The University of Nottingham

UNITED KINGDOM · CHINA · MALAYSIA

Annotation of Image Segmentation on Leaf

Abstract

Image segmentation is one of the most common and crucial tasks in image processing and computer vision. Segmentation is defined as the process of partitioning an image into a set of nonoverlapping regions, we can group the pixels according to local image properties such as intensity of the original images, patterns or textures that are distinctive to each type of region. Many segmentation techniques and algorithms have been implemented and yet, except for those relatively uncomplicated scenes, the problem of segmentation remains unsolved. Many approaches have been taken to tackle the image segmentation problem, and some of the famous segmentation methods such as Clustering, Edge-based and Region-based. In this report, I will be using watershed segmentation, which is an edge-based segmentation technique. The source code available at https://github.com/justin-sem/Image-Segmentation-on-Leaf.

Plant001

Plant002

Plant003

Images shown above are the sample input images for the MATLAB program. Since we are working with green plant, so it is great to compute the greenness from these RGB images.

greenness = greenChannel – (redChannel + blueChannel) / 2.0

where colour channels are extracted individually from the original input images. The sample images after greenness are shown below in Figure 1:

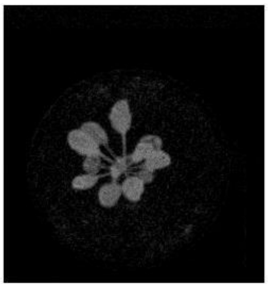

Figure 1(a)

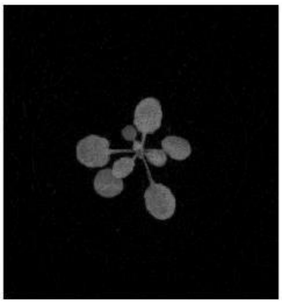

Figure 1(b)

Figure 1(c)

This pre-processing technique is followed by intensity-based methods, which rely on pixel statistics.

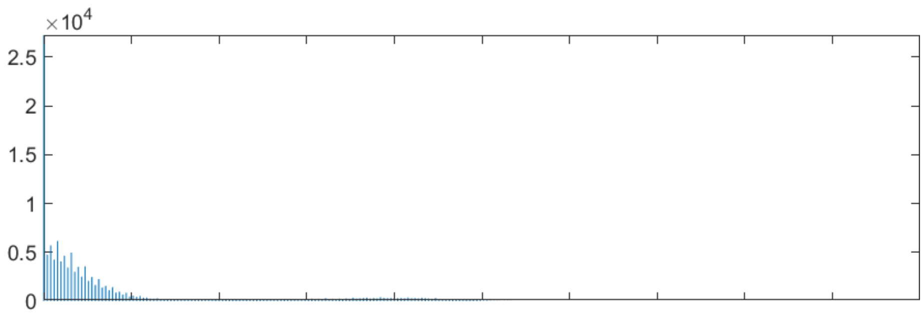

Figure 2(a)

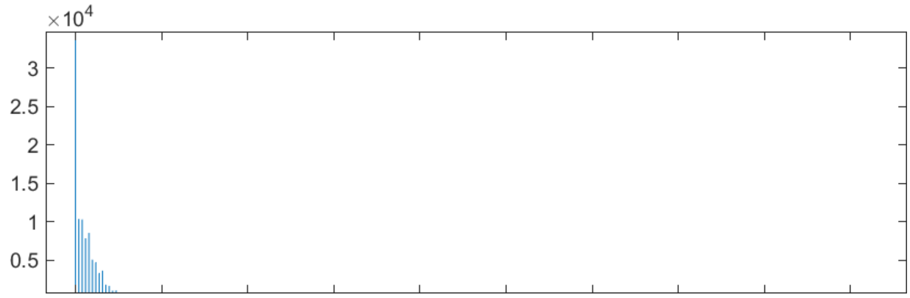

Figure 2(b)

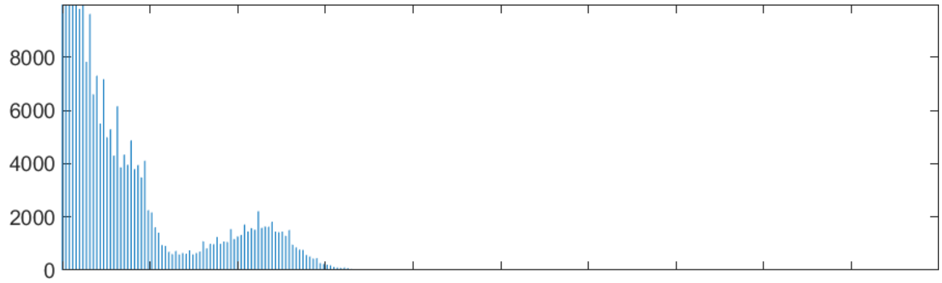

Figure 2(c)

The histogram for the greenness images in Figure 2 displayed each pixel intensity with a reference value. Otsu thresholding is performed here through MATLAB built in function *graythresh*. Therefore, we can thresholding all the greenness images. Mathematically, procedure of thresholding the images $f(x, y)$ and producing a binarizes images $g(x, y)$ can be represent as

$$g(x,y) = \{ \quad \begin{matrix} 1\ if\ f(x,y) > T \\ 0\ otherwise \end{matrix}$$

where T is the threshold value generated Otsu thresholding. However here in this report, only binarize the background and remains foreground object in greyscale.

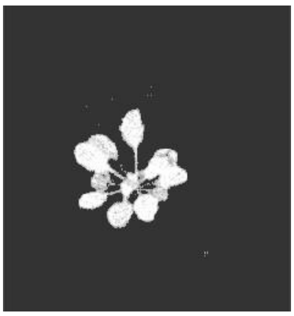

Figure3(a)

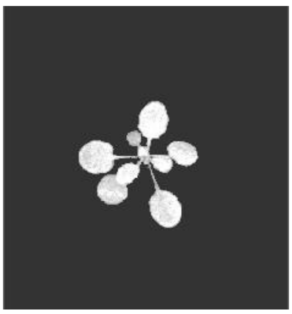

Figure 3(b)

Figure 3(c)

Figure 3 above shown the images after Otsu thresholding by binarize the image background to 0. Images in figure 3 also go through image gained and biased to enhance the contrast and brightness.

Before using watershed segmentation on the leaf, foreground object i.e., leaf is marked. Here, a morphological technique reconstruction-based opening and closing is used to clean and remove small flaw without disturbing the overall shapes of the foreground objects. The regional maxima are calculated and the foreground markers after reconstruction by superimpose on the original image are shown below:

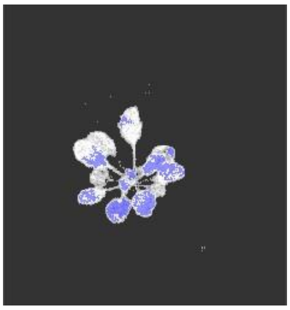

Figure 4(a)

Figure 4(b)

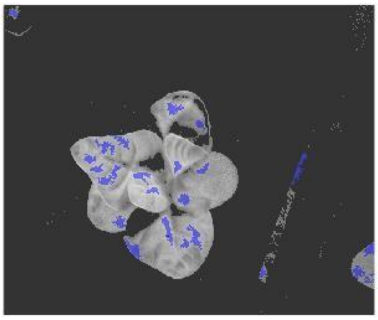

Figure 4(c)

In MATLAB, the IPT function *watershed* implements the watershed transform. This work used function *bwdist* computes the distance from every pixel to the nearest nonzero-valued pixel, which also known as the distance transform. Afterwards, we compute the watershed-based segmentation. The results are shown below by superimpose on the original image:

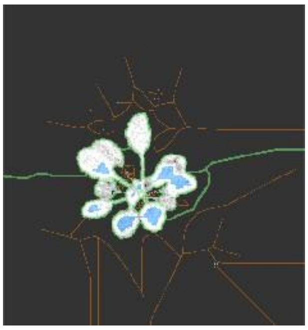

Figure 5(a)

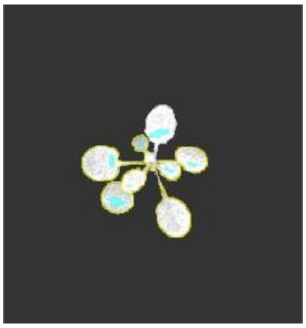

Figure 5(b)

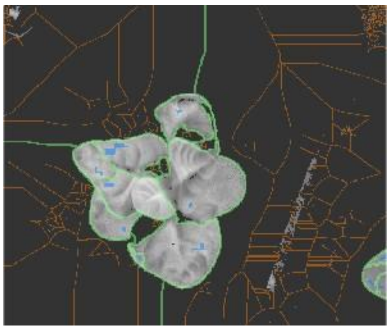

Figure 5(c)

A visualization technique is used here to display the label matrix as a RGB image, by doing so we can visualize the segment of leaf in different colour. Function *label2rgb* is used here with *shuffle* as order parameter which will pseudo-randomly assigns colormap colours to the leaf. The final output results are displayed below:

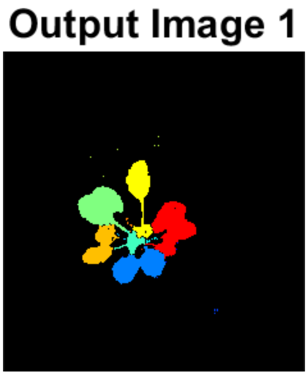

Figure 6(a)

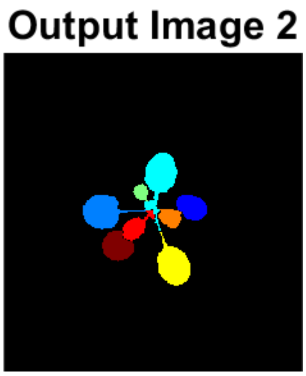

Figure 6(b)

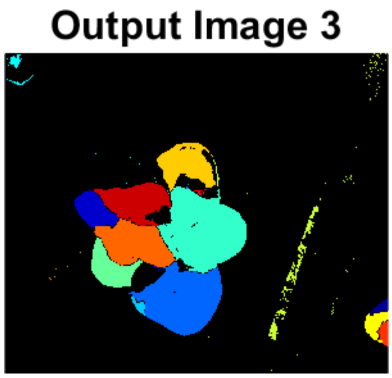

Figure 6(c)

From the output images shown in Figure 6, we can see and conclude that by only using watershed segmentation to segment the leaf is neither enough nor perfect. In output image 2, the segmentation is close to perfect as it is considered as a relatively ‘easy’ scene, however when segmentation comes on nontrivial images is a very complicated problem, such as image 1 and image 3. As there are some complications such as poor contrast between foreground objects and background, overlapping among objects, no uniform lighting and etc. All these issues become the obstacle for the perfect image segmentation.