import numpy as np

import cv2

from matplotlib import pyplot as plt

i = int(input('enter option 1.Contour 2.Edge Detection 3.Feature Matching 4.corner Detection'))

if i == 1:

image = cv2.imread('shapes.png')

gray\_image = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

ret, thresh = cv2.threshold(gray\_image, 127, 255, 0)

contours, hierarchy = cv2.findContours(

thresh, cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)

with\_contours = cv2.drawContours(image, contours, -1, (0, 255, 0), 3)

plt.imshow(with\_contours)

plt.show()

elif i == 2:

image = cv2.imread('edge.jpg')

edges = cv2.Canny(image, 100, 200)

plt.imshow(edges)

plt.show()

elif i == 3:

img1 = cv2.imread('messi1.jpg', 0)

img2 = cv2.imread('messi2.jpg', 0)

orb = cv2.ORB\_create()

kp1, des1 = orb.detectAndCompute(img1, None)

kp2, des2 = orb.detectAndCompute(img2, None)

bf = cv2.BFMatcher(cv2.NORM\_HAMMING, crossCheck=True)

matches = bf.match(des1, des2)

matches = sorted(matches, key=lambda x: x.distance)

img3 = cv2.drawMatches(img1, kp1, img2, kp2, matches[:10], None, flags=2)

plt.imshow(img3), plt.show()

else:

img = cv2.imread("chess.png")

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

gray = np.float32(gray)

dst = cv2.cornerHarris(gray, 2, 3, 0.04)

dst = cv2.dilate(dst, None)

img[dst > 0.01\*dst.max()] = [0, 0, 255]

cv2.imshow('dst', img)

if cv2.waitKey(0) & 0xff == 27:

cv2.destroyAllWindows()