

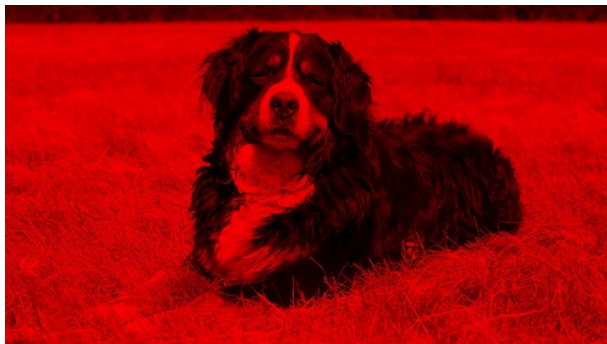
## Projeto da Disciplina – Processamento Digital – Prof. Fabio Botelho:

Criação dos códigos estudados em sala:

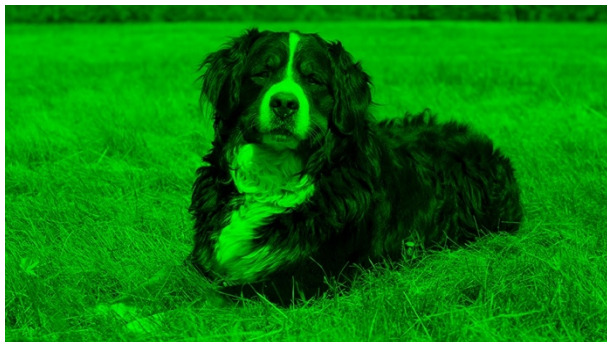
1. Separação de imagem colorida entre os canais vermelho, verde e azul:



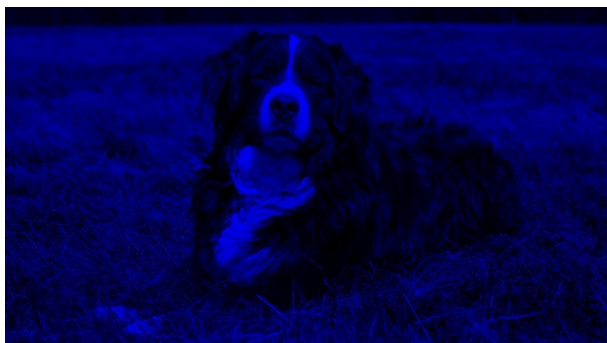
Bernese\_Mountain\_Dog.jpg



canal\_vermelho.jpg



canal\_verde.jpg

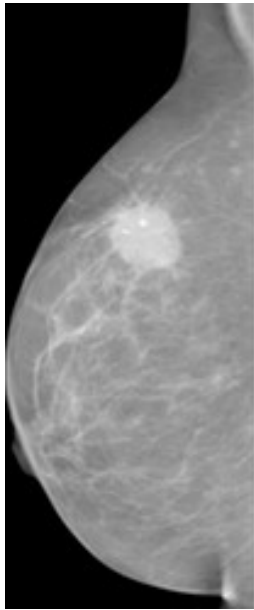


canal\_azul.jpg

Código:

```
codigo-1.py > ...
1  # Separação de imagem colorida entre os canais vermelho, verde e azul (RGB)
2
3  import cv2
4  import numpy as np
5  import matplotlib.pyplot as plt
6
7  # Carregando a imagem
8  imagem_cachorro = cv2.imread('Bernese_Mountain_Dog.jpg')
9
10 # Separação dos canais de cor (RGB)
11 azul, verde, vermelho = cv2.split(imagem_cachorro)
12
13 # Criando imagens dos canais separados
14 blank = np.zeros(imagem_cachorro.shape[:2], dtype='uint8')
15 canal_azul = cv2.merge([azul, blank, blank])
16 canal_verde = cv2.merge([blank, verde, blank])
17 canal_vermelho = cv2.merge([blank, blank, vermelho])
18
19 # Exibindo as imagens separadas dos canais RGB
20 fig, axes = plt.subplots(1, 3, figsize=(15, 5))
21 axes[0].imshow(cv2.cvtColor(canal_vermelho, cv2.COLOR_BGR2RGB))
22 axes[0].set_title('Canal Vermelho')
23 axes[1].imshow(cv2.cvtColor(canal_verde, cv2.COLOR_BGR2RGB))
24 axes[1].set_title('Canal Verde')
25 axes[2].imshow(cv2.cvtColor(canal_azul, cv2.COLOR_BGR2RGB))
26 axes[2].set_title('Canal Azul')
27
28 for ax in axes:
29     ax.axis('off')
30
31 plt.show()
32
33 # Salvando os canais em arquivos
34 cv2.imwrite('canal_vermelho.jpg', canal_vermelho)
35 cv2.imwrite('canal_verde.jpg', canal_verde)
36 cv2.imwrite('canal_azul.jpg', canal_azul)
37
```

## 2. Pré-processamento de imagem de câncer de mama:



mama.jpg



mama\_contornos\_final.jpg

Código:

```
codigo-2.py > ...
1  # Pré-processamento de imagem de câncer de mama
2
3  from PIL import Image, ImageDraw, ImageEnhance
4  from skimage import measure
5  import numpy as np
6  import matplotlib.pyplot as plt
7
8  # Carregando a imagem de câncer de mama
9  imagem_mama = Image.open('mama.jpg')
10
11 # Convertendo a imagem para tons de cinza
12 imagem_cinza = imagem_mama.convert('L')
13
14 # Convertendo a imagem para uma matriz numpy
15 matriz_imagem = np.array(imagem_cinza)
16
17 # Encontrando contornos usando a função find_contours da skimage
18 contornos = measure.find_contours(matriz_imagem, 0.8)
19
20 # Desenhando os contornos na imagem
21 desenhar = ImageDraw.Draw(imagem_mama)
22 for contorno in contornos:
23     for i in range(len(contorno) - 1):
24         desenhar.line(
25             (
26                 contorno[i][1],
27                 contorno[i][0],
28                 contorno[i + 1][1],
29                 contorno[i + 1][0],
30             ),
31             fill='red',
32             width=2,
33         )
34
35 # Realçando o contraste da imagem
36 realcar = ImageEnhance.Contrast(imagem_mama)
37 imagem_realcada = realcar.enhance(15.5)
38
39 # Salvando a imagem final
40 imagem_realcada.save('mama_contornos_final.jpg')
41
42 # Exibindo a imagem resultante
43 plt.imshow(imagem_realcada, cmap='gray')
44 plt.title('Imagem com Contornos e Contraste Aumentado')
45 plt.axis('off')
46 plt.show()
47
```

3. Análise de imagem de câncer de mama para inferir presença ou ausência de câncer de mama:



mama\_contornos\_final\_analise.jpg



mama\_contornos\_pre\_processada.jpg



Código:

```
codigo-3.py > ...
1  # Análise de imagem de câncer de mama para inferir presença ou ausência de câncer
2
3  import cv2
4  import numpy as np
5
6  # Carregando a imagem pré-processada com contornos e contraste realçado
7  imagem_cv2 = cv2.imread('mama_contornos_final.jpg', cv2.IMREAD_GRAYSCALE)
8
9  # Calculando o número de pixels brancos e pretos
10 numero_pixels Branco = np.sum(imagem_cv2 == 255)
11 numero_pixels preto = np.sum(imagem_cv2 == 0)
12
13 # Calculando o percentual de pixels brancos
14 percentual_pixels_branco = (
15     numero_pixels_branco / (numero_pixels_branco + numero_pixels_preto) * 100
16 )
17
18 # Inferindo a presença de câncer com base no percentual de pixels brancos
19 if percentual_pixels_branco >= 30:
20     print('Imagem com câncer')
21 else:
22     print('Imagem sem câncer')
23
24 print(f'Percentual de pixels brancos: {percentual_pixels_branco:.2f}%')
25
```

Link do Github: <https://github.com/CalebeRRdev/PROCESSAMENTO-DIGITAL>

Integrantes do Grupo:

- Calebe Rodrigues Rolim – 2221533
- Gabriel Reis Costa - 2111508