

Nome: Henrique Vital Carvalho RA: 22.119.078-8 Nome: Antonio Gustavo da Silva RA: 22.119.001-0

```
import math

import numpy as np
import cv2
import matplotlib.pyplot as plt

#Importa e converte para RGB
img = cv2.imread('./destro.jpeg')
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
img_copy = img.copy()

#Convertendo para preto e branco (RGB -> Gray Scale -> BW)
img_gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
a = img_gray.max()
_, thresh = cv2.threshold(img_gray, a/2*1.7, a, cv2.THRESH_BINARY_INV)

tamanhoKernel = 5
kernel = np.ones((tamanhoKernel, tamanhoKernel), np.uint8)
thresh_open = cv2.morphologyEx(thresh, cv2.MORPH_OPEN, kernel)

#Filtro de ruído (bluring)
img_blur = cv2.blur(img_gray, ksize=(tamanhoKernel, tamanhoKernel))

img_open = cv2.morphologyEx(img_blur, cv2.MORPH_OPEN, kernel)
img_close = cv2.morphologyEx(img_open, cv2.MORPH_CLOSE, kernel)

# NOVO

img_open = cv2.morphologyEx(img_open, cv2.MORPH_OPEN, kernel)
img_close = cv2.morphologyEx(img_close, cv2.MORPH_CLOSE, kernel)

# fi novo

# Detecção borda com Canny (com blurry)
edges_gray = cv2.Canny(image=img_close, threshold1=a/2,
threshold2=a/2)

#_, thresh_final = cv2.threshold(edges_gray, a/2*1.7,
a, cv2.THRESH_BINARY_INV)
_, thresh_final = cv2.threshold(edges_gray, a/2, a, cv2.THRESH_BINARY)

# contorno
```

```

contours, hierarchy = cv2.findContours(
    image = thresh_final,
    mode = cv2.RETR_TREE,
    method = cv2.CHAIN_APPROX_SIMPLE)
contours = sorted(contours, key = cv2.contourArea, reverse = True)
final = cv2.drawContours(img_copy, contours, contourIdx = -1,
    color = (255, 0, 0), thickness = 2)

```

#plot imagens

```

imagens = [img, img_gray, img_blur, img_open, img_close, edges_gray,
thresh,thresh_open,final]
formatoX = math.ceil(len(imagens)**.5)
if (formatoX**2-len(imagens))>formatoX:
    formatoY = formatoX-1
else:
    formatoY = formatoX
for i in range(len(imagens)):
    plt.subplot(formatoY, formatoX, i + 1)
    plt.imshow(imagens[i], 'gray')
    plt.xticks([],plt.yticks([]))
plt.show()

plt.imshow(imagens[len(imagens) - 1], 'gray')
plt.show()

```



