Why Plant Pathology and objectives for mode

Plant Pathology is not only a serious threat to our food security, but also a trouble for smallholder farmers who live on crops, the largest proportion of hungry people around 50 percent worldwide. However, current crop disease diagnosis by human scouting is time-consuming and inefficient.

Multiple ways have been implemented to prevent plant loss by diseases. First is Computer-Vision based models, although it increases efficiency, the accuracy is decreased due to multiple variances in symptoms.

Second is training a model by using images of the dataset, it is also the method we use in the project. So, the model aimed at accurately identifying the healthy leaf or different diseased category through the test dataset, distinguishes different diseases on more than one leaf, disposes depth perception such as angle and brightness of the leaf, and combines professional knowledge in identification, guidance and explanation for computer vision models to retrieval for relevant attributes during learning.

Intro to data

We are given multiple apple leaf images, and the goal is to distinguish the leaves which are healthy, which are infected with apple rust, which with scab, and which have more than one disease.

Before preprocessing data, we have train.csv including the sheet identifying apple leaves into four categories with image id, Images' folder contains all training and testing images with jpg format, and test.csv including 1821 images for testing data at the end.

When we preprocess the image data, first resizing images to 256*256 pixels, and performing both disease prediction and model optimization based on these downscaled images. Then we split our dataset into 2 parts, 80 percent of the whole dataset used for training, and 20 percent for validation data. What's more, we also split the leaves into 4 different categories including healthy, scab, rust and combination under the train folder and valid folder.

When it comes to data augmentation, images are rescaled, rotated, flipped and changed brightness by ImageDataGenerator. Also, we apply our image augmentation to training and validation datasets. Thus finished our data processing.

Model Selection

Since our goal is to identify a given image from the testing dataset to see whether it is a diseased leaf or a healthy leaf, we focus on different classification models.

Firstly, we choose CNN(Convolutional Neural Network) architecture as our model. CNN is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. It is the most commonly applied model to analyze visual imagery. Usually, CNN models consist of convolutional layers, pooling layers, flatten layers, dense layers and an output layer. In order to see whether we have an overfitting problem, we build two CNN models, one has a dropout layer, the other one doesn't. The Dropout layer randomly sets input units to 0 with a frequency of rate at each step during training time, which helps prevent overfitting. After training two models with preprocessed training dataset and validation dataset, CNN model with dropout layer has the best accuracy, which is about 79%.

The CNN model’s accuracy is good, but could we improve our performance if we choose some other classification models? After some research, we decided to use VGG16 architecture. VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition”. The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. The architecture of VGG16 is similar to CNN model but much more complex. Number 16 refers that VGG16 has a total of 16 layers that has some weights. Since Keras library has a pre-trained VGG16 model, we use it directly without building by ourselves. We do the same step as what we do to our CNN model, and finally we got the best accuracy of VGG16 which is about 65%.

Test Result

We perform automatic hyper-parameter tuning with Keras Tuner and Tensorflow 2.0 to boost accuracy on our CNN model，change activation method(relu, sigmoid), the number of filters(32, 64) , and dense units from 32 to 512.

After hyper-parameter tuning, the best model accuracy is lower than the default model accuracy, from 0.7996 to 0.7341. That means the customized parameters and values we have chosen for tuning should be continuously optimized.