Machine Learning Engineer Nanodegree

Capstone Proposal—Plant Pathology

Junth Basnet

June 14, 2020

Proposal

Domain Background

Currently the US apple industry is worth approximately $15 billion annually [The Plant Pathology 2020 challenge dataset to classify foliar disease of apples]. This makes apples one of the most valuable crops in the US. It is estimated that the industry experiences losses in the millions, due to factors such as insects, fungal and viral pathogens, and other infectious threat. Such threat can lower quality of fruit, decrease crop yields, and in some extreme cases, destroy existing fruit bearing trees. These threats lead to adverse economic and environmental outcomes.

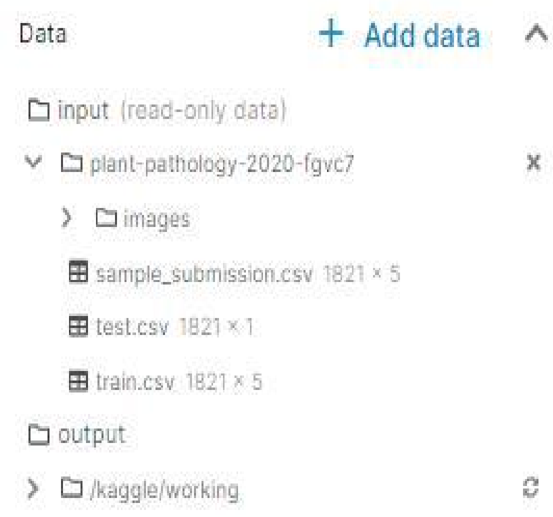
It is imperative that early action is taken to detect such threats; allowing farmers to take the necessary treatment tailored towards the specific threat. Current treatment is limited to manual inspection of apple trees. This approach brings challenges: the first being that human inspection is both costly and largely inefficient—If we could autotomize this process, we could then speed up detection and treatment. The second challenge with manual scouting is that scouts require a great deal of expertise before they can identify and accurately diagnose an orchard. In the following proposal I will attempt to solve this problem by presenting a solution, minimizing loss, maximizing output, and hopefully doing so in a timely manner.

Problem Statement

It is imperative that early action is taken to detect such threats; allowing farmers to take the necessary treatment tailored towards the specific threat. This is a simple **MULTI-CLASS CLASSIFICATION** problem where the network will take in an image as input and output probabilities of the image belongs to each class. Another problem with multi-class classification is how the data is represented in the training set. Luckily, the data is balanced and each class has equal representation.

Datasets and Inputs

This project was inspired in part by the Kaggle "Plant Pathology 2020 -FGVC7" challenge[https://www.kaggle.com/c/plant-pathology-2020-fgvc7]. I must also thank the researchers who created the dataset, which consists of 3,651 high-quality manually captured images of leaves.



The data lives in the 'input' folder. Inside this folder, there is a folder named 'plant-pathology-2020-fgvc7' where there are three .csv files, namely, 'train.csv', 'test.csv' and 'sample_submission.csv'.

The train.csv file consists of a dataframe of 5 columns and 1,821 rows. Along the column axis, there is a column for each class namely, 'healthy', 'multiple_diseases', 'rust' and 'scab' as well as an index column containing the image id. If the image instance belongs a class (e.g. 'healthy'), there will be a '1' in that column and a '0' in other columns.

The test.csv file only contains a single column for the image id. This is standard and to be expected. We are not supposed to know the class of the image, it is for our model to decide. Like I said there is also a sample_submission.csv file in the folder as well however, this is only relevant in the context of the Kaggle competition from where I found the data, so this is not the business of Udacity and I will not go further in explaining it.

In addition to the .csv files, there is also an 'images' folder within the 'plant-pathology-2020-fgvc7' directory. Unsurprisingly, this folder contains the images I will use to train, test, and validate the model. As I previously stated, there are 3,651 high-quality images of leaves. These images were manually captured by the research team and I greatly appreciate them for making it so easy for me.

Solution Statement

This is a simple multi-class classification problem. I will begin by training a Convolutional Neural Network to accurately classify images into different category of health and disease. I will also address differences in image color, angle, light, and shade by using data augmentation. Data augmentation is a strategy to increase diversity of data by slightly changing up variables like the ones I previously listed.

Benchmark Model

For this project, I will be evaluating my model performance using the ROC AUC (the average of the area under the curve of each predicted column) mainly because this is the evaluation metric used in the Kaggle competition to assess performance

Evaluation Metrics

In addition, I will be using the results outlined in the original paper above as my benchmark. Using an out-of-box ResNet50 pre-trained on ImageNet, the researchers were able to achieve 97% accuracy. My goal is to meet or exceed this performance.

Project Design

PYTORCH

For this project I will be using the PyTorch framework. PyTorch is my framework of choice because it is very 'Pythonic' and reasonably simple to pick up. I also enjoy the control it gives me while maintaining its abstraction.

First, I will explore the data so I can get a feel for how it is distributed amongst the classes and address unbalances if need be. The next thing I will do is create a transform operation that will be applied to image batches to address differences in how the images are composed (shadows, angles, shape, etc.). Next, I will create a DataLoader and custom Dataset objects that will prepare our data to be fed into the model.

Now that the data is processed, the next step is to create a custom network class that extends the torch.nn.Module class. I will be using a Convolutional Neural Network to classify images. Starting out with a ResNet50 architecture, I will experiment with different hyperparameters and then train via forward propagation. I shall utilize Dense Cross Entropy to calculate loss and Adam for optimization.

Lastly, I will be practicing my web development skills by deploying my end model to a web application. I plan on experimenting with FastAPI; a light weight, high-performance and easy to learn framework for building web apps.

FastAPI