



TRƯỜNG ĐẠI HỌC VINH
VINH UNIVERSITY

Nơi tạo dựng tương lai cho tuổi trẻ



Chương 1. Tổng quan về xử lý ảnh

ThS. Nguyễn Thị Minh Tâm
Email: tamntm@vinhuni.edu.vn

Đại học Vinh
Viện Kỹ thuật Công nghệ

ĐẠI HỌC VINH - 2022



NỘI DUNG MÔN HỌC

1. Chương 1: Giới thiệu chung

Trình bày tổng quan về xử lý ảnh, các khái niệm cơ bản, sơ đồ tổng quát của một hệ thống xử lý ảnh và các vấn đề cơ bản trong xử lý ảnh.

2. Chương 2: Các phép toán cơ bản và phương pháp xử lý ảnh số

Trình bày các vấn đề liên quan đến một số phép biến đổi ảnh, bao gồm các kỹ thuật phân ngưỡng, thay đổi kích thước ảnh, cân bằng lược đồ mức xám

3. Chương 3: Nâng cao chất lượng ảnh

Trình bày các kỹ thuật liên quan đến lược đồ xám; Trình bày các kỹ thuật lọc ảnh nhằm nâng cao chất lượng ảnh.

4. Chương 4: Phát hiện và tách biên

Trình bày các kỹ thuật cơ bản trong việc phát hiện biên của các đối tượng ảnh theo cả hai khuynh hướng: phát hiện biên trực tiếp và phát hiện biên gián tiếp.

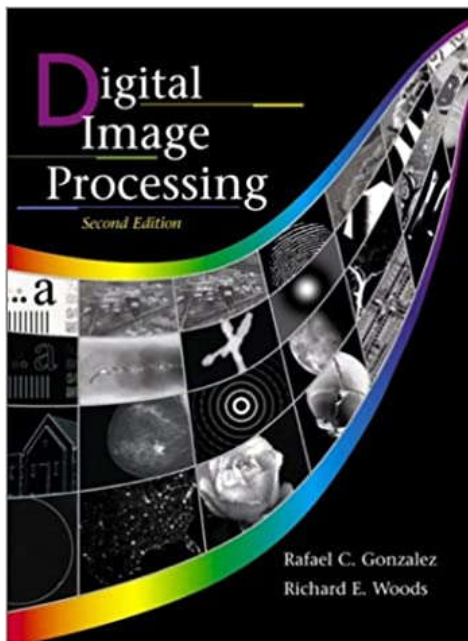
5. Chương 5: Phân vùng ảnh

Các kỹ thuật phân vùng ảnh, đây là khâu quan trọng hỗ trợ cho việc trích chọn các thuộc tính của ảnh để tiến tới hiểu ảnh.

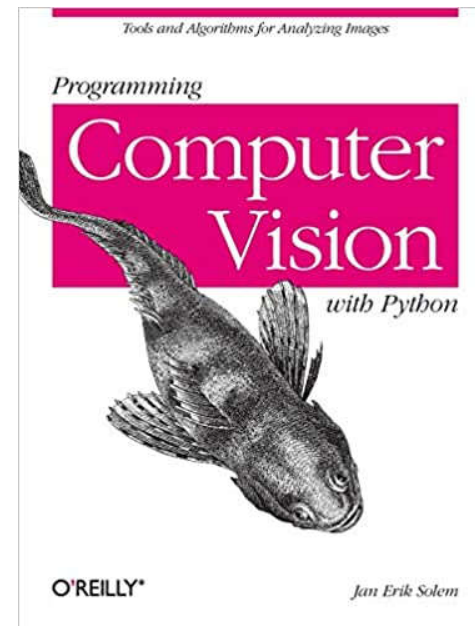


Tài liệu tham khảo

1. *Digital Image processing*
(second edition). Gonzales



2. *Programming Computer Vision with Python*, Jan Erik Solem





Chương 1: Giới thiệu chung

▪ Mục đích:

- Tổng quan về một hệ thống xử lý ảnh
- Một số khái niệm trong xử lý ảnh
- Màu và phương pháp biểu diễn màu
- Phương pháp biểu diễn ảnh trong máy tính điện tử
- Cấu trúc chung của tệp ảnh bitmap



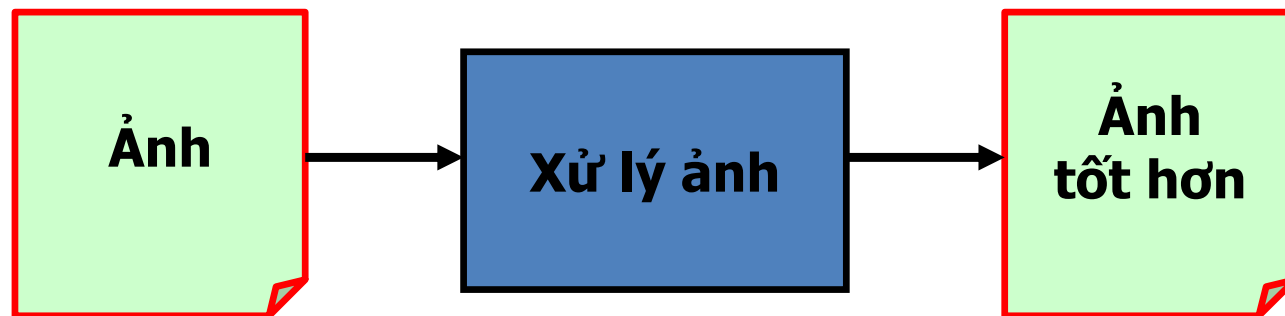
1.1 Tổng quan về một hệ thống xử lý ảnh

- Hình ảnh đóng vai trò quan trọng trong việc trao đổi thông tin.
- Tính trực quan của hình ảnh giúp hiểu rõ và sâu sắc hơn về thông tin.
- Phân tích, tổng hợp hình ảnh theo ý tưởng, mục đích của người sử dụng.



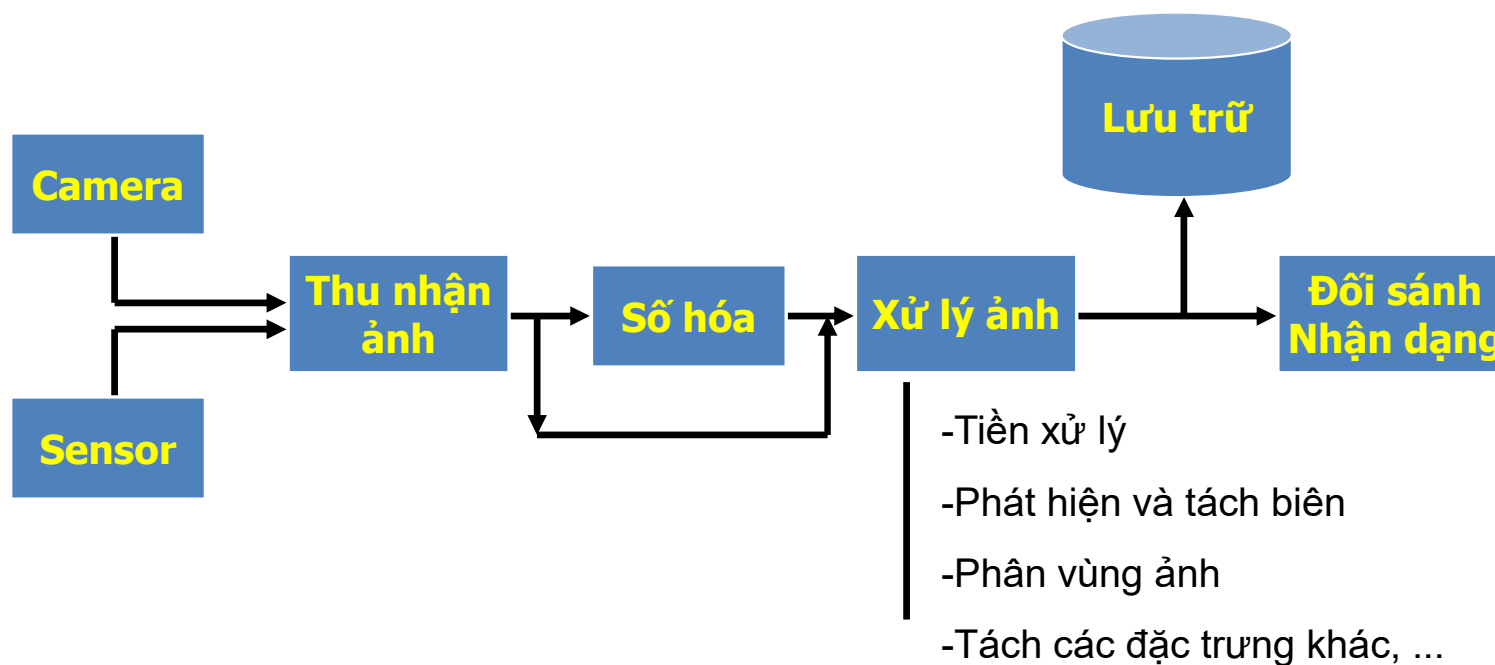
Xử lý ảnh

- Là một phần của lĩnh vực xử lý tín hiệu số.
- Tăng cường chất lượng thông tin hình ảnh đối với quá trình tri giác của con người và biểu diễn trên máy tính.





Các giai đoạn chính trong XLA





Các giai đoạn chính trong XLA

- **Thu nhận ảnh:** ảnh được thu nhận từ thế giới thực qua máy chụp hình, từ tranh ảnh qua máy quét hoặc từ vệ tinh dưới dạng tín hiệu số hoặc tương tự
- **Số hóa ảnh:** Biến đổi tín hiệu tương tự sang tín hiệu rời rạc
 - Rời rạc hóa (lấy mẫu)
 - Lượng tử hóa.
- **Xử lý ảnh:** Gồm nhiều thao tác: Tăng cường ảnh, khử nhiễu, chỉnh méo, phát hiện tách biên, phân vùng ảnh, trích chọn đặc trưng khác ...
- **Lưu trữ ảnh:** Dữ liệu ảnh cũng như các dữ liệu khác để có thể chuyển giao cho quá trình khác cần phải được lưu trữ
- **Đối sánh nhận dạng:** Phân lớp phục vụ cho các mục đích khác nhau



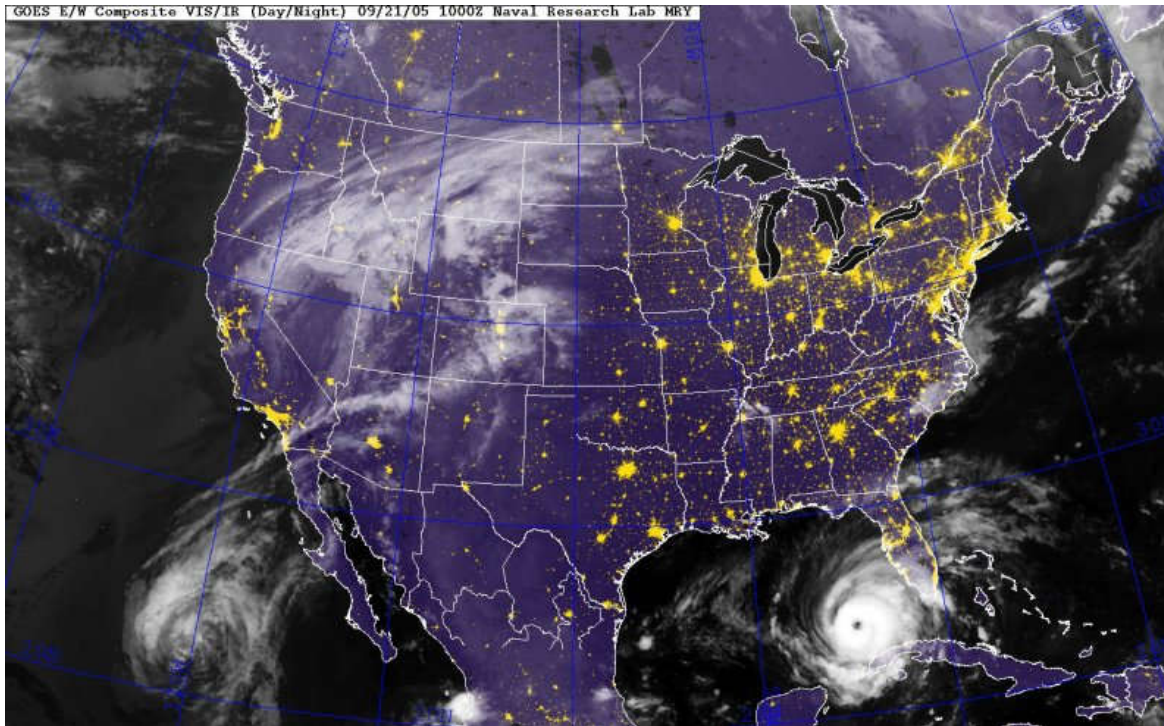
Ứng dụng của xử lý ảnh

- Ứng dụng trong Y khoa
 - Xem xét và giải thích hình ảnh X quang
 - Phân tích hình ảnh tế bào
 - ...
- Ứng dụng trong nông nghiệp
 - Xử lý hình ảnh thu được từ vệ tinh của các khu đất -> phân loại đất -> tìm loại cây trồng phù hợp.
 - Phân tích tình hình sâu, bệnh thông qua xử lý màu lá cây.
 - ...
- Ứng dụng trong công nghiệp
 - Kiểm tra sản phẩm trong dây chuyền sản xuất tự động
 - Kiểm tra mẫu giấy



Ứng dụng của xử lý ảnh

- Ảnh hồng ngoại



Ảnh hồng ngoại chụp bề mặt trái đất. Những nơi có ánh sáng mạnh là những nơi có nguồn nhiệt lớn.



Ứng dụng của xử lý ảnh

- Nhận dạng dấu vân tay, vân môi của con người.
- Nhận dạng gương mặt con người.
- Nhận dạng chữ viết tay
- Nhận dạng ngôn ngữ ký hiệu.
- Nhận dạng chữ ký.
- Nhận dạng màu da người.
- Phân tích chữ viết tay.
- Chấm điểm trắc nghiệm bằng máy.
- ...

Ví dụ về xử lý ảnh



(a) The original image



(b) Result after "sharpening"

Ví dụ về xử lý ảnh



(a) The original image

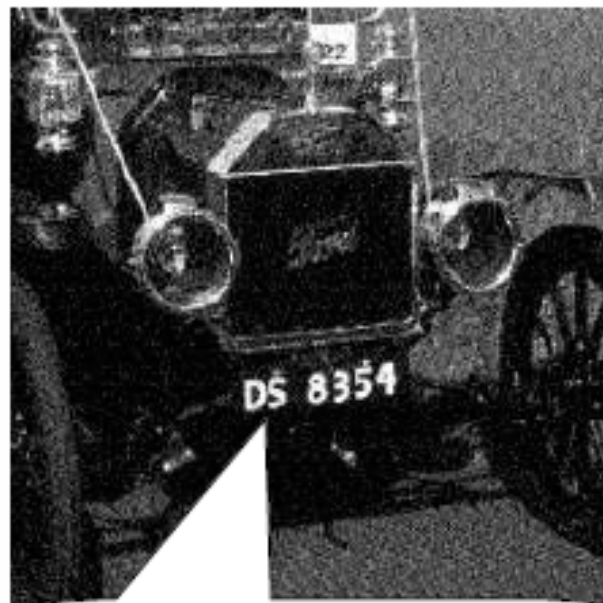


(b) After removing noise

Ví dụ về xử lý ảnh



Ảnh gốc nhòe, không nhìn
rõ biển số xe

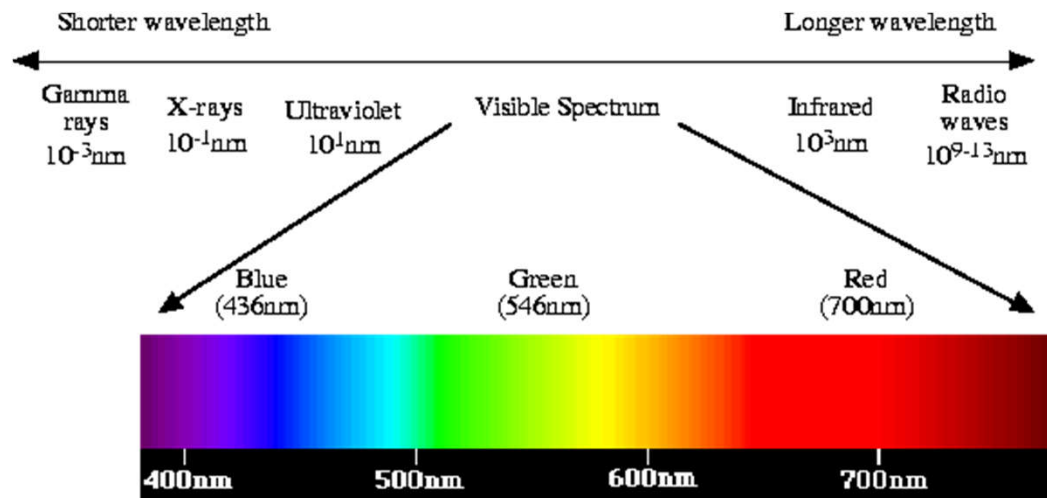


Ảnh đã được làm bớt
nhòe, thấy rõ biển số xe



1.2 Biểu diễn màu

- Khái niệm: Theo quang học màu là một sóng ánh sáng có một bước sóng nhất định nằm trong dải ánh sáng có thể nhìn thấy được.





1.2 Biểu diễn màu (tiếp)

- Để biểu diễn và lưu trữ màu người ta có thể biểu diễn và lưu trữ thông qua bước sóng của màu đó.
- Thuyết 3 màu của Thomas Young: một màu bất kì có thể được tổ hợp từ 3 màu cơ bản: Red, Green, Blue (R, G, B)
- Do đó, trong máy tính người ta thường biểu diễn và lưu trữ màu dựa trên 3 thành phần cơ bản của màu đó. Theo chuẩn CIE:
 - Red: có bước sóng: $\lambda = 700 \text{ nm}$
 - Green có bước sóng: $\lambda = 546.1 \text{ nm}$
 - Blue có bước sóng: $\lambda = 435.8 \text{ nm}$



1.3 Biểu diễn ảnh số

- Pixel (picture element): điểm ảnh hay phần tử ảnh. Mỗi pixel gồm một cặp tọa độ x, y và màu.
- Mức xám (grey level): Mức xám là kết quả sự mã hoá tương ứng một cường độ sáng của mỗi điểm ảnh với một giá trị số - kết quả của quá trình lượng hoá.
- Độ phân giải (resolution): Cặp tọa độ x, y tạo nên độ phân giải.



1.3 Biểu diễn ảnh số (tiếp)

- Để biểu diễn ảnh số người ta sử dụng một ma trận 2 chiều $X(m,n)$, trong đó:
 - $X(m,n)$: biểu diễn màu tại điểm có tọa độ (m,n)
- Mỗi màu được biểu diễn thông qua 3 thành phần màu cơ bản
→ biểu diễn ảnh số qua 3 ma trận hai chiều:
 $R(m,n)$; $G(m,n)$; $B(m,n)$



Ảnh đen trắng

- Ảnh đen trắng: là ảnh chỉ chứa các mức xám được trải từ màu đen tới màu trắng
 - Nếu số mức xám $L = 2$: ảnh nhị phân
 - Thường xử lý ảnh 256 mức, ta gọi là ảnh đen trắng 256 mức.
 - Các mức ở đây là một màu và 3 thành phần màu có giá trị bằng nhau. ($R = G = B$).
 - Do đó ta có 4 ma trận bằng nhau: $R(m,n) = G(m,n) = B(m,n) = X(m,n)$
- ⇒ Khi xử lý ảnh chỉ cần xử lý trên $X(m,n)$.



Ảnh màu

- Ảnh màu: là ảnh chứa các màu là tổ hợp của 3 thành phần R, G, B. Thông thường:

$$R(m,n) \neq G(m,n) \neq B(m,n) \neq X(m,n)$$

- Ba thành phần cơ bản được chia thành các mức. Nếu mỗi thành phần được chia thành 256 mức = 2^8 ta có ảnh 24 bit màu.
- Do đó, khi xử lý ảnh ta phải xử lý trên 3 ma trận thành phần, sau đó tổ hợp lại ta được ma trận X



Lưu ý

- Khi áp dụng các thuật toán xử lý ảnh chỉ áp dụng trên một ma trận:
 - Nếu ảnh đen trắng: $X(m,n)$
 - Nếu ảnh màu: áp dụng thuật toán rời rạc trên 3 ma trận R, G, B sau đó tổ hợp lại.



Các thư viện mã nguồn mở

▪ Chúng ta có thể tận dụng những thành quả của cộng đồng mã nguồn mở để áp dụng cho ứng dụng xử lý ảnh và thị giác máy tính. Hiện nay trên thế giới có rất nhiều thư viện mã nguồn mở cho ta sử dụng, như:

- Intel OpenCV
- CMCVision
- ImLib 3D
- ImaLab...



Thư viện OpenCV

- OpenCV là một thư viện mã nguồn mở hàng đầu cho xử lý ảnh và thị giác máy tính





Thư viện OpenCV

OpenCV is an image processing library created by Intel and later supported by Willow Garage and now maintained by Itseez. opencv is available on Mac, Windows, Linux. Works in C, C++, and Python. It support cross-platform integration and it is flexible. It is Open Source and free. opencv is easy to use and install.



Cài đặt

IDE (Integrated Development Environment)

Editor to write Code

Most of the IDE contain

PYTHON AS INTERNAL PACKAGE

**Choose IDE as
per your
compatibility**



Cài đặt



**Anaconda Navigator contains
so many libraries.**



Cài đặt

- Anaconda là một nền tảng phân phối ngôn ngữ lập trình python miễn phí, phục vụ cho tính toán khoa học và phát triển trí tuệ nhân tạo. Anaconda bao gồm ngôn ngữ lập trình python cộng với các thư viện và gói package cần thiết thường dùng, với mục đích quản lý và triển khai môi trường lập trình python dễ dàng hơn.
- Thông thường, để sử dụng Python, sau khi cài đặt Python, chúng ta cần phải cài đặt thêm các thư viện và các **gói package cần thiết** để chạy chương trình. Ví dụ như để phát triển AI bằng python, chúng ta cần cài đặt thêm Tensorflow, **matplotlib**, Open CV chẳng hạn. Tuy nhiên, nếu bạn cài đặt anaconda python, không những python mà các thư viện và gói package thường dùng cũng sẽ được cài đặt luôn theo đó.



Cài đặt








Cài đặt Anaconda

- Cài anaconda python individual edition

▪ <https://www.anaconda.com/products/individual>

Windows 	MacOS 	Linux 
Python 3.9 64-Bit Graphical Installer (510 MB) 32-Bit Graphical Installer (404 MB)	Python 3.9 64-Bit Graphical Installer (515 MB) 64-Bit Command Line Installer (508 MB)	Python 3.9 64-Bit (x86) Installer (581 MB) 64-Bit (Power8 and Power9) Installer (255 MB) 64-Bit (AWS Graviton2 / ARM64) Installer (488 M) 64-bit (Linux on IBM Z & LinuxONE) Installer (242 M)

- Cài OpenCV:

- Vào command prompt, gõ
- `pip install opencv-python`

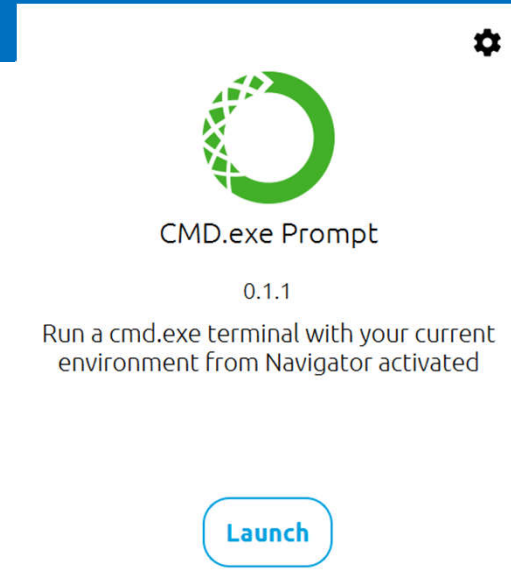
- Mở spyder, gõ `import cv2`

- Chạy chương trình để test



Cài đặt OpenCV

- Cài OpenCV:
- Vào command prompt, gõ
- `pip install opencv-python`
- Mở spyder, gõ `import cv2`



```
(base) C:\Users\LENOVO>pip install opencv-python
Collecting opencv-python
  Downloading opencv_python-4.5.5.62-cp36-abi3-win_amd64.whl (35.4 MB)
    | 35.4 MB 3.3 MB/s
Requirement already satisfied: numpy>=1.14.5 in g:\anaconda\lib\site-packages (from opencv-python) (1.20.3)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.5.5.62
```



Ảnh và ma trận

- Trong OpenCV, sử dụng mảng 2 chiều để lưu trữ ảnh



j →

↓ i

62	79	23	119	120	105	4	0
10	10	9	62	12	78	34	0
10	58	197	46	46	0	0	48
176	135	5	188	191	68	0	49
2	1	1	29	26	37	0	77
0	89	144	147	187	102	62	208
255	252	0	166	123	62	0	31
166	63	127	17	1	0	99	30



Thư viện Numpy

- Numpy (Numeric Python): là một thư viện toán học phổ biến và mạnh mẽ của Python. Cho phép làm việc hiệu quả với ma trận và mảng, đặc biệt là dữ liệu ma trận và mảng lớn với tốc độ xử lý nhanh hơn nhiều lần khi chỉ sử dụng “core Python” đơn thuần.
- Cài đặt thư viện Numpy: Mở Command Prompt và gõ lệnh: `pip install numpy`
- Khai báo thư viện: `import numpy as np`
- Khởi tạo mảng một chiều với kiểu dữ liệu các phần tử là Integer
 - `arr = np.array([1,3,4,5,6], dtype = int)`
- Khởi tạo mảng hai chiều
 - `arr1 = np.array([(4,5,6), (1,2,3)], dtype = int)`
- Tạo mảng hai chiều các phần tử 0 với kích thước 3x4
 - `np.zeros((3,4), dtype = int)`

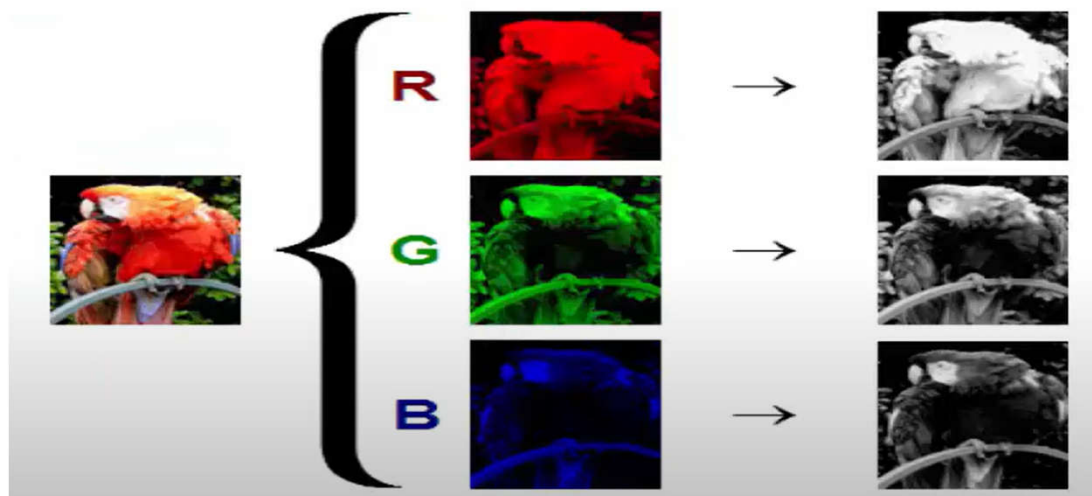


Không gian màu – color space

- Cấu tạo tự nhiên của mắt nhạy với 3 màu Red-Green-Blue. Dựa vào đó các thiết bị hiển thị màu đa số đều cấu tạo để phù hợp với mắt người. Các hệ màu khác được sử dụng chủ yếu cho in ấn, xử lý dữ liệu hoặc cần hiển thị màu đẹp hơn.
- Ví dụ: màn hình LCD máy tính cấu tạo mỗi pixel màu từ 3 kênh màu, để thay đổi giá trị của pixel màu người ta thay đổi giá trị của từng kênh.

Màu sắc của ảnh

- Màu được tổ hợp từ 3 màu RGB (Red, Green, Blue)
- Trong OpenCV, thứ tự màu là BGR (Blue, Green, Red)





Hệ màu RGB

- Hình ảnh đa số cấu tạo từ 3 kênh màu Red-Green-Blue. Tùy theo từng trường hợp sử dụng mà có thêm kênh thứ tư là kênh Alpha (độ trong suốt). Khi tách mỗi kênh màu ta được 1 ma trận các giá trị từ $[0;255]$ với 8 bits, tương tự là $[0;65535]$ với 16 bits



Ảnh nhị phân và ảnh xám

- Ảnh nhị phân và ảnh xám được tạo bởi 1 kênh màu, mỗi pixel được tạo bởi 1 màu duy nhất. Các hệ màu khác được tạo từ nhiều kênh màu, có nghĩa là tách từng kênh ra ta sẽ được ảnh xám



Màu sắc của ảnh

- Ảnh xám: 1 kênh màu (0-255)
- Ảnh màu: 3 kênh màu



SINGLE CHANNEL



3 - CHANNEL



Màu sắc của ảnh



$$Pixel = \begin{cases} R \\ G \\ B \end{cases}$$

Pixel =

$$\begin{aligned} &0.2989 * R \\ &+ 0.5870 * G \\ &+ 0.1140 * B \end{aligned}$$



Pixel =

$$\begin{cases} 0, \text{nếu } Pixel < T \\ 1, \text{nếu } Pixel \geq T \end{cases}$$





Bài tập

- Bài tập 1: Viết chương trình chuyển ảnh màu sang ảnh xám, ảnh nhị phân



Đọc hiện ảnh

```
import cv2  
img = cv2.imread("đường dẫn") # "C:\\Picture\\Apple.jpg"  
cv2.imshow("Picture Apple",img)  
cv2.waitKey(0)  
cv2.destroyAllWindows()
```




Hệ màu HSV

- Hệ màu HSV dùng để xử lý màu rất thuận tiện
- Mỗi màu sẽ có 1 giá trị Hue $[0;360]$, còn Value và Saturation chỉ thị độ sáng và độ bão hoà của màu. Hệ màu này được ứng dụng để theo dõi đối tượng theo màu
- Hệ màu HSB tương tự HSV nhưng Value được thay bằng Brightness
- Chuyển hệ màu RGB sang HSV:



Hàm chuyển đổi hệ màu

- `image = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY)`
- `image = cv2.cvtColor(src, cv2.COLOR_BGR2HSV)`



Một số hàm thuộc tính ảnh

- Kích thước ảnh: `print(img.shape)`
- Nếu ảnh xám, trả về bộ gồm số hàng và số cột. Nếu ảnh màu, trả về bộ 3 gồm chiều cao, chiều rộng, độ sâu → Có thể dùng để kiểm tra ảnh xám hay ảnh màu

```
img = cv2.imread('example.jpg')  
(h, w, d) = img.shape  
print("width={}, height={}, depth={}".format(w, h, d))
```

- Output: width=580, height=326, depth=3
- Tổng số điểm ảnh: `print(img.size)`
- Image datatype: `print(img.dtype)` #uint8



Lấy kích thước ảnh

Kiểm tra ảnh đọc vào là ảnh màu hay ảnh xám

```
import cv2
img = cv2.imread("G:\\Codes\\rgb.jpg",1)
try:
    (h, w, d) = img.shape
    print("width={}, height={}, depth={}".format(w, h, d))
    print('anh màu')
except:
    print('anh xám')
    (h, w) = img.shape
    print("width={}, height={}".format(w, h))
```



Một số hàm thuộc tính ảnh

- Tách màu ảnh: `b,g,r = cv2.split(img)`
- Hoặc: `img_blue=img[:, :, 0]`
- Gộp màu ảnh: `img = cv2.merge((b,g,r))`



Lấy giá trị màu ở một điểm ảnh

```
(B, G, R) = img[50, 50]
```

```
print("R={}, G={}, B={}".format(R, G, B))
```

output: R=96, G=100, B=111

```
(B, G, R) = img[50, 50]  
print("R={}, G={}, B={}".format(R, G, B))  
output: R=96, G=100, B=111
```



Tham khảo

- <https://www.geeksforgeeks.org/opencv-python-tutorial/?ref=lbp>
- https://docs.opencv.org/3.4/d2/d96/tutorial_py_table_of_contents_imgproc.html



Đọc video

```
import cv2

vid_capture =
cv2.VideoCapture('video_demo.mp4')

if (vid_capture.isOpened() == False):
    print("Lỗi mở file")
else:
    fps = vid_capture.get(5)
    #CAP_PROP_FPS frame per second
    print('Số hình trong 1 giây:', fps)
    frame_count = vid_capture.get(7)
    #CAP_PROP_FRAME_COUNT
    print('Số khung hình:', frame_count)
```

```
while(vid_capture.isOpened()):
    ret,frame= vid_capture.read()
    if ret == True:
        cv2.imshow('Frame',frame)
        key = cv2.waitKey(20)
        if key == ord('q'):
            break
    else:
        break
vid_capture.release()
cv2.destroyAllWindows()
```




Đọc chuỗi ảnh

- Sử dụng ký hiệu %nd
- Ví dụ:

```
vid_capture = cv2.VideoCapture('Resources/Image_sequence/Cars%04d.jpg')
```

- Cars%04d.jpg: trong đó %04d chỉ ra quy ước đặt tên theo dãy gồm bốn chữ số (ví dụ: Cars0001.jpg, Cars0002.jpg, Cars0003.jpg,...).
- Nếu “Race_Cars_%02d.jpg” thì sẽ tìm kiếm các tệp có dạng: (Race_Cars_01.jpg, Race_Cars_02.jpg, Race_Cars_03.jpg, ...).



Đọc video từ webcam

- `vid_capture = cv2.VideoCapture(0)`
- Nếu có nhiều camera, thay đổi 0 thành 1, 2, ...



Ghi Video

- `VideoWriter(filename, apiPreference, fourcc, fps, frameSize[, isColor])`
- `apiPreference`: Mã định danh phụ trợ API
- `fourcc`: Mã 4 ký tự của codec, được sử dụng để nén khung hình
- AVI: ('M','J','P','G')
- MP4: (*'XVID')
- `fps`: Tốc độ khung hình của luồng video đã tạo
- `frame_size`: Kích thước của khung video
- `isColor`: Nếu không phải là 0, bộ mã hóa sẽ mong đợi và mã hóa các khung

```
output = cv2.VideoWriter('Resources/output_video_from_file.avi',  
                        cv2.VideoWriter_fourcc('M','J','P','G'), 20, frame_size)
```

Thank you!



TRƯỜNG ĐẠI HỌC VINH
VINH UNIVERSITY

Nơi tạo dựng tương lai cho tuổi trẻ

