial for lightweight models designed for real-time applications.

Second, EfficientNetV2-Small incorporated variants of MobileNetV3-Small and Fused-MBConv. This block replaced the depth-wise separable convolution with standard convolution operations, thereby reducing the computational costs. This design enhanced the model's ability to capture complex features while maintaining a low computational footprint, making it suitable for deployment on devices with limited resources.

Third, EfficientNetV2-Small benefited from an efficient architecture discovered through neural architecture search (NAS). During this search process, the optimal network configurations, such as the size of the convolution kernel and the expansion ratio, were automatically determined to achieve specific performance goals. This data-driven approach ensured that the model was optimized for the given task, leading to improved performance metrics.

Last, the utilization of transfer learning showed significant potential for precision agriculture. By leveraging a pre-trained model on the ImageNet dataset, we obtained a robust method of feature extraction and applied it to the recognition of sunflower disease images. This approach not only mitigated the challenge of limited labeled data but also enhanced the model's ability to generalize to new, unseen data.

5. Conclusions

This study demonstrated the effectiveness of lightweight models in the classification and recognition of sunflower disease images, with EfficientNetV2-Small emerging as the superior model. The combination of compound scaling, optimized convolutional blocks, architecture search, and transfer learning enabled EfficientNetV2-Small to achieve high precision, recall, F1 score, and accuracy, making it a promising candidate for real-world applications in precision agriculture.

The image dataset used in this study was relatively small in scale, covering only four types of sunflower diseases, and the number of images for each type was limited. Such limitations of the dataset may lead to the following problems:

- Risk of overfitting: Due to the limited amount of data, the model may overfit the training data during the training process, resulting in poor performance on new, unseen data. This, to some extent, restricts the generalization ability of the model.
- Lack of diversity: The insufficient diversity of the dataset may fail to cover all possible disease scenarios and environmental conditions, thereby affecting the robustness of the model in practical applications.
- Insufficient representativeness: With a small number of images in the dataset, it may not fully represent the characteristics of sunflower diseases in different regions, seasons, and growth stages, leading to biases in the model's practical application.

Moving forward, we plan to expand our research based on the findings of this study. One important direction is to expand the dataset by collecting additional images of sunflower diseases. The dataset used in this study was relatively small and not locally captured in China. To ensure the applicability of our research results in local areas, it is essential to gather more diverse and locally relevant image data. This will not only improve the robustness of our models but also enhance their ability to generalize to different environmental conditions. Another approach is to explore feature extraction methods to identify subtle differences in easily confused disease images. The confusion matrix we obtained showed that downy mildew and leaf scar were two diseases that were easily confused. Therefore, it is necessary to further explore the subtle features of the images to improve the accuracy of the model.

Another avenue for future work is to develop applications and deploy them on mobile terminals or other embedded terminals, so as to achieve real-time image collection and recognition in the fields. This will enable farmers to perform real-time disease identification in the field by using these mobile terminals, facilitating timely intervention and management. We believe that through these continued efforts, the accuracy of sunflower disease identification can be further improved, ultimately contributing to reduce the impact of diseases on crop yields and increase economic benefits for farmers.

This research also had limitations. During model training, the used images were sourced from abroad rather than collected in local areas. Therefore, the disease traits and characteristics may differ from those of sunflower diseases in local area. This could lead to inaccuracies in our model when classifying local sunflower diseases.

Consequently, in order to promote the use of this model locally, it is necessary to collect a large number of images of various local sunflower diseases, fine-tune the model, and conduct in-depth training to obtain a lightweight deep-learning model that conforms to the characteristics of local sunflower diseases. This is also the direction of our future work.

Author Contributions: Conceptualization, methodology, software, writing—original draft preparation, and writing—review and editing, L.Z.; validation, X.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Interdisciplinary Research Fund of Inner Mongolia Agricultural University under Grant No. BR231506. This research also was supported by the Inner Mongolia Autonomous Region Key Laboratory of Big Data Research and Application of Agriculture and Animal Husbandry (Hohhot, Inner Mongolia, China).

Data Availability Statement: We used a public dataset [13].

Acknowledgments: Thanks to the researchers who have provided us with the open datasets.

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

References

1. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Hinton. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
2. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556. [CrossRef]
3. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016. [CrossRef]
4. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018. [CrossRef]
5. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269. [CrossRef]
6. Góngora, C.E.; Silva, M.d.C. Sustainable Strategies for the Control of Crop Diseases and Pests to Reduce Pesticides. *Agronomy* **2024**, *14*, 2158. [CrossRef]
7. Puttha, R.; Venkatachalam, K.; Hanpakdeesakul, S.; Wongsa, J.; Parametthanuwat, T.; Srean, P.; Pakeechai, K.; Charoenphun, N. Exploring the Potential of Sunflowers: Agronomy, Applications, and Opportunities within Bio-Circular-Green Economy. *Horticulturae* **2023**, *9*, 1079. [CrossRef]
8. Production Volume of Sunflower Seed in Major Producer Countries in 2023/2024. Available online: https://www.statista.com/statistics/263928/production-of-sunflower-seed-since-2000-by-major-countries/#:~:text=Sunflower%20seed%20production%20in%20major%20countries%202022/2023&text=During%20that%20time%20period,%20Russia,metric%20tons%20in%202022/2023 (accessed on 10 January 2025).
9. Sravanthi, G.; Moparthi, N.R. An efficient IoT based crop disease prediction and crop recommendation for precision agriculture. *Clust. Comput.* **2024**, *27*, 5755–5782. [CrossRef]
10. Qi, L.L.; Seiler, G.J. Registration of HA-DM15 and HA-DM16 oilseed sunflower germplasms with resistance to sunflower downy mildew. *J. Plant Regist.* **2023**, *18*, 173–178. [CrossRef]
11. Tomioka, K.; Sato, T. Gray mold of yacon and sunflower caused by Botrytis cinerea. *J. Gen. Plant Pathol.* **2011**, *77*, 217–219. [CrossRef]
12. Sui, L.; Lu, Y.; Yang, H.; Liu, Z.; Wang, L.; Zou, X.; Li, Q.; Zhang, Z. Endophytic Beauveria bassiana promotes sunflower growth and confers resistance against sclerotinia disease. *BioControl* **2024**, *70*, 119–130. [CrossRef]
13. Sara, U.; Rajbongshi, A.; Shakil, R.; Akter, B.; Sazzad, S.; Uddin, M.S. An extensive sunflower dataset representation for successful identification and classification of sunflower diseases. *Data Brief* **2022**, *42*, 108043. [CrossRef]
14. Markell, S.G.; Harveson, R.M.; Block, C.C.; Gulya, T.J. 4-Sunflower Diseases. In *Sunflower*; AOCS Press: Champaign, IL, USA, 2015; pp. 93–128. [CrossRef]
15. Berghuis, B.; Friskop, A.; Gilley, M.; Halvorson, J.; Hansen, B.; Fitterer, S.; Carruth, D.; Schatz, B.; Benson, B.; Humann, R.; et al. Evaluation of Fungicide Efficacy on Sunflower Rust (Puccinia helianthi) on Oilseed and Confection Sunflower. *Plant Health Prog.* **2022**, *23*, 140–146. [CrossRef]

16. SS, V.C.; Hareendran, A.; Albaaji, G.F. Precision farming for sustainability: An agricultural intelligence model. *Comput. Electron. Agric.* **2024**, *226*, 109386. [CrossRef]
17. Yang, H.; Deng, X.; Shen, H.; Lei, Q.; Zhang, S.; Liu, N. Disease Detection and Identification of Rice Leaf Based on Improved Detection Transformer. *Agriculture* **2023**, *13*, 1361. [CrossRef]
18. Francis, M.; Deisy, C. Disease detection and classification in agricultural plants using convolutional neural networks—A visual understanding. In Proceedings of the 2019 IEEE 6th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 7–8 March 2019; pp. 1063–1068. [CrossRef]
19. Zhong, Y.; Tong, M. TeenyNet: A novel lightweight attention model for sunflower disease detection. *Meas. Sci. Technol.* **2024**, *35*, 035701. [CrossRef]
20. Raju, K.K.; Sudhakar Ilango, S. A survey on training issues in chili leaf diseases identification using deep learning techniques. *Procedia Comput. Sci.* **2023**, *218*, 2123–2132. [CrossRef]
21. Islam, M.M.; Adil, M.A.A.; Talukder, M.A.; Ahamed, M.K.U.; Uddin, M.A.; Hasan, M.K.; Debnath, S.K. DeepCrop: Deep learning-based crop disease prediction with web application. *J. Agric. Food Res.* **2023**, *14*, 100764. [CrossRef]
22. Ma, L.; Yu, Q.; Yu, H.; Zhang, J. Maize Leaf Disease Identification Based on YOLOv5n Algorithm Incorporating Attention Mechanism. *Agronomy* **2023**, *13*, 521. [CrossRef]
23. Kumar, S.; Ratan, R.; Desai, J.V. Cotton Disease Detection Using TensorFlow Machine Learning Technique. *Adv. Multimed.* **2022**, *2022*, 1812025. [CrossRef]
24. Chen, S.; Lv, F.; Huo, P. Improved detection of yolov4 sunflower leaf diseases. In Proceedings of the 2021 2nd International Symposium on Computer Engineering and Intelligent Communications (ISCEIC), Nanjing, China, 6–8 August 2021. [CrossRef]
25. Sirohi, A.; Malik, A. A Hybrid Model for the Classification of Sunflower Diseases Using Deep Learning. In Proceedings of the 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), London, UK, 28–30 April 2021; pp. 58–62. [CrossRef]
26. Kainat, J.; Ullah, S.S.; Alharithi, F.S.; Alroobaea, R.; Hussain, S.; Nazir, S. Blended Features Classification of Leaf-Based Cucumber Disease Using Image Processing Techniques. *Complexity* **2021**, *2021*, 9736179. [CrossRef]
27. Zhong, Y.; Zhao, M. Research on deep learning in apple leaf disease recognition. *Comput. Electron. Agric.* **2020**, *168*, 105146. [CrossRef]
28. Sethy, P.K.; Barpanda, N.K.; Rath, A.K.; Behera, S.K. Deep feature based rice leaf disease identification using support vector machine. *Comput. Electron. Agric.* **2020**, *175*, 105527. [CrossRef]
29. Ghosh, P.; Mondal, A.K.; Chatterjee, S.; Masud, M.; Meshref, H.; Bairagi, A.K. Recognition of Sunflower Diseases Using Hybrid Deep Learning and Its Explainability with AI. *Mathematics* **2023**, *11*, 2241. [CrossRef]
30. Sathi, T.A.; Hasan, A.; Alam, M.J. SunNet: A Deep Learning Approach to Detect Sunflower Disease. In Proceedings of the 2023 7th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 11–13 April 2023; pp. 1210–1216. [CrossRef]
31. Gulzar, Y.; Ünal, Z.; Aktaş, H.; Mir, M.S. Harnessing the Power of Transfer Learning in Sunflower Disease Detection: A Comparative Study. *Agriculture* **2023**, *13*, 1479. [CrossRef]
32. Dai, G.; Tian, Z.; Fan, J.; Sunil, C.K.; Dewi, C. DFN-PSAN: Multi-level deep information feature fusion extraction network for interpretable plant disease classification. *Comput. Electron. Agric.* **2024**, *216*, 108481. [CrossRef]
33. Iandola, F.N. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360. [CrossRef]
34. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. *arXiv* **2017**, arXiv:1707.01083. [CrossRef]
35. Ma, N.; Zhang, X.; Zheng, H.-T.; Sun, J. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*; Springer: Cham, Switzerland, 2018; pp. 11648–11656. [CrossRef]
36. Tan, M.; Chen, B.; Pang, R.; Vasudevan, V.; Sandler, M.; Howard, A.; Le, Q.V. MnasNet-A1: Platform-Aware Neural Architecture Search for Mobile. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019. [CrossRef]
37. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017. [CrossRef]
38. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520. [CrossRef]
39. Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324. [CrossRef]

40. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114. [CrossRef]
41. Tan, M.; Le, Q.V. EfficientNetV2: Smaller Models and Faster Training. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021. [CrossRef]

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