



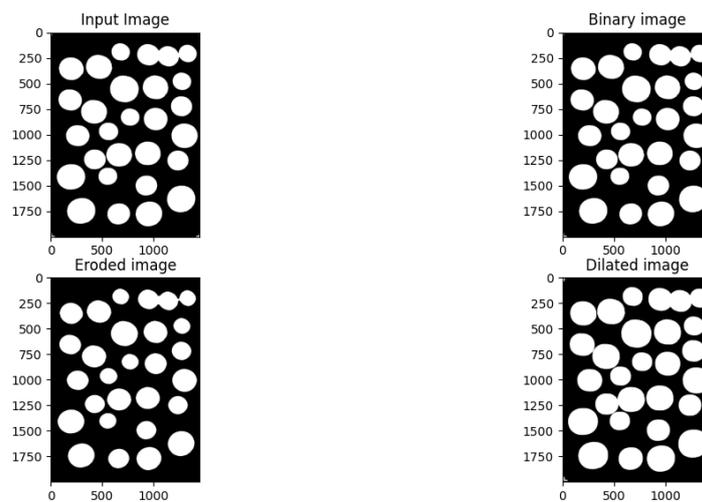
Manara university
Engineering college
Faculty of Informatics engineering
Computer vision
2025 – 2026

Lab session 8: Morphological operations (Dilation and erosion)

Tasks:

1. Read (circles.jpg) image in the gray scale.
2. Define the kernel ones matrix of the size [15, 15].
3. Apply the erosion operation on the image .
4. Apply the dilation operation on the image .
5. Apply the erosion operation followed by the dilation operation to open the image .
6. Display the results .

Results :



Code :

```
path = r'\\'  
img = cv2.imread(path+'circles.jpg', 0)  
  
# Taking a matrix of size 15 as the kernel  
kernel = np.ones((15, 15), np.uint8)
```

```
img_erosion = cv2.erode(img, kernel, iterations=1)
img_dilation = cv2.dilate(img, kernel, iterations=1)
img_open = cv2.morphologyEx(img, cv2.MORPH_OPEN, kernel)

fig = plt.figure(figsize = (12, 12))
ax1 = fig.add_subplot(2, 2, 1)
ax1.imshow(img, cmap='gray')
ax1.title.set_text('Input Image')

ax2 = fig.add_subplot(2, 2, 2)
ax2.imshow(img_erosion, cmap='gray')
ax2.title.set_text('eroded image')

ax3 = fig.add_subplot(2, 2, 3)
ax3.imshow(img_dilation, cmap='gray')
ax3.title.set_text('dilated image')

ax4 = fig.add_subplot(2, 2, 4)
ax4.imshow(img_open, cmap='gray')
ax4.title.set_text('opened image')

plt.show()
```

Code from scratch :

```
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt

# Reading the image
path = r'\\'
img = cv.imread(path+'circles.jpg', 0)
# Parameters
th = 100
r, c = img.shape
kernel = np.ones((15, 15))
x, y = kernel.shape
binary = np.zeros((r, c))
eroded_img = np.zeros((r, c))
dilated_img = np.zeros((r, c))
# Calculation of the ones in the element structure (regarding to the
central pixel value)
if kernel[x//2, y//2] == 1:
    ref = int(np.sum(kernel))-1
else:
    ref = int(np.sum(kernel))
# converting the image from grayscale into binary one
for i in range(r):
    for j in range(c):
        if img[i, j] > th:
            binary[i, j] = 1
# Erosion and Dilation operation
for i in range(x//2, r-(x//2)):
    for j in range(y//2, c-(y//2)):
        img_window = binary[i-(x//2):i+(x//2)+1, j-(y//2):j+(y//2)+1]
        if kernel[x//2, y//2] == 1:
            intersection = int( np.sum(cv.bitwise_and(img_window, kernel)) ) -
1
        else:
```

```
intersection = int( np.sum(cv.bitwise_and(img_window, kernel)) )
if intersection == ref:
    eroded_img [i, j] = 255
if intersection >= 1:
    dilated_img [i, j] = 255

fig = plt.figure(figsize = (12, 12))
ax1 = fig.add_subplot(2, 2, 1)
ax1.imshow(img, cmap='gray')
ax1.title.set_text('Input Image')

ax2 = fig.add_subplot(2, 2, 2)
ax2.imshow(binary, cmap='gray')
ax2.title.set_text('Binary image')

ax3 = fig.add_subplot(2, 2, 3)
ax3.imshow(eroded_img, cmap='gray')
ax3.title.set_text('Eroded image')

ax4 = fig.add_subplot(2, 2, 4)
ax4.imshow(dilated_img, cmap='gray')
ax4.title.set_text('Dilated image')

plt.show()
```