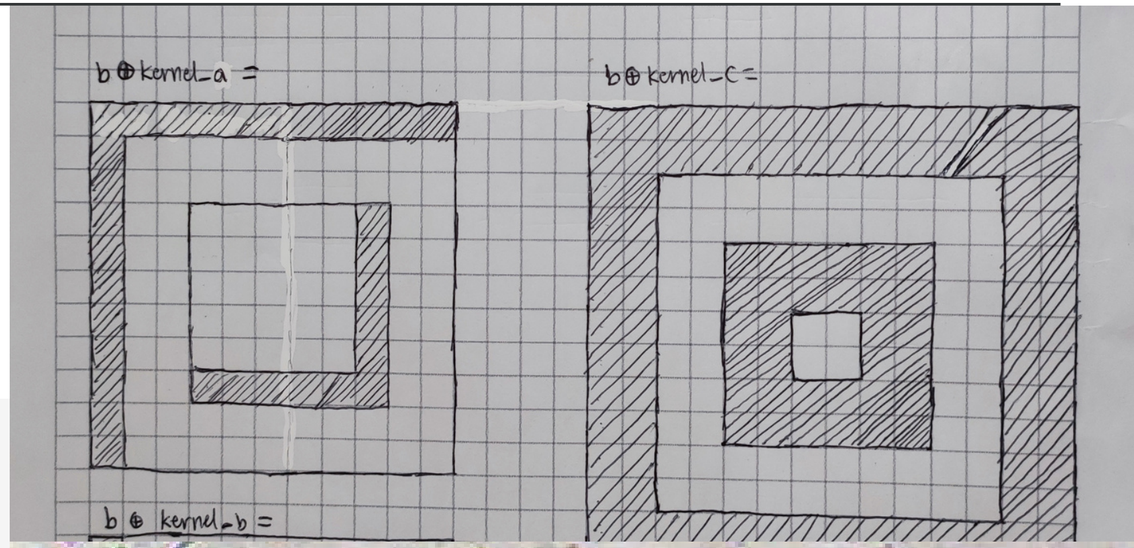
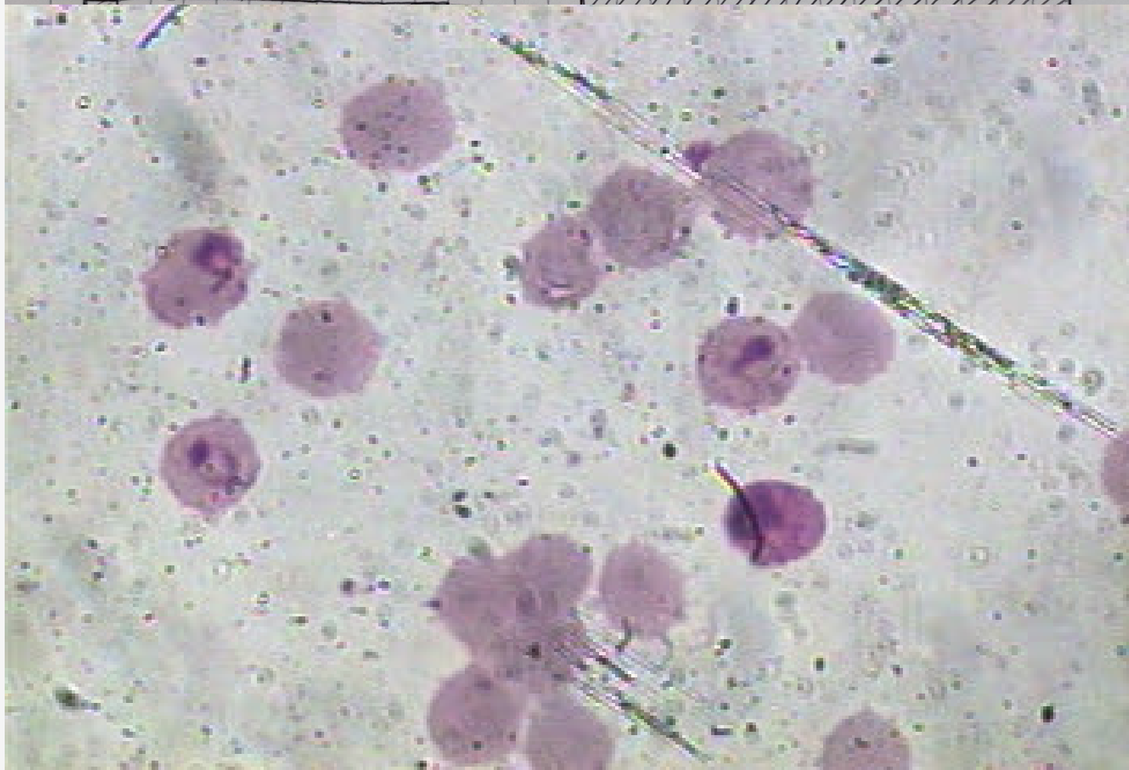
ACTIVITY 6. MORPHOLOGICAL OPERATIONS

Julian Christopher L. Maypa
2020-07587
App Physics 157 WFY-FX-1

Objectives

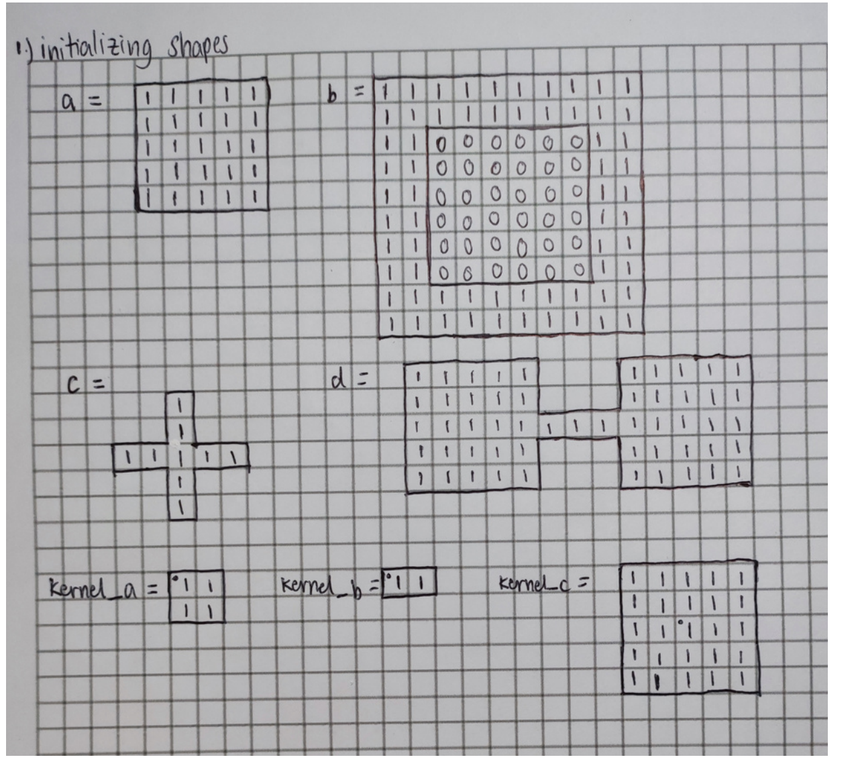
- Apply dilation and erosion on hand drawn shapes.
- Apply different morphological operations for blob detection.





6.1 Dilation and Erosion on hand-drawn shapes

I applied erosion and dilation on different shapes with different structuring elements and I followed the their respective dimensions that are stated in the laboratory manual [1]. The image to the right are my drawings of shapes named *a*, *b*, *c*, and *d*, and the structuring elements named *kernel_a*, *kernel_b*, and *kernel_c*.

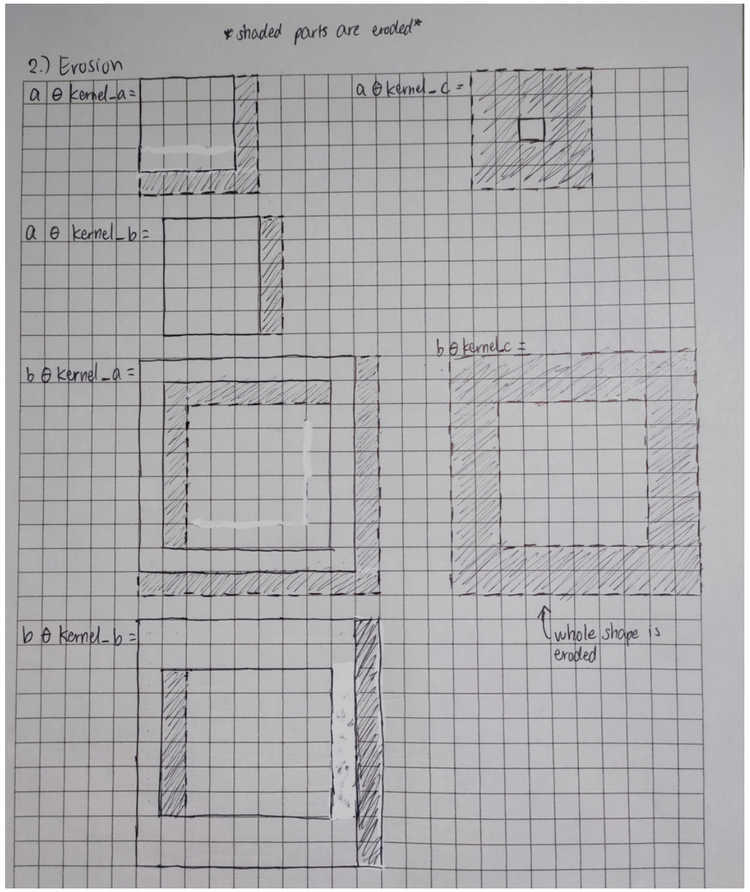


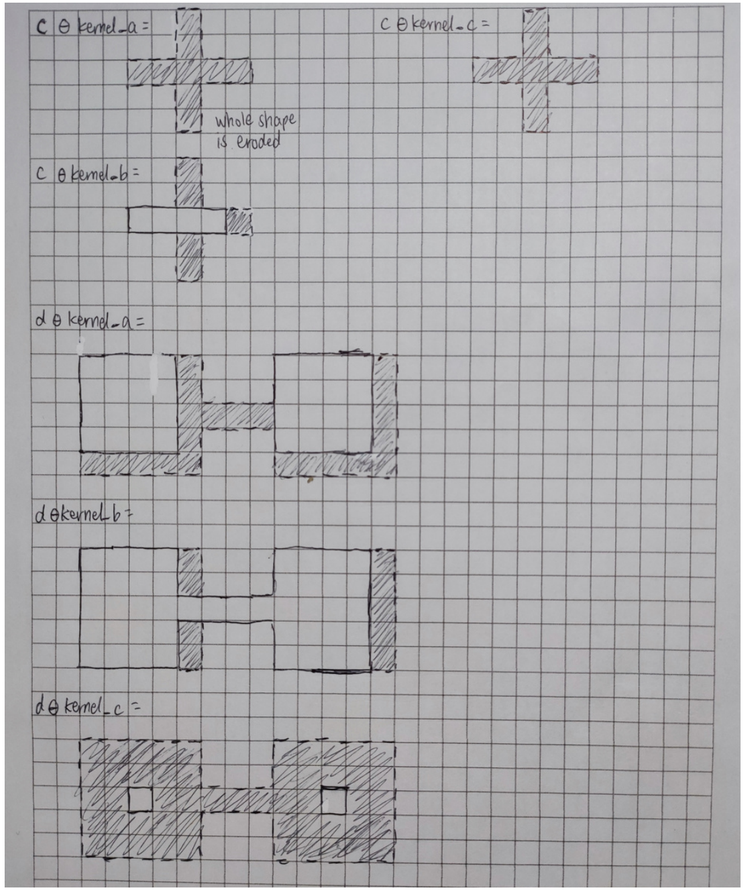
Erosion

I eroded all the shapes with each structuring element and my results can be seen on the right. The shaded region shows the part of the shape that was eroded, the solid lines are the parts of the shapes that are left after erosion, and the broken lines shows the original shape before erosion.

I eroded the shape by running the anchor of the structuring element across each pixel of the shape. If there is at least one pixel of the structuring element that does not overlap with the shape, the pixel of the shape that overlaps with the anchor is set to 0.

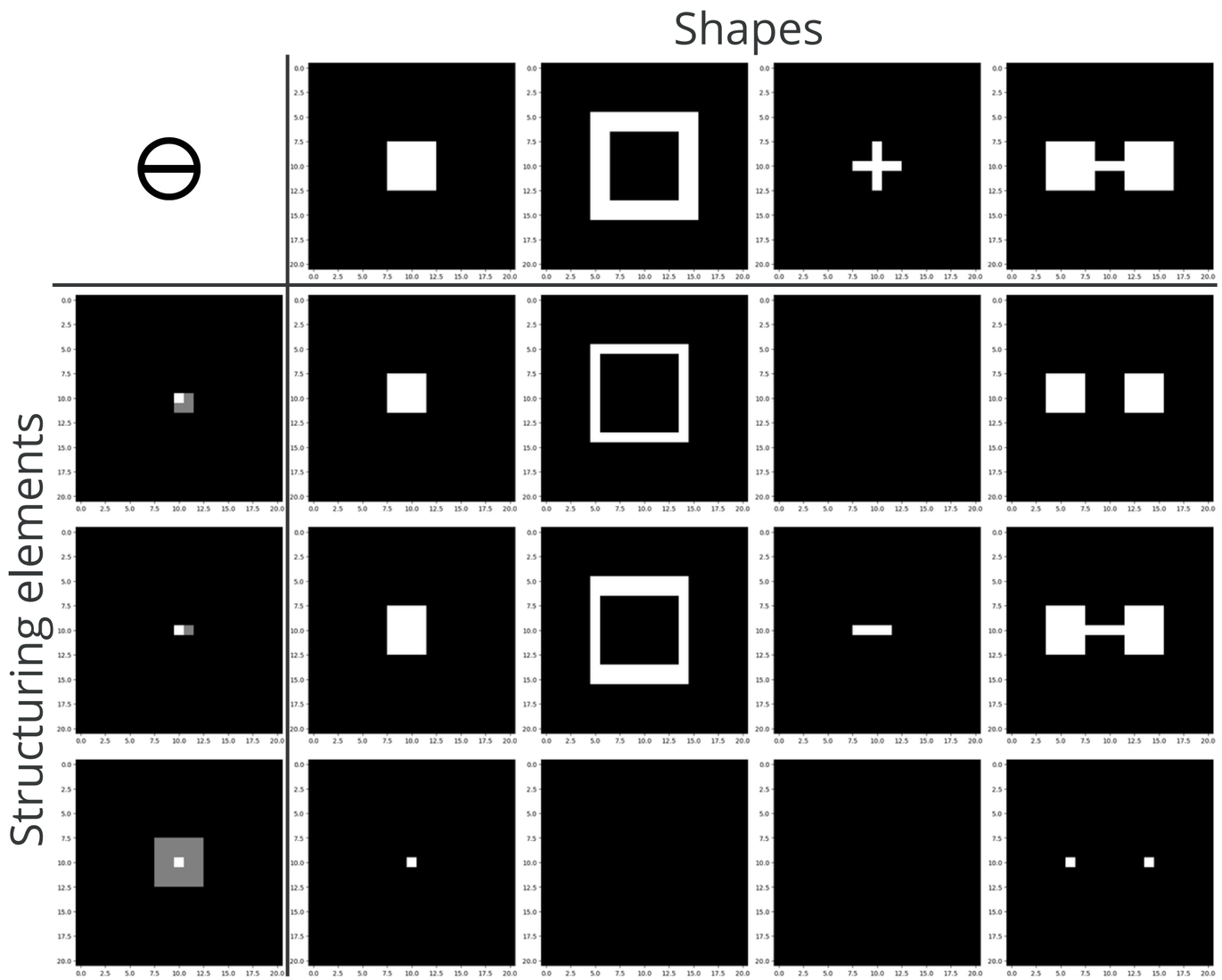
Looking at my results, it looks like erosion with *kernel_a* removes the bottom and right edges of the shape. *kernel_b* removes the right edges of the shape. While *kernel_c* removes 2 pixels from all sides of the shape.





I verified my eroded drawings by using Python to erode the shapes with the respective structuring elements. The bright pixel in the structuring elements is the anchor.

Comparing the output of my code to my hand drawn output, it can be seen that they are the same. Thus, my hand drawn results are correct and I have built up the intuition of using erosion on shapes with different structuring elements.

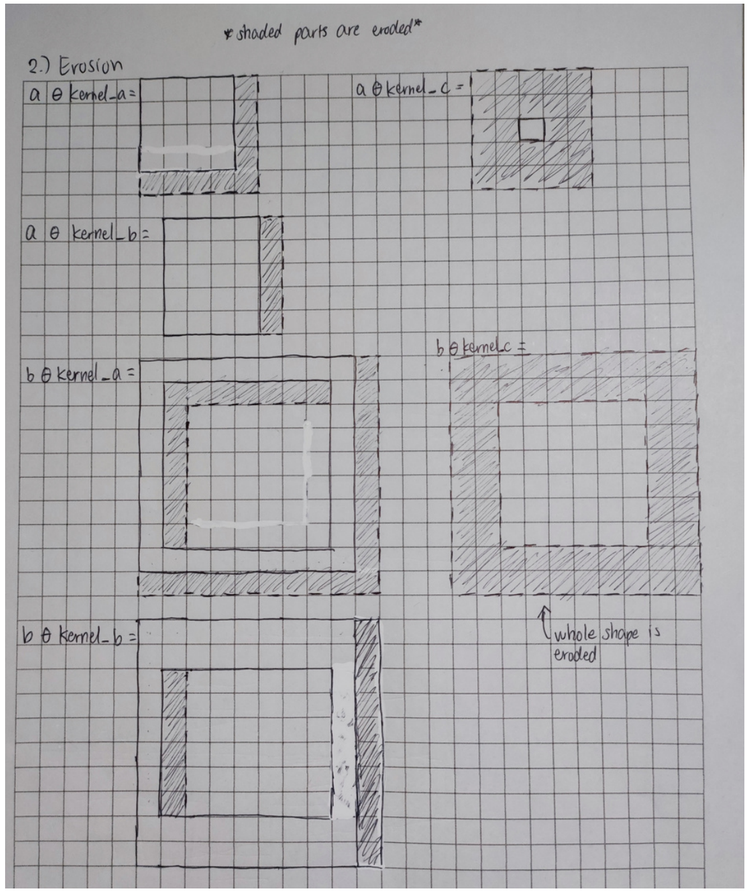


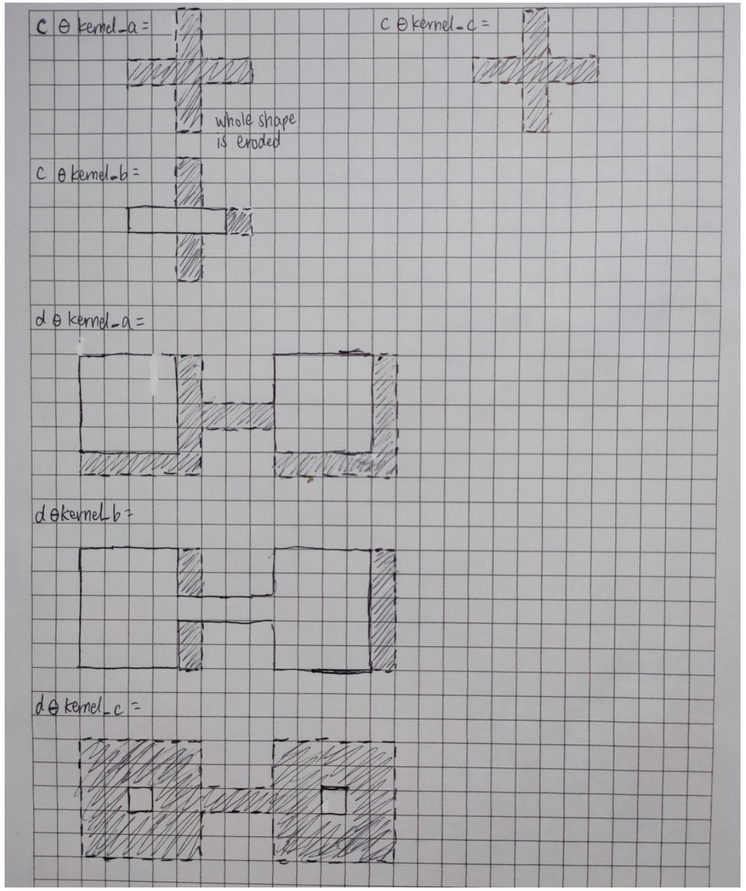
Dilation

I dilated all the shapes with each structuring element and my results can be seen on the right. The shaded region shows the new pixels added to the shape after dilation, and the solid lines show the original shape before dilation.

I dilated the picture by running the anchor of the structuring element across the pixels outside the shape. If there is at least one pixel of the structuring element that overlaps with the shape, the pixel that overlaps with the anchor is set to 1.

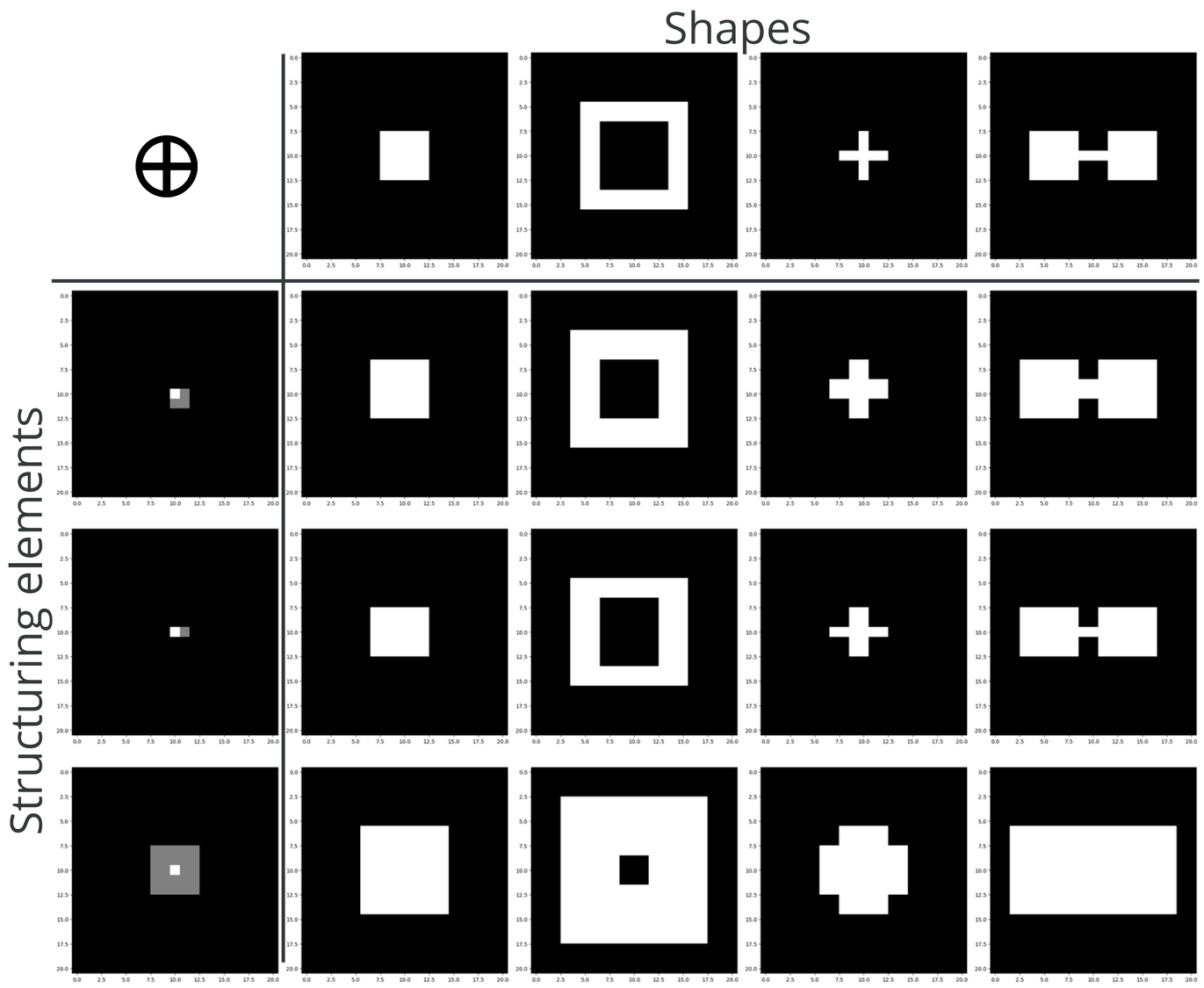
Looking at my results, it looks like dilation with *kernel_a* adds one pixel to the top and left edges of the shape. *kernel_b* adds one pixel to the left edges of the shape. While *kernel_c* adds 2 pixels to all sides of the shape.





I verified my dilated drawings by using Python to dilate the shapes with the respective structuring elements. The bright pixel in the structuring elements is the anchor.

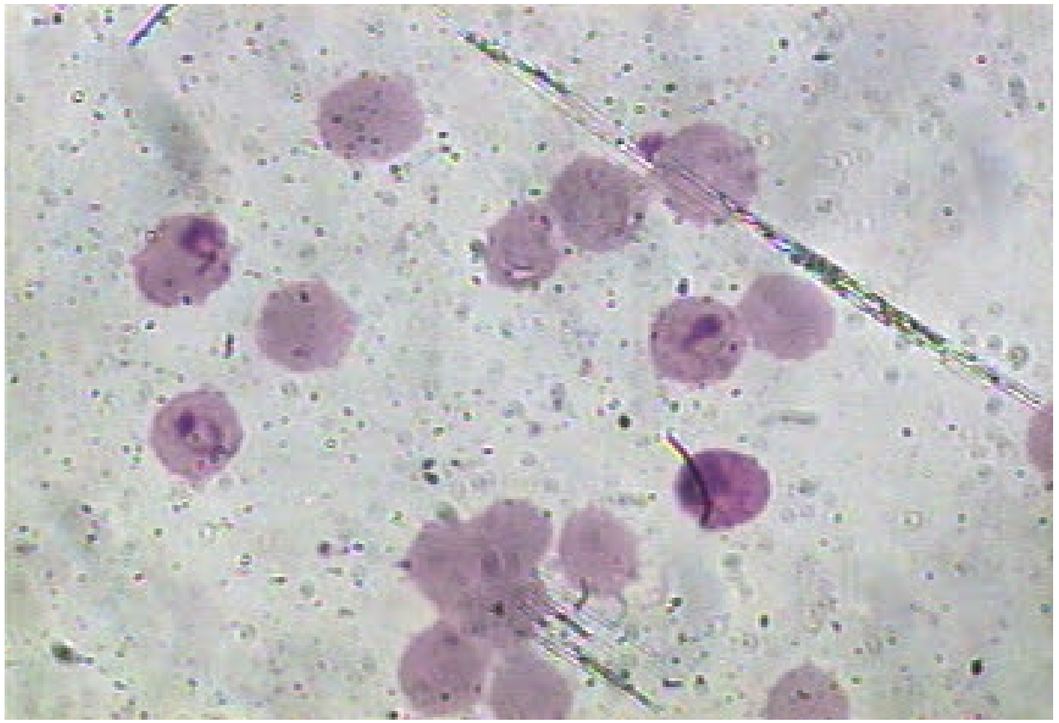
Comparing the output of my code to my hand drawn output, it can be seen that they are the same. Thus, my hand drawn results are correct and I have built up the intuition of using dilation on shapes with different structuring elements.

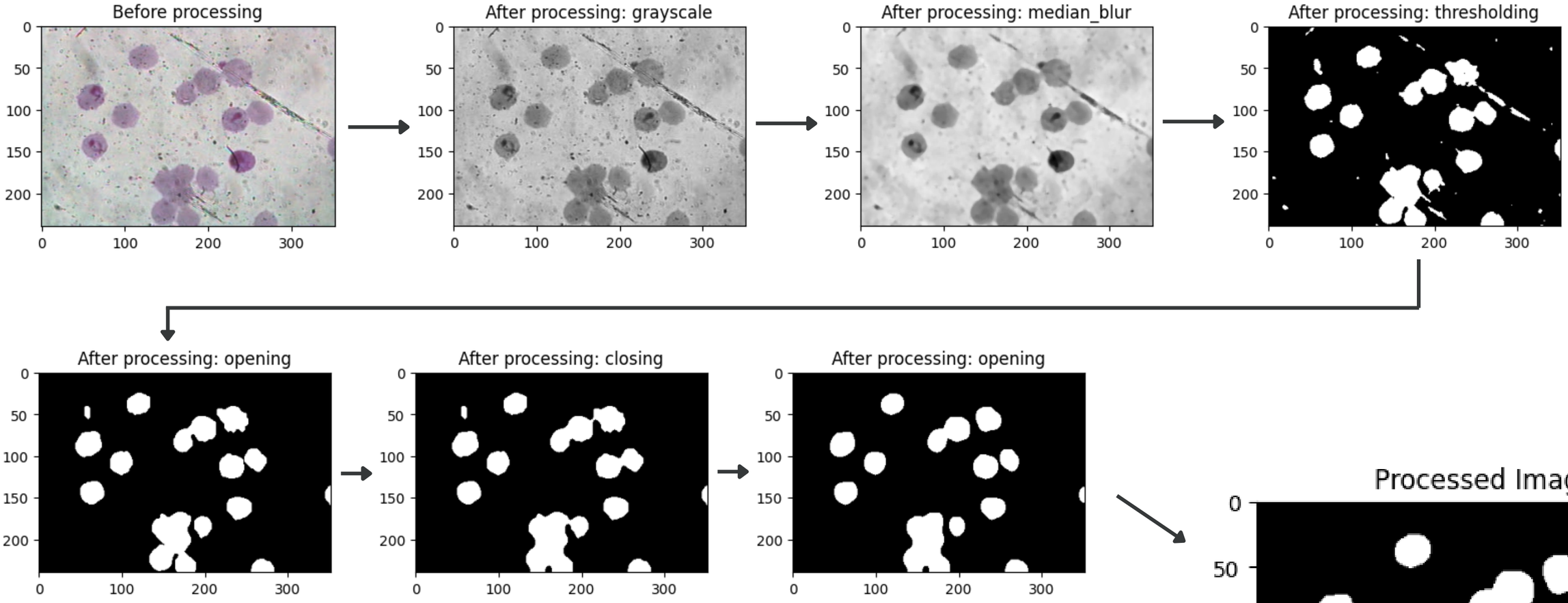


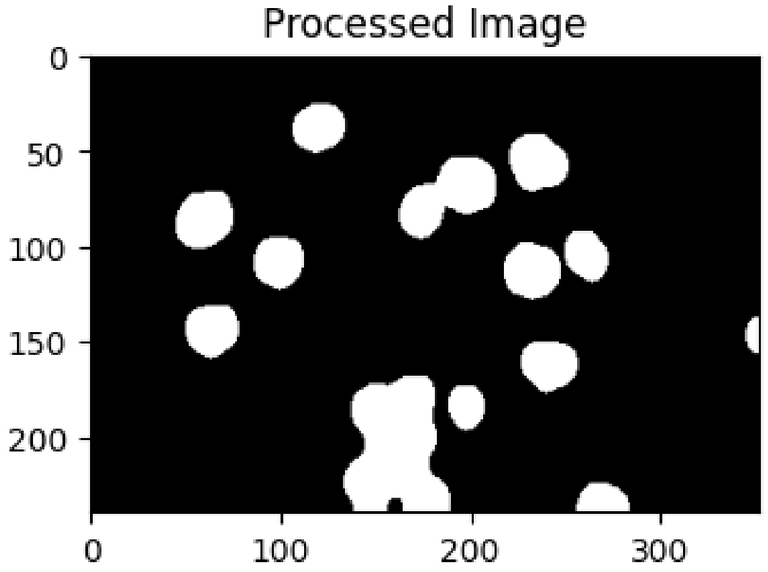
6.2 Application: Blob Detection

Now for the fun part, I applied multiple morphological operations to detect the malaria cells in the given picture. The picture was provided by my professor, Dr. Maricor Soriano.

The following slides shows my process in detecting the cells.



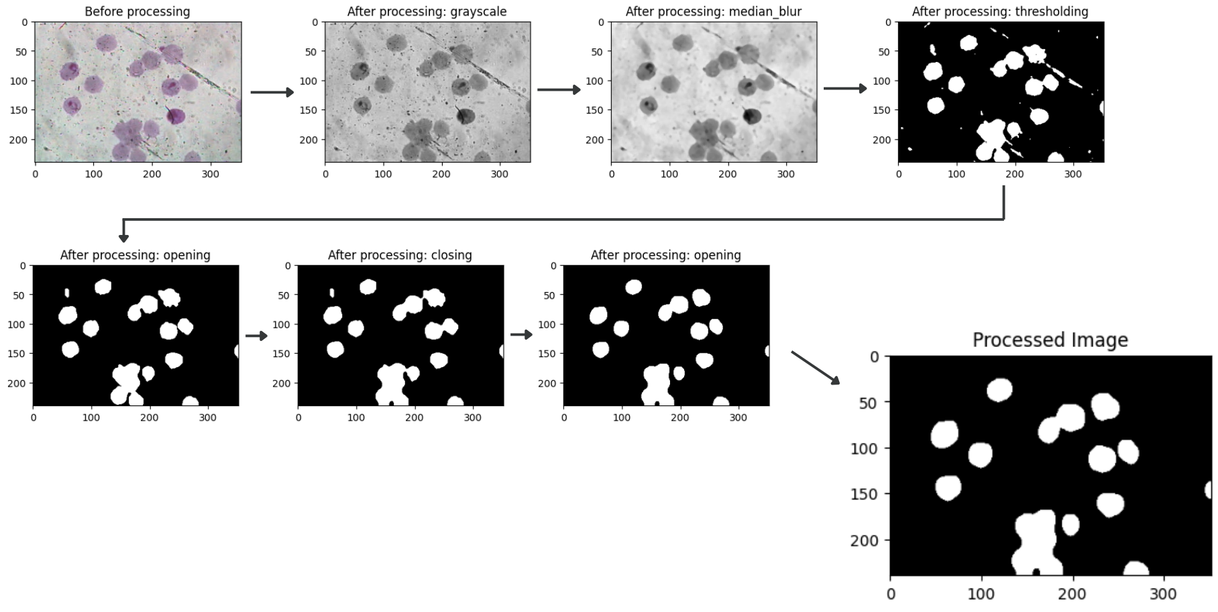




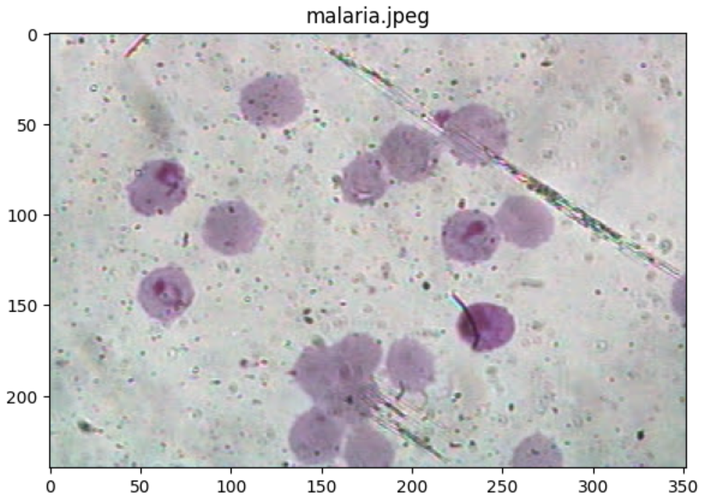
These are the morphological operations that I used to get the final processed image.

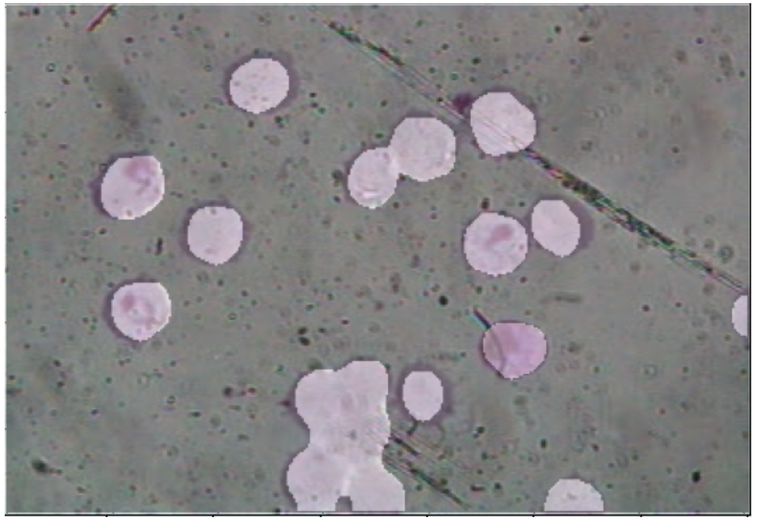
To be more specific with my steps, here is what I did:

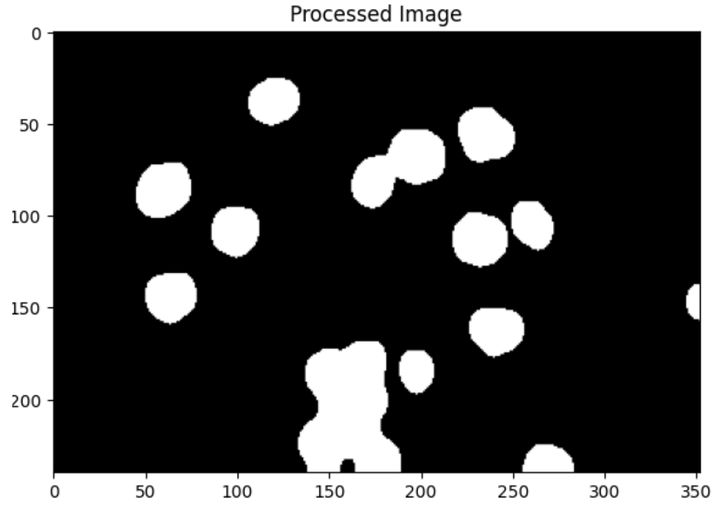
- I first converted the image to grayscale
- Then applied median filtering by using OpenCV's *median_blur* function. The kernel that I used was a 5 by 5 square. I used this filter to blur out the background noise while maintaining the edges of the malaria cells [2].
- I applied thresholding. I did this by getting the median and standard deviation of the pixel values. All the pixel values that fall within the range [median - 1.65*(stdev), median + 1.65*(stdev)] are set to zero while the other pixel values are set to 1.
- I applied the opening morphological operation with a 5 by 10 ellipse as the structuring element. The anchor of the structuring element is at the center of the matrix.
- I applied the closing morphological operation with a 7 by 13 ellipse as the structuring element. The anchor of the structuring element is at the center of the matrix.
- I applied the opening morphological operation with a 15 by 20 ellipse as the structuring element. The anchor of the structuring element is at the center of the matrix.



Below is the comparison of the original image and the processed image. It can be visually observed that the processed image overlaps with the original image quite well. Although the detected blobs are a bit smaller, they still have the general shape and location of the malaria cells. Thus I have successfully used morphological operations to detect the malaria cells.







Reflection

Overall, I believe that all of my results are correct. This is because my code made sense and they are the expected results. Although the detection of the malaria cells is not perfect, I have still successfully detected them using morphological operations.

The most tedious part for me is the part where I had to make the hand drawn sketches of dilation and erosion. Although it gave me an insight for erosion and dilation, I still found it very tedious. But still, I believe it was very necessary to do and it was very helpful. Other than that, the morphological operations were easy to do because of OpenCV. It was just finding the correct combinations of operations that was hard.

I'd like to thank my instructors, Sir Rene Principe Jr. and Sir Kenneth Leo, for guiding me throughout the activity. I would also like to thank my professor, Ma'am Jing, for guiding me in my coding while my classmates and I worked in R202. I would also like to acknowledge my classmates: Abdel, Johnenn, Jonabel, Richmond, Lovely, Hans, Genesis, Jeruine, Rusher, and Ron for helping me complete this activity.

Self Grade

Technical Correctness	I understood the lesson and met all the objectives. My results are complete and I got the expected results.	35
Quality of Presentation	The images I added to this report are of good quality and all the graphs are properly labelled. My code is also properly organized and labelled.	35
Self Reflection	I got the expected results, and acknowledged the contributions of my peers while doing this activity. I also properly cited online references. But I submitted my report late. So I'll take deductions from this.	10
Initiative	Apart from doing the required tasks, I also helped my classmates with their code and helped them by cross-referencing my results with theirs.	10
	Total	90

References

[1] Applied Physics 157 Laboratory Manual. A6 - Morphological Operations

[2] Sekhon, M. (2019, August 11). Image filters in Python. Medium. https://towardsdatascience.com/image-filters-in-python-26ee938e57d2