

**COLE.VN**  
connecting knowledge

**Chủ đề:**  
**Xử lý ảnh (Image Processing)**

# Nội dung chính

**Tổng quan về xử lý ảnh**

**Những vấn đề trong xử lý ảnh**

# Thị giác máy tính ?

- Xử lý ảnh
  - Làm việc trên ảnh như một ma trận số
  - Đầu vào: ảnh số ở đầu ra: ảnh số (ma trận) – Hỗ trợ kiểm tra và sửa đổi ảnh
- Thị giác máy tính
  - Làm máy tính hiểu nội dung ảnh số và video số – Ảnh và video coi như dữ liệu đầu vào
  - Đầu ra: thông tin ngữ nghĩa, thông tin 3D

# Ứng dụng thị giác máy tính

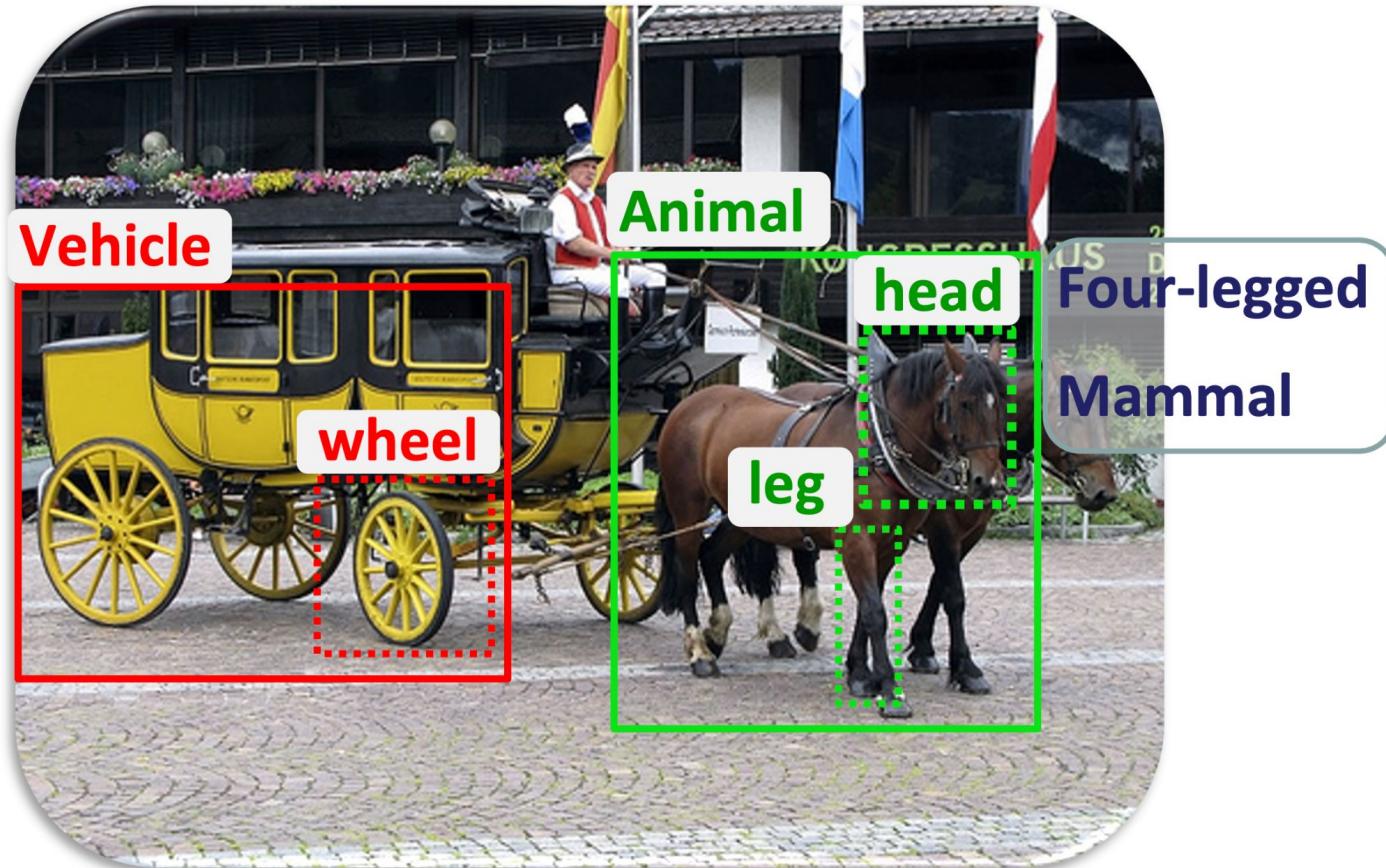
- Ảnh, video:

- Nguồn dữ liệu dồi dào, môi trường đa dạng
- Giàu thông tin  
=> Lĩnh vực thu hút nhiều quan tâm đặc biệt trong kỷ nguyên mới ..



# Ứng dụng thị giác máy tính

## Hiểu nội dung ảnh



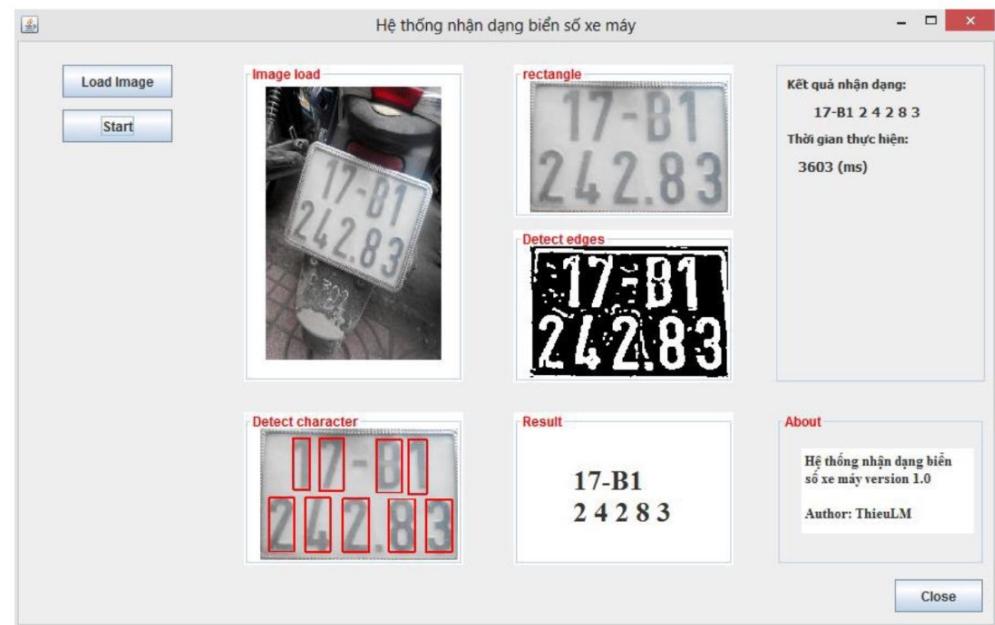
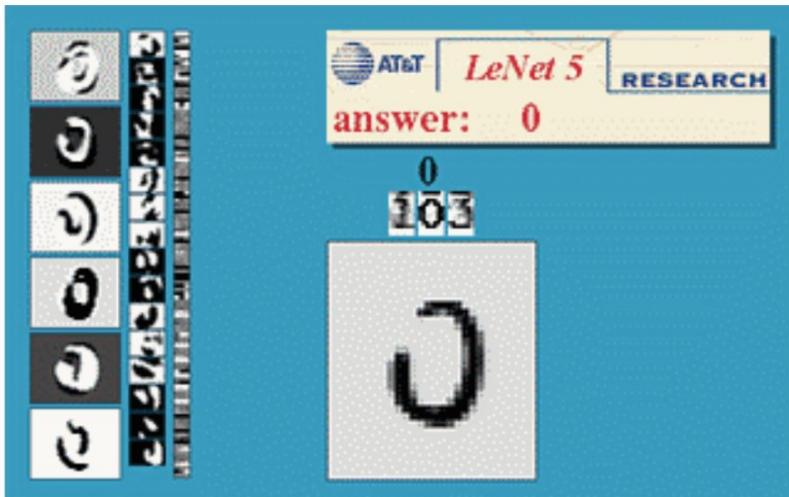
# Ứng dụng thị giác máy tính

## Hiểu nội dung ảnh



# Ứng dụng thị giác máy tính

## OCR (Optical character recognition)



# Ứng dụng thị giác máy tính

Phát hiện mặt người: được tích hợp trong hầu hết các camera để focus tự động, cho phép có các bức ảnh đẹp



# Ứng dụng thị giác máy tính

Ghép ảnh toàn cảnh



# Một số topics trong CV

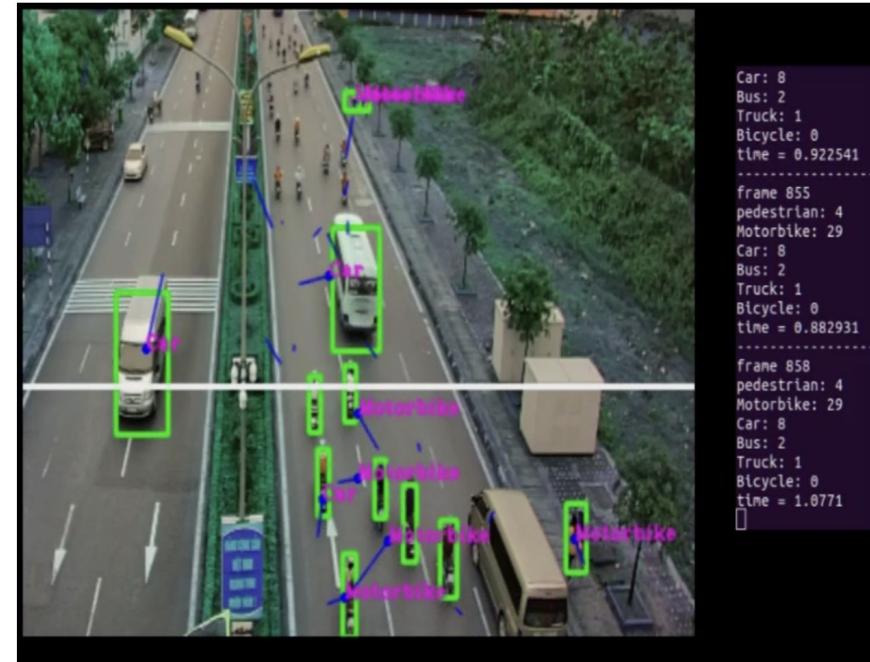
Camera giám sát:

- Đếm số lượng khách hàng trong cửa hàng
- Phát hiện hành động bất thường
- Đo mức độ hài lòng của khách hàng

Phát hiện và nhận dạng đối tượng

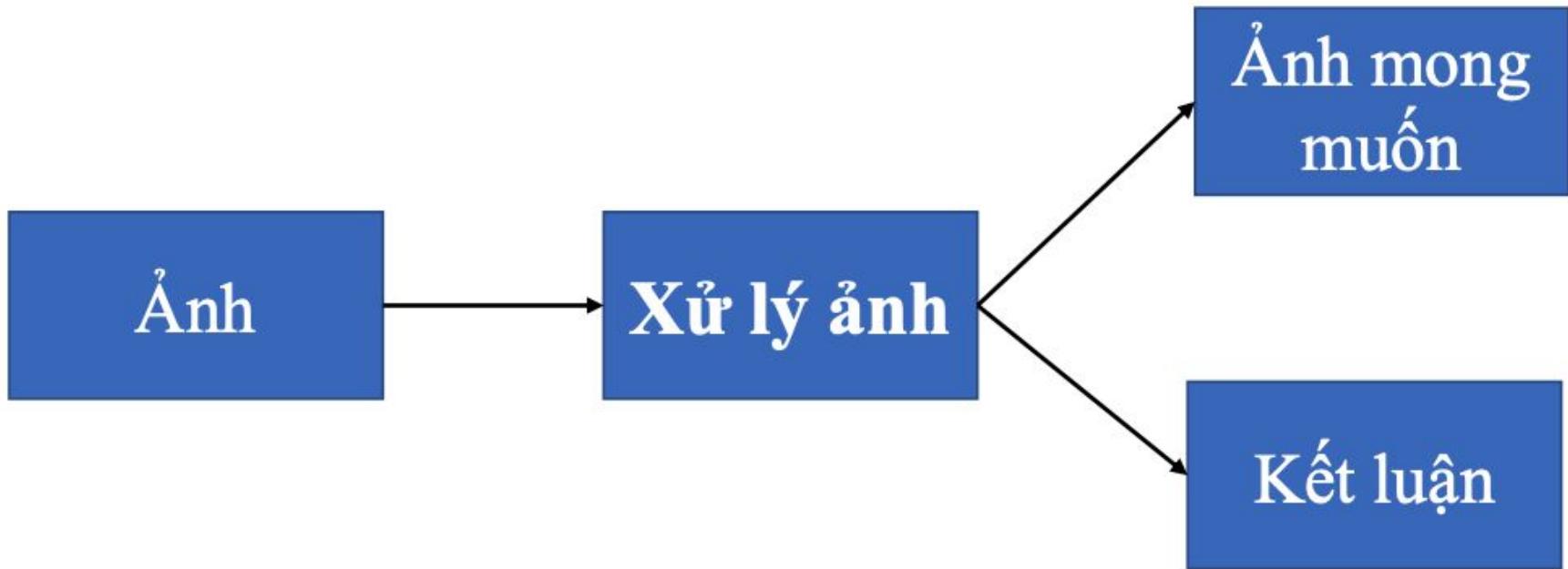
Phát hiện mặt/mắt/người

- Nhận dạng hoạt động
- Phát hiện lỗi
- Gán nhãn ảnh số
- Phát hiện nhận dạng ký tự
- Đọc card visit, CMT, biển số,...
- Xây dựng đối tượng 3D từ ảnh 2D

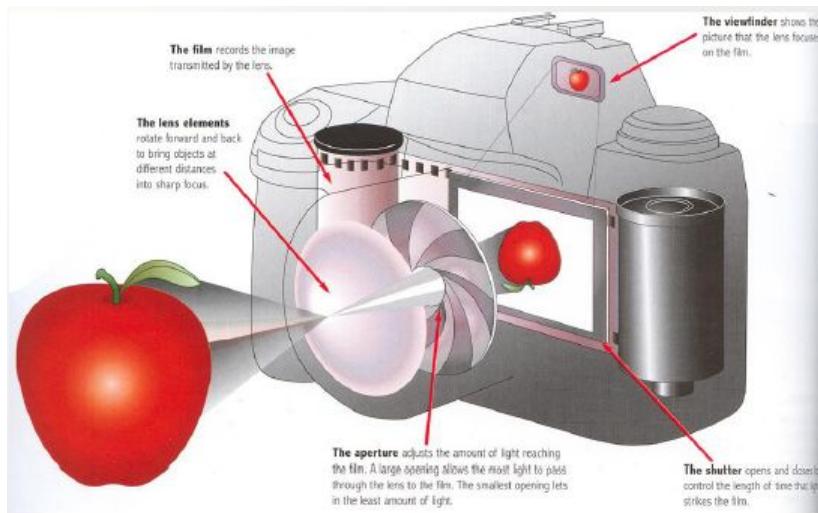


# Tổng quan các bài toán XLA

- Khái niệm xử lý ảnh: là quá trình thao tác ảnh đầu vào nhằm cho ra kết quả mong muốn.
- Kết quả đầu ra của một quá trình xử lý ảnh có thể là một ảnh “tốt hơn” hoặc là một kết luận

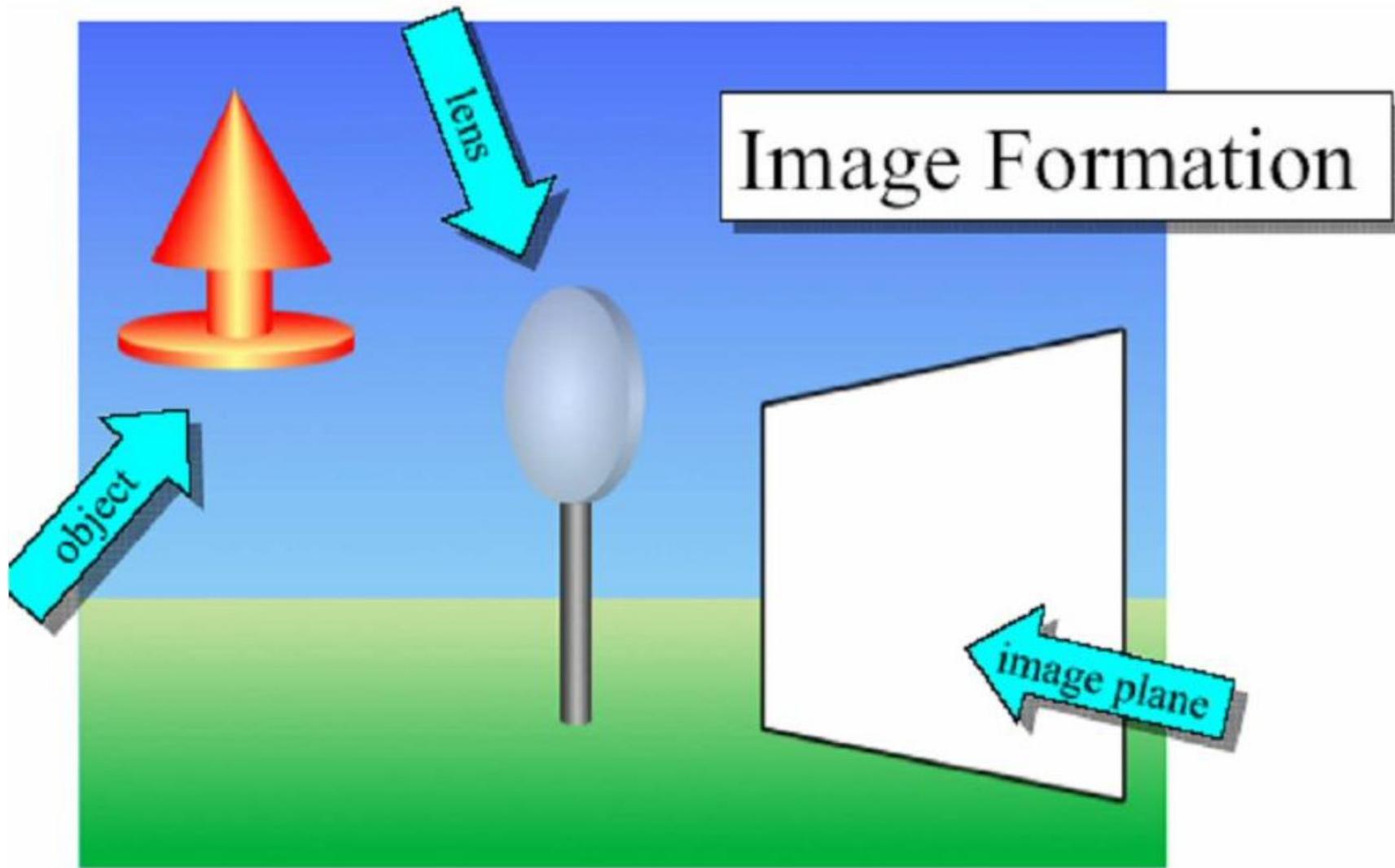


# 1. Quá trình thu nhận và hình thành ảnh



- Ống kính và điểm nhìn xác định phối cảnh
- Độ mở ống kính, tốc độ đóng cửa sổ chụp quyết định độ sáng của ảnh
- Độ mở, khoảng cách focal length quyết định độ sâu
- Film hay cảm biến cho phép lưu ảnh

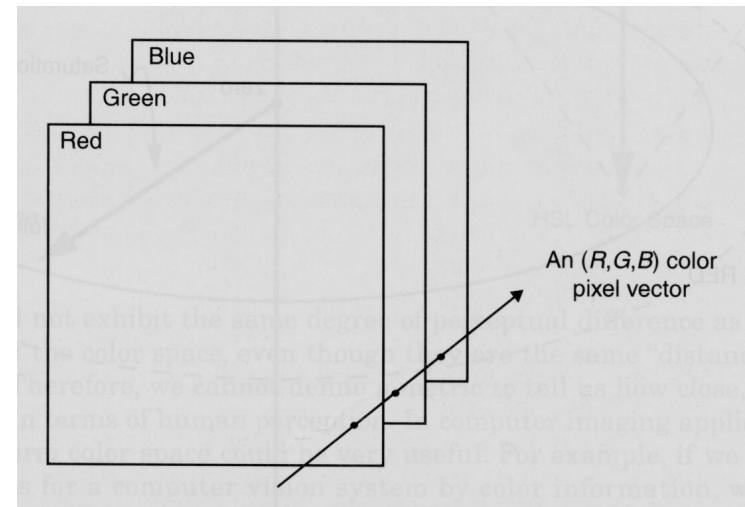
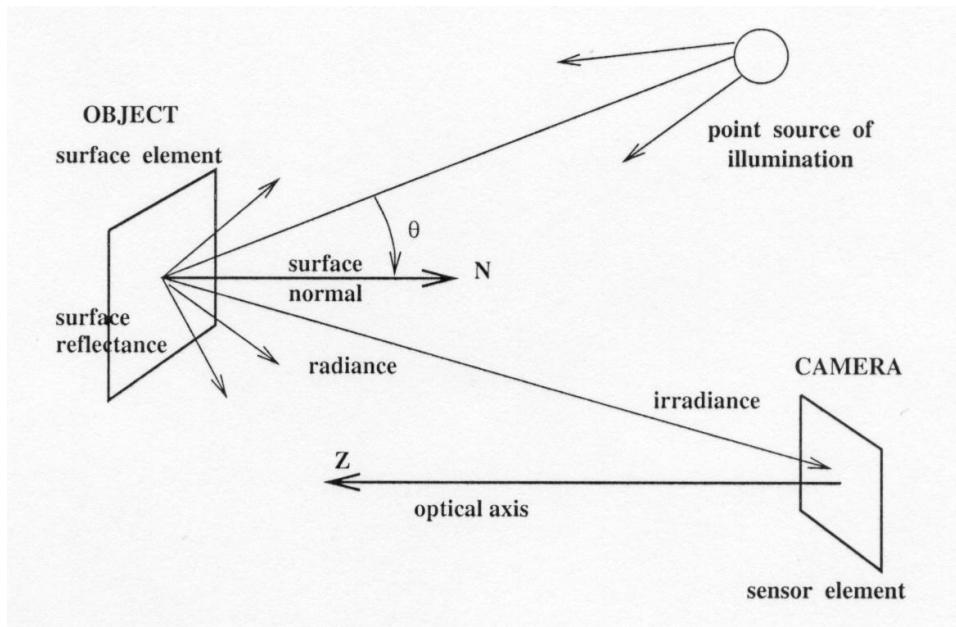
# Mô hình pin-hole camera



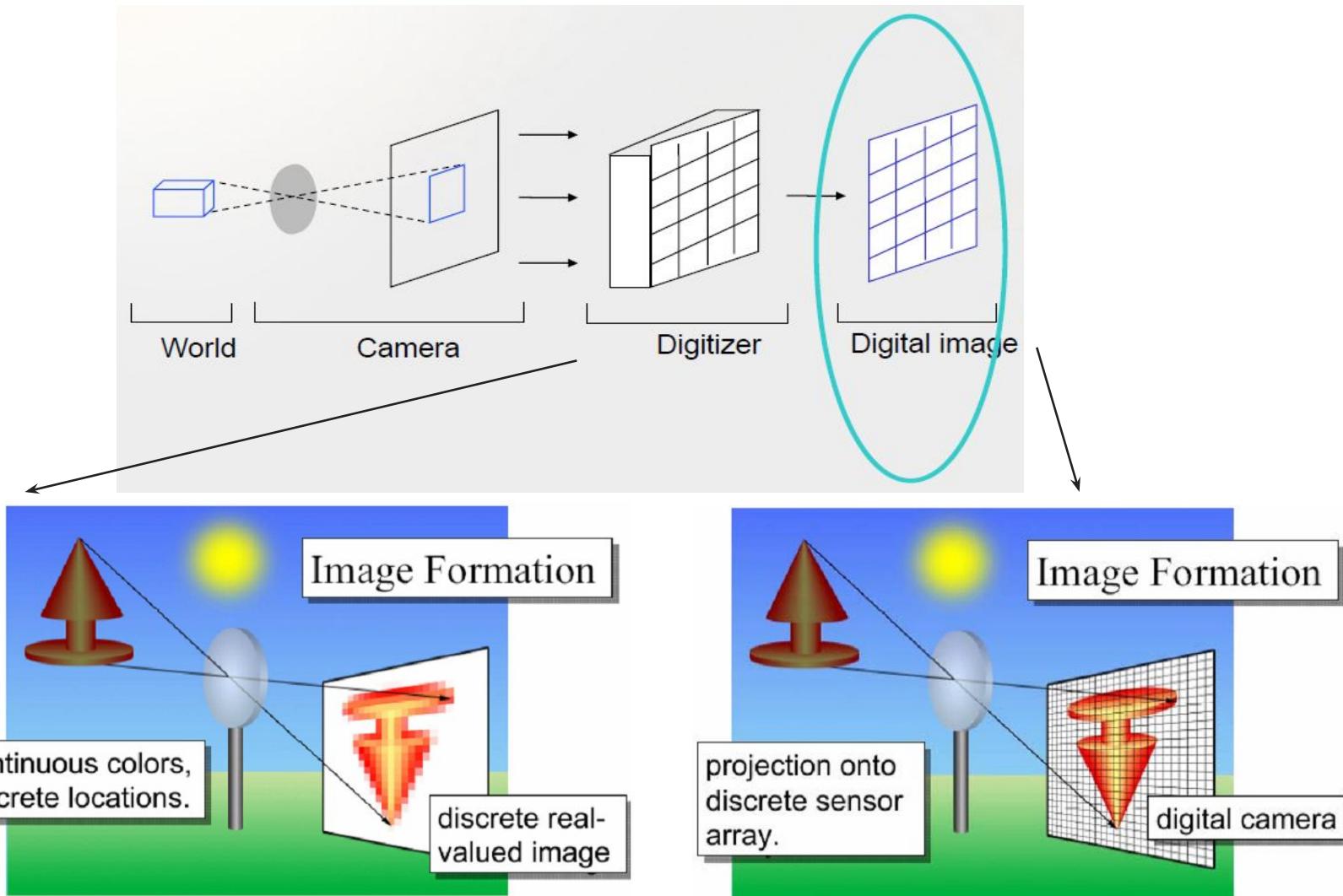
# Quá trình thu nhận ảnh

- ◆ Nguồn sáng chiếu vào đối tượng
- ◆ Phản xạ từ bề mặt đối tượng
- ◆ Hướng nhìn của camera

}  **Ảnh hưởng tới  
giá trị điểm  
ảnh**

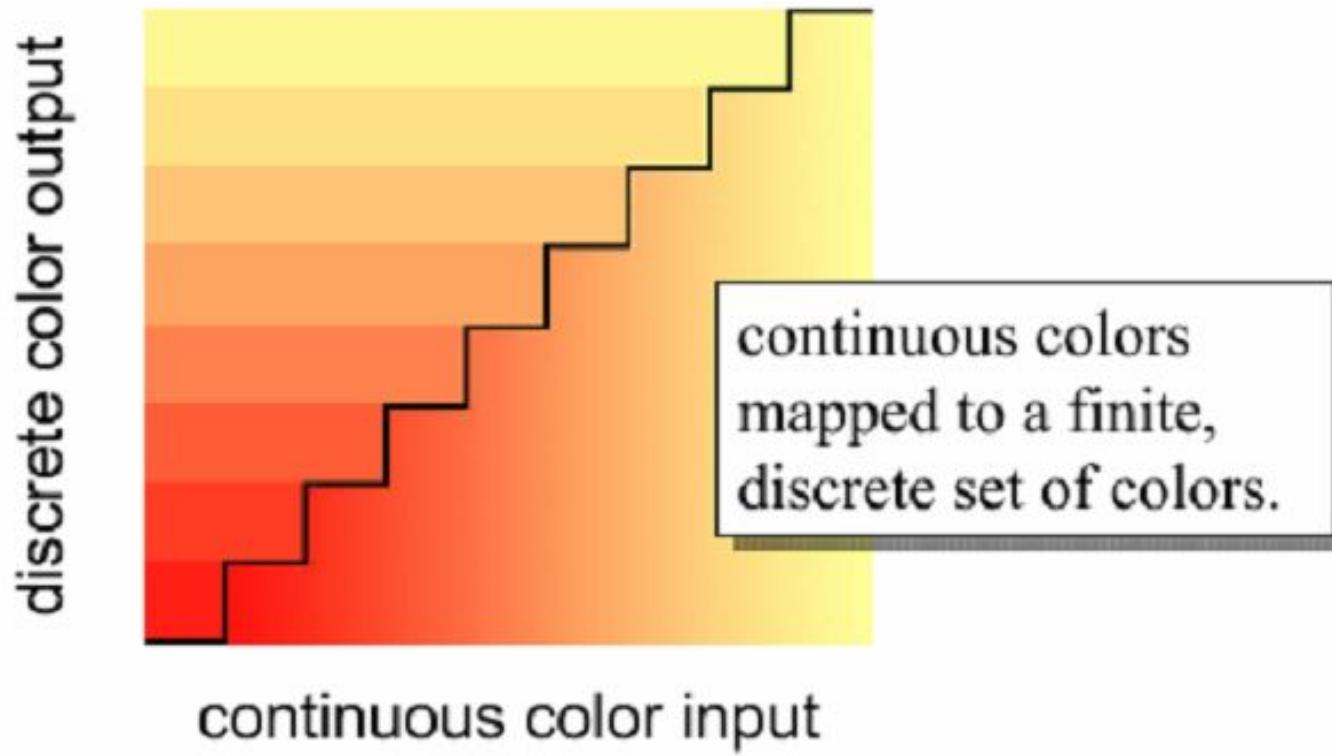


# Quá trình hình thành ảnh số

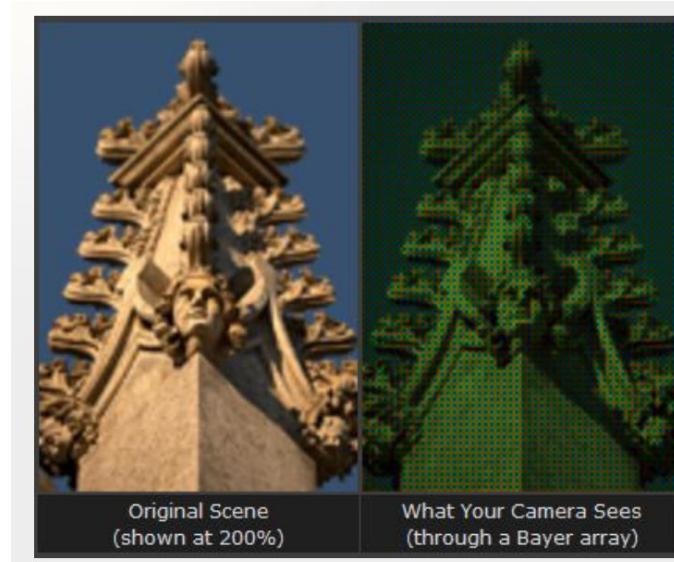
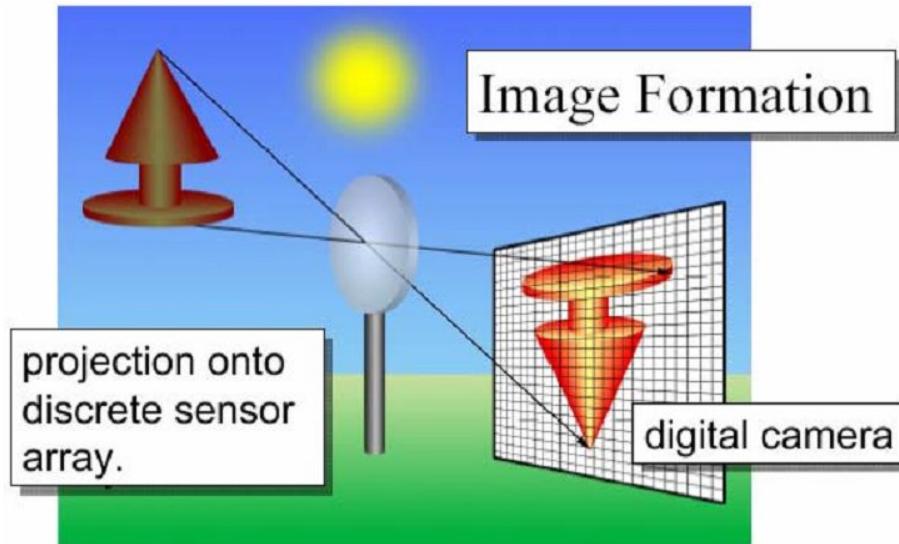


## Hình thành ảnh số

- ◆ Hình thành ảnh số là: ánh xạ miền màu liên tục (vô hạn) vào miền hữu hạn (tập màu rời rạc)



# Hình thành ảnh số



Quá trình hình thành ảnh  
(Số hóa – Digitization)

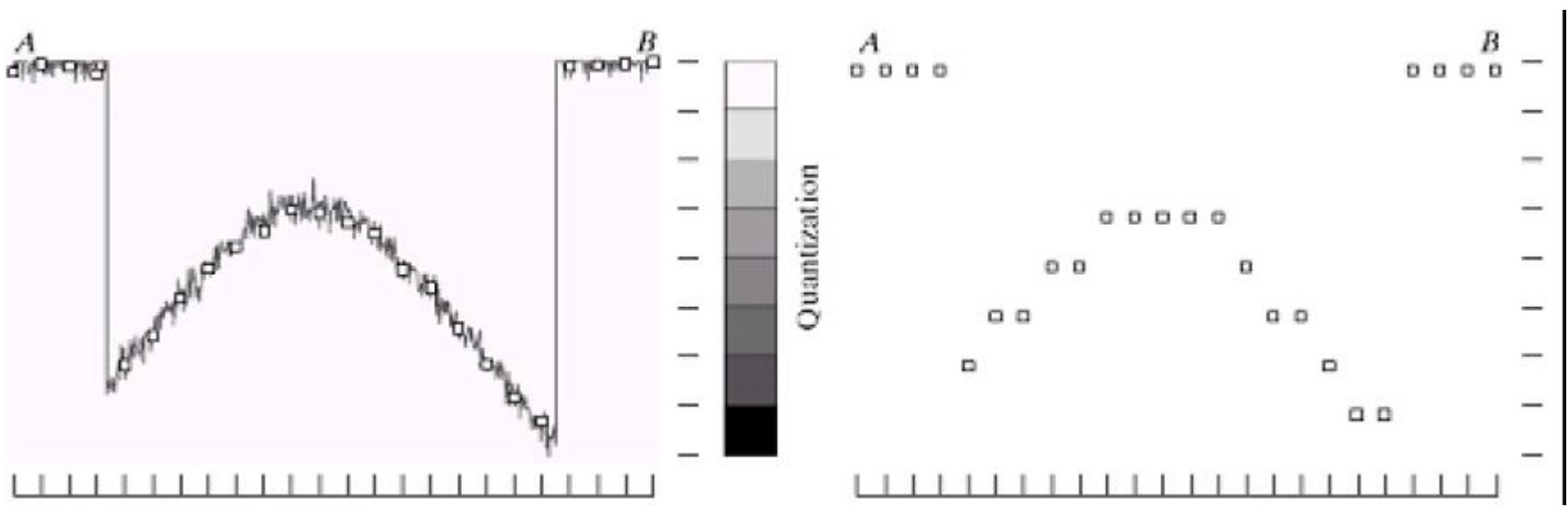
Ảnh  
gốc

Ảnh đã  
được số  
hóa

Digitization = Sampling + Quantization  
(Lấy mẫu + Lượng tử hóa)

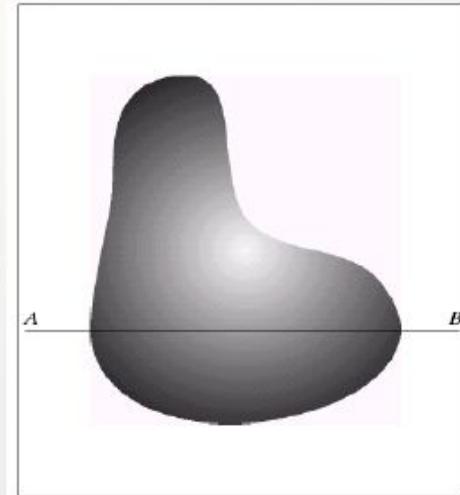
# Quá trình lấy mẫu và lượng tử hóa

- ◆ Sampling (Lấy mẫu): Bị giới hạn (quy định) bởi kích thước của cảm biến (kích thước của ma trận điểm ảnh trên cảm biến)
- ◆ Mức lượng tử hóa: bị hạn chế (quy định) bởi số mức ánh sáng định nghĩa trong 1 giải liên tục

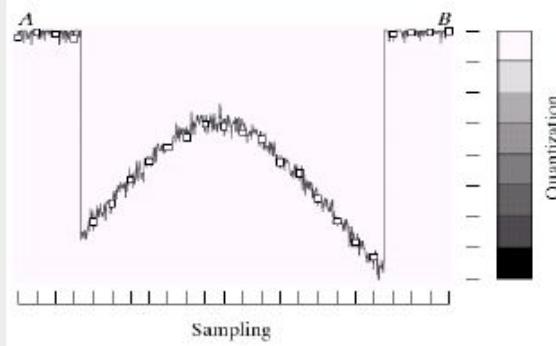


# Quá trình lấy mẫu và lượng tử hóa

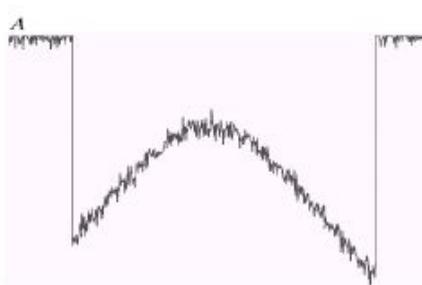
Ảnh gốc



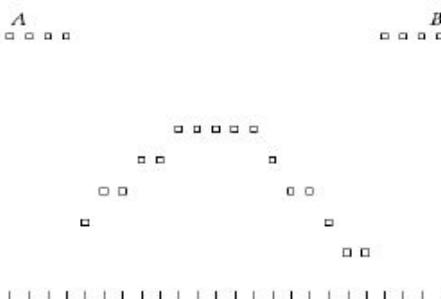
Lấy mẫu



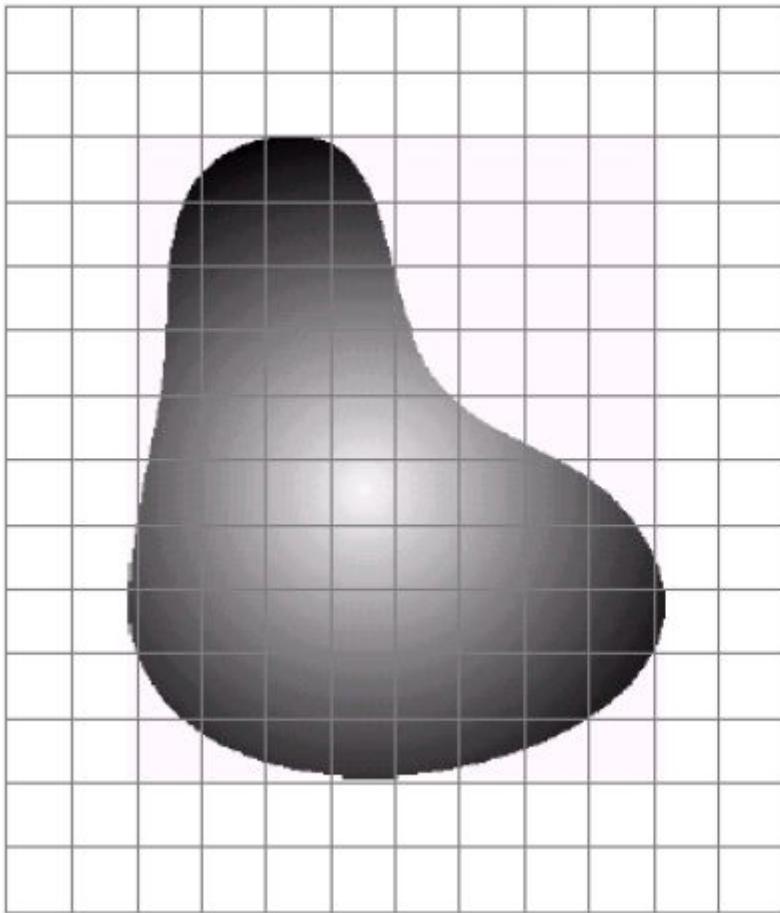
Mức sáng gốc  
trên đường AB



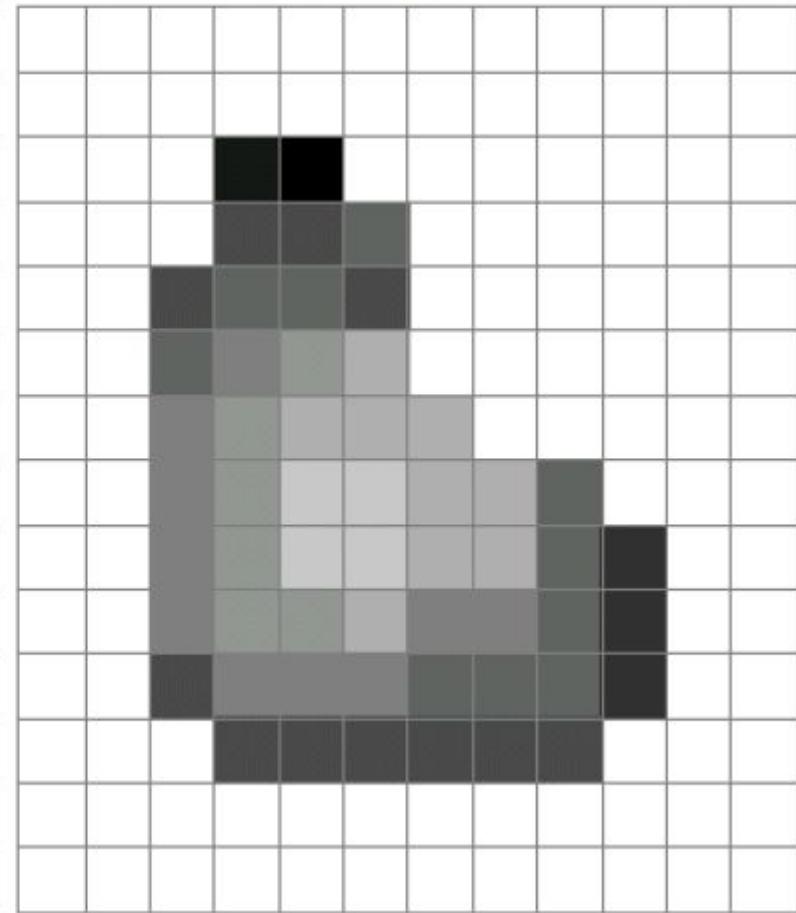
Lượng tử hóa  
theo các mức



# Quá trình lấy mẫu và lượng tử hóa



Ảnh liên tục



Ảnh đã được số hóa  
Qua quá trình lấy mẫu và lượng tử hóa

# Biểu diễn ảnh số

Loại ảnh chính

- Ảnh nhị phân:
  - $I(x,y) \in \{0,1\}$
  - 1 pixel: 1 bit
- Ảnh đa mức xám:
  - $I(x,y) \in [0..255]$
  - 1 pixel: 8 bits (1 byte)
- Ảnh màu:
  - $I_R(x,y), I_G(x,y), I_B(x,y) \in [0..255]$
  - 1 pixel: 24 bits (3 bytes )
- Khác : ảnh đa phông, ảnh độ sâu,...

Binary



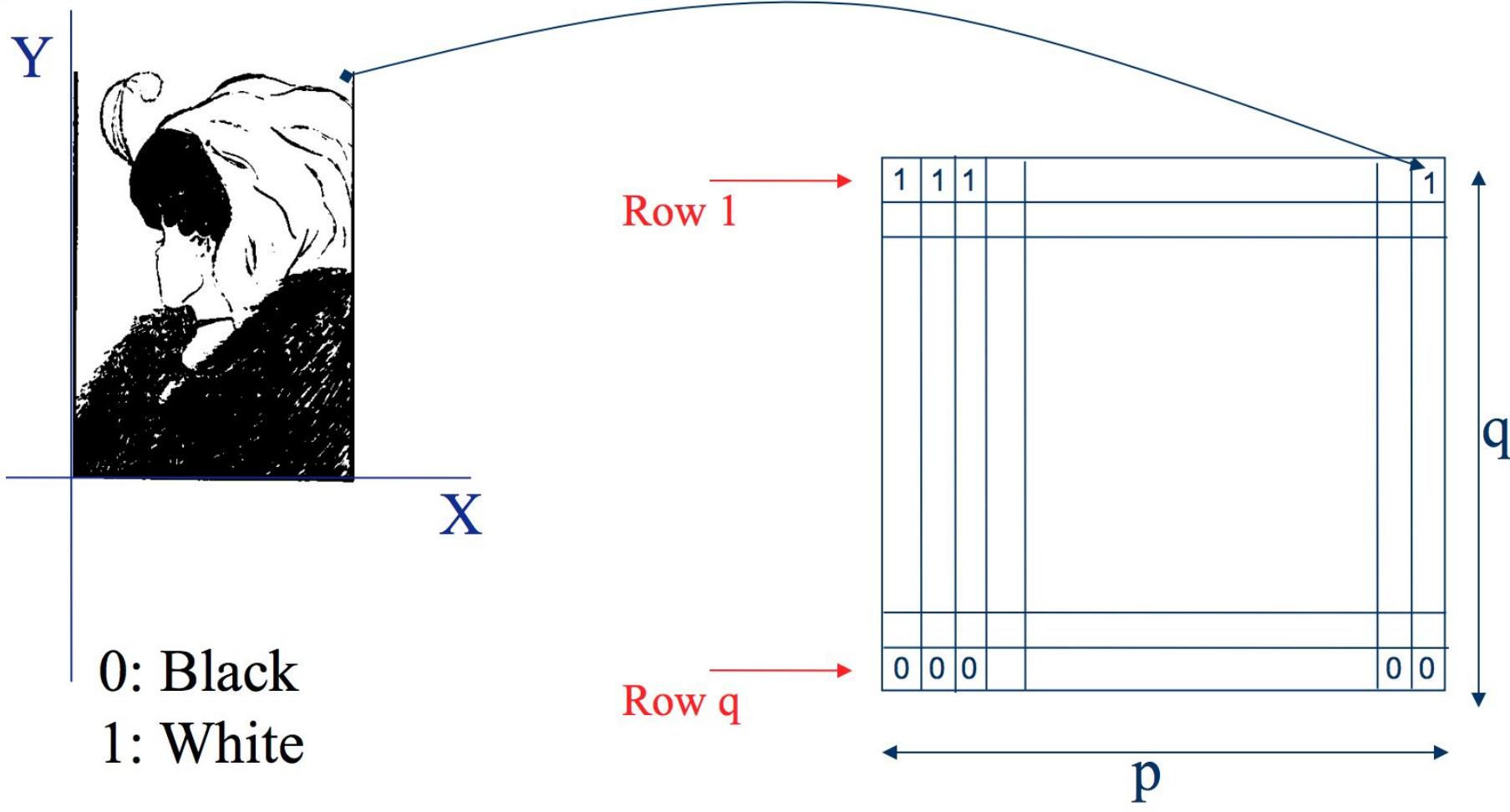
Gray Scale



Color



# Biểu diễn ảnh nhị phân



Ma trận: [ hàng x cột ]  
Giá trị: 0 | 1 (or 0 | 255)

# Biểu diễn ảnh nhị phân



A 3x8 grid of binary values represented by small squares. The first three columns contain the numbers 10, 5, and 9 respectively. The remaining five columns are empty. A blue arrow points from the top-left corner of the grid to the first cell containing the value 10. Another blue arrow points from the bottom-right corner of the grid to the cell containing the value 100. To the right of the grid, there is a vertical color bar with a grayscale gradient and numerical labels: 0, 50, 100, 150, 200, and 250.

10	5	9					

Ma trận: [ hàng x cột ]  
Giá trị: 0 .. 255

# Biểu diễn ảnh màu



Phil Noble / AP



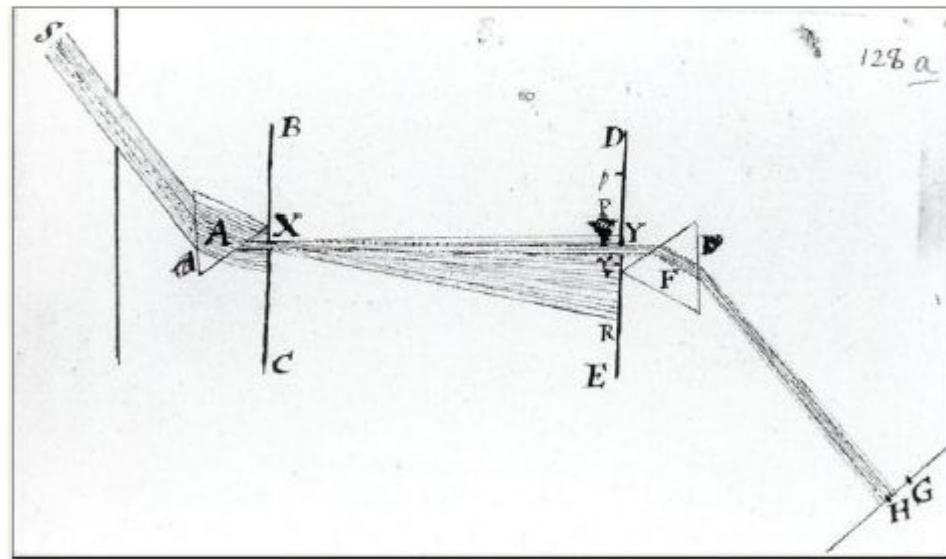
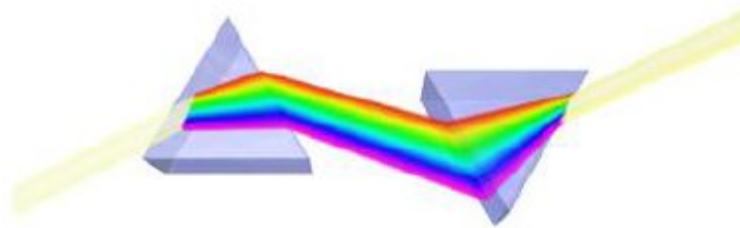
Phil Noble / AP



Phil Noble / AP

# Vấn đề màu sắc trong ảnh

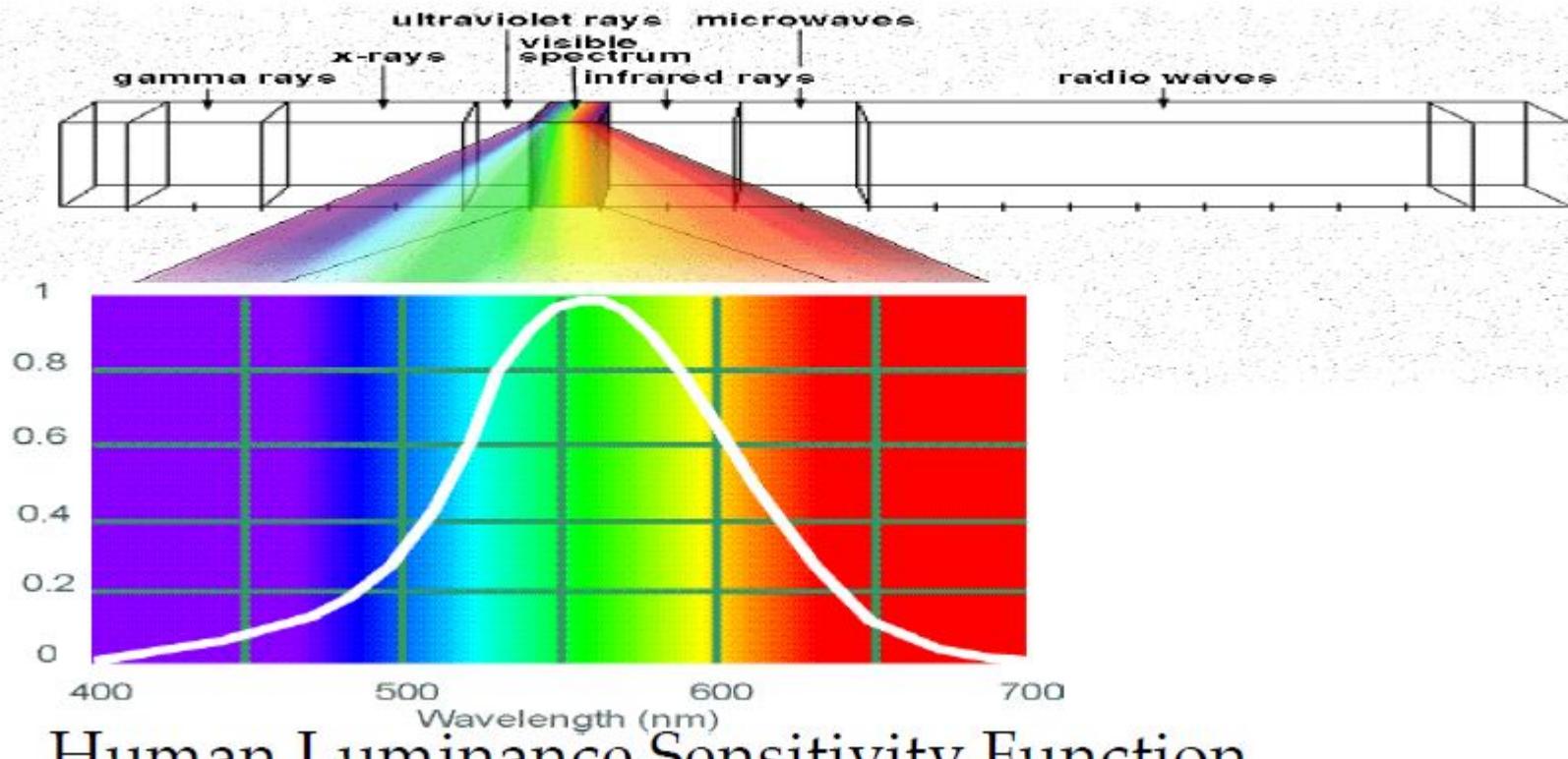
- Hình thành ảnh là do nguồn sáng (gồm 1 chùm các tia sáng chiếu tới)



Newton 1665

# Ánh sáng trắng

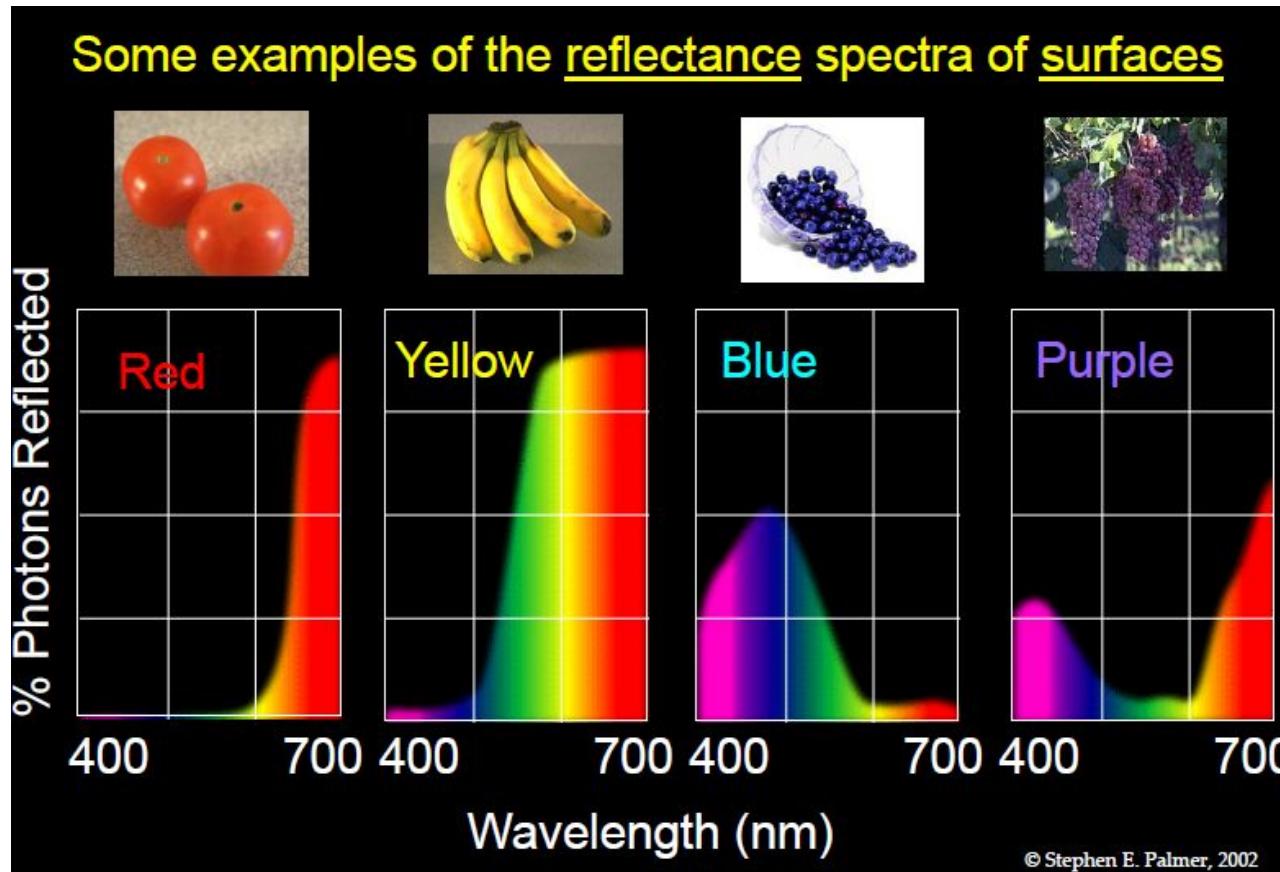
- Dải quan sát được bởi mắt người (visible light)



Human Luminance Sensitivity Function

# Cảm nhận được màu sắc

- Là do đáp ứng bề mặt của vật liệu khác nhau với chùm sáng chiếu tới

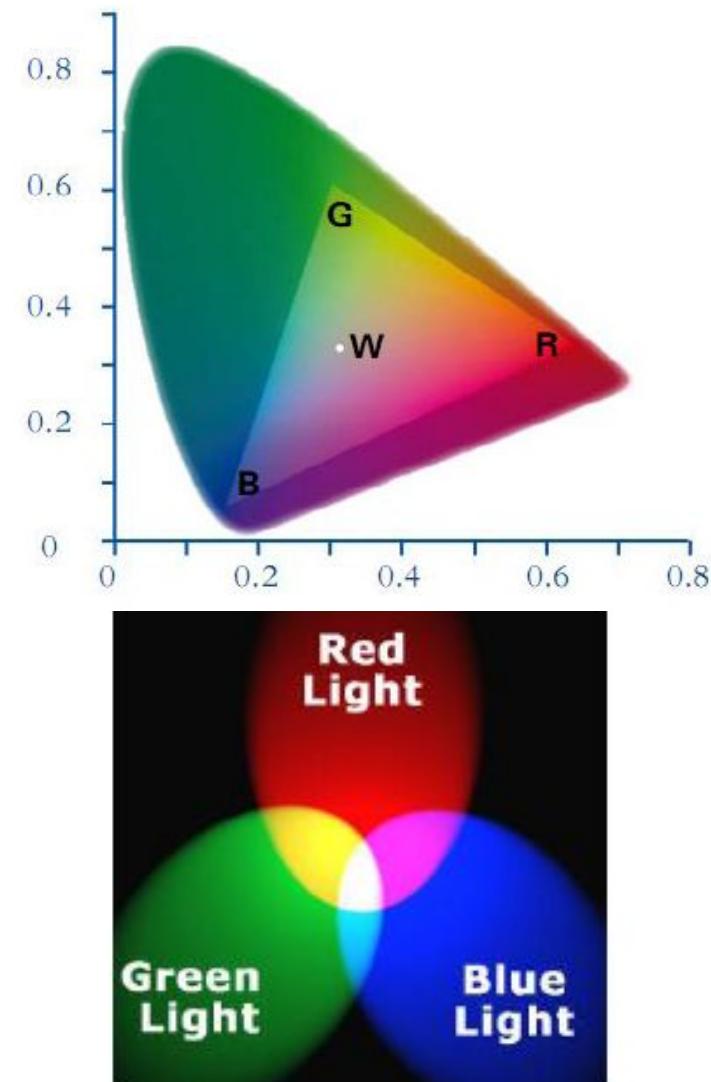


# Không gian màu sắc

- Định nghĩa: Là không gian gồm các thành phần màu được kết hợp để biểu diễn màu sắc
- Các không gian màu thường sử dụng:
  - RGB, HSV, CMY ( Cyan, Megenta, Yellow) , etc
- Tại sao có nhiều hơn một không gian màu (?)
  - Do đặc thù của ứng dụng (in ấn màu, hiển thị (monitor) màu)
  - Một số không gian màu độc lập hoặc phụ thuộc thiết bị
  - Một số không gian màu được cảm nhận tuyến tính, một số khác không tuyến tính
  - etc

# Không gian màu RGB

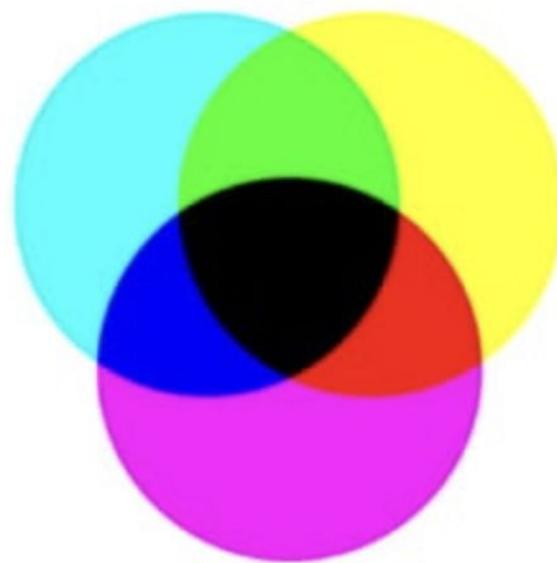
- Đây là không gian màu phổ biến, được dùng trong các thiết bị hiển thị
- Diễn hình của hệ thống phối màu cộng : chồng 3 kênh R, G, B tạo nên một màu
- Không gian màu RGB thường không tuyến tính với việc cảm nhận màu của mắt người



# Không gian màu CMY (Cyan, Magenta, Yellow)

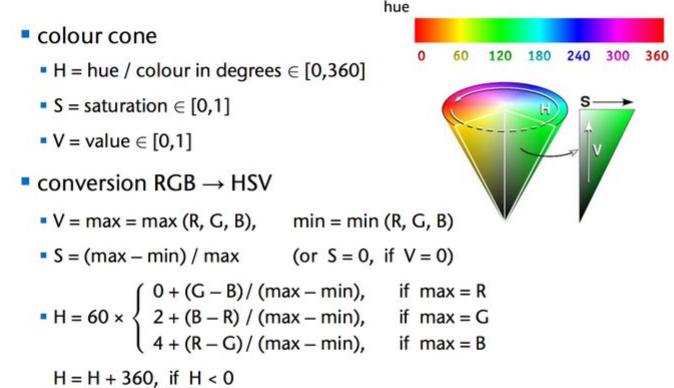
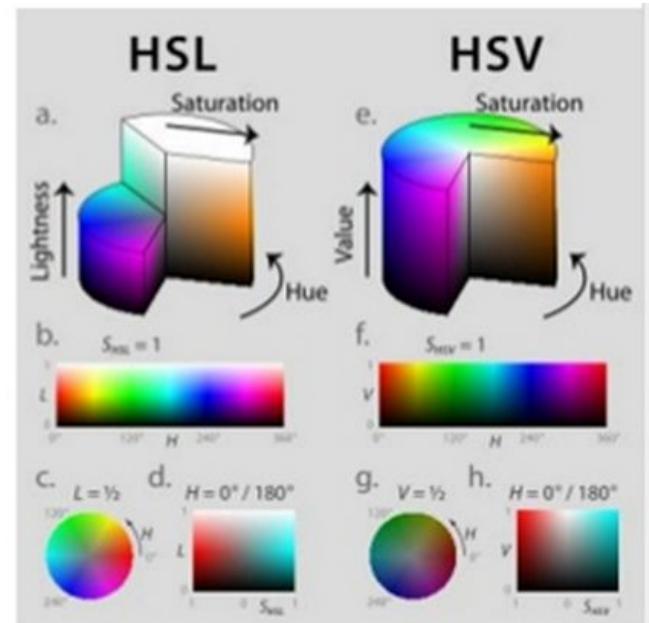
- Thường sử dụng cho máy in , photocopy màu
- Là không gian màu phụ thuộc thiết bị,
- Không tuyến tính với cảm nhận của mắt người, không trực quan
- Không độc lập với thiết bị
- Diễn hình của hệ thống phối màu trừ

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$



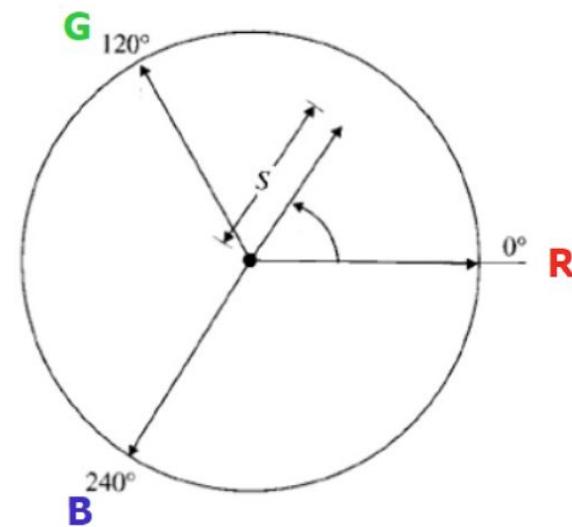
# Không gian màu HSV/HSL

- Mô hình màu **HSV**(Hue, Saturation, Value) và **HSL**(Hue, Saturation, Lightness)
- Mô hình màu này muốn sắp xếp lại hệ màu **RGB** hay **CMY** theo một cách dễ dàng hơn.
- Trong đó Hue là tông màu, Saturation là sắc độ, Value là giá trị cường độ sáng còn Lightness là độ sáng (từ đen đến trắng)



# Không gian màu HSV/HSL

- Hue (H) được mã hóa như 1 góc thay đổi giữa 0 và 360
- Saturation (S) được mã hóa như độ dài của bán kính, giá trị từ 0 đến 1
  - S = 0: xám
  - S = 1: màu tinh khiết
- Value (V) = MAX (Red, Green, Blue)

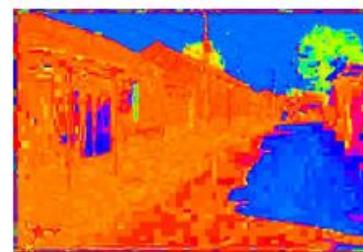
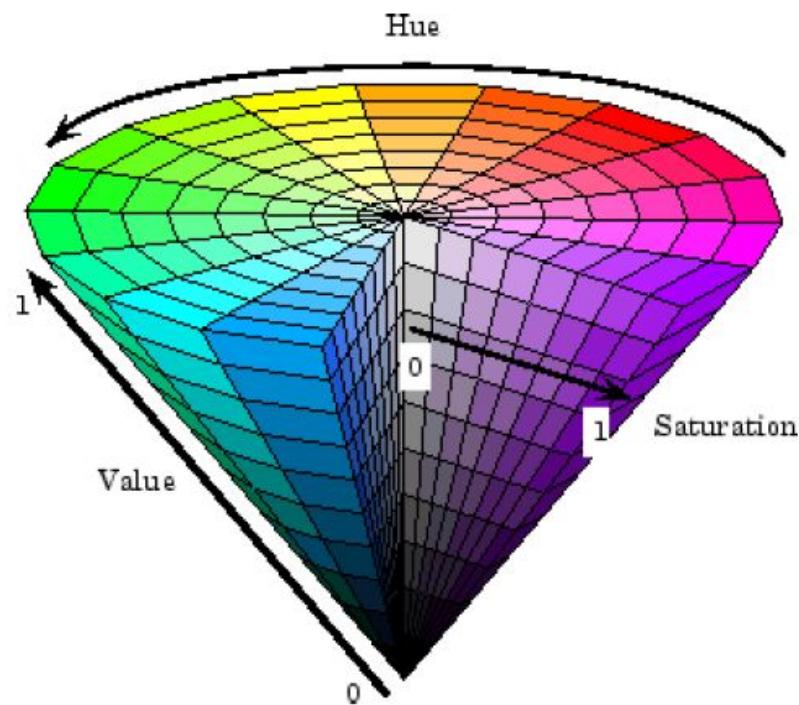


# Ví dụ minh họa

## Color spaces: HSV



Intuitive color space



**H**  
( $S=1, V=1$ )



**S**  
( $H=1, V=1$ )



**V**  
( $H=1, S=1$ )

**R**

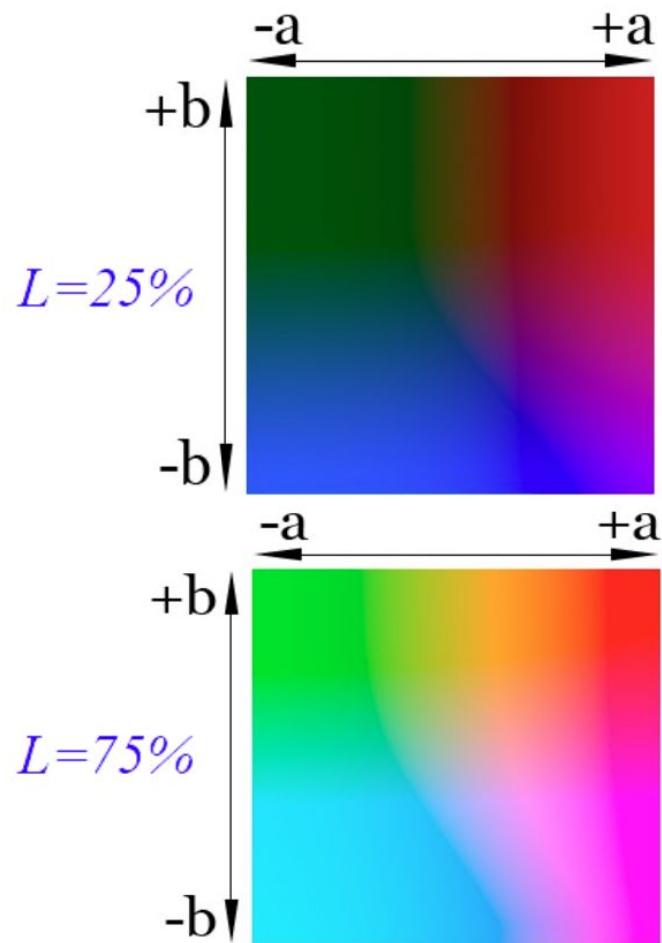
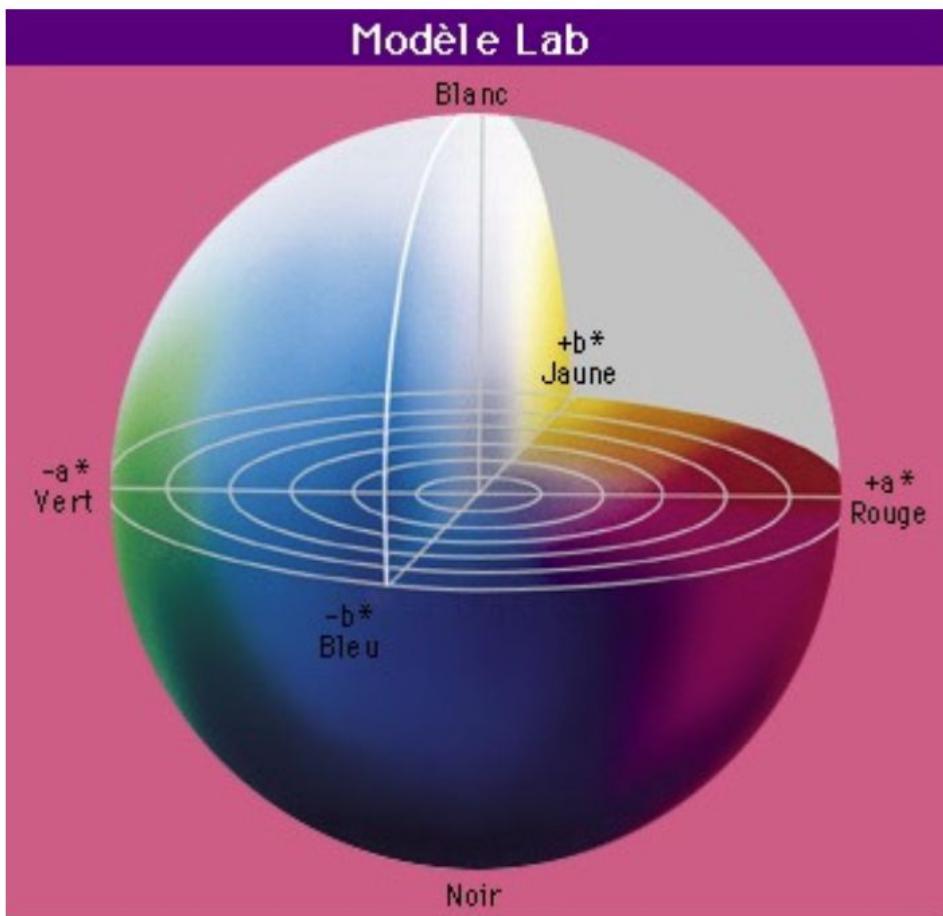
**G**

**B**

# Không gian màu CIE Luv/Lab

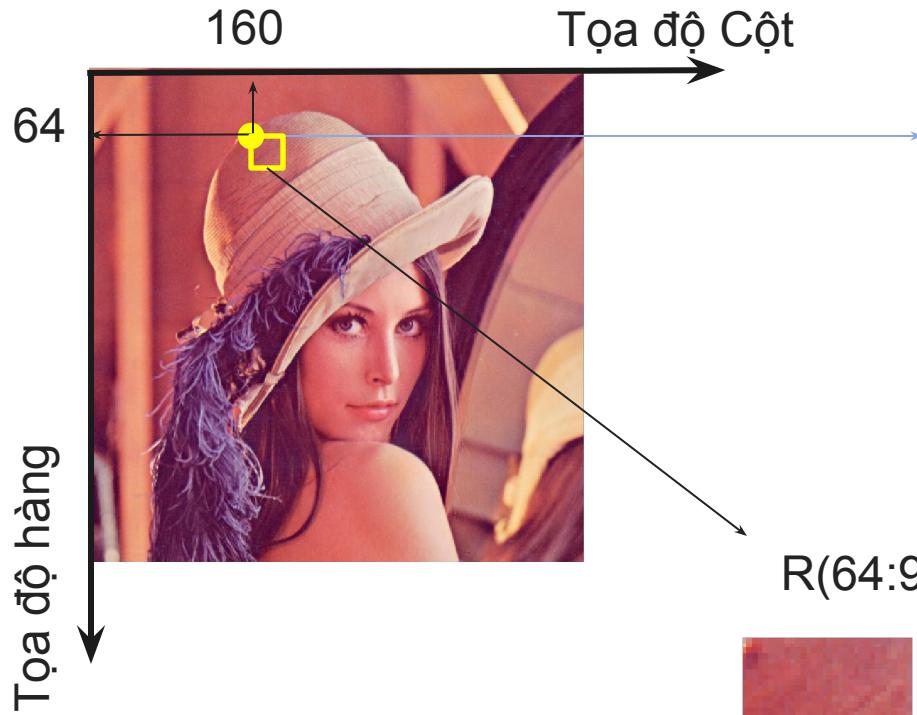
- **Lab** (thi thoảng gọi  $L^*a^*b^*$ ) dựa trên một nghiên cứu về thị giác người
  - Độc lập với tất cả các công nghệ
  - Thể hiện màu sắc như mắt người nhìn thấy
- Màu được xác định bởi 3 giá trị
  - L (luminance) – độ sáng: từ 0% (black) đến 100% (white)
  - $a^*$  biểu diễn trực màu từ màu xanh lá (negative value, -127) tới màu đỏ (positive value, +127)
  - $b^*$  biểu diễn trực màu từ xanh dương (negative value, -127) tới vàng (positive value, +127)

# Không gian màu CIE Luv/Lab



# Biểu diễn ảnh màu

- ◆ Giá trị màu tại mỗi điểm ảnh, vùng ảnh



$$I = [1:512, 1:512, 1:3]$$
$$p(64, 160, 1:3) = [205, 163, 182]$$

Số hàng

Số cột

Số kênh màu

R(64:95, 160:191, :) : 32 x 32 pixels



[205	203	188	181
186	194	182	188
181	187	191	194
195	192	189	183]

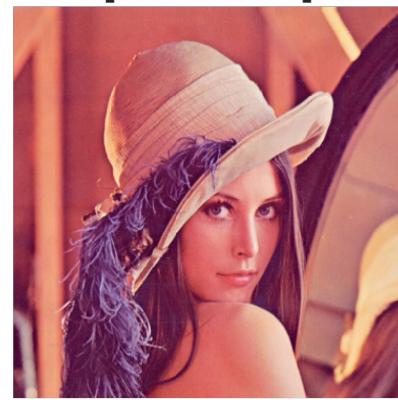
95	92	87	78]
90	98	101	89]

# Thay đổi độ phân giải ảnh

## ◆ Độ phân giải (Resolution)



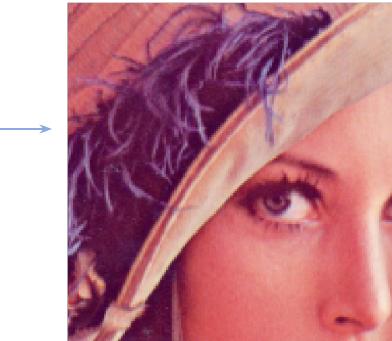
[512 x 512]



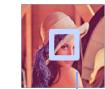
[384 x 384]



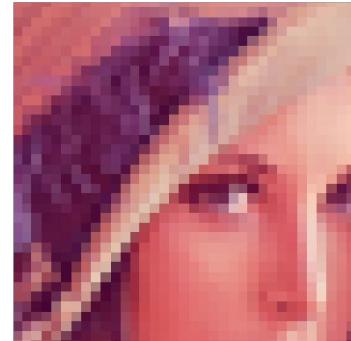
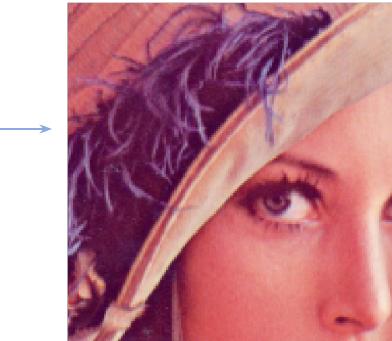
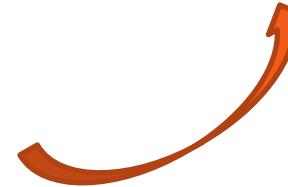
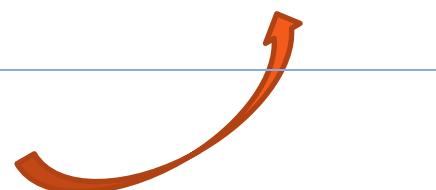
[256 x 256]



[128 x 128]



[64 x 64]



# Thay đổi mức lượng tử hóa

- ◆ 256 gray levels (8bits/pixel)      32 gray levels (5 bits/pixel)      16 gray levels (4 bits/pixel)



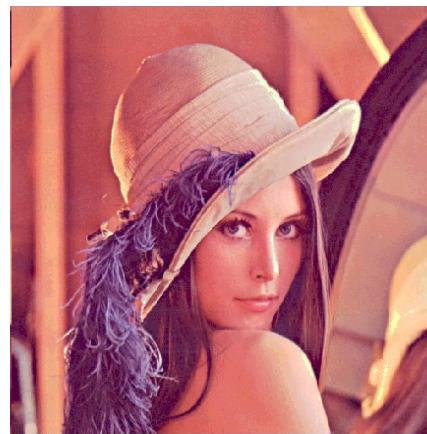
- ◆ 8 gray levels (3 bits/pixel)      4 gray levels (2 bits/pixel)      2 gray levels (1 bit/pixel)



# Thay đổi mức lượng tử hóa



Ảnh gốc



16 mức



12 mức



7 mức

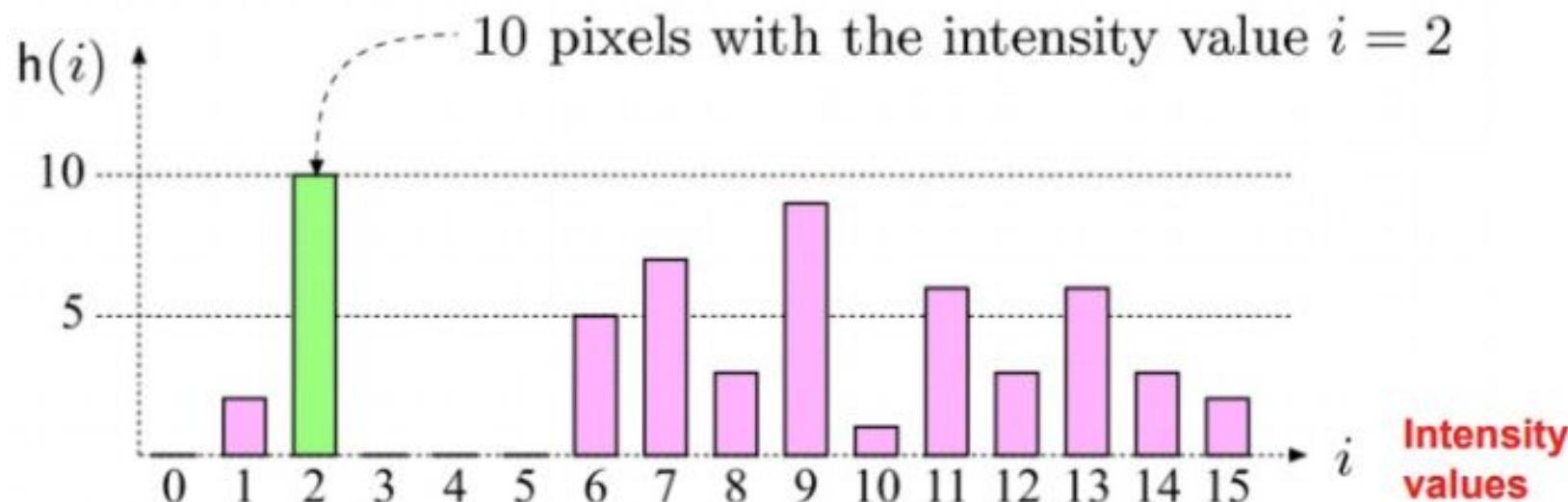
Đặc điểm	Ảnh gốc	16 mức	12 mức	7 mức
Độ phân giải	[512 x 512 x 3]			
Biểu diễn	24 bits	24 bits	24 bits	24 bits
Kích thước	768 K	775 K	769 K	760 K
Số lượng màu	148279	848	468	190

# Một số toán tử xử lý ảnh cơ bản

- Lược đồ ảnh, cân bằng sáng
- Bộ lọc, tiền xử lý và khử nhiễu
- Một số phương pháp nhị phân hóa ảnh
- Chuyển đổi không gian màu

# Lược đồ xám của ảnh (Image histogram)

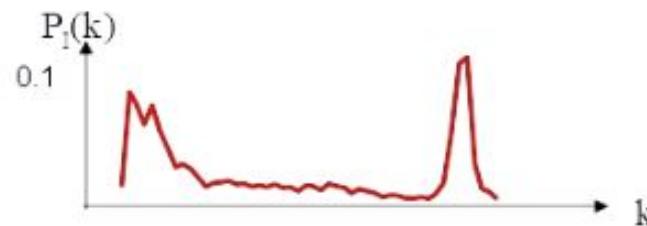
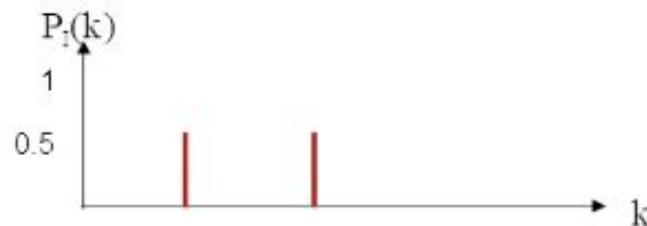
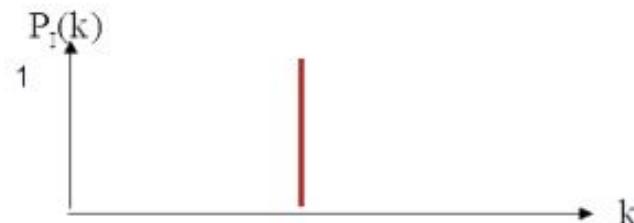
- Là biểu diễn đồ thị sự phân bố màu sắc của các điểm ảnh ảnh số



$h(i)$	0	2	10	0	0	0	5	7	3	9	1	6	3	6	3	2
$i$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

# Lược đồ xám của ảnh (Image histogram)

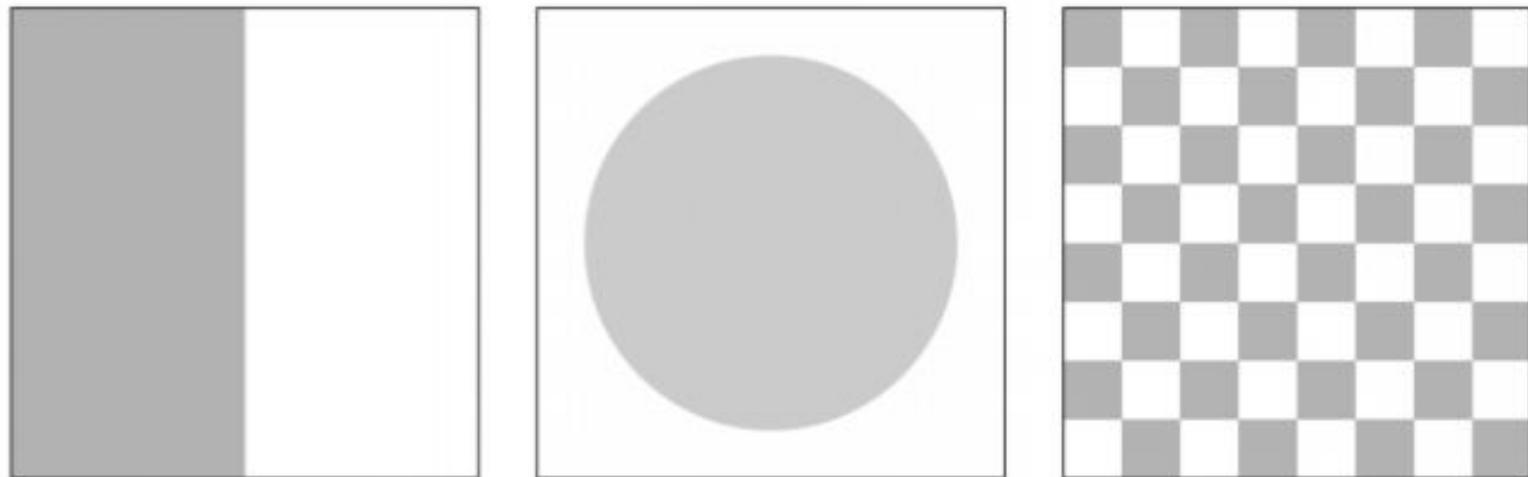
- Histogram
  - Phải chuẩn hóa bằng cách chia cho tổng số điểm ảnh trên ảnh



*Image dynamic range = [min\_value, max\_value]*

# Lược đồ xám của ảnh (Image histogram)

- Histogram
  - Chỉ thông tin thống kê
  - Không có thông tin về mặt không gian của các điểm ảnh
  - Ảnh khác nhau có thể có histogram giống nhau



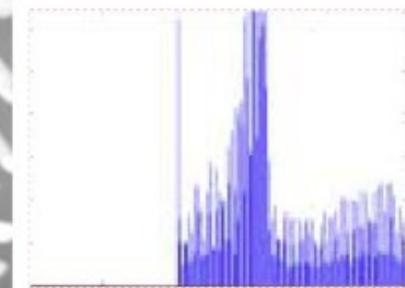
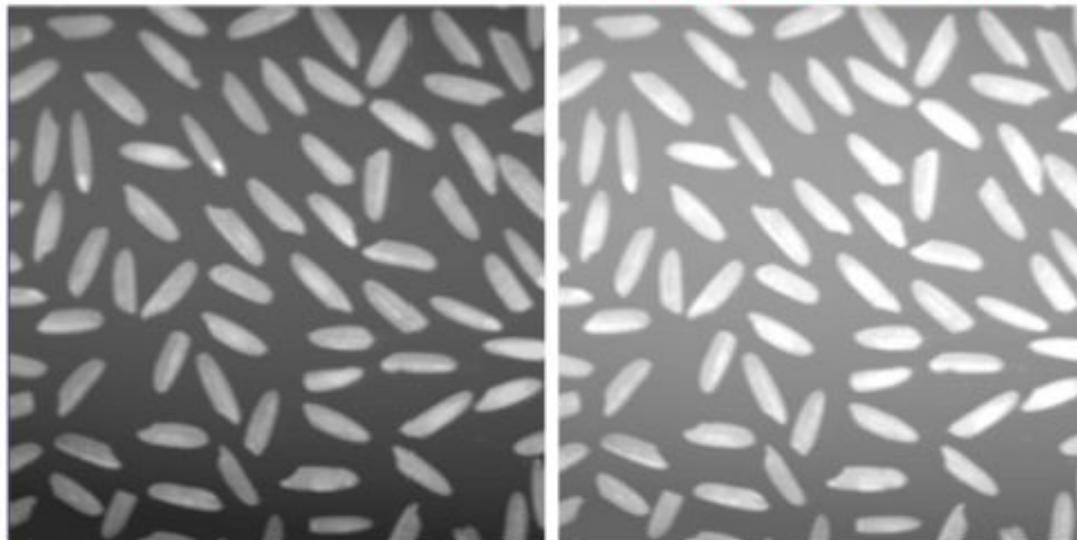
# Độ sáng (Brightness)

- Là giá trị trung bình cường độ sáng trung bình của tất cả các điểm ảnh trên ảnh: độ sáng/tối của ảnh

$$B(I) = \frac{1}{wh} \sum_{v=1}^h \sum_{u=1}^w I(u, v)$$

Divide by total number of pixels

Sum up all pixel intensities



# Độ sáng (Brightness)

- Độ tương phản của ảnh số thể hiện mức độ dễ dàng phân biệt của đối tượng trong ảnh
- Một số cách tính:
  - Độ lệch chuẩn các giá trị điểm ảnh

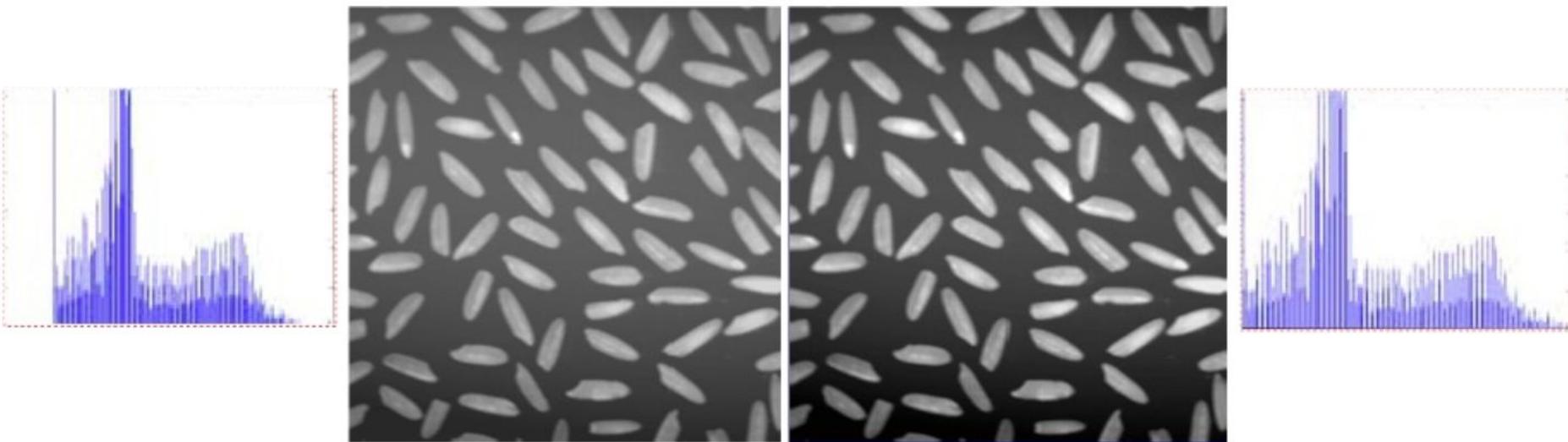
$$C = \sqrt{\frac{1}{M \times N} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (f(x,y) - Moy)^2}$$

- Khác biệt giữa giá trị lớn nhất và nhỏ nhất của điểm ảnh trên ảnh

$$C = \frac{\max[f(x, y)] - \min[f(x, y)]}{\max[f(x, y)] + \min[f(x, y)]}$$

# Độ tương phản

## Contrast vs histogram



# Tăng cường độ tương phản

- Thay đổi giá trị điểm ảnh để có độ tương phản cao hơn
- Một số phương pháp:
  - Kéo giãn dải động ảnh (Linear stretching of intensity range):
    - Linear transform
    - Linear transform with saturation
    - Piecewise linear transform
- Biến đổi phi tuyến. VD: Gamma correction
- Cân bằng histogram

# Giới thiệu về lọc ảnh

- **Ý nghĩa việc lọc ảnh (Image Filtering)**

- Hình thành ảnh mới sao cho những giá trị điểm ảnh đạt được hiệu ứng nào đó từ ảnh gốc
- Nhiều thông tin hữu ích sẽ được thu nhận thông qua quá trình lọc ảnh như:
  - Làm nổi bật các đặc trưng trên ảnh: biên, góc, hình khối
  - Cải thiện/tăng cường chất lượng ảnh, khử nhiễu trong ảnh
  - Tạo hiệu ứng: ảnh độ phân giải cao, sửa ảnh (in-painting)

# Giới thiệu về lọc ảnh

De-noising



Salt and pepper noise

Super-resolution



In-painting



Một số minh họa “hiệu ứng” của lọc ảnh

# Quá trình lọc ảnh

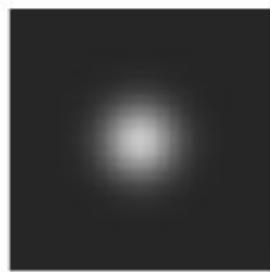
- ◆ Thực hiện trong miền không gian:
  - Lọc ảnh là quá trình thực hiện toán tử trong một lưới (cửa sổ) theo không gian các điểm ảnh
  - Ứng dụng: làm mờ, làm chi tiết hóa, đo thông số về cấu trúc (texture), etc.
- ◆ Thực hiện trong miền tần số:
  - Lọc ảnh là quá trình biến đổi thành phần tần số trong ảnh
  - Ứng dụng: khử nhiễu, lấy mẫu, nén ảnh
- ◆ Bản chất của quá trình lọc ảnh là **tính tích chập**
  - Tích chập trong miền không gian là tích (vô hướng) trong miền không tần số

# Tích chập (Convolution)



Original image

\*



=



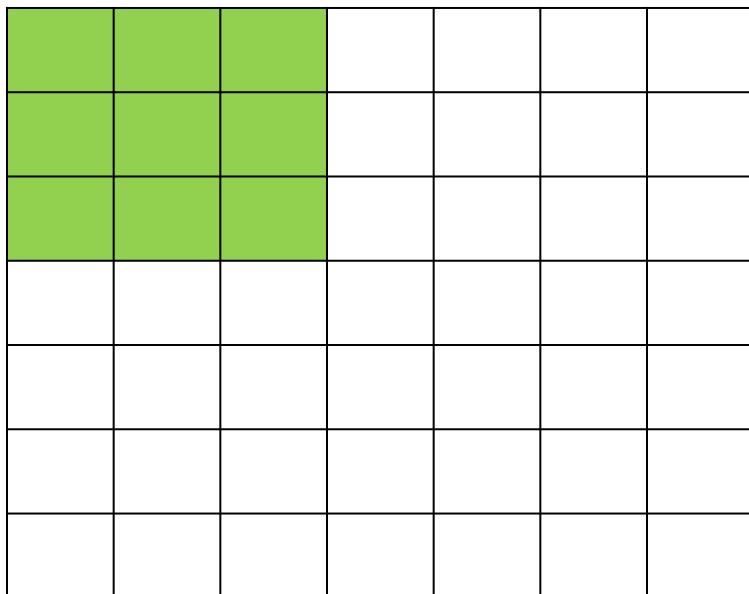
Mask (kernel)

Filtered image

# Lọc ảnh bằng tích chập 2D

2D convolution dịch cửa sổ theo cả hai chiều (x,y)

$$f[n, m] * h[n, m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k, l] h[n - k, m - l]$$

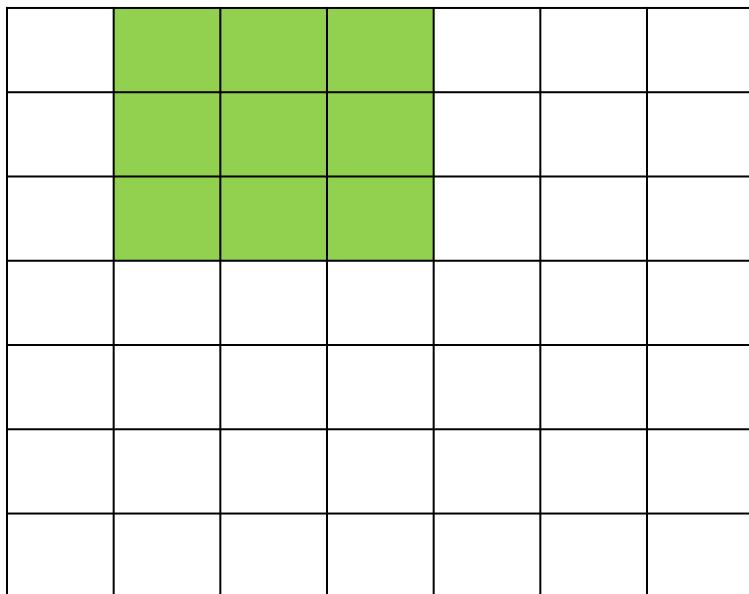


Giả định rằng ta có một filter( $h[.]$ ) có size là  $3 \times 3$ . và một hình ảnh ( $f[.]$ ) có size  $7 \times 7$ .

# Lọc ảnh bằng tích chập 2D

- 2D convolution dịch cửa sổ theo cả hai chiều (x,y)

$$f[n, m] * h[n, m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k, l] h[n - k, m - l]$$

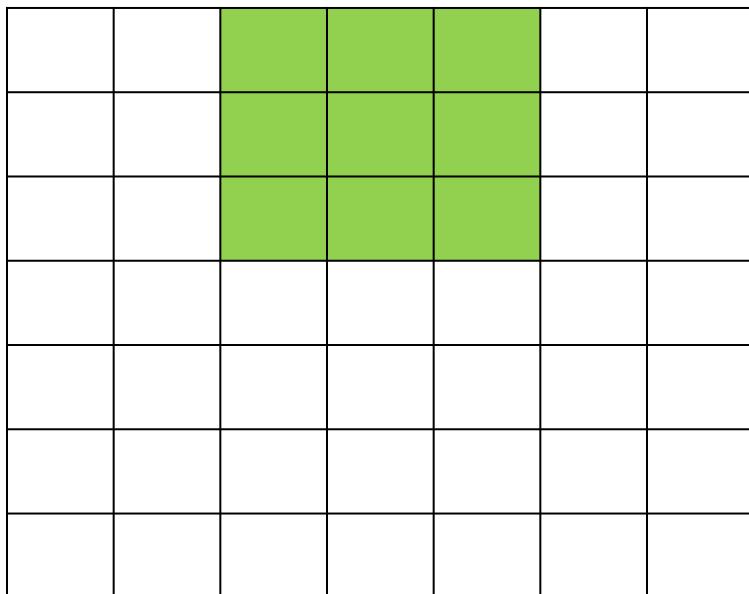


Giả định rằng ta có một filter( $h[.]$ ) có size là  $3 \times 3$ . và một hình ảnh ( $f[.]$ ) có size  $7 \times 7$ .

# Lọc ảnh bằng tích chập 2D

- 2D convolution dịch cửa sổ theo cả hai chiều (x,y)

$$f[n, m] * h[n, m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k, l] h[n - k, m - l]$$

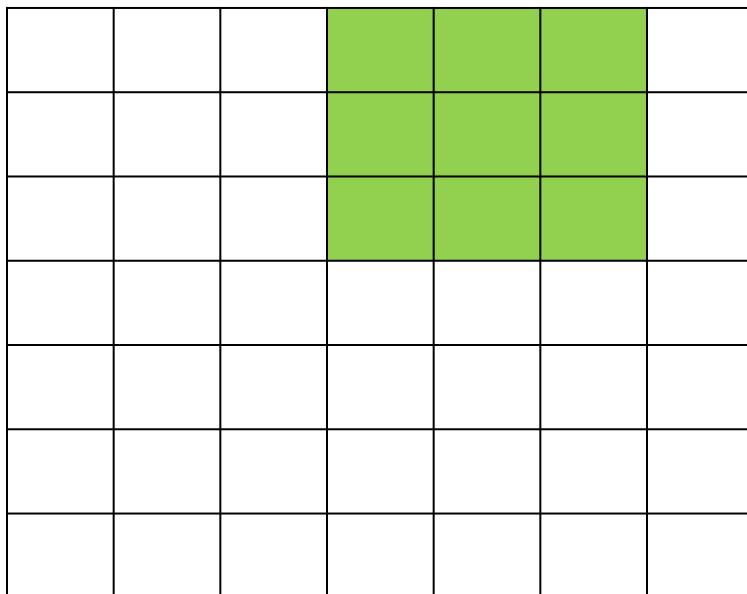


Giả định rằng ta có một filter( $h[.]$ ) có size là  $3 \times 3$ . và một hình ảnh ( $f[.]$ ) có size  $7 \times 7$ .

# Lọc ảnh bằng tích chập 2D

- 2D convolution dịch cửa sổ theo cả hai chiều (x,y)

$$f[n, m] * h[n, m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k, l] h[n - k, m - l]$$

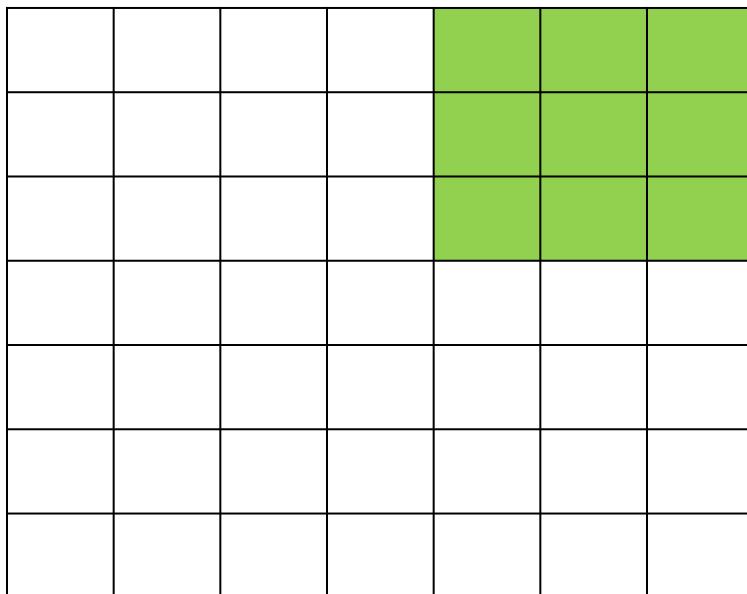


Giả định rằng ta có một filter( $h[.]$ ) có size là  $3 \times 3$ . và một hình ảnh ( $f[.]$ ) có size  $7 \times 7$ .

# Lọc ảnh bằng tích chập 2D

- 2D convolution dịch cửa sổ theo cả hai chiều (x,y)

$$f[n, m] * h[n, m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k, l] h[n - k, m - l]$$

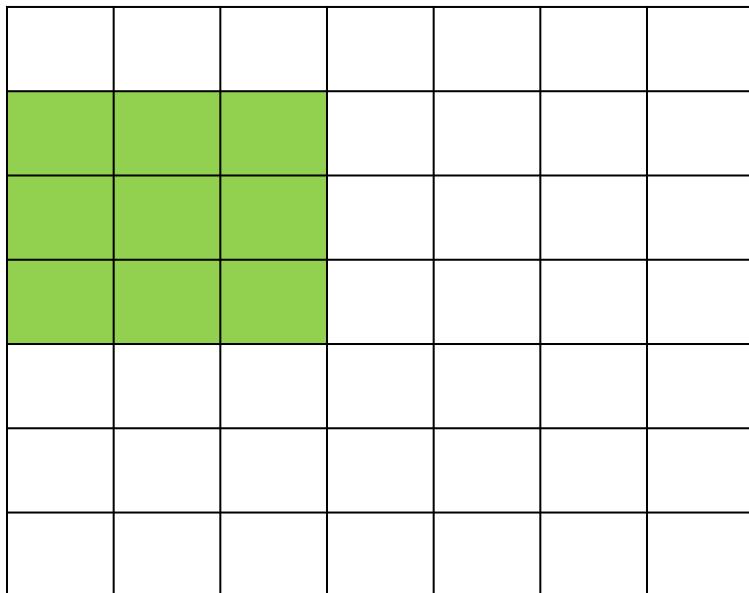


Giả định rằng ta có một filter( $h[.]$ ) có size là  $3 \times 3$ . và một hình ảnh ( $f[.]$ ) có size  $7 \times 7$ .

# Lọc ảnh bằng tích chập 2D

- 2D convolution dịch cửa sổ theo cả hai chiều (x,y)

$$f[n, m] * h[n, m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} f[k, l] h[n - k, m - l]$$



Giả định rằng ta có một filter( $h[.]$ ) có size là  $3 \times 3$ . và một hình ảnh ( $f[.]$ ) có size  $7 \times 7$ .

## Ví dụ về tích chập 2-D

1	2	1		
0	0	0	3	
-1	-2	-1	6	
4	5	7	8	9

$$\begin{aligned}y[0,0] &= x[-1,-1] \cdot h[1,1] + x[0,-1] \cdot h[0,1] + x[1,-1] \cdot h[-1,1] \\&\quad + x[-1,0] \cdot h[1,0] + x[0,0] \cdot h[0,0] + x[1,0] \cdot h[-1,0] \\&\quad + x[-1,1] \cdot h[1,-1] + x[0,1] \cdot h[0,-1] + x[1,1] \cdot h[-1,-1] \\&= 0 \cdot 1 + 0 \cdot 2 + 0 \cdot 1 + 0 \cdot 0 + 1 \cdot 0 + 2 \cdot 0 + 0 \cdot (-1) + 4 \cdot (-2) + 5 \cdot (-1) = -13\end{aligned}$$

-13	-20	-17
-18	-24	-18
13	20	17

Output

## Ví dụ về tích chập 2-D

1	2	1
0	0	0
1	2	3
-1	-2	-1
4	5	6
7	8	9

$$\begin{aligned}y[1,0] &= x[0,-1] \cdot h[1,1] + x[1,-1] \cdot h[0,1] + x[2,-1] \cdot h[-1,1] \\&\quad + x[0,0] \cdot h[1,0] + x[1,0] \cdot h[0,0] + x[2,0] \cdot h[-1,0] \\&\quad + x[0,1] \cdot h[1,-1] + x[1,1] \cdot h[0,-1] + x[2,1] \cdot h[-1,-1] \\&= 0 \cdot 1 + 0 \cdot 2 + 0 \cdot 1 + 1 \cdot 0 + 2 \cdot 0 + 3 \cdot 0 + 4 \cdot (-1) + 5 \cdot (-2) + 6 \cdot (-1) = -20\end{aligned}$$

-13	-20	-17
-18	-24	-18
13	20	17

Output

## Ví dụ về tích chập 2-D

	1	2	1
1	0	0	0
4	-1	-2	-1
7	8	9	

$$\begin{aligned}&= x[1,-1] \cdot h[1,1] + x[2,-1] \cdot h[0,1] + x[3,-1] \cdot h[-1,1] \\&\quad + x[1,0] \cdot h[1,0] + x[2,0] \cdot h[0,0] + x[3,0] \cdot h[-1,0] \\&\quad + x[1,1] \cdot h[1,-1] + x[2,1] \cdot h[0,-1] + x[3,1] \cdot h[-1,-1] \\&= 0 \cdot 1 + 0 \cdot 2 + 0 \cdot 1 + 2 \cdot 0 + 3 \cdot 0 + 0 \cdot 0 + 5 \cdot (-1) + 6 \cdot (-2) + 0 \cdot (-1) = -17\end{aligned}$$

-13	-20	-17
-18	-24	-18
13	20	17

Output

## Ví dụ về tích chập 2-D

1	2	1		3
0	0	0		6
-1	-2	-1		9

$$\begin{aligned}y[0,1] &= x[-1,0] \cdot h[1,1] + x[0,0] \cdot h[0,1] + x[1,0] \cdot h[-1,1] \\&\quad + x[-1,1] \cdot h[1,0] + x[0,1] \cdot h[0,0] + x[1,1] \cdot h[-1,0] \\&\quad + x[-1,2] \cdot h[1,-1] + x[0,2] \cdot h[0,-1] + x[1,2] \cdot h[-1,-1] \\&= 0 \cdot 1 + 1 \cdot 2 + 2 \cdot 1 + 0 \cdot 0 + 4 \cdot 0 + 5 \cdot 0 + 0 \cdot (-1) + 7 \cdot (-2) + 8 \cdot (-1) = -18\end{aligned}$$

-13	-20	-17
-18	-24	-18
13	20	17

Output

## Ví dụ về tích chập 2-D

1	2	1
1	2	3
0	0	0
4	5	6

$$\begin{aligned}y[1,1] &= x[0,0] \cdot h[1,1] + x[1,0] \cdot h[0,1] + x[2,0] \cdot h[-1,1] \\&\quad + x[0,1] \cdot h[1,0] + x[1,1] \cdot h[0,0] + x[2,1] \cdot h[-1,0] \\&\quad + x[0,2] \cdot h[1,-1] + x[1,2] \cdot h[0,-1] + x[2,2] \cdot h[-1,-1] \\&= 1 \cdot 1 + 2 \cdot 2 + 3 \cdot 1 + 4 \cdot 0 + 5 \cdot 0 + 6 \cdot 0 + 7 \cdot (-1) + 8 \cdot (-2) + 9 \cdot (-1) = -24\end{aligned}$$

-13	-20	-17
-18	-24	-18
13	20	17

Output

## Ví dụ về tích chập 2-D

1	1	2	3	1
4	0	5	0	6
7	-1	8	-2	9

$$\begin{aligned} &= x[1,0] \cdot h[1,1] + x[2,0] \cdot h[0,1] + x[3,0] \cdot h[-1,1] \\ &\quad + x[1,1] \cdot h[1,0] + x[2,1] \cdot h[0,0] + x[3,1] \cdot h[-1,0] \\ &\quad + x[1,2] \cdot h[1,-1] + x[2,2] \cdot h[0,-1] + x[3,2] \cdot h[-1,-1] \\ &= 2 \cdot 1 + 3 \cdot 2 + 0 \cdot 1 + 5 \cdot 0 + 6 \cdot 0 + 0 \cdot 0 + 8 \cdot (-1) + 9 \cdot (-2) + 0 \cdot (-1) = -18 \end{aligned}$$

-13	-20	-17
-18	-24	-18
13	20	17

Output

# 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
  - .....

# Sử dụng filter trung bình

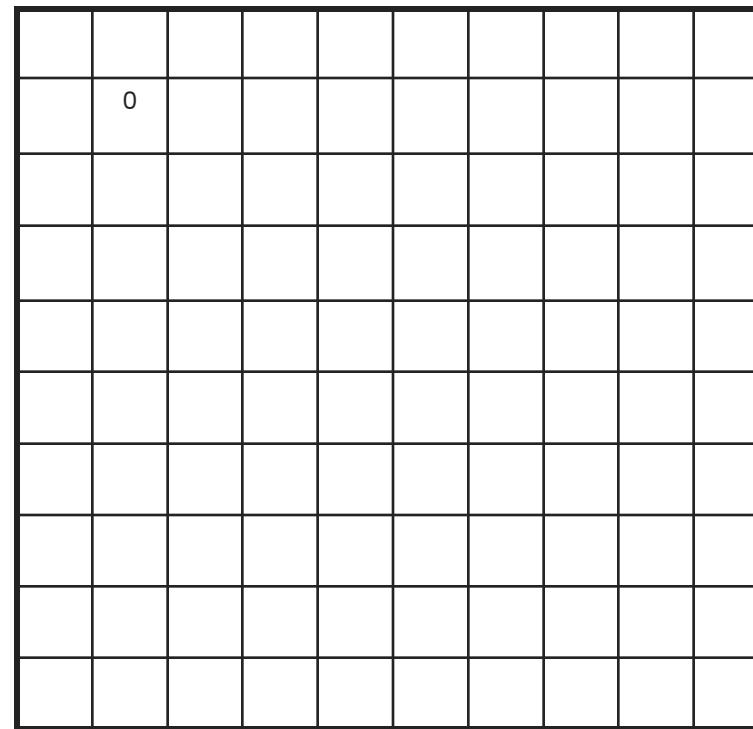
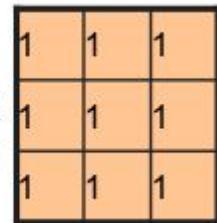
$I[.,.]$

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

$h[.,.]$

$f[.,.]$

$\frac{1}{9}$



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

# Sử dụng filter trung bình

$I[.,.]$

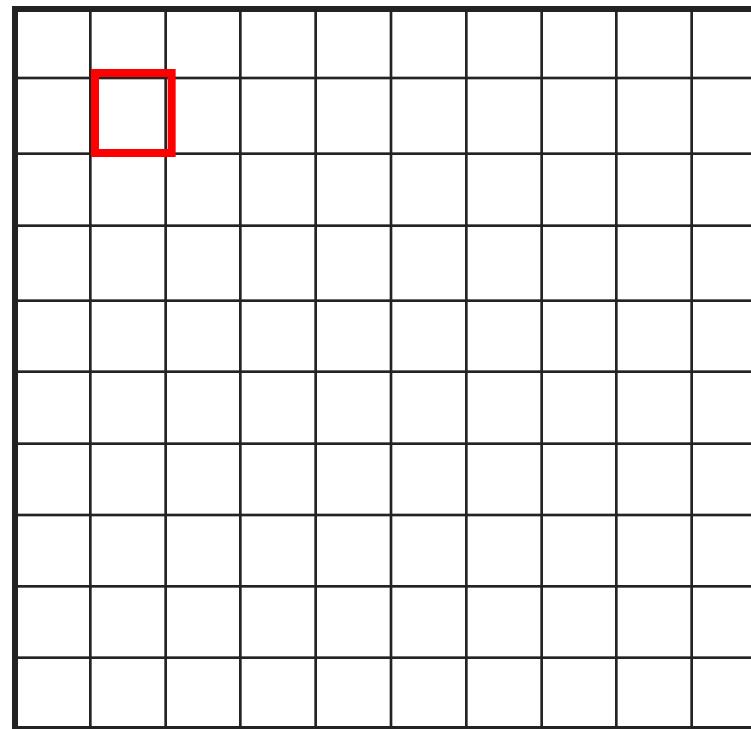
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

$f[.,.]$

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1



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

# Sử dụng filter trung bình

$I[.,.]$

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

$h[.,.]$

$f[.,.]$

$$\frac{1}{9}$$

1	1	1
1	1	1
1	1	1

0	10	20								

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

# Sử dụng filter trung bình

$I[.,.]$

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	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

$h[.,.]$

$f[\cdot, \cdot]$

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


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

# Sử dụng filter trung bình

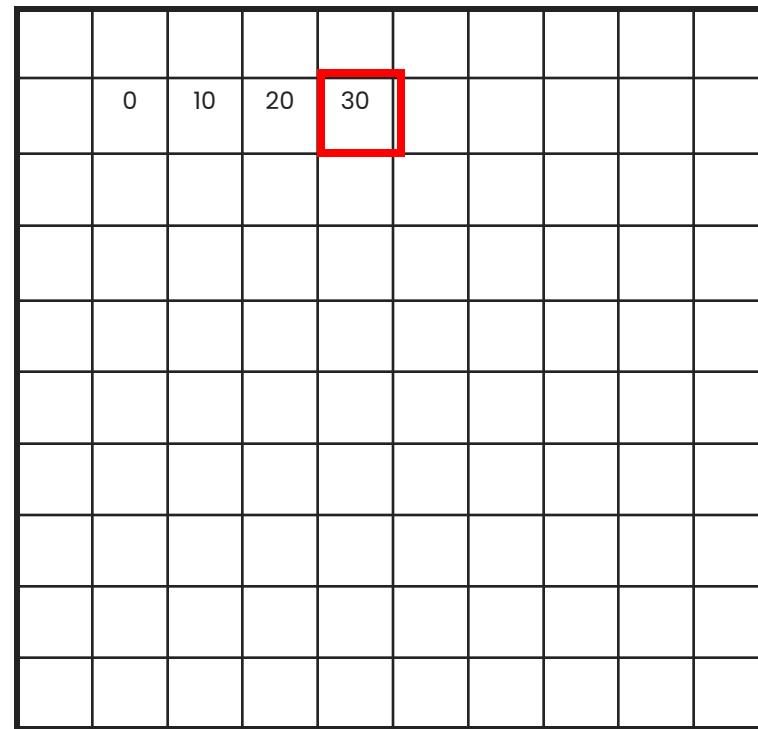
$I[.,.]$

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	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

$h[.,.]$

$f[\cdot, \cdot]$

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



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

# Sử dụng filter trung bình

$I[.,.]$

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

$h[.,.]$

$f[.,.]$

$\frac{1}{9}$

1	1	1
1	1	1
1	1	1

	0	10	20	30	30						

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

# Sử dụng filter trung bình

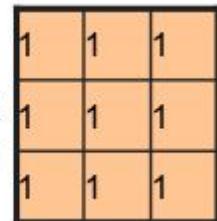
$I[.,.]$

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

$h[.,.]$

$f[.,.]$

$$\frac{1}{9}$$



	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	

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

# Làm mờ bằng bộ lọc trung bình



## Một số ví dụ về lọc ảnh



Original

$$\ast \quad \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} =$$



Filtered (no change)

Dịch sang bên phải 1 pixel



$$\ast \quad \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 0 & 1 \\ \hline 0 & 0 & 0 \\ \hline \end{array} =$$



## Một số ví dụ về lọc ảnh



$$\text{Input Image} * \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \text{Output Image}$$



Làm mờ



$$\text{Input Image} - \frac{1}{9} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \text{Output Image}$$



Áp dụng kết hợp 2 bộ lọc tăng độ chi tiết

# Một số ví dụ về lọc ảnh



Ảnh ban đầu



Ảnh đã được làm mờ



Các chi tiết



Ảnh ban đầu

+ a



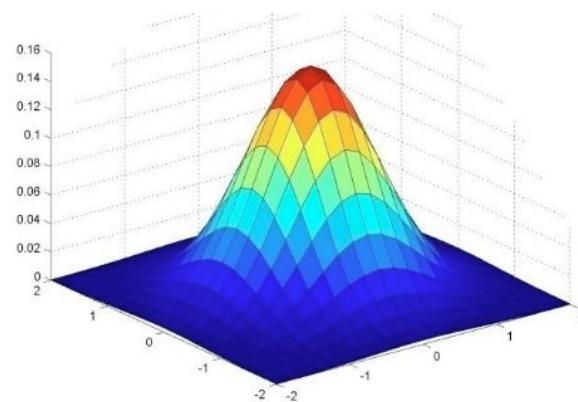
Các chi tiết



Tăng độ sắc nét

# Bộ lọc Gaussian

## Gaussian filter



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

Gaussian filter with size 5 x5 , sigma =1

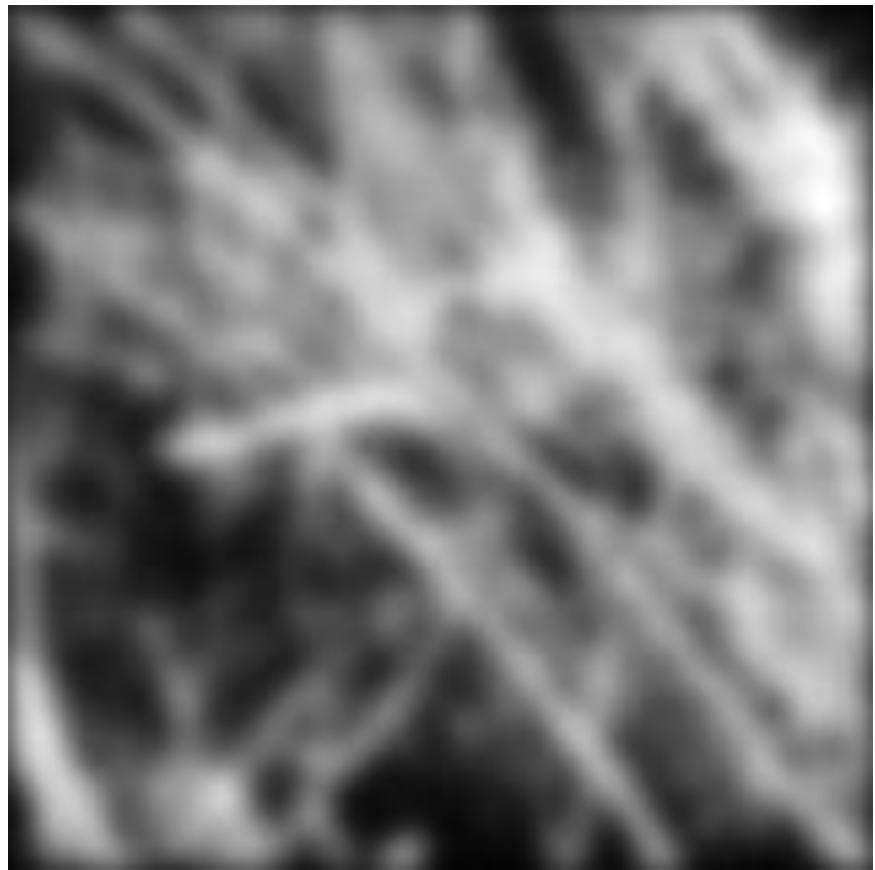


$$G_{\sigma} = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

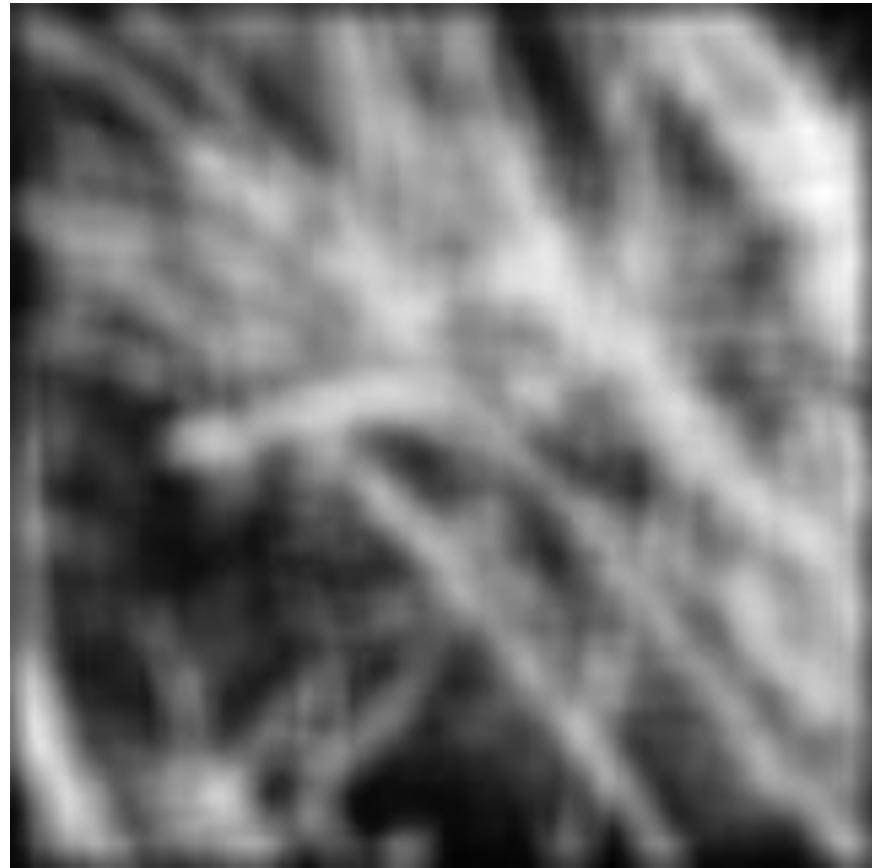
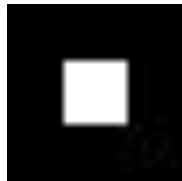
# Bộ lọc Gaussian

- **Bộ lọc thông thấp:** loại bỏ các thành phần tương ứng tần số cao trên ảnh
  - Ảnh trơn hơn
  - Tốt hơn bộ lọc trung bình
- Nhập chập Gauss với chính nó ta được một hàm Gauss
  - Lặp nhận chập với bộ lọc có kích thước nhỏ => thu được kết quả như nhận chập với bộ lọc có kích thước lớn hơn.
  - Nhận chập **2 lần** với bộ lọc Gauss có độ rộng  $\sigma$  giống như nhận chập **1 lần** với bộ lọc có độ rộng  $\sigma\sqrt{2}$ :  $I^*G_{\sigma} * G_{\sigma} = I^*G_{\sigma\sqrt{2}}$
- Bộ lọc có thể phân tách được: Hàm Gauss 2D có thể được biểu diễn như tích của 2 hàm 1D: 1 hàm theo x và 1 hàm theo y:  $G_{\sigma}(x,y) = G_{\sigma}(x) \cdot G_{\sigma}(y)$

# Làm trơn ảnh với bộ lọc Gaussian



# Làm trơn ảnh với bộ lọc Box



# Làm trơn ảnh với bộ lọc Box



Original image

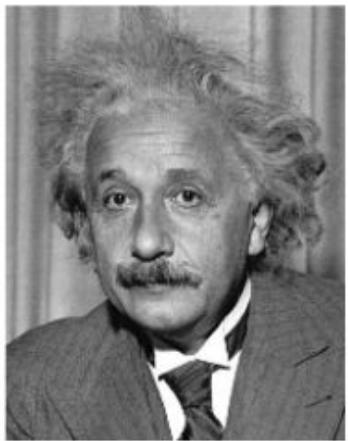


Filtered image  
with box size 5x5



Filtered image  
with box size 11x11

# Lọc ảnh với bộ lọc Sobel (tính đạo hàm)



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



Vertical Edge

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



Horizontal Edge

# Nhị phân hóa ảnh

Chuyển từ ảnh màu thành ảnh xám:  $value = 0.2126*R + 0.7152*G + 0.0722*B$



Ảnh mức xám  
Gray scale



Ảnh Màu RGB



Ảnh trắng/đen  
BW

- Chuyển từ ảnh xám  ảnh nhị phân (trắng/đen): quá trình nhị phân hóa (phân đoạn ảnh)
- 3 Phương pháp
  - Tách ngưỡng xác định trước
  - Bó cụm
  - Tìm ngưỡng tự động

# Tách ngưỡng cố định

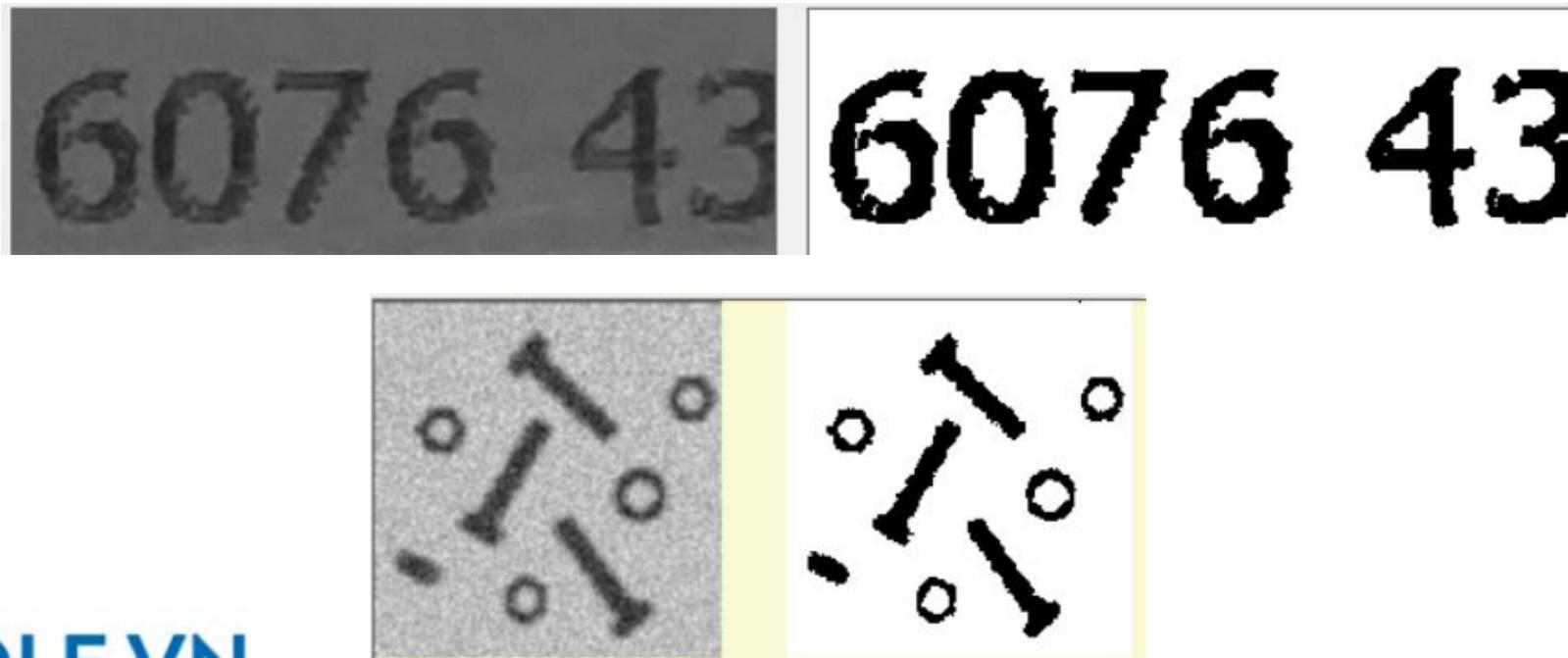
- Giả sử ảnh I có kích thước  $m \times n$ , hai số Min, Max và ngưỡng  $\theta$  khi đó: Kỹ thuật tách ngưỡng được thể hiện

```
for (i = 0; i < m; i + +)  
    for (j = 0; j < n; j + +)  
        I [i, j] = I [i, j] > =  $\theta$ ? Max : Min;
```

- Ứng dụng:
  - Nếu Min = 0, Max = 1 kỹ thuật chuyển ảnh thành ảnh đen trắng được ứng dụng khi quét và nhận dạng văn bản.
  - Có thể xảy ra sai sót nền thành ảnh hoặc ảnh thành nền dẫn đến ảnh bị đứt nét hoặc dính.

## Tách ngưỡng cố định

- Ngưỡng cố định được xác định bởi:
  - Tìm được do thống kê
  - Do người dùng cung cấp
- Ví dụ:



# Tổng kết buổi học

- Các khái niệm, quá trình hình thành ảnh số
- Các kỹ thuật phân tích, xử lý hình ảnh

# THANK YOU !

**COLE.VN**  
Connecting knowledge



[www.cole.vn](http://www.cole.vn)