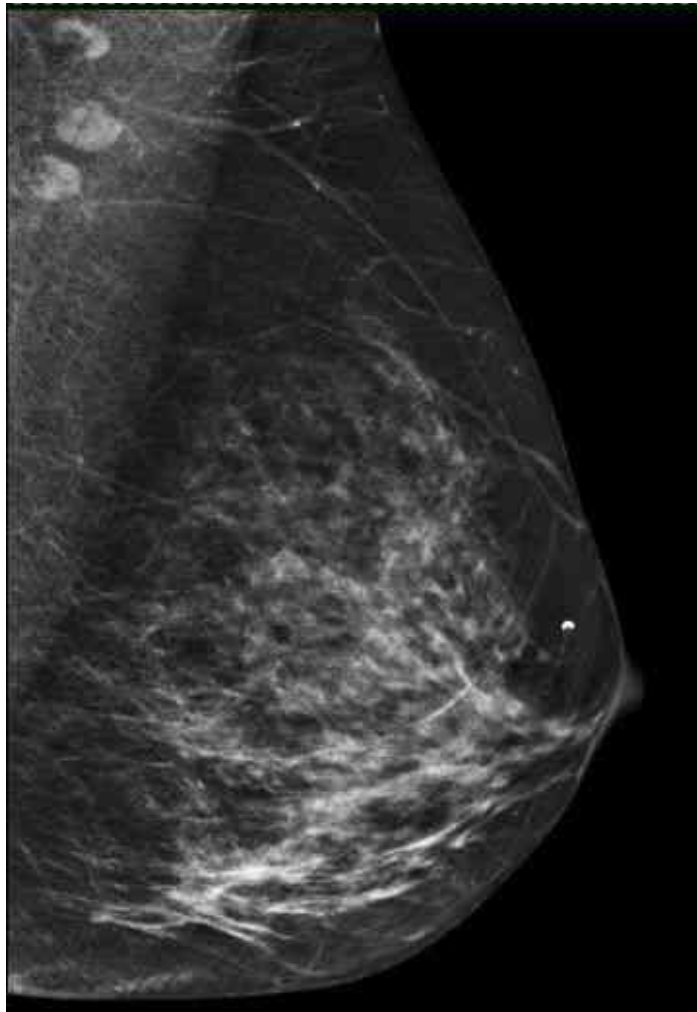


AUTORES:

Hyago Brucy Dias Martins - 2420709
Jean Carlos Araujo Medrado - 2212305

ANÁLISE DE IMAGEM DA MAMA



- Para realizar a análise de câncer nesta imagem, utilizaremos um código em Python que irá destacar os pixels brancos, melhorando a clareza da imagem e permitindo que o algoritmo faça uma análise mais precisa e eficiente:

```
from PIL import Image, ImageDraw, ImageEnhance
from skimage import measure
import numpy as np

image_path = '/mnt/data/RaioXmama.jpg'

imagem = Image.open(image_path)

imagem_cinza = imagem.convert('L')

matriz_imagem = np.array(imagem_cinza)

contornos = measure.find_contours(matriz_imagem, 0.8)

desenhar = ImageDraw.Draw(imagem)

for contorno in contornos:

    for i in range(len(contorno) - 1):

        desenhar.line((contorno[i][1], contorno[i][0], contorno[i+1][1],
contorno[i+1][0]), fill='red', width=2)

realcar = ImageEnhance.Contrast(imagem)

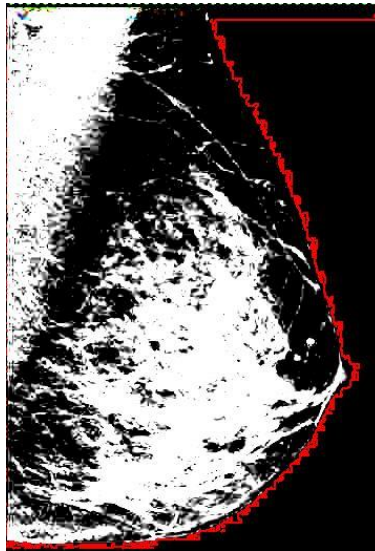
imagem = realcar.enhance(15.5)

output_image_path = '/mnt/data/mama_contornos.jpg'

imagem.save(output_image_path)

output_image_path
```

- Após aplicarmos esse código, irá retornar a seguinte imagem de saída:



- Com isso utilizaremos um código para calcular a porcentagem (%) de pixels brancos dentro dessa imagem determinando se há ou não a ocorrência de um câncer de mama:

```
import cv2

# Carregar a imagem gerada anteriormente

img = cv2.imread(output_image_path)

# Calcular o número de pixels brancos e pretos

numero_pixels Branco = np.sum(img == 255)

numero_pixels preto = np.sum(img == 0)

# Calcular o percentual de pixels brancos

percentual_pixels_branco = numero_pixels_branco / (numero_pixels_branco +
numero_pixels_preto) * 100

# Verificar se a imagem indica potencial câncer

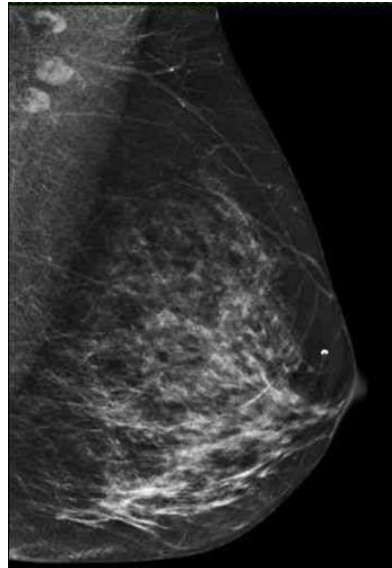
resultado = "imagem com cancer" if percentual_pixels_branco >= 30 else "imagem sem
cancer"

numero_pixels_branco, numero_pixels_preto, percentual_pixels_branco, resultado
```

- O resultado passando esse código na imagem de saída anterior, será:
- Número de pixels pretos: 229.745
- Número de pixels brancos: 113.711
- Percentual de pixels brancos: 33,11%

Logo a imagem acima trata-se de uma imagem com câncer de mama.

SEPARAÇÃO DE IMAGEM COLORIDA ENTRE OS CANAIS RGB



- Vamos utilizar o código abaixo para realização da separação dos canais de cores:

```
import cv2
import numpy as np

#carregaer a imagem
imagem = cv2.imread('mama.jpg')

#separando as camadas de cor em azul, verde e vermelho
azul, verde, vermelho = cv2.split(imagem)

#mesclando os canais de cor na ordem AVV
imagem_mesclada = cv2.merge ((azul, verde, vermelho))

#Invertendo os canais de cor
imagem_invertida = cv2.merge ((vermelho, verde, azul))

#criando uma imagem branca nas dimensoes da imagem lida
blank = np.zeros (imagem.shape[:2], dtype = 'uint8')

#abrindo as imagens por canais e mesclando com as matrizes de zeros
canal_azul = cv2.merge([azul, blank, blank])
canal_verde = cv2.merge([blank,verde,blank])
canal_vermelho = cv2.merge([blank, blank,vermelho])

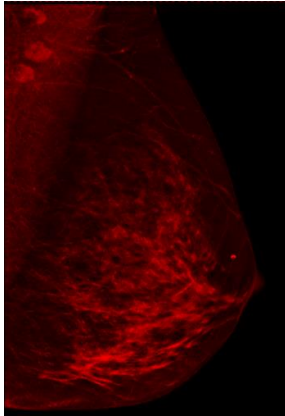
#visualizando
cv2.imwrite ('azul.png', canal_azul)
cv2.imwrite ('verde.png', canal_verde)
cv2.imwrite ('vermelho.png',canal_vermelho)

cv2.imwrite ("imagem_mesclando.png", imagem_mesclada)

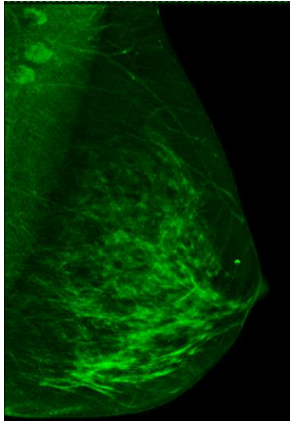
cv2.imwrite("imagem_invertida.png", imagem_invertida)
```

- Logo após, passarmos a imagem através do código, teremos 4 imagens

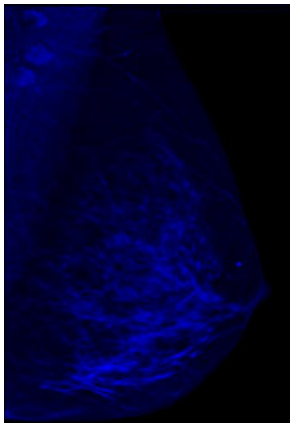
- Vermelho:



- Verde:



- Azul:



- Imagem invertida

