Soyabean Leaf Disease Classification

Name: Shivam Pawar
Id: 5492083

1 Introduction:

Plants play a crucial role in people's daily life in the form of vegetables, fruits and various consumer goods. So, identifying plant leaf diseases is very important for farmers in enhancing agricultural productivity. But, due to a wide variety of diseases, leaf disease identification by the human eye is time-consuming, difficult and also less accurate. Therefore, there is a need for Automatic Leaf Disease Identification techniques which helps farmers to identify various diseases with less effort, less time and more accuracy. These techniques also help in healthy monitoring of fields that ensure quality agricultural products.

Given a leaf, identify the disease that leaf has, among various diseases with high accuracy using image processing techniques.

2 Method:

- 1. Predict the disease for a leaf that is not yet identified.
- 2. We want to find probabilities of leaf having various diseases so that we can choose high probability as predicted disease (Softmax).
- 3. Minimize the difference between predicted and actual disease (Categorical Cross Entropy).
- Constraints:
    1. No strict latency concerns.

3 Experiments:

3.1 Dataset:

The dataset contains leaf images of soyabean plant. Data is collected from Digipathos Repository.
Data source: https://www.digipathos-rep.cnptia.embrapa.br/jspui/ (https://www.digipathos-rep.cnptia.embrapa.br/jspui/)

3.2 Evaluation metrics:

- Accuracy - It is the ratio of no. of true predictions to the total no. of predictions.
- Confusion Matrix - Matrix(table) that helps in visualizing the performance of model.

3.3 Results:

We have experimented two different cnn model architectures on original, cropped and segmented datasets.
The results obtained from the above models are provided in the below table.

Original leaf images have more background pixels than the actual leaf pixels with non-uniform background.
Since only actual leaf pixels are important, we can crop images or perform some image segmentation techniques.
So, we will experiment with 3 different classification models :

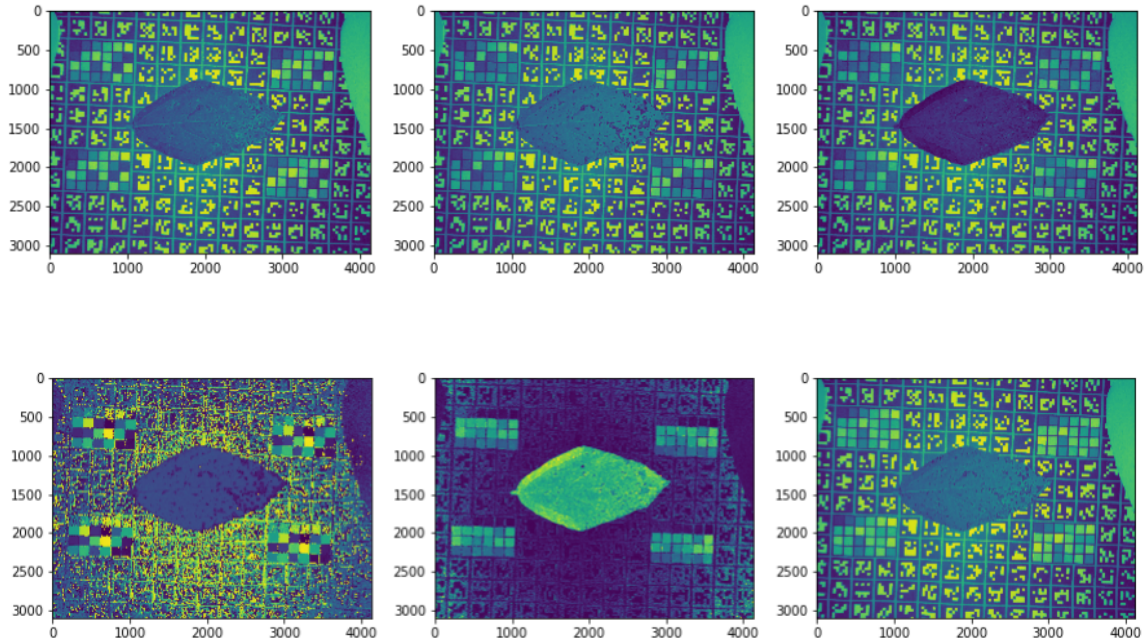
- 1. Classification Model on Original Leaf Images
- 2. Classification Model on Cropped Leaf Images
- 3. Classification Model on Segmented Leaf Images

```
# Please compare all your models using Prettytable library
# http://zetcode.com/python/prettytable/
from prettytable import PrettyTable

x = PrettyTable()
x.field_names = ["Dataset Type", "CNN Model", "Train Accuracy", "Test Accuracy" ]
x.add_row(["Original Images", "Model 1 ", 97.3015, 84.4827])
x.add_row(["Original Images", "Model 2", 96.5079, 77.5862 ])
x.add_row(["Cropped Images", "Model 1", 49.2063, 65.5172 ])
x.add_row(["Cropped Images", "Model 2", 98.7301, 81.0344])
x.add_row(["Segmented Images", "Model 1", 18.2539, 24.1379 ])
x.add_row(["Segmented Images", "Model 2", 96.1905, 82.7586 ])

print(x)
```

```
+------------------+-----------+----------------+---------------+
|   Dataset Type   | CNN Model | Train Accuracy | Test Accuracy |
+------------------+-----------+----------------+---------------+
| Original Images  |  Model 1  |    97.3015     |    84.4827    |
| Original Images  |  Model 2  |    96.5079     |    77.5862    |
|  Cropped Images  |  Model 1  |    49.2063     |    65.5172    |
|  Cropped Images  |  Model 2  |    98.7301     |    81.0344    |
| Segmented Images |  Model 1  |    18.2539     |    24.1379    |
| Segmented Images |  Model 2  |    96.1905     |    82.7586    |
+------------------+-----------+----------------+---------------+
```

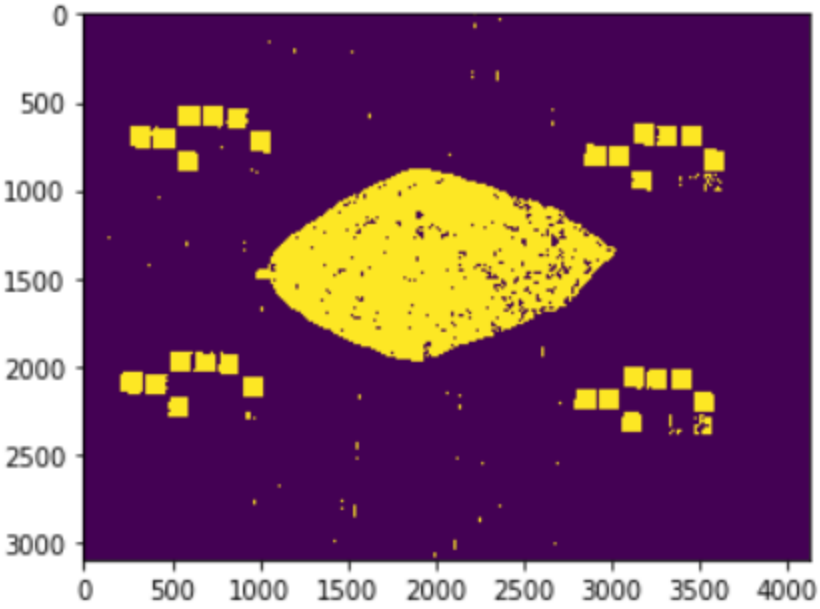
3.4 Analysis and discussions:

a) Finding the best color space channel for color segmentation.

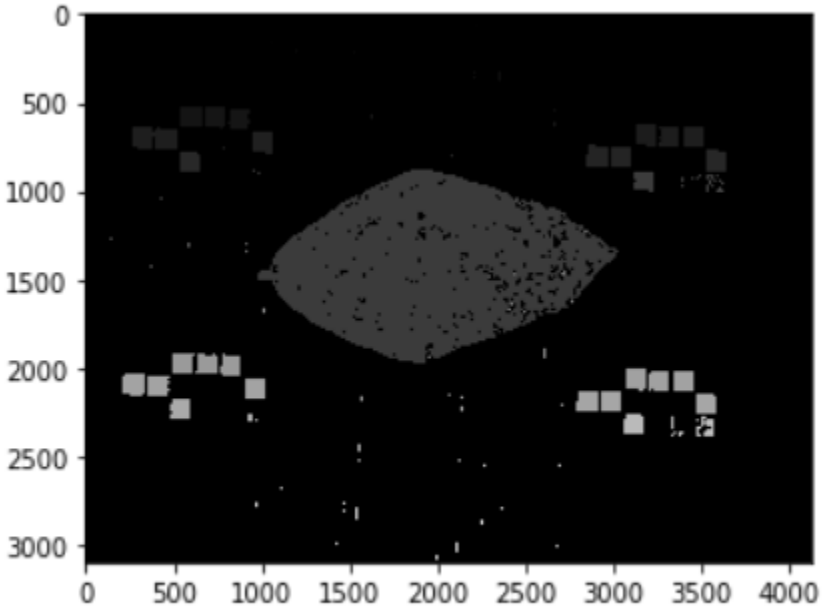


From above, we can conclude the 'b' channel from the LAB color model is useful for color segmentation.

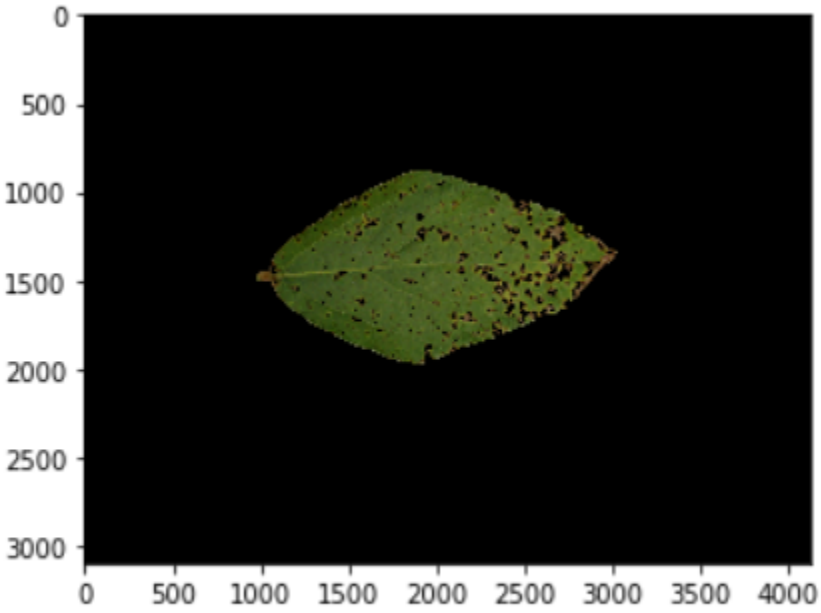
(b) K-means clustering on b channel of Lab color space

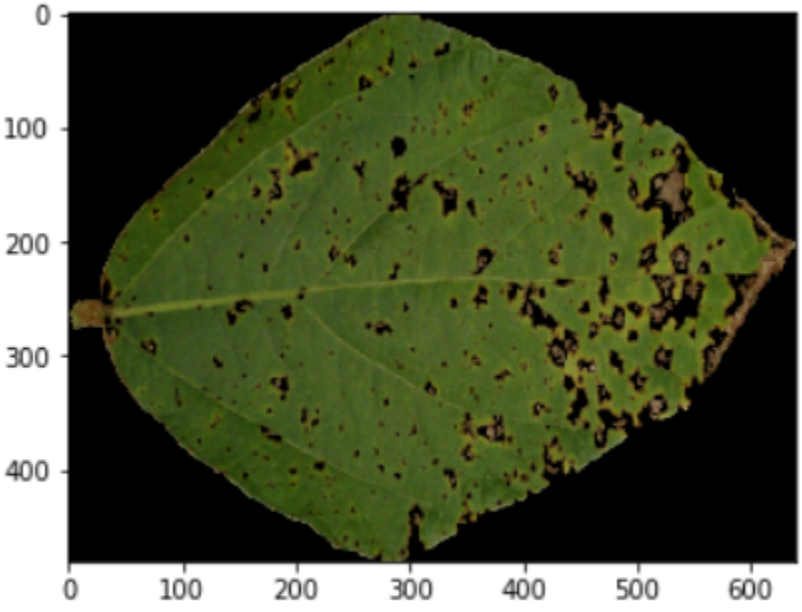


(c) Finding connected components in 2-clustered pixel labels



(d) Finding the main cluster to which actual leaf belongs





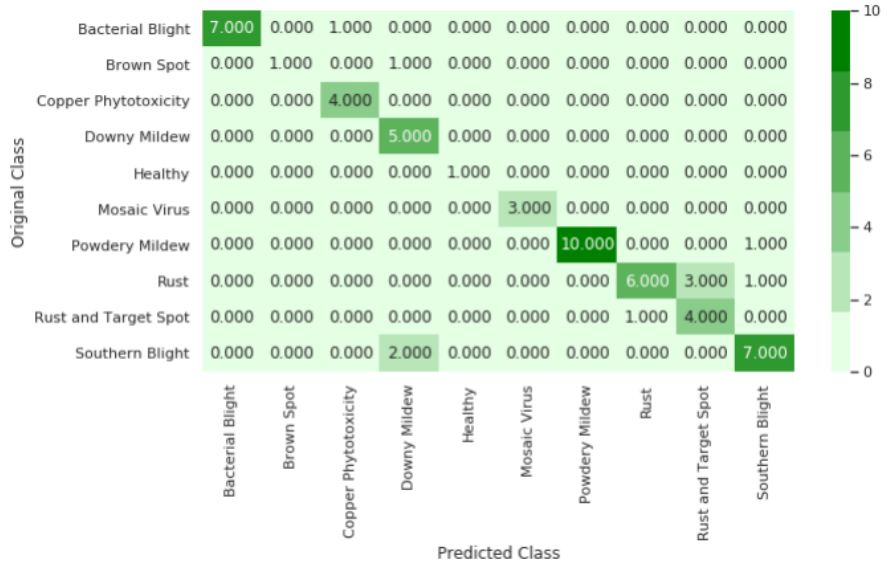

0.8
0.6
0.4
0.2
Train Accuracy
Test Accuracy
0
50
100
150
200
250


Train Loss
Test Loss
10
8
6
4
2
0
0
50
100
150
200
250

Number of misclassified points 17.24137931034483

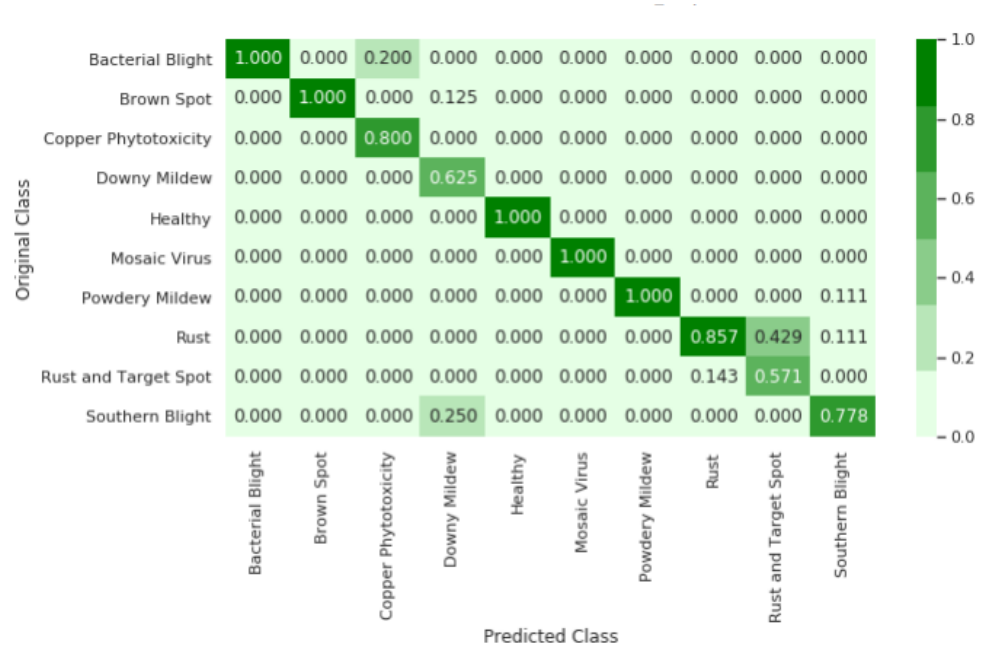
---------- Confusion matrix ----------

-



---------- Precision matrix ----------

-



Sum of columns in precision matrix [1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]

---------- Recall matrix ----------


```
Sum of rows in precision matrix [1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
```

4. Conclusions:

- 1. In this case study, two different CNN architectures are built on original, cropped and segmented soyabean leaf images.
- 2. Since, dataset is too small, data augmentation is performed.
- 3. Due to a highly imbalanced dataset, upsampling techniques are used.
- 4. Because of small and highly imbalanced datasets, the performance of models on some datasets is poor.
- 5. Both model1 and model2 gave good results for original images.
- 6. From confusion, precision and recall matrices, it is observed that 'Rust' and 'Rust and Target Spot' classes are misclassified because images of these classes are mostly similar.
- 7. And also it is observed that some classes have high variance in colors for images belonging to the same class. Due to this, achieving best results is difficult.
- 8. Although, dataset is very small and highly imbalanced, better results are achieved in following cases: Model 1 - Original Images Model 2 - Segmented and Cropped Images

- 9. Because of the dataset structure, the best model cannot be justified. But, among all models, model 1 on original images is the best.
- 10. CNN Models on original images have given better performance than cropped images and segmented images.
- 11. If the dataset is large, the performance of the models might further improve.
- 12. It seems that image segmentation techniques might also give better results for larger results.

5. Contribution:

- **Data Preprocessing**
- **Data Cleaning**
- **Work on CNN models**