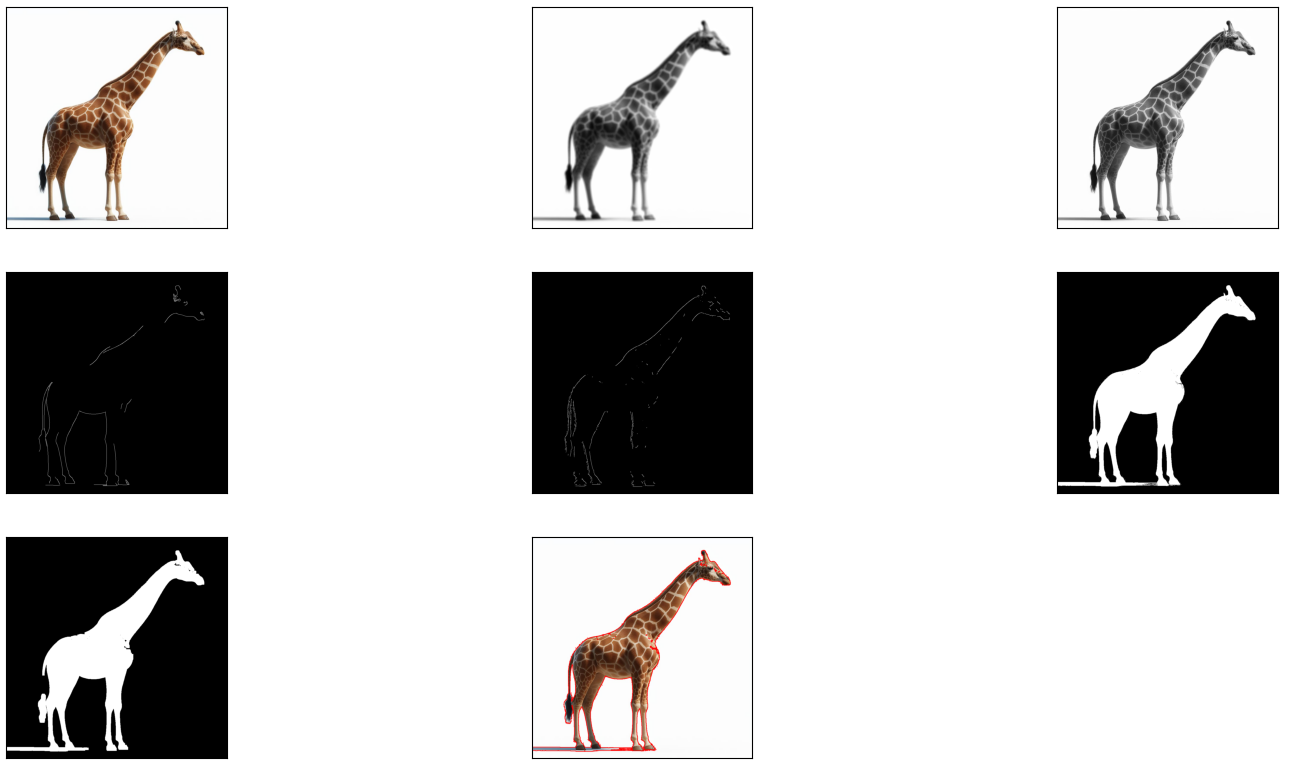
Nomes:

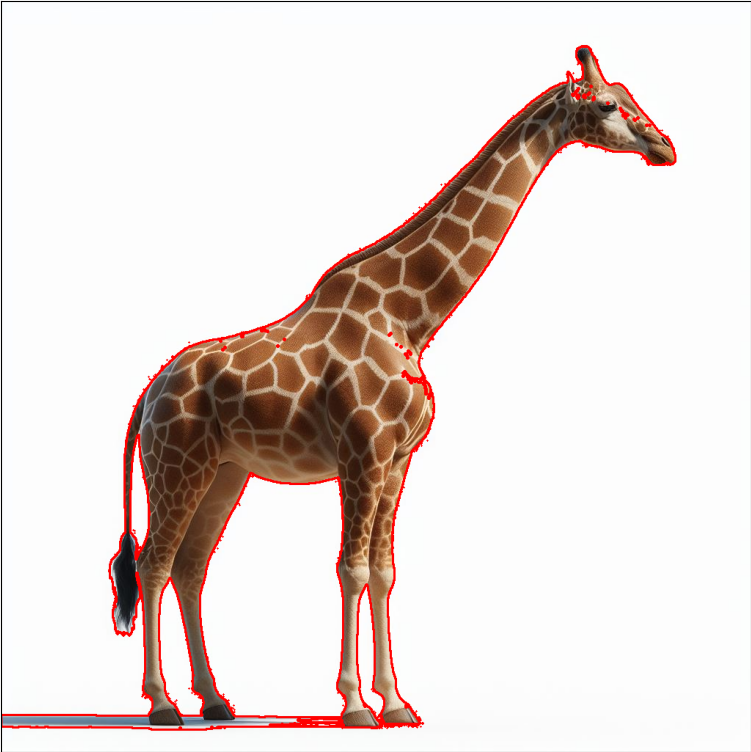
Henrique Kenji Konishi 22121030-5

Erico Medeiros Correia da Silva 22221051-0

Fernando Chan Lin

**GIRAFA.jpeg**





#pip install opencv-python

import math

import numpy as np

import cv2

import matplotlib.pyplot as plt

#Importa e converta para RGB

img = cv2.imread('./GIRAFA.jpeg')

img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

#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\*1.9, a,cv2.THRESH\_BINARY\_INV)

tamanhoKernel = 11

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))

# Detecção borda com Canny (sem blurry)

edges\_gray = cv2.Canny(image=img\_gray, threshold1=a/0.7, threshold2=a/0.3)

# Detecção borda com Canny (com blurry)

edges\_blur = cv2.Canny(image=img\_blur, threshold1=a/2, threshold2=a/2)

# contorno

contours, hierarchy = cv2.findContours(

                                   image = thresh,

                                   mode = cv2.RETR\_TREE,

                                   method = cv2.CHAIN\_APPROX\_SIMPLE)

contours = sorted(contours, key = cv2.contourArea, reverse = True)

img\_copy = img.copy()

final = cv2.drawContours(img\_copy, contours, contourIdx = -1,

                         color = (255, 0, 0), thickness = 2)

#plot imagens

# imagens = [img,img\_blur,img\_gray,edges\_gray,edges\_blur,thresh,thresh\_open,final]

imagens = [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()

**SATELITE.jpeg**