PIL

August 28, 2024

1 Python Imaging Library:

Python Imaging Library is a free and open-source additional library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats.

All the images are stored in the form of array

Image –> Array –> Pixels

Every image is broken down into 0-255 pixels

0 - Black

255 - Brightest

[3]:
```
import numpy as np
```

[5]:
```
ones_arr = np.ones((3,3),dtype=int)
```

[7]:
```
ones_arr
```

[7]:
```
array([[1, 1, 1],
       [1, 1, 1],
       [1, 1, 1]])
```

[9]:
```
import matplotlib.pyplot as plt
```

[101]:
```
%matplotlib inline
```

[13]:
```
from PIL import Image
```

[15]:
```
horse_img = Image.open(r'E:\Users\TANISHQ\Downloads\horse.jpg')
```

[17]:
```
horse_img
```

[17]:

[19]:
```
type(horse_img)
```

[19]:
```
PIL.JpegImagePlugin.JpegImageFile
```

1.0.1 .asarray( ) –> Converts input to an array

[21]:
```
horse_arr = np.asarray(horse_img)
horse_arr
```

[21]:
```
array([[[15, 17, 29],
        [15, 17, 29],
        [15, 17, 29],
        …,
        [25, 37, 35],
        [19, 34, 31],
        [14, 30, 27]],

       [[15, 17, 29],
        [15, 17, 29],
        [15, 17, 29],
        …,
        [26, 38, 36],
        [22, 37, 34],
```

```
        [20, 36, 33]],

       [[15, 17, 29],
        [15, 17, 29],
        [15, 17, 29],
        …,
        [28, 40, 38],
        [25, 40, 37],
        [24, 40, 37]],

       …,

       [[49, 50, 44],
        [40, 41, 35],
        [35, 35, 27],
        …,
        [14, 30, 29],
        [13, 25, 25],
        [12, 22, 23]],

       [[45, 50, 44],
        [38, 43, 37],
        [31, 36, 30],
        …,
        [11, 25, 25],
        [12, 24, 24],
        [16, 26, 27]],

       [[31, 41, 33],
        [31, 41, 33],
        [32, 39, 32],
        …,
        [14, 26, 26],
        [16, 26, 27],
        [23, 31, 33]]], dtype=uint8)
```

[145]:
```
horse_red = horse_arr.copy()
```

[147]:
```
horse_red
```

[147]:
```
array([[[15, 17, 29],
        [15, 17, 29],
        [15, 17, 29],
        …,
        [25, 37, 35],
        [19, 34, 31],
        [14, 30, 27]],
```

```
       [[15, 17, 29],
        [15, 17, 29],
        [15, 17, 29],
        …,
        [26, 38, 36],
        [22, 37, 34],
        [20, 36, 33]],

       [[15, 17, 29],
        [15, 17, 29],
        [15, 17, 29],
        …,
        [28, 40, 38],
        [25, 40, 37],
        [24, 40, 37]],

       …,

       [[49, 50, 44],
        [40, 41, 35],
        [35, 35, 27],
        …,
        [14, 30, 29],
        [13, 25, 25],
        [12, 22, 23]],

       [[45, 50, 44],
        [38, 43, 37],
        [31, 36, 30],
        …,
        [11, 25, 25],
        [12, 24, 24],
        [16, 26, 27]],

       [[31, 41, 33],
        [31, 41, 33],
        [32, 39, 32],
        …,
        [14, 26, 26],
        [16, 26, 27],
        [23, 31, 33]]], dtype=uint8)
```

[149]:
```
horse_arr == horse_red
```

[149]:
```
array([[[ True,  True,  True],
        [ True,  True,  True],
```

```
 [ True,  True,  True],
 …,
 [ True,  True,  True],
 [ True,  True,  True],
 [ True,  True,  True]],

[[ True,  True,  True],
 [ True,  True,  True],
 [ True,  True,  True],
 …,
 [ True,  True,  True],
 [ True,  True,  True],
 [ True,  True,  True]],

[[ True,  True,  True],
 [ True,  True,  True],
 [ True,  True,  True],
 …,
 [ True,  True,  True],
 [ True,  True,  True],
 [ True,  True,  True]],

…,

[[ True,  True,  True],
 [ True,  True,  True],
 [ True,  True,  True],
 …,
 [ True,  True,  True],
 [ True,  True,  True],
 [ True,  True,  True]],

[[ True,  True,  True],
 [ True,  True,  True],
 [ True,  True,  True],
 …,
 [ True,  True,  True],
 [ True,  True,  True],
 [ True,  True,  True]],

[[ True,  True,  True],
 [ True,  True,  True],
 [ True,  True,  True],
 …,
 [ True,  True,  True],
 [ True,  True,  True],
 [ True,  True,  True]]])
```

1.0.2 .imshow( ): Displays data as an image with its width and height, i.e., on a 2D regular raster.

[151]:
```
plt.imshow(horse_red)
```

[151]:
```
<matplotlib.image.AxesImage at 0x2a89eafced0>
```

[29]:
```
horse_red.shape  # (height,width,channel)
```

[29]:
```
(2334, 3502, 3)
```

Here we have 3 columns means 3D channel

In PIL if it is

2D channel –> Black and White

3D channel –> RGB > For 3D channel:

0 means Red

1 means Green

2 means Blue

[153]:
```
horse_red[:,:,0]
```

[153]:
```
array([[15, 15, 15, …, 25, 19, 14],
       [15, 15, 15, …, 26, 22, 20],
       [15, 15, 15, …, 28, 25, 24],
       …,
       [49, 40, 35, …, 14, 13, 12],
       [45, 38, 31, …, 11, 12, 16],
       [31, 31, 32, …, 14, 16, 23]], dtype=uint8)
```

[155]:
```
plt.imshow(horse_red[:,:,0])
```

[155]:
```
<matplotlib.image.AxesImage at 0x2a89ebccfd0>
```

[157]:
```
plt.imshow(horse_red[:,:,1])
```

[157]:
```
<matplotlib.image.AxesImage at 0x2a895c57490>
```

[53]:
```
plt.imshow(horse_red[:,:,2])
```

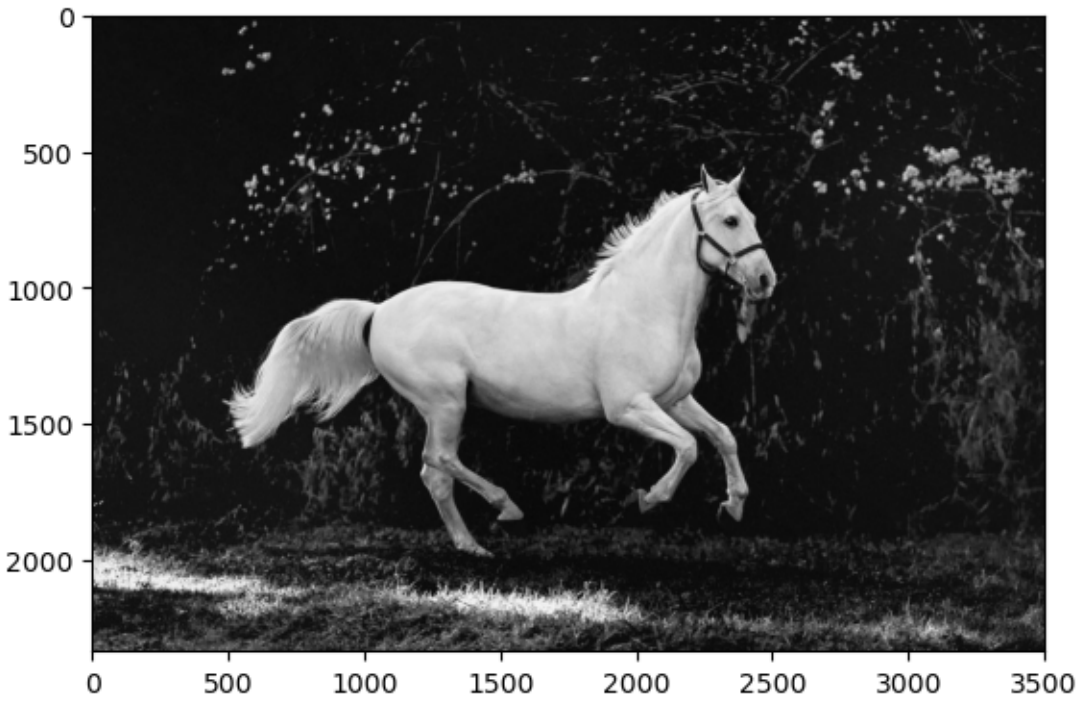
[53]: <matplotlib.image.AxesImage at 0x2a88980d0d0>

[159]:
```
plt.imshow(horse_red[:,:,0], cmap='Greys')
# "Greys" gives Inverted Black and White
```

[159]: <matplotlib.image.AxesImage at 0x2a8a5205dd0>

[161]:
```
plt.imshow(horse_red[:,:,0], cmap='grey')
```

[161]: <matplotlib.image.AxesImage at 0x2a8a526ddd0>



[49]:
```
plt.imshow(horse_red[:,:,1], cmap='grey')
```

[49]: <matplotlib.image.AxesImage at 0x2a8876f6750>

[51]:
```
plt.imshow(horse_red[:,:,2], cmap='grey')
```

[51]:
```
<matplotlib.image.AxesImage at 0x2a889731f10>
```

You can choose colors from google 'matplotlib colormaps' for cmap

[163]:
```
horse_red[:,:,1] = 0
```

[165]:
```
horse_red[:,:,1]
```

[165]:
```
array([[0, 0, 0, …, 0, 0, 0],
       [0, 0, 0, …, 0, 0, 0],
       [0, 0, 0, …, 0, 0, 0],
       …,
       [0, 0, 0, …, 0, 0, 0],
       [0, 0, 0, …, 0, 0, 0],
       [0, 0, 0, …, 0, 0, 0]], dtype=uint8)
```

[167]:
```
plt.imshow(horse_red)
```

[167]:
```
<matplotlib.image.AxesImage at 0x2a89ea57b50>
```

[127]:
```
horse_red[:,:,2]
```

[127]:
```
array([[29, 29, 29, …, 35, 31, 27],
       [29, 29, 29, …, 36, 34, 33],
       [29, 29, 29, …, 38, 37, 37],
       …,
       [44, 35, 27, …, 29, 25, 23],
       [44, 37, 30, …, 25, 24, 27],
       [33, 33, 32, …, 26, 27, 33]], dtype=uint8)
```

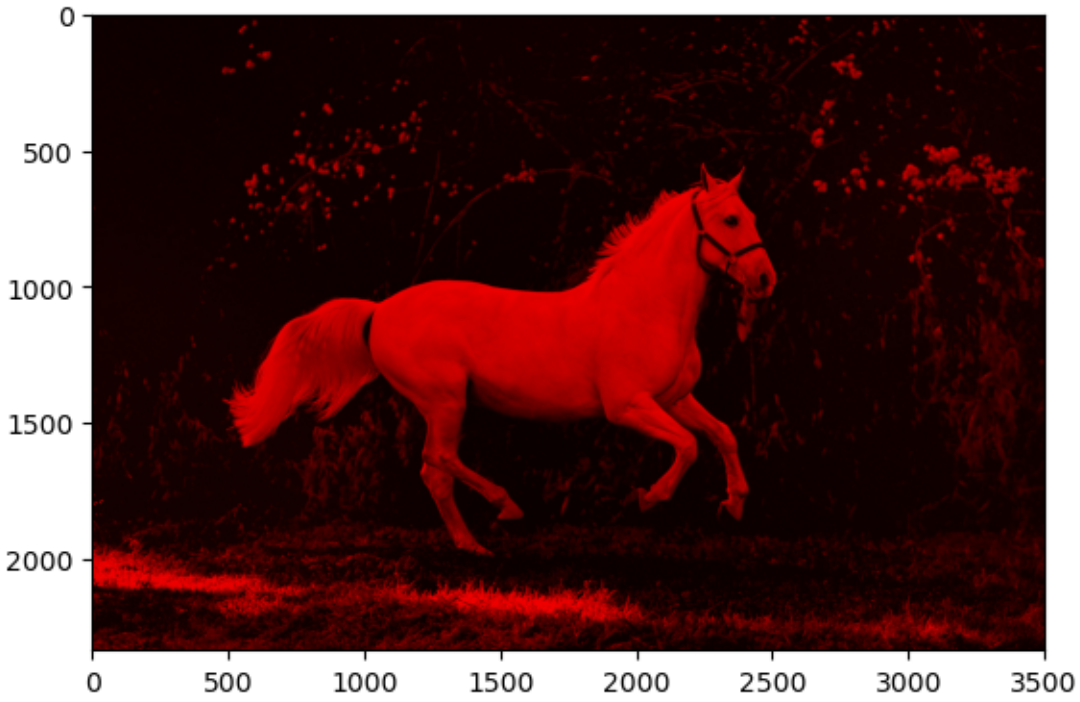
[129]:
```
horse_red[:,:,2]
```

[129]:
```
array([[29, 29, 29, …, 35, 31, 27],
       [29, 29, 29, …, 36, 34, 33],
       [29, 29, 29, …, 38, 37, 37],
       …,
       [44, 35, 27, …, 29, 25, 23],
       [44, 37, 30, …, 25, 24, 27],
       [33, 33, 32, …, 26, 27, 33]], dtype=uint8)
```

[131]:
```
horse_red[:,:,2] = 0
```

[141]:
```
plt.imshow(horse_red)
```

[141]: <matplotlib.image.AxesImage at 0x2a895bb0810>



[182]:
```
horse_r = horse_arr.copy()
```

[172]:
```
horse_g = horse_arr.copy()
```

[174]:
```
horse_b = horse_arr.copy()
```

1.0.3 Here we make 1st element (0 in array) of 3D channel i.e Red=0. Means intensity of Red is 0 that gives image colour of green and blue

[209]:
```
horse_r[:,:,0] = 0   # All the arrays of Red became 0, means Red pixels␣
↪intensity became 0
```

[186]:
```
plt.imshow(horse_r)
```

[186]: <matplotlib.image.AxesImage at 0x2a8a91e8d50>

Below, we maxed intensity of all red pixel arrays to 255

[221]:
```
horse_r[:,:,0] = 255
```

[223]:
```
plt.imshow(horse_r)
```

[223]:
```
<matplotlib.image.AxesImage at 0x2a8b1f5ced0>
```

1.0.4 Here we make 2nd element (0 in array) of 3D channel i.e Green=0. Means intensity of Green is 0 that gives image colour of red and blue

[188]:
```
horse_g[:,:,1] = 0
```

[190]:
```
plt.imshow(horse_g)
```

[190]:
```
<matplotlib.image.AxesImage at 0x2a8a9225dd0>
```

1.0.5 Here we make 2nd element (0 in array) of 3D channel i.e Blue=0. Means intensity of Blue is 0 that gives image colour of green and red

[192]:
```
horse_b[:,:,2] = 0
```

[195]:
```
plt.imshow(horse_b)
```

[195]:
```
<matplotlib.image.AxesImage at 0x2a8b20a8310>
```

[232]:
```
horse_b[:,:,2] = 200  # Here we increased intensity of all BLue array pixels
```

[228]:
```
plt.imshow(horse_b)
```

[228]:
```
<matplotlib.image.AxesImage at 0x2a8b1fef490>
```


0
500
1000
1500
2000
0
500
1000
1500
2000
2500
3000
3500