Segmentation of PlantVillage Leaf Images

Author : Boris Conforty

The Images

PlantVillage has gathered a collection of more than 54'000 images over time. Pictures were shot by different people, using different cameras with different automatic adjustments, under varying lightning conditions. Some are already segmented. Here are a few examples:

Click one of the images above to change the image used as an example in the whole page.

Original Corrected Final result

The Dream

Technology, particularly computer vision, now makes it possible to "teach" a computer to recognise features in images. If one has enough images with known features, an "educated machine" can achieve a level of performance in recognition that competes with that of trained humans.

One of our dreams at PlantVillage was that our images be used to train an algorithm which, in turn, could be distributed over the world to diagnose plant diseases with a smartphone.

The Challenge

So-called deep learning with neural networks is the tool of choice to train a computer to recognise features. The idea is to have the neural network ingest an image, and you tell it what it has just seen. If you do that thousands or millions of times, it, little by little, "learns". Image-based diagnosis is mainly about visual features (as opposed to context, for instance), so our dream could turn real.

One big concern about teaching a machine is that you should make sure it learns what you want and not something else. If you want to teach a computer to tell between apples and bananas, and all your images of apples are black and white, whereas your images of bananas are in color, it might simply reach the obvious conclusion that all black and white pictures in the world are pictures of apples.

Of course, at PlantVillage, our images are all in color. But, amongst all images, one could easily spot features that are specific to one image set or another. For instance, most of our images of healthy apple leaves could be blueish, or most of our tomato early blight could be on a light background with strong and sharp shadows. We, as humans, easily spot these features as non-relevant; it's not as obvious for a computer.

So came the idea to process our images so as to remove the background and keep the leaf only. This process is called *segmentation*.

Overview of the method

Our approach was empirical. We took an image and tried to find a way to separate the leaf from the background. Then we adjusted the technique so that it worked with more and more images, until we reached a near-perfect or at least decent result with almost all of them.

The steps are:

- Improve/standardise the image
- Detect the color characteristics of the background
- Reconstruct a better image
- Create a collection of masks based on various characteristics of the image
- Process the masks and combine them in a final mask
- Use the final mask to make the background black

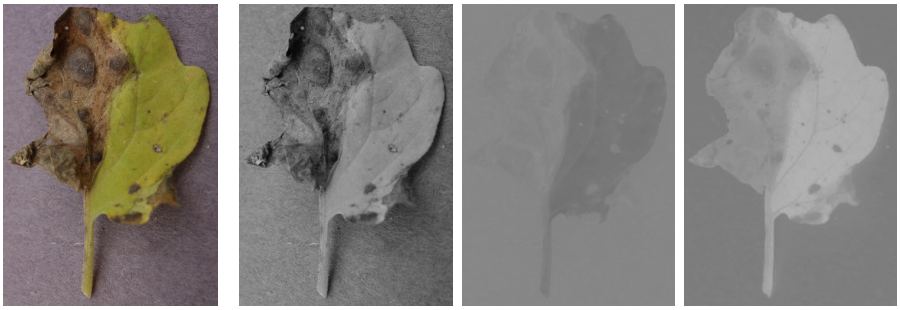
Improve the image

Conversion to Lab color space

Note: R, G and B stand for red, green and blue. L, a and b stand for lightness and a and b are color-opponent dimensions. (For technical details about color spaces, see the dedicated Wikipedia article for color space, RGB color model and Lab color space)

If you take a picture of a uniform background under non-uniform lightning conditions, you'll end up with a gradient in all R, G and B channels (in the RGB color space), whereas only the L channel will be affected (in the Lab color space).
[IMAGES]

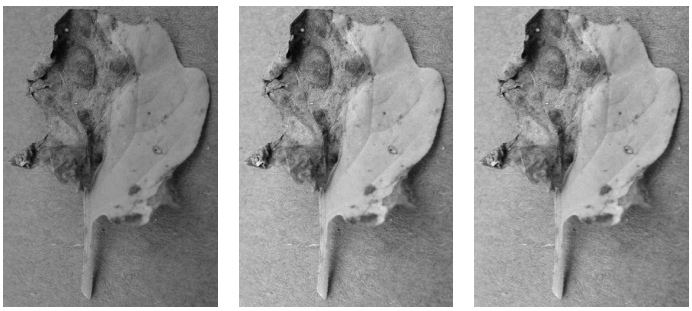
We convert the original RGB image into the Lab color space:



Original L a b

L channel adjustments

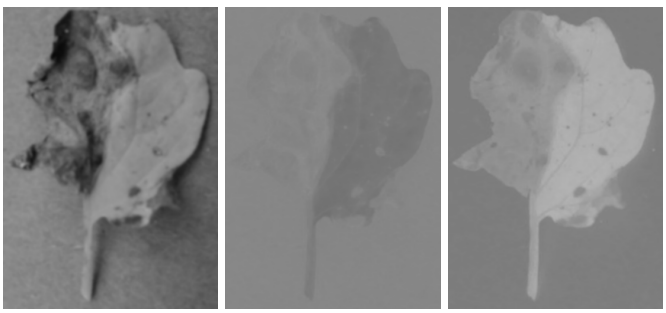
The contrast is automatically adjusted. Then, slight lightness gradients are partly removed with band masking in the frequency space via Fourier transform.



L Contrasted Filtered

Blurring

Channels L, a and b are slightly blurred to decrease noise:

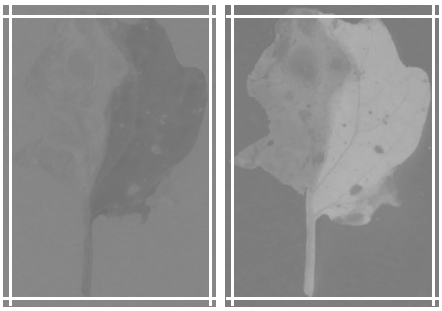


L a b

Detect the color characteristics of the background

Edge cutting

We expect at least one of the borders of the image to show the background. We thus extract the 4 few-pixels-wide edges:



a b

Median color

The median level of each edge image is calculated for the a and b channels.

Selection of the median color

The border median color that is farthest away from green is considered the be the most likely to be that of the background and will be the starting point for the background detection

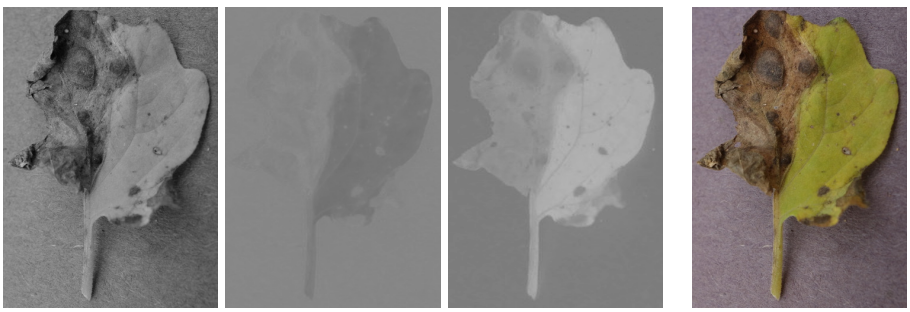
If the color is too close to green, we consider that no background was found and skip the rest of the processing.

Reconstruct a better image

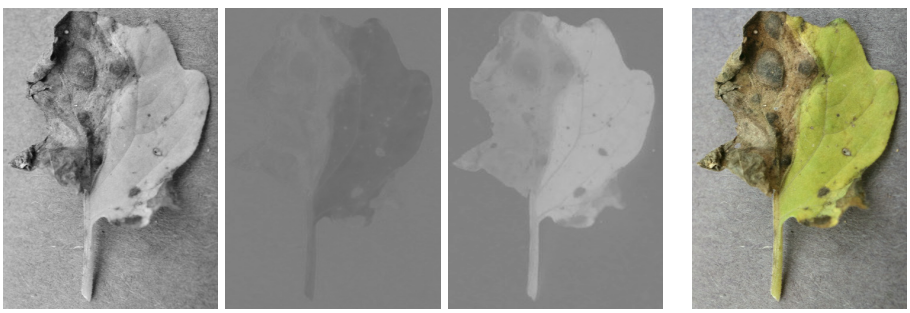
a and b channel correction and reconstruction

Since we know that our pictures were taken over a light-gray background, the values of a and b for all the pixels are shifted so that the median values found in the previous step become that of gray (0.5),

By correcting the a and b channels, we make sure that the background is the right color, and the leaf recovers its natural color.



Original L Original a Original b Original



Corrected L Corrected a Corrected b Improved

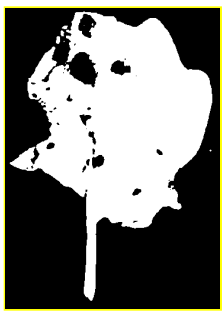
Create a collection of masks based on various characteristics of the image

This part is where most of the empirical tweaking went. A mask is an image with all pixels set to white or black, corresponding to a binary choice (good/bad, keep/don't keep, etc.). After several steps, a set of masks is created, ending up with the following masks:

1. Color mask - Selection of the pixels whose color is close to that of the background
2. Shadow mask - Selection of dark areas that are probably shadows

Color mask

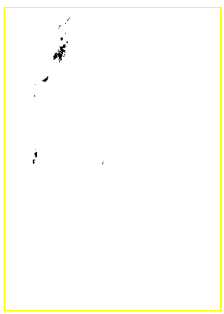
This mask is generated by comparing the value of each pixel, in the a and b channels, to the corresponding of the background. A threshold is applied to keep only pixels that are supposedly not part of the background.



Color mask

Shadow mask

The shadow mask tries to keep only pixels that are not too dark, and with a color not too far from that of the background.

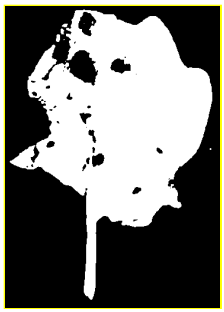


Shadow mask

Final steps

Combine the masks together

The color and shadow masks are combined so that the shadows areas are removed from the color mask.



Intermediate mask

Remove artefacts and detect leaf borders

The mask is processed to remove isolated white or black pixels, through erosion and dilatation. A flood fill is then applied from the border to select the whole background and the mask is slightly blurred.

Intermediate mask

Mask the corrected image

The reconstructed image with color fix is now masked, i.e. the background is turned to black.

Original

Corrected

Final result