

Lecture 6: LỌC ẢNH

(Image filtering)

Problems

- Convolution and Image filtering
 - Nhân chập
 - Một số bộ lọc tuyến tính
- Smoothing – làm mịn
 - Noise removal
 - Detail preserving image smoothing
- Sharpen filter

Approaches

- Sử dụng toán tử hoặc bộ lọc trên miền không gian (giá trị pixel hiện tại phụ thuộc vào chính nó và các pixel xung quanh)
 - Linear filtering
 - Non-linear filtering

Lọc ảnh (image filtering)

- Lọc ảnh: Với mỗi điểm ảnh, tính giá trị mới của điểm ảnh dựa trên 1 hàm theo các điểm trong lân cận của nó
 - Lọc trong **miền không gian**:
 - Với mỗi điểm ảnh, tính giá trị mới của điểm ảnh dựa trên **1 hàm theo các điểm trong lân cận** của nó
 - Lọc ảnh trong **miền tần số** (Project)
 - Ảnh đầu vào và ra thường có **cùng kích thước**
- Nhân chập: phép lọc tuyến tính, hàm số là tổng có trọng số của các điểm ảnh trong lân cận của điểm ảnh xét.

$$I' = I * K$$

- Có vai trò quan trọng!
 - Tăng cường ảnh: giảm nhiễu, làm trơn, tăng độ tương phản, ...
 - Trích chọn thông tin từ ảnh: Texture, edges, distinctive points, etc.
 - Phát hiện mẫu (template matching)

Lọc ảnh (image filtering)

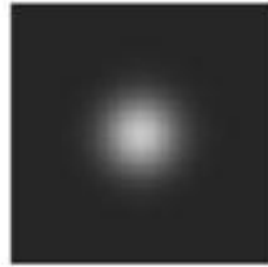
- Lọc trong miền không gian
 - Biến đổi cục bộ được thực hiện trên ma trận điểm ảnh
$$g(x,y) = T(f(x,y))$$
 - T: hàm tính toán dựa trên giá trị điểm (x,y) và các điểm hàng xóm (K)
(K: Filter/Mask/Kernel/Window/Template Processing)
 - T giống nhau tại mọi vị trí trên ảnh

Lọc ảnh



Original image

*



Mask (kernel)

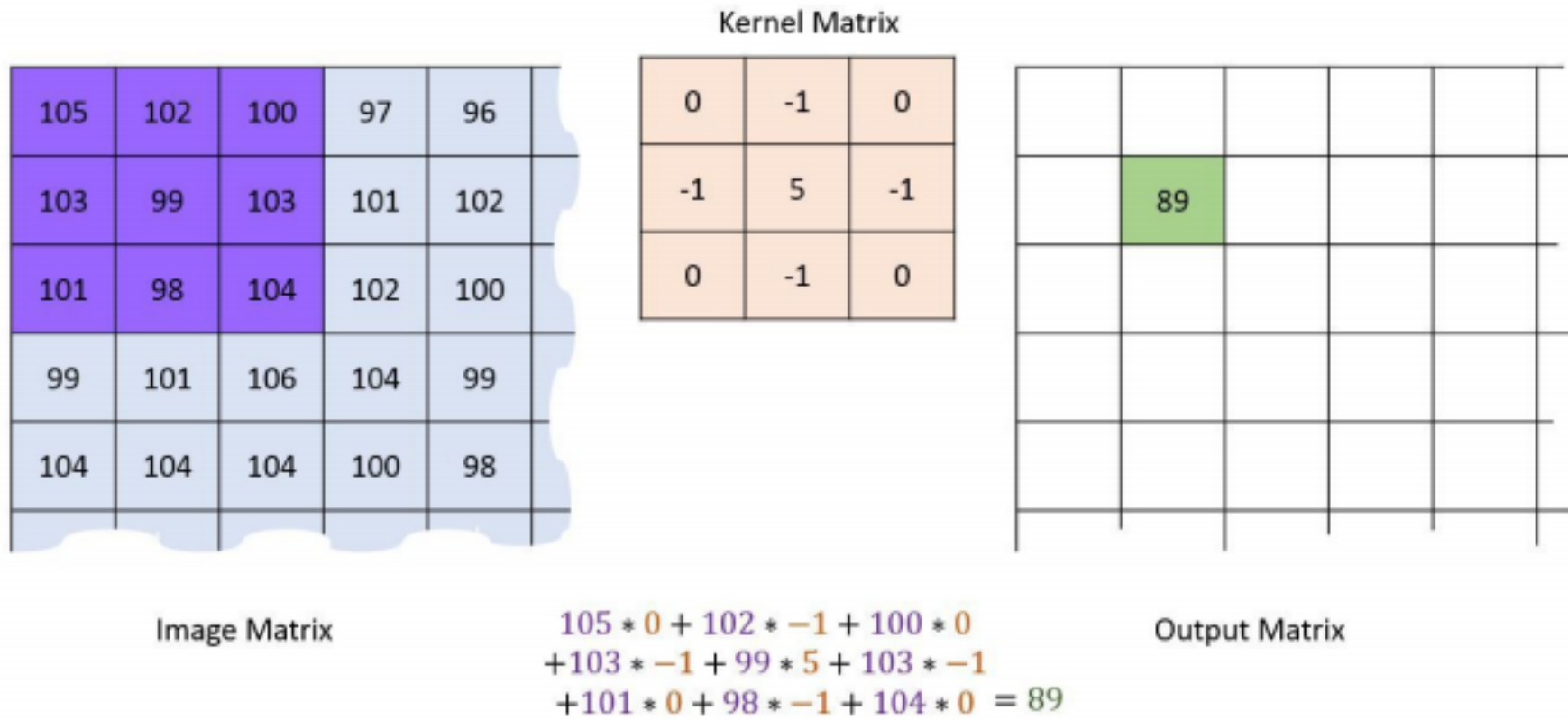
=



Filtered image

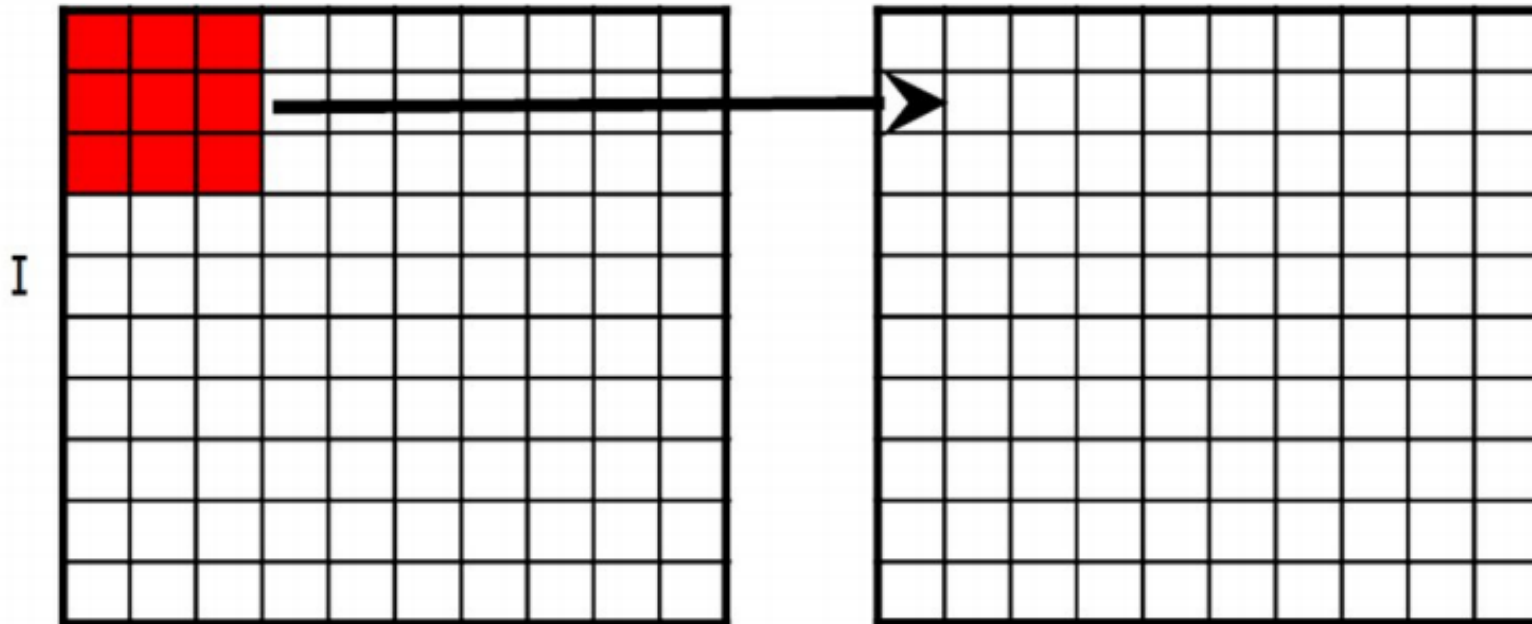
Nhân chập

- New value of a pixel(i,j) is a weighted sum of its neighbors

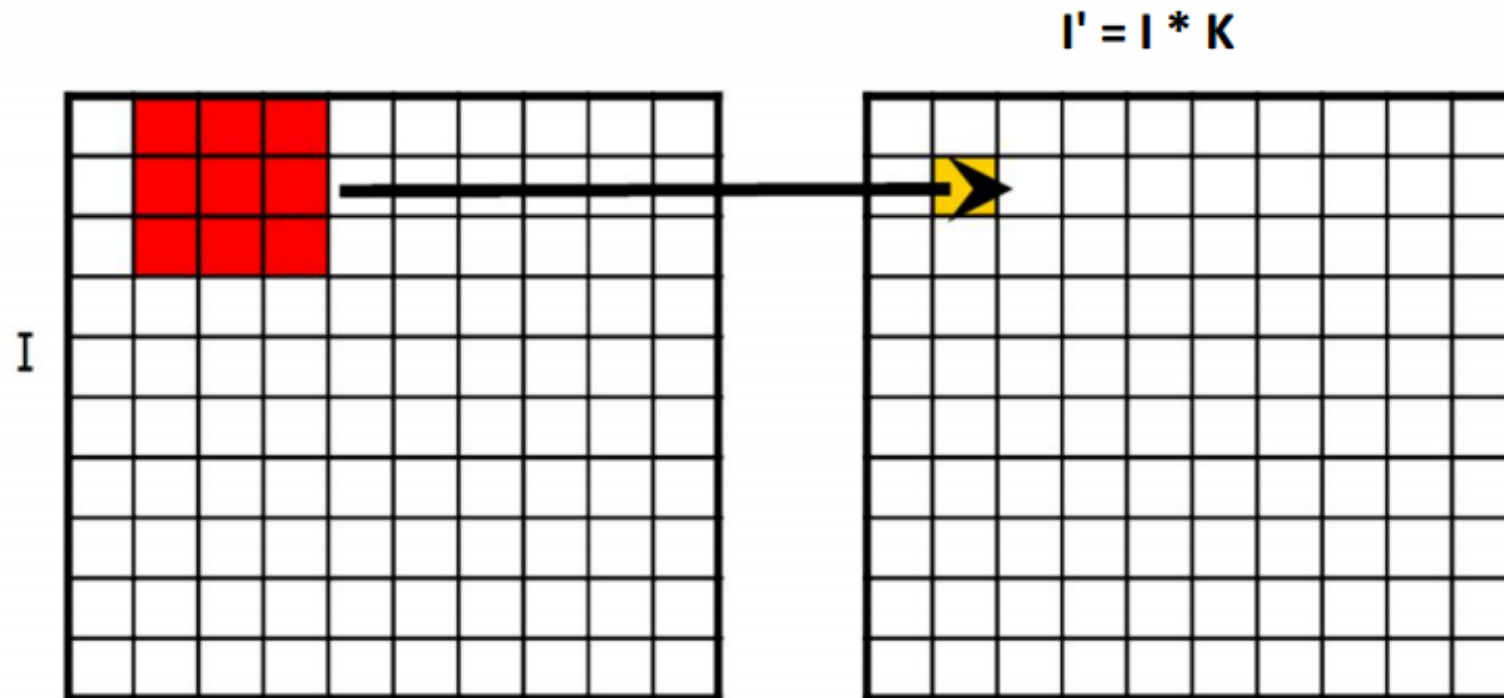


Nhân chập

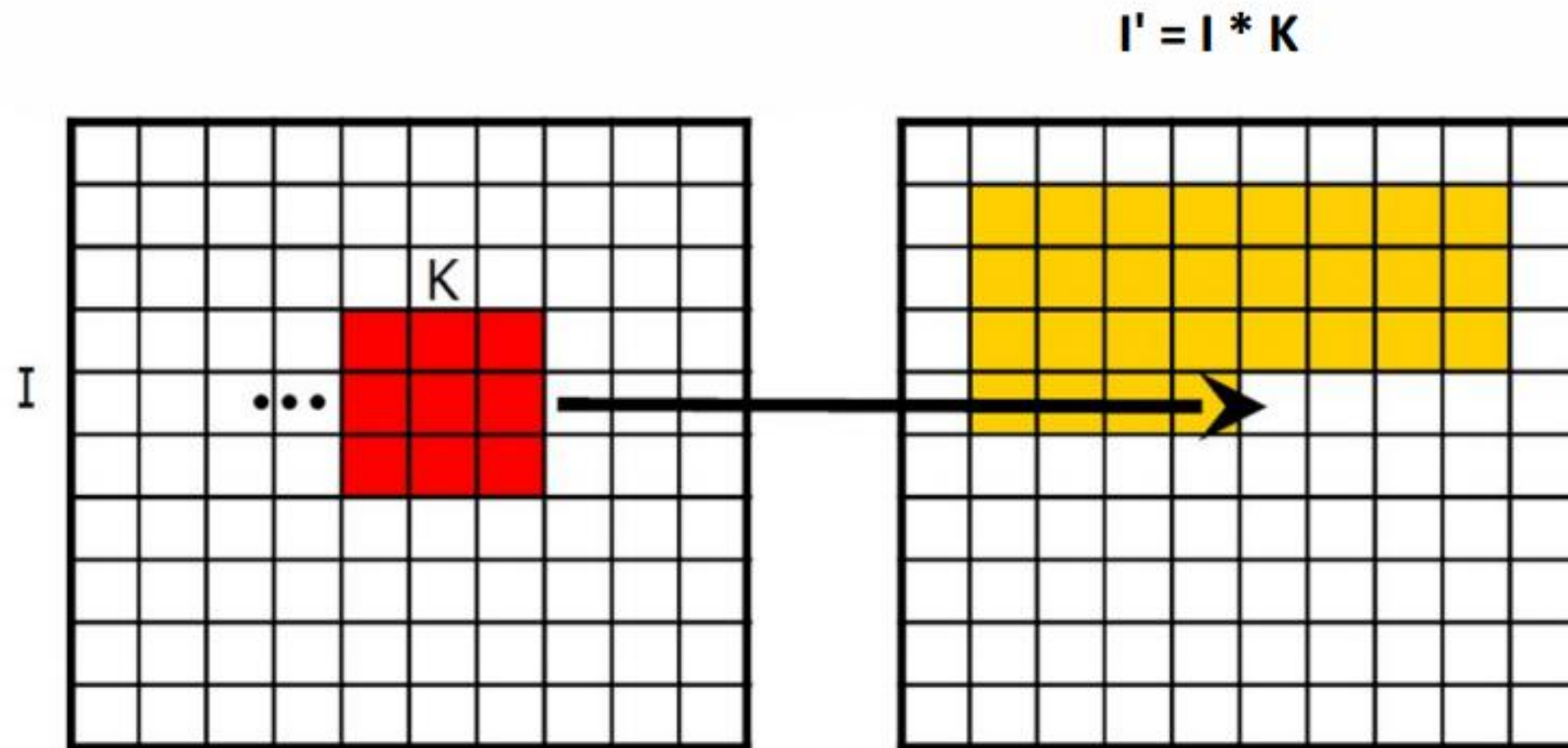
$$I' = I * K$$



Nhân chập



Nhân chập



Nhân chập

- Vấn đề ở cạnh ảnh?
 - Thêm dòng/cột 0 vào ma trận đầu vào
 - Đối xứng gương:
 - $f(-x,y) = f(x,y)$
 - $f(-x,-y) = f(x,y)$

?	?	?	?	?	?	?	?	?	?
?									?
?									?
?									?
?									?
?									?
?									?
?									?
?									?
?	?	?	?	?	?	?	?	?	?

0	0	0	0	0	0
0	105	102	100	97	96
0	103	99	103	101	102
0	101	98	104	102	100
0	99	101	106	104	99
0	104	104	104	100	98

105	105	102	100	97	96
105	105	102	100	97	96
103	103	99	103	101	102
101	101	98	104	102	100
99	99	101	106	104	99
104	104	104	104	100	98



Nhân chập

0	0	0	0	0	0
0	105	102	100	97	96
0	103	99	103	101	102
0	101	98	104	102	100
0	99	101	106	104	99
0	104	104	104	100	98

Image Matrix

Kernel Matrix		
0	-1	0
-1	5	-1
0	-1	0

320				
210	89	111		

Output Matrix

$$\begin{aligned}
 &0 \times 0 + 0 \times -1 + 0 \times 0 \\
 &+ 0 \times -1 + 105 \times 5 + 102 \times -1 \\
 &+ 0 \times 0 + 103 \times -1 + 99 \times 0 = 320
 \end{aligned}$$

105	105	102	100	97	96
105	105	102	100	97	96
103	103	99	103	101	102
101	101	98	104	102	100
99	99	101	106	104	99
104	104	104	104	100	98

Kernel Matrix		
0	-1	0
-1	5	-1
0	-1	0

110				
	89	111		

Source: <http://machinelearningguru.com>

Một số bộ lọc (Some kernels)



Original image

*

0	0	0
0	1	0
0	0	0



Filtered image
(no change)



Original image

*

0	0	0
1	0	0
0	0	0



Filtered image
(shifted left by 1 pixel)

Một số bộ lọc (Some kernels)

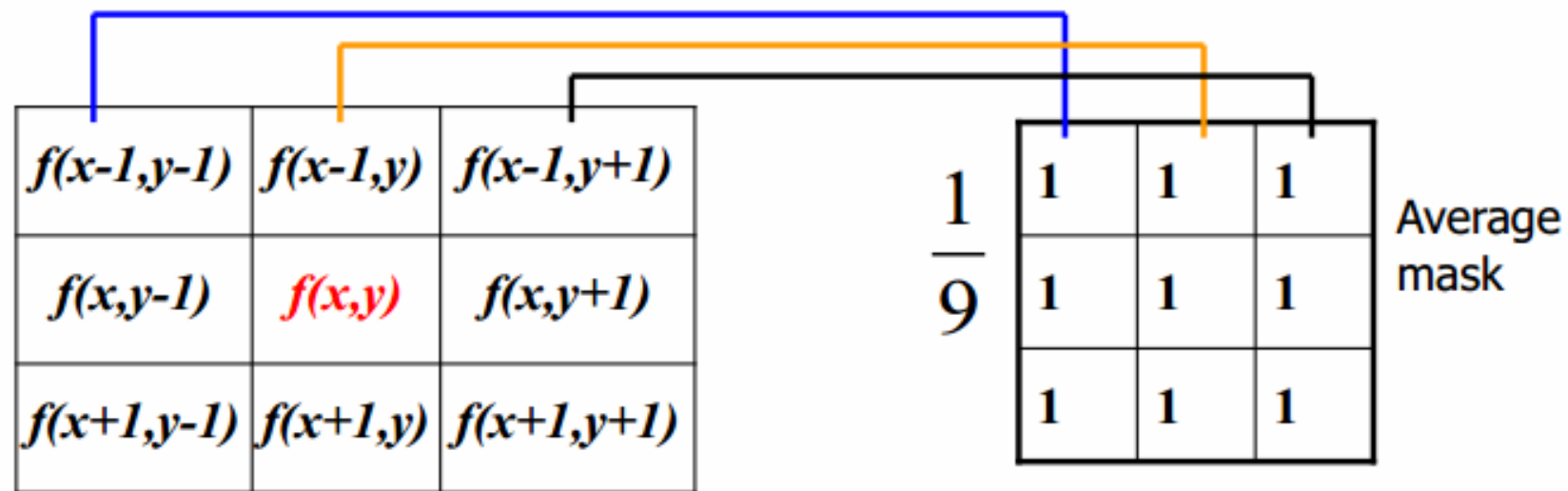
- Nhân chập 2D
 - Chủ yếu được sử dụng để trích chọn đặc trưng trên ảnh
 - Được sử dụng như phép toán trong khối cơ sở của mạng Neuron tích chập: Convolutional Neural Networks (CNNs)
- Mỗi bộ lọc có hiệu ứng riêng và hữu ích cho các nhiệm vụ cụ thể như:
 - Làm mờ (lọc nhiễu),
 - Làm nét biên,
 - Phát hiện cạnh,
 -

Bộ lọc làm trơn ảnh

- Mục đích
 - Lọc nhiễu
 - Làm trơn ảnh
 - Còn gọi là bộ lọc thông thấp
- Một số bộ lọc thông thấp
 - Bộ lọc trung bình
 - Bộ lọc Gauss
- Để tránh thay đổi độ sáng của ảnh, **tổng các hệ số trong mặt nạ phải = 1**

Bộ lọc làm trơn ảnh

- Lọc trung bình (**mean filter**):



$$g(x, y) = \frac{1}{9} [f(x-1, y-1) + f(x-1, y) + f(x-1, y+1) + \\ f(x, y-1) + f(x, y) + f(x, y+1) + \\ f(x+1, y-1) + f(x+1, y) + f(x+1, y+1)]$$

Bộ lọc làm trơn ảnh

- Lọc trung bình (**mean filter**):
 - Thay giá trị bởi giá trị trung bình của các hàng xóm
 - Ảnh được làm trơn

$$1/9 \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



Original image



Filtered image
with box size 5x5



Filtered image
with box size 11x11

Mean filter/Box filter

Diagram illustrating the Mean filter/Box filter operation:

The input image F is a 10x10 grid. A 3x3 neighborhood is highlighted with a red box, centered at row 6, column 4 (0-indexed). The values in this neighborhood are:

0	0	0
0	0	90
0	90	0

The kernel H is a 3x3 grid of ones:

1	1	1
1	1	1
1	1	1

The output image G is a 10x10 grid. The value at the center of the neighborhood (row 6, column 4) is calculated as the mean of the neighborhood values:

$$\frac{0 + 0 + 0 + 0 + 0 + 90 + 90 + 90 + 0}{9} = 20$$

The resulting value 20 is shown in the output image G at the corresponding position (row 6, column 4), also highlighted with a red box.

Bộ lọc làm trơn ảnh

- Lọc trung bình có trọng số
 - The pixel corresponding to the center of the mask is more important than the other ones.

$\frac{1}{16} \times$	1	2	1
	2	4	2
	1	2	1

$$g(x, y) = \frac{1}{16} [f(x-1, y-1) + 2f(x-1, y) + f(x-1, y+1) \\ + 2f(x, y-1) + 4f(x, y) + 2f(x, y+1) \\ + f(x+1, y-1) + 2f(x+1, y) + f(x+1, y+1)]$$

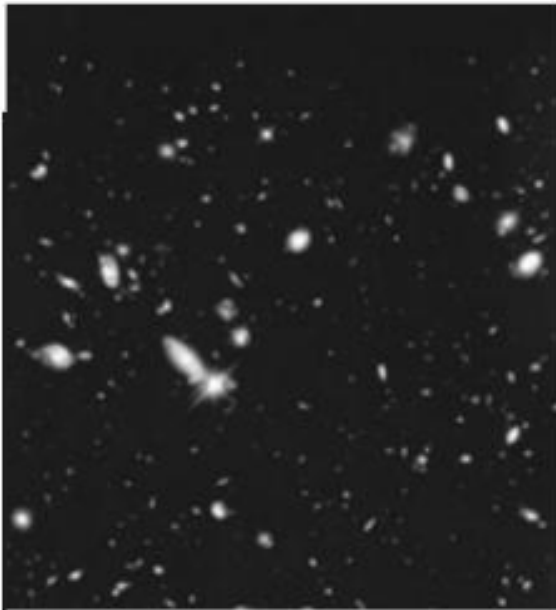
Common Smoothing Filters

$$\frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}; \quad \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}; \quad \frac{1}{(b+2)^2} \begin{bmatrix} 1 & b & 1 \\ b & b^2 & b \\ 1 & b & 1 \end{bmatrix}, \text{ with } b \geq 1.$$

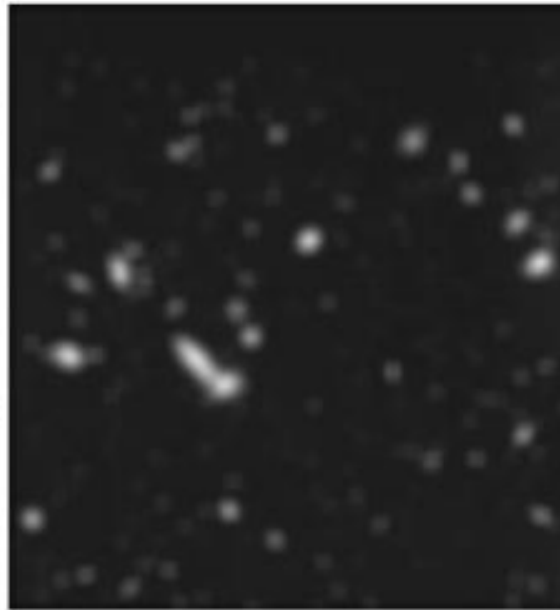
Bộ lọc làm trơn ảnh

- VD: sử dụng bộ lọc thông thấp để loại bỏ các vùng nhỏ

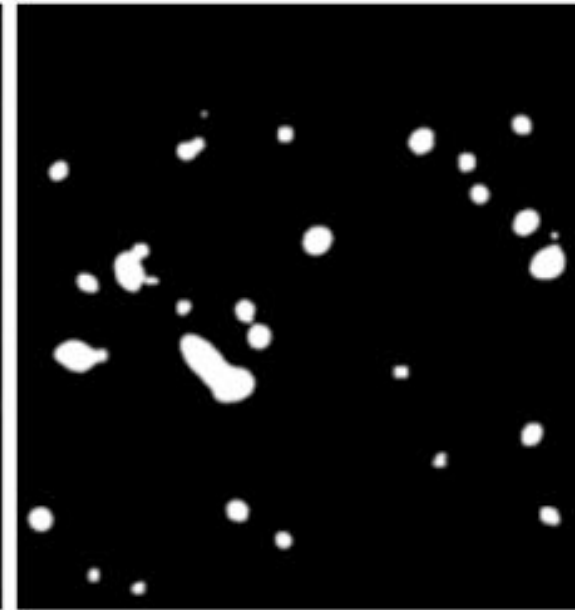
Original image



Average filtering: 15x15



Thresholding of blurring image



Median filters - Trung vị

110	120	90	130
91	94	98	200
90	95	99	100
82	96	85	90

becomes

95

Steps:

1. Sort the pixels in ascending order:

90, 90, 91, 94, 95, 98, 99, 110, 120

2. replace the original pixel value by the median: 95

Median filters – Trung vị

- Bộ lọc trung vị hoạt động trên một cửa sổ bằng cách chọn cường độ trung vị trong cửa sổ.



Image credit: Wikipedia – page
on Median Filter

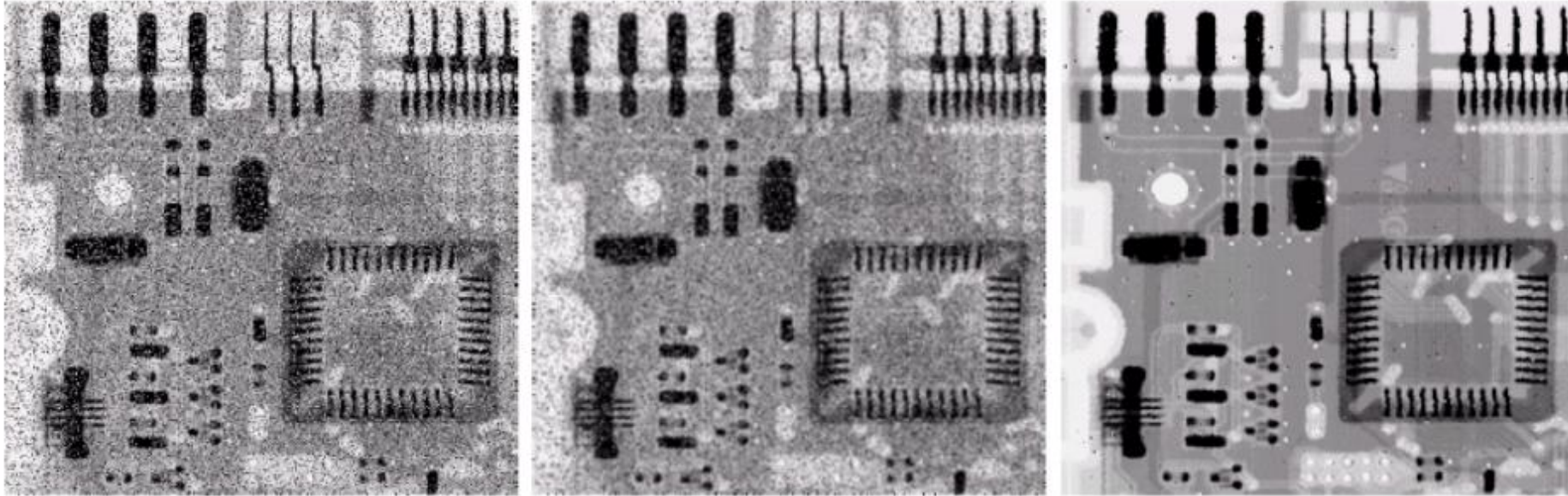
Ưu điểm của Median so với Mean

- **Khử nhiễu muối tiêu (Salt & Pepper noise) tốt hơn.**
 - Mean filter chỉ làm mờ nhiễu, điểm trắng/đen cực trị vẫn còn.
 - Median filter loại bỏ hẳn các điểm nhiễu đơn lẻ.
- **Giữ biên cạnh tốt hơn.**
 - Mean filter làm nhòe biên cạnh do tính trung bình qua vùng chuyển tiếp.
 - Median filter chọn giá trị pixel thực tế, nên biên ít bị mờ.
- **Chống outlier tốt.**
 - Các giá trị cực trị không ảnh hưởng mạnh đến kết quả.

Nhược điểm

- Tính toán chậm hơn Mean (vì phải sắp xếp các giá trị trong cửa sổ).
- Với nhiễu Gaussian (dạng phân bố liên tục), Mean filter thường hiệu quả hơn Median.

Example



a b c

FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Gaussian Kernel

- Bộ lọc Gaussian là bộ lọc làm mượt (smoothing filter) dựa trên phân phối chuẩn (Gaussian distribution).
- Khác với mean filter (trọng số các pixel trong cửa sổ bằng nhau), Gaussian filter gán trọng số lớn hơn cho pixel gần tâm cửa sổ, và trọng số nhỏ dần cho pixel xa hơn.
- Điều này giúp:
 - Làm mượt ảnh tự nhiên hơn.
 - Giữ lại biên cạnh tốt hơn so với mean filter.

Công thức toán học

Hàm Gaussian 2D:

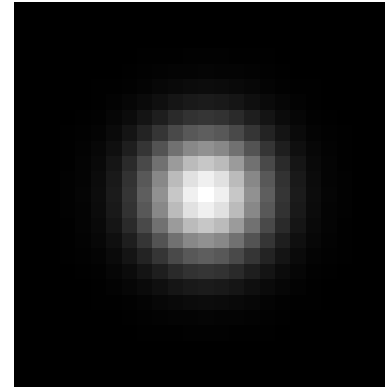
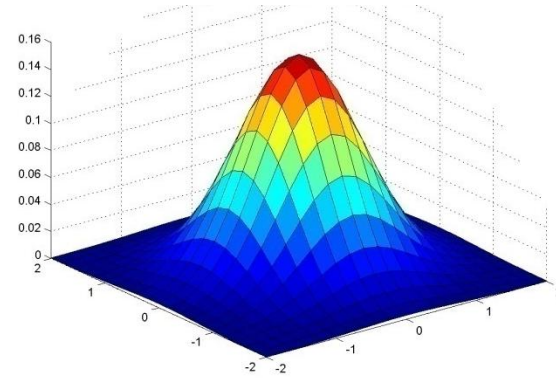
$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

Trong đó:

- x, y : tọa độ pixel tương đối so với tâm kernel.
- σ : độ lệch chuẩn \rightarrow quyết định mức độ làm mờ.
- Kernel thu được sẽ đối xứng và có dạng hình "chuông" (bell-shape).

Sau đó kernel được **chuẩn hóa** để tổng các phần tử = 1, đảm bảo ảnh không bị thay đổi độ sáng trung bình.

Gaussian Kernel



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

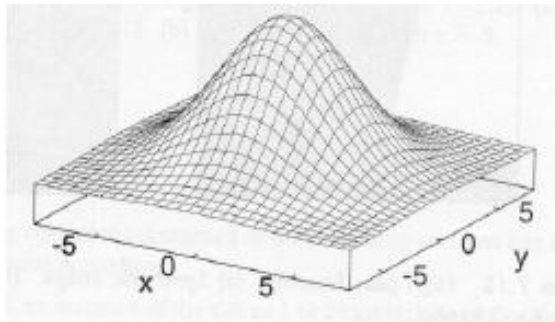
Đặc điểm quan trọng

- Tham số σ :
 - σ nhỏ \rightarrow làm mờ ít.
 - σ lớn \rightarrow làm mờ nhiều.
- Kích thước kernel: thường chọn sao cho size $\approx 6\sigma + 1$ để bao phủ đủ vùng Gaussian.
- Tính chất:
 - Kernel **đối xứng** và **separable** (có thể tách thành nhân theo hàng và cột để tăng tốc tính toán).
 - Là **low-pass filter** \rightarrow loại bỏ nhiễu tần số cao.

Gaussian Smoothing

- **Idea:** replace each pixel by a weighted average of its neighbors
- Mask weights are computed by sampling a Gaussian function

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



7 × 7 Gaussian mask

1	1	2	2	2	1	1
1	2	2	4	2	2	1
2	2	4	8	4	2	2
2	4	8	16	8	4	2
2	2	4	8	4	2	2
1	2	2	4	2	2	1
1	1	2	2	2	1	1

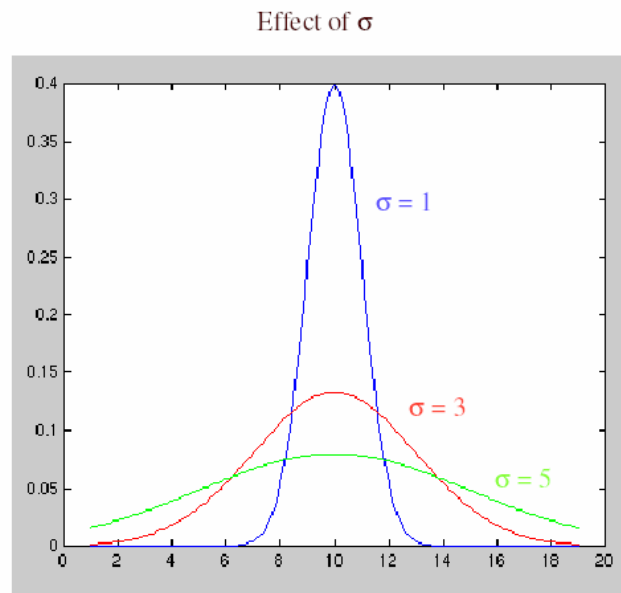
Note: weight values decrease with distance from mask center!

Gaussian Smoothing (cont'd)

mask size depends on σ : *height = width = 5σ* (subtends 98.76% of the area)

- σ determines the degree of smoothing!

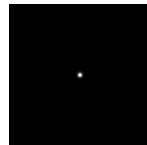
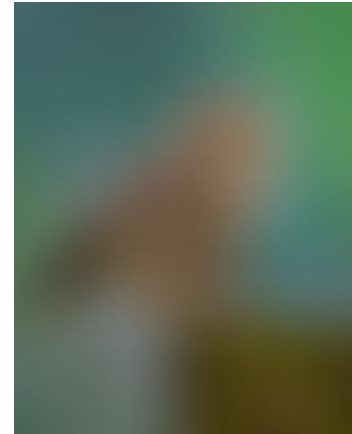
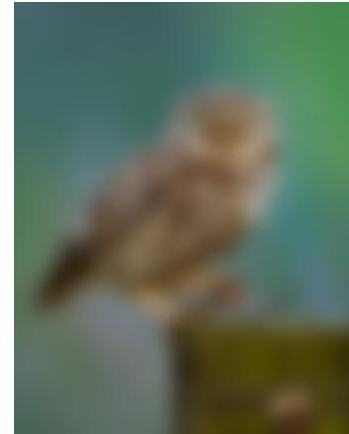
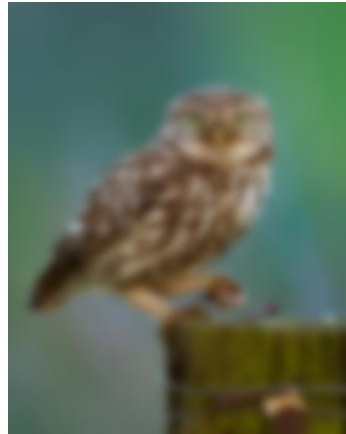
$$\sigma=3$$



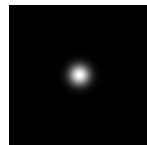
15 × 15 Gaussian mask

2	2	3	4	5	5	6	6	6	5	5	4	3	2	2
2	3	4	5	7	7	8	8	8	7	7	5	4	3	2
3	4	6	7	9	10	10	11	10	10	9	7	6	4	3
4	5	7	9	10	12	13	13	13	12	10	9	7	5	4
5	7	9	11	13	14	15	16	15	14	13	11	9	7	5
5	7	10	12	14	16	17	18	17	16	14	12	10	7	5
6	8	10	13	15	17	19	19	19	17	15	13	10	8	6
6	8	11	13	16	18	19	20	19	18	16	13	11	8	6
6	8	10	13	15	17	19	19	19	17	15	13	10	8	6
5	7	10	12	14	16	17	18	17	16	14	12	10	7	5
5	7	9	11	13	14	15	16	15	14	13	11	9	7	5
4	5	7	9	10	12	13	13	13	12	10	9	7	5	4
3	4	6	7	9	10	10	11	10	10	9	7	6	4	3
2	3	4	5	7	7	8	8	8	7	7	5	4	3	2
2	2	3	4	5	5	6	6	6	5	5	4	3	2	2

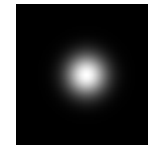
Gaussian filters



$\sigma = 1$ pixel



$\sigma = 5$ pixels



$\sigma = 10$ pixels



$\sigma = 30$ pixels

So sánh với các bộ lọc khác

- **Mean filter:** làm mờ đều → gây nhòe cạnh mạnh hơn.
- **Median filter:** tốt cho nhiễu muối tiêu nhưng không phù hợp với nhiễu Gaussian.
- **Gaussian filter:** tốt để giảm nhiễu Gaussian, giữ biên cạnh tốt hơn Mean.

Salt & Pepper Noise



Gaussian noise

3x3

Mean



Gaussian



Median



5x5



7x7



Effect of mean filters

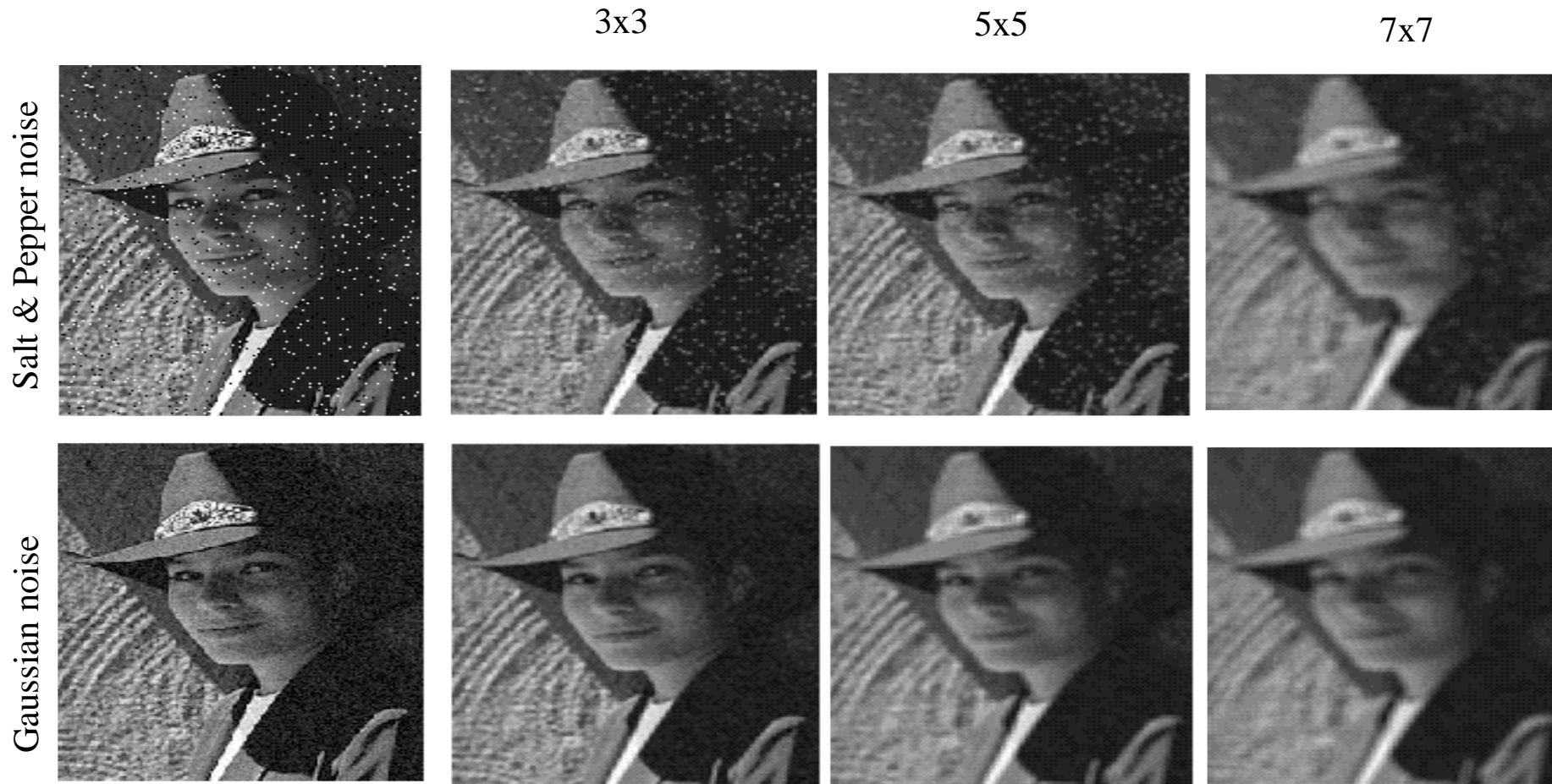
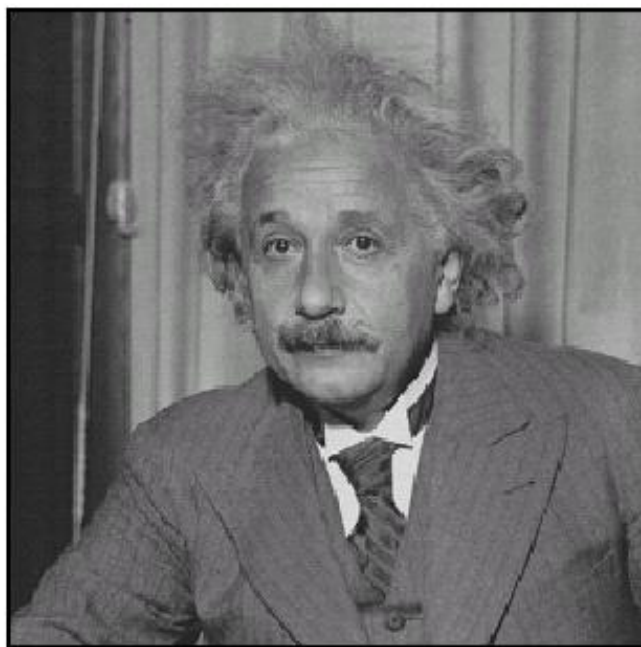
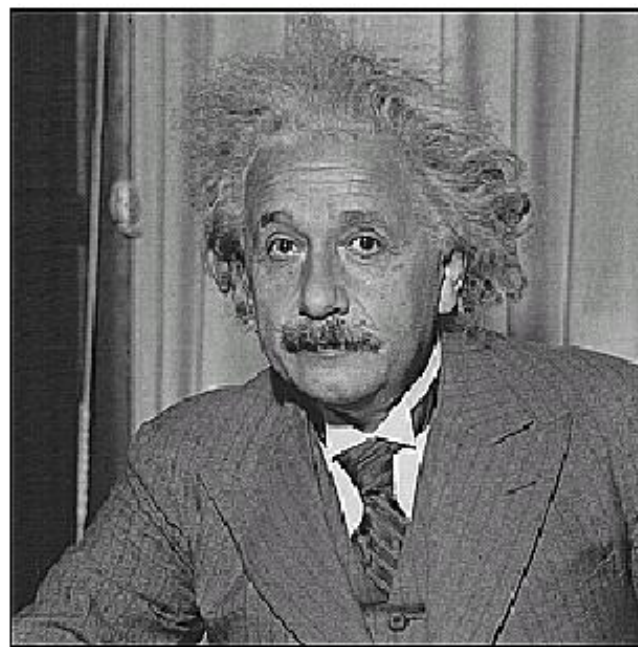


Image Sharpening



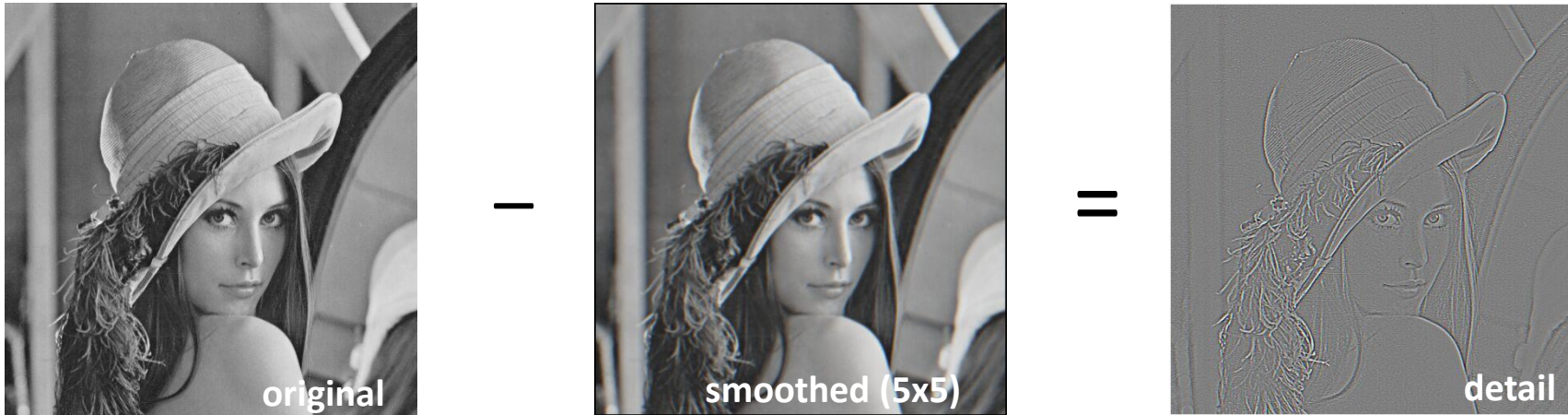
before



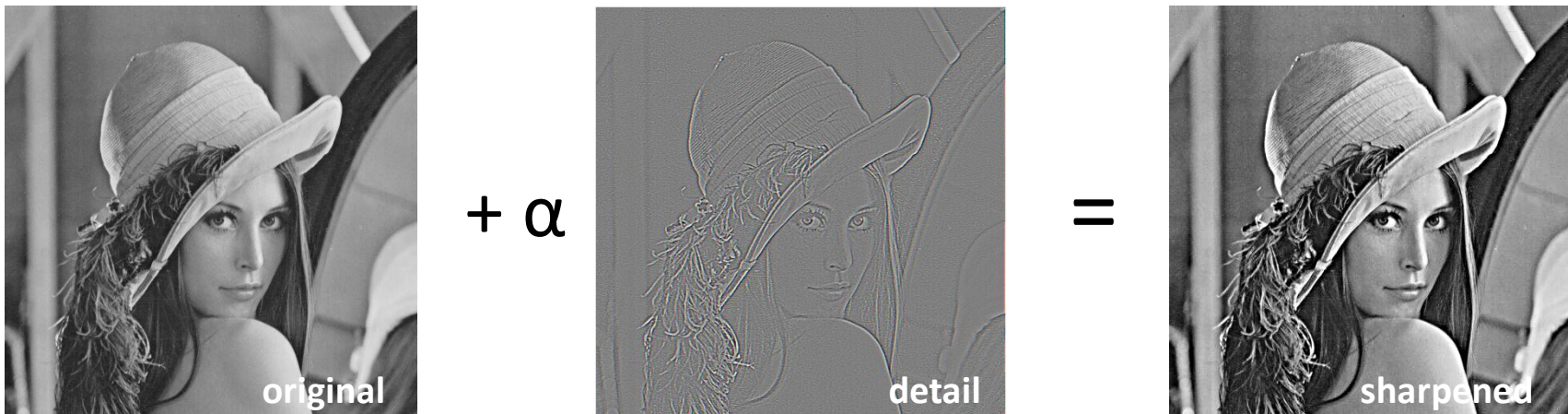
after

Sharpening filter

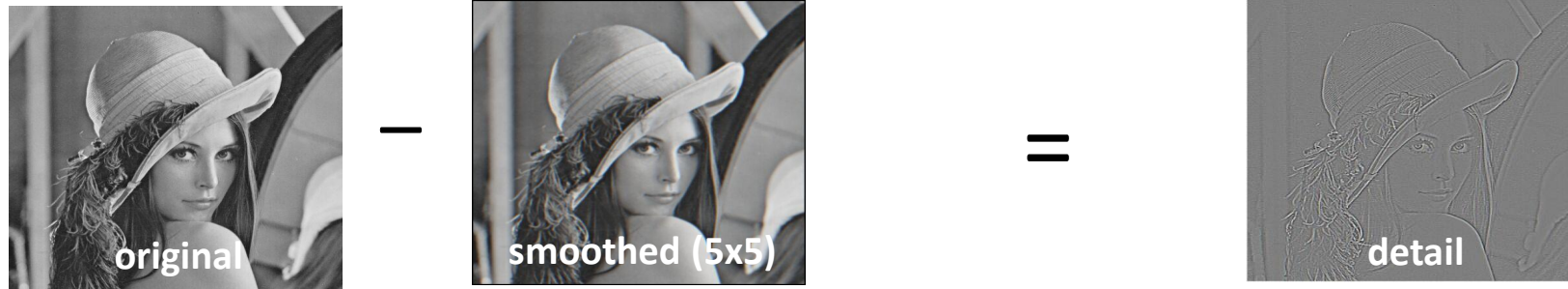
- Step 1: Original - Smoothed = "Details"



- Step 2: Original + "Details" = Sharpened



- Step 1: Original - Smoothed = "Details"




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

- Step 2: Original + "Details" = Sharpened



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


Result:



•0	•0	•0
•0	•2	•0
•0	•0	•0

 $- \frac{1}{9}$

•1	•1	•1
•1	•1	•1
•1	•1	•1

 $=$ 

Original

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

$$M = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -7 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Noise – nhiễu

■ Types of noise:

- Salt and pepper noise
- Impulse noise: nhiễu xung
- Gaussian noise:

■ Nguyên nhân

- Lỗi truyền tải - transmission errors
- Đốm trên ống kính
- Do cảm biến
- ...



Original



Salt and pepper noise



Impulse noise



Gaussian noise

Noise



Salt and pepper noise



Impulse noise



Gaussian noise

Noise

```
def add_salt_pepper(img, prob=0.02):  
    noisy = img.copy()  
    rnd = np.random.rand(*img.shape)  
  
    noisy[rnd < (prob/2)] = 0          # Pepper (đen)  
    noisy[rnd > 1 - (prob/2)] = 255  # Salt (trắng)  
  
    return noisy  
  
noisy_img = add_salt_pepper(img, prob=0.05)
```

Noise

```
# Tạo nhiễu Gaussian (mean=0, sigma=25)  
noise = np.random.normal(0, 25, img.shape)  
  
# Thêm vào ảnh gốc  
noisy_img = img + noise  
  
# Giới hạn về [0,255] và chuyển về uint8  
noisy_img = np.clip(noisy_img, 0, 255).astype(np.uint8)
```

Noise Removal (Image Smoothing)

■ Noise removal:

- Loại bỏ các chấm/đốm trên ảnh.
- Các dot có thể mô hình hóa như các impulses -xung (salt-and-pepper or speckle) or continuously varying (Gaussian noise)
- Có thể được loại bỏ bằng cách lấy giá trị trung bình hoặc trung vị của các pixel lân cận.

Exercises

1) Lọc ảnh sau:

100	100	100	100	100
100	200	205	203	100
100	195	200	200	100
100	200	205	195	100
100	100	100	100	100

- a) Mean filter
- b) Gaussian filter
- c) Median filter

Exercises

2) Làm sắc nét ảnh sau:

1	6	9	10	2	8
2	5	1	8	4	2
3	7	4	9	10	3
9	8	3	6	7	9
8	0	9	4	7	2
9	10	12	6	9	8