

## Processamento de Imagem - Professor Elcio

### Alunos

Bruno Alexandre Lassakoski D51IGB0

Yan Gabriel Furlan N306629

### Processamento de imagem em Python

Github <https://github.com/richardson-souza/processamento-de-imagem#aplica%C3%A7%C3%B5es>

Processamento de imagens ( Image Processing ) apresentam técnicas para manipular informações representadas na imagem como por exemplo, realçar bordas e remover resíduos.

No campo de Visão Computacional ( Computer Vision ), técnicas de processamento de imagem são aplicadas para inferir informações de baixo nível em partes da imagem. Os algoritmos de visão computacional têm como entrada uma ou mais imagens digitais que são pré-processadas por meio de técnicas de processamento de imagem para que informações úteis possam ser extraídas.

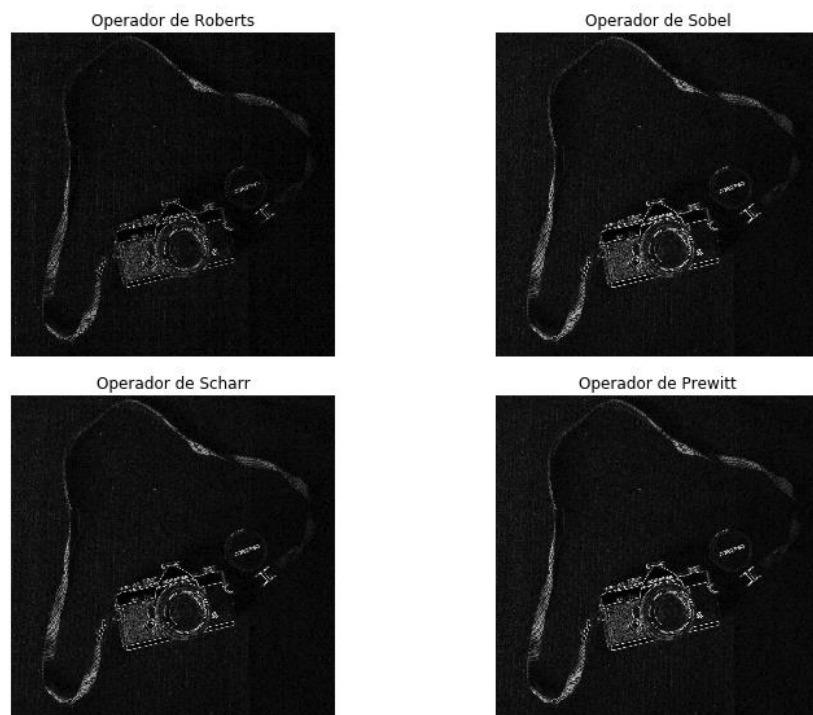
Existem inúmeras bibliotecas disponíveis para auxiliar no processamento de imagens como o OpenCV.

Para os exemplos apresentados foi usada a biblioteca scikit-image[2] que é uma coleção de algoritmos para processamento de imagens. Está disponível gratuitamente e sem restrições.

- Imagem base



- Visualizando os resultados



- Código

Importando bibliotecas necessárias:

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

```
from skimage.io import imread
from skimage.filters import roberts, sobel, scharr, prewitt
```

Usando os operadores:

```
img = imread("input/camera_01.jpg", as_gray=True) # carregando imagem

op_roberts = roberts(img)
op_sobel = sobel(img)
op_scharr = scharr(img)
op_prewitt = prewitt(img)
```

Visualizando os resultados:

```
fig, axes = plt.subplots(nrows=2, ncols=2, sharex=True, sharey=True, figsize=(12, 12))
```

```
ax = axes.ravel()
```

```
ax[0].imshow(op_roberts, cmap=plt.cm.gray)
```

```
ax[0].set_title('Operador de Roberts')
```

```
ax[1].imshow(op_sobel, cmap=plt.cm.gray)
```

```
ax[1].set_title('Operador de Sobel')
```

```
ax[2].imshow(op_scharr, cmap=plt.cm.gray)
```

```
ax[2].set_title('Operador de Scharr')
```

```
ax[3].imshow(op_prewitt, cmap=plt.cm.gray)
```

```
ax[3].set_title('Operador de Prewitt')
```

```
for a in ax:
```

```
    a.axis('off')
```

```
plt.tight_layout()
```

```
plt.show()
```

## Relatório

O resultado como podemos ver. São pequenas diferenças de iluminação e de cores de branco das imagens, cada uma sendo uma melhoria da outra na qualidade para a real imagem, diferenciando cada os objetos na imagem