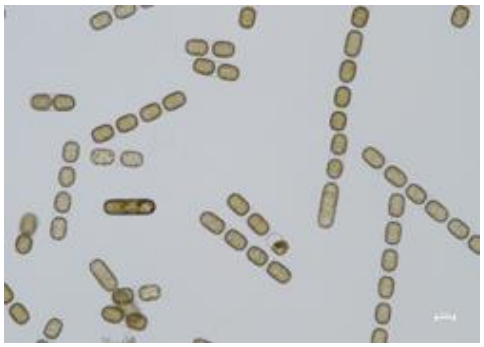
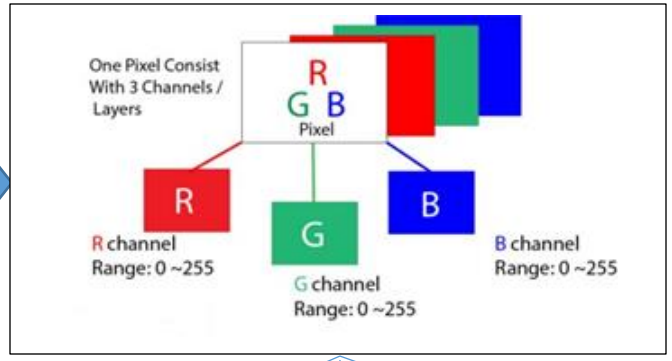
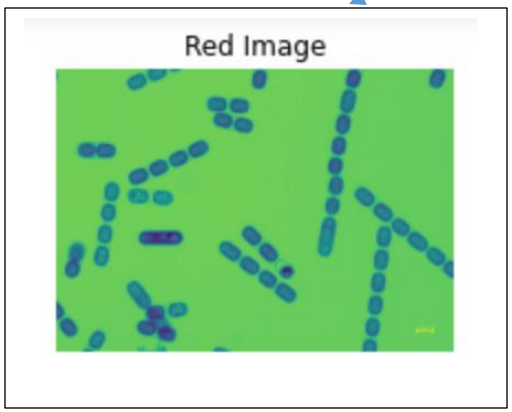
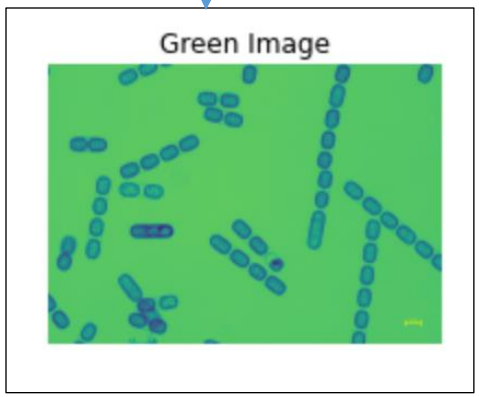
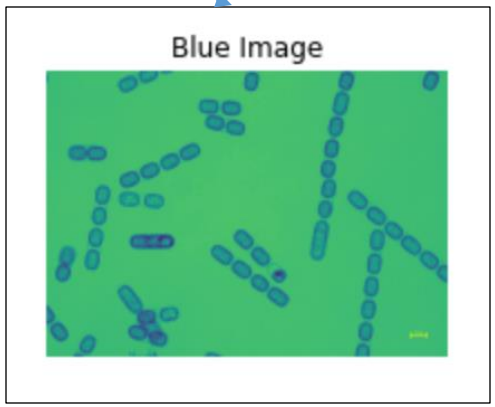
ORIGINAL PICTURE









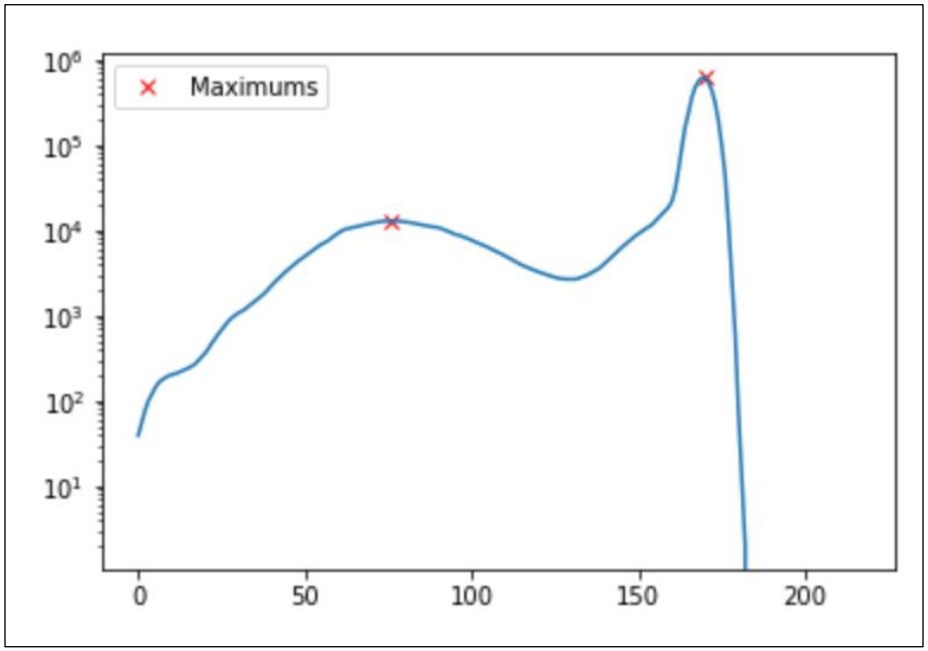


- I selected the filter in base at your contrast between the cell and the environment**.**
- With the selected filter **(RED)** I did representations with histogram and plotters to see the distribution of the intensity of the pixels in all the image.
- Calculated the median and means and plotter it.
- I got the peaks of the histogram and I used it to definite intervals around of the peaks, it represent intensity the pixels where there is probably to find cells.

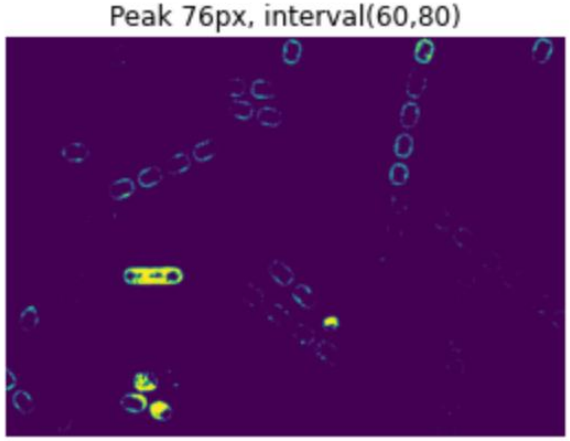
PEAKS

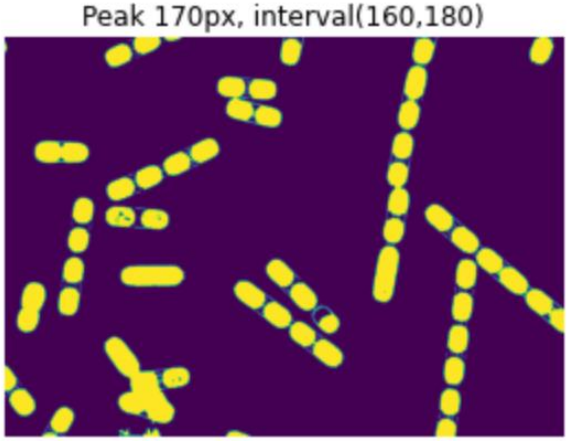
76 px

170 px



Segmentation





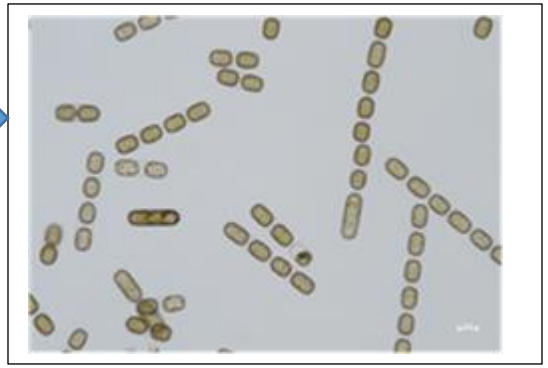
- **INSIDE OF INTERVAL---→THERE IS CELLS------→TRUE-→BOOLEAN VALUE-----→1**
- **OUTSIDE OF INTERVAL→THERE IS NOT CELLS→FALSE→BOOLEAN VALUE----→0**

DETECTION

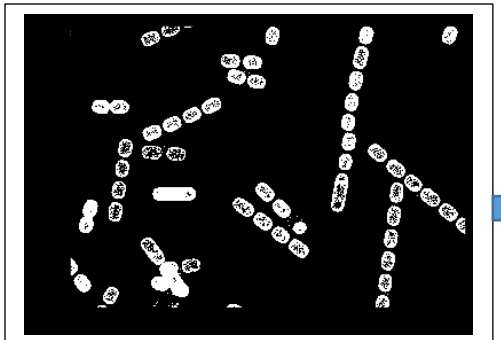
METHOD 1

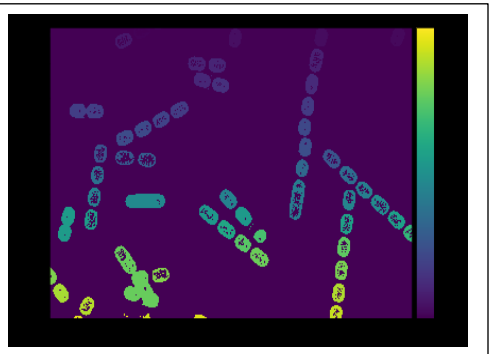
METHOD 2

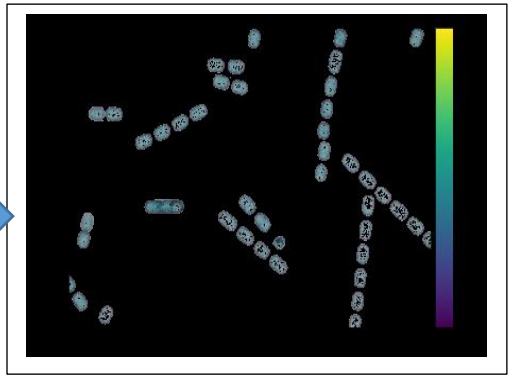
INPUT RGB SCALE



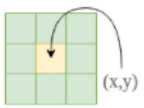
METHOD 1 ELIPSIS







SIMPLE SCALE (BLACK AND WHITE)

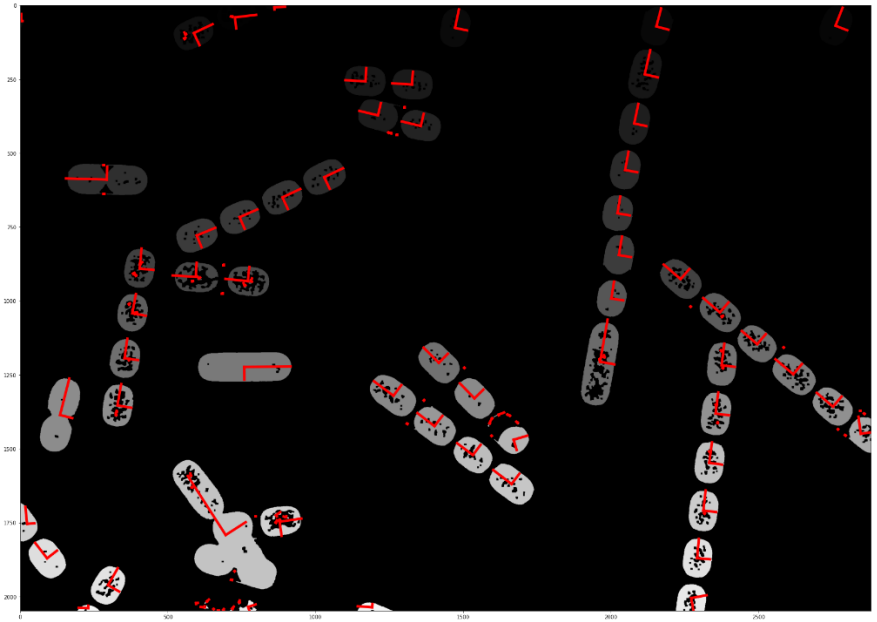


3X3 neighbourhood of (x,y)

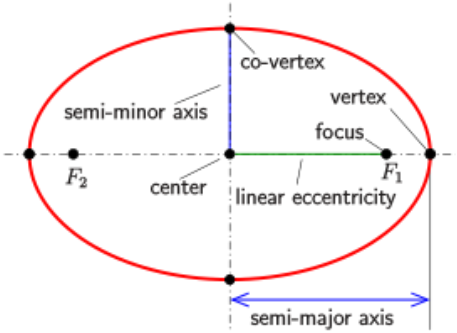
Labels the regions of the image depending on the connectivity of the pixels to each other. As long as neighboring pixels share the same value if intensity, they will be labeled as a single region, a region can be a cell.

After I got information of each region as: area, perimeter, major axis lengths, minor axis length, etc.

Define a CONDITION to determine which regions will be used: area and regions with a convex_area to area ratio that is approximately equal to 1 (to ensure that the object is indeed rectangular).

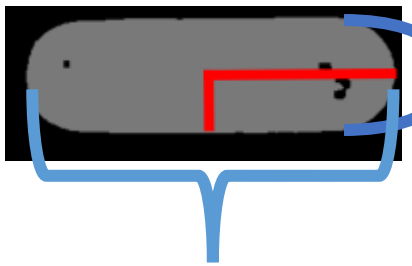


- EACH REGION IS EVALUATED IN BASE AT THE SHAPE OF AN ELIPSIS.
- GENERATED A DATAFRAME WITH PROPERTIES OF EACH REGION.



axis_minor_length: The length of the minor axis of the ellipse.

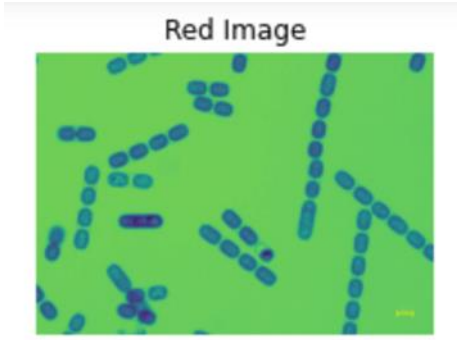
axis_major_length: The length of the major axis of the ellipse.

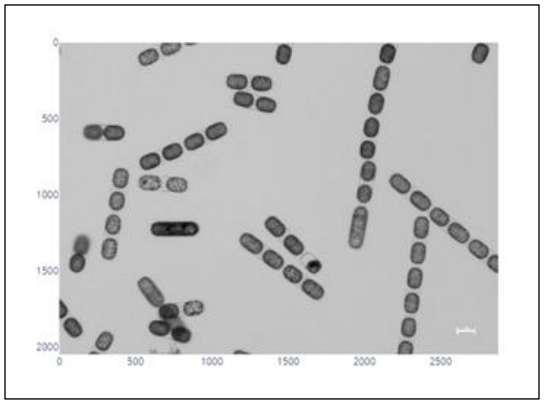


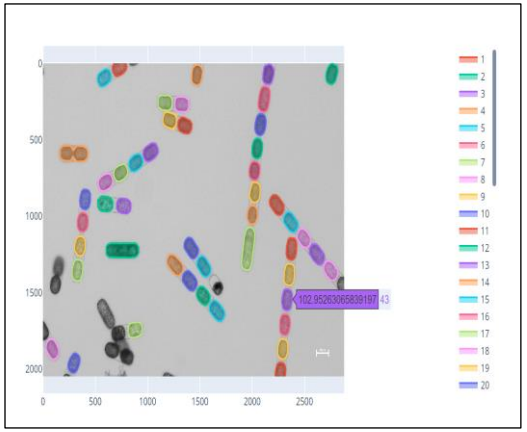
$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1.$$

$$(x, y) = (a\cos(t), b\sin(t)) \quad \text{for} \quad 0 \leq t \leq 2\pi.$$

METHOD 2 THRESHOLDING



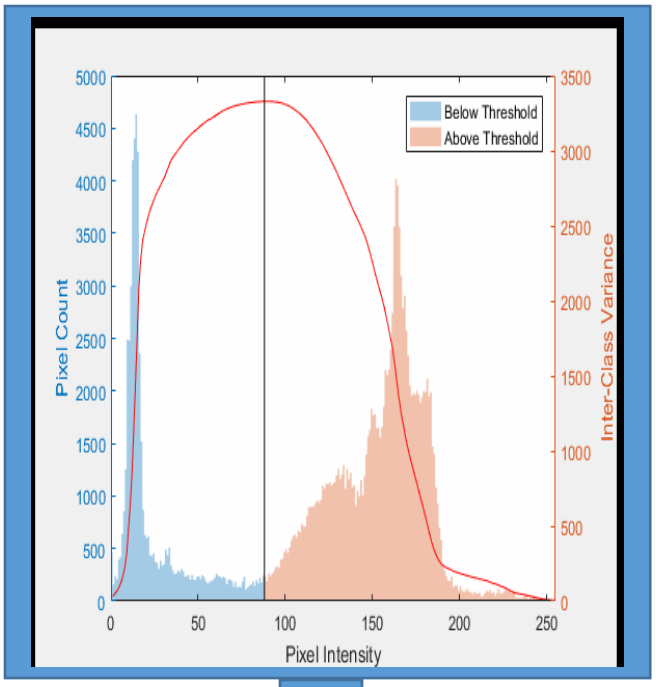




RGB SCALE → RED → GRAYSCALE →

Thresholding

Replace each pixel in an image with a black pixel if the image intensity is less than some fixed constant T- curvature or a white pixel if the image intensity is greater than that constant.

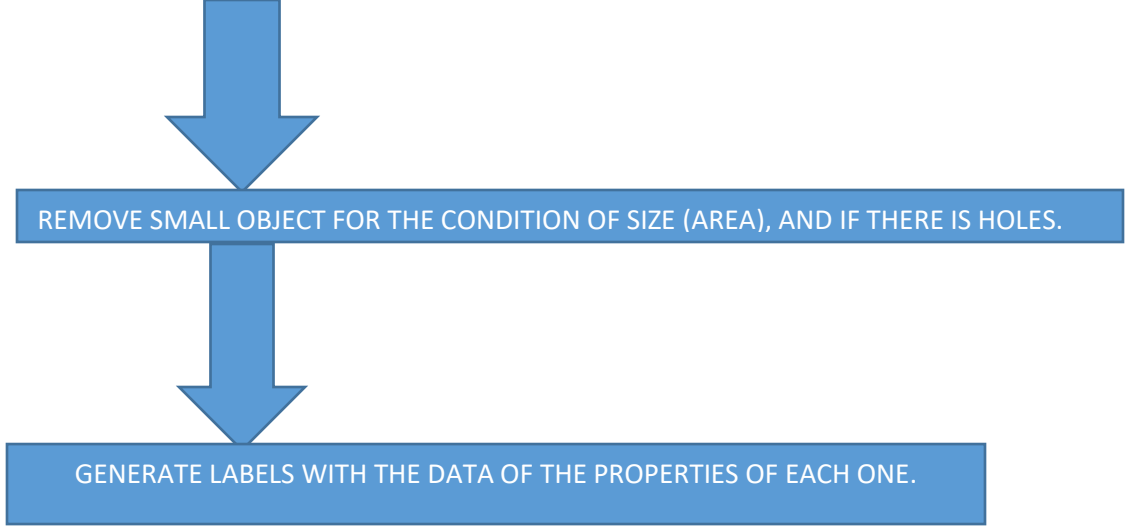


Returns a single intensity threshold that separate pixels into two classes, foreground and background.

Use the variance generated in the histogram of intensity of each pixel in the image.

Peaks, valleys and curvatures of the histogram are analyzed.

CREATE A BINARY IMAGE (WHITE AND BLACK)



The threshold value is calculated from the histogram of the image. The first and the second arguments are values corresponding to the minimum and maximum of the histogram. Out-of-range values will be placed to the first and the last bins, respectively. The third argument is the count of bins in the histogram.