Capstone Project II: Milestone Report

E. Erik Larsen

January 6th Cohort

Problem Statement & Motivation

Throughout history gardening and crop production has been plagued by the presence of undesired plant species mixing with the desired yield. The advent of machine learning makes it possible to identify faces and objects, and even plant species, by picture alone. The ability to discern undesirable infiltrators in personal gardens and production level farms could greatly improve the performance of both home gardening and large scale agricultural operations. By simply taking a picture of a seedling, these problematic 'weeds' can be identified and removed before robbing the desired plants of nutrients, space, and sunlight. Some immediate benefits include an increase in crop health and yield, a decrease in man-hours spent eliminating weeds, and could possibly lead to a decrease in the herbicides used to treat large cash crops.

Data Description

The data set can be found on the Kaggle website at https://www.kaggle.com/c/plant-seedlings-classification/data. It consists of pictures of twelve different species of plant seedlings. These include vital cash crops such as Maize, and undesirable weeds such as the Chickweed picture seen below.

Chickweed example using the PIL Image function

This set is large, and takes up approximately 1.69GB when compressed on the website for download. There are a total of 4750 pictures of 960 individual plants. These have been labeled and sorted into specific folders in Kaggle, setting the stage for supervised learning techniques. To complicate the matter, each picture is of a different size; there is no uniformity within or among the separate species. The largest picture belongs to the Loose Silky-bent species, with dimensions of 3457x3457; the smallest pictures are 49x49 and belong to both

Scentless Mayweed and Sugar Beet varieties. They are color pictures, so each also has three layers corresponding to the red, green, and blue intensities.

Each plant species also has a different number of pictures. Loose Silky-bent again has the largest number, with 654 pictures. The smallest number belongs to both Maize and Common Wheat varieties, with only 221 pictures in each. The average picture size in each species ranges from 226x226 to 661x661, belonging to Scentless Mayweed and Black-grass, respectively. The median average picture size across all plants is 370x370. Clearly reshaping must be done to make the pictures uniform for a neural network algorithm to examine, while ensuring that the preponderance of data from the larger specimens is not lost.

Data Wrangling

The first step is to download the entire data set from Kaggle. Because of its size, this can take several minutes and uses a lot of internal storage space. Several packages can be used to examine the pictures using Python, including PIL, skimage, and the standard matplotlib pyplot libraries. Resizing the pictures to uniform dimensions is accomplished with PIL's Image.resize() function, with chosen dimensions of 256x256. These can then be transformed into arrays via the numpy asarray() function, making the numerical data easy to analyze and manipulate. The data type for all pictures is then confirmed as numpy.float32.

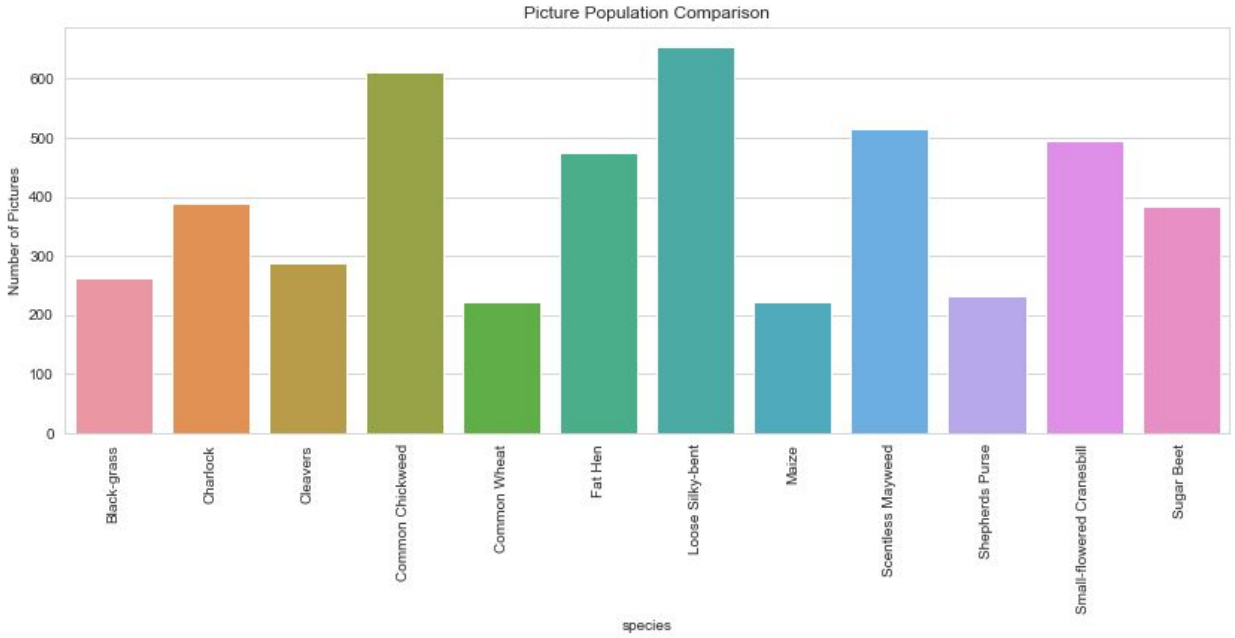
Next, it is necessary to load each picture into a list which is sorted by species. This is accomplished using the glob.glob() function to retrieve the data from the local drive. The result is a rather large list of twelve lists, each containing its own species' pictures as numerical arrays. The Shepherd's Purse variety did not immediately load, prompting investigation as to why these pictures were left out when all of the others loaded without issue.

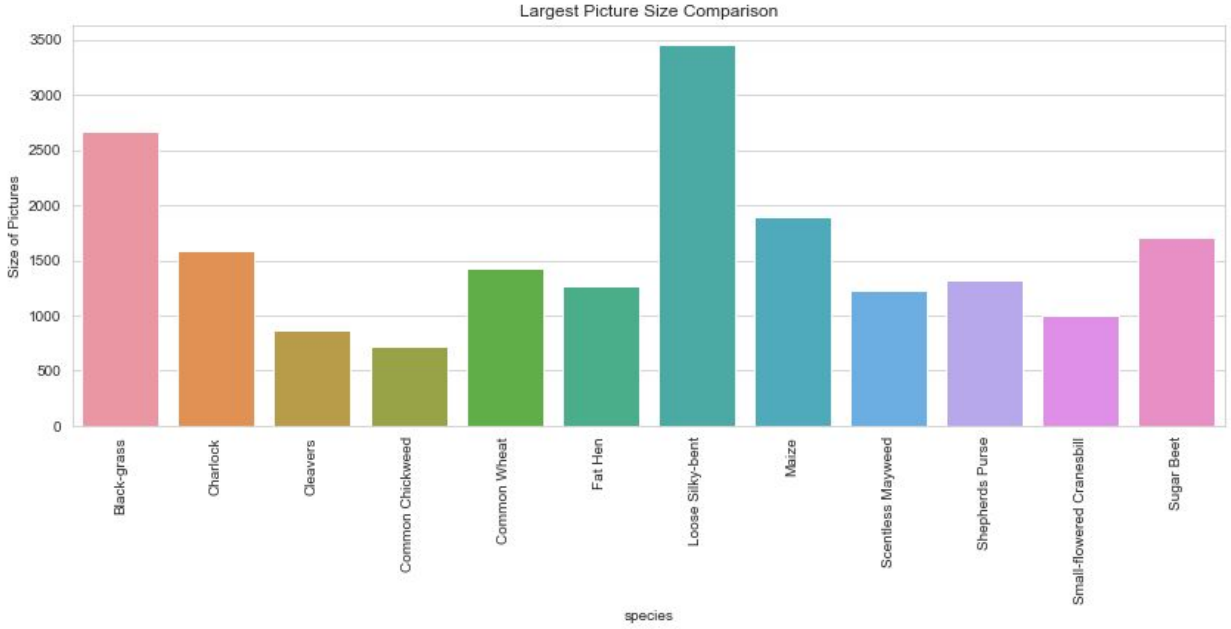
Once loaded in this list format (which takes approximately 15 - 25 minutes) the pictures for each species can be counted and analyzed for largest and smallest sizes, etc., as mentioned above. Some simple statistics about the data set are then calculated and placed into a data frame for quick observation and easy plotting.

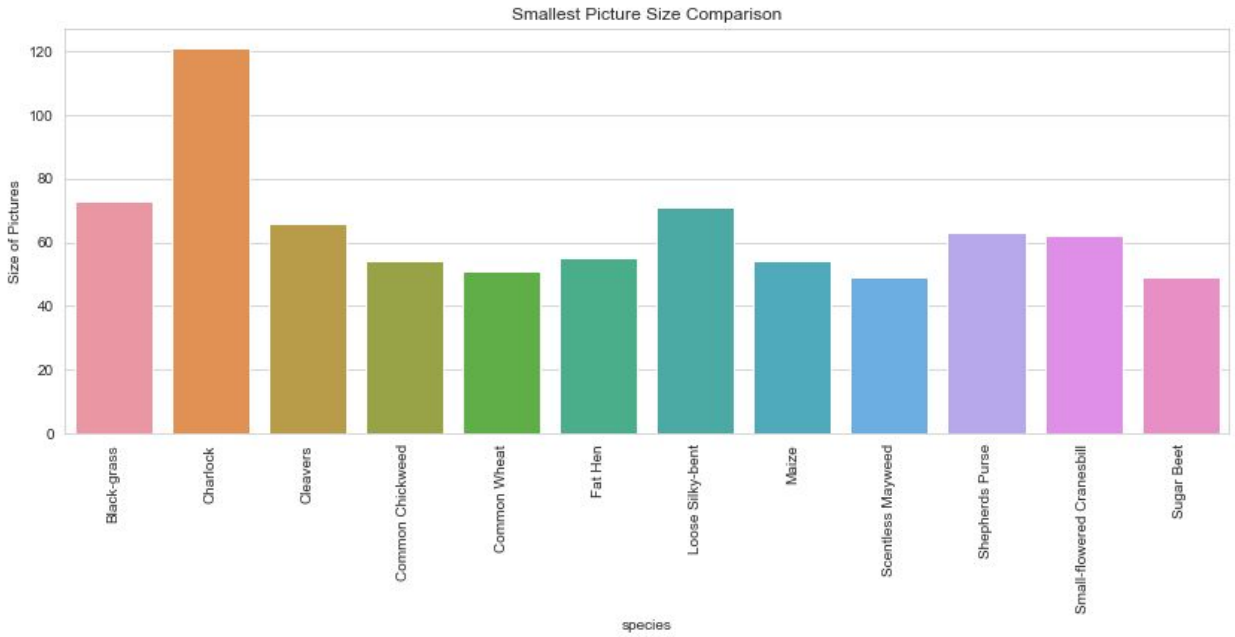
	species	num_pics	largest_pic	smallest_pic	avg_pic_size	std_dev	data_type
0	Black-grass	263	2670	73	661	498.0	<class 'numpy.float32'>
1	Charlock	390	1582	121	432	287.0	<class 'numpy.float32'>
2	Cleavers	287	866	66	310	152.0	<class 'numpy.float32'>
3	Common Chickweed	611	718	54	231	161.0	<class 'numpy.float32'>
4	Common Wheat	221	1432	51	440	303.0	<class 'numpy.float32'>
5	Fat Hen	475	1273	55	303	211.0	<class 'numpy.float32'>
6	Loose Silky-bent	654	3457	71	398	409.0	<class 'numpy.float32'>
7	Maize	221	1900	54	483	416.0	<class 'numpy.float32'>
8	Scentless Mayweed	516	1227	49	226	212.0	<class 'numpy.float32'>
9	Shepherds Purse	231	1317	63	289	248.0	<class 'numpy.float32'>
10	Small-flowered Cranesbill	496	1006	62	342	194.0	<class 'numpy.float32'>
11	Sugar Beet	385	1715	49	590	335.0	<class 'numpy.float32'>

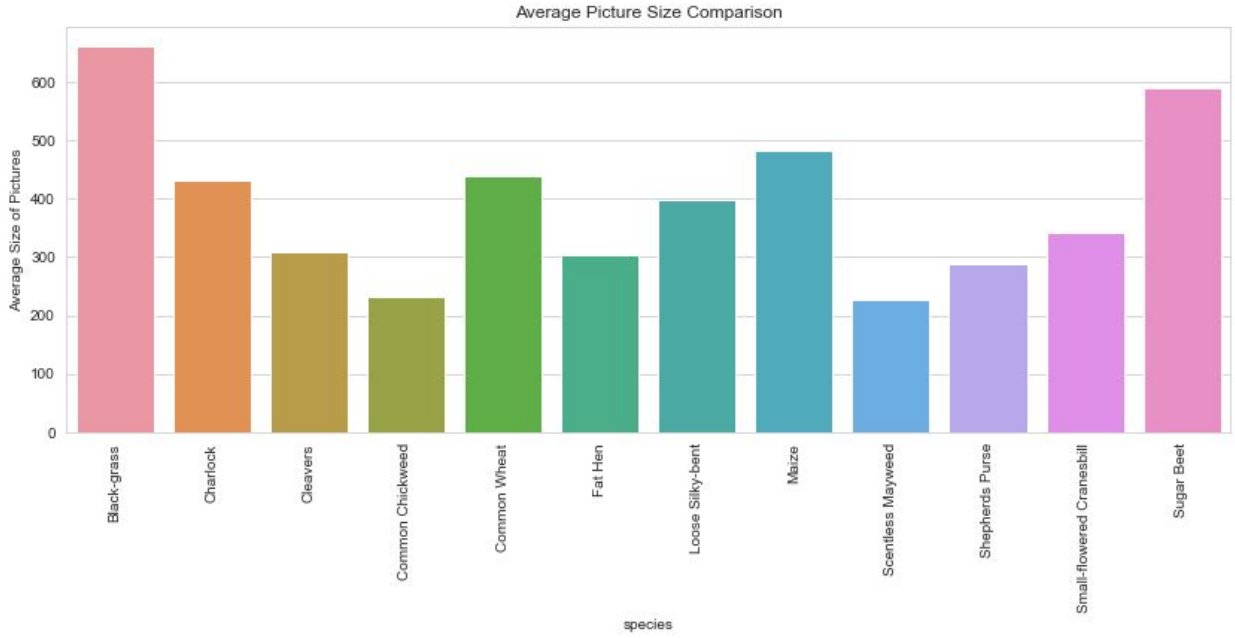
Exploratory Data Analysis

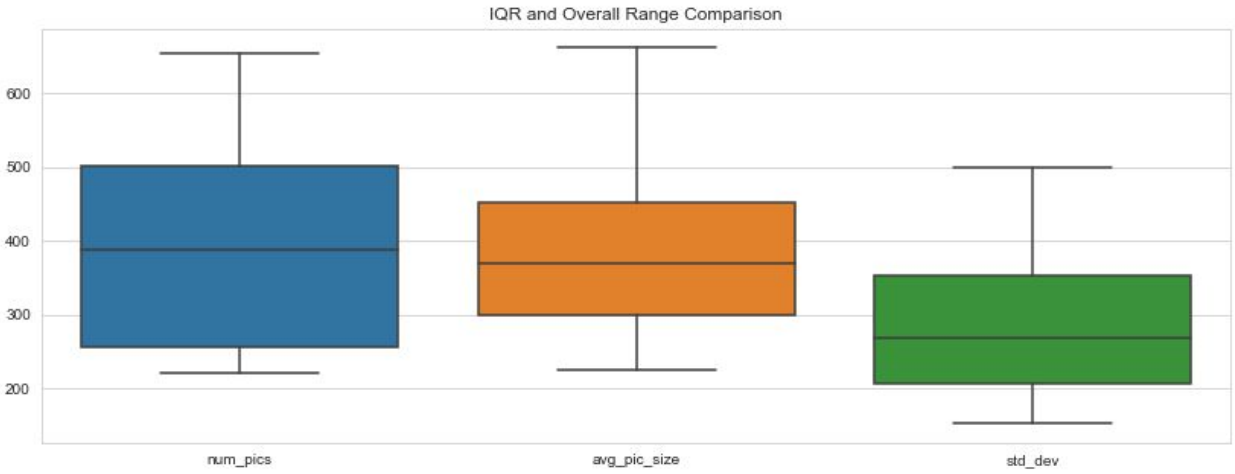
The differences in each group are easily seen when visualized. The following plots show that the data is not uniform in any sense, and that some correction is required to ensure they all have the same input size for a neural network. The differing number of pictures in each group will necessitate the use of the ‘stratify’ hyperparameter when creating a neural net.





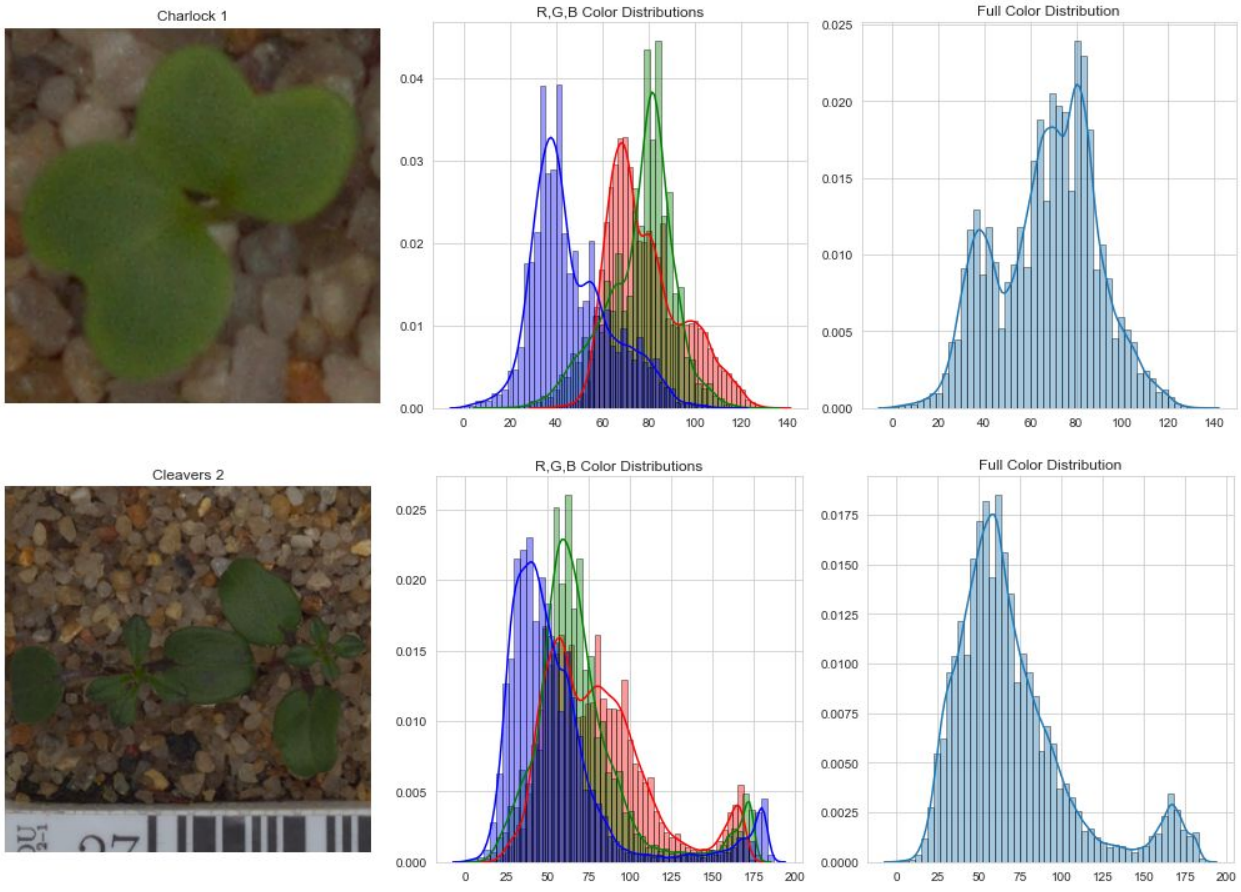

Smallest Picture Size Comparison
Size of Pictures
0
20
40
60
80
100
120
Black-grass
Charlock
Cleavers
Common Chickweed
Common Wheat
Fat Hen
Loose Silky-bent
Maize
Scentless Mayweed
Shepherds Purse
Small-flowered Cranesbill
Sugar Beet
species


Average Picture Size Comparison
Average Size of Pictures
0
100
200
300
400
500
600
Black-grass
Charlock
Cleavers
Common Chickweed
Common Wheat
Fat Hen
Loose Silky-bent
Maize
Scentless Mayweed
Shepherds Purse
Small-flowered Cranesbill
Sugar Beet
species

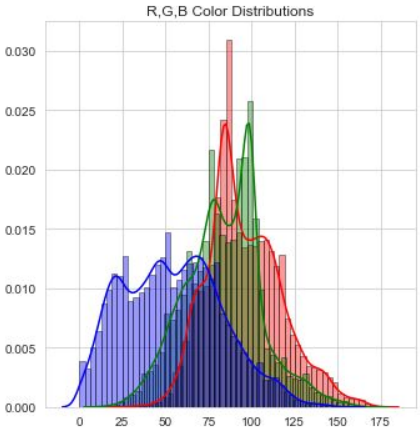


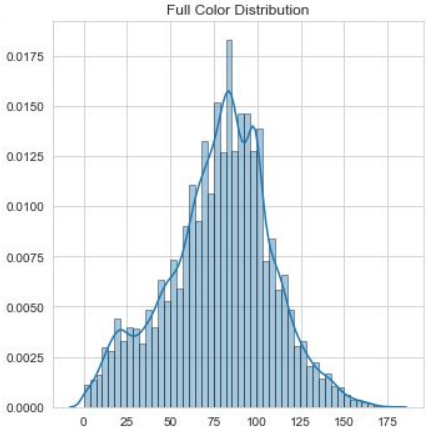
Pixel Distribution Comparisons

Each picture has its own unique distribution of red, green, and blue pixels. Comparing these visually could give insight into feature engineering or other ways to improve model performance. Examples of pictures and their respective distributions clearly show this.

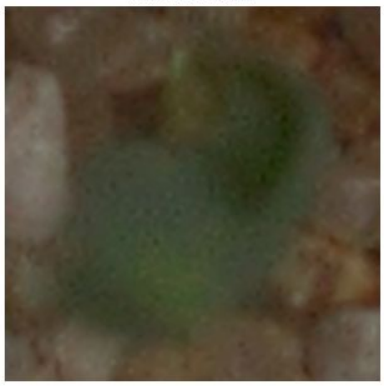


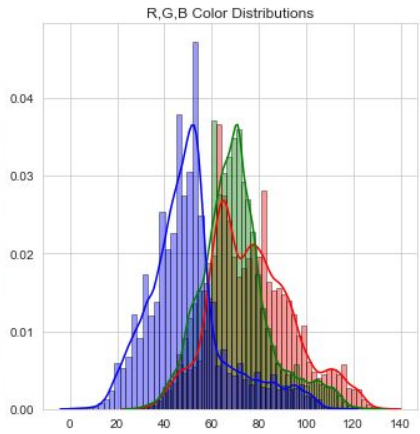
Common Chickweed 3

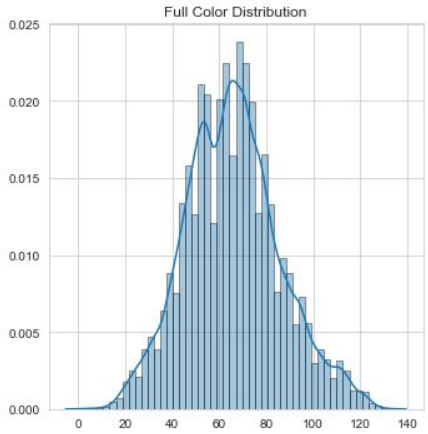




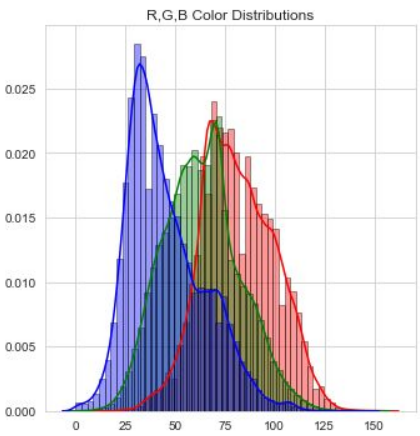
Common Wheat 4

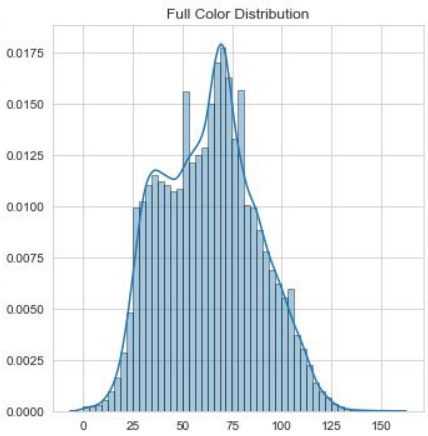




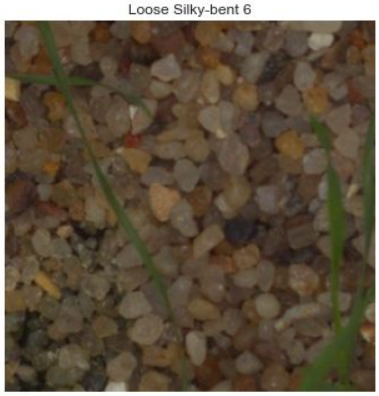


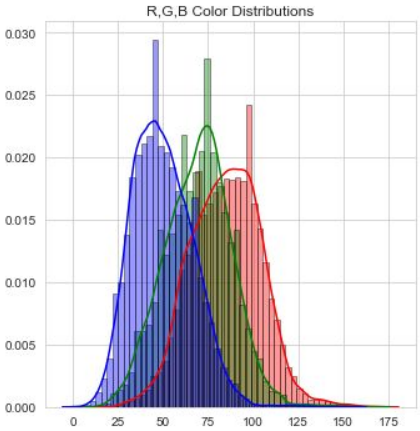
Fat Hen 5

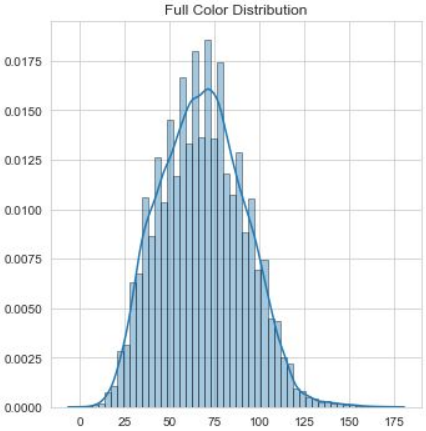




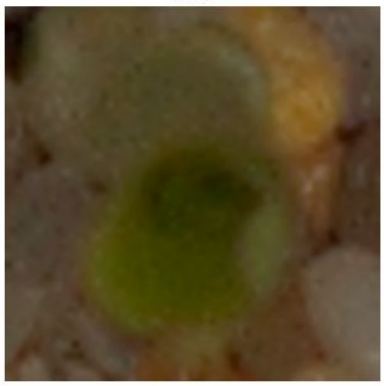
Loose Silky-bent 6

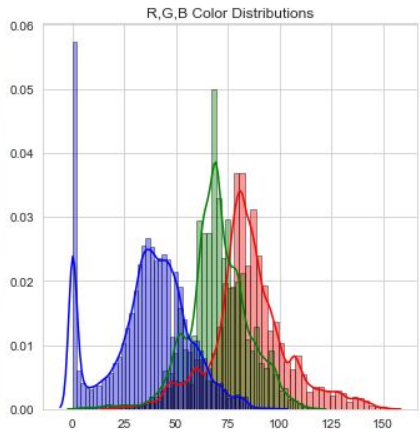


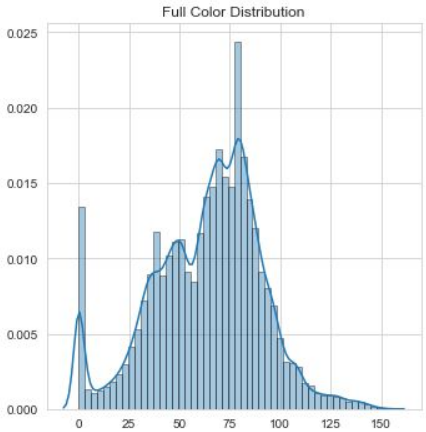




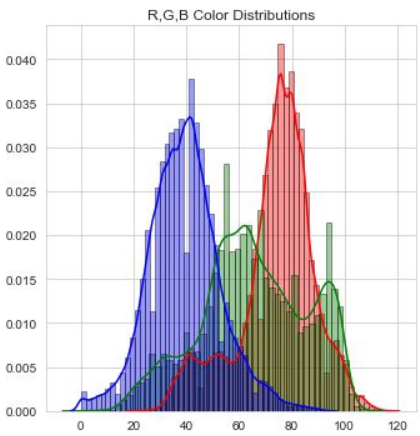
Maize 7

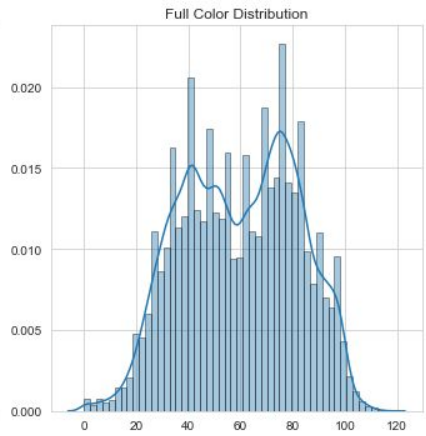


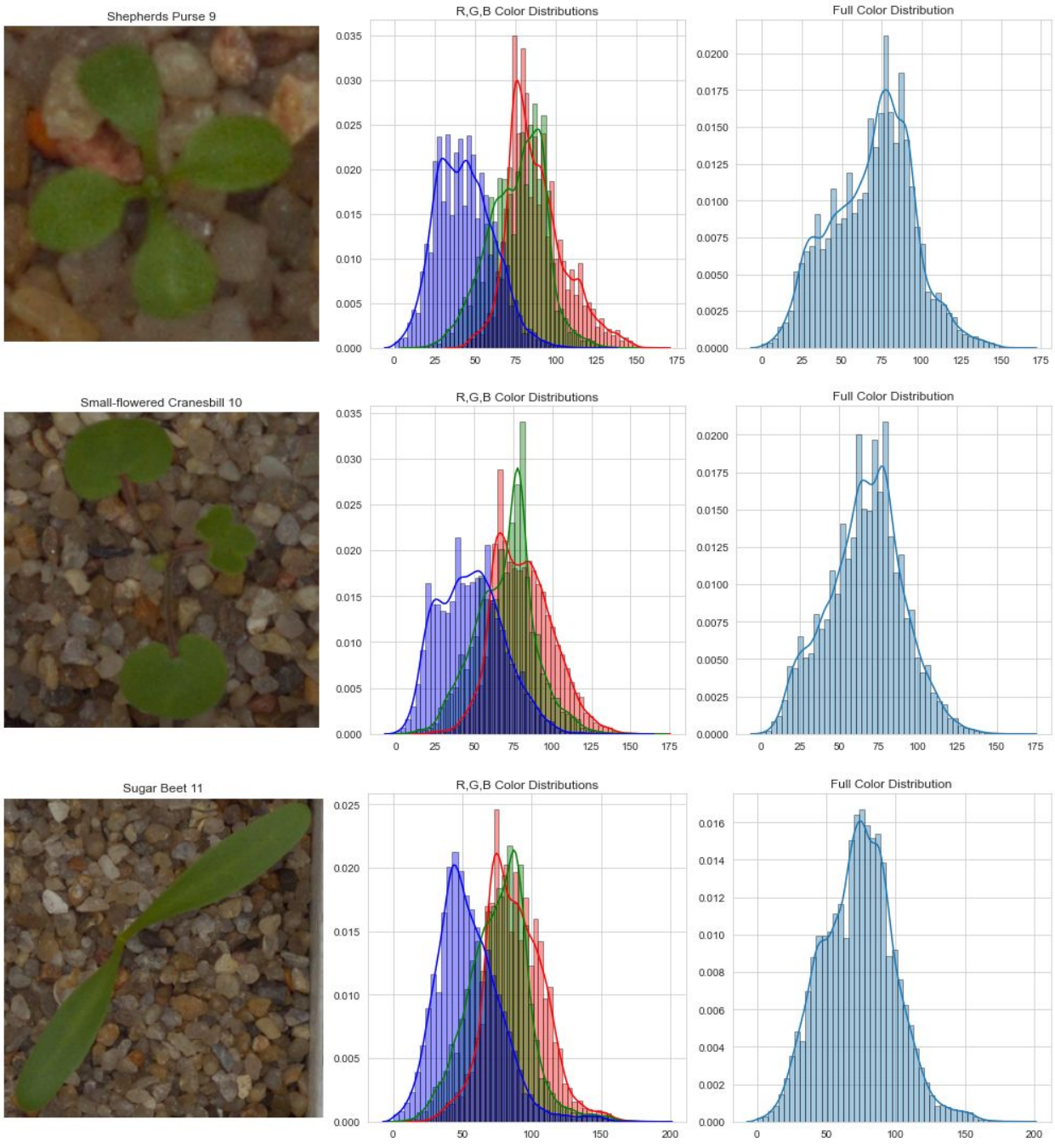




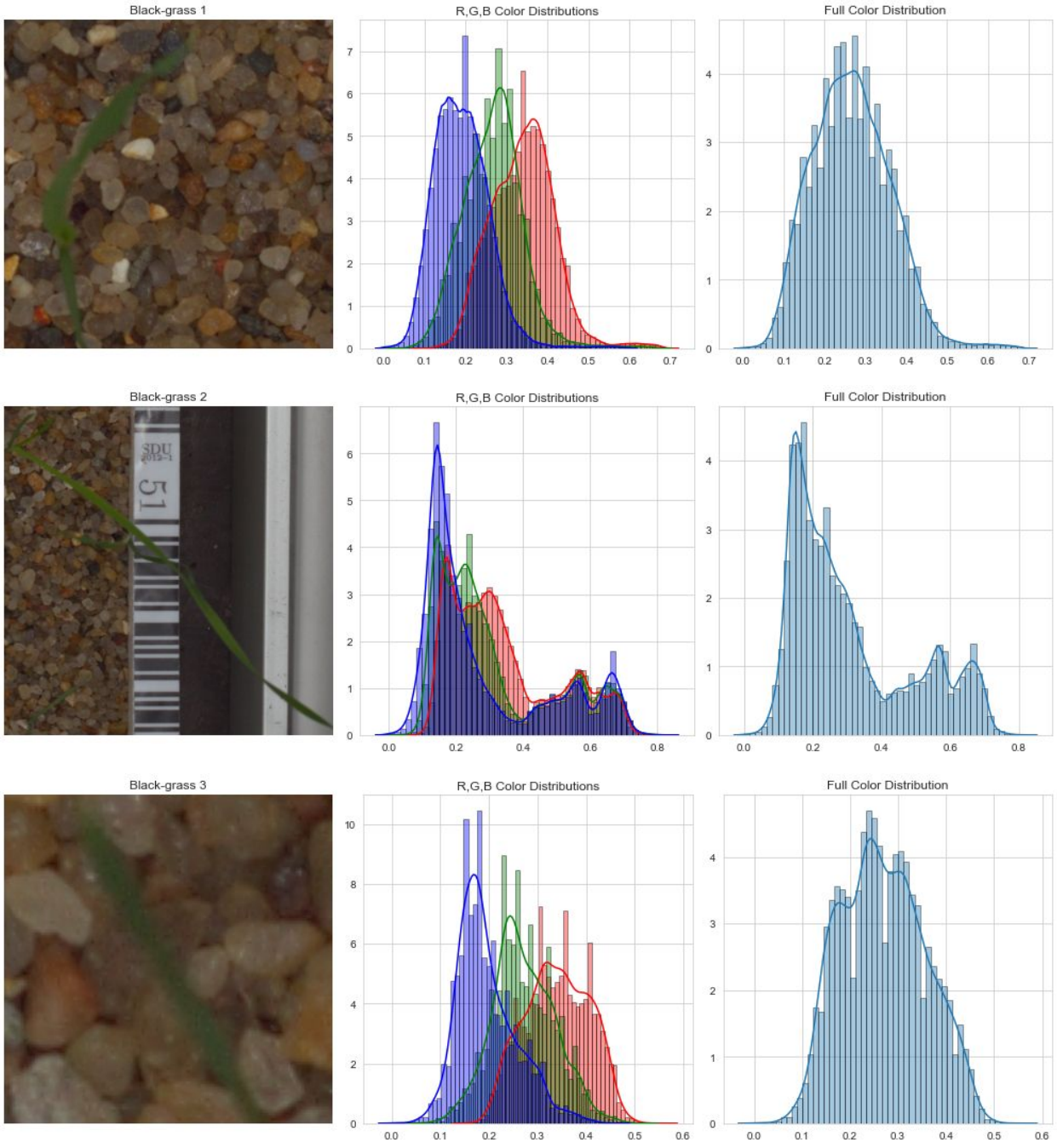
Scentless Mayweed 8



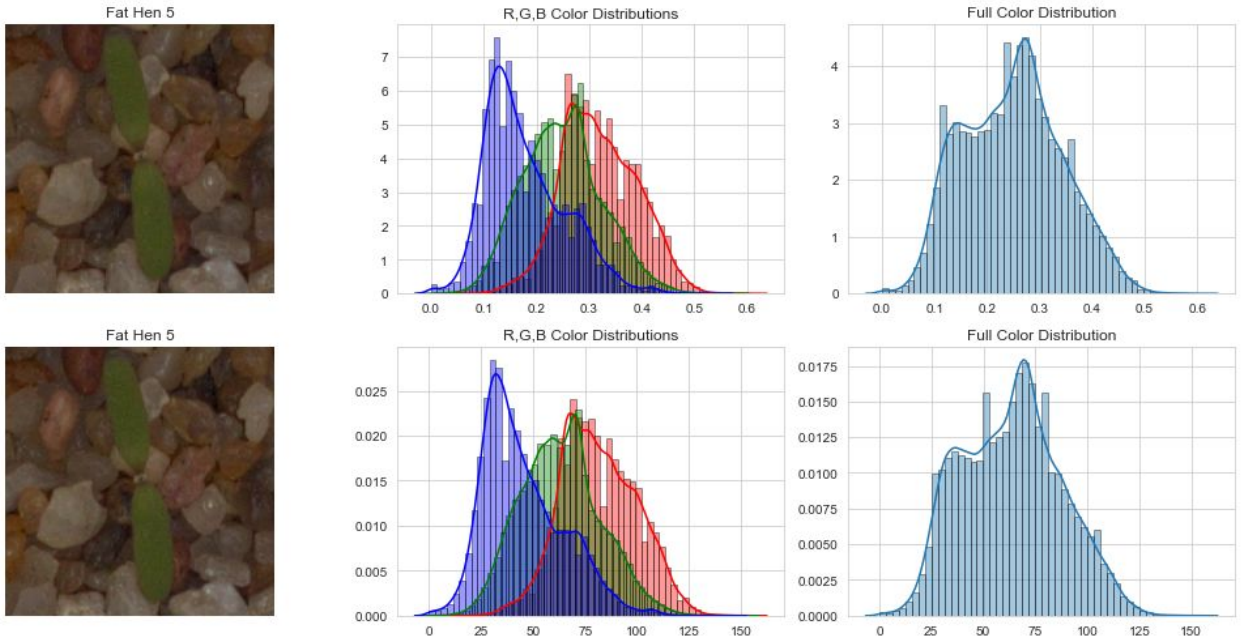




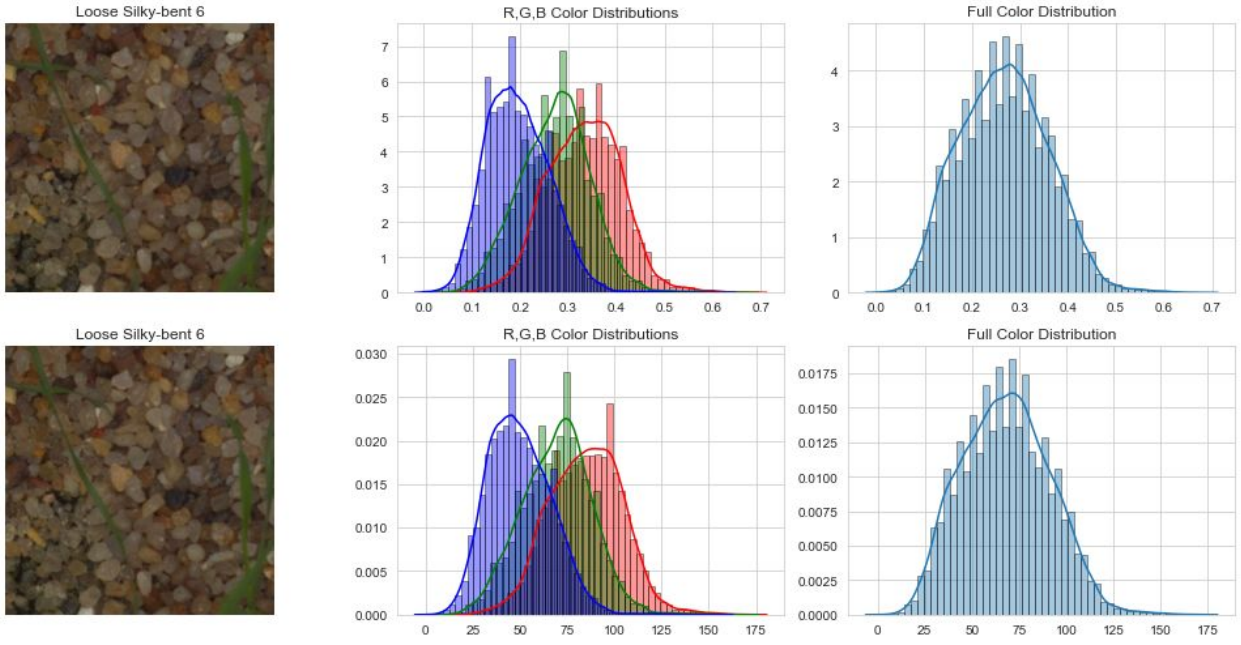
Even plants of the same species show differences between the plants at different stages of growth, as seen below with Blackgrass.



Comparing the distributions before and after reshaping shows that the overall distributions shapes remain basically the same, but very small differences are noticeable. This will hopefully translate to a very small loss of data from the original picture if it was quite large.



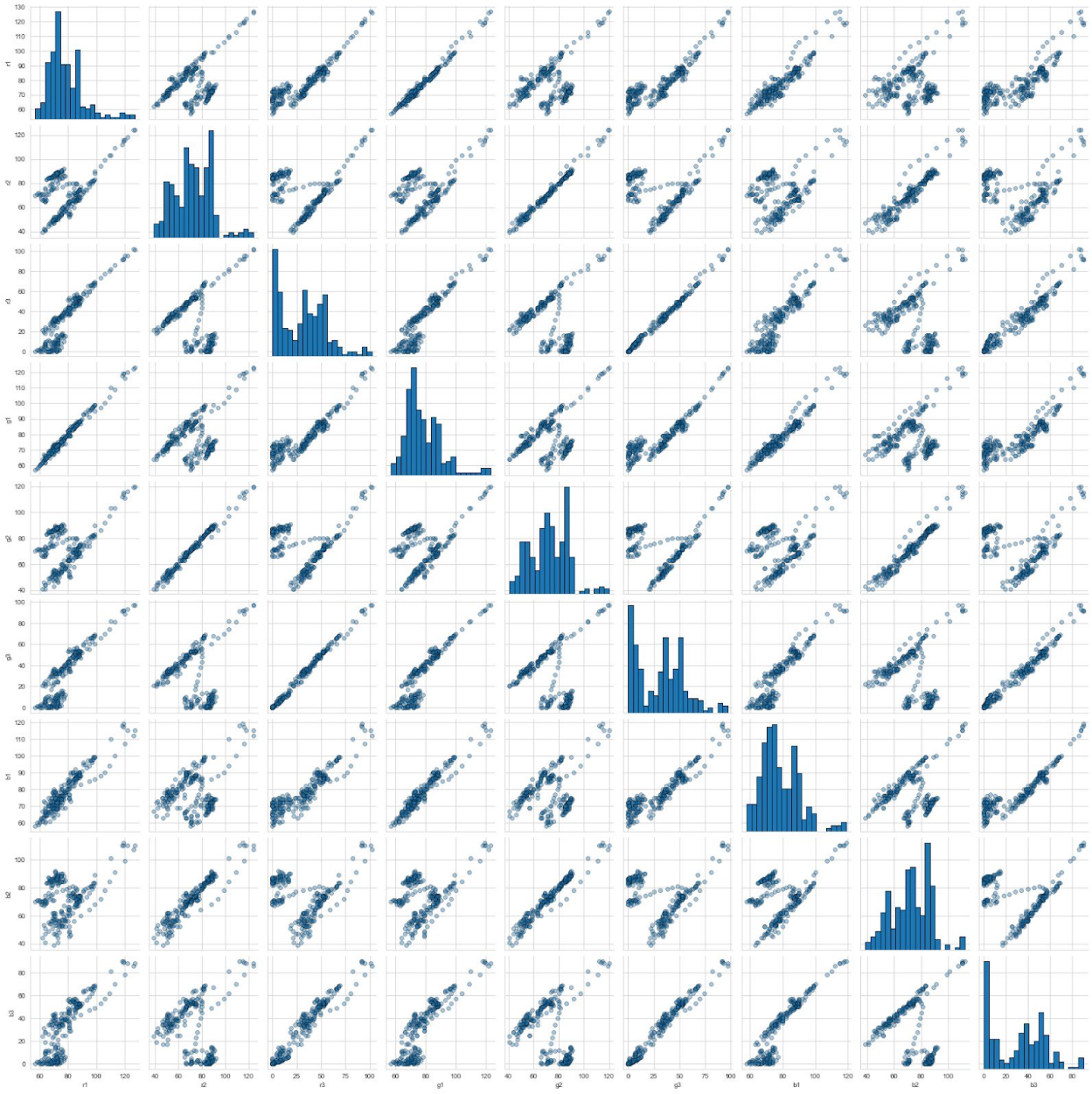
Top: original picture Bottom: reshaped picture



Top: original picture Bottom: reshaped picture

Scatterplots

Plotting the red, green, and blue pixels against each other in a scatterplot shows clear linear correlation. Some of the plots, however, have clusters of points that are not located near the main line-shaped grouping. A similar trend across plants of all species may help an algorithm to decipher the correct species when trained.



Common Chickweed pixels compared in scatterplots

Conclusion & Next Steps

The data is not uniform and requires manipulation to ready it for training neural network models. First, each picture must be reshaped to the same size, which has been chosen to be 256x256. Second, because the number of pictures is not uniform across all classes, the stratify method must be used when creating training and test sets. EDA shows differences in both original and reshaped pixel distributions between the species and individual plants of the same species. Reshaped distributions closely matched the original picture distributions, which is indicative of negligible data loss or inference.Next, several different neural network models will be trained. These will include simple to more complex hidden layer architectures, and

convolutional neural networks (CNN). In addition to different models, various numbers of filters, edge detectors, and other well known computer vision techniques will be explored in order to achieve the best performance.