



**Image Processing for Analyzing  
Illuminated Plant Coursework Report**

**Student name:** Teo Shi Bin

**Student ID:** 20183717

**Course of study:** CSAI

**Date of Submission:** 24/4/2021

## Contents

Introduction .....	3
Implementation .....	3
Function Structure .....	3
Count Nuclei with Regional Max.....	4
Count Nuclei with Watershed .....	5
Count Cells with Watershed .....	8
Connected Component Result .....	9
Calculate Statistic.....	10
Object Properties .....	10
Object counts .....	11
Evaluation .....	12
Conclusion.....	12
Attachments.....	13
README .....	13
Video Presentation .....	14
Bibliography .....	15
Figure 1. function structure .....	3
Figure 2. count nuclei with regional max function process .....	5
Figure 3 piece-wise linear stretching with coordinates of (30, 64), (255 255) .....	6
Figure 4 count nuclei with watershed process .....	7
Figure 5 count cells function process.....	8
Figure 6 result - connected components .....	9
Figure 7 statistic table.....	10
Figure 8 more properties calculated from one table.....	10
Figure 9 number of objects.....	11
Table 1 statistic property name and calculation.....	11

## Introduction

This report will go through most of the function implementations and discuss the reason and purpose for its implementation. Furthermore, I will evaluate the pros and cons of the approach and talk about other alternative solutions and improvements. The GitHub link and video link will be placed at the end of the attachments. Due to the length of the output result, the second and third analysis will not be included within this report. However, all analysis result was manually saved in the attachments folder within the report folder.

## Implementation

The idea that I came up for dealing with the image is actually analyzing different color channels separately for dedicated purpose. I've created 3 functions that deal with red and green channel independently. The red channel consist only the cell walls and the green channel consist only the nuclei, so the best way will be using them separately.

## Function Structure

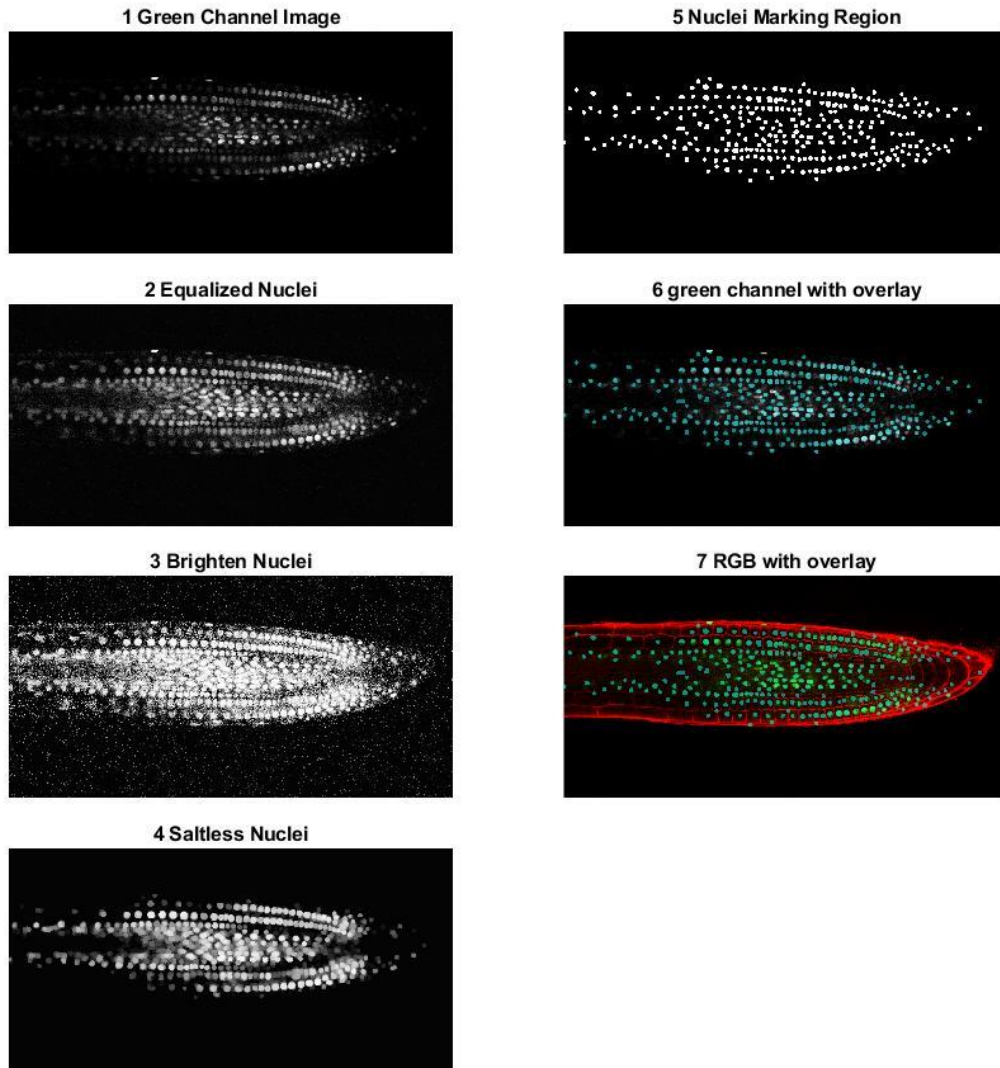
Before going straight into all of the functions, a brief understanding of all functions is required. For the clarification all functions within the folder is fully written by myself.

```
.
├─ main.m (analyse 3 given image)
│   ├─ analyse_plant_illumination.m (analyse an image)
│   │   ├─ count_nuclei.m (analyse green channel with regional max)
│   │   │   └─ autoimfill.m (region fill with automated seed generation)
│   │   ├─ count_nuclei_2.m (analyse green channel with watershed)
│   │   ├─ count_cells.m (analyse red channel with watershed)
│   │   ├─ whole_statistic.m (calculate and print statistic of an image)
│   │   │   └─ calculate_statistic.m (calculate statistic of a property)
│   │   └─ piecewiseLinear.m (image transformation method)
│   └─ subplottight.m (custom subplot)
└─ my_disp.m (custom disp)
```

Figure 1. function structure

## Count Nuclei with Regional Max

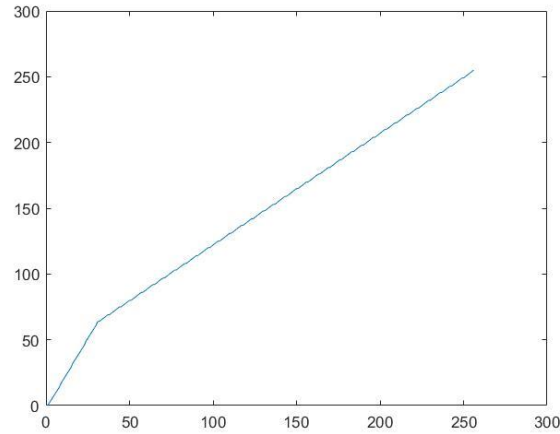
Figure 2 indicates the process of the entire count nuclei function. First of all, split out the green channel that contains nuclei. The split image is then enhanced using histogram equalization in order to reveal even more nuclei. The image will then be brighten using `imlocalbrighten` [1] function by raising the brightness of the image dynamically based on the intensity of the region. During the process of local region brightening, it introduces a lot of salt noise as well, so this is why the following procedure is to denoise the image using morphology opening operation. The final step is to extract local intensity maxima using a morphology operation called `imregionalmax` [2] so that the brighter point of the nuclei is then extracted and will be spaced separately. Ultimately the local max intensity image will be used solely for calculating the number of nuclei but not for the statistical values, mainly because the image doesn't necessarily represent the nuclei region, it is just the bright spot of the nuclei. The image is then fed into `bwconncomp` [3] to get the labeled connected components in Figure 6.



*Figure 2. count nuclei with regional max function process*

### Count Nuclei with Watershed

First in Figure 4, I split the green channel from the RGB image. Next, applying median, average and unsharpening [4] to reduce the noise and enhance it which will improve the clarity of the image when we do the binarization. Additionally, using piece-wise linear transformation with the plot shown below to brighten the image.



*Figure 3 piece-wise linear stretching with coordinates of (30, 64), (255 255)*

Furthermore, I am using a built-in morphological operation called `imreconstruct` [5] to perform opening operation that doesn't affect the original shape of the object as compare to the actual opening operation [6]. For extra information, a complete list of morphological operation can be found here [7]. Next, thresholding the reconstructed image with adaptive binarization. Now, if we feed the binarized image directly into watershed algorithm, it will cause over segmentation due to the fact that it is still considerably noisy in the binarized image. Thus, instead of doing that I transform the image with Euclidean distance transformation [8] then parse it into watershed. After, merging the binarized image with the watershed label image we get the image shown in Figure 4 count nuclei with watershed step 8. The result was then overlaid on top of green channel and original image in step 9 and 10. Finally, the image will be fed into `bwconncomp` to calculate the number of objects and compute the statistics.

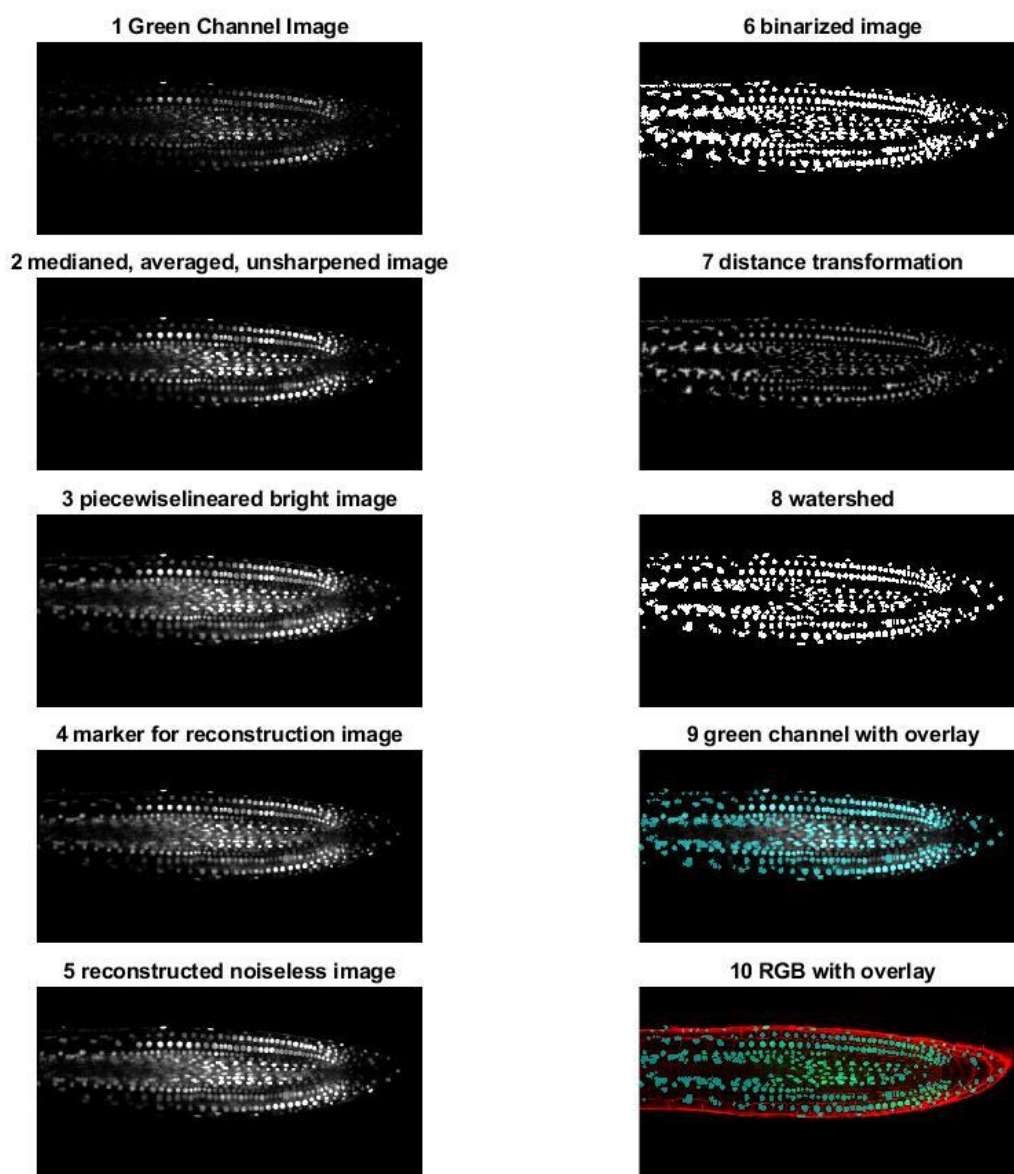


Figure 4 count nuclei with watershed process

## Count Cells with Watershed

Figure 5 indicates the process of analyzing using red channel. The first step is to remove some salt noise so median is used. After blurring the image with median transformation, un-sharpening is executed to sharpen the image back to its clear state. Next, piecewise linear stretching is done to enhance the contrast. After applying those enhancements, the image is being binarized. The binarized image is then dilated to thicken the cell walls due to the reason that object counting function require the object to space out properly. In the 5<sup>th</sup> image the border is filled with color white using a function that is created by me that called autoimfill. It is capable of generating seeds that is located outside of the object itself. Due to the complexity of autoimfill, I will not include it in this report as it has nothing to do image processing, it is more likely to be an algorithm topic, therefore the entire logic behind the scene will be in the comments of the code. Furthermore, the region filled image will be inverted to compute the Euclidean distance and fed into watershed, at the end drop small region using morphology opening. Finally, the segmented image is parse into bwconncomp to generate the result in Figure 5

**Error! Reference source not found..**

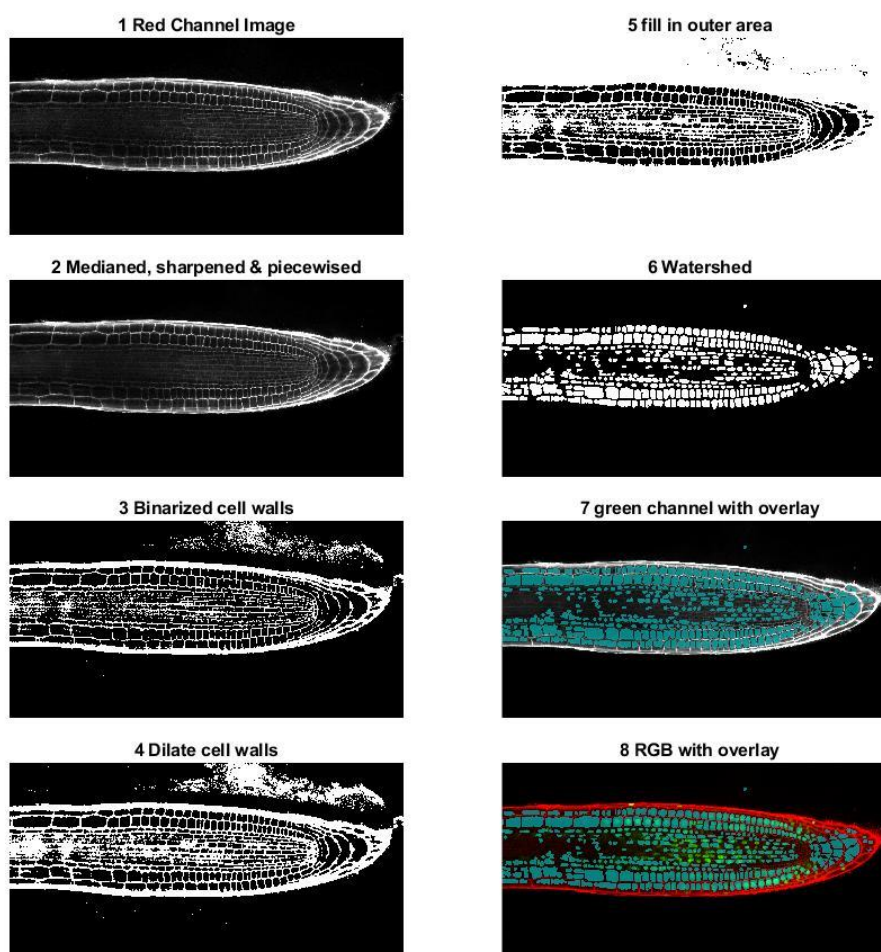


Figure 5 count cells function process



## Connected Component Result

This is the complete result after parsing the same image into 3 different analyze function where using watershed, regional max for green channel and watershed again for red channel, we get the result shown below.

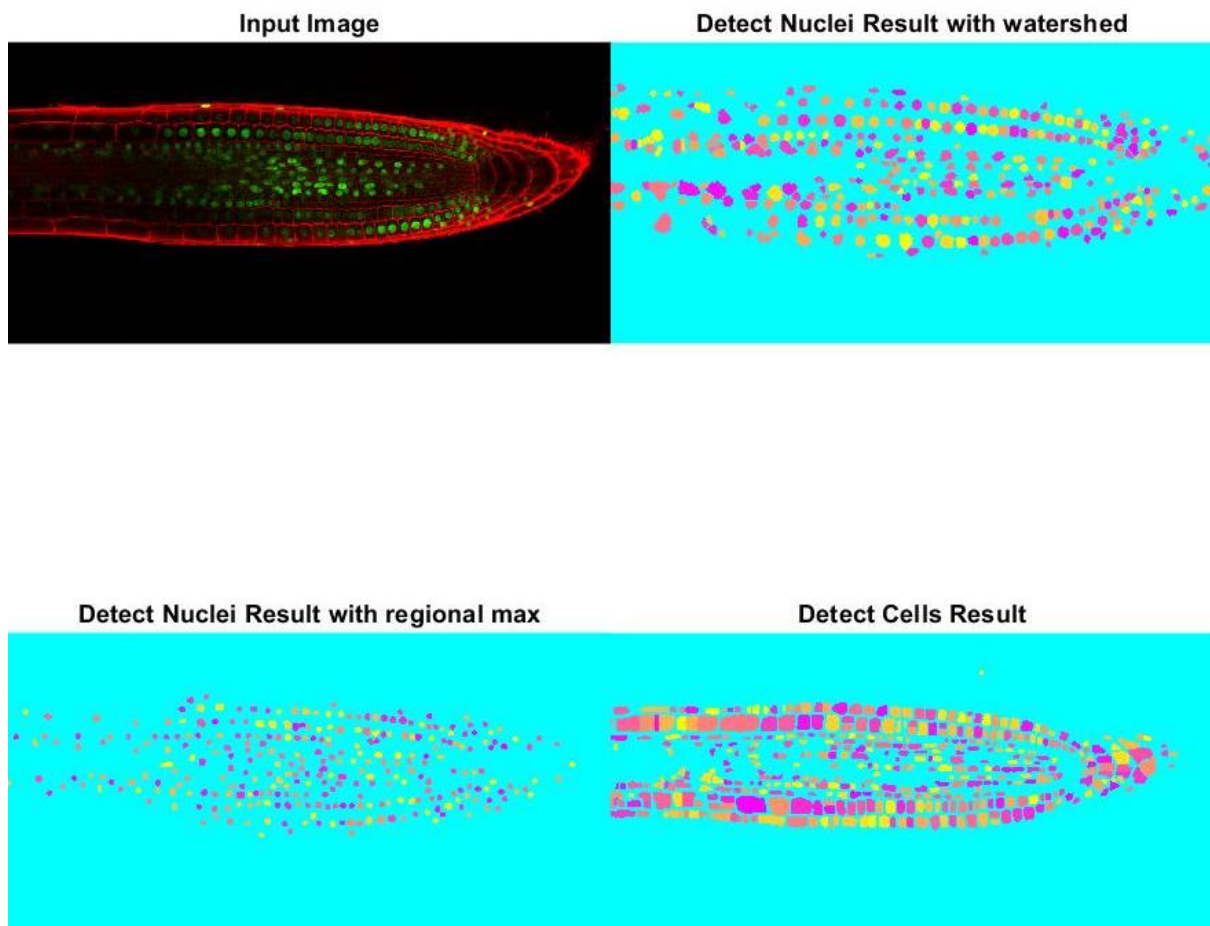


Figure 6 result - connected components

## Calculate Statistic

### Object Properties

The Statistic table shown in Figure 7 is calculated through regionprops [9] function that is a built-in function in MATLAB, it allows all sorts of statistical analysis to be done. Figure 7 in this report is showing only the first 5 rows of objects' properties, to view the whole table please run the script that I've given. This table is generated using the connected connection struct that is return by bwconncomp along with its corresponding image channel. Further calculation using the table Figure 7, return more results shown in Figure 8, this calculation is done for each Figure 7 statistic table. Furthermore, the descriptions of Figure 8 is shown in Table 1.

Statistical Analysis of Nuclei with Regional Max =							
Area	MajorAxisLength	MinorAxisLength	Circularity	Perimeter	MeanIntensity	MinIntensity	MaxIntensity
39	9.5273	5.4608	1.1374	20.758	25.282	9	53
69	9.4086	9.4086	1.2632	26.2	31.043	10	64
69	9.4086	9.4086	1.2632	26.2	10.29	1	28
91	11.742	9.9362	1.2062	30.79	22.769	8	48
105	12.699	10.626	1.1767	33.486	27.486	12	50
Statistical Analysis of Nuclei with Watershed =							
Area	MajorAxisLength	MinorAxisLength	Circularity	Perimeter	MeanIntensity	MinIntensity	MaxIntensity
162	18.448	11.399	0.99732	45.18	15.549	0	53
350	24.793	19.77	0.66291	81.454	9.0514	0	35
109	15.329	9.7044	0.90733	38.854	17.229	0	56
78	10.492	10.171	0.97185	31.758	6.9744	0	23
54	9.6185	7.6756	0.9121	27.276	2.8889	0	15
Statistical Analysis of Cell Walls with Watershed =							
Area	MajorAxisLength	MinorAxisLength	Circularity	Perimeter	MeanIntensity	MinIntensity	MaxIntensity
64	19.046	4.846	0.58693	37.017	18.109	7	42
242	30.789	10.186	0.70831	65.524	16.136	1	34
180	25.413	9.3879	0.75327	54.798	13.556	3	36
99	23.14	7.0664	0.554	47.388	21.535	5	67
491	47.007	13.589	0.62763	99.15	28.817	8	52

Figure 7 statistic table

```
Min & Max Intensity Stats =
    IndividualMean: [422x1 double]
        MinorMean: 11.0995
        MajorMean: 57.9739
        PopulationMean: 34.5367
        MinorMedian: 10
        MajorMedian: 53
        PopulationMedian: 32
        MinorMinimum: 0
        MinorMaximum: 64
        MajorMinimum: 8
        MajorMaximum: 145
        StandardDeviation: 15.1323
        Variance: 228.9862

Area Stats =
    mean: 184.8294
    median: 123
    StandardDeviation: 176.6159
    Variance: 3.1193e+04

Circularity Stats =
    mean: 1.0043
    median: 1.0132
    StandardDeviation: 0.1500
    Variance: 0.0225

Perimeter Stats =
    mean: 45.7947
    median: 39.4490
    StandardDeviation: 19.8871
    Variance: 395.4972

Local Region Mean Intensity Stats =
    mean: 28.2632
    median: 25.8707
    StandardDeviation: 12.3777
    Variance: 153.2084

Min & Max Length Stats =
    IndividualMean: [422x1 double]
        MinorMean: 11.5020
        MajorMean: 18.7262
        PopulationMean: 15.1141
        MinorMedian: 9.7018
        MajorMedian: 16.9934
        PopulationMedian: 13.3042
        MinorMinimum: 4.8460
        MinorMaximum: 32.2822
        MajorMinimum: 8.1775
        MajorMaximum: 71.8476
        StandardDeviation: 5.9265
        Variance: 35.1236
```

Figure 8 more properties calculated from one table

### *n-by-1 matrix attributes*

They are fairly straight forward. Mean, median, SD, variance can all be calculated using 1-D vector

### *n-by-2 matrix attributes*

These attributes are grouped data where there will be upper and lower bound to it.

*Table 1 statistic property name and calculation*

Property name	Description
Individual mean	Mean of pair. E.g. $(a + b) / 2$
Major mean / Minor mean	Mean of minimum values or maximum values
Population mean	Mean of individual means, the overall mean
Major median / Minor median	Median of minimum values or maximum values
Population median	Median of individual means
Major / Minor – Maximum / Minimum	Maximum or minimum value of the bounds
Standard Deviation	Calculated using individual means
variance	Power of SD

### Object counts

The approximated number of cells is shown as Figure 9 in the command window.

```
Numbers of nuclei detected with method 1 =  
299  
  
Numbers of nuclei detected with method 2 =  
360  
  
Numbers of cells detected =  
422  
  
Hit Enter to proceed to next analysis
```

*Figure 9 number of objects*

## Evaluation

Analyzing different color channel separately can have a major flaw if the initial illumination images are illuminated using different light source. The main reason is that the program is assuming that nuclei will be green and cell walls will be red, without the proper color illumination, wrong object in different color channel will be fed into the wrong algorithm. The best approach to eliminate this problem is to design a more general gray scaled analyzer so we can feed any color into it due to the fact that the input image will be gray scaled anyway. However, implementing such algorithm require techniques that will split the nuclei and cell walls since the nuclei and cell walls are all in one image in that scenario.

Secondly, it is possible to detect even more cells even when certain cells are darker than most of the cells. Some cells are too dim to be detected and it is crucial if we do want to get the greatest number of nuclei out of it.



## Conclusion

The image is going through one after another transformation to achieve certain patterns and with that we're able to detect the edges, shapes and all sort of useful application, I am fairly happy with the implementation that I manage to come up with, but I can't wait to improve the algorithm even more.

## Attachments

README

GitHub Link: [https://github.com/teoshibin/COMP2032\\_IIPCW](https://github.com/teoshibin/COMP2032_IIPCW)

 README.md 

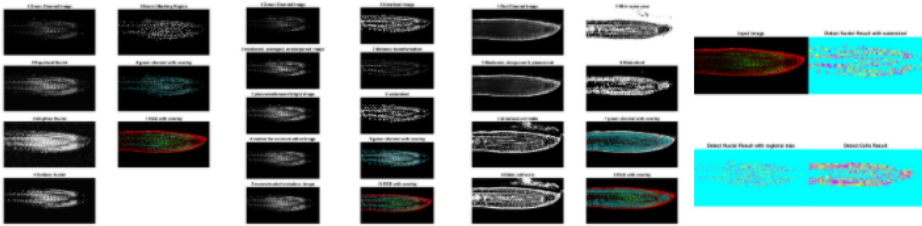
---

# Image Processing - plant illumination analysis

---

## Results

---



The Results section displays a grid of 24 small images arranged in 4 rows and 6 columns. The first 5 columns show various stages of image processing for three different plant images. The 6th column shows statistical results for each image, represented as heatmaps. The first row shows the original images. The second row shows the images after background removal. The third row shows the images after thresholding. The fourth row shows the images after edge detection. The sixth column shows the statistical results for each image, represented as heatmaps.

### Statistical Results

---

## USAGE

---

### Run Example Script

1. Navigate the directory of MATLAB into the folder containing "main.m"
2. type "main" in the console and hit enter, it will run analysis using the first image.
3. Hit enter in the console to execute the next analysis of next image (there are 3 images in total)

### How to analyse other image

1. Place preferred image into [Image folder](#)
2. call `analyse_plant_illumination` with specified image to generate the result

see also

Example run script implementation [main.m](#)

## Video Presentation

The presentation video can be found within the same folder as the MATLAB code or you can use the link below to access it via YouTube. The video is around 4 minutes long, this is as fast as it possibly gets to cover everything about the result without diving into any of the code.

Video Presentation	<a href="https://youtu.be/qHij8u6A-cg">https://youtu.be/qHij8u6A-cg</a>	
--------------------	---	---

## Bibliography

- [1] "Brighten low-light image - MATLAB imlocalbrighten."  
<https://www.mathworks.com/help/images/ref/imlocalbrighten.html> (accessed Apr. 17, 2021).
- [2] "Regional maxima - MATLAB imregionalmax."  
<https://www.mathworks.com/help/images/ref/imregionalmax.html> (accessed Apr. 17, 2021).
- [3] "Find connected components in binary image - MATLAB bwconncomp."  
<https://www.mathworks.com/help/images/ref/bwconncomp.html> (accessed Apr. 17, 2021).
- [4] "Sharpen image using unsharp masking - MATLAB imsharpen."  
<https://www.mathworks.com/help/images/ref/imsharpen.html> (accessed Apr. 17, 2021).
- [5] "Morphological reconstruction - MATLAB imreconstruct."  
<https://www.mathworks.com/help/images/ref/imreconstruct.html> (accessed Apr. 17, 2021).
- [6] "Marker-Controlled Watershed Segmentation - MATLAB & Simulink Example."  
<https://www.mathworks.com/help/images/marker-controlled-watershed-segmentation.html> (accessed Apr. 16, 2021).
- [7] "Morphological Operations - MATLAB & Simulink."  
<https://www.mathworks.com/help/images/morphological-filtering.html> (accessed Apr. 16, 2021).
- [8] "Watershed transform question from tech support," *Steve on Image Processing with MATLAB*.  
<https://blogs.mathworks.com/steve/2013/11/19/watershed-transform-question-from-tech-support/> (accessed Apr. 16, 2021).
- [9] "Measure properties of image regions - MATLAB regionprops."  
<https://www.mathworks.com/help/images/ref/regionprops.html> (accessed Apr. 17, 2021).