Classifying apple leaf disease with CNN and transfer learning.

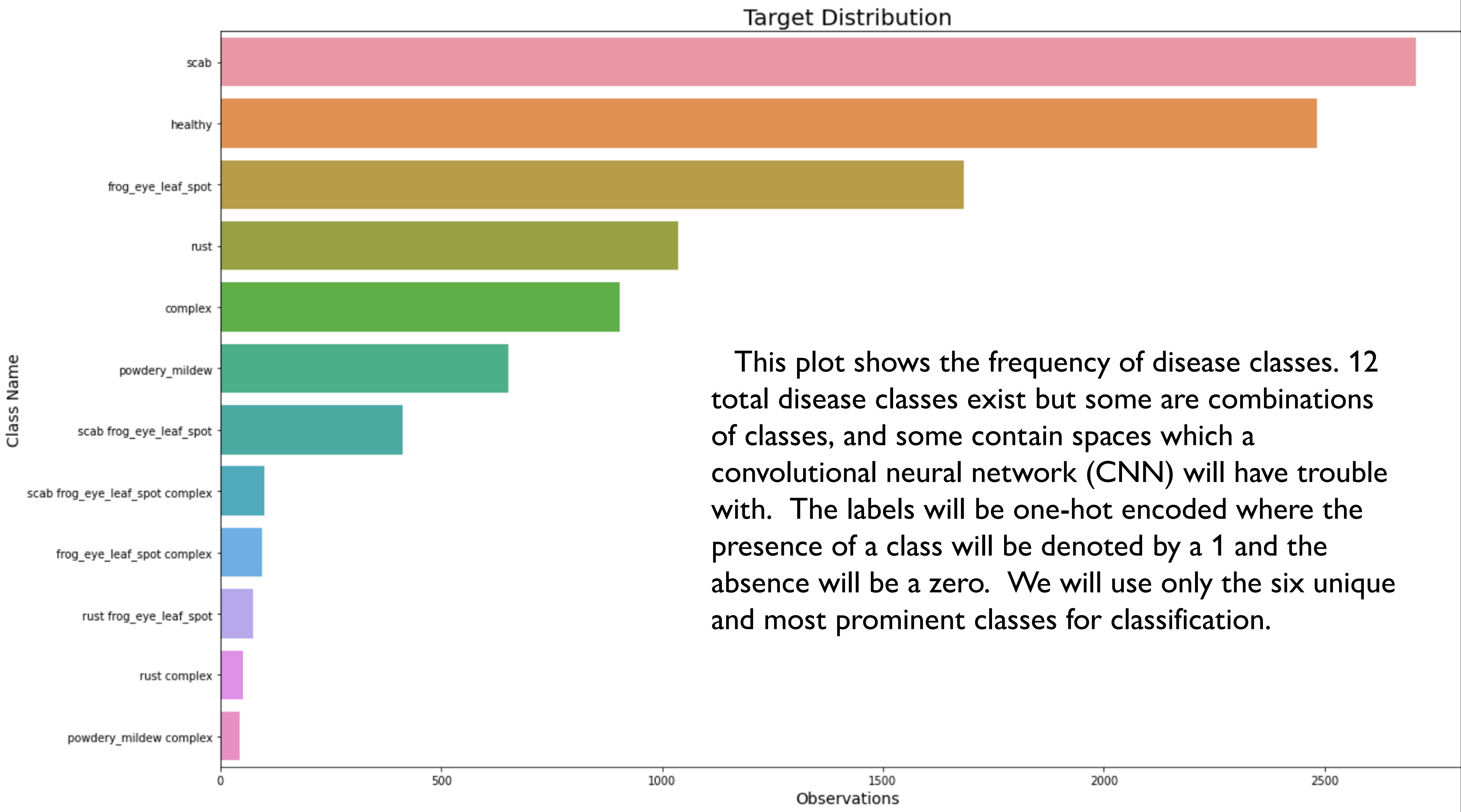
by Brian Ingram

Why is this important?

- Apples are one of the most important temperate fruit crops.
- Diseased plants can have a strong impact on harvest yield and quality.
- Apple leaf disease is found by hand which is time consuming and costly.
- Plants are people too! Plants are living things and improving plant life is important.

The data

- This data set is from the Plant Pathology 2021 – FGVC8 code competition.
- The training set contains about 18,600 RGB Fine Grained Visual Cat images of apple leaves.
- There is also a corresponding csv file that contains the image ID and labels which are target disease classes.
- Memory limitations will not allow 18,000 images to be used. A smaller training set of 10,240 images will be trained and processed.
- A test set will be made from the remaining training set (2559 images)



This plot shows the frequency of disease classes. 12 total disease classes exist but some are combinations of classes, and some contain spaces which a convolutional neural network (CNN) will have trouble with. The labels will be one-hot encoded where the presence of a class will be denoted by a 1 and the absence will be a zero. We will use only the six unique and most prominent classes for classification.

Disease classes - Healthy

These leaves should exhibit no evidence of disease exclusive of the other classes and serve as a baseline.

Disease classes - Scab

Scabs are a fungal disease that forms pale yellow or olive-green spots on the upper surface of leaves. Dark, velvety spots may appear on the lower surface. Severely infected leaves become twisted and puckered and may drop early in the summer.

Disease classes - Rust

Circular, yellow spots (lesions) appear on the upper surfaces of the leaves shortly after bloom. In late summer, brownish clusters of threads appear beneath the yellow leaf spots or on fruits and twigs. The spores associated with the threads the leaves during wet, warm weather.

Powdery mildew is a fungal disease that affects a wide range of plants and is one of the easier plant diseases to identify. Infected plants display white powdery spots on the leaves and stems. As the disease progresses, the spots get larger and denser, and the mildew may spread up and down the length of the plant.

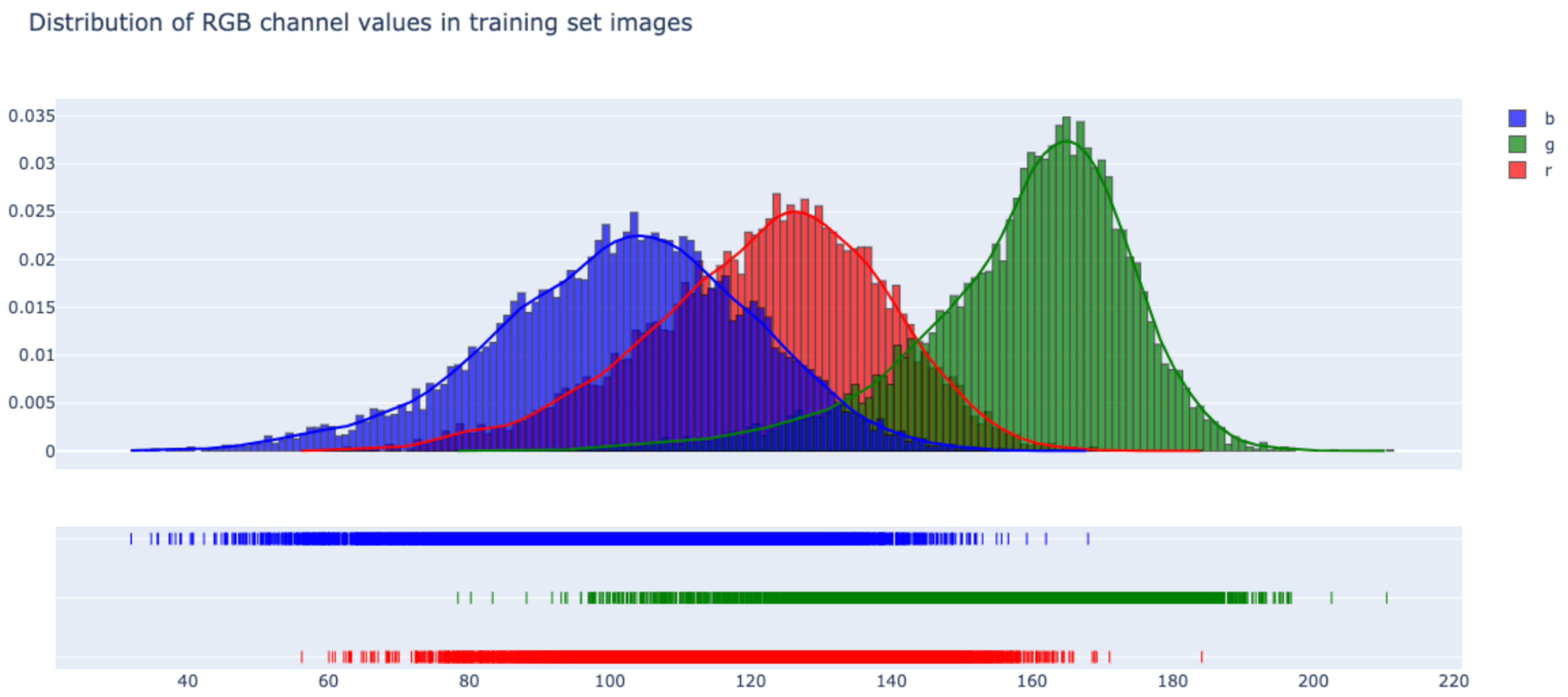
Disease classes – Powdery Mildew

Disease classes – Frog eye leaf spot

First, small purple spots form on the leaves. These spots gradually enlarge and eventually develop into lesions with a light tan interior, surrounded by a dark purple perimeter. Heavy infections of frog-eye leaf spot can cause leaves to turn yellow and drop.

Disease classes - Complex

Unhealthy leaves with too many diseases to classify visually will have the complex class and may also have a subset of the diseases identified.


Distribution of RGB channel values in training set images
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
b
g
r
40
60
80
100
120
140
160
180
200
220

My CNN

Even with 1,391,014 parameters, this neural network does not have enough parameters to classify over 10,000 images of apple leaves. On to **transfer learning.**

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 224, 224, 8)       80
_________________________________________________________________
batch_normalization (BatchNo (None, 224, 224, 8)       32
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 112, 112, 8)       0
_________________________________________________________________
dropout (Dropout)            (None, 112, 112, 8)       0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 110, 110, 16)      1168
_________________________________________________________________
batch_normalization_1 (Batch (None, 110, 110, 16)      64
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 55, 55, 16)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 55, 55, 16)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 53, 53, 32)        4640
_________________________________________________________________
batch_normalization_2 (Batch (None, 53, 53, 32)        128
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 26, 26, 32)        0
_________________________________________________________________
dropout_2 (Dropout)          (None, 26, 26, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 21632)             0
_________________________________________________________________
dense (Dense)                (None, 64)                1384512
_________________________________________________________________
dense_1 (Dense)              (None, 6)                 390
=================================================================
Total params: 1,391,014
Trainable params: 1,390,902
Non-trainable params: 112
```

What is transfer learning?

- Transfer learning is a supervised learning technique that reuses parts of a previously trained model on a new network tasked for a different but similar problem.
- The models we will explore have been trained on ImageNet which is an image database organized according to the WordNet hierarchy in which each node of the hierarchy is depicted by hundreds and thousands of images. This means the model could be used on other similar problems.

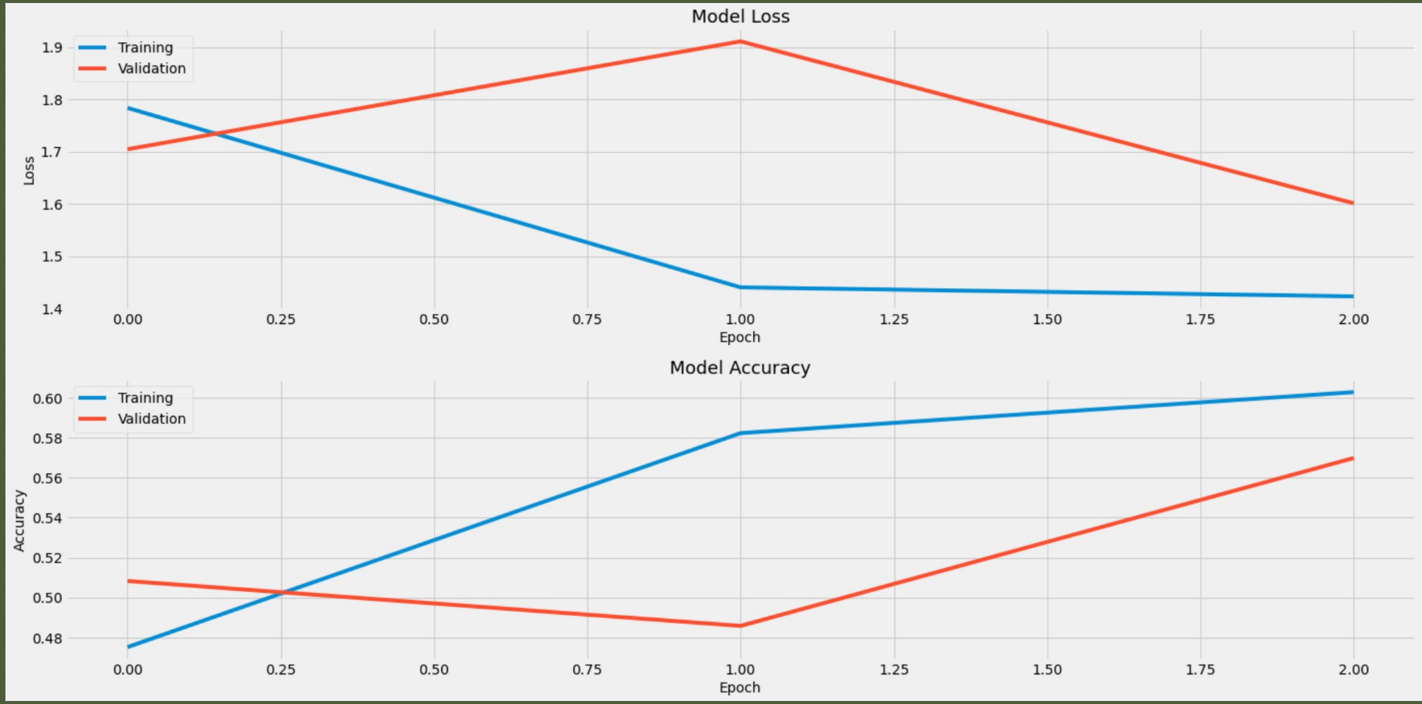
VGG19

Only 3×3 convolutional layers stacked on top of each other in increasing depth. Reducing volume size is handled by max pooling. Long time to train and large network architecture weights. May need more than 3 epochs.

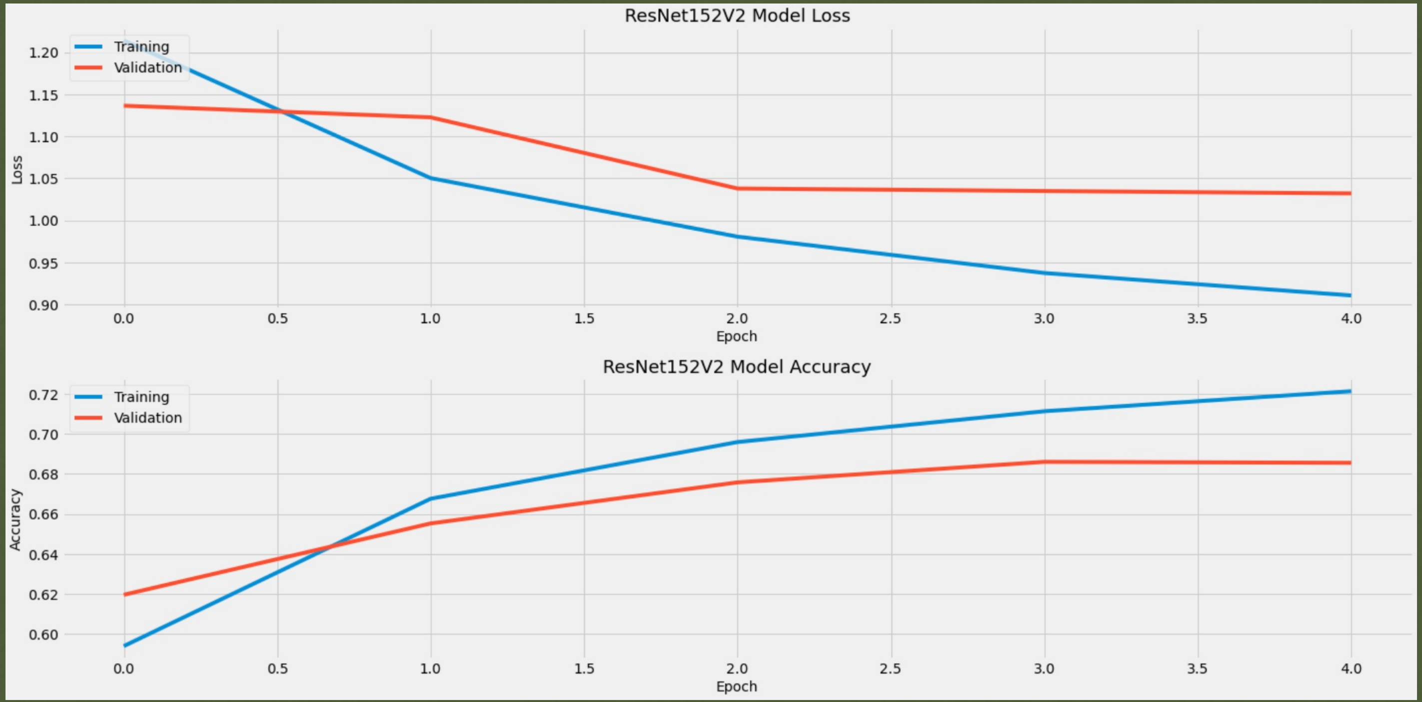
Model: "VGG19"

Layer (type)	Output Shape	Param #
input_3 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv4 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv4 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv4 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
global_average_pooling2d_3 (	(None, 512)	0
dense_8 (Dense)	(None, 512)	262656
dense_9 (Dense)	(None, 6)	3078

Total params: 20,290,118
Trainable params: 265,734
Non-trainable params: 20,024,384



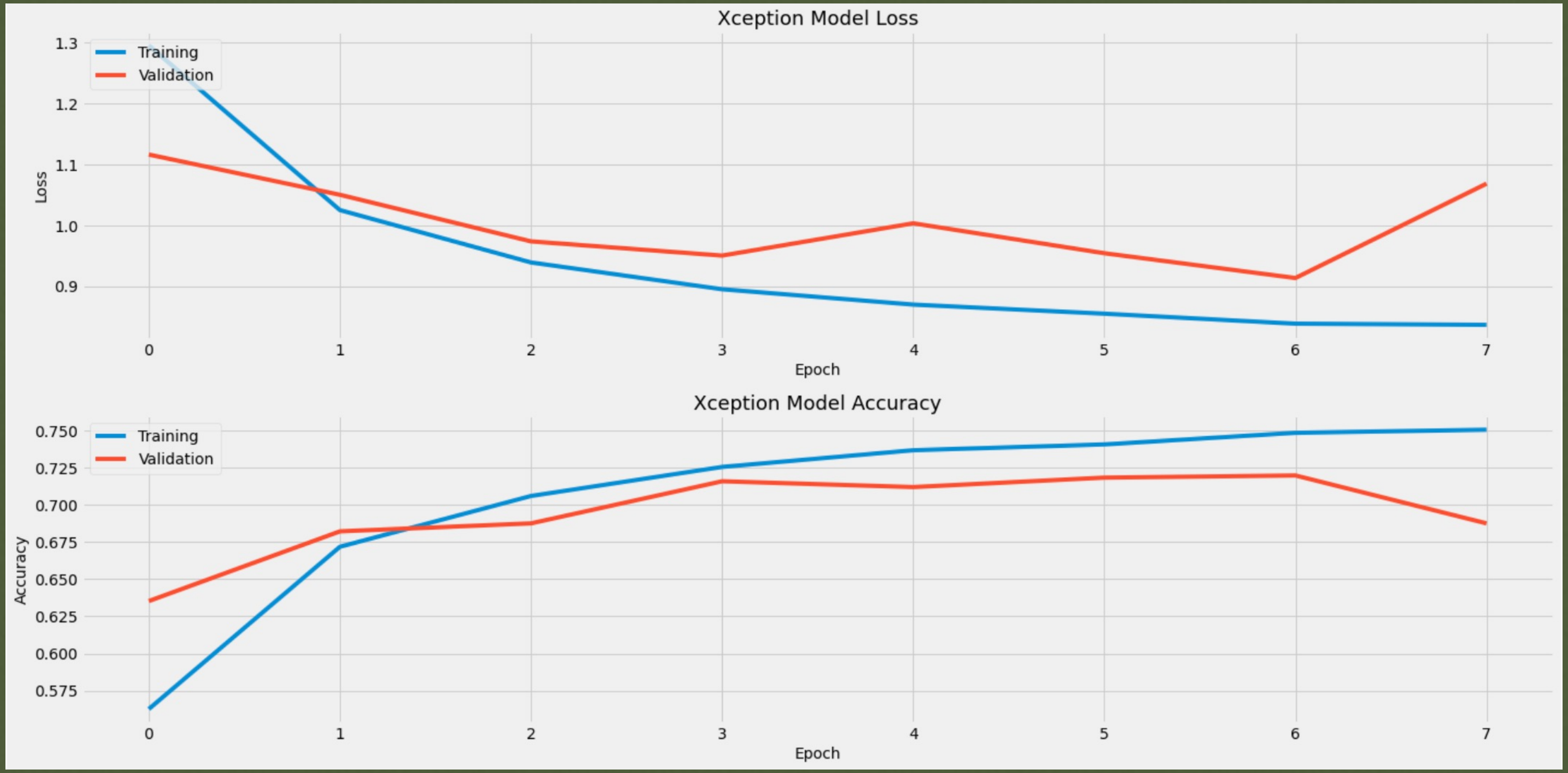
ResNet152V2



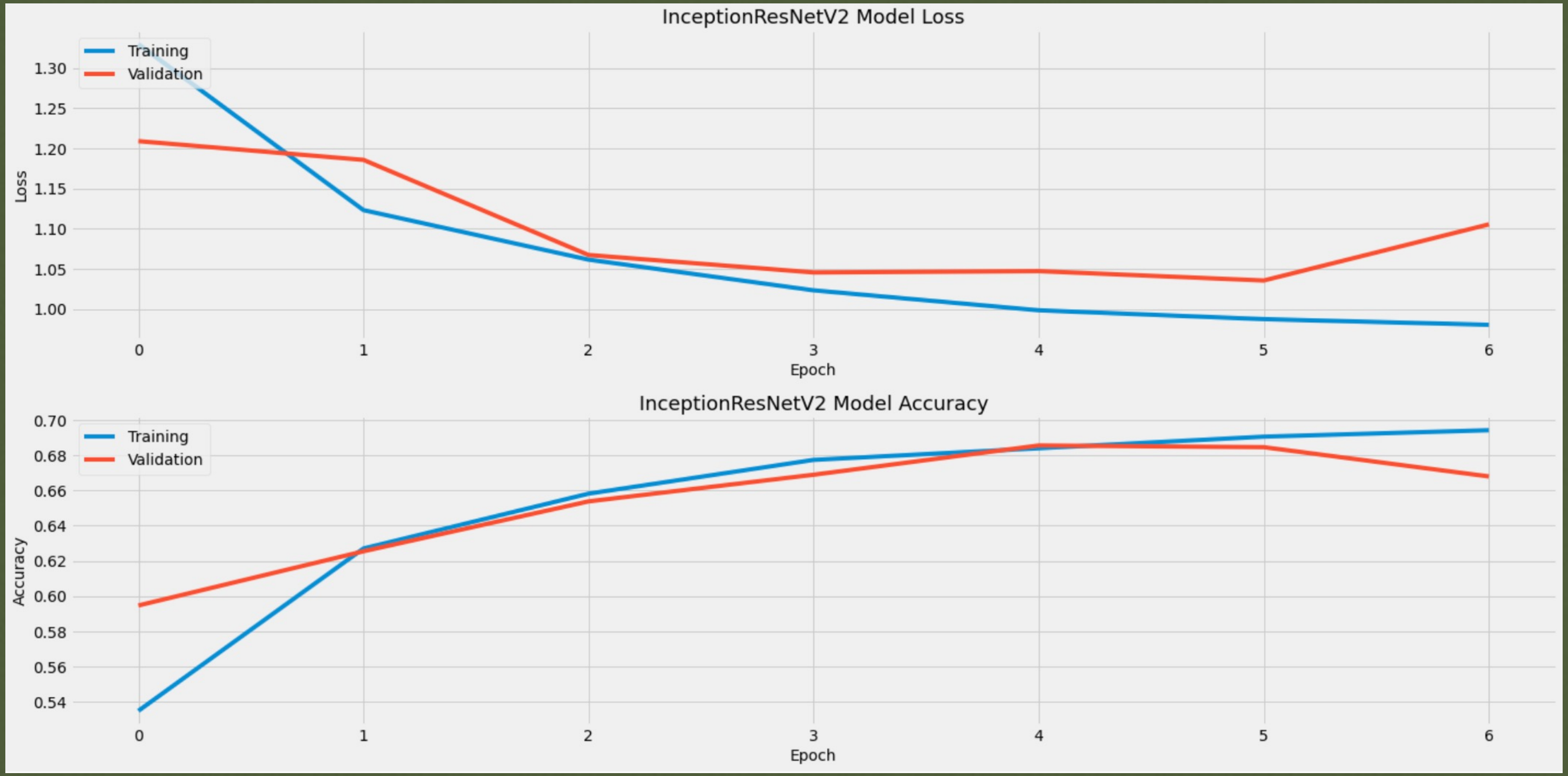
ResNet relies on micro-architecture modules called network-in-network architectures that are much deeper than VGG19 and the model size is smaller. ResNet stands for the use of residual modules that can train extremely deep networks with stochastic gradient descent.

Xception

Xception, which was developed by the creator of Keras, François Chollet, is an extension of the Inception architecture which replaces the standard Inception modules with depth-wise separable convolutions.



InceptionResNetV2



InceptionResNetV2 is a convolutional neural architecture that builds on the Inception family of architectures but incorporates residual connections replacing the filter concatenation stage of the Inception architecture.

Transfer learning metrics summary

Model	Accuracy	Precision	Recall	f1 Score	Total Parameters	Epoch time (s)
Sequential CNN	0.334	N/A	N/A	N/A	1,391,014	5
VGG19	0.589	0.676	0.518	0.587	20,290,118	429
ResNet152V2	0.709	0.812	0.597	0.688	59,383,814	425
Xception	0.696	0.78	0.6	0.678	21,913,646	25
InceptionResNetV2	0.692	0.774	0.619	0.688	55,126,758	38

The figure above depicts accuracy, precision and recall for the CNN models on the test set of images after training on the training set. The ResNet152V2, Xception and InceptionResNetV2 perform very similarly on the test data. The VGG19 model did not perform as well. Xception and InceptionResNetV2 fit very quickly where ResNet152V2 and VGG19 were sluggish. All models were run on a TPU with high-RAM on Google Colab.

What could be improved?

- There was not enough RAM available to read all the images provided in the initial Kaggle training set into Google Colab. A larger data set could provide more accuracy and precision during training.
- Leaf color and leaf morphology, non-uniform image background, and different light illumination during imaging may require more image pre-processing before modeling.
- PyTorch is more advanced with some of the object-oriented programming functions that can be utilized which could improve training efficiency and modeling accuracy.

Uses for these models

The ResNet152V2, Xception or InceptionResNetV2 models can be used to identify the 5 most popular classes of apple leaf disease with respect to healthy leaves with close to 70% accuracy.

These models can reduce labor costs and increase profits due to increased efficiency during harvest and cultivation.

These models can also be used on other cases of plant pathology improving the health of various types of plants.

These models can be used as a basis for development of models that can predict plant disease before it begins.