ĐỀ CƯƠNG ÔN TẬP

Một số bài tập có lời giải để tham khảo, cần chỉnh sửa tiếp.

1 Cân bằng histogram của kênh S của ảnh Ihsv.

```
img_equal_hist = cv2.equalizeHist(S)
f, axes = plt.subplots(2,2, figsize=(30,20))
axes[0, 0].imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
axes[0, 0].set_title('origin')
axes[0, 1].imshow(cv2.cvtColor(img_equal_hist, cv2.COLOR_BGR2RGB))
axes[0, 1].set_title('hist equal')
axes[1, 0].hist(img.flatten(), 256, [0,256])
axes[1, 1].hist(img_equal_hist.flatten(), 256, [0,256])
plt.show()
plt.close()
```

2 Chuyển ảnh Igray sang ảnh nhị phân Ib với ngưỡng Otsu.

```
ret, Ib = cv2.threshold(Igray, 0, 255, cv2.THRESH_OTSU)
cv2.imshow("anh nhi phan theo otsu", Ib)
```

3 Chuyển ảnh mầu Img sang ảnh đa cấp xám

(grayscale) theo công thức xác định mức độ xám từ tổ hợp các thành phần mầu (r, g, b) theo tỷ lệ (0.39, 0.5, 0.11), được ma trận ảnh Ig.

Hiển thị ảnh Ig.

```
h,w,k = img.shape
Igray = np.zeros((h, w), dtype='uint8')
b,g,r = cv2.split(img)
Igray[:, :] = 0.39 * r + 0.5 * g + 0.11 * b
```

4 Chuyển ảnh mầu I sang ảnh đa cấp xám

(grayscale) theo công thức xác định mức độ xám từ tổ hợp các thành phần mầu (r, g, b) theo tỷ lệ (0.39, 0.5, 0.11), được ma trận ảnh Ig.

Hiển thị ảnh Ig.

Chuyển ảnh Ig sang ảnh nhị phân Ib với ngưỡng Otsu.

5 Chuyển ảnh mầu I sang ảnh đa cấp xám (grayscale)

theo công thức xác định mức độ xám từ tổ hợp các thành phần mầu (r, g, b) theo tỷ lệ (0.39, 0.5, 0.11), được ma trận ảnh Ig.

Hiển thị ảnh Ig.

Xác định mức xám lớn nhất của ảnh Ig.

6 Chuyển ảnh mầu I sang ảnh đa cấp xám

(grayscale) theo công thức xác định mức độ xám từ tổ hợp các thành phần mầu (r, g, b) theo tỷ lệ (0.39, 0.5, 0.11), được ma trận ảnh Ig.

Hiển thị ảnh Ig.

Xác định mức xám trung bình của ảnh Ig.

7 Chuyển ảnh mầu I sang ảnh HSV.

Hiển thị kênh S.

Hiển thị các giá trị điểm ảnh trong cửa sổ lân cận 5x5 của điểm ảnh với tọa độ dòng y=10, cột x=20.

```
print(Ihsv[y-2:y+2, x-2:x+2])
```

8 Chuyển ảnh sang biểu diễn HSV được ma trận Ihsv.

Hiển thị kênh H của Ihsv.

Xác định giá trị mức sáng lớn nhất của kênh S của ảnh Ihsv.

9 Chuyển ảnh sang biểu diễn HSV được ma trận Ihsv.

Hiển thị kênh H của Ihsv.

Xác định giá trị mức sáng trung bình của kênh S của ảnh Ihsv.

```
Ihsv = cv2.cvtColor(I,cv2.COLOR_BGR2HSV)
cv2.imshow('kenh H',Ihsv[:,:,0])
```

10 Chuyển ảnh sang biểu diễn HSV được ma trận Ihsv.

Hiển thị kênh S của Ihsv.

Xác định giá trị mức sáng lớn nhất của kênh V của ảnh Ihsv.

11 Chuyển ảnh sang biểu diễn HSV được ma trận Ihsv.

Hiển thị kênh V của Ihsv.

Xác định giá trị mức sáng lớn nhất và nhỏ nhất của kênh S của ảnh Ihsv.

12 Chuyển đổi ảnh I sang ảnh HSV được ma trận ảnh Ihsv.

Hiển thị kênh S của ảnh Ihsv.

13 Chuyển kênh S của ảnh sang ảnh nhị phân Ib với ngưỡng Otsu.

Hiển thị ảnh nhị phân Ib.

Hiển thị ảnh I

14 Hiển thị ảnh I và giá trị tỉ lệ giữa độ cao và độ rộng của ảnh.

15 Hiển thị các độ xám của của cửa sổ lân cận 5x5 của pixel

có tọa độ dòng y=109, cột x=130 của ảnh Ig.

```
Ig = cv2.cvtColor(I, cv2.COLOR_BGR2GRAY)
cv2.imshow("Image Gray", Ig)
print(Ig[y-2:y+2,x-2:x+2)
```

16 Hiển thị kênh B của ảnh I.

```
b,g,r = cv2.split(img)
```

17 Hiển thị kênh H của Ihsv.

Xác định giá trị mức sáng lớn nhất của kênh S của ảnh Ihsv.

18 Hiển thị kênh S của Ihsv.

Xác định giá trị mức sáng trung bình của kênh V của ảnh Ihsv.

19 Hiển thị tỷ lệ giữa giá trị độ cao và độ rộng của ảnh I.

20 Hiệu chỉnh lại ảnh I với size mới là độ cao 256,

ảnh giữ nguyên tỷ lệ so với ảnh gốc, được ảnh mới .

Hiển thị ảnh.

```
np.reshape
```

21 23 Hiệu chỉnh lại ảnh I với size mới

là độ cao 256, độ rộng 256 được ảnh mới

Hiển thị ảnh .

```
h = I.shape[0]
w = I.shape[1]
print(h/w)
```

```
np.reshape
```

22 Kiểm tra pixel có tọa độ dòng y=100, cột x=300 có là điểm biên

của ảnh Ig theo phép dò biên Canny không?

```
Icanny = cv2.Canny(image=I, threshold1=100, threshold2=200)
cv2.imshow('Icanny',Icanny)
Icanny[100,300]==1
```

23 Làm trơn ảnh Ig theo bộ lọc trung bình cộng với lân cận cửa sổ

kích thước 5x5 thu được ảnh Im.

```
kernel_averaging_5_5 = np.ones((5, 5), np.float32)/25
smooth_image_f2D_5_5 = cv2.filter2D(image, -1, kernel_averaging_5_5)
```

24 Làm trơn ảnh Ig theo bộ lọc trung bình cộng với lân cận cửa sổ kích thước 5x5 thu được ảnh Im.

25 Làm trơn ảnh kênh S của ảnh theo bộ lọc trung bình cộng

với lân cận cửa sổ kích thước 5x5 thu được ảnh Im. Hiển thị ảnh kết quả Im.

26 Làm trơn ảnh kênh S của Ihsv theo bộ lọc median,

kích thước cửa sổ lân cận là 5x5 được ảnh Is. Hiển thị ảnh Is.

```
Im = cv2.medianBlur(Ig, 5)
cv2.imshow("loc median", Im)
```

27 Hiển thị các giá trị mức xám

trong lân cận 5x5 của điểm ảnh có tọa độ dòng y=9, cột x=11.

```
print(Ig[y-2:y+2,x-2:x+2]
```

28 Làm trơn ảnh kênh V của Ihsv theo bộ lọc trung bình cộng,

Kích thước cửa sổ lân cận là 3x3 được ảnh Iv. Hiển thị ảnh Iv.

29 Làm trơn kênh S của ảnh Ihsv

với bộ lọc median kích thước cửa sổ 3x

Biến đổi ngược ảnh Ihsv về biểu diễn mầu RGB được ảnh I.

Hiển thị ảnh I3.

30 Lấy biên của ảnh Ig theo phương pháp Canny

được ảnh biên Ie là ảnh nhị phân nền đen.

Hiển thị ảnh Ie.

Kiểm tra các điểm ảnh của pixel có tọa độ dòng y=109, cột x=130 có phải là điểm biên của Ig theo phương pháp dò biên Canny.

```
Icanny = cv2.Canny(image=I, threshold1=100, threshold2=200)
cv2.imshow('Icanny',Icanny)
```

31 Lấy biên của ảnh Ig theo phương pháp Canny được ảnh biên Ie

là ảnh nhị phân nền đen.

Kiểm tra các điểm ảnh của pixel có tọa độ dòng y=100, cột x=120 có phải là điểm biên của Ig theo phương pháp dò biên Canny.

32 Lấy biên của ảnh Ig theo phương pháp Canny được ảnh biên Ie

là ảnh nhị phân nền đen.

Kiểm tra pixel có tọa độ dòng y=160, cột x=326 có là điểm biên của ảnh Ig theo phép dò biên Canny không?

33 Nhị phân ảnh Ig theo ngưỡng Otsu

được ảnh nhị phân nền đen được ảnh Ib.

```
ret, Ib = cv2.threshold(Ig, 90, 255, cv2.THRESH_OTSU)
cv2.imshow("anh nhi phan theo otsu", Ib)
```

34 Nhị phân ảnh Ig theo ngưỡng Otsu được ảnh nhị phân nền đen Ib.

Xác định các đường contour của ảnh Ib gần tương tự với đường tròn.

Vẽ các đường contour trên lên ảnh gốc I.

```
thresh, I_bina= cv2.threshold(I_gray,0, 255, cv2.THRESH_OTSU)
I_copy=I.copy()
contours, hierarchy = cv2.findContours(I_bina, cv2.RETR_TREE,
        cv2.CHAIN_APPROX_SIMPLE)
area_cnt = [cv2.contourArea(cnt) for cnt in contours]
max = area_cnt[0]
for i in range(len(area_cnt)):
        if max < area_cnt[i]:
                max = area_cnt[i]
for contour in contours:
        if cv2.contourArea(contour) > (max /(5)):
                cv2.drawContours(I_copy, [contour], -1, (0, 255, 255), 3)
```

```
cv2.imshow('contours',I_copy)
```

35 Nhị phân ảnh Ig theo ngưỡng Otsu được ảnh nhị phân nền đen Ib.

Xác định các đường contour của ảnh Ib.

Vẽ các đường contour trên lên ảnh gốc I.

36 Nhị phân ảnh Ig theo ngưỡng Otsu được ảnh nhị phân nền đen Ib.

Xác đường contour của ảnh Ib có diện tích lớn nhất.

Vẽ đường contour tìm được trên lên ảnh gốc I.

37 Nhị phân hóa ảnh 255- Is (ảnh nghịch đảo) theo ngưỡng Otsu được

ảnh nhị phân Ib.

Xác định đường contour có chu vi lớn nhất của ảnh Ib.

Vẽ đường contour trên ảnh gốc I.

38 Nhị phân hóa ảnh Is theo ngưỡng Otsu được ảnh nhị phân Ib.

Xác định đường contour có chu vi lớn nhất của ảnh Ib.

Vẽ đường contour đó trên ảnh gốc I.

39 Nhị phân hóa ảnh Is theo ngưỡng Otsu được ảnh nhị phân Ib.

Xác định đường contour có chu vi lớn nhất của ảnh Ib.

Vẽ đường contour trên ảnh I.

40 Nhị phân hóa của ảnh 255-Is theo ngưỡng Otsu

được ảnh nhị phân Ib.

Xác định đường contour có chu vi lớn nhất của ảnh Ib.

Vẽ đường contour trên ảnh gốc I và hiển thị ảnh I.

41 Tăng độ sáng của kênh V của ảnh Ihsv

bằng phương pháp cân bằng histogram.

Biến đổi ngược ảnh Ihsv về biểu diễn mầu RGB được ảnh I.

42 Tăng độ sáng của kênh V của ảnh Ihsv

bằng phương pháp giãn mức xám.

Biến đổi ngược ảnh Ihsv về biểu diễn mầu RGB được ảnh I mới.

43 Tăng độ sáng của kênh V của ảnh Ihsv

bằng phương pháp giãn tuyến tính các giá trị mức xám.

Biến đổi ngược ảnh Ihsv về biểu diễn mầu RGB được ảnh I.

Hiển thị lại ảnh I.

44 Tăng độ sáng của kênh V của ảnh Ihsv

bằng phương pháp giãn tuyến tính giá trị mức xám.

Biến đổi ngược ảnh Ihsv về biểu diễn mầu RGB được ảnh I.

Hiển thị lại ảnh I.

45 Xác định biên theo phương pháp Canny của kênh V

của ảnh Ihsv được ảnh nhị phân Ivb.

46 Xác định các contour của ảnh Im và vẽ vị trí các contour lên ảnh gốc I ban đầu.

47 Xác định các đường contour của ảnh Ib, tìm giá trị max_area

là diện tích lớn nhất trong các contour trên.

Vẽ các contours có diện tích > max_area/5.0 lên ảnh gốc I với mầu vàng bgr = (0, 255, 255).

Hiển thị ảnh I.

48 Xác định các đường contour của ảnh Ib,

tìm giá trị max_chuvi là chu vi lớn nhất trong các contour trên.

Vẽ các contours có chu vi > max_chuvi/3.0 lên ảnh gốc I.

49 Xác định các đường contour của ảnh Ib,

tìm giá trị max_cv là chu vi lớn nhất trong các contour trên. V

ẽ các contours có chu vi lớn nhất lên ảnh gốc I với mầu vàng bgr = (0, 255, 255).

Hiển thị ảnh I.

50 Xác định đường contour có chu vi lớn nhất của ảnh Ib.

Vẽ đường contour trên ảnh gốc I.

```
thresh, I_bina= cv2.threshold(I_gray,0, 255, cv2.THRESH_OTSU)
I_copy=I.copy()
contours, hierarchy = cv2.findContours(I_bina, cv2.RETR_TREE,
        cv2.CHAIN_APPROX_SIMPLE)
length_cnt = [cv2.arcLength(cnt,True) for cnt in contours]
max = length _cnt[0]
for i in range(len(length _cnt)):
        if max < length _cnt[i]:
```

```
                max = length_cnt[i]
    for contour in contours:
        if cv2.arcLength (contour) > (max /(5)):
            cv2.drawContours(I_copy, [contour], -1, (0, 255, 255), 3)
    cv2.imshow('contours',I_copy)
```

51 Xác định đường contour có diện tích lớn nhất của ảnh Ib.

Vẽ đường contour trên ảnh gốc I.

```
    thresh, I_bina= cv2.threshold(I_gray,0, 255, cv2.THRESH_OTSU)
    I_copy=I.copy()
    contours, hierarchy = cv2.findContours(I_bina, cv2.RETR_TREE,
    cv2.CHAIN_APPROX_SIMPLE)
    area_cnt = [cv2.contourArea(cnt) for cnt in contours]
    max = area_cnt[0]
    for i in range(len(area_cnt)):
        if max < area_cnt[i]:
            max = area_cnt[i]
    for contour in contours:
        if cv2.contourArea(contour) > (max /(5)):
            cv2.drawContours(I_copy, [contour], -1, (0, 255, 255), 3)
    cv2.imshow('contours',I_copy)
```

52 Xác định đường contour có tỉ lệ giữa chu vi và diện tích là lớn nhất

của ảnh Ib.

Vẽ đường contour trên ảnh gốc I

Hiển thị ảnh

53 Xác định giá trị mức xám nhỏ nhất, lớn nhất và trung bình của ảnh Ig. Hiển thị các giá trị này.

54 Xác định ma trận gradient theo hướng x của Ig

sử dụng toán tử Sobel và hiển thị ma trận kết quả.

```
    sobelx = cv2.Sobel(Im,ddepth=cv2.CV_64F, dx=1, dy=0, ksize=5)
    print(sobelx)
    cv2.imshow('Sobel X', sobelx)
```

55 Xác định ma trận gradient theo hướng y và theo hướng x của Ig

sử dụng toán tử Sobel và hiển thị 2 ma trận kết quả.

56 Xác định và vẽ histogram của kênh S của ảnh Ihsv.

```
hB_ = cv2.calcHist(Ihsv[:, :, 1],[0],None,[256],[0,256])
plt.plot(hB_)
plt.title("hist kenh s ")
plt.show()
```

57 Xác định và vẽ histogram của kênh V của ảnh Ihsv.