Machine Learning Engineer Nanodegree

Capstone Proposal

Deep learning for Plant Seedlings Classification

Domain Background

Weed control, which directly affect the efficiency of algoricoutul, has been researched for several decades. Keeping crop free of weeds is critical in good farm management. Farmers should plan their farm activities well so that they do weeding at the right time and in the right way. For this purpose, to differentiate weed from crop seedling is the essential step. Different seedlings have their unique features on their colors, shape, and outlines. Multiple technologies have been adopted including GPS mapping, ground-based vision identification, remote sensors and so on. This project focuses on to identify and classify the seedlings from the image using machine learning based computer vision. Fortunately Aarhus University and University of Southern Denmark recently released a dataset containing images of approximately 960 unique plants belonging to 12 species at several growth stages. those labeled data provides the researchers an opportunity to experiment with different image recognition techniques(for instance, K-nearest, SVM, CNN, and other high performance deep learning models). Kaggle also helped to organize an online competition providing a platform to let researchers communicate and cross-compare the solutions.

Citation

- PAPER: A Public Image Database for Benchmark of Plant Seedling Classification Algorithms (https://arxiv.org/abs/1711.05458 (https://arxiv.org/abs/1711.05458))
- Kaggle Competition: https://www.kaggle.com/c/plant-seedlings-classification (https://www.kaggle.com/c/plant-seedlings-classification)
- Automated Classification of Seedlings Using Computer Vision (http://plant_recognition.sdu.dk/files/AutomatedClassificationOfSeedlingsUsingComputerVision.pdf (http://plant_recognition.sdu.dk/files/AutomatedClassificationOfSeedlingsUsingComputerVision.pdf))

Problem Statement

The topic of this project is to use provided dataset training weed recognition algorithms. The most critical part is to select and adjust the architecture of the model togher with its hyper-paramaters to push the limits of the accuracy. To get inspiration from previous research results, transform learning becomes a good start point. The dataset from AU & SDU consists of different seeding images and labeled classification will be used for training and testing separately. The performance will be evaluated on MeanFScore, which at Kaggle is a micro-averaged F1-score.

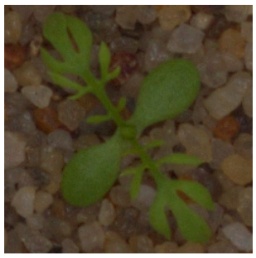
Datasets and Inputs

Datasets are devided by a training set and a test set of images of plant seedlings at various stages of grown. it comprises annotated RGB images with a physical resolution of roughly 10 pixels per mm. Each image has a filename that is its unique id. The dataset comprises 12 plant species, including: *(Samples of each species from the database https://vision.eng.au.dk/plant-seedlings-dataset/ (https://vision.eng.au.dk/plant-seedlings-dataset/))*

Maize

Common wheat

Sugar beet



Scentless Mayweed

Chickweed

Shepherd's Purse

Cleavers

Charlock

Fat Hen

Cranesbill

Black-grass

Loose Silky-bent

File descriptions

train dataset: Images are all in .png format with RGB color space. Dimension is various from image to image. Images are stored in different folders with the name of its seedling category.

./train

```
/Black-grass (263 images)
/Charlock (390 images)
/Cleavers (287 images)
/Common Chickweed (611 images)
/Common wheat (221 images)
/Fat Hen (475 images)
/Loose Silky-bent (654 images)
/Maize (221 images)
/Scentless Mayweed (516 images)
/Shepherds Purse (231 images)
/Small-flowered Cranesbill (496 images)
/Sugar beet (385 images)
```

test dataset: All .png images are put in one folder.

./test

```
795 images
```

train.csv - the training set, with plant species organized by folder
test.csv - the test set, need to predict the species of each image
sample_submission.csv - a sample submission file in the correct format

Considering it is a small dataset, all examples of the dataset will be used at this project for training, validating and testing.

Solution Statement

The goal of the project is to train a model/classifier that can recognize the seedlings from the image input. Considering the type of the input as the image, Convolutional Neural Networks was considered as a good option to start given its abilities on visual applications. Different architectures of CNN with transfer learning will be evaluated and tuned for the use case. ImageNet database and its relevant projects provide multiple high potential models worthy to take a test. Data visualiztion method like tensorbord will be used to monitor the performance during the training. The model will be measured using MeanFScore.

Benchmark Model

Kaggle presents a leaderboard (https://www.kaggle.com/c/plant-seedlings-classification/leaderboard (https://www.kaggle.com/c/plant-seedlings-classification/leaderboard)) to compare the performance among different solutions. Up til know the top 50s have reached an excellent score (>0.98110). The best score is 0.99622. Several kernels (SimpleNet, Xception network, VGG based classifier and so on) have also been published on the kaggle platform.

To start with, a CNN model is picked as a benchmark. Its architecture presents here:

In [ ]:

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 48, 48, 3)	0
conv2d_1 (Conv2D)	(None, 46, 46, 16)	448
batch_normalization_1 (Batch	(None, 46, 46, 16)	64
activation_1 (Activation)	(None, 46, 46, 16)	0
conv2d_2 (Conv2D)	(None, 44, 44, 16)	2320
batch_normalization_2 (Batch	(None, 44, 44, 16)	64
activation_2 (Activation)	(None, 44, 44, 16)	0
max_pooling2d_1 (MaxPooling2	(None, 22, 22, 16)	0
conv2d_3 (Conv2D)	(None, 20, 20, 32)	4640
batch_normalization_3 (Batch	(None, 20, 20, 32)	128
activation_3 (Activation)	(None, 20, 20, 32)	0
conv2d_4 (Conv2D)	(None, 18, 18, 32)	9248
batch_normalization_4 (Batch	(None, 18, 18, 32)	128
activation_4 (Activation)	(None, 18, 18, 32)	0
global_max_pooling2d_1 (Glob	(None, 32)	0
dense_1 (Dense)	(None, 64)	2112
dropout_1 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 32)	2080
dropout_2 (Dropout)	(None, 32)	0
dense_3 (Dense)	(None, 12)	396

Evaluation Metrics

Solution is evaluated on MeanFScore (micro-averaged F1-score).
Given positive/negative rates for each class **k**, the resulting score is computed this way:

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

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

F1-score is the harmonic mean of precision and recall

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

Project Design

Overview:

A theoretical workflow for this project will be

- Data prepare
- Pre-processing
- Iteration of:
  - Model training and validation
  - Parameter tuning
- Model test

Data Prepare

The first step is to get available datasets. The dataset will be downloaded from kaggle including a test set and a train set (https://www.kaggle.com/c/plant-seedlings-classification/data (https://www.kaggle.com/c/plant-seedlings-classification/data)). Data will also be split for validation. Tools like numpy, pandas will be used.

Pre-processing

Images of the provided dataset have different dimensions. Thus a resize step is needed to convert all images to the same size. The RGB data of all pixels will be normalized from 255 to 1. Data argumentation will also be implemented by adding shift, rotation, and flip. Tools like keras ImageDataGenerator will be used.

Model training and validation

More and more computer vision challenges have been successfully launched. There are a few of architectures published showing a really good performance on image recognition. Keras has pretrained models of those architectures including Xception, VGG, Inception, etc. It will help me to quickly test different models and evaluate their performances. NVIDIA CNN model(https://arxiv.org/pdf/1604.07316.pdf (https://arxiv.org/pdf/1604.07316.pdf)) is another interesting option. In this step, tools like tensorflow and keras will be used.

Parameter tuning

Adjust hyperparameters over the course of training runs to achieve an optimal result. Considering the requirements of large computing power, cloud computing service(AWS EC2) may be adopted to accelerate the hyperparameter tuning iteration.

Model test

At last, the model will be test with an all-new test data. The evaluation metrics (MeanFScore) will be calculated by comparing with the ground truth.