Un círculo se representa matemáticamente como( x- \_Xcentro)2+(y−ycentro)2=r2dónde(Xcentro,ycentro)es el centro del círculo, yres el radio del círculo. De la ecuación, podemos ver que tenemos 3 parámetros, por lo que necesitamos un acumulador 3D para una transformación grande, lo que sería muy ineficaz. Entonces OpenCV usa un método más complicado, el método de gradiente de Hough , que utiliza la información de gradiente de los bordes.

La función que usamos aquí es cv.HoughCircles() . Tiene muchos argumentos que están bien explicados en la documentación. Entonces vamos directamente al código.

import numpy as np  
import cv2 as cv  
img = cv.imread('resources/opencv-logo-white.png', cv.IMREAD\_GRAYSCALE)  
assert img is not None, "file could not be read, check with os.path.exists()"  
img = cv.medianBlur(img,5)  
cimg = cv.cvtColor(img,cv.COLOR\_GRAY2BGR)  
circles = cv.HoughCircles(img,cv.HOUGH\_GRADIENT,1,20,  
 param1=50,param2=30,minRadius=0,maxRadius=0)  
circles = np.uint16(np.around(circles))  
for i in circles[0,:]:  
 # draw the outer circle  
 cv.circle(cimg,(i[0],i[1]),i[2],(0,255,0),2)  
 # draw the center of the circle  
 cv.circle(cimg,(i[0],i[1]),2,(0,0,255),3)  
cv.imshow('detected circles',cimg)  
cv.waitKey(0)  
cv.destroyAllWindows()

qt.qpa.plugin: Could not find the Qt platform plugin "wayland" in "/home/dread/miniconda3/envs/tf/lib/python3.9/site-packages/cv2/qt/plugins"