Plant Seedling Classification

Project #5 AIML for Business

Matthew Clark
July 13, 2024

Contents / Agenda

Executive Summary

- Agriculture is a trillion-dollar industry that is ready for technological innovation. Many current practices are heavily dependent on manual labor. One such task is identifying different plant seedlings and weeds. Although there have been many advances in agricultural technologies plant identification is an area that is ripe for improvement.

- The introduction of Artificial Intelligence and Deep Learning could greatly reduce the time, efforts, and labor needed to complete seedling identification through Computer Vision.

- We recommend that agricultural operators implement Computer Vision using Convolutional Neural Network modeling to identify plant seedlings.

- Human labor is expensive and sometimes difficult to find in agricultural areas. Implementing these improved efficiencies through Computer Vision could free up experienced human labor for higher order decision making and plant care, resulting in better crop yields, and long-term sustainability.

- Agriculture should embrace the power of Computer Vision to ensure long-term sustainability.

Business Problem Overview and Solution Approach

- Many tasks in agriculture are heavily dependent on manual labor. This includes plant seedling identification which is often done by experienced human workers. The introduction of Artificial Intelligence and Deep Learning has the potential to greatly improve the efficiencies of plant seedling identification. The savings in workers time and energy could provide benefits across the entire operation, freeing human labor up for the most important tasks, and this could result in improved crop yield and sustainability.

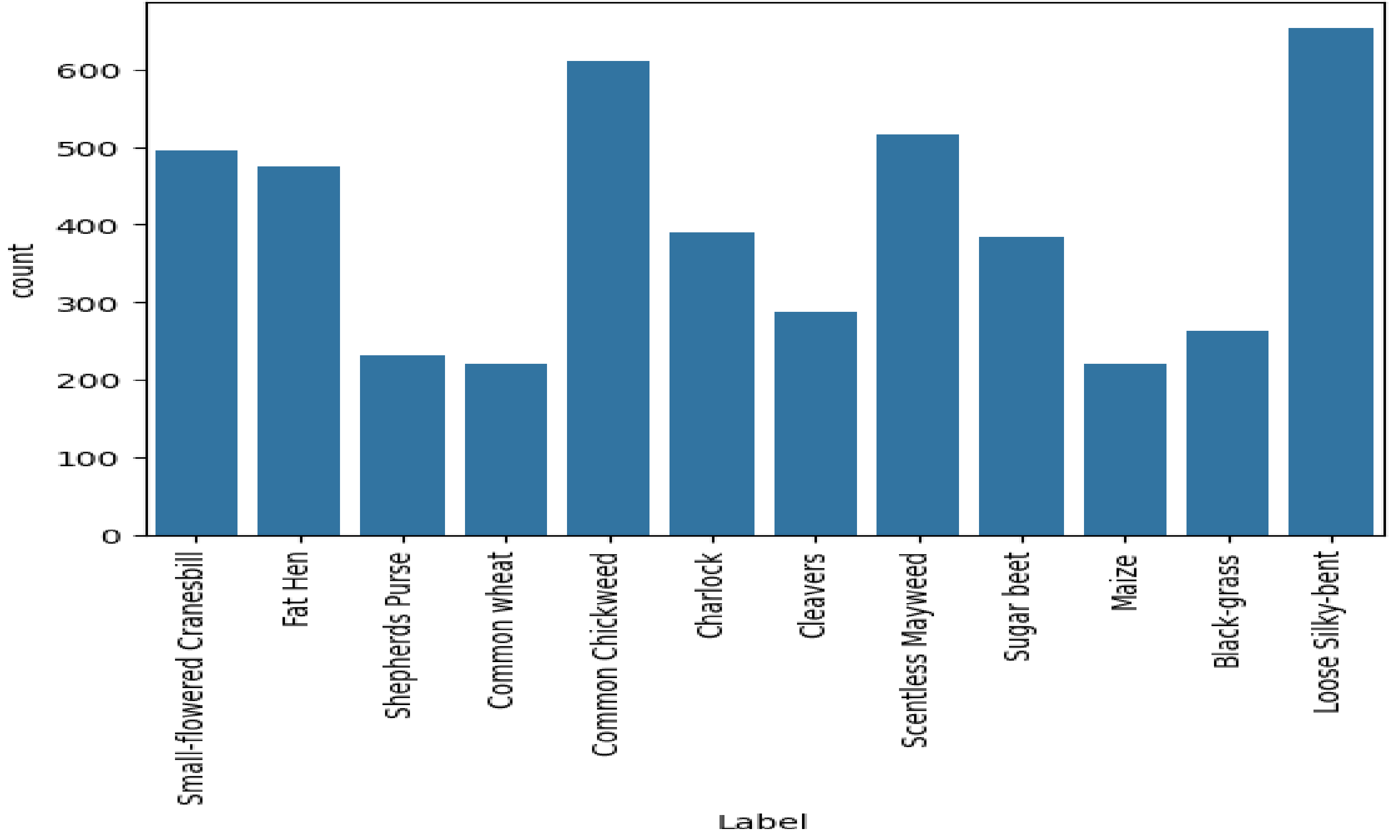
- The Data Science Team has been tasked with creating a Convolutional Neural Network model which will classify plant seedlings into their respective 12 categories.

- Our approach to this project will begin with processing and analyzing the dataset to ensure accuracy and best modeling. Next, we will construct a Convolutional Neural Network model and train and test the dataset. Based on the results of that model we will then tune the hyperparameters of the CNN model with a second model to improve the overall performance.

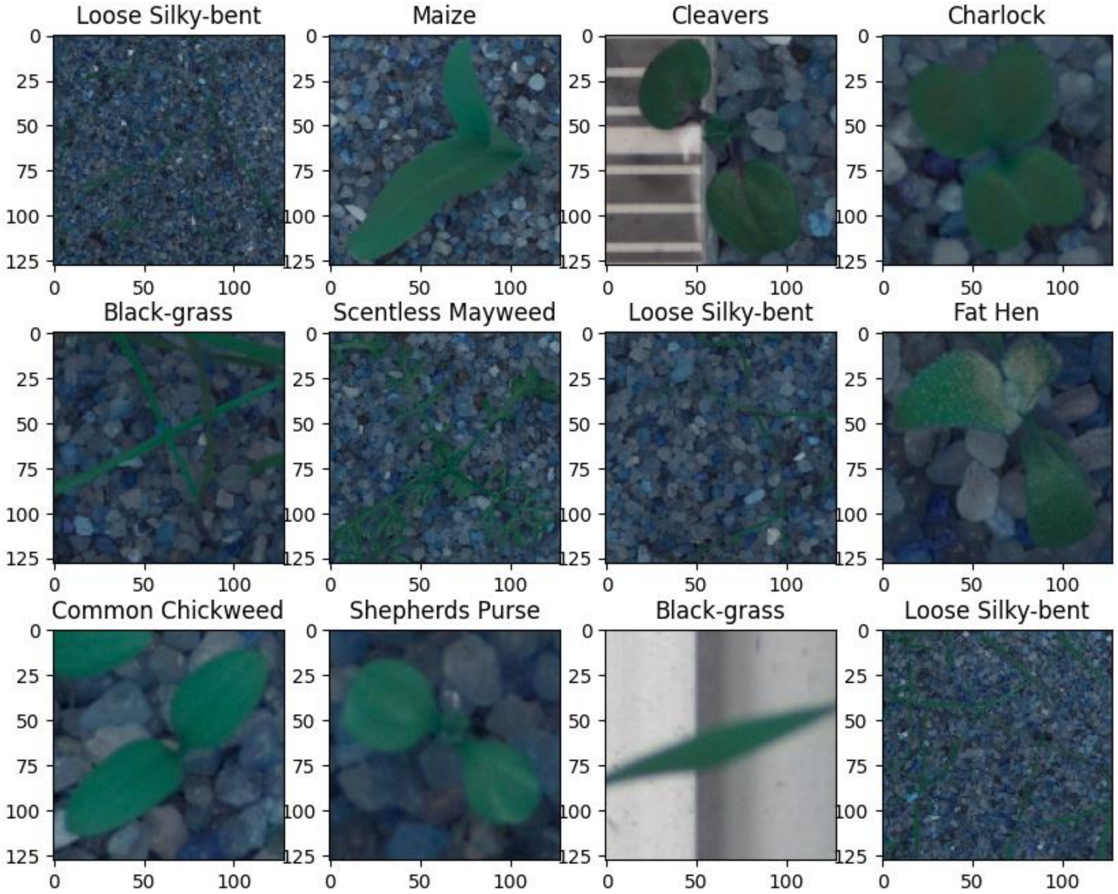
EDA Results

- The data set contains 4750 images.
- The data set provided is imbalanced among the 12 different types of seedlings being classified.
- Some classes feature more than 600 images while others have just over 200 images.
- Randomly generated images from the data set show the plant seedlings have a variations in size and shape.

Distribution of Target Variables



Plotting Random Images from each Class

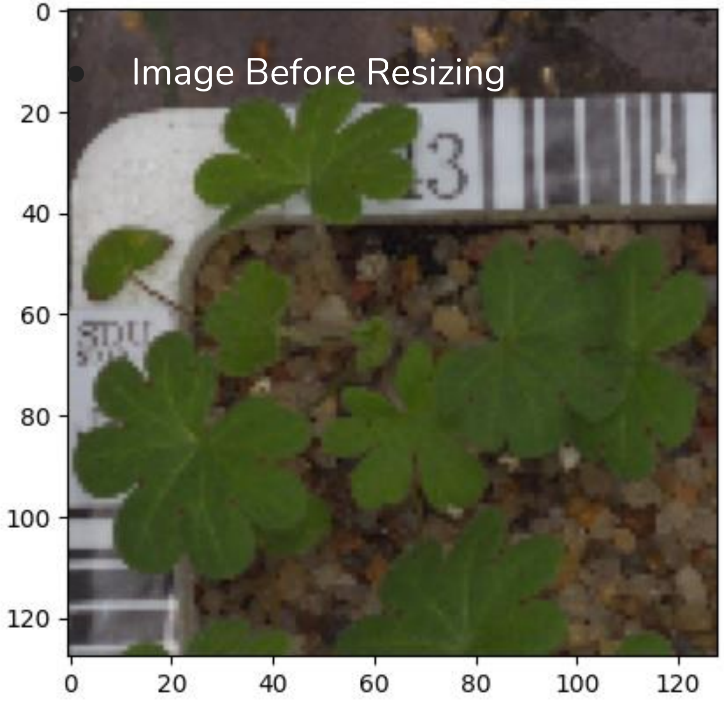


Data Preprocessing

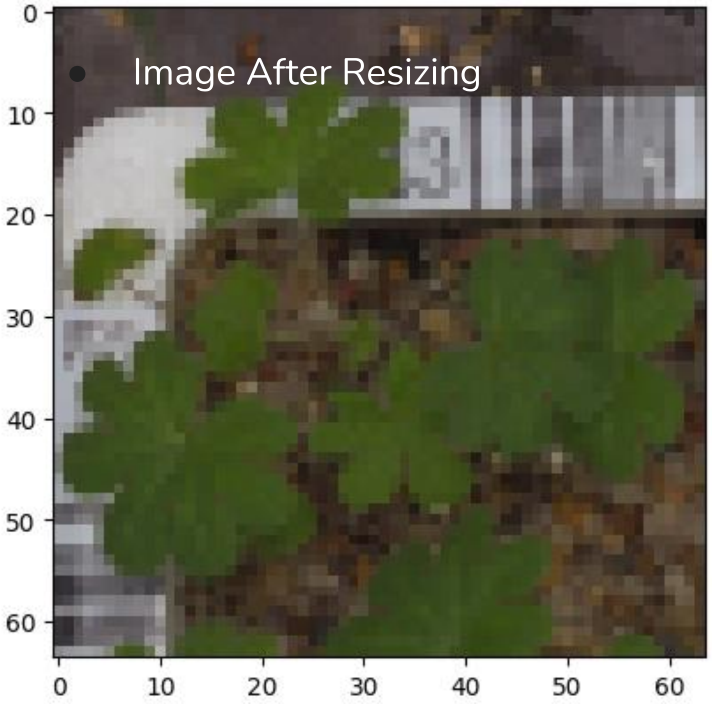
- First the images in the data were converted from BGR to RGB using the cvtColor function of OpenCV.
- Next, we resized the images. The size of the images in the data set is large, and as a result the computational costs could be high. Reducing the size will prevent high costs. The images in the data set were sized at 128. We reduced the size to 64.
- Before more processing the data set was split into training, validation, and testing. 80% for training, 10% for validation, and 10% for testing.
- The next step was to encode the target labels using labelBinarizer.
- Finally we completed data normalization using scaling. By dividing the pixel values by 255 to standardize the images to have values betweed 0 and 1.

Resized Photos

- Image Before Resizing



- Image After Resizing



Model Performance Summary

- Model 1 is a Convolutional Nueral Network.
- There are 3 Convolutional layers with different numbers of filters, a 3x3 kernel size, and 2x2 padding using Relu for activation.
- There are 3 Max Pooling layers to provide the output size remains the same as the input size.
- The next layer flattens the convolutional layers output to make it ready for dense connections.
- Next is a dense layer with 16 nuerons using Relu for activation.
- This is followed by a dropout layer with a dropout rate of 0.3
- And finally an output layer using softmax for activation.
- The model has a batch size of 32 with 30 epochs. The optimizer is Adam and the loss function is categorical cross entropy.

Model 1 Summary

CONVOLUTIONAL NUERAL NETWORK - MODEL 1		
LAYER	**OUTPUT SHAPE**	**PARAM#**
conv2d (Conv2D)	(None, 64, 64, 128)	3584
max_pooling2d (MaxPooling2D)	(None, 32, 32, 128)	0
conv2d_1 (Conv2D)	(None, 32, 32, 64)	73792
max_pooling2d_1 (MaxPooling2D)	(None, 16, 16, 64)	0
conv2d_2 (Conv2D)	(None, 16, 16, 32)	18464
max_pooling2d_2g2D	(None, 8, 8, 32)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 16)	32784
dropout (Dropout)	(None, 16)	0
dense_1 (Dense)	(None, 12)	204
Total Parameters:	128828	
Trainable Parameters:	128828	
Non-Trainable Parameters:	0	

Model 1 Classification Report

CNN - MODEL 1				
Classes	Precision	Recall	F1-Score	Support
0	0.00	0.00	0.00	26
1	0.79	0.67	0.72	39
2	0.69	0.76	0.72	29
3	0.81	0.90	0.85	61
4	0.00	0.00	0.00	22
5	0.71	0.77	0.74	48
6	0.52	0.98	0.68	65
7	0.86	0.55	0.67	22
8	0.65	0.63	0.64	52
9	0.69	0.39	0.50	23
10	0.87	0.78	0.82	50
11	0.64	0.74	0.68	38
Accuracy			0.68	475
Macro Avg	0.60	0.60	0.59	475
Weighted Avg	0.64	0.68	0.65	475

Model Tuning

- To improve the metrics of the model we made changes to Model 1 to create Model 2.
- The learning rate was slowed down using the ReduceLRonPlateau() function. This may start decreasing the loss at a smaller learning rate.
- Because the data set is imbalanced data augmentation was added. The data was augmented by setting the rotation range to 20.
- The filter sizes were lowered of the convolutional layers. One convolutional later was removed and replaced with a batch normalization layer.
- Activation and optimization remain the same with Relu and Adam.
- The batch size was increased to 64. The epochs remain at 30.

Model2 Summary

CONVOLUTIONAL NUERAL NETWORK - MODEL 2		
LAYER	**OUTPUT SHAPE**	**PARAM#**
conv2d (Conv2D)	(None, 64, 64, 64)	1792
max_pooling2d (MaxPooling2D)	(None, 32, 32, 64)	0
conv2d_1 (Conv2D)	(None, 32, 32, 32)	18464
max_pooling2d_1 (MaxPooling2D)	(None, 16, 16, 32)	0
batch_normalization (Batch Normalization)	(None, 16, 16, 32)	128
flatten (Flatten)	(None, 8192)	0
dense (Dense)	(None, 16)	131088
dropout (Dropout)	(None, 16)	0
dense_1 (Dense)	(None, 12)	204
Total Parameters:	151676	
Trainable Parameters:	151612	
Non-Trainable Parameters:	64	

Model 2 Classification Report

CNN - MODEL 2				
Classes	Precision	Recall	F1-Score	Support
0.00	0.25	0.12	0.16	26
1.00	0.82	0.92	0.87	39
2.00	0.70	0.72	0.71	29
3.00	0.93	0.90	0.92	61
4.00	0.71	0.68	0.70	22
5.00	0.72	0.75	0.73	48
6.00	0.71	0.82	0.76	65
7.00	0.70	0.95	0.81	22
8.00	0.77	0.85	0.81	52
9.00	0.73	0.48	0.58	23
10.00	0.93	0.80	0.86	50
11.00	0.79	0.82	0.81	38
Accuracy			0.77	475
Macro Avg	0.73	0.73	0.73	475
Weighted Avg	0.76	0.77	0.76	475

Conclusion

- Model 2 performed much better than Model 1.
- Tuning the model to account for the imbalance in the classes through data augmentation provided a large boost to accuracy.
- The accuracy improved from 68% to 77% on the training data.
- The test data accuracy on Model 2 was nearly 78%.
- Model 2 is the model that should be used for plant seedling identification.

APPENDIX

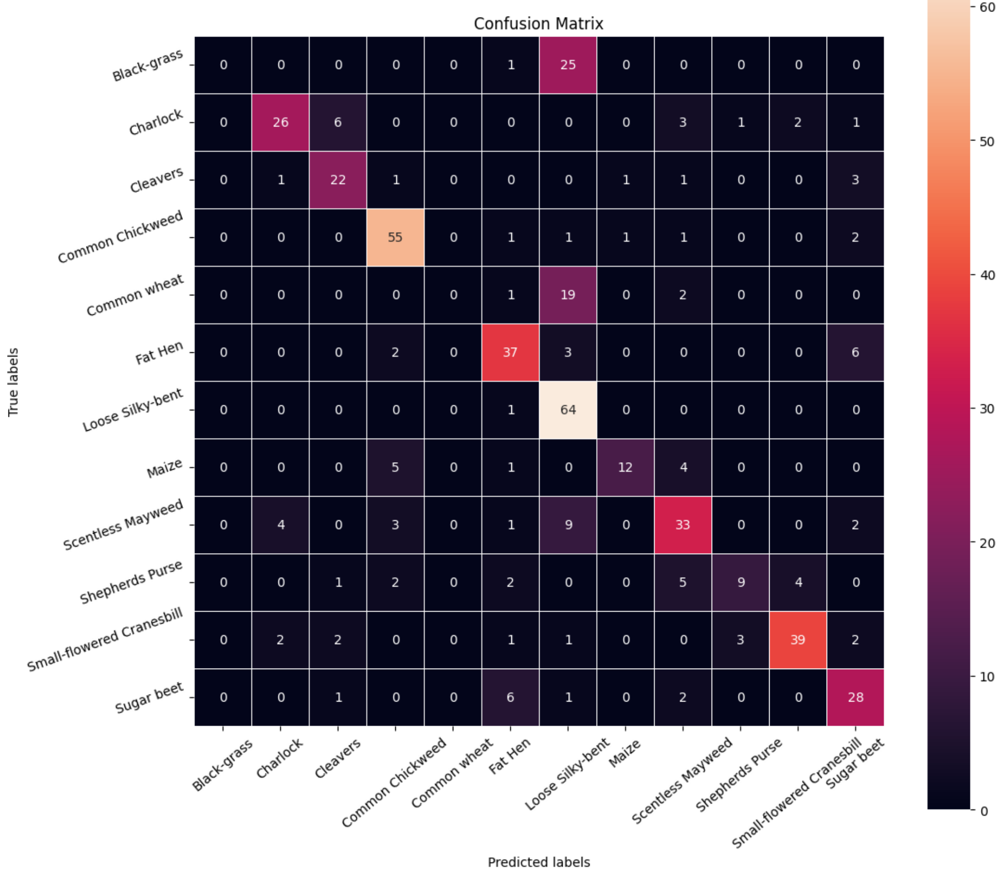
Data Background and Contents

- The Aarhus University Signal Processing group, in collaboration with the University of Southern Denmark, has provided the data containing images of unique plants belonging to 12 different species.

- List of Plant species:

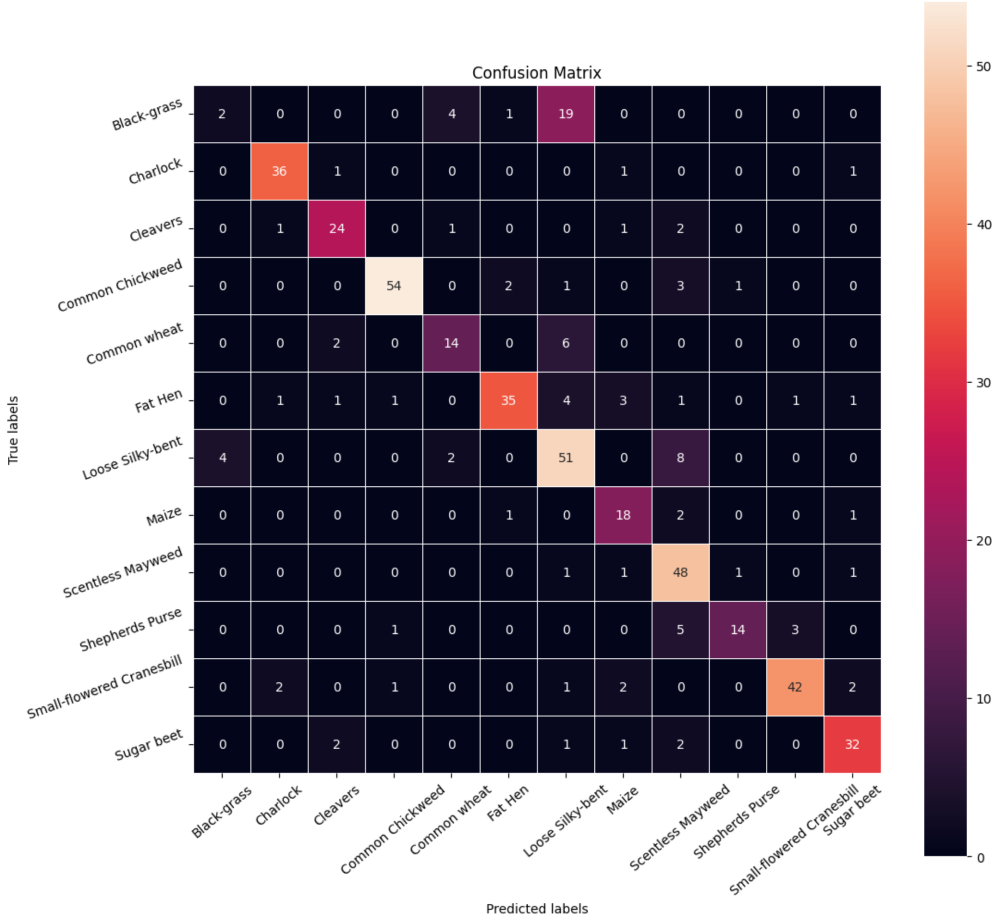
- Black-grass
- Charlock
- Cleavers
- Common Chickweed
- Common Wheat
- Fat Hen
- Loose Silky-bent
- Maize
- Scentless Mayweed
- Shepherds Purse
- Small-flowered Cranesbill
- Sugar beet

Model 1 Confusion Matrix



- The matrix shows that some classes were more difficult to correctly identify than other classes.

Model 2 Confusion Matrix



- Model 2 shows improvement in correctly identifying classes.
- The variance in size and shape seen in the random images generated during EDA makes it understandable that any model may struggle with some identification.

Great Learning

Happy Learning !