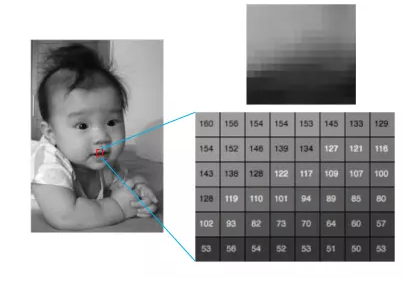
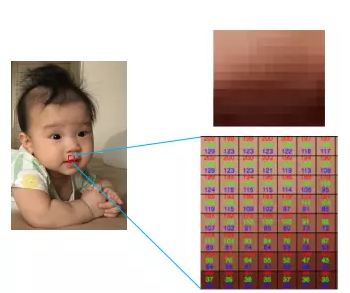
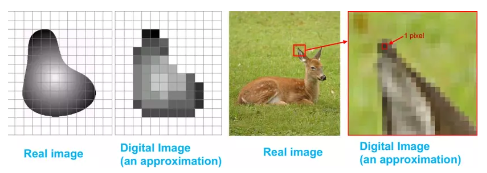
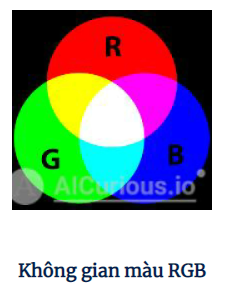
1. ảnh số và điểm ảnh

* điểm ảnh (pixel) có vị trí (x, y) và có độ xám I (x, y).
* ảnh trắng đen:
* 
* ảnh màu: mỗi điểm ảnh sẽ có 3 giá trị tương ứng với độ sáng của các màu đỏ, lục, xanh dương (RGB).
* 
* Ảnh số: " một hình ảnh có thể được định nghĩa là hàm hai chiều, f (x, y), trong đó x và y là tọa độ không gian (mặt phẳng) và biên độ của f tại bất kỳ cặp tọa độ (x, y) nào được gọi là cường độ hoặc mức độ màu xám của hình ảnh tại điểm đó. Khi x, y và các giá trị cường độ của f đều là các đại lượng hữu hạn, rời rạc, chúng ta gọi hình ảnh là hình ảnh kỹ thuật số".
* Ảnh số: " một hình ảnh có thể được định nghĩa là hàm hai chiều, f (x, y), trong đó x và y là tọa độ không gian (mặt phẳng) và biên độ của f tại bất kỳ cặp tọa độ (x, y) nào được gọi là cường độ hoặc mức độ màu xám của hình ảnh tại điểm đó. Khi x, y và các giá trị cường độ của f đều là các đại lượng hữu hạn, rời rạc, chúng ta gọi hình ảnh là hình ảnh kỹ thuật số".
* 
* Điểm ảnh: "hình ảnh kỹ thuật số chứa một số lượng hữu hạn các hàng và cột của các phần tử. Mỗi phần tử được gọi là pixel "

1. Các hệ không gian màu
   1. Không gian màu RGB

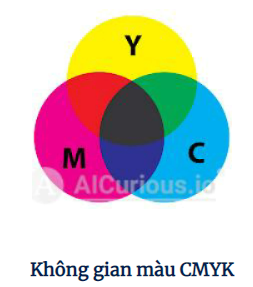


Nguyên lý cơ bản là sử dụng 3 màu sắc cơ bản R (red ), G (green) và B (blue) để biểu diễn tất cả các màu sắc.

Thông thường, trong mô hình 24 bit, trong OpenCV đảo 2 kênh R và B, trở thành BGR. Mỗi màu sử dụng 8 bit để biểu diễn. Giá trị sẽ nằm trong khoảng 0 – 255. Bộ 3 số này biểu diễn cho từng điểm ảnh, mỗi số biểu diễn cho cường độ của một màu.

Số lượng màu tối đa: 255 x 255 x 255 = 16581375.

* 1. Không gian màu CMYK



Không gian màu CMYK chỉ mô hình màu loại trừ, thường dùng trong in ấn. Mô hình này dựa trên cơ sở trộn các chất màu của các màu sau:

C=Cyan trong tiếng Anh có nghĩa là màu xanh lơ hay cánh chả

M=Magenta trong tiếng Anh có nghĩa là màu cánh sen hay hồng sẫm

Y=Yellow trong tiếng Anh có nghĩa là màu vàng

K=Key (trong tiếng Anh nên hiểu theo nghĩa là cái gì đó then chốt hay chủ yếu để ám chỉ màu đen mặc dù màu này có tên tiếng Anh là black do chữ B đã được sử dụng để chỉ màu xanh lam (blue) trong mô hình màu RGB

Nguyên lý làm việc của CMYK là trên cơ sở hấp thụ ánh sáng. Những màu người ta nhìn thấy là phần tử của ánh sáng không bị hấp thụ. Trong CMYK hồng sẫm cộng với vàng sẽ cho màu đỏ, cánh sen cộng với xanh lơ cho màu xanh lam, xanh lơ cộng với vàng sinh ra màu xanh lá cây và tổ hợp của các màu xanh lơ , cánh sen và vàng tạo ra màu đen.

* 1. Không gian màu HSV (HSB)



Không gian màu HSV ( hay HSB) là một cách tự nhiên hơn để mô tả màu sắc, dựa trên 3 số liệu:

H: (Hue) vùng màu

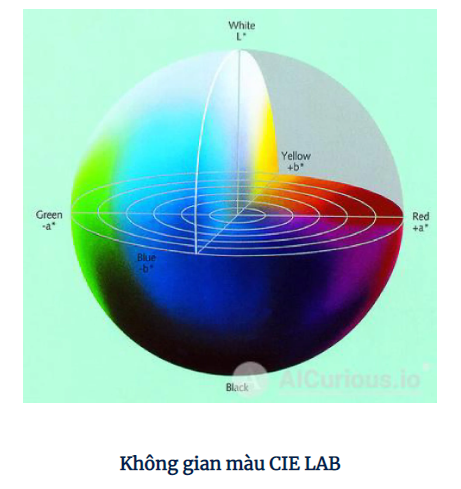
S: (saturation) độ bão hòa màu

B (hay V): (bright hay value) độ sáng

* 1. Không gian màu CLE LAB

Không gain màu CLE LAB là không gian màu có sự đồng đều trong dải màu sắc, do vậy phù hợp để so sánh sự khác biệt giữa màu sắc này với màu sắc khác. Các giá trị Lab mô tả tất cả những màu mà mắt mỗi người bình thường có thể nhìn thấy được.

Lab được xem là một mô hình màu sắc độc lập đối với thiết bị và thường được sử dụng như một cơ sở tham chiếu khi chuyển đổi một màu từ không gian màu này sang một không gian màu khác.



Theo mô hình Lab, tất cả các màu có cùng một độ sáng sẽ nằm trên cùng một mặt phằn có dạng hình tròn theo truc a và b. Màu có giá trị a dường thì ngả đỏ, màu có giá trị a âm thì ngả lục. tương tự b dương ngả vàng và b âm ngả lam. Còn độ sáng của màu thay đổi theo trục dọc.

1. Chuyển đổi giữa các không gian màu trong OpenCV

Vì đang sử dụng python, câu lệnh cv2.cvtColor()

Ví dụ chuyển đổi ảnh img từ hệ màu BRG (hệ màu mặc định trong OpenCV) sang hệ màu HSV:

Hsv = cv2.cvtColor(img, cv2.COLOR\_BRG2HSV)

1. Xử lý ảnh: thuật toán cân bằng histogram ảnh
2. Lý thuyết

Histogram là biểu đồ tần xuất được dùng để thống kê số lần xuất hiện các mức sáng trong ảnh.

Cân bằng histogram là sự điều chỉnh histogram về trạng thái cân bằng, làm cho phân bổ giá trị pixel không bị co cụm lại một khoảng hẹp mà được “kéo dãn” ra.

1. Trong OpenCV

ảnh xám

import cv2  
import numpy as np  
import matplotlib.pyplot as plt  
  
img = cv2.imread("anhthe.jpg", 0)  
  
#hàm tính histogram của một ảnh  
def compute\_hist(img):  
 hist = np.zeros((256,), np.uint8)  
 h, w = img.shape[:2]  
 for i in range(h):  
 for j in range(w):  
 hist[img[i][j]] += 1  
 return hist  
  
# ham can bang anh  
def equal\_hist(hist):  
 cumulator = np.zeros\_like(hist, np.float64)  
 for i in range(len(cumulator)):  
 cumulator[i] = hist[:i].sum()  
 #print(cumulator)  
 new\_hist = (cumulator - cumulator.min())/(cumulator.max() - cumulator.min())\*255  
 new\_hist = np.uint8(new\_hist)  
 return new\_hist  
  
#chay  
hist = compute\_hist(img).ravel()  
new\_hist = equal\_hist(hist)  
  
h, w = img.shape[:2]  
for i in range(h):  
 for j in range(w):  
 img[i, j] = new\_hist[img[i, j]]  
  
fig = plt.figure()  
ax = plt.subplot(121)  
plt.imshow(img, cmap='gray')  
  
plt.subplot(122)  
plt.plot(new\_hist)  
plt.show()

1. Xử lý ảnh
2. Erosion

Gọi là xói mòn.

Mục đích của phương pháp này sẽ giúp:

* Loại bỏ những pixel nhiễu cô lập
* Loại bỏ những pixel nhiễu xung quanh đối tượng giúp cho phần viền (cạnh) của đối tượng trở nên mịn hơn
* Loại bỏ lớp viền (cạnh) của đối tượng giúp đối tượng trở nên nhỏ hơn và đặt những pixel viền đó trở thành lớp nền của đối tượng
* img = cv2.medianBlur(image, 3)

1. Dilation

Dilation hay còn còn là sự giãn nở là toán tử còn lại mà đã nêu ở trên, nó ngược lại với erosion cũng được áp dụng trong các ảnh nhị phân. Mục đích của phương pháp này sẽ giúp:

* Với những hình ảnh bị đứt nét có thể giúp nối liền ảnh lại
* Với những pixel nhiễu xung quanh đối tượng sẽ trở thành viền của đối tượng
* Giúp nổi bật đối tượng trong ảnh hơn
* img = cv2.convertScaleAbs(image, 1.1, 5)