Diseases in Apple Trees

Multi-Class Image Classification
Springboard Data Science
Adrien Clay

Frog Eye Leaf Spot Example

The Problem

The current process for diagnosing disease in apple orchards is time consuming, manual labor that becomes extremely expensive over time.

•••

The objective for the problem at hand is to develop a model that can accurately classify given images over a range of diseases and varying photographic conditions.

The Data

The data consists of 18,000+ images ranging over 12 conditions. In all there are 6 unique diseases, and the other six classes are comprised of a combination of the six unique diseases.

Data Wrangling

The images were loaded as 400px by 400px with three color channels, and passed through a data augmentation layer to give the data more variation.

•••

The images were in one folder and needed to split into directories by class to be loaded into a Tensorflow input pipeline (tf.data.Dataset)

Data Wrangling

A train.csv was provided. It contained image filenames and associated labels. Using this, I was able to apply a for loop that would automatically parse the images into separate folders by class.

•••

Additionally, 2000 images were held out of the training set to be used as a true test set down the line.
After this, the images were ready to be loaded into a Tensorflow Dataset.

EDA

Images were plotted out by class, and a subsequent KMeans clustering was performed on the images to see if images could be grouped by visual similarities.

This process was really a testament to how difficult the data science problem is - The images seemed to cluster not by disease, but by photographic characteristics.

EDA

Cluster 5




•••

Cluster 6




EDA

As for the images on the previous slides, one thing I personally noticed is that the algorithm appeared to cluster by similar color tones, and photographic attributes rather than anything in relation to disease. This was red flag number one for the problem at hand.

Modelling

Many transfer learning models were evaluated including:

- InceptionV3
- VGG16
- ResNet18
- EfficientNetB0

...

Modelling

The final model used was EfficientNetB3 - along with a data augmentation layer, a global average pooling layer, dropout, and a final dense output layer to 12 possible output classes.

•••

To a certain degree, this model was really only chosen because of the results - which topped all other models chosen both in validation loss and in general accuracy.

Results

The model performed very well in training, with an accuracy of 88%+, and a validation loss reduced from over 1.0 to as little as .3626, although admittedly I had hoped for lower.

•••

However, when it came time to test the model on the 2000 test images, the model was hitting at 73% accuracy, with an F-1 Score of .71

Results

The most important take away that I found from the evaluation metrics was the performance on Healthy images, and powdery_mildew images.

These images are distinct from the rest in they their color tones are extremely consistent, and that they really just have a LOT of green. The recall for healthy images was .95 for healthy plants, and .94 for powdery mildew.

Recommendations

Based on the results, I would recommend the client consider a quality control process to control the conditions of the photographic attributes to the best of their ability.

...

What I found is that the attributes of the photograph seem to matter more to the model than the diseases themselves - which is unfortunate because that's hardly the point of the problem.

Future Improvement

Coming to the end of this problem, I believe that this is a problem for object detection within photographs, which would be much more suited to pick up on the nuances of each disease, its color tones, and texture, rather than looking at each image as a whole.

...

This would also reduce the need for extreme consistency across images given to the model, as the model would be more focused on the disease found rather than the overall attributes of the images.