Mansoura University
Faculty of Computers and Information
Department of Information Technology
Second Semester- 2024-2025

[IT324P] Digital Image Processing

Grade: 3 IT

Aya Ayad

Outline

- Neighborhood operations: Filters
  - **Spatial Domain Methods**
    1. Linear Spatial Filter
       - Common Linear Operations
         i. Convolution
         ii. Correlation
    2. Non-linear Spatial Filter
  - **Smoothing Spatial Filters**

Neighborhood Operations

- Neighborhood operations are image processing techniques where the value of each pixel in the output image is determined by applying a specific operation to a predefined group of surrounding pixels (called a neighborhood) in the input image.
- This neighborhood is typically centered on the target pixel and can be of any shape or size (commonly square, like 3×3, 5×5, etc.).
- These operations are used for a variety of image enhancements and analyses such as smoothing, sharpening, edge detection, and noise reduction.

STEPS OF ENHANCEMENT

1. **Select a Pixel (x, y):**
   Choose a target pixel location in the image where the operation will be applied.
2. **Define a Neighborhood:**
   Identify a predefined region of pixels surrounding the target pixel (for example, a 3×3 or 5×5 square centered at (x, y)).
3. **Apply the Operation:**
   Perform a specific operation (like averaging, finding the maximum, applying a mask/kernel, etc.) using the values of the pixels within this neighborhood.
4. **Store the Result:**
   Assign the result of this operation to the corresponding position (x, y) in the output image.
5. **Move to the Next Pixel:**
   Shift the neighborhood window to the next pixel position (usually one pixel to the right, moving row by row) and repeat steps 2–4.
6. **Repeat for the Entire Image:**
   Continue this process for all pixels in the image.

Filtering

- Filtering is a technique for modifying or enhancing an image like highlighting certain features or removing other features.
- Image filtering includes smoothing , sharping, and edge enhancement .
- Image filtering may be applied in either
  - Spatial domain
  - Frequency domain

Filtering

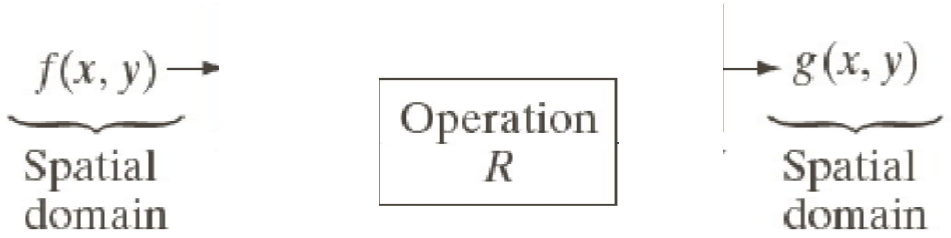
f(x, y)

Filtering

g(x, y)

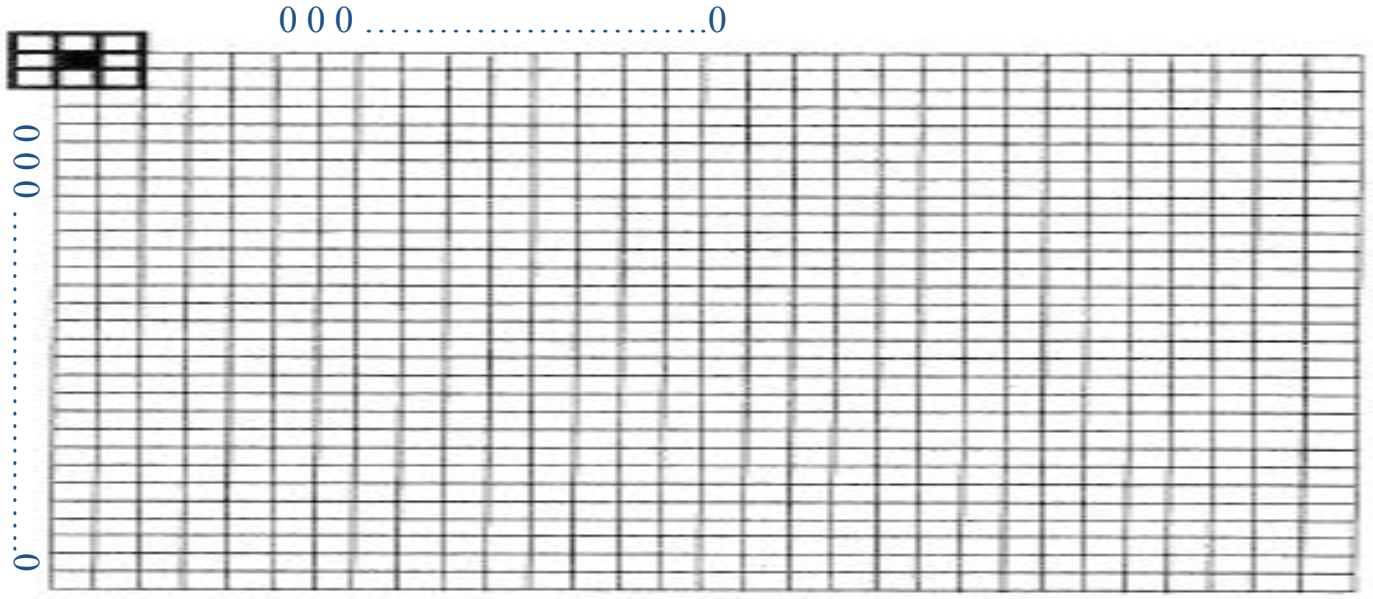
Image Filtering Methods

- **Spatial Domain**



Handling Pixels Close To Boundaries

Pad with zeroes



Spatial Domain Filters

There are two types:

1. Linear Spatial Filter

   1.1Convolution

   1.2 Correlation

2. Non-linear Spatial Filter

Convolution

Convolution kernel, ω

1	-1	-1
1	2	-1
1	1	1

Rotate 180°

1	1	1
-1	2	1
-1	-1	1

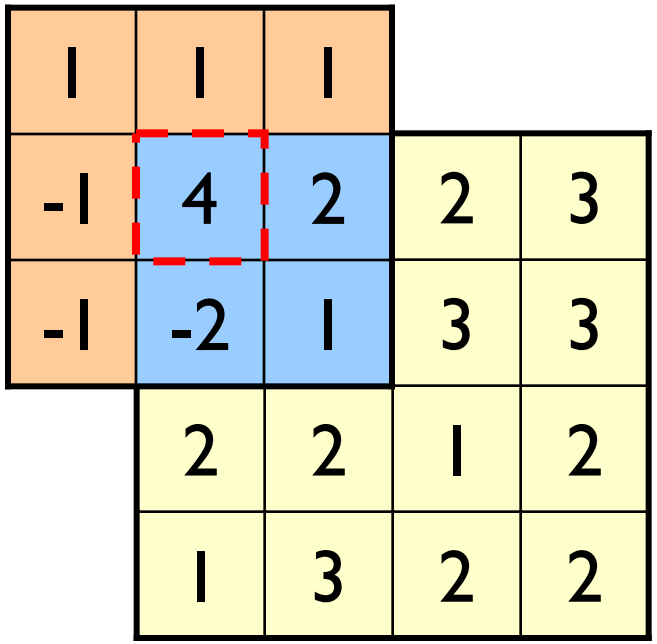
Input Image, f

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2

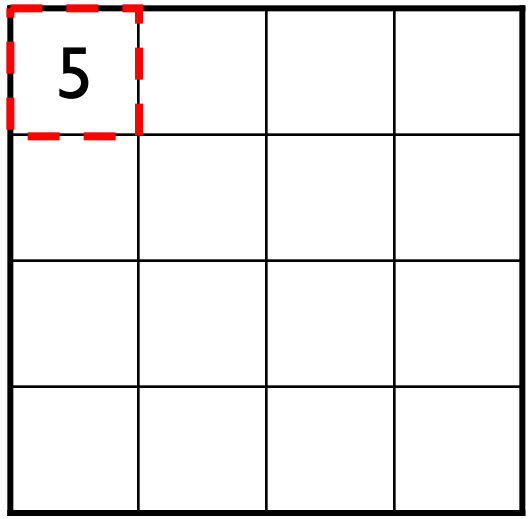
Convolution

1	1	1
-1	2	1
-1	-1	1

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2



Input Image, *f*



Output Image, *g*

Convolution

5	4	4	-2
9	6	14	5
11	7	6	5
9	12	8	5

Final output Image, *g*

Correlation

correlation kernel, ω

1	-1	-1
1	2	-1
1	1	1

Don't rotate use it directly

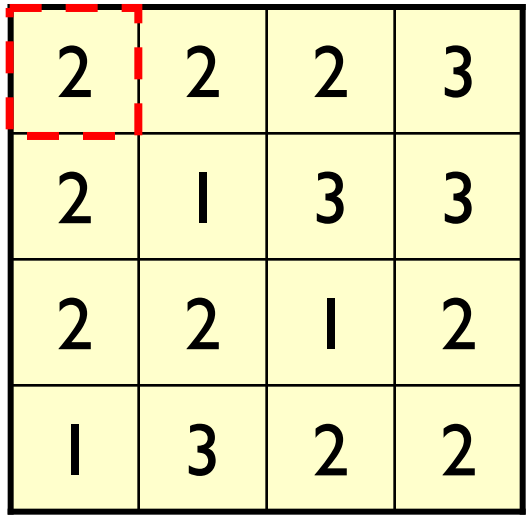
Input Image f

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2

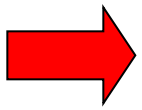
Correlation

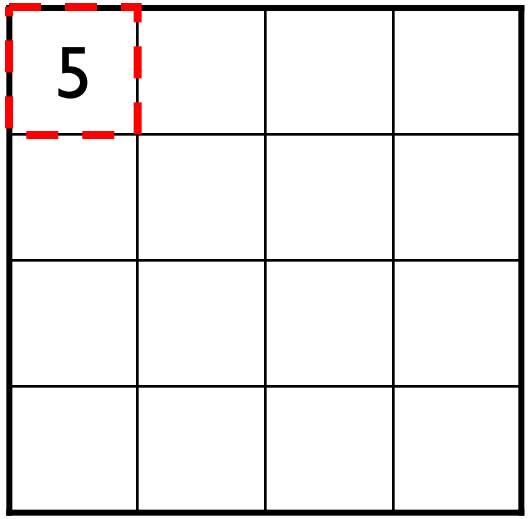
1	-1	-1
1	2	-1
1	1	1

2	2	2	3
2	1	3	3
2	2	1	2
1	3	2	2



Input Image, *f*





Output Image, *g*

Correlation

5	10	10	15
3	4	6	11
7	11	4	9
-5	4	4	5

Final output Image, *g*

Smoothing Spatial Filters

Smoothing Spatial Filters divided into two types

1. Smoothing Linear Filters

   a)Mean Filter

   - Average Filter
   - Weighted Filter
   - Gaussian Filter

2. Smoothing Non-Linear Filters

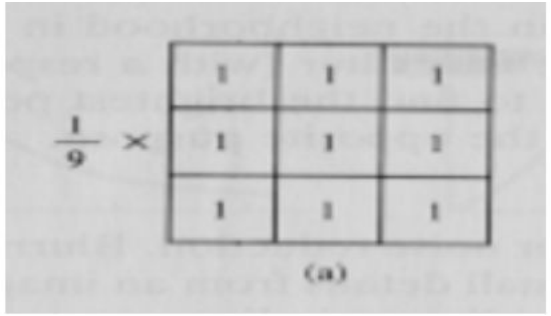
   a)Order statistics filter

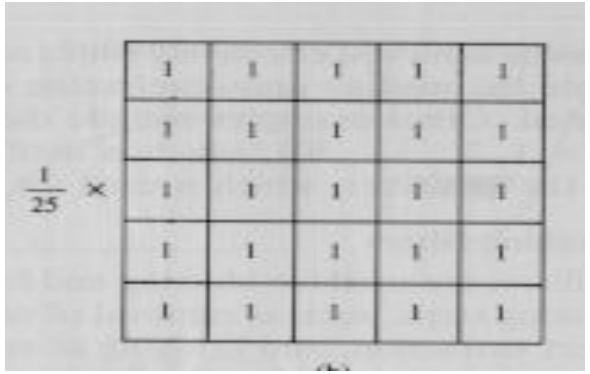
   - Median Filter
   - Maximum filter
   - Minimum filter

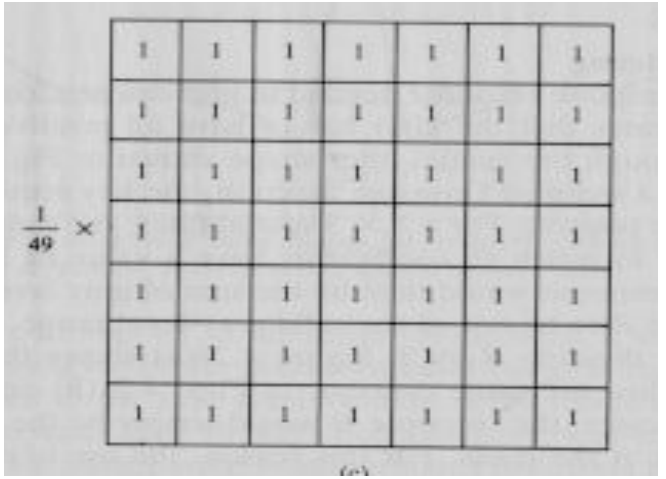
Smoothing Using Mean Filter

- Linear spatial filter is simply the average of the pixels contained in the neighborhood of the filter mask.

- The idea is replacing the value of every pixel in an image by the average of the grey levels in the neighborhood define by the filter mask.





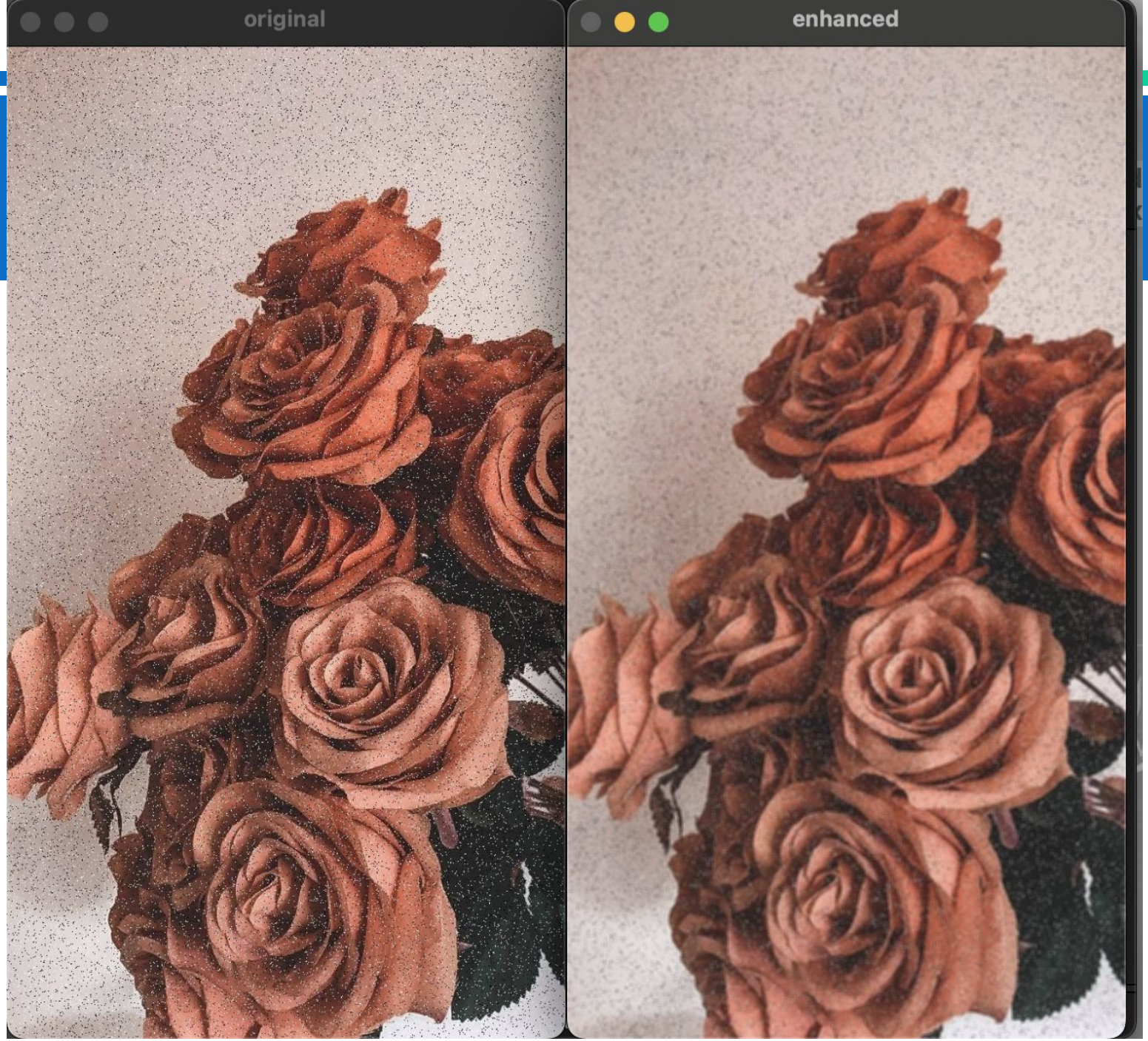


Types Of Mean Filter

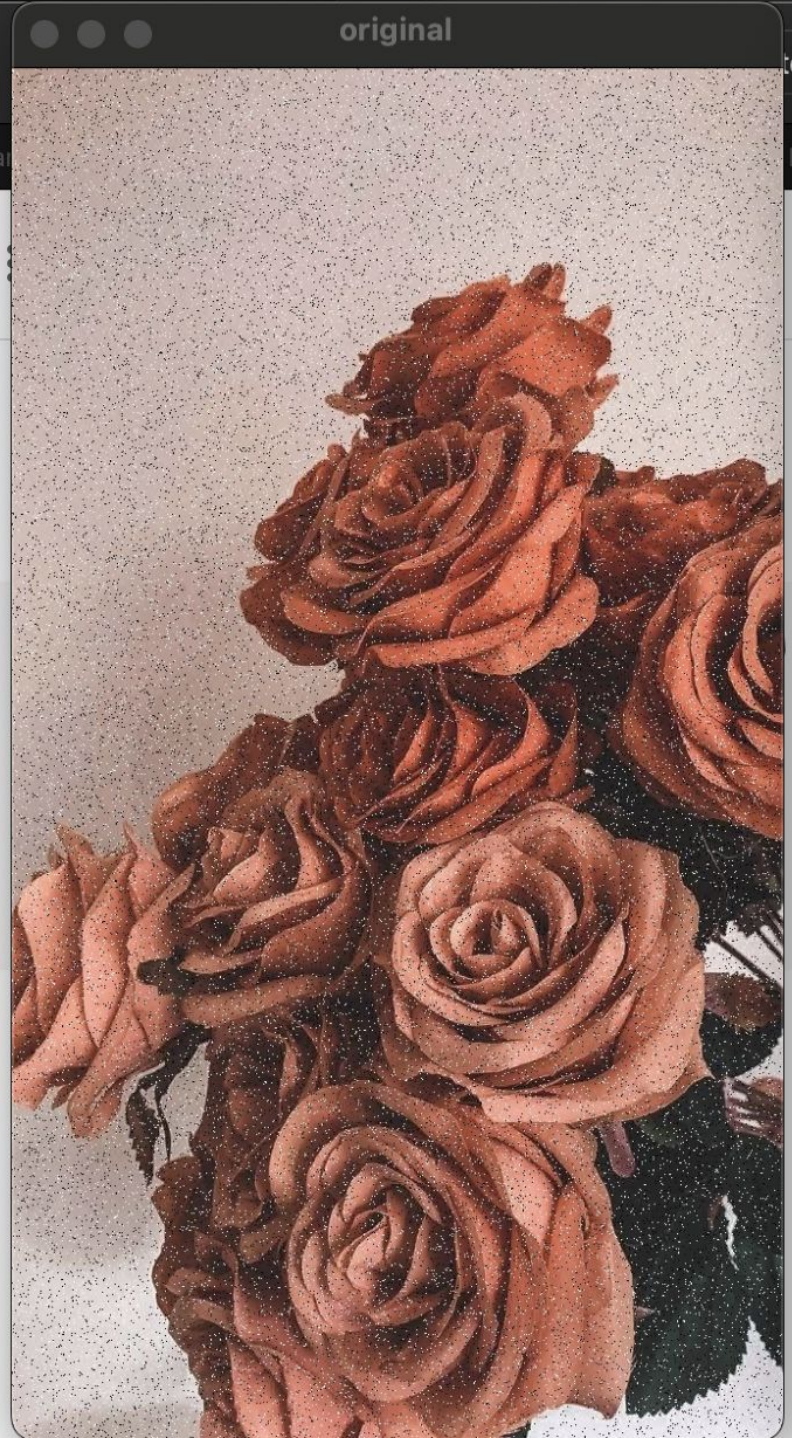
1. **Averaging filter (Box filter)**:
    1. This is the simplest type of mean filter, where all pixel values within the filter window are given equal weight (coefficients are all the same). The output pixel is the average of all pixels in the kernel**. This helps to blur the image and reduce fine details.**
2. **Weighted averaging filter**:
    1. In this type, the center pixel is given more weight than the surrounding pixels. The coefficients are not all the same, but rather they are chosen based on a predefined weight distribution. The center pixel often gets the highest weight, which makes this filter give more importance to the central pixel.
3. **Gaussian filter**:
    1. The Gaussian filter is a type of weighted averaging filter where the coefficients are determined by a Gaussian function, by averaging pixel values with a weighted Gaussian distribution. This is a commonly used filter for blurring or noise reduction in images**, especially for preserving edges better than the simple averaging filter.**

Average Filter(mean Filter)

```
import cv2
img=cv2.imread("noisy_img.jpg")
enhanced=cv2.blur(img,(5,5))
cv2.imshow("original",img)
cv2.imshow("enhanced",enhanced)
cv2.waitKey()
```


original
enhanced

```python
import cv2
img=cv2.imread("noi
with_padding=cv2.bl
borderType=cv2.BORD
enhanced2=cv2.blur(
cv2.imshow("origina
cv2.imshow("enhance
```



CODE OF WEIGHTED AVERAGING FILTER

ω_1	ω_2	ω_3
ω_4	ω_5	ω_6
ω_7	ω_8	ω_9

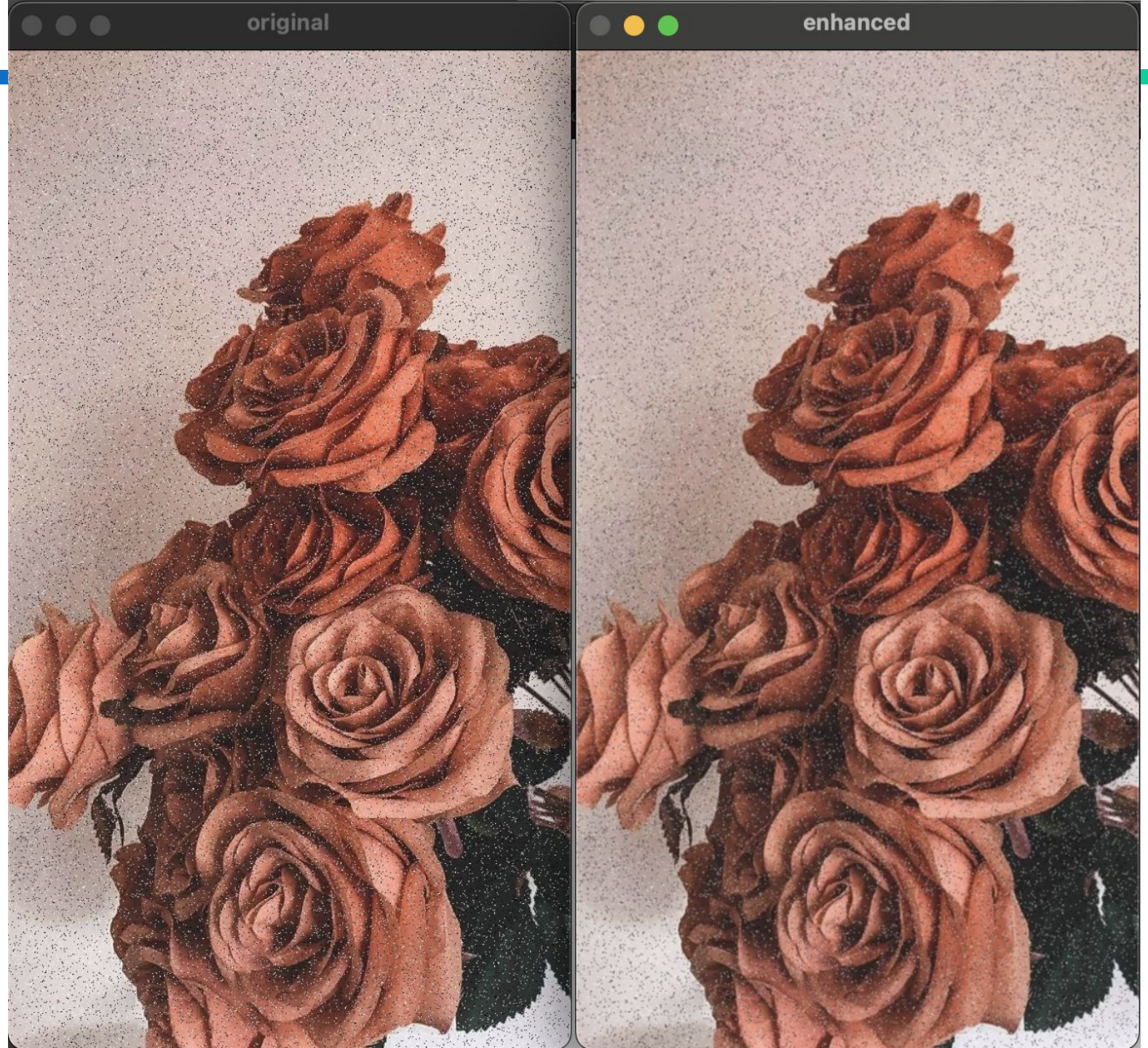
(a) A representation of a general 3×3 filter mask

$$\frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

(b) A 3×3 weighted average filter mask

CODE OF WEIGHTED AVERAGING FILTER

```
import numpy as np
img=cv2.imread("noisy_img.jpg")
kernel=np.array([[1,2,1],
                 [2,4,2],
                 [1,2,1]])
k=kernel/np.sum(kernel)
enhanced=cv2.filter2D(img,-1,k)
```

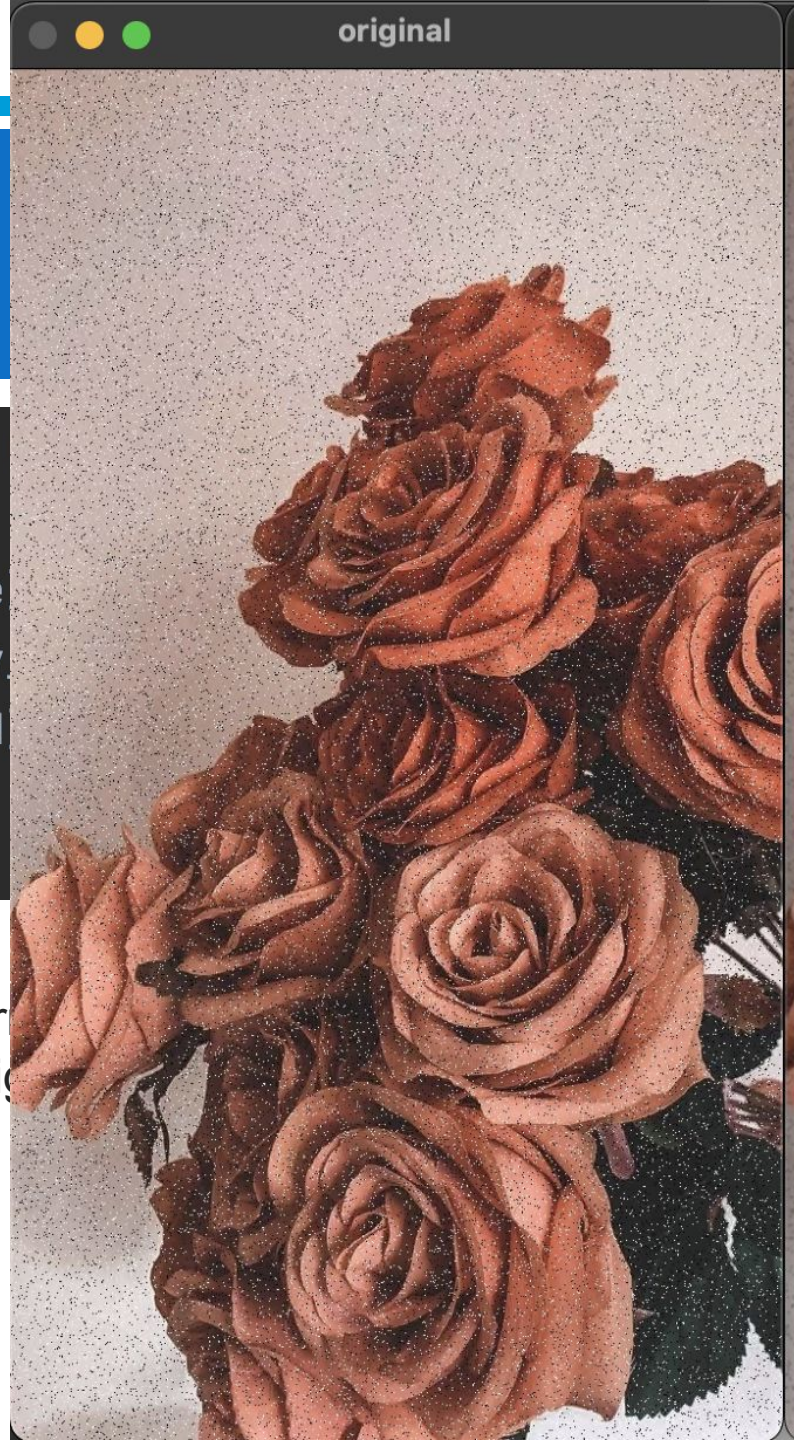

original
enhanced

GAUSSIAN FILTER

```
blurred = cv2.GaussianBlur(image,

cv2.imshow('original image', image
cv2.imshow('noisy_sp image', noisy
cv2.imshow('blured image', blurred
cv2.waitKey()
cv2.destroyAllWindows()
```

0: is the sigma of this Gaussian function deter
amount of smoothing. The higher number of si
it is the higher of smoothness too.



Smoothing Non-linear Filters

Order Statistics Filter:

- It is based on the ordering the pixels contained in the image area encompassed by the filter.
- It replaces the value of the center pixel with the value determined by the ranking result.
- Edges are better preserved in this filtering.

Smoothing Non-linear Filters

- **Types of Order statistics filter:**

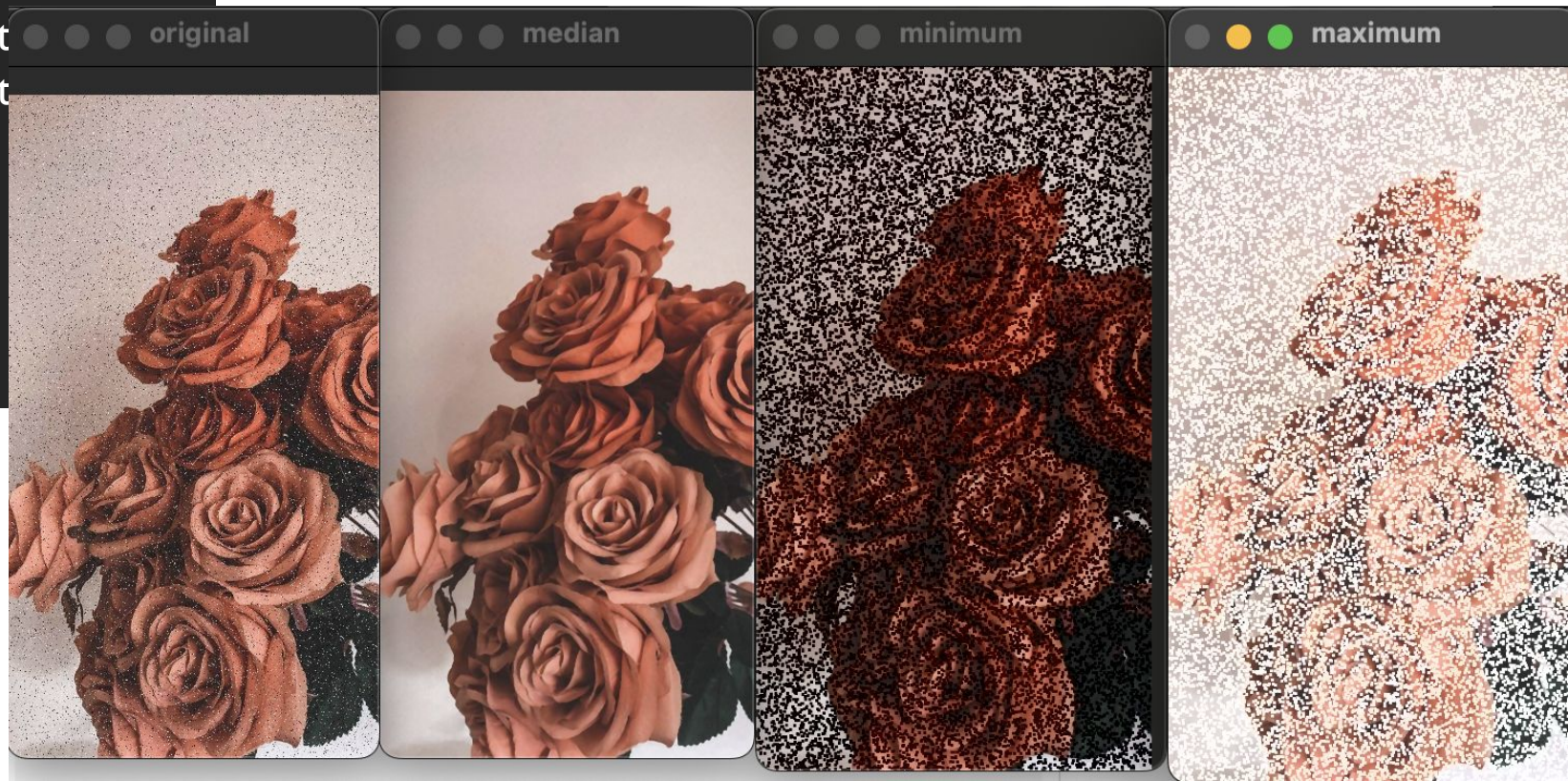
(i) Minimum filter: The value of the center is replaced by the smallest value in the window.

(ii) Maximum filter: The value of the center is replaced by the largest value in the window.

(iii) Median filter: Each pixel in the image is considered. First, neighboring pixels are sorted, and original values of the pixel is replaced by the median of the list.

Smoothing Non-linear Filters

```
import cv2
import numpy as np
img=cv2.imread("noisy_img.jpg")
median=cv2.medianBlur(img, 5)
mini=cv2.erode(img,np.ones((5,5),np.uint
maxi=cv2.dilate(img,np.ones((5,5),np.uint
cv2.imshow("original",img)
cv2.imshow("median", median)
cv2.imshow("minimum", mini)
cv2.imshow("maximum", maxi)
cv2.waitKey()
```



Thank you