Machine Learning Engineer Nanodegree

Capstone Proposal

Praveen Maniyan
April 2nd, 2018

Proposal

Domain Background

The domain of this project will be Botany, and especially related to Plant Species identification.

Problem Statement

The Plant species classification can be best achieved by using the leaf samples, as they are unique to species and is a challenging task. The application of this research work is in identifying a plant species from a leaf sample, identify how close a very new leaf sample might belong to a known species, and during fossil extraction, to which variety can this fossil be linked to the current active species. We can try to use Neural Networks with computer vision to solve the classification problem.

Datasets and Inputs

The dataset for building this Machine Learning problem can be taken from Folio Data Set found at https://archive.ics.uci.edu/ml/datasets/Folio

This dataset has 20 photos of leaves for each of 32 different species.

Solution Statement

I can use the Transfer Learning mechanism, either use VGG16 or ResNet. I would have to provide some filter's that filters' the color (green or violet, etc.), and have a layer from Keras model that detects edges. However, an extra feature which I would like to try is to add layers that might identify the veins of the leaf. Given the shape (edges') and the vein pattern, these should be great distinguisher for the species classification problem.

Benchmark Model

Based on the paper titled, "Data Augmentation for Plant Classification", Authors, Pornntiwa Pawara, Emmanuel Okafor, Lambert Schomaker, and Marco Wiering, for the Folio Dataset they have achieved a Test accuracy of 99.04%.

Either I can try to achieve this or improve the accuracy by using the Leaf Vein feature.

Evaluation Metrics

I would like to use the validation loss and test accuracy parameter for evaluation.

Project Design

- Download the data from https://archive.ics.uci.edu/ml/machine-learning-databases/00338/

- Do some organizing of the data to get the filename (image) and its label, and represent it in the dataset for Python/Keras

- Load the images and dataset, split it into train, test

- Scale the images to a standard pixel size

- Either build a 4 to 5 layer model (filter color, edge detect, vein detect, output layer), or use any of the existing Keras model that does the above (some research and study work has to happen on this)

- Carry out some Image augmentation, especially rotation, flip and contrast

- Train the model with a small batch size and epoch of 25, 50 and 100

- Get the best val_loss value and test it on the test set