

# parte1

April 14, 2021

## 1 Parte 1 - Imagens coloridas

TIAGO PEREIRA DALL'OCA - 206341

```
[1]: from scipy import misc
     from scipy import ndimage
     import cv2
     import numpy as np
     import matplotlib.pyplot as plt
```

```
[2]: img = cv2.imread('imagens/baboon.png')
```

```
[3]: img.shape
```

```
[3]: (512, 512, 3)
```

### 1.1 a)

Aqui é bem autoexplicativo. É criado uma matriz que irá multiplicar os vetores que representam os três canais de cores de cada pixel.

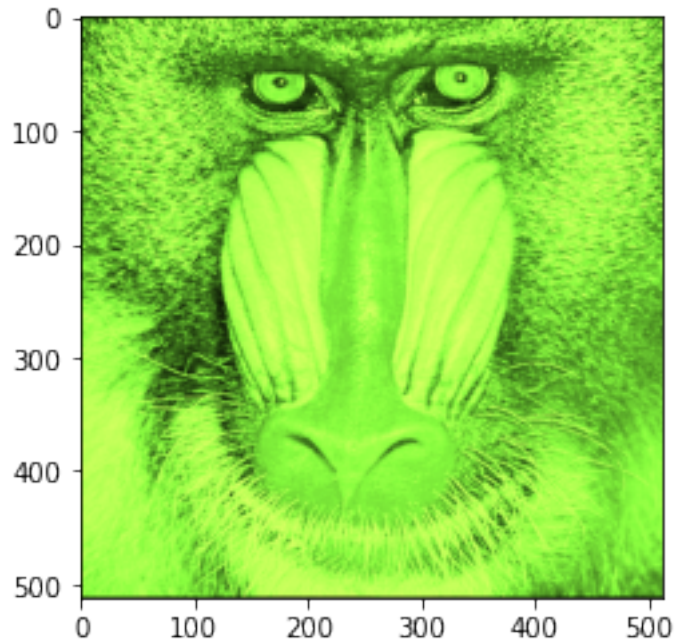
```
[5]: matriz_a = np.array([[0.393, 0.769, 0.189],
                          [0.394, 0.686, 0.168],
                          [0.272, 0.534, 0.131]])
```

```
[15]: img_a = np.dot(img, matriz_a)/255
      img_a = img_a.clip(max=[1,1,1])
      img_a.shape
```

```
[15]: (512, 512, 3)
```

```
[16]: plt.imshow(img_a)
```

```
[16]: <matplotlib.image.AxesImage at 0x7f07f499d5e0>
```



## 1.2 b)

Semelhante ao item “a”, porém agora faremos uma multiplicação vetorial que nos resultará em uma imagem de canal único (imagem cinza).

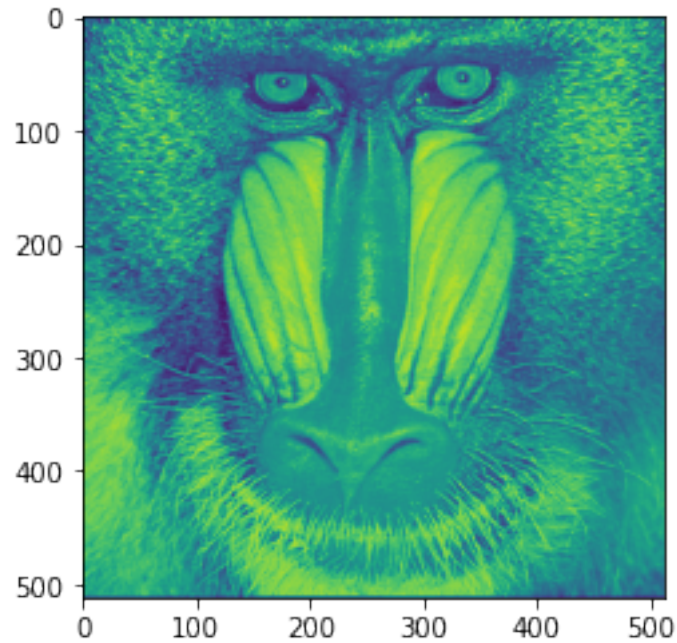
```
[36]: vetor_b = np.array([0.2989, 0.5870, 0.1140])
```

```
[38]: img_b = np.tensordot(img, vetor_b, axes=([2], [0]))/255  
img_b = img_b.clip(max=1).reshape(img.shape[0:2])  
img_b.shape
```

```
[38]: (512, 512)
```

```
[39]: plt.imshow(img_b)
```

```
[39]: <matplotlib.image.AxesImage at 0x7f07f4034640>
```



### 1.3 O Programa

**Executar:**

```
python3 python3 parte1_imagens_coloridas.py imagens_coloridas/baboon.png [...]
```

As imagens resultado da aplicação dos filtros vão estar na pasta `imagens_mascaradas_parte1/` local.

As imagens estão prefixadas com “a” e “b” indicando a qual item pertencem.