# The Implementation of A Crop Diseases APP Based on Deep Transfer Learning

Mengji Yang
School of Information Science and Engineering
Chengdu University
Chengdu, Sichuan, China
e-mail: ymjmg@foxmail.com

Daiwei Li
Department of Software Engineering
Chengdu University of Information Technology
Chengdu, Sichuan, China
e-mail: ldwcuit@cuit.edu.cn

Minquan Chen
School of Information Science and Engineering
Chengdu University
Chengdu, Sichuan, China
e-mail: landy_orton@yeah.net

Abdelaziz Bouras
Department of Computer Science
Qatar University
ictQATAR, Box. 2731, Doha, Qatar
e-mail: abdelaziz.bouras@qu.edu.qa

Yiqian Tang
School of Information Science and Engineering
Chengdu University
Chengdu, Sichuan, China
e-mail: tangyiqian@cdu.edu.cn

Xi Yu*
School of Information Science and Engineering
Chengdu University
Chengdu, Sichuan, China
Key Laboratory of Pattern Recognition and Intelligent Information Processing in Sichuan
e-mail: yuxi@cdu.edu.cn

***Abstract*—Classifying the severity of crop diseases is the staple-basic element of the plant pathology for making disease prevent and control strategies. The diagnosis of disease needs timeliness and accuracy. Thanks to the development and popularity of smart phones and mobile networks, this makes possibly to develop mobile applications that can be widely accepted by users in the agricultural community. This paper provides a system that can detect the severity of crop diseases automatically and intelligently through taking photos. The development of this mobile app is based on deep transfer learning that we proposed an improved method with nearly 92% accuracy based on ResNet 50. The significantly high success rate makes the model a very useful advisory or warning tool. This project provides a new idea and solution for the detection of crop diseases in agriculture.**

***Keywords-mobile app; crop disease; ResNet 50; deep transfer learning***

## I. INTRODUCTION

In recent years, food production and security are given serious attention[1]. Food security is affected by climate change and crop diseases[2]. Crop diseases are one of the major challenges in agricultural sciences and catastrophic crop diseases intensify the shortage of food supplies. In developed nations, agricultural modernization has got a flourish development that contrasts to most less developed countries. In developed countries, enough technology and policy support were adopted to minimize crop losses. In developing world, crop diseases can lead to disastrous consequences, because there have plenty of smallholder farmers whose livelihoods depend on healthy crops[3]. If smallholder farmers and plant protection units unable to detect crop diseases and take appropriate measures timely, there could be serious losses. For instance, in China, according to relevant statistics and analysis, from 2006 to 2015, crop diseases were in serious situation with the disaster area of crop diseases ranged from 463.5 million to 507.5 million $hm^2$. On the basis of projections by NATESC (The National Agro-Tech Extension and Service Center), the disaster area of crop diseases will reach 300 million $hm^2$ and caused considerable economic damage[4].

Formerly, the identification of crop diseases had to be done by experts or local plant protective station. In recent years, agricultural expert knowledge base was established and various agricultural websites have appeared, people can get some advice through expert system online or search by themselves. In fact, this approach has some limitations. First, the types of crop diseases are too great, just take the potato as an example, there are more than a hundred kinds of diseases related to potatoes[5]. Meanwhile the relevant experts are limited in the field of crop diseases. Second, due to limited plant protection workers, some areas are remote and inadequate transportation, many crop diseases cannot be identification on site in time. Because of these factors,

978-1-7281-9741-8/20/$31.00 ©2020 IEEE

people unable to get the efficacious prevention suggestions timely.

With the development of smart phones and mobile network as well as the computer technology evolves, many developers choose computer vision technology[6] as the basis to develop mobile apps and detect the crop diseases. In computer vision field, one of the typical applications is image recognition. The researchers often use convolutional neural network (CNN)[7] to do image recognition. CNN can easily extract image features at the pixel level. This characteristic makes CNN get sufficiently finer performance on image identification.

A lot of researches have been done on using CNN to identify crop diseases. However, as the amount of dataset increases rapidly, training a model from scratch is quite consuming time and occupying computing resource. Sometimes these elements are not easy to implement. All of these factors make mobile apps hard to commercialize quickly.

Therefore, we apply transfer learning[8] to the field of deep learning. In this thesis, we have presented an improved deep transfer learning approach called CDCNNv2 based on ResNet 50[9] (original model pre-trained from ImageNet[10]) to classify the severity of diseases (exempli causa, healthy, general or severe). We used 'AI Challenger 2018' crop dataset and conduct the experiments. We got 92% accuracy by doing deeper analysis of the dataset. Based on CDCNNv2, we deployed the trained model to the GPU server. By using TensorFlow Serving[11] and Docker container[12], we succeeded in image teleprocessing and obtained the detection results. Based on this principle, we have successfully developed a mobile app about detection of crop diseases. Anything else, app also provides prevention advice, expert consultation, and crop encyclopedias and so forth.

The composition of this paper is as follows. Section II has enumerated leading studies in crop diseases detection technology and correlative mobile apps status. Section III, we applied the specific implementation steps of the method we proposed. In section IV, we discussed and analyzed the large number of experiments and summarized the results. Section V introduced the rationale of model deployment and invoking block in detail. The basic information of the APP has been displayed in section VI. Finally, we will summarize the status and discuss future developments of the mobile app in section VII.

## II. Related Work

In this section, we will discuss the related work in two parts. One is about arithmetic researches, another is research status of mobile apps.

In recent studies, Anyela Camargo et al. detected banana leaf disease called Black Sigatoka with different stages of the disease by using digital image processing[13]. And Jayme Garcia Arnal Barbedo et al. identified multiple plant diseases by using digital image processing[14]. Some other researchers also had performed related study by using machine learning. Rumpf, T et al. used support vector machines to detect and classify sugar beet leaves[15]. With the development of deep learning, Amanda Ramcharan et al. used Inception v3 network to classify the cassava diseases[16]. Ferentinos, K. P. and Sharada P. Mohanty et al. both used an oppositely large dataset to detection various crops with different diseases[17, 18].

According to the above algorithms, each of them have some good points and some disadvantages. For instance, traditional digital image processing equipped with a wide range of application, high processing accuracy, high flexibility and so forth. But the processed images require to observe and evaluate by professionals guiding. Meanwhile, we may not be able to find the experts who know various crop diseases timely. Machine learning algorithms can identify crop disease automatically. Nonetheless, before the model training, original images needed experts to mark out the feature area of crop diseases manually. If the dataset is large, the process will be very time-consuming and energy-consuming. Deep learning methods can minimize above-mentioned problems, it just needs professionals to label out whether the original images are diseased. Based on these simple labels merely, overall process realizes complete automation from training to classification.

Other studies focus on program development, Sarah J. Pethybridge et al. developed an IOS app used image SADs to assess the severity of the disease[19]. This app is only for the severity of beets and semi-automatic, users need to choose comparison criteria. Wayne Goodridge et al. established a variety of disease models by using a set of characteristics which are weighted for each disease using two types of weights[20]. The disadvantage is cannot identify severity. Scot C. Nelson et al. calculated the area of the affected to assess the severity[21]. The operation of the app requires users have a certain professional basis.

In this study, the purpose of our mobile app is for easy to use, real-time and full-automatic detection, get results on time, users without professional basis.

The next section, the key algorithm of our model will be detailed introduction.

## III. Method

### A. Selecting the Appropriate Pre-trained Model

Selecting the appropriate pre-trained model is the key to the success of transfer learning. How to choose the correct pre-trained model is useful for our own training, the most important point is the dataset of the pre-trained model (source dataset) and our dataset (target dataset) are related or similar. Such as, some or other pre-trained model is about bicycle identification, and we think this pre-trained model might be useful to us for motorcycle identification.

In this paper, we used the ResNet 50 model which has been pre-trained on ImageNet. ImageNet is a large image database which has more than 14 million natural images and include over 20,000 categories (contains a large number of crop images). However, the pre-trained model from ImageNet is not optimize for crop diseases, so the pre-trained model performed not excellent on our crop diseases dataset. Therefore, we need to improve the pre-trained model accordingly. The next part, we will expound what have we improved.

### B. Approach

The main strategy that we used is fine-tuning. The fine-tuning process is that before using the weights of the pre-trained model, we can adjust one or more layers. Then about the base framework, we use the ResNet 50, transfer the weights of the pre-trained model to the base framework. Employed the fine-tuning manner and appended some layers to adapt our task of identification on crop diseases rank.

The proposed improved method called CDCNNv2 that comprises the basic steps as follows:

*1)* Before the images entering the model, we added a zero padding layer with padding window is 3*3. Using value of '0' to pad around the edge of the target crop images, in order to extract the edge feature information of the crop images preferably.

*2)* Using ResNet 50 as the base framework, transfer the weights of convolution layers from pre-trained model. Then let the weights of convolution layers update with training.

*3)* After the convolution layers, an average pooling layer with the pooling window is 2*2 was added. Calculating the average value of the 2*2 matrix region of the images is beneficial to preserve more detail imformation of the images.

*4)* The next, the flatten function was added as a flatten layer. The purpose is making multi-dimensional inputs into one-dimensional. This layer can expedite the calculation.

*5)* After the flatten layer, two fully connected layers were added. There has a batch normalization function between them. The prior fully connected layer adopted outputs dimensions of 1,024 and the activation function is 'Relu'. Whereafter, added a batch normalization function with the purpose of getting the faster training, in the meantime, the classification accuracy able to be improved after convergence. In odder to distinguish 61 level of crop diseases, the final fully connected layer with output dimensions of 61, and we use the 'Softmax' function as the final activation function.

In Fig. 1, we show the concise model structure. Next, we will establish experiment of this method.

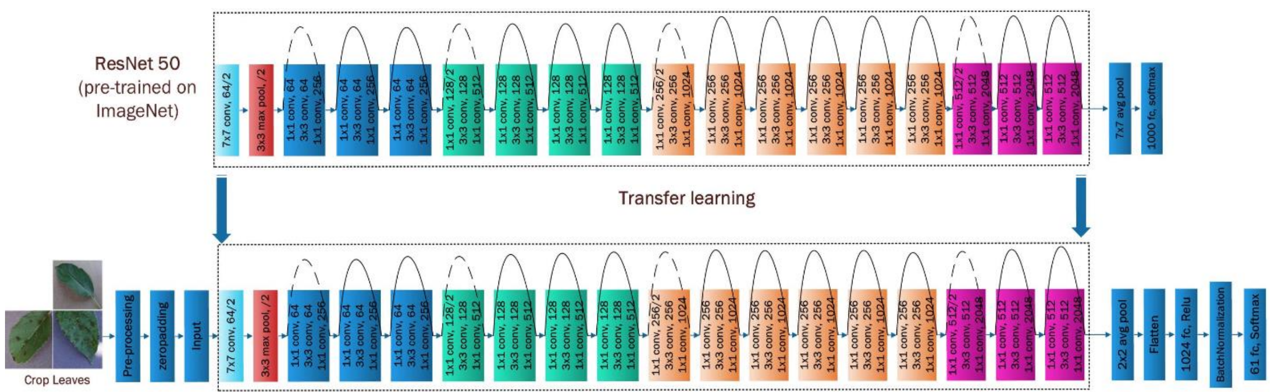



Figure 1. The concise model structure. The wights of the ResNet 50 pre-trained from ImageNet are transferred to our CNCNNv2 model. The weights of the convolution layers are updated with the whole training process.

## IV. Experiments and Results

### A. Experimental Environment

Experimental environment: the operating system is CentOS 7, we use GPU of Tesla P100 for training. Use the open source artificial neural network library Keras.

### B. Dataset Description

As important as pre-trained model selecting, dataset selection and collection are the same key, for making the model have better generalization ability. We chose one called 'AI Challenger 2018'. The dataset contains 36,261 labeled crop images. There are 31,721 images in the training set and the verification set has 4,540 images. The dataset has 10 species (apple, orange, grape, cherry, peach, strawberry, maize, pepper, tomato, potato) and 27 diseases in total. There are 61 classification (species-disease-severity) in total. For instance, in Fig. 2, we showed one example.

### C. Image Preprocessing

In the part of image preprocessing, we have adopted the method of data augmentation. The purpose of this operation contains improving the generalization ability of the model and increasing the noise to improve the robustness of the model. In the real life, images taken by photographic equipment are uneven quality. This strategy can also simulate the shooting environment realistically. In the preprocessing, we change the original images by increasing or decreasing the brightness randomly, rotating and flipping.

The specific equation as follows, when we regulated the brightness randomly as follows:

$$\phi = \gamma \cdot \alpha + \beta. \tag{1}$$

The '$\phi$' signify the changed image, '$\gamma$' is original image, '$\alpha$' and '$\beta$' represent contrast and brightness. '$\alpha$' $\in$ (0.9, 1.1), '$\beta$' $\in$ (-10, 10) with part of integer.

And then we normalize the image to (-1, 1). After that, we set a center point and rotate the image randomly by an angle to construct the rotation matrix. Then, take the matrix affine and mirror flip the picture randomly. Finally, unify the size of the image.

In Fig. 3, we will demonstrate the operations mentioned above.

(1) (2) (3)

Figure 2. Example for maize leaves. (1) Healthy maize leaf, (2) Maize grey leaf spot-general, (3) Maize grey leaf spot-serious.

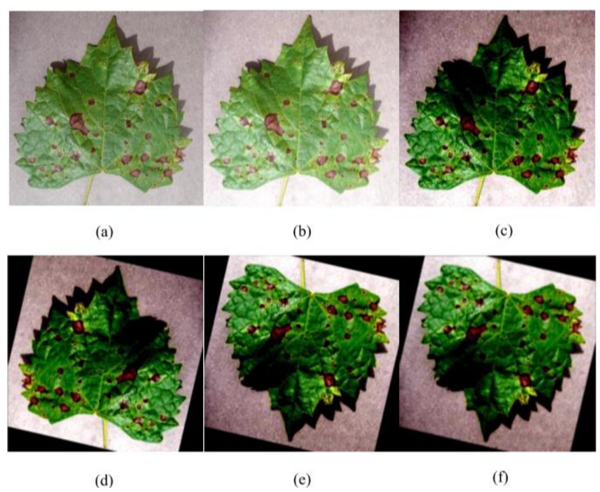



Figure 3. (a) Original picture, (b) regulation of the brightness, (c) Normalization, (d) Rotate, (e) Mirror flip, (f) Uniform in size.

### D. Results

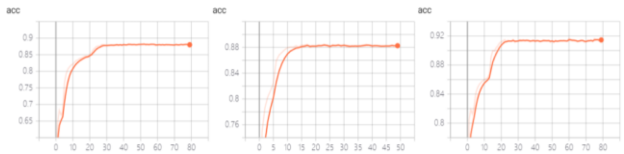
For the sake of the effect of our model that we proposed, we did two comparison experiments, one used approximate condition without transfer learning, another used Xception network framework with transfer learning. All experiments use 64 batch sizes and 80 epochs. Next, in the TABLE I. , we showed the results of the three experiments.

TABLE I. THE RESULTS OF EXPERIMENTS

| **Experiments** | **Criterion** | | |
|---|---|---|---|
| | ***Average accuracy(%)*** | ***Average converge time(h)*** | ***Number of Exp*** |
| Original model | 87.52 | 9.78 | 10 |
| Xception | 88.48 | 7.02 | 10 |
| CDCNNv2 | 91.51 | 8.48 | 10 |

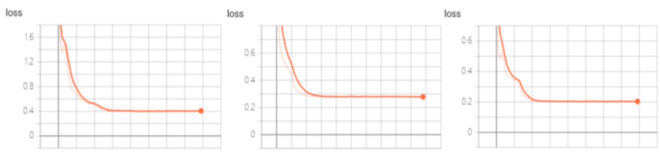
As shown from the Table I, under the same experiment conditions, experimental results show that our accuracy is better than non-transfer learning method and Xception. Our method is not the best at convergence time, but compare with non-transfer learning we saved 13% of the convergence time. Obviously, our method works better.

As can be seen from the Fig. 4, non-transfer learn method converged at around 30$^{th}$ epoch, but using transfer learning strategy converged at around 25th epoch and Xception even converged faster. It is certified that using transfer learning makes the convergence speed faster and higher accuracy rate.



(a) (b) (c)

Figure 4. (a) Non-transfer learning model, (b) Xception, (c) CDCNNv2.



(a) (b) (c)

Figure 5. (a) loss of non-transfer learning, (b) Xception, (c) CNCNNv2.

Identically, we can see the Fig. 5, all experiments achieved low losses. Nonetheless, loss of our algorithm is the best.

Combined with the above results, convergence speed of our model is faster and has better accuracy. We not only saved training time but also saved computing resource.

In the next part, we will elaborate how do we deploy and invoke the model.

## V. Model Deployment and Invocation

Before deploying the model, there are two important tools that we need to understand. One of them is TensorFlow Serving, another is Docker container.

TensorFlow Serving is a flexible, high-performance serving system for machine learning, designed for production environments. It can serve multiple models, or multiple versions of the same model simultaneously, and exposes both gRPC as well as HTTP inference endpoints. Docker container makes the model easy to deploy to the TensorFlow Serving with a couple lines of code.

Next, we will paste in part of the deployed critical code.

```
model_filePath = './my_model_resnet_v2.h5'
model = load_model(model_filePath)
export_path_base = "serving"
export_version = '2';
export_path=os.path.join(export_path_base,export_ver
sion)

builder=tf.compat.v1.saved_model.builder.SavedMdel
Builder(export_path)

tensor_info_input=tf.compat.v1.saved_model.utils.buil
d_tensor_info(model.input)

tensor_info_output=tf.compat.v1.saved_model.utils.bu
ild_tensor_info(model.output)
```

This part of the operation is to define service version and assign where the model needs to be saved. Then, gets the inputs and outputs of the model.

After that, we need to package the inputs and outputs for receiving parameters from the client expediently. The abbreviated code is as follows:

```
prediction_signature=
(
tf.compat.v1.saved_model.signature_def_utils.build_si
gnature_def
        (
inputs={'images':tensor_info_inpu},
outputs{'result':tensor_info_output},
method_name=tf.saved_model.PREDICT_METHOD_
NAME)
)
builder.add_meta_graph_and_variables
(
K.get_session(),
[tf.saved_model.SERVING],
signature_def_map={'predict_images':prediction_signa
ture,}
)
builder.save()
```

The next step is to start the online service. In this step, we will use the Docker container with following these steps:

```
Docker run -p port1 -p port2
-mount type = blind, source = THE SOURCE TO
MOUNT, target = THE DESTINATION TO MOUNT
t tensorflow/serving
--model_config_file=/model/models.config
```

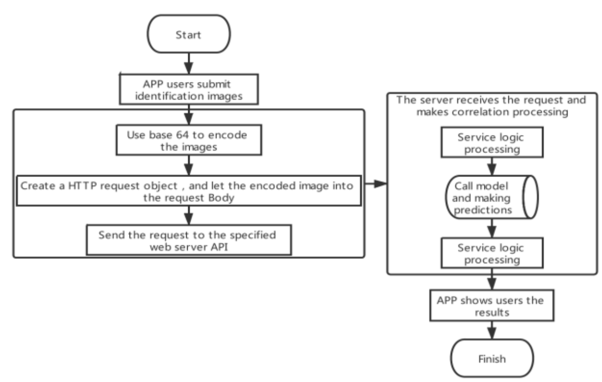



Figure 6. The flow chart of app interacting with the server.

The above steps are for starting Docker container, and the ‘port1’ is the Host Port, you can change it at will, but the ‘port2’ is fixed port, unable to change. Using config file and mounting by Docker can achieve deploy multiple models if you write the path and name of the model.

Sending and receiving data on android side are not complicated. We will show the consist of the steps as follows:

*1)* Encode the image in 64 bits.

*2)* Create a new RequestBody instance including format, the encoding type and encoding to encapsulate.

*3)* Request server connection, using the .url to get the server IP address. And .post sends the encapsulation to sever.

*4)* Through Response operation to get the results after defining the receiving of the data format.

In Fig. 6, we showed the flow chart of the above steps.

That is the whole process of model deployment and invocation. In the next section, we will introduce some cases of the APP.

## VI. BRIEF INTRODUCTION OF THE APPLICATION

For the limitation of thesis, we only introduce the main function of mobile app and detection function in detail. We have implemented the deployment and invocation of the model, around this point, we deigned the following features in Fig. 7.

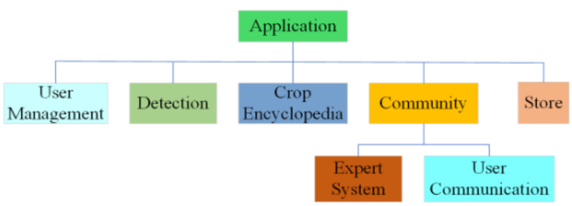



Figure 7. Application function diagram.


Figure 8. Interface of the detection results.

The brief introduction of each function module is as follows:

*1)* User management: Allows users to change nicknames, password, mailbox or phone number.

*2)* Detection: Allows users to take photos or web upload crop images. Then get the diagnosis results in seconds, and obtain the prevention suggestions or durgs recommended.

*3)* Crop encyclopedia: Allows users to search and query crop information.

*4)* Community: Allows users to exchange of crop relevant information. Meanwhile, opening help channel for users to ask experts for help.

*5)* Store: later period, we will introduce thirdparty sellers, links jump by drugs recommendations of detection function.

Identification function displayed as Fig. 8.

We used apple leaf with venturia inaequalis serious to test. According to the above picture, it contains several results including degree of the disease, disease description, detection reliability, prevention suggestion and drugs recommended.

In order to verify accuracy of our app, we conducted a total of 200 identification tests, every test we all documented by log file as Fig. 9. In TABLE II. we will show the precision rate of the results. For example, venturia inaequalis include two types of degree (general and serious).

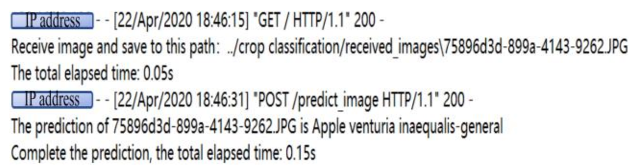



Figure 9. The part of log file.

TABLE II. THE RESULTS OF TEST

| Species | Criterion | | | |
|---|---|---|---|---|
| | ***Image Num*** | ***Correct Num*** | ***Wrong Num*** | ***Accuracy (%)*** |
| Apple health | 20 | 20 | 0 | 100 |
| Venturia inaequalis | 20 | 17 | 3 | 85 |
| Apple gray spot | 20 | 17 | 3 | 85 |
| Leaf mold of tomato | 20 | 17 | 3 | 85 |
| Tomato early blight | 20 | 19 | 1 | 95 |
| Corn rust disease | 20 | 16 | 4 | 80 |
| Grape health | 20 | 20 | 0 | 100 |
| Potato early blight | 20 | 17 | 3 | 85 |
| Orange health | 20 | 20 | 0 | 100 |
| Strawberry leaf blight | 20 | 18 | 2 | 90 |
| Total | 200 | 181 | 19 | 90.5 |

As shown in the TABLE II. , after 200 tests, we can conclude that the recognition function of our mobile app has a certain robustness and actual use condition. Please visit this website: https://www.bilibili.com/video/BV1c64y1M7Lw/ to get video information. The next part, we will discuss the above outcomes.

## VII. Discussion

For the past several years, with the continuous development of deep learning, great progress has been made in the field of image recognition. But with various network framework layers increasing constantly and the data size growth quickly, training seems to become very difficult, because of the huge computational resource consumption and time-consuming. The emergence of transfer learning makes a project commercialize commendably and quickly.

The experiment results express the algorithm of our model has robustness and practicality. Based on this model, we developed an application successfully that allow the user take the images of crops diseases to the server and get the diagnosis results in seconds. This makes our app a very useful advisory or early warning tool and brings great convenience to the diagnose of crop diseases in agricultural.

In the later, we will carry on improving our model for more precise and generalization. At the same time continue to ameliorate the function of the application for a better promotion.

## Acknowledgment

This research grateful for ‘AI Challenger 2018’ providing the dataset of crop diseases. It is supported by the open fund (MSSB-2019-02) of Key Laboratory of Pattern Recognition and Intelligent Information Processing, Chengdu University, China and Erasmus+ SHYFTE project (598649-EPP-1-2018-1-FR-EPPKA2-CBHE-JP) which funded with support from the European Commission.

## References

[1] Strange, R.N. and P.R.J.A.r.o.p. Scott, Plant disease: a threat to global food security. 2005. 43.

[2] Tai, A.P., M.V. Martin, and C.L.J.N.C.C. Heald, Threat to future global food security from climate change and ozone air pollution. 2014. 4(9): p. 817-821.

[3] Harvey, C.A., et al., Extreme vulnerability of smallholder farmers to agricultural risks and climate change in Madagascar. 2014. 369(1639): p. 20130089.

[4] Yuying Jiang, Wancai Liu, et al., 2020 Forecast of major crop diseases in China. China Plant Protection. 2020. 40(01): p. 102-103.

[5] Hooker, W.J., Compendium of potato diseases. 1981: International Potato Center.

[6] Forsyth, D.A. and J. Ponce, Computer vision: a modern approach. 2002: Prentice Hall Professional Technical Reference.

[7] Xu, L., et al. Deep convolutional neural network for image deconvolution. in Advances in neural information processing systems. 2014.

[8] Tan, C., et al. A survey on deep transfer learning. in International conference on artificial neural networks. 2018. Springer.

[9] He, K., et al. Deep residual learning for image recognition. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[10] Krizhevsky, A., I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. in Advances in neural information processing systems. 2012.

[11] Olston, C., et al., Tensorflow-serving: Flexible, high-performance ml serving. 2017.

[12] Merkel, D.J.L.j., Docker: lightweight linux containers for consistent development and deployment. 2014. 2014(239): p. 2.

[13] Camargo, A. and J.J.B.e. Smith, An image-processing based algorithm to automatically identify plant disease visual symptoms. 2009. 102(1): p. 9-21.

[14] Barbedo, J.G.A., L.V. Koenigkan, and T.T.J.B.e. Santos, Identifying multiple plant diseases using digital image processing. 2016. 147: p. 104-116.

[15] Rumpf, T., et al., Early detection and classification of plant diseases with support vector machines based on hyperspectral reflectance. 2010. 74(1): p. 91-99.

[16] Ramcharan, A., et al., Deep learning for image-based cassava disease detection. 2017. 8: p. 1852.

[17] Ferentinos, K.P.J.C. and E.i. Agriculture, Deep learning models for plant disease detection and diagnosis. 2018. 145: p. 311-318.

[18] Mohanty, S.P., D.P. Hughes, and M.J.F.i.p.s. Salathé, Using deep learning for image-based plant disease detection. 2016. 7: p. 1419.

[19] Pethybridge, S.J. and S.C.J.P.d. Nelson, Estimate, a new iPad application for assessment of plant disease severity using photographic standard area diagrams. 2018. 102(2): p. 276-281.

[20] Goodridge, W., et al., Intelligent diagnosis of diseases in plants using a hybrid Multi-Criteria decision making technique. 2017. 133: p. 80-87.

[21] Pethybridge, S.J. and S.C.J.P.d. Nelson, Leaf Doctor: A new portable application for quantifying plant disease severity. 2015. 99(10): p. 1310-1316.