Bài thực hành 5

Trần Văn Quyền

MSSV: 19574802010166

Câu 1:

import cv2

import numpy as np

import matplotlib.pyplot as plt

img = cv2.imread('D:\Tran Quyen\Document\Image\kp.png')

gray\_img = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

ret, binary\_img = cv2.threshold(gray\_img, 127, 255, cv2.THRESH\_BINARY)

adaptive\_img = cv2.adaptiveThreshold(gray\_img, 255, cv2.ADAPTIVE\_THRESH\_MEAN\_C, cv2.THRESH\_BINARY, 11, 2)

ret, otsu\_img = cv2.threshold(gray\_img, 0, 255, cv2.THRESH\_BINARY + cv2.THRESH\_OTSU)

plt.subplot(2,2,1), plt.imshow(img)

plt.title('Original Image'), plt.xticks([]), plt.yticks([])

plt.subplot(2,2,2), plt.imshow(binary\_img, cmap='gray')

plt.title('Binary Thresholding'), plt.xticks([]), plt.yticks([])

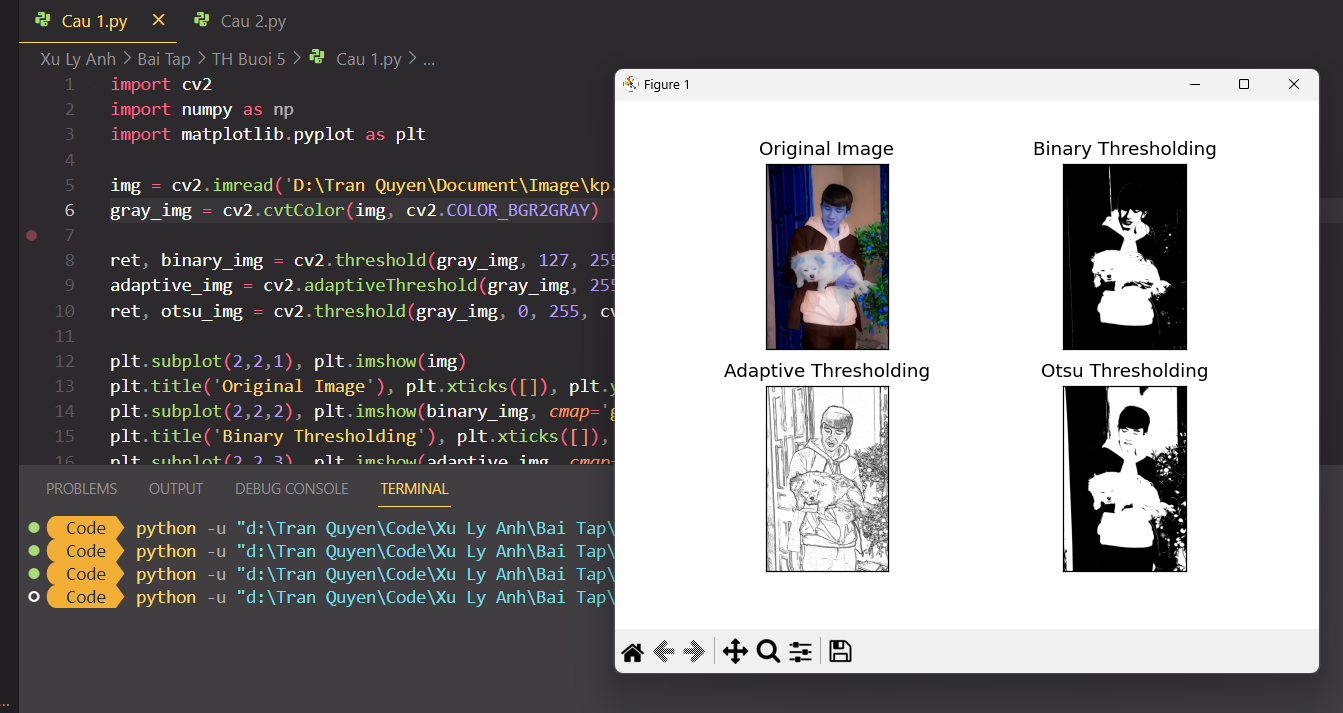
plt.subplot(2,2,3), plt.imshow(adaptive\_img, cmap='gray')

plt.title('Adaptive Thresholding'), plt.xticks([]), plt.yticks([])

plt.subplot(2,2,4), plt.imshow(otsu\_img, cmap='gray')

plt.title('Otsu Thresholding'), plt.xticks([]), plt.yticks([])

plt.show()



Câu 2:

import cv2

import numpy as np

image = cv2.imread('D:\Tran Quyen\Document\Image\kp.png')

img = cv2.resize(image, (400, 400))

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

ret, thresh = cv2.threshold(gray, 127, 255, cv2.THRESH\_BINARY)

contours, hierarchy = cv2.findContours(thresh, cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)

contour\_img = np.zeros\_like(img)

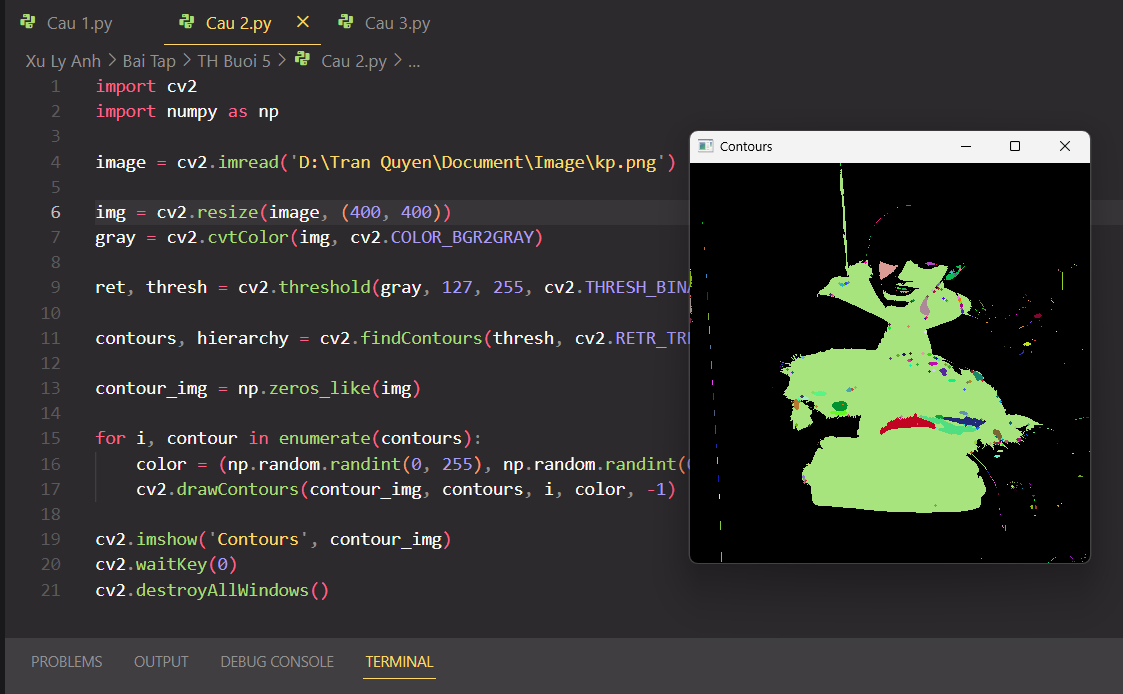
for i, contour in enumerate(contours):

color = (np.random.randint(0, 255), np.random.randint(0, 255), np.random.randint(0, 255))

cv2.drawContours(contour\_img, contours, i, color, -1)

cv2.imshow('Contours', contour\_img)

cv2.waitKey(0)

cv2.destroyAllWindows()

Câu 3:

import cv2

import numpy as np

image = cv2.imread('D:\Tran Quyen\Document\Image\kp.png')

if image is None:

print('Error: Could not load image!')

exit()

img = cv2.resize(image, (400, 500))

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

grad = cv2.morphologyEx(gray, cv2.MORPH\_GRADIENT, np.ones((5,5),np.uint8))

opening = cv2.morphologyEx(gray, cv2.MORPH\_OPEN, np.ones((3,3),np.uint8))

closing = cv2.morphologyEx(gray, cv2.MORPH\_CLOSE, np.ones((5,5),np.uint8))

sure\_bg = cv2.dilate(opening,np.ones((3,3),np.uint8),iterations=2)

dist\_transform = cv2.distanceTransform(opening,cv2.DIST\_L2,5)

ret, sure\_fg = cv2.threshold(dist\_transform,0.7\*dist\_transform.max(),255,0)

sure\_fg = np.uint8(sure\_fg)

unknown = cv2.subtract(sure\_bg,sure\_fg)

ret, markers = cv2.connectedComponents(sure\_fg)

markers = markers+1

markers[unknown==255] = 0

markers = cv2.watershed(img,markers)

img[markers == -1] = [255,0,0]

cv2.imshow('Result', img)

cv2.waitKey(0)

cv2.destroyAllWindows()

