

¹ Flooding-based mobilenet to identify cucumber diseases from leaf ² images in natural scenes

³ Liu Yiming^a, Wang Zhngle^b, Wang Rujia^c, Chen Jiasi^d and *Gao Hongju^b

⁴ ^aSchool of Computer Science(National Pilot Software Engineering School),Beijing University of Posts and
⁵ Telecommunications,100876,China;liuyimingbyr@bupt.edu.cn

⁶ ^bCollege of Information and Electrical Engineering, China Agricultural University, Beijing 100083,
⁷ China;wangzhngle@cau.edu.cn;hjgao@cau.edu.cn

⁸ ^cSchool of Information and Communication Engineering,School of Computer Science,Beijing University of Posts and
⁹ Telecommunications,100876,China;WNHwrj@bupt.edu.cn

¹⁰ ^d School of Artificial Intelligence,School of Computer Science,Beijing University of Posts and
¹¹ Telecommunications,100876,China;chenjsi@bupt.edu.cn

¹² ARTICLE INFO

¹⁶ *Keywords:*

¹⁷ Flooding;Lightweight;MobileNet v3;
¹⁸ Cucumber disease;Crop QA

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. In this paper, cucumber leaf image data in batches is collected on agricultural website, and simply processed. The system to identify cucumber diseases from leaf images in natural scenes is established so that famers can detect diseases more quickly. Farmers can upload cucumber pictures by taking photos, and the system can quickly identify and judge with high accuracy. With a lightweight and fast MobileNetv3 network structure, seven kinds of cucumber leaf disease classification can be quickly and accurately completed. The 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 is introduced in this paper, replacing the strategy of loss decline, which finally achieves 83.3% accuracy on our data set. Finally, two public data sets of PlantVillage and apple disease are selected for another experiment. The accuracy is up to 99% and 98.1%, which proves the universality of the strategy proposed in this paper. The code for all the experiments will be open source in https://github.com/YiQuanMarx/Agricultural_Diseases_Dentification for reference.

³³

³⁴ CRedit authorship contribution statement

³⁵ **Liu Yiming:** Responsible for paper experiment conception, data processing, main experiment realization, data
³⁶ processing, picture drawing and paper writing and polishing. **Wang Zhngle:** Participate in the preliminary research
³⁷ of the paper, responsible for the partial realization of the paper experiment and the preparation of the paper. **Wang**
³⁸ **Rujia:** Responsible for data processing, drawing and editing of the paper. **Chen Jiasi:** Participate in some experiments
³⁹ of the paper and polishing of the writing part. ***Gao Hongju:** Responsible for framing the paper, providing data,
⁴⁰ guiding the writing of the paper, and polishing the paper.

⁴¹ 1. Introduction

⁴² The cucumber, *Cucumis sativus*, is a widely cultivated creeping vine in the gourd family that usually bears cylindri-
⁴³ cal fruits and is used as a vegetable. According to statistics, in 2019, the world produced 88 million tons of cucumbers
⁴⁴ and gherkins, of which China accounted for 80 percent. However, the global yield of cucumber is declining due to
⁴⁵ various diseases.

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

46 Traditional disease detection methods require manual inspection of diseased leaves through visual clues. Due
47 to human error, it is easy to lead to low detection efficiency and poor reliability. In addition, this labor-intensive task
48 becomes complicated and time-consuming due to the large area to be detected and the millimeter-scale size of the early
49 symptoms to be detected. Compounding the problem is a lack of expertise among farmers, and not enough agricultural
50 experts able to spot these diseases also hinder overall harvests. Therefore, if the early detection and classification of
51 cucumber disease can be provided to farmers in terms of tools and technology, it can greatly alleviate all the above
52 problems. The emergence of online question-and-answer system provides us with a suitable processing method. We
53 can let farmers take photos of their cucumber leaves through mobile devices such as mobile phones and upload them.
54 After receiving the images, the system will process and analyze the pathological results of the cucumber. At present,
55 online question-and-answer system has been used in all aspects of life, including customer service of various shopping
56 softwares, online hospital consultation, etc. Therefore, A lightweight, efficient cucumber diseases detaction network
57 is needed which can be used on mobile phone and make free from manual work.

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

64 However, there are still many problems in the current method. The first problem is that existing methods cannot
65 correctly identify fruit leaf diseases in the China region, because all current practices are trained only on the PlantVil-
66 lage data set, which is based on images from farms in the United States and Switzerland. Fruit diseases also differ
67 from other regions due to differences in leaf shapes, varieties, and environmental factors. In addition, as about 80%
68 of cucumbers are produced in China, there are few widely used data sets for cucumber leaf detection model training.
69 It is difficult for Chinese farmers to obtain cucumber disease detection technology with high detection accuracy. We
70 urgently need to develop a new data set to detect the disease of cucumber leaves in China, so that Chinese farmers
71 can determine the disease of cucumber at an early stage, increase their income and boost the country's economic
72 development.

73 Another problem is that the data sets widely used in training models are mostly shot by professional experts and
74 photographers. But in reality, most of these photos are taken by farmers who cannot take perfect photos for analysis.
75 The photos may have various backgrounds, colors, and sizes. Therefore, it is necessary to train the model in a data
76 set containing non-specialized leaf images. The last problem is that in practical applications such as agriculture, most
77 Chinese farmers do not have high-precision equipment, and generally use mobile terminal devices such as mobile

78 phones. Therefore, we need small, low-latency models explicitly tailored for devices with small memory and low
79 computing power. At the same time, the results of pathological tests are accurate.

80 Although previous work has achieved high classification accuracy on its data set of images of natural cultivation
81 conditions, there are still several problems. First, deep learning-based disease diagnosis methods require many train-
82 ing images. Unlike other general computer vision tasks, marking disease data sets requires professional background
83 knowledge, which is difficult for farmers to master. In addition, in order to collect perfect images of large data sets,
84 plants must grow in a tightly controlled environment, which is labor-intensive and very expensive. Secondly, the over-
85 fitting problem is particularly acute in plant diagnostic tasks because clues related to the disease are often unclear,
86 and other factors, such as the image's background, often have a significantly impact on the final decision. Not only
87 that, the over-fitting caused by potential similarity of the data set often leads to a significant decline in the accuracy of
88 another data set (for example, images from other farms). For example, in cucumber disease diagnosis from wide-angle
89 images, the diagnostic performance in the same farm showed 86.0% in F1 scores but decreased to 20.7% in different
90 farms. To solve the over-fitting problem, Saikawa et al. proposed a method to remove background from the region of
91 interest (RoI) as a preprocessing step (Saikawa et al., 2019). The results showed that they could improve accuracy
92 by 12.2%. However, they also point out that this approach requires a large amount of expensive shielding data, which
93 may eliminate the surrounding information essential to for diagnosis (Cap et al., 2020).

94 This paper is expected to use a lightweight and fast MobileNetv3 network structure to make the network structure
95 suitable for mobile terminal device recognition processing. The network model is achieved by selecting appropriate
96 parameters, optimizer, and batch capacity through the single variable method to further improve its accuracy. With
97 the increase of epoch, if the loss on the test set reaches a certain threshold and continues training to reduce loss, over-
98 fitting will occur. A new training set loss training strategy is introduced in this paper replacing the strategy which is
99 just reducing loss to solve the over-fitting problem on the test set. The flooding method is expected to improve the
100 situation. Finally, it will reach a higher accuracy and be better applicable to the daily agricultural life of cucumber
101 pathological judgment.

102 2. Related work

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

109 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.

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

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

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

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

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

139 Kawasaki et al. (Kawasaki et al., 2015) trained a three-layer convolutional neural network, which can automatically
140 acquire the features required for classification and obtain high classification performance to diagnose three kinds of

₁₄₁ cucumber diseases on real farm images, in which the target object had a complex background. Under the four-fold
₁₄₂ cross-validation strategy, the author's model achieved an average accuracy of 94.9%.

₁₄₃ DeChant et al. (DeChant et al., 2017) proposed an automatic system consisting of several layers of convolutional
₁₄₄ neural networks (CNN) for identifying large spot blight lesions on images obtained from maize plant fields and achiev-
₁₄₅ ing an accuracy of 96.7% on the test set.

₁₄₆ The above studies have achieved high accuracy of judgment through various convolutional neural networks, but
₁₄₇ they were all based on standardized images with transparent background. Once the background is blurred, its accuracy
₁₄₈ will be significantly reduced and cannot meet the requirements.

₁₄₉ Ye Zhonghua et al. studied the real agricultural production environment and finally adopted the SSD target de-
₁₅₀ tection model through the comparison and improvement of different models to realize the prediction of crop image
₁₅₁ disease regions with complex backgrounds (Zhonghua et al., 2021). The experimental results showed that the average
₁₅₂ accuracy of the final model in the test set reached 69.894%.

₁₅₃ 3. Material and Methods

₁₅₄ 3.1. Material

₁₅₅ Most of the data sets used in previous studies are from the public data set PlantVillage, which has standard image
₁₅₆ specifications, simple and clear background, and accurate shooting details. However, the simple background patho-
₁₅₇ logical judgment does not applicable to actual agricultural life.

₁₅₈ In this paper, we collect a large number of cucumber leaf image data on Chinese websites in batches, which means
₁₅₉ that these images come from all over China, and most of these images are randomly taken by farmers using mobile
₁₆₀ terminal devices. In real life, most farmers use mobile phones to shoot, so there is no suitable equipment to take
₁₆₁ photos with high enough definition. And due to the different models and specifications of mobile phones, the size and
₁₆₂ definition of the images are also different, which requires us to process them further. Moreover, the sample images will
₁₆₃ be shot directly in the farmland without destroying crops, so the background of images is complex and changeable,
₁₆₄ and the shooting angles are diverse, as shown in Figure1.

₁₆₅ With the help of plant pathologists, these images are labeled and became the data set of this experiment. The
₁₆₆ data set consist of 2392 images, of which 80% are used for the training set and 598 images (20%), are used as the
₁₆₇ test set. As shown in the Figure2, we propose a lightweight and fast MobileNetv3 network structure that can quickly
₁₆₈ and accurately complete the classification of seven kinds of cucumber leaf diseases. The seven pathologic conditions
₁₆₉ are downy mildew, powdery mildew, bacterial angular leaf spot, target leaf spot, gummy stem blight, fusarium wilt,
₁₇₀ and anthracnose. Therefore, the machine vision system for cucumber pathological diagnosis proposed in this paper
₁₇₁ includes three steps: image acquisition, preprocessing and classification, and then the network model is optimized.

₁₇₂ This is shown in Figure2.

₁₇₃ 3.2. MobileNet v3

₁₇₄ MobileNetV3 is a lightweight network. MobileNetV3 uses a network architecture search (NAS) to search the
₁₇₅ global network structure by optimizing each network block, supplemented by the NetAdapt algorithm. This technique
₁₇₆ can efficiently determine an optimal model for a given hardware platform. In addition, MobileNetV3 uses the h-swish
₁₇₇ activation function to improve accuracy (Howard et al., 2019)

₁₇₈ In contrast to other classification models, it operates a single convolution at each depth of the input image, rather
₁₇₉ than combining and flattening all the depths of the input image, which is achieved through depth-oriented separable
₁₈₀ convolution. This deep convolution divides the convolution process into two layers, one for filtering and the other for
₁₈₁ merging. This combination reduces the size of the model. MobileNetv3 consists of four 2D convolution layers, two
₁₈₂ (112×122) bottleneck layers, two (56×56) bottleneck layers, three (28×28) bottleneck layers, seven (14×14) bottleneck
₁₈₃ layers, and two (7×7) bottleneck layers, in which Swish and Relu are used as activation. Use a aggregation layer (7×7)
₁₈₄ before two dense layers. Extrusion and excitation layers are also included to make it faster and lighter. This addition
₁₈₅ assigns unequal weights to channels when creating a map of output elements. Finally, a dense layer with 1024 units
₁₈₆ is applied to obtain the feature vector. The following Table1 shows the network structure of MobileNetv3 large. In
₁₈₇ the table, Input is the size of the input image, Operator is the convolution layer or the reciprocal residual structure,
₁₈₈ Exp size and Out are the number of convolution kernels in the first and last layer of the reciprocal residual structure,
₁₈₉ respectively, and SE is whether to use the SE module. NL is the activation function used in the first and second layers
₁₉₀ of the reciprocal residual structure, and S is the step length of the deep convolution layer of the reciprocal residual
₁₉₁ structure.

₁₉₂ 3.3. Flooding

In this paper, the superiority of the network model is judged mainly by the loss size. Firstly, we introduce the generation method of loss function. 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:

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

₁₉₃ 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 our chapter 4

Table 1*MobileNet_v3_large* network structure

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

experiment. That is,2:

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

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

Consider input variable $p \in \mathbb{J}^d$ and output variable $q \in [C] := \{1, \dots, C\}$, where C is the number of classes. They follow an unknown joint probability distribution with density $p(p, q)$. We denote the score function by $f : \mathbb{J}^d \rightarrow \mathbb{J}^C$. For any test data point p_0 , our prediction of the output label will be given by $\hat{q}_0 := \arg \max_{z \in [C]} f_z(p_0)$, 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 $\ell : \mathbb{J}^C \times [C] \rightarrow \mathbb{J}$ denote a loss function. ℓ can be the zero-one loss, where $w := (w_1, \dots, w_C)^\top \in \mathbb{J}^C$, or a surrogate loss such as the

softmax cross-entropy loss,3:

$$\ell_{\text{CE}}(\mathbf{w}, z') := -\log \frac{\exp(w_{z'})}{\sum_{z \in [C]} \exp(w_z)}. \quad (3)$$

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 $J(f)$ instead to make the problem more tractable. Furthermore, since $p(p, q)$ is usually unknown and there is no way to exactly evaluate $J(f)$, we minimize its empirical version calculated from the training data instead4:

$$\hat{J}(f) := \frac{1}{m} \sum_{i=1}^m \ell(f(p_i), q_i) \quad (4)$$

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

Definition1. The flooded empirical risk is defined as 4

$$\tilde{J}(f) = |\hat{J}(f) - b| + b \quad (5)$$

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

207 4. Results and Discussion

208 In this experiment, we first preprocess the image, and the PyTorch framework is used to scale the image to 448×448
 209 for data standardization. In this paper, the Mobilenet v3 network model is selected as well as optimizer ASGD, the
 210 learning rate is set to 0.001, the L1 regularity coefficients are all 0.01, the batch size is 12, and 300 rounds of iterative
 211 training are conducted on the training set and the test set respectively. In order to prevent over-fitting phenomenon in
 212 the experiment, we also apply the algorithm of Dropout to randomly deactivate the neural nodes in the network before
 213 network training, reduce the interdependence between neurons, so as to ensure the extraction of important features that
 214 are independent of each other and improve the generalization ability of the model. As shown in the figure, it can be
 215 clearly seen that neurons randomly deactivate seven neural nodes in the network.

216 **4.1. Contrast test**

217 On the same cucumber pathological leaf image data set, we select seven current mainstream network models and
218 MobileNet v3 network for comparative test. and accurately reflect the advantages of MobileNet v3 network from the
219 experimental data.

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

226 The loss function and accuracy of the training set of Alexnet model converge well. When epoch=100, they begin
227 to converge and gradually become stable. However, the effect on the test set is not good, the fluctuation of loss and
228 accuracy is large, and the convergence effect is not good. Compared with other models, its loss in the test set is highest,
229 its accuracy is lowest, and its performance is poor.

230 The training set of the VGG model converges quickly, and the loss image of the data set begins to converge in the
231 70th round of iteration. The accuracy image of the training set and the test set converge faster and it has converged in
232 about 30 iteration rounds. However, the degree of fitting in the test set is not high, and the test set loss and accuracy
233 image of the VGG model show no convergence trend. The average accuracy has not reached 70%.

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

238 The results of Efficientnet v2 and Efficientnet v7 models are basically the same. The loss and accuracy of the
239 test set tend to converge, and the fitting degree is also higher compared with the training set. However, the degree of
240 over-fitting of the loss image of the test set is too high and fluctuates wildly.

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

244 In the MobileNet v2 model, the training set starts to converge from the 30th iteration, and the loss finally keeps
245 approaching 0, and the accuracy also keeps increasing with the training rounds. The result trend of the test set also
246 roughly fits the training set, but there is an over-fitting phenomenon with low generalization and large fluctuation.

247 Considering the fitting effect of each model test set comprehensively, the test set result trend of Mobilenet v3

248 network is consistent with the training set trend, and its maximum accuracy is relatively the highest, reaching 81.3% and
249 above. Moreover, Mobilenet v3 converges 66% faster than Resnet network and much faster than any other network due
250 to its lightweight framework. Therefore, Mobilenet v3 network is finnaly chosen for the next optimization experiment.

251 4.2. The choice of optimizer

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

261 As shown in the figure 5, although the four optimizers, RAdam, NAdam, Adam, and RMSprop, all have a conver-
262 gence trend at last, their loss value in the training set are very high, and their highest accuracy is not more than 60%.
263 Compared with other optimizers, the effect of those four optimizers is poor. RAdam, NAdam, Adam, and RMSprop
264 are unsuitable for this paper's network model.

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

268 Adamax optimizer begins to converge after 100 iteration rounds of the training set, and its loss function image
269 value in the test set is higher than that of the ten optimizers, its accuracy is low, with an average accuracy of less than
270 65%.

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

276 The data set images of the two optimizers, ASGD and SGD, almost coincide, converge in both the training set
277 and the test set, and the convergence is faster. The training set begins to converge in the 80th iteration round, and the
278 test set begins to converge in the 20th iteration. And the peak value of test set accuracy is the highest, up to 81.4%.

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

279 Numerically, the ASGD has less fluctuation than the SGD optimizer.

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

283 4.3. The choice of batch size

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

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

291 It can be seen from the figure6 that the convergence speed increases with the increase of batch size. According to
 292 the numerical results in 2, with the increase of batch size, the maximum loss function, average loss function, and the
 293 standard deviation of the loss function, namely the volatility, of the test set gradually decrease. However, after batch
 294 *size* = 12, each loss value changes little with the increase in batch size. In addition, the maximum accuracy after
 295 convergence increases weakly and sometimes even regresses. Moreover, after batch *size* = 12, the standard deviation
 296 of the accuracy of the test set began to rise continuously, indicating that the model's generalization ability declined.

297 Before batch *size* = 12, the test set's accuracy increases while the loss fluctuation decreases. When batch *size* = 12,
 298 the standard deviation of loss is minimum, and the anti-aliasing effect is best. At the same time, the accuracy of the
 299 test set increased to 82.1%.

300 The experiment in this paper is carried out under a blurred background image, so we need as much generalization
 301 ability as possible. Moreover, the model proposed in this paper should apply to mobile terminal devices, should be as
 302 lightweight as possible, and need to select the smallest batch size value possible. Therefore, from the perspective of

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

³⁰³ background requirements and image data analysis, batch size = 12 is selected as the optimal experimental parameter
³⁰⁴ in this paper.

³⁰⁵ 4.4. Flooding

³⁰⁶ In Chapter 3, we introduce the basic principle and used a flooding mode. that is, by changing the loss function
³⁰⁷ and adding a threshold, the loss eventually fluctuated around the threshold. Flooding allows us to directly select the
³⁰⁸ level of training loss, which is difficult to achieve with other regularizers. There is an over-fitting phenomenon in the
³⁰⁹ loss result in images of the Mobilenet v3 experiment mentioned above. In this section, flooding is used to realize the
³¹⁰ secondary decrease of data set loss and prevent over-fitting.

³¹¹ In this experiment, the optimizer uses ASGD, the learning rate is 0.001, the regularization coefficients of L1 and
³¹² L2 are 0.01, and 300 iteration experiments were conducted. The loss threshold is set with 15 different values for
³¹³ comparative analysis of images and data.

³¹⁴ As seen from the image,⁷ after flooding is added, the over-fitting rising trend of the loss function image of the test
³¹⁵ set is effectively suppressed. When $b = 0.310, 0.348$, and 0.290 , the flooding not only resulted in good inhibition but
³¹⁶ also resulted in secondary descending, which solved the over-fitting problem.

³¹⁷ According to a series of comparisons of the table data, after adding flooding to 3, the mean test set accuracy
³¹⁸ increased by 0.2%, and the maximum test set accuracy increased by 0.5%. The final goal of this paper is to select the
³¹⁹ test set with the highest accuracy to achieve the best pathological recognition effect of the cucumber leaf image. The
³²⁰ final selection threshold is 0.310, at which time the over-fitting of the loss function is well suppressed, and the accuracy
³²¹ is up to 83.3%.

³²² We compare the two experiments without using flooding and using flooding method. The results are shown in

323 the figure 8. The method with flooding often improves the test accuracy than the baseline method without flooding.
324 Continuing to train the model without flooding, the loss may rise and accuracy may decline. However, according to
325 the results which used flooding, the model has good predictive performance. The results means that flooding helps
326 improve test accuracy in the later training. During training with flooding, test losses became lower and flatter. On
327 the other hand, the training loss reachse a secondary decline and continues to float around the flooding threshold with
328 stability.

329 **4.5. Discussion**

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

335 As can be seen from the image results,⁹ the loss function of the training set and the test set is constantly close to
336 0, and the accuracy also increases with the increase of iteration rounds. The accuracy of PlantVillage public data set
337 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
338 as 98.1%, which is far higher than the 76.5% accuracy of Zhou Minmin's apple-leaf-disease-detection-system based
339 on transfer learning.(Minmin, 2019) It is proved that compared with the existing strategies, the proposed strategies are
340 universal, accurate, and less time-consuming and can better meet the needs of Chinese farmers for crop pathological
341 judgment in today's society.

342 **5. Conclusion**

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

347 To solve these problems, a flood based Mobilenet v3 is proposed in this paper to identify crop leaves in fuzzy
348 scenes. It satisfied the requirement of mobile terminal using a lightweight framework and could quickly and accurately
349 judge crop pathological conditions through farmers' shooting pictures. The network solved most important issue about
350 identify cucumber diseases from leaf images in natural scenes which enable QA system for famers can be realized.

351 In this paper, cucumber leaf images are randomly collected from Chinese agricultural websites and labeled. A data
352 set with complex image background is constructed, and seven kinds of cucumber leaf pathology judgments are made.

353 Through the control variable method, the network model, the optimizer, and batch size, three rounds of experiments
354 are compared and analyzed to achieve the optimal network model. In this paper, flooding method is used to replace
355 an evaluation strategy of loss. The accuracy of the test set is increased by 0.5% again, reaching the highest 83.3%.
356 Finally, two public data sets of Plant-Village and apple disease are selected for the experiment again. The accuracy is
357 up to 99% and 98.1%, respectively, which proved the universality of the proposed strategy and its high practical value.

358 References

- 359 Benfenati, A., Causin, P., Oberti, R., Stefanello, G., 2021. Unsupervised deep learning techniques for powdery mildew recognition based on
360 multispectral imaging. arXiv preprint arXiv:2112.11242 .
- 361 Cap, Q.H., Uga, H., Kagiwada, S., Iyatomi, H., 2020. Leafgan: An effective data augmentation method for practical plant disease diagnosis. IEEE
362 Transactions on Automation Science and Engineering .
- 363 DeChant, C., Wiesner-Hanks, T., Chen, S., Stewart, E.L., Yosinski, J., Gore, M.A., Nelson, R.J., Lipson, H., 2017. Automated identification of
364 northern leaf blight-infected maize plants from field imagery using deep learning. Phytopathology 107, 1426–1432.
- 365 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
366 mobilenetv3, in: Proceedings of the IEEE/CVF international conference on computer vision, pp. 1314–1324.
- 367 Hughes, D., Salathé, M., et al., 2015. An open access repository of images on plant health to enable the development of mobile disease diagnostics.
368 arXiv preprint arXiv:1511.08060 .
- 369 Ishida, T., Yamane, I., Sakai, T., Niu, G., Sugiyama, M., 2020. Do we need zero training loss after achieving zero training error? arXiv preprint
370 arXiv:2002.08709 .
- 371 Karthik, R., Hariharan, M., Anand, S., Mathikshara, P., Johnson, A., Menaka, R., 2020. Attention embedded residual cnn for disease detection in
372 tomato leaves. Applied Soft Computing 86, 105933.
- 373 Kawasaki, Y., Uga, H., Kagiwada, S., Iyatomi, H., 2015. Basic study of automated diagnosis of viral plant diseases using convolutional neural
374 networks, in: International symposium on visual computing, Springer. pp. 638–645.
- 375 Martinelli, F., Scalenghe, R., Davino, S., Panno, S., Scuderi, G., Ruisi, P., Villa, P., Stroppiana, D., Boschetti, M., Goulart, L.R., et al., 2015.
376 Advanced methods of plant disease detection. a review. Agronomy for Sustainable Development 35, 1–25.
- 377 Minmin, Z., 2019. Research on Android Detection System of Apple Leaf Disease Based on Transfer Learning. Master's thesis. Northwest A&F
378 University.
- 379 Prasanna Mohanty, S., Hughes, D., Salathe, M., 2016. Using deep learning for image-based plant disease detection. arXiv e-prints , arXiv–1604.
- 380 Saikawa, T., Cap, Q.H., Kagiwada, S., Uga, H., Iyatomi, H., 2019. Aop: An anti-overfitting pretreatment for practical image-based plant diagnosis,
381 in: 2019 IEEE International Conference on Big Data (Big Data), IEEE. pp. 5177–5182.
- 382 Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., Stefanovic, D., 2016. Deep neural networks based recognition of plant diseases by leaf
383 image classification. Computational intelligence and neuroscience 2016.
- 384 Wspanialy, P., Moussa, M., 2020. A detection and severity estimation system for generic diseases of tomato greenhouse plants. Computers and
385 Electronics in Agriculture 178, 105701.
- 386 Zhang, Y., Song, C., Zhang, D., 2020. Deep learning-based object detection improvement for tomato disease. IEEE Access 8, 56607–56614.
- 387 Zhonghua, Y., Mingxia, Z., Lu, J., 2021. Research on crop disease image recognition with complex background. Transactions of the Chinese
388 Society for Agricultural Machinery .

389 **List of Figures**

390 1	Cucumber complex background data set presentation	16
391 2	Article algorithm	17
392 3	A comparison of the neural network before and after using dropout	18
393 4	Comparison Experiment between mobilenet v3 and Mainstream Image Classification Algorithms	19
394 5	Comparison Experiment between ASGD and Mainstream Optimizer Experiment	20
395 6	Comparative Experiments with Different batch Sizes	21
396 7	Comparative Experiment on Different Values of Parameter b After Using flooding	22
397 8	Comparative Experiment Before and After Flooding	23
398 9	Effect experiment of mobile v3 based on flooding on different data sets	24



Figure 1: Cucumber complex background data set presentation

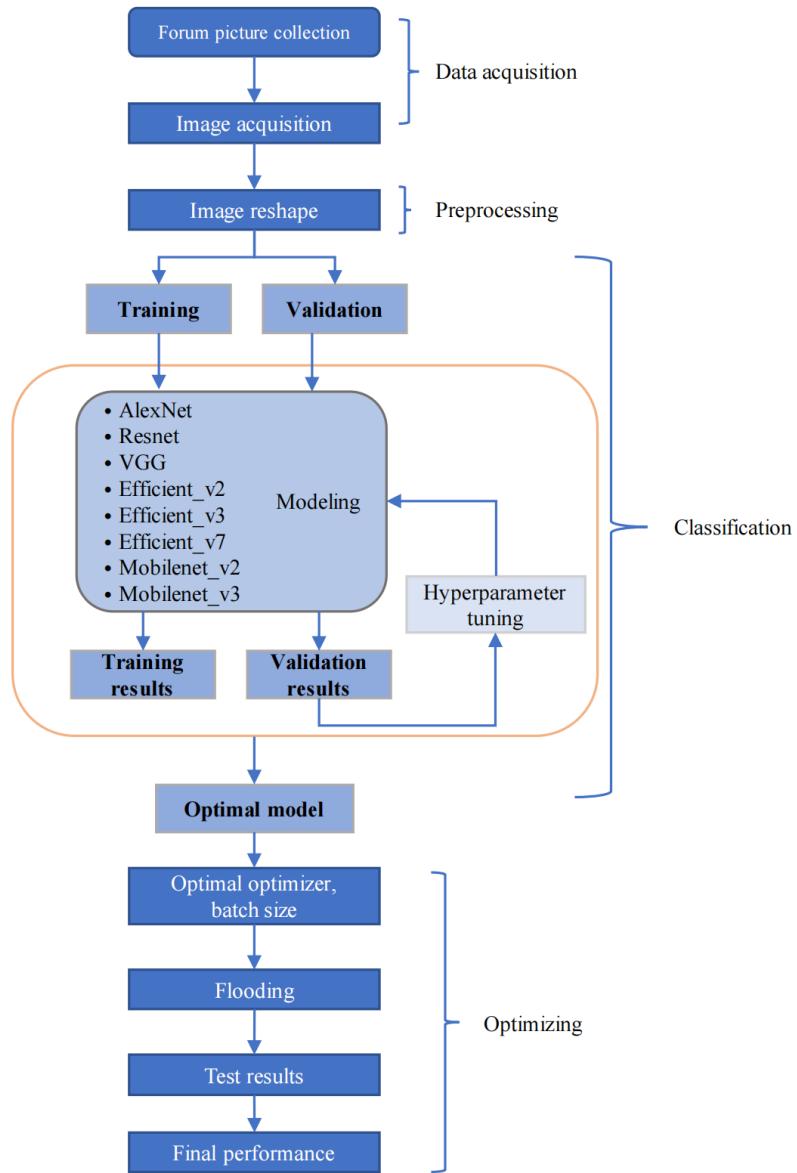
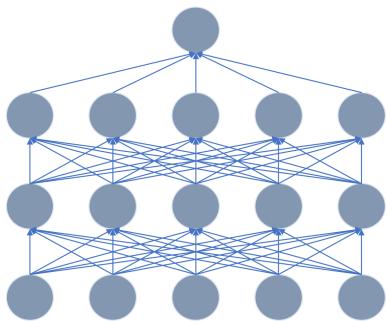
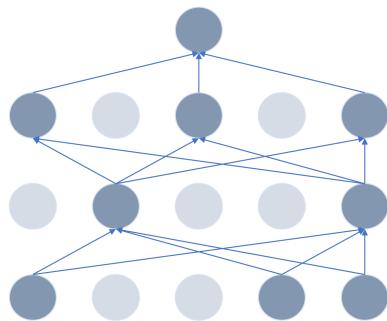


Figure 2: Article algorithm



(a) Standard Neural Net



(b) After applying dropout

Figure 3: A comparison of the neural network before and after using dropout

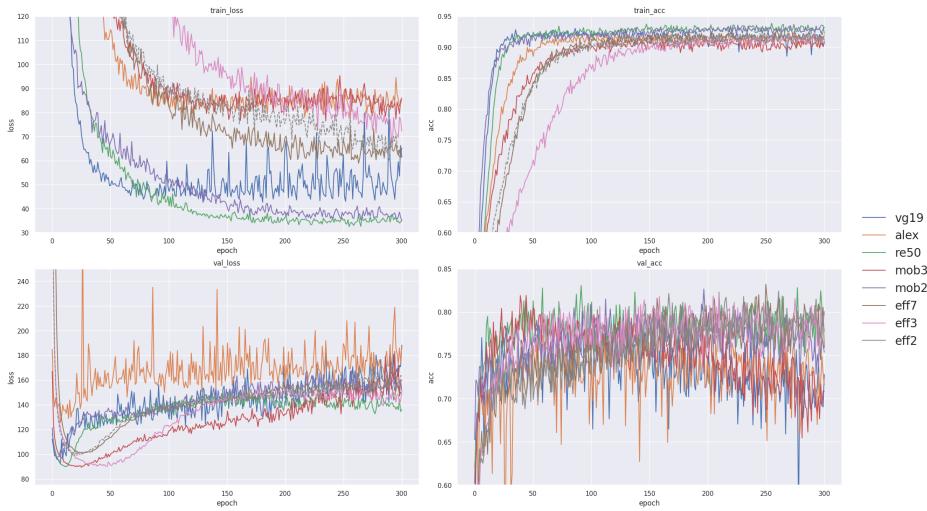


Figure 4: Comparison Experiment between mobilenet v3 and Mainstream Image Classification Algorithms

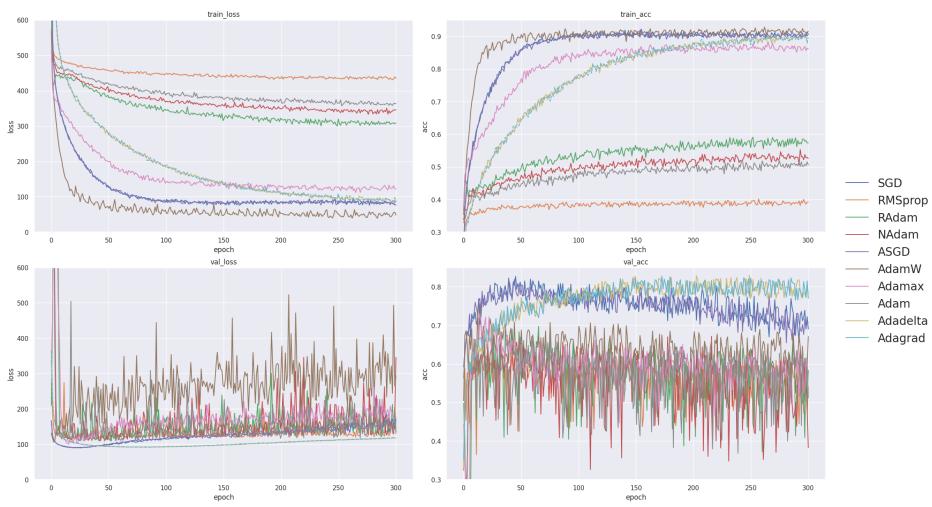


Figure 5: Comparison Experiment between ASGD and Mainstream Optimizer Experiment

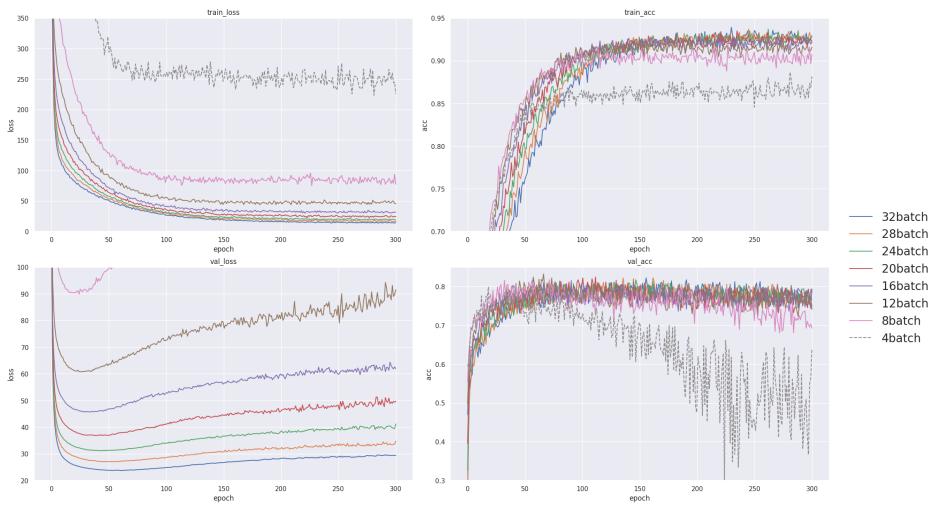


Figure 6: Comparative Experiments with Different batch Sizes

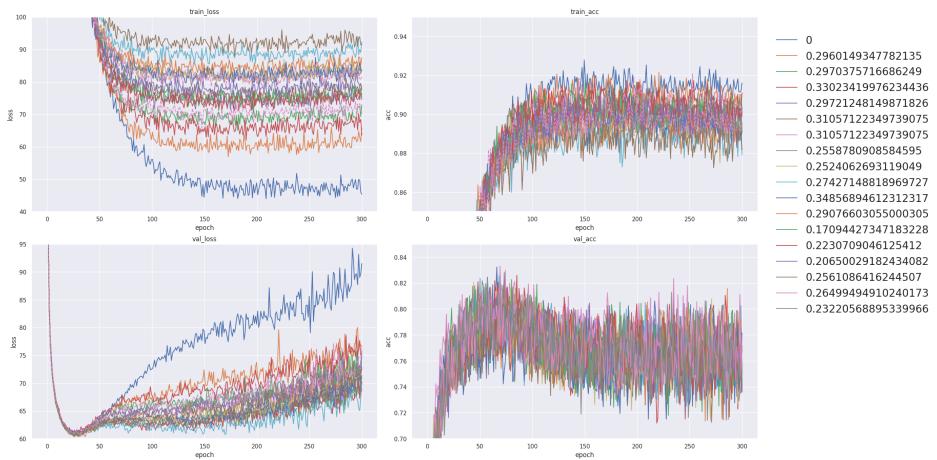


Figure 7: Comparative Experiment on Different Values of Parameter b After Using flooding

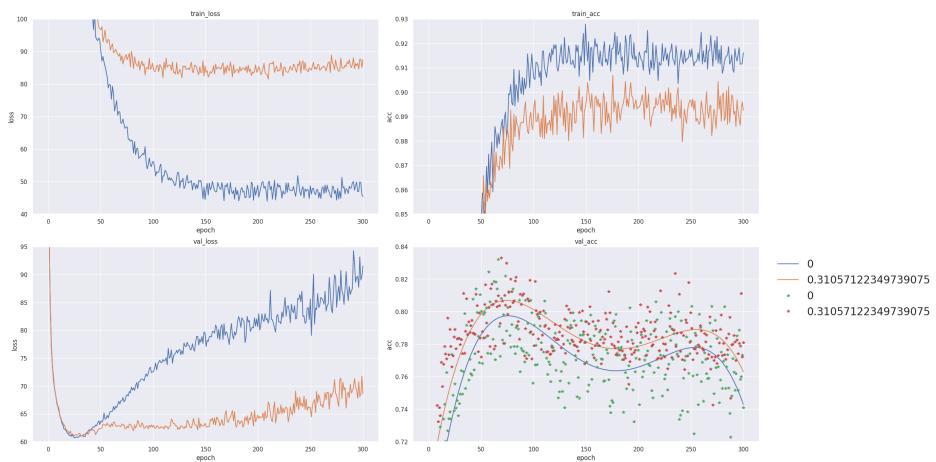


Figure 8: Comparative Experiment Before and After Flooding

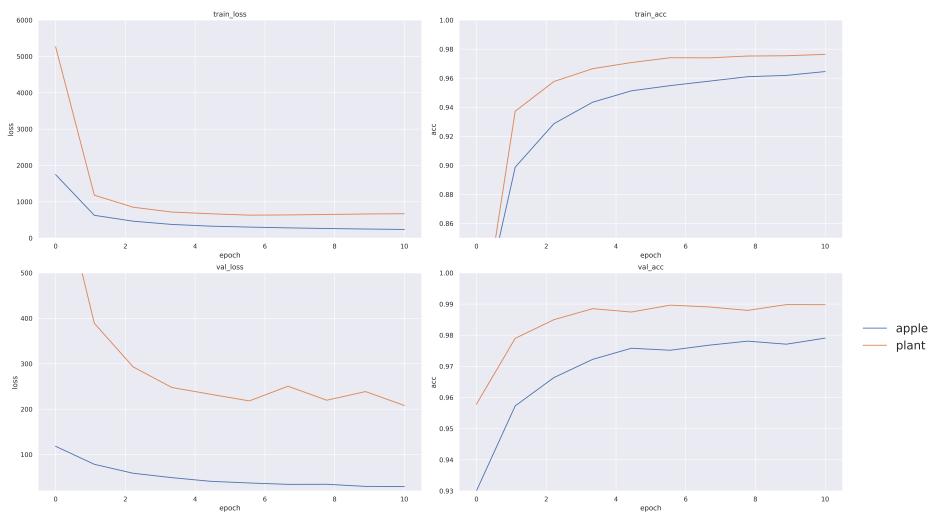


Figure 9: Effect experiment of mobile v3 based on flooding on different data sets