BÀI THU HOẠCH MÔN XỬ LÝ ẢNH

I.1. Viết về biểu diễn mầu HSV (bắt buộc), LAB (có thể bỏ qua), pixel điểm ảnh (bắt buộc

+HSV:

Không gian màu HSV (còn gọi là 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

Ứng dụng điển hình nhất của HSV là trong việc lọc màu. Giả sử chúng ta có 1 bài toán là nhận dạng màu sắc của đèn đường giao thông và đầu vào là ảnh dưới và yêu cầu ở đây là nhận dạng đèn màu xanh lá cây.

Với bài toán này thì hướng tiếp cận đơn giản nhất chính là lọc màu xanh lá cây ra khỏi ảnh. Từ kiến thức ở bên trên thì mỗi một màu sẽ được thể hiện tương đối qua một khoảng giá trị với mỗi hệ màu tương ứng

+LAB

Không gian màu CIE L_a_b* 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 một người bình thường có thể nhìn thấy được.

Lab được xem là một mô hình màu độc lập đối với thiết bị và thường được sử dụng như một cơ sở tham chiếu khi chuyển đổi một màu từ một 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 phẳng có dạng hình tròn theo 2 trục a* và b_. Màu có giá trị a_ dương thì ngả đỏ, màu có giá trị a* âm thì ngả lục. Tương tự b* dương thì ngả vàng và b* âm thì ngả lam. Còn độ sáng của màu thì thay đổi theo trục dọc.

+pixel điểm ảnh

- **pixel điểm ảnh**

Pixel hay còn có tên gọi khác là pel, là cụm từ viết tắt của "**picture element**" với nghĩa tiếng Việt là **điểm ảnh**. Đây thực chất là một khối màu nhỏ hoặc một **điểm ảnh raster, bitmap** với những thông số màu sắc khác nhau. Vì thế, pixel chính là đơn vị **cơ bản nhất** cấu tạo nên một tấm ảnh hiển thị trên màn hình Led.

Một tấm hình mà thông số pixel **càng cao** thì khi phóng to lên sẽ **càng rõ nét** hơn so với những tấm ảnh có độ phân giải thấp. Ngoài ra, để tính độ sắc nét của hình ảnh hiển thị người ta sẽ sử dụng **pixel dimensions**- nghĩa là **độ phân giải** của hình ảnh.

- 1 megapixel = 1 triệu pixel
- Công thưc tính pixcel chính xác nhất là:

  Lượng pixel có chiều dài x lượng pixel ở chiều rộng
- Ý nghĩa của pixel trong thiết kế đồ họa:
  + Xác định dung lượng ảnh
  + Xác định kích thước ảnh
  + Xác định độ nét của hình ảnh

I.2. Ảnh gray, ảnh nhị phân (có thể bỏ qua)

+Ảnh Gray

ảnh xám (GrayScale) đơn giản là một hình ảnh trong đó các màu là các sắc thái của màu xám. Lý do cần phải phân biệt giữa ảnh xám và các ảnh khác nằm ở việc ảnh xám cung cấp ít thông tin hơn cho mỗi pixel. Với ảnh thông thường thì mỗi pixel thường được cung cấp 3 trường thông tin trong khi với ảnh xám chỉ có 1 trường thông tin, việc giảm khối lượng thông tin giúp tăng tốc độ xử lý nhưng vẫn đảm bảo các tác vụ cần thiết

+ảnh binary

Ảnh nhị phân (binary) được xem như là 1 bước giảm thông tin nữa so với ảnh xám. Với ảnh xám thì ngưỡng xám của mỗi pixel được miêu tả trong khoảng [0;255] nhưng với ảnh nhị phân thì mỗi pixel chỉ có giá trị bằng 0 hoặc 255. Tuy thông tin bị giảm khá nhiều nhưng ảnh nhị phân lại thể hiện được rất rõ các yếu tố về góc cạnh và hình dàng của vật thể, điều đó rất có lợi trong các bài toán lọc nhiễu, nhận dạng vật thể (object detection), v..v

I.3. Lân cận cửa sổ (bắt buộc)

+Lân cận điểm ảnh

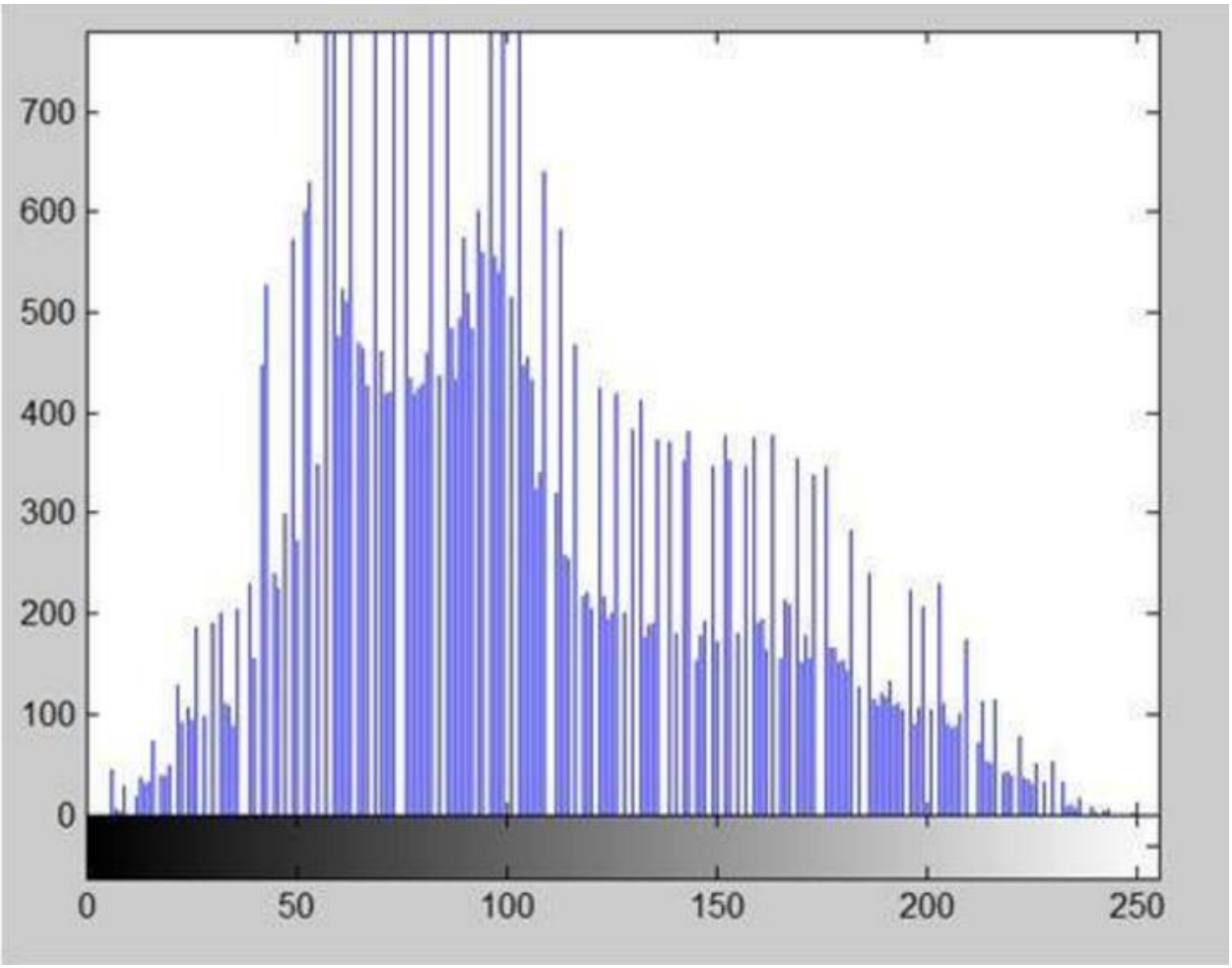
ảnh I (biến bộ nhớ, ma trận ảnh), cao h, rộng w. Giả sử là 1 ảnh grayscale hoặc 1 kênh ảnh của ảnh đa kênh (kênh R, G,B).
Pixel: tọa độ y,x. Sách viết I(x,y), nhưng môi trường lập trình lại là I(y,x).
giá trị xám g=I(x,y).
Lân cận cửa sổ (2d+1)x(2d+1) tâm tại x,y.

Có (2d+1)*(2d+1) pixel.
Khái niệm 4-lân cận; 8-lân cận lại khác 1 chút.

I.4. Histogram của ảnh gray (bắt buộc), ảnh tối, ảnh tương phản thấp có hình dạng histogram như thế nào (bắt buộc).Mục đích của phép cân bằng histogram của ảnh gray (bắt buộc), ảnh thu được sau phép cân bằng có histogram so với histogram của ảnh gốc như thế nào
+ Histogram của ảnh gray

- Histogram là một đồ thị. Histogram thể hiện tần số của mọi thứ. Thông thường Histogram có các thanh đại diện cho tần số xuất hiện của dữ liệu hay nói cách khác là Histogram hiển thị sự phân bố tần số của một tập dữ liệu.
- Một Histogram có hai trục: trục x và trục y.
+ Trục x chứa các điểm có sự phân bố tần số.
+ Trục y chứa tần số.

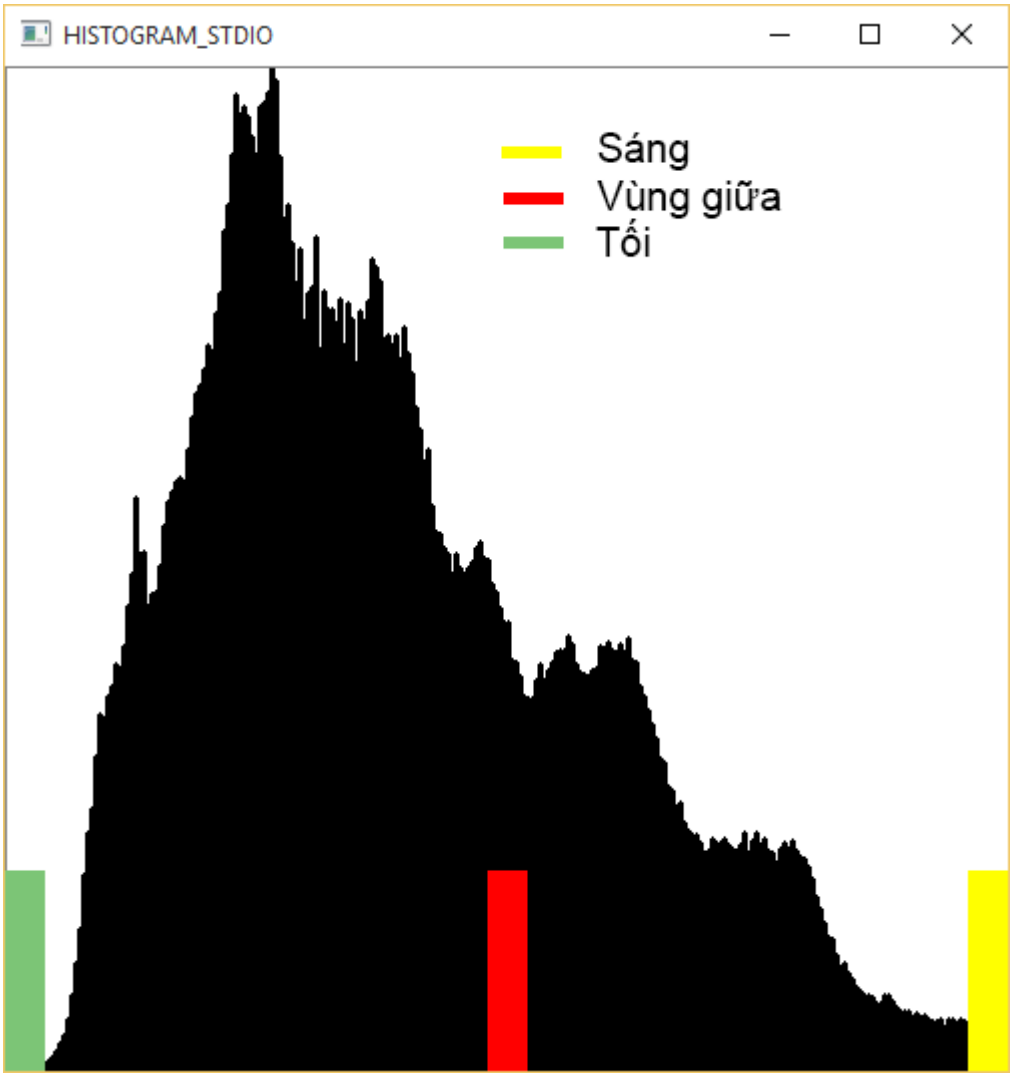
Histogram của một hình ảnh cũng giống như histograms khác cũng cho thấy sự phân bố tần số. Nhưng biểu đồ hình ảnh hiển thị tần số của các giá trị cường độ điểm ảnh. Trong một biểu đồ hình ảnh, trục x hiển thị cường độ mức xám và trục y hiển thị tần số của cường độ.



- Trục tung (Oy) biểu diễn số lượng điểm ảnh (Pixel) của mức xám.
- Trục hoành (Ox) biểu diễn mức xám.
- Giá trị lớn nhất của trục hoành chính là số lượng điểm ảnh (Pixel) có trong một bức ảnh.
- Với ảnh màu như RGB thì có tới 3 biểu đồ Histogram thể hiện từng kênh màu.
- Một biểu đồ tốt à biểu đồ có số lượng điểm ảnh nhiều nhất ở vùng giữa (Độ sáng trung bình) và ít dần ra 2 vùng sáng tối (Ngọn núi).

- Ứng dụng của Histogram: dùng để phân tích hình ảnh, cân bằng một hình ảnh, tạo ngưỡng cho ảnh. Chúng ta có thể dự đoán về một hình ảnh bằng cách chỉ nhìn vào biểu đồ của nó.

+ ảnh tối



- Ảnh tối là ảnh có tập trung quá nhiều điểm ảnh bên vùng tối.
- Ảnh sáng là ảnh có tập trung quá nhiều điểm ảnh bên vùng sáng.
- Ảnh có độ tương phản cao là ảnh có điểm ảnh tập trung nhiều ở 2 vùng sáng và tối và tập trung ít ở vùng giữa.
- Ảnh có độ tương phản thấp là ảnh có điểm ảnh tập trung nhiều ở vùng giữa và tập trung rất ít ở vùng hai vùng sáng và tối.

+ảnh tương phản thấp

Ảnh đen trắng(nhị phân), trắng tuyệt đối, size ảnh mxn thì mảng histogram có tập giá trị xác định ?

Đ/s: hist[255]=m*n, hist[g]=0 với mọi g<255

+Ảnh đen trắng, đen tuyệt đối, size ảnh mxn thì mảng histogram có tập giá trị xác định ?

Đ/s: hist[0]=m*n, hist[g]=0 với mọi g=1,2,..,255

+Ảnh grayscale rất tối thì khi vẽ biểu đồ histogram, thể hiện thế nào?
Đ/s: histogram dồn về vùng có độ xám thấp.
+Ảnh grayscale độ tương phản thấp thì khi vẽ biểu đồ histogram, thể hiện thế nào?
Đ/s: vùng mức xám có tần xuất (số lần xuất hiện nhiều) là hẹp.
+Ảnh grayscale độ tương phản cao thì khi vẽ biểu đồ histogram, thể hiện thế nào?
Đ/s: vùng mức xám có tần xuất (số lần xuất hiện đồng đều) là rộng.

+Mục đích của phép cân bằng histogram của ảnh gray (bắt buộc),

- Cân bằng sáng thường được dùng ở bước **tiền xử lý**.
- Nhằm giảm sự ảnh hưởng do chiếu sáng (chói), thiếu ánh sáng (ảnh tối), …
- Ta có thể hiểu cân bằng sáng giúp ta "chuẩn hóa" ảnh đầu vào trước khi tiến hành xử lý.
- Các giải thuật xử lý ảnh thường nhạy cảm với ánh sáng, cùng nội dung ảnh nhưng với các điều kiện ánh sáng khác nhau có thể làm sai lệch kết quả xử lý (giả sử trong bài toán phát hiện đối tượng, bài toán nhận dạng, bài toán đếm đối tượng, …). Do đó, cân bằng sáng ở bước tiền xử lý là một trong những cách giúp làm giảm các ảnh hưởng này.

I.5 Điểm biên của ảnh gray, giá trị gradient của một điểm ảnh và phép tìm kiếm ảnh nhị phân là ảnh biên của ảnh gray dựa trên ma trận gradient của ảnh (bắt buộc).

+I ảnh grayscale. Có 2 cách xác định ảnh nhị phân là ảnh biên của I: sử dụng phương pháp gradient và phương pháp toán tử hình thái. Trong đó phương pháp gradient là thông dụng nhất.

+ cách xác định ma trận gradient (2 chiều, cùng size với ảnh I) của ảnh I theo Sobel.

+ Cấp phát 1 ma trận số thực 2 chiều Igradient cùng size với I.
Duyệt trên mỗi điểm ảnh I(y,x) của I:
+ Tại mỗi điểm ảnh I(y,x) tính giá trị các đạo hàm riêng theo y và theo x sử dụng ma trận
trọng số Sobel.
+ Tính tổng bình phương 2 giá trị đạo hàm riêng tính được.

+ Khai căn bậc 2 giá trị này.
+ Gán giá trị tính được cho ma trận Iradient mong muốn tại điểm (y,x): Igradien(y,x)=giá trị vừa tính trên.

II. Tham số các hàm thư viện của opencv

II.1. Giải thích tham số của hàm cân bằng histogram của opencv

Igray_eq = cv2.equalizeHist(Igray)

- Tham số truyền vào chỉ là 1 ảnh xám.
- ảnh sau khi cân bằng có histogram đều hơn, các mức xám xuất hiện và khá tương đồng về số lượng. Do đó ảnh sáng hơn, và trực giác cho thấy độ tương phản cao hơn ảnh gốc. Tuy nhiên có thể có những ảnh mà sau khi cân bằng bị biến dạng

II.2. Giải thích tham số của hàm xác định ảnh biên của ảnh gray sử dụng thuật toán Canny (bắt buộc).

cv2.Canny(image, threshold1, threshold2, apertureSize, L2gradient)

Các thông số cần thiết là:

- image: Nguồn / Hình ảnh đầu vào của mảng n chiều.
- threshold1: Đây là giá trị ngưỡng Cao của gradient cường độ.
- threshold2: Đây là giá trị ngưỡng Thấp của gradient cường độ.

Các thông số tùy chọn là:

- apertureSize:Thứ tự của Kernel (ma trận) cho bộ lọc Sobel. Giá trị mặc định của nó là (3 x 3) và giá trị của nó phải là số lẻ trong khoảng từ 3 đến 7. Nó được sử dụng để tìm độ dốc hình ảnh. Bộ lọc được sử dụng để làm mịn và làm sắc nét hình ảnh.
- L2gradient:Điều này xác định phương trình để tìm độ lớn của gradient. L2gradient thuộc loại boolean và giá trị mặc định của nó là False.

II.3. Giải thích tham số của hàm nhị phân của ảnh gray sử dụng thuật toán Otsu (bắt buộc).

cv2.THRESH_OTSU: Sử dụng thuật toán Otsu để xác định giá trị ngưỡng.

cv2.threshold(Ig,0,255, cv2.THRESH_OTSU)

cv::threshold (InputArray src, double thresh, double maxval, int type)

Hàm sử dụng là **threshold** ,

- InputArray src tiên là 1 ảnh xám,
- double thresh giá trị ngưỡng,
- double maxval là giá trị được gán nếu giá pixel lớn hơn giá trị ngưỡng,
- int type là loại phân ngưỡng Cv2.THRESH_OTSU: Sử dụng thuật toán Otsu để xác định giá trị ngưỡng.

II.4. Giải thích tham số hàm xác định histogram của ảnh gray.
cv2.calcHist

hist = cv2.calcHist([Igray],[0],None,[256],[0,256])

cv2.calcHist(images, channels, mask, histSize, ranges)

- **images: :** Đây là hình ảnh dưới dạng 1 danh sách [img]
- **channels:** Danh sách các chỉ mục. Để tính toán biểu đồ của một hình ảnh xám, danh sách sẽ là[0]. Để tính toán biểu đồ cho cả ba kênh đỏ, lục và lam, danh sách kênh sẽ là[0 , 1 , 2]
- **mask:** biểu đồ sẽ chỉ được tính cho các *pixel bị che*. Nếu không có mask hoặc không muốn áp dụng, chúng tôi chỉ có thể cung cấp giá trị là None
- **histSize:** This is the number of bins we want to use when computing a histogram. Again, this is a list, one for each channel we are computing a histogram for. The bin sizes do not all have to be the same. Here is an example of 32 bins for each channel: [32, 32, 32]
- **ranges:** Phạm vi giá trị pixel có thể có. Thông thường, đây là *[0, 256]*

II.5. Giải thích tham số các hàm lọc trung bình cộng (bắt buộc), lọc trung vị (bắt buộc) và lọc tổng quát của ảnh.
Tham số: kích thước cửa sổ lân cận của điểm ảnh:3x3,5x5...
+ Lọc trung bình:
Trung bình cộng: từ 1 ảnh gray tạo ra 1 ảnh gray mới, trơn hơn (smooth, blur)
Duyệt các điểm ảnh của ảnh I
Với mỗi điểm ảnh, xét lân cận (2d+1)x(2d+1) (d cho trước), có thể có nh vị trí không đủ điểm (mép ảnh) vẫn tính là (2d+1)x(2d+1) điểm trong lân cận
Tính giá trị trung bình cộng, lấy phần nguyên, gán vào cùng vị trí tâm lân cận cho ảnh mới với mức xám này.
Chú ý: có thể xuất hiện mức xám không có trong ảnh gốc.

Cú pháp: *cv2.blur (src, ksize [, dst [, anchor [, borderType]]])*
Các thông số:

- ***src:*** *Là hình ảnh được làm mờ.*
- ***ksize:*** *Một bộ giá trị đại diện cho kích thước hạt nhân làm mờ.*
- ***dst:*** *Là hình ảnh đầu ra có cùng kích thước và kiểu với src.*
- ***anchor:*** *Nó là một biến kiểu số nguyên đại diện cho điểm neo và giá trị mặc định của nó. Điểm là (-1, -1) có nghĩa là mỏ neo nằm ở tâm hạt nhân.*
- ***borderType:*** *Nó mô tả loại đường viền sẽ được thêm vào. Nó được xác định bởi các cờ như* ***cv2.BORDER_CONSTANT*** *,* ***cv2.BORDER_REFLECT*** *, v.v.*
- ***Giá trị*** *trả về* ***:*** *Nó trả về một hình ảnh.*

+ Lọc trung vị: median filter.
Duyệt các điểm ảnh của ảnh I
Với mỗi điểm ảnh, xét lân cận (2d+1)x(2d+1) (d cho trước), có thể có nh vị trí không đủ điểm (mép ảnh) vẫn tính là (2d+1)x(2d+1) điểm trong lân cận
Tính giá trị median của các giá trị mức xám trong lân cận, gán vào cùng vị trí tâm lân cận cho ảnh mới với mức xám này.
Chú ý: Mức xám được gán cho ảnh mới vẫn thuộc tập giá trị mức xám của ảnh gốc.

```
 img = cv2.imread('I04.jpg') # Load image
img_median = cv2.medianBlur(img, 7) # Add median filter to image
 cv2.imshow('img', img_median) # Display img with median filter
```

Cú pháp: *cv2.blur (src, ksize)*
Các thông số:

- ***src:*** *Là hình ảnh*
- ***ksize:*** *giá tri int nhưng phải lẻ*

+ lọc tổng quát của ảnh.

```
I1 = cv2.filter2D(I,-1,matran_trongso)
```

II.6. Giải thích tham số hàm xác định ma trận gradient theo hướng x và hướng y sử dụng thuật toán Sobel.

```
Ie_x = cv.Sobel(Im, cv.CV_64F, 1, 0, 3)

Ie_y = cv.Sobel(Im, cv.CV_64F, 0, 1, 3)

Ie = np.sqrt(Ie_x**2 + Ie_y**2)

cv.imshow('Image Ie', Ie)
```

Sobel(src, dst, ddepth, dx, dy)

Phương thức này chấp nhận các tham số sau:

- **src** - Một đối tượng của lớp **Mat** đại diện cho ảnh nguồn (đầu vào).
- **dst** - Một đối tượng của lớp **Mat** đại diện cho hình ảnh đích (đầu ra).
- **ddepth** - Một biến số nguyên đại diện cho độ sâu của hình ảnh (-1)
- **dx** - Một biến số nguyên biểu diễn đạo hàm x. (0 hoặc 1)
- **dy** - Một biến số nguyên đại diện cho đạo hàm y. (0 hoặc 1)

II.7. Giải thích các tham số của acsc lệnh xác định contour của ảnh nhị phân, lệnh vẽ các contour lên ảnh (bắt buộc)

```
Contours, hierarchy = cv.findContours(Im, cv.RETR_TREE,
cv.CHAIN_APPROX_SIMPLE) cv.drawContours(I, Contours, -1, (0, 0, 255), 1)
cv.imshow('Image', I)
```

cv2.findContours(src, contour_retrieval, contours_approximation)

Thông số

Bạn cần truyền ba tham số cho phương thức findContours ()

- src: Hình ảnh đầu vào của mảng n chiều (n = 2,3) nhưng ưu tiên hình ảnh nhị phân 2 mờ để có kết quả tốt hơn.
- contour_retrieval:Đây là chế độ truy xuất đường viền. Các giá trị có thể có là:
  a) cv2.RETR_TREE
  b) cv2.RETR_EXTERNAL

c) cv2.RETR_LIST
d) cv2.RETR_CCOMP, v.v.

contours_approximation:Đây là phương pháp xấp xỉ đường viền. Các giá trị có thể có là:
a) cv2.CHAIN_APPROX_NONE
b) cv2.CHAIN_APPROX_SIMPLE

III. Bài tập

1.

```
import cv2
import numpy as np

def tinh(name_img):
    img = cv2.imread(name_img)
    # cv2.imshow("Image", img)

    print("max kenh R:", np.max(img[:, :, 2]))
    print("min kenh R:", np.min(img[:, :, 2]))
    print("trung binh kenh R:",np.mean(img[:, :, 2]))
    print("Do lech chuan kenh R:",np.std(img[:, :, 2]))

    print("max kenh G:", np.max(img[:, :, 1]))
    print("min kenh G:", np.min(img[:, :, 1]))
    print("trung binh kenh G:",np.mean(img[:, :, 1]))
    print("Do lech chuan kenh G :",np.std(img[:, :, 1]))

    print("max kenh B:", np.max(img[:, :, 0]))
    print("min kenh B:", np.min(img[:, :, 0]))
    print("trung binh kenh B:",np.mean(img[:, :, 0]))
    print("Do lech chuan kenh B :",np.std(img[:, :, 0]))
    cv2.waitKey()

if __name__ == "__main__":
    file_img = "./anhthi/dark.jpg"
    tinh(file_img)
```

2.

```
import os
import cv2
FJoin = os.path.join
def GetFiles(path):
    file_list = []
    for dir, subdirs, files in os.walk(path):
        file_list.extend([FJoin(dir, f) for f in files])
```

```
    return file_list

if __name__ == "__main__":
    files = GetFiles(os.path.expanduser("./test"))
    for file in files:
        try:
            img = cv2.imread(file)
            img = cv2.resize(img, (500, 300))
            print(img.shape[0])
            print(img.shape[1])
            # cv2.imshow("aaaa", img)
            cv2.waitKey()
        except:
            print("Khong doc duoc anh ", file)
```

3.

```
import cv2
import numpy as np

I = cv2.imread("./anhthi/Traffic.jpg")

# chuyển ảnh qua gray
Ig = cv2.cvtColor(I, cv2.COLOR_BGR2GRAY)
# print(I.shape)
# print(Ig.shape)


# chuyển về ma trận ảnh vê 1 chiều
x = I.reshape(-1)
y = Ig.reshape(-1)

print(x)
print(y)

# cv2.imshow("Image", Ig)
cv2.waitKey()
```