Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/5/20230651

A lightweight plant disease recognition network based on ResNet

Yunlong Duan[1, †], Ziyu Han[2, †] and Zhening Tang[3, 4, †]

[1] Guangzhou University Sontan College, Guangzhou, 510275, China
[2] Jiangxi University of Finance and Economics, Nanchang, 330013, China
[3] Shanghai University of Finance and Economics, Shanghai, 200433, China

[4] tangzhening@163.sufe.edu.cn
[†] These authors contributed equally.

Abstract. Identification of foliar diseases is very important for the cultivation of plants. If no diseases are found, the cultivation results may decline, resulting in serious losses of related industries. Most of the early automatic recognition methods of plant leaves are based on manual features and classifiers, and the recognition performance is often unable to meet the actual complex application scenarios. Thanks to the rapid development of convolutional neural networks, such as ResNet, the accuracy of plant disease identification based on deep learning has made a breakthrough. However, convolutive neural network tends to have too many parameters, large amount of calculation and slow training speed, which is difficult to be used in various small and medium-sized plant cultivation industries, especially in small edge computing devices deployed in the field. This paper designs a new lightweight Resnet network structure, namely Resnet-9. The number of network layers in traditional Resnet is reduced. Compared with other commonly used plant disease recognition methods, the accuracy of Resnet is guaranteed and the network is more lightweight. The parameter of this model occupies only 6.6M memory and achieves 99.23% accuracy on public datasets. Even in the other data sets, the accuracy was still 95.15%. The effectiveness of the method is verified by comparative experiment.

Keywords: plant leaf diseases; deep learning; ResNet; lightweight

1. Introduction

In agricultural production, illnesses have a significant impact on crop productivity and quality. Therefore, how to rapidly and properly spot crop illnesses is an important means to improve crop yield and promote agricultural modernization. A growing number of researchers have started using machine learning to identify agricultural illnesses in recent years. The denoising, segmentation, image enhancement, feature extraction, and feature classification and processing are the foundations of the conventional crop disease image detection system. Although the traditional crop disease recognition method can accurately identify the disease, the image processing process is cumbersome, and the accumulated errors between steps will have a certain impact on the results [1].

Convolutional neural networks are able to input photos as data without the need for any sophisticated processes like feature extraction or image processing, as opposed to more conventional

DOI: 10.54254/2755-2721/5/20230651

image classification techniques. In addition, neural networks are used to learn the features of training samples independently, which can not only learn the low-level features of images, but also learn high-level abstract semantic features. However, the accuracy of the model will decline as the network layer is deepened. The gradient descent optimization approach performs worse as the model gets more complicated, making it impossible for the model to have a satisfactory learning effect and causing the issue of learning deterioration. He [2] et al. proposed residual network (ResNet) on the basis of VGG network to solve the problem of learning degradation. Residual network is to add residual learning to traditional neural network. Compared with the ordinary network, which adds a short circuit mechanism between every two layers, ResNet directly skips the data of one layer in the previous layers to the input part of the data layer in the later layer, which forms the residual learning and avoids the phenomenon of learning degradation with the deepening of the network model layer. At present, the fields of object identification and picture classification make extensive use of residual networks, and it is an important part of neural networks in computer vision tasks. Typical residual networks include ResNet-18 and ResNet-50.

In the field of crop disease classification, many researchers use ResNet network or improved ResNet network. Brahmaji [3] et al., used the ResNet-152 v2 model to recognize more than 7000 images of plant leaf diseases, and the accuracy of the model in predicting plant leaf diseases was 95%. Vinod [4] et al., used ResNet-34 network to identify and classify 38 types of plant diseases, and the model identification accuracy on average was 95.48%. On this basis, the authors compare the performance indicators of ResNet-34 with SVM, k-NN, decision tree and logistic regression. The experimental results show the superiority of ResNet-34 network in plant disease classification tasks. Xin [5] et al., respectively used VGG-16, ResNet-34 and ResNet-18 algorithms to classify apple leaves and identify normal and diseased leaves. Through experimental comparison, it is found that the recognition effect of ResNet algorithm is better. The ResNet-18 network has fewer network layers, and it superior to ResNet-34 in recognition accuracy and loss rate. In order to better identify grape downy mildew field photos, He [6] et al. proposed an enhanced residual network model.

To further enhance the network feature extraction capability, the model alters the main branch structure and shortcut branch structure of the model residual block based on ResNet-50. The average recognition accuracy on the dataset with various enhancement methods can reach 99.92 %. Li [7] et al. proposed an Asymmetric Convolution Attention ResNet (Asymmetric Convolution Attention ResNet, ACA-ResNet). On the basis of residual network, the model introduces asymmetric convolution structure and attention mechanism, and the average recognition accuracy of corn disease can reach 97.25%, which is significantly improved compared with the training speed of the original Resne-50. For the purpose of identifying photos of many citrus diseases, Jun Tie [8] et al., suggested the F-ResNet fusion model, which is an upgraded version of ResNet-34. Experimental results show that F-ResNet solves the problems of weak generalization ability and poor robustness in a single model, and the recognition accuracy of citrus disease images in natural environments reaches 93.6 %. The Multi-scale-SE-ResNet-18 model was enhanced by Huang Linsheng [9] et al., to recognize crop disease photos in difficult field situations. This model improves the capability of feature extraction and increases the resilience of the model by incorporating the Inception module and adding attention mechanism on the basis of ResNet-18. The average success rate for diagnosing complex field crop diseases is 95.62%. Compared with the original ResNet-18 model, the accuracy is increased by 10.92 percentage points.

The aforementioned study produced positive crop disease detection findings, and ResNet network accuracy was kept at a high level. In the classic ResNet network, the shallow network structure is ResNet-18 and ResNet-34. Despite being the lightest ResNet-18 network, it still has too many parameters, too much computation, and too long running time. This certainly increases the need for hardware facilities. The utilization of sophisticated network architectures has demanding requirements on hardware and demands considerable memory overhead, which is challenging to install on embedded systems in the majority of small-scale plant culture and planting enterprises [10]. High

DOI: 10.54254/2755-2721/5/20230651

hardware requirements and training of human recognition experts will lead to high cost and tedious recognition process.

However, the research of Vita et al. showed that the paths in ResNet actually have collective behavior and there are many independent and effective paths [11]. It is considered that fewer layers in the traditional ResNet network to make the structure more lightweight. However, when most networks are too shallow, the accuracy can greatly reduced. Therefore, this paper takes ResNet as the reference model, constructs a nine-layer network and modifies part of the structure to design a new lightweight network model ResNet-9 which is used to identify and detect different diseases of different kinds of plant leaves and ensures the accuracy of the results while carrying out lightweight.

The rest of this article is structured as following. Chapter 2 explains the structure of the ResNet-9 network. Chapter 3 carries out a series of comparative experiments. ResNet-9 was applied to PlantVillage dataset and mainstream plant disease recognition network for precision comparison experiment. Then it is applied to the PlantifyDr dataset to compare the accuracy with ResNet of different layers. Finally, it is compared with the classical ResNet network in terms of Parameters and FLOPs to measure the lightweight degree of the model. Then Chapter 4 will summarize the work of this paper and discusses the future work.

2. Methods

In this section, we describe the designed models for object completion. Given pictures of plant leaves, our goal is to classify the type of this plant and judge its health condition as well. The output will show users specific vegetative diseases if the plant is in poor health. This approach is modeled after the ResNet baseline, and we modify the architecture in order to make our method more lightweight.

2.1. Classical structure of Resnet

He et al. [2] proposed the Residual neural network. As the core concept of ResNet, residual blocks are divided into two main types of structure, which can be seen in Figure 1. The right one is a bottleneck block architecture, whose architecture consists of two 1×1 conv layers and a 3×3 conv layer. The 3×3 conv layer is between those two 1×1 conv layers. In bottleneck block architecture, the 1×1 conv layers are used to descend dimension and then raise dimension again. On the other hand, the left one, called basic block, uses two 3×3 conv layers instead. There's no doubt that this architecture seems more simple and practical.

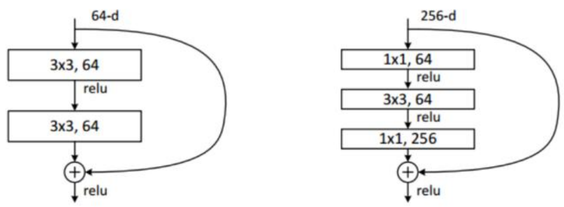

Figure 1. Basic block (left) and bottleneck block (right).

2.2. Our Residual Block

Based on the original network model of ResNet, we design our unique residual block, to retain the characteristics of ResNet and to be more consistent with the overall network structure design as well. The specific architecture is showed in Figure 2.

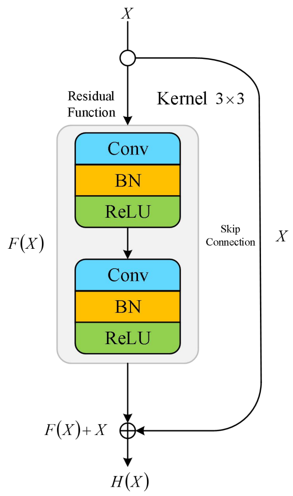

Figure 2. Residual block of our ResNet network.

Because we want the model to reduce unnecessary computation as much as possible, the overall network layer we design is relatively shallow. Hence, there is no need to choose a bottleneck architecture to reduce and increase the dimension of features. So we choose the basic block, whose main body is two 3×3 convolution layers. After each convolution layer, we will first normalize the feature information through the Batch Normalization layer. BN layer not only helps us improve the training speed, but also effectively avoids the problems of gradient disappearance and overfitting. As we use Batch Normaliztion, the need for Dropout can be eliminated and a much higher learning rate is guaranteed as well [12]. Then, we will use the activation function of ReLU layer (rectified linear units) to provide nonlinear learning ability for neural networks [13]. At the same time, ReLU function is very simple, which can save a lot of computation for the whole neural network. As for our residual block, we also set the shortcut path to ensure the identity mapping except for the path F(x). In this way, even if our module has no gradient, it can pass through this shortcut path directly, which avoids the problem of information loss in common convolution layer. It also makes sure that forward information flow and reverse gradient flow are both smooth.

2.3. The Overall Model

Figure 3 shows the architecture of our nueral network which contains residual blocks as a guarentee of accuracy. Given a picture of plant leaves in size of 256×256 pixels, the main features are extracted by convolution layers and residual blocks efficiently. Then BN layers and ReLU layers help our network to be more accurate and lightweight. Processed by max pooling layers, the number of parameters and dimensions to be trained are reduced sharply, which statisfies our requirement for computation reduction and speed enhancement. And we designed fully connected layers as well.

DOI: 10.54254/2755-2721/5/20230651

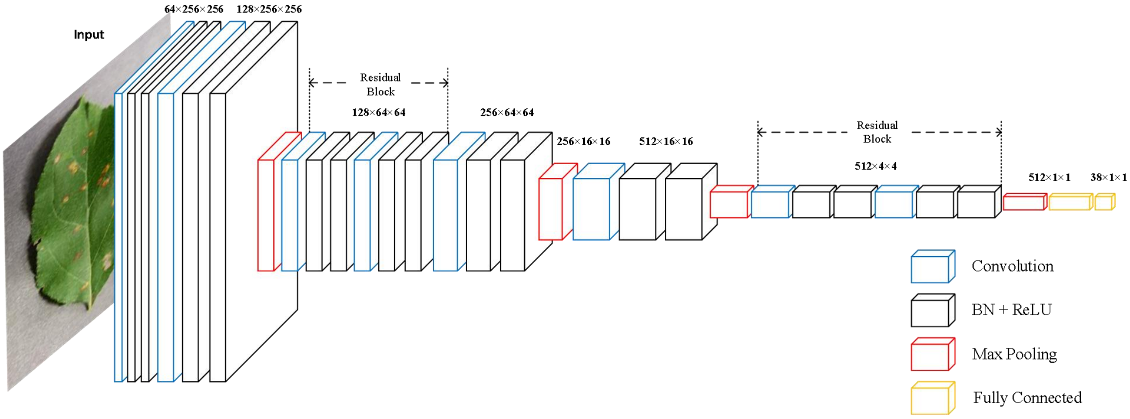

Figure 3. Neural Network Architecture of ResNet-9.

Figure 4 shows our overall model. First, we have relatively common convolution modules, which is only composed of convolution layers, BN layers and ReLU layers. Second, we added a max pooling layers to reduce the dimensions. As shown in the figure, we designed a basic convolution module and three convolution modules with pooling layers. In the middle, we interspersed the residual blocks twice to help our model learn the feature information. Finally, we move through the pooling layer to Flatten layer, which converts the multidimensional input to only one dimension. In the end, information enters Linear, the fully connected layer, so that it can better map features to the classification space of the specified dimension.

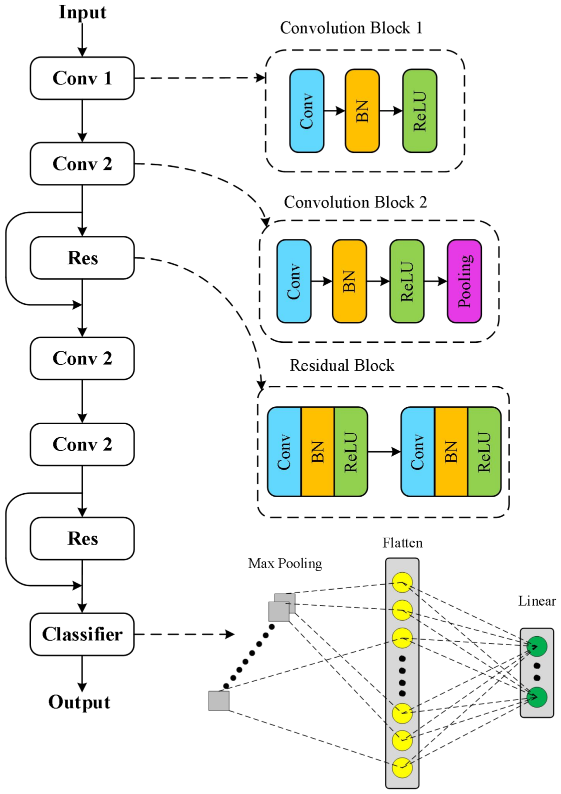

Figure 4. Details of the ResNet-9 neural network architecture

.

2.4. Details

In this part, we will show some relevant details in our design, including the main working principles. First off, we'd like to introduce the related formula of our residual blocks. For forward propagation, we have:

$$a^{l_2} = a^{l_2-1} + F\left(a^{l_2-1}\right) = \left(a^{l_2-2} + F\left(a^{l_2-2}\right)\right) + F\left(a^{l_2-1}\right) = \dots = a^{l_1} + \sum_{i=l_1}^{l_2-1} F\left(a^{i}\right) \quad (1)$$

For backward propagation, we have:

$$\frac{\partial \epsilon}{\partial a^{l_1}} = \frac{\partial \epsilon}{\partial a^{l_2}} \left(1 + \frac{\partial \sum_{i=l_1}^{l_2-1} F(a^i)}{a^{l_1}}\right) \quad (2)$$

For Batch Normalization, we have:

$$\begin{gathered} \text{batch mean} : \mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \\ \text{batch variance} : \sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2 \\ \text{normalize} : \hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \\ \text{scale and shift} : y_i \leftarrow \hat{x}_i + \beta \equiv BN_{\gamma,\beta}(x_i) \end{gathered} \quad (3)$$

And then, our training parameters are listed. We set batch size to 32, max_lr to 0.01, grad_clip to 0.1, and weight_decay to 1e-4. Finally, the cost function we use is cross-entropy, which ensures the speed of gradient updating. For cross-entropy, we have:

$$H(A,B) = -\sum_i P_A(x_i) \log\left(P_B(x_i)\right) \quad (4)$$

Since correctly labeled data can be differentiated from wrongly labeled data in our dataset and the data to be traind is large-scale, cross-entropy would work well [14].

3. Experiments

3.1. Datasets and evaluation metrics

To test the performance of our neural network architecture, we carried out several experiments on two datasets of PlantVillage Dataset [15] and PlantifyDr Dataset [16]. PlantVillage Dataset contains about 87K images, which are categorized into 38 different types. This dataset divides healthy plant leaves apart from diseased ones well. And it is furtherly divided as two parts, valid set and training set. The ratio of these two sets is about one to four, which helps preserve the directory structure and determine if the model is overfitting. PlantVillage Dataset is mainly used to compare our network with others in the aspect of identification accuracy. Since this dataset is famous in related fields, there has been a lot of other people's work to learn from and plenty of data is available. However, we found that the accuracy of our training in PlantVillage Dataset was too high to support more relevant experiments. So we chose a new dataset for further experiments. As for PlantifyDr Dataset, we made some modifications about the ratio of training set and valid set. After adjusting the ratio to ten to one, this dataset contains over 125K images of 10 different plant types,which is composed of 37 different types. Because PlantifyDr dataset has not been wisely used in other people's work, we used it to verify the performance and speed of our neural network compared with more layers of ResNets. In addition, we cited Parameters and FLOPs(Floating-point operations per second) as our evaluation metrics, so that the overall performance can be assessed objectedly.

FLOPs are used to measure the number of operations of the model, which characterizes the time complexity of the model. To compute FLOPs, we suppose that nonlinear functions are computationally free. For convolutional kernals we have:

$$FLOPs = 2HW\left(C_{in}K^2 + 1\right)C_{out} \tag{5}$$

where H is height, W is width, K is the kernal width, C_{in} and C_{out} correspond to the number of channels of the input and output feature maps, respectively. For fully connected layers we have:

$$FLOPs = (2I - 1)O \tag{6}$$

where I and O correspond to the input and output dimension, respectively [17].

3.2. Accuracy comparison with other mainstream networks

In this part of our experiments, we used PlantVillage Dataset to make an accuracy comparison with other mainstream networks focused on image classification. As a matter of fact, plant disease classification can be done perfectly on training dataset by several conventional neural networks. But when it comes to validation or test, the accuracy would drop drastically. To avoid such a sharp fall in accuracy, our network used ResNet to tackle the vanishing gradient problem and get rid of overfitting. So the training error percentage is decreased and the accuracy of our network is quite good. The concrete data is as follows.

Table 1. Accuracy of different methods on classifying plant diseases

Method	Datasets	Selected Plants	Accuracy
VGG-Inception[18]	Real Environment	Apple	78.80%
DCNN[19]	Real Environment	Wheat	85.12%
LeNet[20]	PlantVillage	Banana	98.61%
GoogleNet[21]	PlantVillage	Tomato	99.18%
MobileNet[22]	PlantVillage	24 types of plants	98.34%
Ours	PlantVillage	14 types of plants	99.23%

In Table 1, we cite many experimental results of others for comparison. As we can see, VGG-Inception and DCNN worked not quite well because they were trained by real envirnment. So both methods have terrible accuracy and are specific to only one plant. When it comes to methods trained by PlantVillage Dataset, there has been a noticeable improvement in overall accuracy. Among these four networks, ours has the first accuracy and the second training range, showing an overwhelming advantge over the other three. Consequently, the accuracy of our network is guaranteed to be within a relatively high range, which is also the reason why we chose ResNet as the main body of our network.

3.3. Accuracy and time comparison of ResNets with different layers

Since the result trained by PlantVillage Dataset is much better than expected, we used PlantifyDr Dataset to reduce the accuracy, which is beneficial for us to make further experiments. After making sure that our network has an accuracy advantage over other networks, we hope to perform experiments on ResNets with different number of layers to prove that our architecture has both high accuracy and lightweight features. So we did this experiment focusing on accuracy and running time.

Table 2. Accuracy and running time of ResNets with different layers

Layers	Accuracy			Time
	Epoch 3	Epoch 4	Epoch 5	
5	88.42%	90.28%	90.65%	0.82 t
9 (Ours)	90.67%	93.72%	95.15%	t
18	92.41%	93.89%	95.86%	2.11 t
34	93.54%	94.66%	96.78%	4.37 t

From the table above, we proved that our architecture was superb in both aspects of accuracy and lightweight. Compared with the 5-layer structure, our design only increases the running time by about 22%, and has a relatively significant improvement in accuracy. Moreover, the 5-layer structure enters the bottleneck of accuracy when it reaches close to 91%, and the room for improvement in subsequent training is quite small. However, our architecture with 9-layer avoids this problem and provides a steady improvement in accuracy per epoch. When it comes to more layers such as 18 or 34, the experiment data shows that the accuracy admittedly increases by about one percent, but the cost of running time is bound to increase substantially. Therfore, considering both accuracy and lightweight, our architecture with 9-layer is undoubtedly a wiser choice.

3.4. Comparison of FLOPs and Parameters for ResNets with different layers

As the running time metioned above is depended on computing machine a lot, we assume that it is not a suitable evaluation metric. So we cite FLOPs(floating-point operations) and Parameters as our key evaluation metrics, which are much more rigorous in proving the lightness of our architecture.

In addition, we hoped to predict parameters and reduce them as much as possible. As the weights in all kinds of neural networks are mostly inclined to be structured, we could use specific techniques to decrease the number of free parameters [23]. With this technique, we decompose the weight matrix into two smaller matrices, which satisfies the requirement of smaller memory space and less computation time. If we need to get the weight matrix, we can take the product of smaller ones. By this way, the size of the parameterization can be controlled because we can get the rank of the original matrix. And then we make related experiments to test its feasibility.

Table 3. FLOPs and parameters of ResNets with different layers

Model	FLOPs	Parameters
ResNet-9 (Ours)	0.8×10^9	6.6 M
ResNet-18	1.8×10^9	11.4 M
ResNet-34	3.6×10^9	21.5 M
ResNet-50	3.8×10^9	23.9 M
ResNet-101	7.6×10^9	42.8 M
ResNet-152	11.3×10^9	58.5 M

In this experiment, we examined the FLOPs and parameters of our architecture, and then we made a simple comparison. First off, the data of FLOPs of ResNets with different layers was provided in the paper which puts forward ResNet [2]. And our model has a lower number of FLOPs since we reduce the layers, which means a lower computing complexity. Besides, the parameters of ResNets with different layers was also provided in other people's paper [24]. So we used it to make a further comparison with our model. As is shown in Table 3, the result of parameter size in our model is also lower than those with more layers.

In consequence, our architecture was proven to be a more lightweight model. So our model might be wisely used in a range of figure classification areas which require both high speed and relatively reliable accuracy.

4. Conclusion & Discussion

DOI: 10.54254/2755-2721/5/20230651

In order to ensure the accuracy of plant disease recognition and not be affected by hardware limitations, the deployment efficiency is improved. The nine-layer ResNet model proposed in this paper reduces the number of network layers of traditional ResNet and the optimized BN layer and ReLU layer are added to make the model more lightweight and ensure the accuracy of disease recognition. Then, the model is trained in two different datasets which is called PlantVillage and PlantifyDr Dataset. The feasibility of the model is proved by designing different comparative experiments.

In the future development, we will increase the research on related application fields, and further analyze the spot size of the classified plant diseases through neural networks, so as to obtain the severity of the disease and give corresponding treatment suggestions.

References

[1] Q. Huang and Q. Dong, "Image Classification of Crop Diseases Based on Convolutional Neural Network" *China Computer & Communication*, vol. 34, pp. 138–142+146, 2022.

[2] K. He, X. Zhang, S. Ren, et al., "Deep Residual Learning for Image Recognition," *IEEE Confer ence on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[3] B. Godi, A. S. Muttipati, M. P. Rao, et al., "ResNet Model to Forecast Plant Leaf Disease," In *2 022 International Conference on Computing, Communication and Power Technology (IC3P)*, 2022, pp. 38–43.

[4] V. Kumar, H. Arora and J. Sisodia, "ResNet-based approach for Detection and Classification of Plant Leaf Diseases," In *2020 International Conference on Electronics and Sustainable Com munication Systems (ICESC)*, 2020, pp. 495–502.

[5] X. Li and L. Rai, "Apple Leaf Disease Identification and Classification using ResNet Models," I n *2020 IEEE 3rd International Conference on Electronic Information and Communication T echnology (ICEICT)*, 2020, pp. 738–742.

[6] D. He, P. Wang, T. Niu, et al., "Classification Model of Grape Downy Mildew Disease Degree i n Field Based on Improved Residual Network," *Journal of Agricultural Machinery*, vol. 53, pp. 235–243, 2022.

[7] Q. Li, N. Miao, X. Zhang, et al., "Image recognition of maize disease based on asymmetric conv olutional attention residual network and transfer learning," *Science Technology and Engineer ing*, vol. 21, pp. 6249–6256, 2021.

[8] J. Tie, J. Luo, L. Zheng, et al., "Citrus disease recognition based on improved residual network," *Journal of South-Central University for Nationalities (Natural Science Edition)*, vol. 40, pp. 621–630, 2021.

[9] L. Huang, Y. Luo, X. Yang, et al., "Crop Disease Recognition Based on Attention Mechanism a nd Multi-scale Residual Network," *Journal of Agricultural Machinery*, vol.52, pp. 264–271, 2021.

[10] J. Chen, D. Zhang, A. Zeb, et al., "Identification of rice plant diseases using lightweight attentio n networks," *Expert Systems with Applications*, vol. 169, pp. 114514, 2021.

[11] A. Veit, M.J. Wilber, S. Belongie, "Residual Networks Behave Like Ensembles of Relatively Sh allow Networks," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[12] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reduci ng Internal Covariate Shift," In *International Conference on Machine Learning*, PMLR, 201 5, pp. 448–456.

[13] A.F. Agarap, "Deep Learning using Rectified Linear Units (ReLU)," *arXiv preprint arXiv:1803. 08375*, 2018.

[14] Z. Zhang, M. Sabuncu, "Generalized Cross Entropy Loss for Training Deep Neural Networks w ith Noisy Labels," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[15] github: Dataset of diseased plant leaf images and corresponding labels. [Online]. *Available: http s://github.com/spMohanty/PlantVillage-Dataset*

DOI: 10.54254/2755-2721/5/20230651

[16] kaggle: PlantifyDr Dataset. [Online]. *Available: https://www.kaggle.com/datasets/lavaman151/plantifydr-dataset*

[17] P. Molchanov, S. Tyree, T. Karras, et al., "Pruning Convolutional Neural Networks for Resource Efficient Inference," *In Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[18] P. Jiang, Y. Chen, B. Liu, et al., "Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks," *IEEE Access*, vol. 7, pp. 59069–59080, 2019.

[19] X. Zhang, L. Han, Y. Dong, et al., "A Deep Learning-Based Approach for Automated Yellow Rust Disease Detection from High-Resolution Hyperspectral UAV Images," *Remote Sens*, vol. 11, no. 13, pp. 1554, 2019.

[20] J. Amara, B. Bouaziz and A. Algergawy, "A Deep Learning-based Approach for Banana Leaf Diseases Classification," In *BTW (Workshops)*, 2017, pp. 79–88.

[21] M. Brahimi, K. Boukhalfa and A. Moussaoui, "Deep learning for tomato diseases: Classification and symptoms visualization," *Applied Artificial Intelligence*, vol. 31, no. 4, pp. 299–315, 2017.

[22] K.C. Kamal, Z. Yin, M. Wu, et al., "Depthwise separable convolution architectures for plant disease classification," *Computers and Electronics in Agriculture*, vol. 165, pp. 104948, 2019.

[23] M. Denil, B. Shakibi, L. Dinh, et al., "Predicting Parameters in Deep Learning," *In Conference and Workshop on Neural Information Processing Systems (NIPS)*, vol. 26, 2013.

[24] M.C. Leong, D.K. Prasad, Y.T. Lee, et al., "Semi-CNN architecture for effective spatio-temporal learning in action recognition," Applied Sciences, vol. 10, no.2, pp. 557, 2020.