YCbCr Leaf Segmentation with Deep Learning Classification

Eduardo Antônio de Lucena Lisboa

The problem

A common problem in Brazilian crops is diseases and pests. All kinds of plantations can suffer from such issues and preventing them is a highly sought subject.

One such plant is the Brazilian Arabica Coffee, which is known worldwide and one of Brazil’s most important agricultural commodity, can be affected by diseases and pests and this can have enormous economical implications.

Knowing which part of the plant has a disease or a pest can help to identify where to focus the treatment.

What is the YCbCr color space?

Developed in the 1950s to be used in analog colored television whilst maintaining backwards compatibility with black and white televisions. Today it is used to transmit, store and process digital video.

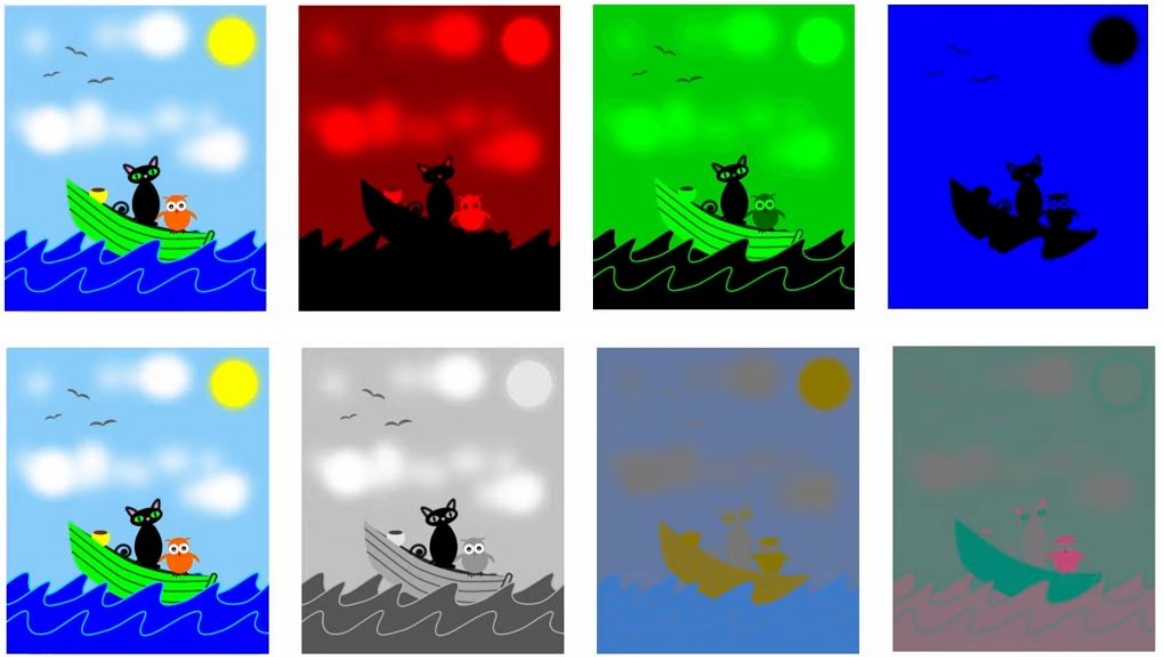
- **Y** is the Luma component, which is the brightness of the color.
- **Cb** is the chrominance blue component
- **Cr** is the chrominance red component

Why use YCbCr instead of RGB?

The human eye is more sensitive to differences in brightness than it is to differences in color, therefore, as the brightness of the YCbCr images is encoded separately, the color values can be changed independently of the brightness information.

If we can reduce the color detail of an image without compromising the brightness, we can create a new image that looks just as good to the human eye. This is the base of JPEG image compression.

RGB x YCbCr



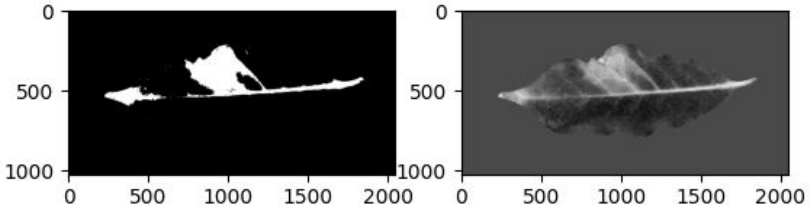
YCbCr and K-Means

For the segmentation, combining YCbCr color spaced images with the K-Means clustering method has been shown to yield great results.

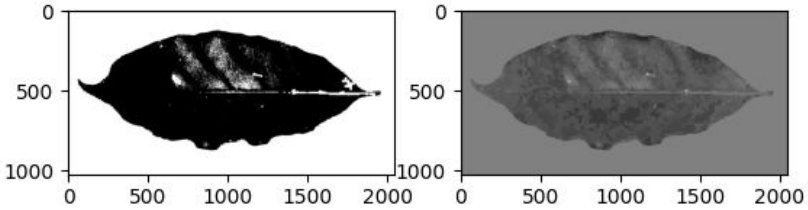
First an input image is converted into the color space, then the Cb and Cr values are separated and put through the K-Means algorithm and finally make the segmentation by applying a morphological transformation on the generated binary image.

Improvement proposal

The original paper has a good segmentation, but it has a big flaw. It doesn’t differentiate between a healthy and a diseased leaf, yielding somewhat confusing results.



Diseased leaf



Healthy leaf

Improvement proposal

To improve this process, I created a Convolutional Neural Network to, before an image is processed, predict if that image is of a healthy or diseased leaf and only segmenting it if this prediction finds it to be a diseased leaf.

The training of the model was made using the Brazilian Arabica Coffee (BRACOL) public domain dataset, which consists of labeled images of healthy and diseased leaves, these diseases being *Phoma*, *Cercospora*, *Miner* and *Rust*, as well as their severity, but this model only predicts if the leaf is healthy or not.

Model training

The models were trained using 80% of the total amount of images in the dataset, being this percentage further divided in 80% for training and 20% for validation.

The hyperparameters adjusted between model trainings were *batch_size* and *epochs*, with a fixed *learning rate* of 0,001.

Data augmentation was performed in the training and validation datasets to prevent overfitting and enhance the amount of images, because the BRACOL dataset has only 1747 images and due to a corruption of the ZIP file, only 1216 images could be used.

Model results

All the trained models were compared using Accuracy, Precision, Recall and F1 scores, then the mean accuracy and standard deviation were calculated using these results.

The model with the best results had these scores:

- Accuracy: 0.9718 (97,18%)
- Precision: 0.9866 (98,66%)
- Recall: 0.9800 (98,00%)
- F1: 0.9833 (98,33%)

Which gives a mean accuracy of 98,04% with a standard deviation of 0,0064.

Combined results

The model was then combined with the original segmentation code. Images considered healthy were ignored and a message is displayed in the terminal and only the images of considered diseased leaves were processed and the resulting segmentation saved in a PNG file.

```
Image 1 is a healthy leaf!

Image 10 is a diseased leaf!
Image saved in "Plots" folder.

Image 2 is a diseased leaf!
Image saved in "Plots" folder.
```

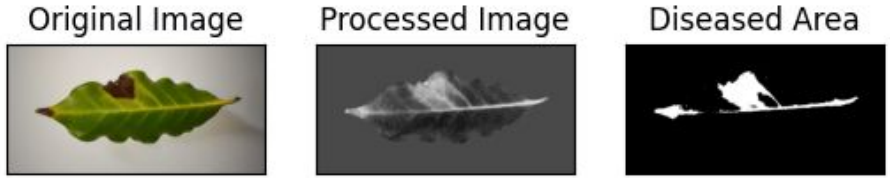


Image 2

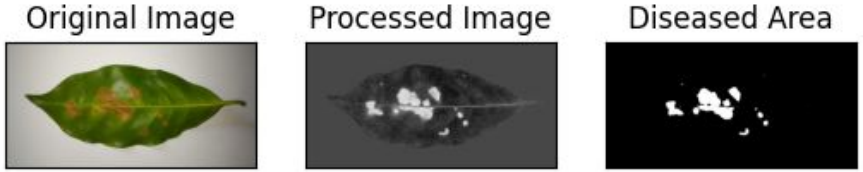


Image 10

Thank you!