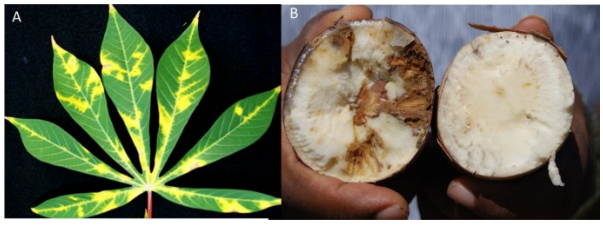
Unsupervised Machine Learning Course Final Project

Predicting Cassava Leaf Disease from Images

Data Summary/Description

The dataset is a set of about 21,000 800x600 images of cassava leaves taken by farmers. Each image shows a cassava plant that is either healthy or has one of four different diseases. For a sample of what each leaf class looks like:

Class	Example Image
Healthy [0] Normal leaf shape and dark green color	
Cassava Bacterial Blight [1] Normal leaf shape and yellow and brown color	

Cassava Brown Streak Disease [2] Normal leaf shape and streaks of yellow	
Cassava Green Mottle [3] Leaves are wrinkled at the edges with normal green color	
Cassava Mosaic Disease [4] Leaves are wrinkled at the edges and have yellow	

The goal is to for every image, predict which disease it has or if it's healthy. Each leaf class corresponds to 5 integer labels 0-4 (each is labeled in the above table).

Data Preparation

Each image was resized down to 160x120, and normalized (so brightness of the images is consistent). The dataset contains about 21,000 images,

however due to memory limitations I only used 1500 images for training/testing. I selected the data to have 300 images for each class so the subset I used would be balanced. I did an 80/20 split on the set so 1200 images would be used for training and 300 images would be used for testing.

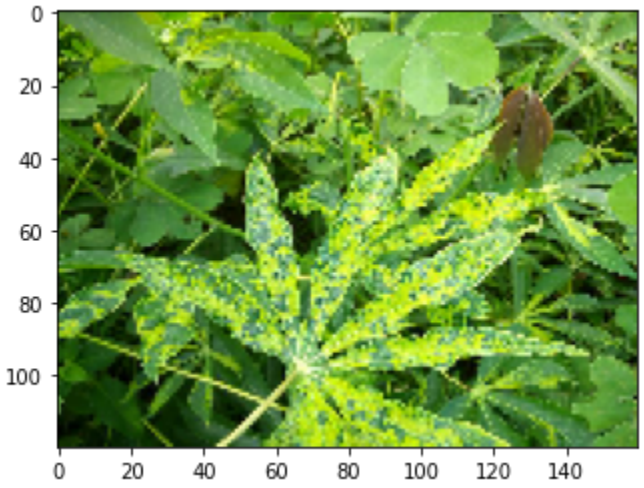
Feature Reduction/Clustering

I used KMeans Clustering to reduce the number of colors in the images. The number of clusters I tried using was 10, 5,and 3.

Basic Output

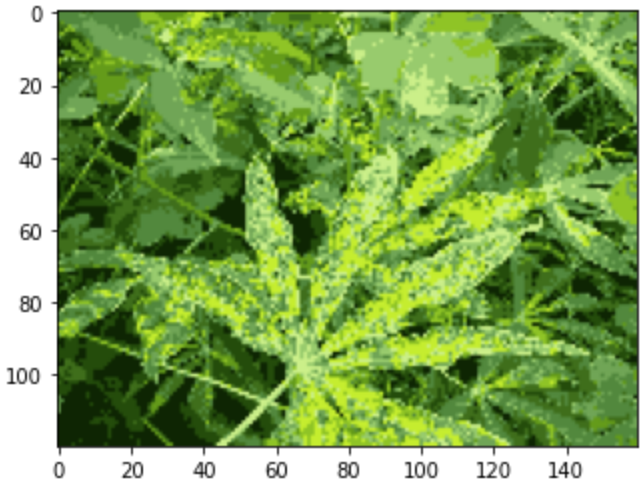
I trained a Convolutional Neural Network to predict the image labels.

Original Sample Image (unclustered):



K=10:

Sample image:



Output from CNN model:

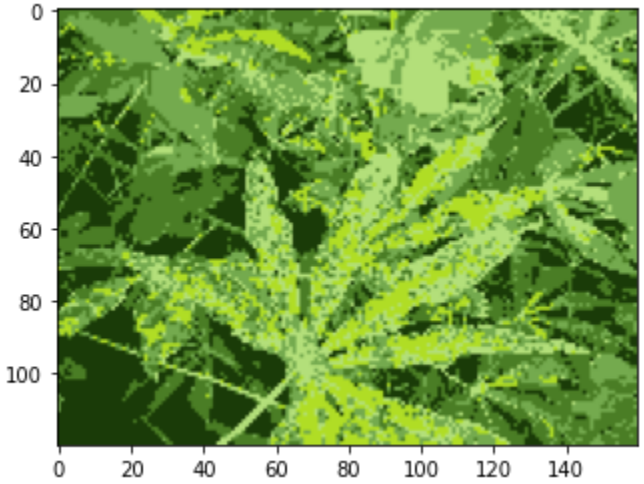
```
Epoch 1/5
38/38 [==============================] - 32s 821ms/step - loss: 0.6867 - accuracy: 0.7933 - val_loss: 1.9314 - val
_accuracy: 0.3600
Epoch 2/5
38/38 [==============================] - 32s 853ms/step - loss: 0.3165 - accuracy: 0.9017 - val_loss: 2.2936 - val
_accuracy: 0.3100
Epoch 3/5
38/38 [==============================] - 33s 864ms/step - loss: 0.2071 - accuracy: 0.9383 - val_loss: 3.1457 - val
_accuracy: 0.3267
Epoch 4/5
38/38 [==============================] - 33s 870ms/step - loss: 0.1278 - accuracy: 0.9642 - val_loss: 3.1276 - val
_accuracy: 0.2800
Epoch 5/5
38/38 [==============================] - 33s 874ms/step - loss: 0.1298 - accuracy: 0.9625 - val_loss: 3.2973 - val
_accuracy: 0.3500
```

	precision	recall	f1-score	support
0	0.51	0.53	0.52	60
1	0.32	0.20	0.24	60
2	0.30	0.47	0.37	60
3	0.41	0.25	0.31	60
4	0.26	0.30	0.28	60
accuracy			0.35	300
macro avg	0.36	0.35	0.34	300
weighted avg	0.36	0.35	0.34	300

	accuracy	precision	recall	f1
0	0.35	0.35	0.35	0.35

K=5:

Sample Image:



Output from CNN model:

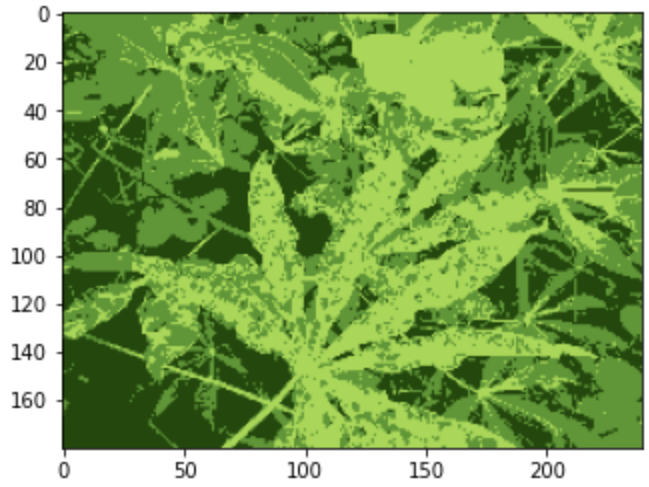
```
Epoch 1/5
38/38 [==============================] - 33s 866ms/step - loss: 0.7458 - accuracy: 0.7317 - val_loss: 2.0127 - val
_accuracy: 0.3067
Epoch 2/5
38/38 [==============================] - 34s 885ms/step - loss: 0.4479 - accuracy: 0.8400 - val_loss: 2.3156 - val
_accuracy: 0.2833
Epoch 3/5
38/38 [==============================] - 34s 908ms/step - loss: 0.2945 - accuracy: 0.9033 - val_loss: 3.1302 - val
_accuracy: 0.2433
Epoch 4/5
38/38 [==============================] - 34s 898ms/step - loss: 0.2240 - accuracy: 0.9283 - val_loss: 3.0687 - val
_accuracy: 0.2933
Epoch 5/5
38/38 [==============================] - 34s 904ms/step - loss: 0.1475 - accuracy: 0.9533 - val_loss: 3.4208 - val
_accuracy: 0.2933
```

	precision	recall	f1-score	support
0	0.35	0.38	0.37	60
1	0.31	0.27	0.29	60
2	0.31	0.23	0.27	60
3	0.28	0.43	0.34	60
4	0.20	0.15	0.17	60
accuracy			0.29	300
macro avg	0.29	0.29	0.29	300
weighted avg	0.29	0.29	0.29	300

	accuracy	precision	recall	f1
0	0.293333	0.293333	0.293333	0.293333

K=3, image size changed to 240X180

Sample Image:



Output from CNN model:

```
Epoch 1/5
38/38 [==============================] - 65s 2s/step - loss: 0.5497 - accuracy: 0.8967 - val_loss: 2.6611 - val_ac
curacy: 0.2967
Epoch 2/5
38/38 [==============================] - 67s 2s/step - loss: 0.1318 - accuracy: 0.9592 - val_loss: 6.2667 - val_ac
curacy: 0.2667
Epoch 3/5
38/38 [==============================] - 69s 2s/step - loss: 0.1676 - accuracy: 0.9558 - val_loss: 2.5568 - val_ac
curacy: 0.2900
Epoch 4/5
38/38 [==============================] - 70s 2s/step - loss: 0.0381 - accuracy: 0.9900 - val_loss: 3.3200 - val_ac
curacy: 0.2667
Epoch 5/5
38/38 [==============================] - 70s 2s/step - loss: 0.1592 - accuracy: 0.9742 - val_loss: 4.3109 - val_ac
curacy: 0.2767
```

	precision	recall	f1-score	support
0	0.45	0.17	0.24	60
1	0.37	0.28	0.32	60
2	0.23	0.78	0.36	60
3	0.28	0.08	0.13	60
4	0.36	0.07	0.11	60
accuracy			0.28	300
macro avg	0.34	0.28	0.23	300
weighted avg	0.34	0.28	0.23	300

	accuracy	precision	recall	f1
0	0.276667	0.276667	0.276667	0.276667

Evaluation
The CNN model performed similarly on all three variations of the clustering I tried (with around 30% accuracy for its predictions). However, the most commonly predicted label did vary with the clustering variations.

For clustering with K=10, it had a higher F1 score on healthy leaves (0.52) which means it had an easier time distinguishing healthy leaves from diseased leaves. This is likely because at K=10, more yellow and brown (not in the background) is retained in the image and in the clustering methods of K=5, and K=3 there is only dark and light green. Dark and light green can appear on healthy leaves, but yellow and brown generally do not, so having those extra colors may have allowed this model to distinguish healthy leaves better. It is important to note that one of the diseases (Cassava Green Mottle) doesn't have yellow or brown coloring on the leaves (only the leaf shape is different), so these examples were likely confused with the healthy leaves, lowering the accuracy.

For clustering with K=5, it mostly predicted every class equally, so the features here were reduced too much. It lost too much color information to be able to distinguish any leaves from each other well.

For clustering with K=3, I increased the resolution to see if the model could learn leaf shape better. Two of the diseases (Green Mottle and Mosaic Disease) have wrinkly leaves and I thought increasing the resolution might help the model learn that better (the lower resolution images make the edges of the leaves more unclear). Unfortunately increasing the resolution drastically increased the number of features and caused the model to overfit much faster, which lowered the accuracy on the test data (though all models were overfit). This model most commonly predicted Cassava Brown Streak Disease (normal leaves with yellow coloring). The only colors in these images were brown (background), dark green, and light green. It would have been unable to distinguish between the yellow on a diseased leaf (because the yellow would have been clustered as light green), and naturally occurring light green or brighter lighting. Likely every leaf appeared to it to have the characteristics (that it could see) of the Cassava

Brown Streak Disease, so that was the most common prediction. It also performed very poorly predicting the diseases which relied on leaf shape (the mottle/mosaic diseases), so increasing the resolution did not improve its recognition of leaf shape.

Conclusions

Overall, I believe that clustering with K=10 performed the best, as the images retained enough color that the model could at least do a decent job of predicting healthy leaves from diseased leaves. Overall each clustering variation only had an overall accuracy of around 30%. This could likely be improved by using the whole dataset (as I only used a very small portion due to memory issues and long runtimes), adding more layers into the CNN, and adding more regulation to the model to prevent overfitting. Cropping the images to only show one leaf might also be beneficial, so there is less background noise in the image.