




## Article

# Crop Guidance Photography Algorithm for Mobile Terminals

Yunsong Jia , Qingxin Zhao , Yi Xiong , Xin Chen  and Xiang Li \*

College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China; jia.ys@cau.edu.cn (Y.J.); qingxinxi99@outlook.com (Q.Z.); xyyyy0428@gmail.com (Y.X.); chxin@cau.edu.cn (X.C.)  
\* Correspondence: cqlixiang@cau.edu.cn; Tel.: +86-1561-136-2365

**Abstract:** The issues of inadequate digital proficiency among agricultural practitioners and the suboptimal image quality captured using mobile smart devices have been addressed by providing appropriate guidance to photographers to properly position their mobile devices during image capture. An application for crop guidance photography was developed, which involved classifying and identifying crops from various orientations and providing guidance prompts. Three steps were executed, including increasing sample randomness, model pruning, and knowledge distillation, to improve the MobileNet model for constructing a smartphone-based orientation detection model with high accuracy and low computational requirements. Subsequently, the application was realized by utilizing the classification results for guidance prompts. The test demonstrated that this method effectively and seamlessly guided agricultural practitioners in capturing high-quality crop images, providing effective photographic guidance for farmers.

**Keywords:** guidance prompts; lightweight; mobilenet model; orientation detection

## 1. Introduction

Algorithms such as target detection [1], crop recognition [2], pest and disease identification [3], and phenotype analysis [4] in farmland all require the use of crop images. The clarity and degree of distortion in image capture, the accuracy of brightness and color tone, the significance of the captured subject in the image [5], and the correct orientation of the subject within the image [5] all directly impact the recognition performance of these algorithms. Problems related to image distortion and color accuracy can be addressed by improving the quality of the camera. However, problems related to shooting distance and angles are difficult for fixed cameras, while they are easy for movable cameras, as the former cannot be adjusted once installed. Moreover, in practical agricultural production environments, fixed cameras are generally not deployed throughout the entire planting area due to high cost and maintenance difficulties. Therefore, it is challenging to meet the complex capturing needs with fixed cameras, and using mobile phones to capture crop images [6] is a more accessible approach for widespread adoption. At present, when using mobile phones for photography, the angles and distances, based on people's intuition and preferences and factors such as low professional skills and a lack of responsibility, also contribute to issues in the captured images, such as improper orientation, off-center composition, or size anomalies. As a result, the image quality from mobile devices typically used by agricultural workers is generally subpar. These lead to a significant decrease in algorithm recognition accuracy and make it challenging to achieve the desired outcomes. Improving the way images are captured and their perspectives before using other image algorithms can lead to the following benefits: first, it can maximize the highlighting of the main features of crops, thereby enhancing the accuracy of crop classification [7,8] and detection [9] algorithms. Second, aligning the camera vertically with the measuring plane can bring the measured length or area closer to the actual values, thus improving the accuracy of phenotype analysis [10]. Third, standardizing the way agricultural workers capture data can enhance the image quality in agricultural datasets [11], making it easier



**Citation:** Jia, Y.; Zhao, Q.; Xiong, Y.; Chen, X.; Li, X. Crop Guidance Photography Algorithm for Mobile Terminals. *Agriculture* **2024**, *14*, 271. <https://doi.org/10.3390/agriculture14020271>

Academic Editors: Francesco Marinello and Valentin Vlăduț

Received: 22 December 2023

Revised: 31 January 2024

Accepted: 5 February 2024

Published: 7 February 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

for models to be trained to achieve better results. In conclusion, guiding the process of taking photographs is of great significance in improving the effectiveness of image-related agricultural intelligent algorithms.

This study proposes a crop guidance algorithm that aims to improve the quality of agricultural photos by providing prompts and instructions on mobile devices. The algorithm is designed to enable agricultural workers to find suitable shooting positions and angles, while the mobile device captures the crops at the appropriate time, ensuring clear features, proper size, and correct posture in the captured images. The research focuses on four key requirements: high accuracy, smooth operation on mobile devices, low latency with no lag, and the ability to handle high usage pressure. By incorporating these elements, the algorithm effectively guides agricultural workers in taking photos and enhances the overall quality of the images captured.

At the current stage, the prompts and intelligent control methods for adjusting camera poses can generally be divided into two categories: human-machine collaborative control [12] and pure machine control [13,14]. They are often applied in intelligent camera control scenarios such as robot control [14], drone control [13,15], sports and artistic photography [16], etc. For example, they can be used for tasks like express package sorting [17], moving object monitoring [18], and subject tracking [12,18]. These methods can automatically track specific targets set by humans or appearing in the field of view, and guide either machines or humans to adjust the camera poses to keep them within the field of view. In human-machine collaborative control, machines play the role of guiding operators to adjust the poses and perform semi-automatic locking and fine-tuning. With predefined targets, machines can be locked through human instructions or existing algorithms. Wang et al. [12] used a short-focus camera to display the target framing box and a long-focus camera for preview images, guiding users to quickly lock the target object within the long-focus lens. Xie et al. [16] designed an aerial photography algorithm that can complete flight guidance and local camera movement control based on predefined viewpoints, constructing the optimal global camera trajectory for observing a series of target landmarks. In pure machine control, machines take full control of the photography equipment, automatically searching, locking, and tracking targets, and adjusting the position and posture of the camera in real time to achieve appropriate shooting positions and angles for capturing, recording, monitoring, and picking up the targets. Wang et al. [15] designed an unmanned aerial vehicle (UAV) dynamic tracking and positioning system based on object detection and tracking, realizing the guided control of the UAV using monocular vision. Feng et al. [19] combined voice stimulation to guide the camera towards the speaker and perform distance adjustment and real-time tracking. Xie et al. [13] determined the flight data to guide the camera mounted on the aircraft to adjust its direction to avoid backlighting and ensure the quality of the captured image. Yamanaka [20] developed an intelligent robotic camera system that automatically tracks and reproduces historical compositions. In the field of adjusting shooting angles to assist in recognition and detection, facial pose estimation [21,22] is commonly used to prompt users to align their faces properly before capturing images for face matching, thereby improving the accuracy of face detection [23]. However, there is limited research on assisting object recognition and detection, particularly in the agricultural domain.

This study aims to design a set of guided shooting methods and mobile applications for agricultural crops, based on lightweight CNN models. The goal is to achieve the real-time recognition and tracking of crop targets on mobile devices such as smartphones. By controlling the operator, especially for agricultural personnel who use mobile phones for image data collection and crop detection, or control algorithm of various forms of guiding devices, the user can move to positions that are suitable for capturing high-quality target image data. This approach addresses issues such as low accuracy in intelligent algorithms, difficulties in recognition and detection, and incomplete analysis caused by low image quality. We will take the classification of assisted crop pot picking as an example, take

guided photos of round leaf pepper and grass pot picking, and finally evaluate the guidance method and auxiliary effects.

## 2. Approach and Dataset Construction

### 2.1. Guidance Methods and Classification Reduction

To enable the operator to reach a moderate distance and align the object properly within the limited prompts and steps, the orientation of the object in the photo can be constrained to 11 types, as shown in Table 1, each corresponding to different guiding cues. Here, the image frame refers to the rectangular frame formed by the camera viewport during shooting, while the object frame refers to the rectangular frame in the image that is tangential to the edges of the target object and parallel to the image frame. The nine-grid region of the frame refers to the nine regions formed by dividing the frame into three equal parts horizontally and vertically.

**Table 1.** Orientation types, constraints, and guiding prompts for objects in photographs.

Direction	Meaning	Guidance
Center	The object is in the center of the image frame, with some distance between the two frames.	None
Oversized	The object box encompasses the image frame, with a length ratio $> 2$ .	Move away from the object
Undersized	The image frame encompasses the object box, with a length ratio $> 2$ .	Move closer to the object
Up	The center point of the object box is located in the upper area of the image frame, with the two boxes intersecting.	Rotate or move upwards
Down	The center point of the object box is located in the lower area of the image frame, with the two boxes intersecting.	Rotate or move downwards
Right	The center point of the object box is located in the right area of the image frame, with the two boxes intersecting.	Rotate or move towards the right
Upper Right	The center point of the object box is located in the upper-right area of the image frame, with the two boxes intersecting.	Rotate or move towards the upper-right
Lower Right	The center point of the object box is located in the lower-right area of the image frame, with the two boxes intersecting.	Rotate or move towards the lower-right
Left	The center point of the object box is located in the left area of the image frame, with the two boxes intersecting.	Rotate or move towards the left
Upper Left	The center point of the object box is located in the upper-left area of the image frame, with the two boxes intersecting.	Rotate or move towards the upper-left
Lower Left	The center point of the object box is located in the lower-left area of the image frame, with the two boxes intersecting.	Rotate or move towards the lower-left

To identify the orientation of objects in images, this study will employ an image classification approach for photo guidance. There are two reasons for this choice: firstly, image classification algorithms have fast computation speeds and small model sizes, making them suitable for real-time photo needs while providing accurate orientation indications. Secondly, image classification techniques have advantages in multi-object scenarios, where they prioritize larger objects in the image during recognition, ignoring distant or smaller objects. The following section will introduce the collection and processing methods for the relevant classification dataset. If there are multiple targets in the image at the same time, according to the above classification method, they will be classified as "Undersized". In the guidance process (as in Section 4.3 later), the operator will be prompted to approach until the situation of having an unclear subject is avoided, preventing the model from having difficulty in distinguishing which is the primary target that needs guidance.

## 2.2. Data Collection and Preprocessing

The dataset was collected from the modern glass greenhouse of the East Campus of China Agricultural University. The collection targets are potted groups of Peppergrass with round leaves, and the collection took place in the afternoon of 22 March 2022. A regular mobile phone was used as the collection device, and the image size was  $1080 \times 1080$ . We manually labeled different images using the features described in Table 1. The data collection process ensured sufficient lighting in the environment, proper growth posture for potted plants, and the approximate alignment of plants at the same growth stage. The original dataset comprised a total of 1580 potted plant photos, which were augmented to 7900 images based on the 11 aforementioned orientation categories. Table 2 and Figure 1 present specific information for each category in the dataset.

**Table 2.** Categorizing of dataset.

Category Number	Orientation Category	Number of Images
0	Center	800
1	Oversized	725
2	Undersized	505
3	Up	695
4	Down	925
5	Right	505
6	Upper Right	765
7	Lower Right	820
8	Left	445
9	Upper Left	935
10	Lower Left	780
Total		7900

To highlight the crop features in the image and enhance the generalization capability of the data augmentation model, as well as to match the input–output requirements of the model, the following preprocessing steps were performed on the dataset:

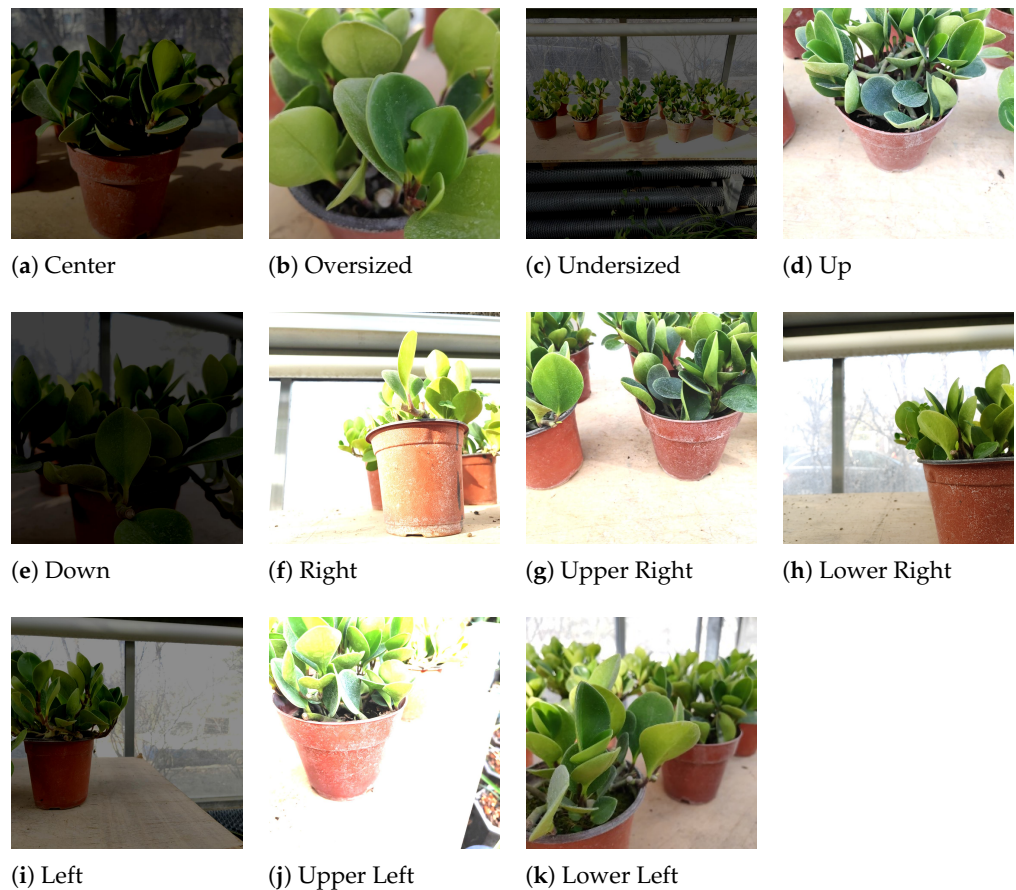
- Step 1. Resizing the image resolution to  $224 \times 224$  pixels. The advantage of applying this step is that it can reduce hardware load and improve network training speed without significantly reducing accuracy [24].
- Step 2. Using Equation (1), individual pixels' RGB channels are separately processed to randomly alter the brightness, contrast, and saturation of an image, thereby augmenting the dataset. This process enables the trained model to adapt to varying lighting conditions and color biases. The 'rand' function generates floating-point random numbers within a specific range.

$$pixel = \min(255, \max(pixel^{rand(0.9,1.1)} \times rand(0.5,2) + rand(-40,40), 0)) \quad (1)$$

Step 3. Performing normalization on the images, following Equation (2), to ensure that the pixel values of the images are within the range of  $-1$  to  $1$ . Here, for an individual pixel, the input grayscale value is represented as  $gray_{input}$ , and the output grayscale value is represented as  $gray_{output}$ .

$$gray_{output} = gray_{input} \div 127.5 - 1 \quad (2)$$

After preprocessing the dataset with the aforementioned steps, it is necessary to split the dataset into training and testing sets. This is carried out in order to create mutually exclusive subsets of data to be used during the model training and evaluation processes. By doing so, a more accurate assessment of the model can be conducted, and the performance of the model on unseen data can be validated. In this study, 30% of the crop photo dataset will be allocated as the testing set, while the remaining 70% will serve as the training set. We segment the original images and their augmented counterparts as a single entity to prevent highly similar images from appearing simultaneously in both the training and testing datasets, thereby avoiding artificially inflated training accuracy.



**Figure 1.** Dataset samples after [Step 2](#).

### 3. Orientation Discrimination Model

Due to the requirement of deploying the model on mobile devices, it is necessary to reduce the parameter size of the model while maintaining its accuracy. This process is known as model lightweight optimization. In this study, MobileNet V2 [25] was selected for the optimized design to ensure that the model can be frequently used on smartphones for orientation discrimination without compromising accuracy.

### 3.1. Model Selection

During the process of guiding photography, the mobile phone needs to call this model in real-time. In order to achieve better guidance effects on mobile devices, the model's parameter size and speed need to be appropriately limited. The threshold for human visual persistence is generally between 15 to 20 Hz, while the inverse of the screen refresh rate and Flops exhibit an linear relationship, and the explanation is as follows: if the model's inference rate is  $H$  Hz, then the time for one inference of the model is  $x = \frac{1}{H}$  s. If the floating-point operations of the model are represented by  $y$  Flops, and the mobile device performs floating-point operations at a frequency of  $v$  Hz, the inference time would be  $\frac{y}{v}$  s. Assuming there is a fixed computation time  $t$  for other operations in the testing environment, we have  $\frac{y}{v} + t = x$ , which implies  $y = \frac{1}{v} \cdot x - \frac{t}{v}$ .

As shown in Figure 2, which illustrates the relationship between common model Flops and the screen refresh rate during mobile phone operation. After performing exponential regression fitting (orange diagonal line) and converting the frequency, the Flops of the model when running during human visual persistence should be between 75,181,987 and 248,285,473 (indicated by the vertical brown-black line range). The  $R^2$  value is greater than 0.95, indicating that the results are reliable. Therefore, the Flops of the model should be less than 248,285,473 in order to provide a better user guidance experience.

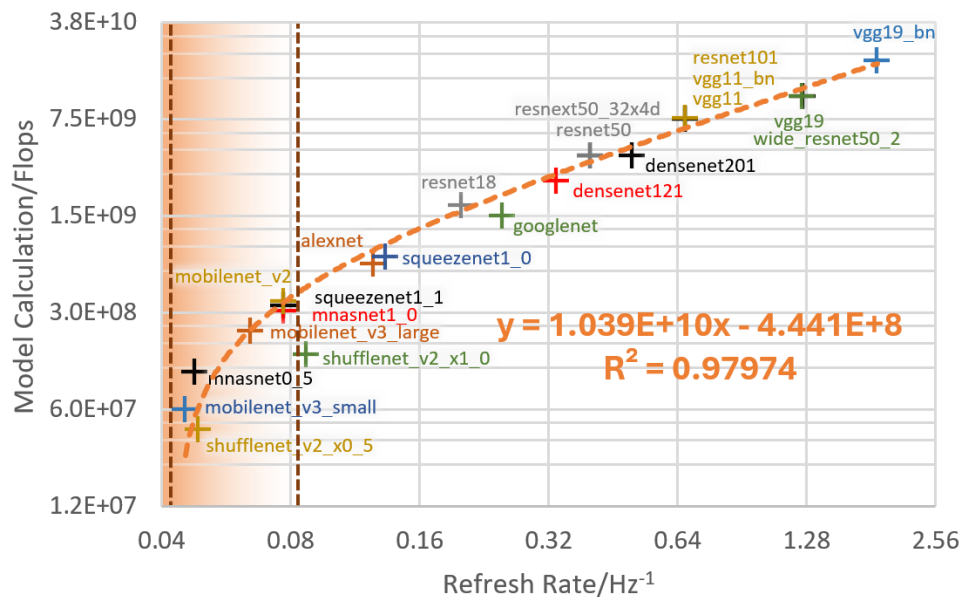


Figure 2. Relationship between model FLOPs and refresh rate during mobile phone operation.

This study aims to improve upon the MobileNet V2 model. Traditional classification models such as AlexNet, VGG, and ResNet are known to be large in size and have slower computational speeds, making them unsuitable for meeting the Flops requirements and the high-frequency invocation demands of mobile devices. To address the issue of model execution on embedded devices, researchers have proposed a series of lightweight models, including SqueezeNet, ShuffleNet, and MobileNet, most of which meet the Flops requirements. However, the direct use of these models often yields a weaker performance compared to larger models.

In this experiment, MobileNet V2 exhibits superior performance among the lightweight models, possibly due to its design strategies such as linear bottlenecks, inverted residuals, depthwise convolutions, and multi-scale feature fusion, resulting in higher accuracy. Specifically, it possesses the following distinctive structures:

- Linear Bottleneck: In each convolutional layer, a  $1 \times 1$  convolutional kernel is used for feature compression, reducing the number of input channels. Then, a non-linear acti-

vation function, ReLU6, is applied. This design allows for a reduction in the number of model parameters while maintaining good feature representation capabilities.

- **Inverted Residuals:** Traditional residual blocks perform feature expansion followed by feature compression. Inverted residuals, on the other hand, reverse this process. Inverted residuals begin with a  $1 \times 1$  convolutional kernel for feature compression, followed by a  $3 \times 3$  depthwise separable convolution for feature expansion. Finally, another  $1 \times 1$  convolutional kernel is used for feature compression. This design helps improve the model's non-linear expressive power while reducing computational complexity.

### 3.2. Model Architecture and Optimization

Common methods for reducing model complexity and improving the performance of lightweight models include model pruning, knowledge distillation, sparse constraints, parameter quantization, and binary weights. Three methods will be employed in this study—increasing sample randomness, model pruning, and knowledge distillation—to optimize the model, resulting in a model with very high accuracy and extremely low computational overhead.

Specifically, as shown in Figure 3, we increased the accuracy of all models through the method of increasing sample randomness (in green), and selected the high-performance large model VGG Net and the lightweight model MobileNet V2. Subsequently, we performed sparse training on the latter to produce a more efficient and easily prunable MobileNet V2-1. Following this, pruning (in yellow) led to the reduced computational load of MobileNet V2-2, and finally, knowledge distillation (in blue) using AlexNet as the teacher resulted in the computationally efficient and highly accurate MobileNet V2-3.

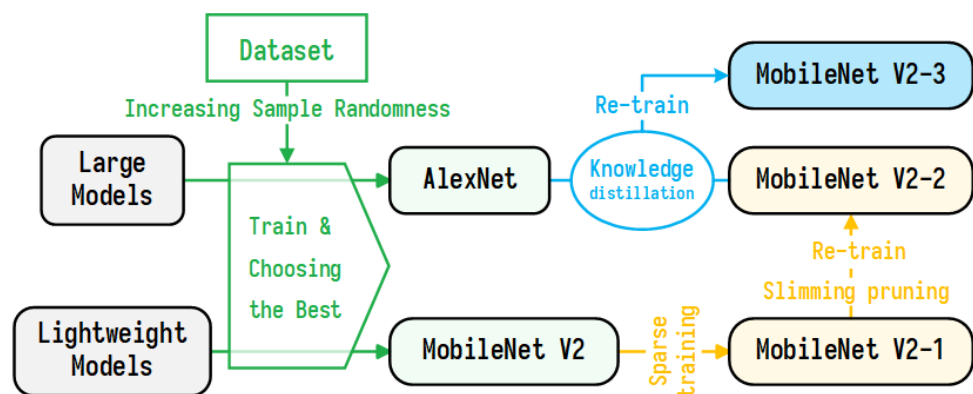


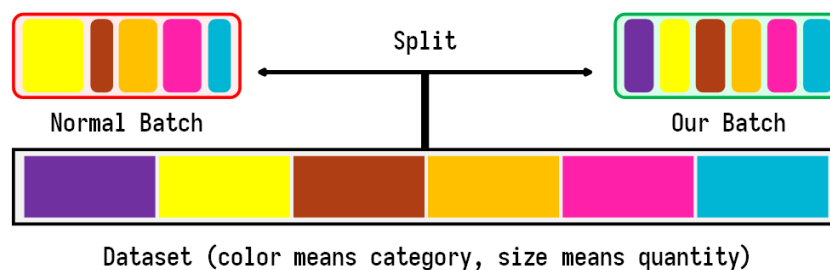
Figure 3. MobileNet V2 structure and training optimization overall process.

#### 3.2.1. Increasing Sample Randomness

To enhance the optimization of the model, the training process will incorporate increased sample randomness. This approach involves handling mini-batch data with augmented sample randomness. Due to the enlarged size of the augmented dataset, direct batch gradient descent becomes impractical. When training with mini-batch gradient descent without shuffling, the model's training process may struggle to converge towards the ideal minimum loss point, as a single batch of samples may not adequately represent the average characteristics of the entire dataset. Consequently, the model's performance may not be fully realized. Including as many different categories as possible in a single batch of samples, with approximately equal numbers of instances for each category, can, to some extent, suppress the occurrence of this situation.

The increase in sample randomness will be achieved through the following methods: at the beginning of each training epoch, the sample order within the training set will be randomly shuffled and stratified sampled (as shown in Figure 4) to enhance the randomness of the samples and the representativeness of the batches. In the implementation process, this segmentation process will be integrated into the data preprocessing stage to avoid

redundant computations during training. It will involve loading preprocessed variable files directly when creating data loaders, facilitating the iteration of batch data in the training and testing sets, providing data support for model training and evaluation, simplifying the data processing process, and improving training efficiency.



**Figure 4.** Illustration of the difference between stratified sampling mini-batches and regular mini-batches.

### 3.2.2. Model Pruning

Model pruning [26] is a commonly used model optimization technique aimed at reducing the size and complexity of deep learning models to improve their storage efficiency, computational efficiency, and generalization ability. The basic idea of model pruning is to reduce model complexity by removing redundant connections, reducing the number of parameters, or decreasing the number of layers while maintaining the model's performance on training and test data.

When pruning network weights, it is necessary to iterate through each parameter of the model and calculate the importance of each parameter based on the selected criteria. Parameter importance metrics are primarily used to evaluate the significance of each parameter. Common metrics include gradient magnitude, the sensitivity of parameters to outputs, and parameter information entropy. These metrics can help determine the extent to which parameters affect the model's output. For gradient-related metrics, the gradients of parameters can be computed through backpropagation and analyzed for their magnitudes. For other metrics, specific evaluation methods may be required for calculation and analysis.

Finally, based on the evaluation results, sort the parameters to determine their importance order, and then select whether to keep or prune the parameters based on their level of importance.

Considering that the Batch Normalization layers account for a significant proportion in the MobileNet V2 network, the Slimming pruning method is primarily employed for channel pruning. This method utilizes the scale parameters of the Batch Normalization layers to assess importance and subsequently prunes the non-important channels. As shown in Figure 5, the implementation of this method follows the steps outlined below:

- Conduct training on the training set, achieving sparsity by applying L1 regularization gradients to the Batch Normalization layer.
- Compute the absolute values of the scale parameters for all BN layers, calculate the average importance per channel, and use it as a metric for the importance of the channel.
- According to a predetermined proportion, prune the weights associated with channels of lower importance to obtain the pruned network.
- Due to changes in the network structure, the new network may be in an underfitting state, so perform secondary training to improve accuracy.

In the implementation process, it is necessary to create a copy of the model first to ensure that the performance of the original model is not affected when evaluating the importance of the parameters. The parameters of the copied model will be modified and analyzed during the evaluation process. The copied model is set to evaluation mode to ensure that no training is performed during the evaluation process, and only parameter analysis takes place.



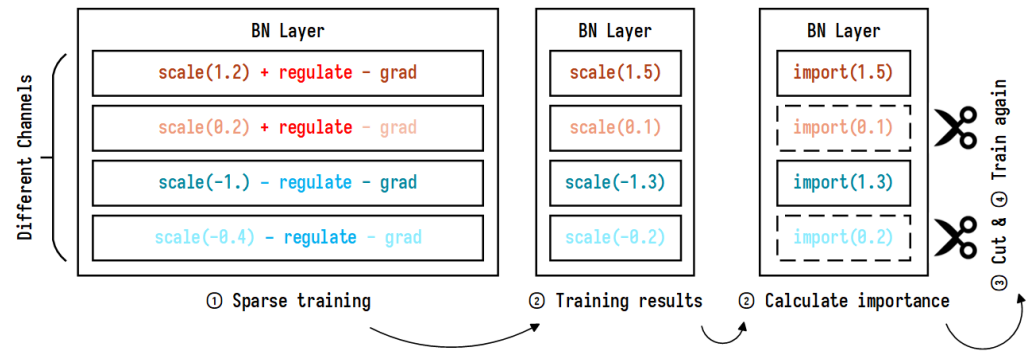


Figure 5. Pruning steps based on BN layer importance assessment.

### 3.2.3. Knowledge Distillation

Knowledge distillation [27] is a model compression technique that trains a student model by leveraging the knowledge from a teacher model, aiming to achieve a performance close to the teacher model while having a smaller and more lightweight model. The basic principle is to use the soft targets generated by the teacher model to guide the training of the student model. Soft targets refer to the class probability distribution outputted by the teacher model, which contains richer information compared to the one-hot encoded hard targets. This richer information helps the student model better understand the data. Through knowledge distillation, the student model can acquire valuable knowledge from the teacher model, transferring the complexity and performance advantages of the larger model to the smaller model, thus achieving model compression and optimization. This technique is particularly valuable in resource-constrained environments such as mobile and embedded devices, where it can simultaneously meet the requirements of model size and computational efficiency.

Due to the small size of the dataset and the relatively low difficulty of model training, we employ the method of knowledge distillation on the pruned MobileNet V2 model through offline distillation. The process is outlined as follows:

1. Training the large pre-trained model with the training set and retaining the best-performing model as the teacher model.
2. Fixing the teacher model and conducting one training session for the student model (same as step 3), while performing a grid search for hyperparameters. The loss calculation is shown in Equation (3), where the parameters are defined as follows:
  - (a) “real” represents the actual one-hot label.
  - (b) “pred” represents the predicted one-hot label.
  - (c) “CE” denotes the cross-entropy loss function.
  - (d) “KL” denotes the Kullback–Leibler divergence loss function.
  - (e) The hyperparameter  $\alpha$  serves as a weight to adjust the emphasis of the student model’s learning toward the teacher model and the real labels.
  - (f) The hyperparameter “temperature” can soften the probability distribution of the model output labels. A larger temperature value leads to a more softened distribution, while a smaller temperature value may amplify the probability of misclassification and introduce unnecessary noise.

$$Loss = (1 - \alpha) \times CE(real, pred_{student}) + \alpha \times KL\left(\log\_softmax\left(\frac{pred_{student}}{temperature}\right), softmax\left(\frac{pred_{teacher}}{temperature}\right)\right) \quad (3)$$

3. Utilizing the optimal hyperparameters obtained from grid search for offline distillation (as shown in Figure 6).
  - (a) Making predictions using the teacher model to obtain the soft targets.
  - (b) Making predictions using the student model to obtain the outputs to be optimized.

- (c) Computing the loss using the soft targets, hard targets (actual labels), and the outputs to be optimized.
  - (d) Performing backpropagation of the loss and updating the student model.
  - (e) Returning to step “a” until the model converges and the training is completed.
4. Conducting a second round of training for the student model directly using the training set to enhance the model’s learning of the original labels.

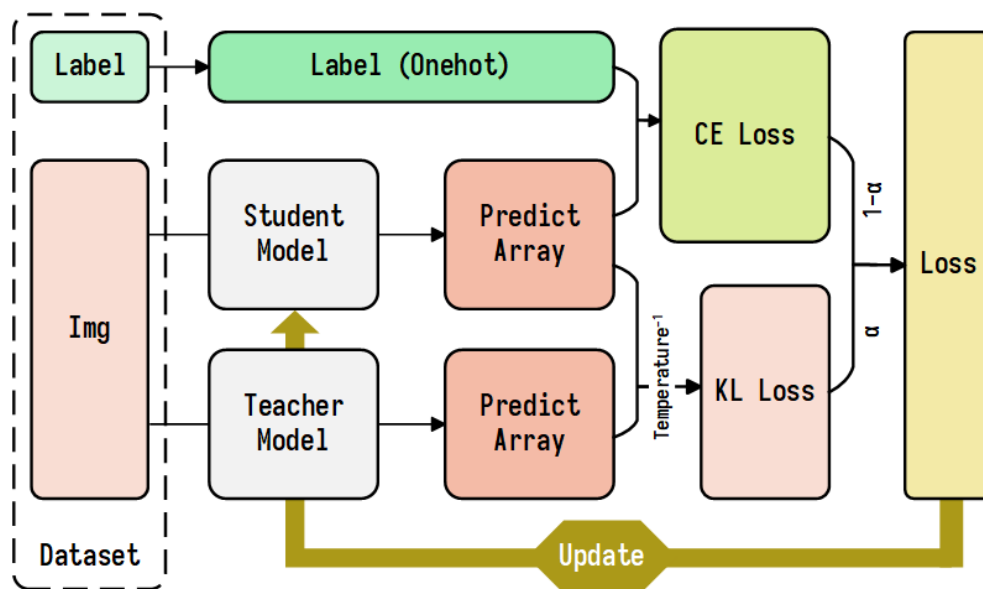


Figure 6. Offline distillation loss calculation and model update.

## 4. Experimental Analysis and Testing

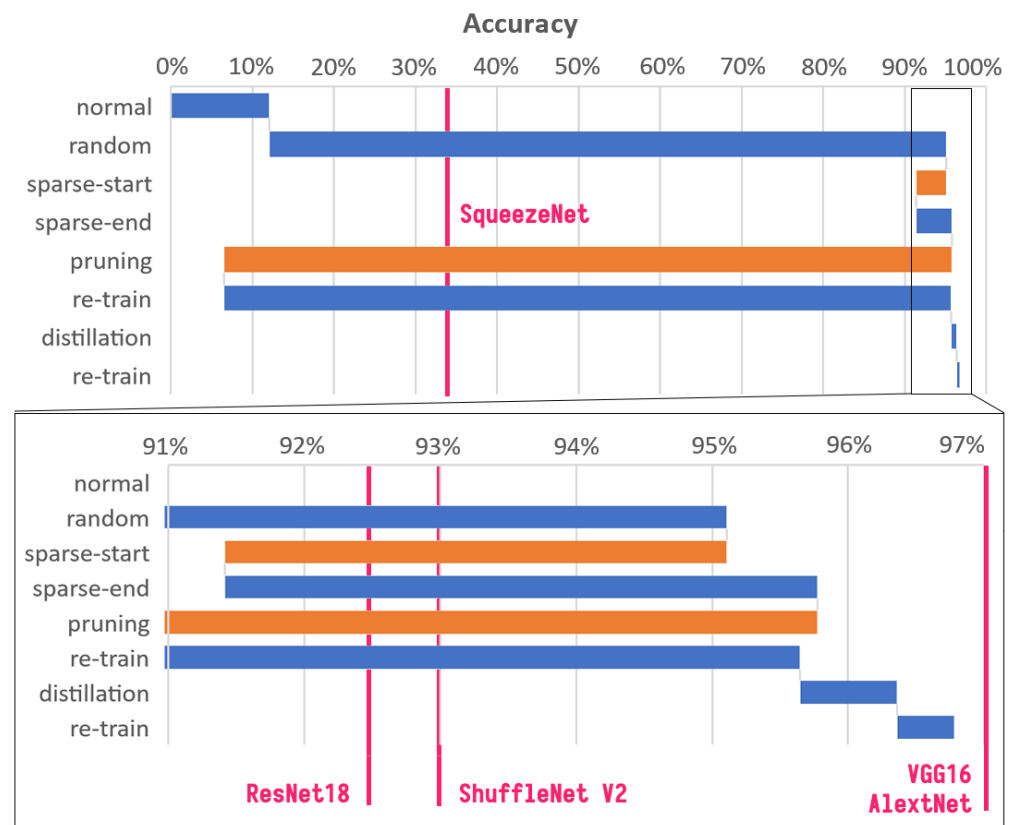
### 4.1. Overview

In this study, the MobileNet V2 [25] model was used as the base model for actual training. The model was optimized through three methods: increasing sample randomness, model pruning, and knowledge distillation. To evaluate the effectiveness of the model improvement strategies, corresponding ablation experiments were designed and compared with traditional models. In this experiment, traditional large models, namely AlexNet [28], VGG16 [29], and ResNet18 [30], were selected as references, along with traditional small models, SqueezeNet 1.0 [31] and ShuffleNet V2 x1.0 [32]. The MobileNet V2 model was used to construct the MobileNet V2-1 model, which was trained by increasing sample randomness. Then, the MobileNet V2-2 model was constructed through model pruning, and finally, the MobileNet V2-3 model was trained using knowledge distillation. The traditional models are widely used in image processing in agriculture, with large models mainly used for fruit and vegetable classification [33], pesticide residue detection [34], and disease and pest detection [35,36] on the server side, while lightweight models are mainly used in mobile devices [37,38] and drones [39].

Both the traditional models and their pre-trained weights are from torchvision.models. The training and improvement results of each model are shown in Table 3. The middle section of the table shows the Accuracy, Precision, Recall, and F1 score of each model in the test set under the original conditions, after adding sample randomness, sparse training, model pruning, and knowledge distillation. The greener the color, the higher the accuracy, and red indicates values below 90%, with darker shades of red indicating lower values. On the right side of the table are the computational loads of the models, with values less than 248,285,473 marked in bold. The improvement effects of each step on the accuracy of the MobileNet model are shown in Figure 7, where the bottom part is an enlarged version of the top part. Orange represents negative improvement, blue represents positive improvement, and the red line represents the accuracy of other models optimized through increased sample randomness during training.

**Table 3.** Accuracy, Precision, Recall, F1 score, and FLOPs for different models.

Stage	Model Name	Accuracy	Precision	Recall	F1 score	FLOPs/M
Normal	AlexNet	94.98%	94.97%	94.59%	94.74%	710.15
	VGG16	95.82%	95.22%	94.75%	94.92%	1044.45
	ResNet18	9.83%	0.59%	9.09%	1.11%	1826.01
	SqueezeNet 1.0	21.13%	22.09%	20.59%	19.82%	153.65
	ShuffleNet V2	12.89%	1.03%	9.09%	1.85%	733.35
	MobileNet V2	12.09%	17.10%	9.39%	2.45%	332.96
Random	AlexNet	97.07%	96.95%	96.67%	96.78%	710.15
	VGG16	96.95%	95.95%	95.62%	95.63%	1044.45
	ResNet18	92.97%	91.87%	91.33%	91.48%	1826.01
	SqueezeNet 1.0	33.72%	29.34%	29.99%	29.15%	153.65
	ShuffleNet V2	92.43%	91.68%	91.23%	91.28%	733.35
Sparse Pruning Re-train Distillation Re-train	MobileNet V2	95.10%	95.12%	94.53%	94.77%	332.96
		95.77%	95.58%	95.46%	95.38%	332.96
		6.49%	0.59%	9.09%	1.11%	160.35
		95.65%	95.46%	95.56%	95.46%	160.35
		96.36%	96.32%	95.99%	96.11%	160.35
96.78%	96.44%	96.39%	96.38%	160.35		



**Figure 7.** MobileNet model accuracy improvement waterfall diagram.

Based on Table 3, it is evident that the MobileNet V2 model, optimized through three steps, demonstrates outstanding performance in terms of accuracy and computational efficiency, while occupying relatively little storage space. This makes it more suitable for deployment and usage on resource-constrained mobile devices. Specifically, after optimizing the training by increasing sample randomness, the accuracy of the small model has significantly improved, indicating that this method can significantly reduce the training difficulty of small models and improve accuracy. Among the large models, AlexNet performs the best, while, among the lightweight models, MobileNet V2 excels, hence the

selection of the latter for transformation optimization. After sparse training, the accuracy of the MobileNetV2 model saw a slight improvement, but the computational load did not yet meet the scene’s requirements. Following pruning, the model parameters reduced sharply by 30%; however, the accuracy decreased drastically. After retraining, the accuracy saw a slight improvement compared to the original model. Utilizing AlexNet as the teacher model for knowledge distillation training and retraining resulted in a significant improvement in model accuracy, achieving the high accuracy of large models and the low computational load of the smallest lightweight model.

#### 4.2. Concrete Analysis

##### 4.2.1. Traditional Model Training Results

The accuracy and loss variations of the selected six traditional models on the validation set during training with our dataset are shown in Figure 8. During training, the batch size was set to 64, utilizing the SGD optimizer with a learning rate of 0.001 and momentum of 0.9. In order to ensure the convergence of all models, a total of 65 epochs were trained. The larger models exhibited notably higher accuracy compared to the lightweight models, with AlexNet and VGG Net demonstrating the best performance, as AlexNet achieved lower loss. For the other four model categories, the loss reached its lowest value within the first 10 epochs, and the accuracy did not improve further, indicating that the models had already reached their performance limits without optimization. The main reason for this situation is the high similarity between samples of different categories, with only spatial differences. If there is a slight deviation in the direction of gradient descent, it is highly likely to miss the low valley region of the loss function, resulting in difficulty in reducing the loss. Therefore, increasing the randomness of the samples is necessary, aiming to include a similar proportion of all categories in each batch, allowing the optimizer to more accurately provide the direction for reducing the loss.

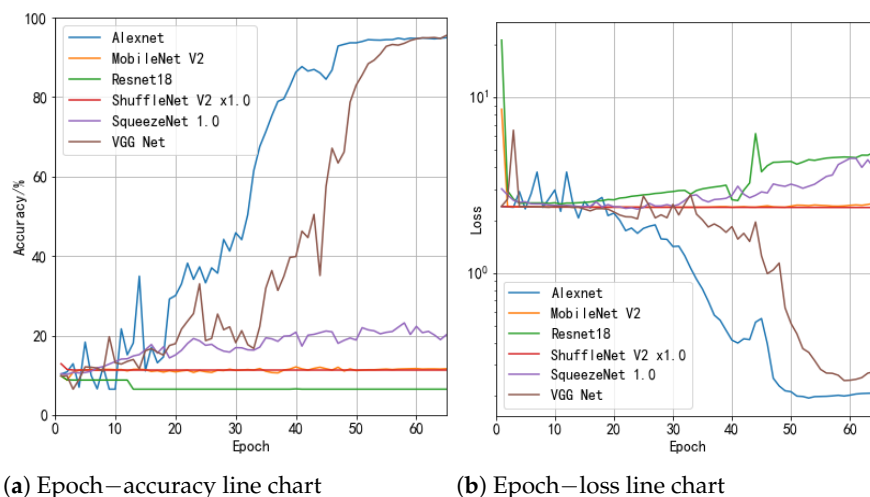
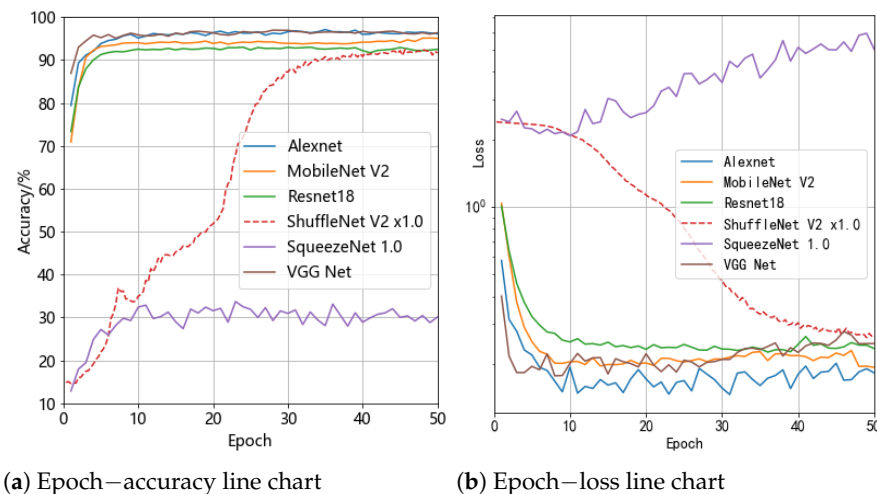


Figure 8. Accuracy and loss variation chart of traditional models in the test set.

##### 4.2.2. Increased Randomness

We increased the randomness of each batch of samples by using random shuffling and layer-wise sampling, and conducted the same training process as the above-mentioned experiment. The performance of each model on the validation set is shown in Figure 9. In particular, the dashed line represents the curve of the ShuffleNet V2 x1.0 model, which took a total of 150 epochs to converge, whereas the other models converged in 50 epochs. To clearly compare the differences between the starting and ending points of different models, we divided the training epoch values of the former by three for plotting in the figure.



**Figure 9.** Accuracy and loss variation of traditional models after increasing randomness processing (the true  $x$  value of the dashed line is three times that shown in the figure).

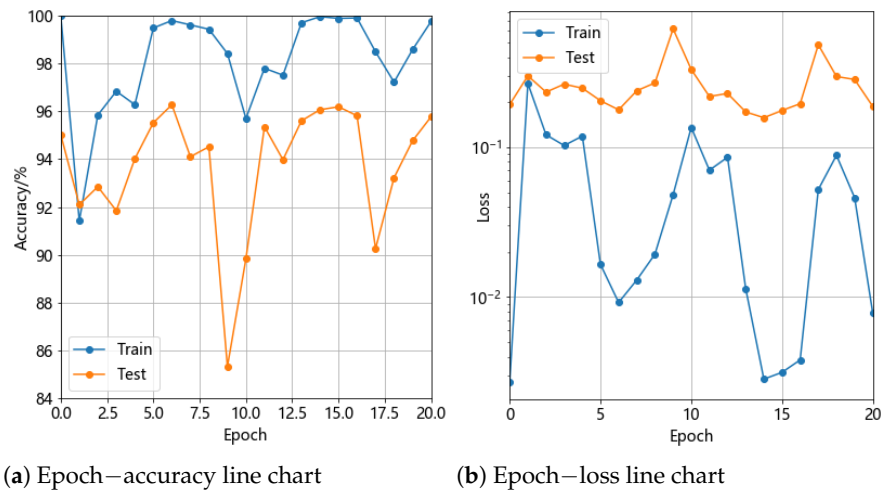
It is evident that all models converge normally, with AlexNet achieving the highest accuracy of 97%. Among the three lightweight models, MobileNetV2 converges rapidly and achieves high accuracy, while ShuffleNet converges slowly with moderate accuracy, and SqueezeNet has extremely low accuracy (34%), but still higher than the random classification level ( $1/11 \approx 9\%$ ). This may be due to the high proportion of  $1 \times 1$  convolutional layers in the SqueezeNet model (30–40%), which limits its ability to capture and represent complex spatial features. In contrast, in the lightweight models, MobileNet and ShuffleNet, the proportion of  $1 \times 1$  convolutional layers is relatively low (10–20%), resulting in less loss of spatial information. Additionally, they adopt more complex structures and feature extraction methods, such as depthwise separable convolution and channel shuffle, which can more effectively learn and represent global spatial features, thereby demonstrating stronger performance in object position classification tasks.

#### 4.2.3. Model Pruning

This study implements sparse training, pruning, and retraining operations through the custom pruner class.

First, we modified the MetaPruner class in the torch\_pruning library to achieve sparse training gradients for all BN layers layer by layer, with a regularization coefficient set to  $1 \times 10^{-5}$  and a learning rate of 0.001. The Adam optimizer was utilized for training, and convergence was reached after 20 epochs. The model's accuracy on the validation set slightly increased from 95.1% in the original model to 95.77%, while also achieving sparsity in the BN layers. We refer to the MobileNet model processed in this manner as MobileNet V2-1. Figure 10 shows the accuracy and loss variation details of sparse training.

By utilizing the Importance class in the torch\_pruning library, the significance of each convolutional layer's output features, i.e., the absolute values of the scale parameters in the BN layer, was evaluated. The pruning operation was executed using pruner.step(), removing 30% of the channels at a time, with the exception of ignoring the final classification layer. Simultaneously, the torch\_pruning.utils.count\_ops\_and\_params function was employed to calculate the MACs (multiply-accumulate operations) of the pruned model. Upon completion of the pruning process, the feature dimension was reduced from 32 to 22.



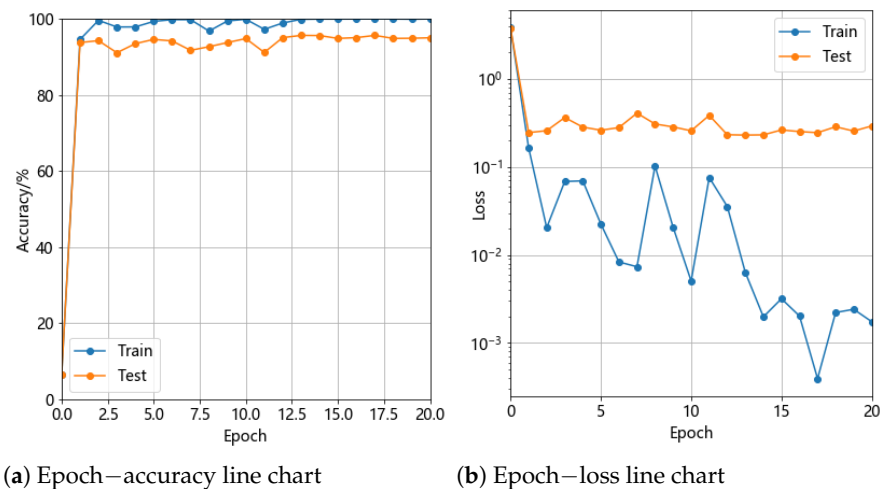
**Figure 10.** Accuracy and loss variation graph of MobileNet in sparse training.

As shown in Table 4, compared to MobileNet V2-1, the pruned model achieved a reduction in computational complexity below the target FLOPs and a decrease in the number of parameters from 2.239244 M to 1.117966 M. This reduction is advantageous for deploying the model on mobile devices.

**Table 4.** Changes in MobileNet V2 model before and after pruning.

State	FLOPs/M	Param/M	Accuracy/%
Before Pruning	332.9616	2.238	95.77
After Pruning	160.3491	1.117	6.49

Due to pruning, the model structure changed, resulting in a loss of certain expressive capabilities, thus making it unable to adapt well to training and testing data, leading to a significant decrease in model accuracy. Therefore, retraining was necessary, with parameters set the same as during sparse training. We refer to this retrained MobileNet model as MobileNet V2-2. As shown in Figure 11, after the second training, the optimal model accuracy was 95.65%, which was slightly lower compared to MobileNet V2-1. To further improve the model’s performance, knowledge distillation training is required.



**Figure 11.** Accuracy and loss variation graph of MobileNet during retraining.

#### 4.2.4. Knowledge Distillation

Based on the results of training with increased sample randomness, the model AlexNet with the highest accuracy was chosen as the teacher model to conduct offline knowledge distillation training on MobileNet V2-2. In order to select the involved hyperparameters alpha and temperature, we fixed the batch size, set the learning rate to 0.001, momentum to 0.9, weight decay to  $5 \times 10^{-4}$ , used the SGD optimizer, and performed grid search for one epoch on the same model. Ultimately, regarding the optimal hyperparameters, alpha was determined to be 0.6 and temperature to be 2, resulting in an accuracy of 96.11% after one training session. The grid search results are shown in Figure 12.

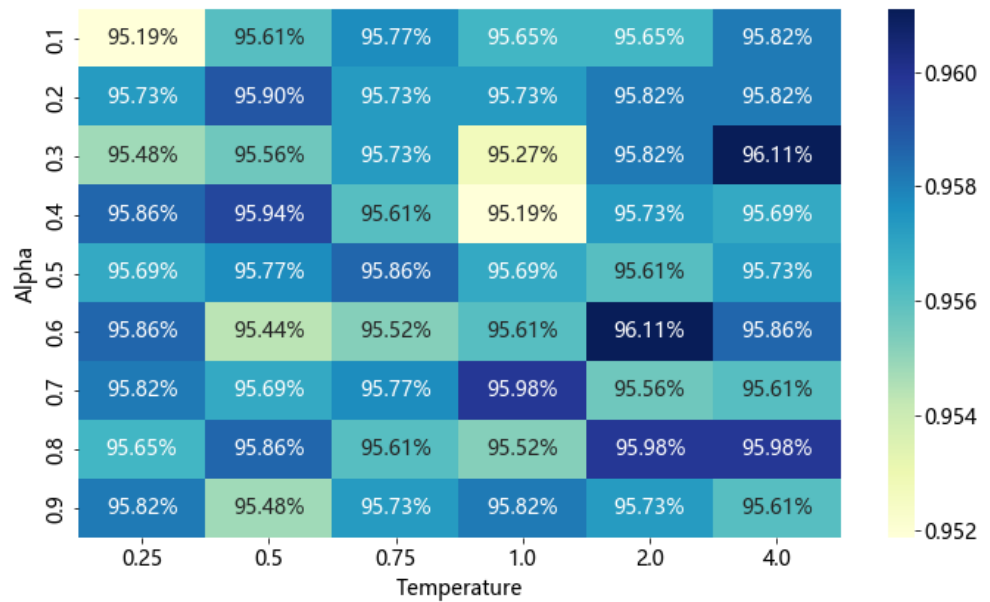
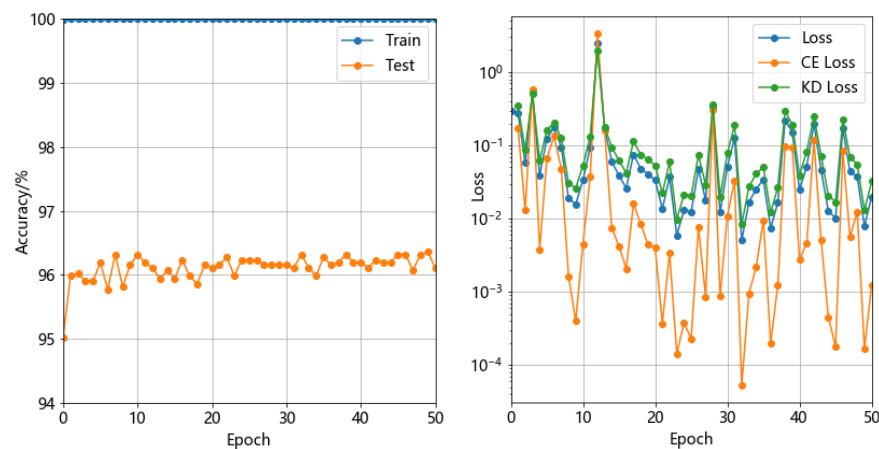


Figure 12. Knowledge distillation hyperparameter grid search accuracy heatmap.

Using the same optimizer, the model underwent 50 epochs of offline distillation training, and the corresponding accuracy and loss changes are shown in Figure 13. It can be observed that the model’s accuracy experienced a small leap, increasing from 95.65% to 96.36%. This represents that the model has learned more accurate data features through distillation training.

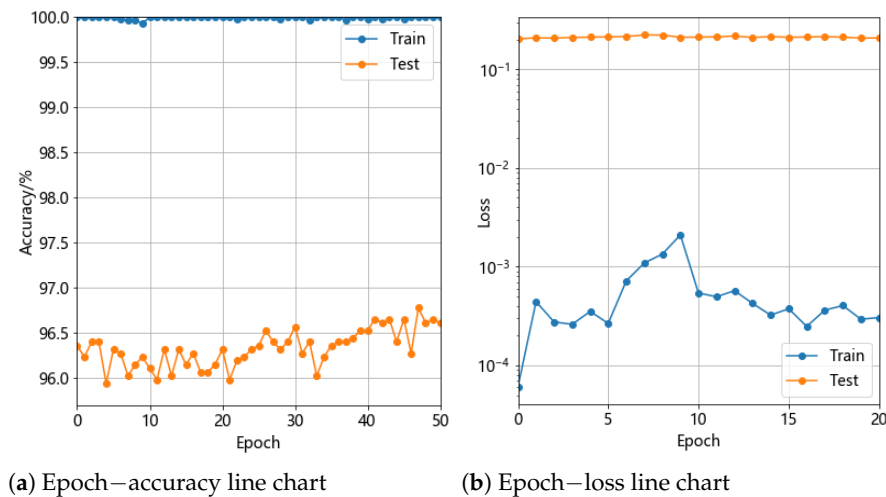


(a) Epoch–accuracy line chart

(b) Epoch–loss line chart

Figure 13. Accuracy and loss variation chart of MobileNet in offline distillation.

Finally, the model was fine-tuned using the original dataset directly, employing cross-entropy loss for the final adjustment of the model. The optimizer and other parameters remained consistent with the aforementioned distillation training, resulting in the ultimate model, MobileNet V2-3. It is evident from Figure 14 that the model's accuracy steadily increased from 96.36% to 96.78%, indicating that fine-tuning the model using the original dataset after distillation had a positive effect.



**Figure 14.** Accuracy and loss variation of MobileNet during distillation and retraining.

#### 4.3. Real Machine Testing

To further validate the effectiveness of the guidance method, tests were conducted with actual devices in similar times and scenes as during the data collection process. The mobile devices used are shown in Table 5, covering different levels of devices, the most popular chipsets, various versions of the Android operating system, and different camera configurations. The efficiency and performance of the algorithm were extensively tested. During testing, we found that the model was unable to be guided under low-light conditions. To address this issue, we employed the data augmentation method as described earlier and turned on the flashlight of the mobile phone.

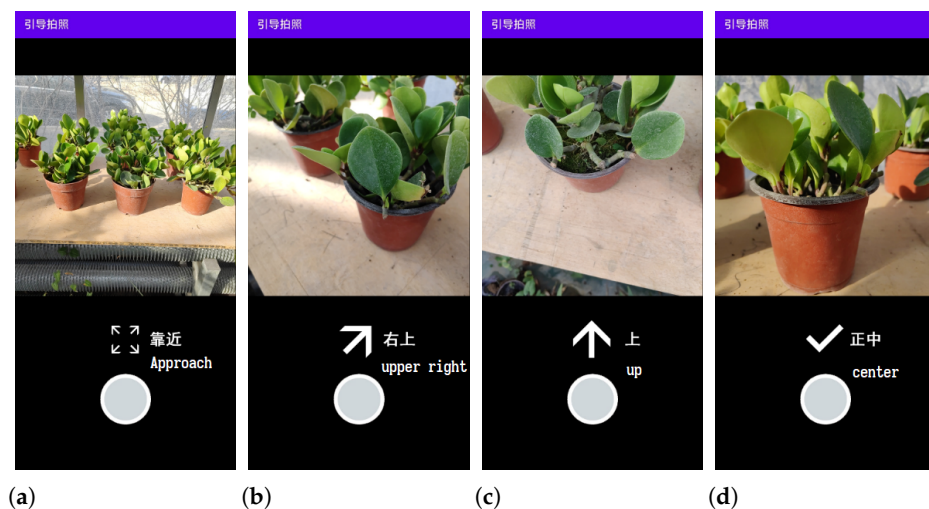
**Table 5.** Devices used in the experiment and their main parameters.

Device	Android	Processor	RAM	Camera	Announced
Xiaomi Redmi Note 9 4 G	12	Qualcomm SM6115 Snapdragon 662 (11 nm/2.0 GHz)	6 GB	48 MP, f/1.8, 26 mm (wide), 1/2.0", 0.8 $\mu\text{m}$ , PDAF	26 November 2020
Xiaomi Redmi Note 12 T Pro	13	Mediatek Dimensity 8200 Ultra (4 nm/3.1 GHz)	8 GB	64 MP, f/1.8, 23 mm (wide), 1/2", 0.7 $\mu\text{m}$ , PDAF	29 May 2023
Huawei Enjoy 20 SE	10	Kirin 710 A (14 nm/2.0 GHz)	4 GB	13 MP, f/1.8, 26 mm (wide), PDAF	23 December 2020
Vivo iQOO Z5	13	Qualcomm SM7325 Snapdragon 778 G 5 G (6 nm/2.4 GHz)	8 GB	64 MP, f/1.8, 26 mm (wide), 1/1.97", 0.7 $\mu\text{m}$ , PDAF	23 September 2021
Vivo Pad	13	Qualcomm SM8250-AC Snapdragon 870 5 G (7 nm/ 3.2 GHz)	8 GB	13 MP, f/2.2, 112° (ultrawide), 1.12 $\mu\text{m}$ , AF	11 April 2022

The operational effect on the Redmi Note 9 4 G device is shown in Figure 15, with the interface using a combination of simple text and icons to guide the photography process,

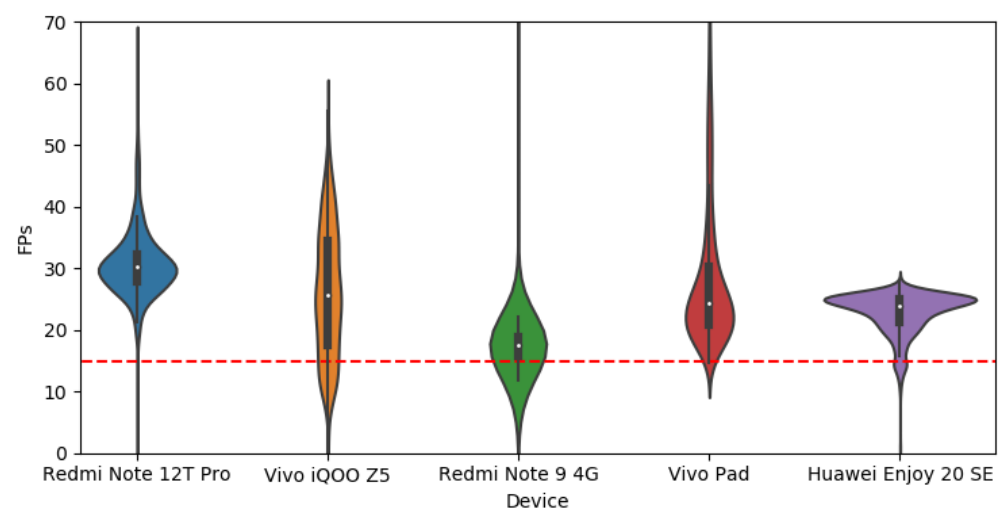


thus reducing the level of difficulty in understanding. During testing, agricultural workers were able to correctly comprehend the guidance prompts and adjust the device to the specified position as per the given instructions. The consistency observed between the guidance prompts and the images indicates a good guidance effect, enabling accurate directional guidance for users to adjust to the specified orientation. In Figure 15a, the target is too small, and the interface prompts the user to move the phone closer to the target. In Figure 15b, the target is located in the top right corner of the image, and the interface prompts the user to move the phone towards the upper right direction. In Figure 15c, the target is located at the top of the image, and the interface prompts the user to move the phone upwards. In Figure 15d, after a series of operations, the target moves to the center of the image, with an appropriate size. The interface prompts the user that the operation is correct, and the guided process is concluded.



**Figure 15.** Screenshot of real machine-guided testing and frame rate box diagram. (a) Target: small. (b) Target: upper right. (c) Target: top. (d) Target: center.

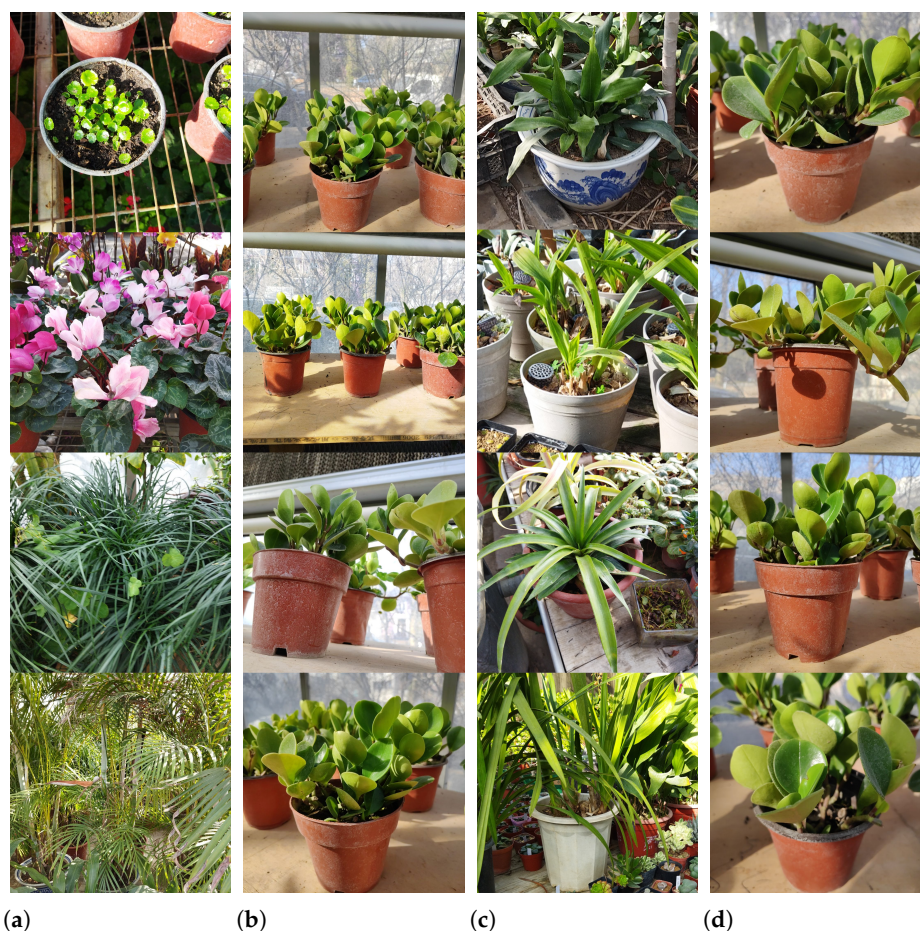
Figure 16 displays a violin plot of the frame rates when running on different devices, where the curved lines on both sides represent the density distribution, the white dots in the middle indicate the median, the black boxes represent the quartiles, and the red line at the bottom corresponds to 15 FPS. It can be observed that the main part of the model's operating frame rates (the black boxes) remains above 15 FPS, indicating very smooth model operation without lagging, which meets the user's experiential requirements.



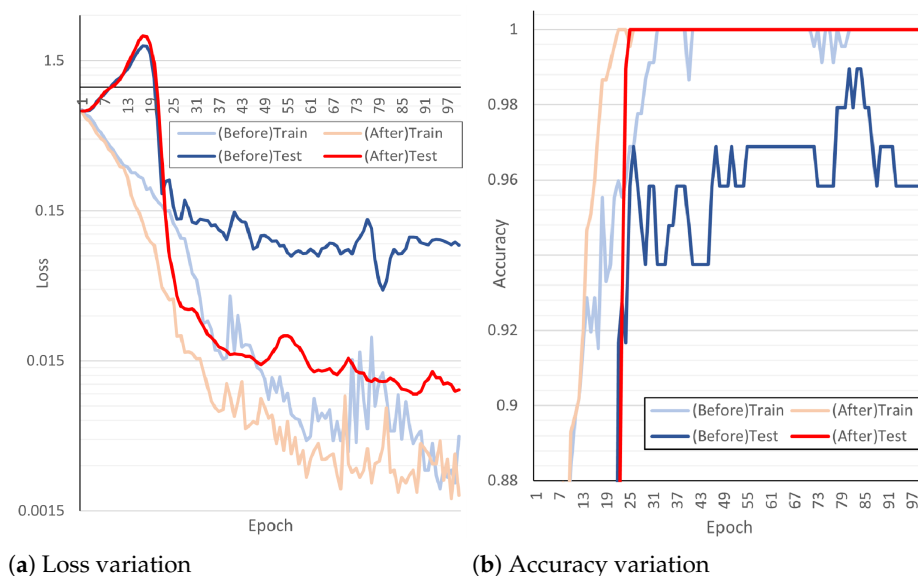
**Figure 16.** FPS violin plot.

#### 4.4. Effect Inspection

To better assess the auxiliary effect of guidance on other crop image detection algorithms, a dataset of various crop images was collected in both normal scenes (Before) and guided assistance scenes (After). The latter utilizes the application mentioned earlier. The MobileNet V2 model was utilized for crop classification training and testing. Figure 17 shows examples of the small-scale binary dataset with a total of 640 images. Figure 18 displays the changes in loss (Figure 18a) and accuracy (Figure 18b) of the model when differentiating between normal scenes and guided assistance scenes during the iterations. In Figure 18, blue represents the effects in general scene acquisition, while red represents the effects under guided photo assistance. As for loss, although the performance of both approaches ultimately converges in the training set, in the testing set, the loss under guided assistance is significantly lower by an order of magnitude compared to that in the general scene. Regarding accuracy, the dataset acquired through guidance enables the model to converge faster during training, saving nearly two-fifths of the iteration time compared to the general scene convergence speed. In particular, under the guided scene, when the training set reaches 100% accuracy, the testing set accuracy continues to improve, while in the general scene, the testing set exhibits early signs of overfitting. This indicates that, under guided photo assistance, similar samples will have higher consistency, and the number of samples that are difficult to distinguish between different classes is greatly reduced, providing a scenario with more distinct features for training and application and thereby enhancing the detection effectiveness of agricultural intelligent algorithms related to image data quality.



**Figure 17.** Crop classification dataset in normal scenes (Before) and guided assistance scenes (After). (a) (Before) negative. (b) (Before) positive. (c) (After) negative. (d) (After) positive.



**Figure 18.** Crop classification in normal (Before) and guided assistance (After) scenes.

## 5. Summary and Outlook

Due to the direct correlation between the effectiveness of agricultural image intelligence algorithms and the positive and significant presence of target objects, coupled with the limited digital proficiency of agricultural practitioners, the image quality captured using mobile smart devices is often subpar. Therefore, it is necessary to provide relevant guidance to instruct agricultural practitioners to adjust the camera pose during image capture. The application of guided photography techniques can facilitate the collection of well-defined and proper angled crop images. These provide a more distinct and accurate dataset for training agricultural image intelligence algorithms, including target detection, crop identification, pest/disease recognition, and phenotypic analysis. By leveraging guided photography, the dataset collected benefits the practical implementation of the follow-up algorithms by providing easily detectable image data, including effectively enhancing the upper limit of accuracy for these algorithms and being beneficial for their practical deployment in the agricultural sector. In this study, based on the MobileNet V2 model, we constructed a smartphone-based orientation detection model with high accuracy and low computational requirements by incorporating three steps: increasing sample randomness, model pruning, and knowledge distillation. Building upon the classification results, we implemented a guidance system for image capture. The experimental results demonstrate that this method provides accurate and smooth guidance, enabling farmers to capture high-quality photos of crops, and effectively improves the performance of intelligent algorithms for agricultural images, including crop classification.

The contributions of this study are as follows: first, it provides a feasible guidance solution for adjusting camera poses during image capture, which can effectively guide agricultural workers to capture high-quality agricultural images, thus providing a clearer and more accurate dataset for training agricultural image intelligent algorithms. In addition, this study has developed a smartphone-based orientation detection model with high accuracy and low computational requirements, making it possible to guide the shooting process on mobile phones. Finally, this study has implemented a guided photography system, enabling farmers to capture high-quality crop photos using their smartphones, which can effectively improve the performance of agricultural image intelligent algorithms, including crop classification.

There are still several areas for improvement in this study. Firstly, further optimization of the deep learning model can be explored to enhance the accuracy and speed of orientation classification, allowing it to accommodate a wider range of crop types and

shooting scenarios. Secondly, expanding the dataset to include a greater variety of crop species and pest/disease conditions can improve the model's generalization ability and accuracy. Thirdly, integrating the guidance model with target detection, crop recognition, pest/disease classification, and phenotype analysis techniques can facilitate the design of more fine-grained guidance models tailored to specific scenario requirements, thereby advancing the practical implementation and application of this technology. Fourthly, further testing and analysis could be conducted on a wider range of mobile phone models and diverse systems (such as the iOS system) to enhance the coverage and universality of this technology. Fifthly, considering the testing environment, which is generally open and characterized by orderly and distinct plant growth, potential obstacles such as occlusions, difficulty in reaching designated positions, and curved or inclined plants may be encountered in other scenarios. Therefore, a detailed analysis of feasible guiding paths is required, along with the collection of occluded images for testing. Further research is needed for more extensive testing and optimization.

**Author Contributions:** Conceptualization, Y.J. and X.L.; methodology, Y.J., Q.Z. and Y.X.; software, Y.J. and Q.Z.; validation, Q.Z. and Y.X.; formal analysis, Y.J.; investigation, Y.J., Q.Z. and Y.X.; resources, Y.J., Y.X. and X.L.; data curation, Y.J.; writing—original draft preparation, Y.J., Q.Z. and Y.X.; writing—review and editing, Y.J., Q.Z. and Y.X.; visualization, Y.J. and Q.Z.; supervision, X.L. and X.C.; project administration, X.L. and X.C.; funding acquisition, X.L. and X.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by National Natural Science Foundation of China grant number 61601471. The APC was funded by National Natural Science Foundation of China grant number 61601471.

**Institutional Review Board Statement:** Not applicable.

**Data Availability Statement:** The dataset for guiding photography can be accessed via the following link: <https://1drv.ms/u/s!Aps4jbGXizHrgQF4f-NXDWcjQXP5?e=Usd9Sw>, accessed on 21 December 2023.

**Conflicts of Interest:** The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## References

1. Lv, Z.; Zhang, F.; Wei, X.; Huang, Y.; Li, J.; Zhang, Z. Tomato Flower and Fruit Recognition in Greenhouse Using Enhanced YOLOX-ViT Collaboration. *Trans. Chin. Soc. Agric. Eng.* **2023**, *39*, 124–134.
2. Li, Z.; Jiang, H.; Yang, X.; Cao, Z. Detection Method of Mung Bean Seedling and Weed Based on Lightweight Deep Learning Model. *J. Agric. Equip. Veh. Eng.* **2022**, *60*, 98–102.
3. Han, H.; Zhang, Y.; Qi, L. Review of Crop Disease and Pest Detection Based on Convolutional Neural Networks. *Smart Agric. Guide* **2023**, *3*, 6–9.
4. Song, H.; Jiao, Y.; Hua, Z.; Li, R.; Xu, X. Detection of Embryo Crack in Soaked Corn Based on YOLO v5-OBB and CT. *Trans. Chin. Soc. Agric. Mach.* **2023**, *54*, 394–401.
5. Li, H. *Statistical Learning Methods*; Tsinghua University Press: Beijing, China, 2012.
6. Zhang, Y. Research on Detection of Tomato Growth Status in Sunlight Greenhouse Based on Digital Image Technology. Ph.D. Thesis, Northwest A&F University, Xianyang, China, 2022.
7. Zhong, L.; Hu, L.; Zhou, H. Deep learning based multi-temporal crop classification. *Remote Sens. Environ.* **2019**, *2019*, 430–443. [[CrossRef](#)]
8. Aslan, M.F. Comparative Analysis of CNN Models and Bayesian Optimization-Based Machine Learning Algorithms in Leaf Type Classification. *Balk. J. Electr. Comput. Eng.* **2021**, *11*, 13–24. [[CrossRef](#)]
9. Lu, J.; Tan, L.; Jiang, H. Review on Convolutional Neural Network (CNN) Applied to Plant Leaf Disease Classification. *Agriculture* **2021**, *11*, 707. [[CrossRef](#)]
10. Chen, D.; Neumann, K.; Friedel, S.; Kilian, B.; Chen, M.; Altmann, T.; Klukas, C. Dissecting the phenotypic components of crop plant growth and drought responses based on high-throughput image analysis. *Plant Cell* **2014**, *26*, 4636–4655. [[CrossRef](#)]
11. Haug, S.; Ostermann, J. A crop/weed field image dataset for the evaluation of computer vision based precision agriculture tasks. *Comput. Vis. ECCV 2014 Work.* **2014**, *9*, 105–116.
12. Wang, S.; Hu, D.; Kou, D. A Shooting Method and Device. CN Patent CN110445978A[P], 15 December 2020.
13. Xie, Y.; Wu, K.; Liu, H. Control Method and Device for Aircraft, and Aircraft. CN Patent CN106125767B[P], 17 March 2020.

14. Wang, D.; Xie, F.; Yang, J.; Liu, Y. Industry Robotic Motion and Pose Recognition Method Based on Camera Pose Estimation and Neural Network. *Int. J. Adv. Robot. Syst.* **2021**, *18*, 17298814211018549. [CrossRef]
15. Wang, H.; Su, B.; Han, J. A Visual-Based Dynamic Object Tracking and Localization Method for Unmanned Aerial Vehicles. CN Patent CN103149939B[P], 21 October 2015.
16. Xie, K.; Yang, H.; Huang, S.; Lischinski, D.; Christie, M.; Xu, K.; Gong, M.; Cohen-Or, D.; Huang, H. Creating and Chaining Camera Moves for Quadrotor Videography. *ACM Trans. Graphs (TOG)* **2018**, *37*, 1–13. [CrossRef]
17. How Robots Can Pick Unknown Objects. Available online: <https://sereact.ai/posts/how-robots-can-pick-unknown-objects> (accessed on 5 April 2023).
18. Zhu, H.; Peng, X.; Wang, H. Selfie Guidance Method and Device for Selfie Terminals. CN Patent CN106911886A[P], 30 June 2017.
19. Feng, J.; Shu, P.; Denglapu, W.; Gamell, J. Video Conferencing Endpoint with Multiple Voice Tracking Cameras. CN Patent CN102256098B[P], 4 June 2014.
20. Yamanaka, N.; Yamamura, Y.; Mitsuzuka, K. An intelligent robotic camera system. *SMPTE J.* **1995**, *104*, 23–25. [CrossRef]
21. McKenna, S.J.; Gong, S. Real-time face pose estimation. *Real-Time Imaging* **1998**, *4*, 333–347. [CrossRef]
22. Breitenstein, M.D.; Daniel, K.; Thibaut, W.; Luc, V.G.; Hanspeter, P. Real-time face pose estimation from single range images. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 24–26 June 2008; pp. 1–8.
23. Erik, H.; Boon, K.L. Face Detection: A Survey. *Comput. Vis. Image Underst.* **2001**, *83*, 236–274.
24. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *CoRR* **2019**, *1*, 6105–6114.
25. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
26. Cohen, R.A.; Choi, H.; Baji, C.I.V. Lightweight compression of intermediate neural network features for collaborative intelligence. *IEEE Open J. Circuits Syst.* **2021**, *2*, 350–362. [CrossRef]
27. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.
28. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. Acn* **2017**, *60*, 84–90. [CrossRef]
29. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
30. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
31. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
32. Ma, N.; Zhang, X.; Zheng, H.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.
33. Zhu, L.; Li, Z.; Li, C.; Wu, J.; Yue, J. High performance vegetable classification from images based on alexnet deep learning model. *Int. J. Agric. Biol. Eng.* **2018**, *11*, 217–223. [CrossRef]
34. Jiang, B.; He, J.; Yang, S.; Fu, H.; Li, T.; Song, H.; He, D. Fusion of machine vision technology and AlexNet-CNNs deep learning network for the detection of postharvest apple pesticide residues. *Artif. Intell. Agric.* **2019**, *1*, 1–8.
35. Paymode, A.S.; Malode, V.B. Transfer learning for multi-crop leaf disease image classification using convolutional neural network VGG. *Artif. Intell. Agric.* **2022**, *6*, 23–33. [CrossRef]
36. Kumar, V.; Arora, H.; Sisodia, J. Resnet-based approach for detection and classification of plant leaf diseases. In Proceedings of the 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2–4 July 2020; IEEE: Delhi, India, 2020; pp. 495–502.
37. Bi, C.; Wang, J.; Duan, Y.; Fu, B.; Kang, J.-R.; Shi, Y. MobileNet based apple leaf diseases identification. In *Mobile Networks and Applications*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 1–9.
38. Hidayatuloh, A.; Nursalman, M.; Nugraha, E. Identification of tomato plant diseases by Leaf image using squeezenet model. In Proceedings of the 2018 International Conference on Information Technology Systems and Innovation (ICITSI), Bandung, Padang, 22–26 October 2018; pp. 199–204.
39. Sun, W.; Fu, B.; Zhang, Z. Maize Nitrogen Grading Estimation Method Based on UAV Images and an Improved Shufflenet Network. *Agronomy* **2023**, *13*, 1974. [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.