Listki

Mateusz Garczyński, Sonya Pobiedimska, Zuzanna Twardowska, Wiktor Wołek, Dawid Wysocki

Introduction

The aim of the project

The purpose of this project is detecting apple trees' diseases from the photos of their leaves.

Our goal was not only to detect if any disease is present but also to be able to categorize all of the known diseases.

On success, our tool would be helpful in early detection of diseases in the orchards and could prevent yield decreases without time-consuming manual work.

Dataset

Our dataset consists of the photos of leaves representing both different diseases and healthy leaves.

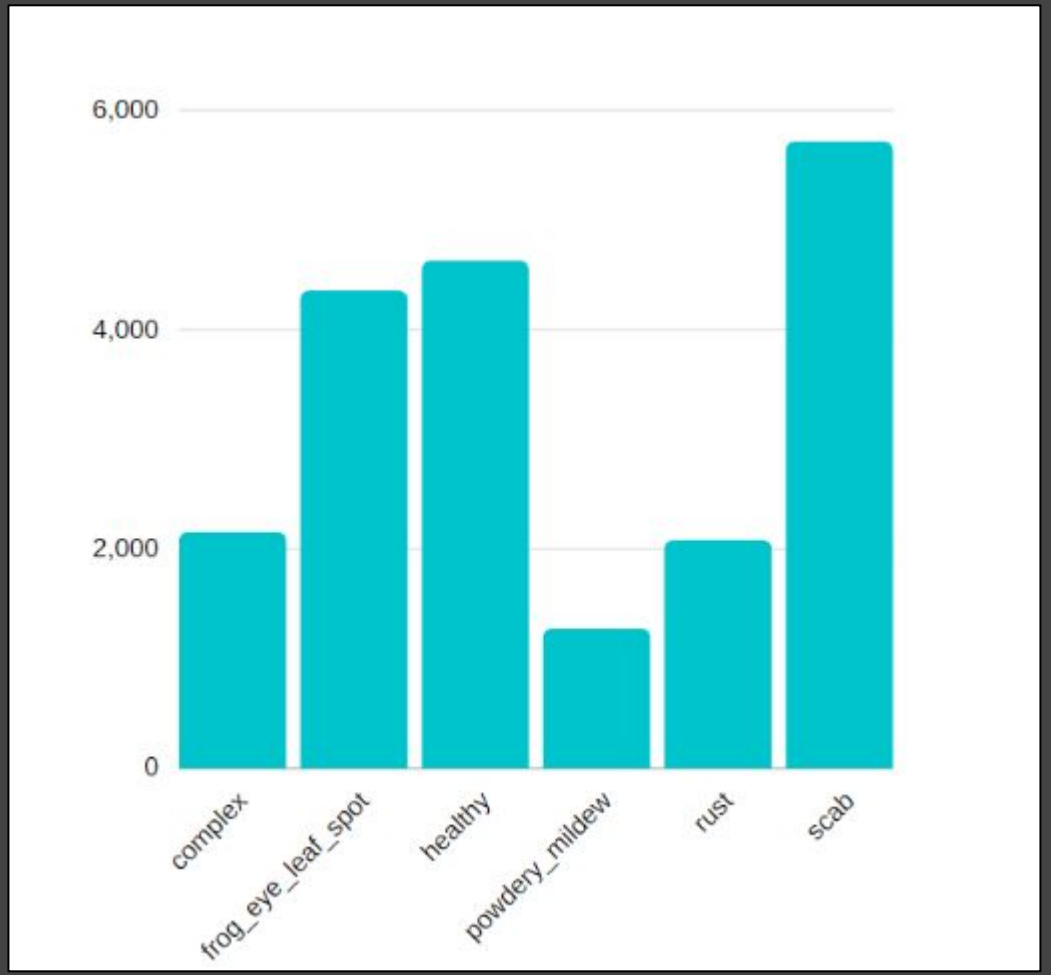
Every photo is categorized into one the six groups.

There are four groups for different diseases, one for the healthy leaves and one that is named ‘complex’.

The ‘complex’ group consists of leaves that present too many diseases to identify them correctly.

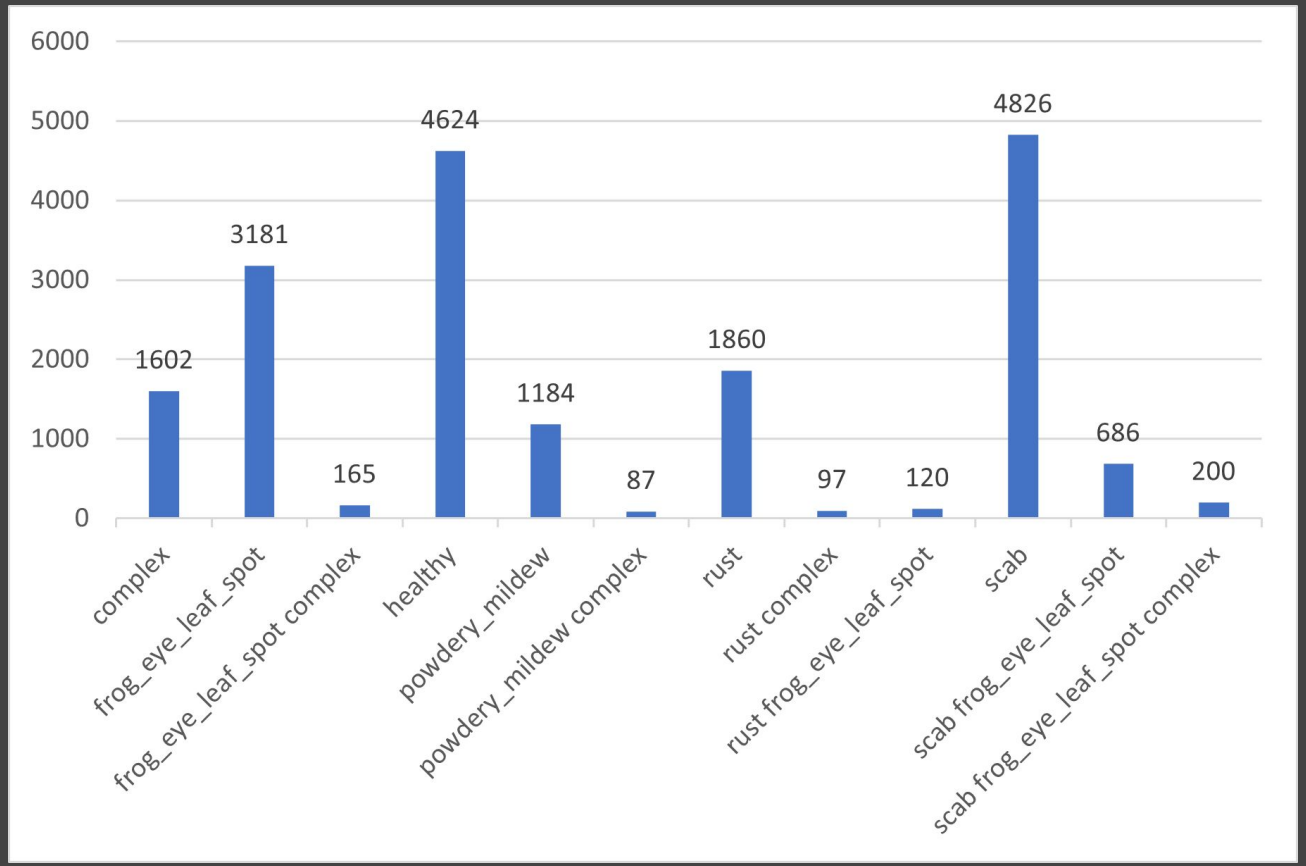
Dataset Analysis

The quantity of photos in each group



Dataset Analysis

The quantity of photos in each group combination



Problems we have encountered

The dataset is really big - we have tried different environments/storages, however, finally we decided to store data and train models on our PCs.

Many leaves were present in each of the photos while only one of them should be examined.

Training models took too much time to enable us to check all of our ideas thoroughly - we had to focus on the best ones.

Our model

Structure

```
model = keras.Sequential([

    input_layer,

    intercept_layer,

    layers.GlobalAveragePooling2D(),

    layers.Dense(classes_amount, kernel_initializer = 'uniform',
    activation = "softmax")

])
```

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input (Sequential)          (None, 256, 256, 3)       0

 inception_v3 (Functional)   (None, None, None, 2048)  21802784

 global_average_pooling2d (G  (None, 2048)             0
 lobalAveragePooling2D)

 dense (Dense)               (None, 12)                24588

=================================================================
Total params: 21,827,372
Trainable params: 24,588
Non-trainable params: 21,802,784
_________________________________________________________________
```

Input layer

Images were preprocessed before entering InceptionV3.

Preprocessing consisted of randomly flipping, rotating and zooming the images.

It allowed us to increase the amount of images for training and avoid overfitting the model.

Images fed to InceptionV3 were in 256 by 256 resolution with all rgb colors in sRGB color space.

InceptionV3 as base model

InceptionV3 is a convolutional neural network that was introduced by Google to allow deeper networks while keeping the amount of parameters not too large.

It is used in object detection and image analysis.

It was chosen due to its relatively low amount of parameters (under 25 million), which increased training speed and due to its high accuracy with minor tweaks.

Global average pooling layer

The purpose of global average pooling is to create one feature map for every category of classification task, instead of using fully connected layers.

Then we take the average value of each map and feed it into the next layer. It helps enforce correspondence between feature maps and categories.

What’s more, because this layer has no parameters to optimize, this layer is resistant to overfitting.

Fitting

Loss function: categorical cross entropy - fits with multiple label classification provided in a one-hot representation which is found in our problem.

Optimizer Adam with default learning rate (0.001) was yielding better results faster than RMSProp and NAdam.

Results

Accuracy on validation set

Validation set consists of 3726 images randomly chosen from the dataset.

The final score is 0.68.

This runs in pair with accuracies on the training dataset, so the problem of overfitting doesn’t occur in our case.

Confusion matrix

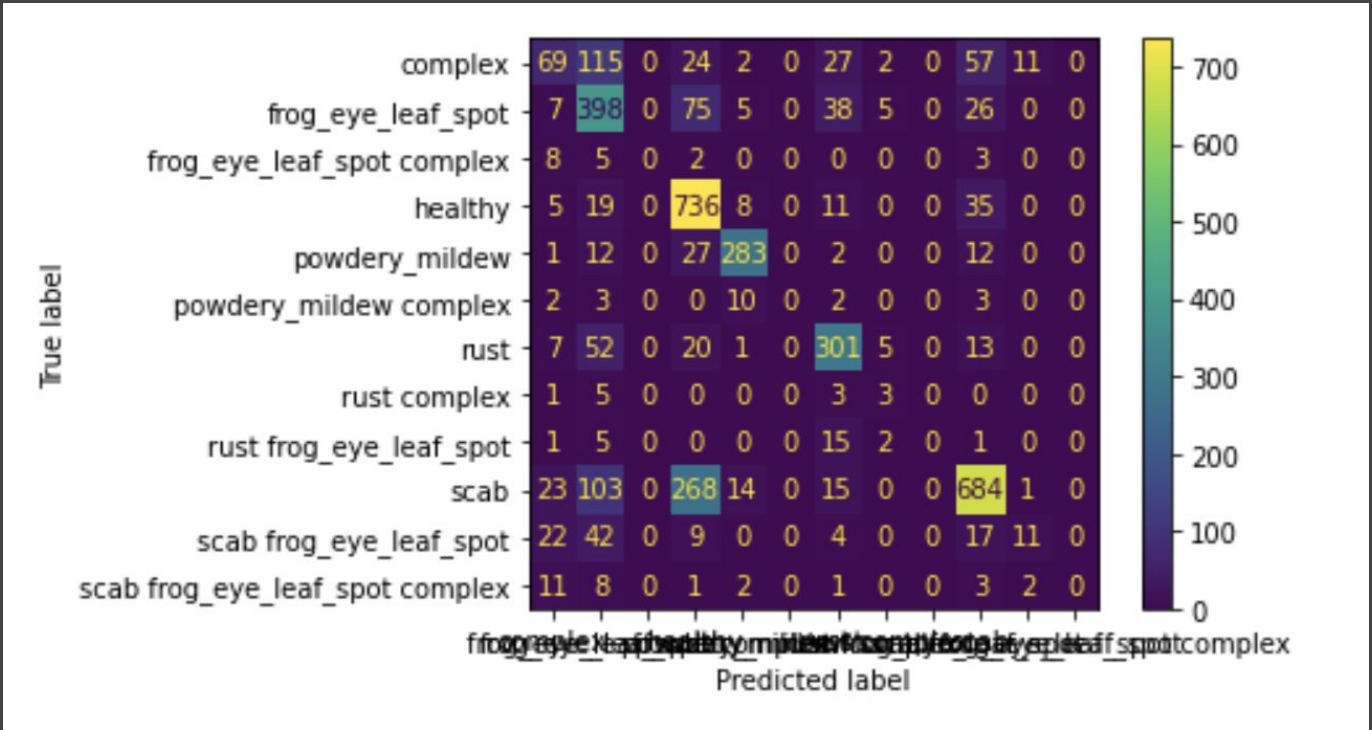


Photo Tests

Real class: scab

Model’s prediction: scab

In [176]:
```
test_images, _ = next(iter(test_generator))
model_final_predictions2 = model_final.predict(test_images)
model_final_predictions2
```

Out[176]:
```
array([[2.9489707e-02, 5.9427951e-02, 5.4278225e-03, 9.8196536e-02,
        3.2741621e-02, 2.0545751e-03, 1.3362162e-02, 1.5190626e-03,
        3.8433820e-04, 7.3017818e-01, 2.5942473e-02, 1.2756570e-03]],
      dtype=float32)
```

Photo Tests

Real class: healthy

Model’s prediction: healthy

In [186]:
```
test_images, _ = next(iter(test_generator))
model_final_predictions2 = model_final.predict(test_images)
model_final_predictions2
```

Out[186]:
```
array([[8.6102085e-03, 7.0557058e-02, 8.6589530e-04, 8.5988849e-01,
        2.2688059e-02, 2.3451625e-04, 1.3234187e-02, 2.6339109e-03,
        9.5051026e-04, 1.6855236e-02, 3.1894832e-03, 2.9240374e-04]],
      dtype=float32)
```

Photo Tests

Real class: complex

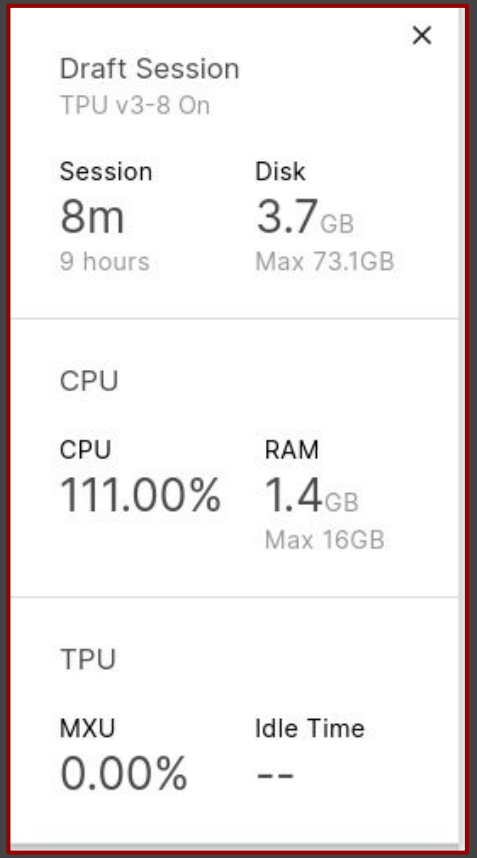
Model’s prediction: complex

Photo Tests

Real class: powdery mildew

Model’s prediction: powdery mildew

Thank you!


Draft Session
TPU v3-8 On
Session
8m
9 hours
Disk
3.7GB
Max 73.1GB
CPU
CPU
111.00%
RAM
1.4GB
Max 16GB
TPU
MXU
0.00%
Idle Time
--