

▼ Laboratório 2 - Imagem e Reprodução e Captura de Vídeo

Incorporação das bibliotecas OpenCV e Numpy

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

Estes comandos são para incorporar meu google drive aqui no notebook. O Google irá perguntar e te enviar um hyperlink, que depois de copiado, efetua a montagem

```
from google.colab import drive
drive.mount('/content/drive/')
```

Mounted at /content/drive/

Subindo Arquivos usados neste Laboratório

Na pasta /Colab Notebooks do google drive faça upload dos notebooks fornecidos, assim como dos arquivos de imagem e vídeo.

Verifique se os arquivos subiram todos:

```
!ls -l
!ls "/content/drive/My Drive/Colab Notebooks/"

total 8
drwx----- 6 root root 4096 Sep 28 22:11 drive
drwxr-xr-x 1 root root 4096 Sep 16 13:40 sample_data
Modelo#6-OptHyper.ipynb  train.csv      Treinar_Modelo.ipynb  Untitled1.ipynb
Multimidia           train_pca.csv  Untitled0.ipynb
```

Verificando a Resolução e as Cores

```
imgmessi = cv.imread('/content/drive/My Drive/Colab Notebooks/Multimidia/messi5.jpg')
altura, largura, camadas = imgmessi.shape
print("Resolução: ", largura, " X ", altura, " PIXELS. ", camadas, " camadas. " )
plt.imshow(imgmessi)
plt.xticks([]), plt.yticks([])
plt.show()
```

Resolução: 548 X 342 PIXELS. 3 camadas.

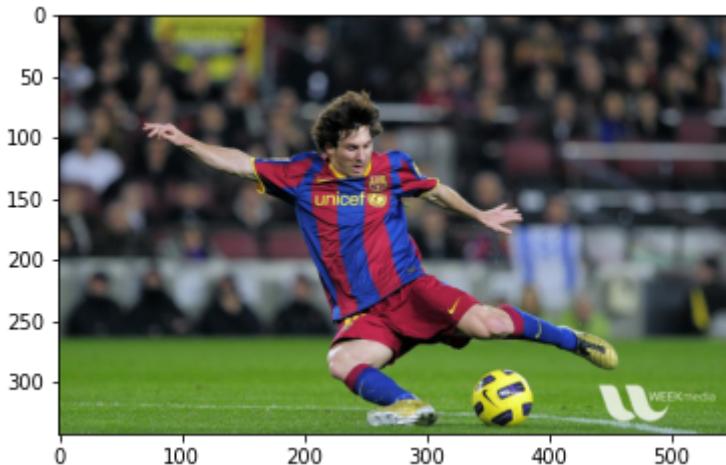


Versões de imread(), no OpenCV e na biblioteca matplotlib:

O opencv (cv) com o imread() lê as cores trocadas... Vejam o imread() do matplotlib (plt):

```
imagem=plt.imread("/content/drive/My Drive/Colab Notebooks/Multimidia/messi5.jpg") # imrea
print("Formato da imagem: ", imagem.shape)
print(f"Esta imagem contém {imagem.shape[0] * imagem.shape[1]} pixels")
plt.imshow(imagem);
plt.show()
```

Formato da imagem: (342, 548, 3)
Esta imagem contém 187416 pixels

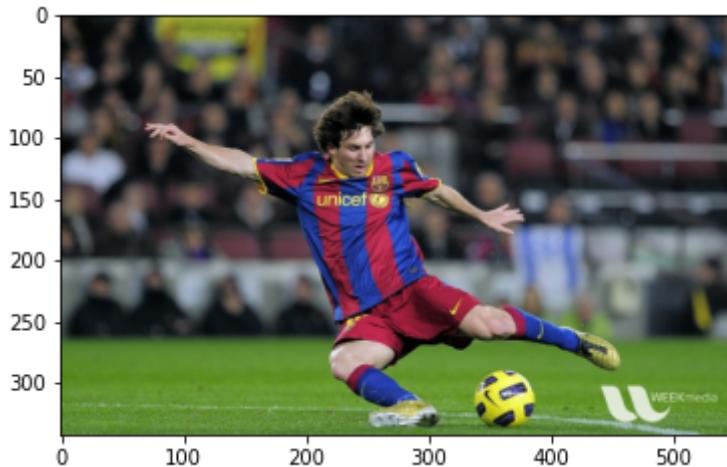


```
import cv2 as cv
import os
image_path = "/content/drive/My Drive/Colab Notebooks/Multimidia/"
img = cv.imread(os.path.join(image_path, 'messi5.jpg'), cv.IMREAD_COLOR)
# BGR
plt.imshow(img)
plt.show()
```



```
# Para termos RGB invertem-se a ordem das cores:
```

```
imgRGB = img[:, :, ::-1]
plt.imshow(imgRGB)
plt.show()
```



Vejam estes códigos:

```
a = 'MARIO MINAMI'
print(a)
```

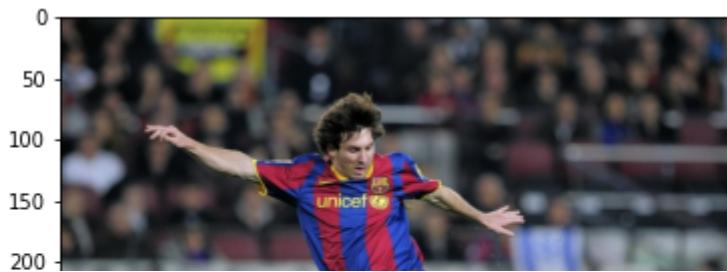
```
MARIO MINAMI
```

```
b = a[::-1]
print(b)
```

```
IMANIM OIRAM
```

O OpenCV tem uma função para isso, cvtColor():

```
# BGR to RGB
imgRGB = cv.cvtColor(img, cv.COLOR_BGR2RGB)
plt.imshow(imgRGB)
plt.show()
```



```
# Escala de Cinza, o plt.imshow está dando errado...
img_gray = cv.imread(os.path.join(image_path, 'messi5.jpg'), cv.IMREAD_GRAYSCALE)
plt.imshow(img_gray, 'gray')
plt.show()
```



Usando uma versão do google para o colab, do imshow:

```
from google.colab.patches import cv2_imshow # cv2_imshow
cv2_imshow(img_gray)
```



Podemos usar resize() do OpenCV para mudar a resolução da imagem, façamos com a nossa P&B:

```
print(img_gray.shape)
img2 = cv.resize(img_gray, None, fx=1.25, fy=1.25, interpolation = cv.INTER_CUBIC)
print(img2.shape)

(342, 548)
(428, 685)

cv2_imshow(img2)
```



▼ REPRODUÇÃO DE VÍDEO

Vamos usar o HTML para exibir vídeos no Colab!

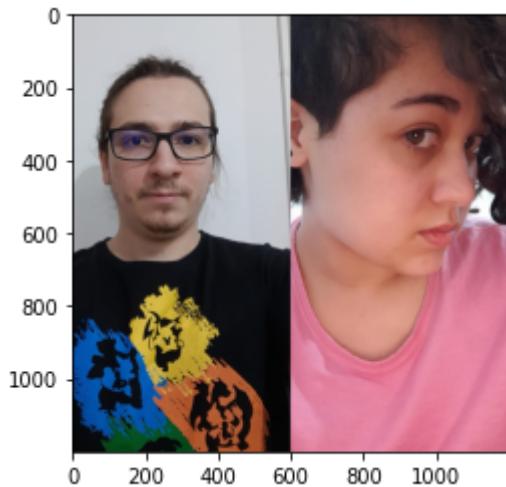
```
from IPython.display import HTML
from base64 import b64encode
video_path = "/content/drive/My Drive/Colab Notebooks/Multimidia/Pikachu.mp4"
mp4 = open(video_path,'rb').read()
video_decodificado = "data:video/mp4;base64," + b64encode(mp4).decode()
HTML(f'<video width=720 controls><source src= {video_decodificado} type="video/mp4"></vide
```

0:00 / 0:06

▼ Captura de Imagem, usar o Notebook Anexo 1

```
#Foto com todos os membros do grupo
foto = 'fotos.jpg'
imagem=plt.imread("/content/drive/My Drive/Colab Notebooks/Multimidia/" + foto) # imread d
print("Formato da imagem: ", imagem.shape)
print(f"Esta imagem contém {imagem.shape[0] * imagem.shape[1]} pixels")
plt.imshow(imagem);
plt.show()
```

Formato da imagem: (1200, 1200, 3)
Esta imagem contém 1440000 pixels

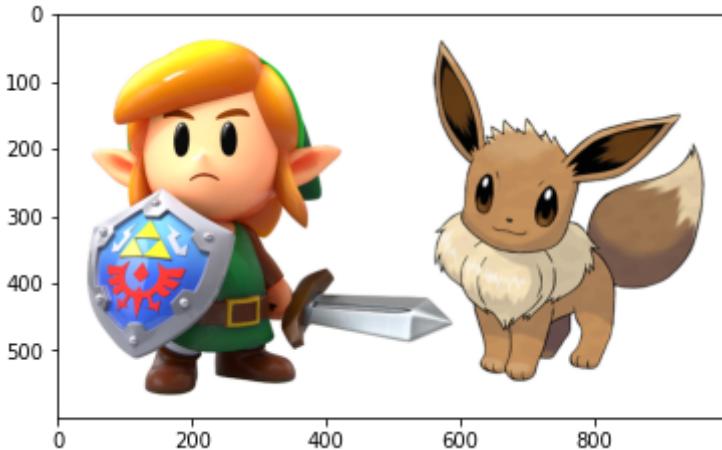


```
#Foto com todos os avatares do grupo
foto = 'avatares.png'
```

<https://colab.research.google.com/drive/1s7UL2e3rBVHvC2KljZc1uzsQTnq-tXsB?authuser=1#scrollTo=oc89cstHIACX>

```
imagem=plt.imread("/content/drive/My Drive/Colab Notebooks/Multimidia/" + foto) # imread d
print("Formato da imagem: ", imagem.shape)
print(f"Esta imagem contém {imagem.shape[0] * imagem.shape[1]} pixels")
plt.imshow(imagem);
plt.show()
```

Formato da imagem: (600, 1000, 3)
Esta imagem contém 600000 pixels



▼ Captura de Vídeo, usar o Notebook Anexo 2

```
# Video em câmera lenta
from IPython.display import HTML
from base64 import b64encode
video = "video_lento.mp4"
video_path = "/content/drive/My Drive/Colab Notebooks/Multimidia/" + video
mp4 = open(video_path,'rb').read()
video_decodificado = "data:video/mp4;base64," + b64encode(mp4).decode()
HTML(f'<video width=720 controls><source src= {video_decodificado} type="video/mp4"></video>')
```

```
# Video em câmera rápida
from IPython.display import HTML
from base64 import b64encode
video = "video_rapido.mp4"
```

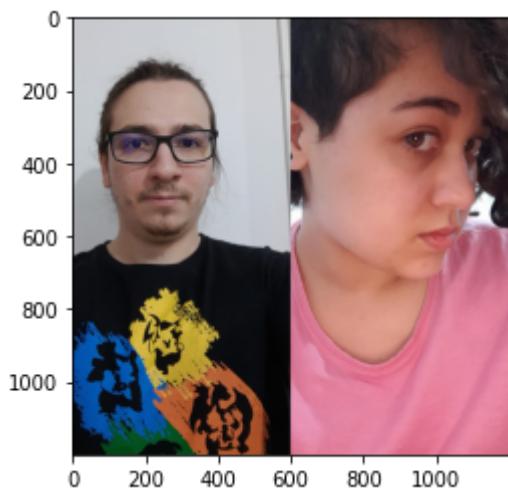
```
video_path = "/content/drive/My Drive/Colab Notebooks/Multimidia/" + video
mp4 = open(video_path, 'rb').read()
video_decodificado = "data:video/mp4;base64," + b64encode(mp4).decode()
HTML(f'<video width=720 controls ><source src= {video_decodificado} type="video/mp4" ></vi
```

▼ Edições para o Relatório

▼ Fotos Originais

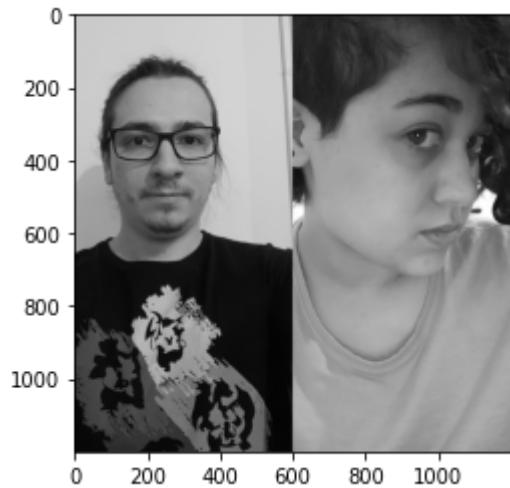
```
foto = "fotos.jpg"
imagem=plt.imread("/content/drive/My Drive/Colab Notebooks/Multimidia/" + foto) # imread d
print('Imagen do Matheus')
print("Formato da imagem: ", imagem.shape)
print(f"Esta imagem contém {imagem.shape[0] * imagem.shape[1]} pixels")
plt.imshow(imagem);
plt.show()
```

Imagen do Matheus
Formato da imagem: (1200, 1200, 3)
Esta imagem contém 1440000 pixels



▼ Fotos P/B

```
foto = "otos.jpg"
img_gray = cv.imread(os.path.join(image_path,foto), cv.IMREAD_GRAYSCALE)
plt.imshow(img_gray, 'gray')
plt.show()
```



▼ Fotos com resoluções diferentes

```
# Matheus Ramos Ribeiro
foto = "otos.jpg"
imagem=plt.imread("/content/drive/My Drive/Colab Notebooks/Multimidia/" + foto)
print(f'escala: {imagem.shape}')
cv2_imshow(imagem)
imagem = cv.resize(imagem, None, fx=0.5, fy=0.5, interpolation = cv.INTER_CUBIC)
print(f'escala: {imagem.shape}')
cv2_imshow(imagem)
imagem = cv.resize(imagem, None, fx=0.25, fy=0.25, interpolation = cv.INTER_CUBIC)
print(f'escala: {imagem.shape}')
cv2_imshow(imagem)
```

