Manara university
Engineering college

Faculty of Informatics engineering

Computer vision

2025 - 2026

Lab session 5:
Spatial filters (Average and Gaussian)

Tasks:

1. Design a function called (convolution) that takes two parameters: (img, mask) and return an 2D array as the same size of the (img) array, and represents the result of convolution of (img) with (mask).
2. Read noisy image in the gray scale.
3. Define mask1 as:

$$mask1 = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

4. Define mask2 as:

$$mask2 = \frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

5. Define mask3 as:

$$mask3 = \begin{bmatrix} 0.0318 & 0.0375 & 0.0397 & 0.0375 & 0.0318 \\ 0.0375 & 0.0443 & 0.0469 & 0.0443 & 0.0375 \\ 0.0397 & 0.0469 & 0.0495 & 0.0469 & 0.0397 \\ 0.0375 & 0.0443 & 0.0469 & 0.0443 & 0.0375 \\ 0.0318 & 0.0375 & 0.0397 & 0.0375 & 0.0318 \end{bmatrix}$$

6. Filter the noisy image using convolution function and the mask1.
7. Filter the noisy image using convolution function and the mask2.
8. Filter the noisy image using convolution function and the mask3.
9. Display the results.

Needed Syntaxes:

 # Definition of a function in python language

```
def convolution (img, mask):
        # statements
        convolved_img = np.zeros( (r1, c1), dtype=float )
        # …
        Return convolved_img
```

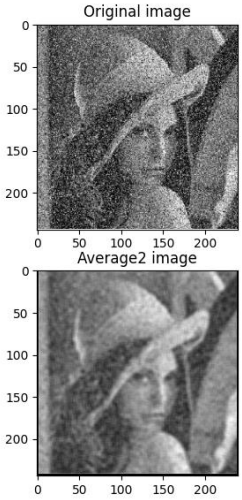
```
Mask1 = (1/9) * np.array( [[1, 1, 1], [1, 1, 1], [1, 1, 1]] )

print( len(Mask1) )                # print the number of Mask1 rows.

 Print( len (Mask1) // 2 )         # print the integer division of Mask1's rows by 2.

print ( np.sum(Mask1) )    )       # print the sum of Mask's elements.

print( img[0:3, 0:3] )    )        # cropping the 1st  three rows and columns
                                    from img array.
```

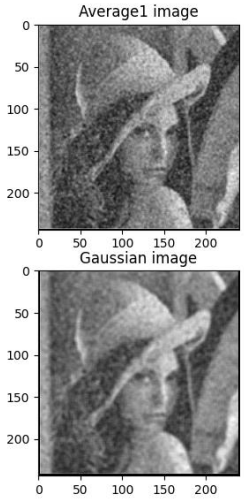
Code:

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

def convolution(img, mask):
    r1, c1 = img.shape
    r2, c2 = mask.shape
    convolved_img = np.zeros((r1, c1), dtype=float)

    for i in range(r2//2,r1-r2//2):
        for j in range(c2//2,c1-c2//2):
            window = img[i-r2//2:i+r2//2+1, j-c2//2:j+c2//2+1]
            convolved_img[i, j] = np.sum( window * mask )
    return convolved_img

path =r'\\'
img = cv.imread(path+'noisy.png', 0)

Mask1 = (1/9) * np.array( [[1, 1, 1], [1, 1, 1], [1, 1, 1]] )
convolved_img1 = convolution(img, Mask1)

Mask2 = (1/25) * np.ones((5, 5))
convolved_img2 = convolution(img, Mask2)
```

```
Mask3 = (1/9) * np.array( [[0.0318, 0.0375, 0.0397, 0.0375, 0.0318],
                           [0.0375, 0.0443, 0.0469, 0.0443, 0.0375],
                           [0.0397, 0.0469, 0.0495, 0.0469, 0.0397],
                           [0.0375, 0.0443, 0.0469, 0.0443, 0.0375],
                           [0.0318, 0.0375, 0.0397, 0.0375, 0.0318]] )
convolved_img3 = convolution(img, Mask3)

fig, axis = plt.subplots(2, 2)
axis[0][0].set_title('Original image')
axis[0][0].imshow(img, cmap='gray')

axis[0][1].set_title('Average1 image')
axis[0][1].imshow(convolved_img1, cmap='gray')

axis[1][0].set_title('Average2 image')
axis[1][0].imshow(convolved_img2, cmap='gray')

axis[1][1].set_title('Gaussian image')
axis[1][1].imshow(convolved_img3, cmap='gray')
plt.show()
```

Results:


Original image
Average2 image


Average1 image
Gaussian image