

COM 3105

Processamento Digital de Imagens

LN03-03 - 2023

Frameworks e Bibliotecas para o Processamento de Imagens

Matplotlib

Prof. Eng. Marcos Cordeiro d'Ornellas, Ph.D.

Revisado em: 26.03.2023.

1. Matplotlib

1.1 Obtendo e Instalando o Matplotlib

1.2 Verificando a Versão Atual (Estável)

In [1]:

```
1 try:
2     import matplotlib
3     print(matplotlib.__version__)
4 except ImportError:
5     print(matplotlib is not installed)
```

executed in 174ms, finished 18:32:08 2023-03-26

3.5.2



1.3 Operações Básicas

1.3.1 Carregando as Bibliotecas e Funções

In [2]:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import matplotlib.image as mpimg
```

executed in 502ms, finished 18:32:08 2023-03-26

1.3.2 Abrindo e Visualizando uma Imagem

Matplotlib é uma biblioteca em Python e é uma extensão numérica - matemática da biblioteca NumPy. Pyplot é uma interface baseada em estado para um módulo Matplotlib que fornece uma interface semelhante ao MATLAB.

Syntax: `matplotlib.pyplot.imread(fname, format=None)`

Parameters:

- `fname` or file-like: The image file to read: a filename, a URL or a file-like object opened in read-binary mode.
- `format`, optional: The image file format assumed for reading the data. If not given, the format is deduced from the filename. If nothing can be deduced, PNG is tried.

Returns: image data

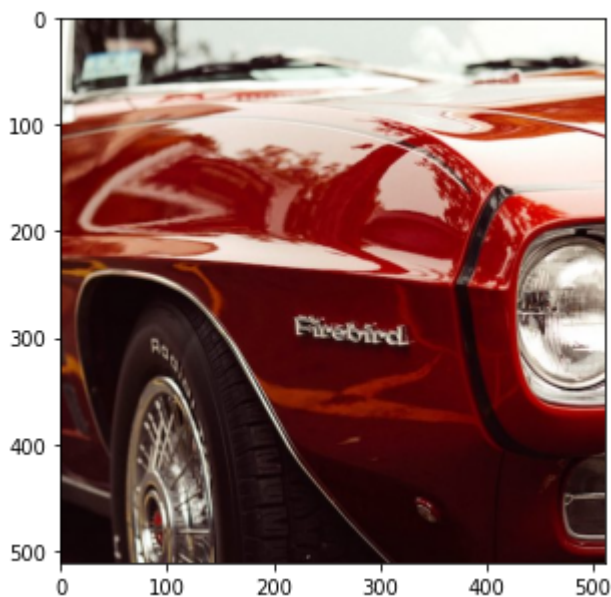
In [3]:

```
1 img = mpimg.imread('image_folder/redcar512.jpg')
2
3 # show the image
4 # The show() function will display the image using your operating systems de
5 # img.show()
6
7 # imshow() from matplotlib
8 plt.figure(figsize=(5,5))
9 plt.imshow(img)
```

executed in 196ms, finished 18:32:08 2023-03-26

Out[3]:

<matplotlib.image.AxesImage at 0x7fed0850b8b0>



1.3.3 Atributos da Imagem

Os atributos da imagem podem obtidos a partir dos seguintes comandos:

In [4]:

```
1 # summarize some details about the image
2 print(img.shape)
3 print(img.size)
4 print(img.dtype)
5 print(img[20,20])
6
7 [rows,cols,pages]=np.shape(img)
8
9 # Get Resolution of Image
10 img_resolution_pixels=rows*cols
11 img_resolution_Mpixels=img_resolution_pixels/(1024*1024)
12 print('resolution=',np.round(img_resolution_Mpixels,2),'Mpixels')
```

executed in 4ms, finished 18:32:08 2023-03-26

```
(512, 512, 3)
786432
uint8
[64 49 46]
resolution= 0.25 Mpixels
```

1.3.4 Salvando uma Imagem

Como o nome sugere, o método `savefig()` é usado para salvar a figura criada após a plotagem dos dados. A figura criada pode ser salva em nossas máquinas locais usando este método.

Syntax: `savefig(fname, dpi=None, facecolor='w', edgecolor='w', orientation='portrait', papertype=None, format=None, transparent=False, bbox_inches=None, pad_inches=0.1, frameon=None, metadata=None)`

Parameters:

- `fname`: Filename .png for image, .pdf for pdf format. File location can also be specified here.
- `dpi`: Number of dots per inch.(picture quality)
- `papertype` Paper type could be “a0 to a10”, “executive”, “b0 to b10”, “letter”, “legal”, “ledger”.
- `format`: File format such as .png, .pdf.
- `facecolor` and `edgecolor` Default as White.
- `bbox_inches`: Set it as “tight” for proper fit of the saved figure.
- `pad_inches`: Padding around the saved figure.
- `transparent`: Makes background of the picture transparent.
- `Orientation`: Landscape or Portrait.

In [5]:

```
1 # plt.savefig('image_folder/saved.png', format='PNG')
```

executed in 5ms, finished 18:32:08 2023-03-26

1.3.5 Convertendo uma Imagem

In [6]:

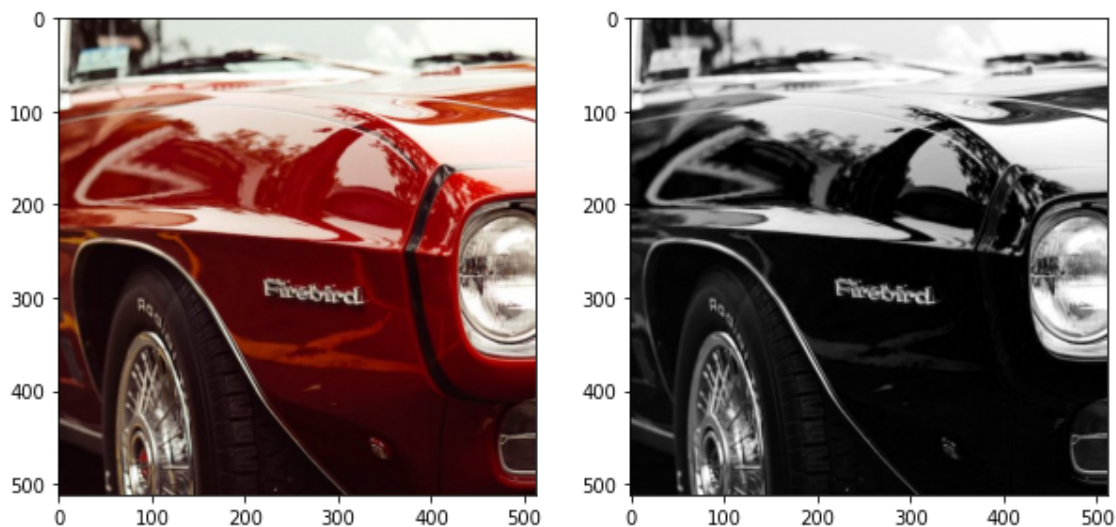
```
1 lum_img = img[:, :, 1]
2
3 print(lum_img[20, 20])
4
5 plt.figure(figsize=(10, 10))
6
7 plt.subplot(121)
8 plt.imshow(img)
9
10 plt.subplot(122)
11 plt.imshow(lum_img, cmap='gray', vmin = 0, vmax = 255)
```

executed in 295ms, finished 18:32:09 2023-03-26

49

Out[6]:

<matplotlib.image.AxesImage at 0x7fed38f99460>



1.4 Operações Geométricas

1.4.1 Recortando, Colando e Rotacionando Imagens

In [7]:

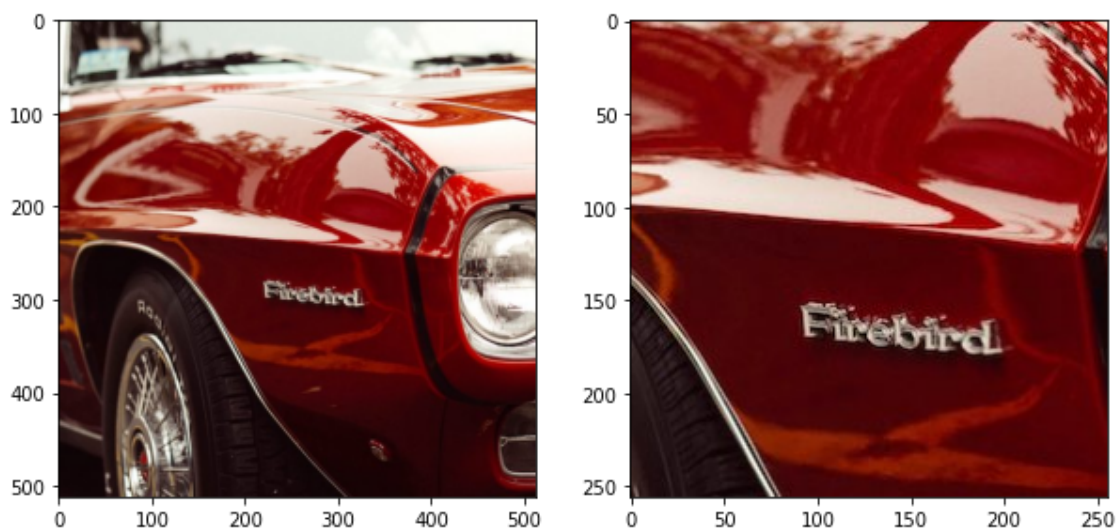
```
1 crop = img[128:-128, 128:-128, :]  
2  
3 fig = plt.figure()  
4 fig.tight_layout()  
5  
6 plt.figure(figsize=(10,10))  
7  
8 plt.subplot(121)  
9 plt.imshow(img)  
10  
11 plt.subplot(122)  
12 plt.imshow(crop)
```

executed in 344ms, finished 18:32:09 2023-03-26

Out[7]:

<matplotlib.image.AxesImage at 0x7fed38ff7e80>

<Figure size 432x288 with 0 Axes>



In [8]:

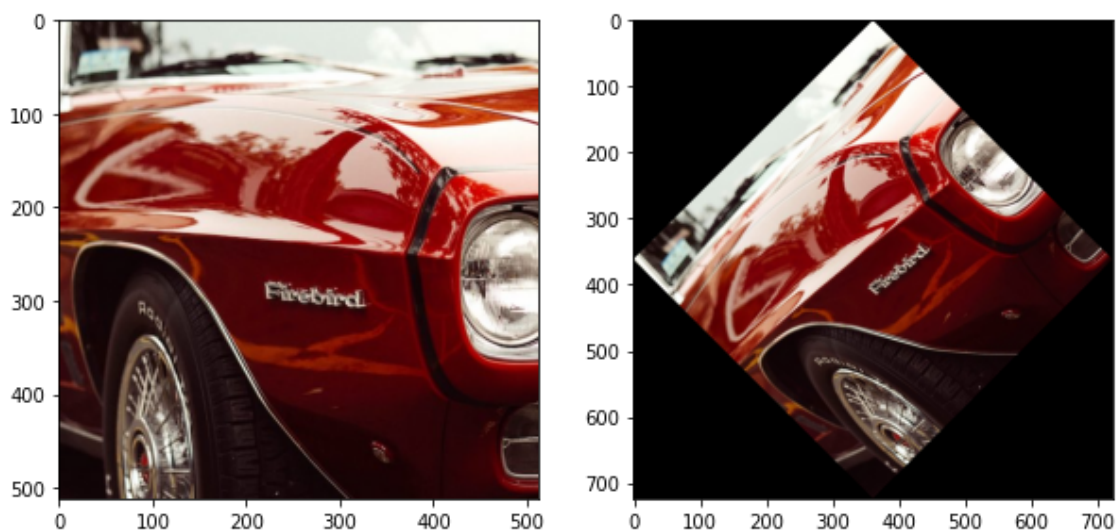
```
1 import scipy.ndimage as ndimage
2
3 rotate=ndimage.rotate(img, 45)
4
5 fig = plt.figure()
6 fig.tight_layout()
7
8 plt.figure(figsize=(10,10))
9
10 plt.subplot(121)
11 plt.imshow(img)
12
13 plt.subplot(122)
14 plt.imshow(rotate)
```

executed in 1.07s, finished 18:32:10 2023-03-26

Out[8]:

<matplotlib.image.AxesImage at 0x7fed19027220>

<Figure size 432x288 with 0 Axes>



1.4.2 Mudando o tamanho da Imagem

Para mudar o tamanho de uma imagem e realizar uma mudança de escala podemos utilizar a função `scale` definida localmente:

In [9]:

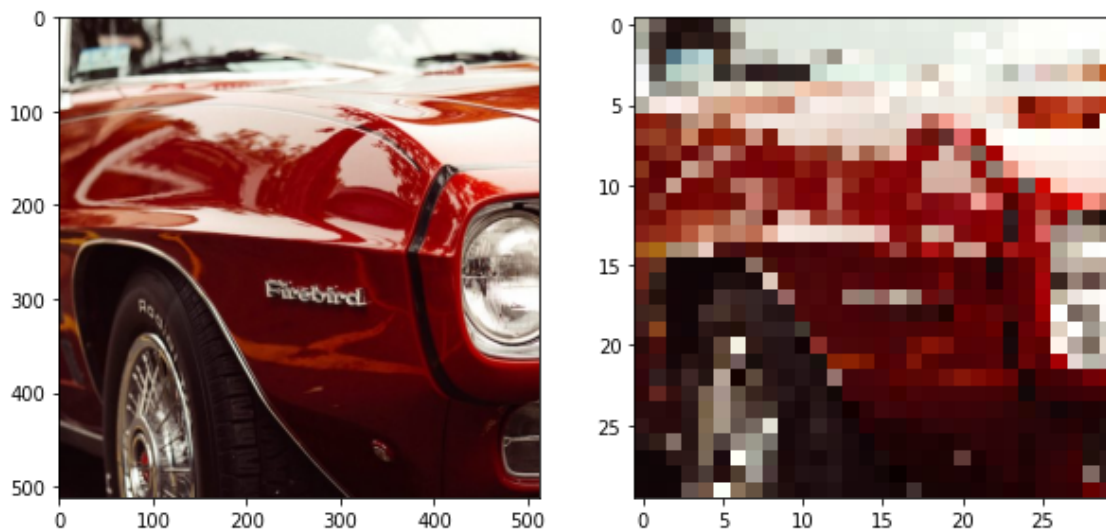
```
1 def scale(im, nR, nC):
2     number_rows = len(im)      # source number of rows
3     number_columns = len(im[0]) # source number of columns
4     return [[ im[int(number_rows * r / nR)][int(number_columns * c / nC)]
5              for c in range(nC)] for r in range(nR)]
6
7 scaled = scale(img,30,30)
8
9 fig = plt.figure()
10 fig.tight_layout()
11
12 plt.figure(figsize=(10,10))
13
14 plt.subplot(121)
15 plt.imshow(img)
16
17 plt.subplot(122)
18 plt.imshow(scaled)
```

executed in 236ms, finished 18:32:10 2023-03-26

Out[9]:

<matplotlib.image.AxesImage at 0x7fed08c2e3d0>

<Figure size 432x288 with 0 Axes>



1.5 Manipulação dos Canais | Bandas da Imagem

1.5.1 Separando Canais | Bandas

In [10]:

```
1  png = mpimg.imread('image_folder/redcar512.png')
2
3  # Dimensions of img
4  [rows,cols,pages]=np.shape(img)
5
6  # Split image "pages" into separate r g b a channels
7  r=img[:, :, 0]
8  g=img[:, :, 1]
9  b=img[:, :, 2]
10 # alpha=img[:, :, 3]
11
12 # Create an empty array where rgb=0,a=alpha
13 imlempty=np.zeros([rows,cols,pages])
14 # imlempty[:, :, 3]=alpha
15 # Create an array with r=r, g=0, b=0, a=1
16 r2=np.copy(imlempty)
17 r2[:, :, 0]=r2[:, :, 0]+r
18 # Create an array with r=0, g=g, b=0, a=1
19 g2=np.copy(imlempty)
20 g2[:, :, 1]=g2[:, :, 1]+g
21 # Create an array with r=0, g=0, b=b, a=1
22 b2=np.copy(imlempty)
23 b2[:, :, 2]=b2[:, :, 2]+b
24
25 fig = plt.figure()
26 fig.tight_layout()
27
28 plt.figure(figsize=(10,10))
29
30 plt.subplot(2,2,1)
31 plt.imshow(img)
32 # Plot Red only as subplot 2
33 plt.subplot(2,2,2)
34 plt.imshow((r2 * 255).astype(np.uint8))
35 # Plot Green only as subplot 3
36 plt.subplot(2,2,3)
37 plt.imshow((g2 * 255).astype(np.uint8))
38 # Plot Blue only as subplot 4
39 plt.subplot(2,2,4)
40 plt.imshow((b2 * 255).astype(np.uint8))
```

executed in 743ms, finished 18:32:11 2023-03-26

Out[10]:

<matplotlib.image.AxesImage at 0x7fed292c0c10>

<Figure size 432x288 with 0 Axes>

