University of Nottingham
UK | CHINA | MALAYSIA

Image Processing Pipeline for Nuclei Extraction and Analysis from Confocal Laser Microscopic Images

COMP2032

Name : Ang Jia Hau

Student ID : 20150707

Course : Computer Science with Artificial Intelligence

APPENDIX

1. Introduction

Confocal Laser Scanning Microscopy (CLSM) is an optical imaging technique heavily used in biological sciences which can be used to analyze samples such as the tip of a plant root. Though, quantitative cell analysis can pose to be a tedious task without automation and the accuracy levels vary within the capability of individual. Meanwhile, the efficiency and accuracy within the capability of computing in image processing can be exponential if done properly. Image processing can significantly speed up and potentially produce more accurate quantitative cell analysis results, as well as lift some load off the shoulders of tired researchers hence allowing them to focus more attention on other segments of their work.

This report showcases the pipeline used to extract the quantitative data of a given set of confocal laser microscopic images of plant roots.

The pipeline can be simplified to 6 levels: RGB Processing, Colour Space Conversion, Pre Processing, Thresholding, Binary Image Processing and Analysis Details.

2.1 RGB Processing

To begin the funnel, the goal was to pre-process the RGB image to be as clear as possible before the channel extraction process. As seen in **Figure 2.1.1,** the images are dark at certain regions and therefore some details could be missing.

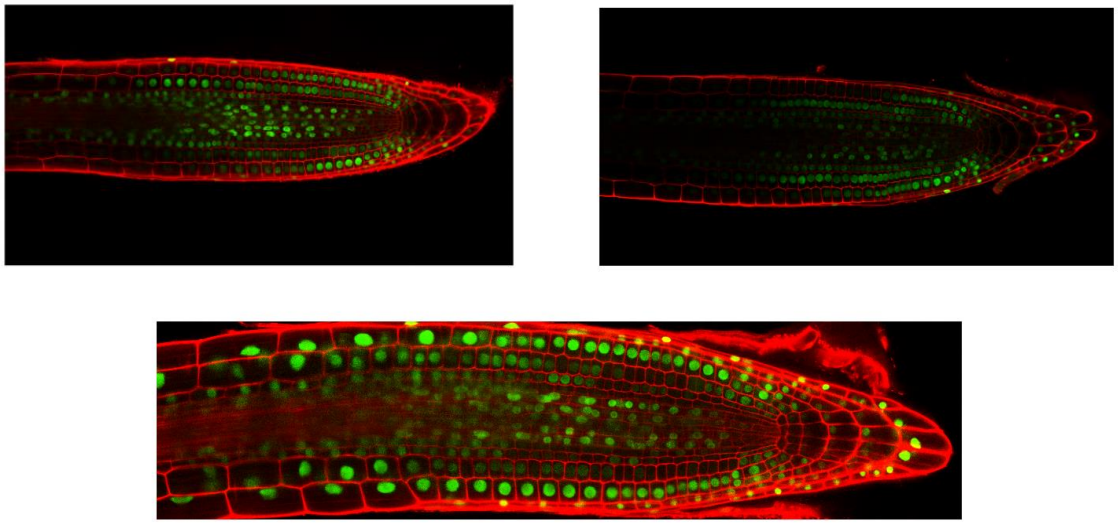


Figure 2.1.1 – Original Image(Not Processed)

Therefore, the method used to improve this segment of the process is to apply Gamma Correction to the linear values based on sRGB standard followed by correcting the linear values based on Adobe RGB(1998) standard which produces the images seen in **Figure 2.1.2**

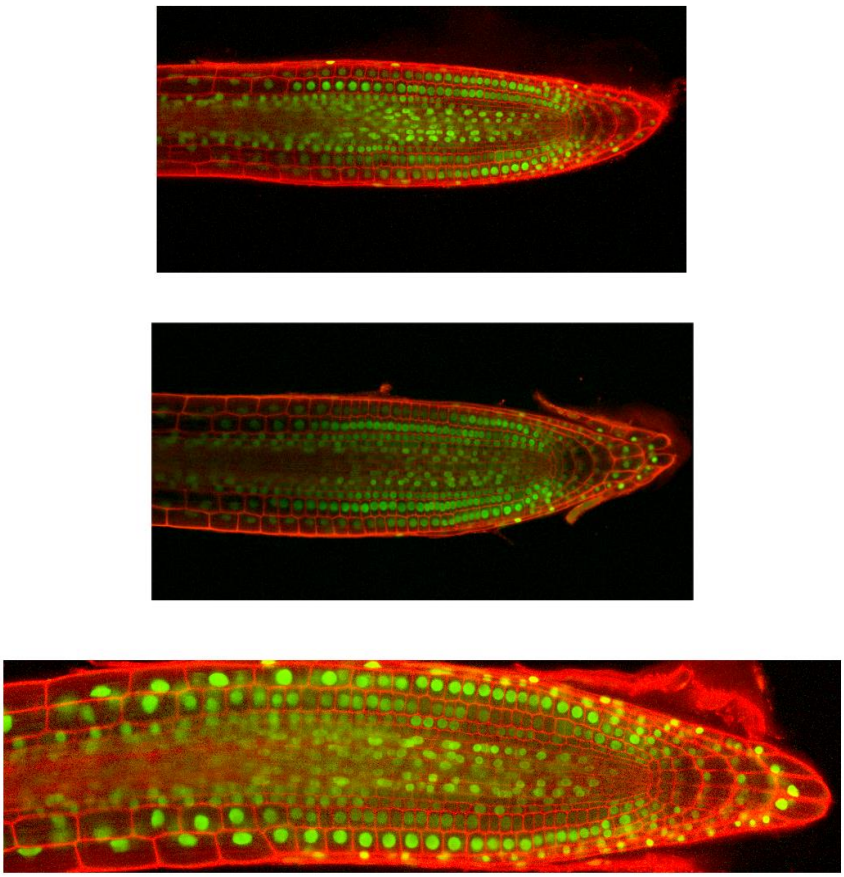


Figure 2.1.2 – Gamma Corrected Images

An alternate solution for this point would be to use the histogram equalization method which will increase the contrast of a given image by equalizing it's values. However, this makes the image too bright, as seen in **Figure 2.1.3(R).** Besides that, another colour option to work with would be Linear RGB Values whereby we could work with the values directly to carry out mathematical operations. However, it causes the images to darken too much and therefore losing its details as seen in **Figure 2.1.3(L).**

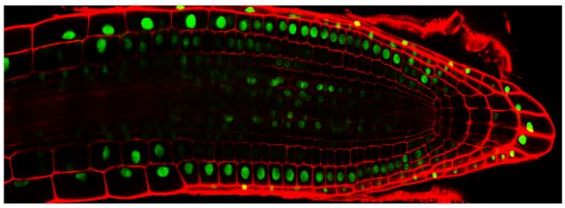


Figure 2.1.3(L) - Linear RGB

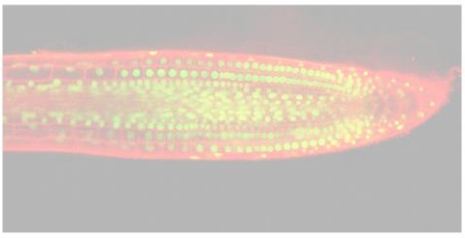


Figure 2.1.3(R) – Histogram Equalisation

2.2 Colour Space Conversion

After applying gamma correction to the image, we convert the colour space to HSV through $a = rgb2hsv(b)$, where a is the output HSV image and b is the input RGB image, followed by extracting the hue channel and increasing the contrast by $X = immultiply(Y, Z)$ function where X is the output image, Y is the input image, and Z is a constant factor of 1.3. This gives the layer of the plant root's nuclei as seen in **Figure 2.2.1.**

HSV is the best option to work with as the red-yellow hue angle falls between 0 to 60 degrees whilst the green hue angle falls between 75 to 165 degrees. The green hue has higher luminance than the red hue, and therefore will stand out more. This fits our criteria, as it serves as the first step in segmenting the nuclei.

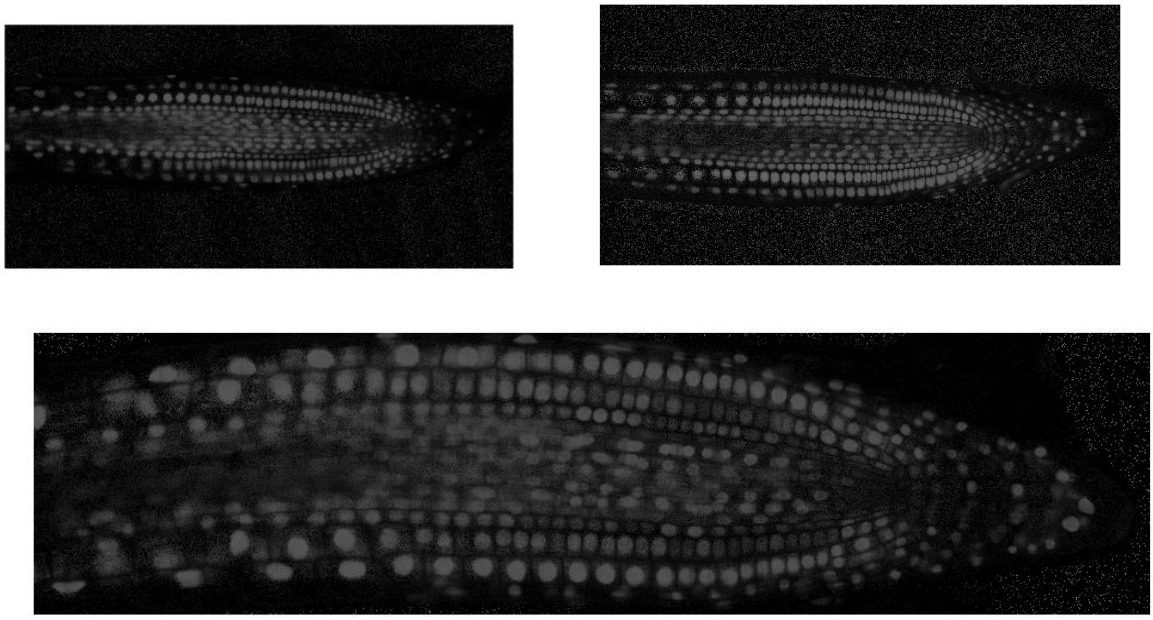


Figure 2.2.1 – Hue Channel

Two other available channels to work with are LAB and RGB. We could extract the green channel from the RGB image, though an overlap with the red values would occur and therefore the segmentation would include the cell wall as seen in **Figure 2.2.2(R).** In LAB, we could extract the green which lies within the a* zone. However, this would be inefficient as it requires more pre-processing steps as seen in **Figure 2.2.2(L).**

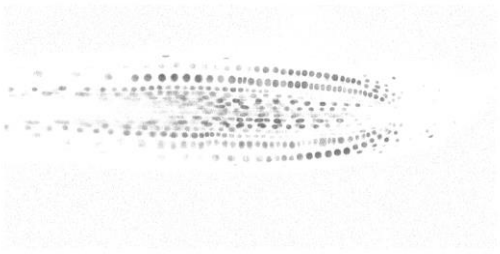


Figure 2.2.2(L) – ClELab Channel

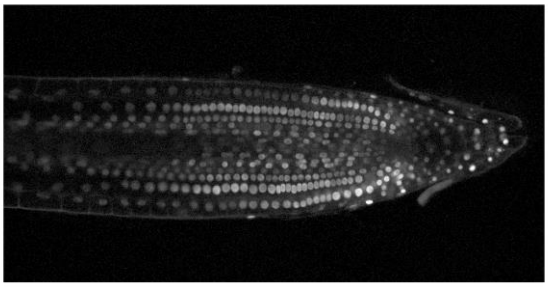


Figure 2.2.2(R) – RGB Green

2.3 Pre-Processing

2.3.1 Noise Reduction

After extracting the hue levels, we can see that there is noise within the previous image **(Figure 2.2.1 Top Right).** Therefore, I have used the Median Filter for noise reduction followed by Adaptive Histogram Equalization to produce the images shown in **Figure 2.3.1.**

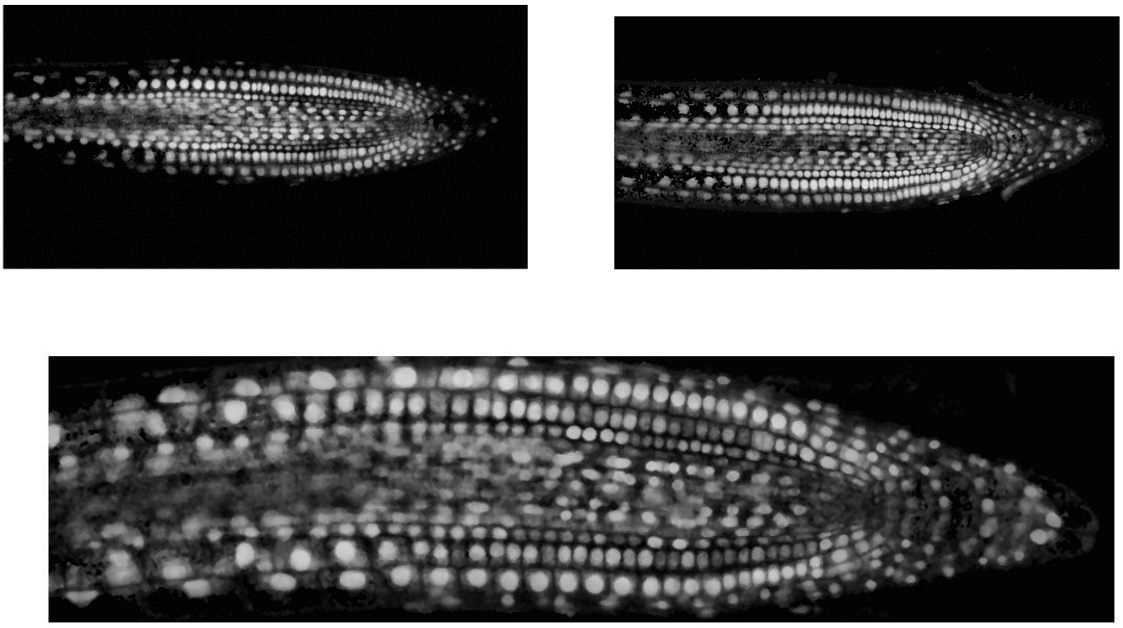
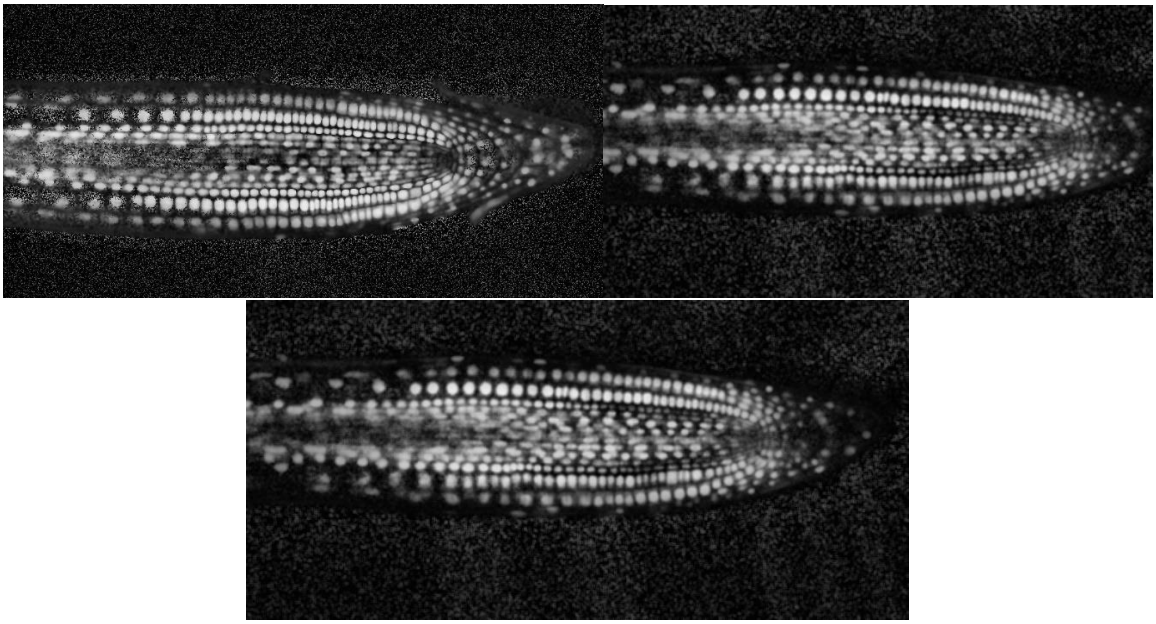


Figure 2.3.1 – Noise Reduction

There are many methods for noise reduction such as Mean Filter,Gaussian Filter, Bilateral Filter, and Anistropic Difussion. However, there are flaws for each selection which makes Median Filter the most suitable candidate for this position at this stage.

This is because Gaussian Filter, Mean Filter and Bilateral Filter all blurs and smoothen the image too much, hence causing major clumps and adding more blur to the image, causing problems identifying individual nuclei. Anistropic Difussion on the other hand, does not show promising results even though it is known to remove noise whilst retaining edges. Below shows results for anistropic diffusion(Top left), Mean Filter(Top right) and Gaussian Filter(Bottom).



2.3.2 Sharpening and Edge Detection

After noise reduction, there is a little bit of blur and the edges are rounded due to the properties brought by Median Filter. Therefore, we need to sharpen the images to get some clear properties of the nuclei. This is done by defining a Laplacian kernel and applying it as a spatial filter to the image, then combining it with the original image, and then smoothing with a Gaussian filter which produces the image shown in **Figure 2.3.2.**

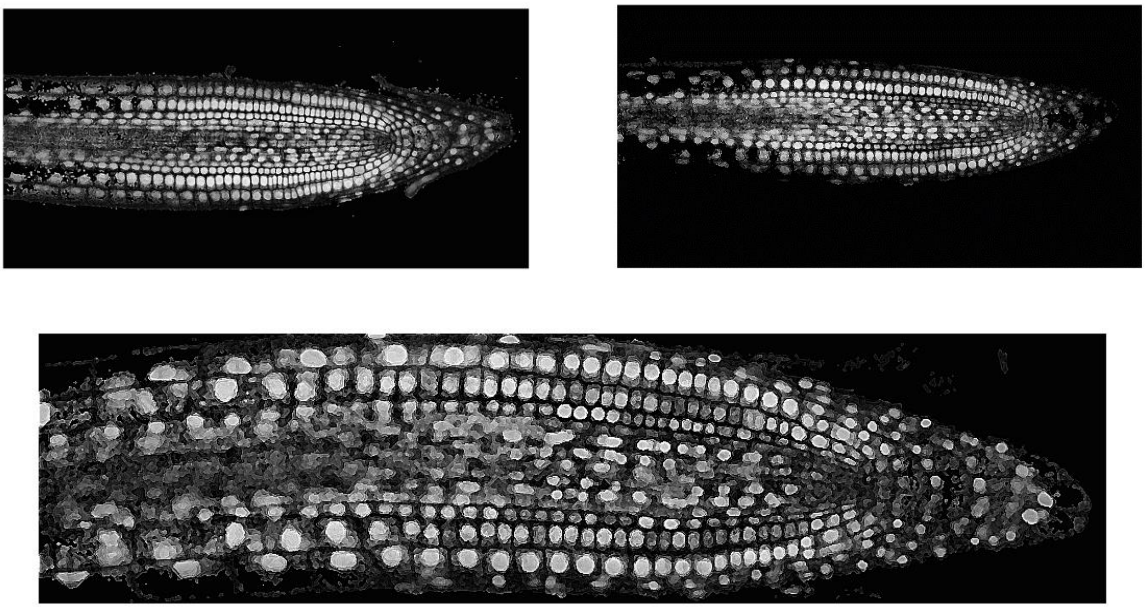


Figure 2.3.2 – Laplacian Sharpening

There are other edge detection options such as Canny and Sobel but they produce results that are over the edge. Figure 2.3.2(A) and Figure 2.3.2(B) shows the product of using Canny and Sobel edge for sharpening respectively. Although the segmentation looks decent, the colour shades poses a problem within the thresholding stage, which resulted in less nuclei being detected overall. Upon comparison, laplacian filter gives the most balanced results which is what we're going for. There are 3 images with varying properties. Therefore, we must go for the solution that is viable for all kinds of images.

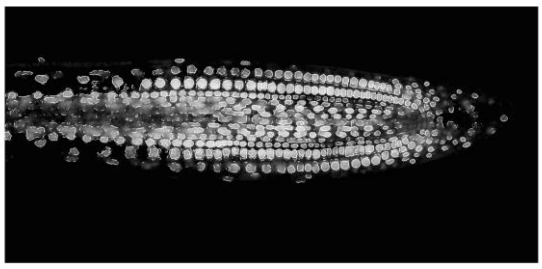


Figure 2.3.2(A) - Canny Edge

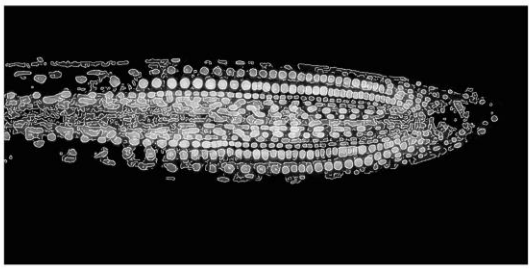


Figure 2.3.2(B) – Sobel Edge

2.3.3 Morphological Transformation

For the last part of the pre-processing, we use top hat and bottom hat filtering to get more separation of the nuclei using a disk element of radius 10. This is done by applying top hat filter to the image then adding the original image with the top hat filtered image. After that, we subtract it with the bottom-hat filtered image. In matlab syntax, it is

$$im_h = imsubtract\left(imadd\left(K, imtophat(K, se)\right), imbothat(K, se)\right),$$

where im_h *is the output image,* K *is the image to be improved, and se is a disk element with radius of* 10

This produces a clear segmentation between nuclei, and an increase of sharpness and contrast as seen in **Figure 2.3.3**

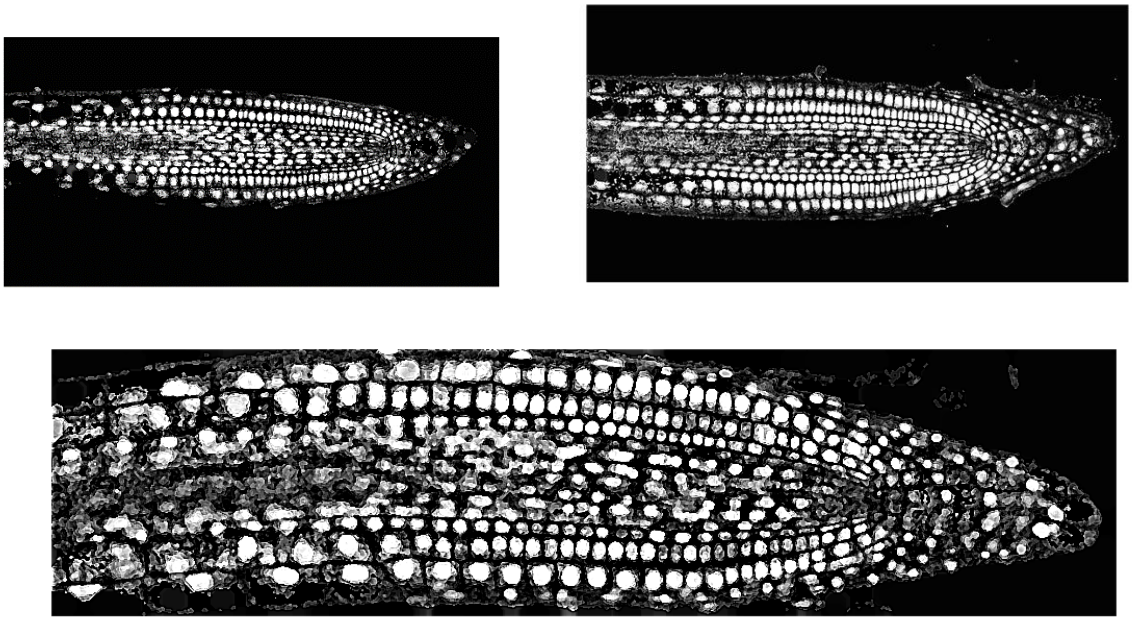


Figure 2.3.3 – Top Hat and Bottom Hat Filtering(Morphological Transform)

2.4 Thresholding

Now that the image is sharp and pre-processed enough, the thresholding process comes in place. Otsu Thresholding is chosen as it computes a global threshold from the morphologically transformed image. Therefore, this is the best option as other thresholding methods pick up too much noise as seen in **Figure 2.4.2**.

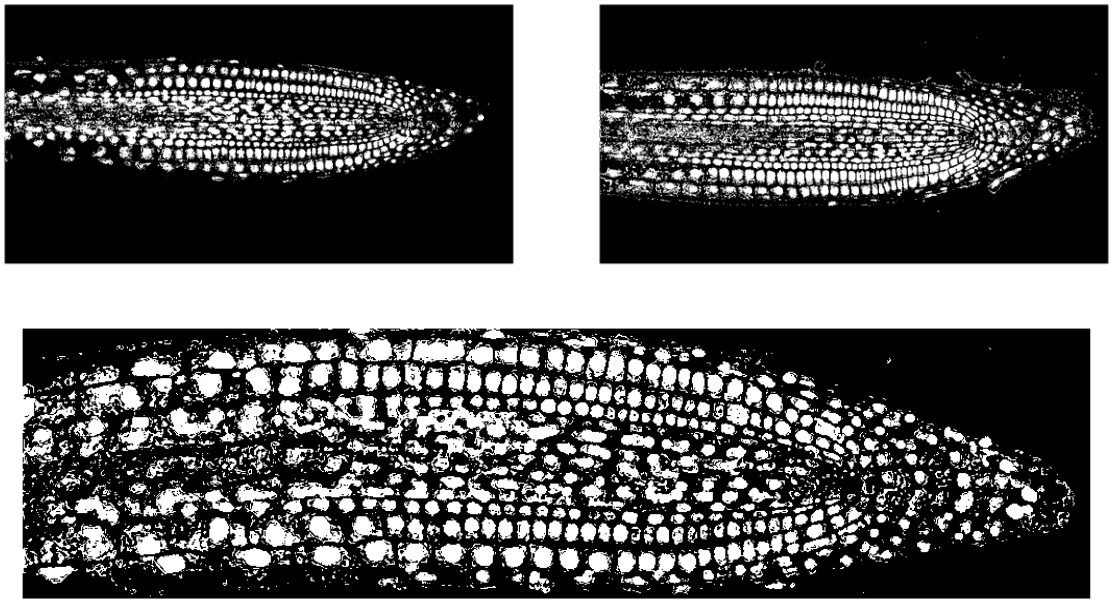


Figure 2.4.1 – Otsu Thresholding

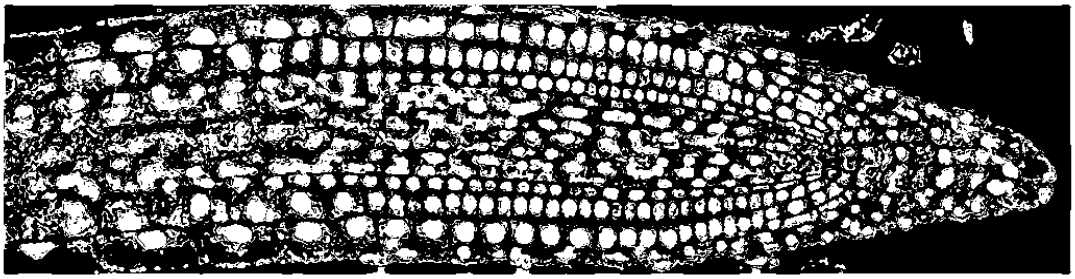


Figure 2.4.2 – Adaptive Method Thresholding

2.5 Binary Image Processing

From the thresholded image, it is obvious that although the segments can be seen, and the edges of the nuclei are sharp, there is still some noise and un-segmented nuclei, resulting in sizes of all distributions. Although nuclei sizes differ, there should not be such a fluctuation. The solution to this problem would be to firstly eliminate the small speckles. This can be done by using finding the connected components within the image, use region props to get the area properties of the components, label them, then only include regions with pixel area above 55 within the final result. This is followed by a morphologically opening the image with a disk element of radius 2, which produces the results shown in **Figure 2.5.1.**

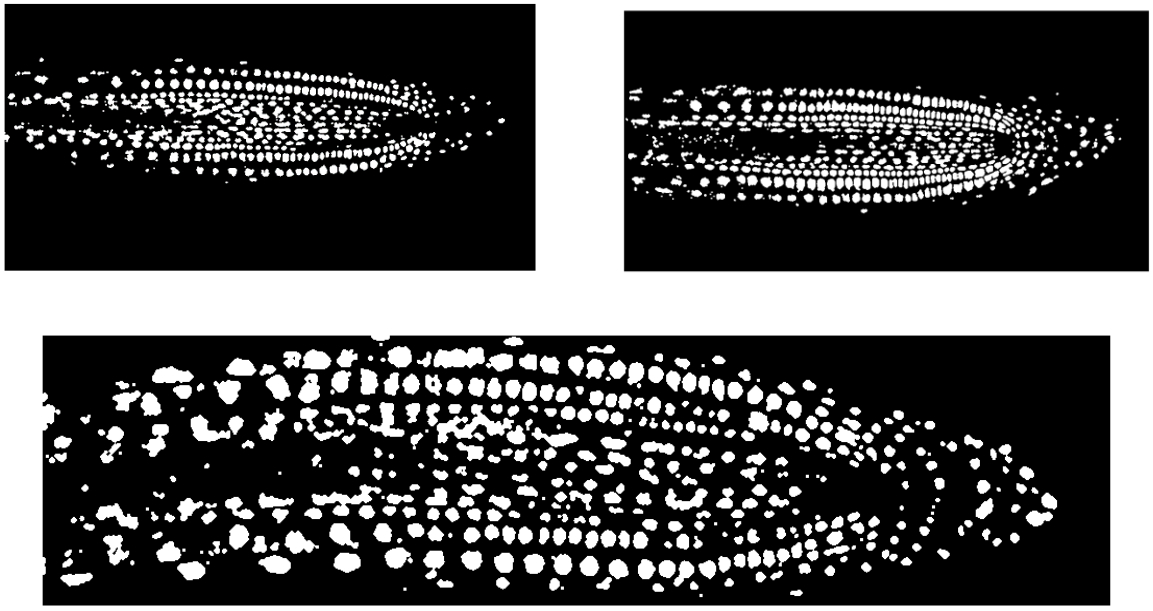
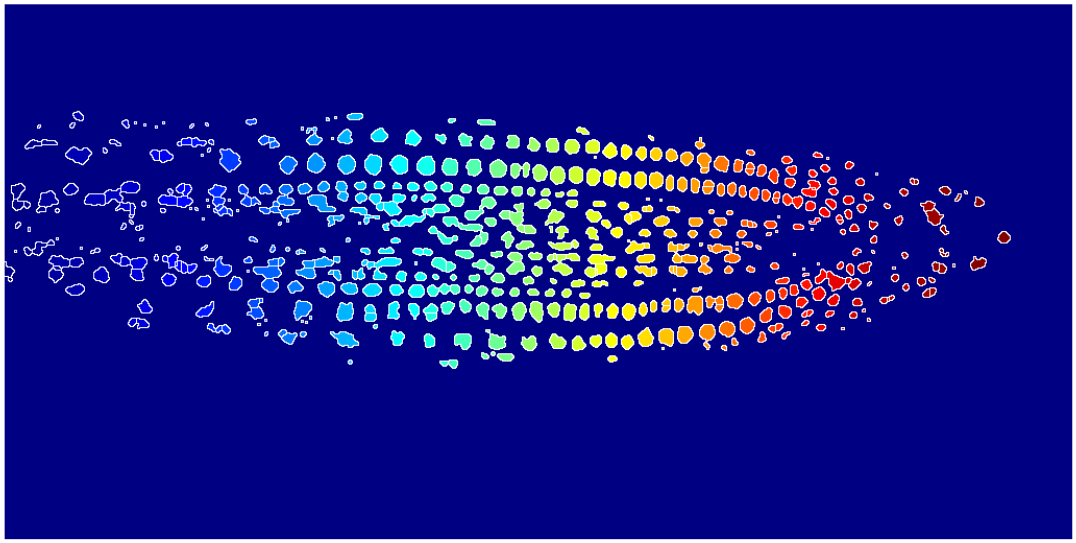
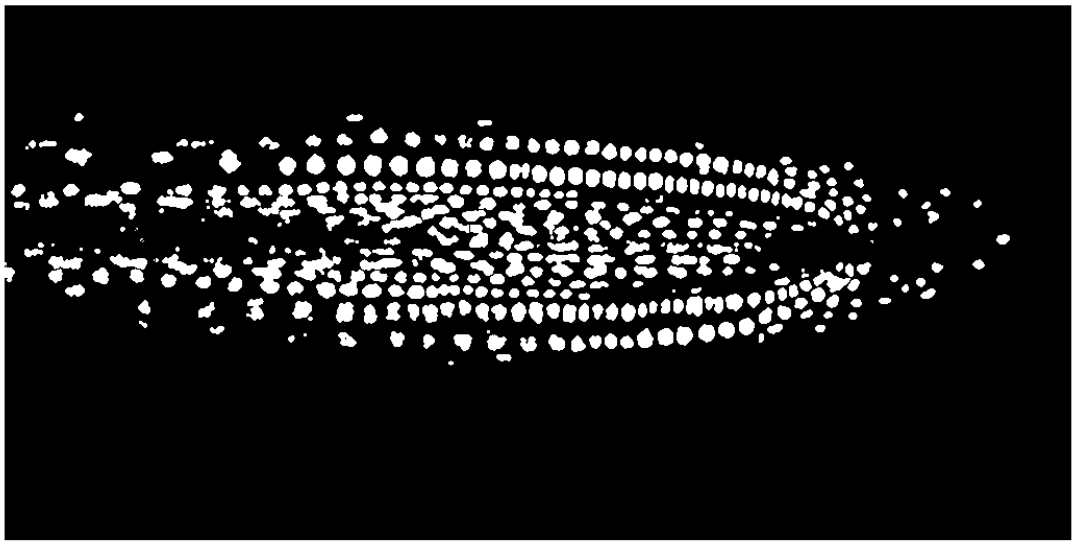


Figure 2.5.1 – Noise Removal and Morphological Transform

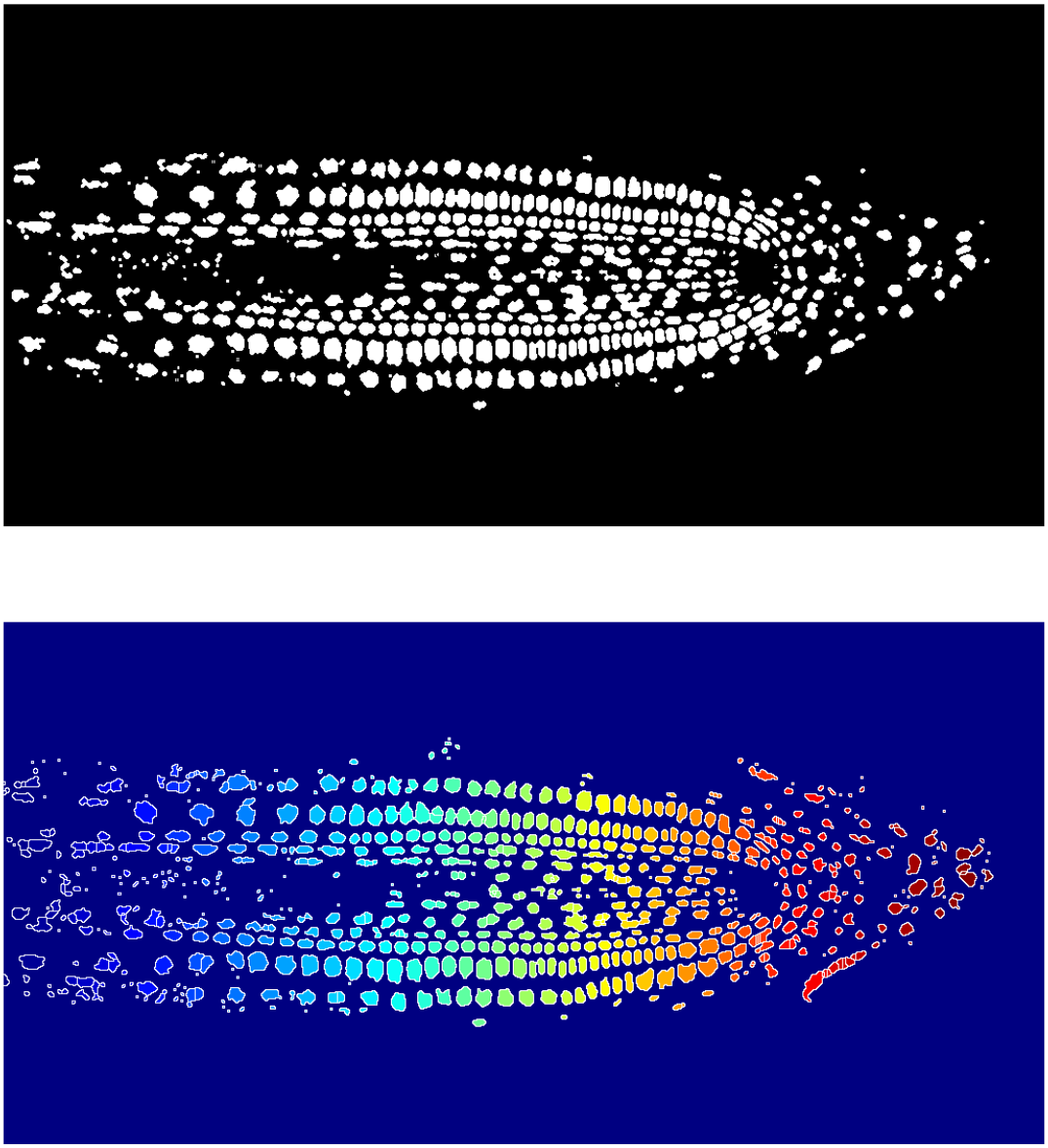
2.6 Watershed and Final Results

As seen in **Figure 2.5.1**, the nuclei is now separated and the noise speckles has been removed. However, there is inconsistency in terms of nuclei size, and many fluctuations. This is due to a lack of segmentation. Therefore, I have used watershedding to segment the nuclei. An alternative segmentation method would be Region Growing which chooses initial seed points and checks neighbouring pixels to determine if they should be added to the region. The challenge comes in finding the ideal seed points, which could be done through the combination of K-means clustering. However, watershedding would be an ideal and a more efficient method which requires less steps which satisfactory results. The watershed and final results can be seen below.

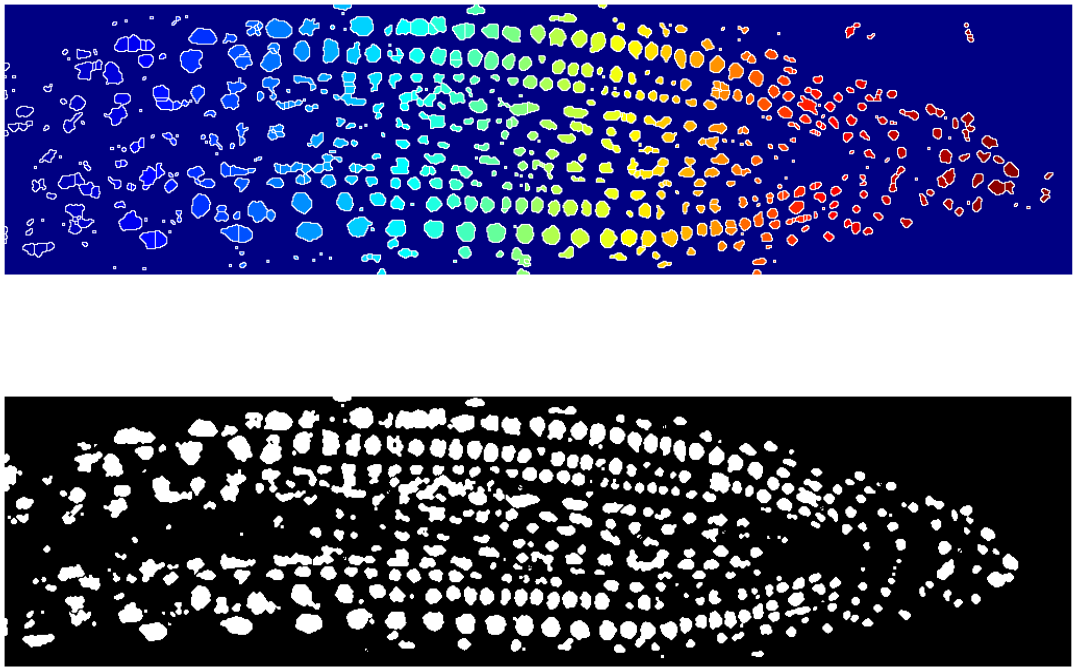




StackNinja1.bmp



StackNinja2.bmp



StackNinja3.bmp

3.0 Image Overlay

Below are the final results for the nuclei overlaid upon the plant root. The top image shows a fused image of the identified nuclei on top of the original image. The middle image shows the nuclei location being circled, and the bottom image shows the original and unprocessed hue channel being overlapped with the plant root. Which displays a lack of nuclei as compared to the processed version. This indicates that we could have extracted the hue level of the gamma correction image to get the nuclei, but it will not be as accurate compared to having the processing funnel above in this report.

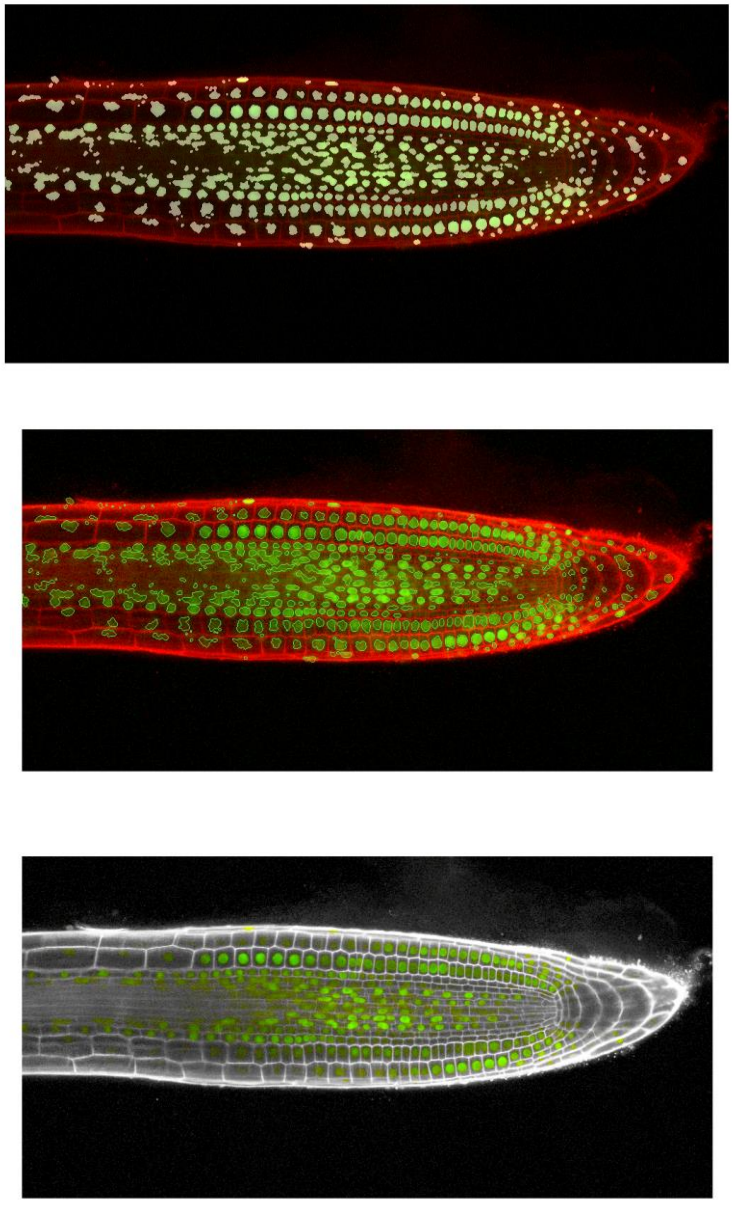


Figure 3.0.1 – StackNinja1

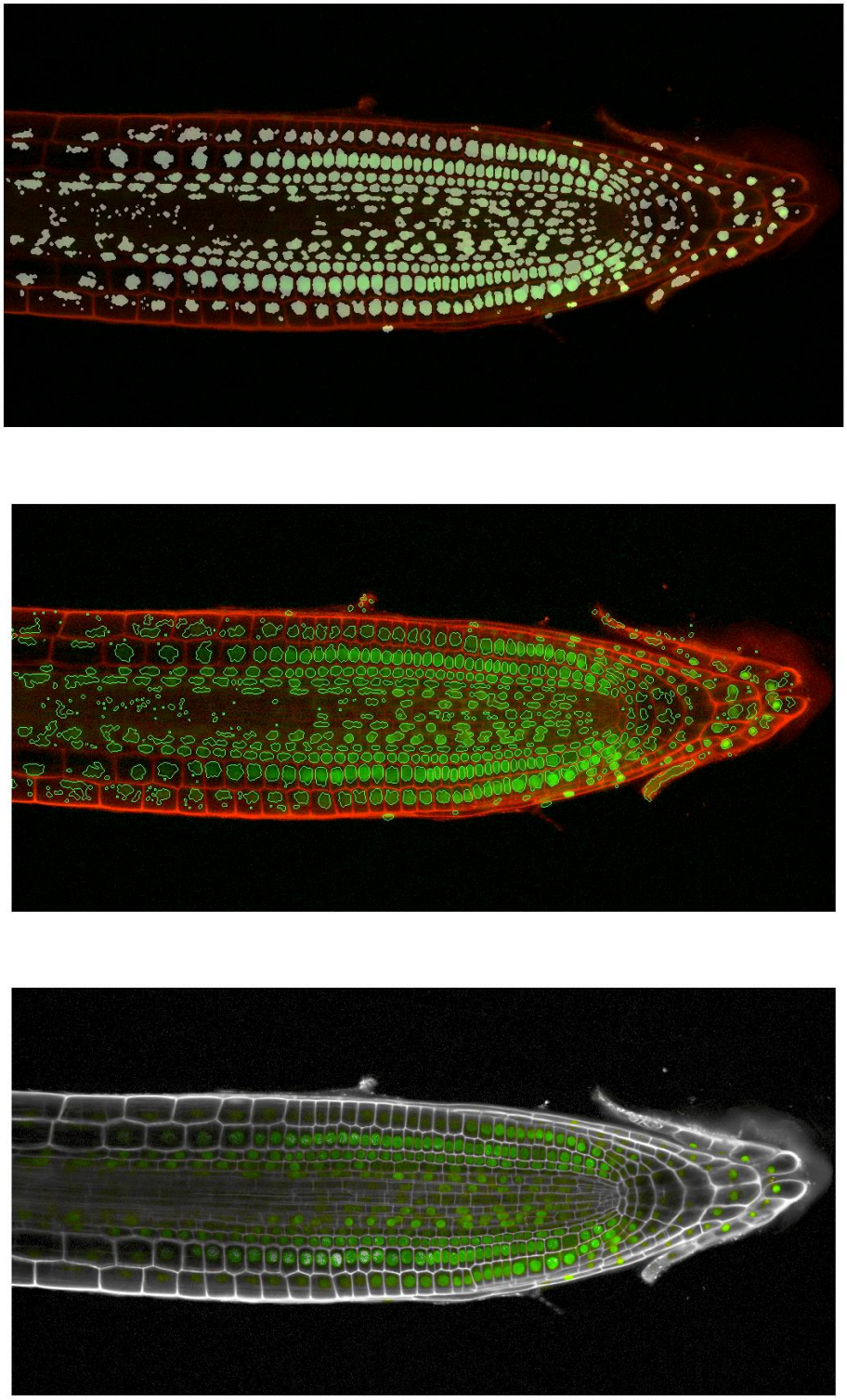


Figure 3.0.2 – StackNinja2

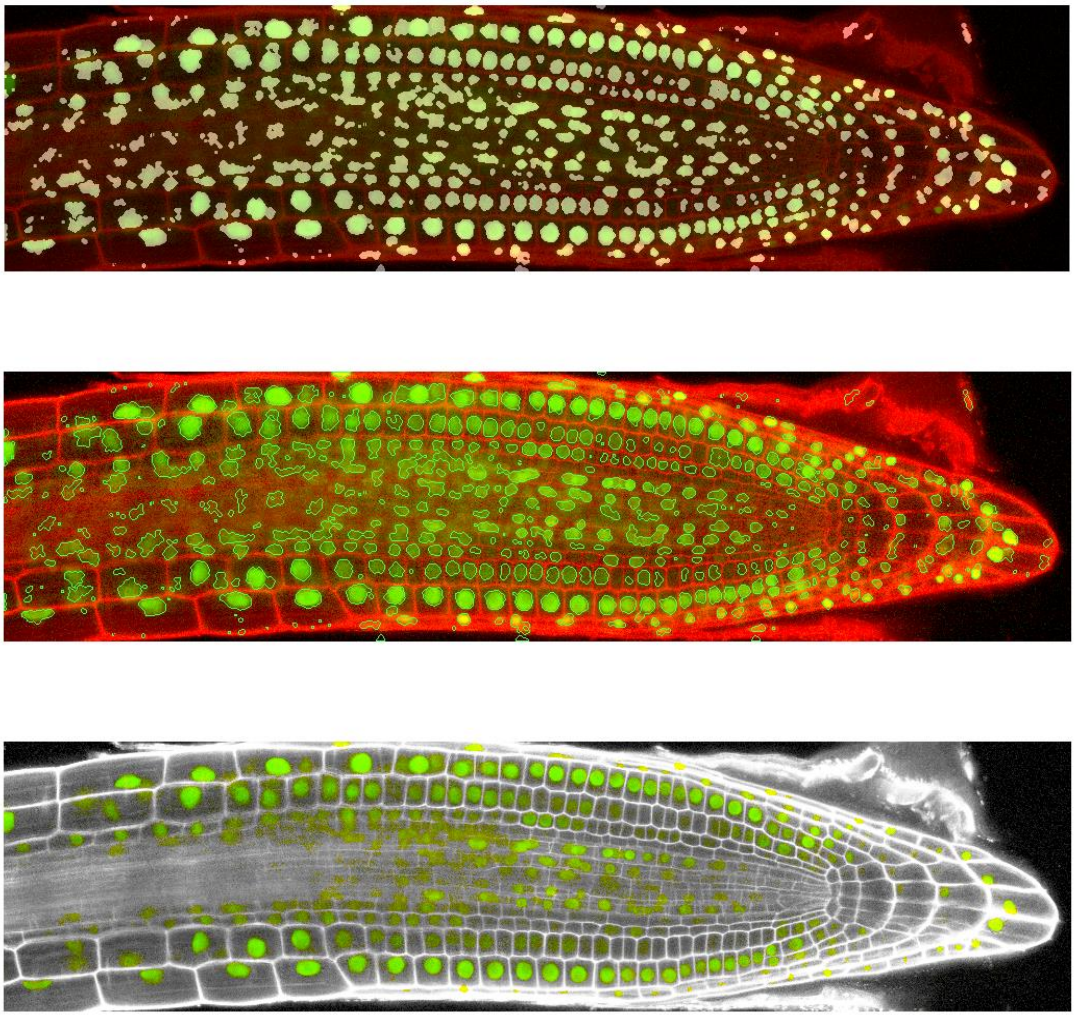


Figure 3.0.3 – StackNinja3

4.0 Data Analysis

This section consists of the analysis carried out and the methods used to accurately measure the properties of the nuclei. After that, the final image with nuclei count, graphs and histograms plotted will be displayed.

4.0.1 Nuclei Count

For the nuclei count, we first find connected components within the image of the final result, and then return the number of objects found which can be extracted from the structured dataset.

4.0.2 Size Distribution

The size or area of a given object can be defined as the number of pixels within a given shape. A normal distribution histogram for the size of nuclei can be plotted through utilizing the regionprops method, which is able to extract the area of each 8-connected component in the final result binary image. For the size distribution histogram, the y-axis indicates the number of components, whilst the x-axis indicates the region size. If we take the overall average size of all 3 plant roots, there is an average size of 120.3 pixels.

4.0.3 Brightness Analysis

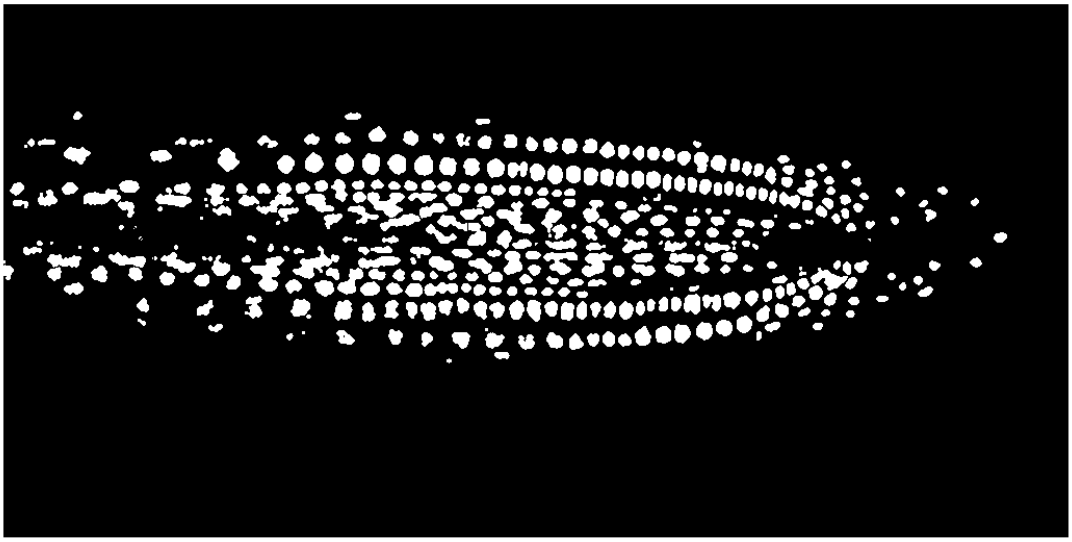
The brightness of a nuclei can be determined as the intensity. This will give an indicator of how bright or dark the green nuclei is. Saturation indicates the “purity” of a given colour, but not the brightness. Meanwhile hue indicates the wavelength of the colour. Hence, we took the value channel from the original HSV image and not the saturation or hue channel, to be passed as the parameter when finding the ‘MeanIntensity’ with the regionprop filter. However, this returns the values in the form of struct. Therefore, the struct2cell function is used so that we may plot the histogram. The brightness scale is from 0 to 1, where 0 represents the darkest nuclei and 1 represents the brightest nuclei.

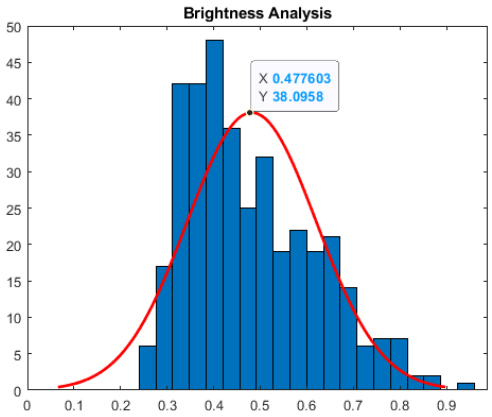
4.0.4 Shape Analysis

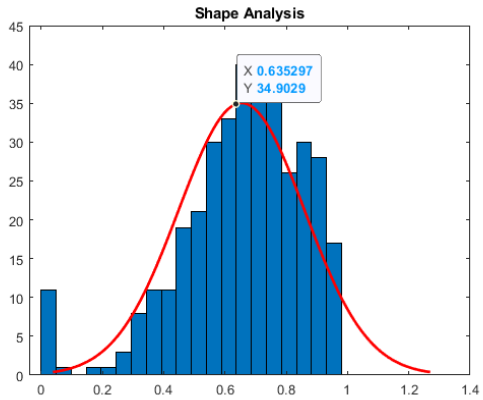
To run analysis on the shape of the nuclei, we use Eccentricity which can sometimes be known as ellipticity. This can be defined as the the short axis length divided by the long axis length of a given component, which gives a value between 0 and 1. A value leaning towards the side of 0 indicates a shape that is more rounded, whilst a value leaning towards the side of 1 indicates a shape that is more linear.

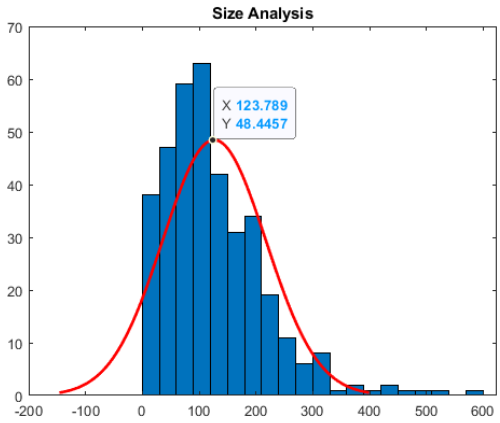
4.1 Analysis Results

4.1.1 First Image(StackNinja1.bmp)



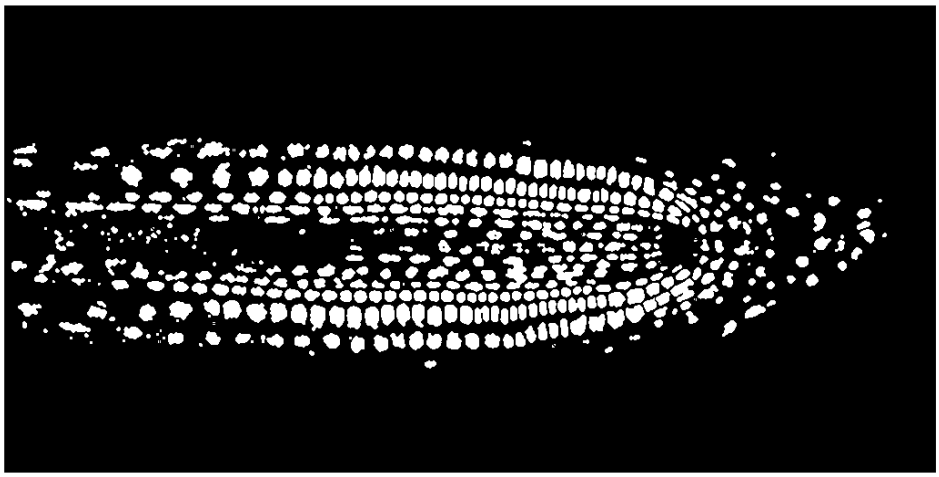


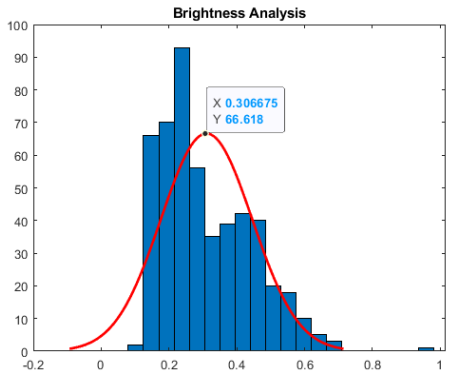


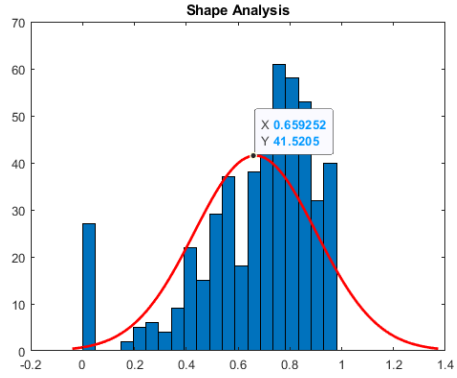


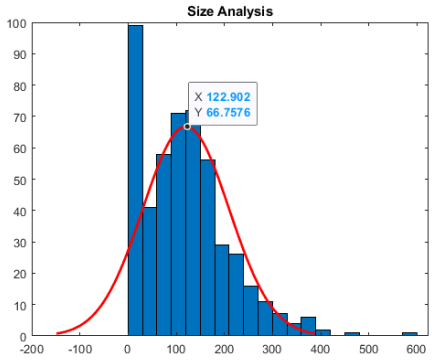
The nuclei count of the first image is 368. In the brightness analysis, the median value on the x-axis indicates a balanced level of 0.447. The shape analysis shows a median value of 0.63 on the x-axis, indicating that the shape is circular but also leaning more towards a linear side. On the size analysis for the first image, the average pixel in a shape is 123.8.

4.1.2 Second Image(StackNinja2.bmp)



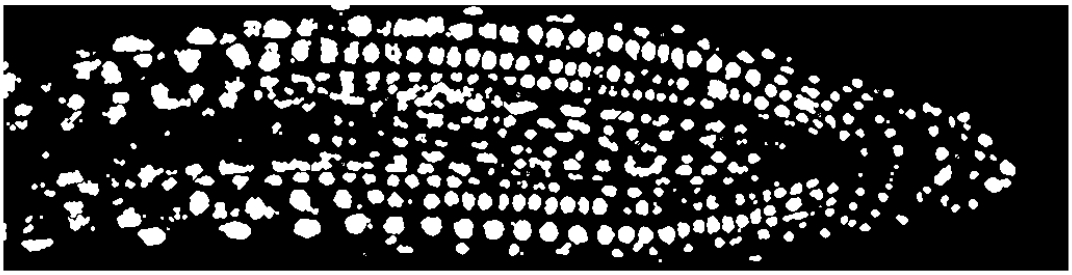


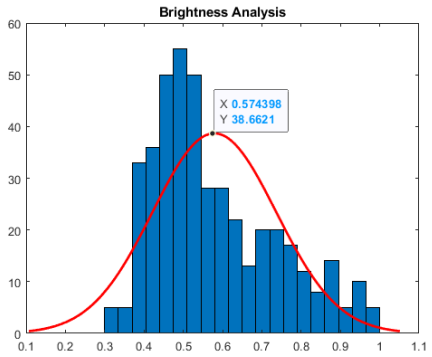


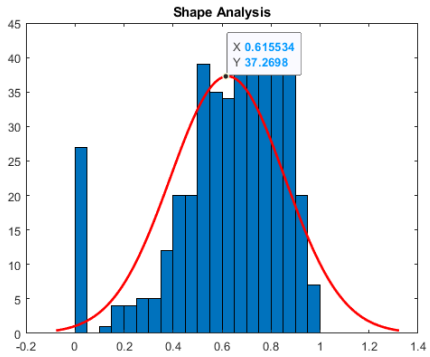


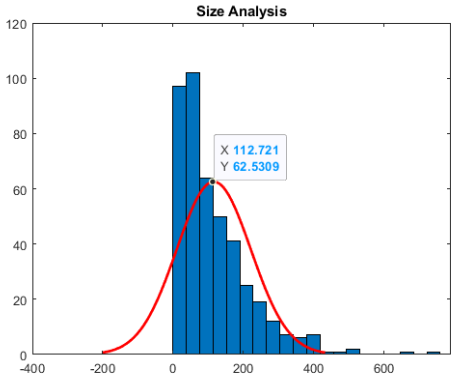
In the second image, there is a total nuclei count of 500, higher than the first image. The median X value of the brightness analysis shows that it is slightly dimmer than the first. Whilst the median value of the shape analysis indicates that the nuclei in this second image is similar to the first. Lastly, the size analysis shows that the total pixels in a shape has a value of 122.9, slightly smaller than the first image.

4.1.2 Third Image(StackNinja3.bmp)









Lastly, the final image has a nuclei count of 436. The brightness analysis proves it is the brightest image out of the 3.

5.0 Conclusion

This paper shows an image processing funnel which segments nuclei in microscopic images from CLSM, and further using quantitive analysis to confirm image properties through proven datasets, such as the third image being the brightest out and the second being the dimmest which we can clearly see from our naked eye. This proves the accuracy of the methods use in this paper.

A full demo of the software and details can be viewed from this youtube link : https://youtu.be/TJmvuYbv5pE

6.0 References

1. Michael A. Wirth, 2004, “Shape Analysis and Measurement”,Computing and Information Science Image Processing Group

2. Matlab, “Region Props, Measure properties of Image Regions”, 2021, https://www.mathworks.com/help/images/ref/regionprops.html

3. Matlab, “Graythresh, Global image threshold using Otsu’s method”, 2021, https://www.mathworks.com/help/images/ref/regionprops.html

4. Steve Eddins, “The Watershed Transform : Strategies for Image Segmentation”, Mathworks, https://www.mathworks.com/company/newsletters/articles/the-watershed-transform-strategies-for-image-segmentation.html

5. Matlab, “Watershed, Watershed Transform”, Mathworks, https://www.mathworks.com/help/images/ref/watershed.html

6. Matlab, “imbothat, Bottom Hat Filtering”, Mathworks, https://www.mathworks.com/help/images/ref/imbothat.html

7. Steve Eddins, 2013, “Watershed transform question from tech support”, Mathworks, https://blogs.mathworks.com/steve/2013/11/19/watershed-transform-question-from-tech-support/

8. Steve Eddins, 2006, “Cell Segmentation”, Mathworks, https://blogs.mathworks.com/steve/2006/06/02/cell-segmentation/

9. Wikipedia, 2021, “Hue”, < https://en.wikipedia.org/wiki/Hue>