Foliar Diseases in Apple Trees

MULTI-CLASS IMAGE CLASSIFICATION

Adrien Clay | Springboard Data Science | 04/19/2021

Problem

The current process for diagnosing diseases in apple orchards is manual scouting by humans. This is both time consuming and expensive. That said, there are severe limitations in the past on computer vision giving useful insight into the diseases that can be found, specifically in relation to new varieties, or extreme variation in the visual symptoms present in a single disease. A few examples are as follows:

- Leaf Color
- Age of Infected Tissues
- Non-uniformity across image backgrounds
- Different light illumination

OBJECTIVE

The main objective is to develop a model that can accurately classify a given leaf image from the test dataset to a particular disease category, and to identify an individual disease from multiple disease symptoms on a single leaf image.

Example of a Healthy Leaf

The Data

The data consists of 18,000+ images, spread across twelve different possible classes of disease. The classes of diseases are as follows:

- Health
- Scab Frog Eye Leaf Spot Complex
- Scab
- Complex
- Rust
- Frog Eye Leaf Spot
- Powdery Mildew
- Scab Frog Eye Leave Spot
- Frog Eye Leave Spot Complex
- Rust Frog Eye Leaf Spot
- Powdery Mildew Complex
- Rust complex

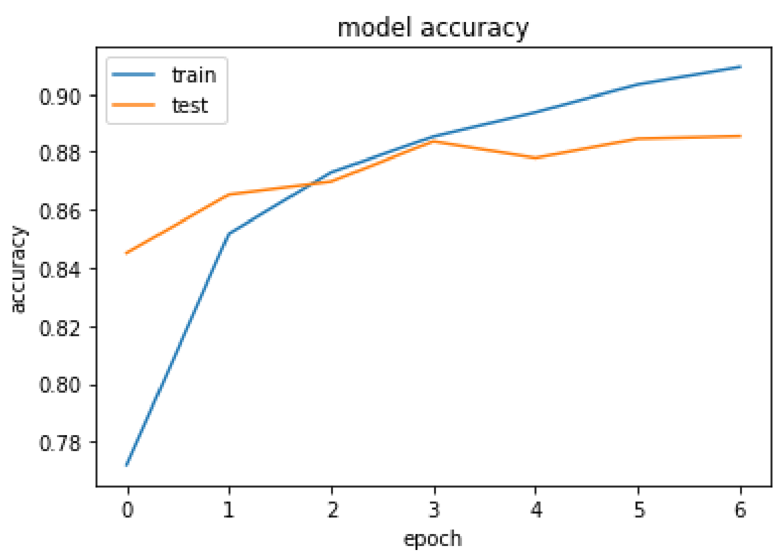
The Model

For this project, I utilized a transfer learning CNN model known as **EfficientNetB3.**

The pipeline is as follows:

- Load Images (400x400, 3 color channels)
- Train/Validation Split
- Data Augmentation
- Global Average Pooling (2d)
- Dropout
- Output

The model is then compiled and fit to the training data over twenty-five epochs. It utilizes a cosine decay as a learning rate scheduler. Finally, it uses a Keras callback to an early stopping mechanism so the model wouldn't be overfit to the data.



Other Modelling Work

Although it is not shown in the associated GitHub repository, many other transfer learning models were evaluated throughout this process. A list of these is as follows:

- InceptionV3
- VGG16
- EfficientNetB0
- ResNet18

While these models performed adequately, it became clear after a few trials that EfficientNetB3 was hitting consistent accuracy and consistent loss reduction on a downward slope without any hiccups, whereas the other models would bounce around a bit, or were just too slow to train. The overall run time for training EfficientNetB3 with all additions was around 2.5 hours.

Recommendations for Use:

In the end, the model was only hitting at around 72% accuracy when tested on 2000 extra images that were held out of training. While this is not awful, it is something to take with a grain of salt. If a farmer assumed there to be one disease plaguing the entire crop, it would be wise to have graphical confirmation over a range of 10 – 20 images to see if a prediction made itself more obvious than others.

METRICS:

	precision	recall	f1-score	support
complex	0.49	0.57	0.53	165
frog_eye_leaf_spot	0.79	0.76	0.78	298
frog_eye_leaf_spot complex	0.00	0.00	0.00	5
healthy	0.65	0.95	0.77	436
powdery_mildew	0.90	0.94	0.92	173
powdery_mildew complex	0.25	0.09	0.13	11
rust	0.66	0.90	0.76	202
rust complex	0.00	0.00	0.00	8
rust frog_eye_leaf_spot	0.00	0.00	0.00	15
scab	0.91	0.60	0.72	614
scab frog_eye_leaf_spot	0.35	0.11	0.16	57
scab frog_eye_leaf_spot complex	0.00	0.00	0.00	16
accuracy			0.73	2000
macro avg	0.42	0.41	0.40	2000
weighted avg	0.73	0.73	0.71	2000

Current Issues

When looking at the metrics table, I noticed a few things right off the bat. One of those things is that it appears to have no problem identifying the more obvious diseases (or lack thereof in the case of 'healthy')

Healthy has a precision of 65% and a recall of 95%, and subsequently powdery mildew has a precision of 90% and a recall of 94%. I found this interesting because both of those classes are very distinct in the way that they look.

Many of the healthy pictures are extremely green.

Powdery Mildew is a pale green, which is easily distinguishable.

Frog Eye Leaf Spots - Similar to the healthy image, lots of green tones here. The only distinguishing factors for the model would be the 3 spots with brown information.

Rust: Once again, the model would have to be able to distinguish these brown tones and their pattern from the above Frog Eye Leaf Spot tones.

Future Improvements

As of now, the challenge of a problem like this is apparent. I almost did not realize how in depth the issue is until looking over the images for this documentation – the slight visual difference in leaves that have one disease (let alone two at the same time) provides a difficult playing field for the model to be able to distinguish one disease from the next. I believe a lot of this relies in similarity in color tones. One thing I noted about the model is that it did not seem to have a hard time telling if there was a disease or not, just exactly which one, and sometimes simply did not provide any useful insight on a leaf having two diseases at a time.

In the future, I think that it would be useful to have images that focus less on the leaves, and more on the spots that contain disease, because I believe the model is extremely aware of the green tones in each image. There is a possibility of overfitting to healthy and powdery mildew images, and they appear to contain the most green tones per pixel of all the photographs. If someone had time to crop each image to the portion of the leaf that is in question, I would be interested to see how the model performed on images when they were subsequently zoomed out, as I feel the model would be trained to look for the color tones in question, and the data that is unique to each disease both in texture and appearance.