ŞEKİL TANIMA VE MERKEZ NOKTASINI BELİRLEME

1- Gerekli Kütüphaneleri dahil ediyoruz.

import cv2  
import imutils

2- Resmi okuyoruz.

image = cv2.imread("im\_shape.png")

3- Resmi yeniden boyutlandırıyoruz ve yükseklik oranını hesaplıyoruz.

resized = imutils.resize(image, width=300)  
ratio = image.shape[0] / float(resized.shape[0])

4- Gri tonlamaya dönüştürüyoruz.

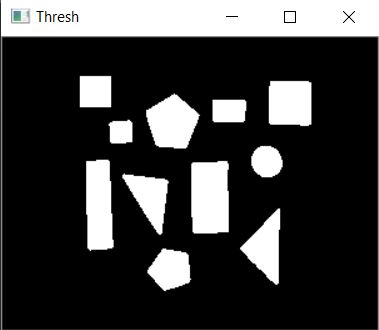
gray = cv2.cvtColor(resized, cv2.COLOR\_BGR2GRAY)

5- Gürültüleri azaltmak için filtre uyguluyoruz.

blurred = cv2.GaussianBlur(gray, (5,5), 0)

6- Şekillerin (renklerin) ayırt edilmesi için eşikleme yapıyoruz.

thresh = cv2.threshold(blurred, 60, 255, cv2.THRESH\_BINARY)[1]



7- Konturları buluyoruz.

contour = cv2.findContours(thresh.copy(), cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)

8- OpenCV 2.4, 3 veya 4'ü kullanıp kullanmadığımıza bağlı olarak uygun tuple değerini alır.

contour = imutils.grab\_contours(contour)

9- Konturlar üzerinde döngü oluşturuyoruz.

for i in contour:

10- şekillerin merkezini buluyoruz ve (x,y) olarak ayırıyoruz.

M = cv2.moments(i)  
ix = int((M["m10"] / M["m00"]) \* ratio)  
iy = int((M["m01"] / M["m00"]) \* ratio)

11- Kontur çevresini hesaplıyoruz.

peri = cv2.arcLength(i, True)

12- Köşe sayısını (kordinatlarını) elde ediyoruz.

approx = cv2.approxPolyDP(i, 0.04 \* peri, True)

13-

Köşe sayısı 3 ise üçgen,

if len(approx) == 3:  
 shape = "Üçgen"

Köşe sayısı 4 ve kenar uzunlukları eşit ise kare, değilse dikdörtgen,

elif len(approx) == 4:  
 (x,y,w,h) = cv2.boundingRect(approx)  
 ar = w / float(h)  
  
 if ar == 1:  
 shape = "Kare"  
 else:  
 shape = "Dikdörtgen"

Köşe sayısı 5 ise beşgen,

elif len(approx) == 5:  
 shape = "Beşgen"

Köşe sayısı bunların dışında ise daire,

else:  
 shape = "Daire"

14- Konturları ve merkez (x, y) koordinatlarını yeniden boyutlandırma oranımızla çarpıyoruz.

i = i.astype("float")  
i\*=ratio  
i = i.astype("int")

15- Konturları çiziyoruz.

cv2.drawContours(image, [i], -1, (0,255,0), 2)

16- Şekil isimlerini yazdırıyoruz.

cv2.putText(image, shape, (ix+5,iy+5), cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (255,255,255), 2)

17- Merkez noktasına siyah renkli daire çiziyoruz.

cv2.circle(image, (ix, iy), 3, (0, 0, 0), -1)

18- Resmi ekranda gösteriyoruz.

cv2.imshow('Window',image)

19- Herhangi bir tuşa basılana kadar resmi ekranda tutuyoruz.

cv2.waitKey(0)

