1.0 Preprocessing techniques

1.1 Image1

Firstly, read in the original image of plant1

Figure 1.1.1

The first pre-processing technique for image 1 is to take the green channel of the image. The reason why green channel is used is because since the plants that we are interested to segment out is in green colour, therefore the value of the pixels of the plants in green channel would be higher compared to other regions and this would make it easier to binarize the image later.

Figure 1.1.2

Next, apply Fast local Laplacian to make the edge more obvious. Fast Local Laplacian Filter is an edge preserving filtering method. The sigma and alpha value for the Laplacian filter is 0.4 and 0.5 respectively.

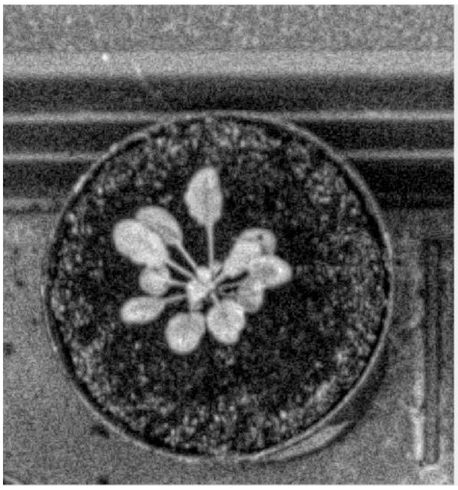


Figure 1.1.3

The Laplacian filtered image is then converted to a binary image using imbinarize

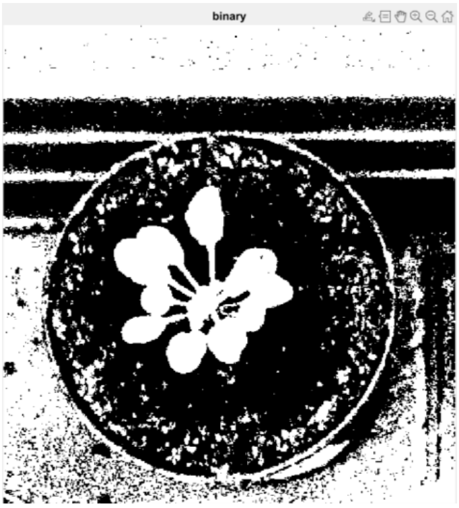


Figure 1.1.4

Use bwareafilt to extract out the region of interest.bwareafilt is a built in function in matlab that extracts obejcts from binary image by size.Figure 1.1.5 shows the 3 largest connected areas while Figure 1.1.6 shows the 2 largest connected areas. Figure 1.1.7 is obtained by subtracting Figure 1.1.6 from Figure 1.1.5.

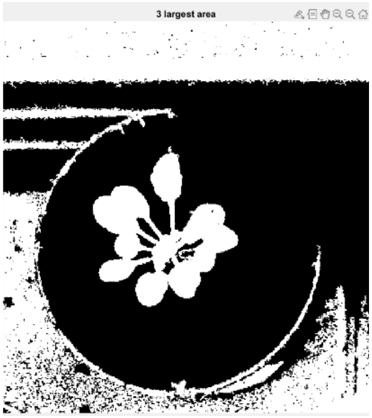


Figure 1.1.5

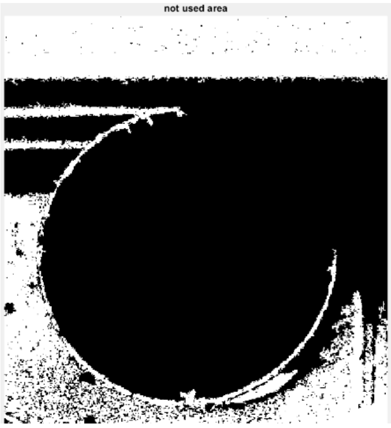


Figure 1.1.6

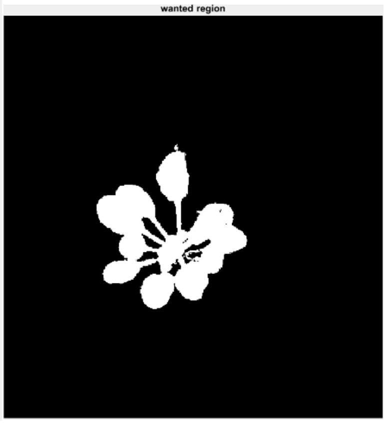


Figure 1.1.7

1.2 Image 2

For the second Image, the first step is to read in the original Image.

Figure 1.2.1

The green channel of the original image is extracted out.

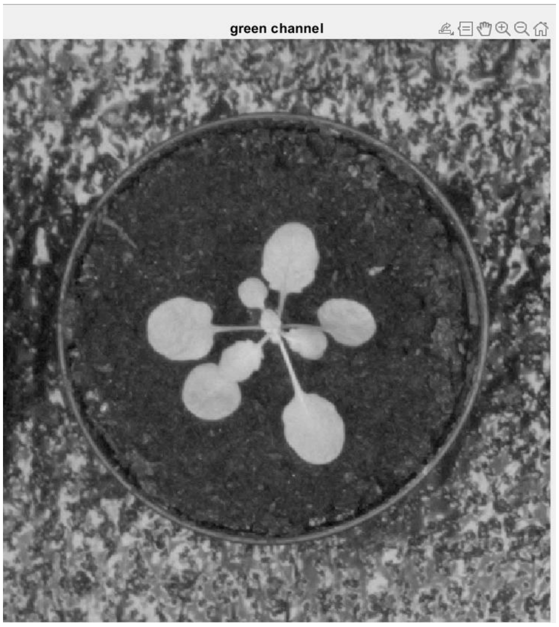


Figure 1.2.2

Filter the image using fast local Laplacian filter. The parameters of the filter (alpha and beta value) is the same as image 1.

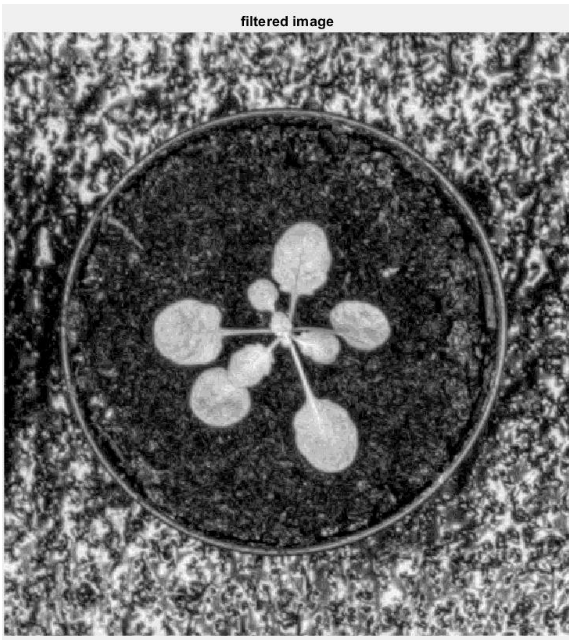


Figure 1.2.3

The filtered image is binarized.

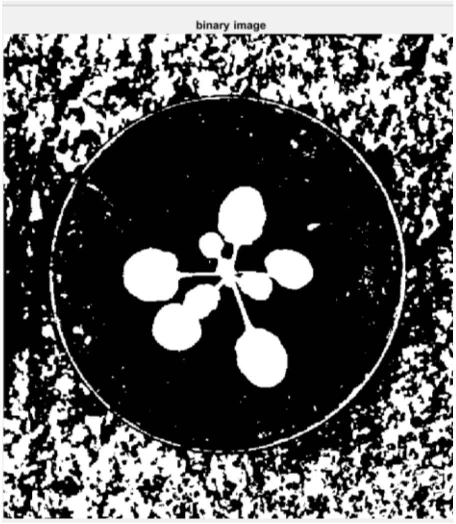
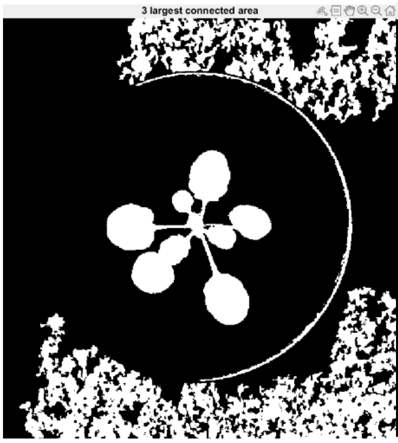
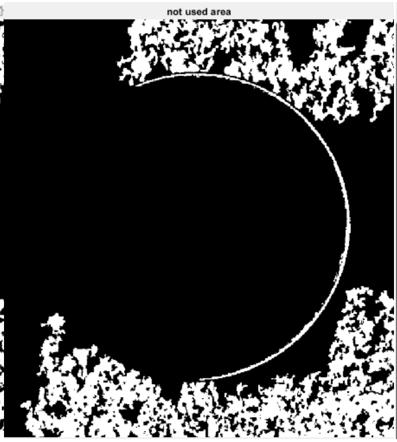


Figure 1.2.4

Use bwareafilt to extract out the region of interest. Figure 1.2.5 shows the 3 largest connected areas while Figure 1.2.6 shows the 2 largest connected areas. Figure 1.2.7 is obtained by subtracting Figure 1.2.6 from Figure 1.2.5





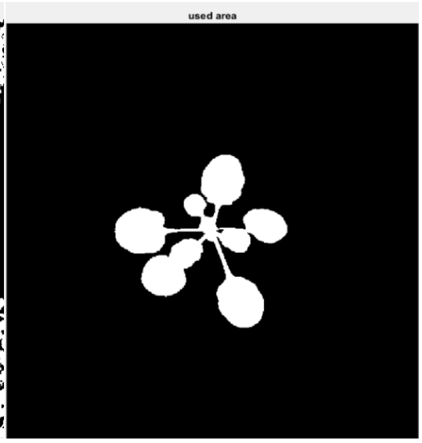


Figure 1.2.5

Figure1.2.6

Figure 1.2.7

1.3 Image 3

Read in original image of plant 3

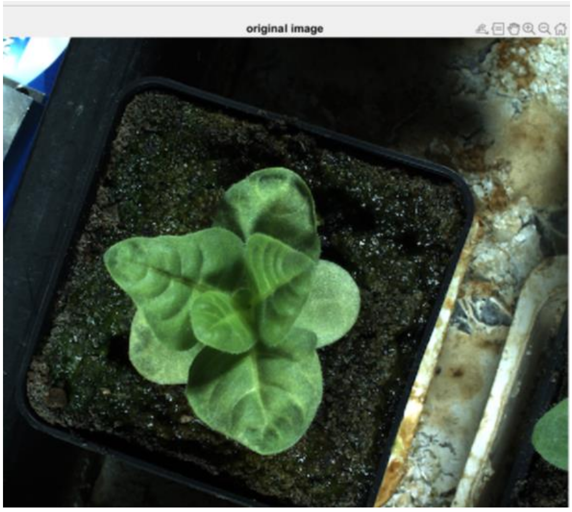


Figure 1.3.1

The green channel is extracted out from the original image

Figure 1.3.3

The image is sharpened using Laplacian filter with the same sigma and alpha value as in image 1 and image 2

Figure 1.3.2

The filtered image is binarized.

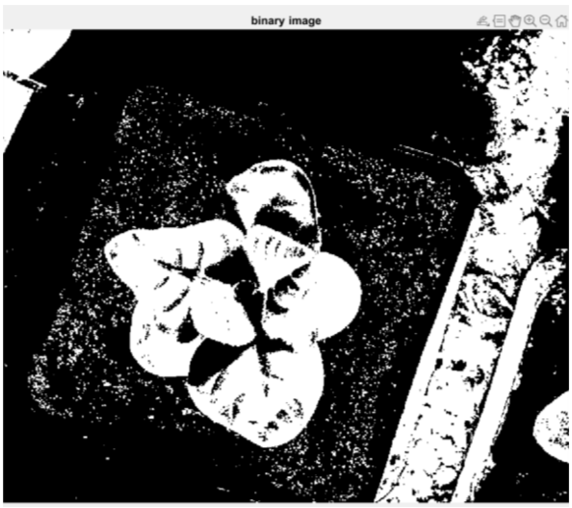


Figure 1.3.4

Use bwareafilt to get the desired area. Since the largest connected area in the image is the plant itself, so the region of the plant is extracted out directly. （Different version of Matlab may yield different result, the version of Matlab I'm using is R2021b）

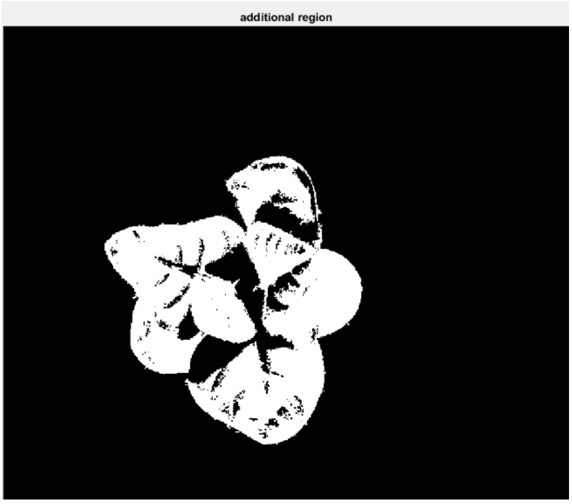


Figure 1.3.5

Dilate the image using structured elements of disk shape and neighbouring,nhood = 5. The purpose of dilation is to fill up the holes especially the textures of the leaves.

Figure 1.3.6

2.0 Watershed Segmentation

Ultimately, I have decided to go with watershed segmentation. Bernhard & Charl (2014) stated that to understand the watershed, one can think of an image as a surface where the bright pixels represent mountaintops and the dark pixels valleys. The surface is punctured in some of the valleys, and then slowly submerged into a water bath. The water will pour in each puncture and start to fill the valleys. However, the water from different punctures is not allowed to mix, and therefore the dams need to be built at the points of first contact. These dams are the boundaries of the water basins, and also the boundaries of image objects. Rosebrock (2015) stated that The watershed algorithm is a classic algorithm used for segmentation and is especially useful when extracting touching or overlapping objects in images, such as the coins in Figure 2.0.1.

Figure 2.0.1

Rosebrock. (2015) also stated that traditional image processing methods such as thresholding and contour detection, we would be unable to extract each individual coin from the image. Since the original image of the plants contain overlapping leaves and some of the leaves are very close together, therefore I have decided to use watershed segmentation.

The steps to perform watershed segmentation is as follow:

Calculate the distance transform of the complement of the binary image containing the outline of the plant. The distance transform of an image is the distance between every pixels in the image to the nearest nonzero. The distance method that I used for calculating the distance transform is Euclidean.

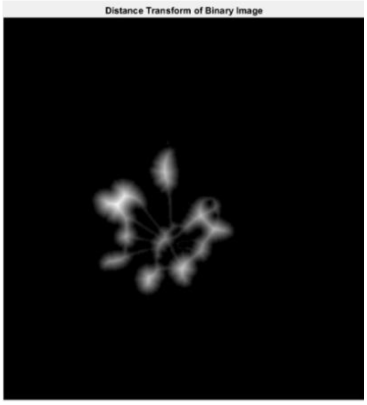


Figure 2.0.2.

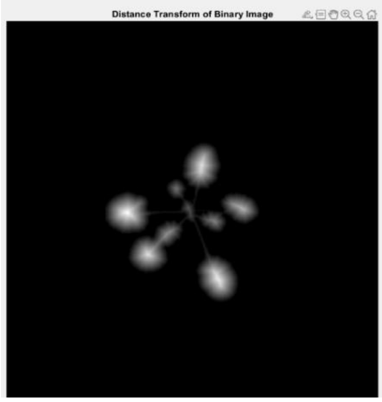


Figure 2.0.3

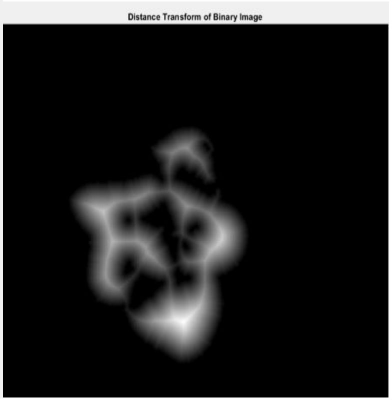


Figure 2.0.4

The complement of the distance transformed image is taken so that light pixels represent high elevations and dark pixels represent low elevations for the watershed transform.

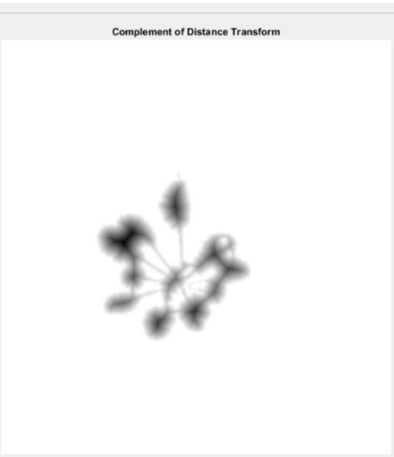


Figure 2.0.5

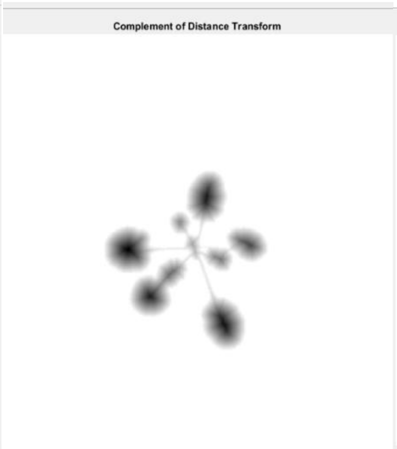


Figure 2.0.6

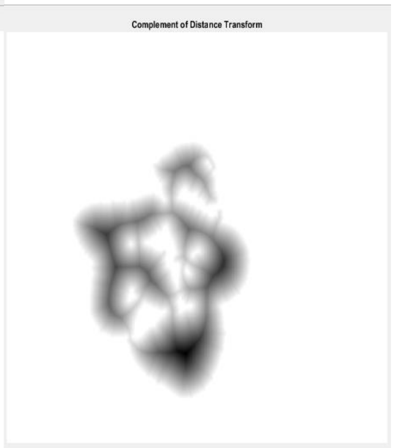


Figure 2.0.7

Calculate the watershed transform. The Matlab function, imhmin is used to suppress minima whose depth is less than h. For plant 1, the h value is 0.5, 1 and 3 for plant 1,plant 2 and plant 3 respectively. Pixels that are outside the ROI(the plant) is set to 0. The image shown below is inverted to provide a better visualization of the result from watershed transform

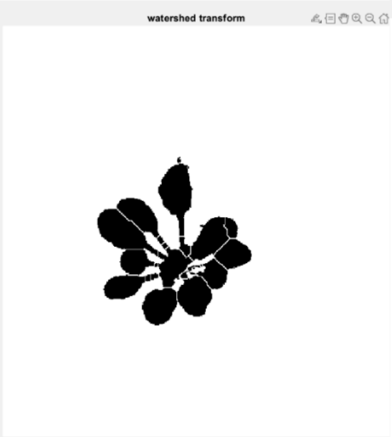


Figure 2.0.8

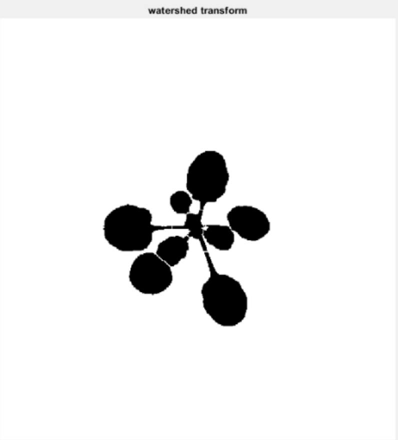


Figure 2.0.9

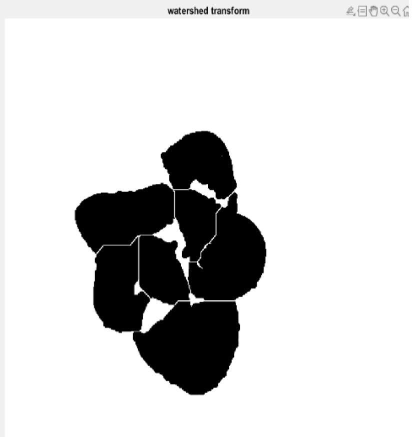


Figure 2.0.10

Display final Image using the Matlab function, [label2rgb](). The colourmap is choosen at random to produce random colour every run. The background is changed to black colour.

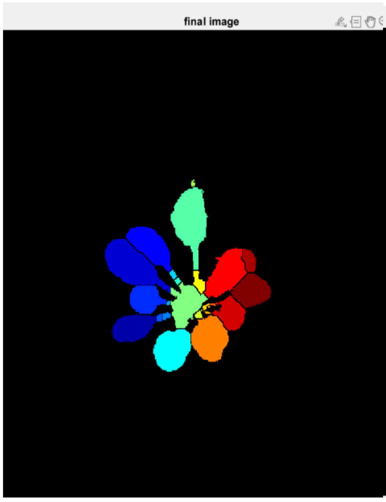


Figure 2.0.11

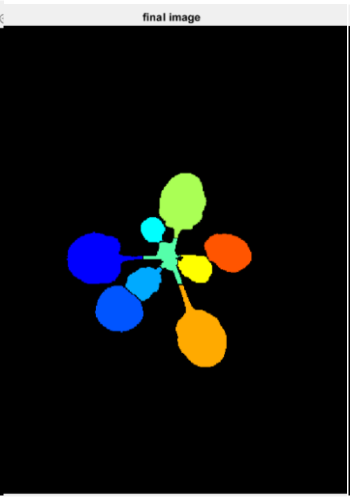


Figure 2.0.12

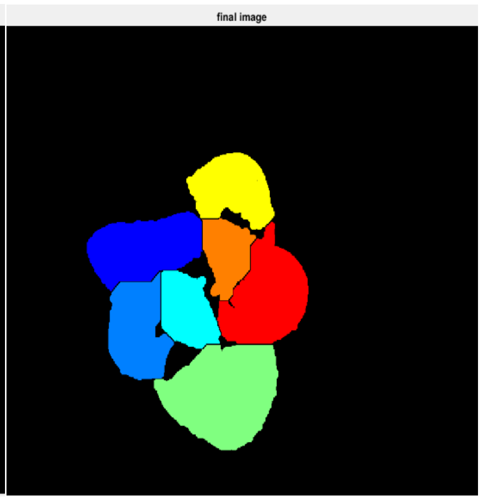


Figure 2.0.13

3.0 Critical analysis

3.1 Fast Local Laplacian Filter

Aubry, Paris, Hasinoff, Kautz and Durand.(2015) stated that Fast Local Laplacian is a faster version of Laplacian filter and can yields result that is 50 times faster on grayscale image. Since Laplacian filter is a 2nd derivative filter, it has stronger response to fine details and easier implementation. The reason why Laplacian filter is used is to enhance the edges of the leaves more as in Figure 1.1.1, Figure 1.2.1 and Figure 1.3.1, there are a number of overlapping leaves. In order to better detect separate leaves, I decided to use a more sensitive filtering method while preserving the edge. The local Laplacian filter is applied with a sigma value of 0.4 and alpha value of 0.5. The sigma value is the amplitude of details to smooth while the alpha value is the amount of smoothing to apply. From the Mathworks website for [local Laplacian filter](), an alpha value of less than 1 increases the detail of the input image and this effectively enhances the local contrast of the image without affecting edges or introducing halos. In order to avoid over-filtering or under-filtering, an optimum value of 0.5 is chosen after trying out with other values e.g. 0.1,0.9,0.6.

The diagram below shows local Laplacian filter with alpha value of 0.9 and 0.1 respectively. The strength of fast local Laplacian filter is that it can filter the image to remove noise while preserving the edges in the image. Besides that, it is also faster than the conventional Laplacian filter. However, since Laplacian filter is sensitive to fine details, it would be vital to choose the optimal sigma and alpha value to ensure the image is not oversegmented in the end.

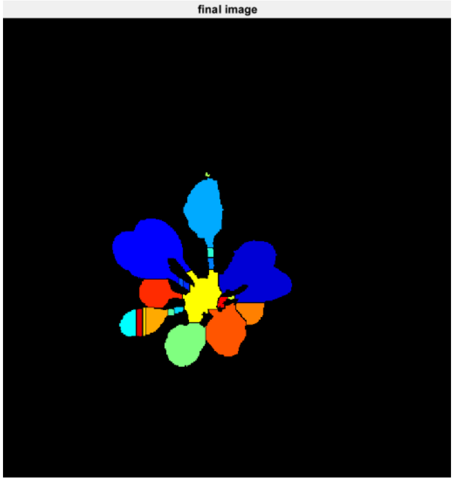


Figure 3.1.1

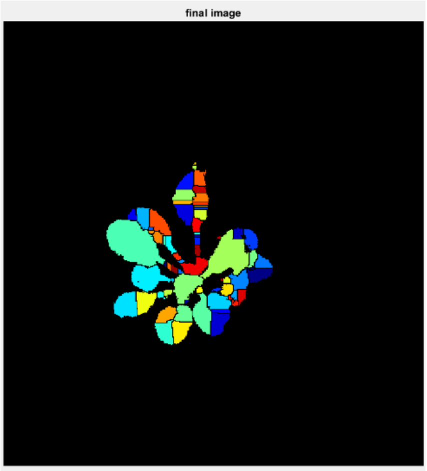


Figure 3.1.2

3.2 BW area filter

The strength of bwareafilt is that it allows the user to extract largest area of connected components and ignore small regions with pixel value. The weakness is that user is required to use trial and error method to get the desired region of interest.

3.3 Dilation

Textures on the leaves in plant 3 (Figure 1.3.1) is being regarded as edges. This will eventually create small patches of black pixels on the leaves which is being regarded as the background when the image is converted into binary image after Laplacian filtered as shown in Figure 1.3.5. Therefore, applying dilation could help fill up the black pixels especially on the leaves which is helpful during the watershed segmentation later. The size of the structuring element is crucial because if the size of structuring element is too big, the area of dilation will be too big and this would significantly decrease the number of leaves segmented during watershed segmentation. Figure 3.3.1 shows the final image using dilation with structuring element of size 6 in 'disk' shape.

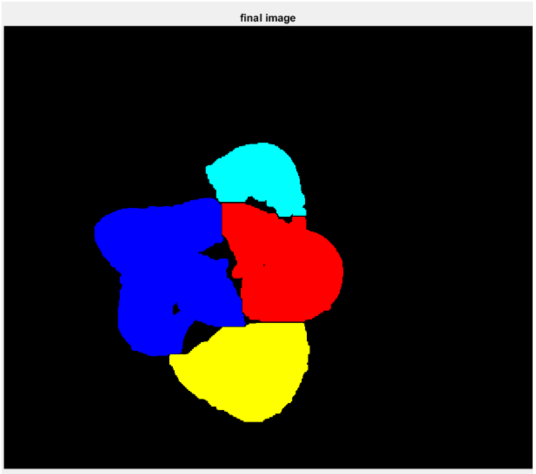
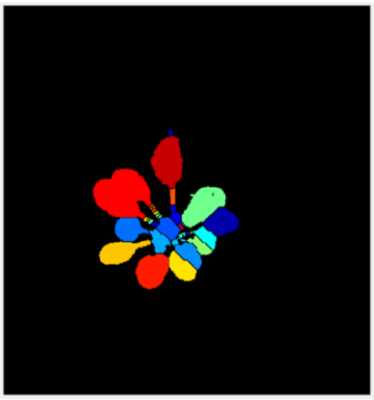
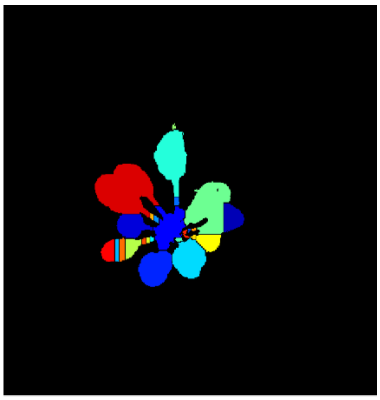


Figure 3.3.1

3.4 Distance Transform

For the distance transform in Figure2.0.2, Figure 2.0.3, and Figure 2.0.4, the distance method that I used is the Euclidean method. This is because the Euclidean distance method works better than other distance method like cityblock, chessboard and quasi Euclidean. Figure 3.4.1, 3.4.2 and 3.4.3 shows the result produced using distance method of chessboard, cityblock and quasi Euclidean respectively using the same preprocessing parameters as the final result.





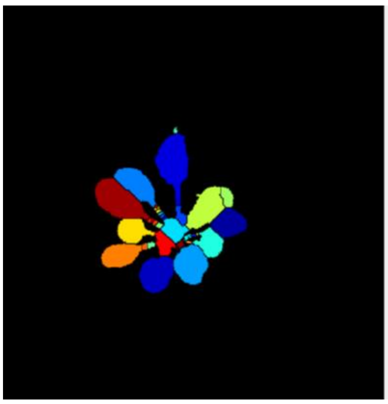


Figure 3.4.1 Figure 3.4.2 Figure 3.4.3

Fisher, Perkins, Walker & Wolfart. (2003) stated that in order for distance transform to produce a good result, it is important that the binary input image is a good representation of the object that we want to process. Fisher et al (2003) also stated that simple thresholding is often not enough and it might be necessary to further process the image before applying the distance transform.Therefore, if the amount of preprocessing is not sufficient, the result from distance transform is not ideal.

3.5 Watershed Segmentation

Amandeep & Aayushi (2014) stated that the result of watershed algorithm is global segmentation, border closure and high accuracy. It can achieve one-pixel wide, connected, closed and exact location of outline.

From the final images shows above (Figure 2.0.11,Figure 2.0.12 and Figure2.0.13), we can see that watershed segmentation is able to locate and separate most of the distinct leaves. This shows that watershed algorithm is high in precision and efficiency. Watershed segmentation is able to separate out regions that are connected together in the binary image of the 3 plants (Figure 1.1.7, Figure 1.2.7, Figure 1.3.6)

Nicket & Rameh. (2013) stated that the main disadvantage of the Watershed Transform is that for most natural images it produces excessive over-segmentation. Ravimal (2014) stated that watershed segmentation suffers from oversegmentation due to the lack of knowledge of the objects being classified. Eddins. (n.d.) stated that oversegmentation occurs because every regional minimum, even if tiny and insignificant, forms its own catchment basin.From Figure 3.5.1 and Figure 3.5.2, we can notice that some of the colours on the branches are disconnected and is regarded as 2 different regions by watershed segmentation.

Oversegmentation can be reduced using the imhmin (h-minima transform) function in matlab which suppresses all minima whose depth is below a specified value. However, if the value is too big, we might not be able to segment out leaves that are overlapping in particular. The imhmin value, is 0.5,1 and 3 respectively for plant 1,2 and 3 respectively. The reason why the imhmin value in plant 1 is lower is because if the value is bigger, some of the edges separating the plants would disappear. Therefore, we can see that there area many disjoint sections in the branches of plant 1.

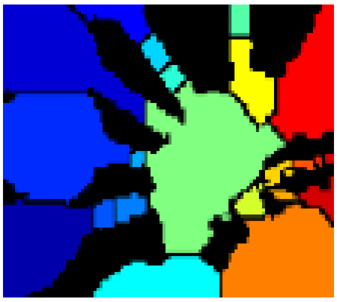


Figure 3.5.1

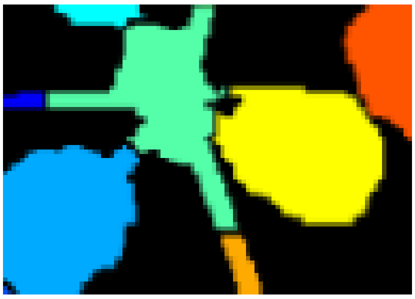


Figure 3.5.2

Although most of the distinct leaves are identified and segmented out, we can see that some of the edges separating the leaves do not reflect the true outline of the leaves in the original image as shown in Figure 3.5.3 and Figure 3.5.4. Figure 3.5.5 and Figure 3.5.6 shows the region of the original image. The situation arrises when the leaves are overlapping or when they are too close together. This is because when binarizing the image, the overlapping regions would become one single connected region in the binarized image. So, when watershed is applied to the binary image, there will be some changes in the edge of the leaves.

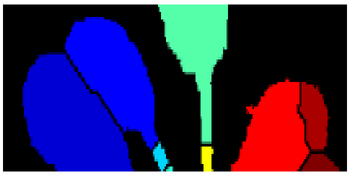


Figure 3.5.3

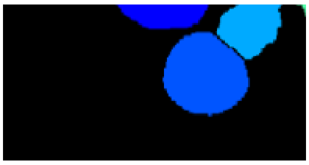


Figure 3.5.4

Figure 3.5.5

Figure 3.5.6

From Figure 2.0.13, we can see that the result in plant 3 is not as good as plant 1 (Figure 2.0.11) and plant 2 (Figure 2.0.12). This is because in image 3, the leaves are very close together and some regions of the leaves are overshadowed by the leaves above it. When the image is converted to binary image (Figure 1.3.6). The region which is overshadowed would be regarded as the background due to its pixel value being below of the threshold. Due to the inability of pre-processing to remove the shadow, watershed segmentation therefore couldn't produce a perfect result for plant 3. Besides that, leaves that are overlapping will become one single connected area with the overlapped leaves when being converted to binary image and therefore the edges of separate leaves being segmented out is not perfect.

4.0 References

1. Niket Amoda & Ramesh K Kulkarni. (2013).Efficient Image Segmentation Using Watershed Transform. Retrieved from http://ijcst.com/vol42/2/niket.pdf

2. Amandeep Kaur & Aayushi. (2014). Image Segmentation using Watershed Transform. Retrieved from https://www.ijsce.org/wpcontent/uploads/papers/v4i1/A2060034114.pdf

3. Bernhard.P. & Charl.B. (2014). *Visual Computing for Medicine, 2nd Edition.* Morgan Kaufmann Publishers. https://www.sciencedirect.com/topics/computer-science/watershed-segmentation

4. Ravimal Bandar. (2014). Image Segmentation using Unsupervised Watershed Algorithm with an Over-segmentation Reduction Technique. Retrieved from https://arxiv.org/ftp/arxiv/papers/1810/1810.03908.pdf

5. Rosebrock, A. (2015). Watershed OpenCV. Retrieved from https://pyimagesearch.com/2015/11/02/watershed-opencv/

6. Eddins, S. (n.d.). The Watershed Transform: Strategies for Image Segmentation. Retrieved from https://www.mathworks.com/company/newsletters/articles/the-watershed-transform-strategies-for-image-segmentation.html

7. Aubry, M., Paris, S., Hasinoff, S, W., Kautz, J., Durand, F., (2014) "Fast Local Laplacian Filters: Theory and Applications." Acm Transactions on Graphics 33 .