**ATIVIDADE 5**

**ATENÇÃO:**

1. **Esta Atividade deverá ser feita em GRUPO DE PELO MENOS 06 ALUNOS E DE NO MÁXIMO 08 ALUNOS embora a entrega deverá ser feita INDIVIDUALMENTE .**
2. **Atividades feitas individualmente ou entregues com atraso NÃO SERÃO CONSIDERADAS.**
3. **Todas as respostas devem ser escritas aqui no espaço destacado em** **COR AZUL abaixo.**

.

**Grupo**

Carlos Armando Munhoz Vilela          | RA:822138213

Giulio Enrico Miranda Maciotta    | RA:822159355

Gustavo Morais Cardoso             | RA:822154343

Felipe Mori Ferreira             | RA:822129885

1. Qual a importância da técnica de segmentação de imagem?
2. Pesquise e cite algumas ferramentas matemáticas envolvidas em segmentação de imagens.
3. Faça a segmentação de uma imagem. Para isso, procure algoritmos prontos em Python com essa finalidade (como o que aparece no link abaixo). Na resposta, escreva abaixo o código Python completo e de um *print* das figuras empregadas neste processo.

<https://offsouza.medium.com/segmentando-objetos-pela-cor-opencv-487d5181b473>

1. Implemente um código em Python que faça a segmentação de pelo menos 3 imagens. Compare a segmentação e faça comentários sobre o resultado delas.

**RESPOSTA DO ALUNO**

1. A segmentação de imagem é crucial na visão computacional, pois divide imagens em regiões ou objetos de interesse, facilitando a extração de informações e a redução da complexidade dos dados. Ela é aplicada em áreas como medicina, para identificar órgãos ou tumores, em veículos autônomos, na detecção de objetos, e em imagens de satélites para monitoramento ambiental.
2. Alguns métodos encontrados foram:

Métodos de Limiarização (Thresholding)

Transformada de Watershed

Modelos de Nível de Contorno



import cv2

import numpy as np

# Carregar a imagem 'teste.jpg'

image = cv2.imread('imgs/teste.jpg')

def segment\_color(image, lower\_bound, upper\_bound):

hsv = cv2.cvtColor(image, cv2.COLOR\_BGR2HSV)

mask = cv2.inRange(hsv, lower\_bound, upper\_bound)

result = cv2.bitwise\_and(image, image, mask=mask)

return result

# Definir os limites para a cor roxa, roxo claro e roxo escuro

lower\_purple\_light = np.array([130, 50, 50]) # Limite inferior para roxo claro

upper\_purple\_light = np.array([160, 255, 255]) # Limite superior para roxo claro

lower\_purple\_dark = np.array([125, 50, 50]) # Limite inferior para roxo escuro

upper\_purple\_dark = np.array([130, 255, 255]) # Limite superior para roxo escuro

lower\_magenta = np.array([140, 50, 50]) # Limite inferior para magenta

upper\_magenta = np.array([160, 255, 255]) # Limite superior para magenta

lower\_lilac = np.array([125, 50, 70]) # Limite inferior para lilás

upper\_lilac = np.array([140, 255, 255]) # Limite superior para lilás

# Aplicar a segmentação das cores

mask\_light = segment\_color(image, lower\_purple\_light, upper\_purple\_light)

mask\_dark = segment\_color(image, lower\_purple\_dark, upper\_purple\_dark)

# Combinar as máscaras

segmented\_image = cv2.bitwise\_or(mask\_light, mask\_dark)

# Salvar a imagem resultante na pasta imgs

cv2.imwrite('imgs/resultado\_roxo.jpg', segmented\_image)

# Exibir as imagens

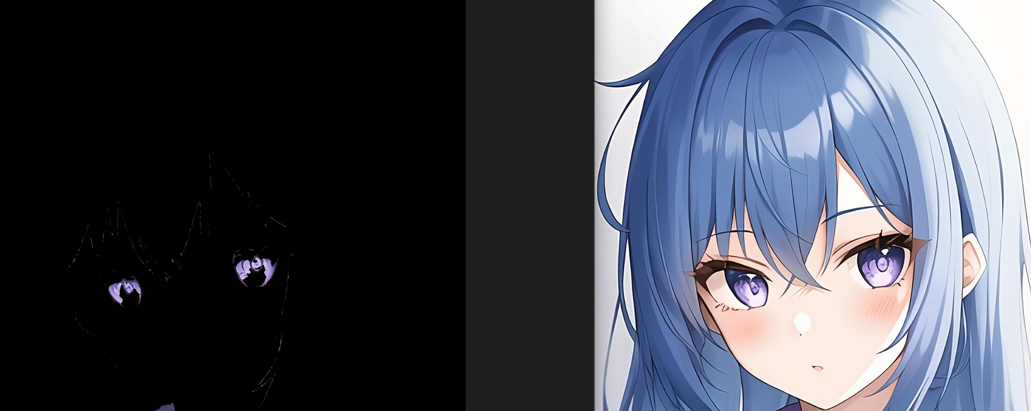
cv2.imshow('Original', image)

cv2.imshow('Segmented', segmented\_image)

# Aguardar até que uma tecla seja pressionada

cv2.waitKey(0)

cv2.destroyAllWindows()



Foi feito a segmentação de forma a pode separar os olhos do resto da imagem

4)

import cv2

import numpy as np

from google.colab.patches import cv2\_imshow

from google.colab import files

import io

uploaded = files.upload()

image = cv2.imdecode(np.frombuffer(uploaded[next(iter(uploaded))], np.uint8), cv2.IMREAD\_COLOR)

def segment\_color(image, lower\_bound, upper\_bound):

hsv = cv2.cvtColor(image, cv2.COLOR\_BGR2HSV)

mask = cv2.inRange(hsv, lower\_bound, upper\_bound)

result = cv2.bitwise\_and(image, image, mask=mask)

return result

lower\_red1 = np.array([0, 120, 70])

upper\_red1 = np.array([10, 255, 255])

mask1 = segment\_color(image, lower\_red1, upper\_red1)

mask2 = segment\_color(image, lower\_red2, upper\_red2)

segmented\_image = cv2.bitwise\_or(mask1, mask2)

cv2\_imshow(image)

cv2\_imshow(segmented\_image)

#Todas as imagens somente a cor vermelha ficou visível por exemplo o uniforme do São Paulo ficou somente preto e vermelho removendo o branco e deixando de ser tricolor, uma pokebola deixou de parecer uma esfera.