Morphological Operations

Tolledo, Charmaine S.

OVERVIEW OF THE ACTIVITY

Morphological operations are fundamental techniques used in image processing to analyze and manipulate the shape and structure of objects within images. [1] These operations, based on mathematical set theory, involve erosion and dilation applied to binary or grayscale images using a structuring element. These operations can be combined to execute operations such as opening and closing. Understanding and applying morphological operations is essential for a variety of image processing tasks, such as noise removal, object extraction, and image enhancement. [2]

In this activity, we'll delve into the practical applications of morphological operations in image processing!

OBJECTIVES

- Predict the morphological operations for four geometric objects by drawing them.
- Compare the hand-drawn morphological operations with the operations implemented in MATLAB.
- Apply morphological operations in biological sample.
- Explore other application of morphological operations

ORIGINAL IMAGES/OBJECTS USED

IMAGE 1

IMAGE 2

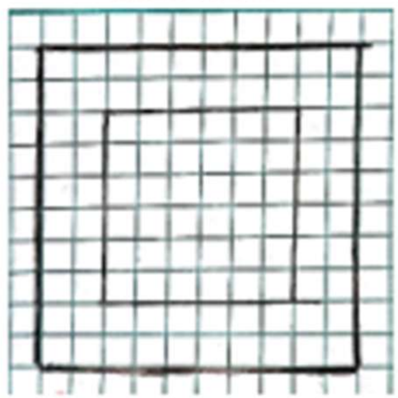


IMAGE 3

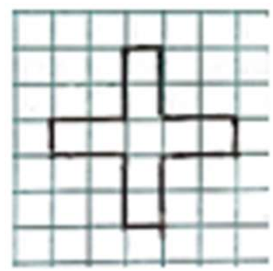
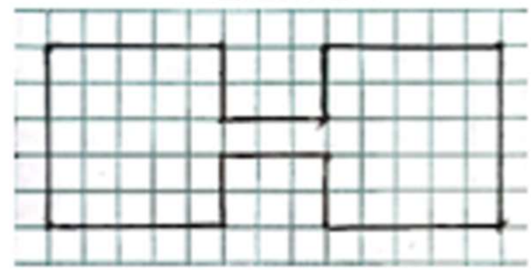
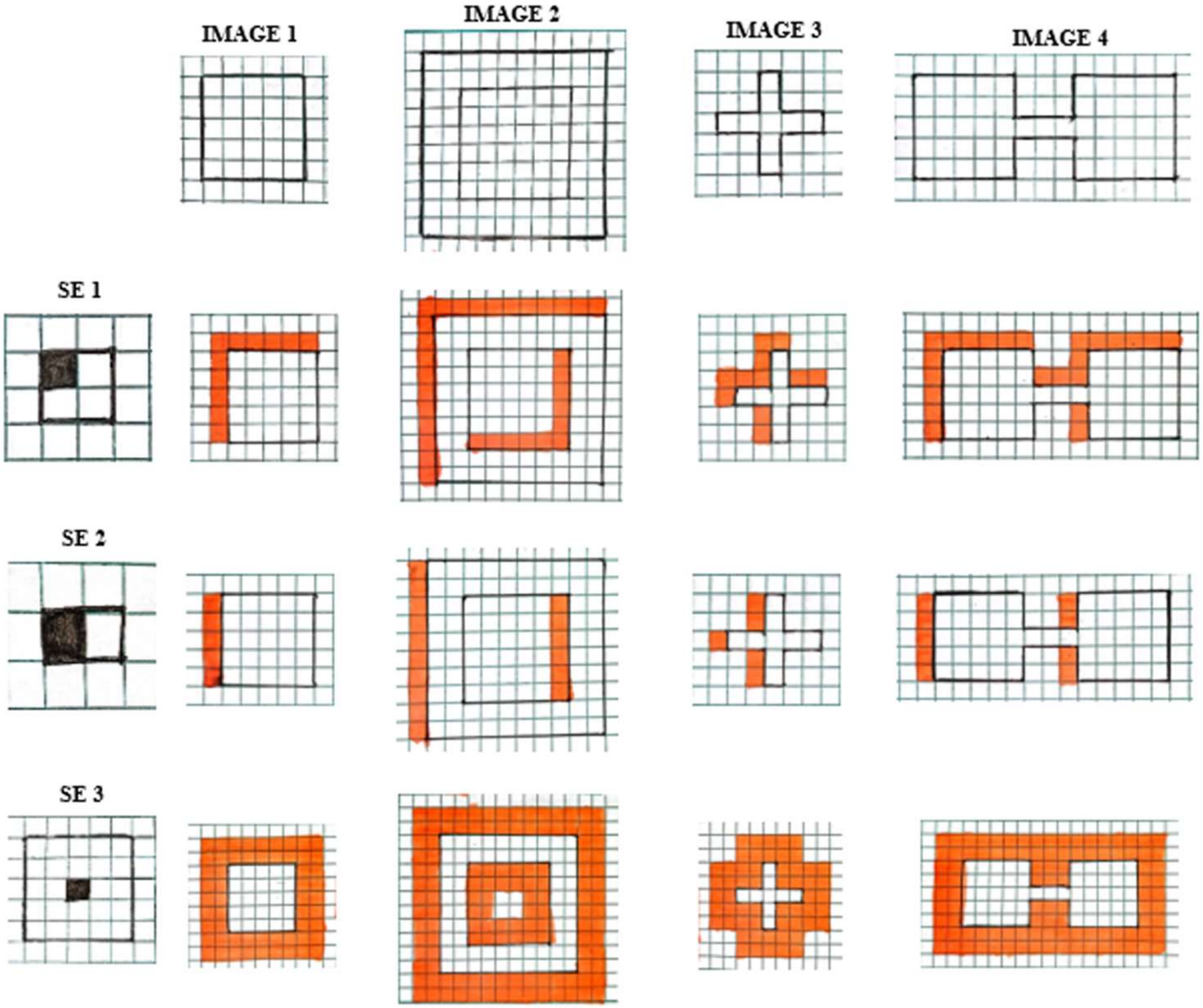


IMAGE 4



These are the geometric objects used in part I. The first image is a 5x5 square, the second is a hollow 10x10 square with a thickness of two boxes, the third is a plus sign with a thickness of one box and five boxes along each line, and the fourth is a dumbbell shape consisting of two 5x5 squares connected by a 3x1 line. We’ll apply dilation and erosion to these diverse geometric objects and observe how the shapes evolved and transformed based on the defined structuring elements.

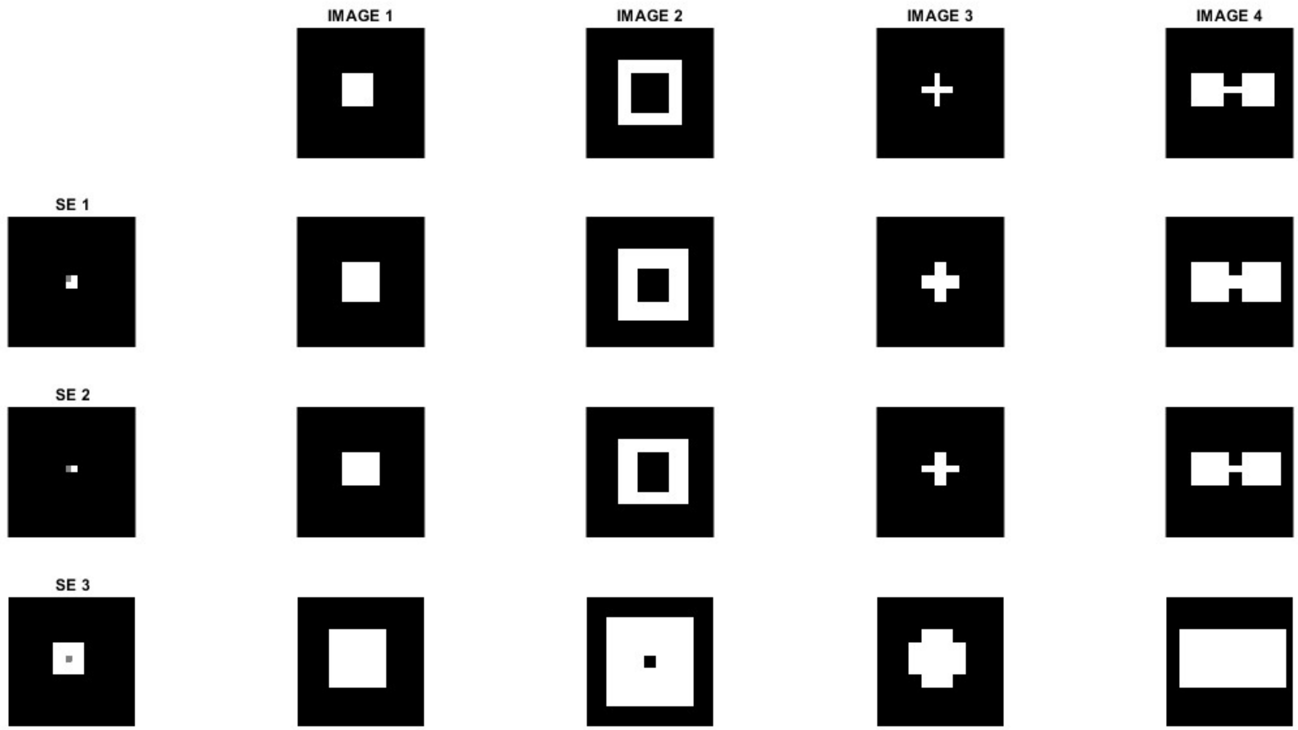
6.1 DILATION BY HAND



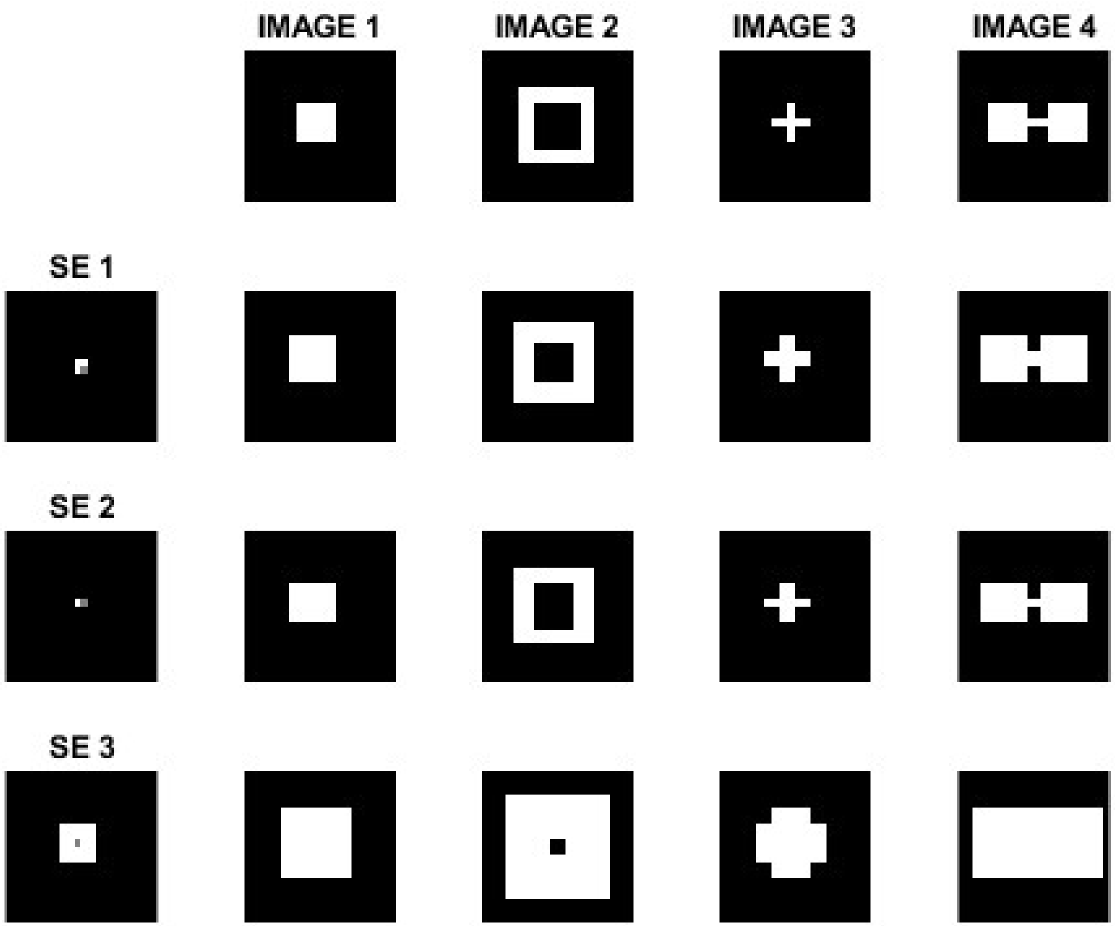
Here, we can see that we used three (3) structuring elements to apply morphological operations on the four geometric shapes. The shaded part represents what part of the structure element is taken as the origin. In my hand-drawn dilation, the red-orange color served as the added layer/pixel, which means that the resulted images in dilation expanded. For SE1, The added pixels were symmetrically placed around the original shape, leading to a uniform growth in size. Meanwhile for SE2, the dilation produced a similar expansion effect. However, since the structuring element was asymmetric, the added pixels were also asymmetrically placed. That's why the shapes of the dilated images were slightly distorted, with more added pixels on the left side. Lastly, for SE3, the dilation is symmetric, with added pixels distributed around the central region of the pattern.

DILATION BY CODING

This is the resulted images as we used MATLAB to perform the dilation operation on the 4 geometric shapes . Specifically, we used the imdilate function. Upon initial observation, we can see that the produced results closely resemble the hand-drawn images from the previous slide. However, if we look closely, we can see a subtle deviation, specifically, for SE1 and SE2, the added pixels were on the right side, which should be on the left. Thus, we can say that MATLAB does not reflect the structuring elements along the axis relative to their origin. To verify this hypothesis, we can proceed to the next slide, where we will incorporate the reflection of our SEs in the MATLAB code.

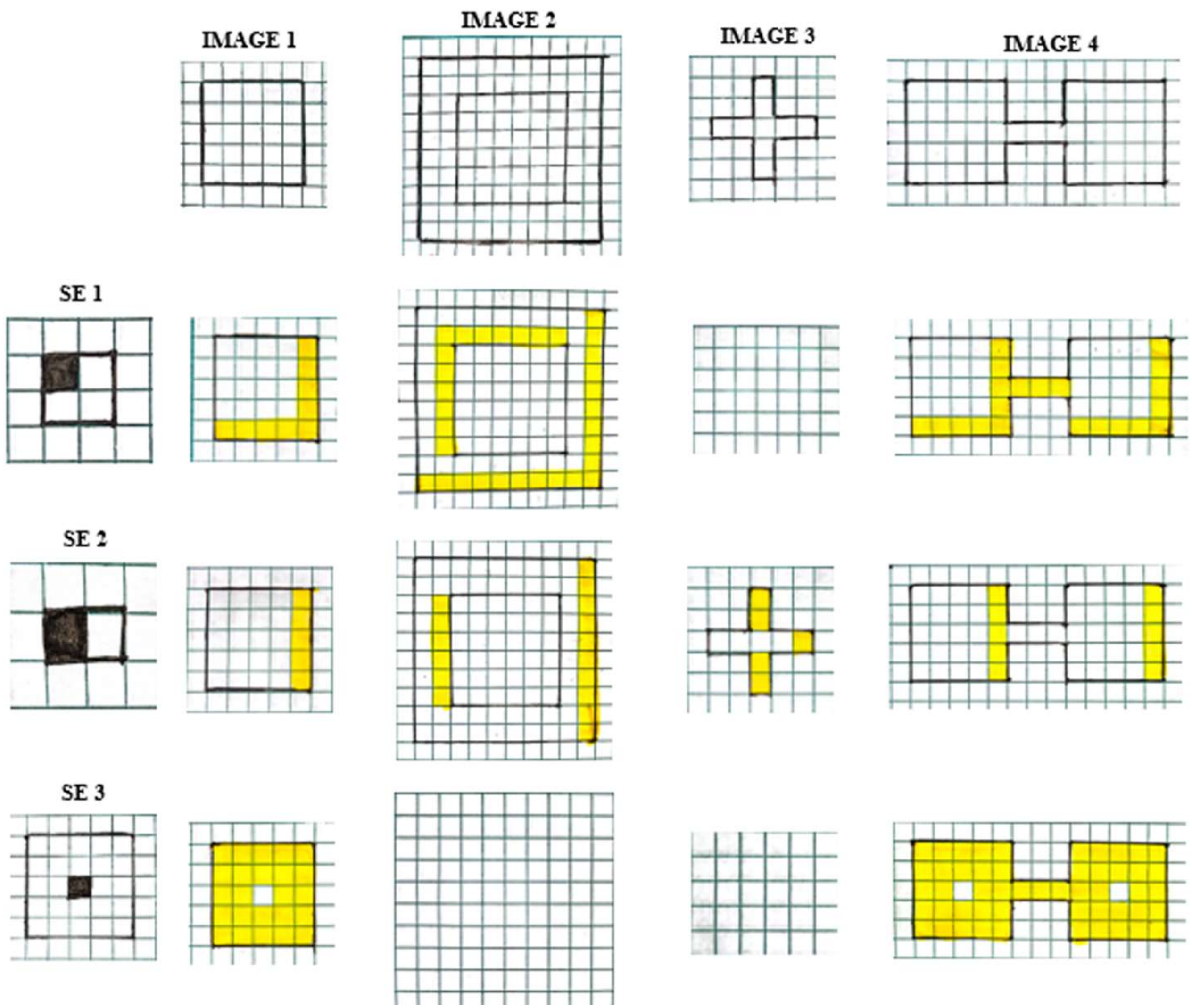


DILATION BY CODING WITH REFLECTED SE



Here, we can see that the pictures produced by the MATLAB function and the previous hand-drawn dilatation are now identical because to the incorporation of the reflected structural elements (SEs). With this, we can conclude that the imdilate() function in MATLAB does not initially reflect the image's structural element. To conclude, we found that during the process of dilation, the objects expand, leading to an overall enlargement of the image. This expansion was carried out by adding pixels or layers to the existing objects, based on the structural elements used.

EROSION BY HAND

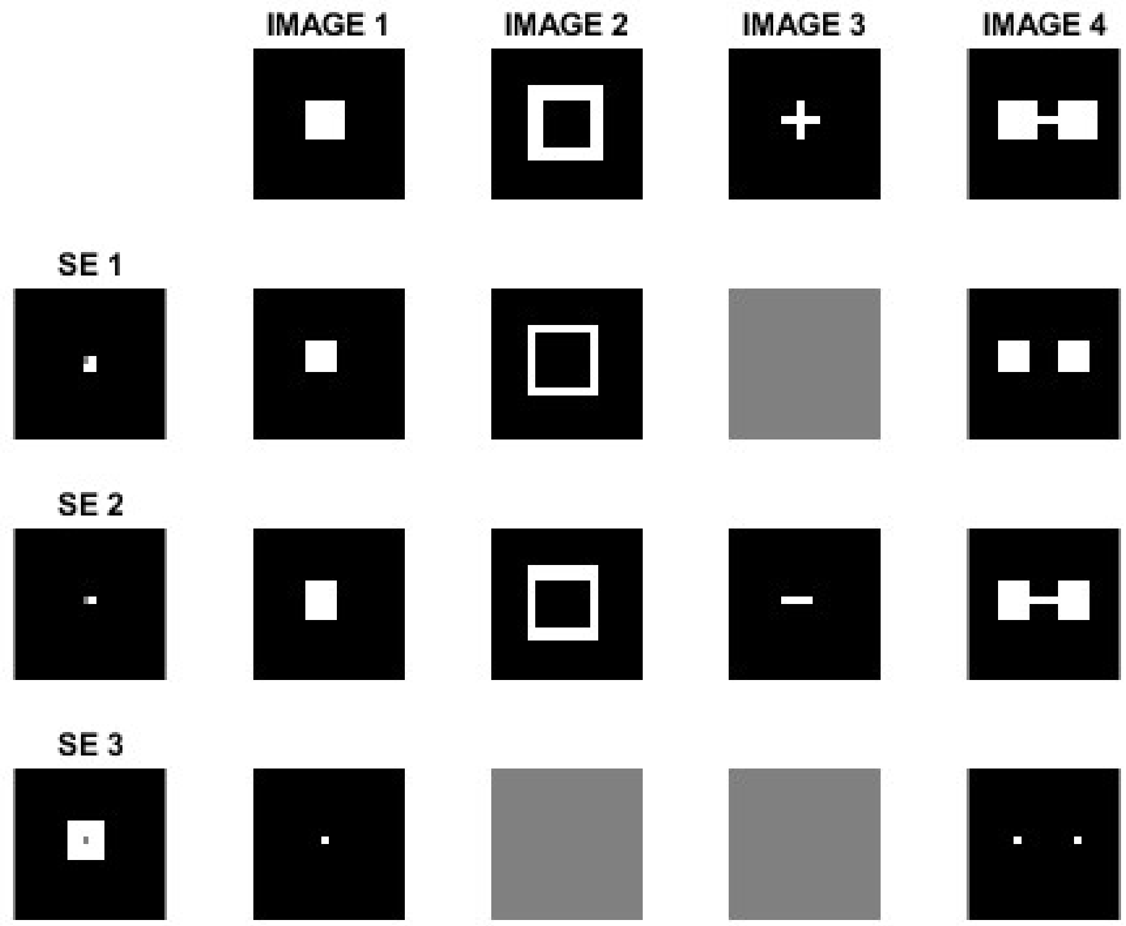


In my hand-drawn erosion, the yellow color served as the removed layer/pixel, which means that the resulted images in dilation shrunk or eroded. As SE1 and SE2 were asymmetric, the resulting eroded images showed shape distortion. More pixels were removed from one side than the other, resulting in an asymmetrical size reduction. Whereas for SE3, erosion occurred uniformly around the central region.

EROSION BY CODING

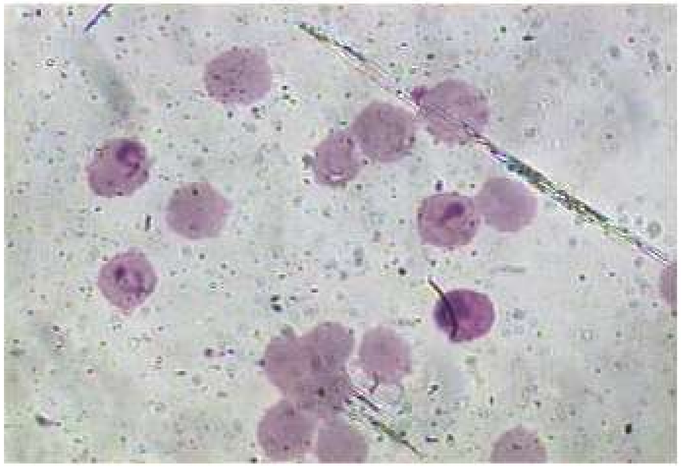
Here, we can see that the images obtained from the MATLAB-coded erosion are identical to the ones we have drawn. For SE1 and SE2, which are asymmetric structuring elements, the resulting eroded images showed slight distortions and more pronounced removal of pixels on the right side.

Also, when we dilate an image, we add pixels which causes them to get bigger. But when we then erode the enlarged image, the original places of the pixels are not fully recovered. Instead, some pixels are lost, and the form of the objects may be changed. This difference between dilation and erosion demonstrates that erosion does not perfectly reverse the effects of dilation. So, we can assume that erosion is not the exact opposite of dilation.

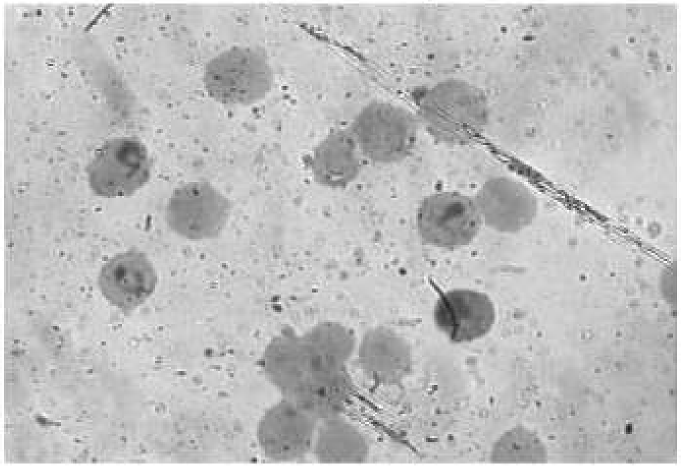


6.2 MORPHOLOGICAL OPERATIONS ON BIOLOGICAL SAMPLE

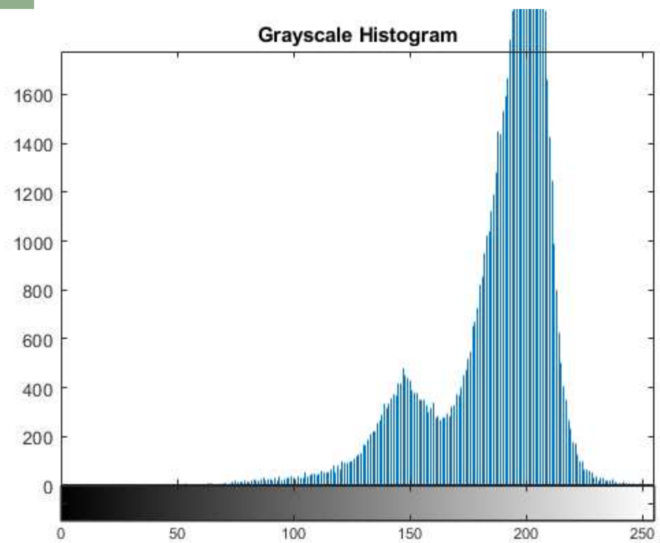
Original Image

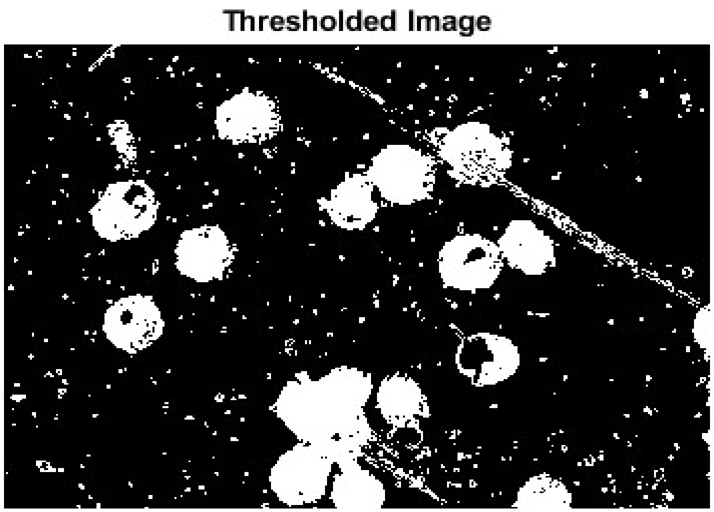


Grayscale Image



Here, we'll apply morphological operations on biological sample, specifically to images of the malaria sample. First, we converted the original image to grayscale. This is to enhance the visibility of important features and structures within the biological sample.

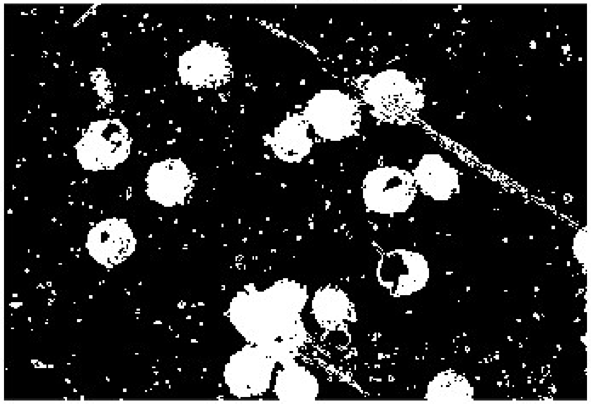




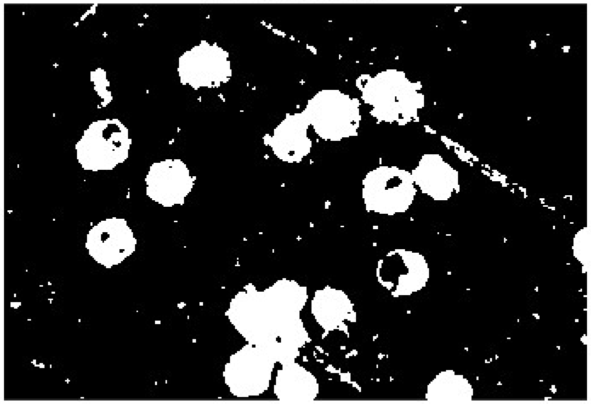
After converting the image to grayscale and analyzing its intensity distribution through a grayscale histogram (as seen in figure on the left), the next step was we applied a thresholding technique to create a binary image. In this case, a threshold value range was defined, specifically considering intensity values that are greater than 100 and less than 164. This is to segment and isolate regions of interest within the grayscale image based on their intensity values.

MORPHED WITH “MAJORITY” FUNCTION

Thresholded Image



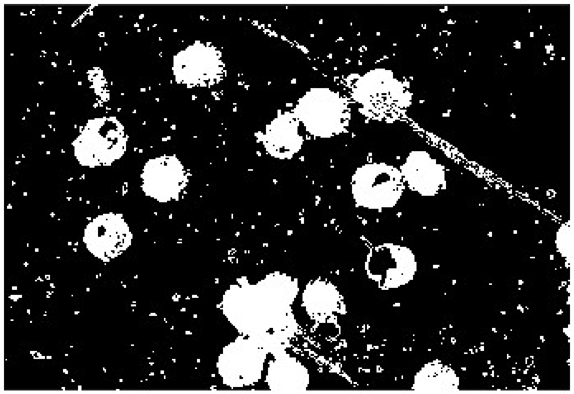
Morphed Image



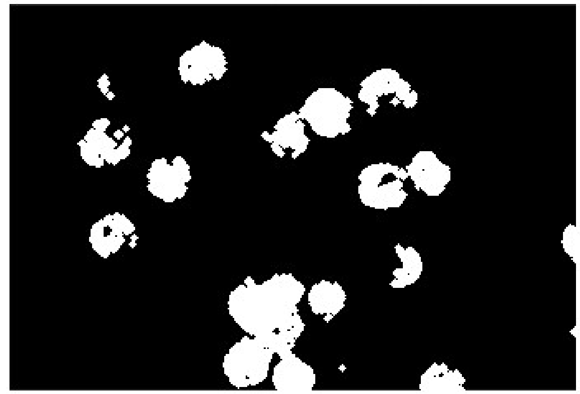
Using the "bwmorph" function in MATLAB, we applied the "majority" morphological operation to the thresholded image. This function determines whether a pixel should be set to white or black based on the presence of at least five white pixels in its immediate 3x3 vicinity. As seen in the morphed image, the majority of spots were effectively removed and the segmented image's outline was cleaned, resulting in a substantially clearer image. This result indicates that this method is highly effective for segmented images containing scattered white spots.

MORPHOLOGICAL OPENING

Thresholded Image

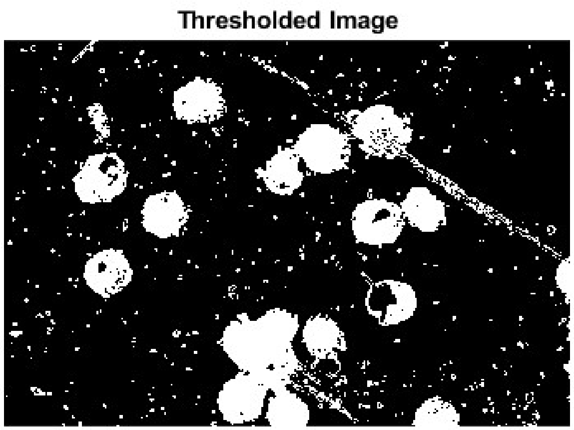


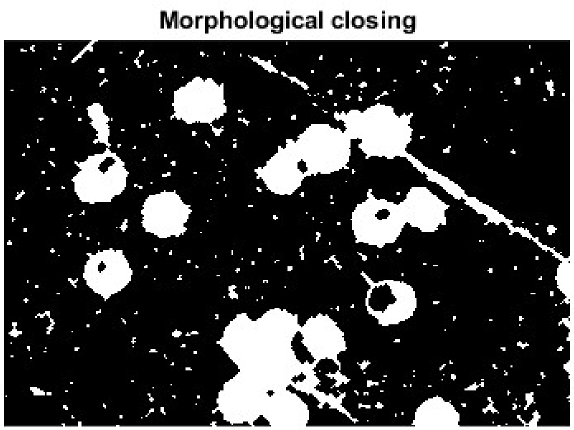
Morphological Opening



Here, we applied the concept of morphological opening in which involves performing an erosion followed by a dilation operation and a disk-shaped structuring element with a radius of 2 (This choice was guided by the principle of using a structuring element that aligns with the desired pixel shape for removal.) And as we can see, the morphological opening smoothed out irregularities and fine details present in the image and successfully eliminated extraneous pixel spots. It literally provided a cleaner background and really emphasized the more significant regions of interest.

MORPHOLOGICAL CLOSING





Here, we applied the concept of morphological closing which combines the dilation followed by the erosion process. On the thresholded image, the "imclose" function in MATLAB was applied using the same disk-shaped structuring element. As can be seen, it connected the structures and filled in the small gaps in the thresholded image. However, it has limitations. It was ineffective at filling large or extensive gaps in the image, as evidenced by the fact that significant holes were still visible in the final image despite the closing operation.

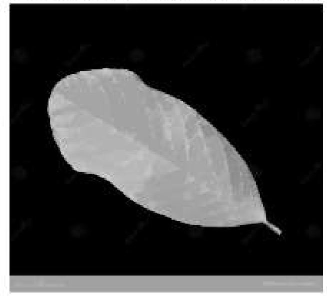
EXPLORATION OF MORPHOLOGICAL OPERATIONS

Now, for the exploration of morphological operations, we have here is an image of a leaf with a small hole. In this case, we'll apply morphological operations to fix some of the flaws in the image, such the tiny gap and hole in this leaf. The goal is to create a more accurate and aesthetically pleasing picture of the leaf by filling in or bridging the gaps using morphological closing and opening.

Leaf with hole

Grayscale leaf with hole

Leaf with Filled Hole using morphological closing



```
%morphological closing to fill small holes
SE = strel('disk', 10);
I_closed = imclose(I, SE);
I_clean = imfill(I_closed, 'holes');
```

Leaf with hole

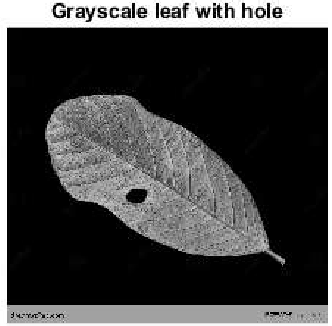
Grayscale leaf with hole

Leaf with Filled Hole using morphological opening

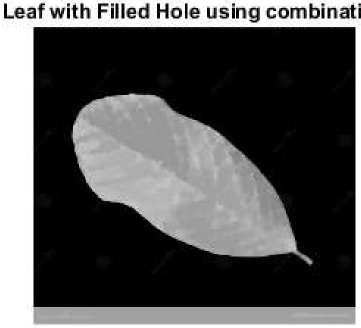
```
%morphological opening to fill small holes
SE = strel('disk', 10);
I_opened = imopen(I, SE);
I_clean = imfill(I_opened, 'holes');
```

Leaf with hole

Grayscale leaf with hole



Leaf with Filled Hole using combination



```
%morphological closing to fill small holes
SE_close = strel('disk', 10);
I_closed = imclose(I, SE_close);

% morphological opening to remove noise or small objects
SE_open = strel('disk', 5);
I_clean = imopen(I_closed, SE_open);
I_clean = imfill(I_clean, 'holes');
```

Morphological operations on filling leaf's hole

As seen in the previous slide, we used the opening and closing morphological operations to fill in a small hole in an image of a leaf. First, the picture is read and turned into grayscale. followed by an inversion to prepare it for further processing. As we can see from the code snippets, the structuring element, which is described as a circle with a certain radius, is a key part of how the process works. The resulting image is then improved by using the imfill function to make sure that the hole is completely filled in. This makes the leaf look better.

In the first image, where morphological closing is used, the resulting image appeared brighter than the original image. This brightening is due to the way the close process works, which includes dilation followed by erosion. The dilation step tends to make the brighter areas bigger, while the erosion step helps get the shape back to how it was before, though the brightness might change. Thus, the filling of the small hole leads to an overall brighter appearance in the resulting image.

In the second image, which uses morphological opening, the resulting image exhibits increased contrast compared to the original image. The opening operation, which involves erosion and then expansion, helps get rid of noise or small objects, making the contrast between the leaf structure and the background stronger. By removing some of the darker areas, like noise, while keeping the brighter parts, like the leaf, the resulting picture has a higher contrast, which makes the details of the leaf structure more pronounced.

In the third image, which combines morphological closing and opening, the resulting image appears even brighter compared to the closing operation alone. This is because closing, which tends to make things brighter, and opening, which gets rid of noise and boosts contrast, both have this affect. The closing operation fills the hole and makes the image brighter. The opening operation, which comes next, helps improve the image by getting rid of extra noise or unwanted structures. The combined effect results in an image that is brighter overall compared to the closing operation alone.

Despite the differences in brightness and contrast, all three processes successfully fill the small hole in the leaf image.

REFLECTION

Throughout the process of working on this activity, I initially thought it to be a relatively short task. However, I soon realized that creating the drawings themselves took a significant amount of time. Not only did I struggle with my drawing skills, but I also had to carefully consider the result of applying dilation or erosion to the given images using the provided structuring elements. This made the drawing process even more time-consuming. If the drawing itself took so long, I could only imagine the additional time and effort required for the coding part. Thankfully, MATLAB's built-in functions for morphological operations proved to be immensely helpful, simplifying the coding process and saving me time. Despite some moments of frustration due to a lack of grand ideas to apply these operations to, I still managed to explore morphological operations successfully. This sense of accomplishment, combined with the learning experience gained, makes me feel deserving of a perfect score: **100/100.**

REFERENCES

- P Soille. “Morphological Image Analysis, Principles and Applications”, 1999.
- https://towardsdatascience.com/understanding-morphological-image-processing-and-its-operations-7bcf1ed11756
- Activity 6 lab manual
- https://www.mathworks.com/help/images/ref/imfill.html
- https://epochabuse.com/hole-filling-in-image-processing/#:~:text=Hole%20filling%20or%20region%20filling,a%20border%20of%20foreground%20pixels.

End