

## 1ª Atividade

### Manipulações básicas de imagens coloridas

O código abaixo mostra como utilizar a biblioteca OpenCV para abrir uma imagem colorida e exibi-la utilizando o pacote de imagens incluso com o Python.

---

```
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt

%matplotlib inline

filename = "SmallSample.png"
img = cv.imread(filename,1)
plt.imshow(img)
```

---

Utilizando o código acima como base, escreva um código que realize as seguintes tarefas:

1. Exiba a imagem em escala de cinza.
2. Exiba os canais R, G e B de uma imagem em gráficos do Matplotlib.
3. Converta uma imagem em BGR ou RGB para HSV, e exiba os canais H, S e V em gráficos.
4. Reconstitua a imagem original a partir dos canais individuais para ambos os casos.
5. Formule e aplique um filtro de média aritmética de dimensão  $3 \times 3$  a todos os canais de uma imagem colorida e exiba o resultado. O preenchimento deve ser feito com zeros.
6. Aplique o filtro resultante a apenas a um dos canais em uma imagem RGB. Compare os resultados.
7. Aplique o filtro resultante a apenas ao canal V de uma imagem HSV. Compare os resultados.

## Funções de referência

Algumas funções de referência necessárias para implementar as tarefas acima são descritas abaixo:

---

```
filename = "example.png"
cv.imread(filename,mode)
```

---

- Retorna uma matriz onde cada um dos elementos é um vetor de três elementos, correspondendo a uma imagem. O modo define como a imagem será interpretada:
  - $mode = 0$  → Escala de cinza.
  - $mode = 1$  → Imagem colorida em padrão BGR, ignora canais de transparência. É a configuração padrão.
  - $mode - 1$  → Carrega a imagem sem alterações.

---

```
# create figure 1
fig1 = plt.figure(1)
# create subplot in fig1
ax1 = fig1.subplots(1,2)
# set 1st plot
ax1[0].imshow(img1, cmap='gray')
ax1[0].title.set_text('I')
ax1[0].set_xticks([])
ax1[0].set_yticks([])
# set 2nd plot
ax1[1].imshow(img2)
ax1[1].title.set_text('II')
ax1[1].set_xticks([])
ax1[1].set_yticks([])
# show figures
plt.show()
```

---

- Exibe um *subplot* contendo duas imagens lado a lado sem marcações nos eixos. A função `imshow` espera uma imagem colorida no padrão RGB.

---

```
img_converted = cv.cvtColor(img,CONVERSION)
```

---

- Retorna uma imagem convertida de um espaço de cores para outro de acordo com o parâmetro `CONVERSION`. A lista valores possíveis para `CONVERSION`, especificando as conversões entre espaços, pode ser encontrada [clikando aqui](#).

---

```
x, y, z = cv.split(img)
```

---

- Retorna os três canais de uma matriz onde cada um dos elementos é um vetor de três elementos, correspondendo a uma imagem colorida. Note que os canais  $x$ ,  $y$  e  $z$  representados acima corresponderão ao formato da imagem (RGB, HSV, etc.).

---

```
img_rgb = cv.merge((r,g,b))  
# or  
img_hsv = cv.merge((h,s,v))
```

---

- Retorna uma matriz onde cada um dos elementos é um vetor de três elementos, correspondendo a uma imagem colorida. Acima são utilizados canais RGB e HSV em ordem como exemplo, mas na prática não há esta restrição.