

CMP\_SC 7650: Digital Image Processing

Homework 5: Detection and Segmentation of Individual Structures of Interest

By: Mikey Joyce

Due: 11/7/2023

#### Abstract:

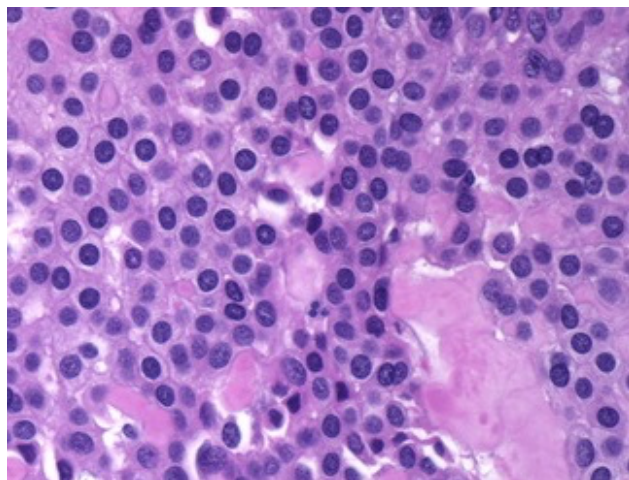
The problem presented in this assignment is to demonstrate using the watershed and connected component algorithms to identify individual structures and produce metrics on them such as total count and area.

#### Introduction:

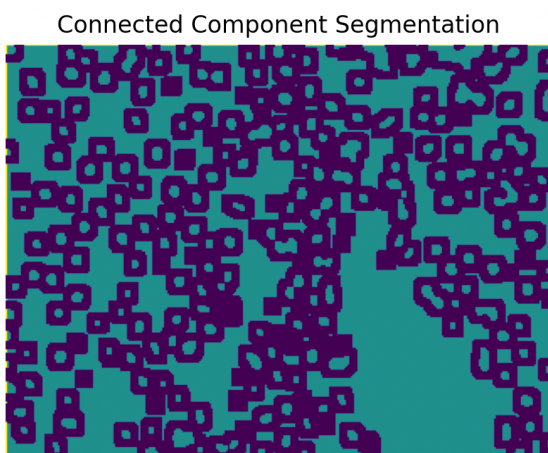
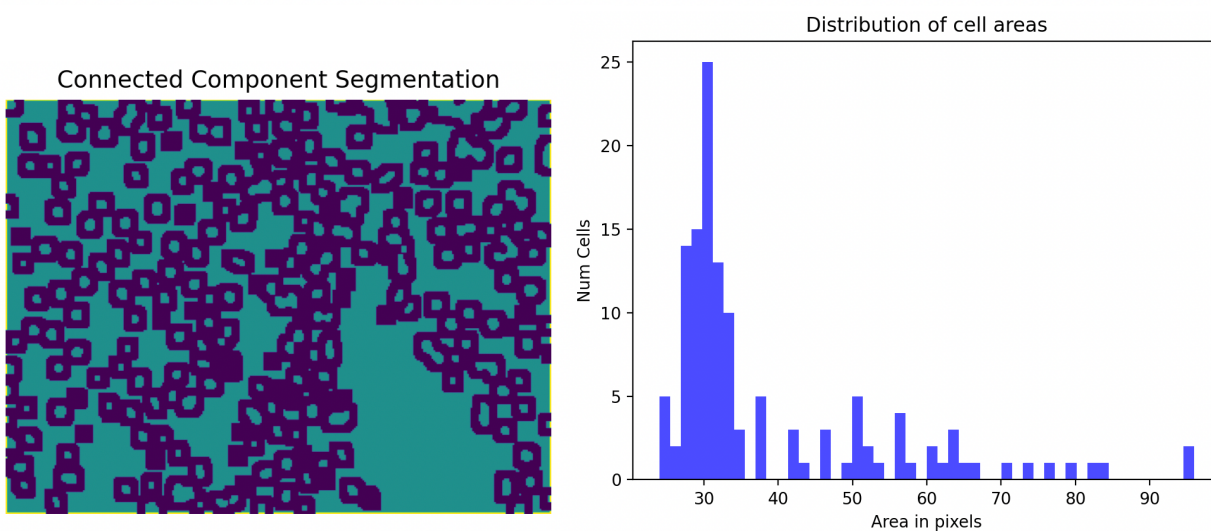
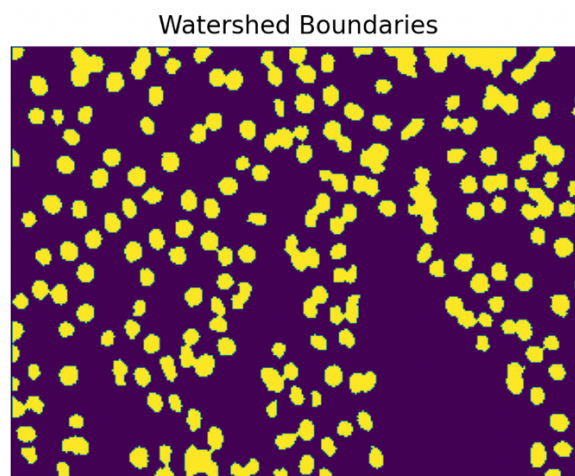
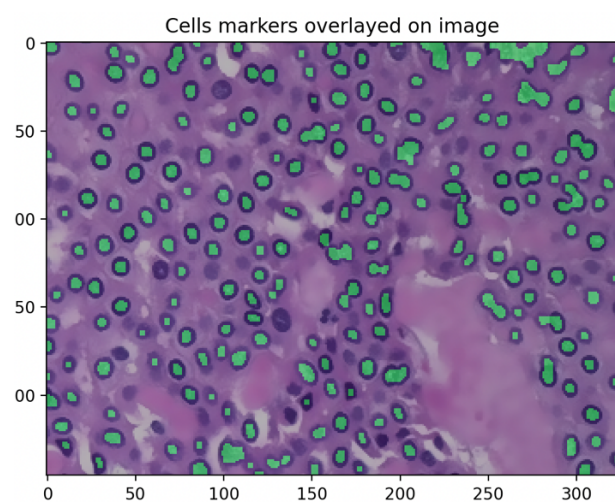
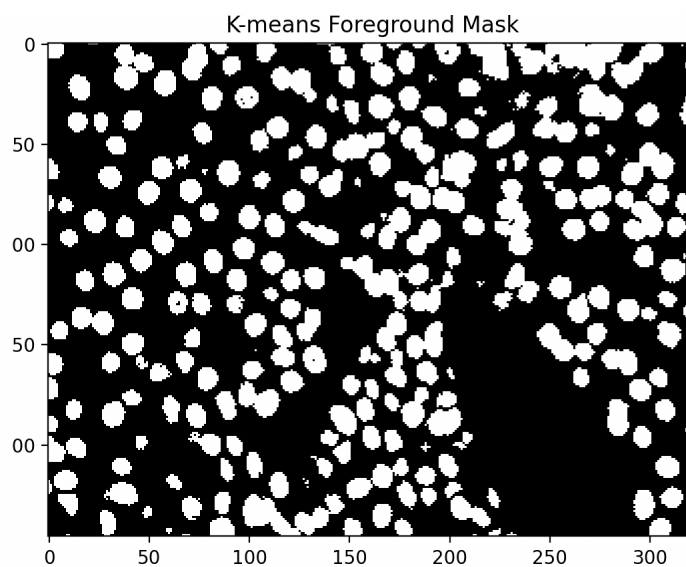
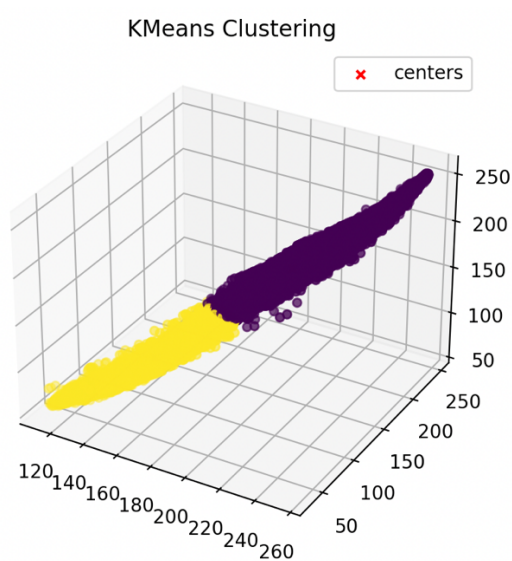
In this assignment only one test image was given. I will show all of the required plots for the given image. We will start with the original image, then use K-means to segment the cells from the image. After this we will use the distance transform function to obtain markers and overlay them over the original image. Watershed will then be applied, showing the output of the algorithm. Lastly, connected component labeling will be applied to obtain the total number of cells and the distribution of the cell areas.

#### Experiments and Results:

Original Image



There are 130 cells according to my watershed implementation.



## Conclusion:

In this conclusion we will discuss how the watershed algorithm and connected component labeling were implemented in this homework assignment. Initially, I read in the image and then apply the bilateral filter on it. I like this filter a lot because it preserves edges and cleans everything up, even though the input image didn't really need it too much, I still applied the bilateral filter. I then craft my feature vector for K-means the feature vector consists of (B, G, R) values. Then K-means is applied, this time when I implemented K-means I added in a loop around it and a line of code that essentially checks if I got the correct segmentation from K-means since the centers are initialized randomly. After this I shared the K-means plot and segmentation in the images above. Next thing I did was apply the opening operation by eroding and dilating the segmented image. I did this because there was some overlap of the cells, and I wanted a cleaner segmentation. Next, I applied my distance transform to get the markers. I got the background by performing a dilation and got the foreground by using the distance transform + a threshold operation followed with an opening operation. The resulting markers are then overlayed over the original image in the results section. My distance transform utilized Euclidean distance and a 5x5 neighborhood size. Then the threshold is obtained proportional to the maximum element in the resulting transform, although OTSU probably could've been used. After this I ensure that the foreground markers are of the correct data type (np.int32) and then apply the watershed algorithm to obtain the watershed boundaries. Lastly, I apply connected component labeling to obtain the segmentation and count the number of labels which means the number of cells. I made sure to subtract 1 to the number of labels because one of the labels is for the background. I was able to obtain the area by counting the number of pixels each marker had and then I plotted this distribution by plotting the histogram with 50 bins. I believe most of my results were satisfactory, K-means worked in a straightforward way and so did the watershed algorithm. I wasn't sure if the chart I produced for the connected component labeling is what you guys were going for, but the labeling still appeared to work correctly because I was able to count the number of cells and obtain the area of each cell.

## References:

- Libraries and tools: PyCharm, OpenCV, NumPy, Matplotlib, Sklearn, Statistics, Preview
- Lec12\_MathematicalMorphology.pdf
- Lec13\_WatershedSegmentation\_longer.pdf