Mihwa Han
January 4, 2018

Plant Seedlings Classification

Udacity Machine Learning Nano Degree
Capstone Project Proposal

1. Domain Background

Can we distinguish a weed from a crop seedling? Many weeds and crop seedlings look similar, so sometimes it can be hard to differentiate weeds from seedlings. Although they look alike when they are young, they will eventually grow into totally different plants. This challenge presents itself in many real-world situations. For example, Charlock is an agricultural weed and an invasive species in some areas outside its native range, while Shepherds Purse, considered an herbal medicine, has seedlings of very similar shape to Charlock. Thus, farmers need to differentiate each type of seedling to successfully cultivate without damaging their plants. The goal of this project is develop a model to effectively classify several species of seedlings. This is based on a Kaggle project called "Plant Seedlings Classification" (https://www.kaggle.com/c/plant-seedlings-classification).

2. Problem Statement

Why is this project meaningful? As described earlier, many types of seedlings look very similar, especially compared to weeds, so even an experienced gardener can have difficulty differentiating weeds from the seedlings of more helpful plants. The images below are examples of each seedling in the Kaggle dataset.

As seen above, some of the plant breeds are pretty similar in appearance. Moreover, each species includes different growth stages, so the character of each seedling could include several different features depending on their stage of growth. Furthermore, I will use low resolution images, which are more likely representative of real life, and because of their low resolution it is not easy for even an experienced human to identify the correct seedling species.

3. Datasets and Inputs

The dataset contains 4,750 training images and 794 test images, each of which belongs to one of twelve species at several different growth stages. I obtained the dataset through a Kaggle competition, but originally the dataset was from the Aarhus University Department of Engineering Signal Processing Group. The training images include the label of seedling species, while the test images do not include labels. Although I do not have the test labels, I can still check the accuracy of my model's predictions because Kaggle will verify my model by returning an F1 score. The list of species is as follows:

Black-grass

Charlock

Cleavers

Common Chickweed

Common wheat

Fat Hen

Loose Silky-bent

Maize

Scentless Mayweed

Shepherds Purse

Small-flowered Cranesbill

Sugar beet

The input images have a range of sizes, with some less than 100x100 pixels and some over 1000x1000 pixels, and each image has RGB colors.

4. Solution Statement

Convolutional Neural Networks (CNNs) are particularly powerful for training on multi-dimensional data, such as images. They are designed to handle 2D spatial information in images particularly well; unlike a regular neural network, CNNs use locally-connected layers and do not use vectors for their hidden layers. I will use a CNN to categorize the seedling species using Python/Keras. More details about the CNN model can be found in the Project Design section.

5. Benchmark Model

For a benchmark model, I will use a simple CNN model just with three convolutional layers and max-pooling layers. I tested this simple CNN on the input dataset, and I achieved a mean F-score of 0.45, and a prediction accuracy of 0.4. This accuracy is much better than random chance (1/12), but still not impressive. The aim of this project is to find a model with significantly better mean F-score and accuracy.

6. Evaluation Metrics

(1) Mean F-score

Here, I will use mean F-score as an evaluation metric. The mean F-score is a weighted average of the *precision* and *recall*, which are given by the equations below:

$$Precision_{micro} = \frac{\sum_{k \in C} TP_k}{\sum_{k \in C} TP_k + FP_k}$$

$$Recall_{micro} = \frac{\sum_{k \in C} TP_k}{\sum_{k \in C} TP_k + FN_k}$$

The summations are over all species of seedling. TP means True Positive, which are cases where the correct seedling species is predicted. FP means False Positive, which are cases where a seedling is incorrectly predicted to be a particular species. FN means False Negative, which are cases where a seedling is not predicted to be its true species.

Therefore, mean F-score is written as the following:

$$MeanFScore = F1_{micro} = \frac{2Precision_{micro}Recall_{micro}}{Precision_{micro} + Recall_{micro}}$$

(2) Accuracy

The definition of accuracy is the number of correct results divided by all results. For my project I aim to achieve better than 40% accuracy.

7. Project Design

Before preprocessing the data, I will separate 4,750 training images into train set (80%), validation set (10%), and test set (10%). Although there is a separate test set provided by Kaggle, I will not use this dataset to validate or test my model. Rather, I will use the Kaggle test set as a hold-out dataset, and will only use it to check the final mean F-score.

First, I need to preprocess the input data for Keras. The images have different sizes with 3 RGB channels, so I need to reshape all of the input images to have the same width and hight (224 pixels by 224 pixels). I will reshape the image arrays as (number of images, 3, 224, 224). Also, I will convert the pixel values to float32 format and normalize all of the arrays. In the case of the label dataset, I will simply convert 1-dimensional class arrays to 12-dimensional class matrices.

Next, I will start with a basic CNN structure with three Convolutional layers and pooling layers. Also, it will have fully connected layer with softmax function before the output layer. Once the basic CNN is set up, I will improve the model in several ways such as:

1) adding more CNN layers
2) different numbers of filters
3) different optimization functions (SGD, Adam)
4) modifying pooling layers
5) Image augmentation

The figure below is an example design I will use. Moreover, I will compare the results from popular models used with transfer learning like InceptionV3 and Xception. I will use mean F-scores to check the performance of each model, and pick the best model. Finally, I will apply the best overall model to the test sets and post the results on Kaggle to check if the model's mean F-score is reliable.