Morphology Mathematics

Luca Guarnera, Ph.D.

Research Fellow
luca.guarnera@unict.it

University of Catania
Dipartimento di Economia e Impresa

Morphology Mathematics

- In the field of image processing, the term **mathematical morphology** denotes the study of the geometric structure of an image.

- It is a useful tool for representing and describing the ***shape of a region***. ***Contours, skeleton***, etc. can be extracted.

- It is a mathematical tool initially defined on **binary images** but easily **<u>extended to gray tone and then color images</u>**.

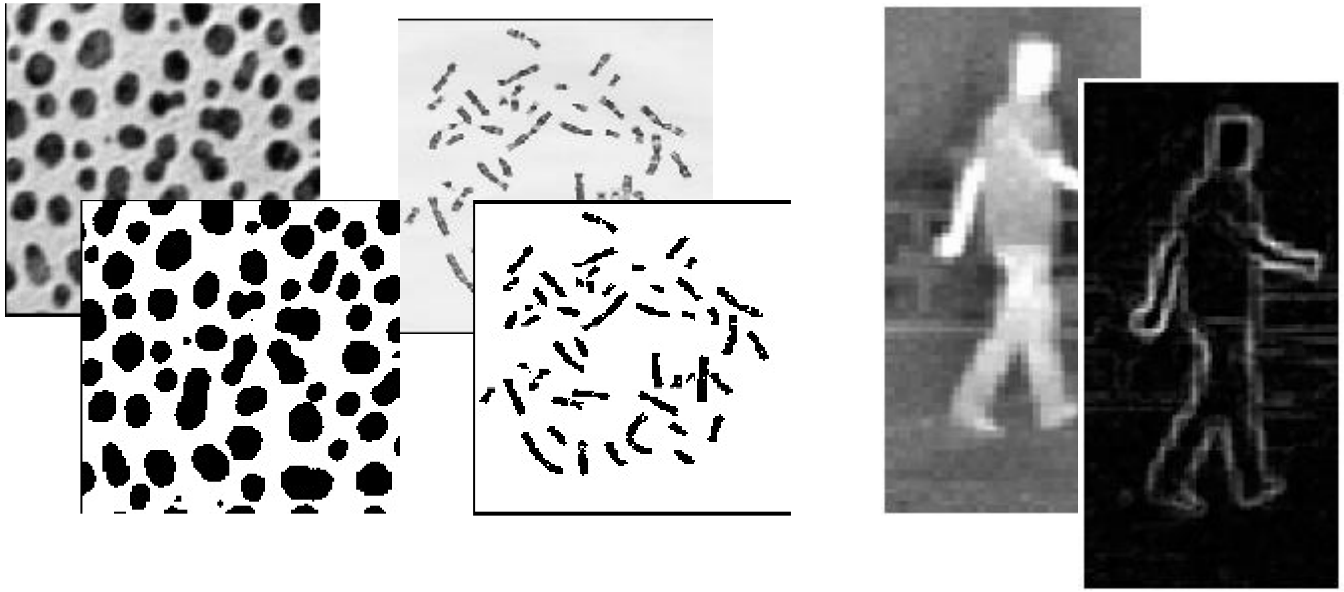
Morphology Mathematics

- **Goal**: ***To distinguish relevant shape information from irrelevant information.***

- Most techniques for analyzing and processing the shape of regions are based on making a shape operator that satisfies the required properties.

Examples

The analysis of an image involves the **extraction of measures characteristic** of the image under consideration.



For example, Geometric measures consist of the position of an "object," orientation, area and length of the perimeter...

Preliminary

- Sets in mathematical morphology represent "**objects**" in an image:

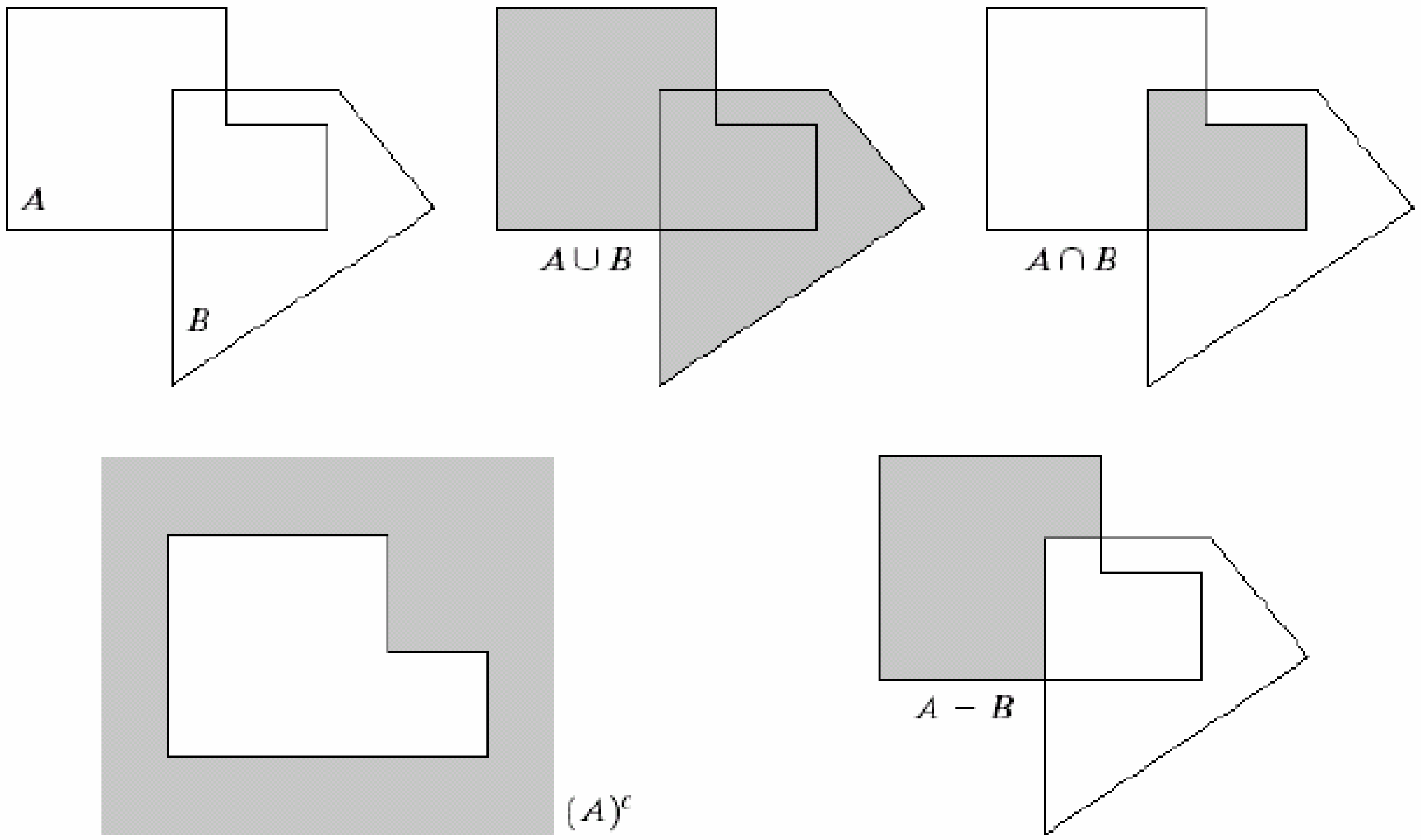
- ***Binary images*** (0 = white, 1 = black): the element of the set corresponds to the (x, y) coordinates of the pixel; The object is defined in Z^2;
- ***Gray tone images***: the set element corresponds to the (x,y) coordinates of the pixel and its intensity value; The object is defined in Z^3;

Preliminaries

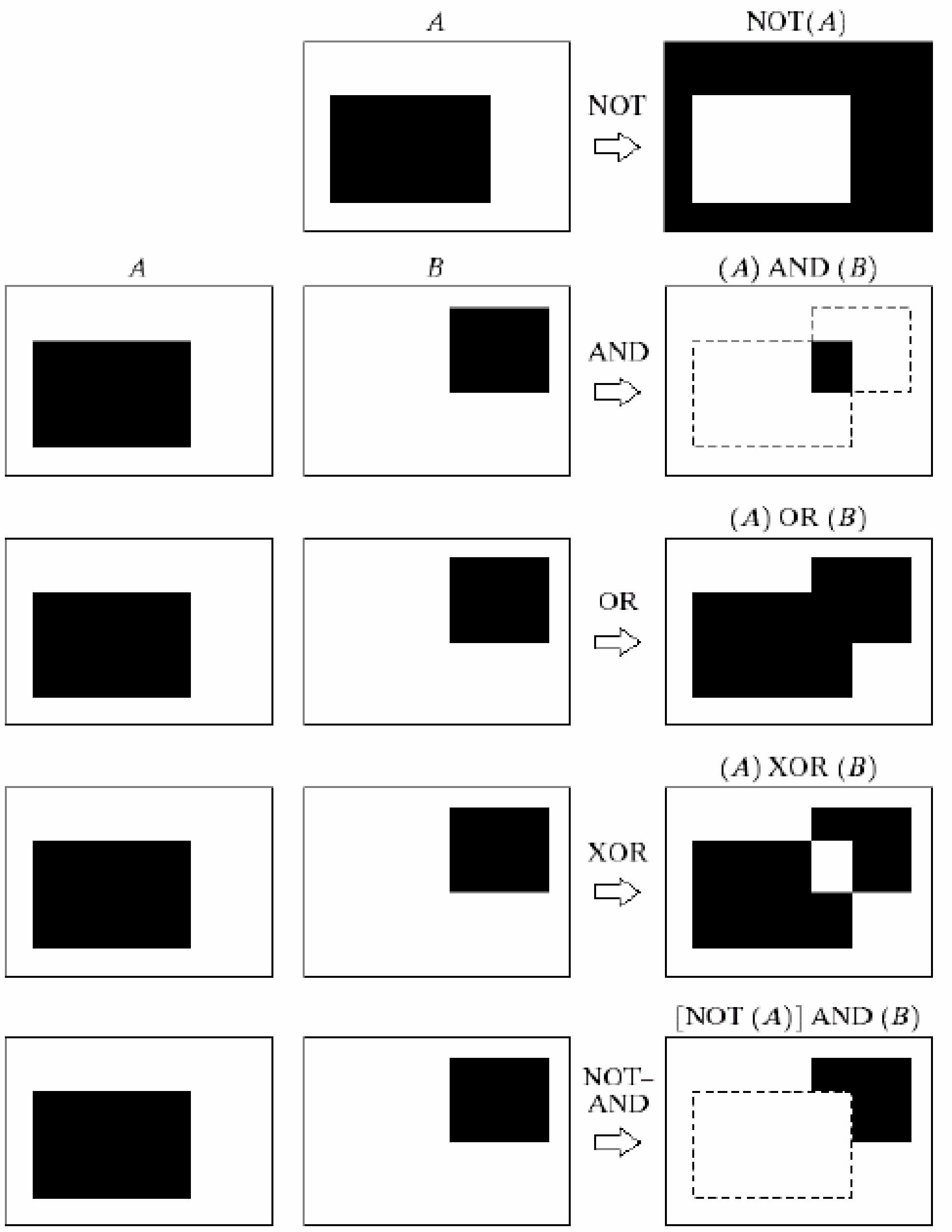
If an element of A is defined as $a=(a_1,a_2)$ the following expressions are well defined:

$a \in \mathrm{A}$	*a* belongs to the set A;
$a \notin \mathrm{A}$	*a* does not belong to the set A;
$\mathrm{A} \subseteq \mathrm{B}$	A is included in B;
$C = \mathrm{A} \cup \mathrm{B}$	Union;
$C = \mathrm{A} \cap \mathrm{B}$	Intersection;
$\mathrm{A} \cap \mathrm{B} = \varnothing$	Empty intersection;
$\mathrm{A}^c = \{w \mid w \notin \mathrm{A}\}$	Complementary of A;
$\mathrm{A} - B = \{w \mid w \in \mathrm{A}, w \notin \mathrm{B}\} = \mathrm{A} \cap \mathrm{B}^c$	Insiemistic difference;

Examples



Logical operations

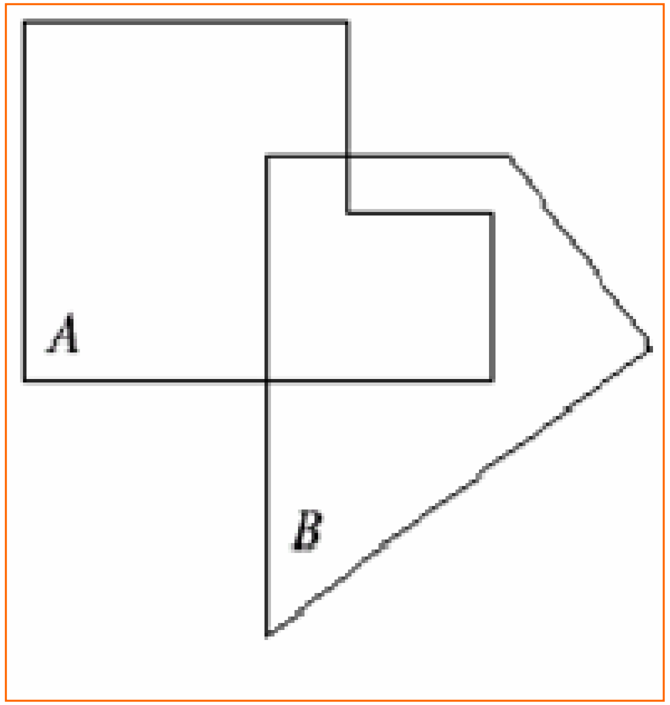


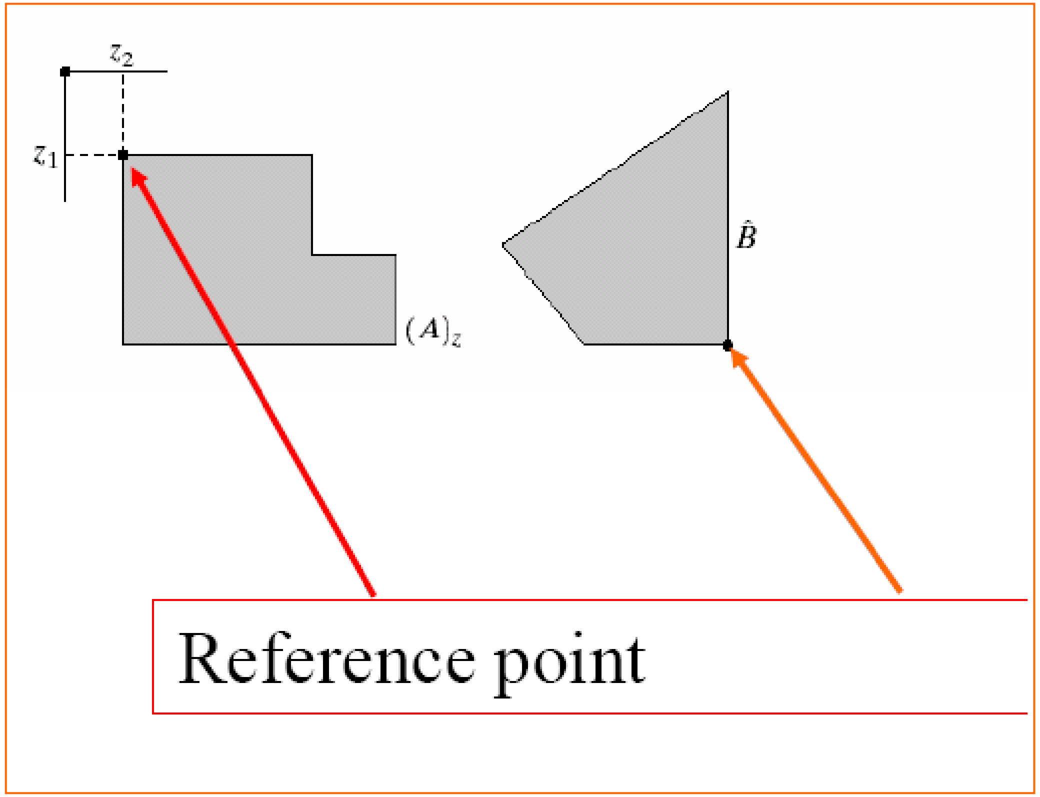
Reflection and Translation

Let A and B be sets in Z^2

$$\hat{B} = \{ w \mid w = -b, \ \forall\, b \in B \}, \ \textit{Riflessione dell'insieme } B$$

$$(A)_z = \{ w \mid w = a + z, \ \forall\, a \in A \}, \quad \textit{Traslazione dell'insieme } A$$





Structuring element

The image structure is "probed" with a user-definable shape set (**structuring element**) usually encoded by a small raster image (3×3 or 5×5).

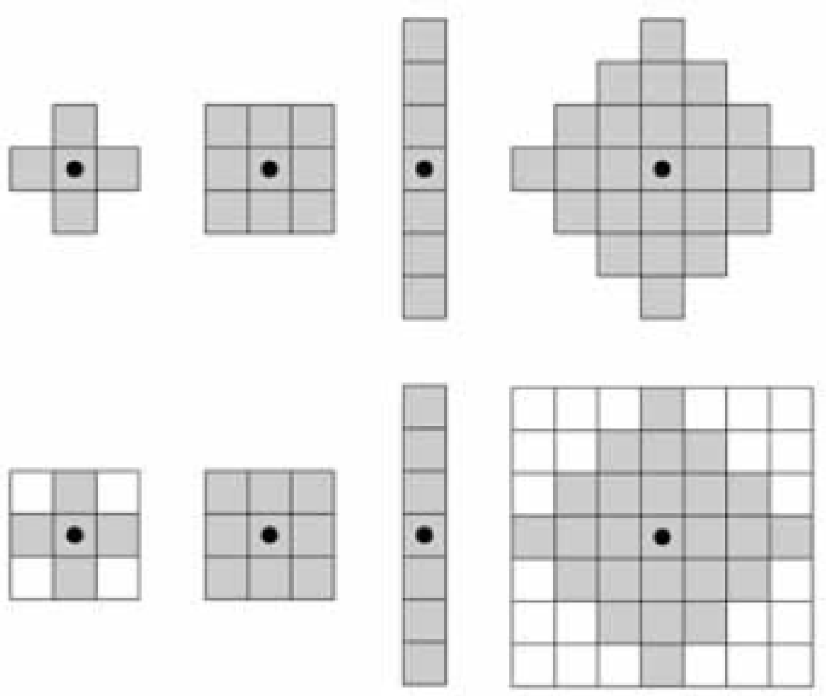


Figura 9.2

Prima riga: esempi di elementi strutturanti. Seconda riga: elementi strutturanti trasformati in matrici rettangolari. I punti indicano i centri degli SE.

Università di Catania

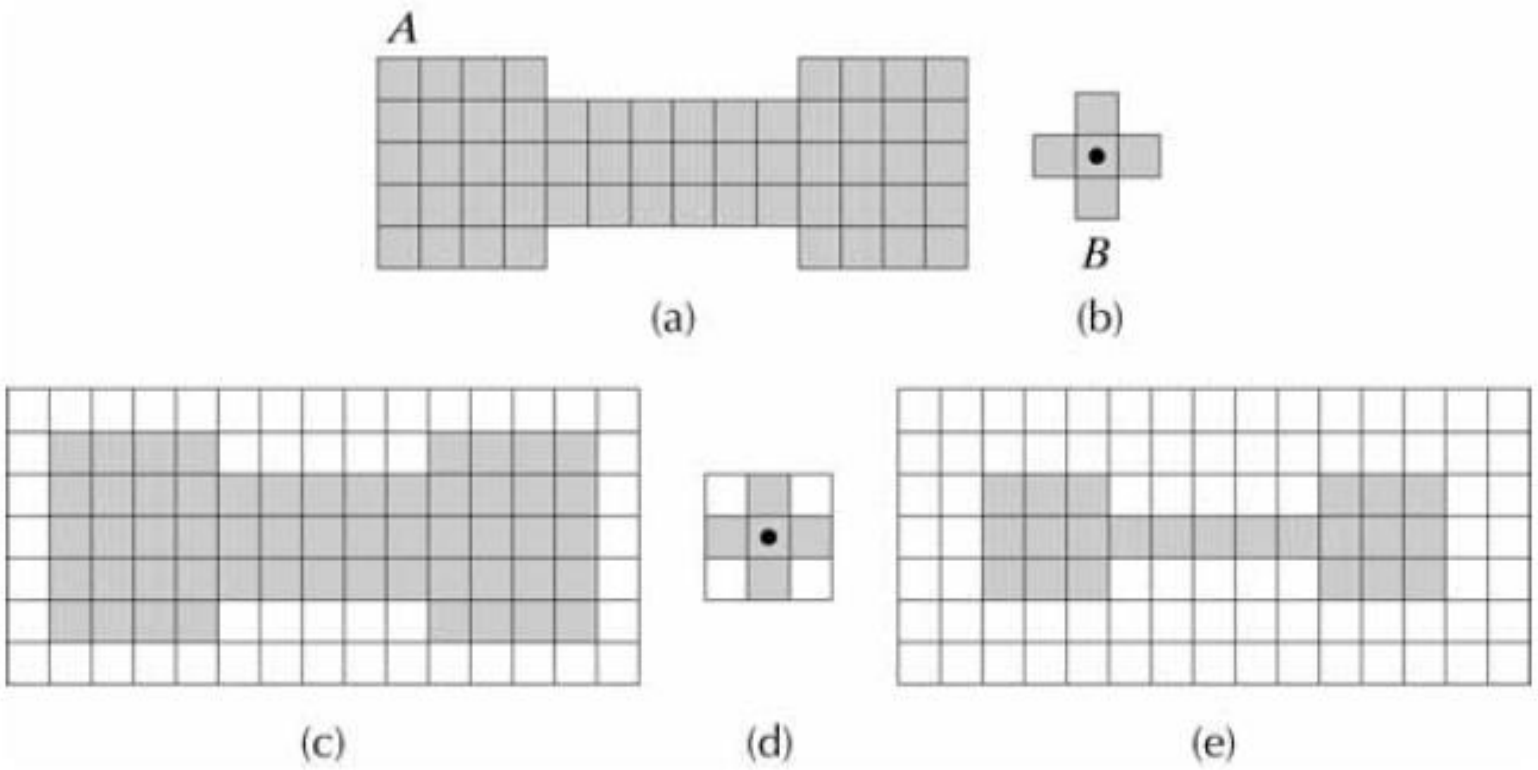
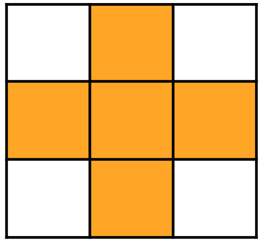
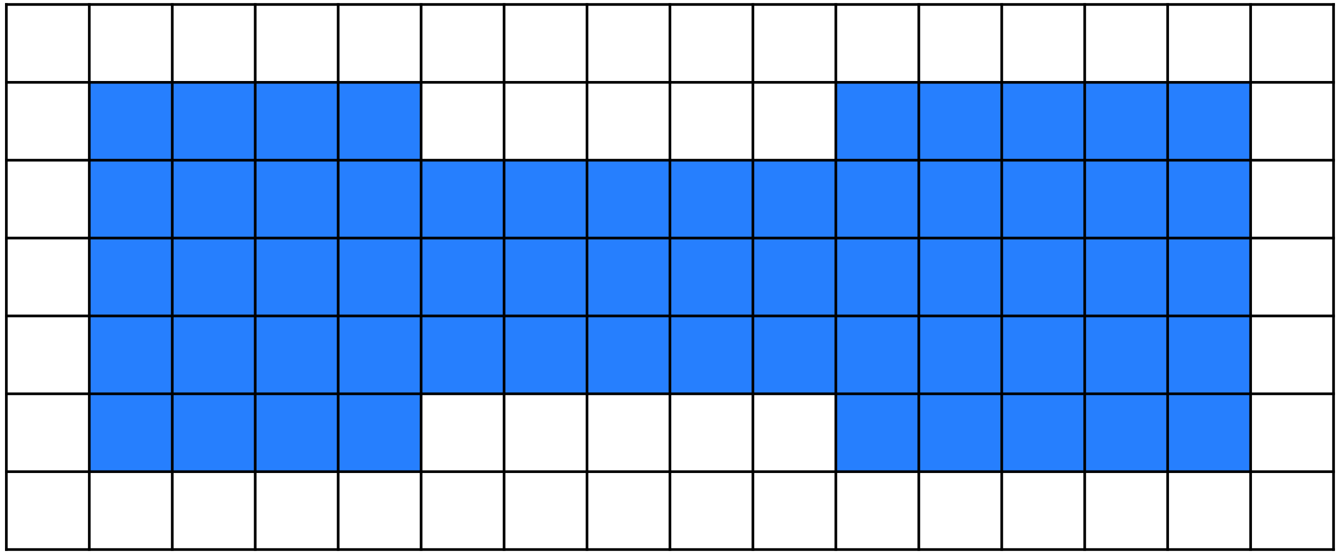


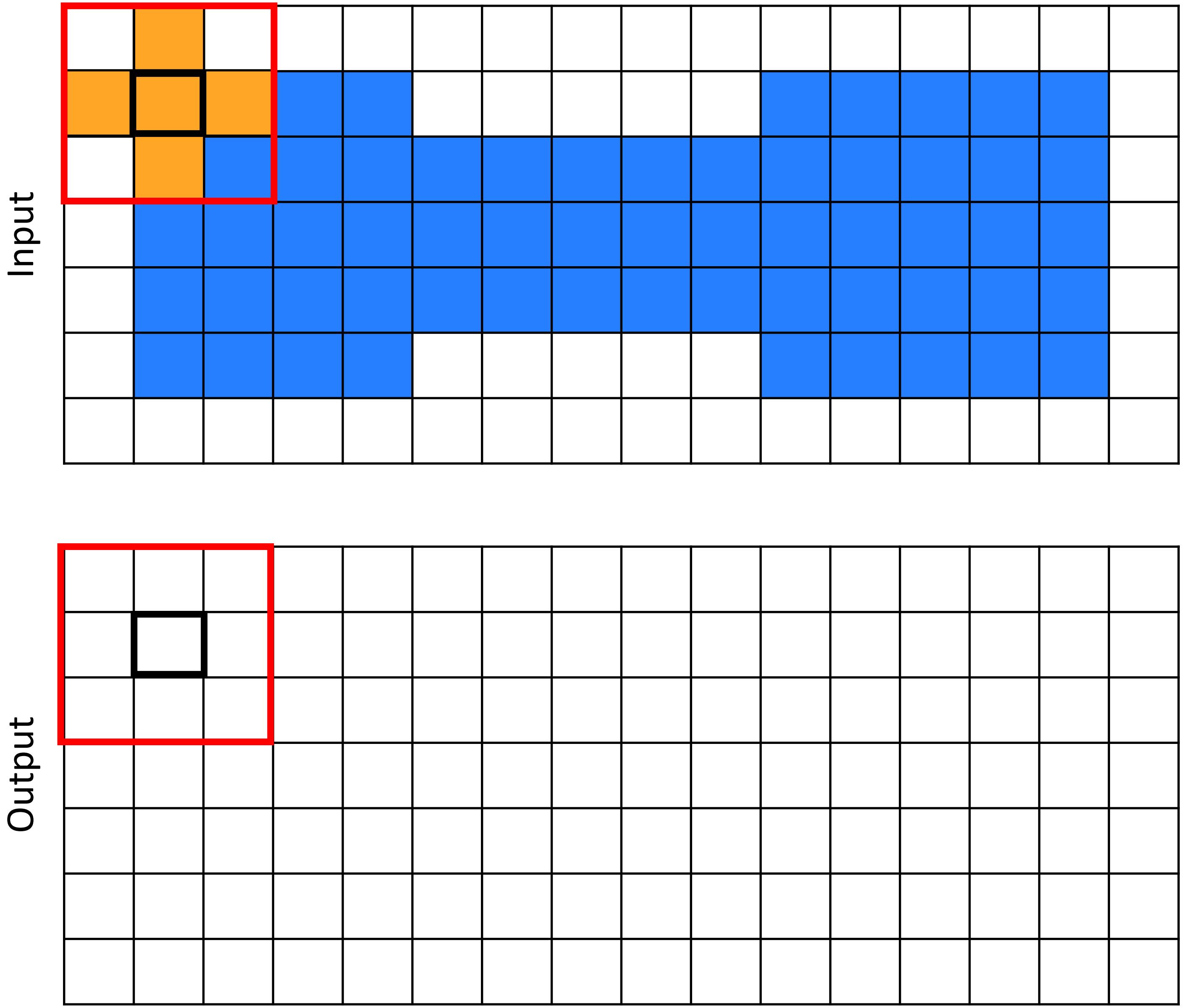
Figura 9.3 (a) Un insieme (ogni quadrato ombreggiato è un membro dell'insieme). (b) Un elemento strutturante. (c) L'insieme riempito con elementi di sfondo per formare una matrice rettangolare rispetto allo sfondo. (d) Elemento strutturante come matrice rettangolare. (e) Insieme elaborato da un elemento strutturante.

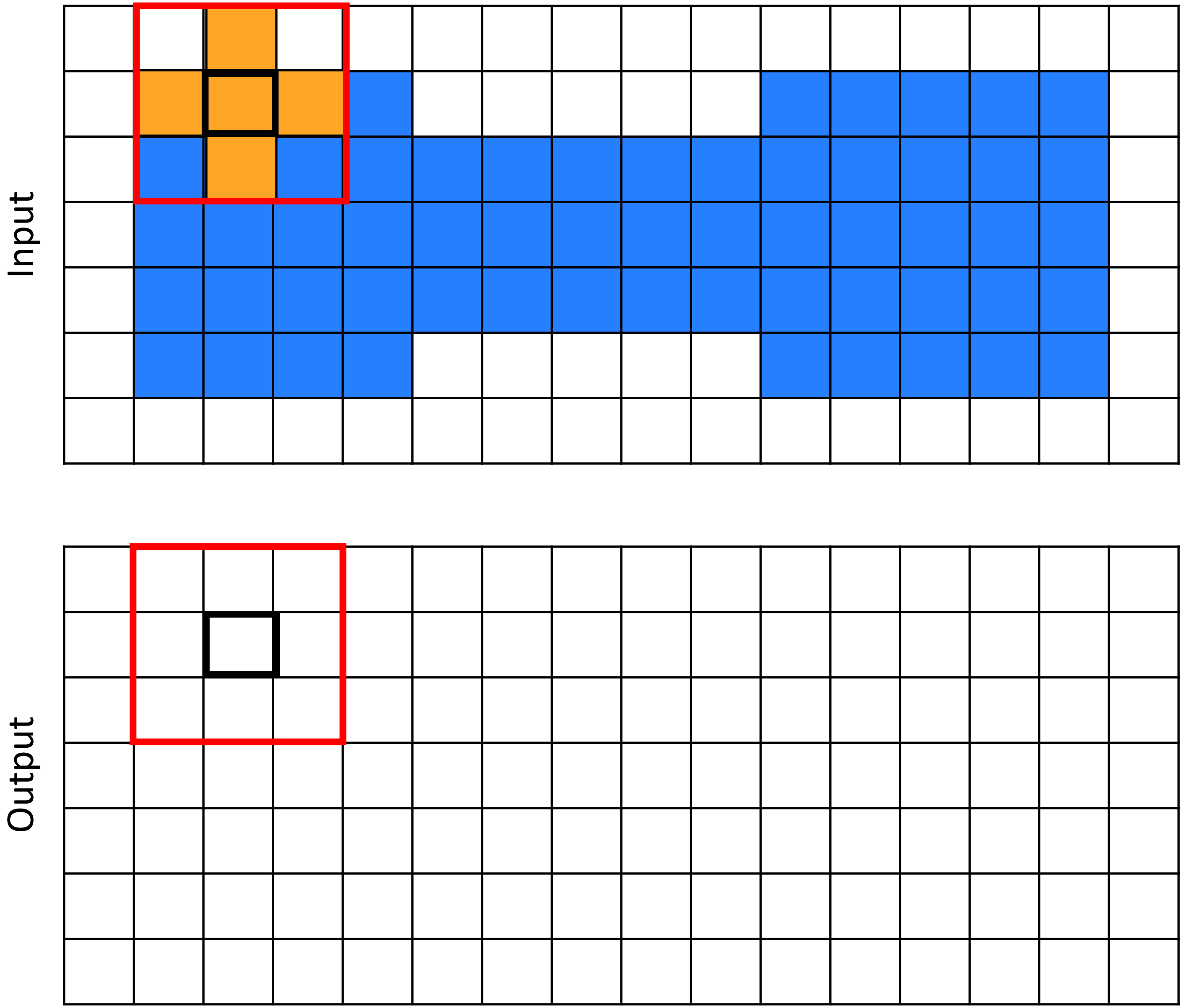


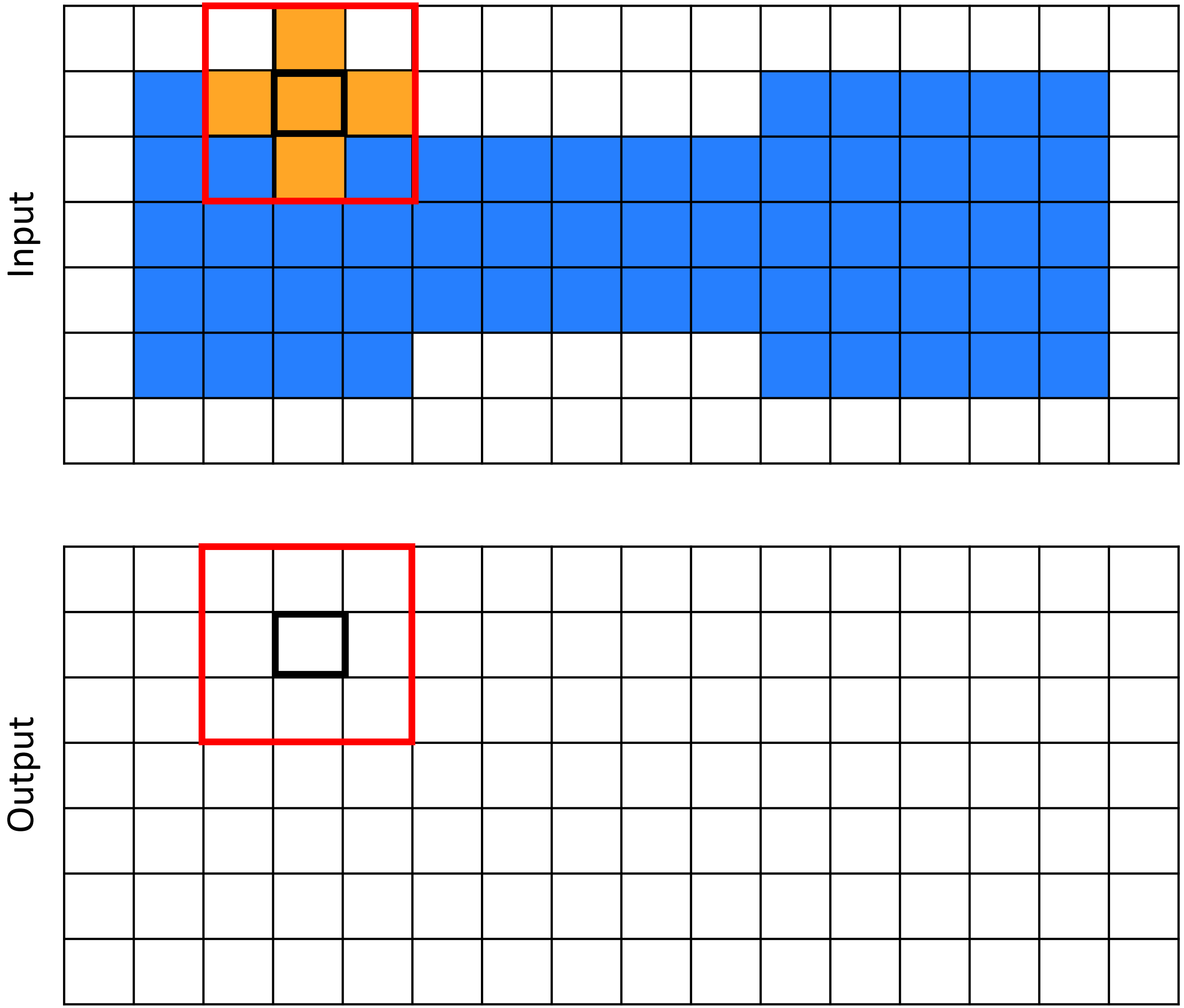
Structuring element

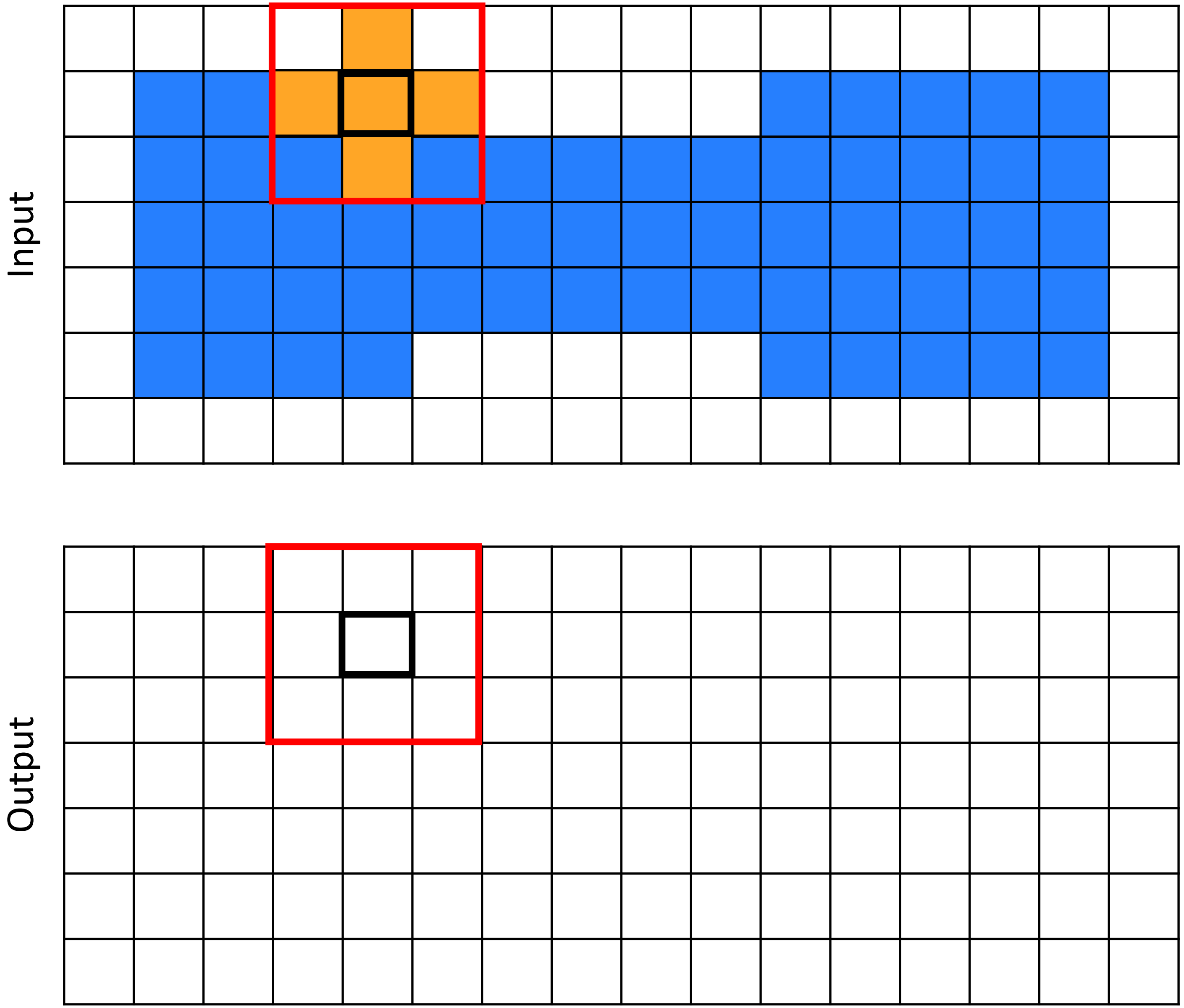
Input

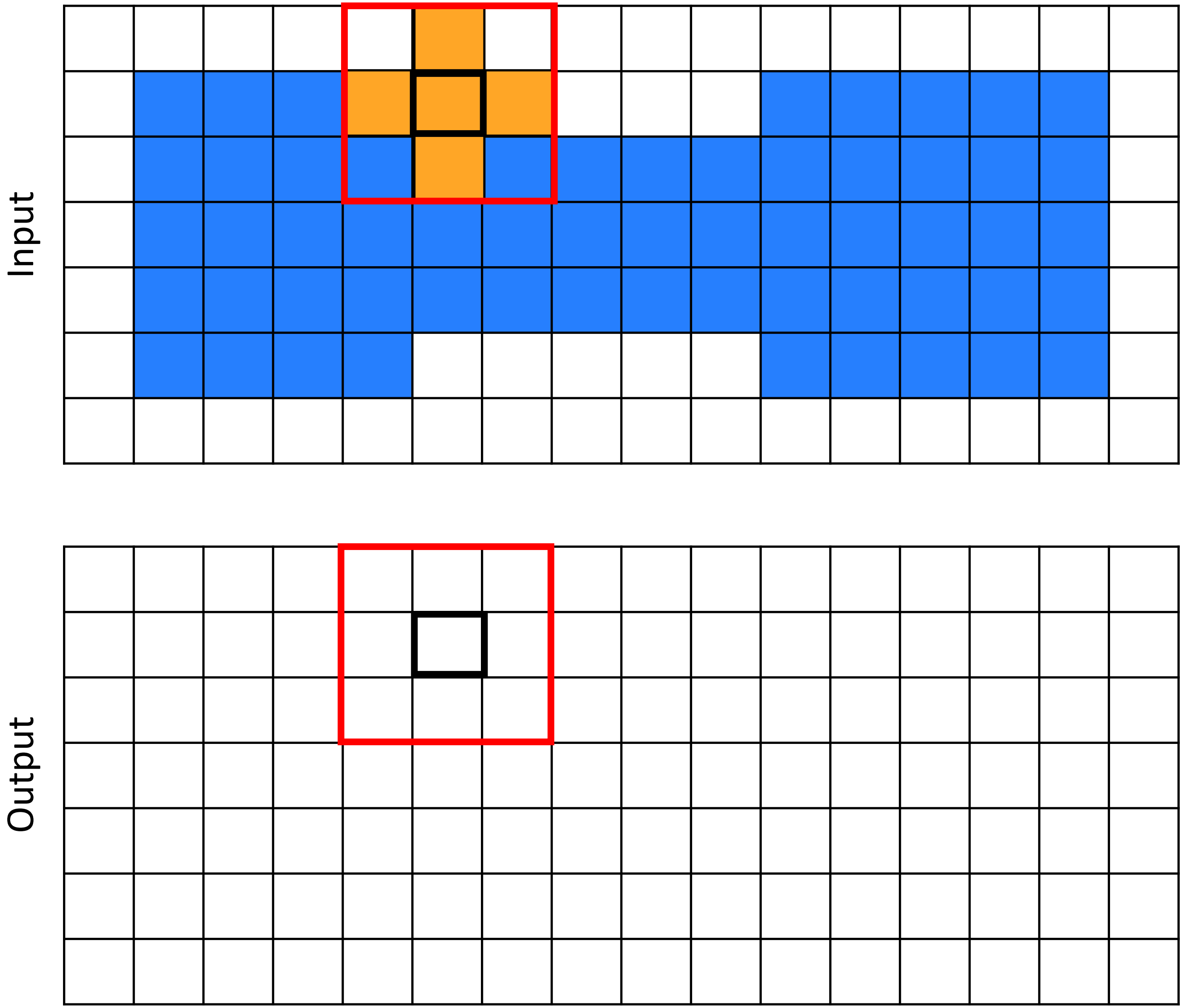



Input
Output

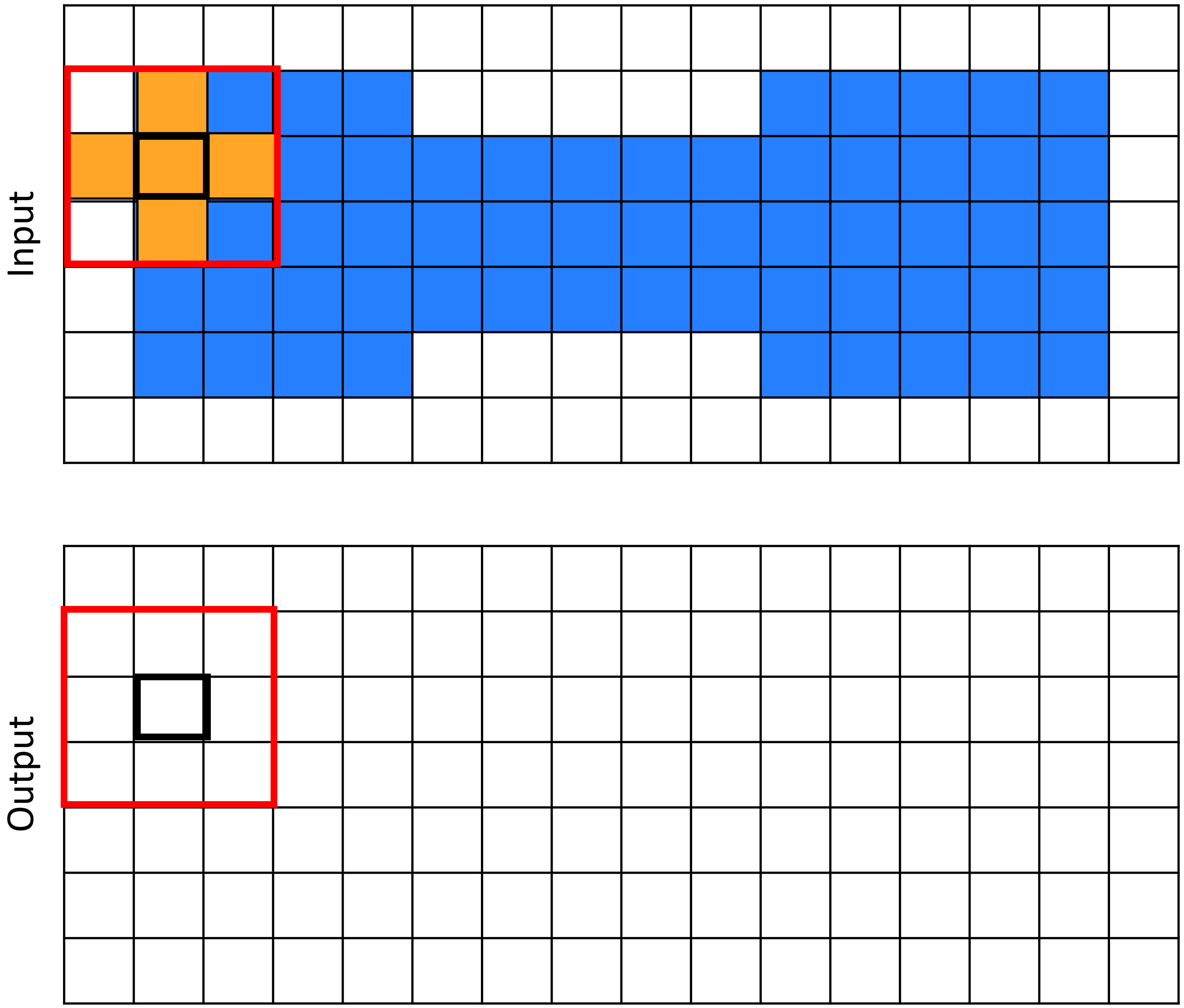

Input
Output

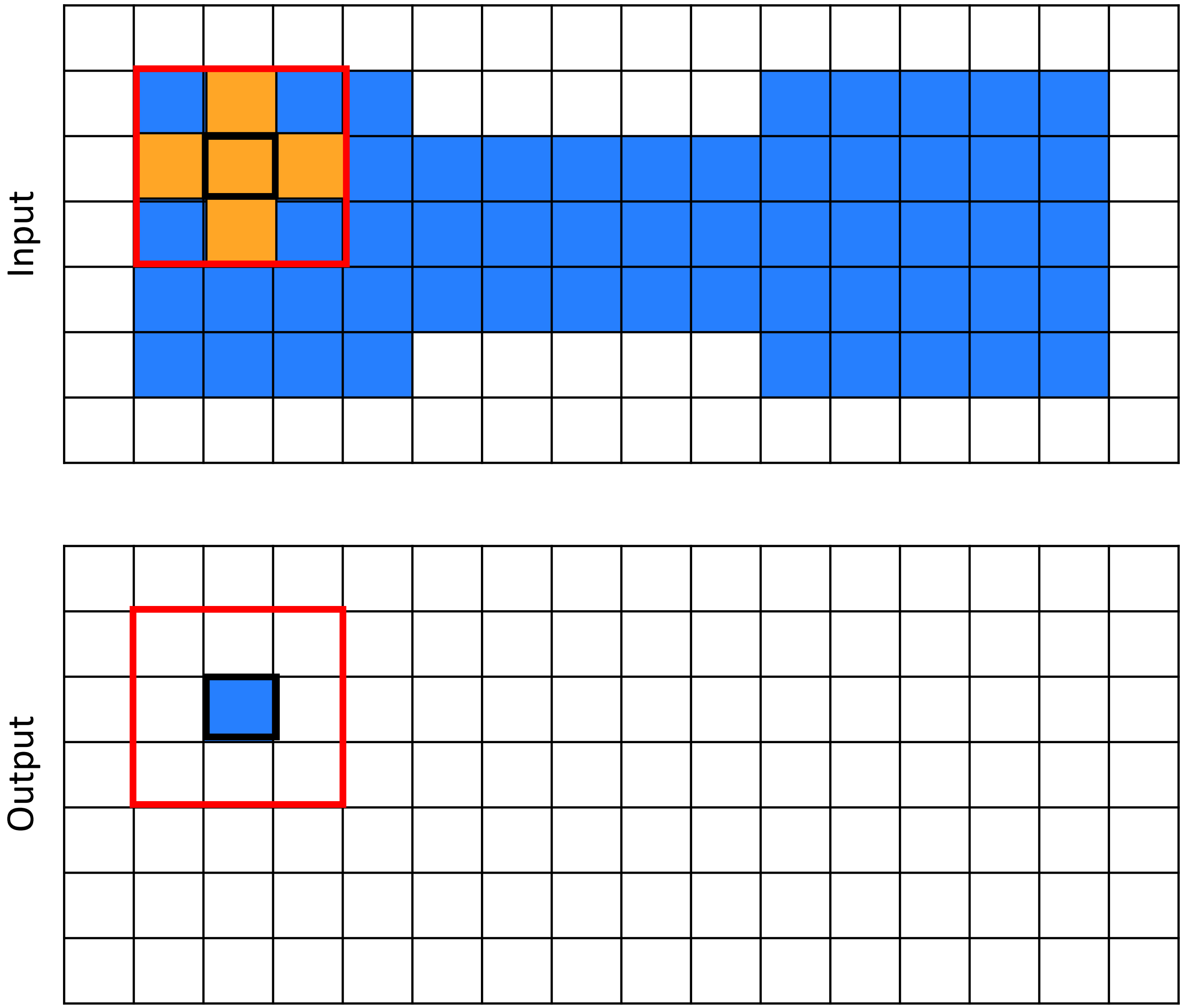

Input
Output


Input
Output

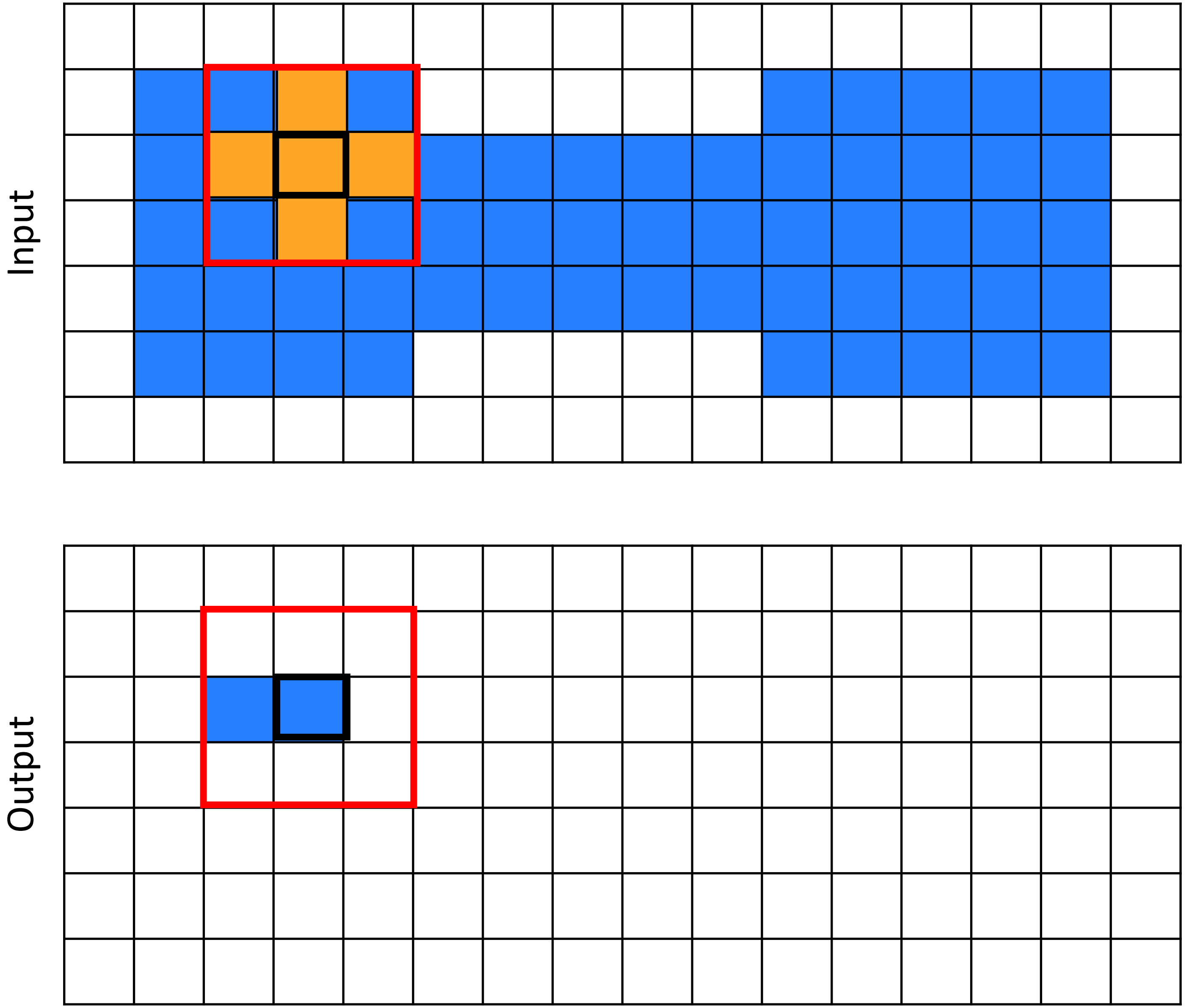

Input
Output

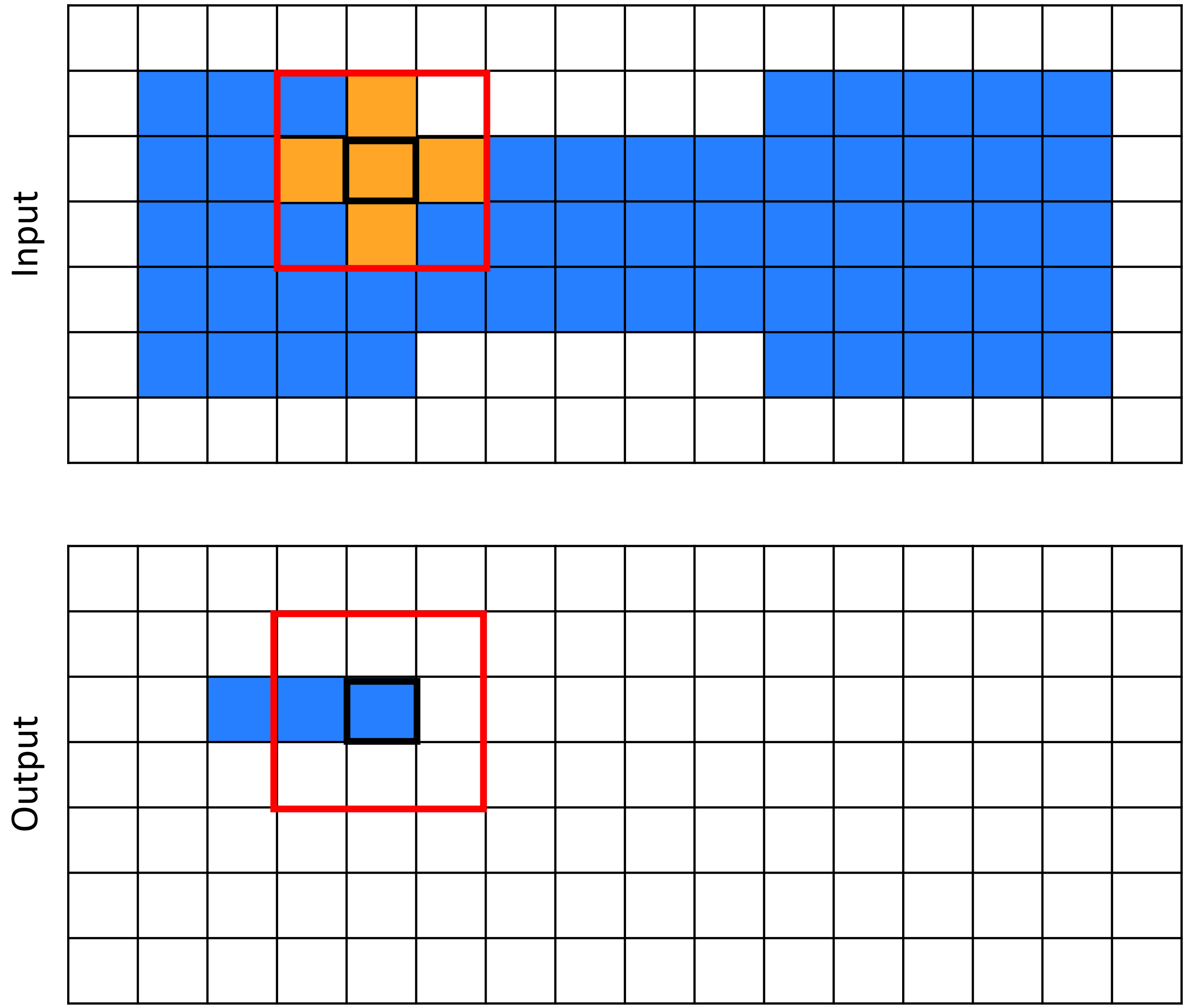
. . .


Input
Output

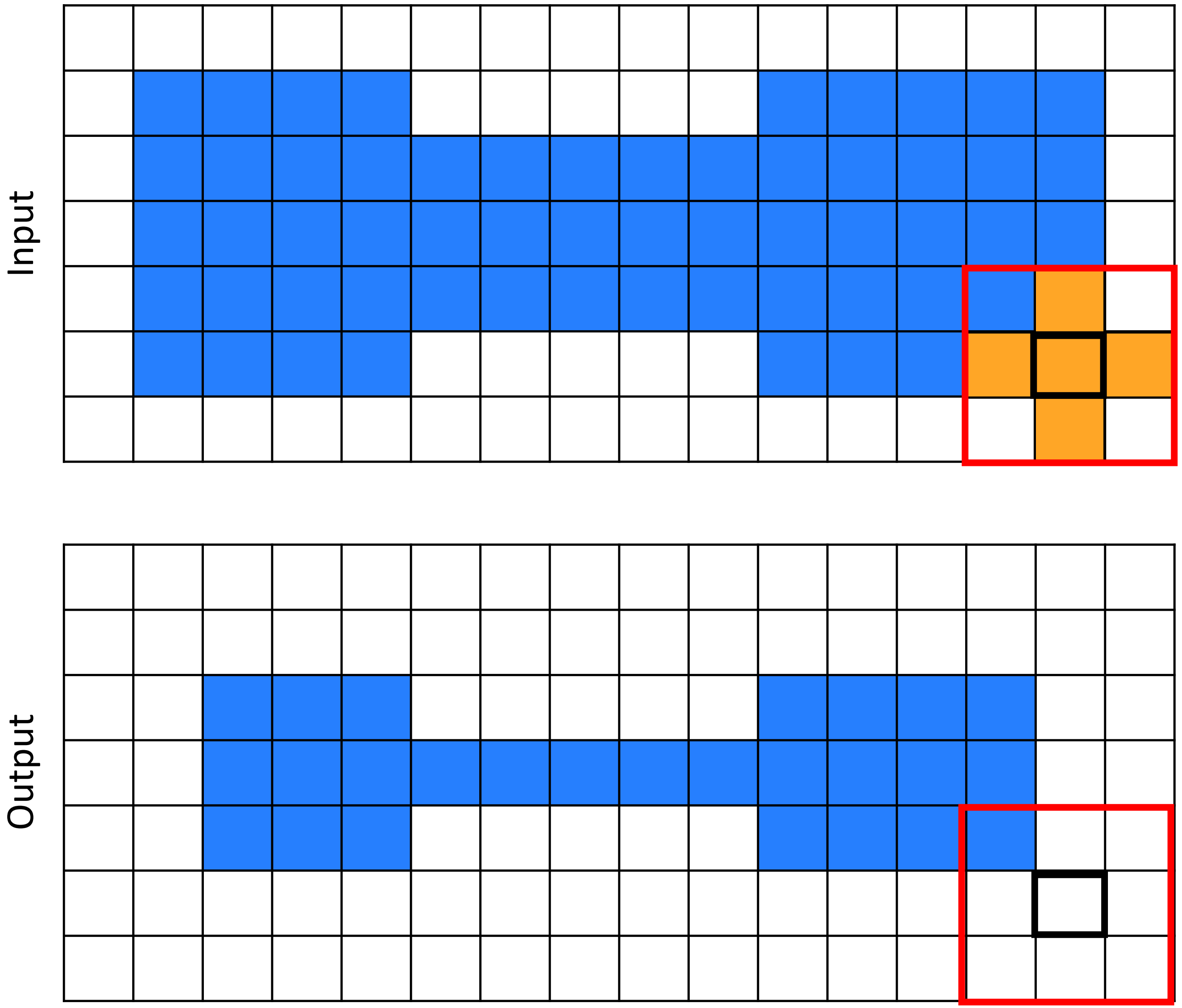


Input
Output


Input
Output

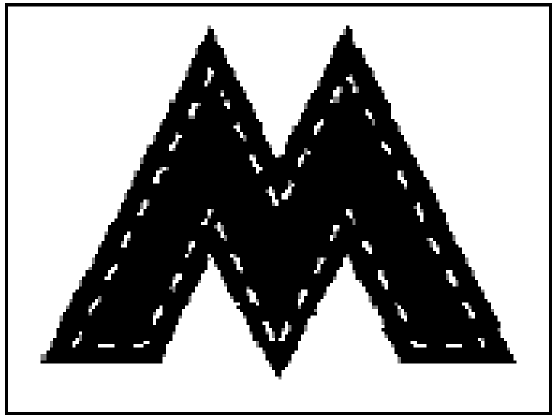

Input
Output

. . .


Input
Output

Dilatation

Expands objects

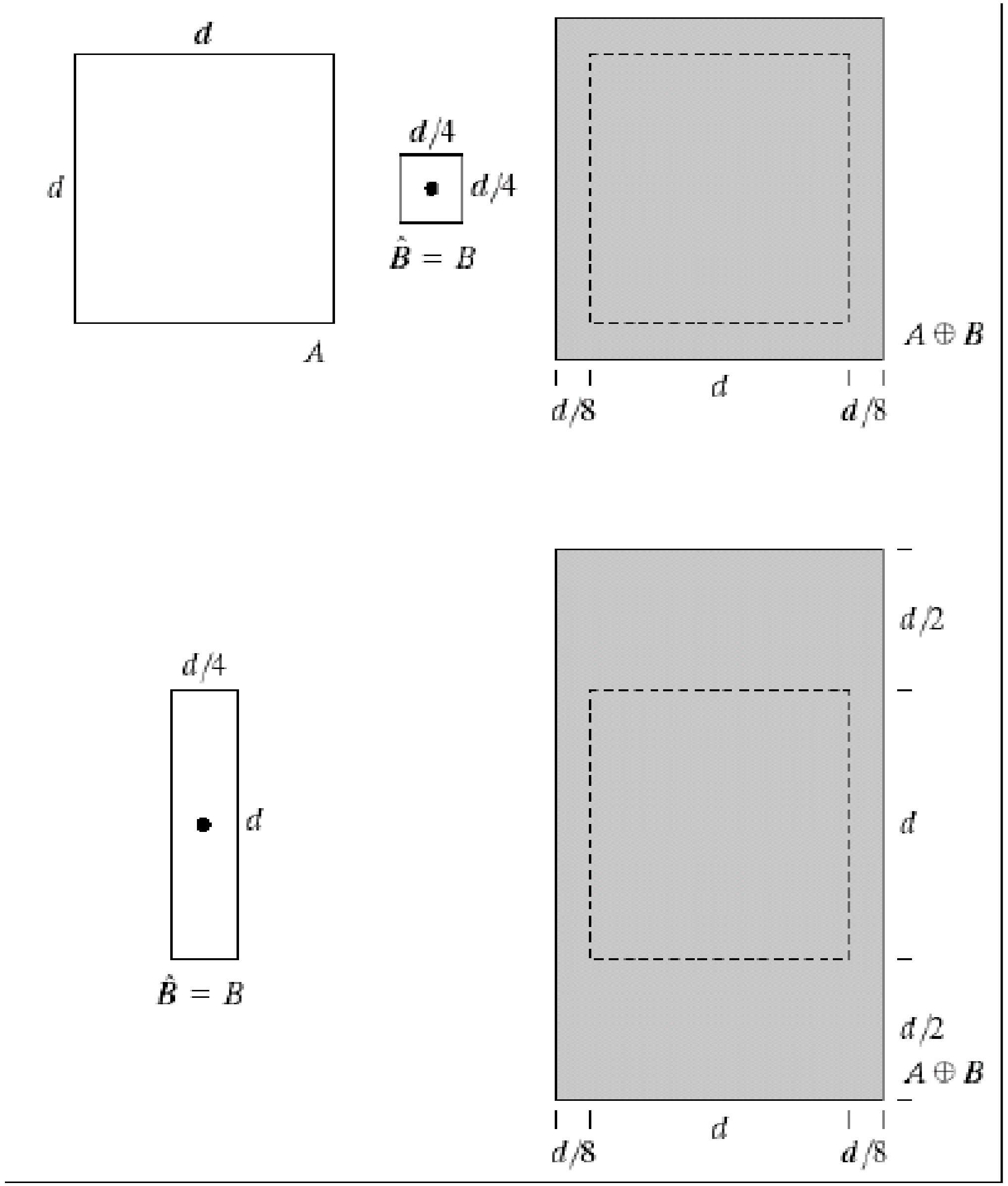


$$A \oplus B = \left\{ z \middle| \left(\hat{B}\right)_z \cap A \neq \varnothing \right\} = \left\{ z \middle| \left(\hat{B}\right)_z \cap A \subseteq A \right\}$$

The expansion effect is due to the application of the **structuring element B near the edges**.

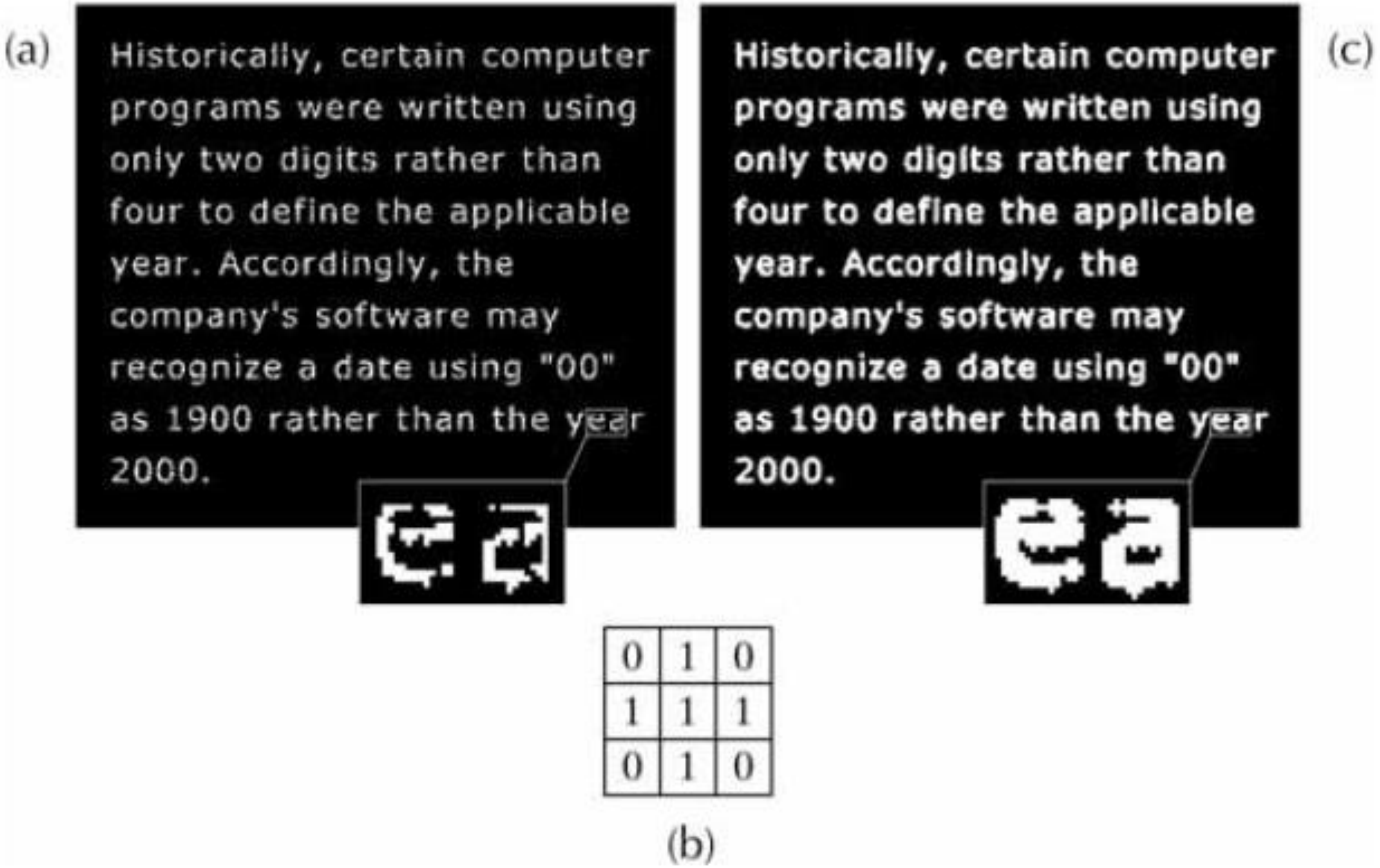
It follows from the definition that the structuring element is flipped with respect to its origin, through the reflection operation, and shifted by z positions through a translation. The result of the operator is the set of z positions such that (B^)z intersects at least one element of A.

Example of Dilation

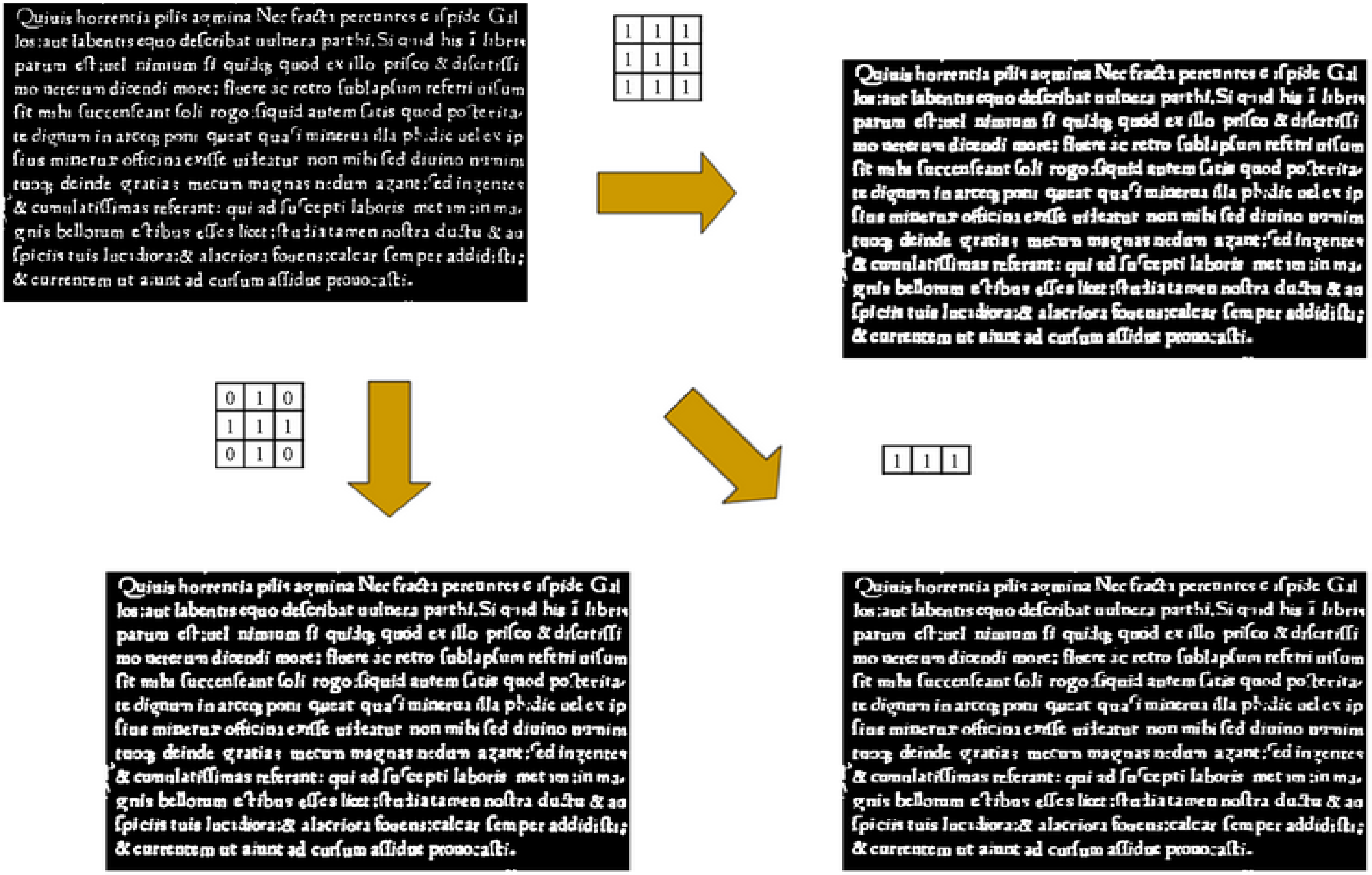


Università di Catania

Example of Dilation

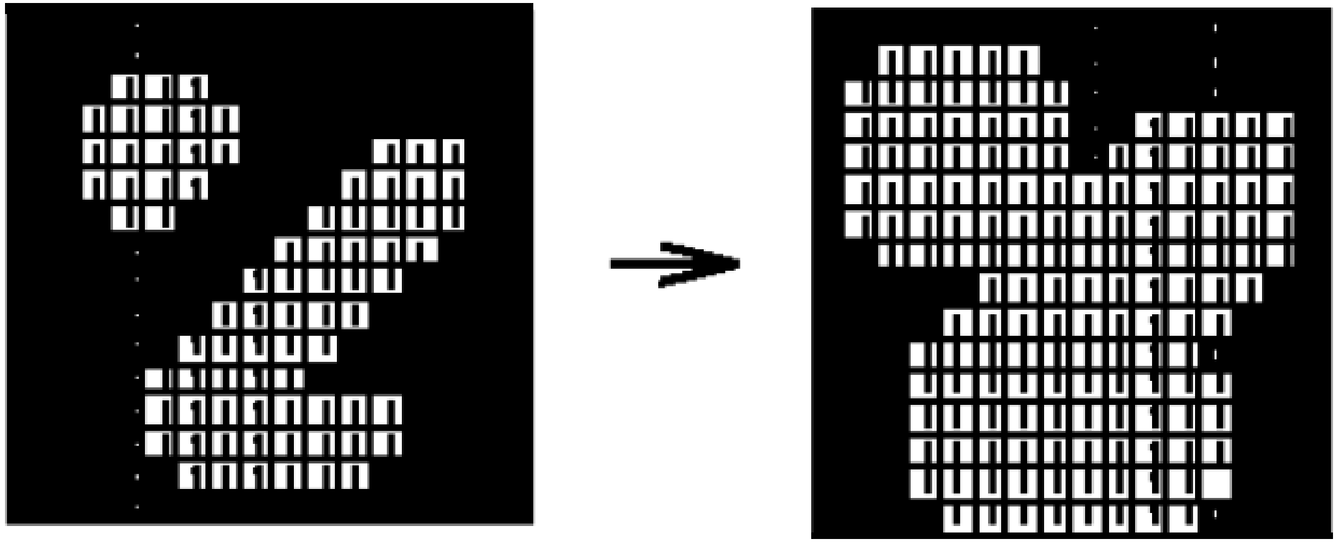


Applications: Filling



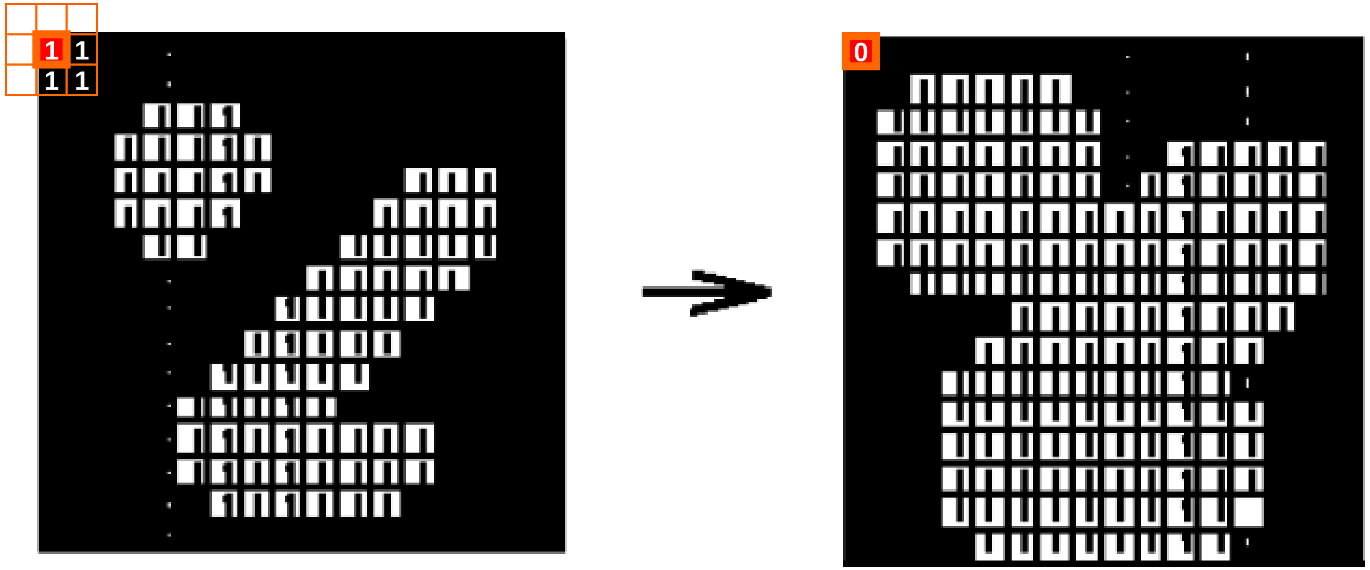
Example of Dilation

> B=ones(3,3)



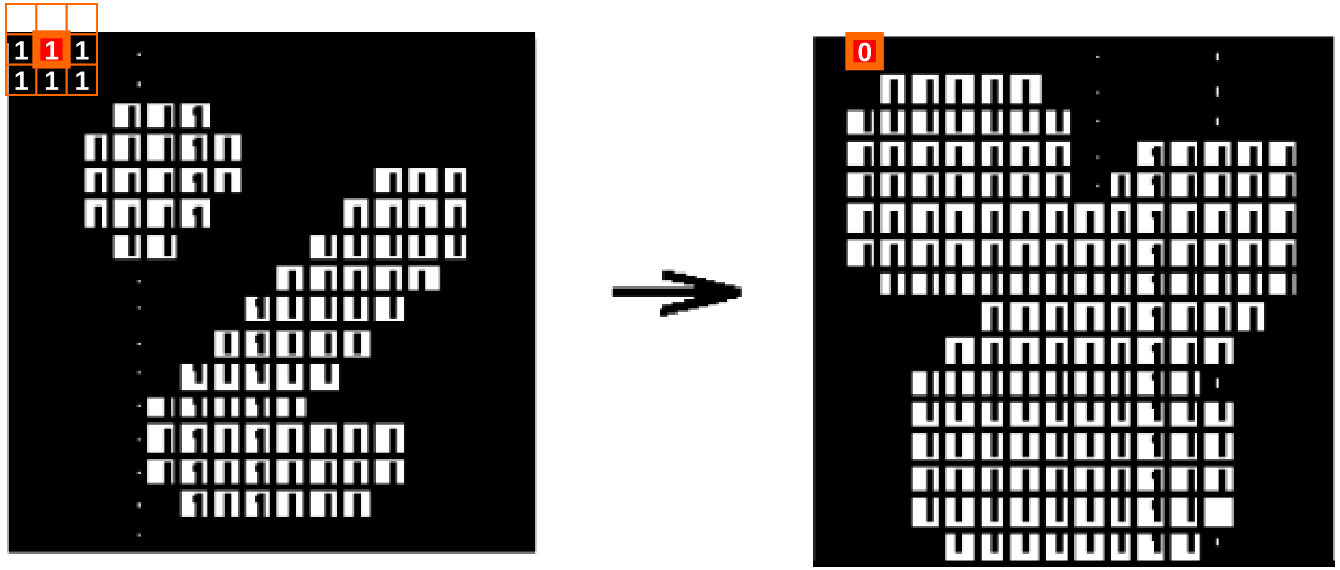
Example of Dilation

> B=ones(3,3)



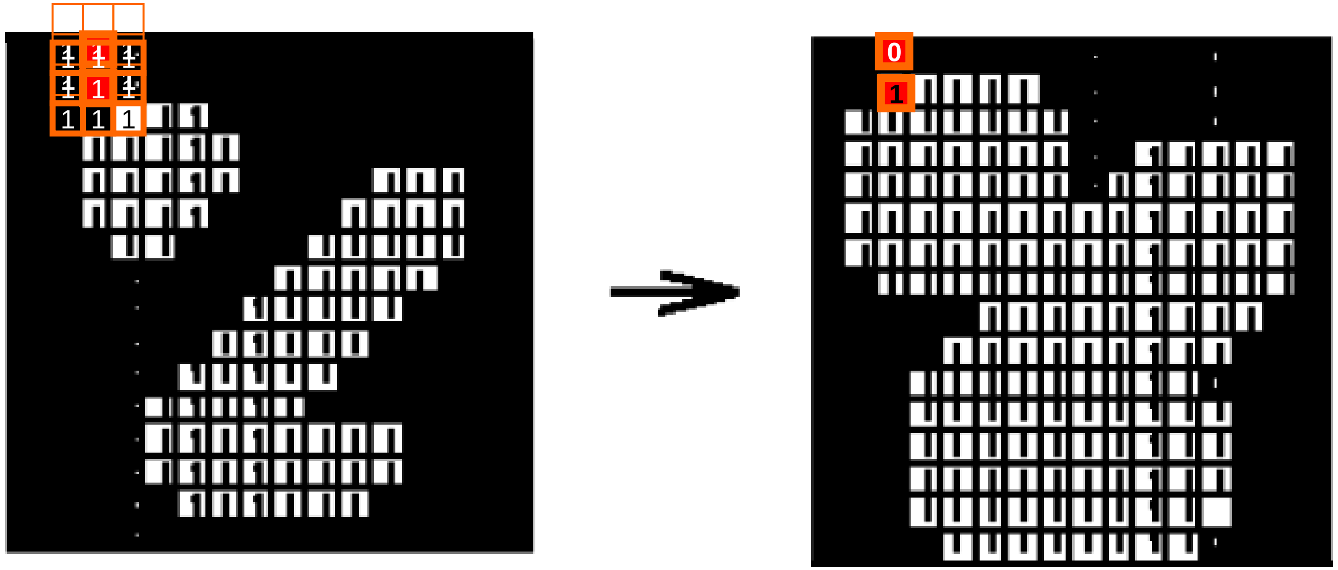
Example of Dilation

> B=ones(3,3)



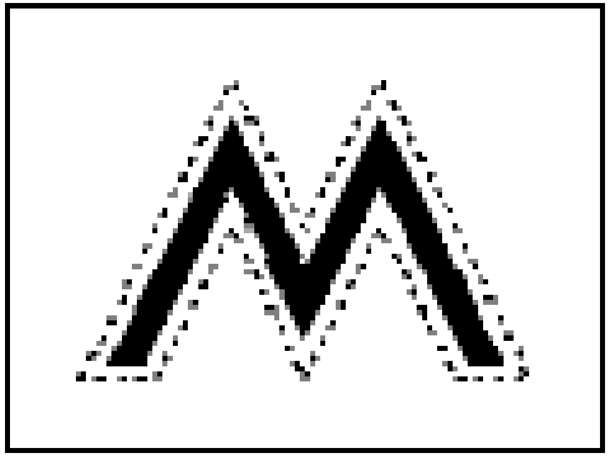
Example of Dilation

> B=ones(3,3)



Erosion

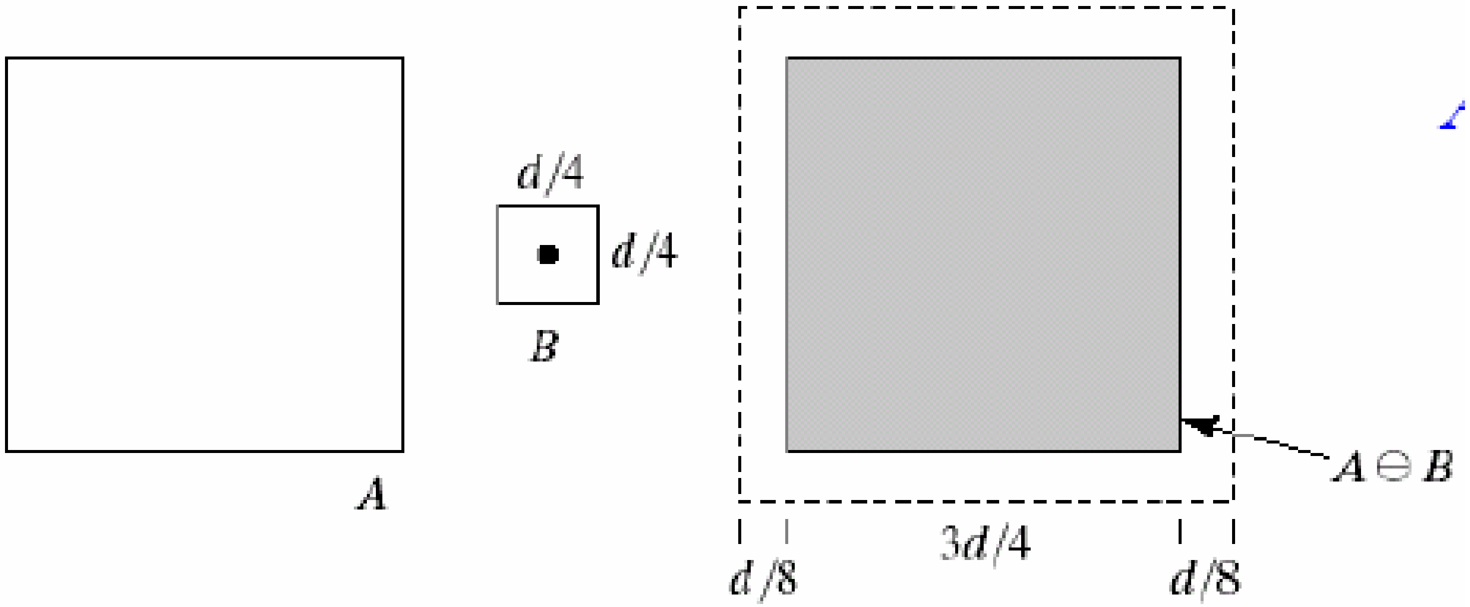
Erodes/shrinks objects

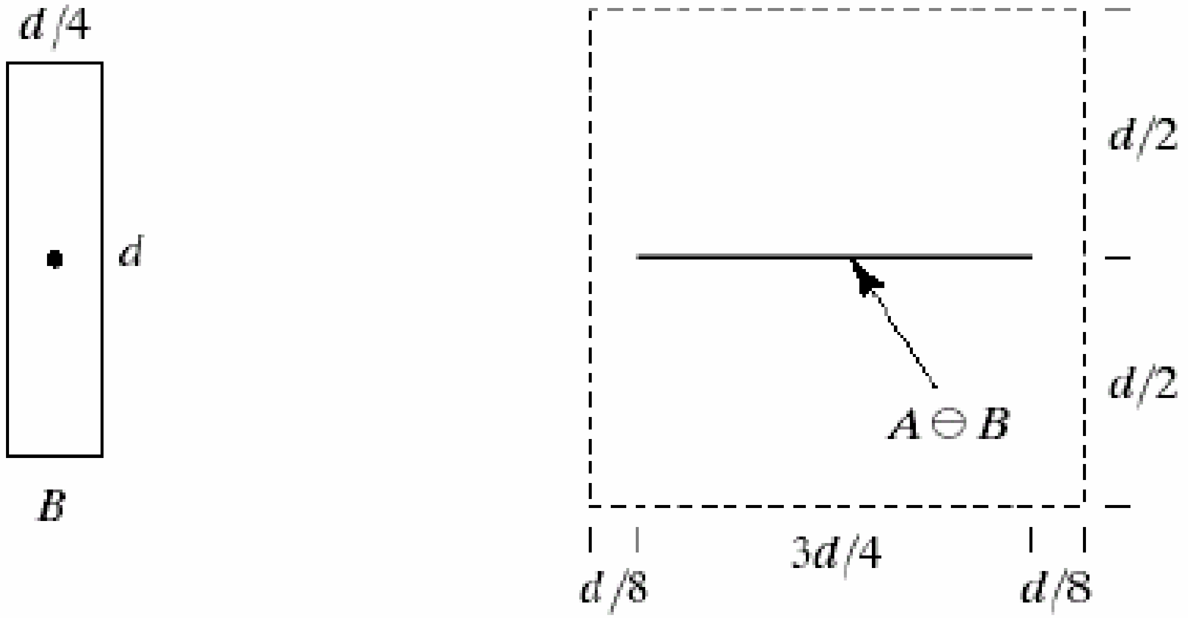


$$A \Theta B = \left\{ z \middle| (B)_z \subseteq A \right\}$$

The erosion effect is due to the fact that when the structuring element B is translated near the edges **it is not completely contained in A**.

Example of Erosion



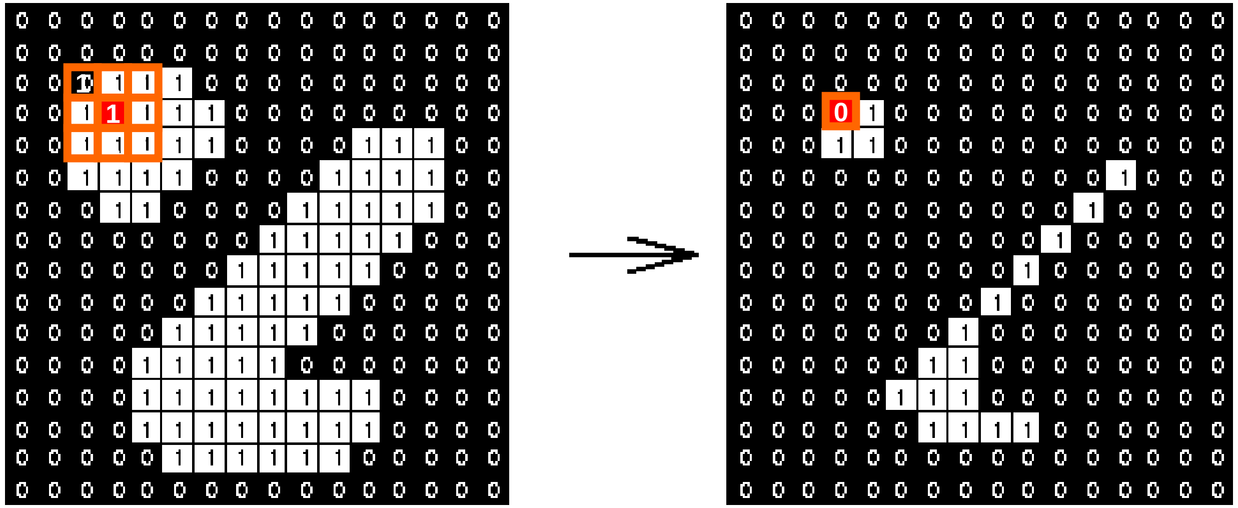


Example of Erosion

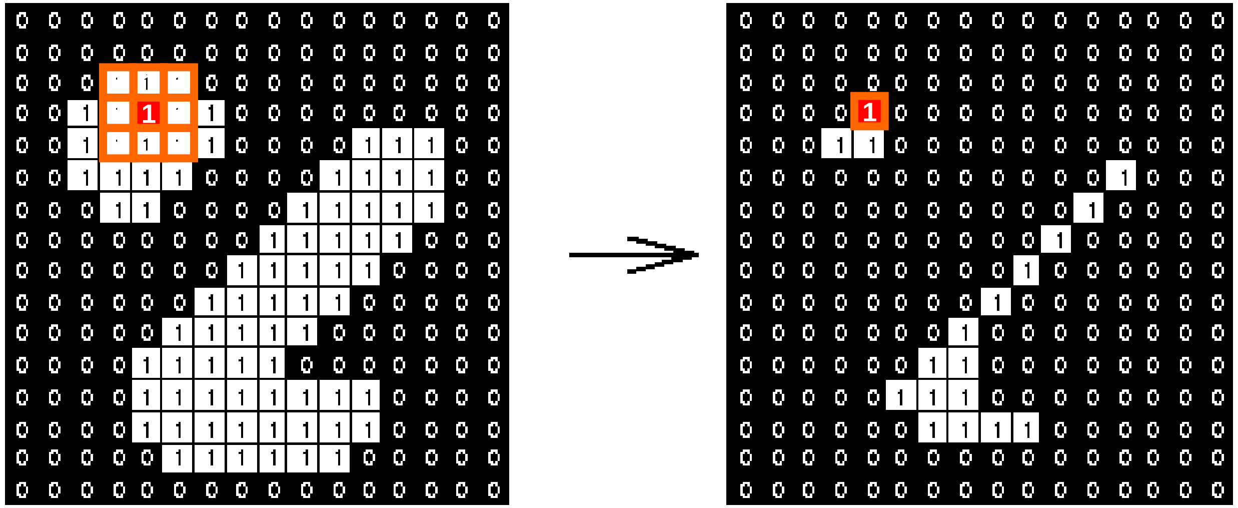
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	1	1	1	0	0
0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
0	0	0	1	1	0	0	0	0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

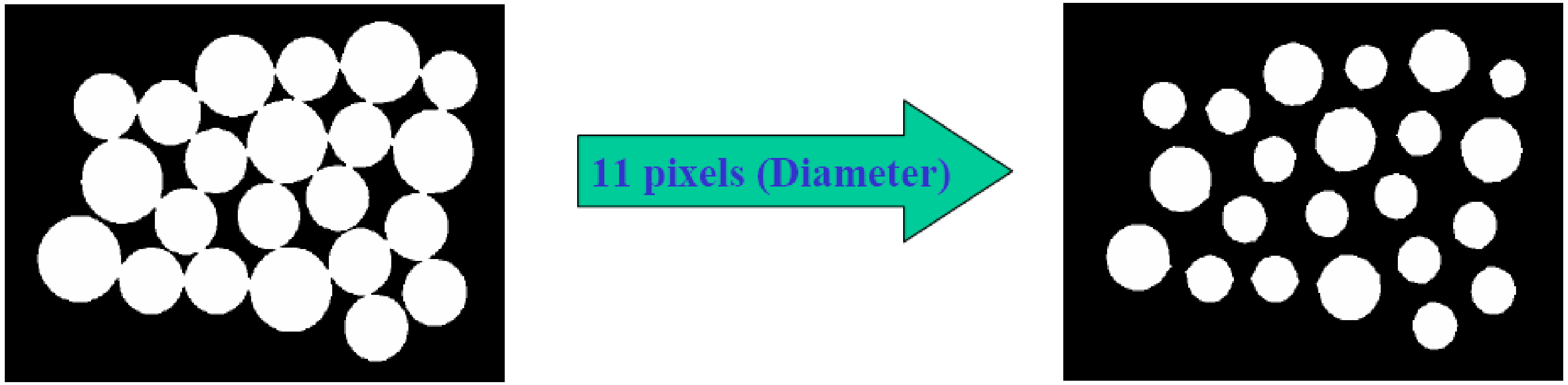
Example of Erosion



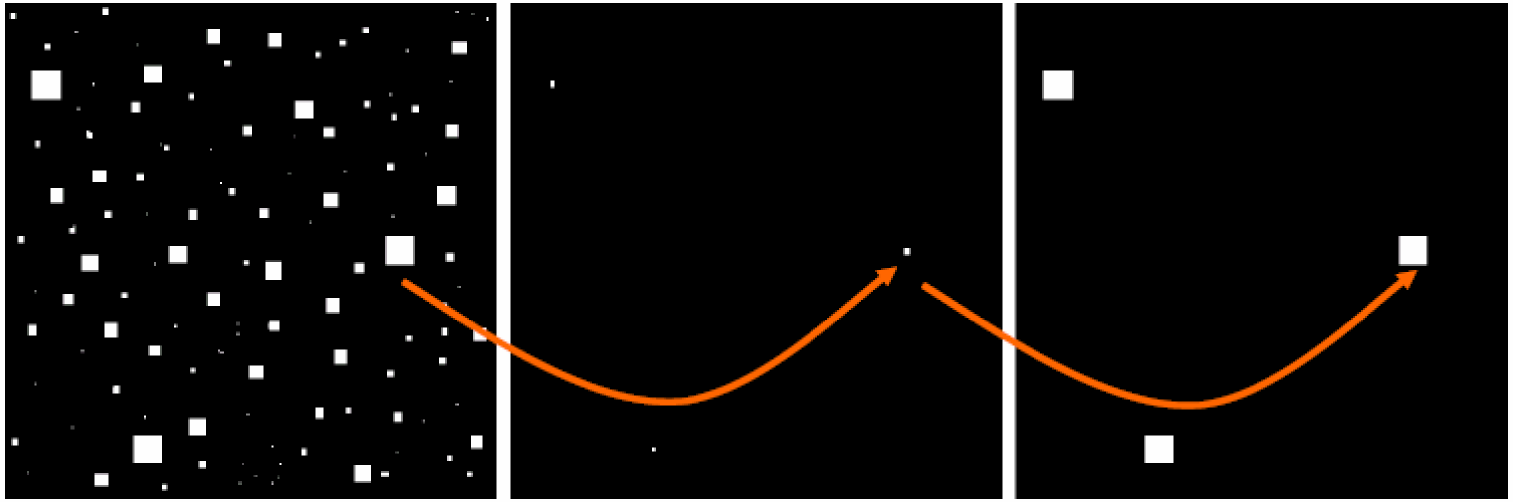
Example of Erosion



Example of Erosion



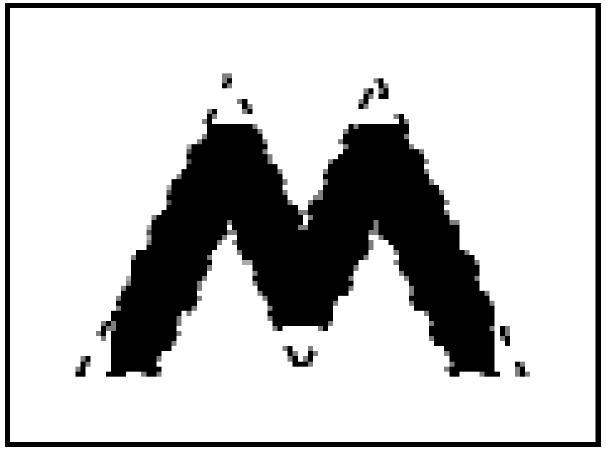
Particle Size Analysis



Shown on the left is an image containing white squares of size 1,3,5,7,9 and 15. In the center is the output of an erosion process with a structural element of side 13. Then applying an expansion with the same structural element results in an elegant **removal of the initial details**

Opening

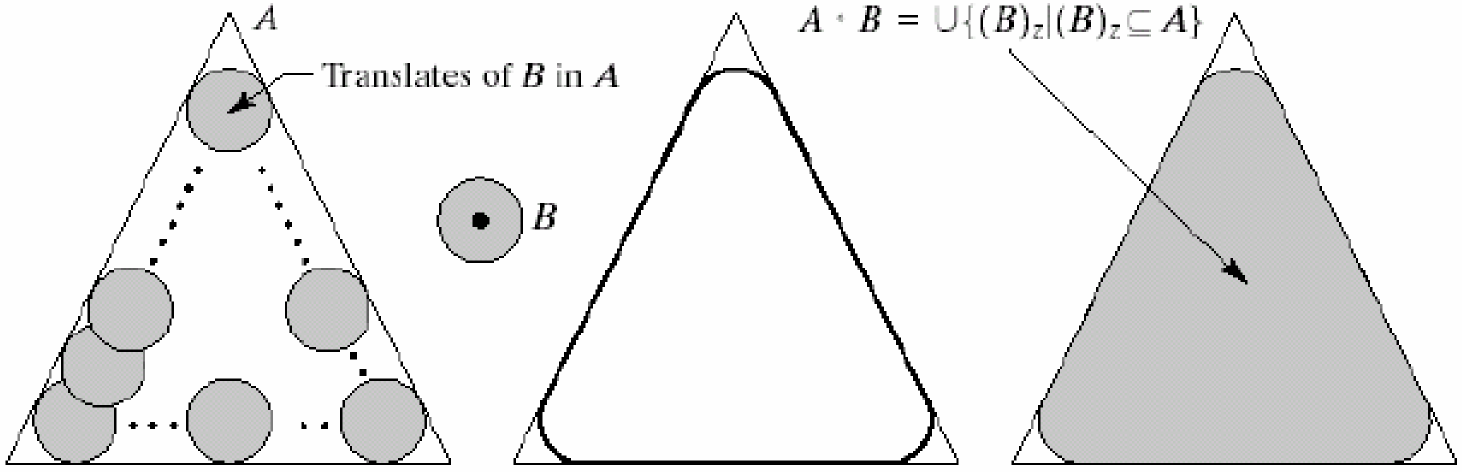
Structured Tip Removal

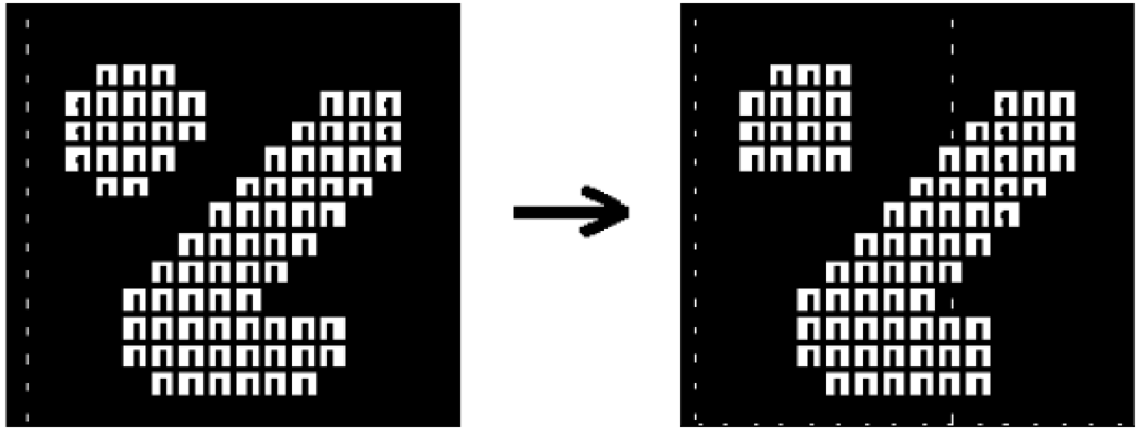


$$Opening(A,B)=A \circ B=(A \Theta B) \oplus B = \cup \{ (B_z) \mid (B_z) \subseteq A \}$$

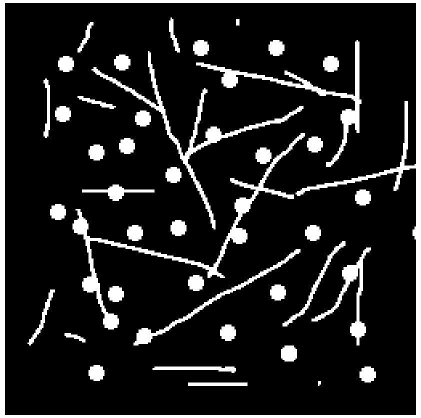
An erosion followed by an expansion using the same structural element. The effect of opening is to preserve regions of similar shape to the structural element as much as possible and to remove different ones. It is a smoothing filter, the power and type of which are determined by the shape and size of B.

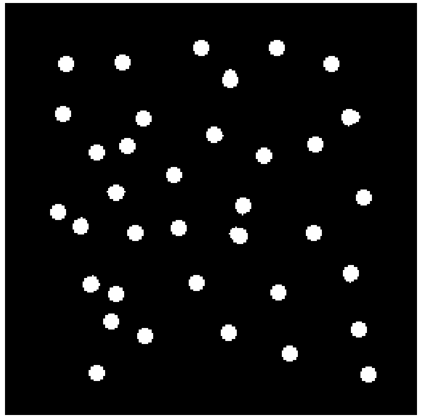
Opening





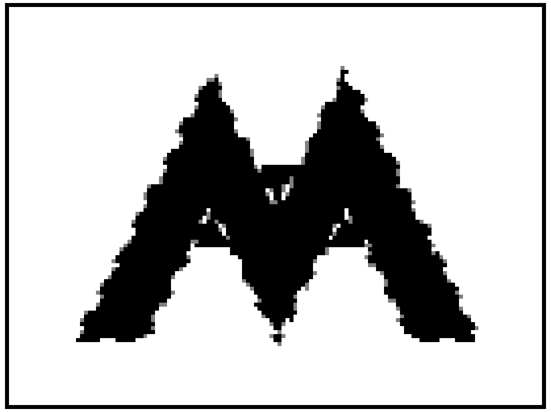
Opening





An example of a problem that requires the application of aperture is the removal of lines from the image in the figure. In this case, a spherical-shaped structural element with a radius equal to that of the circles to be preserved that is greater than the thickness of the lines is used.

Closing

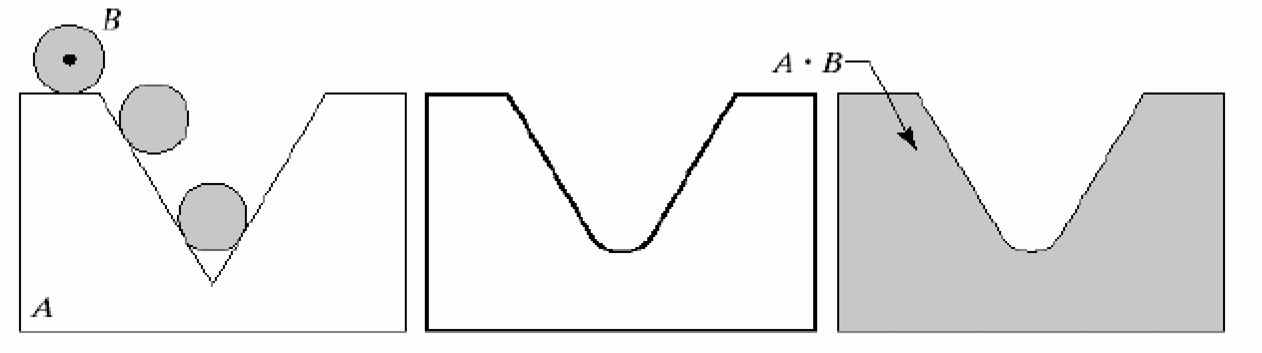


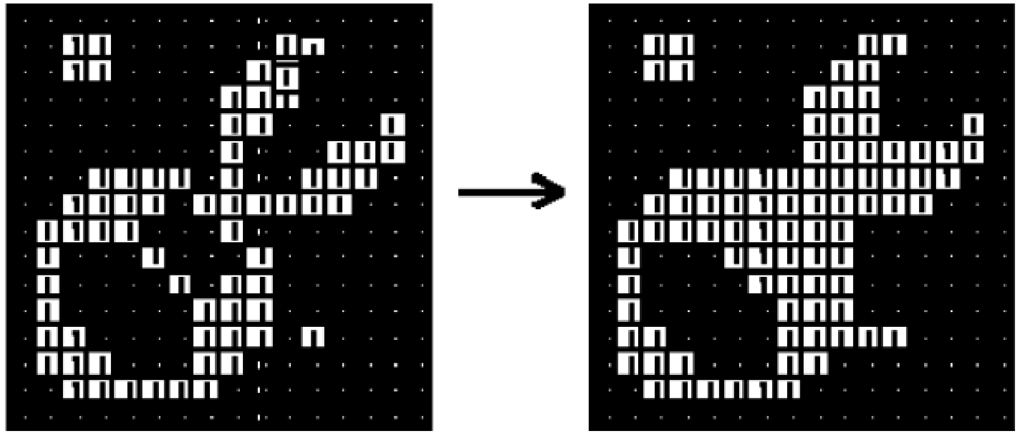
Riempimento strutturato di cavità

$$Closing(A,B) = A \bullet B = (A \oplus B)\ \Theta\ B$$

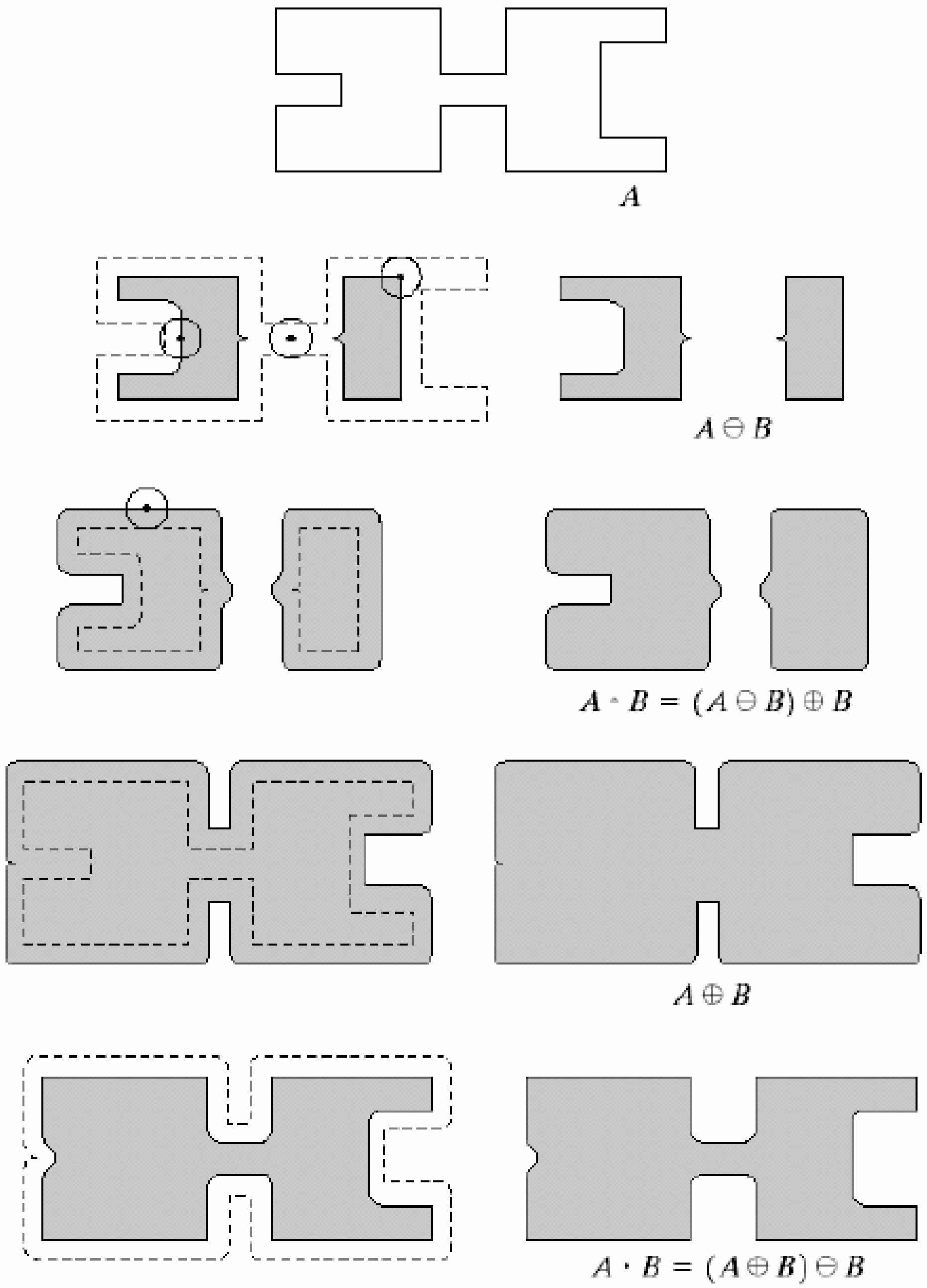
An expansion followed by erosion using the same structural element. The effect of closing is to close any internal holes.

Closing

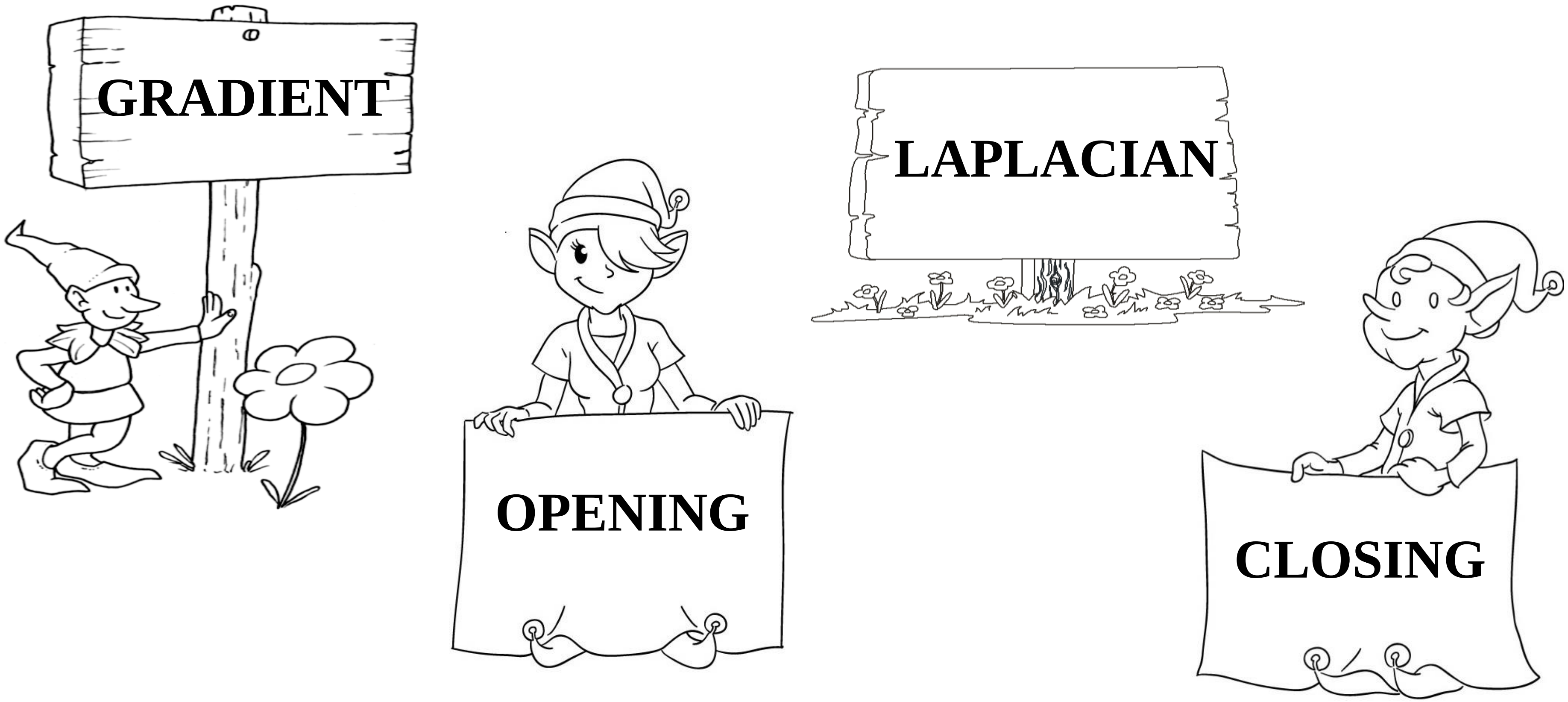




Examples



Expansion and erosion operations can be used in conjunction to achieve other operations. For example:



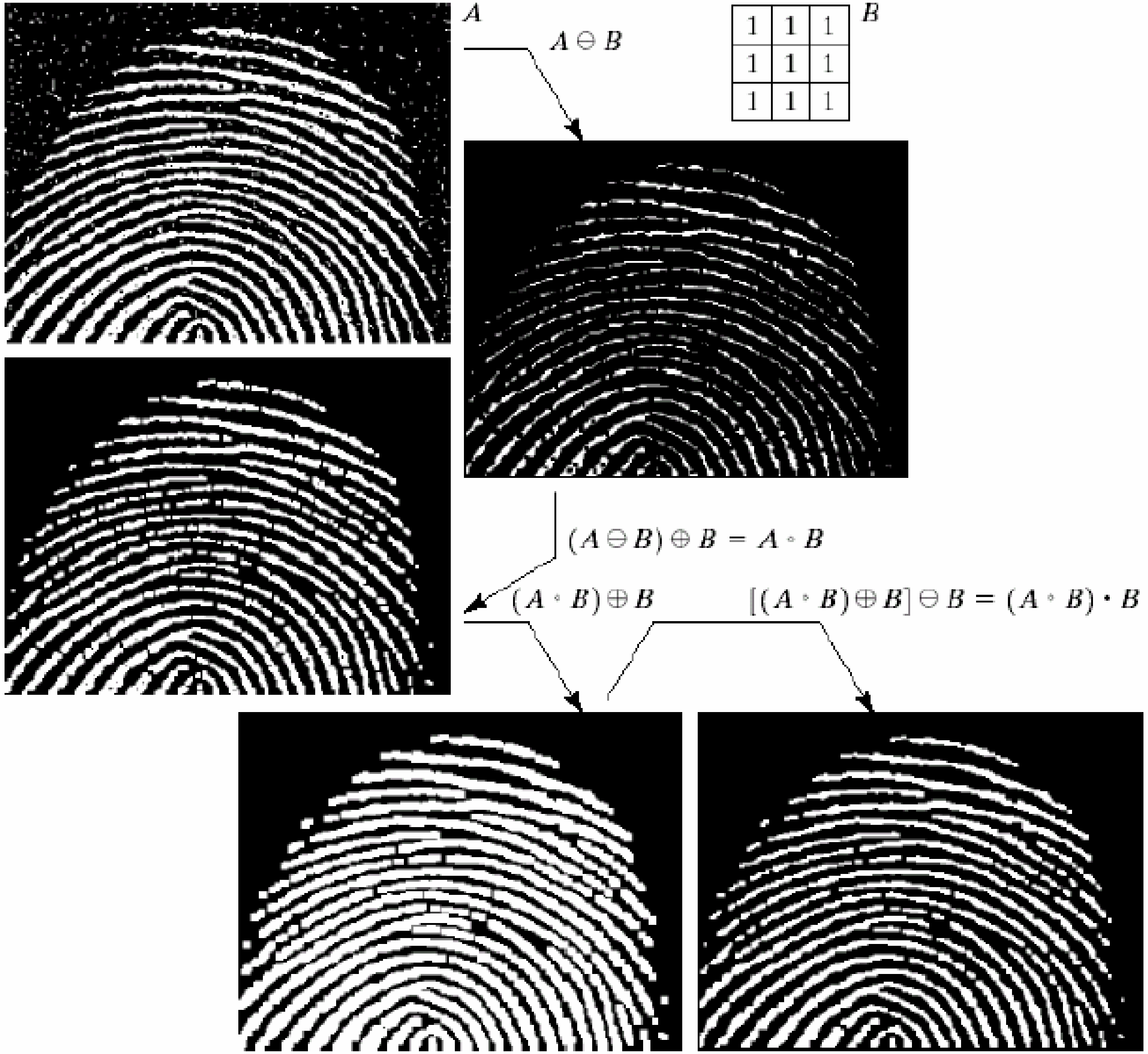
Complex operators

- The combination of the basic operators allows for other complex operations such as
  - contour extraction
  - filling regions
  - connected components
  - convex "shell"
  - Thinning
  - Thickening
  - pruning

Opening/Closing: Noise Reduction

Erosion → Dilatation → **Opening**

Dilatation → Erosion → **Closing**



Estrazione dei contorni

As dilation makes regions thicker and erosion thins them, their difference emphasizes the boundaries between regions. The result is an image in which the edges between objects are clearly seen and in which the contribution of homogeneous regions is not present.

$$g = (f \oplus b) - (f \ominus b)$$

- Binary images
- Grayscale images

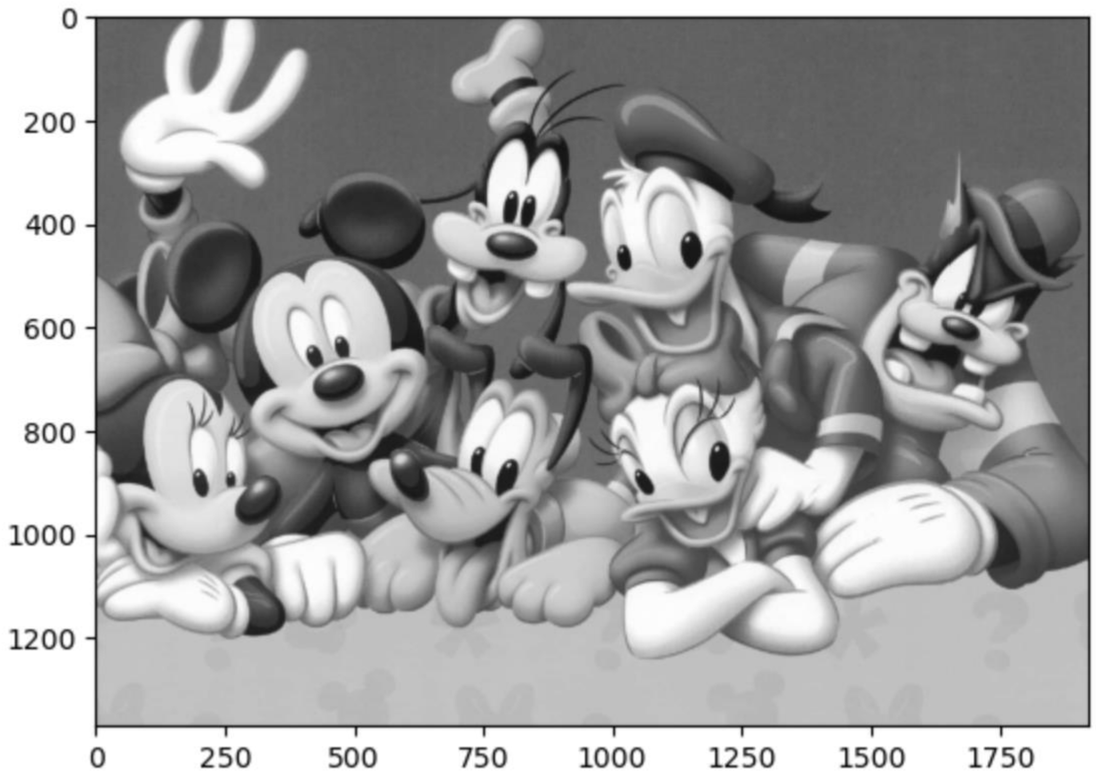
Python and OpenCV

Python and OpenCV

Example with black and white image

```python
import cv2
import numpy as np
img = cv2.imread("image.jpg",0)
plt.imshow(img, cmap="gray")
```

```
<matplotlib.image.AxesImage at 0x2ca4b9ac520>
```



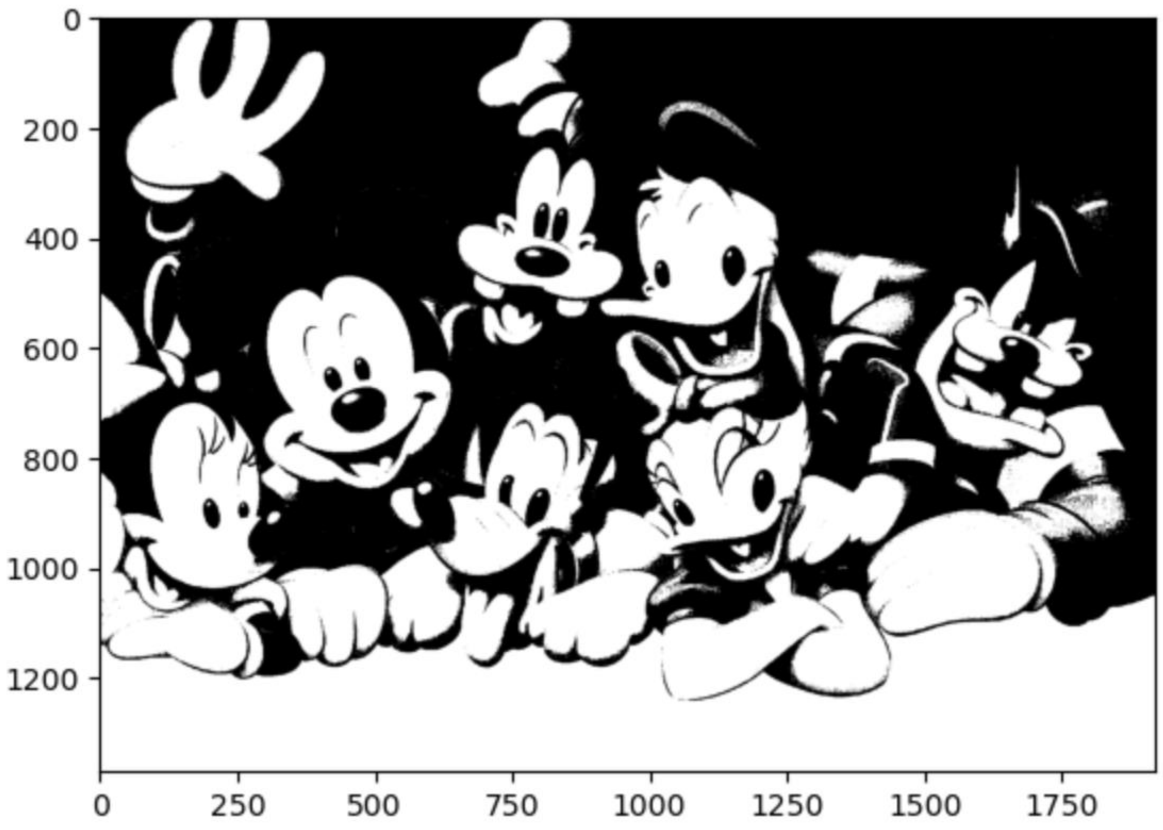
Input

Python and OpenCV

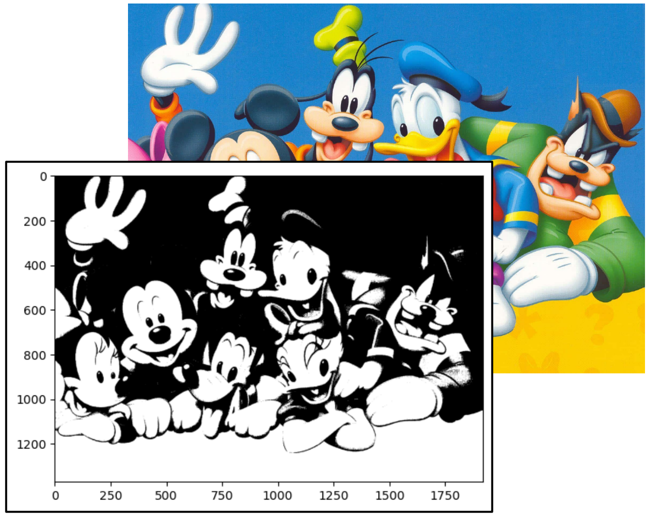
Example with black and white image

```
(thresh, im_bw) = cv2.threshold(img, 128, 255, cv2.THRESH_BINARY | cv2.THRESH_OTSU)
plt.imshow(im_bw, cmap="gray")
```

<matplotlib.image.AxesImage at 0x2ca4ca0bd30>



Input

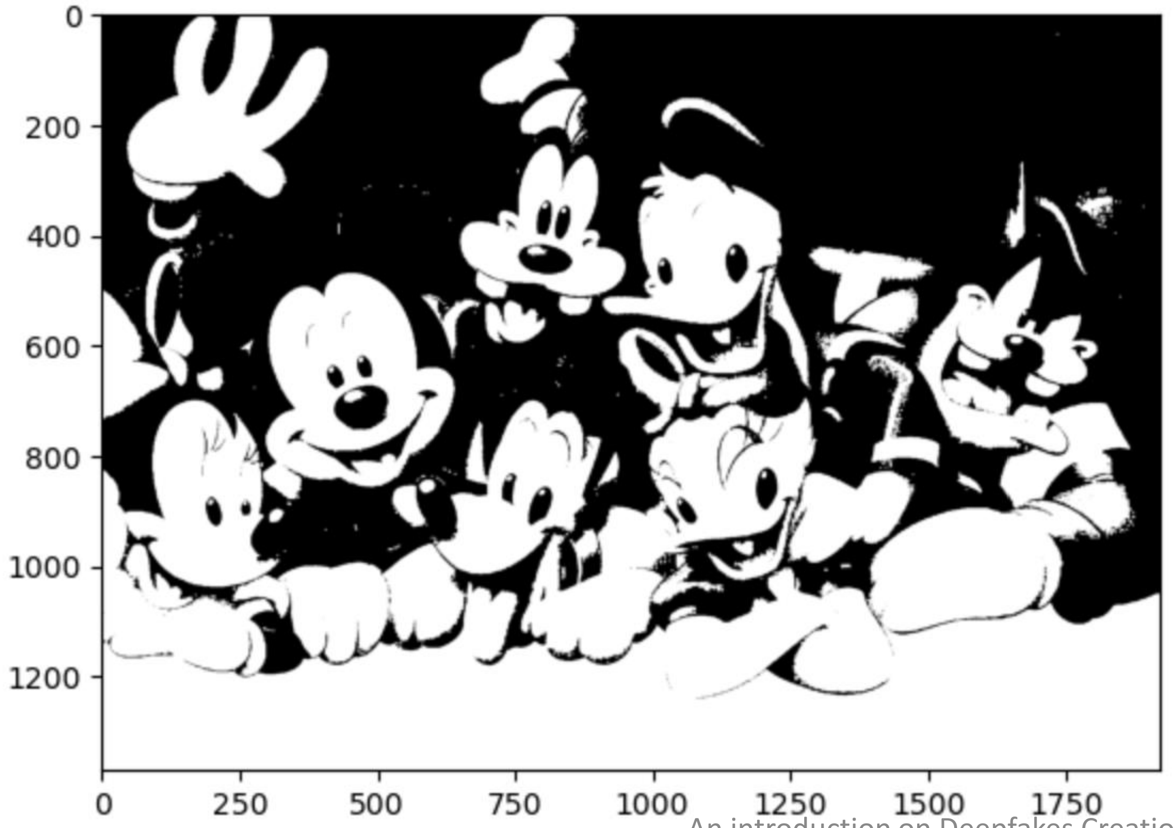


Python and OpenCV

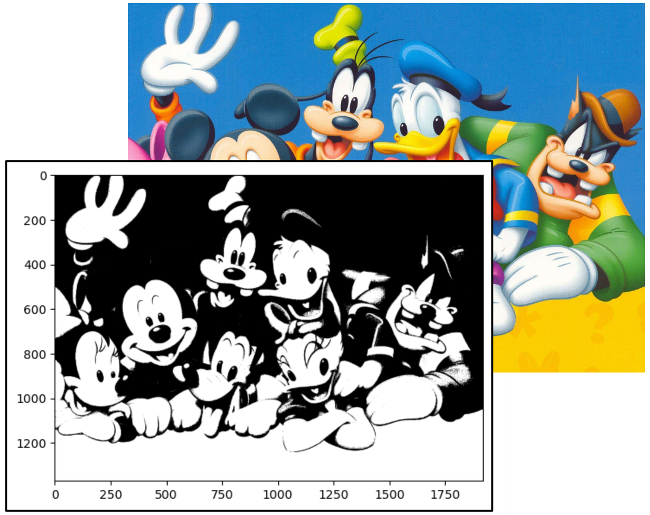
Example with black and white image

```
kernel = np.ones((3,3),np.uint8)
dilation = cv2.dilate(im_bw, kernel, iterations = 1)
plt.imshow(dilation, cmap="gray")
```

```
<matplotlib.image.AxesImage at 0x2ca4d9b43d0>
```



Input

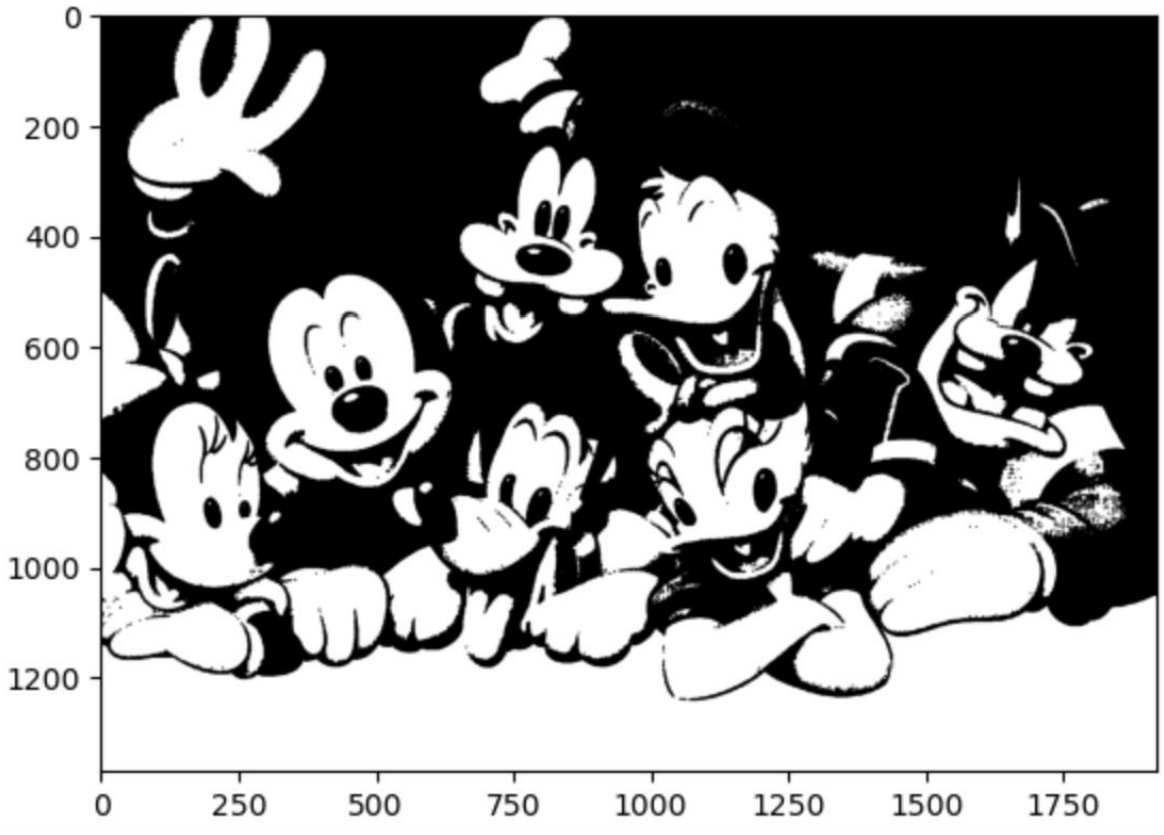


Python and OpenCV

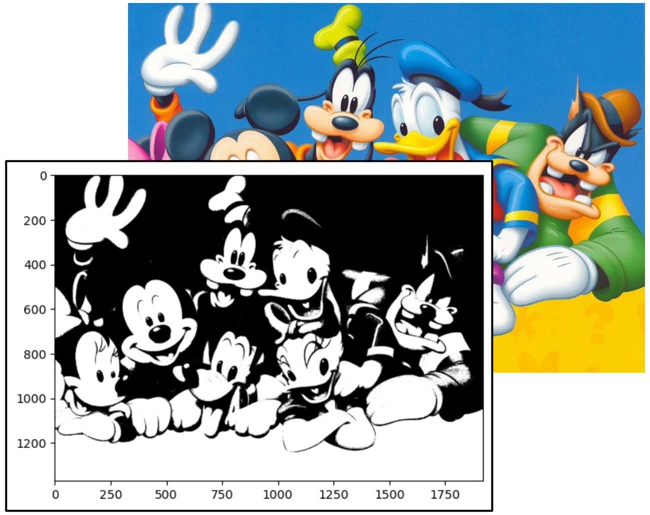
Example with black and white image

```
erosion = cv2.erode(im_bw, kernel, iterations = 1)
plt.imshow(erosion, cmap="gray")
```

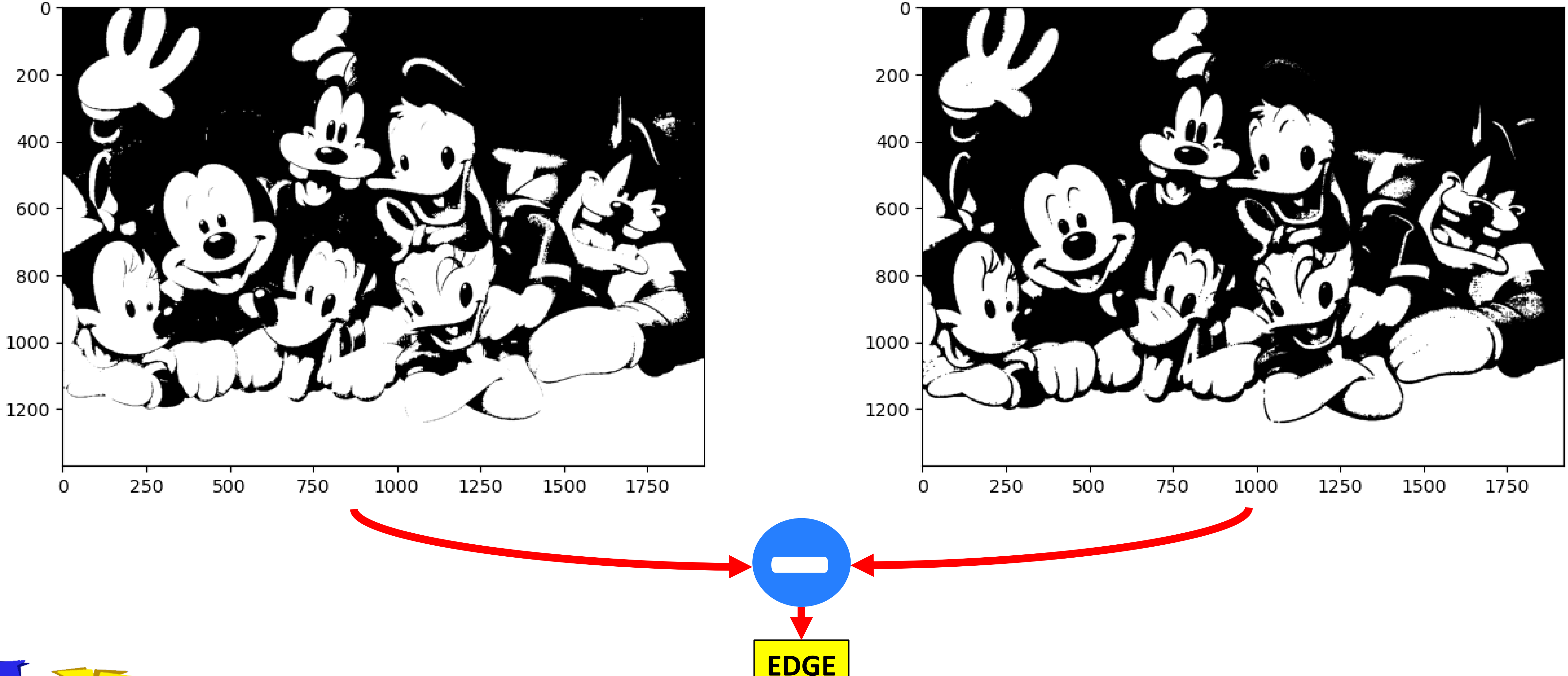
```
<matplotlib.image.AxesImage at 0x2ca4da30340>
```



Input



Dilate and Erode

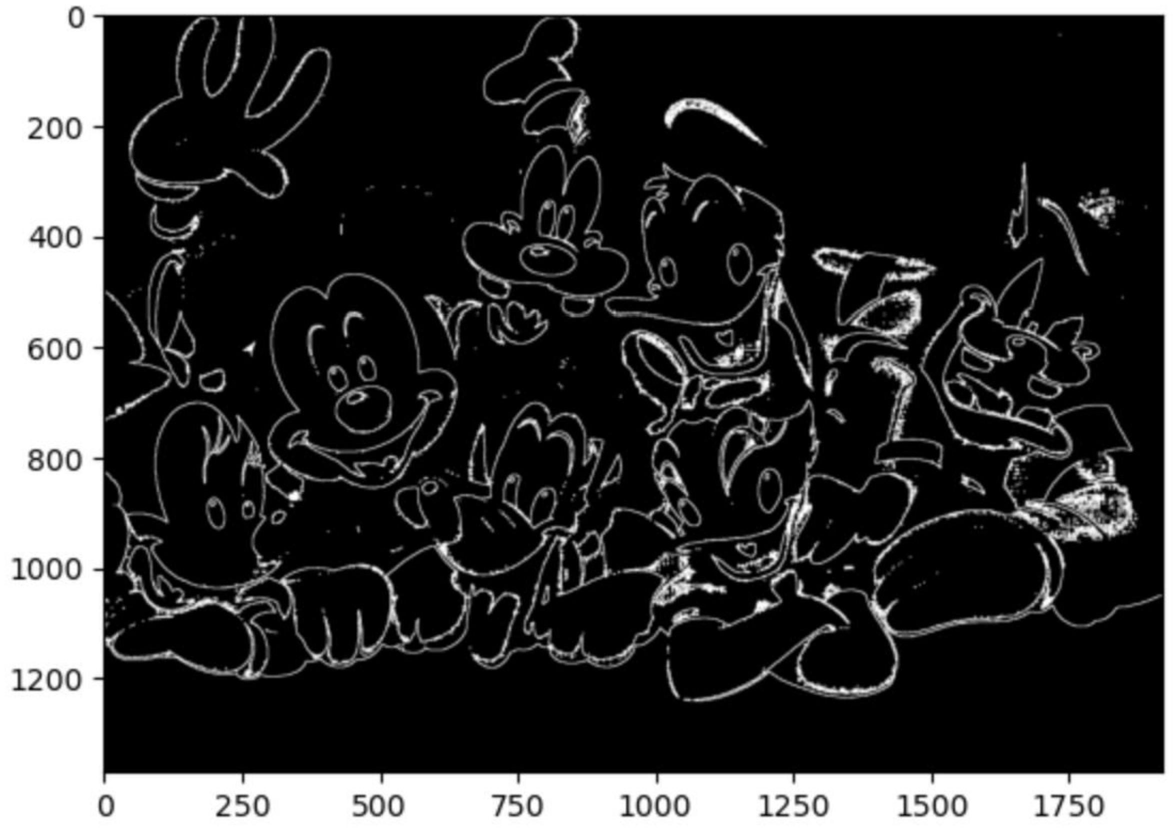


Python and OpenCV

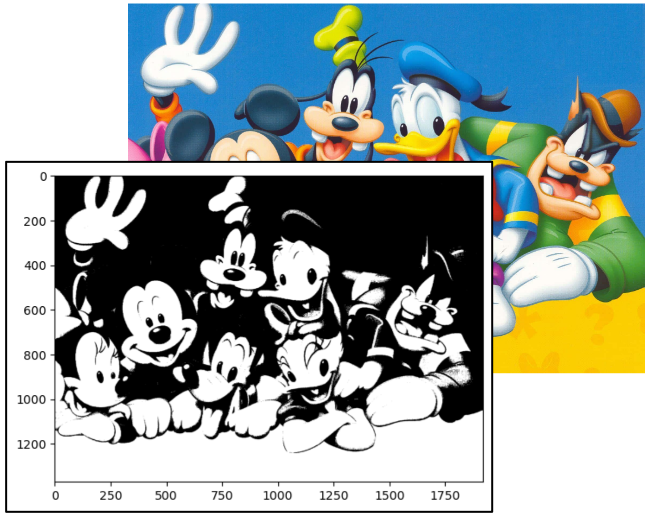
Example with black and white image

```
edge = dilation - erosion
plt.imshow(edge, cmap="gray")
```

<matplotlib.image.AxesImage at 0x2ca4da5b520>



Input

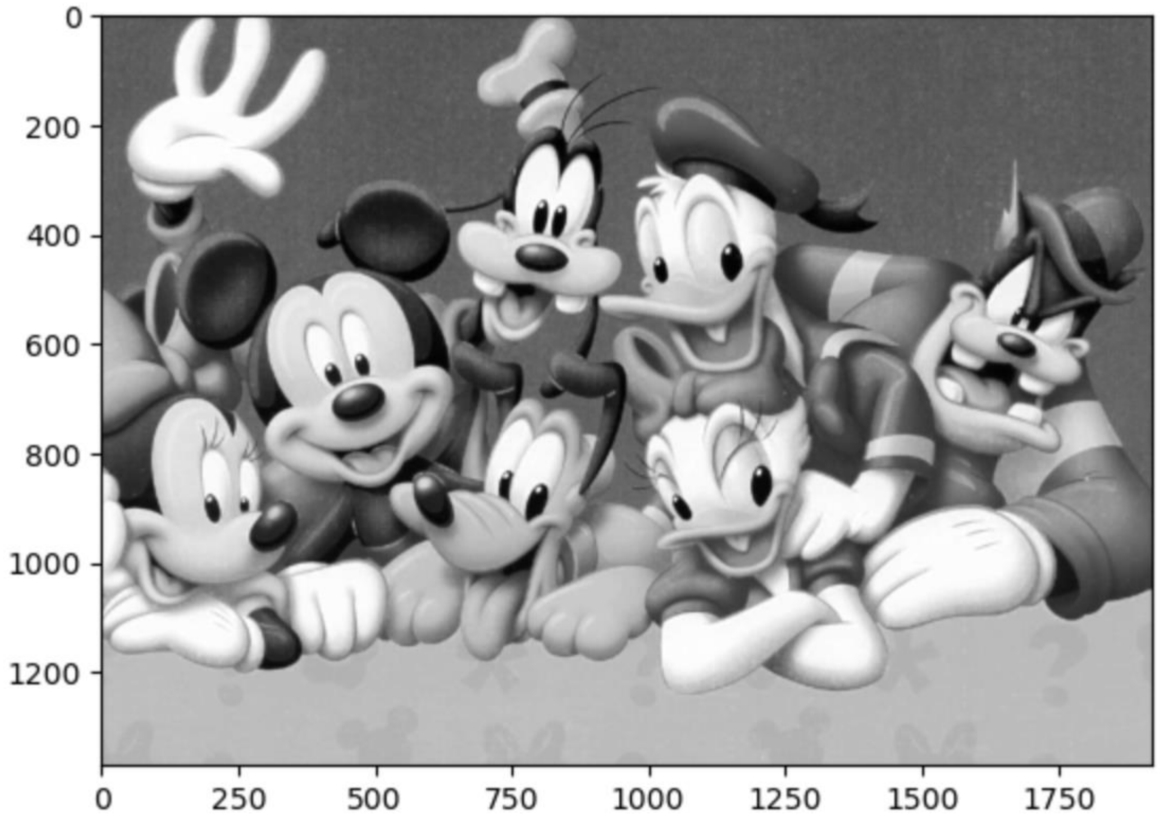


Python and OpenCV

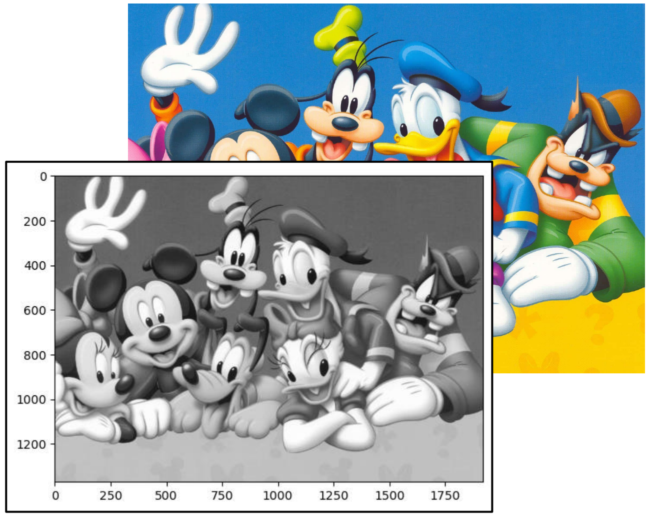
Example with Gray scale image

```
kernel = np.ones((3,3),np.uint8)
dilation = cv2.dilate(img, kernel, iterations = 1)
plt.imshow(dilation, cmap="gray")
```

```
<matplotlib.image.AxesImage at 0x2ca4e2d8df0>
```



Input

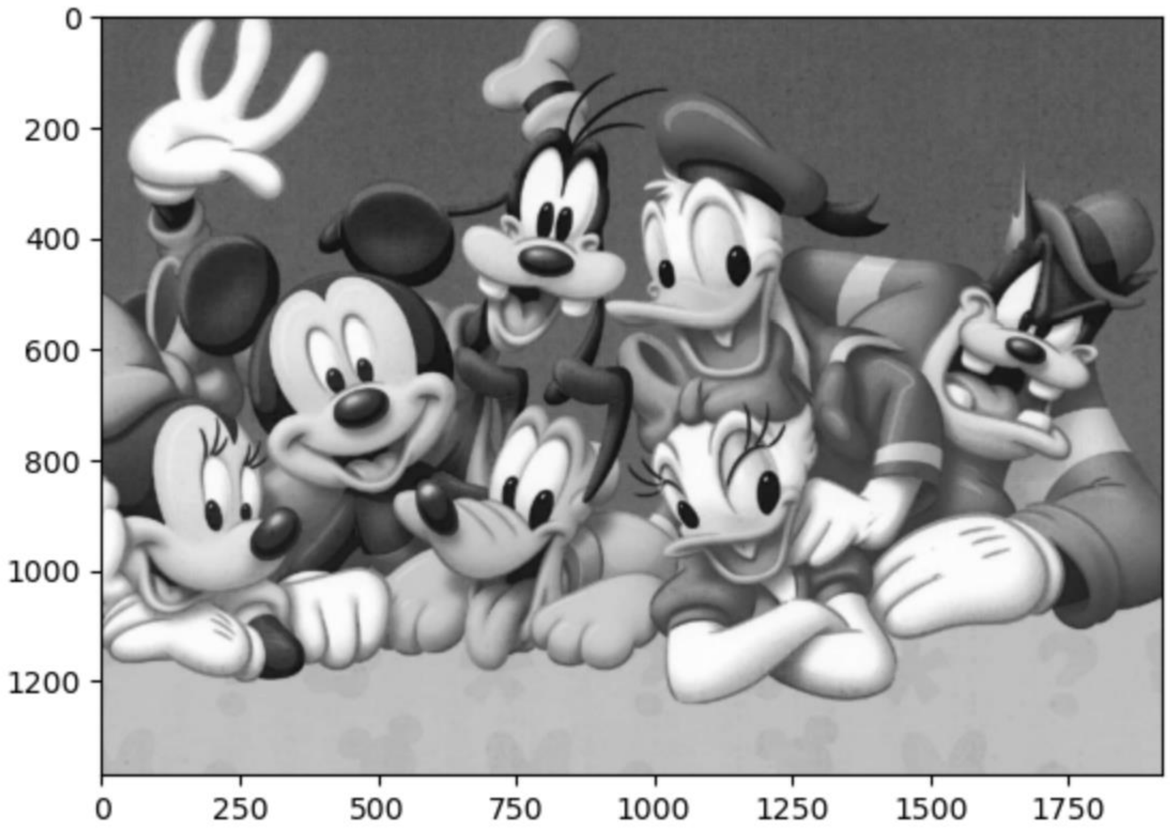


Python and OpenCV

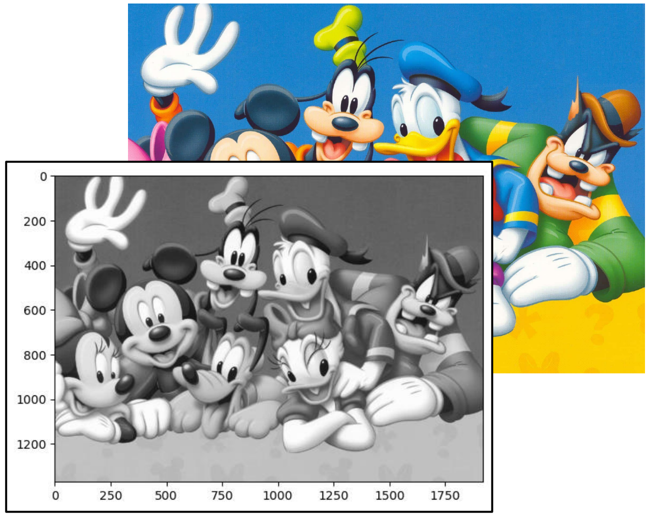
Example with **Gray scale** image

```
erosion = cv2.erode(img, kernel, iterations = 1)
plt.imshow(erosion, cmap="gray")
```

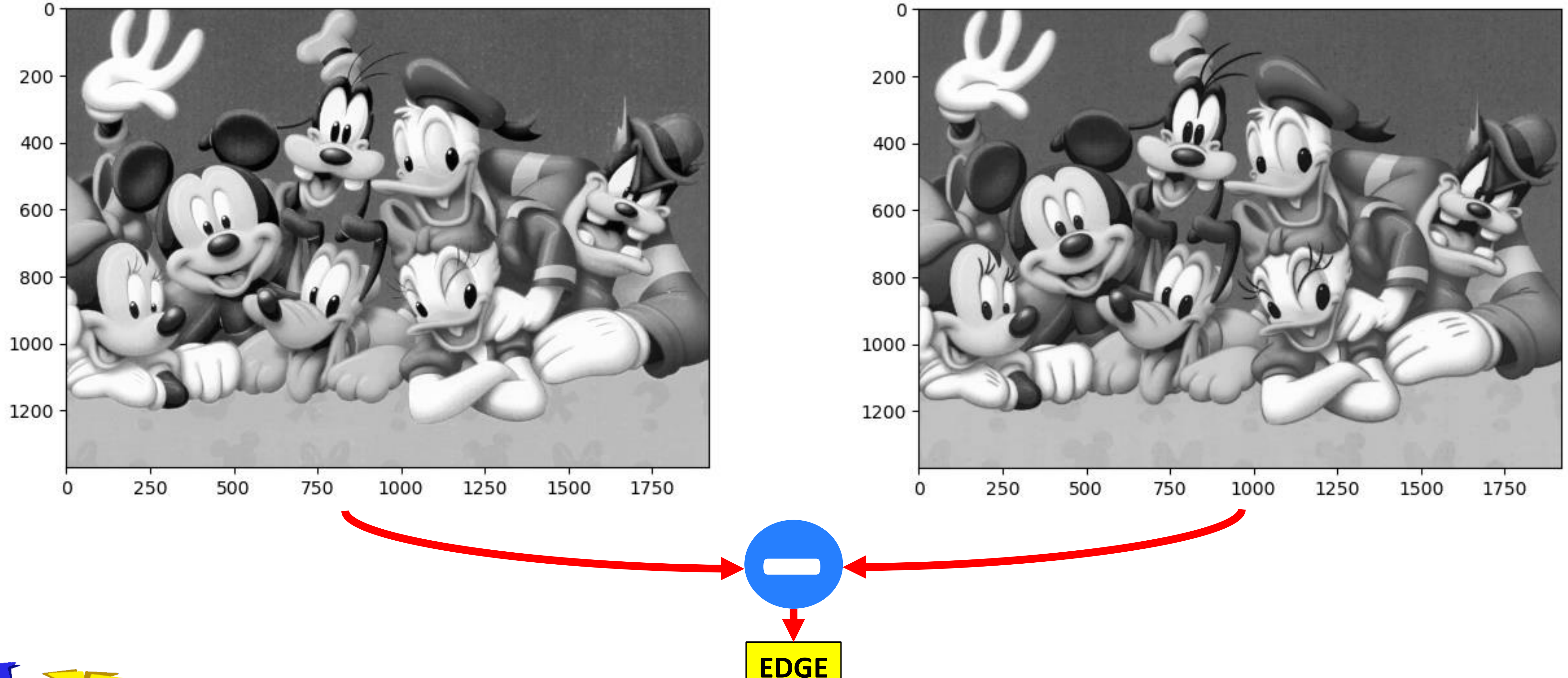
```
<matplotlib.image.AxesImage at 0x2ca4dd3ceb0>
```



Input



Dilate and Erode

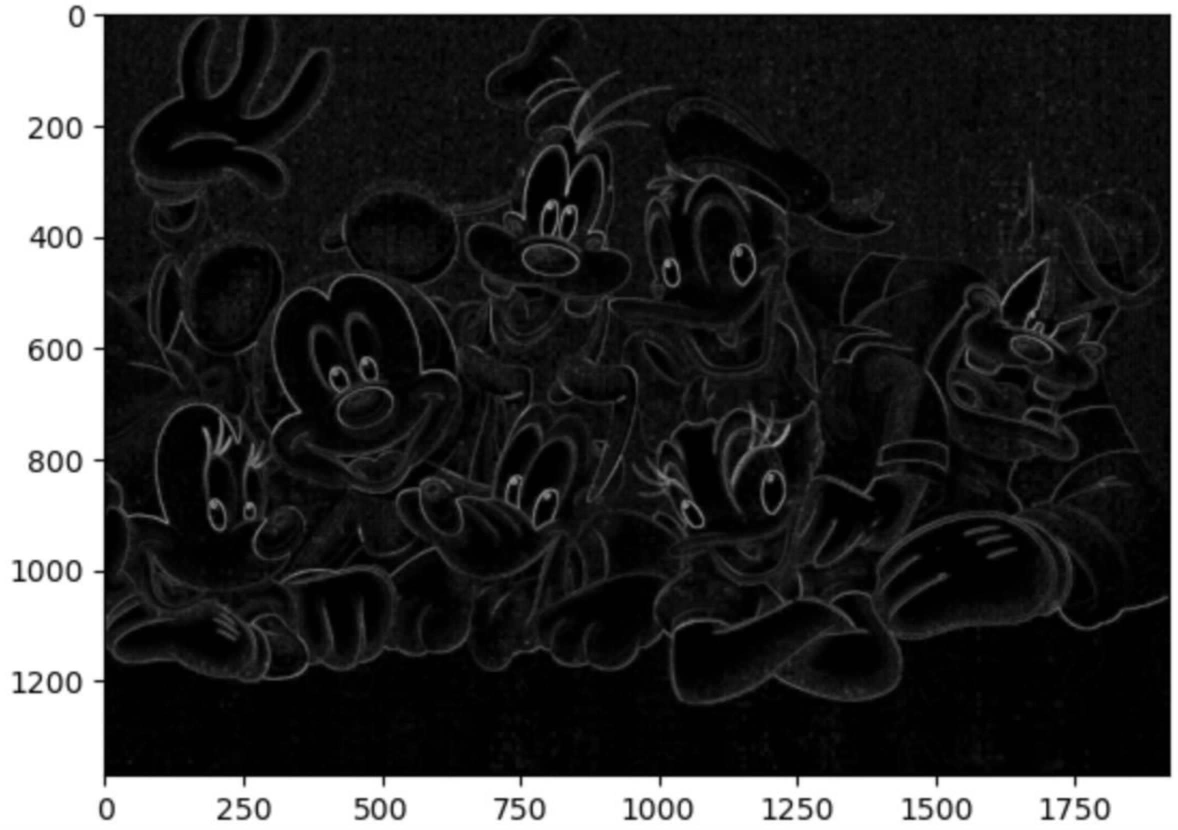


Python and OpenCV

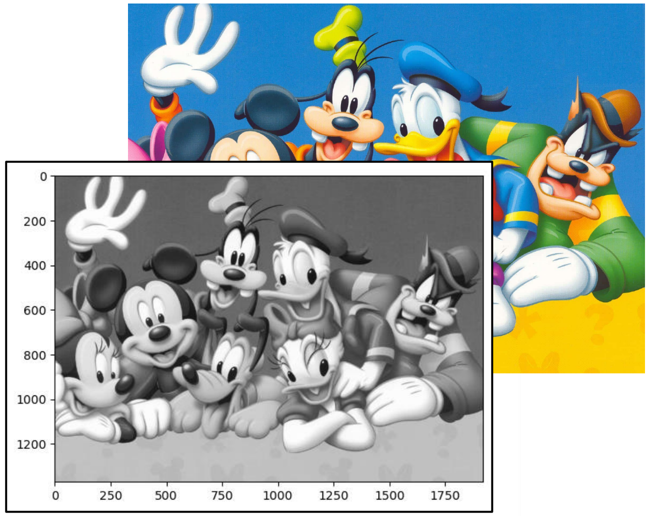
Example with **Gray scale** image

```
edge = dilation - erosion
plt.imshow(edge, cmap="gray")
```

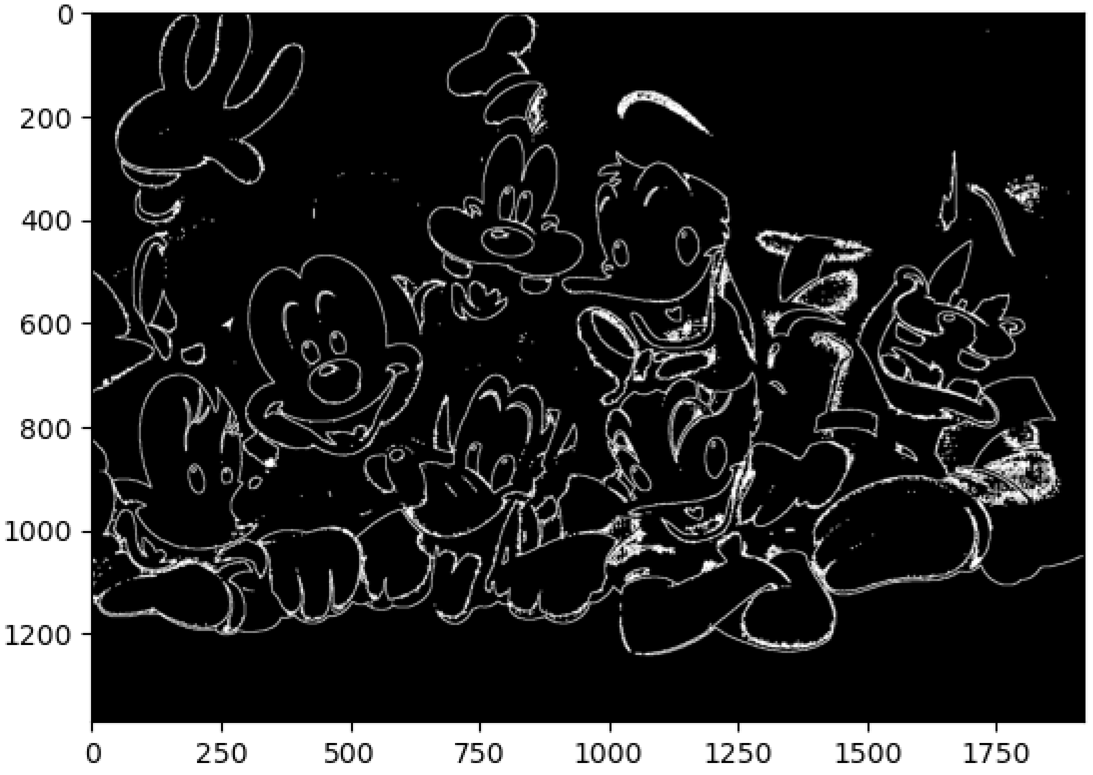
<matplotlib.image.AxesImage at 0x2ca4e0bc190>

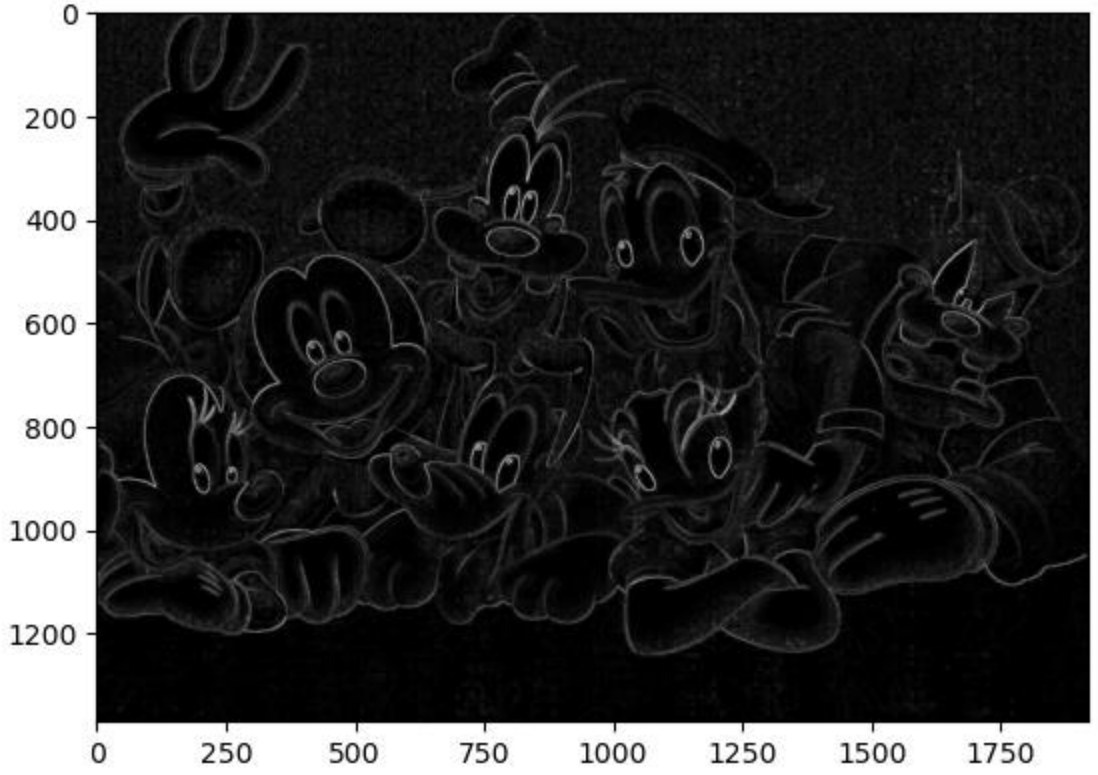


Input



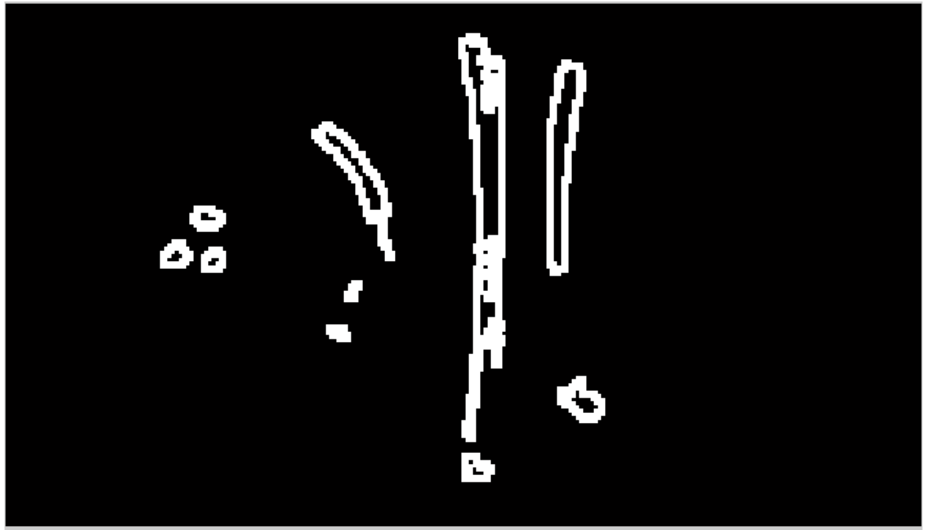
B&W Vs Gray scale



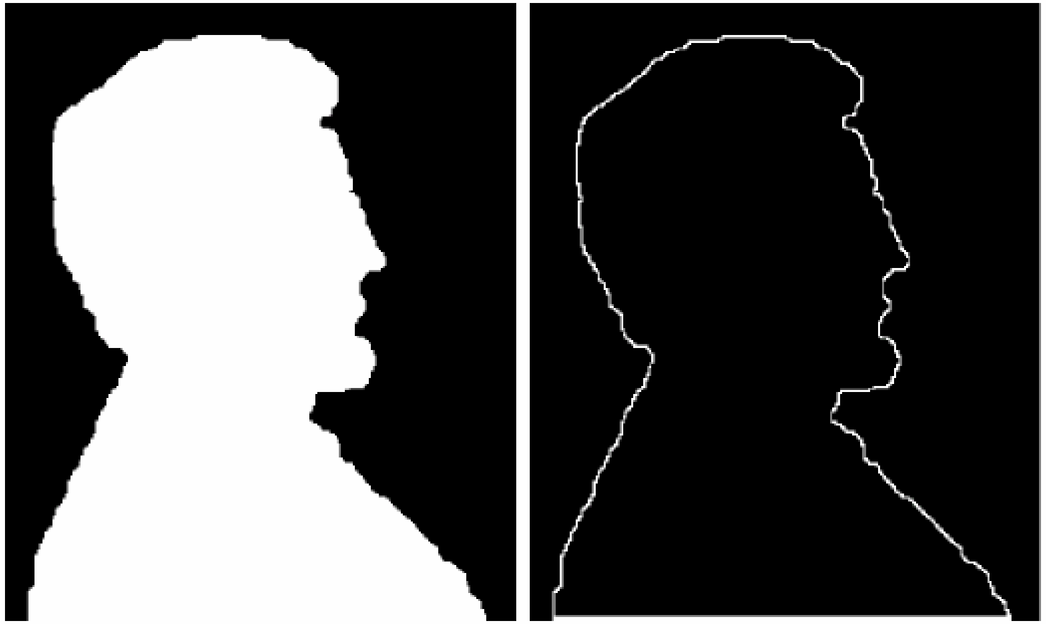


Edge extraction

Edge extraction



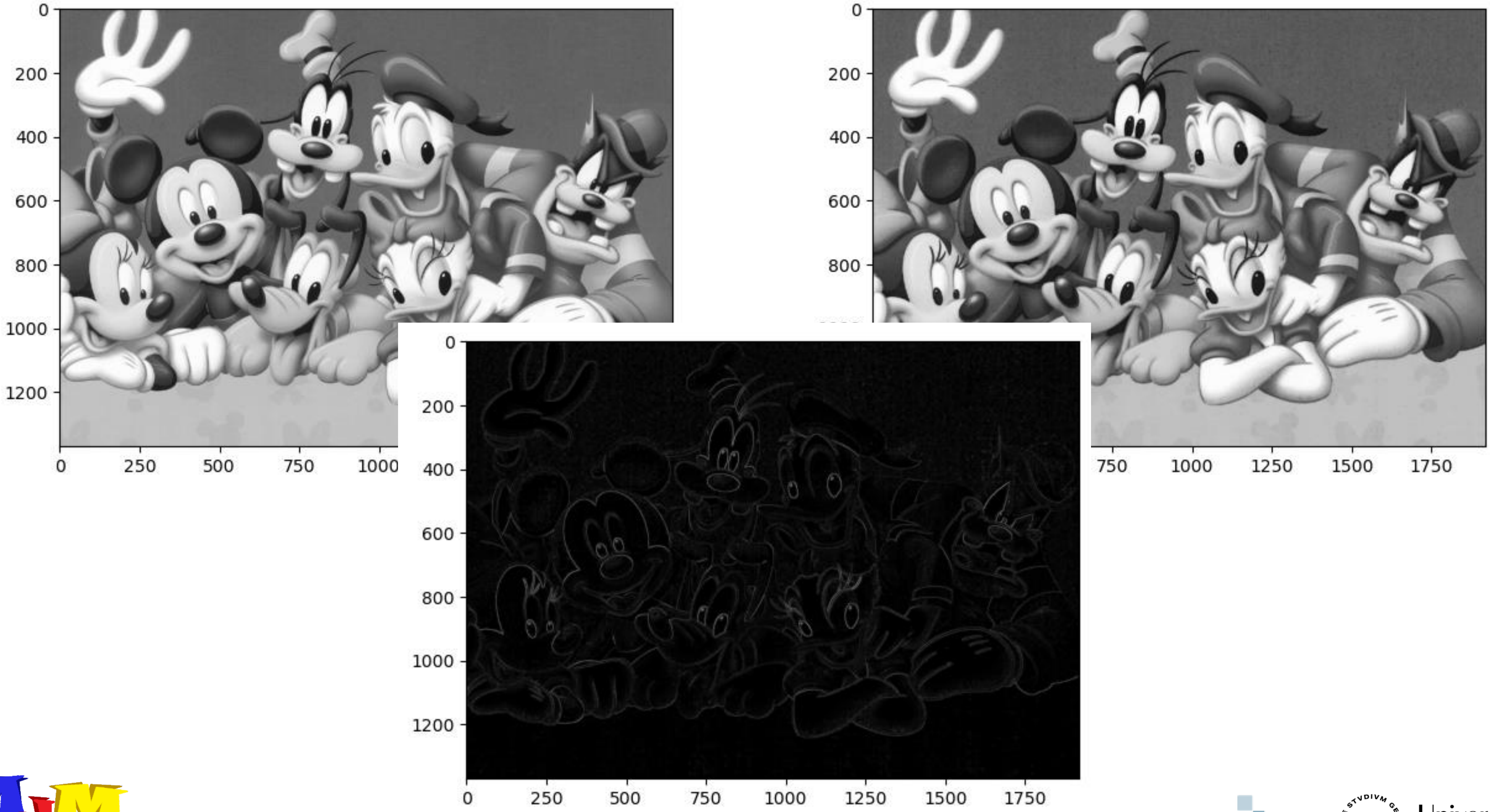
Edge extraction



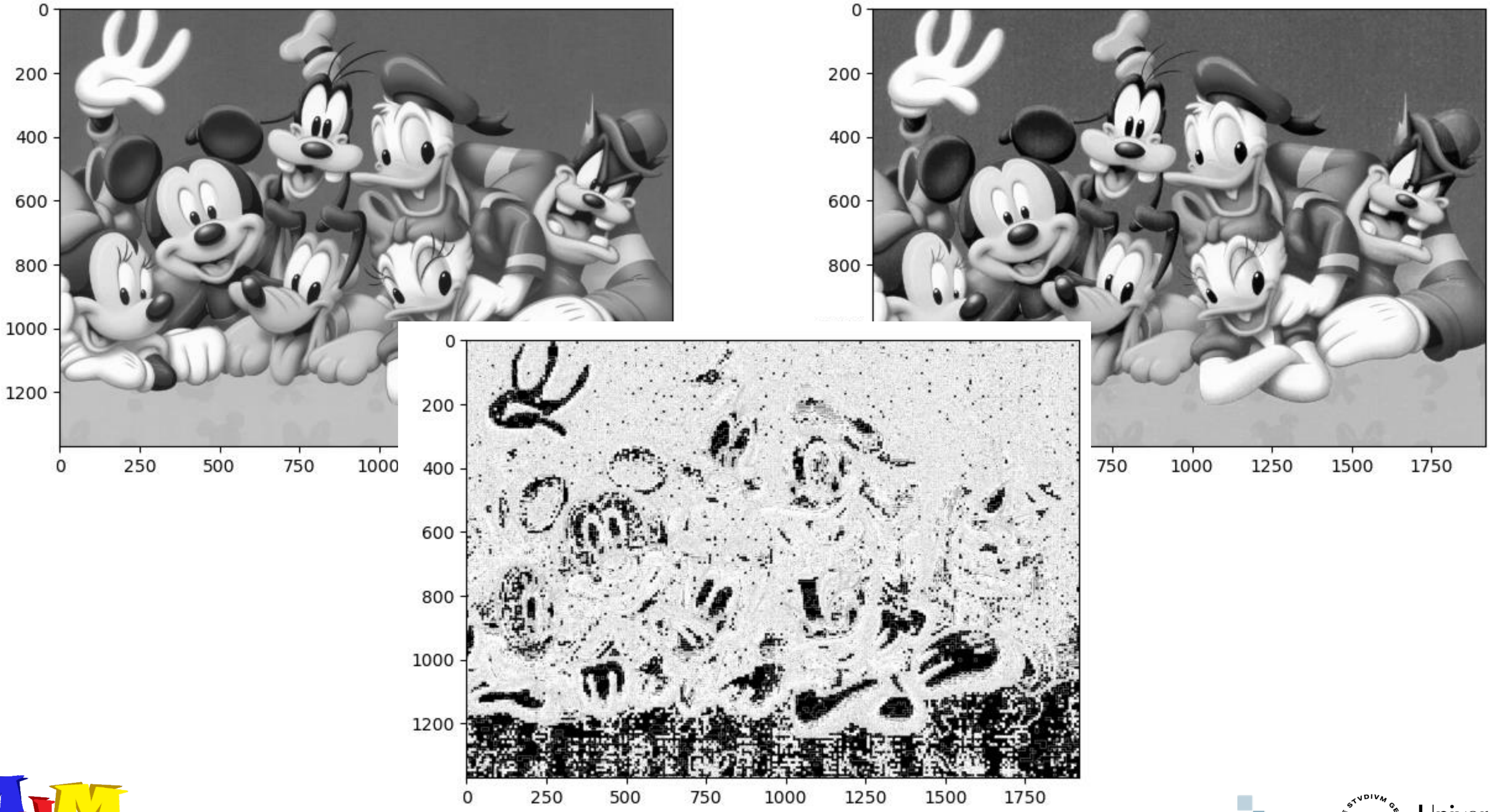
Contorni di A si ottiene facendo A - (A Θ B)

Python:

```
edge = img - erosion
```


0
200
400
600
800
1000
1200
0
250
500
750
1000
1250
1500
1750

l – dilation ?


0
200
400
600
800
1000
1200
0
250
500
750
1000
1250
1500
1750

Laplacian

Un'alternativa al gradiente morfologico è l'operatore laplaciano che tende a formare contorni chiusi.

$$l = (f \oplus b) + (f \ominus b) - 2f$$

- ✓Immagini binarie
- ✓Immagini a scala di grigio

```
A = imread ('image.jpg');
e = strel ('square', 15 );
L = imdilate(A , e) + imerode(A , e) – 2*A;
```

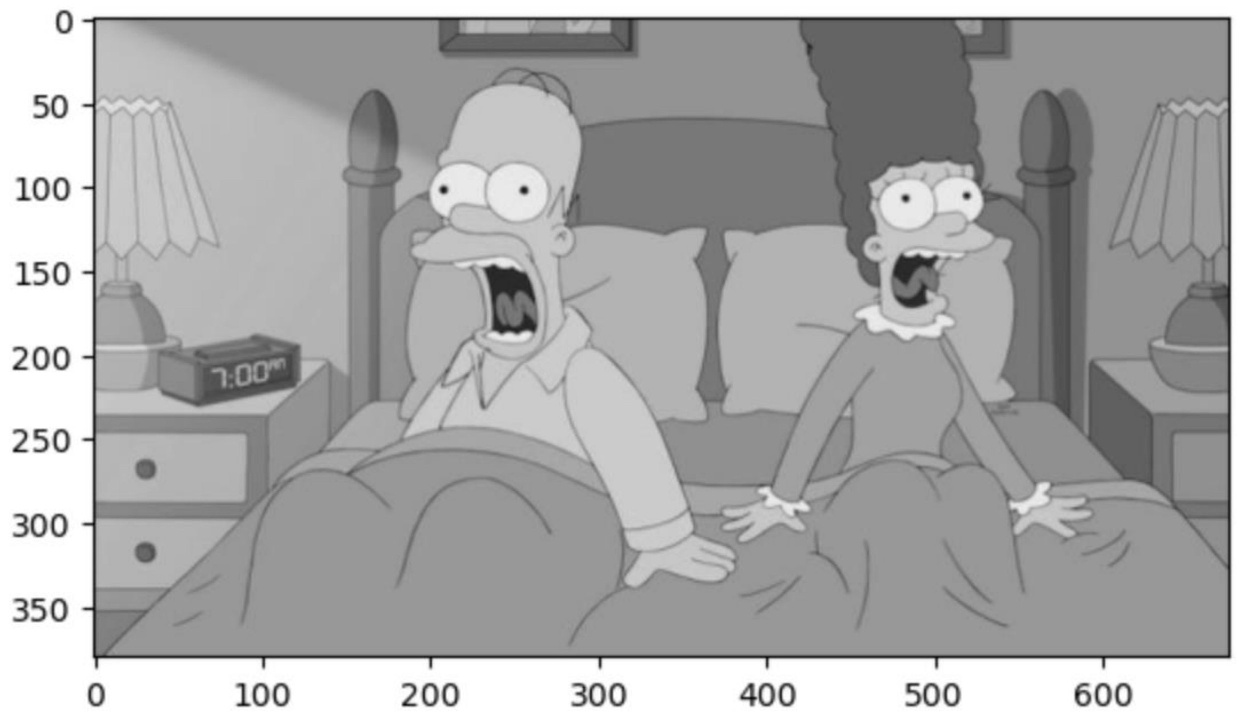
Python and OpenCV

Python and OpenCV

Example with Gray scale image

```
import numpy as np
img2 = cv2.imread("simpson.jpg",0)
plt.imshow(img2, cmap="gray")
```

```
<matplotlib.image.AxesImage at 0x2ca52a4e7a0>
```



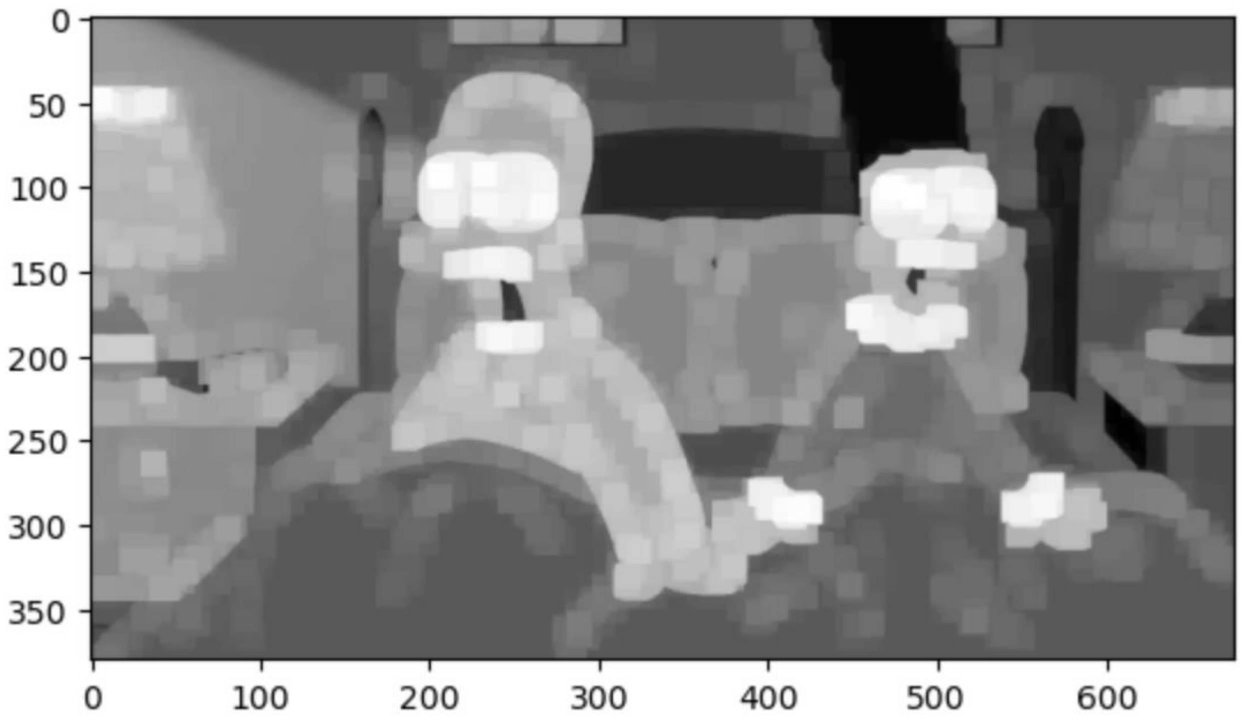
Input

Python and OpenCV

Example with Gray scale image

```
kernel = np.ones((15,15),np.uint8)
dilation = cv2.dilate(img2, kernel, iterations = 1)
plt.imshow(dilation, cmap="gray")
```

```
<matplotlib.image.AxesImage at 0x2ca55270e80>
```



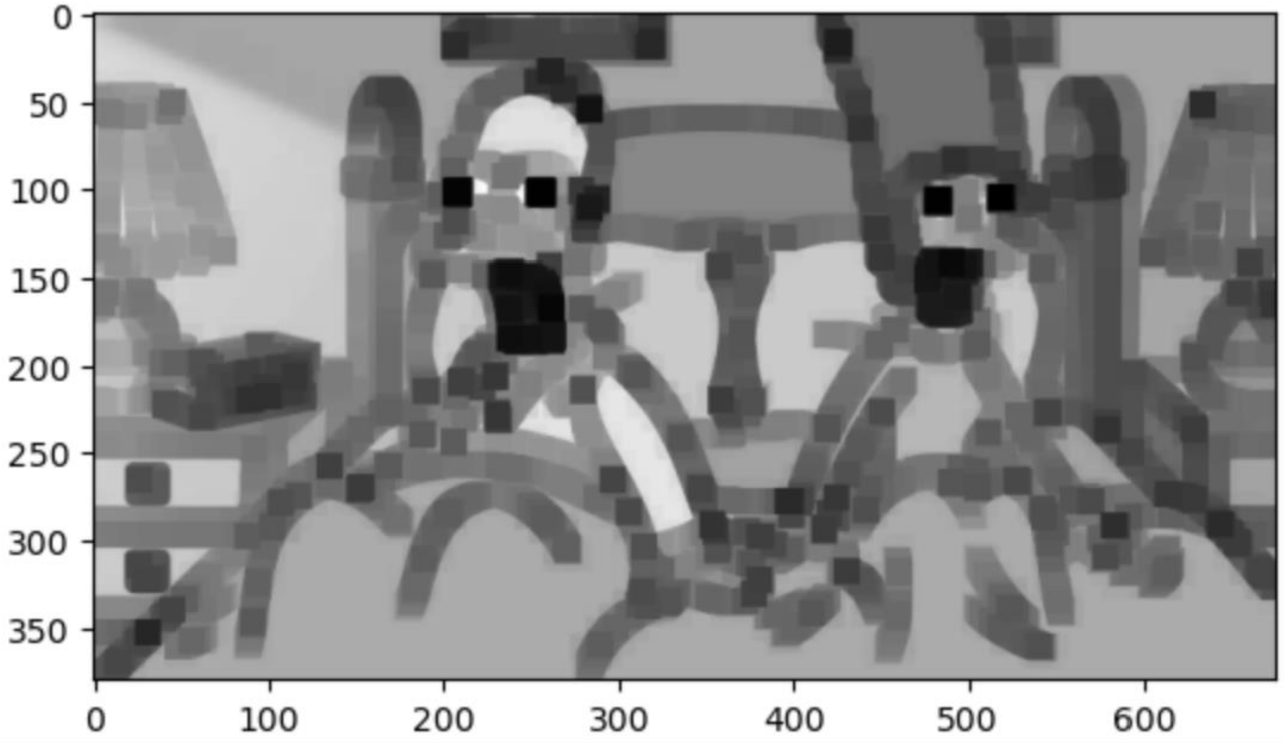
Input

Python and OpenCV

Example with Gray scale image

```
erosion = cv2.erode(img2, kernel, iterations = 1)
plt.imshow(erosion, cmap="gray")
```

```
<matplotlib.image.AxesImage at 0x2ca552e35e0>
```



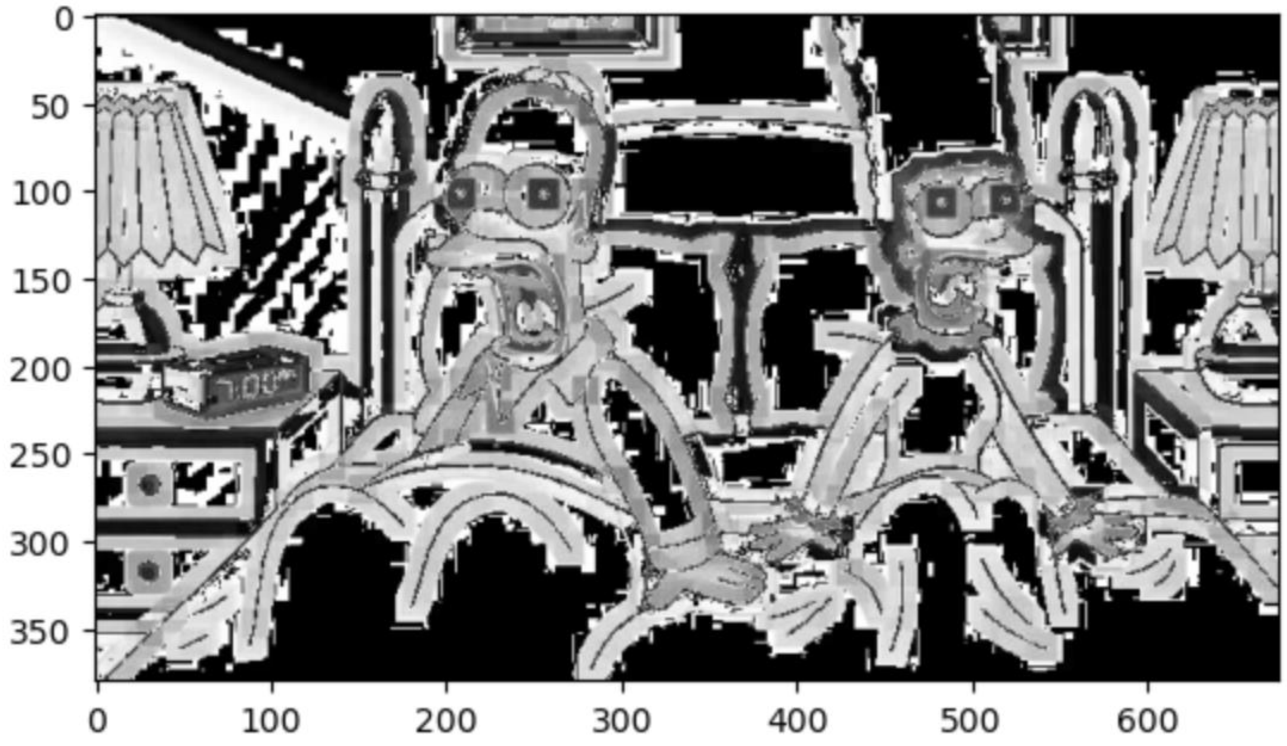
Input

Python and OpenCV

Example with Gray scale image

```
#Edge9
edge = dilation + erosion - 2*img2
plt.imshow(edge, cmap="gray")
```

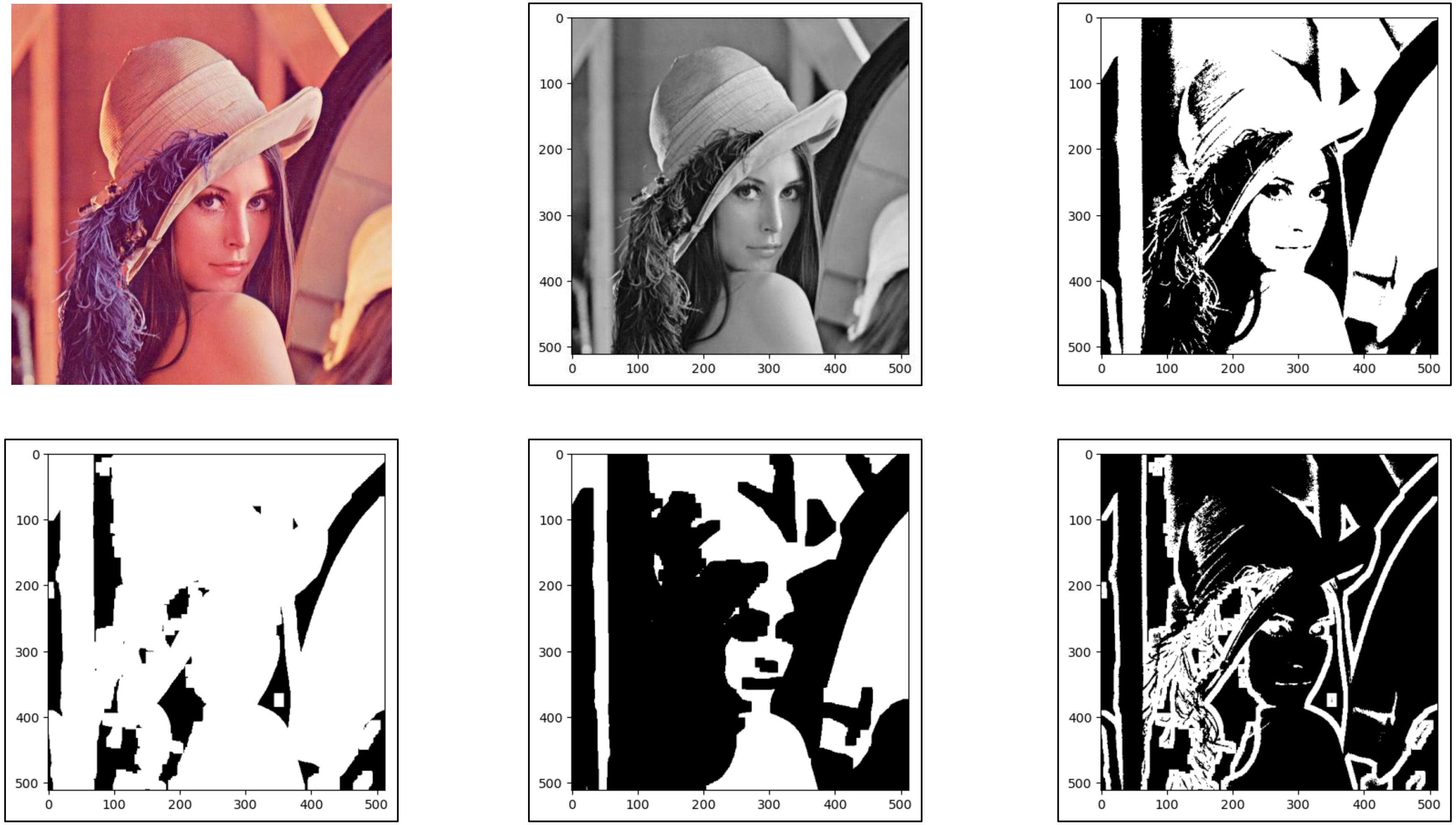
```
<matplotlib.image.AxesImage at 0x2ca55578310>
```



Input

Black and White

Lena



Skeletonization

Skeletonization is a morphological operation that reduces objects in a binary image to a set of thin lines that preserve relevant information about the shape of the object.

$$S(A) = \bigcup_{k=0}^{K} S_k(A) \qquad con\ S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$$

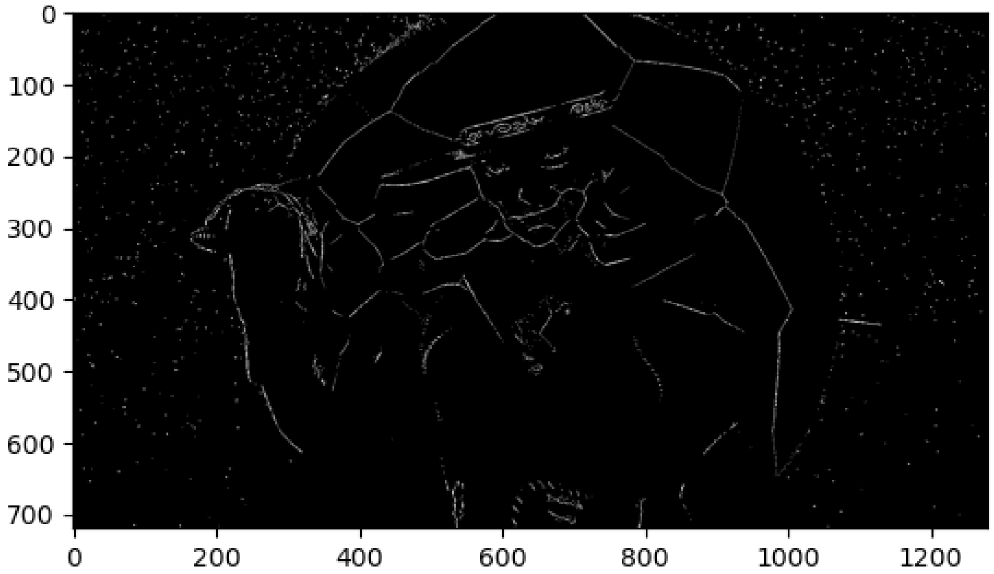
✓Binary image

```
A = imread ('skeletonization.jpg');
B = bwmorph (A , 'skel' , n);
```

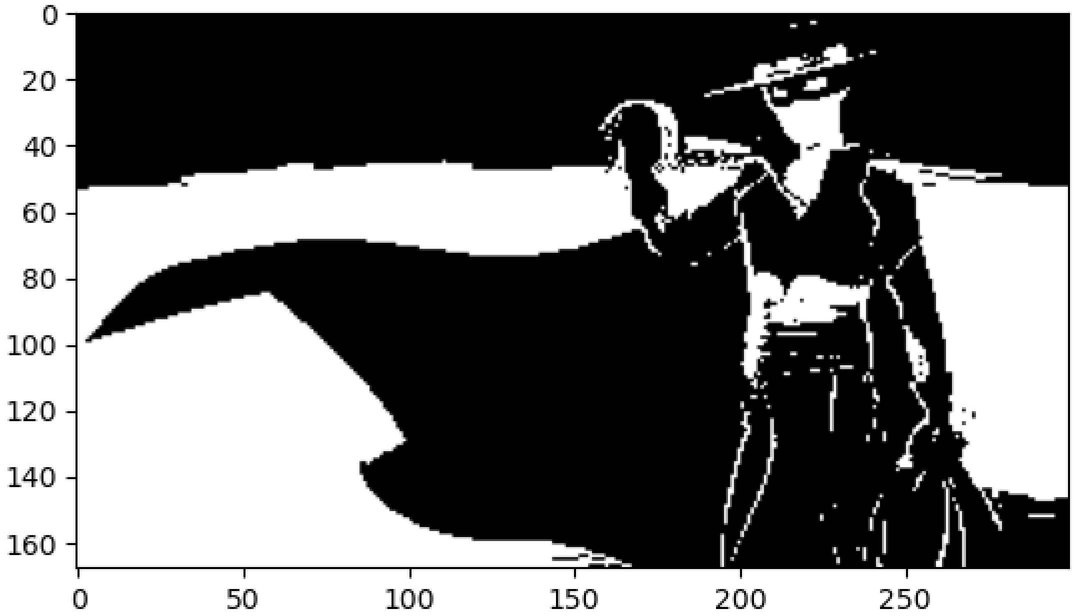
```python
import cv2
import numpy as np

img = cv2.imread('image.jpg',0)
size = np.size(img)
skel = np.zeros(img.shape,np.uint8)

ret,img = cv2.threshold(img,127,255,0)
element = cv2.getStructuringElement(cv2.MORPH_CROSS,(3,3))
print("Element ", element)
done = False

while( not done):
    eroded = cv2.erode(img,element)
    temp = cv2.dilate(eroded,element)
    temp = cv2.subtract(img,temp)
    skel = cv2.bitwise_or(skel,temp)
    img = eroded.copy()

    zeros = size - cv2.countNonZero(img)
    if zeros==size:
        done = True

plt.imshow(skel, cmap="gray")
```

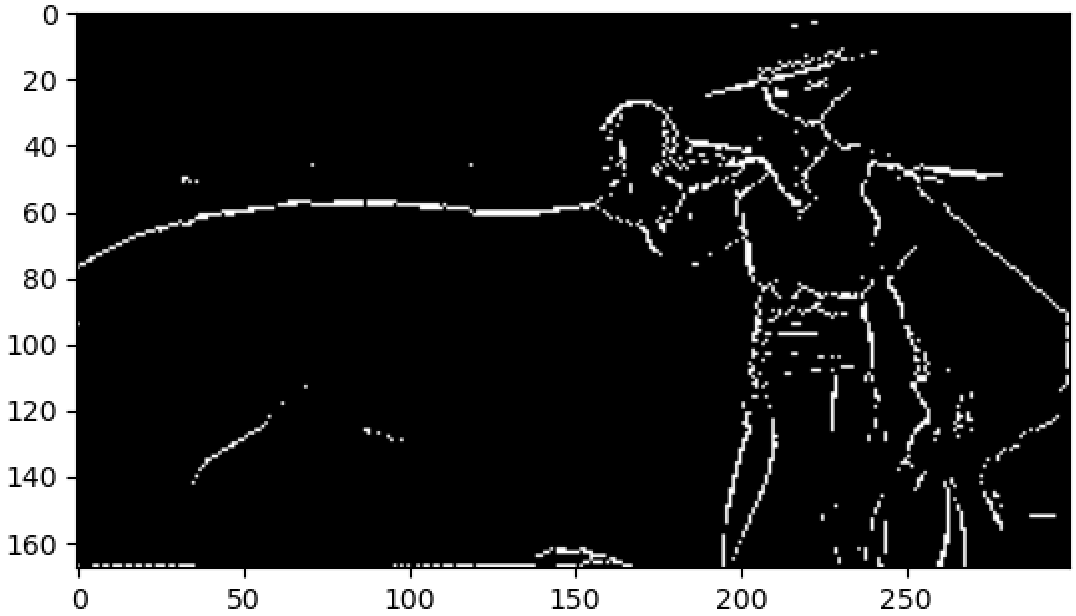
Gray scale


0
100
200
300
400
500
600
700
0
200
400
600
800
1000
1200

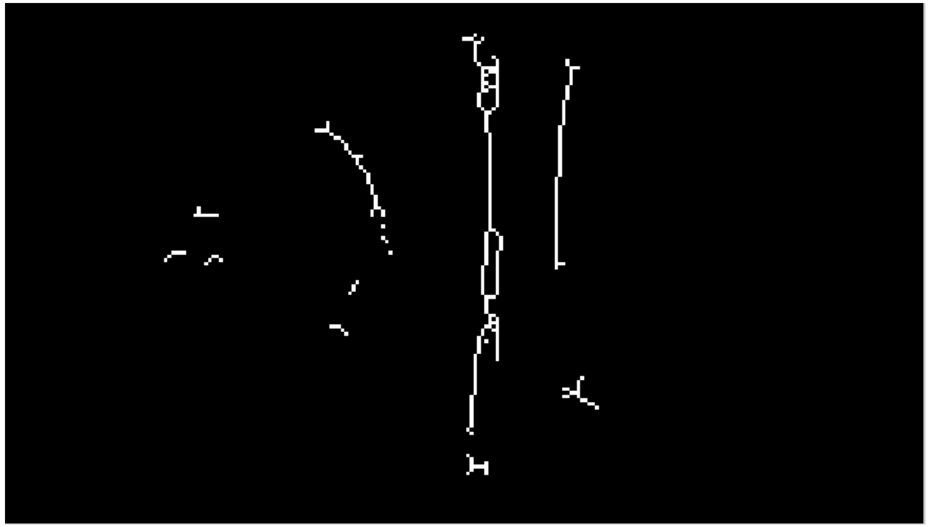

0
100
200
300
400
500
600
700
0
200
400
600
800
1000
1200

Black and White





Skeletonization



Other Morphological Operators

Luca Guarnera

An introduction on Deepfakes Creation and Detection Approaches

Bridge

It searches for groups of unconnected pixels and joins them by changing 0 to 1.

✓ Binary images

ESEMPIO

1	0	0
1	0	1
0	0	1

diventa →

1	0	0
1	1	1
0	0	1

```
A = imread ('image.png');
B = bwmorph (A , 'bridge' , 1);
```

Bridge

Clean

This operation removes isolated pixels. For example, a single 1 surrounded by all 0s.

✓ Binary images

```
A = imread ('image.png');
B = bwmorph (A , 'clean' , 1);
```

ESEMPIO

0	0	0
0	1	0
0	0	0

Clean

Pixel Filling

This operation allows isolated pixels located within a binary image to be filled.

ESEMPIO

1	1	1
1	0	1
1	1	1

- ✓ Binary images

```
A = imread ('pixel.jpg');
B = bwmorph (A , 'fill' , 1);
```

Pixel Filling

Morphological Operators for Gray-Scale Images

Image Processing Laboratory

Università di Catania

Extensions to grayscale images

It is possible to generalize mathematical morphology techniques to gray-level images. In this case:

- $f(x, y)$: input image;
- $b(x, y)$: a structural element (a subimage);
- (x, y): integer coordinates.

f and b are functions that assign a gray level to each distinct pair of integer coordinates.

Dilatation

$$(f \oplus b)(s, t) = \max\{f(s - x,\ t - y) + b(x, y) \mid (s - x) \in D_f, (t - y) \in D_f, (x, y) \in D_b\}$$

- where D_f, D_b represent the domains of f and b, respectively.
- If all values of the structural element are positive the output image tends to be lighter than the input. Dark details are reduced or eliminated depending on their value and the shape and value of b.

Erosion

$$(f \ominus b)(s,t) = \min\{f(s+x, t+y) - b(x,y) \mid (s+x) \in D_f, (t+y) \in D_f, (x,y) \in D_b\}$$

- where D_f , D_b represent the domains of f and b, respectively.
- If all values of the structural element are positive, the image tends to be darker than the input. One can control the degree of lightening of small light details depending on their value and the shape and value of b.

Esempi

Original

Dilation

Erosion

Examples

- Smoothing: $g = ((f \circ b) \bullet b)$
- Gradient: $g = (f \oplus b) - (f \ominus b)$
- Laplacian: $g = (f \oplus b) + (f \ominus b) - 2f$

Dilation Erosion Smoothing Gradient Laplacian

Morphology Mathematics

Luca Guarnera, Ph.D.

Research Fellow

luca.guarnera@unict.it

University of Catania
Dipartimento di Economia e Impresa