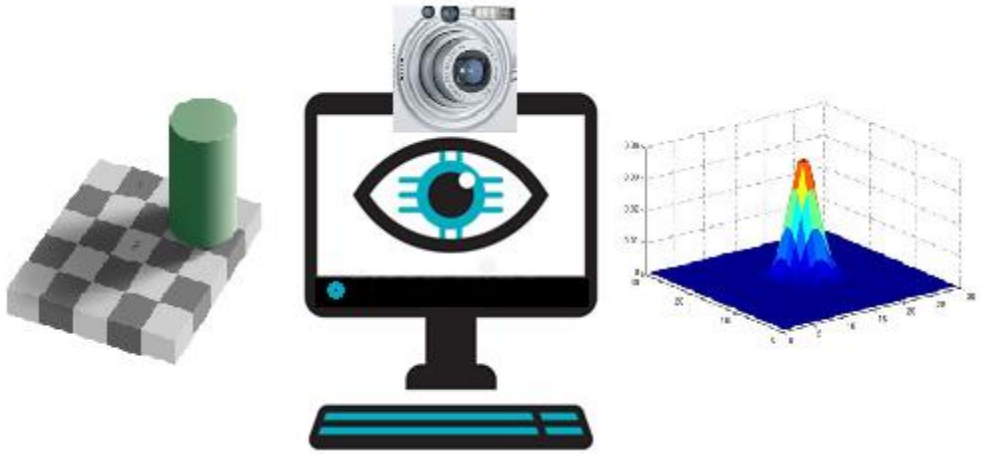


Computer Vision
Lecture 3
Spatial Filtering

Dr. Ali Mahmoud Mayya
Computer Science Dept.
AL Manara University, Syria
2024

Image Filtering

A spatial filter consists of (a) **a neighborhood**, and (b) **a predefined operation**

تتألف عملية الترشيح من جوار وعملية محددة

Linear spatial filtering of an image of size MxN with a filter of size mxn is given by the expression

الترشيح المكاني الخطي لصورة بأبعاد MxN بمرشح ذو أبعاد mxn يعطى بالعلاقة الآتية

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

X

Y

(x-1,y-1)	(x,y-1)	(x+1,y-1)
(x-1,y)	(x,y)	(x+1,y)
(x-1,y+1)	(x,y+1)	(x+1,y+1)

- يهدف الترشيح إلى عدة أمور منها:
- ترشيح الضجيج
- توضيح (تعزيز) الحواف
- كشف مناطق محددة من الصورة

Type of filters

- Linear Filters
  - Mean Filter
  - Gaussian Filter

تستبدل البكسل بمتوسط السويات اللونية أو المتوسط الموزون للبكسل وجواره

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

- Non linear Filters
  - Median
  - Max
  - Min

تستبدل البكسل بأحد بكسلات الجوار (الوسيط، القيمة الأعلى في الجوار، أو القيمة الأدنى في الجوار)

Image filtering-Linear-Average

$$g[\cdot,\cdot]\ \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

$$h[m,n]=\sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Slide credit: S. Seitz

Image filtering-Linear-Average

$$g[\cdot,\cdot]\ \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10							

$$h[m,n]=\sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Slide credit: S. Seitz

Image filtering-Linear-Average

$$g[\cdot,\cdot]\ \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20						

$$h[m,n]=\sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Slide credit: S. Seitz

Image filtering-Linear-Average

$$g[\cdot,\cdot]\ \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30					

$$h[m,n]=\sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Slide credit: S. Seitz

Image filtering-Linear-Average

$$g[\cdot,\cdot]\ \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30				

$$h[m,n]=\sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Slide credit: S. Seitz

Image filtering-Linear-Average

$$g[\cdot,\cdot]\ \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30				
				?					

$$h[m,n]=\sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Slide credit: S. Seitz

Image filtering-Linear-Average

$$g[\cdot,\cdot]\ \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30				
						?			
				50					

$$h[m,n]=\sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Slide credit: S. Seitz

Image filtering-Linear-Average

$$g[\cdot,\cdot]\ \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

$f[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

$$h[m,n]=\sum_{k,l} g[k,l]\ f[m+k,n+l]$$

Slide credit: S. Seitz

Image Filtering- Example

smoothing تنعيم

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

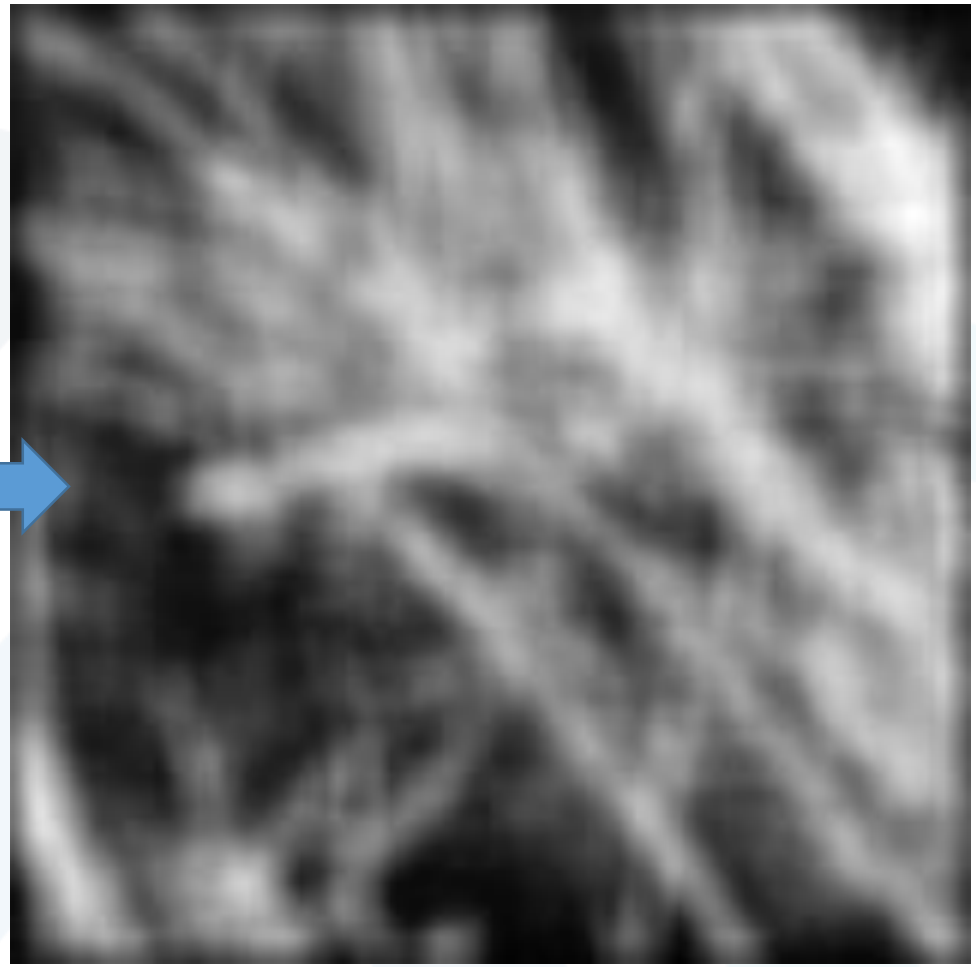


Image Filtering- Example

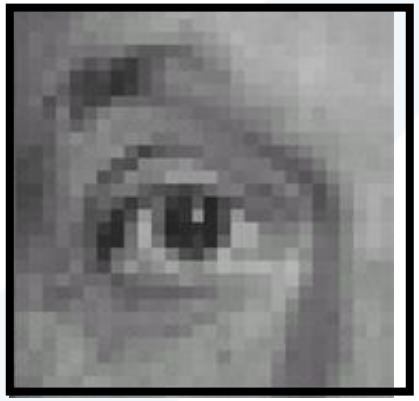
Original

0	0	0
0	1	0
0	0	0

Filtered
(no change)

Original

0	0	0
0	0	1
0	0	0



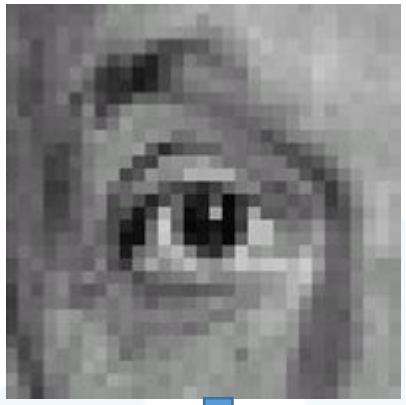
Shifted left
By 1 pixel

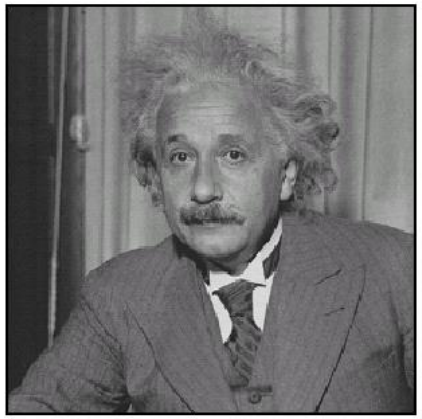
Image Filtering Application- Example

Original

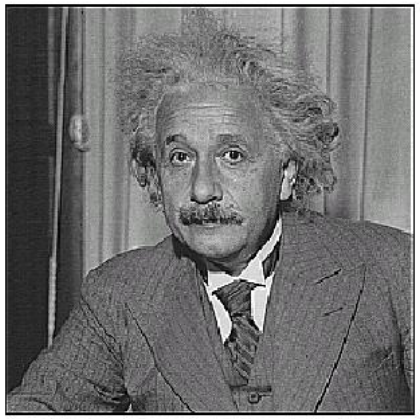
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

(Note that filter sums to 1)





before



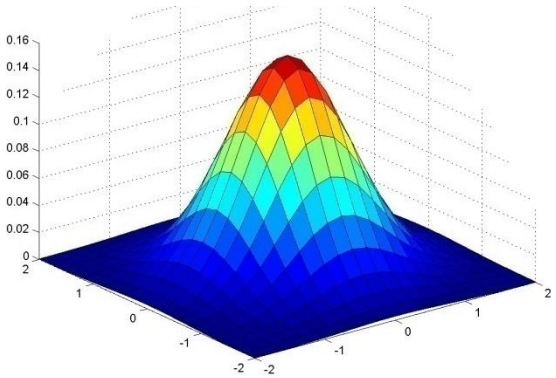
after

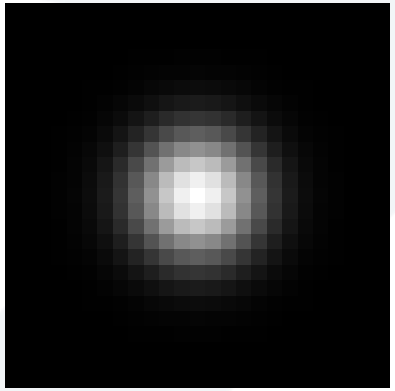
Sharpening filter

توزين القيم للصورة الأصلية – الصورة المنعمة = صورة معززة الحواف

Image Filtering-Linear- Gaussian

تدرج في الوزن من القيمة المركزية باتجاه الأطراف
أخف تأثيراً على الحواف من المتوسط Average
يخفف الضجيج





0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

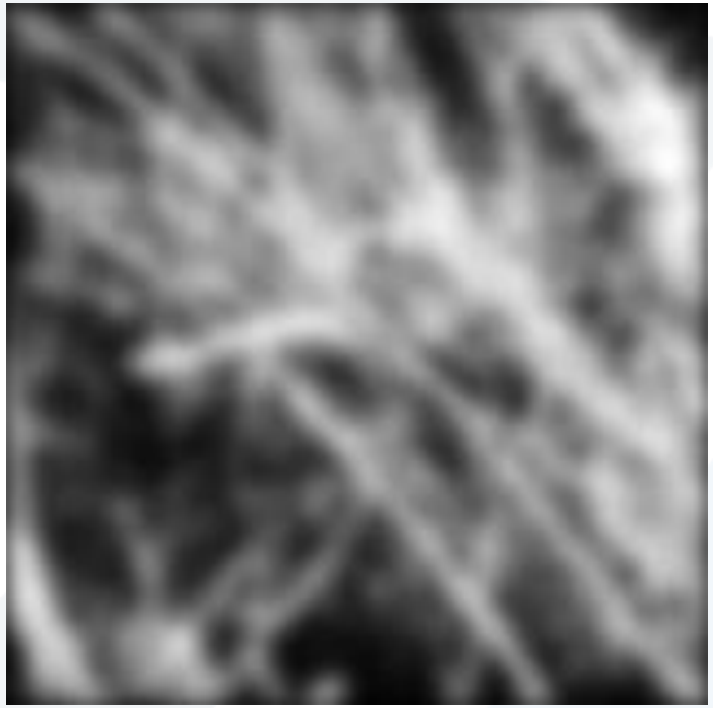
5 x 5, $\sigma = 1$

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Image Filtering-Linear- Gaussian

الصورة الأصلية

الغوصي



المتوسط

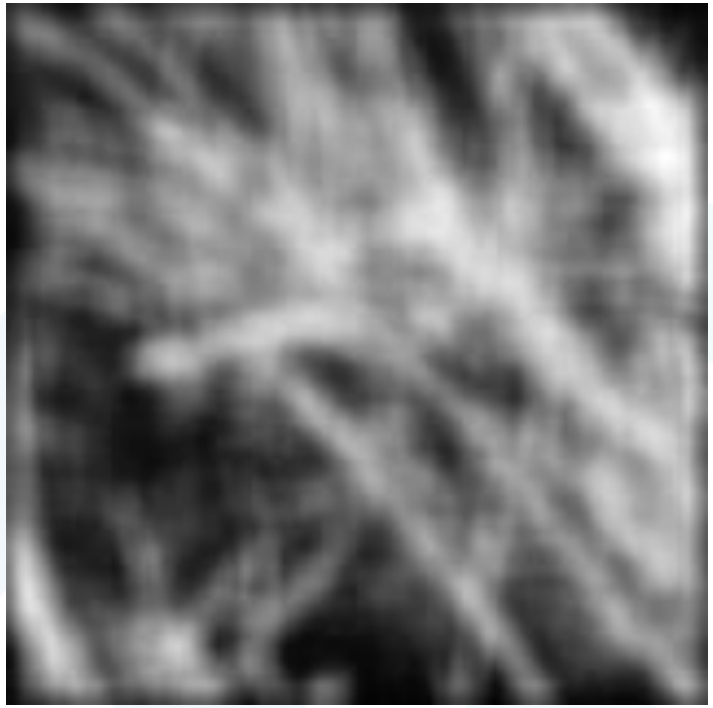


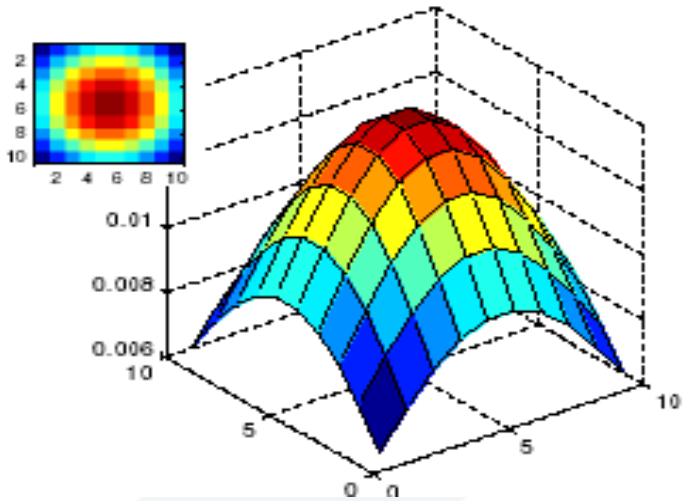
Image Filtering-Linear- Gaussian

What parameters matter here? ما هي البارامترات المهمة للمرشح

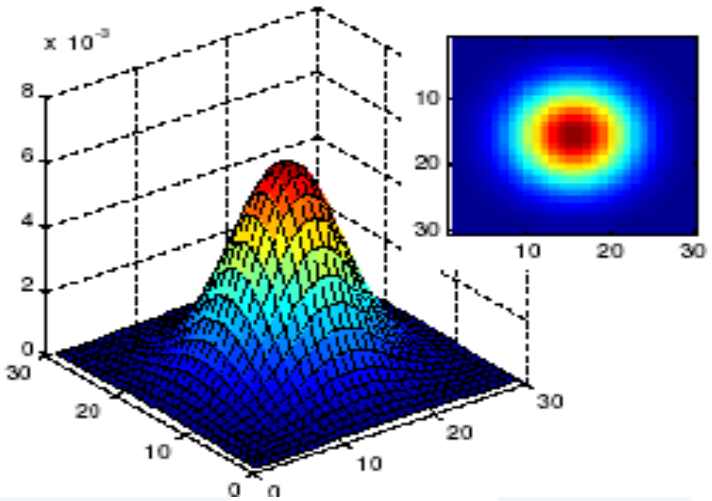
Size of kernel or mask حجم القناع أو نواة المرشح

Note, Gaussian function has infinite support, but discrete filters use finite kernels

يزداد التنعيم مع زيادة حجم المرشح



σ = 5 with 10 x 10 kernel



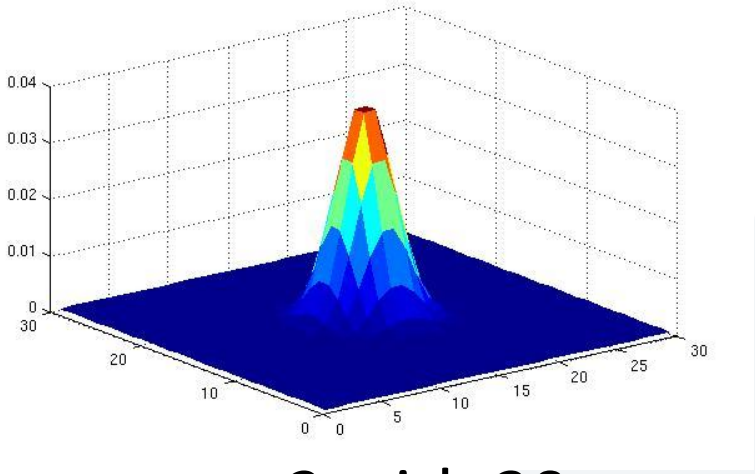
σ = 5 with 30 x 30 kernel

Image Filtering-Linear- Gaussian

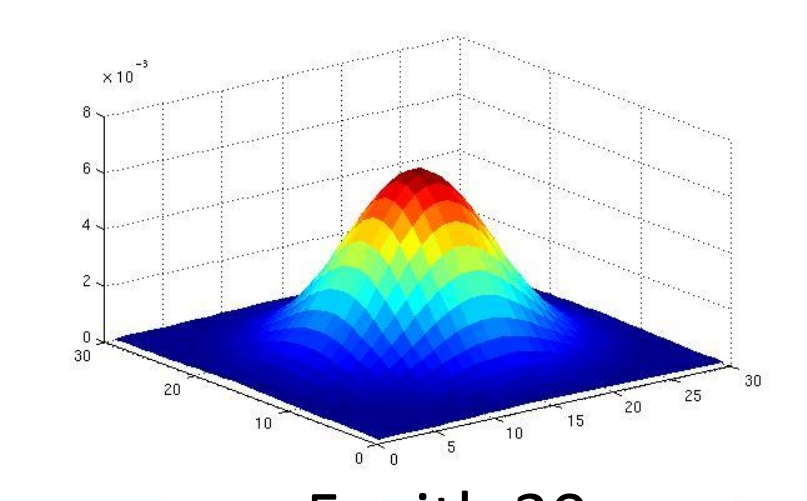
What parameters matter here? ما هي البارامترات المهمة للمرشح

Variance of Gaussian: determines extent of smoothing

الانحراف (امتداد عملية التنعيم) مع زيادة الانحراف يزيد التنعيم والعكس بالعكس



σ = 2 with 30 x 30 kernel



σ = 5 with 30 x 30 kernel

Image Filtering-Linear- Gaussian

Separability خاصية قابلية التقسيم

2D filtering
(center location only)

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 2 & 3 & 3 \\ 3 & 5 & 5 \\ 4 & 4 & 6 \end{bmatrix}$$

$$\begin{aligned} &= 2 + 6 + 3 = 11 \\ &= 6 + 20 + 10 = 36 \\ &= 4 + 8 + 6 = 18 \\ \hline &\ 65 \end{aligned}$$

The filter factors
into a product of 1D
filters:

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

Perform filtering
along rows:

$$\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 2 & 3 & 3 \\ 3 & 5 & 5 \\ 4 & 4 & 6 \end{bmatrix} = \begin{bmatrix} & 11 & \\ & 18 & \\ & 18 & \end{bmatrix}$$

Followed by filtering
along the remaining column:

$$\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \begin{bmatrix} & 11 & \\ & 18 & \\ & 18 & \end{bmatrix} = \begin{bmatrix} & & \\ & 65 & \\ & & \end{bmatrix}$$

Separability means that a 2D convolution can be reduced to two 1D convolutions

Complexity of filtering an $n \times n$ image with an $m \times m$ kernel?
$O(n^2 m^2)$
What if the kernel is separable?
$O(n^2 m)$

Noise الضجيج

Original

Salt and pepper noise

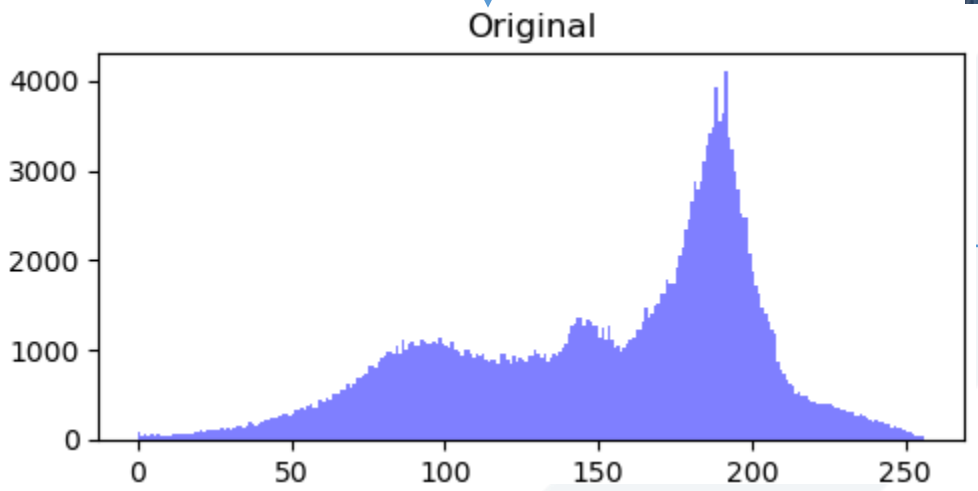
Impulse noise

Gaussian noise

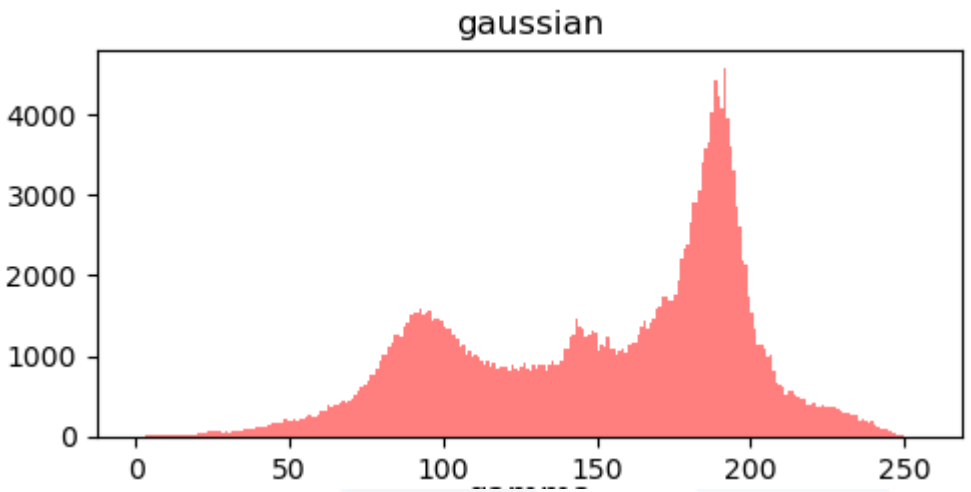
- **Salt and pepper noise**: contains random occurrences of black and white pixels
- **ضجيج S & P** يتضمن بكسلات بيضاء وسوداء موزعة ضمن الصورة
- **Impulse noise:** contains random occurrences of white pixels
- **الضجيج النبضي** يتضمن بكسلات بيضاء موزعة ضمن الصورة
- **Gaussian noise**: variations in intensity drawn from a Gaussian normal distribution
- **الضجيج الغوصي** يمثل تغيرات في السويات الرمادية مماثلة للتوزع الطبيعي لغوص
- **Periodic Noise:**
- **الضجيج الدوري** ويظهر على شكل خطوط طولية أو عرضية أو مائلة أو أشكال هندسية مكررة في الصورة.

- الأنواع الثلاثة الأولى يمكن ترشيحها في المجال الفراغي Spatial Domain.
- النوع الأخير (الضجيج الدوري) يجب ترشيحه حصراً في المجال الترددي Frequency Domain.

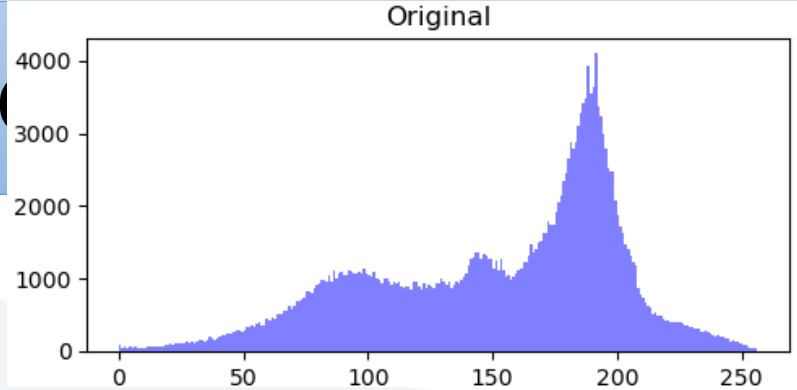
Noise type prediction using Histogram

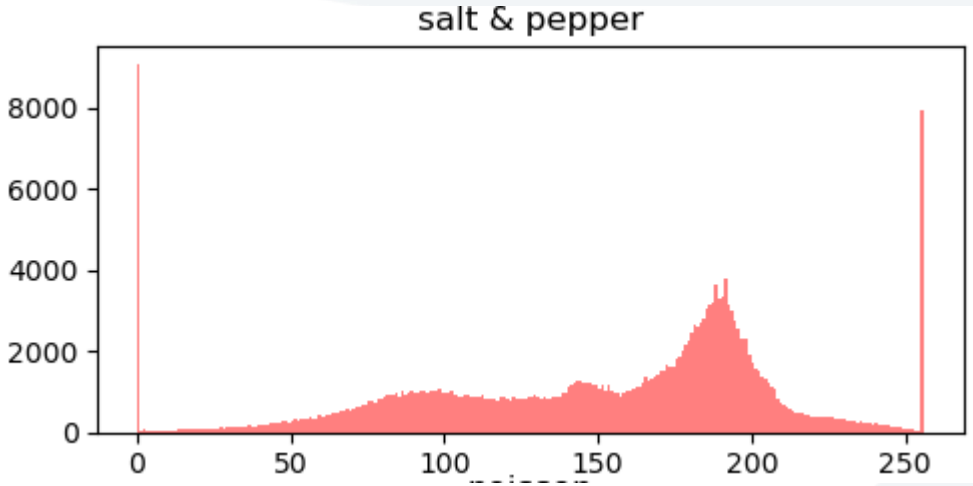


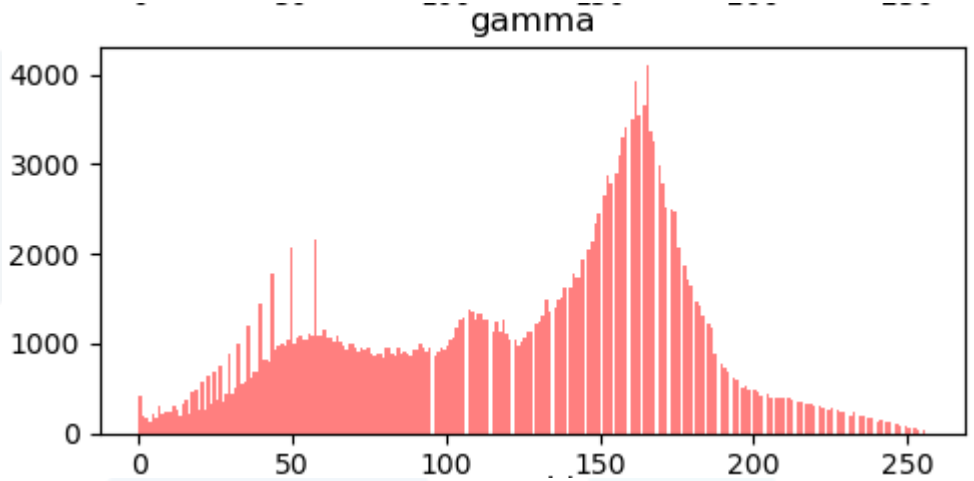
Adding Gaussian noise

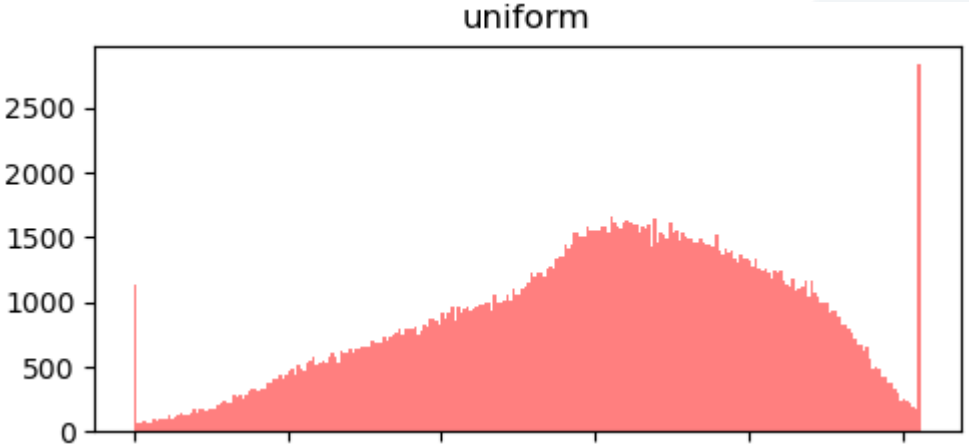


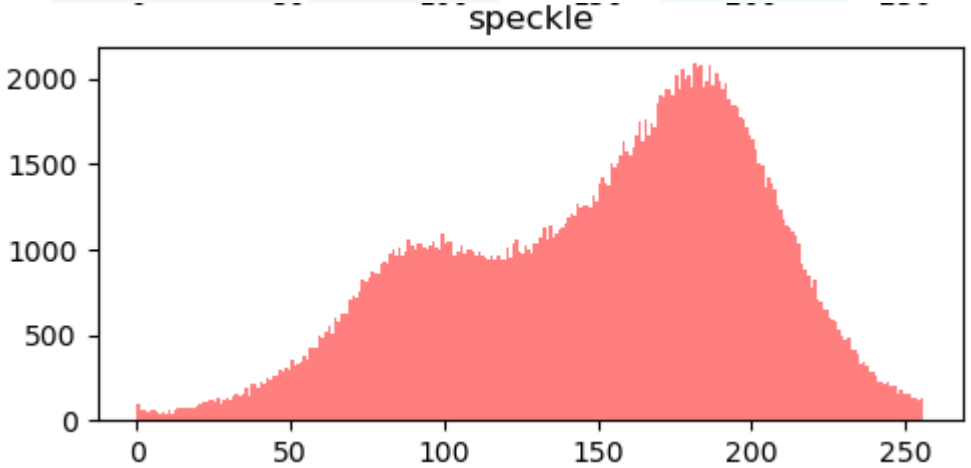
Noise typ g Histogram











Noise type prediction using Histogram - code

```
import cv2    import numpy as np   import matplotlib.pyplot as plt
img = cv2.imread('image.jpg', cv2.IMREAD_GRAYSCALE)
noise_types = ['gaussian', 'salt & pepper', 'gamma', 'poisson',
'speckle']    noisy_imgs = []
for noise_type in noise_types:
    if noise_type == 'gaussian':
        noisy_img = cv2.GaussianBlur(img, (5, 5), 0)
    elif noise_type == 'salt & pepper':
        noisy_img = img.copy()
        noise = np.zeros_like(img)
        cv2.randu(noise, 0, 255)
        noisy_img[noise < 10] = 0
        noisy_img[noise > 245] = 255
    elif noise_type == 'gamma':
        noisy_img = np.power(img /np.max(img), 1.5) * 255
        noisy_img = noisy_img.astype(np.uint8)
    elif noise_type == 'uniform':
        noisy_img = img.copy()
        noise = np.random.uniform(-50, 50, size=img.shape)
        noisy_img = noisy_img + noise
        noisy_img[noisy_img < 0] = 0
        noisy_img[noisy_img > 255] = 255
    elif noise_type == 'speckle':
        noisy_img = img + img * np.random.randn(*img.shape) * 0.1
    noisy_imgs.append(noisy_img)
```

```
#display original image
plt.figure(figsize=(((6,6
plt.imshow(img, cmap='gray')
plt.title('Original Image')
plt.axis('off')
plt.show()
#plot histograms
fig, axs = plt.subplots(((10 ,10)=ezisgfi ,2 ,3
axs[ ,0.5=ahpla ,[256 ,0] ,256 ,()levar.gmi)tsih.[0 ,0
('eulb'=roloc
axs[('lanigirO')eltti_tes.[0 ,0
for i, noise_type in enumerate(noise_types):
    axs[(i+ ,0] ,256 ,()levar.[i]sgmi_ysion)tsih.[%2(1+i) ,2//(1
('der'=roloc ,0.5=ahpla ,[256
    axs[(i+(epyt_esion)eltti_tes.[%2(1+i) ,2//(1
plt.tight_layout()
plt.show()
```

إزالة الضجيج S&P باستخدام المرشحات الخطية

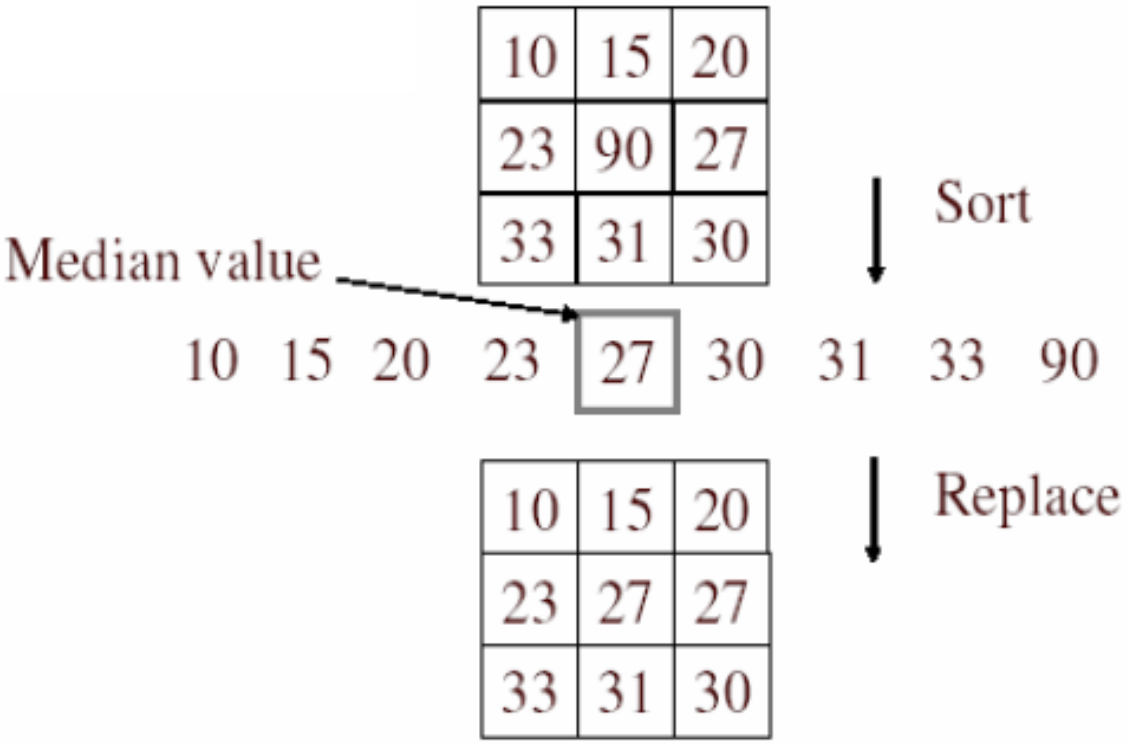
3x3 Average Filter

5x5 Average Filter

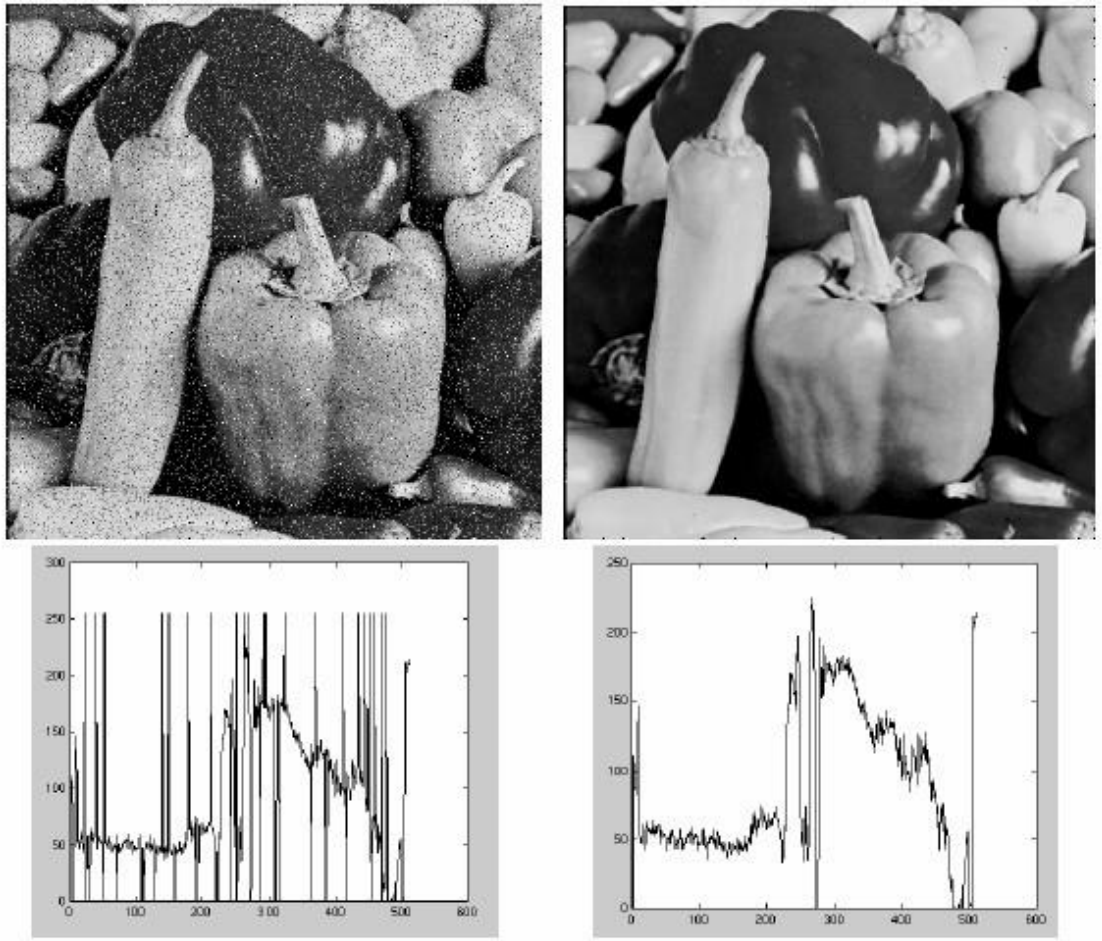
7x7 Average Filter

- What’s wrong with the results?
- المشكلة أنَّ مرشح المتوسط والمرشح الغوصي لا يستطيع ترشيح ضجيج S&P لأنه يستبدل البكسل بمتوسط الشدة اللونية للبكسل ومجاوراته!
- نحتاج لمرشح غير خطي !

إزالة الضجيج S&P باستخدام المرشحات الخطية



Salt-and-pepper noise Median filtered

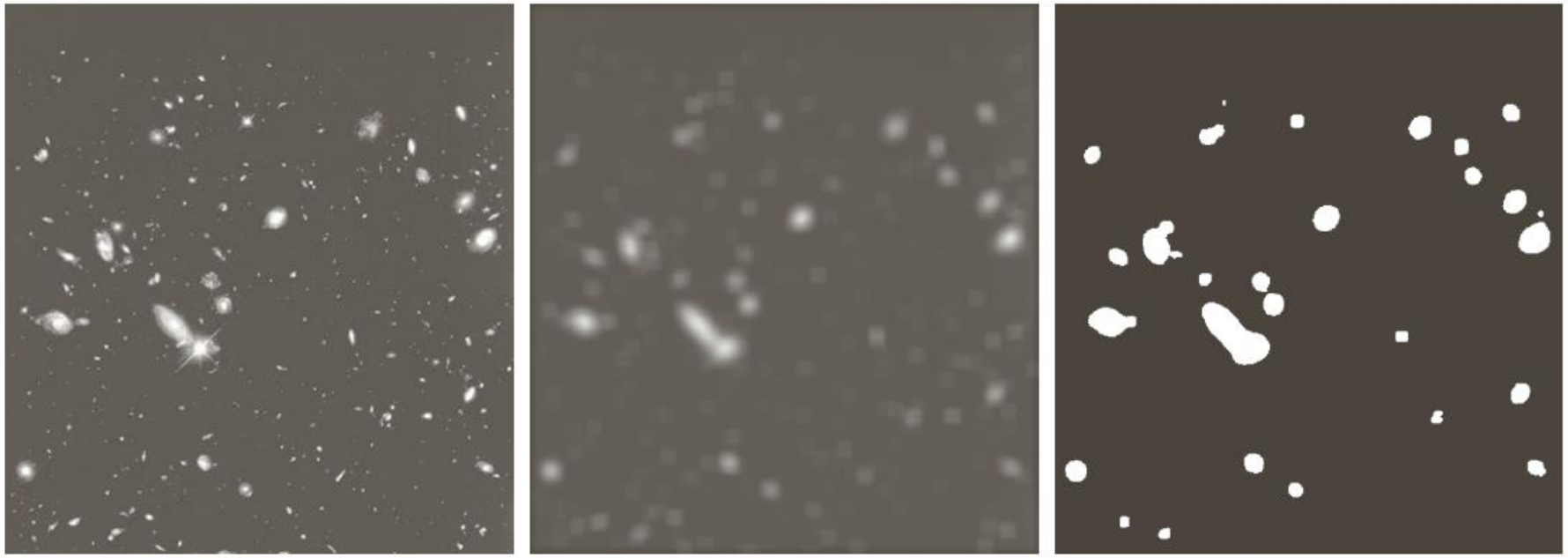


- **نتيجة:**
- يمكن إزالة **ضجيج S&P** باستخدام مرشح الوسيط **Median** اللاخطي

إزالة الضجيج S&P باستخدام المرشحات الخطية

	3x3	5x5	7x7
Gaussian			
Median			

كشف المناطق باستخدام المرشحات الخطية Object Detection



a b c

FIGURE 3.34 (a) Image of size 528 × 485 pixels from the Hubble Space Telescope. (b) Image filtered with a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

مرشحات الحد Sharpening Spatial Filters

- Foundation
- Laplacian Operator
- Unsharp Masking and Highboost Filtering
- Using First-Order Derivatives for Nonlinear Image Sharpening — The Gradient

Sharpening Spatial Filters: Foundation

- The first-order derivative of a one-dimensional function f(x) is the difference

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

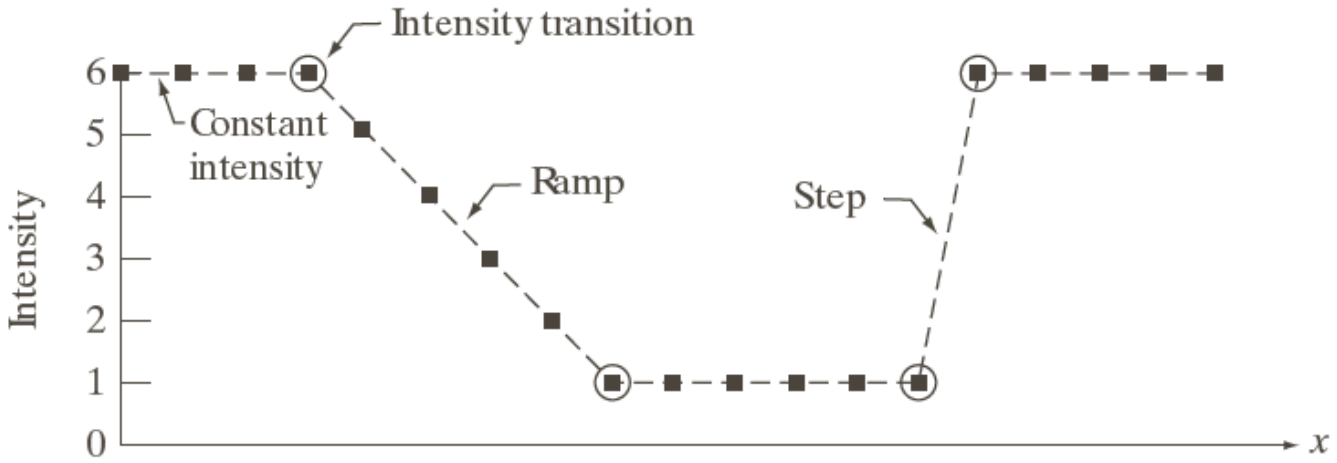
مشتق درجة أولى
(القيمة التالية –
القيمة الحالية)

- The second-order derivative of f(x) as the difference

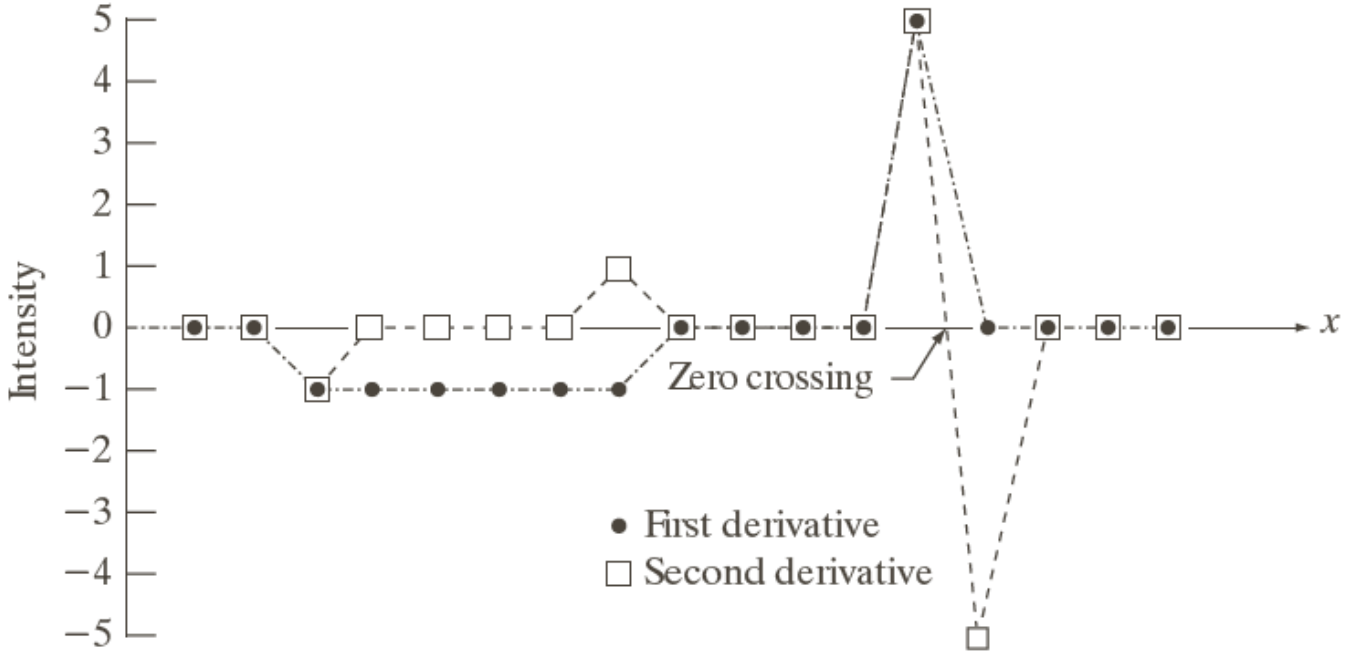
$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

مشتق درجة ثانية
(القيمة التالية +
القيمة السابقة -
ضعفي القيمة
الحالية)

al Filters



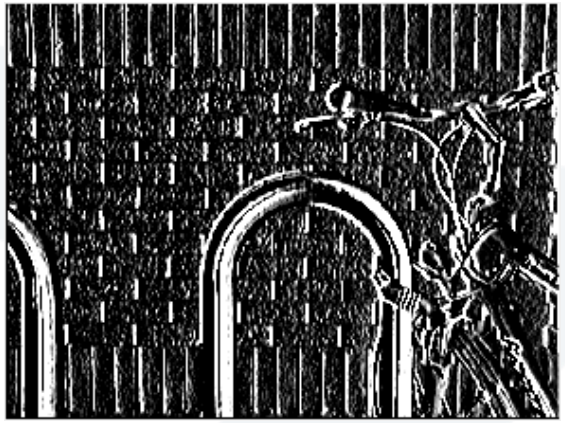
Scan line	6	6	6	6	5	4	3	2	1	1	1	1	1	1	6	6	6	6	6
1st derivative		0	0	−1	−1	−1	−1	−1	0	0	0	0	0	5	0	0	0	0	
2nd derivative		0	0	−1	0	0	0	0	1	0	0	0	0	5	−5	0	0	0	



Sharpening Spatial Filters: First Order- Sobel

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

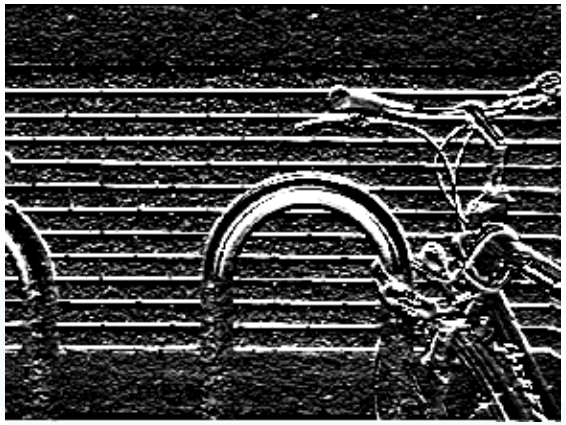
Sobel
مرشح سوبل أفقي Hx
لكشف حواف عمودية



Vertical Edge $\mathbf{g_x}$
(absolute value)

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

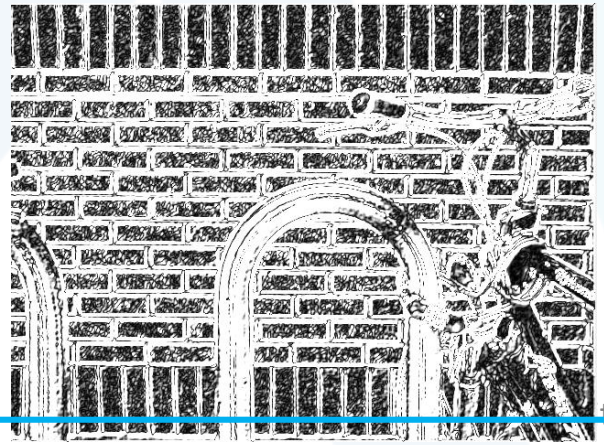
Sobel
مرشح سوبل عمودي Hy
لكشف حواف أفقية



Horizontal Edge $\mathbf{g_y}$
(absolute value)

The *magnitude* of vector ∇f, denoted as $M(x, y)$

$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^{\,2} + g_y^{\,2}}$$



Gradient Image
طويلة التدرج
الحواف الأفقية
والعمودية معاً

t: David Lowe (UBC)

Sharpening Spatial Filters: Second Order- Laplacian Equations & Matrix

The second-order isotropic derivative operator is the Laplacian for a function (image) f(x,y)

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

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

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

$$\nabla^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

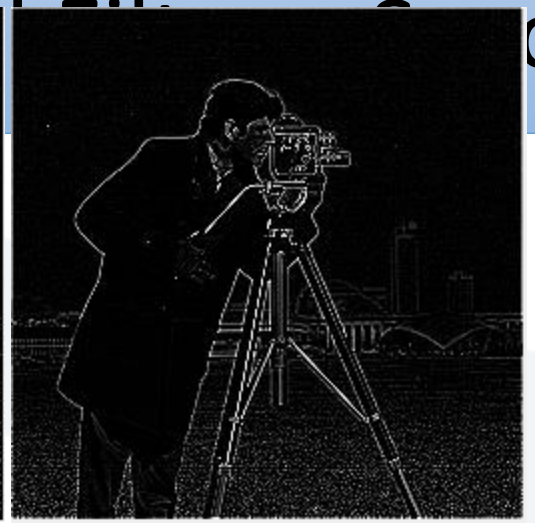
0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

Sharpe[illegible]ond Order- **Edge Detection**

- كشف الحواف- مثال: تأثير اختلاف المرشح

0	1	0
1	-4	1
0	1	0



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

1	1	1
1	-8	1
1	1	1

-1	-1	-1
-1	8	-1
-1	-1	-1

Sharpening Spatial Filters: Second Order- Laplacian Sharpening

يتم إنجاز عملية الحد Sharpening على الصورة باستخدام مرشح اللابلاسيان كالـآتي:

$$g(x,y) = f(x,y) + c\left[\nabla^2 f(x,y)\right]$$

where,

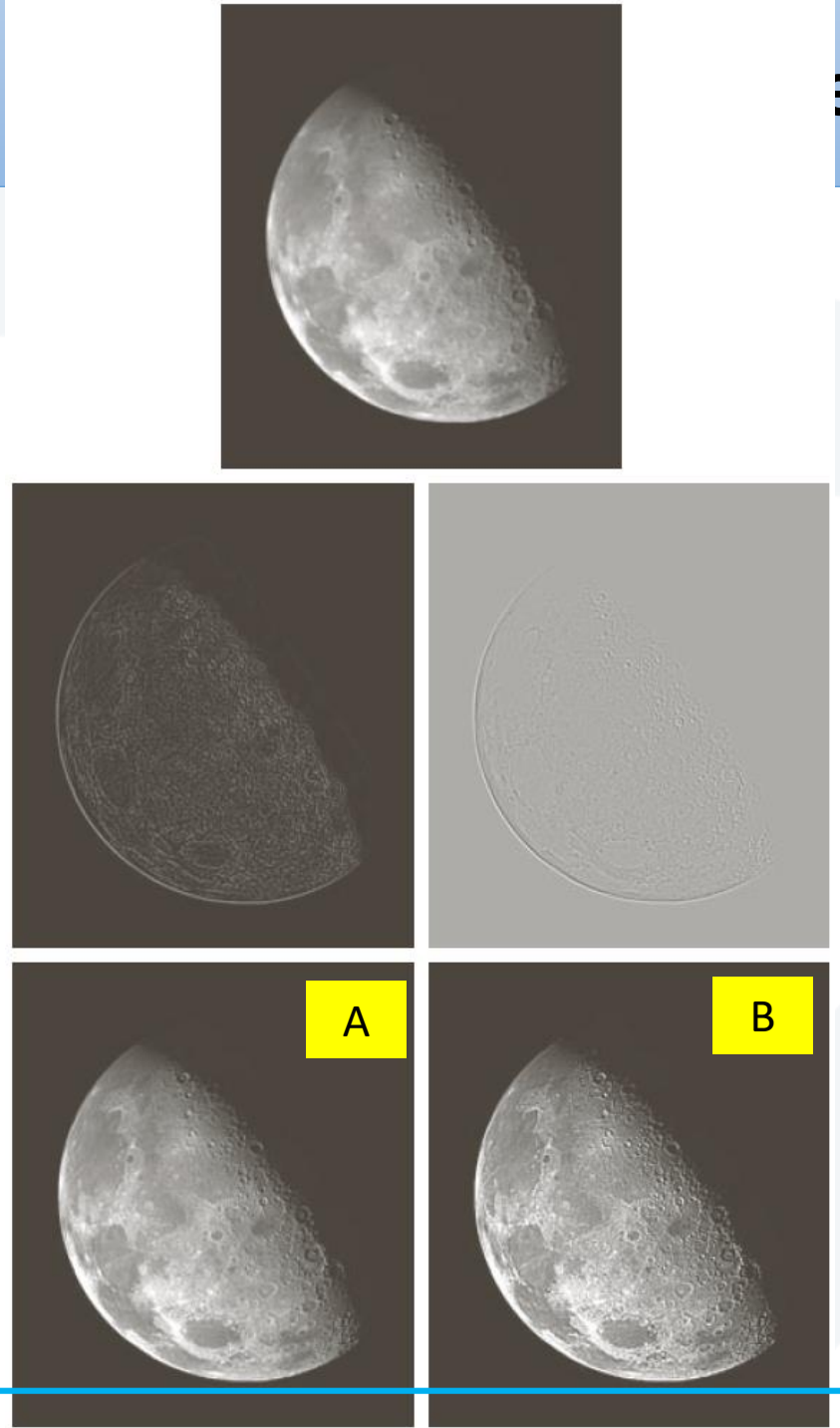
$f(x,y)$ is input image,

$g(x,y)$ is sharpenend images,

$c = -1$ if $\nabla^2 f(x,y)$ corresponding to Fig. 3.37(a) or (b)

and $c = 1$ if either of the other two filters is used.

ers: Laplacian Sharpening الحد باستخدام اللابلاسيان



a
b c
d e

FIGURE 3.38 (a) Blurred image of the North Pole of the moon. (b) Laplacian without scaling. (c) Laplacian with scaling. (d) Image sharpened using the mask in Fig. 3.37(a). (e) Result of using the mask in Fig. 3.37(b). (Original image courtesy of NASA.)

A

0	1	0
1	−4	1
0	1	0

B

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

C

−1	−1	−1
−1	8	−1
−1	−1	−1

D

Sharpening Spatial Filters: Second Order- Unsharp

- Unsharp masking

  Sharpen images consists of subtracting an unsharp (smoothed) version of an image from the original image

  طرح نسخة منعمة للصورة من الصورة الأصلية فيتم بذلك تعزيز الحواف

  e.g., printing and publishing industry لتطبيقات الطباعة

- Steps

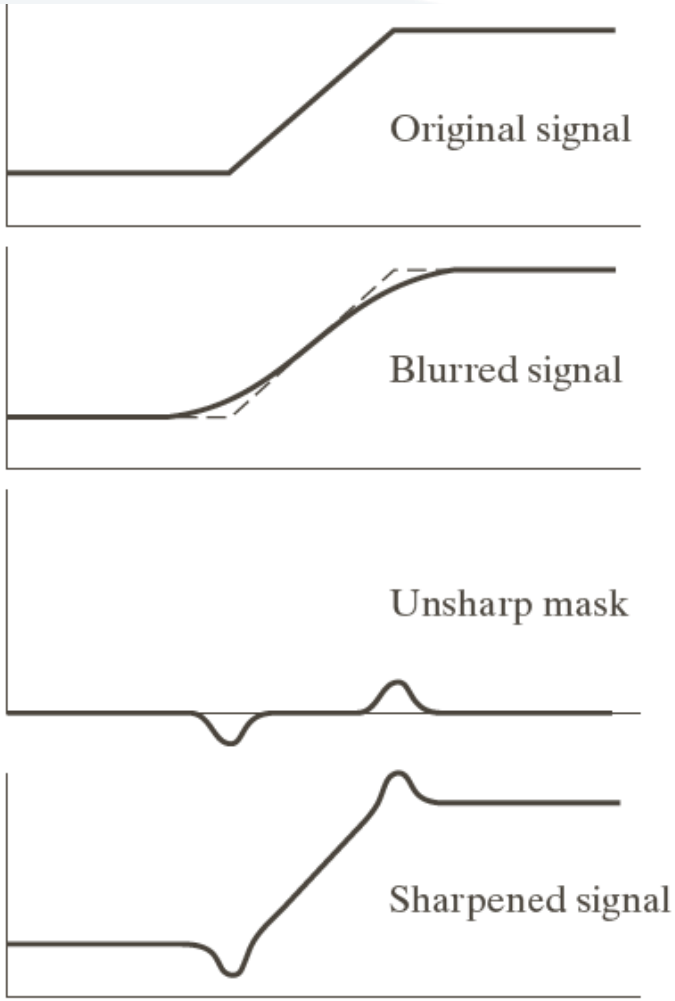
1. Blur the original image تنعيم الصورة
2. Subtract the blurred image from the original طرح الصورة المنعمة من الأصلية

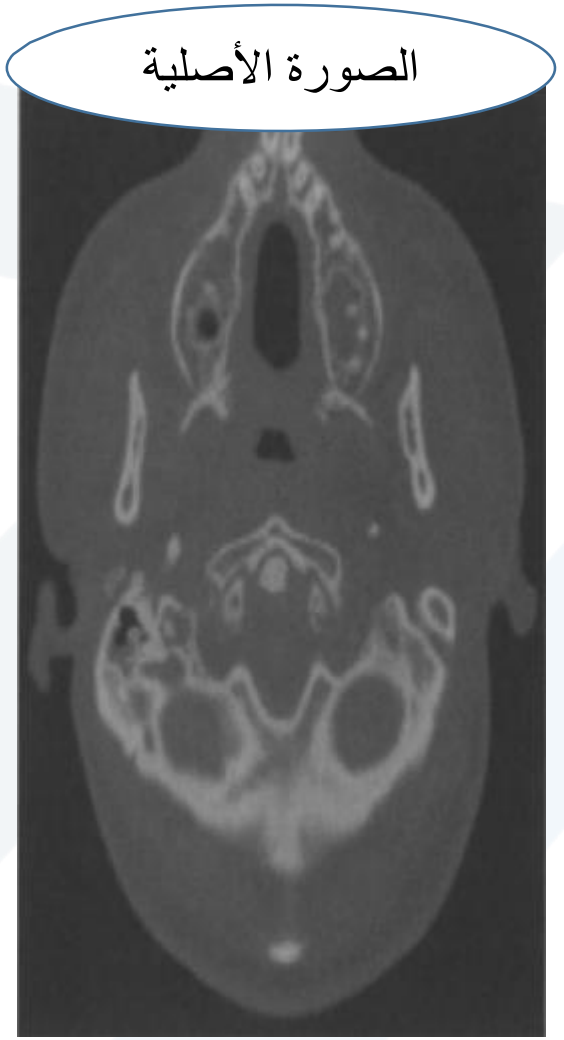
$$g_{mask}(x,y) = f(x,y) - \overline{f}(x,y)$$

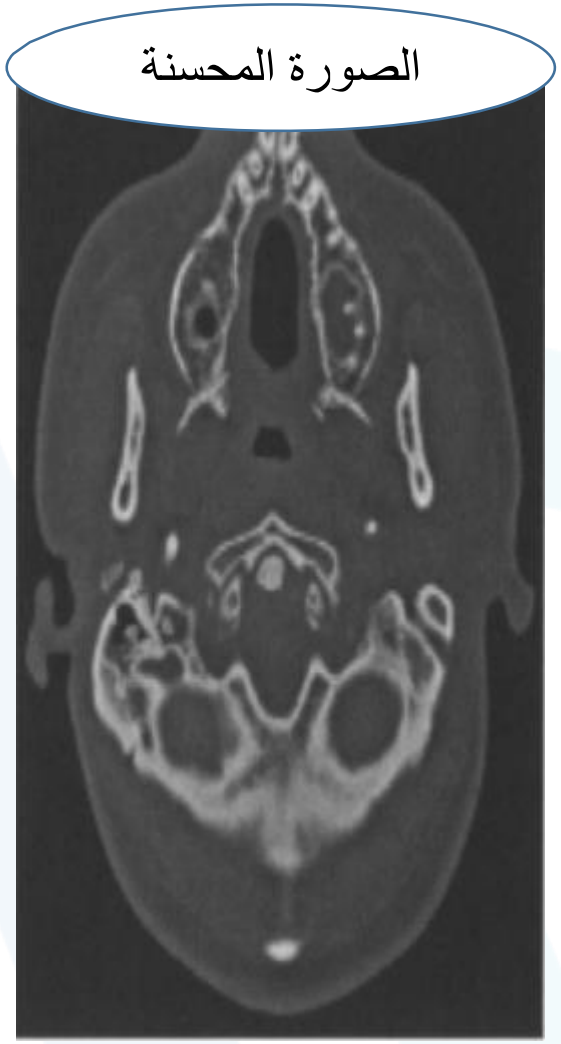
3. Add the mask to the original إضافة ناتج الخطوة 2 للصورة الأصلية

$$g(x,y) = f(x,y) + k * g_{mask}(x,y) \qquad k \geq 0$$

Sharpening Spatial Filters: Second Order- **Unsharp**







https://www.sciencedirect.com/topics/computer-science/unsharp-masking

https://manara.edu.sy/

Spatial Filters- General Concepts

Practice:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

مرشح التنعيم مع الحفاظ على الحواف
Bilateral Filtering

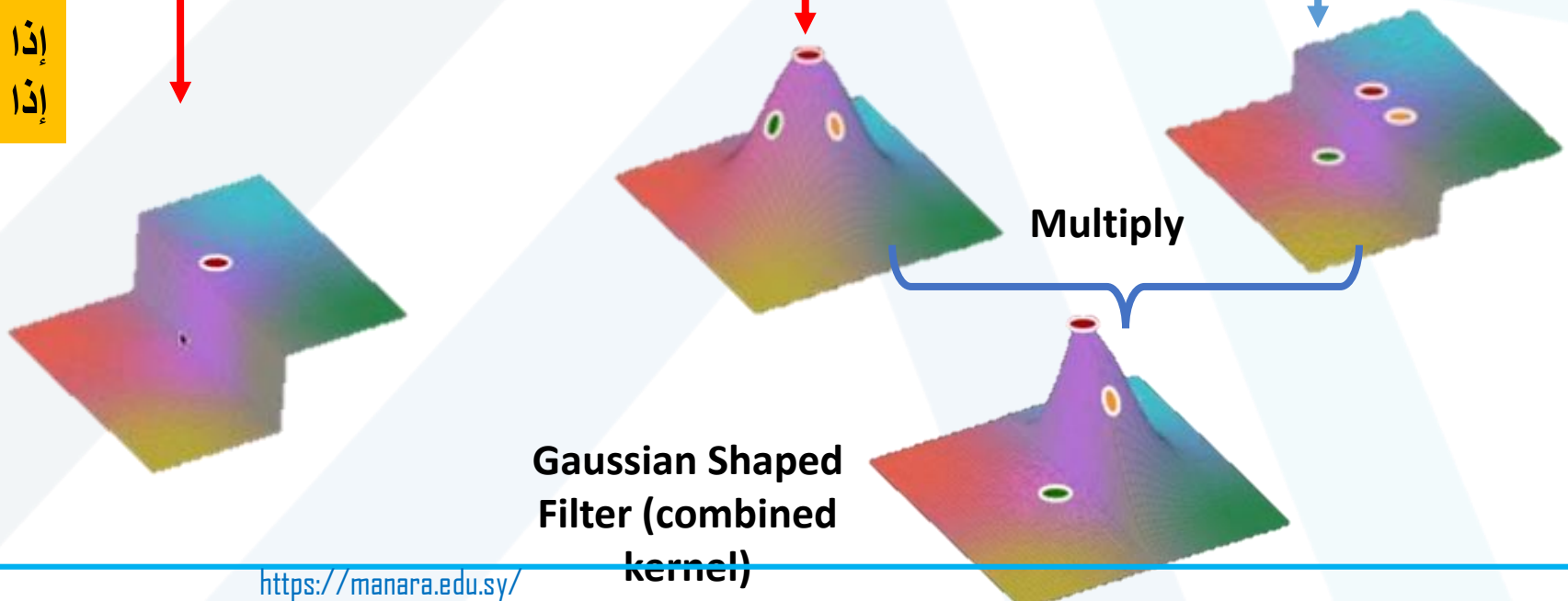
يقوم هذا المرشح بتنعيم الصورة مع **الحفاظ** على الحواف:
إذا كانت السويات الرمادية للبكسل ومجاوراته متقاربة يتم ترشيح البكسل.
إذا كانت السويات الرمادية للبكسل ومجاوراته مختلفة (حافة) لا يتم ترشيح البكسل.

بفرض الصورة تتضمن ضجيج وحافة

لكل بكسل:
نحسب الفرق بين البكسل ومجاوراته
إذا كان الفرق كبير نسند للبكسل وزن صغير
إذا كان الفرق صغير نسند للبكسل وزن كبير

Spatial Gaussian Brightness Gaussian

$$g[i,j] = \frac{1}{W_{sb}} \sum_m \sum_n f[m,n]\, n_{\sigma_s}[i-m, j-n]\, n_{\sigma_b}(f[m,n] - f[i,j])$$



مرشح التنعيم مع الحفاظ على الحواف
Bilateral Filtering

Spatial Gaussian Brightness Gaussian

$$g[i,j] = \frac{1}{W_{sb}} \sum_m \sum_n f[m,n]\, \underbrace{n_{\sigma_s}[i-m, j-n]}_{\text{Spatial Gaussian}}\, \underbrace{n_{\sigma_b}(f[m,n] - f[i,j])}_{\text{Brightness Gaussian}}$$

Spatial Gaussian

$$n_{\sigma_s}[m,n] = \frac{1}{2\pi {\sigma_s}^2} e^{-\frac{1}{2}\left(\frac{m^2+n^2}{\sigma_s^2}\right)}$$

Brightness Gaussian

$$n_{\sigma_b}(k) = \frac{1}{\sqrt{2\pi}\sigma_b} e^{-\frac{1}{2}\left(\frac{k^2}{\sigma_b^2}\right)}$$

Normalization Factor

$$W_{sb} = \sum_m \sum_n n_{\sigma_s}[i-m, j-n]\, n_{\sigma_b}(f[m,n] - f[i,j])$$

مرشح التنعيم مع الحفاظ على الحواف
Bilateral Filtering

blur = cv.bilateralFilter(img,diameter,σ_b, σ_s)