Automatic Leaf Diseases Detection System Based on Multi-stage Recognition

Songyun Deng[1], Lekai Cheng[1], Wenlin Li[2], Wei Sun[1], Yaonan Wang[1], and Qiaokang Liang[1](✉)

[1] College of Electrical and Information Engineering, Hunan University, Changsha 410082, China
qiaokang@hnu.edu.cn

[2] DUT-RU International School of Information Science & Engineering, Dalian University of Technology, Dalian 116620, China

Abstract. The problem of plant diseases is an important issue affecting the growth of world food. However, the existing technology still has many defects in the automatic detection of plant diseases. These defects are mainly concentrated in three aspects: lack of dataset, no algorithm suitable for large range detection and lack of systems used on agricultural production. In this paper, we have made the following contributions to these shortcomings. First, we proposed a multi-stage system which could not only do the plant species classification but also do the disease classification at the same time. Besides, this approach could also reduce the dependence of the model on dataset to some extent. Second, an improved network proposed by us could perform fast calculations while maintaining accuracy. Third, we have realized the function of real-time leaf disease recognition on the embedded platform, which would provide ideas for the plant diseases detection of a wide range.

Keywords: Multi-stage recognition · SEI-ResNet · Embedded platform

1 Introduction

Plant diseases have always been a significant obstacle in agricultural production and gardening fields. In general, the traditional manual methods that people used to detect plant diseases are complicated and subjective [1]. Besides, due to the manual detection methods which require many steps and take a long time, it cannot be applied to the detection of plant diseases in a large range [2]. Today, advances in computer science and technology make it possible for machine vision recognition technology to replace naked eye identification to some extent [3]. It's the reason that plant diseases recognition is an important research task in machine vision field.

Supported by organization x.

Among the organs of plants, leaves are considered as one of the important organs for confirming plant categories [4]. Therefore, in recent years, there were a great number of researches on leaf diseases recognition based on machine vision technologies. These methods can be divided into two categories. The first one is the traditional image processing technique which mainly contains the pattern recognition method based on statistical method, pattern recognition method based on structure and fuzzy pattern recognition method in image classification field [5]. However, these traditional methods require experienced people to use them well and can be limited by the influence of environment, which means that traditional technique is not a perfect solution to automatic identification on leaf diseases [6,7]. The second one is the image processing technique based on machine learning technology. This approach is considered as a more valuable method than traditional methods in leaf diseases detection [3].

In this paper, we propose a solution for automatic identification of leaf diseases including two steps: cropping and classification. In the first step, YOLOv4-Tiny [8] is used to determine the position of leaves and remove redundant background. On the second step, a changed network architecture is used to classify species of plants and categories of leaf diseases. What's more, this classification network was designed for efficient computing speed. Finally, the algorithm is deployed to an embedded platform, and this system can identify leaf diseases in real time.

2 Related Work

The image classification technology based on deep learning has developed rapidly since 2012, which made the automated identification of large-scale leaf diseases possible [9]. In the field of automatic image recognition on leaf diseases, most research focused on the PlantVillage dataset [6,10]. Mohanty et al. [11] tested the performance of some classical network models in the PlantVillage dataset and found that the inception module got the best score in this dataset. Besides, Anjaneya [12] used the methods of transfer learning and discriminant learning to research the performance of ResNet34 and ResNet50 on PlantVillage. The result is that only four epochs were needed to achieve 98% accuracy on this dataset.

The scarcity of dataset made it difficult to implement the automatic leaf diseases identification system. Moreover, using ordinary network models to detect leaf diseases in the natural environment was of difficulty when the training of these models only relied on PlantVillage [13]. For the implementation of this system, Udutalapally et al. [14] set up a fully automatic monitoring and irrigation agricultural equipment on the field. This test lasted for 3 months and completed the task relatively successfully. However, the network structure used on equipment was relatively unitary. It couldn't finish the identification task of complex disease species. In the improvement of the identification method, Brahimi et al. [15] proposed a Teacher/Student Network based on transfer learning to fix this flaw, while Lee et al. [4] used attention-based RNN model to solve this problem. On the other hand, Bi et al. [16] used the Wasserstein way based on gradient punishment with the Label Smooth Regularization(LSR) method to improve classification accuracy.

3 Network Construction Scheme:YOLOv4-Tiny + SEI-ResNet

Our solution is mainly used to solve two difficulties: the scarcity of dataset and the difficulty in implementation of this system. Therefore, we propose a multi-stage recognition scheme, and its specific process is shown in Fig. 1.

3.1 YOLOv4-Tiny: A Fast Method to Remove Redundant Background

The step to remove the unnecessary background needs to be positioned quickly and accurately. However, traditional image segmentation technology is difficult to indicate good results in complex backgrounds, and the image processing speed of image segmentation technology based on deep learning is slower than that of image detection technology. Moreover, the error tolerance rate of image segmentation technology is worse than that of image detection technology. Therefore, image detection technology is the best way to remove redundant background.

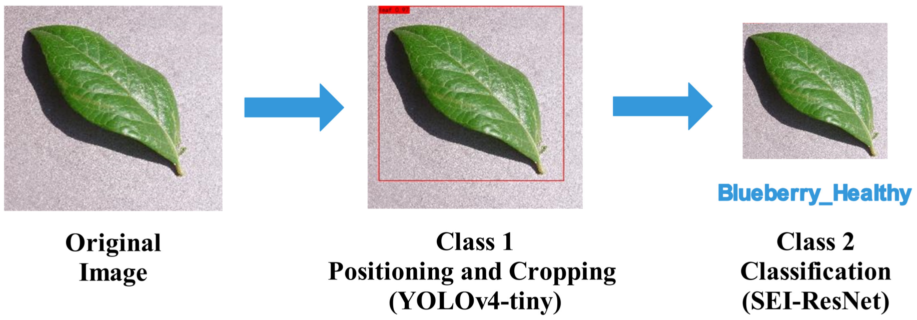
Blueberry_Healthy
Original Image
Class 1 Positioning and Cropping (YOLOv4-tiny)
Class 2 Classification (SEI-ResNet)

Fig. 1. Process of Identification. Our scheme is divided into two steps: the removal of redundant background and the image classification.

On the other hand, YOLOv4-tiny has a faster computing speed compared to the same period of other networks. It can achieve detection speed close to 375fps on 1080ti graphics card while maintaining high positioning accuracy [8].

3.2 SEI-ResNet

In order to enable the network to be deployed in embedded devices for real-time detection, by referring to the speed test of different networks on Jetson Nano [17], the detection speed needs to be significantly higher than ResNet50 and the accuracy should be close to it.

Therefore, we propose an improved CNN model based on Inception module, ResNet and SE-Net [18]: SEI-ResNet. Its overview figure is shown in Fig. 2.

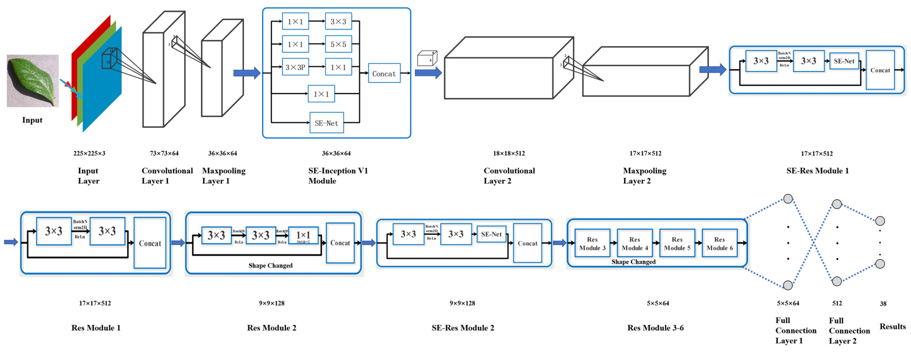

Fig. 2. Overview of SEI-ResNet. It mainly includes six parts: Convolutional Layer, Maxpooling Layer, SE-Inception V1 Module, SE-Res Module, Res Module and Full Connection Layer.

At the front end of this network, we combined SE Module and Inception Module to develop the SE-Inception V1 Module. This type of module is considered to have multi-stage feature extraction capabilities and have good robustness for the extraction of complex features [19]. Besides, SE-Res Module has similar functions in this network. Nevertheless, the powerful feature extraction capabilities as these modules have, too many of which may greatly increase the fragmentation of network operations. Moreover, different channel width may increase the memory access cost (MAC) [20]. Thus, in this network, we used as many "balanced" convolutions (equal channel width) as possible and used as few modules as possible, which may increase the degree of fragmentation.

On the other hand, since the number of layers in the model does not exceed 30, which is less than that of ResNet34, some overfitting methods like Dropout, Batch-Normalization [21] still need to be used in the network. In detail, its structure can be shown on Table 1.

3.3 Network Training

Training for YOLOv4-Tiny. Since the role of YOLOv4-tiny is mainly to locate the position of the leaf, pictures that can only show the local part of the leaf will not be able to be positioned. Therefore, all images of corn and the squash leaf images cannot be used to train. We manually annotate the remaining 33 categories of images, each category is labeled with 50 images. Finally, a total of 1650 images can be used for training. In these images, 70% of them are used for training and the remaining 30% are used for validation. The details of the train environment are shown on Table 2. Moreover, the loss of the model no longer drops significantly after 21000 iterations, and this indication is the termination condition of training.

Table 1. Details of the SEI-ResNet

Network number	Network type	Number of feature maps	Kernel size	Padding size	Step size	Additional operating
Input	Input layer	3	–	–	–	–
C-1	Convolution	64	13	2	3	BatchNorm2d
MP-2	Maxpooling	64	3	0	2	–
SE-In-3	Convolution	128	1, 3, 5	0, 1, 2	1	BatchNorm2d
	Maxpooling		3	1	1	BatchNorm2d
	SE-Net		–	–	–	BatchNorm2d
C-4	Convolution	512	6	2	2	BatchNorm2d
MP-5	Maxpooling	512	2	0	1	–
SE-Res-6	Convolution	512	3	0	1	BatchNorm2d
	SE-Net		–	–	–	BatchNorm2d
Res-7	Convolution	512	3	0	1	BatchNorm2d
Res-8	Convolution	128	1, 3	0	1	BatchNorm2d
SE-Res-9	Convolution	128	3	0	1	BatchNorm2d
	SE-Net		–	–	–	BatchNorm2d
Res-10	Convolution	128	3	0	1	BatchNorm2d
Res-11	Convolution	64	1, 3	0	1	BatchNorm2d
Res-12	Convolution	64	3	0	1	BatchNorm2d
Res-13	Convolution	64	3	0	1	BatchNorm2d
F-14	Fully Connected	Number of Neurons: 1600	–	–	–	BatchNorm1d and Dropout
F-15	Fully Connected	Number of Neurons: 512	–	–	–	–
Output	Output Layer	Number of Categories: 38	–	–	–	–

Table 2. Details of the train environment

Train environment		
Type	Configuration (YOLOv4-tiny)	Configuration (SEI-ResNet)
System	Windows10	Ubuntu 18.04.5 LTS
CPU	Intel(R) Core(TM) i5-8300H CPU@2.30 GHZ	Intel(R) Core(TM) i9-10920X CPU@ 3.50 GHz
GPU	NVIDIA GeForce GTX 1050 (4G)	NVIDIA GeForce RTX 3090 (24G)
CUDA	10.2	11.1
Cudnn	7.6.5	8.0.3
Pytorch	1.8.1	1.7
Python	3.8.5	3.8.5
Train parameters		
Type	Value (YOLOv4-tiny)	Value (SEI-ResNet)
Batch	32	16
Momentum	0.9	–
Decay	0.0005	0.05
Epoch	21000	33
Learning Rate	0–100: 0.001	0.0002 StepLR (Size = 3)
	100–10000: 0.0001	
	10000–21000: 0.00001	

Training for SEI-ResNet. It's necessary for SEI-ResNet to train all images. On the contrary, 80% of these images are trained while 20% of them are used to validate. In order to prevent the model from overfitting, when the accuracy of the validation set exceeds 99.4%, the model will stop training. On the other hand, the Adam optimizer and an optimization algorithm that decreases the learning rate with epoch (StepLR) [22] are used to train the model. Besides, the other training details are recorded in Table 2.

In terms of image preprocessing, preventing overfitting of the model is the primary goal. In this step, online learning of images is used for pictures in every epoch, and its process is shown in Fig. 3.

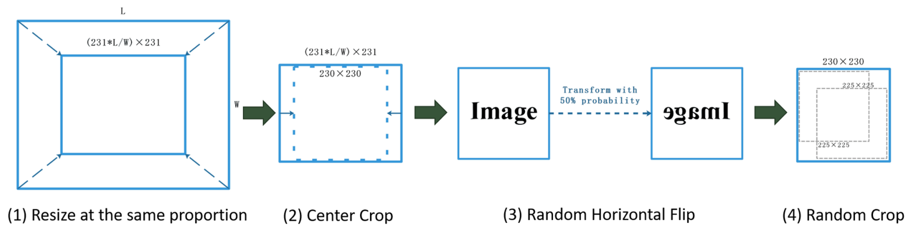

Fig. 3. Process of online learning. During image training, each image must be processed by the 4 steps: Resize, Center Crop, Random Horizontal Flip and Random Crop

4 Experimental Results

4.1 Automatic Leaf Diseases Detection(ALDD) Network Positioning Test

In the field of deep learning, there are many methods to evaluate the quality of a model. In this condition, our model just needs to judge whether there are leaves in the target area. Hence, this model can be considered a two-class classification model, and a discriminant table can be indicated on Table. 3.

Table 3. Matrix for leaves detection

	Leaves (in the forecast)	Non-leaves (in the forecast)
Leaves (in fact)	TP	FN
Non-leaves (in fact)	FP	TN

In this table, TP indicates that there are actually leaves in the target area, and the predicted result is also the same; TN indicates that there are actually no leaves and no leaves are predicted; FP means that there are actually no leaves, but the predicted is the opposite; FN means that there are leaves in reality, but this result is not predicted. Based on these conditions, the formula of Recall and Precision can be demonstrated.

$$Recall = \frac{TP}{TP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

By using the formula (1) and (2), in the validation set containing 330 images, the value of *Recall* is 100% and that of *Precision* is 98.82%. Thus, combining these two values according to the sequence of sample confidence, the average precision (AP) curve can be shown in Fig. 4.

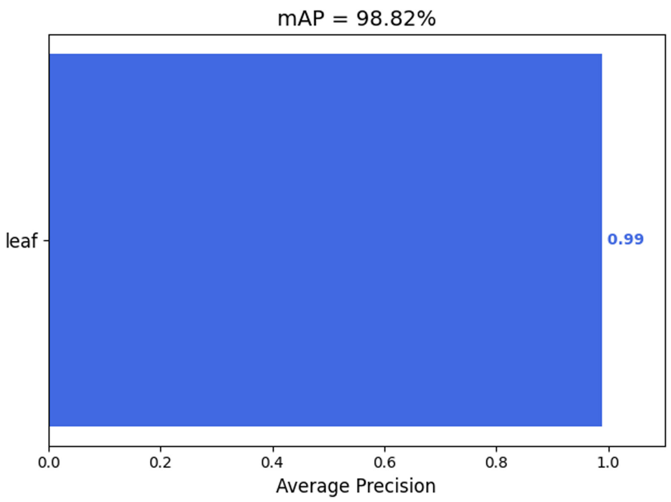

Fig. 4. AP Curve. Because this model only needs to detect one category, AP is the best indicator to evaluate the effect of the precision of the model, and the mean average precision (mAP) is the area under the curve, which can demonstrate the overall precision of this model [3].

4.2 ALDD Network Ablation Test

The effectiveness of the residual structure, SE Module and Inception V1 Module have been proven in many networks [18,19]. However, the effect of these structures used in SEI-ResNet is still undefined. By comparing the differences between these structures and ordinary convolutional structures, their effects in the network can be verified.

In this test, the training process and training duration of the model were used as indicators to evaluate the effectiveness of these structures. Specifically, the kernel size of the ordinary convolutional structure is set as 3, the stride is set to 1, and the padding is 1. Besides, parameters such as learning rate and Decay were set to the same value and the hardware environment was also kept consistent with Table 2 in this test. Moreover, the iteration termination condition was set when the accuracy of the verification set exceeded 99.4%. Figure 5 shows the result of ablation test on SE-Inception V1 module, which indicates the effects on SE Module, Inception V1 Module and SE-Inception V1

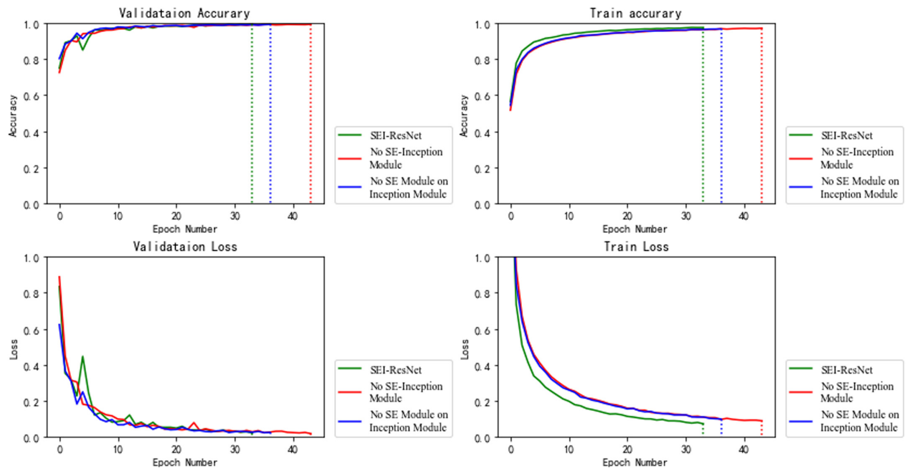

Fig. 5. Ablation Test on SE-Inception V1 Module. The red curve indicates the training process that convolution structure is used to replace the SE-Inception V1 Module while the purple curve shows that the training process that SE-Inception V1 Module is replaced with Inception V1 Module. (Color figure online)

Table 4. Computing speed comparison among different conditions

Model type	Mean training time (s)
No SE-inception module	192.58
No SE module on inception module	213.78
SEI-ResNet	216.56

Module. On the other hand, Table 4 indicates the computing speed comparison among the three conditions.

From the result, it's clear that the Inception V1 Module is a structure that takes a long time to compute and the ability of SE-Inception V1 Module to improve the learning efficiency of models is stronger than other structures. It indicates the necessity of using the SE-Inception V1 Module at the front end of this network. Besides, due to the time-consuming nature of this structure, it was only used once in the network. On the other hand, SE Module can slightly improve the learning performance of the model with almost no additional time. That's the reason that this structure was used repeatedly in residual structures.

4.3 ALDD Network Classification Test

Computing speed and computing efficiency are the two key points for evaluating the performance of classification models. In order for the automatic identification system to have a wide range of leaf disease identification capability, the system needs to have the function of real-time identification. This function has a high demand for model speed. Thus, comparing mainstream networks in a single learning time of PlantVillage, the Table. 5 can be listed.

Table 5. Computing speed comparison among different networks

Model type	Mean training time (s)
DesNet101	640.28
ResNet34	234.76
ResNet50	323.94
SEI-ResNet	**216.56**
VGG16	444.54
SqueezeNet1.1	142.62

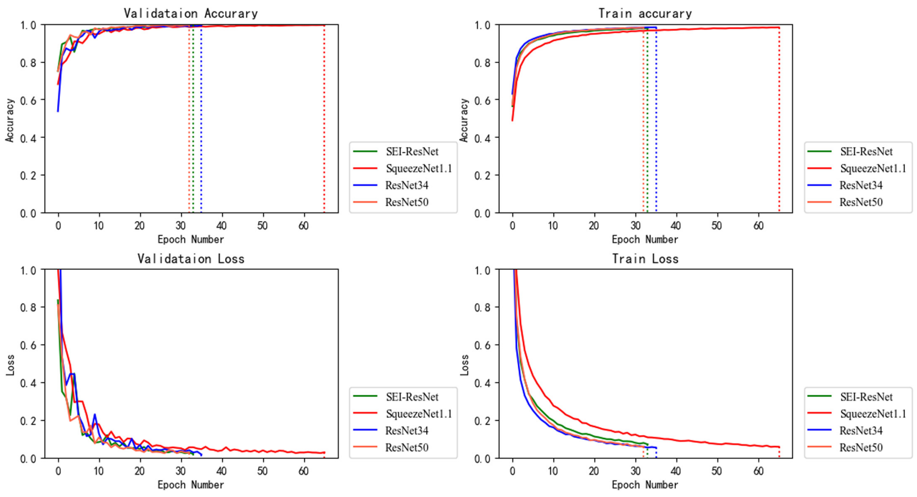

Fig. 6. Accuracy and loss curve for 4 networks.

By selecting the top 4 (short time) networks to compare their training curves, it's clear that there is only one epoch between the SEI-ResNet and the ResNet50 in Fig. 6. However, although ResNet50 exceeds SEI-ResNet by one epoch in training efficiency, its computational speed is 49.58% slower than SEI-ResNet, which means that SEI-ResNet has better overall performance than mainstream classification networks.

4.4 ALDD System Test

All algorithms are finally put into the embedded platform for testing after being trained. For the embedded platform, because of the cost-effective performance, Jetson Nano was selected as our test platform. What's more, the system on Jetson Nano is Ubuntu 18.04 LTS, and the development environment on this platform is Python3.7 with the OpenCV3.4.6, Pytorch1.4 and CUDA10.

After converting the model to TensorRT for acceleration, the recognition speed of our ALDD System can reach 15.32fps. Compared with the test done by Nvidia [17], our system is closer to the speed of SSD, ResNet18, and it surpasses classic networks such as Inception V4 and VGG19.

5 Conclusion

In this paper, we proposed a multi-stage detection algorithm in leaf diseases recognition field. This algorithm has three main contributions. First, it made up for the defect that the general target detection algorithm needed to confirm the location and category of the target at the same time. Second, the core network (SEI-ResNet) proposed by us in this system had better performance than mainstream classification networks as a whole. Third, this system had the ability to automatically detect leaf diseases in real time, and the recognition speed could reach 15.32fps on Jetson Nano. Even though the lack of data set was not enough to provide the system with the ability to detect plant diseases on the spot, the improvement in the performance of the algorithm showed that the system would be capable of real-time identification of a wide range of leaf diseases after getting enough data to learn. In the future, we will collect data provided by plant disease specialists and use these data to conduct small-scale tests by using our system on some farms or orchards. Besides, we will further optimize the computational cost of this architecture.

Acknowledgments. This work was supported by the National Natural Science Foundation of China under Grant 62073129.

References

1. Nalini, S., et al.: Paddy leaf disease detection using an optimized deep neural network. Comput. Mater. Continua **68**, 1117–1128 (2021)
2. Buja, I., et al.: Advances in plant disease detection and monitoring: from traditional assays to in-field diagnostics. Sensors **21**, 2129 (2021)
3. Liu, J., Wang, X.: Plant diseases and pests detection based on deep learning: a review. Plant Meth. **17**, 22 (2021)
4. Lee, S.H., Chan, C.S., Mayo, S.J., Remagnino, P.: How deep learning extracts and learns leaf features for plant classification. Pattern Recogn. **71**, 1–13 (2017)
5. Wallace, T.P., Mitchell, O.R., Fukunaga, K.: Three-dimensional shape analysis using local shape descriptors. IEEE Trans. Pattern Anal. Mach. Intell. PAMI **3**, 310–323 (1981)
6. Boulent, J., Foucher, S., Théau, J., St-Charles, P.-L.: Convolutional neural networks for the automatic identification of plant diseases. Front. Plant Sci. **10**, 941 (2019)

7. Tsaftaris, S.A., Minervini, M., Scharr, H.: Machine learning for plant phenotyping needs image processing. Trends Plant Sci. **21**, 989–991 (2016)
8. Darknet: Open Source Neural Networks. https://github.com/AlexeyAB/darknet. Accessed 10 Apr 2021
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. Commun. ACM **60**, 84–90 (2017)
10. PlantVillage-Dataset: Open data set on plant leaf diseases. https://github.com/spMohanty/PlantVillage-Dataset. Accessed 10 Feb 2021
11. Mohanty, S.P., Hughes, D.P., Salathé, M.: Using deep learning for image-based plant disease detection. Front. Plant Sci. **7**, 1419 (2016)
12. Anjaneya: Plant disease detection from images. arXiv pre-print arXiv:2003.05379 (2020)
13. Ferentinos, K.P.: Deep learning models for plant disease detection and diagnosis. Comput. Electron. Agricult. **145**, 311–318 (2018)
14. Udutalapally, V., Mohanty, S.P., Pallagani, V., Khandelwal, V.: sCrop: a novel device for sustainable automatic disease prediction, crop selection, and irrigation in Internet-of-Agro-things for smart agriculture. IEEE Sens. J. (2020)
15. Brahimi, M., Mahmoudi, S., Boukhalfa, K., Moussaoui, A.: Deep interpretable architecture for plant diseases classification. In: 2019 Signal Processing Algorithms, Architectures, Arrangements, and Applications, pp. 111–116 (2019). IEEE
16. Bi, L., Hu, G.: Improving image-based plant disease classification with generative adversarial network under limited training set. Front. Plant Sci. **11** (2020)
17. Jetson Nano: Deep learning inference benchmarks. https://developer.nvidia.com/embedded/jetson-nano-dl-inference-benchmarks. Accessed 4 Mar 2021
18. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, 18–23 June 2018, pp. 7132–7141 (2018). https://doi.org/10.1109/cvpr.2018.00745
19. Xie, S., Girshick, R., Dollar, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017, pp. 5987–5995 (2017). https://doi.org/10.1109/CVPR.2017.634
20. Ma, N., Zhang, X., Zheng, H.-T., Sun, J.: ShuffleNet V2: practical guidelines for efficient CNN architecture design. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. LNCS, vol. 11218, pp. 122–138. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01264-9_8
21. Goodfellow, I., Bengio, Y., Couville, A.: Deep Learning, 1st edn. MIT Press, Cambridge (2016)
22. Loshchilov, I., Hutter, F.: SGDR: stochastic gradient descent with warm restarts. arXiv pre-print arXiv:1608.03983 (2017)