



ONE LOVE. ONE FUTURE.



ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Nhập môn **Khoa học dữ liệu** (IT4142)

PGS.TS Thân Quang Khoát & PGS.TS Phạm Văn Hải
Team lecturers

ONE LOVE. ONE FUTURE.

Contents

- Lecture 1: Tổng quan về Khoa học dữ liệu
- Lecture 2: Thu thập và tiền xử lý dữ liệu
- Lecture 3: Làm sạch và tích hợp dữ liệu
- Lecture 4: Phân tích và khám phá dữ liệu
- Lecture 5: Trực quan hóa dữ liệu
- Lecture 6: Trực quan hóa dữ liệu đa biến
- Lecture 7: Học máy
- Lecture 8: Phân tích dữ liệu lớn
- Lecture 9: Báo cáo tiến độ bài tập lớn và hướng dẫn
- **Lecture 10+11: Phân tích một số kiểu dữ liệu**
- Lecture 12: Đánh giá kết quả phân tích



3

Nội dung

- Thị giác máy tính và một số ứng dụng
- Khái niệm cơ bản về ảnh số
 - Ảnh số, lược đồ xám, độ sáng độ tương phản, màu,...
 - Thư viện : Opencv
- Nhận dạng và một số bộ lọc cơ bản
 - lọc nhiễu
 - phát hiện biên
- Biểu diễn nội dung ảnh / trích chọn đặc trưng: đặc trưng cục bộ và đặc trưng toàn cục



4

Thị giác máy tính ?

- Xử lý ảnh
 - Làm việc trên ảnh như một ma trận số
 - Đầu vào: ảnh số → 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



What kind of scene?

Where are the cars?

How far is the building?

...



5

Ứng dụng của thị giác máy tính

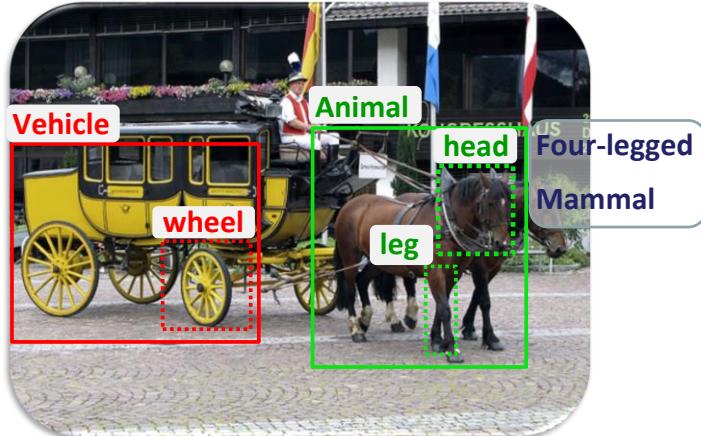
- Ảnh, video:
 - Nguồn dữ liệu dồi dào, môi trường đa dạng
 - Giàu thông tin
- Lĩnh vực thu hút nhiều quan tâm đặc biệt trong kỷ nguyên mới ..



6

Ứng dụng của thị giác máy tính

- Hiểu nội dung ảnh



7

Ứng dụng của thị giác máy tính

- Hiểu nội dung bức ảnh

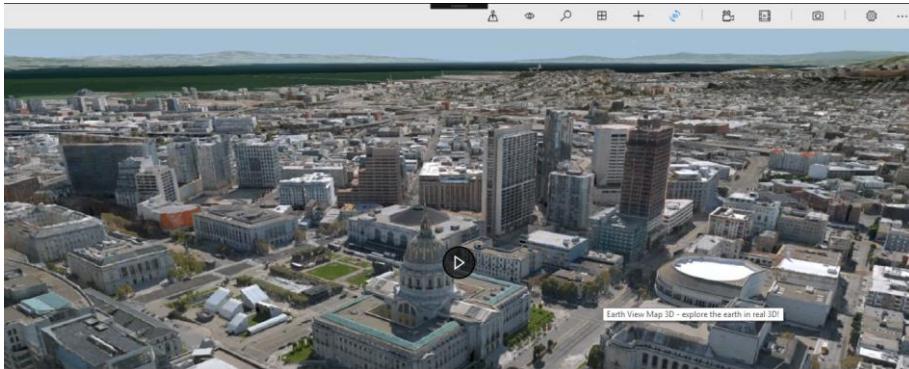


8



Ứng dụng của thị giác máy tính

- Earth View, Google earth (mô hình 3D từ nhiều ảnh 2D): mô hình được sinh tự động + các công trình tiêu biểu cần chi tiết: mô hình được thiết kế thủ công (Golden Gate bridge or Sydney Opera house)

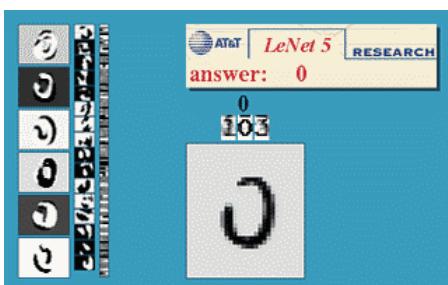


Microsoft's Virtual Earth

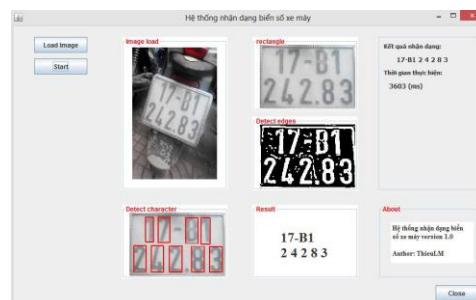
9

Ứng dụng của thị giác máy tính

- OCR (Optical character recognition)
 - Technology to convert scanned docs to text: each scanner came with an OCR software



Digit recognition, AT&T labs
<http://yann.lecun.com/exdb/lenet/>



Licence plate detection and character recognition

10



Ứng dụng của 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



Source: Derek Hoiem, Computer vision, CS 543 / ECE 549, University of Illinois



11

Ứng dụng của thị giác máy tính

- Phát hiện mặt cười: smart camera
 - Camera có thể tự động chọn thời điểm chụp để có bức ảnh có biểu cảm hoàn hảo



Source: Derek Hoiem, Computer vision, CS 543 / ECE 549, University of Illinois



12

Ứng dụng của thị giác máy tính

- Đăng nhập với thông tin sinh trắc học (vân tay, mông mắt, mặt,...)



Bộ quét vân tay được trang bị trên nhiều máy tính cũng như các thiết bị khác

Source: Derek Hoiem, Computer vision, CS 543 / ECE 549, University of Illinois

13

Ứng dụng của thị giác máy tính

- Phát hiện đối tượng (trên mobiles)



[Point & Find, Nokia](#)

[Google Goggles](#)

Source: Derek Hoiem, Computer vision, CS 543 / ECE 549, University of Illinois

14

Ứng dụng của thị giác máy tính

- Truy vấn ảnh dựa trên nội dung

TinEye
Reverse Image Search

22 Results
Searched over 2.1809 billion images in 1.116 seconds.
for file: test_it_213873b.jpg

* These results expire in 72 hours. [Get a new search](#)
• [Report Abuse](#)
• [TinEye is free for non-commercial purposes.](#)

Sort by:
Best Match
Most Changed
Biggest Image

Image	Source	Link
	pa.photobucket.com	pa.photobucket.com/images/
	blogs.telegraph.co.uk	blogs.telegraph.co.uk/steve-jobs/
	tecnologia.ig.com.br	7571361/steve-jobs-225-300.jpg

[View 1 more collection result](#)

Source: Derek Hoiem, Computer vision, CS 543 / ECE 549, University of Illinois

15

Ứng dụng của thị giác máy tính

- Smart cars → autonomous vehicles

Our Vision. Your Safety.

manufacturer products consumer products

rear looking camera forward looking camera side looking camera

EyeQ Vision on a Chip **Vision Applications** Road, Vehicle, Pedestrian Protection and more **AWS** Advance Warning System

News:
/ **Mobileye Advances Technologies Power Volvo Cars World First Collision Warning With Auto Brake System**
/ **Volvo New Collision Warning with Auto Brake Help Prevent Rear-end**
/ **Mobileye at Equip Auto, Paris, France**
/ **Mobileye at SEMA, Las Vegas, NV**

Events:
/ **Mobileye at SEMA, Las Vegas, NV**

Mobileye: vision systems currently in many cars

"In mid 2010 Mobileye will launch a world's first application of full emergency braking for collision mitigation for pedestrians where vision is the key technology for detecting pedestrians"

Source: Derek Hoiem, Computer vision, CS 543 / ECE 549, University of Illinois

16



Ứng dụng của thị giác máy tính

- Ghép ảnh toàn cảnh:



Source: Derek Hoiem, Computer vision, CS 543 / ECE 549, University of Illinois



17

Ứng dụng của thị giác máy tính

- Games / robots:



Vision-based interaction game
(Microsoft's Kinect)



<http://www.robocup.org/>



Robot vacuum cleaner



Source: Derek Hoiem, Computer vision, CS 543 / ECE 549, University of Illinois

18

Ứng dụng của thị giác máy tính

- Tìm hiểu thêm về các ứng dụng và công ty trong lĩnh vực thị giác máy tính, tham khảo trang của D.Lowe:

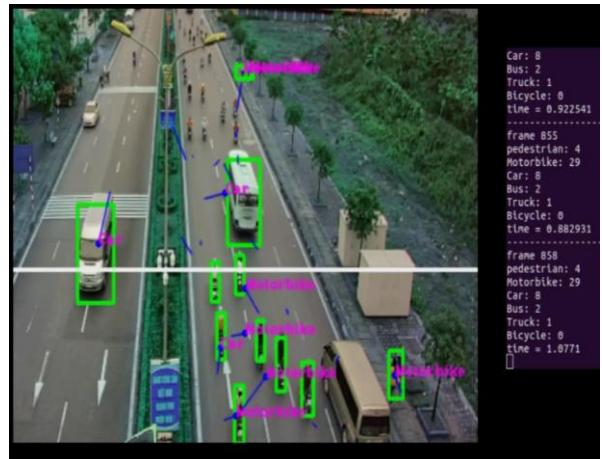
<https://www.cs.ubc.ca/~lowe/vision.html>



19

Một số topics trong CV

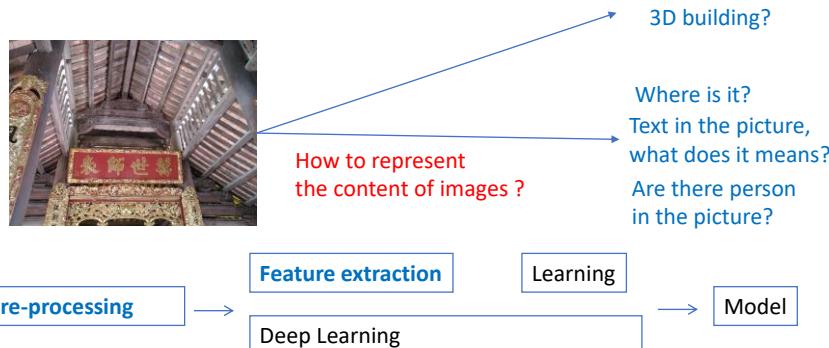
- Camera giám sát:
 - Đếm số lượng khách hàng trong cửa hàng
 - Phát hiện hành động bất thường
 - Đo mức độ hài lòng của khách hàng
 - Object tracking: Someone ran a red light?
- Phát hiện và nhận dạng đối tượng
 - Phát hiện mặt/mắt/người
 - Nhận dạng hoạt động
 - Phát hiện lỗi
- Gán nhãn ảnh số
- Phát hiện nhận dạng ký tự
 - Đọc card visite, CMT, biển số, ...
- Xây dựng đối tượng 3D từ ảnh 2D



20

Nội dung chính sẽ đề cập

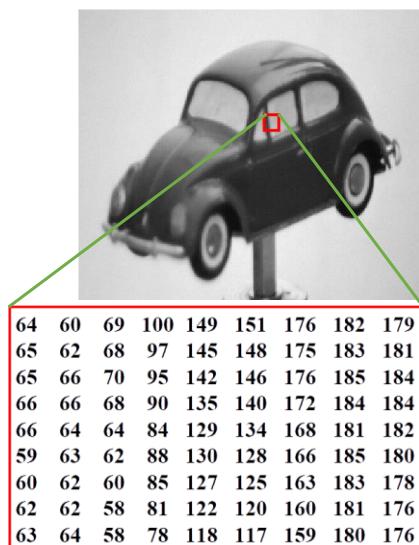
- 2 kiểu thông tin chúng ta trích chọn từ ảnh:
 - Thông tin 3D
 - Thông tin ngữ nghĩa



21

Ảnh số ?

- Con người nhìn thấy gì trong ảnh?
 - Một chiếc oto?
- Máy tính có thể thấy gì?
 - Ảnh là một ma trận các điểm ảnh
 - Ảnh $N \times M$: ma trận $N \times M$
 - 1 điểm ảnh (gray levels):
 - Giá trị cường độ sáng
 - 0-255
 - Đen: 0
 - Trắng: 255

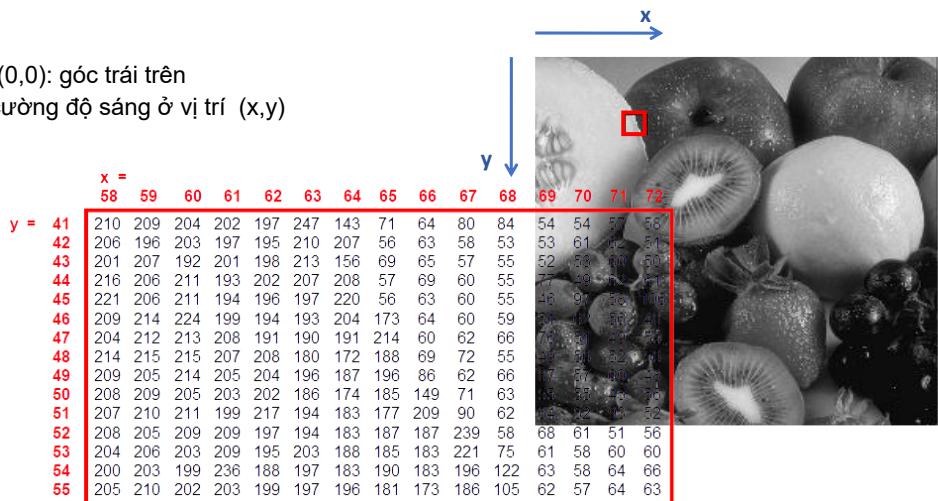


22

Ảnh số ?

- **Ảnh I:**

- Chỉ số (0,0): góc trái trên
- $I(x,y)$: cường độ sáng ở vị trí (x,y)

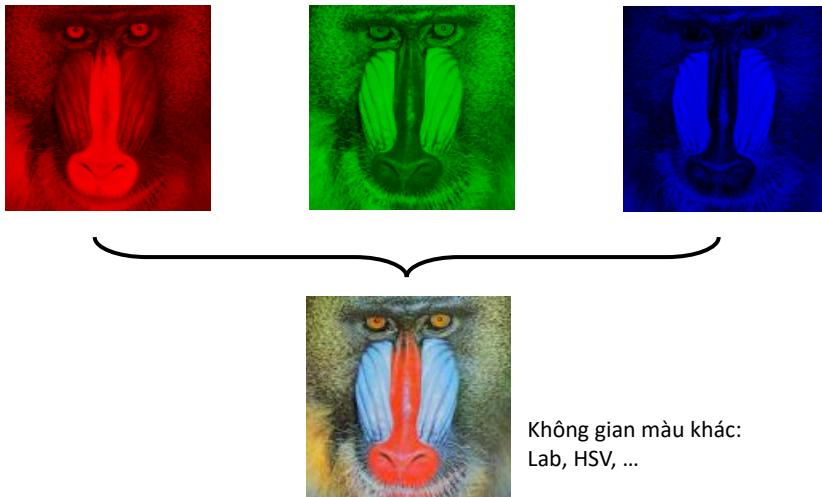


Ảnh số ?

- 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 phẳng, ảnh độ sâu,...



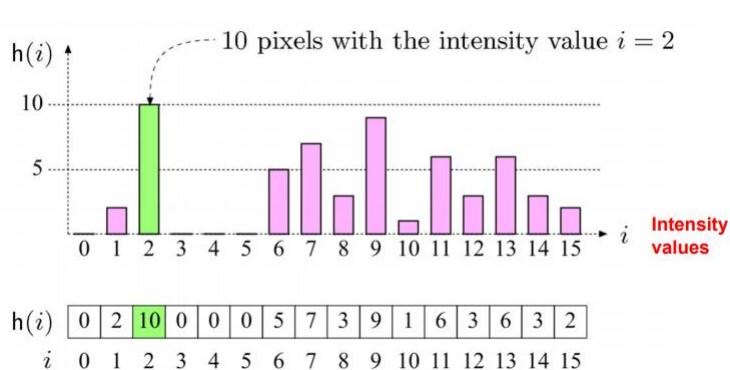
Ảnh màu trong không gian màu RGN



25

Lược đồ xám của ảnh (Image histogram)

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



26

Lược đồ xám của ảnh (Image histogram)

- Histogram

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

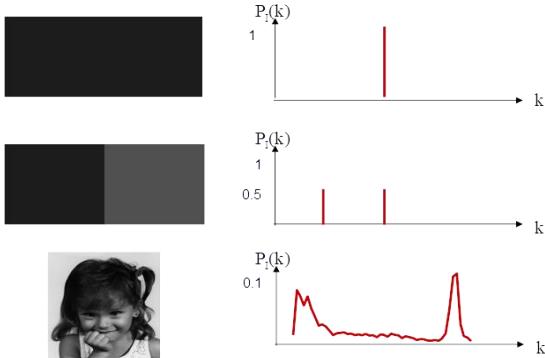


Image dynamic range = [min_value, max_value]

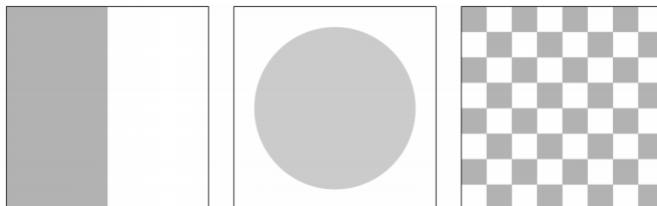


27

Lược đồ xám của ảnh (Image histogram)

- Histogram

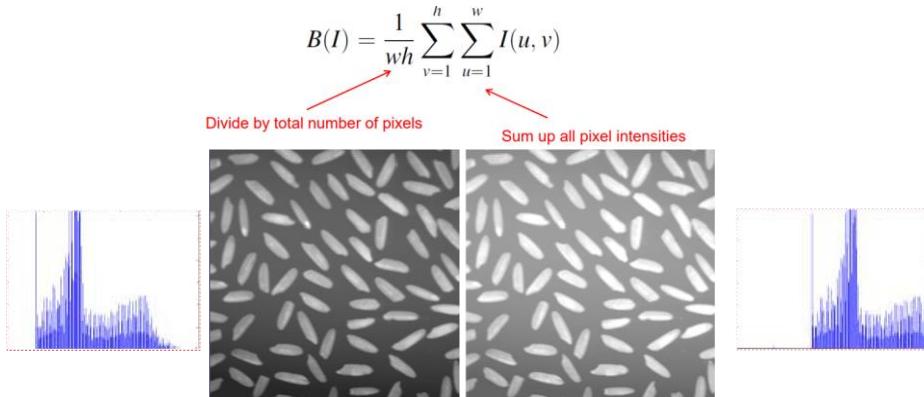
- Chỉ thông tin thống kê
- Không có thông tin về mặt không gian của các điểm ảnh
- Ảnh khác nhau có thể có histogram giống nhau



28

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



Độ tương phản (Contrast)

- 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

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

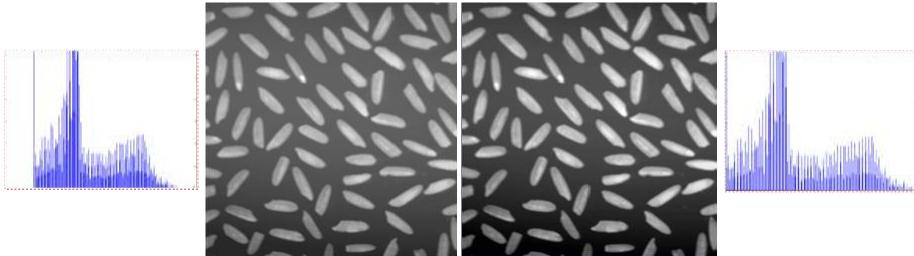
- Khác biệt giữa giá trị lớn nhất và nhỏ nhất của điểm ảnh trên ảnh

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



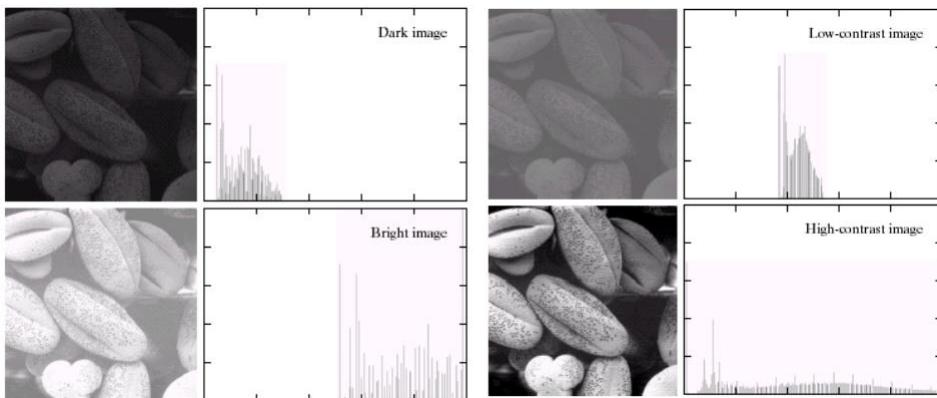
Độ tương phản (Contrast)

- Contrast vs histogram



31

Ví dụ



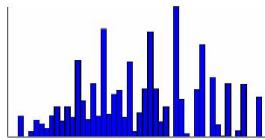
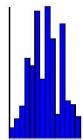
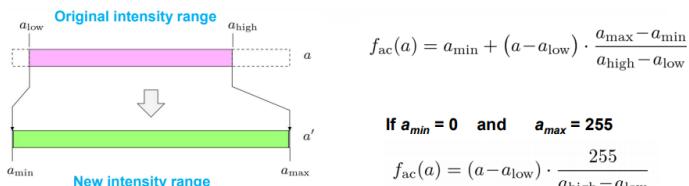
32

Tăng cường độ tương phản

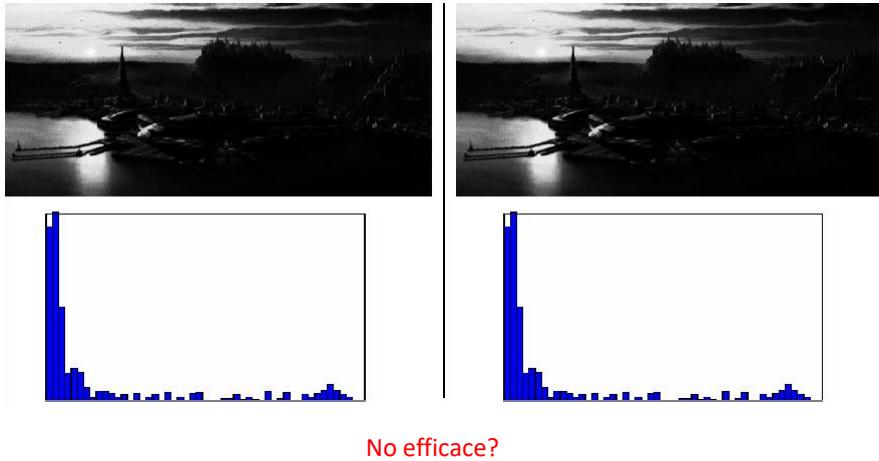
- Thay đổi giá trị điểm ảnh để có độ tương phản cao hơn
- Một số phương pháp:
 - Kéo giãn dài động ảnh (Linear stretching of intensity range):
 - Linear transform
 - Linear transform with saturation
 - Piecewise linear transform
 - Biến đổi phi tuyến. VD: Gama correction
 - Cân bằng histogram



Linear stretching



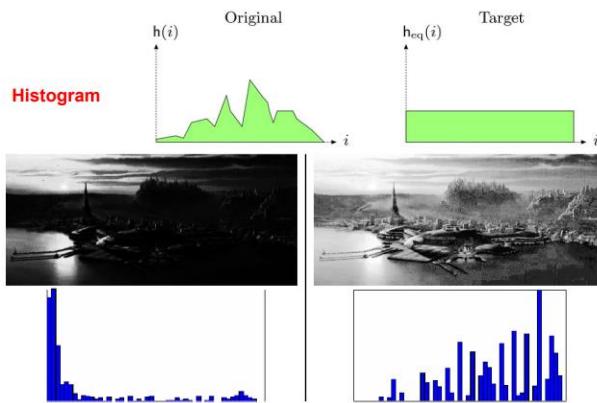
Linear stretching



35

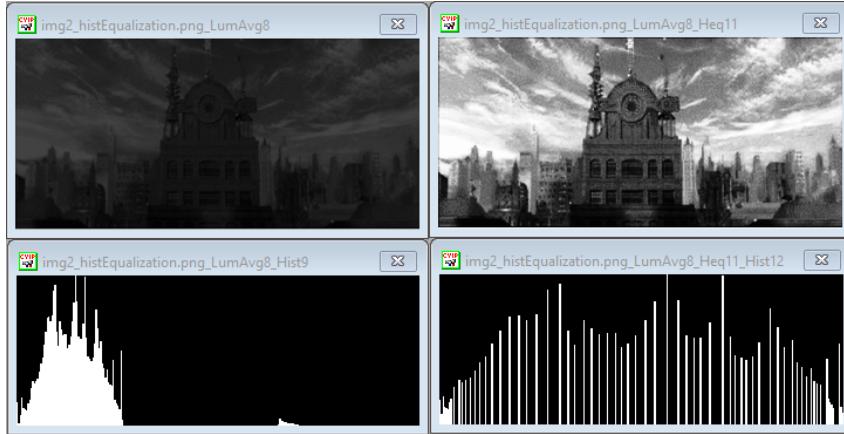
Cân bằng histogram

- Histogram của ảnh sau thay đổi hướng tới phân phối đều
- Không tham số. OpenCV:cv2.equalizeHist(img)



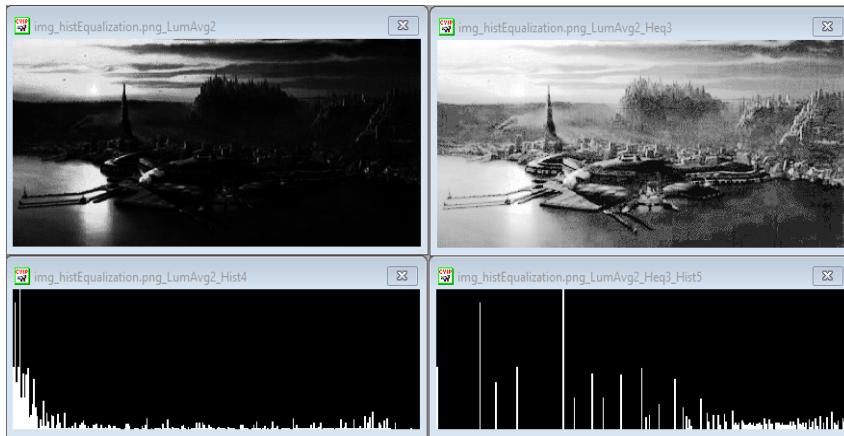
36

Cân bằng histogram



37

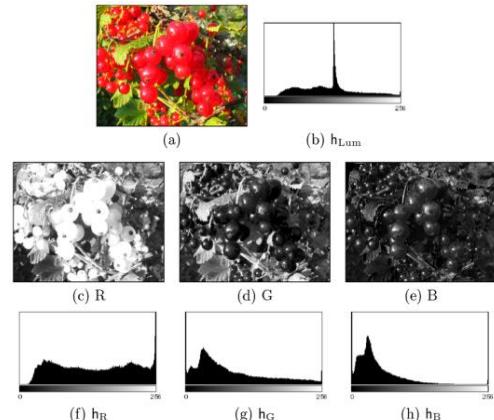
Cân bằng histogram



38

Histogram trên ảnh màu

- Lược đồ xám:**
 - Chuyển ảnh màu sang ảnh xám
 - Tính hist của ảnh xám
- Histogram của các kênh màu riêng:**
3 histograms cho (R,G,B)
- Histogram 3D:**
 - Một màu đc xđ bởi 3 giá trị
 - Không thường đc dụng do kích thước lớn



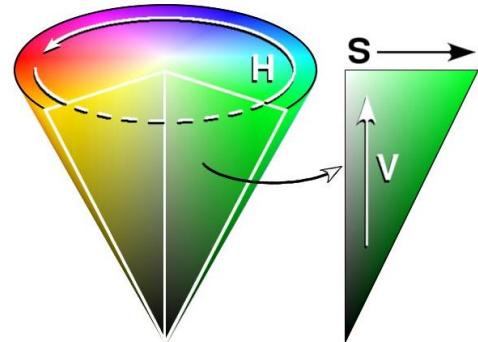
Source: <https://web.cs.wpi.edu/~emmanuel>



39

HSV (Hue – Saturation- Value)

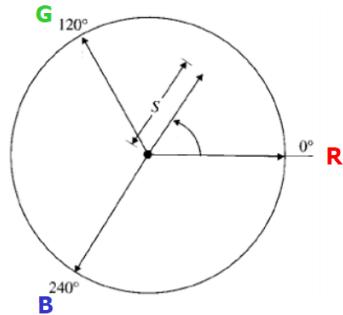
- HSV: không gian màu tốt thường đc sử dụng trong bài toán phân vùng hay nhận dạng
 - Biến đổi phi tuyến từ RGB
 - Biểu diễn trực quan màu sắc
- Mỗi pixel có:
 - Cường độ sáng: **intensity (value)**
 - Màu sắc: **color (hue + saturation)**
- RGB không có sự phân tách như thế này



40

HSV (Hue – Saturation- Value)

- **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)



41

HSV (Hue – Saturation- Value)

- Nếu biết màu của đối tượng tìm kiếm → có thể biểu diễn sử dụng 1 khoảng giá trị **H (Hue)**
- Lưu ý: H có chu kỳ
 - Hue < 60° không có nghĩa
 - 350° nhỏ hay lớn hơn 60°?
 - Cần xác định H trong 1 khoảng giá trị. VD: $350^\circ < H < 60^\circ$
- Khoảng giá trị H có ý nghĩa nếu Saturation > threshold (nếu không là màu xám)
- H, S độc lập với **Value**, Value nhạy cảm hơn với điều kiện chiếu sáng



42

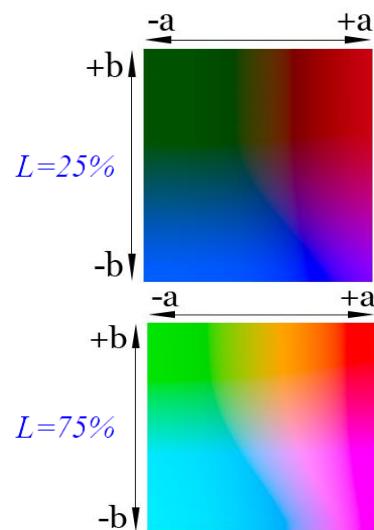
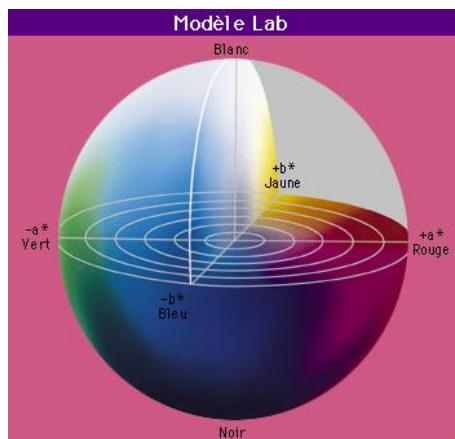
Không gian màu Lab

- **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)



43

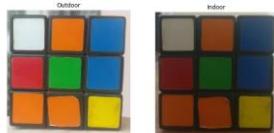
Không gian màu Lab



44

Không gian màu vs. Điều kiện chiếu sáng

- Thu thập 10 ảnh của khối lập phương trong điều kiện chiếu sáng khác nhau



- Cắt riêng từng màu để có 6 bộ cho 6 màu khác nhau



Changes in color due to varying Illumination conditions

- Tính toán phân bố giá trị của màu sắc cụ thể trong các không gian màu khác nhau

Source: Vikas Gupta, Learn OpenCV



45

Không gian màu vs. Điều kiện chiếu sáng

- Điều kiện chiếu sáng giống nhau: giá trị tập trung

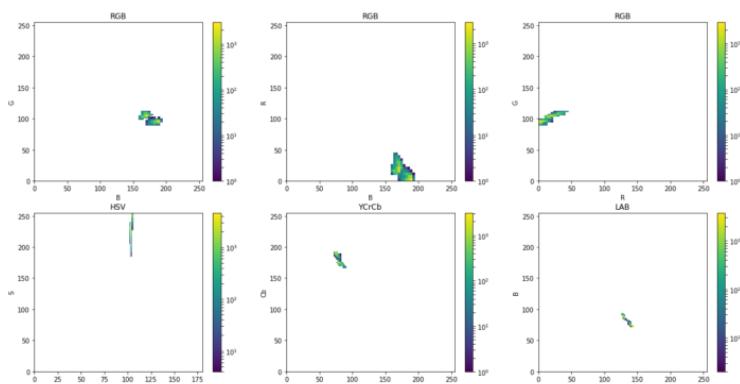


Fig.: Density Plot showing the variation of values in color channels for 2 similar bright images of blue color

Source: Vikas Gupta, Learn OpenCV



46

Không gian màu vs. Điều kiện chiếu sáng

- Điều kiện chiếu sáng giống nhau: giá trị tập trung

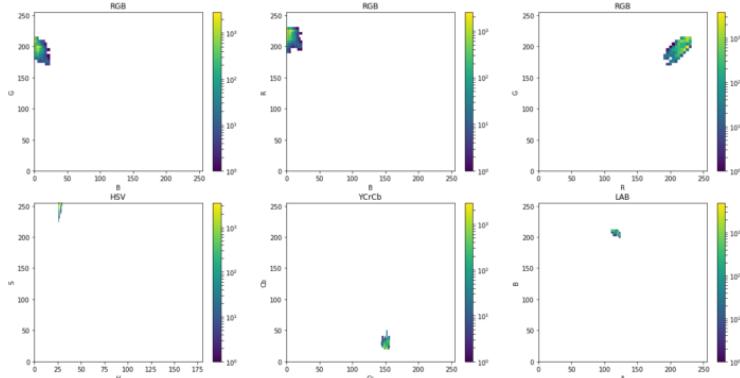


Fig.: Density Plot showing the variation of values in color channels for 2 similar bright images of **yellow color**

Source: Vikas Gupta, Learn OpenCV



47

Không gian màu vs. Điều kiện chiếu sáng

- Điều kiện chiếu sáng khác nhau

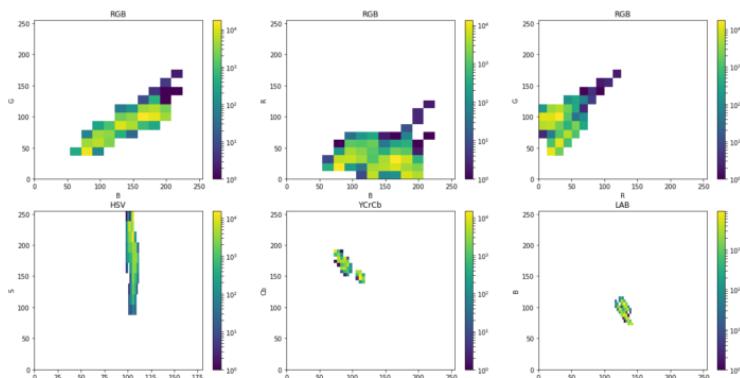


Fig.: Density Plot showing the variation of values in color channels under varying illumination for the **blue color**

Source: Vikas Gupta, Learn OpenCV



48

Không gian màu vs. Điều kiện chiếu sáng

- Điều kiện chiếu sáng khác nhau

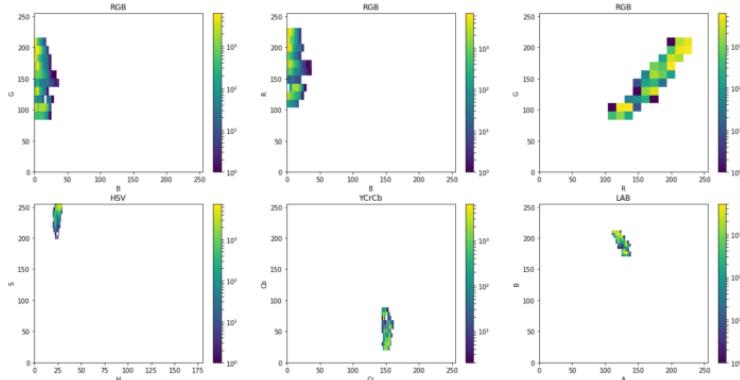


Fig.: Density Plot showing the variation of values in color channels under varying illumination for the **yellow color**

Source: Vikas Gupta, Learn OpenCV



49

Không gian màu vs. Điều kiện chiếu sáng

- Điều kiện chiếu sáng khác nhau:

- RGB : sự biến động giá trị ở các kênh lớn
- HSV: giá trị tập trung ở kênh **H**. Chỉ có kênh **H** chứa thông tin tuyệt đối về màu → 1 lựa chọn
- YCrCb, LAB: giá trị tập trung ở kênh **CrCb** và kênh **AB**
 - Giá trị tập trung tốt hơn ở không gian LAB
- Chuyển đổi giữa các không gian màu (OpenCV):
 - `cvtColor(bgr, ycb, COLOR_BGR2YCrCb);`
 - `cvtColor(bgr, hsv, COLOR_BGR2HSV);`
 - `cvtColor(bgr, lab, COLOR_BGR2Lab);`



50

Nhân chập (Convolution)

- 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
 - **Cùng hàm** được áp trên mỗi điểm ảnh
 - 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 rõ, tăng độ tương phản, ...
 - Trích chọn thông tin từ ảnh:
 - Texture, edges, distinctive points, etc.
 - Phát hiện mẫu
 - Template matching

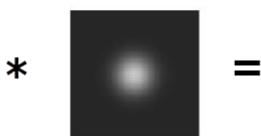


51

Nhân chập (Convolution)



Original image



Mask (kernel)

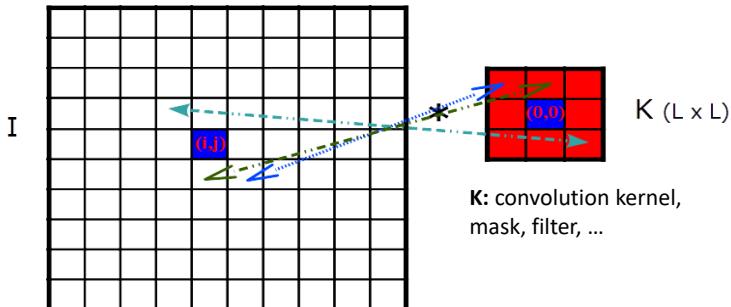


Filtered image



52

Nhân chập (Convolution)

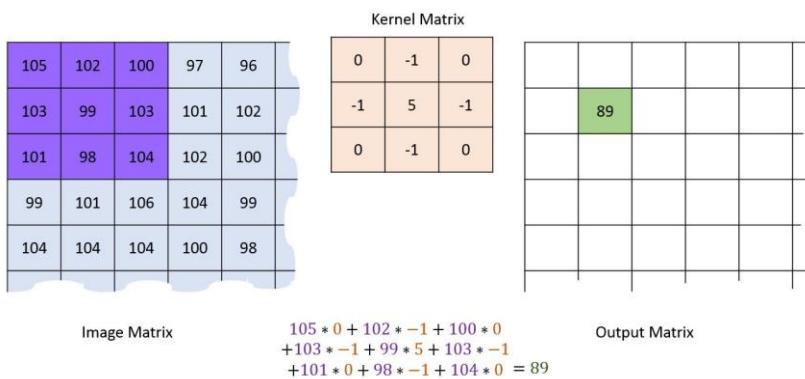


$$I'(i,j) = \sum_{u=-\frac{(L-1)}{2}}^{\frac{(L-1)}{2}} \sum_{v=-\frac{(L-1)}{2}}^{\frac{(L-1)}{2}} I(i-u, j-v) K(u, v)$$



53

Nhân chập (Convolution)



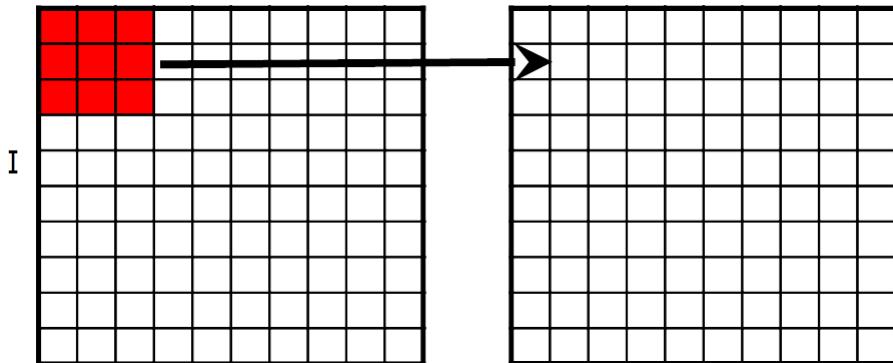
Source: <http://machinelearningguru.com>



54

Nhân chập (Convolution)

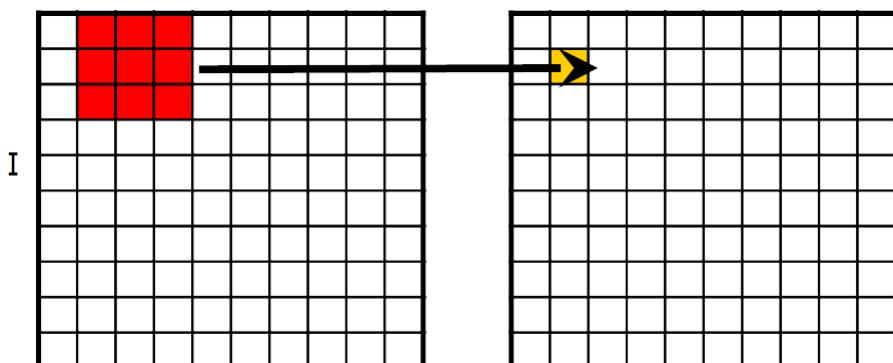
$$I' = I * K$$



55

Nhân chập (Convolution)

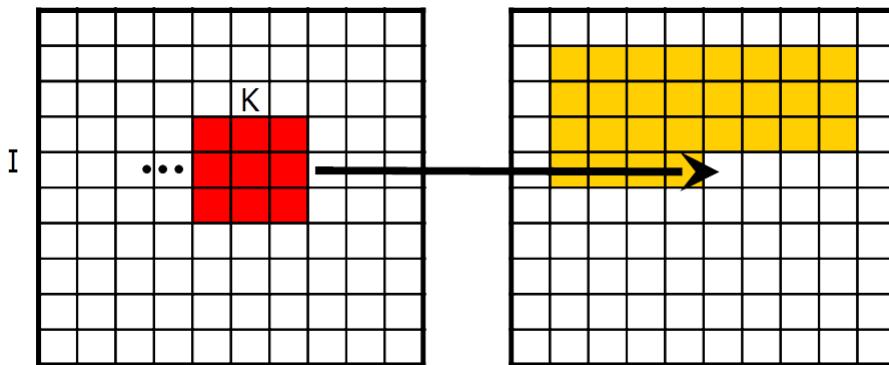
$$I' = I * K$$



56

Nhân chập (Convolution)

$$I' = I * K$$



57

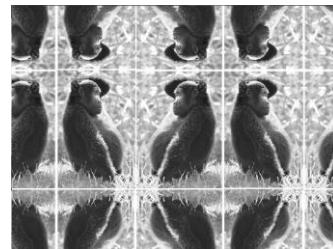
Nhân chập (Convolution)

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

0	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	

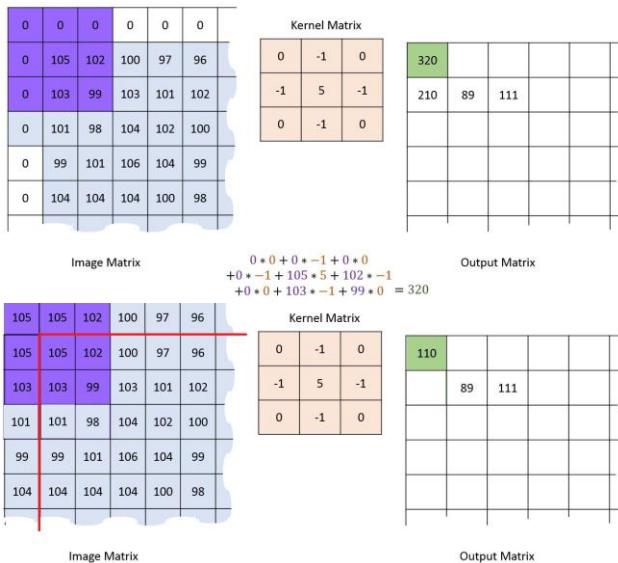
105	105	102	100	97	96	
105	105	102	100	97	96	
103	103	99	103	101	102	
101	101	98	104	102	100	
99	99	101	106	104	99	
104	104	104	104	100	98	

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



58

Nhân chập (Convolution)



Source: <http://machinelearningguru.com>

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,
 -



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)

Source: David Lowe



61

Một số bộ lọc (Some kernels)

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

 $1/9 \times$

1	1	1
1	1	1
1	1	1



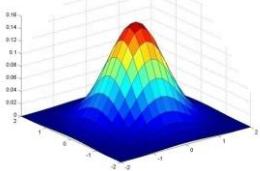
Original image

Filtered image
with box size 5x5Filtered image
with box size 11x11

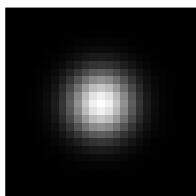
62

Một số bộ lọc (Some kernels)

- Gaussian filter



Gaussian function in 3D



Gaussian image

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

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

Rule for Gaussian filter: set **filter half-width to about 3σ** 

63

Một số bộ lọc (Some kernels)

- Bộ lọc Gauss:
 - 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 nhận chập 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 σ 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:
 - $G_\sigma(x,y) = G_\sigma(x).G_\sigma(y)$



64

Một số bộ lọc (Some kernels)

- Bộ lọc Gauss

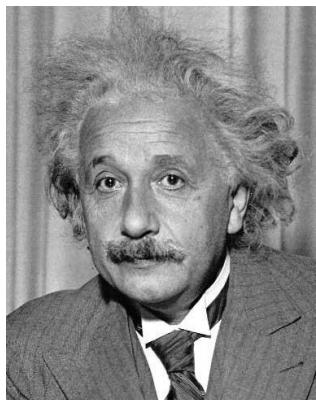


Original image

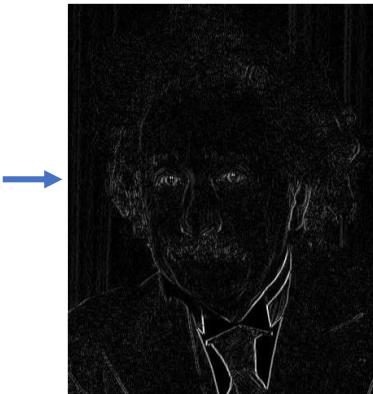
Filtered image
with box size 5x5Filtered image
with box size 11x11

Một số bộ lọc (Some kernels)

- Sobel



$$\begin{matrix} * & \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix} \end{matrix}$$

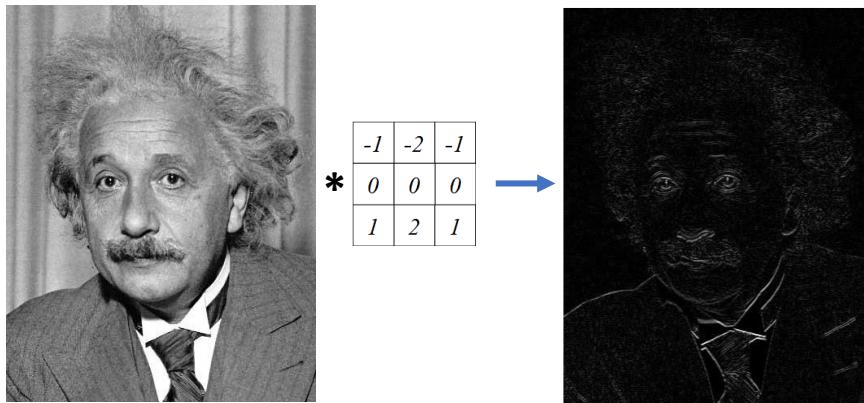


Vertical Edge (absolute value)



Một số bộ lọc (Some kernels)

- Sobel



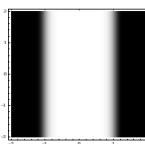
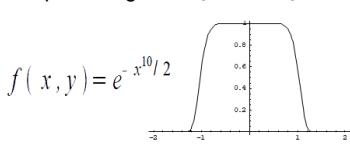
Vertical Edge (absolute value)



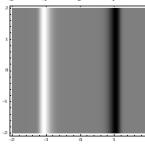
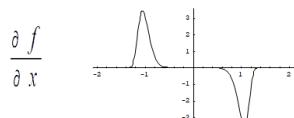
67

Phát hiện biên

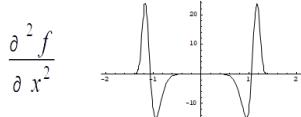
- Vị trí biên:
 - Đạt cực trị trên đạo hàm bậc 1
 - Đổi dấu qua không trên đạo hàm bậc 2



Image



First derivative



Second derivative



68

Phát hiện biên với đạo hàm bậc 1

- Tính kết quả nhân chập giữa ảnh vào bộ lọc để tính đạo hàm bậc 1
 - Bộ lọc để tính đạo hàm bậc 1: Sobel, Prewitt, Robert
 - Các bộ này đều được cài đặt OpenCV library
- Tìm cực trị địa phương
 - Biên bao gồm các điểm ảnh có giá trị **cực đại/cực tiểu** trên **đạo hàm bậc 1** của ảnh.
 - Có thể **dùng ngưỡng để phát hiện nhanh các cạnh**
 - Có thể gồm nhiều bước để tìm được cạnh tối ưu: **Canny detector** (cài đặt trong OpenCV)



Phát hiện biên với đạo hàm bậc 1

- Bộ lọc dùng tính đạo hàm bậc 1 trên ảnh
 - Robert
 - Prewitt
 - ít nhạy cảm với nhiễu
 - Làm tròn với bộ lọc trung bình + tính đạo hàm bậc 1
 - Sobel:
 - ít nhạy cảm với nhiễu
 - Làm tròn với Gauss + tính đạo hàm

1	0
0	-1

0	1
-1	0

-1	-1	-1
0	0	0
1	1	1

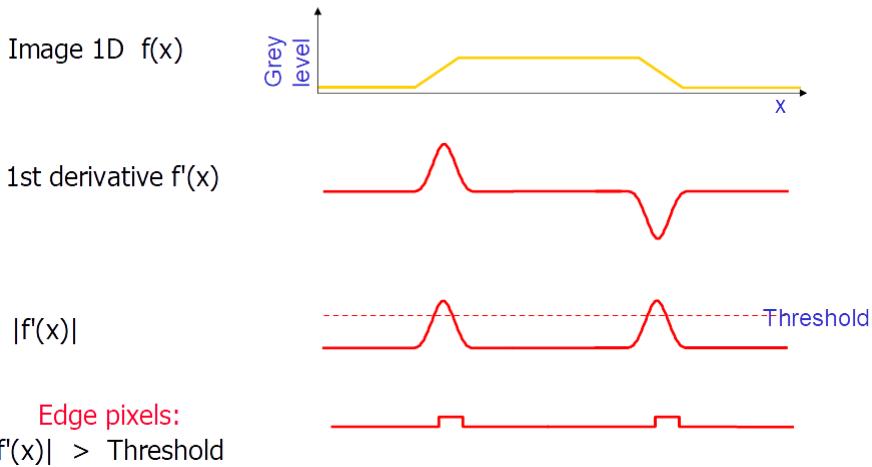
-1	0	1
-1	0	1
-1	0	1

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

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



Phát hiện biên với đạo hàm bậc 1



71

Đạo hàm (Image gradient)

- 1st derivatives :

$$I * \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \rightarrow I_x \quad \text{Đạo hàm bậc 1 theo } x$$

$$I * \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \rightarrow I_y \quad \text{Đạo hàm bậc 1 theo } y$$

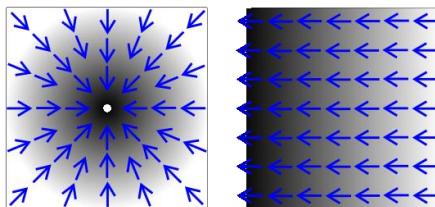
→ Image gradient



72

Đạo hàm (Image gradient)

- Đạo hàm thể hiện hướng thay đổi về cường độ sáng/màu trên ảnh
- Mỗi điểm ảnh có: G_x, G_y
- Tạo thành 1 vector đạo hàm (G_x, G_y) :
 - Thông tin quan trọng để mô tả nội dung của
 - Độ lớn đạo hàm = $\sqrt{(G_x)^2 + (G_y)^2} \approx |G_x| + |G_y|$
 - Hướng của đạo hàm = $\arctan(G_y/G_x)$



Đường màu xanh thể hiện
hướng của đạo hàm: từ
sáng nhất đến tối nhất



Phát hiện biên với đạo hàm bậc 2

- Tính đạo hàm bậc 2
 - Nhân chập ảnh với bộ lọc Laplace

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- Tìm điểm đổi dấu qua không



Đạo hàm bậc 2

- Một mặt nạ xấp xỉ cho hàm Laplace

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



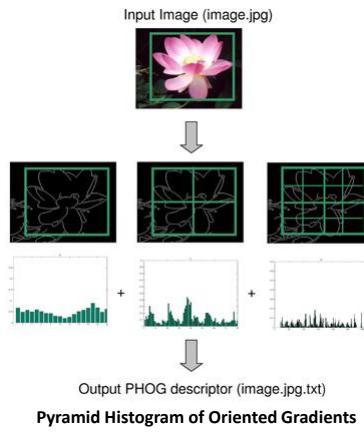
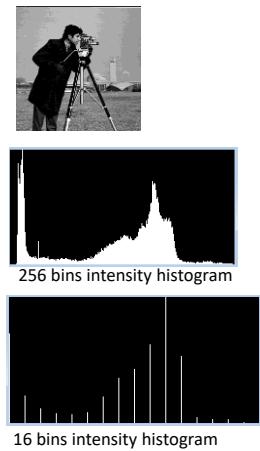
Trích chọn đặc trưng

- Hai loại đặc trưng được trích chọn từ ảnh:
 - Đặc trưng cục bộ và toàn cục
- Đặc trưng toàn cục:**
 - Mô tả toàn bộ ảnh như 1 đối tượng
 - Đặc trưng đường biên, đặc trưng hình dạng, đặc trưng kết cấu
 - Ví dụ: Invariant Moments (Hu, Zernike), Histogram Oriented Gradients (HOG), PHOG, and Co-HOG,...
- Đặc trưng cục bộ:**
 - Mô tả đặc trưng cục bộ mô tả từng vùng nhỏ trong ảnh, từng vùng cục bộ của đối tượng (điểm đặc trưng trong ảnh).
 - Biểu diễn đặc trưng kết cấu/màu sắc trong mỗi vùng cục bộ ảnh
 - Ví dụ: SIFT, SURF, LBP, BRISK, MSER và FREAK, ...



Trích chọn đặc trưng

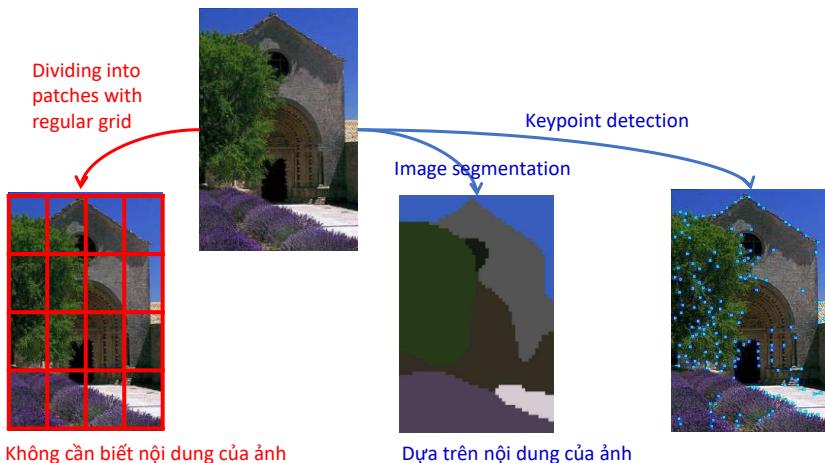
- Đặc trưng toàn cục



77

Trích chọn đặc trưng

- Đặc trưng cục bộ: xác định vùng cục bộ như thế nào?



79

Trích chọn đặc trưng

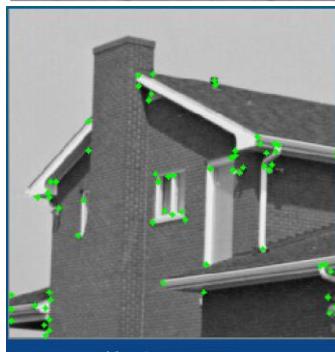
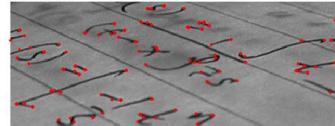
- Phân vùng ảnh
 - Lấy ngưỡng (Thresholding)
 - Chia và hợp (Split and merge)
 - Phát triển vùng (Region growing)
 - Watershed
 - ...



80

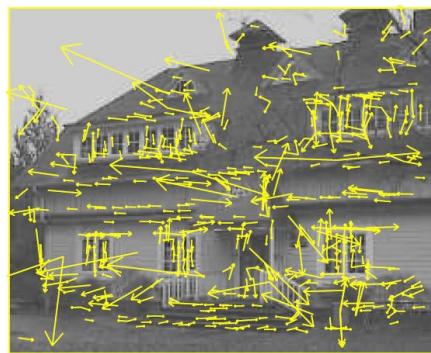
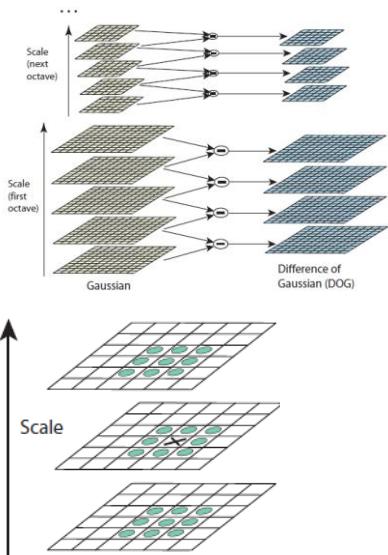
Trích chọn đặc trưng

- Phát hiện các điểm đặc trưng:
 - DoG /SIFT detector
 - Harris corner detector
 - Moravec
 - ...
- Đặc trưng cục bộ: tính trên các vùng cục bộ xung quanh điểm đặc trưng:
 - SIFT,
 - SURF(Speeded Up Robust Features),
 - PCA-SIFT
 - LBP, BRISK, MSER and FREAK, ...



81

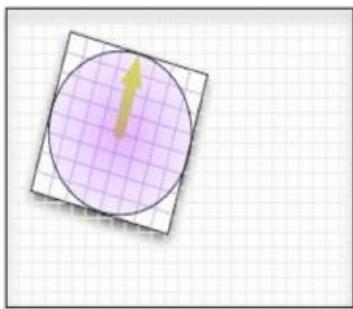
VD: Bộ phát hiện điểm đặc trưng DoG/SIFT



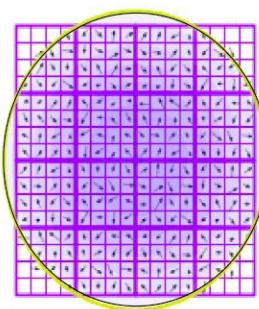
Source: [Distinctive Image Features from Scale-Invariant Keypoints – IJCV 2004](#)

82

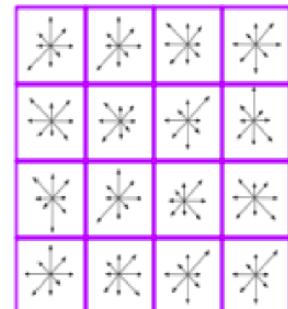
VD: đặc trưng SIFT



Blur the image using the scale of the keypoint (scale invariance)



Compute gradients in respect to the keypoint orientation (rotation invariance)



Compute orientation histogram in 8 directions over 4x4 sample regions

Source: [Distinctive Image Features from Scale-Invariant Keypoints – IJCV 2004](#)
http://campar.in.tum.de/twiki/pub/Chair/TeachingWs13TDCV/feature_descriptors.pdf

83



Trích chọn đặc trưng: Đặc trưng tốt?

- Gọn (Compact)
- Bất biến với
 - Một số phép biến đổi hình học
 - Góc nhìn camera
 - Điều kiện chiếu sáng
- Một trong những đặc trưng tốt nhất: **SIFT** (David Lowe)



Một số bộ đặc trưng cục bộ khác

- Popular features: SURF, HOG, SIFT
http://campar.in.tum.de/twiki/pub/Chair/TeachingWs13TDCV/feature_descriptors.pdf
- Summary some local features:
http://www.cse.iitm.ac.in/~vplab/courses/CV_DIP/PDF/Feature_Detectors_and_Descriptors.pdf



Trích chọn đặc trưng: OpenCV

- SIFT & SURF:
 - Đã đăng ký sở hữu trí tuệ
 - Sử dụng tự do cho mục đích nghiên cứu/giảng dạy
 - Sử dụng trong các ứng dụng **thương mại**: cần giấy phép
- Từ OpenCV 3.0, các thuật toán đã ddk sở hữu trí tuệ:
 - Được loại bỏ khỏi gói cài đặt chuẩn,
 - Được đặt trong gói "non-free" ([opencv-contrib](#), không được cài đặt mặc định)
- Một số các thay thế cho sift, surf:
 - ORB (Oriented FAST and Rotated Brief)
 - BRIEF, BRISK, FREAK, KAZE and AKAZE



Trích chọn đặc trưng: OpenCV

- SIFT
 - **sift.detect()** : tìm các điểm đặc trưng
 - **sift.compute()** : tính đặc trưng cục bộ tại các điểm đặc trưng
 - sift = cv.xfeatures2d.SIFT_create()
 - kp = sift.detect(gray,None)
 - kp,des = sift.compute(gray,kp)
 - Tìm điểm đặc trưng và đặc trưng cục bộ: **sift.detectAndCompute()**
 - sift = cv.xfeatures2d.SIFT_create()
 - kp, des = sift.detectAndCompute(gray,None)
 - https://docs.opencv.org/3.4/da/df5/tutorial_py_sift_intro.html
- SURF: tương tự



Trích chọn đặc trưng: OpenCV

- SURF:

```
>>> img = cv.imread('fly.png',0)
# Create SURF object. You can specify
params here or later.
# Here I set Hessian Threshold to 400
>>> surf = cv.xfeatures2d.SURF_create(400)
# Find keypoints and descriptors directly
>>> kp, des =
surf.detectAndCompute(img,None)
>>> len(kp)
699
```



https://docs.opencv.org/3.4/d4/dd2/tutorial_py_surf_intro.html

88



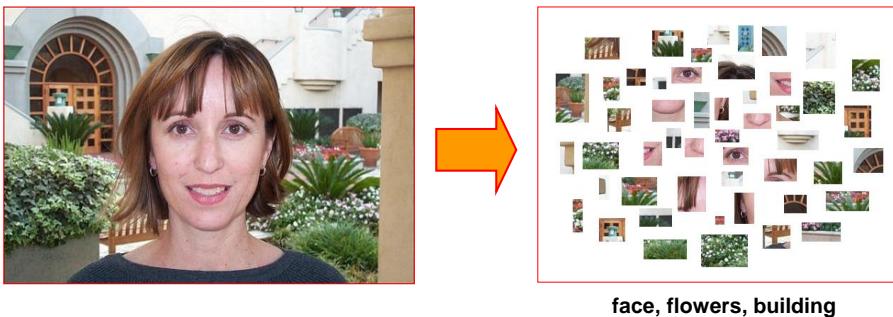
Mô hình túi từ

- Xuất phát từ mô hình biểu diễn văn bản không tính đến thứ tự các từ: chỉ tính đến tần suất xuất hiện của từ trong văn bản Salton & McGill (1983)



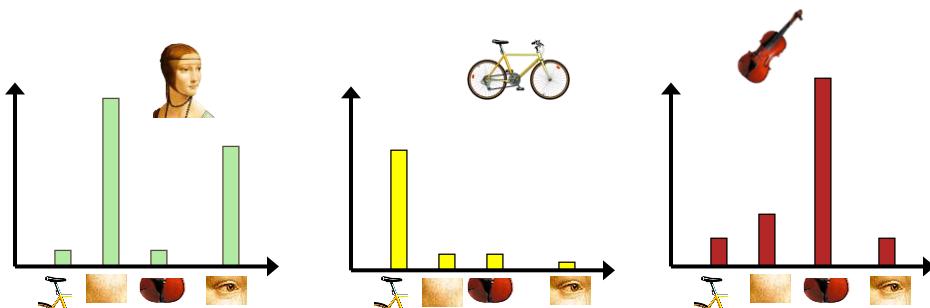
Mô hình túi từ (Bags of features) để nhận dạng đối tượng

- Hoạt động tốt cho phân loại ảnh cũng như bài toán nhận dạng các thể hiện của đối tượng



Mô hình túi từ: các bước chính

1. Trích chọn đặc trưng
2. Học từ điển trực quan (visual vocabulary)
3. Xác định từ tương ứng cho các đặc trưng
4. Biểu diễn ảnh bởi tần suất xuất hiện của các từ trực quan (“visual words”)



References

- CVIP tool to explore the power of computer processing of digital images: Many methods in image processing and computer vision have been implemented
 - <https://cviptools.ece.siu.edu/>
- Library: OpenCV, with C/C++, Python and Java interfaces. OpenCV was designed for computational efficiency and with a strong focus on real-time application: <https://opencv.org/>
- Books:
 - Rafael C. Gonzalez, Richard Eugene Woods, Digital Image Processing, 2nd edition, Prentice-Hall, 2002: Chap 3 (spatial operators), 6 (Color spaces)
 - Richard Szeliski, Computer Vision: Algorithms and Applications, Springer, 2010. <http://szeliski.org/Book/>
- Articles:
 - SIFT (DoG detector and SIFT descriptor): <https://www.cs.ubc.ca/~lowe/keypoints/>
 - SURF: Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, "Speeded Up Robust Features", ETH Zurich, Katholieke Universiteit Leuven
 - GLOH: [Krystian Mikolajczyk and Cordelia Schmid "A performance evaluation of local descriptors", IEEE Transactions on Pattern Analysis and Machine Intelligence, 10, 27, pp 1615--1630, 2005.](#)
 - PHOG: <http://www.robots.ox.ac.uk/~vgg/research/caltech/phog.html>
 - <https://www.learnopencv.com/> : many examples with code in C++/ Python and clear explanation



92

HUST

THANK YOU !