Deep Learning Methods for Recognition of Orchard Crops' Diseases

Baratbek Sabitov[1]†, Saltanat Biibsunova[2]††, Altyn Kashkaroeva[3]††, Bolotbek Biibosunov[4]††

†Kyrgyz National University named after I. Arabaev, Department of Applied Informatics, Bishkek, Kyrgyzstan
††Arabaev Kyrgyz State University, Department of Applied Informatics, Bishkek, Kyrgyzstan

Abstract

Diseases of agricultural plants in recent years have spread greatly across the regions of the Kyrgyz Republic and pose a serious threat to the yield of many crops. The consequences of it can greatly affect the food security for an entire country. Due to force majeure, abnormal cases in climatic conditions, the annual incomes of many farmers and agricultural producers can be destroyed locally. Along with this, the rapid detection of plant diseases also remains difficult in many parts of the regions due to the lack of necessary infrastructure. In this case, it is possible to pave the way for the diagnosis of diseases with the help of the latest achievements due to the possibilities of feedback from the farmer – developer in the formation and updating of the database of sick and healthy plants with the help of advances in computer vision, developing on the basis of machine and deep learning. Currently, model training is increasingly used already on publicly available datasets, i.e. it has become popular to build new models already on trained models. The latter is called as transfer training and is developing very quickly. Using a publicly available data set from PlantVillage, which consists of 54,306 or NewPlantVillage with a data volumed with 87,356 images of sick and healthy plant leaves collected under controlled conditions, it is possible to build a deep convolutional neural network to identify 14 types of crops and 26 diseases. At the same time, the trained model can achieve an accuracy of more than 99% on a specially selected test set.

Keywords:
Diseases of agricultural plants, Orchard Crops' Diseases, Deep Learning Methods, Recognition

1. Introduction

We analyze the updated PlantVillage of images of plant leaves, where each class is assigned with class labels. Each class label represents a pair of crops-diseases, and we are trying to predict a pair of crops-diseases based only on the image of a plant leaf. In all the approaches described in this article, we resize the images to 256×256 pixels and make model optimization and prediction of these reduced images.

In all our experiments, we used color and black–and-white versions of the entire PlantVillage dataset. Let's start with the PlantVillage dataset in color; then we experiment with the gray version of the PlantVillage dataset. This set of experiments was designed to understand whether the neural network is really studying the "concept" of plant diseases or just studying the inherent systematic errors in the data set. [1] [9] [10] Figure 1 shows different versions of the same leaf for a randomly selected set of leaves.

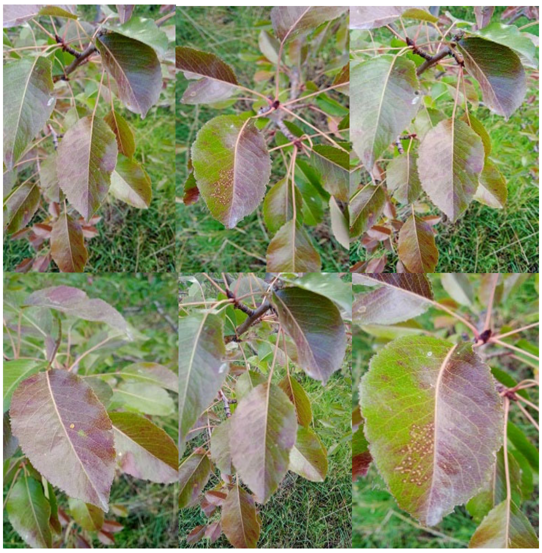


Fig. 1. Sick images of pear leaves obtained via smartphones by farmers

It can be seen from the drawings that some images are obtained in enlarged, reduced size, with rotation by a certain angle. In this regard, deep learning technologies - data augmentation - are used in the work for the main image. The identification and analysis of the leading, main influencing factors also plays a very important role in the identification of plant diseases. This article explores the key factors of influence and recognition of the severity of pear diseases using deep learning based on our database of pear diseases. Using deep learning neural networks, including VGG16, Inception V3, ResNet50, we have developed a neural network scheme that can be used to analyze the main signs in the recognition of diseases. Big cultivation of pears in the republic is noted in the Issyk-Kul region [1]. Diseases of pears and their contagiousness can significantly affect the normal growth of pear trees. For example, only in the Issyk-Kul region of the Kyrgyz Republic, many plantations were ruined because of this problem one after another.

The cause is a bacteriological burn of pears of a certain variety. As a result, scientific diagnostic measures are crucial to avoid misuse of recipes, excessive use of pesticides, pesticide residues, can lead to a significant reduction in pear yields. It can even cause problems with food safety, which is fraught with many consequences. Therefore, when using agricultural agrotechnologies with its increasing costs, combined with human labor, it is very difficult to help farmers in the diagnosis of diseases. Mutual communication is needed, as we noted above, when farmer

Manuscript received October 5, 2022
Manuscript revised October 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.10.33

who will send images of sick plants to the database, and then professionals diagnose plant diseases from the data accumulated on the server. Thus, there is an urgent need for automatic disease recognition technology that could detect, predict, identify and possibly recommend treatment of a particular disease.

Currently, there are basically two different technical ways to study plant diseases: 1) traditional image recognition processing technology, which is based on a small data set and does not require a large number of disease samples. But it includes image preprocessing, disease segmentation, feature extraction, classifier construction, model construction, etc. and all of them have so many manual processing steps [9] [12]. This technology is subjective, time-consuming and error-exposed: 2) Deep Learning (DL) technology is another promising technology, that uses DL network models to automatically extract and identify disease signs based on a large data set. There are fewer stages of manual processing, but there is a great demand for the number of disease samples. Below farmers send some of the unprocessed color images of healthy pears to the server.

Fig. 2. Healthy pear leaves

It has been proven that deep learning is a promising way to recognize objects [5] [11] [8]. DL technology has achieved outstanding achievements in image processing, as well as in the fields of medicine, industry and others. The advantage of deep learning is the training of signs, i.e. the identification of their strengths, for example, to detect diseases of pears.

2. Material and methods

Here it is necessary to choose InceptionV3, because it has fewer parameters compared to other models. Fewer parameters lead to less time to train the model. Since we have collected the number of images with the corresponding pear disease labels it is not enough to support learning from scratch Then we use guided transfer learning in this work. Many studies show that transfer training gives good results without requiring a large number of samples. Transfer learning, as we have already mentioned, is the reuse of a pre-trained model to build a new, more advanced model. Below are the various architectures of the model, which were trained from scratch and the results of the application of transfer training.

We connect the necessary software and relevant data to work on the project. A work disk is mounted in Google for the aim of the project in the form of Jupiter. We determine the path to the database of sick and healthy plants in the mounted GoogleColab disk:

```
image_dir='/content/drive/My
Drive/ColabNotebooks/Agriculture/'
```

Here we upload the plant' images, as sick so healthy plants.

```
healthy_images_dir= image_dir
healthy_images_dir
unhealthy_images_dir=image_dir+'UnHealthy/'
```

The total number of healthy plants is 1591, and the sick is 465. Below are the results of changes in the accuracy of disease recognition in four models during training and testing at a resolution of 224 × 224 pixels (the input image of the disease). The following model results were obtained after applying various neural network architectures.

3. Results and discussions

Model 1:

** Architecture **

Conv2D -> MaxPool -> Conv2D -> MaxPool -> Dense -> Dense -> Sigmoid

** Optimizer **

SGD

Batch size = 32

Epoch = 20

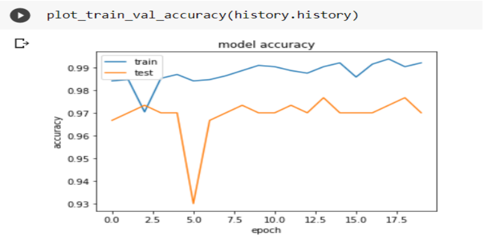
Total params: 188,049

Trainable params: 188,049

Non-trainable params: 0

Now let's train our model

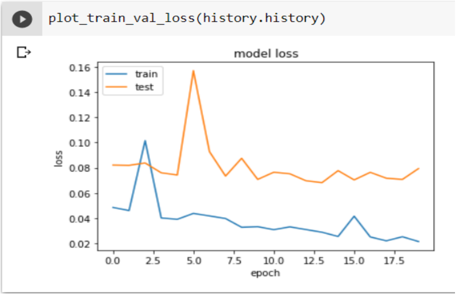
```
Epoch 1/20
55/55 [==============================] - 45s
814ms/step - loss: 0.0486 - accuracy: 0.9841 - val_loss: 0.0822 -
val_accuracy: 0.9667
……………………………………………………………………
………………….
Epoch 20/20
55/55 [==============================] - 45s
815ms/step - loss: 0.0217 - accuracy: 0.9920 - val_loss: 0.0794 -
val_accuracy: 0.9700
```



```
plot_train_val_accuracy(history.history)
```

Fig. 3. Accuracy model by stochastic gradient descent

We have obtained a graph of the accuracy of the model depending on the epoch on the training and test data. Similarly, below we will get a loss of model accuracy depending on the epoch on the training and test data.



```
plot_train_val_loss(history.history)
```

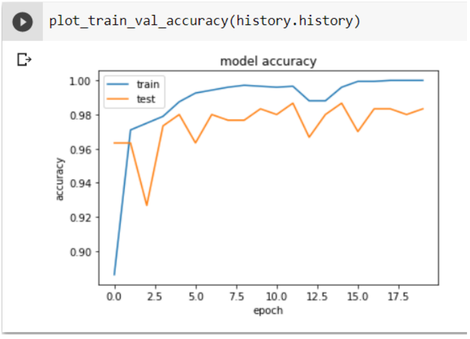
Fig. 4. Error model by stochastic gradient descent

Model 2: Dealing with Variations in GD Loss.
** Architecture **
Conv2D -> MaxPool -> Conv2D -> MaxPool -> Dense -> Dense -> Sigmoid

** *Optimizer* **

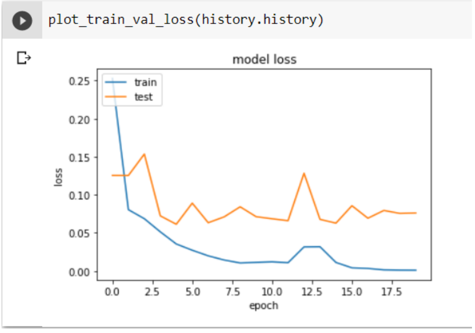
Adam (change)
Batch size = 32
Epoch = 20

Here is the result of model training process



```
plot_train_val_accuracy(history.history)
```

Fig. 5. Result of the model accuracy error with Adam optimizer



```
plot_train_val_loss(history.history)
```

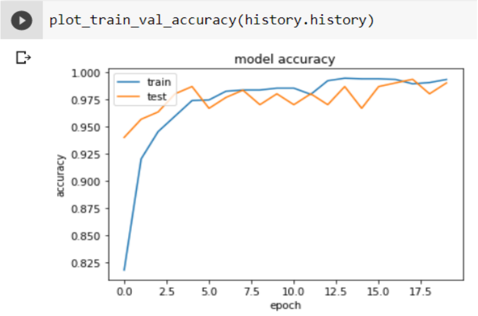
Fig. 6. Model bug with Adam optimizer

Model 3: Retraining. Regularization technology.

While inputting an image into a neural network with a size of 224 × 224 pixels for example, the network model of the neural network causes a problem with retraining. Let's train the model.

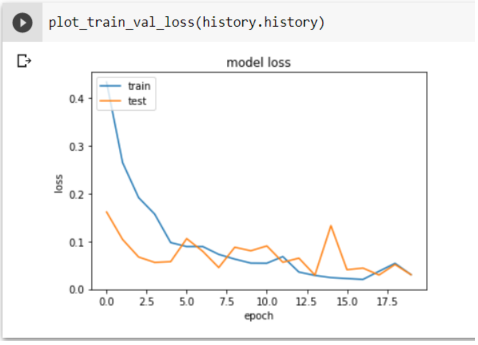
```
Epoch 1/20
55/55 [==============================] - 47s
831ms/step - loss: 0.5179 - accuracy: 0.7479 - val_loss: 0.1611 -
val_accuracy: 0.9400
………………………………………………………………………
…………………..
Epoch 20/20

55/55 [==============================] - 47s
860ms/step - loss: 0.0277 - accuracy: 0.9931 - val_loss: 0.0299 -
val_accuracy: 0.9900
```



```
plot_train_val_accuracy(history.history)
```

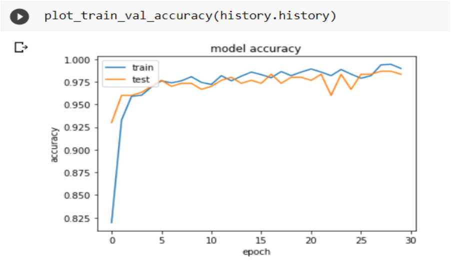
Fig. 7. Model accuracy with Adam optimizer and regularization



```
plot_train_val_loss(history.history)
```

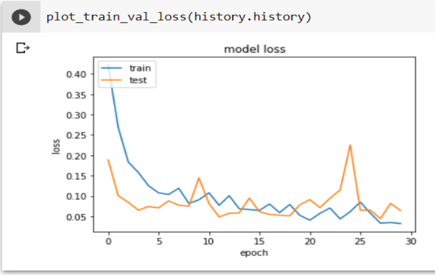
Fig. 8. Model loss with Adam optimizer and regularization

Model 4: Result after accuracy improving with more deep network



```
plot_train_val_accuracy(history.history)
```

Fig. 9. Accuracy improving in model with Adam optimizer and regularization



```
plot_train_val_loss(history.history)
```

Fig. 10. Loss improving in model with Adam optimizer and regularization

Now, using trained networks with the help of transfer learning, we will build up the model based on VGG16. Here are the model features:

** Architecture **

VGG16

** Optimizer **

Adam

Batch size = 64

Epoch = 20

from keras.applications.vgg16 import VGG16

Loading of previously trained weights

K.clear_session()

We will upload only one conv network

conv_model = VGG16(weights='imagenet',

include_top=False,

connected layers

input_shape=image_shape)

Downloading data from:

https://storage.googleapis.com/tensorflow/kerasapplications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5

58892288/58889256 [==============================] - 1s 0us/step

58900480/58889256 [==============================] - 1s 0us/step

Total params: 14,714,688

Trainable params: 14,714,688

Non-trainable params: 0

vgg16_model.compile(optimizer='adam',

loss = 'binary_crossentropy',

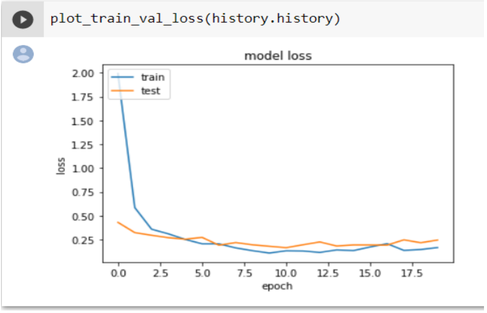
metrics = ['accuracy'])

history = vgg16_model.fit(

X_train_norm,

y_train, # prepared data

Train on 740 samples, validate on 300 samples



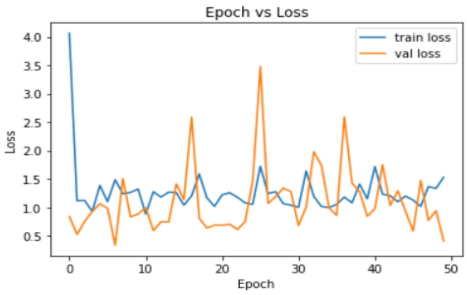
```
plot_train_val_loss(history.history)
```

Fig.11. Model loss in transfer learning, built based on VGG16

For the InceptionV3 model, the size of the input image is image_size = [224,224]. Our input image is a color image, then we need to add +3 to the input. We upload images of an already trained model with weights and exclude the last layers. So we don't want to train the model from scratch, but we train the model with new imaginary paddles. After smoothing the Inception V3 model, we use the Flatten layer. Next we can add layers to the output layer. We will add only one dense layer. We apply softmax to a one-dimensional vector. Let's build a model with a total number of parameters = 22 007 588. But in our case, we will train only 204804 parameters. Compile the model by choosing the number of epochs = 50.Here is the model training code:

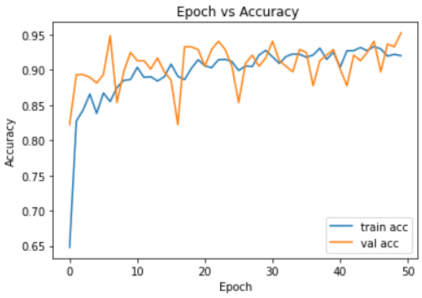
```
result=transfer_learning_model.fit_generator(
train_agumented_set,
validation_data=val_agumented_set,
epochs=50,
steps_per_epoch=len(train_agumented_set),
validation_steps=len(val_agumented_set))
```

After training the model at the end of the 50th epoch, we get the training accuracy = 0.93 and the verification accuracy = 0.94. The figures below show the results of the transfer learning model error, the model error on the training and test data sets, as well as the model accuracy on the training and test data sets.



```
<Figure size 432x288 with 0 Axes>
```

Fig. 12. Loss of accuracy in the model during transfer training based on InceptionV3



```
<Figure size 432x288 with 0 Axes>
```

Fig.13. Accuracy in the transfer learning model based on InceptionV3

It is always better to use a transfer learning model because it can train deep neural networks with less data, and also gives good accuracy compared to CNN from scratch.

4. Conclusion

Modeling and forecasting of agricultural tasks are studied in this paper. One of the most difficult and promising tasks in this area is the recognition of plant diseases, which greatly affects crop yields, the profits of farmers and agricultural producers and the country's economy as a whole. The process of building models from scratch and using transfer learning based on data of various data volumes is studied. Models were obtained by using IVGG16 and InceptionV3 for new data. There is studied issue of retrained models based on regularization technologies. Various models have been obtained for a certain type of plant, namely for pears.

References

[1] Azadbakht M, Ashourloo D, Aghighi H, Radiom S, Alimohammadi A Detection of wheat leaf rust at the canopy scale at different levels using machine learning methods. Comput Electron Agric 156: 119–128 (2019).

[2] Bai X, Li X, Fu Z, Lv X, Zhang L. A fuzzy clustering segmentation method based on grayscale information in the neighborhood to determine images of cucumber leaf spot disease. Comput Electron Agric 136: 157–165 (2017).

[3] DeChant C, Wiesner-Hanks T, Chen S, Stewart EL, Yosinski J, Gore MA, Nelson RJ, Lipson H . Automated identification of corn plants affected by northern late blight from field images using deep learning. Phytopathology 107 (11): 1426–1432. (2017).

[4] Hongfei Zhu, Lianhe Yang, Jianwu Fei, Longgang Zhao, Zhongzhi Han, Recognition of carrot appearance quality based on deep feature and support vector machine, Computers and Electronics in Agriculture, 10.1016/j.compag.2021.106185,186, (106185), (2021).

[5] Hinton GE, Osindero S, Teh YW Algorithm of fast learning for the networks with deep learning. Neural Comput 18 (7): 1527–1554 (2006).

[6] Mohanty SP, Hughes DP and Salathé M (2016) Using Deep Learning for Image-Based Plant Disease Detection. *Front. Plant Sci.*7:1419. doi: 10.3389/fpls.2016.01419.

[7] Robert D Clark, Putting deep learning in perspective for pest management scientists, Pest Management Science, 10.1002/ps.5820,76, 7, (2267 -2275), (2020).

[8] L. Breiman. Random forests. Machine learning, 45(1):5–32, 2001.

[9] D. Cire¸san, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. Arxiv preprint arXiv:1202.2745, 2012.

[10] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Advances In Neural Information Processing Systems, pages 2172–2180, 2016.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in Neural Information Processing Systems, pages 2672–2680, 2014.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

[13] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. Science, 350(6266):1332–1338, 2015.

[14] J¨urgen Schmidhuber. Deep learning in neural networks: An overview. Neural Networks, 61:85–117, 2015.

[15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1–9, 2015.

Baratbek Sabitov. Associated Professor, Kyrgyz National University named after J. Balasagyn, Department of Applied Informatics, Kyrgyzstan, Bishkek. Received the PhD degree in Physics&Math. His interest is on Artificial Intelligence, Machine Learning.

Saltanat Biibosunova Assosiated Professor, Arabaev Kyrgyz State University, Department of Applied Informatics, Kyrgyzstan, Bishkek. Received the PhD degree in Economics and M.S. degrees in Applied Informatics, respectively. Her research interests are on Informaton System and Technology, Data Science, Cyber Security in Education.

Altyn Kashkaroeva Associated Professor, Arabaev Kyrgyz State University, Department of Applied Informatics, Kyrgyzstan, Bishkek. Received the PhD degree in Social Sciences. She got the Docent diploma on “Applied Informatics”. Her interest is on Information Systems and technologies in various activities.

Bolotbek Biibosunov Professor, Doctor of Physics&Math, Doctor of Technical Sciences, Arabaev Kyrgyz State University, Department of Applied Informatics, Kyrgyzstan, Bishkek. He is the chair of “Applied Informatics” Department. His research interest is Information Technologies, Mathematical Modelling.