BASIC OPERATIONS ON IMAGES

1. Hồ Thái Ngọc
2. ThS. Võ Duy Nguyên
3. TS. Nguyễn Tấn Trần Minh Khang

Contents

1. Accessing and Modifying pixel valuesDrawing Line
2. Accessing Image Properties
3. Splitting and Merging Image Channels
4. Making Borders for Images (Padding)

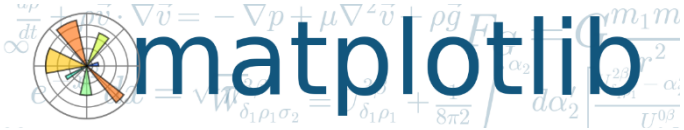
Basic operations on images

KÍCH THƯỚC ẢNH

Bài toán 01 – Kích thước ảnh

- Bài toán: viết chương trình đọc và hiển thị ảnh có tên `Lindsay.tif`. Sau đó xuất kích thước ảnh.

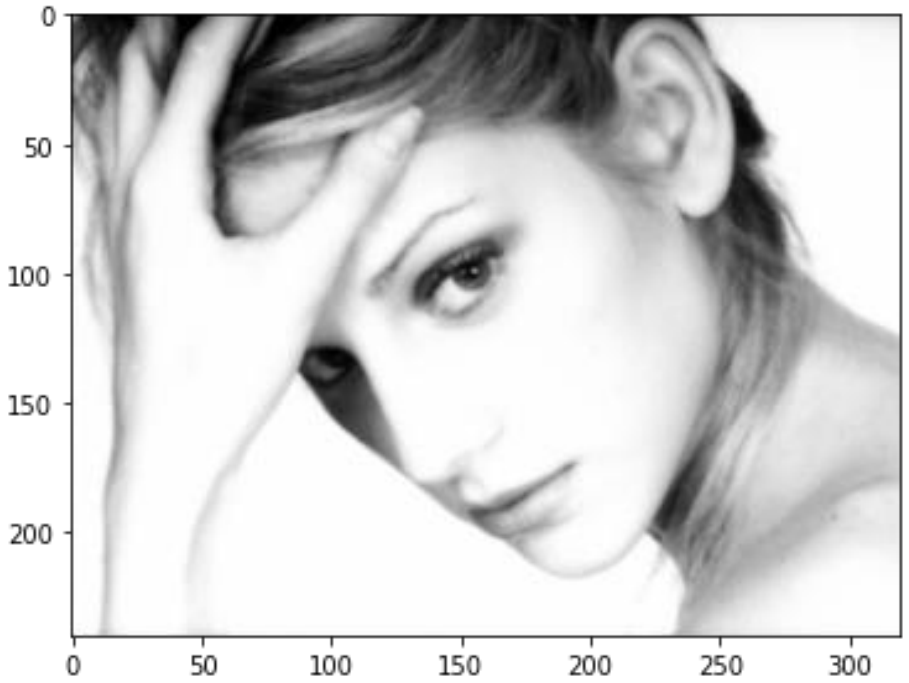
Bài toán 01 – Kích thước ảnh




```
1.import matplotlib.pyplot as plt
2.import numpy as np
3.import cv2
4.img = cv2.imread('lindsay.tif',0)
5.img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
6.plt.figure(figsize=(6, 6))
7.plt.imshow(img)
8.print(img.shape)
```

Bài toán 01 – Kích thước ảnh

```
1.  import matplotlib.pyplot as plt
2.  import numpy as np
3.  import cv2
4.  img = cv2.imread('lindsay.tif',0)
5.  img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
6.  plt.figure(figsize=(6, 6))
7.  plt.imshow(img)
8.  print(img.shape)
```




```
(240, 320, 3)
```

Basic operations on images

ACCESSING AND MODIFYING PIXEL

Accessing and Modifying pixel

- Access a pixel value by its row and column coordinates.
- An array of Blue, Green, Red values (BGR image).
- Corresponding intensity (grayscale image).

Accessing and Modifying pixel

- Access and modify all B,G,R values using numpy array methods.
- array.item() and array.itemset() is considered to be better.

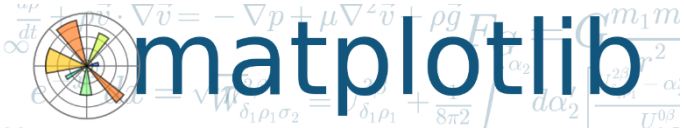
Bài toán 02 – Pixel

- Bài toán: viết chương trình đọc và hiển thị ảnh có tên `Lindsay.tif`. Xuất thông tin của hai điểm ảnh có tọa độ: `[200, 200],[50, 90]`.

OpenCV

matplotlib

Bài toán 02 – Pixel

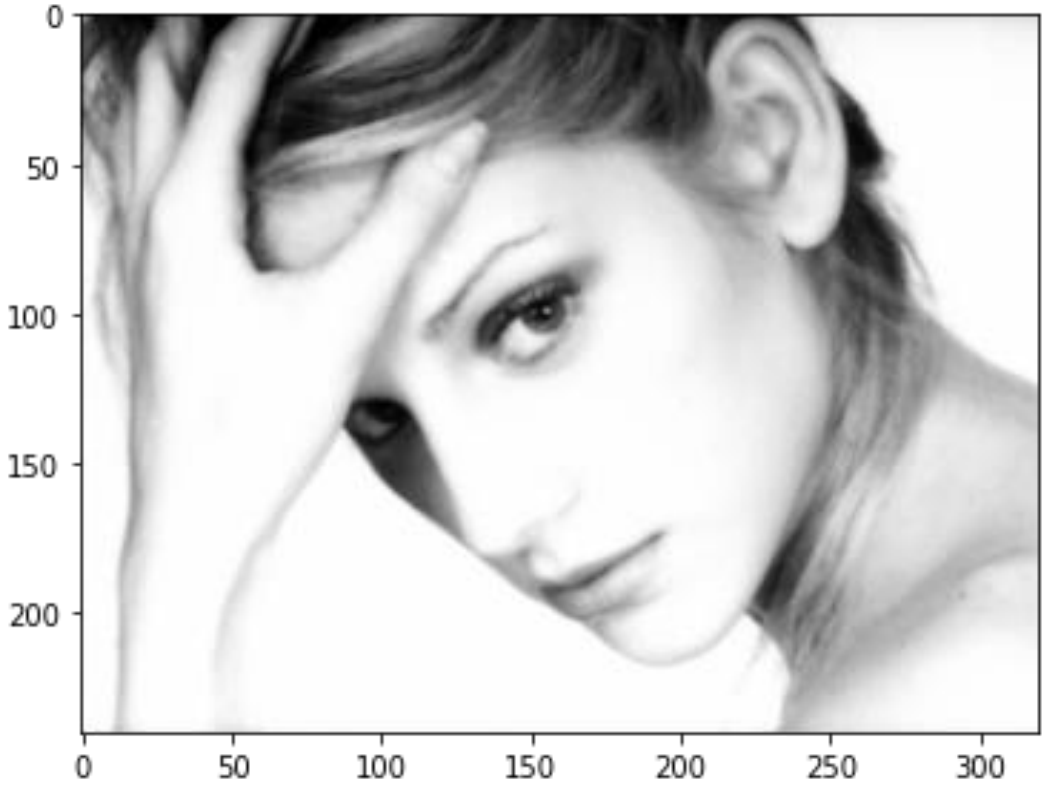



```
11.import matplotlib.pyplot as plt
12.import numpy as np
13.import cv2
14.img = cv2.imread('lindsay.tif',0)
15.img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
16.plt.figure(figsize=(6, 6))
17.plt.imshow(img)
18.px_1 = img[200, 200]
19.px_2 = img[50, 90]
20.print('px1: ', px_1)
21.print('px2: ', px_2)
```

Bài toán 02 – Pixel

```
11. import matplotlib.pyplot as plt
12. import numpy as np
13. import cv2
14. img = cv2.imread('lindsay.tif',0)
15. img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
16. plt.figure(figsize=(6, 6))
17. plt.imshow(img)
18. px_1 = img[200, 200]
19. px_2 = img[50, 90]
20. print('px1: ', px_1)
21. print('px2: ', px_2)
```

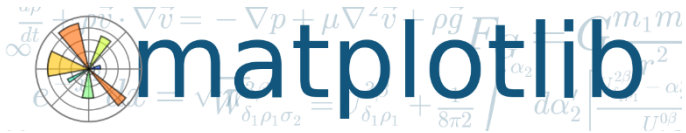
```
px1:  [255 255 255]
px2:  [169 169 169]
```



Bài toán 03 – Thay đổi Pixel

- Bài toán: viết chương trình đọc và hiển thị ảnh có tên `Lindsay.tif`. Thay đổi thông tin của điểm ảnh có tọa độ: `[200,200]` thành `[100,101,102]`.

Bài toán 03 – Thay đổi Pixel




```
11.import matplotlib.pyplot as plt
12.import numpy as np
13.import cv2
14.img = cv2.imread('lindsay.tif',0)
15.img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
16.plt.figure(figsize=(6, 6))
17.plt.imshow(img)
18.px = img[200, 200]
19.print('px: ', px)
20.img.itemset((200, 200,0), 100)
21.img.itemset((200, 200,1), 101)
22.img.itemset((200, 200,2), 103)
23.px = img[200, 200]
24.print('px: ', px)
```

```
px:  [255 255 255]
px:  [100 101 103]
```

Basic operations on images

ACCESSING IMAGE PROPERTIES

Accessing Image Properties

- Image properties include number of rows, columns and channels, type of image data, number of pixels etc.
  - + number of rows, columns and channels using img.shape
  - + type of image data using img.dtype
  - + number of pixels using img.size

Accessing Image Properties

- Shape of image is accessed by img.shape
  - + Image is color: a tuple of number of rows, columns and channels
  - + Image is grayscale: number of rows and columns

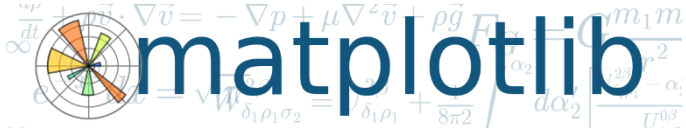
Bài toán 04 – Thuộc tính cơ bản

- Bài toán: viết chương trình đọc và hiển thị ảnh có tên `Lindsay.tif`. Thể hiện thông tin cơ bản của ảnh.

OpenCV

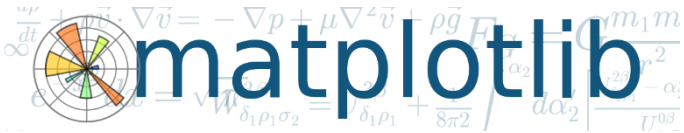
matplotlib

Bài toán 04 – Thuộc tính cơ bản




```
11.import matplotlib.pyplot as plt
12.import numpy as np
13.import cv2
14.img = cv2.imread('lindsay.tif',0)
15.img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
16.plt.figure(figsize=(6, 6))
17.plt.imshow(img)
18.print('Shape image color: ', img.shape)
19.print('Total number of pixels: ', img.size)
20.print('Image datatype: ', img.dtype)
```

Bài toán 04 – Thuộc tính cơ bản




```
11.import matplotlib.pyplot as plt
12.import numpy as np
13.import cv2
14.img = cv2.imread('lindsay.tif',0)
15.img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
16.plt.figure(figsize=(6, 6))
17.plt.imshow(img)
18.print('Shape image color: ', img.shape)
19.print('Total number of pixels: ', img.size)
20.print('Image datatype: ', img.dtype)
```

```
Shape image color:  (240, 320)
Total number of pixels:  76800
Image datatype:  uint8
```

Basic operations on images

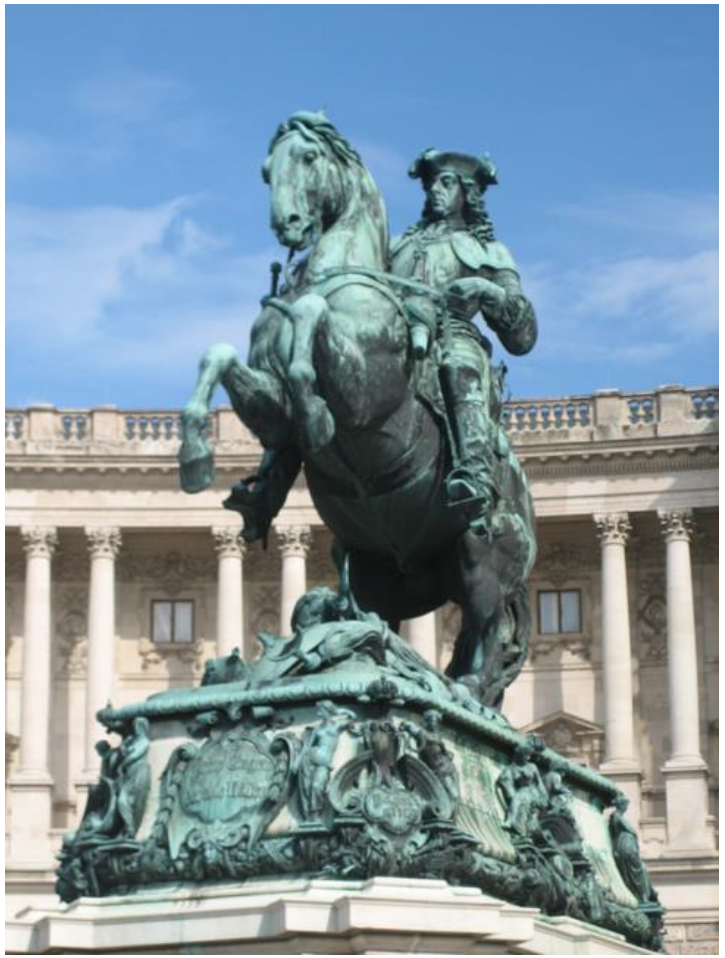
SPLITTING AND MERGING IMAGE CHANNELS

Splitting and Merging Image Channels

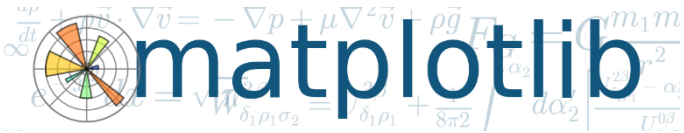
- The B,G,R channels of an image can be split into their individual planes.
- The individual channels can be merged back together to form a BGR image.

Bài toán 05 – Splitting

- Bài toán: viết chương trình đọc và hiển thị ảnh có tên `statue.png`. Tách kênh màu R, G, B của ảnh, sau đó hiển thị ảnh từng kênh.

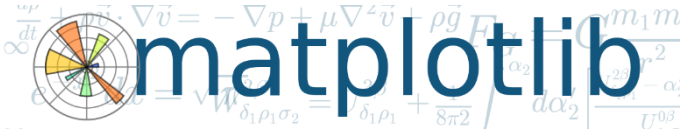


Bài toán 05 – Splitting




```
11.  import matplotlib.pyplot as plt
12.  import numpy as np
13.  import cv2
14.  img = cv2.imread('statue.png', 1)
15.  img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
16.  plt.figure(figsize=(6, 6))
17.  plt.imshow(img)
18. R,G,B = cv2.split(img)
19. print(type(R), R.shape, R.size)
20. print(R)
21. print(type(G), G.shape, G.size)
22. print(G)
23. print(type(B), B.shape, B.size)
24. print(B)
```

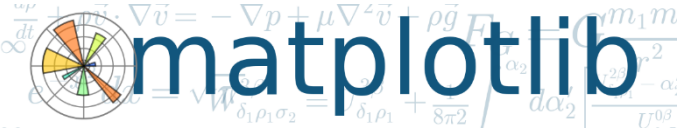
Bài toán 05 – Splitting




```
11.  import matplotlib.pyplot as plt
12.  import numpy as np
13.  import cv2
14.  img = cv2.imread('statue.png', 1)
15.  img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
16.  plt.figure(figsize=(6, 6))
17.  plt.imshow(img)
18.  R,G,B = cv2.split(img)
19.  print(type(R), R.shape, R.size)
20.  print(R)
21.  print(type(G), G.shape, G.size)
22.  print(G)
23.  print(type(B), B.shape, B.size)
24.  print(B)
```

```
R <class 'numpy.ndarray'> (648, 486) 314928
[[ 99 101  99 ...  84  84  85]
 [ 98  97 101 ...  86  85  85]
 [ 99  98  99 ...  84  85  80]
 ...
 [169 175 177 ... 191 192 191]
 [104 102 105 ... 170 185 193]
 [ 90  88  89 ... 161 166 179]]
```

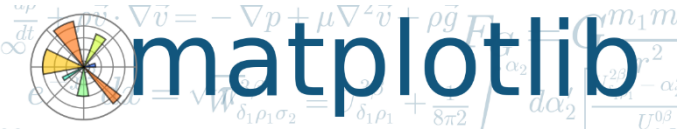
Bài toán 05 – Splitting




```
11.  import matplotlib.pyplot as plt
12.  import numpy as np
13.  import cv2
14.  img = cv2.imread('statue.png', 1)
15.  img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
16.  plt.figure(figsize=(6, 6))
17.  plt.imshow(img)
18.  R,G,B = cv2.split(img)
19.  print(type(R), R.shape, R.size)
20.  print(R)
21.  print(type(G), G.shape, G.size)
22.  print(G)
23.  print(type(B), B.shape, B.size)
24.  print(B)
```

```
G <class 'numpy.ndarray'> (648, 486) 314928
[[149 151 152 ... 134 135 133]
 [151 151 151 ... 134 133 133]
 [152 152 152 ... 135 134 132]
 ...
 [151 156 158 ... 173 175 174]
 [ 88  89  90 ... 153 168 176]
 [ 77  75  76 ... 149 150 162]]
```

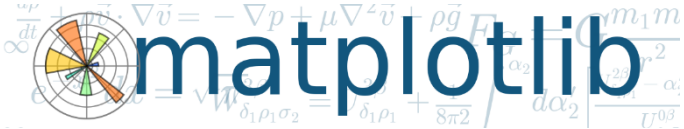
Bài toán 05 – Splitting




```
11.  import matplotlib.pyplot as plt
12.  import numpy as np
13.  import cv2
14.  img = cv2.imread('statue.png', 1)
15.  img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
16.  plt.figure(figsize=(6, 6))
17.  plt.imshow(img)
18.  R,G,B = cv2.split(img)
19.  print(type(R), R.shape, R.size)
20.  print(R)
21.  print(type(G), G.shape, G.size)
22.  print(G)
23.  print(type(B), B.shape, B.size)
24.  print(B)
```

```
B <class 'numpy.ndarray'> (648, 486) 314928
[[205 207 206 ... 191 189 187]
 [206 207 207 ... 189 188 187]
 [208 207 207 ... 189 190 188]
 ...
 [122 128 131 ... 164 160 158]
 [ 63  62  62 ... 140 153 161]
 [ 68  65  64 ... 132 133 146]]
```

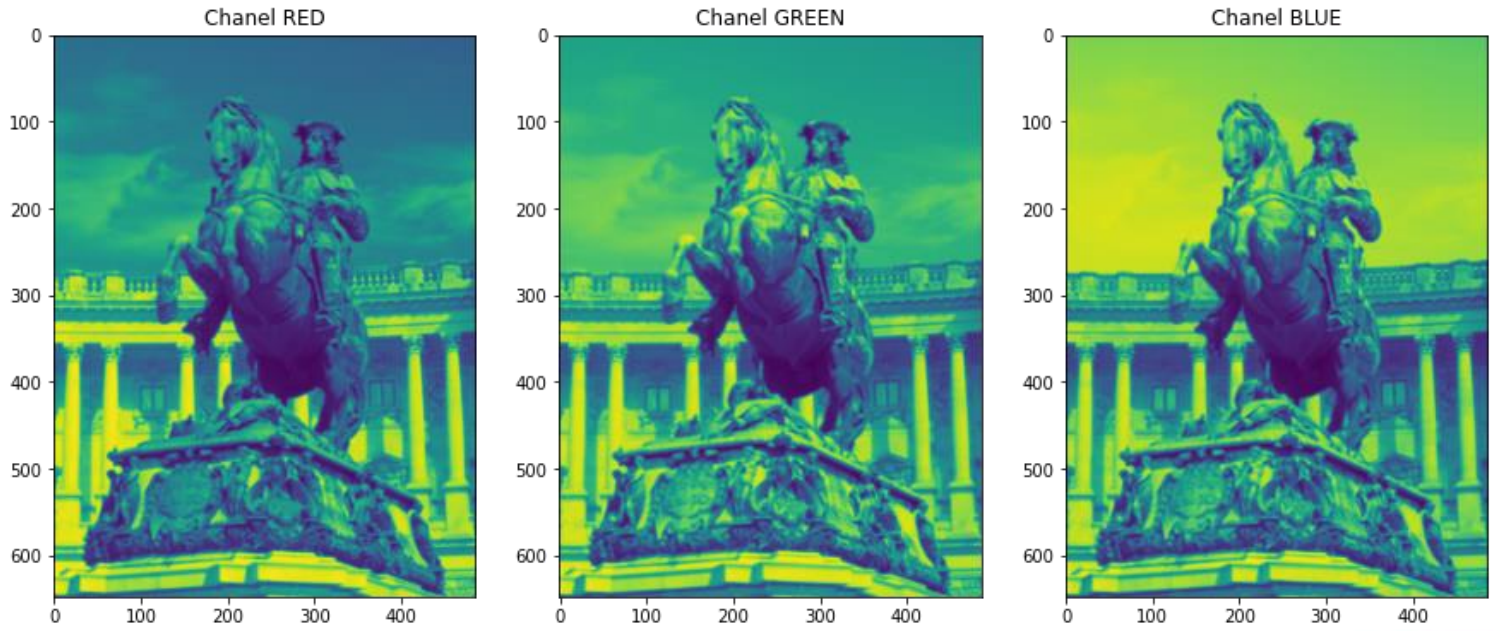
Bài toán 05 – Splitting




```
11.    import matplotlib.pyplot as plt
12.    import numpy as np
13.    import cv2
14.    img = cv2.imread('statue.png', 1)
15.    img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
16.    plt.figure(figsize=(6, 6))
17.    plt.imshow(img)
18.    R,G,B = cv2.split(img)
19.    print(type(R), R.shape, R.size)
20.    print(R)
21.    print(type(G), G.shape, G.size)
22.    print(G)
23.    print(type(B), B.shape, B.size)
24.    print(B)
25.plt.figure(figsize=(15,15))
26.plt.subplot(1,3,1),plt.imshow(R, interpolation = 'bicubic')
27.plt.title('Chanel RED')
28.plt.subplot(1,3,2),plt.imshow(G, interpolation = 'bicubic')
29.plt.title('Chanel GREEN')
30.plt.subplot(1,3,3),plt.imshow(B, interpolation = 'bicubic')
31.plt.title('Chanel BLUE')
```

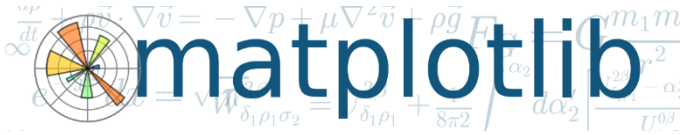
Bài toán 05 – Splitting

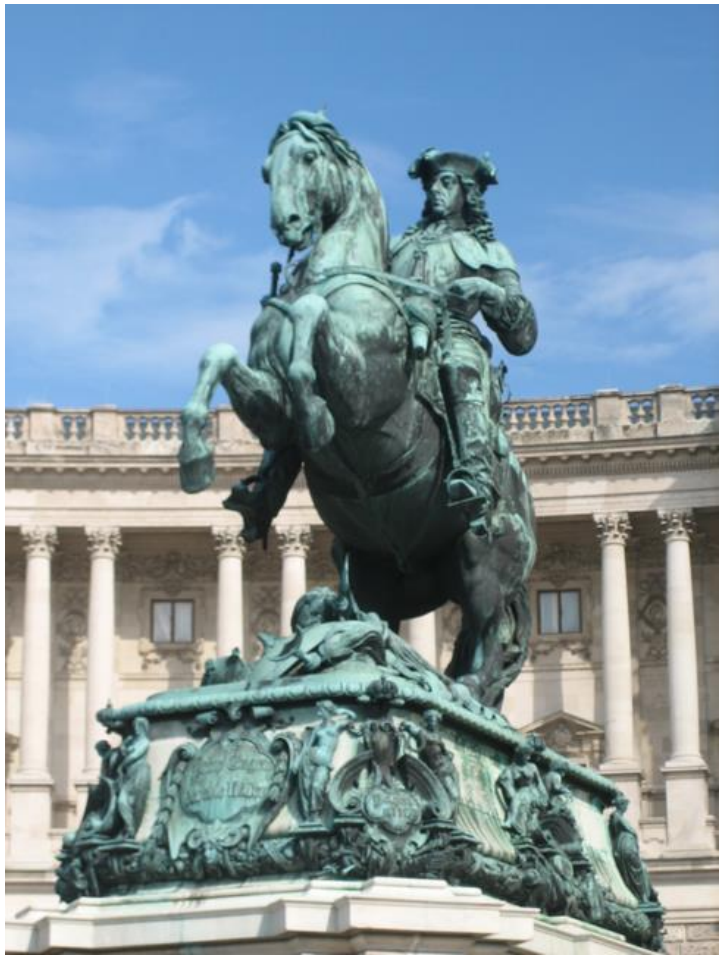
```
11.  import matplotlib.pyplot as plt
12.  import numpy as np
13.  import cv2
14.  img = cv2.imread('statue.png', 1)
15.  img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
16.  plt.figure(figsize=(6, 6))
17.  plt.imshow(img)
18.  R,G,B = cv2.split(img)
19.  print(type(R), R.shape, R.size)
20.  print(R)
21.  print(type(G), G.shape, G.size)
22.  print(G)
23.  print(type(B), B.shape, B.size)
24.  print(B)
25.  plt.figure(figsize=(15,15))
26.  plt.subplot(1,3,1),plt.imshow(R, interpolation = 'bicubic')
27.  plt.title('Chanel RED')
28.  plt.subplot(1,3,2),plt.imshow(G, interpolation = 'bicubic')
29.  plt.title('Chanel GREEN')
30.  plt.subplot(1,3,3),plt.imshow(B, interpolation = 'bicubic')
31.  plt.title('Chanel BLUE')
```



Bài toán 06 – Merging Image Channels

- Bài toán: viết chương trình đọc và hiển thị ảnh có tên `statue.png`. Tách kênh màu R, G, B của ảnh, sau đó gộp lại và hiển thị ảnh kết quả.



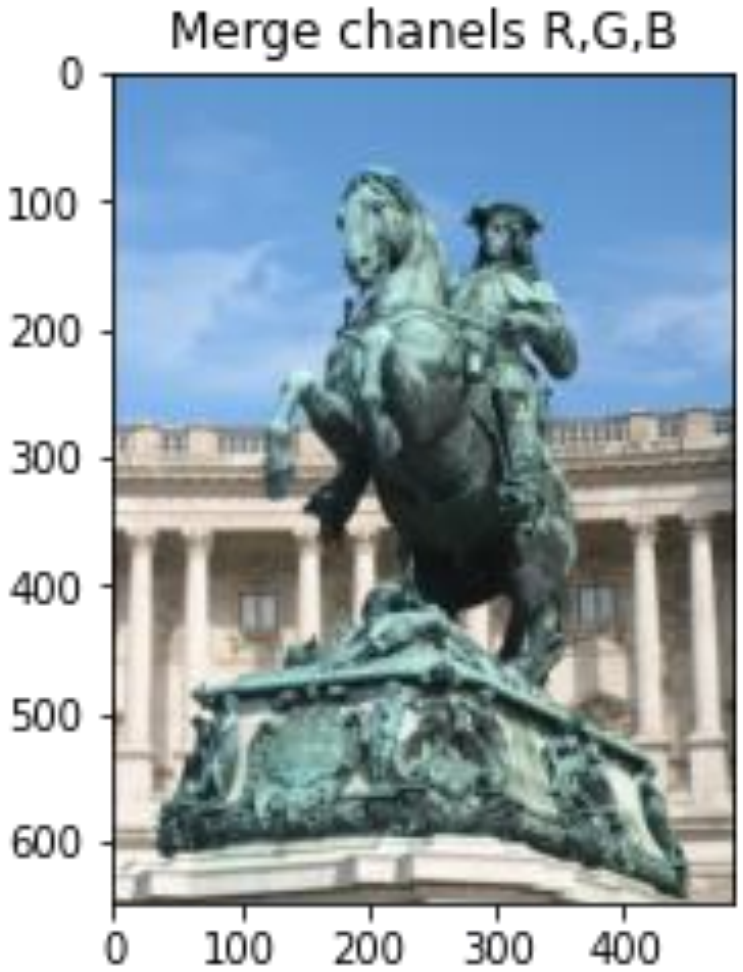


Bài toán 06 – Merging Image Channels


```
11.import matplotlib.pyplot as plt
12.import numpy as np
13.import cv2
14.img = cv2.imread('statue.png', 1)
15.img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
16.plt.figure(figsize=(6, 6))
17.plt.imshow(img)
18.R,G,B = cv2.split(img)
19.img = cv2.merge((R, G, B))
20.plt.figure(figsize=(6,6))
21.plt.subplot(1,2,1),plt.imshow(img)
22.plt.title('Merge chanels R,G,B')
```

Bài toán 06 – Merging Image Channels

```
11.import matplotlib.pyplot as plt
12.import numpy as np
13.import cv2
14.img = cv2.imread('statue.png', 1)
15.img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
16.plt.figure(figsize=(6, 6))
17.plt.imshow(img)
18.R,G,B = cv2.split(img)
19.img = cv2.merge((R, G, B))
20.plt.figure(figsize=(6,6))
21.plt.subplot(1,2,1),plt.imshow(img)
22.plt.title('Merge chanels R,G,B')
```



Basic operations on images

MAKING BORDERS FOR IMAGES (PADDING)

Making Borders for Images (Padding)

- Use `cv2.copyMakeBorder()` to create a border around the image, something like a photo frame.

OpenCV

matplotlib

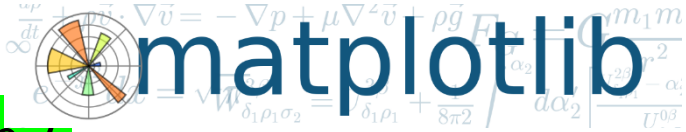
Making Borders for Images (Padding)

- This function takes following arguments:
  - + **src:** input image
  - + **top**, **bottom**, **left**, **right:** border width in number of pixels in corresponding directions
  - + Flag defining what kind of border to be added
    - `cv2.BORDER_CONSTANT`
    - `cv2.BORDER_REFLECT`
    - `cv2.BORDER_REFLECT_101 or cv2.BORDER_DEFAULT`
    - `cv2.BORDER_REPLICATE`
    - `cv2.BORDER_WRAP`
  - + **value:** Color of border if border type is `cv2.BORDER_CONSTANT`

Bài toán 07 – Making Borders

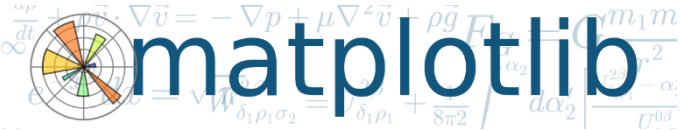
- Bài toán: viết chương trình đọc và hiển thị ảnh có tên `Lindsay.tif`. Thực hiện making borders, sau đó hiển thị ảnh kết quả.

Bài toán 07 – Making Borders



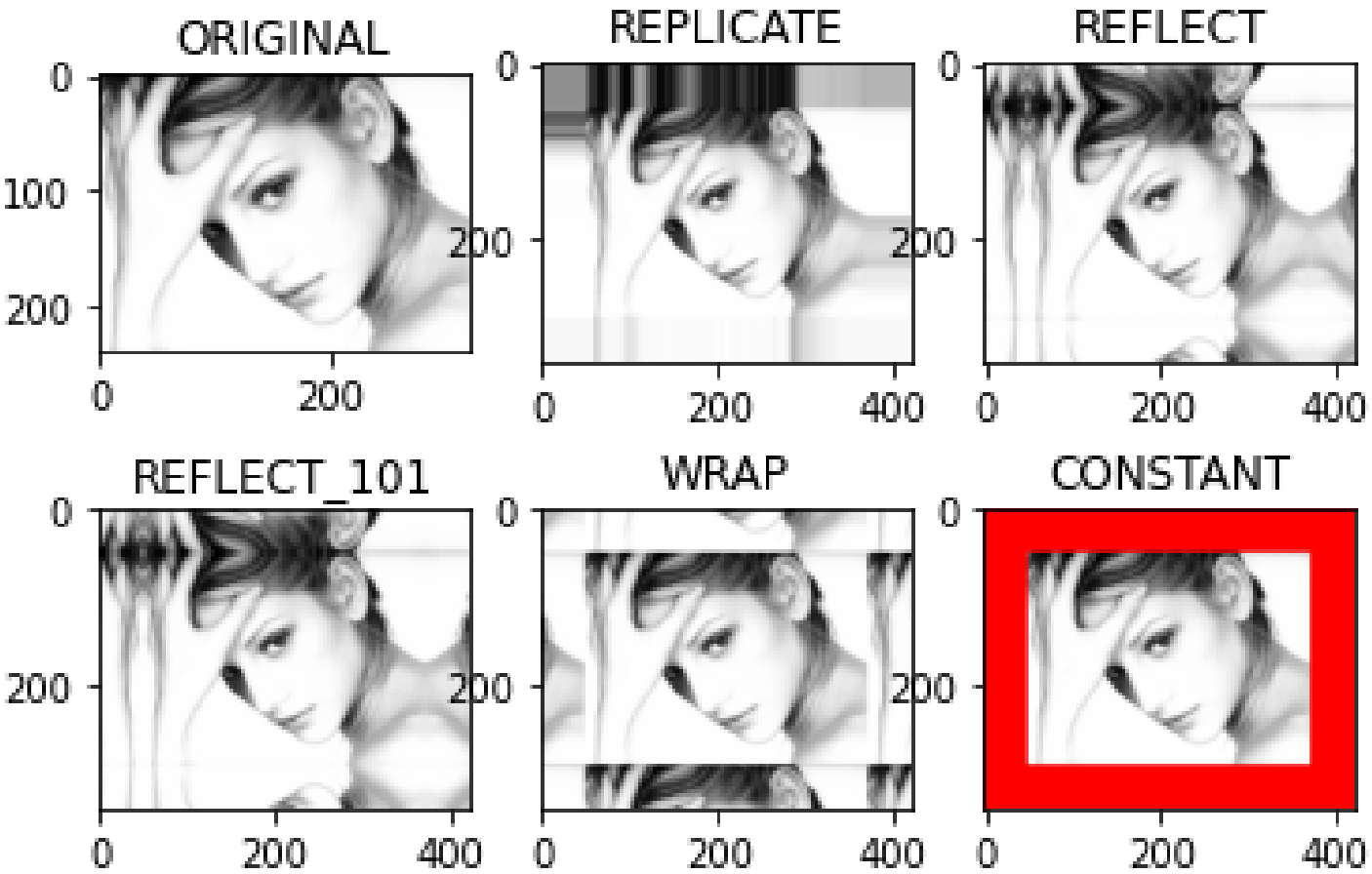

```
11. import matplotlib.pyplot as plt
12. import numpy as np
13. import cv2
14. img = cv2.imread('lindsay.tif', 1)
15. replicate = cv2.copyMakeBorder(
    img,50,50,50,50,cv2.BORDER_REPLICATE)
16. reflect = cv2.copyMakeBorder(
    img,50,50,50,50,cv2.BORDER_REFLECT)
17. reflect101 = cv2.copyMakeBorder(
    img,50,50,50,50,cv2.BORDER_REFLECT_101)
```

Bài toán 07 – Making Borders




```
11.  import matplotlib.pyplot as plt
12.  import numpy as np
13.  import cv2
14.  img = cv2.imread('lindsay.tif', 1)
15.  replicate = cv2.copyMakeBorder(
         img,50,50,50,50,cv2.BORDER_REPLICATE)
16.  reflect = cv2.copyMakeBorder(
         img,50,50,50,50,cv2.BORDER_REFLECT)
17.  reflect101 = cv2.copyMakeBorder(
         img,50,50,50,50,cv2.BORDER_REFLECT_101)
18.wrap = cv2.copyMakeBorder(
     img,50,50,50,50,cv2.BORDER_WRAP)
19.constant= cv2.copyMakeBorder(
     img,50,50,50,50,cv2.BORDER_CONSTANT,valu
e=[255,0,0])
```

Bài toán 07 – Making Borders



OpenCV

matplotlib

Chúc các bạn học tốt

Thân ái chào tạm biệt các bạn

ĐẠI HỌC QUỐC GIA TP.HCM

ĐẠI HỌC CÔNG NGHỆ THÔNG TIN TP.HCM

TOÀN DIỆN – SÁNG TẠO – PHỤNG SỰ