

**Department of Artificial Intelligence**

**College of Computer Science and Information Technology**

1. **Objectives**
2. Understanding Morphological Image Processing and Its Operations

**Due Date: Tuesday September 24, 2024 @ 11:59 PM**

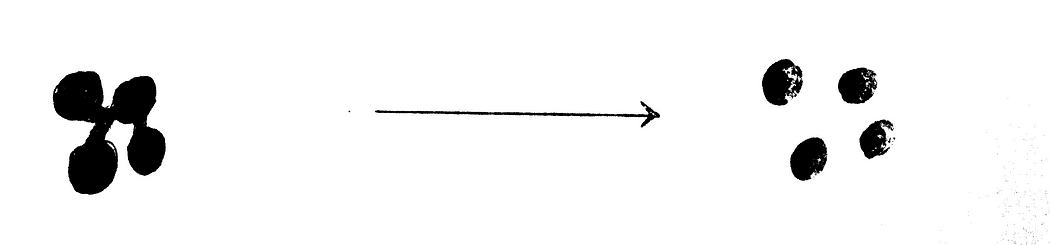
**Late Submissions:**

* Q: Can I skip the lab and submit the solution?
  + You will receive a mark of **zero** if you do not attend the lab, even if you complete the exercise. Attending the labs is compulsory for evaluation. If you have a justified excuse, you may receive a partial mark depending on the circumstances. See the next question for information on late submissions.
* **Q:** If I submit it at 12:00am, you’ll still mark it, right?
  + **A:** 11:59pm and earlier is on time. Anything after 11:59pm is late. Anything late will **NOT** be probably marked. If I find you have a legitimate cause, you will be graded according to the following rules (24 hours after deadline 🡪 assignment is marked out of 75% only, 48 hours after deadline 🡪 assignment is marked out of 50% only, 74 hours after deadline 🡪 assignment is marked out of 25% only).

1. **Introduction (Copied from:** [**https://towardsdatascience.com/understanding-morphological-image-processing-and-its-operations-7bcf1ed11756**](https://towardsdatascience.com/understanding-morphological-image-processing-and-its-operations-7bcf1ed11756)**)**

The word ‘Morphology’ generally represents a branch of biology that deals with the form and structure of animals and plants. However, we use the same term in ‘mathematical morphology’ to extract image components useful in representing region shape, boundaries, etc.



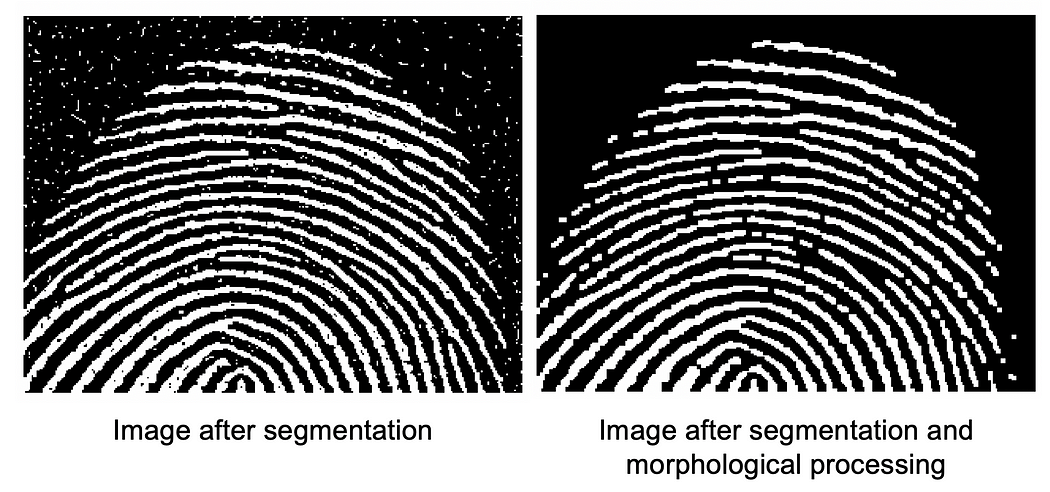


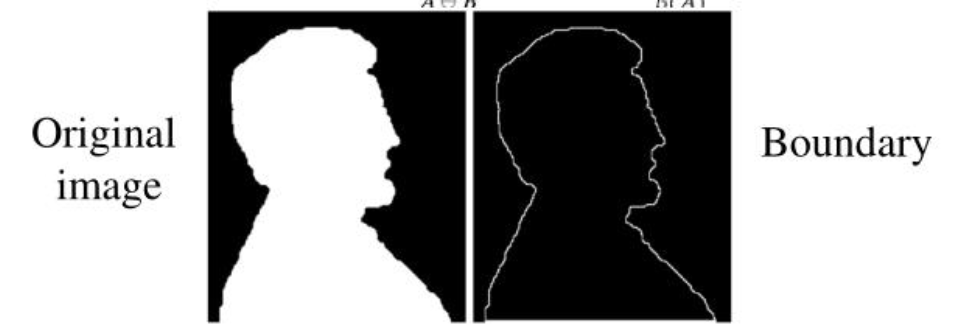
Morphology is a comprehensive set of image processing operations that process images based on shapes [1]. Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors.

There is a slight overlap between Morphology and Image Segmentation. Morphology consists of methods that can be used to pre-process the input data of Image Segmentation or to post-process the output of the Image Segmentation stage. In other words, once the segmentation is complete, morphological operations can be used to remove imperfections in the segmented image and deliver information on the shape and structure of the image as shown in Figure 2.

In this lab we outline the following basic morphological operations:

1. Erosion
2. Dilation
3. Opening
4. Closing
5. White Tophat
6. Black Tophat
7. Skeletonize
8. Convex Hull



Figure 2. Example of Morphological Processing [2].

## Terminologies in Morphological Image Processing

Morphological operation is a technique to process an image based on its shape. It works on comparing the neighboring pixels to structure an image. The process is preferable for binary images ({0,1} or {0,255}).

**How does the process work?**

Before getting familiar with morphological operations, we need to have knowledge about some basic terminologies —**Structuring Element, Miss, Hit and Fit.**

**Structuring Element:**It is a matrix or a small-sized template that is used to traverse an image. The structuring element is positioned at all possible locations in the image, and it is compared with the connected pixels. It can be of any shape.

*Different Structuring Elements —*

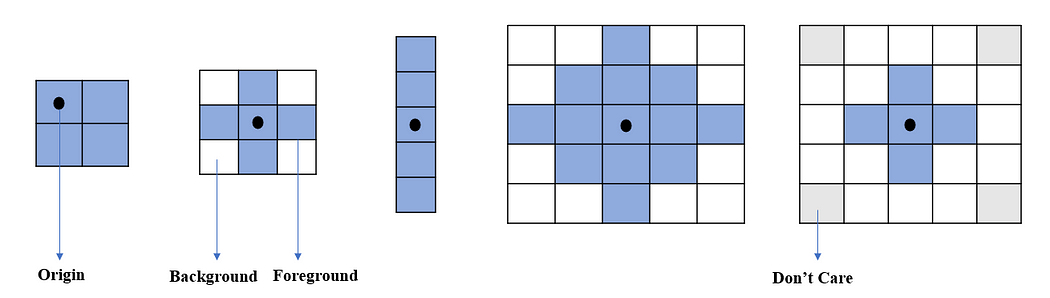


Figure 3: Different Structuring Elements

Structuring elements are designed according to the shape of the image. The size of structuring elements can be of different sizes (2x2, 3x3, 5x1, 5x5, etc.). Structuring elements contains the intensity value of foreground and background (namely 0 or 1). It can also hold the **don’t care** value. One pixel of the structuring element is considered an **origin**. In the above image, the origin pixel with a black dot is mentioned. There is no hard and fast rule to define the origin. It is dependent on you. But conventionally, the origin is considered the center pixel.

Morphological operation is done by propagating the structuring elements through the image. The pixel value is changed in the **origin**position of the image by comparing the pixels under the structuring elements.

**Fit:** When all the pixels in the structuring element cover the pixels of the object, we call it Fit.  
**Hit:**When at least one of the pixels in the structuring element cover the pixels of the object, we call it Hit.  
**Miss:** When no pixel in the structuring element cover the pixels of the object, we call it miss.

Figure 4 shows the visualization of terminologies used in morphological image processing.

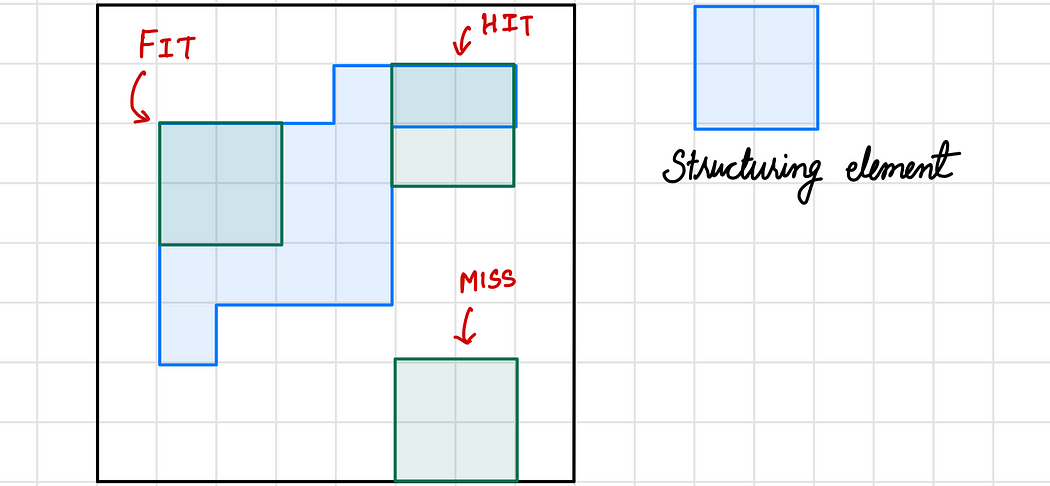


Figure 4. Morphology terminologies explained.

# Morphological Operations

Fundamentally morphological image processing is similar to spatial filtering. The structuring element is moved across every pixel in the original image to give a pixel in a new processed image. The value of this new pixel depends on the morphological operation performed. The two most widely used operations are Erosion and Dilation.

## 1. Erosion

Erosion shrinks the image pixels, or erosion removes pixels on object boundaries. First, we traverse the structuring element over the image object to perform an erosion operation, as shown in Figure 5. The output pixel values are calculated using the following equation.  
Pixel (output) = 1 {if FIT}

Pixel (output) = 0 {otherwise}

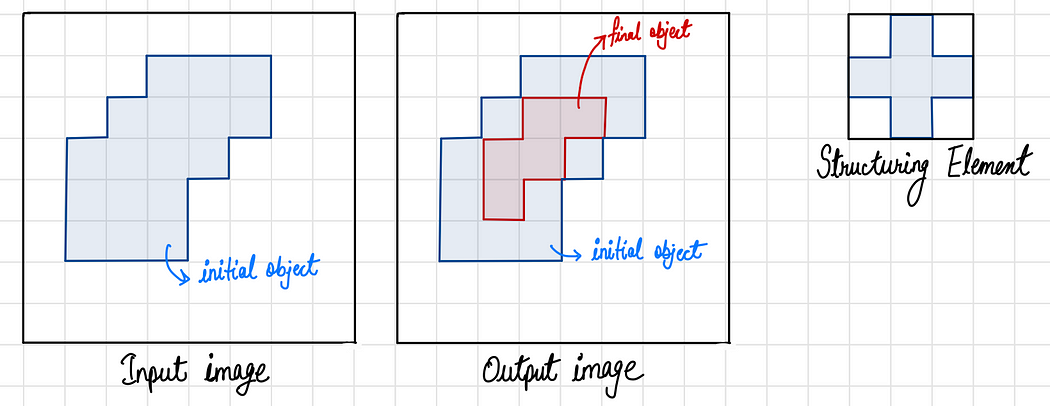


Figure 5. Erosion operation on an input image using a structuring element.

For demonstration purposes, let’s use a 2x2 structuring element with an intensity of 1.

A diagram of a blue square with a black dot and a black arrow

Description automatically generated

Figure-6: Structuring Element

We can take an image with 6x6 pixels. Where the white color elements are considered as 0 intensity value and sky-blue color pixels are considered as intensity value of 1. *Now carefully observe the simulation given below.*

A blue square in a grid

Description automatically generated

Erosion Operation (Gif)

The structuring element convolutes every pixel of the given image. If it satisfies the miss or hit condition, it will change the pixel to 0 in the **origin's**location of the structuring element. In the simulation, I have shown where the pixel is changed from 1 to 0 with red color. Finally, we get the following result.

A diagram of a graph

Description automatically generated with medium confidence

Figure-7: Erosion Result

An example of Erosion is shown in Figure 6. Figure 8(a) represents original image, 8(b) and 8(c) shows processed images after erosion using 3x3 and 5x5 structuring elements respectively.

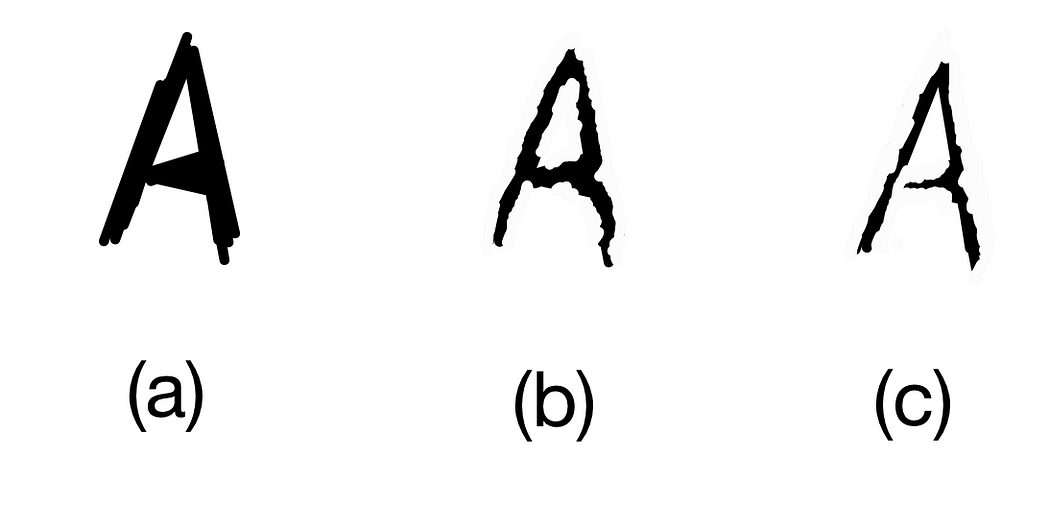


Figure 8. Results of structuring element size in erosion. (Source: Image by the author)

**Properties:**

1. It can split apart joint objects
2. It can strip away extrusions

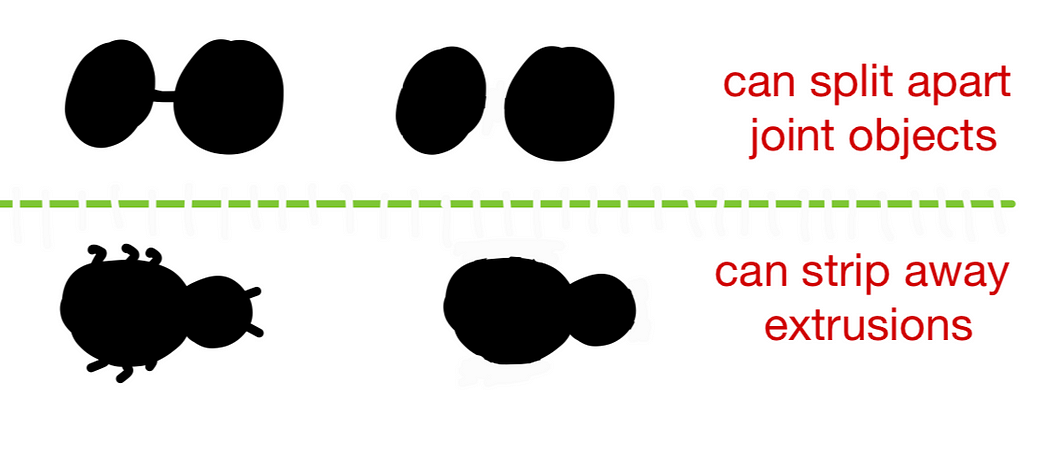
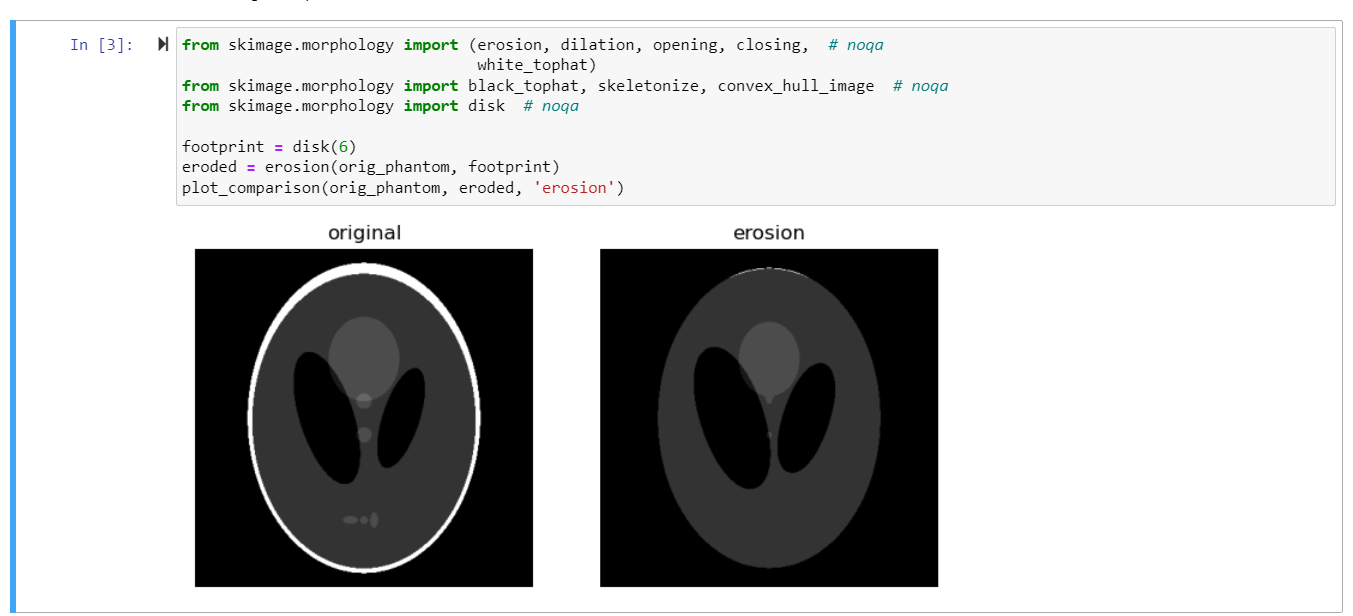


Figure 9 Example use-cases of Erosion.



## 2. Dilation

Dilation expands the image pixels, or it adds pixels on object boundaries. First, we traverse the structuring element over the image object to perform an dilation operation, as shown in Figure 10. The output pixel values are calculated using the following equation.  
Pixel (output) = 1 {if HIT}  
Pixel (output) = 0 {otherwise}

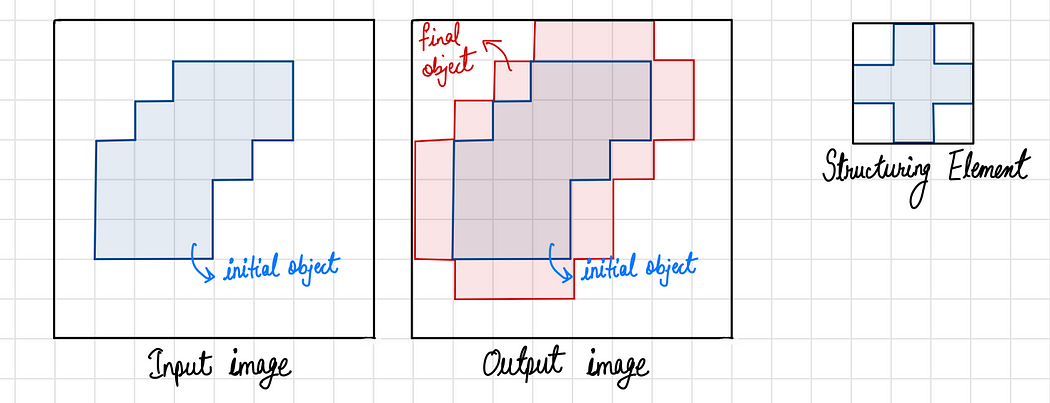
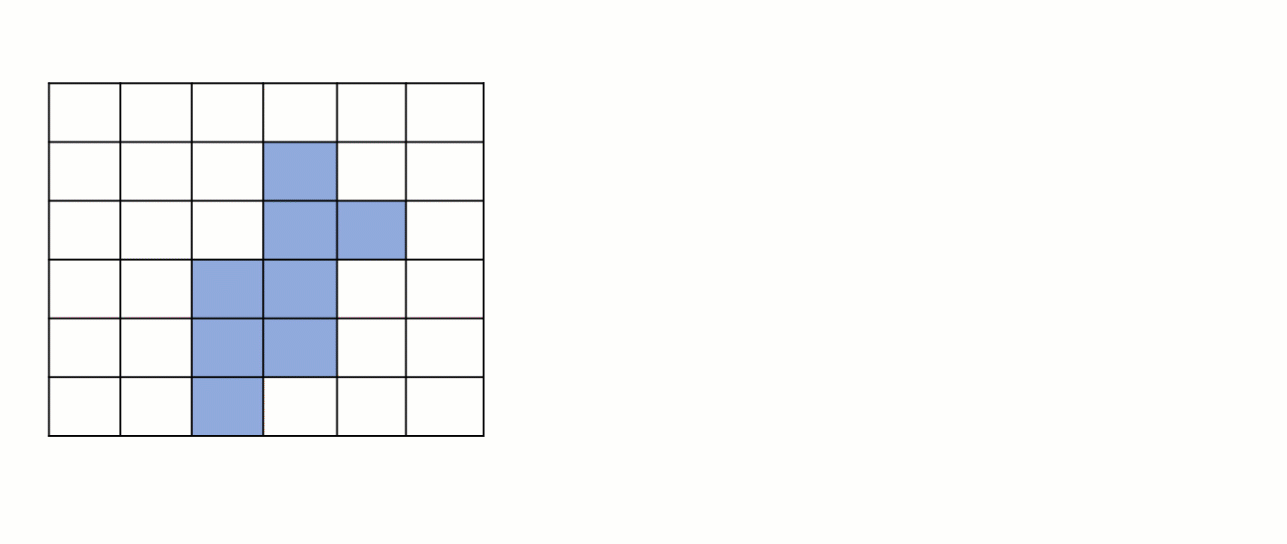


Figure 10. Dilation operation on an input image using a structuring element.

*Now, carefully observe the dilation operation shown below.*



Dilation Operation (Gif by Author)

The structuring element is convoluting in every pixel of the object image from left to right and top to bottom. When it satisfies the hit or miss condition, the pixel in the origin’s location of the structuring element changes from 0 to 1. Otherwise, it remains the same. After completing the operation, it produces the result shown below.

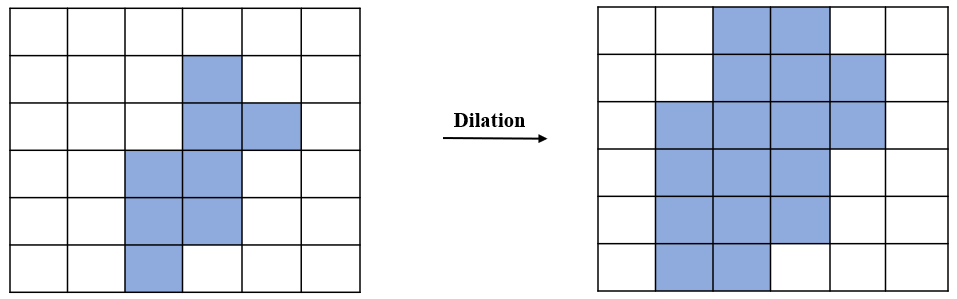


Figure-11: Dilation result

So, dilation increases the pixels of the object image.

An example of Dilation is shown in Figure 12. Figure 12(a) represents original image, 12(b) and 12(c) shows processed images after dilation using 3x3 and 5x5 structuring elements respectively.

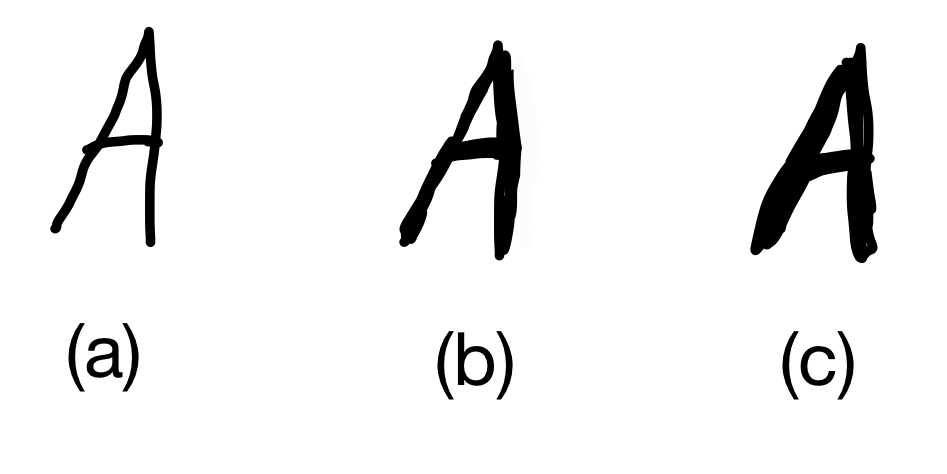


Figure 12. Results of structuring element size in dilation. (Source: Image by the author)

**Properties:**

1. It can repair breaks
2. It can repair intrusions

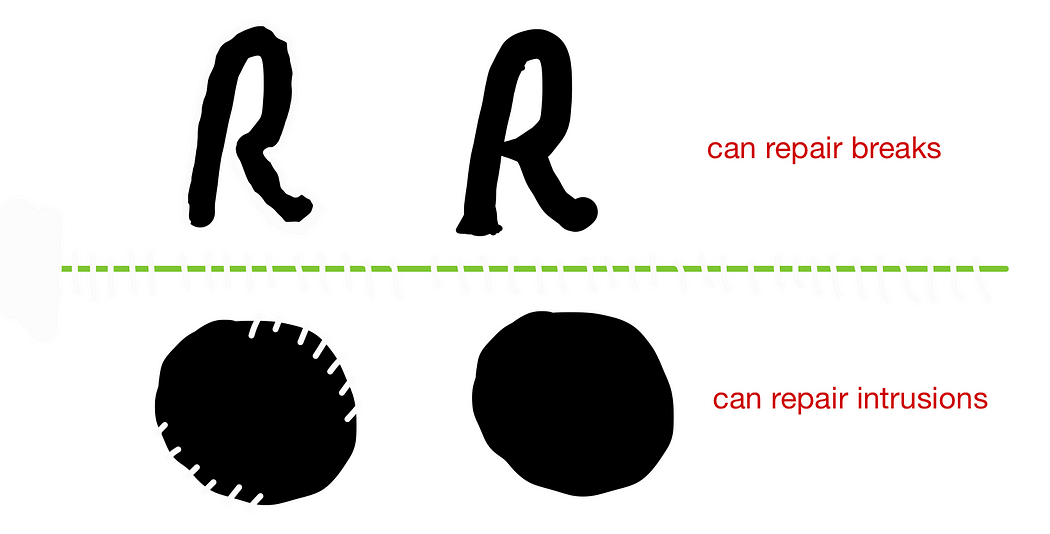


Figure 13. Example use-cases of DIlation. (Source: Image by the author)



# Compound Operations

Most morphological operations are not performed using either dilation or erosion; instead, they are performed by using both. Two most widely used compound operations are: (a) Closing (by first performing dilation and then erosion), and (b) Opening (by first performing erosion and then dilation). Figure 10 shows both compound operations on a single object.

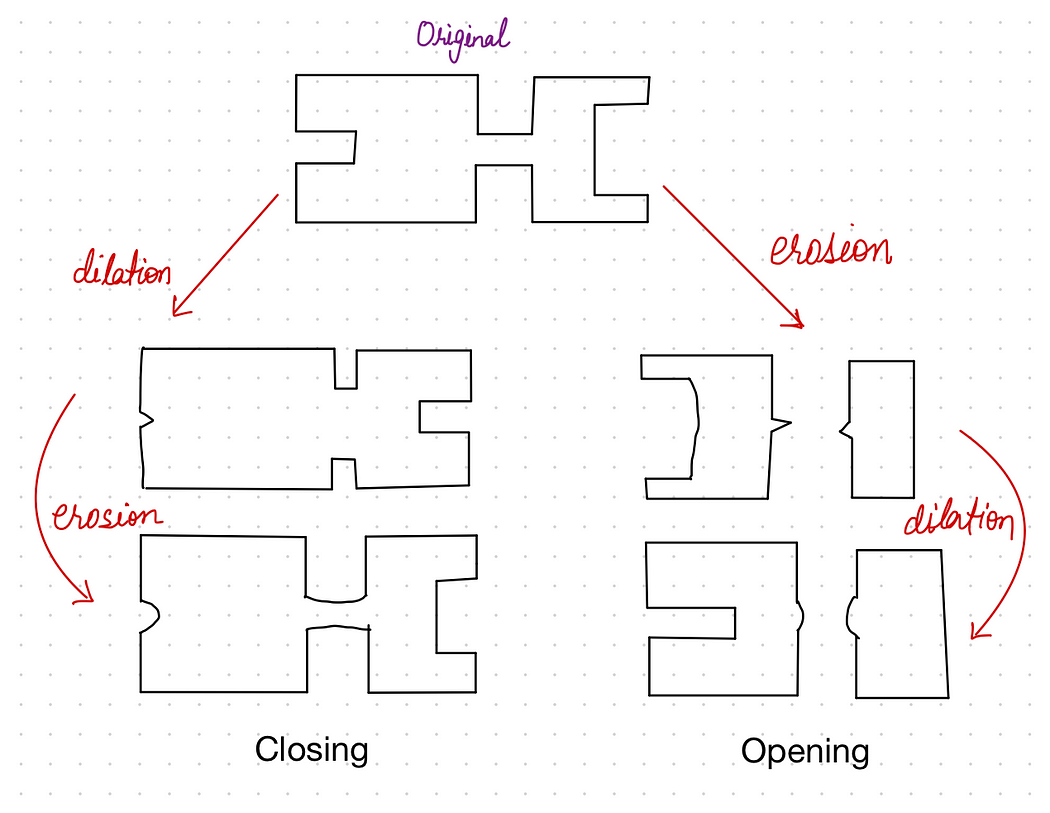
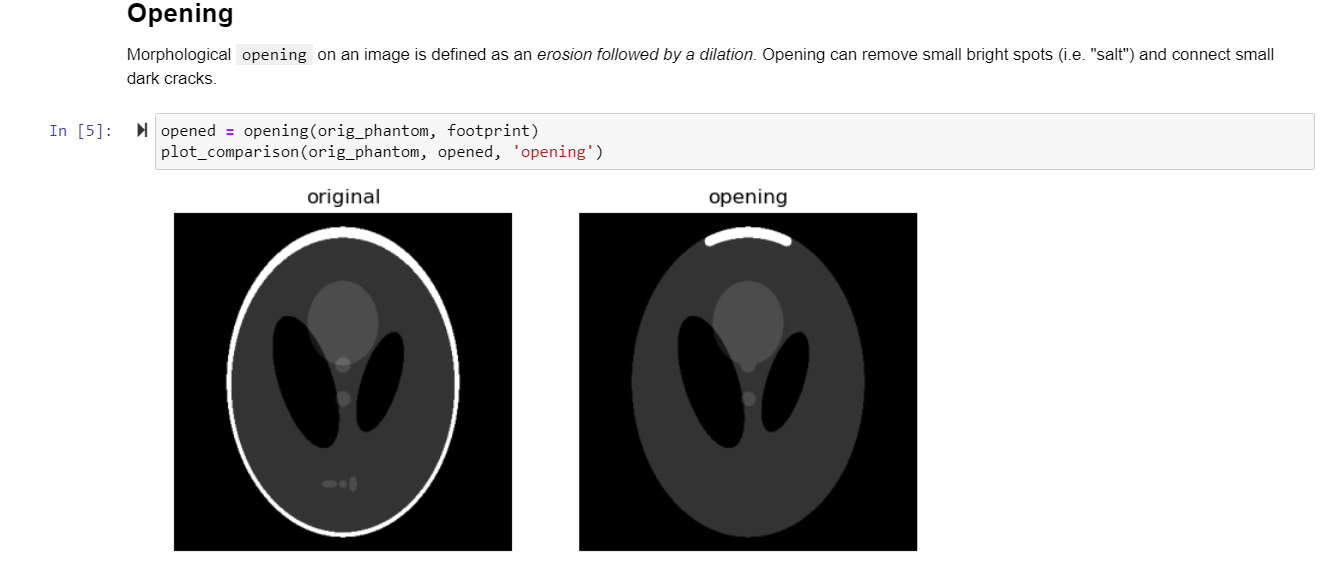


Figure 13. Output of Compound operations on an input object. (Source: Image by the author)



A screenshot of a computer

Description automatically generatedWhite tophat

The white\_tophat of an image is defined as the image minus its morphological opening. This operation returns the bright spots of the image that are smaller than the structuring element.

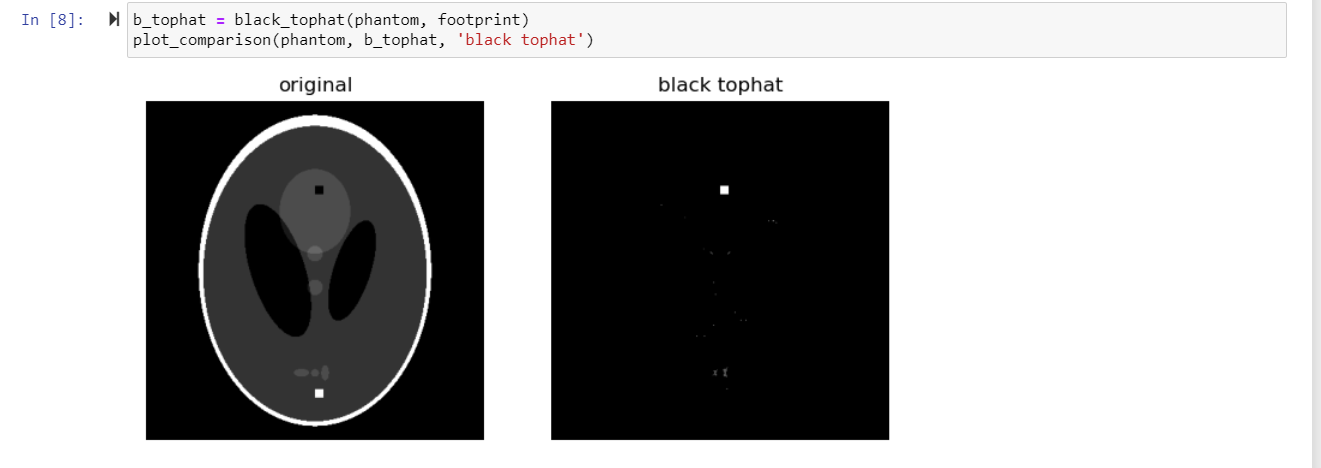
To make things interesting, we'll add bright and dark spots to the image:



As you can see, the 10-pixel wide white square is highlighted since it is smaller than the structuring element. Also, the thin, white edges around most of the ellipse are retained because they're smaller than the structuring element, but the thicker region at the top disappears.

## Black tophat

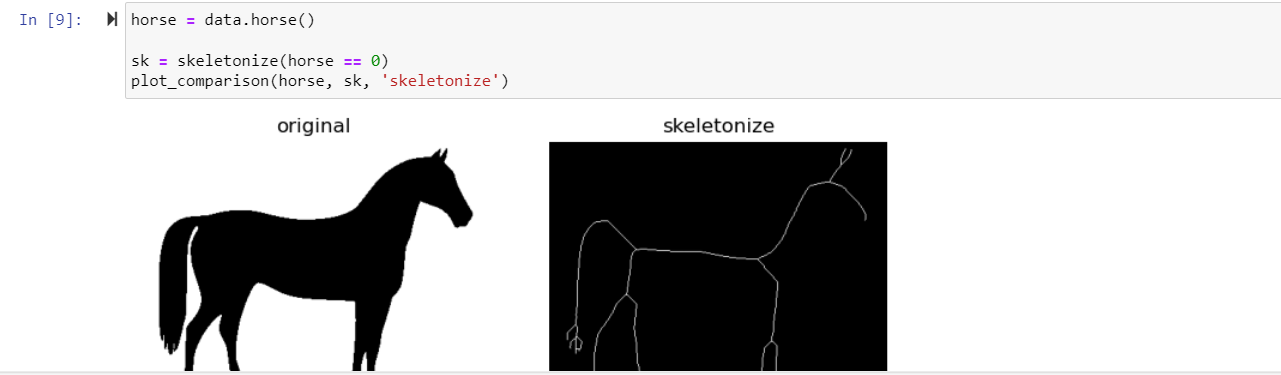
The black\_tophat of an image is defined as its morphological **closing minus the original image**. This operation returns the dark spots of the image that are smaller than the structuring element.



# Skeletonize

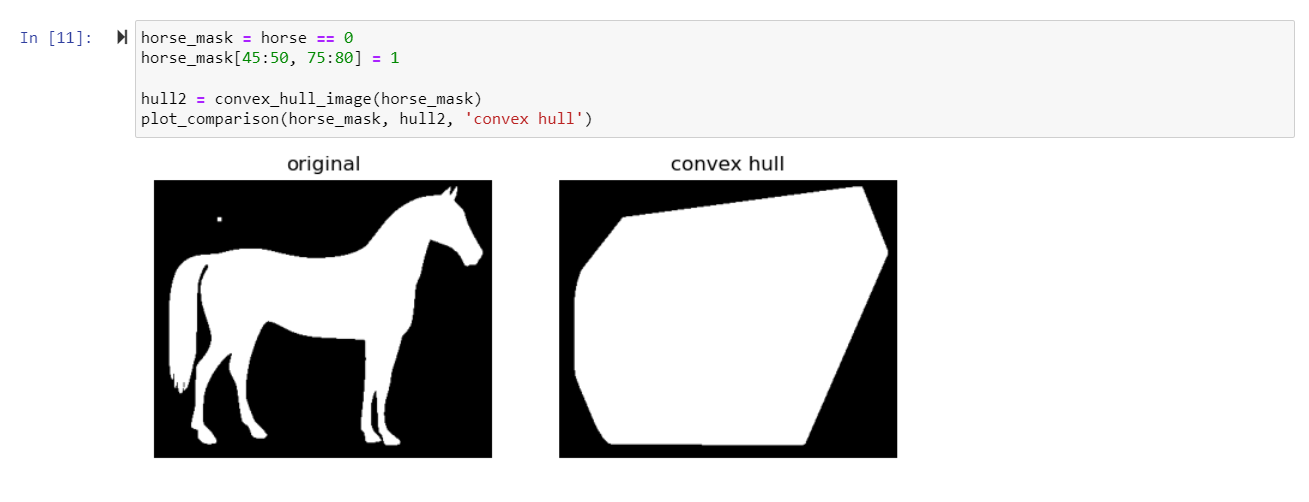
Skeletonization reduces binary objects to 1 pixel wide representations. This can be useful for feature extraction, and/or representing an object’s topology.

skeletonize works by making successive passes of the image. On each pass, border pixels are identified and removed on the condition that they do not break the connectivity of the corresponding object.



# Convex Hull

The convex hull of a binary image is the set of pixels included in the smallest convex polygon that surround all white pixels in the input.



# References

[1] P Soille. “Morphological Image Analysis, Principles and Applications”, 1999.

[2] R. C. Gonzalez, R. E. Woods, “Digital image processing”, 2nd ed. Upper Saddle River, N.J. Prentice Hall, 2002.

[3] P Chhikara, “Understanding Morphological Image Processing and Its Operations”, 2022, <https://towardsdatascience.com/understanding-morphological-image-processing-and-its-operations-7bcf1ed11756>

[5] Md. Zubair “Morphological Operations with Simulation (CV-05) ”, 2023

[4] [skimage.morphology — skimage 0.21.0 documentation (scikit-image.org)](https://scikit-image.org/docs/stable/api/skimage.morphology.html)

Your Task:

Task 1

1. Display a cropped color image of the model space shuttle in the image.
2. Display each color channel of image [0:200, 0:100] in grayscale.
3. Which channel is red?
4. Which channel is blue?
5. Which channel is green?

[Extra challenge] Create a new astronaut image where:

1. all perfectly white (r,g,b=255) pixels have been converted to red (r=255; g,b=0)
2. all perfectly black (r,g,b=0) pixels have been converted to blue (r,g=0; b=255)

Task 2

**Complete your progress in the first lab.**

Return to the second task in the lab 1, and try altering Image Brightness using skimage

Hit: use the adjust\_gamma function in skimage

**Assessment**

1. Each student will show all the above parts running as demo to the Lab Instructor **before leaving the lab.** Total marks for the lab is as follows

|  |  |
| --- | --- |
| Task 1 | Marks (demo + report) |
| 1 | 10 |
| 2 | 10 |
| Total | 20 |

1. Students will prepare a report in which they will submit the snapshots taken while they worked on each part. They will explain the figures to make sure that they understood what they did.

**References:**

[Getting started — skimage 0.21.0 documentation (scikit-image.org)](https://scikit-image.org/docs/stable/user_guide/getting_started.html)

[Rescale, resize, and downscale — skimage 0.21.0 documentation (scikit-image.org)](https://scikit-image.org/docs/stable/auto_examples/transform/plot_rescale.html#sphx-glr-auto-examples-transform-plot-rescale-py)

[Skimage | Skimage Tutorial | Skimage Python (analyticsvidhya.com)](https://www.analyticsvidhya.com/blog/2019/09/9-powerful-tricks-for-working-image-data-skimage-python/)

[Basic\_skimageJAX/lessons/1-Images\_are\_arrays.md at master · TheJacksonLaboratory/Basic\_skimageJAX (github.com)](https://github.com/TheJacksonLaboratory/Basic_skimageJAX/blob/master/lessons/1-Images_are_arrays.md)