**Machine Learning Engineer Nanodegree**

**Capstone Project**

Muthudivya Navaneetha Kannan  
April 20th, 2018

**I. Definition**

*(approx. 1-2 pages)*

**Project Overview**

One of major challenges in agriculture is weed management. Sustainable crop production is the need of the hour as we are struggling to supply food for the increasing global population. For effective crop production, weed management is one of the key factors. The ability to automatically classify and remove weeds among the crop at the seedling level will yield huge benefit and higher harvest. This problem is not something that can be totally eradicated, but a solution to this problem can help solve and mitigate weed problem now and, in the years, to come.

This project is an attempt to tackle one such problem. The dataset used for this problem has been donated by The Aarhus University Signal Processing group, in collaboration with University of Southern Denmark, in the hope “*to provide researchers a foundation for training weed recognition algorithms*” [3]. A benchmark for standardizing the results of this classification problem has been discussed in [arXiv:1711.05458](https://arxiv.org/abs/1711.05458) using the [original dataset](https://vision.eng.au.dk/plant-seedlings-dataset).

**Problem Statement**

Problem is the similarity among different species of plant seedling. In this project, the model should be able to determine the species of a seedling from an image. The dataset contains images of twelve categories of plant seedling, and hence the solution should be able to classify an image into one of twelve categories.

The solution of this problem is to build and train a model that can classify the new unseen image into one of the twelve mentioned categories accurately.

This can be achieved by:

* For any given image, the model outputs one predicted value (i.e., the species of the seedling with highest probability)
* However, the model outputs probability of every class that the image can belong to, we can design the output to be in such a way that the only category with highest probability is shown as prediction for the input image.

**Metrics**

Since the dataset is unbalanced, the appropriate metric to evaluate the performance of this classifier is by calculating the Mean F Score.

The F1 score [1] can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The relate contribution of precision and recall to the F1 score are equal. The formula for the F1 score is given as:

F1 = 2\*(precision \* recall) / (precision + recall)

Precision and recall can be calculated as [2]:

Precision = number of TP / (number of TP + number of FP)

Recall = number of TP / (number of TP + number of FN)

Where, TP is True Positives, FP is False Positives and FN is False Negatives.

As this problem statement has multi-class solution, Mean F1 score is the weighted average of the F1 scores of each class.

**II. Analysis**

*(approx. 2-4 pages)*

**Data Exploration**

Datasets are obtained from Kaggle [4]. A training set and a test set containing them images of plant seedlings at various stages of grown. Each image has a filename that is its unique ID. The dataset comprises of 12 plant species, which are listed below:

Black-grass

Charlock

Cleavers

Common Chickweed

Common wheat

Fat Hen

Loose Silky-bent

Maize

Scentless Mayweed

Shepherds Purse

Small-flowered Cranesbill

Sugar beet

The objective is to build an image classifier that can accurately identify the image and classify the image into one of the above 12 species.

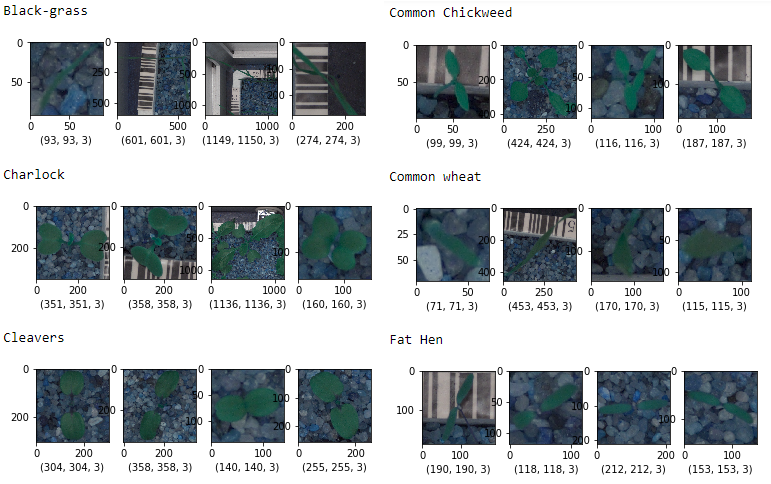
A quick look at the train dataset tells that the number of images available for each class (['Sugar beet', 'Small-flowered Cranesbill', 'Maize', 'Common wheat', 'Common Chickweed', 'Fat Hen', 'Black-grass', 'Cleavers', 'Scentless Mayweed', 'Charlock', 'Loose Silky-bent', 'Shepherds Purse']) is varied (496, 221, 221, 611, 475, 263, 287, 516, 390, 654, 231), i.e. the dataset is highly unbalanced. To combat with the unbalanced dataset, one of the many strategies can be applied. The dataset can be under-sampled, or data augmentation can be applied to balance the under-represented classes. However, the problem of unbalanced dataset can also be dealt by calculating the confusion matrix and F1 score for the classifier, as these metrics give us the more appropriate evaluation of the model. The images are RBG colored and have a background that is undesirable for the problem at hand can be gotten rid of by masking – this can be achieved by converting the RGB images to HSV mode by tuning the related parameters. I intend to use 80:20 as the train to validation split ratio for training my classifier.

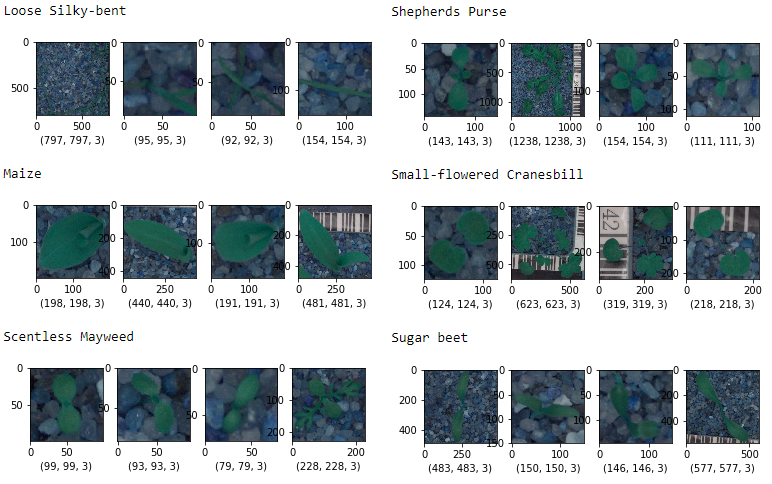
In this section, you will be expected to analyze the data you are using for the problem. This data can either be in the form of a dataset (or datasets), input data (or input files), or even an environment. The type of data should be thoroughly described and, if possible, have basic statistics and information presented (such as discussion of input features or defining characteristics about the input or environment). Any abnormalities or interesting qualities about the data that may need to be addressed have been identified (such as features that need to be transformed or the possibility of outliers). Questions to ask yourself when writing this section:

* *If a dataset is present for this problem, have you thoroughly discussed certain features about the dataset? Has a data sample been provided to the reader?*
* *If a dataset is present for this problem, are statistics about the dataset calculated and reported? Have any relevant results from this calculation been discussed?*
* *If a dataset is****not****present for this problem, has discussion been made about the input space or input data for your problem?*
* *Are there any abnormalities or characteristics about the input space or dataset that need to be addressed? (categorical variables, missing values, outliers, etc.)*

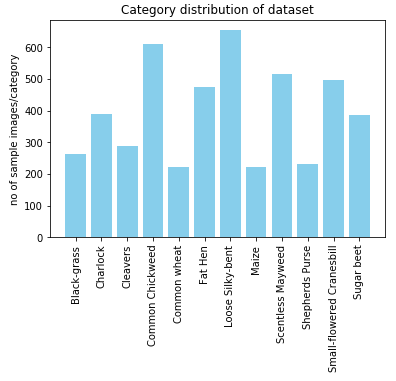
**Exploratory Visualization**

Below are 4 random samples from each of the categories:





Further plotting the number of samples available in the training set for each category gives the below graph:



Above graphical representation of the categorical distribution tells us that the training dataset is unbalanced. Few of the classes have only a little over 200 samples whereas few others have over 500 samples. This shows that dataset is highly unbalanced and hence performance of this classifier can not be measure by its accuracy and hence F1 score is used to measure the performance.

**Algorithms and Techniques**

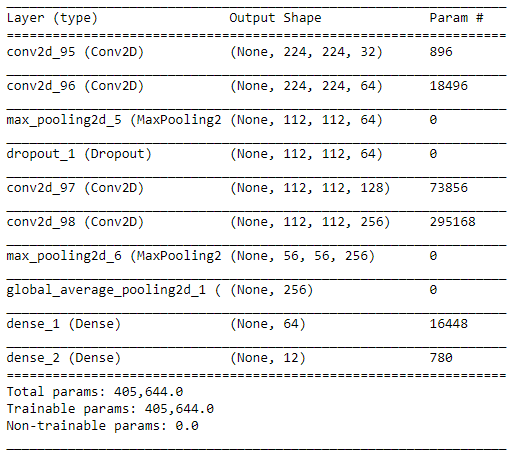
In this section, you will need to discuss the algorithms and techniques you intend to use for solving the problem. You should justify the use of each one based on the characteristics of the problem and the problem domain. Questions to ask yourself when writing this section:

* *Are the algorithms you will use, including any default variables/parameters in the project clearly defined?*
* *Are the techniques to be used thoroughly discussed and justified?*
* *Is it made clear how the input data or datasets will be handled by the algorithms and techniques chosen?*

**Benchmark**

As my benchmark model, I built a simple CNN architecture, trained it on the training set and measured its performance by calculating the F1 score. With over 405k+ trainable parameters, the model’s F1 score is 0.687237026648.

The model has the following architecture:



**III. Methodology**

*(approx. 3-5 pages)*

**Data Preprocessing**

Loading the dataset

The dataset was downloaded from Kaggle. The path of the dataset was fed into ‘load\_dataset’ function that returns a dictionary containing the list of folder names (the category names) as ‘target’, and list of all the individual file names as ‘filenames’.

One-hot encoding

The ‘target’ values are one-hot encoded using np.util.to\_categorical function and returned as an array of one-hot encoded ‘y\_targets’ vector.

Train-validation-test split

The dataset was first split in the ratio of 85:15 as training-validation and test set. The training-validation set containing 4037 images were further split into training and validation set in the ratio 80:20 giving 3229 images for training set and 808 for validation set. The test set with 713 images and labels pair was kept untouched for final evaluation the classifier.

**Implementation**

In this section, the process for which metrics, algorithms, and techniques that you implemented for the given data will need to be clearly documented. It should be abundantly clear how the implementation was carried out, and discussion should be made regarding any complications that occurred during this process. Questions to ask yourself when writing this section:

* *Is it made clear how the algorithms and techniques were implemented with the given datasets or input data?*
* *Were there any complications with the original metrics or techniques that required changing prior to acquiring a solution?*
* *Was there any part of the coding process (e.g., writing complicated functions) that should be documented?*

**Refinement**

In this section, you will need to discuss the process of improvement you made upon the algorithms and techniques you used in your implementation. For example, adjusting parameters for certain models to acquire improved solutions would fall under the refinement category. Your initial and final solutions should be reported, as well as any significant intermediate results as necessary. Questions to ask yourself when writing this section:

* *Has an initial solution been found and clearly reported?*
* *Is the process of improvement clearly documented, such as what techniques were used?*
* *Are intermediate and final solutions clearly reported as the process is improved?*

**IV. Results**

*(approx. 2-3 pages)*

**Model Evaluation and Validation**

In this section, the final model and any supporting qualities should be evaluated in detail. It should be clear how the final model was derived and why this model was chosen. In addition, some type of analysis should be used to validate the robustness of this model and its solution, such as manipulating the input data or environment to see how the model’s solution is affected (this is called sensitivity analysis). Questions to ask yourself when writing this section:

* *Is the final model reasonable and aligning with solution expectations? Are the final parameters of the model appropriate?*
* *Has the final model been tested with various inputs to evaluate whether the model generalizes well to unseen data?*
* *Is the model robust enough for the problem? Do small perturbations (changes) in training data or the input space greatly affect the results?*
* *Can results found from the model be trusted?*

**Justification**

In this section, your model’s final solution and its results should be compared to the benchmark you established earlier in the project using some type of statistical analysis. You should also justify whether these results and the solution are significant enough to have solved the problem posed in the project. Questions to ask yourself when writing this section:

* *Are the final results found stronger than the benchmark result reported earlier?*
* *Have you thoroughly analyzed and discussed the final solution?*
* *Is the final solution significant enough to have solved the problem?*

**V. Conclusion**

*(approx. 1-2 pages)*

**Free-Form Visualization**

In this section, you will need to provide some form of visualization that emphasizes an important quality about the project. It is much more free-form, but should reasonably support a significant result or characteristic about the problem that you want to discuss. Questions to ask yourself when writing this section:

* *Have you visualized a relevant or important quality about the problem, dataset, input data, or results?*
* *Is the visualization thoroughly analyzed and discussed?*
* *If a plot is provided, are the axes, title, and datum clearly defined?*

**Reflection**

In this section, you will summarize the entire end-to-end problem solution and discuss one or two particular aspects of the project you found interesting or difficult. You are expected to reflect on the project as a whole to show that you have a firm understanding of the entire process employed in your work. Questions to ask yourself when writing this section:

* *Have you thoroughly summarized the entire process you used for this project?*
* *Were there any interesting aspects of the project?*
* *Were there any difficult aspects of the project?*
* *Does the final model and solution fit your expectations for the problem, and should it be used in a general setting to solve these types of problems?*

**Improvement**

As an improvement and future work, I would like to try data masking on the training set. Noise from the background of the images can be cancelled by masking images. I believe that without the background noise and restricting the visibility to the green leaves, the model can be trained better, and we may notice significant improvement in the performance.

Another implementation that can be tried is data augmentation. As the dataset is highly unbalanced, augmenting data to the under-represented classes might give a good boost to the total number of training images yielding a well-balanced dataset. Training the model on such dataset may give us significant improvement in the performance.

**Before submitting, ask yourself. . .**

* Does the project report you’ve written follow a well-organized structure similar to that of the project template?
* Is each section (particularly **Analysis** and **Methodology**) written in a clear, concise and specific fashion? Are there any ambiguous terms or phrases that need clarification?
* Would the intended audience of your project be able to understand your analysis, methods, and results?
* Have you properly proof-read your project report to assure there are minimal grammatical and spelling mistakes?
* Are all the resources used for this project correctly cited and referenced?
* Is the code that implements your solution easily readable and properly commented?
* Does the code execute without error and produce results similar to those reported?