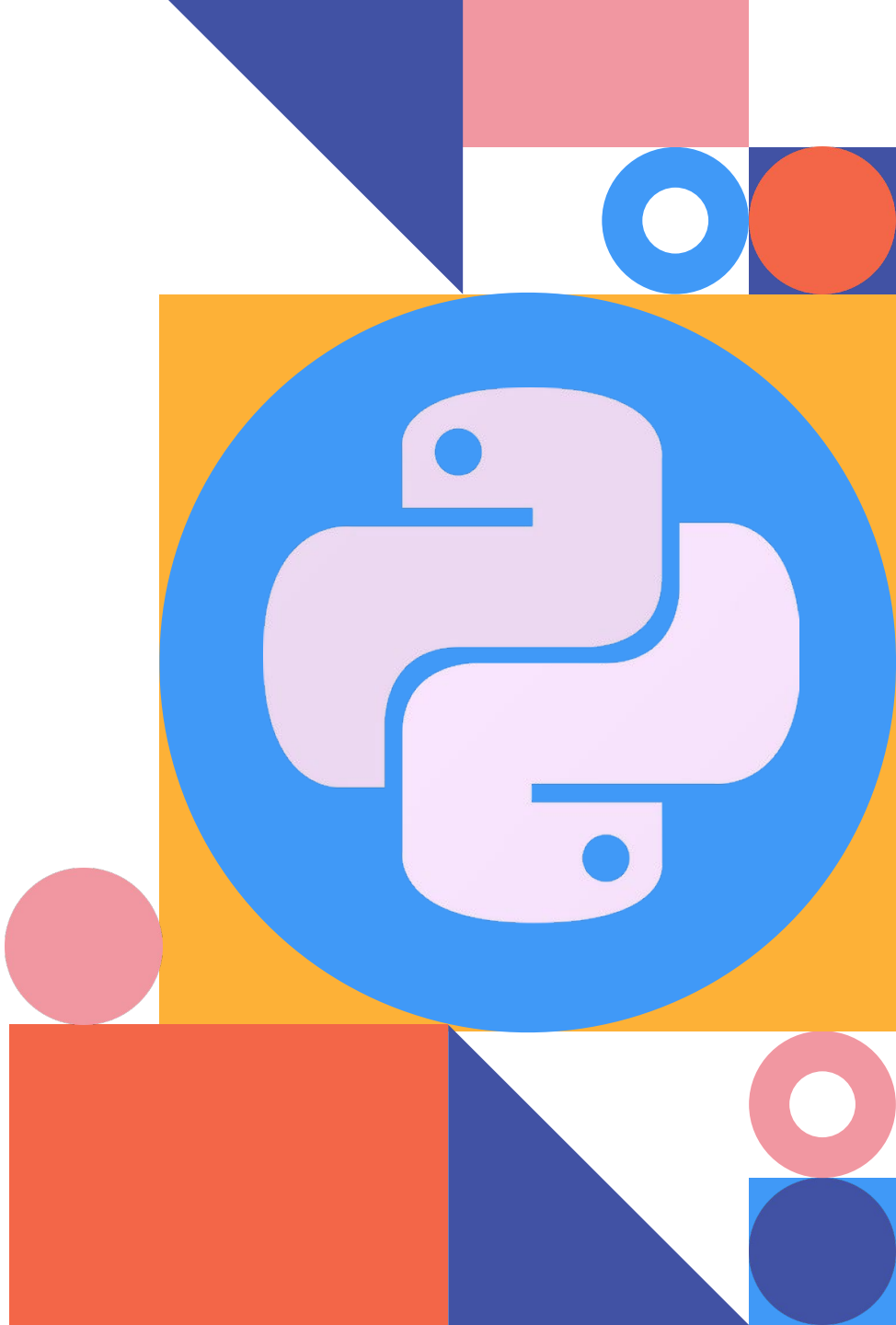
Image Processing III

Yan-Fu Kuo

Dept. of Biomechatronics Engineering

National Taiwan University



Contents

Quick Recap

Import scikit-image

```
import skimage
```

Reading image

```
coins_snp = io.imread('coins_s&p.png')

print(type(coins_snp))
print(coins_snp.shape)
```

Showing image

```
plt.imshow(coins_snp, cmap='gray')
plt.show()
```

01 Spatial Filtering (Cont’d)

Object Detection

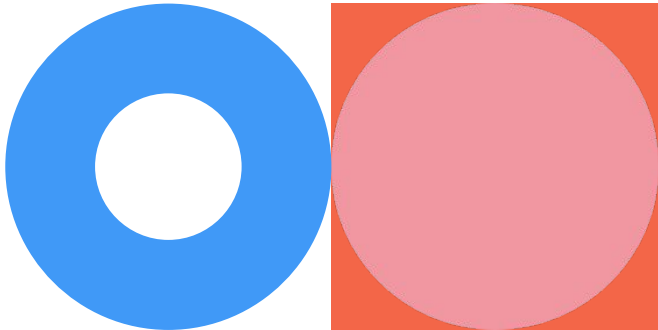
Sobel Filter

-1	0	1
-2	0	2
-1	0	1

Sobel x

1	2	1
0	0	0
-1	-2	-1

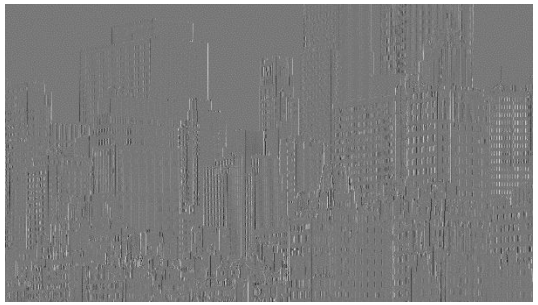
Sobel y



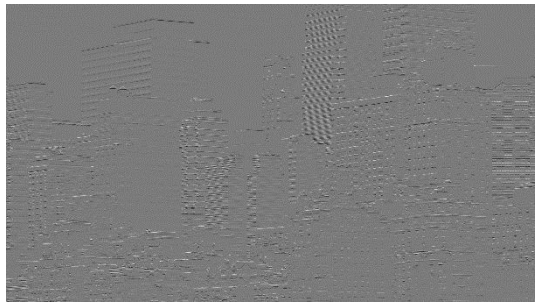
Sobel Filter

```
from skimage import filters

building = io.imread('building.jpg', as_gray=True)

building_sobel_x = filters.sobel_v(building)
building_sobel_y = filters.sobel_h(building)
```

Original image



Sobel x



Sobel y

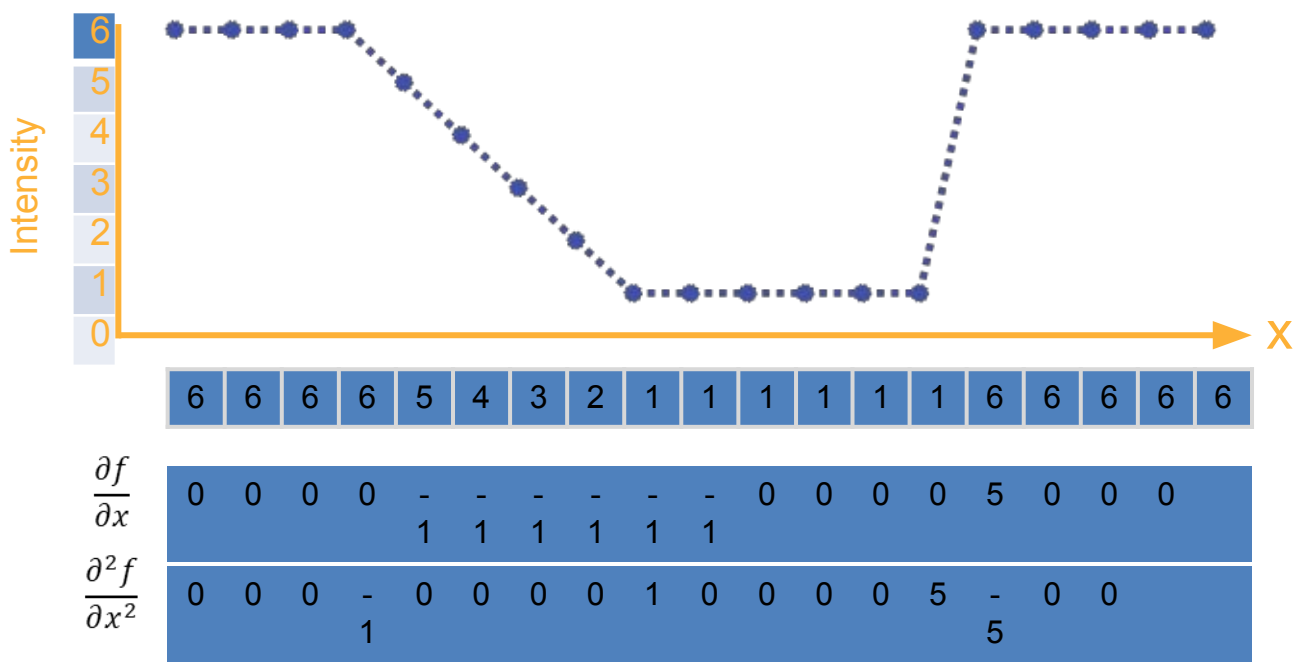
Derivative

- 1st derivative

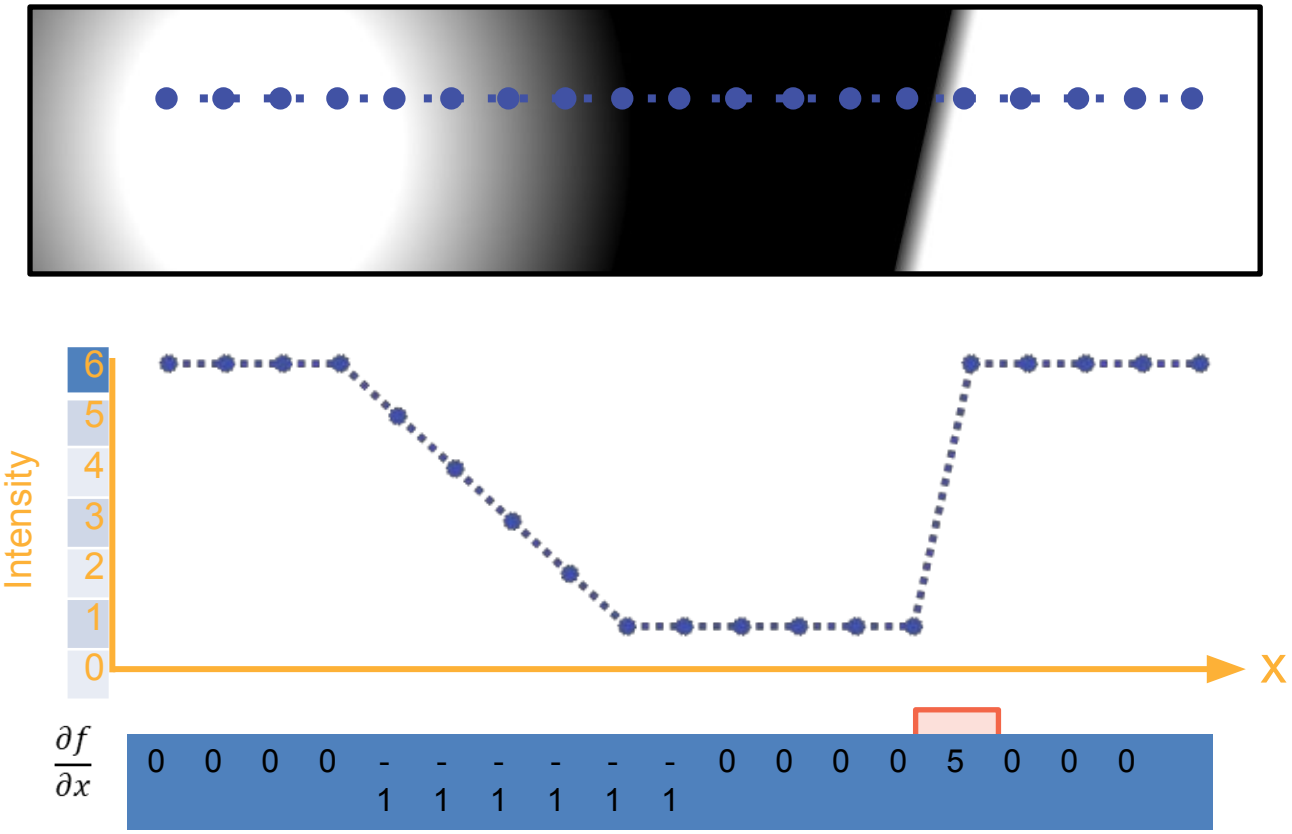
$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

- 2nd derivative

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$



Edges in Image



1st derivative

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

Sobel x filter

-1	0	1
-2	0	2
-1	0	1

Laplacian Filter

0	1	0
1	-4	1
0	1	0

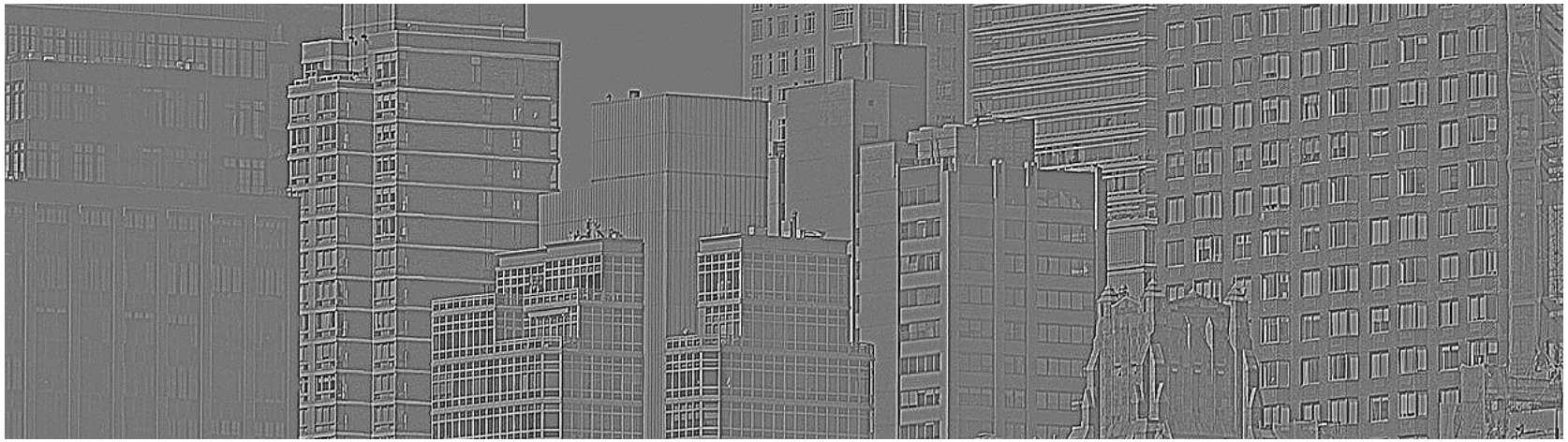
1	1	1
1	-8	1
1	1	1

2^{nd} derivative

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

Laplacian Filter

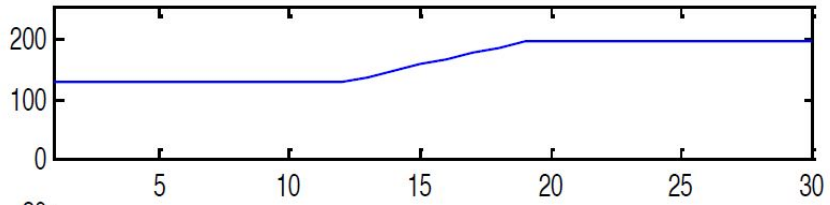
```
building_laplacian = filters.laplace(building)
```



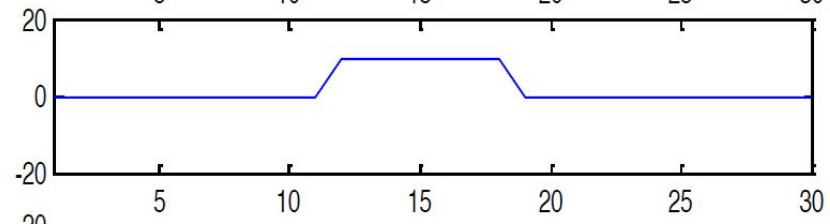
Can Laplacian filter detect round edges?

Edge Enhancement

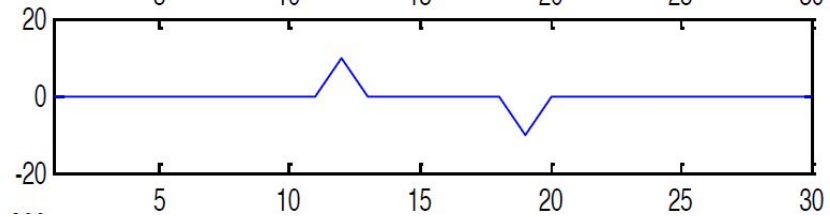
$f(x)$



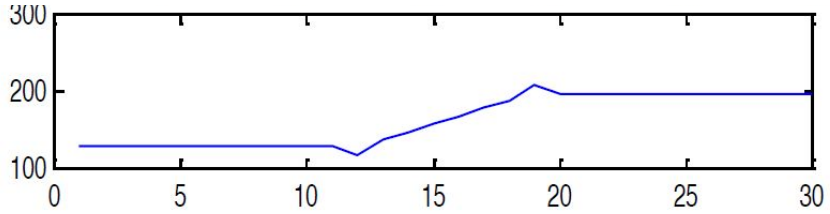
$\frac{df}{dx}$



$\frac{d^2f}{dx^2}$



$f(x) - \frac{d^2f}{dx^2}$



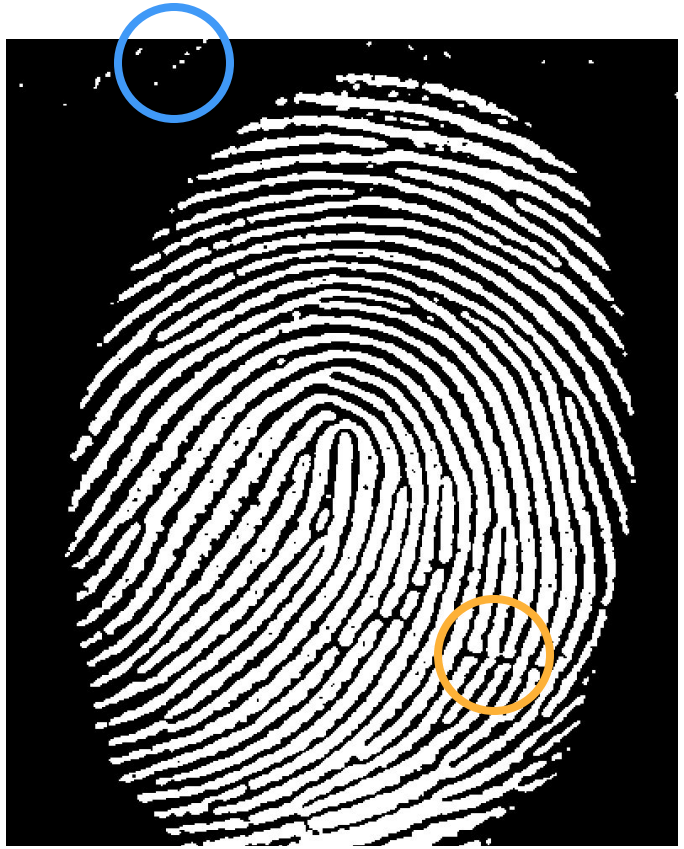
Caution!

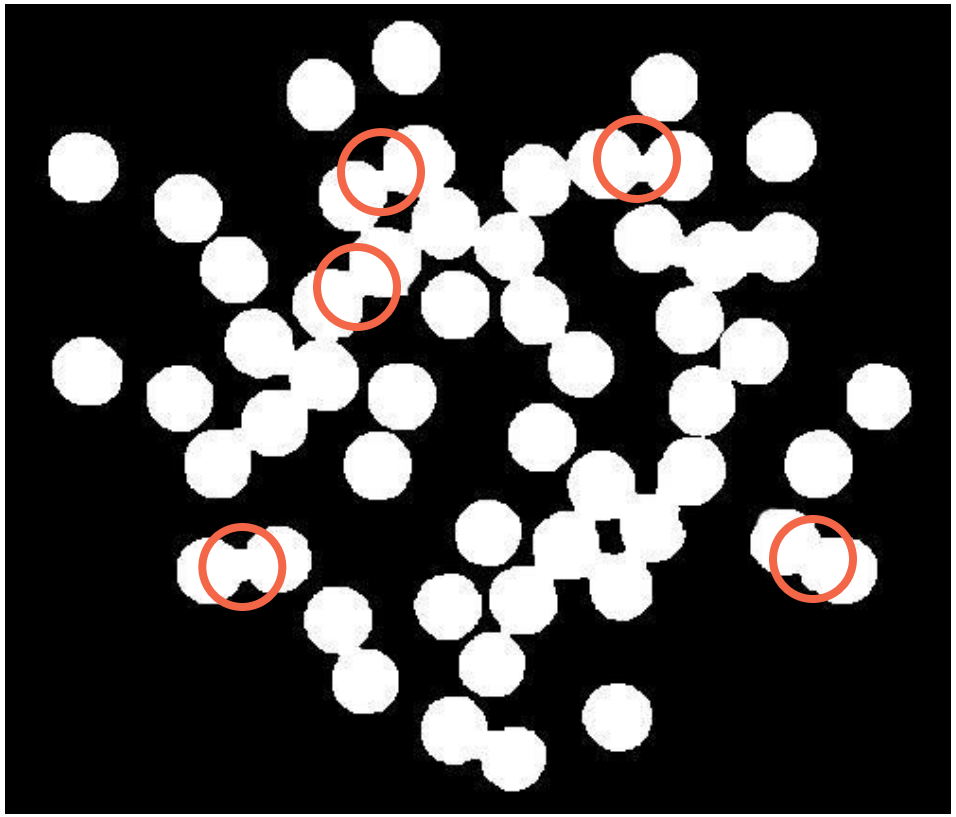
skimage uses the filter below as Laplacian filter

0	-1	0
-1	4	-1
0	-1	0

02 Morphological Image Processing

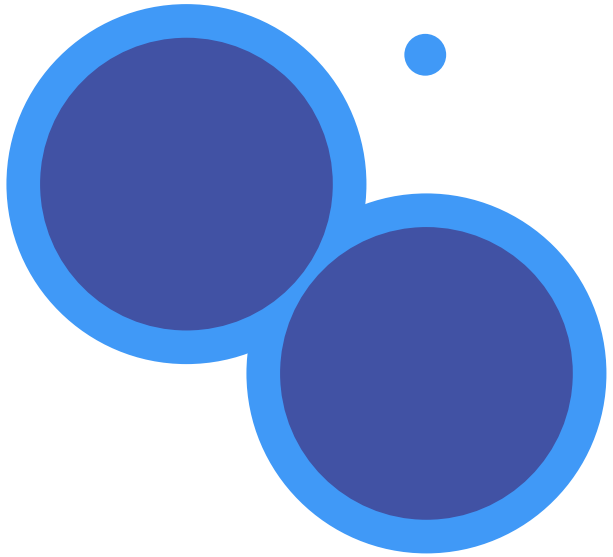
Problem Setup



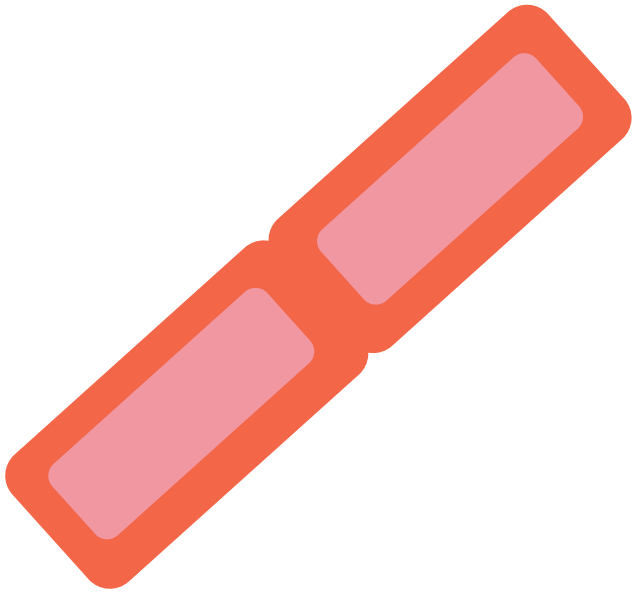


Erosion & Dilation

Erosion

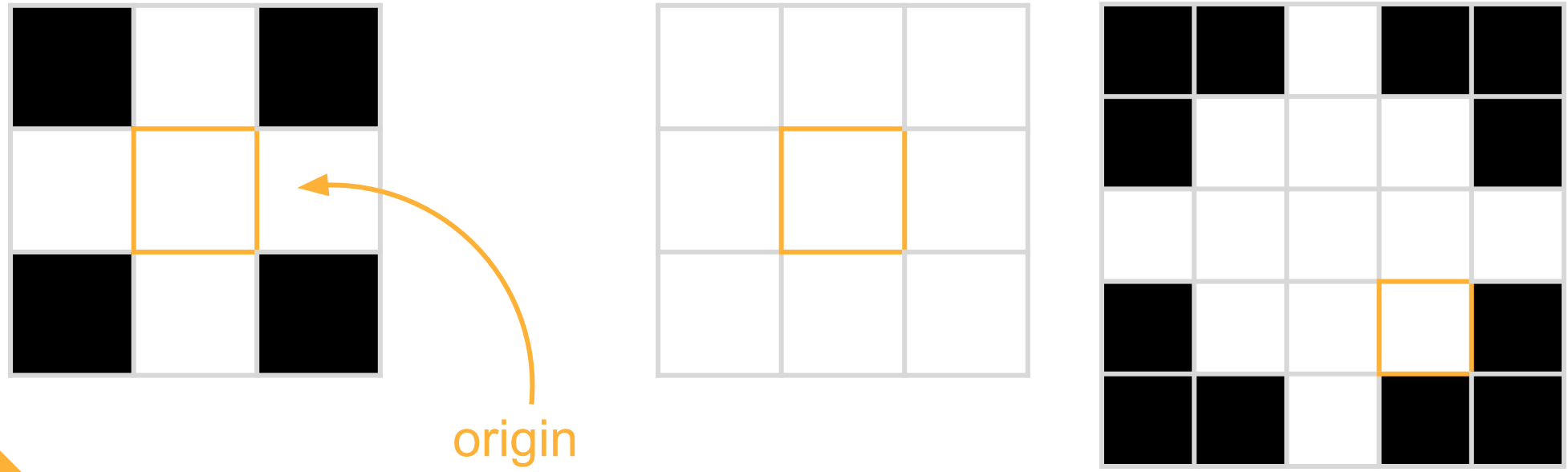


Dilation

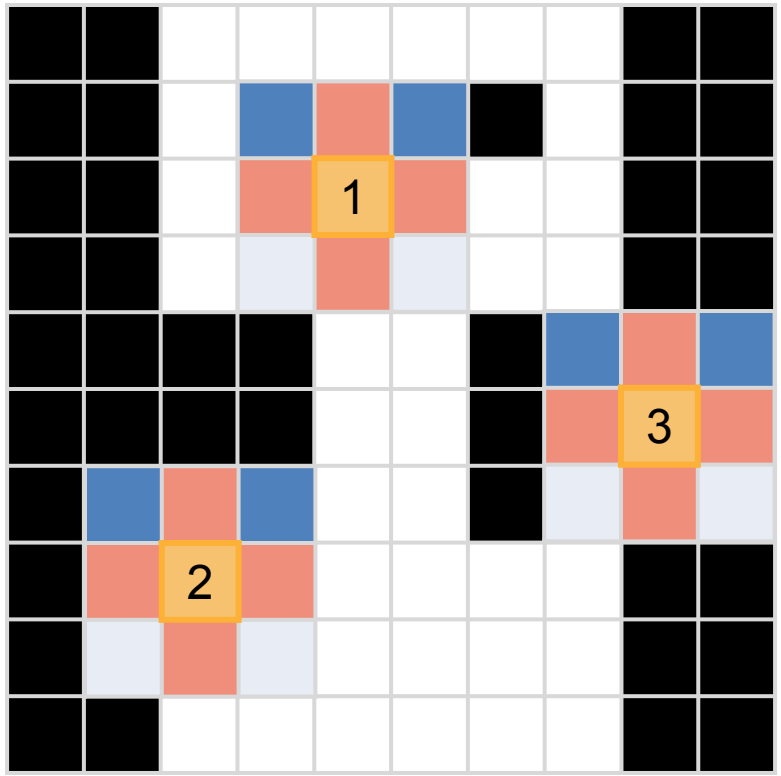


Structuring Elements

```
from skimage.morphology import disk, square, rectangle, diamond
selem = disk(radius=1)
```



Fit, Hit, and Miss



1. Fit

All on pixels in the structuring element superimpose pixels in the image ()

2. Hit

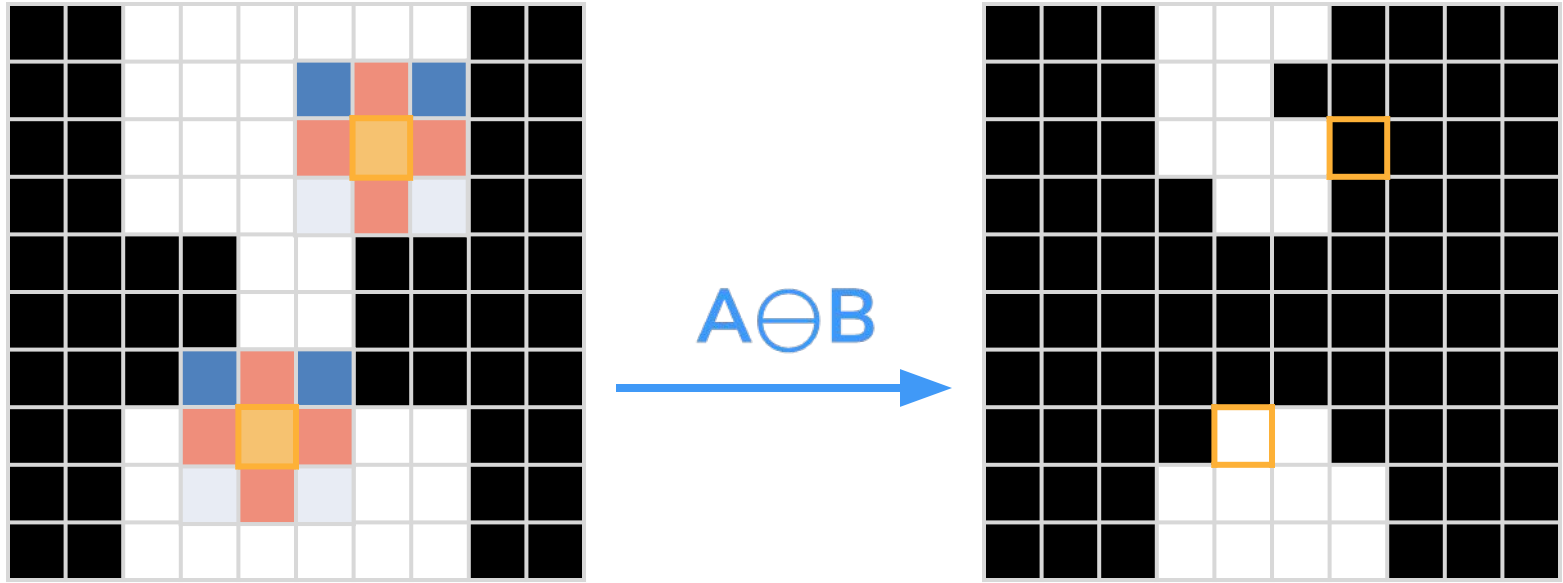
Any on pixel in the structuring element superimpose pixels in the image

3. Miss

No on pixel in the structuring element superimpose pixels in the image

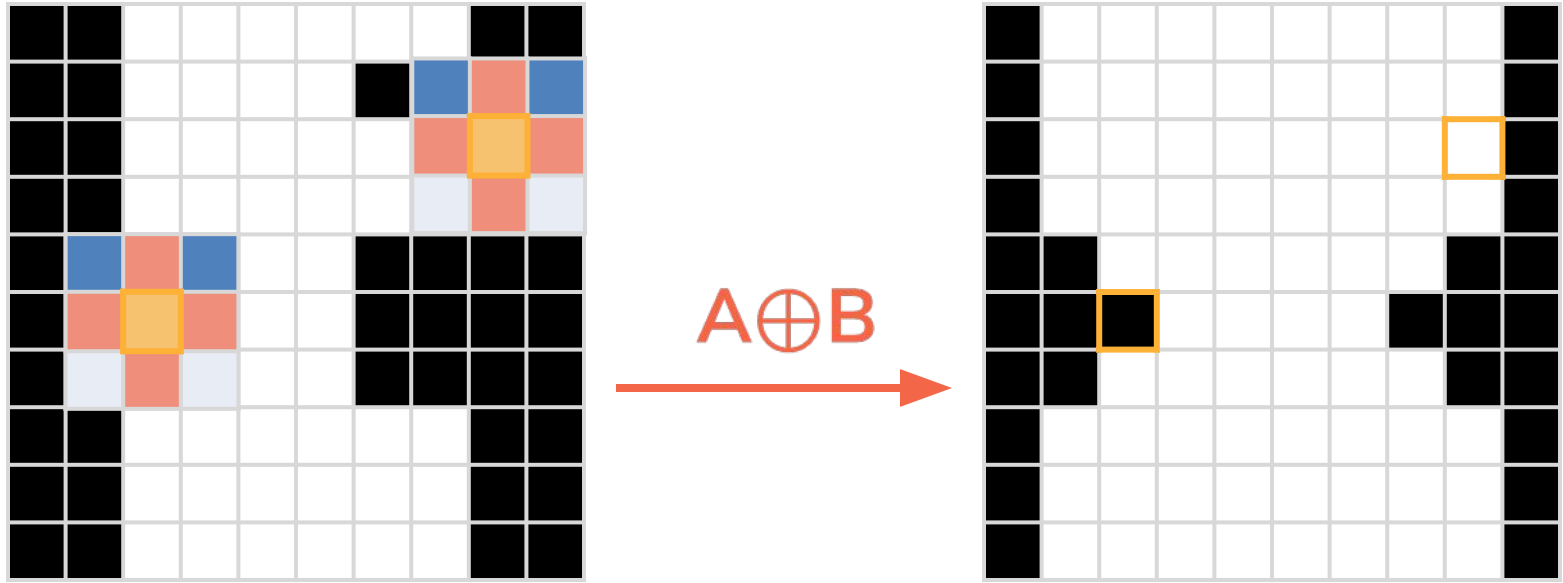
Erosion

For each pixel in A superimpose the origin of B, if B is completely contained by A (**fit**), the pixel is retained, else deleted.

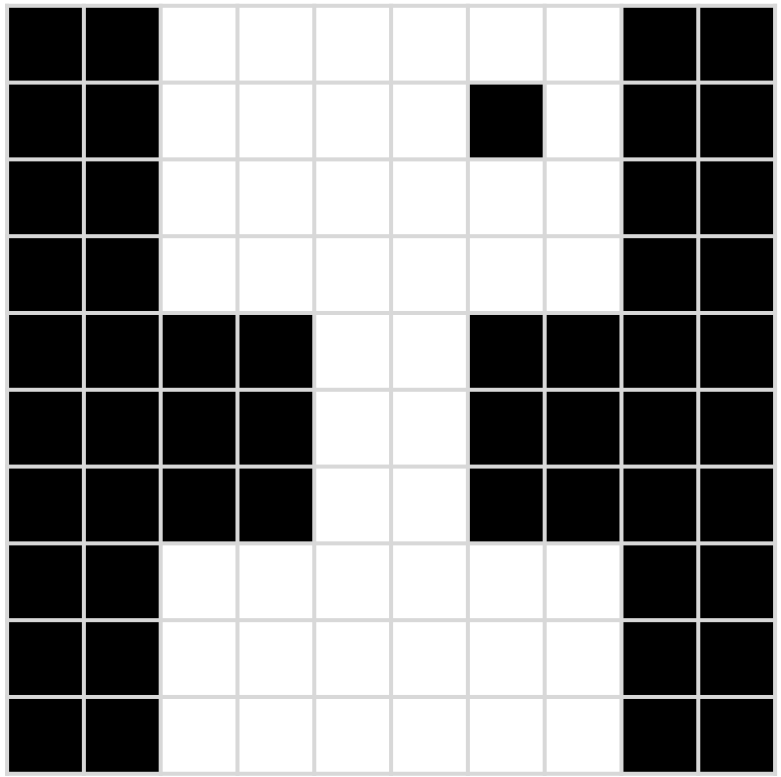


Dilation

For each pixel in A that has a value of 1, superimpose B, with the center of B aligned with the corresponding pixel in A.
Each pixel of every superimposed B is included in the dilation of A by B.

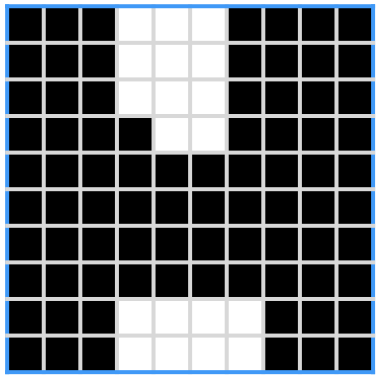


Quiz 1

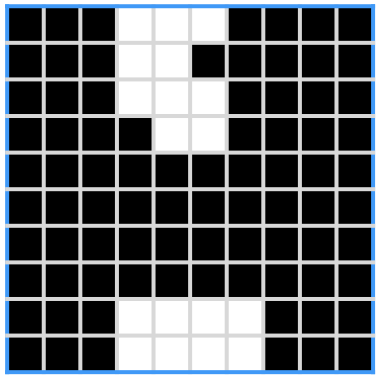


Erosion

(A)

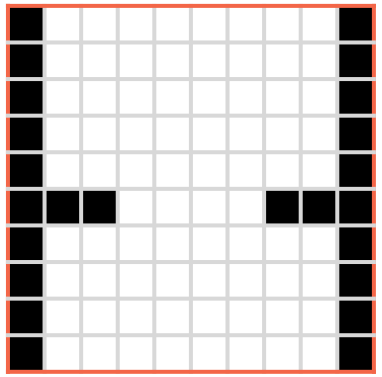


(B)

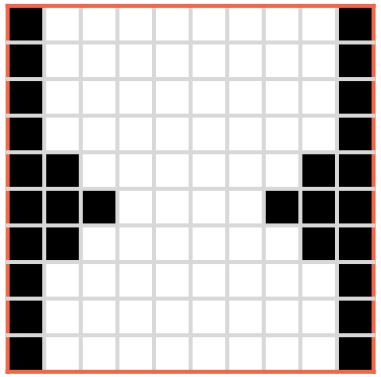


Dilation

(A)



(B)



skimage.morphology

```
from skimage import morphology
from skimage.morphology import disk, square, rectangle, diamond

fingerprint = io.imread('mmbw.jpg')

selem = disk(radius=1)
mm_erosion = morphology.binary_erosion()
```

Also try morphology.binary_dilation

Try changing the structuring element shape

What is the difference between increasing selem size and repeating the erosion operation?

Exercise 1

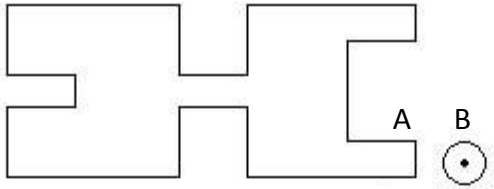
Eliminate the noise by using the erosion operation ($A \ominus B$)

What do you observe on the fingerprint?

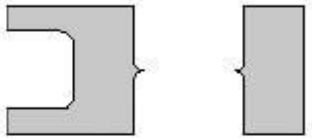
Try performing a dilation operation after the erosion ($(A \ominus B) \oplus B$)

Compound Operations

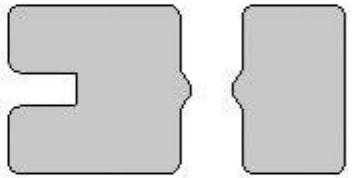
- Opening = erosion → dilation $A \circ B = (A \ominus B) \oplus B$



Original shape

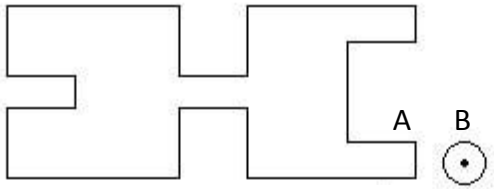


After erosion

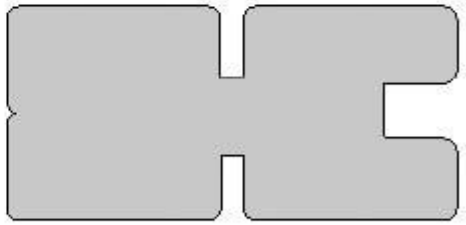


After dilation

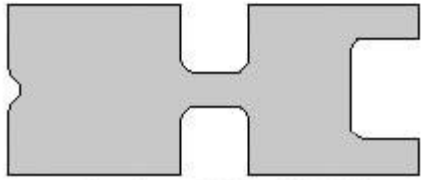
- Closing = dilation → erosion $A \bullet B = (A \oplus B) \ominus B$



Original shape



After dilation



After erosion

More Morphological Operations

`remove_small_objects`

`remove_small_holes`

`skeletonize`

`medial_axis`