

Morphological Operations

Submitted by Mary Franczine Tan

Objectives

- 1 Understand how morphological operations occur by visualizing them on paper
- 2 Simulate morphological operations on python and compare paper results
- 3 Apply morphological operations in cleaning of cell image

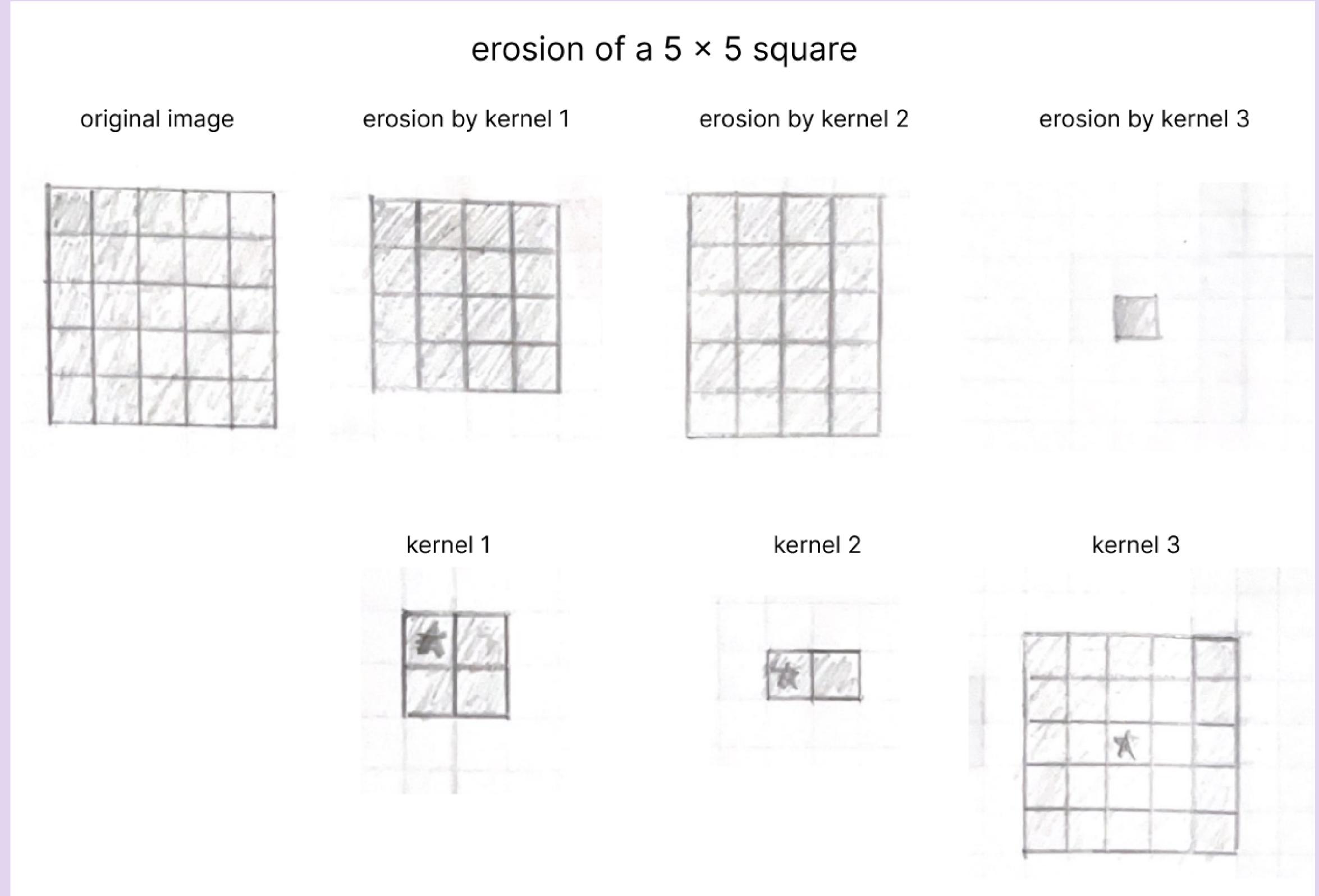
Results and Analysis

Manual Simulation of Morphological Operations

Erosion is the process of removing pixels that don't match with the structuring element given an origin. This results in a shape that is smaller than the original image.

The image to the side shows the effect of erosion on a 5×5 square given different structuring elements.

We see that kernels affect the resulting size and shape of the image. In applications of morphological operations, it is then important to choose the appropriate kernel for your desired effect.



The image to the side shows the **erosion of a 10×10 hollow square, 2 boxes thick** using different structuring elements.

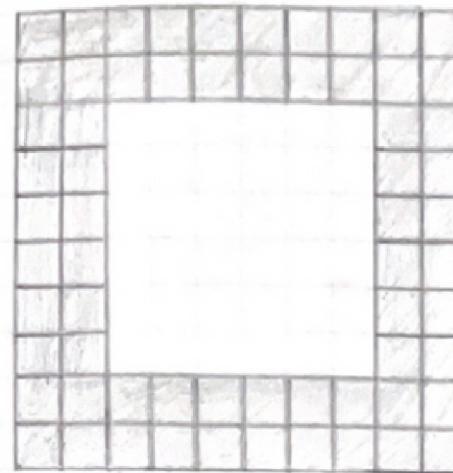
For **kernel 1 and 2**, we observe that **portions of the hollow square got thinner**.

Meanwhile for **kernel 3**, the **hollow square disappeared entirely**. This is because the structuring element is bigger each connected pixel in the image.

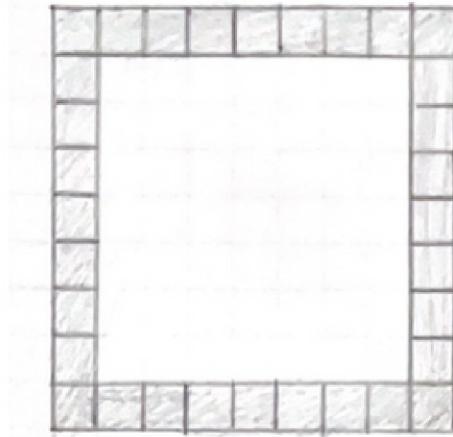
This shows that **erosion** is also a **way of removing unwanted noise or errant pixels in an image**.

erosion of a 10×10 hollow square, 2 boxes thick

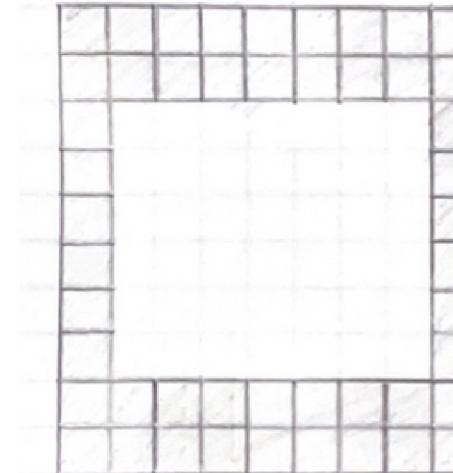
original image



erosion by kernel 1

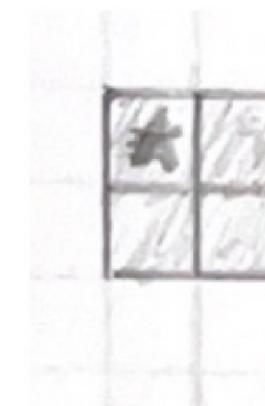


erosion by kernel 2

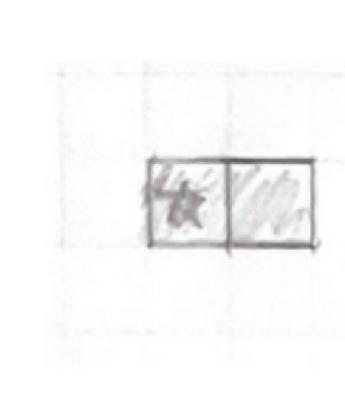


erosion by kernel 3

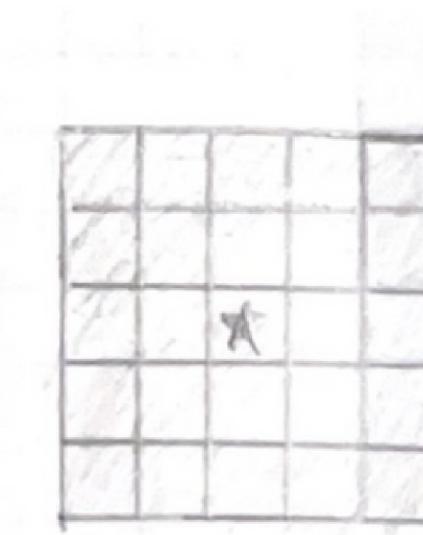
kernel 1



kernel 2

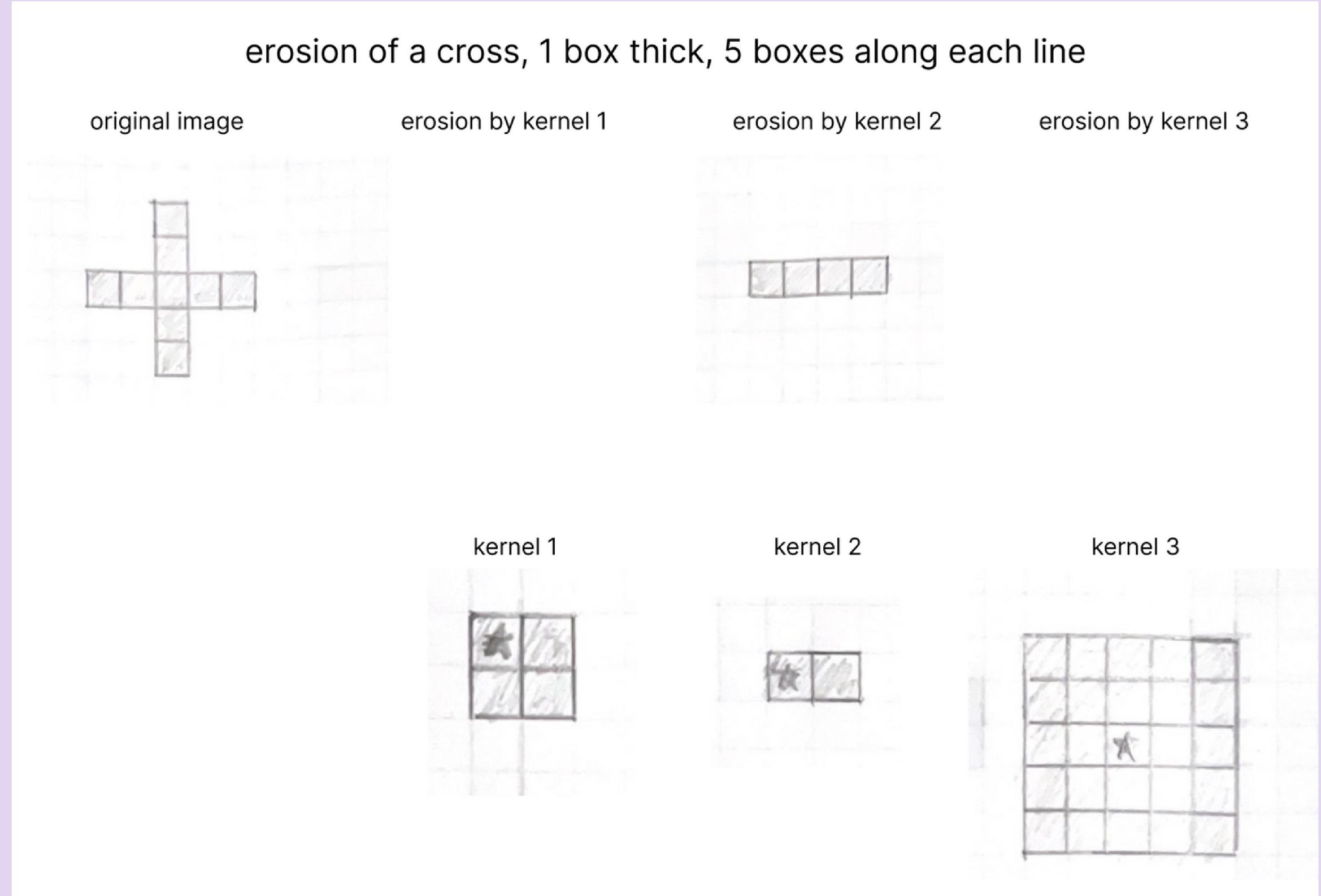


kernel 3



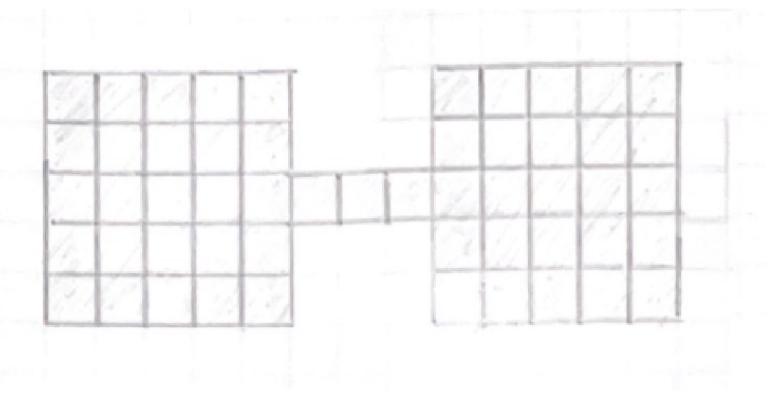
A similar concept is observed with the previous erosions.

For the erosion of a cross that is **1 box thick**, we see that it is **removed entirely with kernel 1 and kernel 3**.

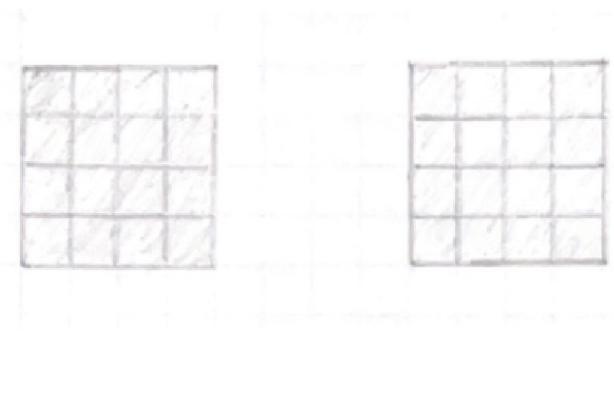


erosion of a dumbbell - two 5×5 squares connected by a 3×1 line

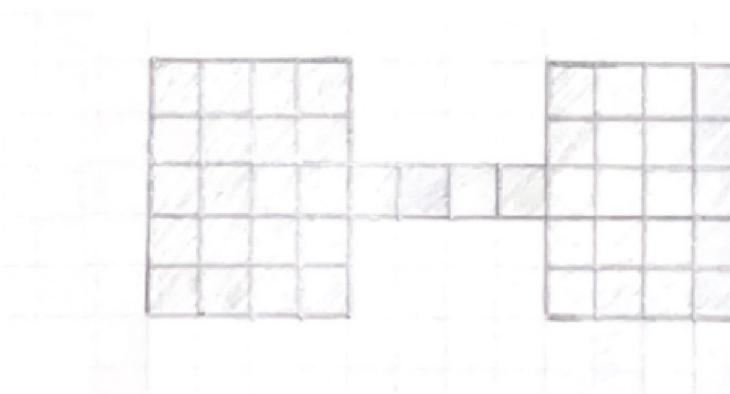
original image



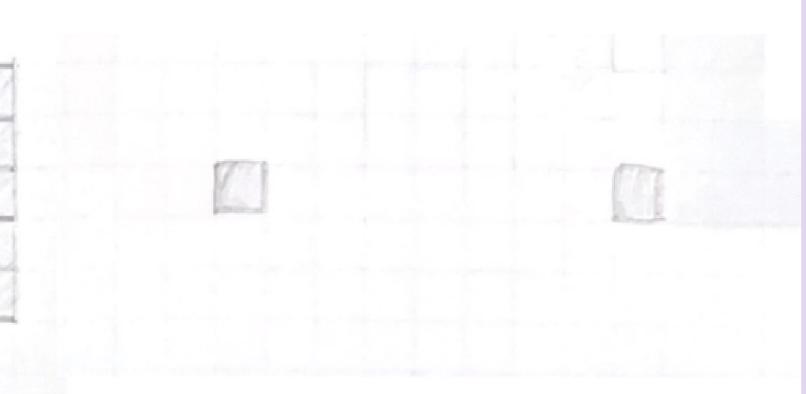
erosion by kernel 1



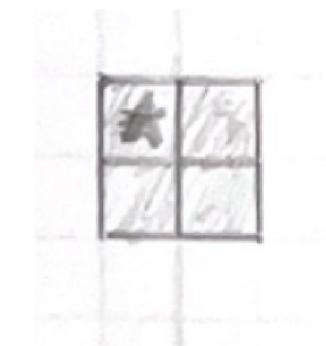
erosion by kernel 2



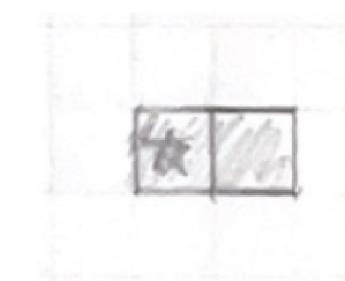
erosion by kernel 3



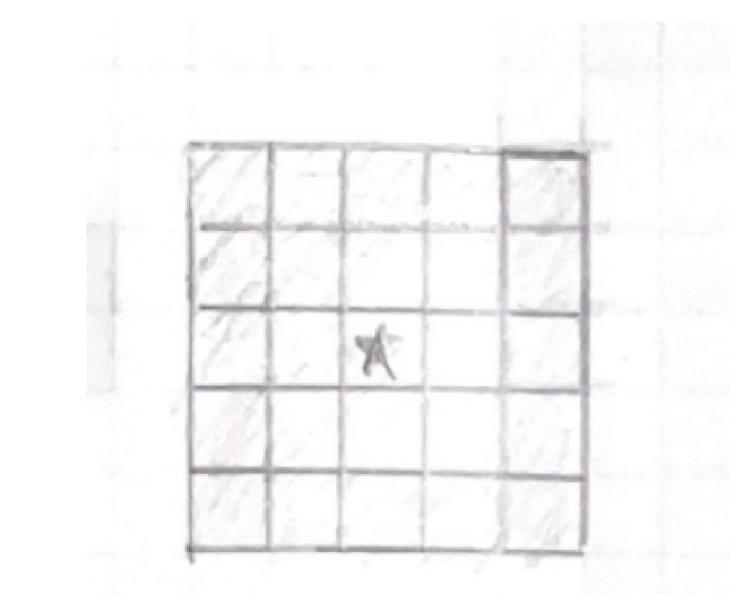
kernel 1



kernel 2

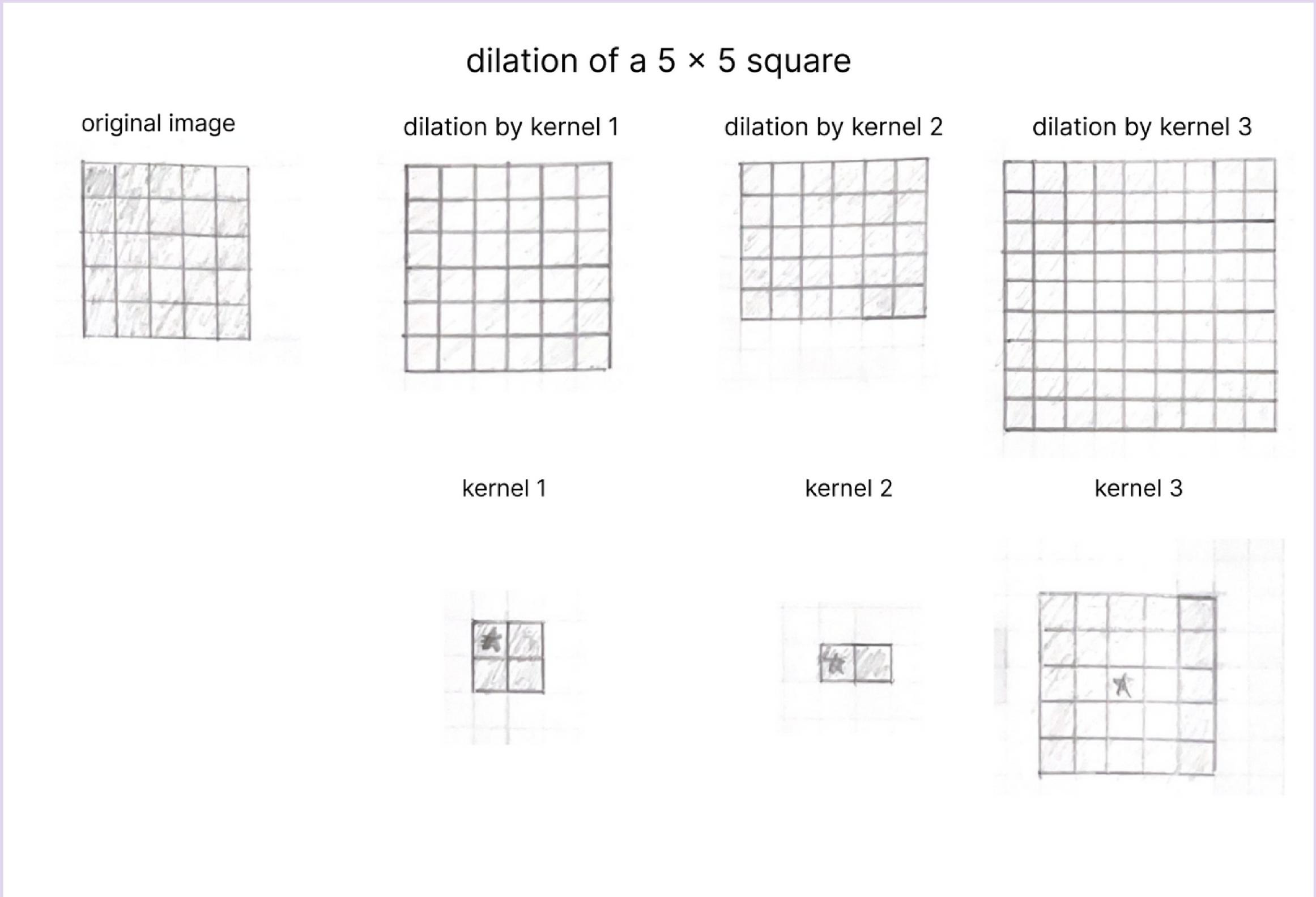


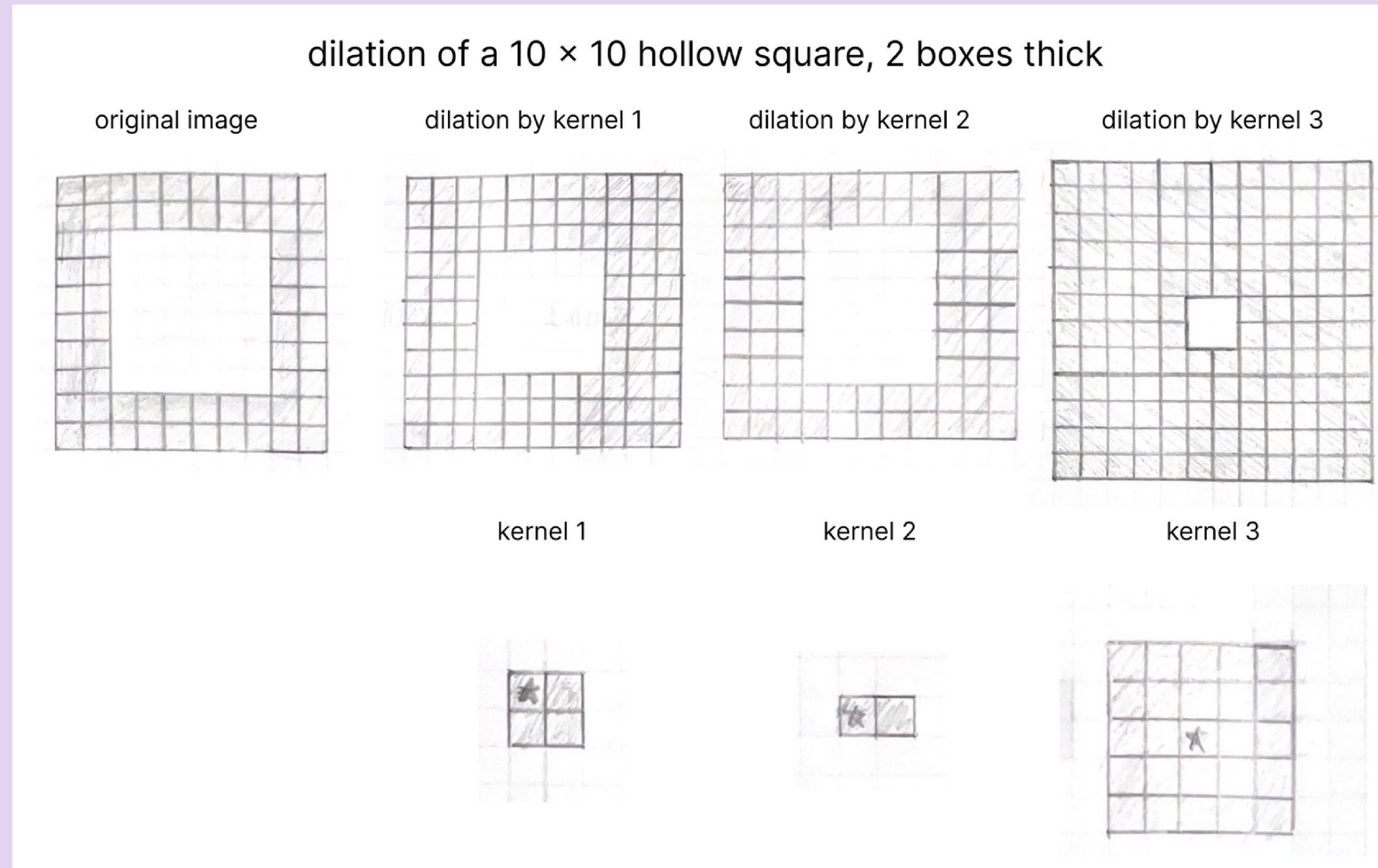
kernel 3



Dilation is the process of adding pixels in areas that don't match with the structuring element given an origin. This results in a shape that is larger than the original image.

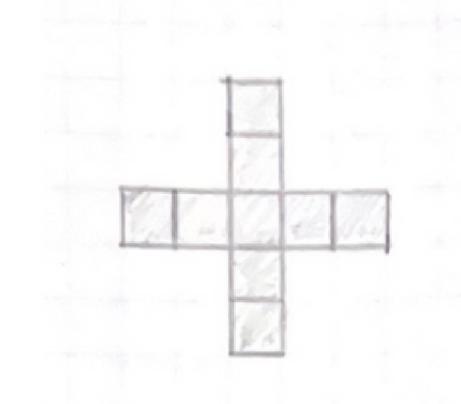
The image to the side shows the effect of dilation on a 5×5 square given different structuring elements.



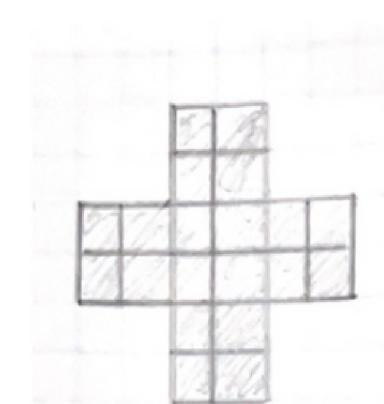


dilation of a cross, 1 box thick, 5 boxes along each line

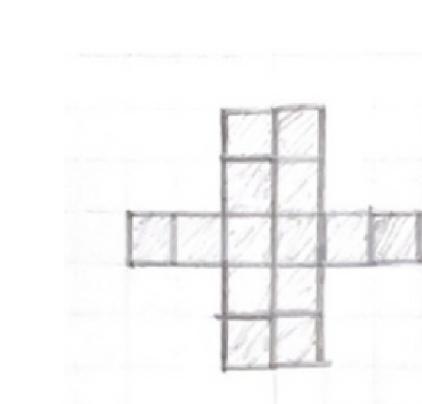
original image



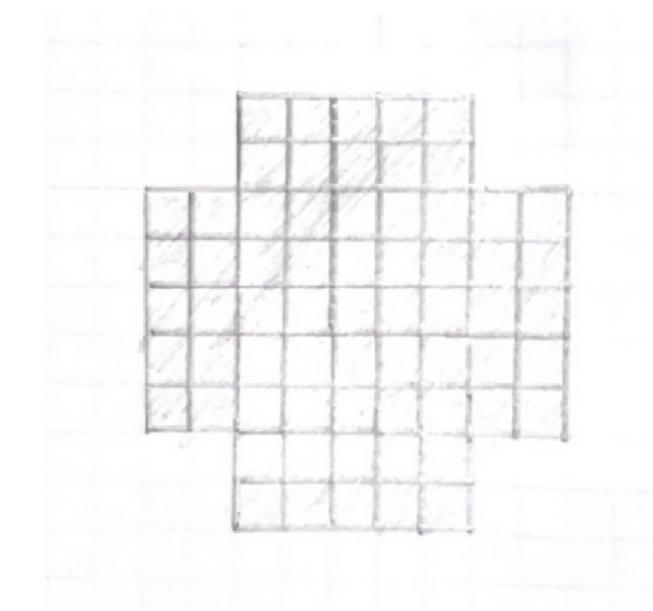
dilation by kernel 1



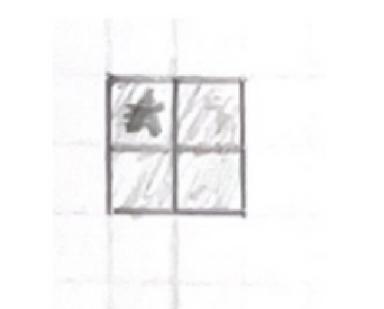
dilation by kernel 2



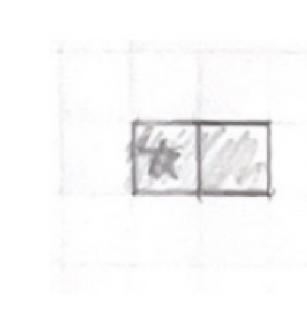
dilation by kernel 3



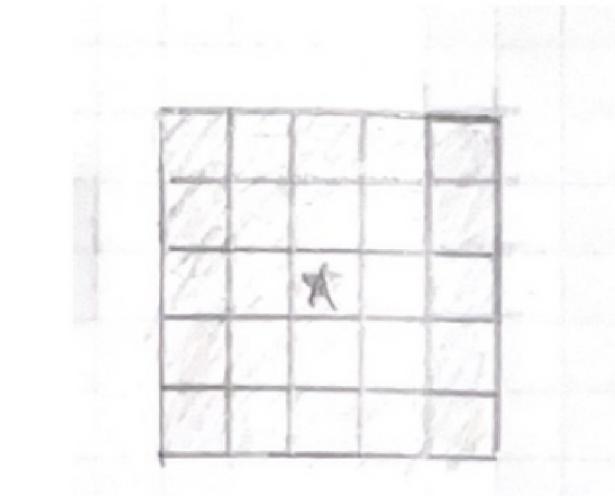
kernel 1



kernel 2



kernel 3

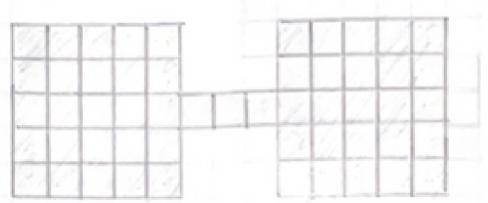


In the dilation of this dumbbell shape by kernel3, we see another purpose of dilation other than enlarging an image.

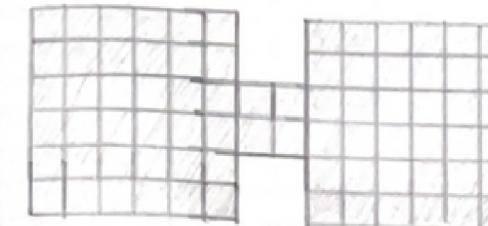
Here, we see that it is capable of **joining together two different blobs of shape into one**.

dilation of a dumbbell - two 5×5 squares connected by a 3×1 line

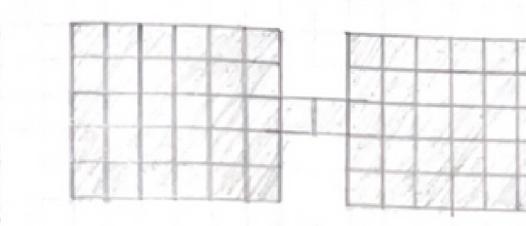
original image



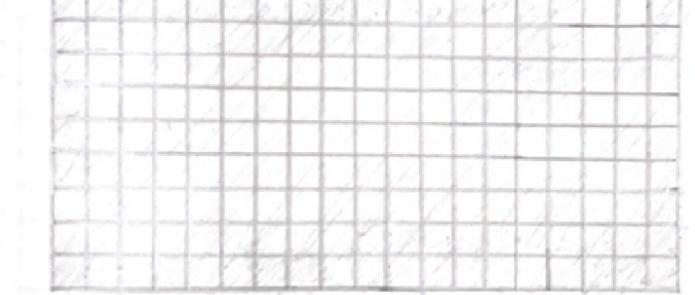
dilation by kernel 1



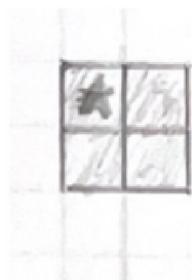
dilation by kernel 2



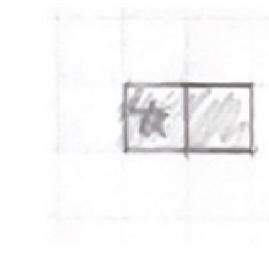
dilation by kernel 3



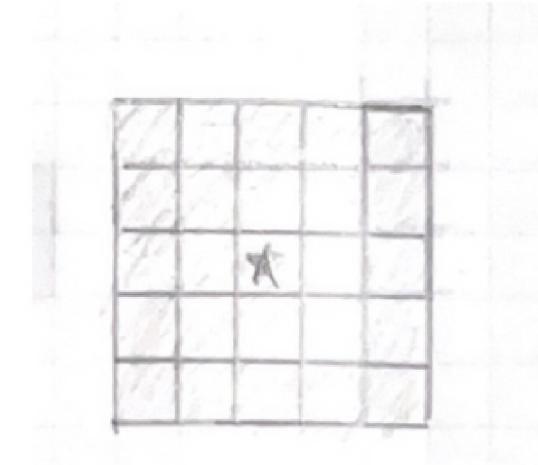
kernel 1



kernel 2

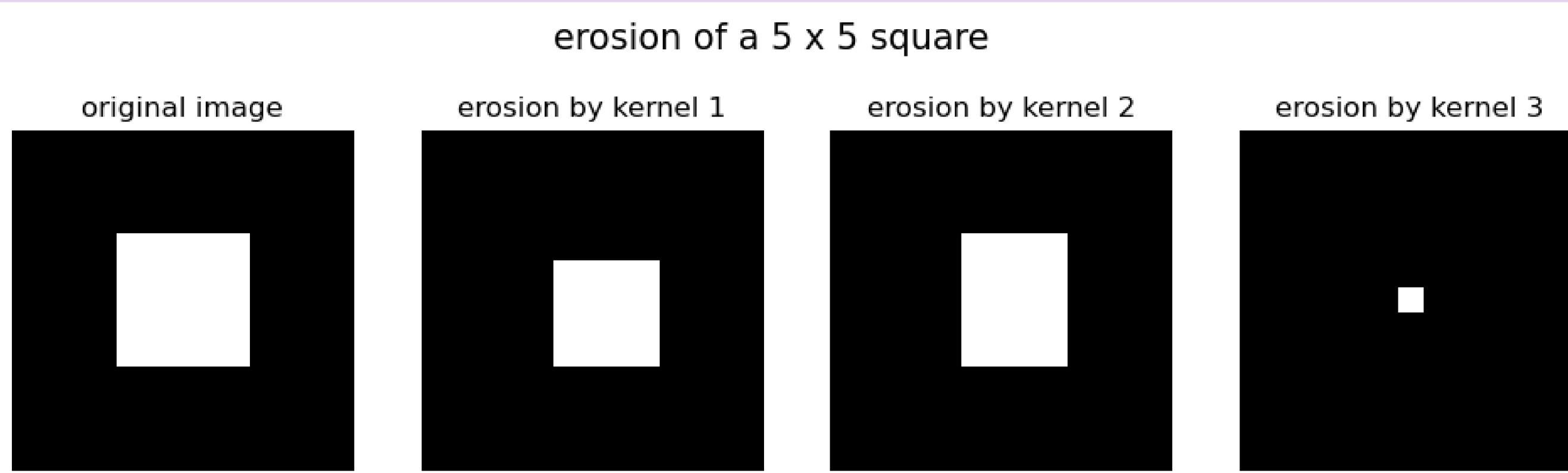


kernel 3



Results and Analysis

Python Simulation of Morphological Operations



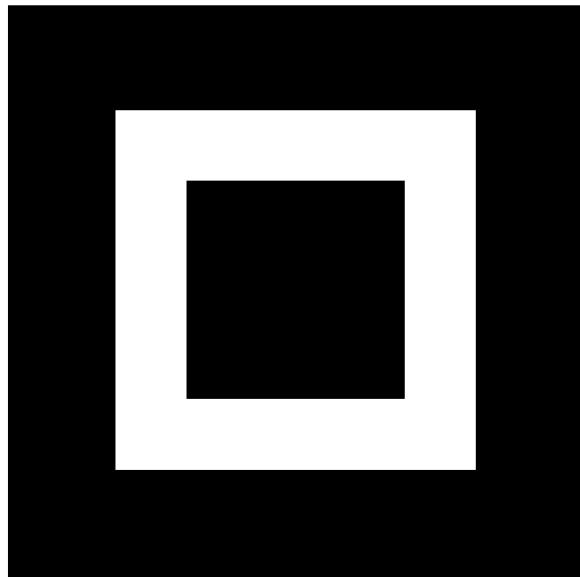
Here, we see that the **square gets smaller, similar to manual simulations**. But, the pixels removed are not the same.

This is because the **origin of the structuring element** in Python is different from that used in the drawing.

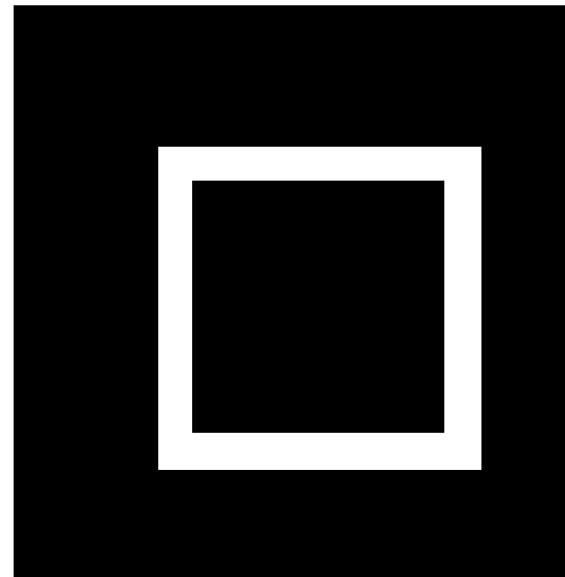
kernel 1: 2×2 oneskernel 2: 2×1 oneskernel 3: 5×5 ones

erosion of a 10 x 10 hollow square, 2 boxes thick

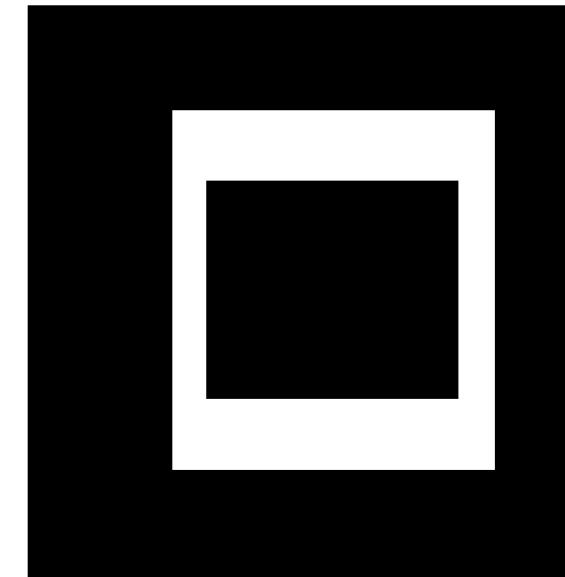
original image



erosion by kernel 1



erosion by kernel 2



erosion by kernel 3



kernel 1: 2 x 2 ones



kernel 2: 2 x 1 ones



kernel 3: 5 x 5 ones

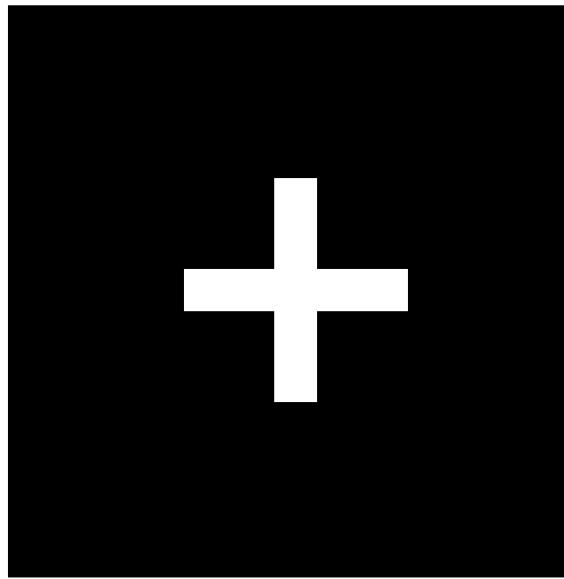


The same is observed for all erosion examples



erosion of a plus sign, 1 box thick, 5 boxes along each line

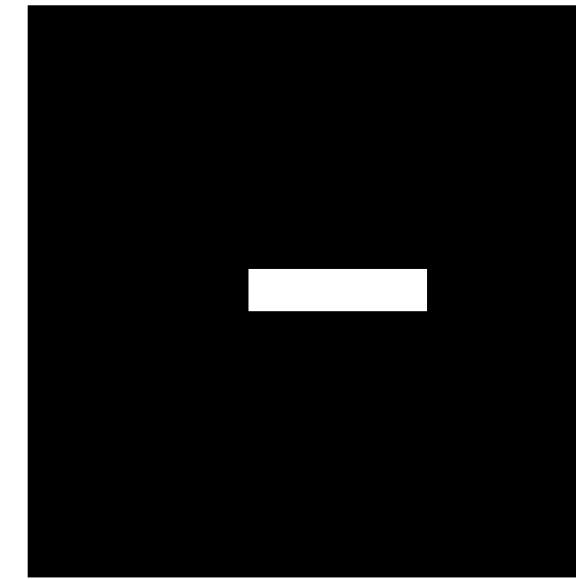
original image



erosion by kernel 1



erosion by kernel 2



erosion by kernel 3



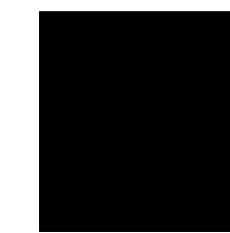
kernel 1: 2 x 2 ones



kernel 2: 2 x 1 ones

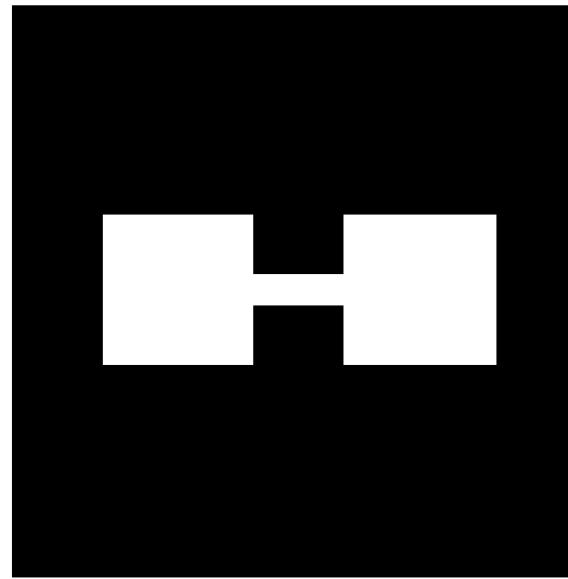


kernel 3: 5 x 5 ones

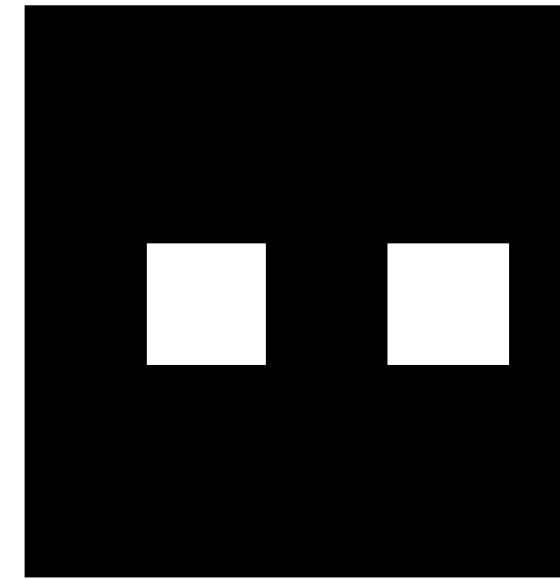


erosion of a dumbbell - two 5x5 squares connected by a 3x1 line

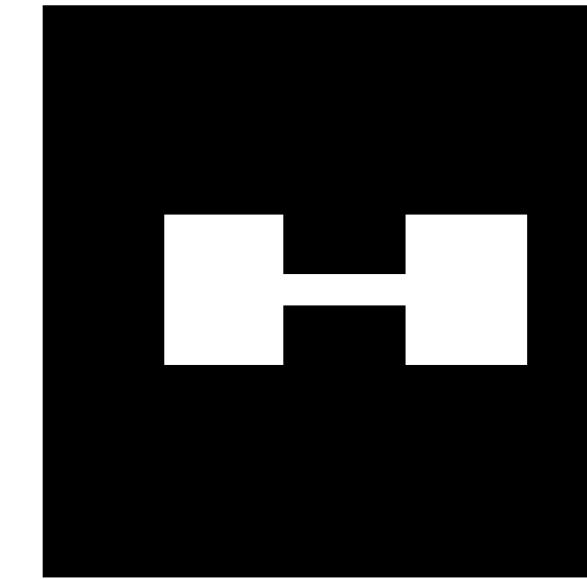
original image



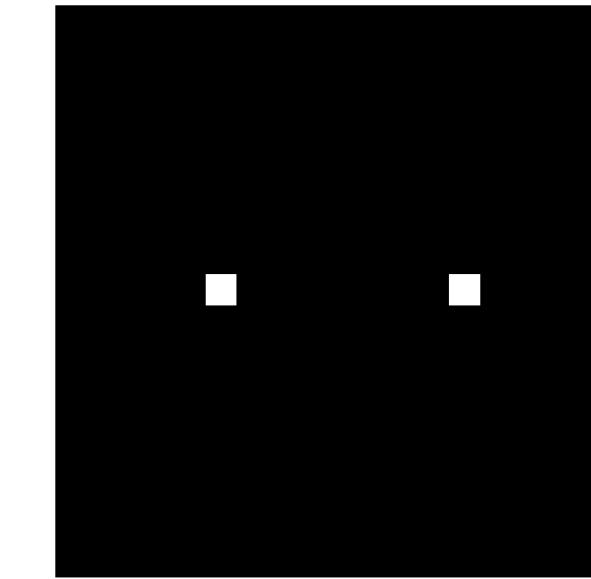
erosion by kernel 1



erosion by kernel 2



erosion by kernel 3



kernel 1: 2 x 2 ones

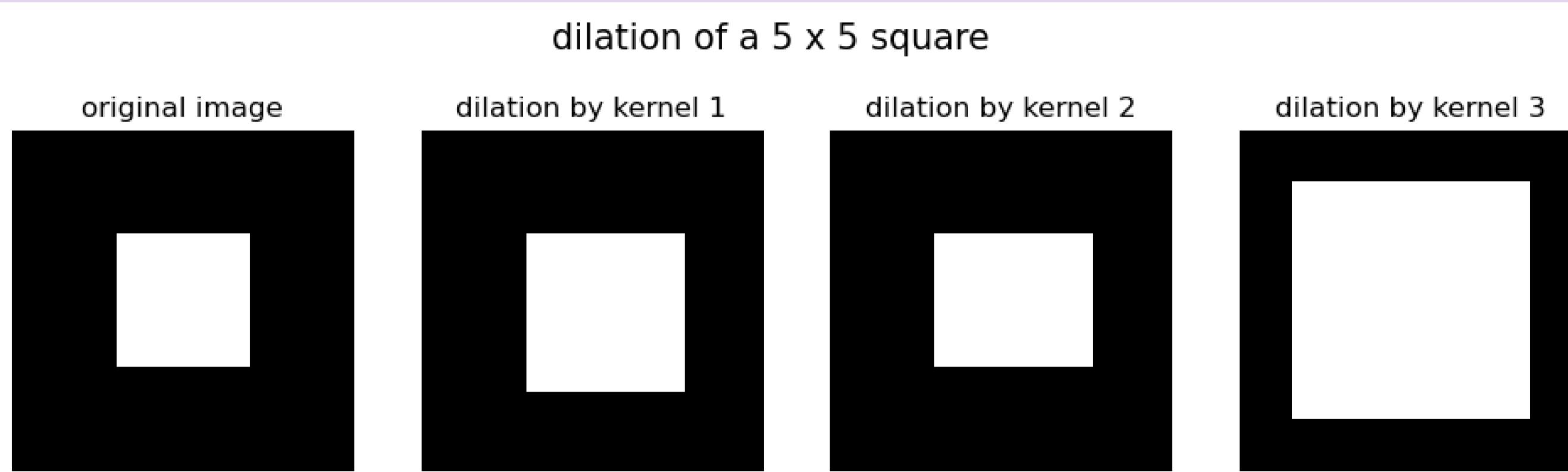


kernel 2: 2 x 1 ones



kernel 3: 5 x 5 ones





kernel 1: 2 x 2 ones



kernel 2: 2 x 1 ones



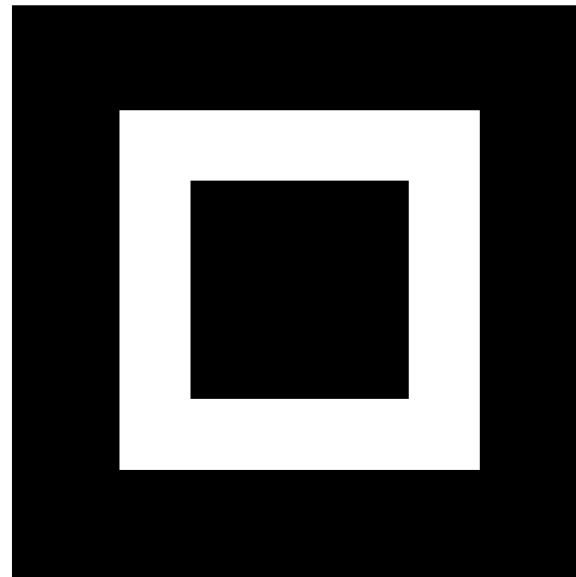
kernel 3: 5 x 5 ones



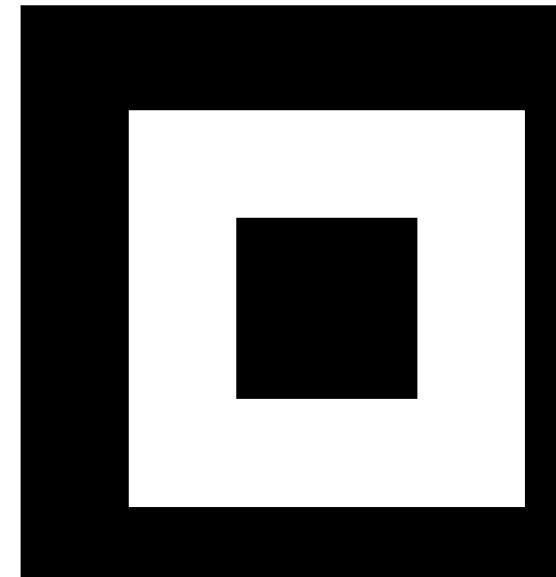
For dilation, we see that the **square gets bigger, also similar to the manual simulations.**

dilation of a 10×10 hollow square, 2 boxes thick

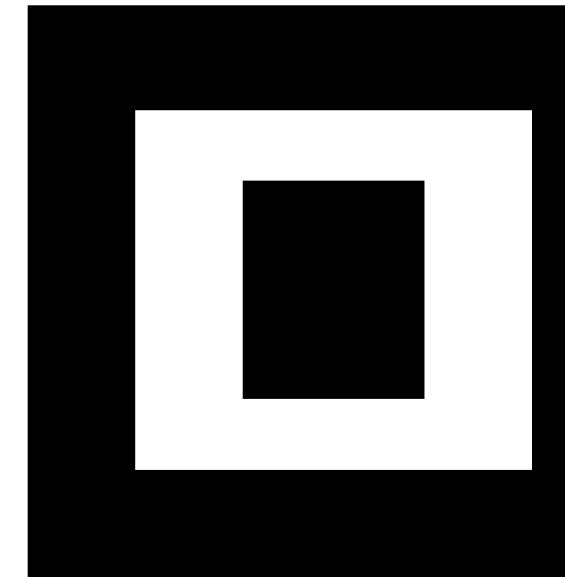
original image



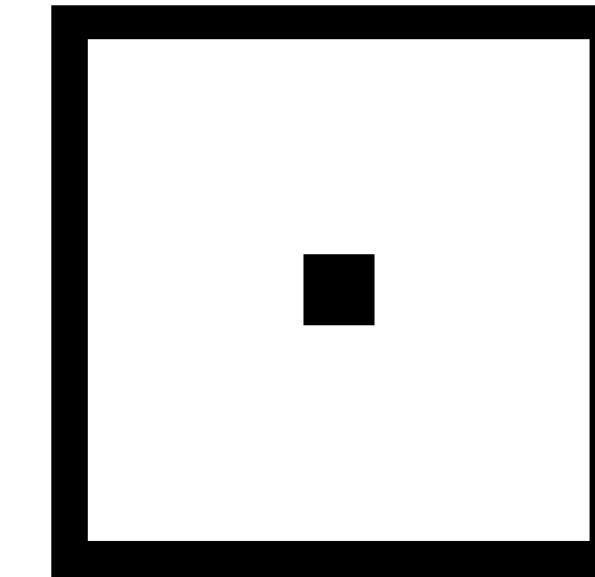
dilation by kernel 1



dilation by kernel 2



dilation by kernel 3



kernel 1: 2 x 2 ones



kernel 2: 2 x 1 ones

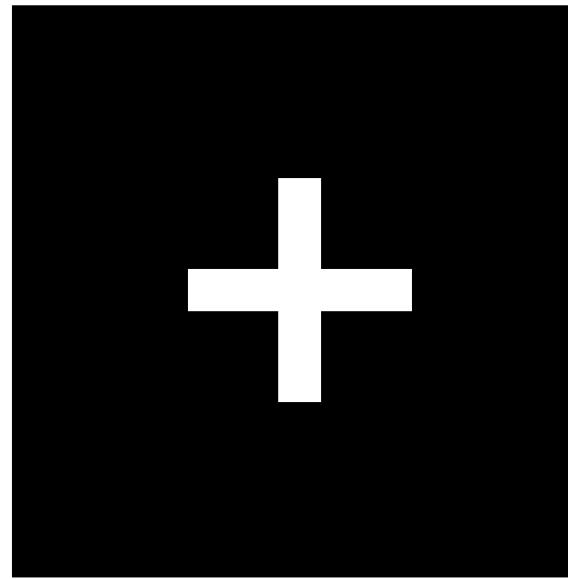


kernel 3: 5 x 5 ones

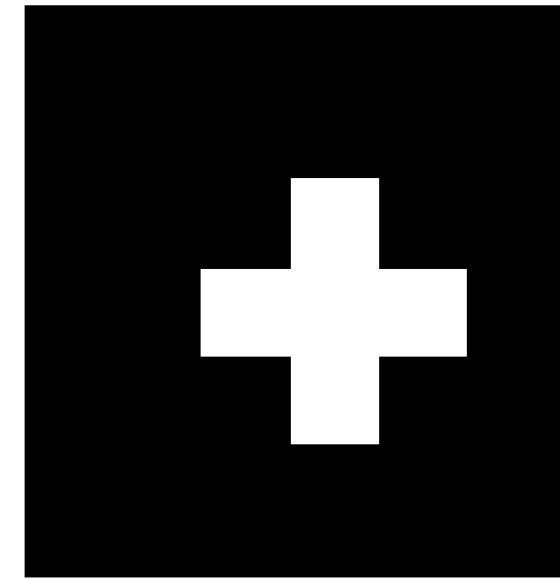


dilation of a plus sign, 1 box thick, 5 boxes along each line

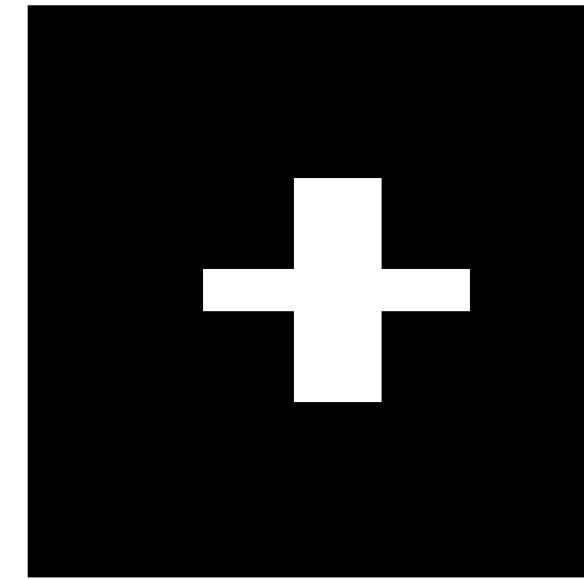
original image



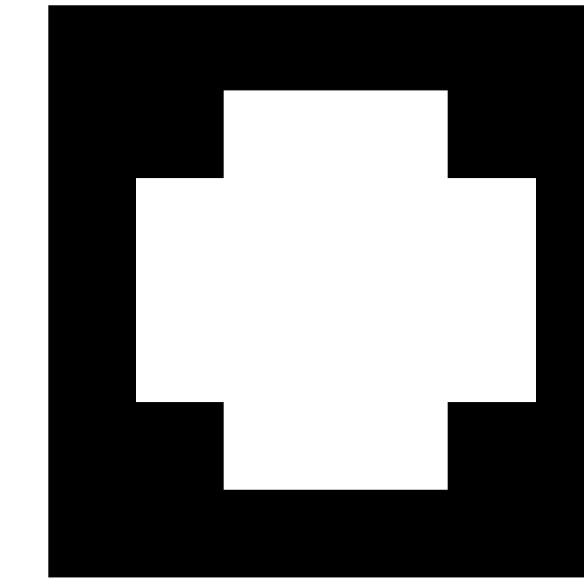
dilation by kernel 1



dilation by kernel 2



dilation by kernel 3



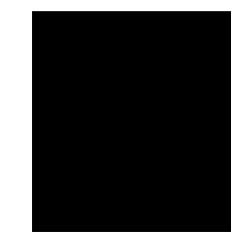
kernel 1: 2 x 2 ones



kernel 2: 2 x 1 ones

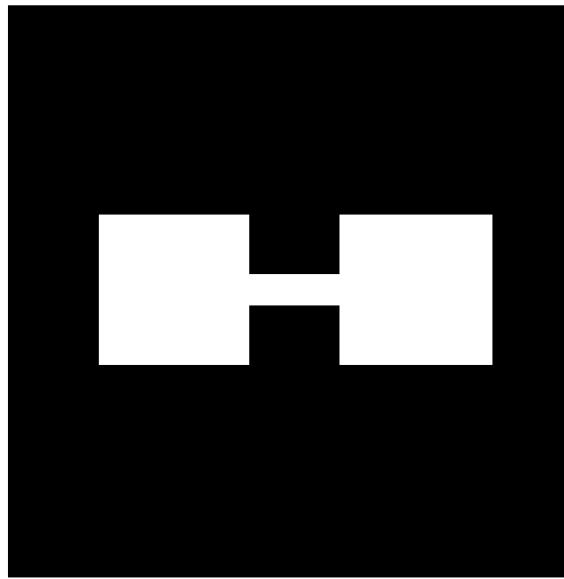


kernel 3: 5 x 5 ones

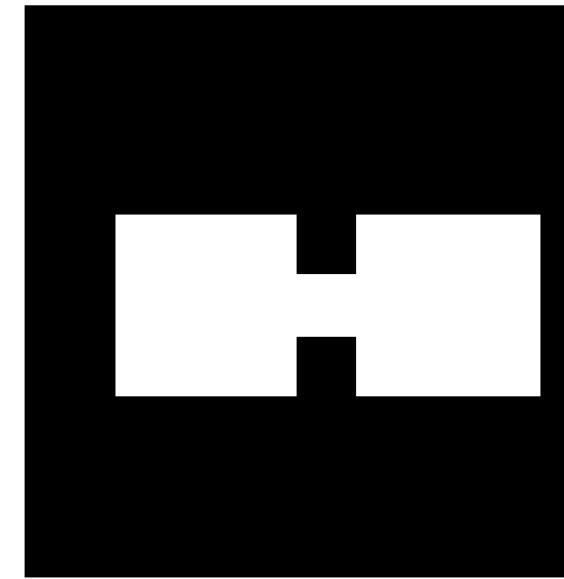


dilation of a dumbbell - two 5x5 squares connected by a 3x1 line

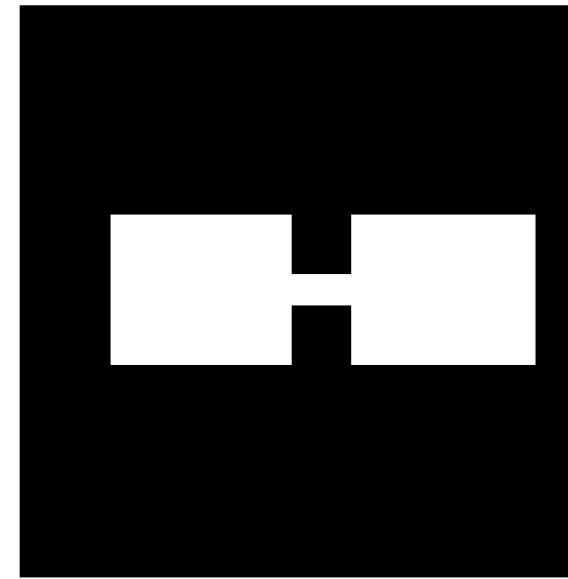
original image



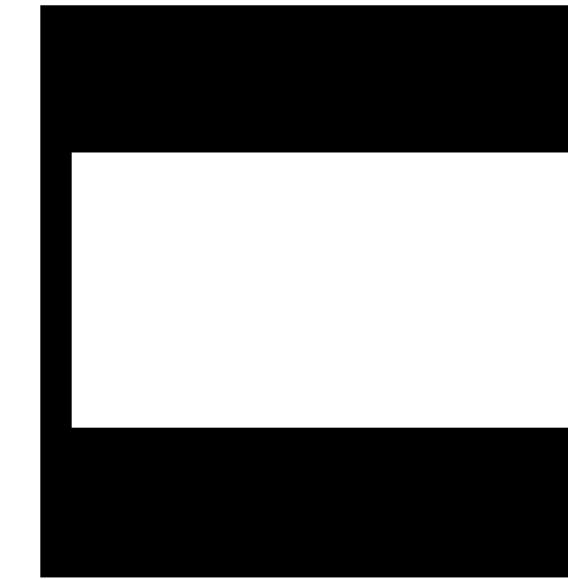
dilation by kernel 1



dilation by kernel 2



dilation by kernel 3



kernel 1: 2 x 2 ones



kernel 2: 2 x 1 ones



kernel 3: 5 x 5 ones

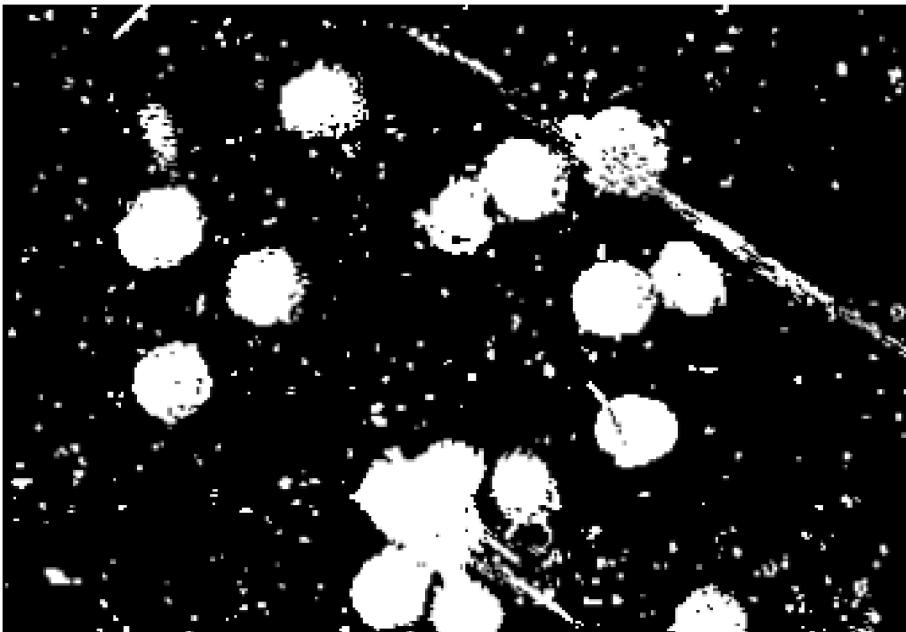


Results and Analysis

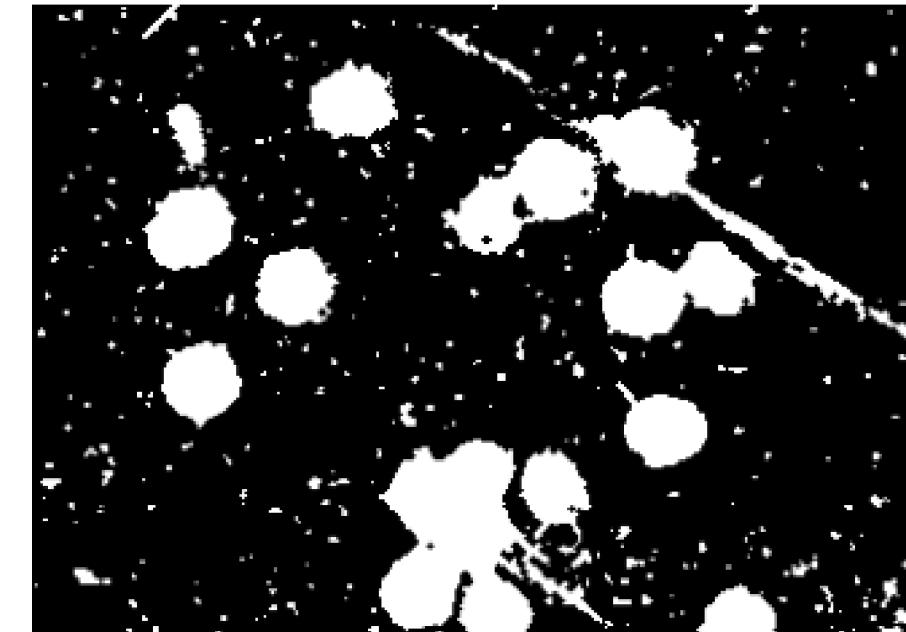
Cleaning Cell Image for Feature Extraction

Clean Up of Cell Images

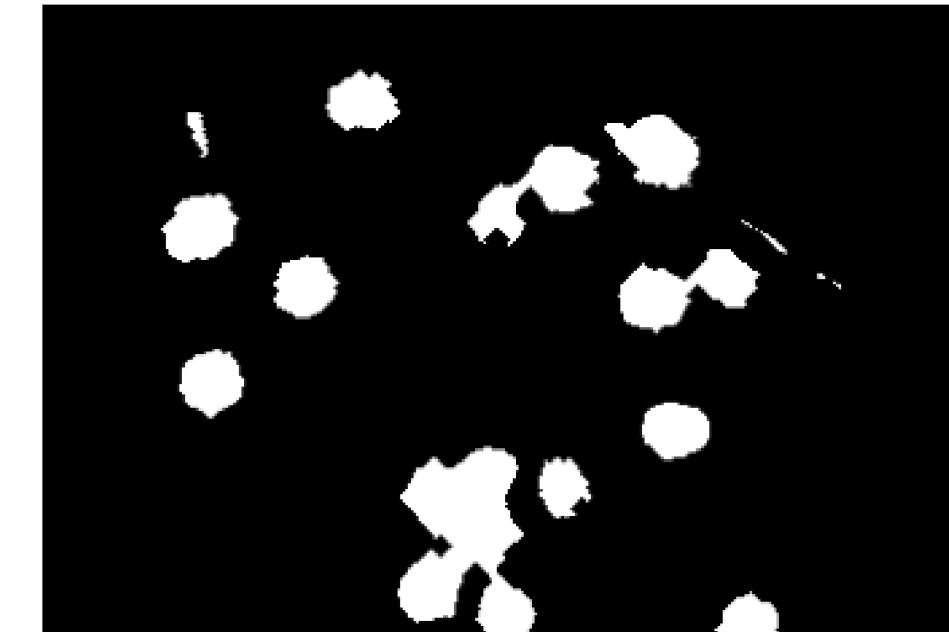
original image



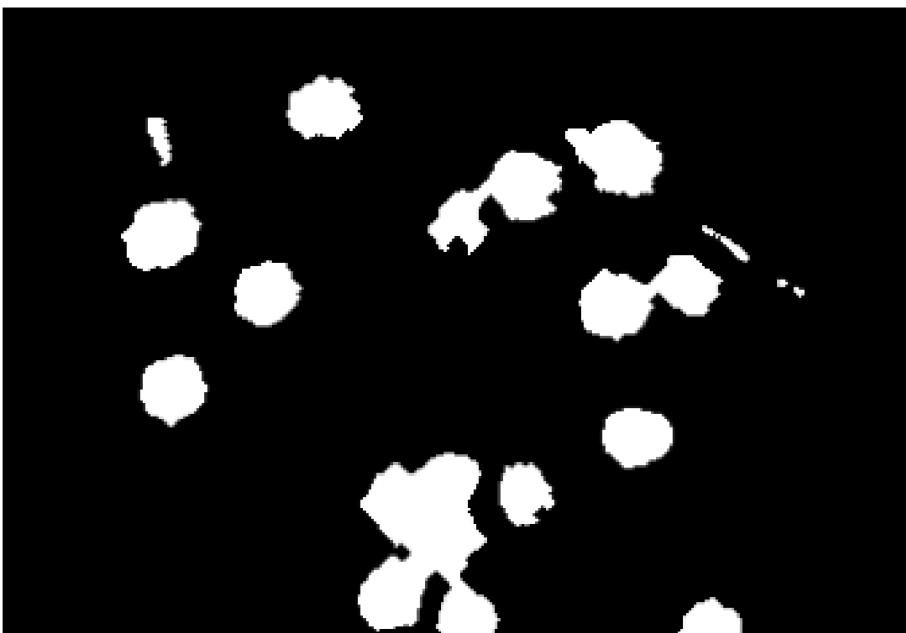
step 1: erosion



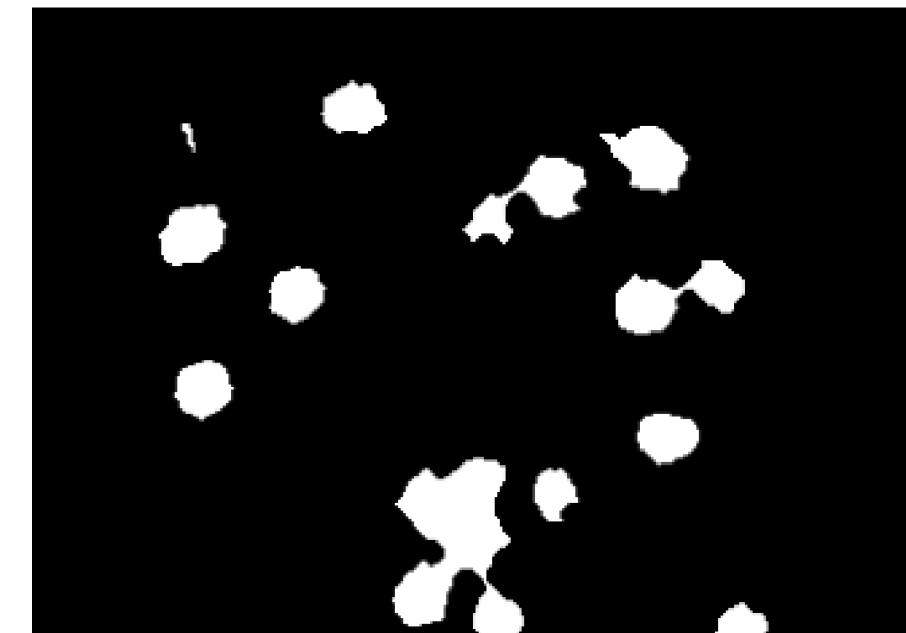
step 2: closing



step 3: erosion

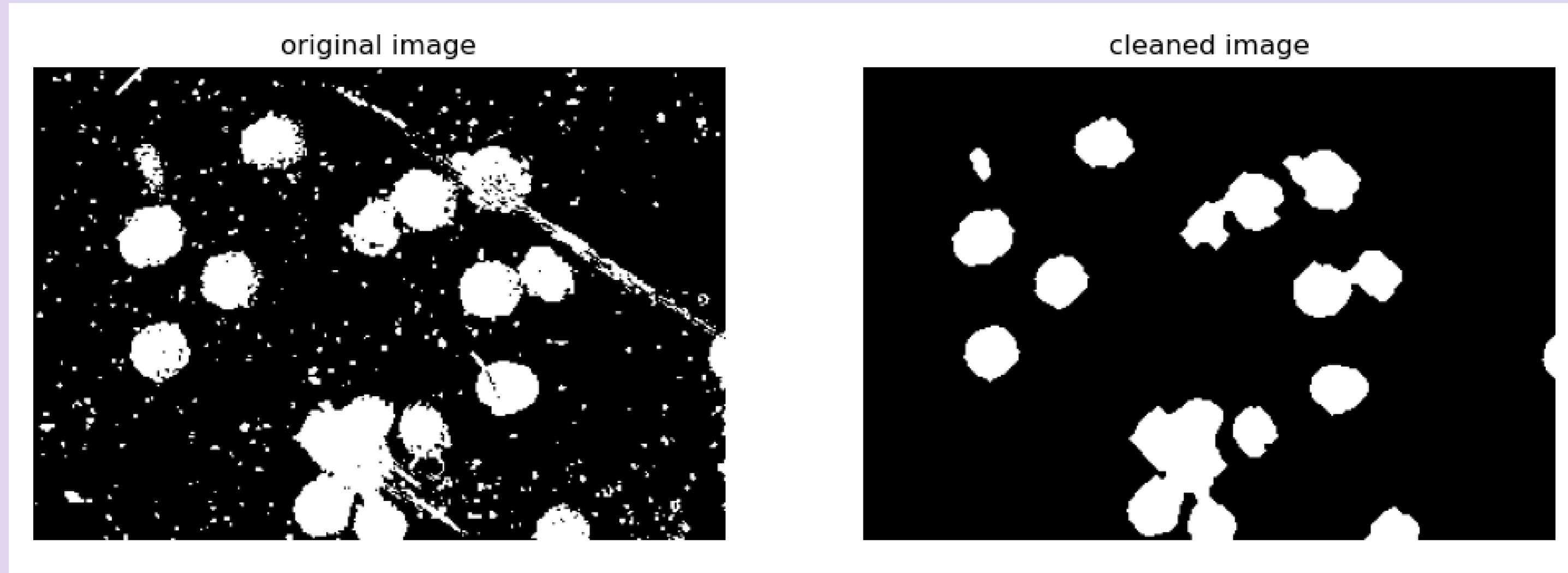


step 4: closing



step 5: dilation





The goal of this portion is to clean the cell image to prepare it for feature extraction. Here, I **made use of a combination of erosion, dilation, and closing morphological operations**. The result of each step and operation is detailed in the next slide.

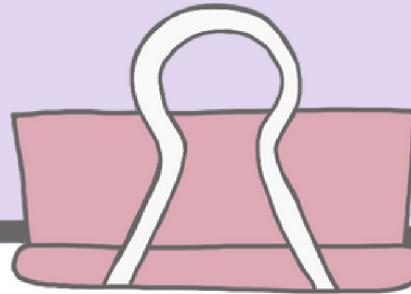
As shown in the results above, we have successfully **cleaned the image**, although some edges of the cells are removed throughout the process. Nevertheless, the **shape and area of the cells appear similar to the original image**.

Reflection

The results of this activity are as expected, as shown by the fact that we have successfully cleaned the cell image using **morphological operations**. It also matched the theoretical results expected from the activity.

This activity actually started poorly for me, cause **performing erosion and dilation would give me the opposite result**, Turns out, my segmentation was wrong (I accidentally switched the 0 and 1 values). Thankfully I caught that mistake, and I'm much happier with my results. I also spent a **very long time trying to determine how to set the structuring element's origin** in cv2. I couldn't find anything online, and thanks to my seat mate Zach Hizon for confirming that it isn't possible in cv2.

Overall, I'm very happy with my results. It was very fun to try different combinations of morphological operations to get the best results.



Self-grading

- Technical Correctness = 35
- Quality of Presentation = 35
- Self-reflecton = 30
- Initiative = 0

Total: 100

References

I would also like to cite and thank the following sources that I used in this activity!

- [1] Soriano, M. (2023). Morphological Operations.

Thanks also to my seat mate Zach Hizon for confirming that there's no way to set the origin of a structuring element in cv2!