Machine Learning Engineer Nanodegree

Capstone Proposal: Plant Seedlings Classification in Kaggle

Yan Weili
March 3rd, 2018

Domain Background

Currently, farmers spray a standard herbicide mixture over the entire field to combat weeds [1]. Such a method is neither economical nor environment-friendly. The herbicides are not only expensive but also may pollute the rivers around. In addition, the effect of the standard mixture of herbicides varies on different weeds. Nowadays, several decision systems are able to reduce the herbicide expenditures by at least 40% by recommending the optimal herbicide dosages for a particular weed [1]. However, the category of the weed must be identified before using the decision system. Recently, a method using a fully convolutional neural network is presented to automatically detect the weeds in color images despite heavy leaf occlusion [2].

Problem Statement

In order to automatically inspect the species of weeds (differentiate a weed from a crop seedling), the Aarhus University Signal Processing group, in collaboration with University of Southern Denmark, has recently released a dataset containing images of approximately 960 unique plants belonging to 12 species at several growth stages [3]. Kaggle are now hosting the Plant Seedlings Classification competition using this dataset to classify the different species based on their images.

We are provided by Kaggle with a training set and a test set of images of plant seedlings at various stages of grown [4]. Each image has a filename that is its unique id. The dataset consists of 12 plant species, which is listed as:

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

We aim to train a convolutional neural network (CNN) with transfer learning to classify the plant's species from a photo.

Datasets and Inputs

The datasets can be downloaded from Kaggle [4], which are departed into the training dataset and the testing dataset. As the class label of the testing dataset in Kaggle is unavailable, we will solely use the original training dataset (training.zip) in Kaggle [4] as the training dataset and the testing dataset in this project.

The original training dataset in Kaggle has images of plant seedlings at various stages of grown. Each image has a filename that is its unique id. Each image is stored in its corresponding species-named file. The dataset has 12 species-named files, and each file has more than 200 images. Number of images for each plant species is listed in Table 1. There are 4750 images in total (1.72GB). As each image has different size and is colorful with three channels RGB, we will reshape each image into fix dimension, for example 299*299*3 when using Xception model in Keras.

Table 1. Number of images for each plant species

Plant Species	Number of Images
Black-grass	263
Charlock	390
Cleavers	287
Common Chickweed	611
Common wheat	221
Fat Hen	475
Loose Silky-bent	654
Maize	221
Scentless Mayweed	516
Shepherds Purse	231
Small-flowered Cranesbill	496
Sugar beet	385

We will randomly choose 10% images in each species-named file as the testing dataset, and the resting 90% images in each species-named file will be used as the training and validation dataset.

Solution Statement

In this project, a convolutional neural network (CNN) with transfer learning will be trained to classify the plant's species from a photo. Transfer learning using the learned weights of the pre-trained networks on a large dataset will be applied to extract features in images. Pre-trained networks such as Xception, RESNET50, InceptionV3, and VGG16 will be explored in this project.

Benchmark Model

Random Guess: The random guess is included as the first benchmark model. It is able to evaluate the performance improvement of the CNN model as compared to a model with zero information about the input data (no features are included).

SVM Classification: The Support Vector Machine (SVM) classifier serves as another benchmark model. The SVM classifier is to be trained on the flattened array of the grayscale images.

The CNN model via transfer learning should have higher performance than both random guess and SVM classification, since the CNN model takes the spatial relationship of image pixels while the flattened feature does not.

Evaluation Metrics

We apply the same evaluation metrics from Kaggle [5]. The prediction results are evaluated on *MeanFScore*, which is actually a micro-averaged F1-score at Kaggle. Given positive/negative rates for each class *k*, the resulting score is calculated as

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

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

The micro-averaged F1-score *MeanFScore* is the harmonic mean of precision and recall, which is calculated as

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

Project Design

- **Libraries to be used:** Pandas, Numpy, Sklearn, Keras, Tensorflow, Opencv, Matplotlib, Seaborn, and so on.
- **Language to be used:** Python 2.7 or higher
- **Work Flow:**

- *Image preprocessing*: As the images in the datasets may not be in the input format of the CNN models, the first step is to reshape the images and turn them into grayscale if needed.
- *Feature extraction*: The second step is the feature extraction stage. For the CNN model, the feature is taken from a fully connected layer of the pre-trained CNN model. For the SVM model, the feature is simply the flatten array of the **reshaped** grayscale image.
- *Model training***:** After feature extraction stage, a fully-connected neuron network (FCNN) and a SVM classifier are trained using the transferred feature and the flattened array, respectively. This stage includes fine tuning the number of layers and the number of neurons of the FCNN, and the parameters of the SVM model.
- *Model evaluation:* The last stage is to evaluate the model performances using the evaluation metrics, and analyze the pros and cons of each model.

Reference

［1］ https://vision.eng.au.dk/roboweedmaps/

［2］ Mads Dyrmann, Rasmus Nyholm Jørgensen, Henrik Skov Midtiby. (2017,7). Detection of Weed Locations in Leaf-occluded Cereal Crops using a Fully-Convolutional Neural Network. Advances in Animal Biosciences Volume 8, Issue 2 (Papers presented at the 11th European Conference on Precision Agriculture (ECPA 2017).

［3］ https://www.kaggle.com/c/plant-seedlings-classification#description

［4］ https://www.kaggle.com/c/plant-seedlings-classification/data

［5］ https://www.kaggle.com/c/plant-seedlings-classification#evaluation