Machine Learning Engineer Nanodegree

Capstone Proposal

Rishi Ohri
March 16, 2019

Proposal

Domain Background

The proposed project is to differentiate a weed (i.e. unwanted plant seedling) from a crop seedling. The ability to do so effectively can mean better crop yields and better care for the environment.

The project problem and dataset are taken directly from Kaggle Competition: Plant Seedlings Classification – Determine the species of a seedling from an image.

The Aarhus University Department of Engineering Signal Processing Group, in collaboration with University of Denmark, has shared a dataset of images (on Kaggle) belonging to 12 species of plant seedlings at various stages of growth.

Related academic research: A Public Image Database for Benchmark of Plant Seedling Classification Algorithms. In this paper, a benchmark based on f1 scores is proposed to standardize the valuation of classification results obtained with the provided database.

Problem Statement

The problem here is a close resemblance among different species of plant seedlings. A potential solution to resolve this problem is to design an Image Classification model using Convolution Neural Network. The model should be able to determine the species of a plant seedling from an image, with a good metrics score.

Datasets and Inputs

Dataset is taken directly from Kaggle which provides a training set and a test set of images of plant seedlings at various stages of growth. Each image has a filename that is its unique id. The dataset comprises of 12 plant species:

- Black-grass
- Charlock
- Cleavers
- Common Chickweed
- Common wheat
- Fat Hen
- Loose Silky-bent
- Maize
- Scentless Mayweed
- Shepherds Purse
- Small-flowered Cranesbill
- Sugar beet

Here, training dataset contains images of plant species organized by folder (i.e. dataset consists of subfolders with name of plant species). However, the test dataset consists of images for which species need to be predicted by our trained model and the predictions need to be submitted to Kaggle competition.

So, here I will divide the training dataset into three parts:

- training dataset will be used to train my model
- validation dataset to cross-validate my model
- test dataset will be used to test accuracy of my model

Solution Statement

The solution to this problem is to build and train a model that can accurately classify an image into one of the 12 categories mentioned above.

Following steps will be performed to find a possible solution:

- Data pre-processing (may include different techniques to visualize the data, image augmentation etc.)
- Creation of simple Convolution Neural Network (from scratch)
- Creation of Convolution Neural Network (using transfer learning)
- Measure performance of both the models.

Benchmark Model

I will consider my simple CNN model as benchmark model. I will try to create another CNN model (using transfer learning) and compare its performance to that of benchmark model using same evaluation metrics.

I will also compare the scores of my two models with those provided on Kaggle Leaderboard.

Evaluation Metrics

As per Kaggle, the evaluation metrics to be used is MeanFScore. So, I will use the same evaluation metrics.

Given positive/negative rates for each class *k*, the resulting score is computed this way:

$$Precision(micro) = \frac{\sum_{k \in c} TP_k}{\sum_{k \in c} TP_k + FP_k}$$

$$Recall(micro) = \frac{\sum_{k \in c} TP_k}{\sum_{k \in c} TP_k + FN_k}$$

Where TP – True Positive, FP – False Positive and FN – False Negative

F1-score is the harmonic mean of precision and recall

MeanFScore:

$$F1_{micro} = 2\ Precision_{micro}\ Recall_{micro} / (Precision_{micro} + Recall_{micro})$$

Project Design

I intend to build my project in following broad steps:

Step 1: Perform some data analysis, visualization and exploration to get better understanding.

Step 2: Divide the data in training folder into three parts:

- training dataset will be used to train my model
- validation dataset to cross-validate my model
- test dataset will be used to test accuracy of my model

Step 3: Try few techniques like data augmentation etc on training dataset if required to further improve accuracy of my model.

Step 4: Build and train two models:

- Simple Convolution Neural Network (from scratch)
- Convolution Neural Network using transfer learning

Step 5: Measure performance based on evaluation metrics.