Flooding based mobilenet v3 identifies cucumber disease leaves in fuzzy scenes

Liu Yiming[a], Wang Zhengle[b], Wang Rujia[c] and *Gao Hongju[b]

[a] *School of Computer Science(National Pilot Software Engineering School),Beijing University of Posts and Telecommunications,100876,China;liuyimingbyr@bupt.edu.cn*

[b] *College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China;wangzhengle@cau.edu.cn;hjgao@cau.edu.cn*

[c] *School of Information and Communication Engineering,School of Computer Science,Beijing University of Posts and Telecommunications,100876,China;WNHwrj@bupt.edu.cn*

ARTICLE INFO

Keywords:
Flooding;Lightweight;Mobilenet v3;Cucumber disease;Crop question and answer

ABSTRACT

Domestic cucumber production is declining due to various pathologic diseases, but the technology of plant pathologic detection is not mature and requires high labor costs. In addition, since the planting site is usually a high-density scene, most photos taken are shot from various angles, and the background is messy, resulting in poor detection reliability. The rise of online question-and-answer systems is an inspiration. This paper wants to establish an online QA system. Farmers can upload cucumber pictures by taking photos, and the system can quickly identify and judge with high accuracy. In this paper, the crawler program is used to collect many cucumber leaf image data in batches on an agricultural website, and simple preprocessing is carried out. With a lightweight and fast MobileNetv3 network structure, it can quickly and accurately complete the seven kinds of cucumber leaf disease classification. The optimal network model is achieved by selecting appropriate parameters, optimizer, and batch capacity through the single variable method. In addition, a new training strategy of data set loss – flooding method was introduced in this paper, replacing the strategy of flooding after the flooding threshold was reached, which finally achieved 88.3% accuracy. Finally, two public data sets of PlantVillage and apple disease were selected for another experiment. The accuracy was up to 99% and 98.1%, respectively, which proved the universality of the strategy proposed in this paper. In this paper, the code will be open source in https://github.com/YiQuanMarx/Agricultural_Diseases_Dentification for reference.

CRediT authorship contribution statement

Liu Yiming: Responsible for paper experiment conception, data processing, main experiment realization, data processing, picture drawing and paper writing and polishing. **Wang Zhengle:** Participate in the preliminary research of the paper, responsible for the partial realization of the paper experiment and the preparation of the paper. **Wang Rujia:** Responsible for data processing, drawing and editing of the paper. ***Gao Hongju:** Responsible for framing the paper, providing data, guiding the writing of the paper, and polishing the paper.

1. Introduction

The cucumber, Cucumis sativus, is a widely cultivated creeping vine in the gourd family that usually bears cylindrical fruits and is used as a vegetable. According to statistics, in 2019, the world produced 88 million tons of cucumbers and gherkins, of which China accounted for 80 percent. However, global production of cucumbers is declining as various diseases plague them.

Traditional disease detection methods require manual inspection of diseased leaves through visual cues, which is easy to lead to low detection efficiency and poor reliability due to human error. In addition, this labor-intensive task is

ORCID(s): 0000-0002-3006-5919 (L. Yiming)

complicated and time-consuming by the large area to be detected and the millimeter-scale size of the early symptoms to be detected. Compounding the problem is a lack of expertise among farmers, and not enough agricultural experts able to spot these diseases also hinder overall harvests. Therefore, if available to farmers tools and techniques exist, early detection and classification of cucumber diseases can significantly alleviate these problems. The emergence of an online question-and-answer system provides a suitable treatment method. We can allow farmers to upload photos of their cucumber leaves through mobile devices such as mobile phones. After receiving the images, the system will process and analyze the pathological results of the cucumbers. The online question-and-answer system has been used in all aspects of life, including various shopping software, customer service, online hospital consultation, etc. However, the current online question-and-answer system is mainly based on manual identification processing, and agricultural manual inspection is unsuitable for this situation. Therefore, this paper needs to seek a kind of online question-and-answer system without a manual.

Currently, there are few methods for the pathological analysis of cucumber, including molecular analysis, spectral analysis, volatile organic compound analysis, etc. However, these methods are expensive and difficult to apply on a commercial scale. Martinelli et al. (2015) In this respect, computer vision has great inherent potential: symptoms of crop diseases often cause a feature on plant leaves that can be detected with image-based techniques and appropriate strategies. Crop diseases are detected and identified by analyzing the images' color, texture, and shape of diseased leaves. Benfenati et al. (2021)

However, many things could still be improved with the current approach. The first problem is that existing methods need to correctly identify fruit leaf diseases in the Chinese region because all current practices are trained only on the PlantVillage dataset, which is based on images from farms in the United States and Switzerland. Fruit diseases also differ from other regions due to differences in leaf shape, variety, and environmental factors. In addition, as about 80% of cucumbers are produced in China, there are few widely used data sets for training cucumber leaf detection models. Therefore, it is difficult for Chinese farmers to obtain cucumber disease detection technology with high accuracy. We urgently need to develop a new data set to detect diseases in cucumber leaves in regions of China so that Chinese farmers can identify diseases in cucumbers early, increase their income and boost the country's economic development.

Another problem is that professional experts and photographers mostly take the data sets widely used in training models. However, most of these photos are taken by farmers who cannot get the perfect shot for analysis, which can come in various backgrounds, colors, and sizes. Therefore, it is necessary to train the model in a data set containing non-specialized leaf images. The last problem is that most Chinese farmers need high-precision equipment for practical applications such as agriculture and generally use mobile terminal devices such as mobile phones. Therefore, we need small, low-latency models explicitly tailored for devices with small memory and low computing power. At the same time, the results of pathological tests are accurate.

Although previous work achieved high classification accuracy on its data set of images of natural cultivation conditions, several problems still need to be solved. First, deep learning-based disease diagnosis methods require many training images. Unlike other general computer vision tasks, labeling disease data sets requires specialized background knowledge difficult for farmers to master. Also, to collect perfect images of large data sets, plants must be grown in tightly controlled environments, which is labor-intensive and very expensive. Second, overfitting problems are particularly acute in plant diagnostic tasks because clues related to the disease are often unclear, and other factors, such as the image's background, often significantly impact the final decision. Not only that, but overfitting due to potential similarities in the data set often results in a significant decline in the accuracy of another data set (for example, images from other farms). For example, in cucumber disease diagnosis from wide-angle images, diagnostic performance on the same farm showed 86.0% in F1 scores but decreased to 20.7% in different farms. To solve the overfitting problem, Saikawa et al. Saikawa et al. (2019) proposed a method to remove background from the region of interest (RoI) as a pre-processing step. The results showed that they could improve accuracy by 12.2%. However, they also point out that this approach requires much more expensive masking data, potentially eliminating surrounding information essential to a diagnosis. Cap et al. (2020)

This paper is expected to use a lightweight and fast mobile ETV3 network structure to make it suitable for mobile terminal device recognition processing. The optimal network model is achieved by selecting appropriate parameters, optimizer, and batch capacity through the single variable method. To further improve its accuracy. With the epoch increase, if the loss on the test set reaches a certain threshold and continues training to reduce loss, overfitting will occur. This paper considers introducing a new training strategy for data set loss to replace the strategy to reduce loss after reaching the threshold to solve the overfitting problem on the test set. The flooding method is expected to improve the situation. Finally, it can achieve high accuracy and better apply to the daily agricultural life of cucumber pathological judgment.

2. Related work

Recent advances in artificial intelligence (AI), machine learning (ML), and computer vision (CV) technologies have opened up new possibilities, paving the way for the use of data from optical sensors in crop detection by automatically identifying relevant features. Deep learning is at the heart of intelligent farming through adopting new devices, technologies, and algorithms in agriculture 4. Deep learning is widely used to solve complex problems, such as feature extraction, transformation, pattern analysis, and image classification, which helps significantly develop, control, and improve agricultural production.

Over the past few decades, many types of deep learning architectures have been proposed for plant disease classification, resulting in several plant disease diagnosis systems tailored to real cultivation conditions.

In Prasanna Mohanty et al. (2016), Mohanty et al., using a large CNN(Google net and Alexnet), classify 26 diseases in 14 crops in 54, PlantVillage Repository Hughes et al. (2015), 306 labeled color images of diseased and healthy plant leaves formed public data and were trained. The trained model achieved 99.35 percent accuracy on the retention test set, demonstrating the feasibility of combining smartphones with computer vision to aid in plant disease diagnosis methods.

In Sladojevic et al. (2016), Sladojevic et al. used the deep learning framework CaffeNet to propose a new method to establish a plant disease recognition model. The developed model was able to identify 13 different types of plant diseases from healthy leaves and was able to distinguish plant leaves from their surroundings. The model was trained with 4483 (increased to 30,880) images downloaded from the Internet, and the PlantVillage dataset was used to evaluate the performance of the proposed technique. Experimental results on the developed model achieved an accuracy of between 91% and 98%, with an average of 96.3% for individual class tests.

In Karthik et al. (2020), Karthik et al. proposed a two-stage deep-learning technique for tomato leaf disease detection. The first architecture applies residual learning to learn essential features of classification. The second layer architecture applies the attention mechanism to the deep residual network. The experiment was conducted using the Plant Village Dataset, which contained three diseases: early blight, late blight, and leaf mold. The author takes advantage of the features CNN uses attention mechanism to learn in various processing hierarchies, and the overall accuracy of the verification set reaches 98% in five-fold cross-validation.

In Zhang et al. (2020), Zhang et al. proposed an improved fast RCNN to detect healthy tomato leaves and four diseases to improve the accuracy of the crop disease leaf recognition model and location of disease leaves. First, the author used a deep residual network instead of VGG16 for image feature extraction to obtain deeper disease features. Secondly, a k-means clustering algorithm is used to cluster the bounding box, and then anchoring is improved according to the clustering results. The improved anchoring framework is the genuine bounding box of the data set. Finally, the author conducts a k-means experiment with three feature extraction networks. The experimental results show that the improved method is 2.71% more accurate than the original fast RCNN, and the detection speed is faster.

Patrick et al. In Wspanialy and Moussa (2020), the authors propose a new computer vision system that can automatically identify several diseases, detect previously undetected diseases, and estimate the severity of each leaf. The model was trained and tested using several modified versions of nine tomato diseases from the PlantVillage tomato dataset and showed how different leaf attributes affect disease detection.

Kawasaki et al. Kawasaki et al. (2015) trained a three-layer convolutional neural network, which can automatically acquire features required for classification and obtain high classification performance to diagnose three types of cucumber diseases on real farm images where the target object has a complex background. Under the four-fold cross-validation strategy, the average accuracy of the model achieved 94.9%.

DeChant et al. DeChant et al. (2017) proposed an automatic system consisting of several layers of convolutional neural networks (CNN) for identifying large spot blight lesions on images obtained from maize plant fields and achieving an accuracy of 96.7% on the test set.

The above studies obtained high judgment accuracy through various convolutional neural networks, but they were all based on standardized images with transparent backgrounds. Once the background was blurred, the accuracy would be significantly reduced, which could not meet the requirements.

In Zhonghua et al. (2021), Ye Zhonghua et al. studied the real agricultural production environment and finally adopted the SSD target detection model through the comparison and improvement of different models to realize the prediction of crop image disease regions with complex backgrounds. The experimental results showed that the average accuracy of the final model in the test set reached 69.894%.

3. Dataset and method

3.1. Dataset

Most of the data sets used in previous studies are from the public data set PlantVillage, which has standard image specifications, simple and clear background, and accurate shooting details. However, the simple background pathological judgment does not apply to agricultural life.

In this paper, we used the crawler program written to collect a large number of cucumber leaf image data in batches on an agricultural website, which means that these images come from all over China, and most of these images are randomly taken by farmers with mobile terminal devices. In real life, most farmers use mobile phones to shoot, so there is no suitable equipment to shoot photos with high enough resolution. Moreover, due to the different models and specifications of mobile phones, the size and resolution of the images are also different, which requires us to process them further. Moreover, sample images will be shot directly on farmland without destroying crops, so the background of images is complex and changeable, and the shooting angles are diverse, as shown in Figure1.

With the help of plant pathologists, these images were labeled and became the data set for the experiment. The data set consisted of 2392 images, of which 80% were used for the training set and 598 images, or 20%, were used for the test set. As shown in the figure, we propose a lightweight and fast MobileNetv3 network structure that can quickly and accurately complete the classification of seven kinds of cucumber leaf diseases. The seven pathologic conditions are downy mildew, powdery mildew, bacterial angular leaf spot, target leaf spot, gummy stem blight, fusarium wilt, and anthracnose. Therefore, the machine vision system proposed in this paper for cucumber pathological diagnosis consists of three steps: image acquisition, preprocessing and classification, and network model optimization. This is shown in Figure2.

3.2. MobileNet v3

MobileNetV3 is also a lightweight network. MobileNetV3 uses a network architecture search (NAS) to search the global network structure by optimizing each network block, supplemented by the NetAdapt algorithm. This technique can efficiently determine an optimal model for a given hardware platform. In addition, MobileNetV3 uses the h-swish activation function to improve accuracy Howard et al. (2019)

In contrast to other classification models, it operates a single convolution at each depth of the input image rather than combining and flattening all the depths of the input, which is achieved by depth-oriented separable convolution. This deep convolution divides the convolution process into two layers, one for filtering and one for merging. This combination reduces the size of the model. MobileNetv3 consists of 4 2D convolution layers, 2 (112x122) bottleneck layers, 2 (56×56) bottleneck layers, 3 (28×28) bottleneck layers, 7 (14×14) bottleneck layers, and 2 (7×7) bottleneck layers, in which Swish and Relu are used for activation. Use a pooling layer (7x7) before two dense layers. Extrusion and excitation layers are also included, making it faster and lighter. This addition assigns unequal weights to channels when creating a map of output elements. Finally, a dense layer with 1024 units is applied to obtain the feature vector. The following Table1 is the network structure diagram of MobileNet v3 large. Input in the table is the size of the input image. The operator is the convolution layer or the reciprocal residual structure, Exp size and Out are the numbers of convolution kernels of the first and last layer of the reciprocal residual structure, respectively, and SE is whether the SE module is used. NL is the activation function used in the first and second layers of the reciprocal residual structure, and S is the step size of the deep convolution layer of the reciprocal residual structure.

3.3. Flooding

In this paper, the superiority of the network model is judged mainly by the loss size. Firstly, the generation mode of the loss function is introduced. This paper's experiment's loss function adopts the cross entropy loss function to classify the pathology of cucumber leaves into seven categories: $C = 7$ and batch capacity $N = 12$. The calculation formula of the loss function is as follows.1:

$$\ell(p, q) = L = \left\{l_1, \ldots, l_N\right\}^{\top}, \quad l_m = -\sum_{c=1}^{C} w_c \log \frac{\exp\left(x_{m,c}\right)}{\sum_{i=1}^{C} \exp\left(x_{m,i}\right)} y_{m,c} \tag{1}$$

Where x is the input,y is the target, w is the weight, and l is the loss function value.

The loss value of each data sample is calculated through the cross-entropy Loss function. Then the total loss function of an epoch is added and calculated according to the batch size to obtain the loss value of the image in the

Table 1
$MobileNet\_v3\_large$ network structure

Input	Operator	Exp size	Out	SE	NL	S
$224 \times 224 \times 3$	conv2d	×	16	×	h-swish	2
$112 \times 112 \times 16$	bneck, 3×3	16	16	×	relu	1
$112 \times 112 \times 16$	bneck, 3×3	64	24	×	relu	2
$56 \times 56 \times 24$	bneck, 3×3	72	24	×	relu	1
$56 \times 56 \times 24$	bneck, 5×5	72	40	√	relu	2
$28 \times 28 \times 40$	bneck, 5×5	120	40	√	relu	1
$28 \times 28 \times 40$	bneck, 5×5	120	40	√	relu	1
$28 \times 28 \times 40$	bneck, 3×3	240	80	×	h-swish	2
$14 \times 14 \times 80$	bneck, 3×3	200	80	×	h-swish	1
$14 \times 14 \times 80$	bneck, 3×3	184	80	×	h-swish	1
$14 \times 14 \times 80$	bneck, 3×3	184	80	×	h-swish	1
$14 \times 14 \times 80$	bneck, 3×3	480	112	×	h-swish	1
$14 \times 14 \times 112$	bneck, 3×3	672	112	√	h-swish	1
$14 \times 14 \times 112$	bneck, 5×5	672	160	√	h-swish	2
$7 \times 7 \times 160$	bneck, 5×5	960	160	√	h-swish	1
$7 \times 7 \times 160$	bneck, 5×5	960	160	√	h-swish	1
$7 \times 7 \times 160$	conv2d, 1×1	×	960	×	h-swish	1
$7 \times 7 \times 960$	pool, 7×7	×	×	×	×	1
$1 \times 1 \times 960$	conv2d 1×1, NBN	×	1280	×	h-swish	1
$1 \times 1 \times 1280$	conv2d 1×1, NBN	×	k	×	×	1

experiment in Chapter 4. That is,2:

$$\ell(p, q) = \sum_{m=1}^{N} l_m \tag{2}$$

In this paper, the size of the loss function is taken as the benchmark for the superiority of the network model. In the follow-up experiments, we will find that the network model we used has an overfitting phenomenon. A loss evaluation strategy needs to be replaced. After reaching a certain threshold, the strategy does not take simple loss decline as the training orientation. In this way, the loss on the test set shows a relatively flat trend, and then the rising speed of the loss on the test set will be reduced, and even a secondary decline may occur. Finally, the accuracy is further improved to some extent. We needed a way to solve this problem, and the flooding method came into being. Ishida et al. (2020)

Consider input variable $\boldsymbol{p} \in \mathbb{J}^d$ and output variable $q \in [C] := \{1, \ldots, C\}$, where C is the number of classes. They follow an unknown joint probability distribution with density $p(\boldsymbol{p}, q)$. We denote the score function by $\boldsymbol{f} : \mathbb{J}^d \to \mathbb{J}^C$. For any test data point p_0 , our prediction of the output label will be given by $\widehat{q}_0 := \arg\max_{z \in [C]} f_z\left(p_0\right)$, where $f_z(\cdot)$ is the z -th element of $f(\cdot)$, and in case of a tie, arg max returns the largest argument. Let $\ell : \mathbb{J}^C \times [C] \to \mathbb{J}$ denote a loss function. ℓ can be the zero-one loss, where $w := \left(w_1, \ldots, w_C\right)^\top \in \mathbb{J}^C$, or a surrogate loss such as the softmax cross-entropy loss,3:

$$\ell_{\mathrm{CE}}\left(\boldsymbol{w}, z'\right) := -\log \frac{\exp\left(w_{z'}\right)}{\sum_{z\in[C]} \exp\left(w_z\right)}. \tag{3}$$

For a surrogate loss ℓ , we denote the classification risk. The goal of multi-class classification is to learn f that minimizes the classification error $J_{01}(f)$. In optimization, we consider the minimization of the risk with a almost surely differentiable surrogate loss $J(\boldsymbol{f})$ instead to make the problem more tractable. Furthermore, since $p(p,q)$ is usually unknown and there is no way to exactly evaluate $J(\boldsymbol{f})$, we minimize its empirical version calculated from the training data instead4:

$$\widehat{J}(\boldsymbol{f}) := \frac{1}{m}\sum_{i=1}^{m} \ell\left(\boldsymbol{f}\left(\boldsymbol{p}_i\right), q_i\right) \tag{4}$$

where $\left\{\left(p_i, q_i\right)\right\}_{i=1}^{m}$ are i.i.d. sampled from $p(p,q)$. We call $\widehat{J}$ the empirical risk.

Definition1. The flooded empirical risk is defined as 4

$$\widetilde{J}(\boldsymbol{f}) = |\widehat{J}(\boldsymbol{f}) - b| + b \tag{5}$$

Note that when $b = 0$, then $\widetilde{J}(\boldsymbol{f}) = \widehat{J}(f)$. The gradient of $\widetilde{J}(f)$ w.r.t. model parameters will point to the same direction as that of $\widehat{J}(\boldsymbol{f})$ when $\widehat{J}(\boldsymbol{f}) > b$ but in the opposite direction when $\widehat{J}(\boldsymbol{f}) < b$. This means that when the learning objective is above the flood level, we perform gradient descent as usual (gravity zone), but when the learning objective is below the flood level, we perform gradient ascent instead (buoyancy zone).Pushing the parameters towards a more stable region keeps the convergence of the loss function near a threshold value, which improves the generalization performance and better resists perturbations.

4. Experiment

In this experiment, the image was preprocessed first, and the PyTorch framework was used to scale the image to 448×448 for data standardization. In this paper, Mobilenet v3 network model was selected as well as optimizer ASGD, the learning rate was set to 0.001, the L1 regularity coefficients were all 0.01, the batch size was 12, and 300 rounds of iterative training were conducted on the training set and the test set respectively. In order to prevent overfitting in the experiment, we also apply the algorithm of Dropout to randomly inactivate the neural nodes in the network before network training, reduce the interdependence between neurons, and thus ensure the extraction of important features that are independent of each other and improve the generalization ability of the model. As shown in the figure, it can be seen that neurons randomly deactivate seven neural nodes in the network.

4.1. Contrast test

We selected seven mainstream network models and MobileNet v3 network for a comparison test on the same cucumber pathological leaf image data set. The experimental data accurately reflected the superiority of MobileNet v3 network.

In this paper, Alexnet, Resnet, VGG, Efficientnet v2, Efficientnet v3, Efficientnet v7, Mobilenet v2, and Mobilenet v3 leaf pathological recognition models were trained, and image training set and test set were used to test and compare them. This way, the network model performance's superiority is tested and further optimized. The figure4 shows the experimental results, in which vg19 represents VGG model, alex represents Alexnet model, re50 represents Resnet model, mob3 represents Mobilenet v3 model, mob2 represents mobilenet v2 model. eff7 represents the Efficientnet v7 model, eff3 represents the Efficientnet v3 model, and eff2 represents the Efficientnet v2 model.

The loss function and accuracy of the training set of Alexnet model converge well. When epoch=100, they begin to converge and gradually become stable. However, the effect on the test set could be better, loss and accuracy fluctuate considerably, and the convergence effect could be better. Compared with other models, its loss in the test set is higher, its accuracy is lowest, and its performance is poor.

The training set of the VGG model converges quickly, and the loss image of the data set begins to converge in the 70th round of iteration. The accuracy image of the training set and the test set converge faster and in about 30 iterations. However, the degree of fitting in the test set is not high, and the test set loss and accuracy image of the VGG model show no convergence trend. The average accuracy is less than 70%.

The training set of Resnet begins to converge when the number of training rounds is around 40, and the convergence speed is breakneck. When testing the test set, we found that the convergence fitting degree is considerably high, and the maximum accuracy is as high as 81.4%. However, the fluctuation range of loss and accuracy of the test set is larger than that of Alexnet network model numerically, and the loss function also appears to be an overfitting phenomenon.

The results of Efficientnet v2 and Efficientnet v7 models are the same. The loss and accuracy of the test set tend to converge, and the fitting degree is higher than that of the training set. However, the degree of overfitting of the loss image of the test set is too high and fluctuates wildly.

The Efficientnet v3 model, where the precision image of the test set begins to converge around the 150th iteration round, is the slowest of all models. The test set of Efficientnet v3 shows a convergence effect, but a severe overfitting phenomenon occurs, and the fluctuation is the largest from the experimental results.

In the MobileNet v2 model, the training set starts to converge from the 30th iteration, the loss finally keeps approaching 0, and the accuracy also keeps increasing with the training rounds. The resulting trend of the test set also roughly fits the training set, but there is an overfitting phenomenon. The generalization degree is shallow, and the fluctuation degree is enormous.

Considering the fitting effect of each model test set comprehensively, the test set result trend of Mobilenet v3 network is consistent with the training set trend, and its maximum accuracy is relatively the highest, reaching 81.3% or above. Moreover, Mobilenet v3 converges 66% faster than other networks due to its lightweight framework. Therefore, we finally chose Mobilenet v3 network for the next optimization experiment.

4.2. The choice of optimizer

After selecting Mobilenet v3 as the final experimental network, this paper optimizes it. The first is the selection of the optimizer. The optimizer is used to update and calculate network parameters that affect model training and model output to approximate or reach the optimal value, thereby minimizing (or maximizing) the loss function. Choosing an appropriate optimizer can make our network model reach convergence faster and achieve better accuracy. On the same cucumber pathological leaf image data set, Mobilenet v3 network was selected in this paper, and the learning rate was set to 0.001, the regularity coefficients of L1 and L2 were both 0.01, and 300 rounds of iterative training were conducted on the training set and test set respectively. In this paper, ASGD, SGD, RMSprop, RAdam, NAdam, AdamW, Adamax, Adadelta, and Agagrad are selected for comparison, and loss functions and accurate images of the training set and test set are obtained, as shown in the figure. 5

As shown in the figure, although the four optimizers, RAdam, NAdam, Adam, and RMSprop, all have a convergence trend at last, their loss value on the training set is very high, and their highest accuracy is not more than 60%. Compared with other optimizers, the effect could be better, and they are unsuitable for this paper's network model.

AdamW optimizer performs well in the training set. The convergence rate of the loss function and accuracy image is the fastest compared with other optimizers. However, its performance in the test set could be better. The loss function and accuracy image have large fluctuations, and its maximum accuracy is at most 70%.

Adamax optimizer begins to converge after 100 iteration rounds of the training set. Its loss function image value in the test set is higher than that of the ten optimizers, and its accuracy is low, with an average accuracy of less than 65%.

The data set images of Adadelta and Adagrad optimizers almost coincide. Both the training set and the test set converge. The loss value gradually decreases with the increase in the number of iterations, which is the lowest among the ten optimizers. The accuracy also increases with the number of iterations, reaching a high accuracy of 83%. However, its convergence speed could be faster. The training set begins to converge in the 250th iteration round, and the test set begins to converge in the 150th iteration round, which takes the longest time.

The data set images of the two optimizers, ASGD and SGD, almost coincide, converge in both the training set and the test set, and the convergence is faster. The training set begins to converge in the 80th iteration round, and the test set begins to converge in the 20th iteration. The accuracy of the test set peaked at 81.4%. Numerically, the ASGD has

Table 2
Use effect of different batch sizes

batch_size	max_acc(%)	mean_acc(%)	sstd_acc(%)	max_loss	mean_loss	std_loss
4	79.870	64.197	10.916	1367.639	367.173	181.912
8	81.794	75.492	3.227	178.613	125.196	20.972
12	82.115	76.568	2.962	135.261	76.619	9.009
16	81.542	76.848	3.556	123.976	55.940	6.853
20	82.451	76.626	4.069	118.699	44.232	6.210
24	81.751	76.854	4.498	117.166	36.638	5.916
28	82.245	76.527	5.140	113.502	31.251	5.801
32	82.230	76.486	5.303	112.958	27.393	5.897

less fluctuation than the SGD optimizer.

Therefore, after comparing convergence speed, fitting degree, accuracy, loss function size, and other aspects, ASGD has a higher peak value, faster convergence, and minor fluctuation. In this article, the ASGD is chosen as the final optimizer.

subsectionThe choice of batch size In this comparison experiment, Mobilenet v3 network model was selected, the optimizer was ASGD, the learning rate was set to 0.001, the regularity coefficients of L1 and L2 were both 0.01, and 300 rounds of iterative training were conducted on the training set and test set respectively. Batch size was selected as 4, 8, 12, 16, 20, 24, 28, and 32. The images and data results were obtained as shown in the figure below. 6

Batch $size = 4$, as the batch size value is too small, the gradient of each layer has high randomness and takes much time. The resulting image also fluctuates, and the final precision effect is considerably poor, resulting in an underfitting phenomenon. The convergence effect is not good enough.

It can be seen from the figure6 that the convergence speed increases with the increase in batch size. According to the numerical results,2 with the increase of batch size, the maximum loss function, average loss function, and the standard deviation of the loss function, namely the volatility, of the test set gradually decrease. However, after batch $size = 12$, each loss value changes little with the increase in batch size. In addition, the maximum accuracy after convergence increases weakly and sometimes even regresses. Moreover, after batch $size = 12$, the standard deviation of the accuracy of the test set began to rise continuously, indicating that the model's generalization ability declined. Before batch $size = 12$, the test set's accuracy increases while the loss fluctuation decreases. When batch $size = 12$, the standard deviation of loss is minimum, and the anti-aliasing effect is best. At the same time, the accuracy of the test set increased to 82.1%.

The experiment in this paper is carried out under a blurred background image, so we need as much generalization ability as possible. Moreover, the model proposed in this paper should apply to mobile terminal devices, should be as lightweight as possible, and need to select the smallest batch size value possible. Therefore, from the perspective of background requirements and image data analysis, batch $size = 12$ was selected as the optimal experimental parameter

Table 3
Use effect after different values of parameter b in flooding

b	max_acc(%)	mean_acc(%)	sstd_acc(%)	max_loss	mean_loss	std_loss
0.349	81.644	76.025	2.980	139.207	63.706	5.395
0.291	81.608	76.508	3.076	136.610	65.388	5.288
0.252	82.068	76.344	3.080	138.879	67.156	5.405
0.311	83.308	78.263	2.658	135.187	64.881	5.256
0.297	81.641	76.315	2.980	139.065	65.060	5.400
0.330	80.890	76.395	3.248	138.475	64.670	5.387
0.297	81.375	76.458	2.953	138.858	64.929	5.319
0.296	81.568	76.385	3.146	137.580	65.504	5.320
0.274	82.557	76.497	3.249	136.909	65.820	5.494
0.171	82.271	76.717	3.136	136.269	69.950	6.173
0.232	82.820	76.610	3.306	136.021	67.736	5.490
0.265	81.786	76.702	3.238	134.727	66.250	5.396
0.256	81.828	76.217	3.118	138.510	66.854	5.473
0.207	81.634	76.496	3.263	137.237	69.357	5.813
0.223	82.588	76.782	3.382	136.995	67.819	5.571

in this paper.

4.3. Flooding

In Chapter 3, we introduced the basic principle and used a flooding mode. By changing the loss function and adding a threshold, the loss eventually fluctuated around the threshold. Flooding allows us to directly select the level of training loss, which is difficult to achieve with other regularizers. There was an overfitting phenomenon in the loss result in images of the Mobilenet v3 experiment mentioned above. In this section, flooding was used to realize the secondary decrease of data set loss and prevent overfitting.

In this experiment, the optimizer used ASGD, the learning rate was 0.001, the regularization coefficients of L1 and L2 were 0.01, and 300 iteration experiments were conducted. The loss threshold is set with 15 different values for comparative analysis of images and data.

As seen from the image,7 after flooding was added, the overfitting rising trend of the loss function image in the test set was effectively suppressed. When $b = 0.310$, 0.348, and 0.290, the flooding not only resulted in good inhibition but also resulted in secondary descending, which solved the overfitting problem.

According to a series of comparisons of table data, after adding flooding to 3, the mean test set accuracy increased by 0.2%, and the maximum test set accuracy increased by 0.5%. The final goal of this paper is to select the test set with the highest accuracy to achieve the best pathological recognition effect of the cucumber leaf image. The final selection threshold is 0.310, at which time the overfitting of the loss function is well suppressed, and the accuracy is up to 83.3%.

We compared the two experiments without flooding with the flooding method. The results are shown in the figure.8 Methods with flooding tend to improve test accuracy compared to baseline methods without flooding. Continue to train the model without flooding until, eventually, the loss function may continue to rise and accuracy may decline. However, according to the results, the final model has good predictive performance when there is flooding, which means that flooding helps improve test accuracy in later training. During training with flooding, test losses became lower and flatter. On the other hand, the training loss reached a secondary decline and continued to float around the flooding threshold with stability.

4.4. Discussion

In this study, the data set was replaced with PlantVillage public data set and another public apple disease data set in China to conduct the pathological judgment experiment of apple leaves. In this experiment, 10 rounds of iterative experiments were conducted. The experimental results are shown in the figure below, where the apple curve represents the experimental results using the apple disease data set. The plant curve represented the experimental results using the PlantVillage public data set.

As can be seen from the image results,9 the loss function of the training set and the test set is constantly close to 0, and the accuracy also increases with the increase of iteration rounds. The accuracy of PlantVillage public data set after applying the strategy in this paper is as high as 99%, and the accuracy of the apple disease data set is also as high as 98.1%, which is far higher than the 76.5% accuracy of Zhou Minmin's apple-leaf-disease-detection-system based on transfer learning.Minmin (2019) It is proved that compared with the existing strategies, the proposed strategies are universal, accurate, and less time-consuming and can better meet the needs of Chinese farmers for crop pathological judgment in today's society.

5. Conclusion

In today's society, the rise of the online QA system has brought great convenience to people's lives, but it is not widely used in agriculture. The pathological judgment of agricultural plants is an essential part of agricultural planting life. Today's crop pathological judgment mostly requires high labor costs, low detection efficiency, and poor reliability because of its dense growing environment and chaotic background.

To solve these problems, this paper proposed a Mobilenet v3 based on flooding to identify crop leaves in fuzzy scenes. It satisfied the requirement of mobile terminal using a lightweight framework and could quickly and accurately judge crop pathological conditions through farmers' shooting pictures.

In this paper, cucumber leaf images were randomly collected from a Chinese agricultural website and labeled. A dataset with complex image background was constructed, and seven kinds of cucumber leaf pathologic judgments were made. Through the control variable method, the network model, the optimizer, and batch size, three rounds of experiments were compared and analyzed to achieve the optimal network model. In this paper, flooding method was used to replace an evaluation strategy of loss. The accuracy of the test set was increased by 0.5% again, reaching the highest 88.3%. Finally, two public data sets of PlantVillage and apple disease were selected for the experiment again. The accuracy was up to 99% and 98.1%, respectively, which proved the universality of the proposed strategy and its high practical value.

References

Benfenati, A., Causin, P., Oberti, R., Stefanello, G., 2021. Unsupervised deep learning techniques for powdery mildew recognition based on multispectral imaging. arXiv preprint arXiv:2112.11242 .

Cap, Q.H., Uga, H., Kagiwada, S., Iyatomi, H., 2020. Leafgan: An effective data augmentation method for practical plant disease diagnosis. IEEE Transactions on Automation Science and Engineering .

DeChant, C., Wiesner-Hanks, T., Chen, S., Stewart, E.L., Yosinski, J., Gore, M.A., Nelson, R.J., Lipson, H., 2017. Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning. Phytopathology 107, 1426–1432.

Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al., 2019. Searching for mobilenetv3, in: Proceedings of the IEEE/CVF international conference on computer vision, pp. 1314–1324.

Hughes, D., Salathé, M., et al., 2015. An open access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv preprint arXiv:1511.08060 .

Ishida, T., Yamane, I., Sakai, T., Niu, G., Sugiyama, M., 2020. Do we need zero training loss after achieving zero training error? arXiv preprint arXiv:2002.08709 .

Karthik, R., Hariharan, M., Anand, S., Mathikshara, P., Johnson, A., Menaka, R., 2020. Attention embedded residual cnn for disease detection in tomato leaves. Applied Soft Computing 86, 105933.

Kawasaki, Y., Uga, H., Kagiwada, S., Iyatomi, H., 2015. Basic study of automated diagnosis of viral plant diseases using convolutional neural networks, in: International symposium on visual computing, Springer. pp. 638–645.

Martinelli, F., Scalenghe, R., Davino, S., Panno, S., Scuderi, G., Ruisi, P., Villa, P., Stroppiana, D., Boschetti, M., Goulart, L.R., et al., 2015. Advanced methods of plant disease detection. a review. Agronomy for Sustainable Development 35, 1–25.

Minmin, Z., 2019. Research on Android Detection System of Apple Leaf Disease Based on Transfer Learning. Master's thesis. Northwest AF University.

Prasanna Mohanty, S., Hughes, D., Salathe, M., 2016. Using deep learning for image-based plant disease detection. arXiv e-prints , arXiv–1604.

Saikawa, T., Cap, Q.H., Kagiwada, S., Uga, H., Iyatomi, H., 2019. Aop: An anti-overfitting pretreatment for practical image-based plant diagnosis, in: 2019 IEEE International Conference on Big Data (Big Data), IEEE. pp. 5177–5182.

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., Stefanovic, D., 2016. Deep neural networks based recognition of plant diseases by leaf image classification. Computational intelligence and neuroscience 2016.

Wspanialy, P., Moussa, M., 2020. A detection and severity estimation system for generic diseases of tomato greenhouse plants. Computers and Electronics in Agriculture 178, 105701.

Zhang, Y., Song, C., Zhang, D., 2020. Deep learning-based object detection improvement for tomato disease. IEEE Access 8, 56607–56614.

Zhonghua, Y., Mingxia, Z., Lu, J., 2021. Research on crop disease image recognition with complex background. Transactions of the Chinese Society for Agricultural Machinery .

List of Figures