# Plant Seedling Classification Using Convolutional Neural Networks

Machine Learning Engineer Nanodegree Capstone Proposal

Brungi Vishwa Sourab

November, 2018

## Domain Background

For several decades, researchers have worked on systems aimed at performing site-specific **weed control.** The approaches used range from weed maps constructed using coarse remote sensing data to real-time precision spraying using ground-based platforms equipped with high-resolution imagery sensors. Common to all approaches is the goal of detecting weeds - either in patches or as single plants. Although some systems are commercially available, a true commercial breakthrough of such systems is still to come despite the construction of several prototypes and case studies showing promising results. The reason may be that a general approach enabling robust classification despite varying conditions and species compositions is yet to be discovered. Previous work has been done in this field. A research paper on similar problem identifying plant species by the shape of leaf is published by Ji-Xiang Du, Xiao Feng Wang & Guo-Jun Zhang, Department of Automation, University of Science and Technology of China.[1]

I strongly believe that now is the right time for such breakthrough systems because of the latest techniques in deep learning. Until very recently, the main problem was acquisition of the image data to be used to build robust systems. But, in November of 2017, to support and encourage the development of species recognition techniques for the agricultural industry, the Computer Vision and Biosystems Signal Processing Group, Department of Engineering-Aarhus University has collected the data and made it available to the public for free.

## Problem Statement

Differentiating a weed from a crop seedling by the image can pave a significant way to intelligent weed control systems and thus eliminating the unwanted plants in its initial stages only. The ability to do so effectively can mean better crop yields and better stewardship of the environment.

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. My goal is to build a classifier that classifies the seedling's class based on the image input, using state of the art deep learning techniques. Here are some sample images of seedlings from the dataset.

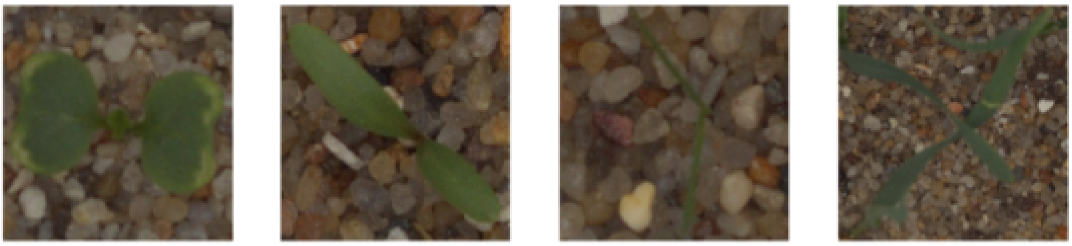


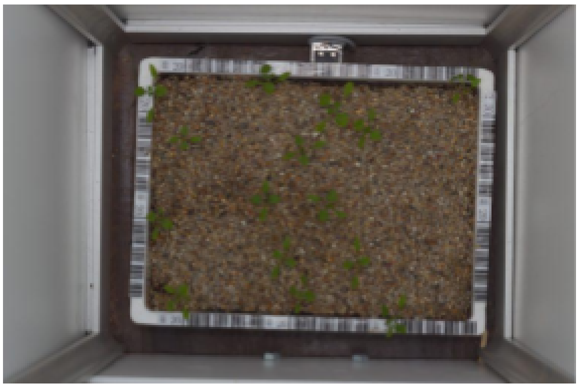
Figure 1: Sample images from the Seedlings dataset

**Dataset**

The Plant Seedlings Dataset contains images of approximately 960 unique plants belonging to 12 species at several growth stages. It comprises annotated RGB images with a physical resolution of roughly 10 pixels per mm.

The database have been recorded at Aarhus University Flakkebjerg Research station in a collaboration between University of Southern Denmark and Aarhus University. You can find more information about the dataset in the PAPER: A Public Image Database for Benchmark of Plant Seedling Classification Algorithms [2]

The datasets are available for download in different versions i.e, Raw images (9.7GB), Cropped Images(1.7 GB) and Segmented Images (258 MB). In my project, I use Cropped Images version of the dataset. A sample of raw type of images is shown below:



Raw chickweed image sample

*Image source: Computer Vision and Biosystems Signal Processing Group, Aarhus University*

Since it is not possible to include too many species in the database, only a subset of high importance to the Danish agricultural industry are chosen to make the database. There are 12 species of images in the database:

*Maize*

*Common wheat*

*Sugar beet*

*Scentless Mayweed*

*Chickweed*

*Shepherd's Purse*

*Cleavers*

*Charlock*

*Fat Hen*

*Cranesbill*

*Black-grass*

*Loose Silky-bent*

The classes in the dataset are not equally balanced. However, to improve the model performance, I will use different balancing techniques. The number of images in each species are:

| | |
|---|---|
| ● Sugar beet | 385 |
| ● Loose Silky-bent | 654 |
| ● Charlock | 390 |
| ● Shepherd's Purse | 231 |
| ● Maize | 221 |
| ● Common | 221 |
| ● Cleavers | 287 |
| ● Cranesbill | 496 |
| ● Black-grass | 263 |
| ● Chickweed | 611 |
| ● Fat Hen | 475 |
| ● Scentless Mayweed | 516 |

**Solution Statement**

I intend to solve this problem by building an image classifier using Convolutional Neural Networks. I will load the dataset and preprocess the images by sharpening, masking and segmenting to extract information such as shape and location of the leaves in the images. Later, I will convert the images to tensors or numpy arrays. I will build a base image classifier using CNN in the beginning and later experiment with the advanced architectures of CNN. I also intend to use the different versions of architectures available on the internet such as RESNET, XCEPTION by transfer learning. After experimenting with all the models, I will choose the best model and use it to build an algorithm that classifies species of the plant given an image as input. I have included detailed workflow in the project design section of this document.

**Benchmark Model**

The project I am doing is a typical image classification problem. I intend to use the benchmark model as a Vanilla Convolutional Neural Network and try to build more complicated networks to act as a classifier. There is also a Kaggle Competition[3] held on this dataset. I will submit my results to the Kaggle and compare my score.

**Evaluation Metrics**

Performance is measured using weighted averages of **f1 scores** for each fold. Given positive/negative rates for each class *k*, the resulting score is computed this way:

True Positive (TP): A **true positive** is an outcome where the model *correctly* predicts the *positive* class.[4]

True Negative (TN): A **true negative** is an outcome where the model *correctly* predicts the *negative* class.[4]

False Positive (FP): A **false positive** is an outcome where the model *incorrectly* predicts the *positive* class.[4]

False Negative (FN): A **false negative** is an outcome where the model *incorrectly* predicts the *negative* class.[4]

Precision (P): Precision is a metric that says, Out of all points that are predicted to be positive, how many are actually Positive.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall (R): Recall is a metric that says, Out of all positive points, how many are actually positive.

$$\text{Recall} = \frac{TP}{TP + FN}$$

**F1 Score:** $F_1$ score is the harmonic mean of precision and recall.

$$\text{F1} = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

**Project Design**

I intend to do the project in two phases. The details and the workflow is listed below:

**Phase 1:**

- **Exploring the data**
  - Importing libraries and data
  - View a sample of training data
  - View number of images in training data
  - View species vise images count
- **Data Preprocessing**
  - Transforming Images (Resize, Converting to Arrays, Normalizing)
  - Sharpening Images
  - Creating Segments of leaves
  - Splitting data for training and Validation
- **Model Building**
  - Basic CNN for benchmarking
  - Complex CNN architectures
  - Hyperparameter Tuning
  - Training and Validating the models
  - Saving best model and weights to disk
- **Algorithm Building**
  - Creating a pipeline for image classification
  - Loading and predicting the class on saved model
  - Exporting the algorithm as a Python API

**Phase 2:**

- **Creating a Web Application**
  - Creating a basic website in Python.
  - Creating a form that takes user input and calls API.
  - Predicting the species and displaying the Output.

**References**

[1] PAPER: Leaf shape based plant species recognition
https://www.sciencedirect.com/science/article/abs/pii/S009630030600806X

[2] PAPER: A Public Image Database for Benchmark of Plant Seedling Classification Algorithms

[3] Kaggle Plant Seedlings Competition
https://www.kaggle.com/c/plant-seedlings-classification

[4] Google Machine Learning Crash Course
https://developers.google.com/machine-learning/crash-course/classification/true-false-positive-negative