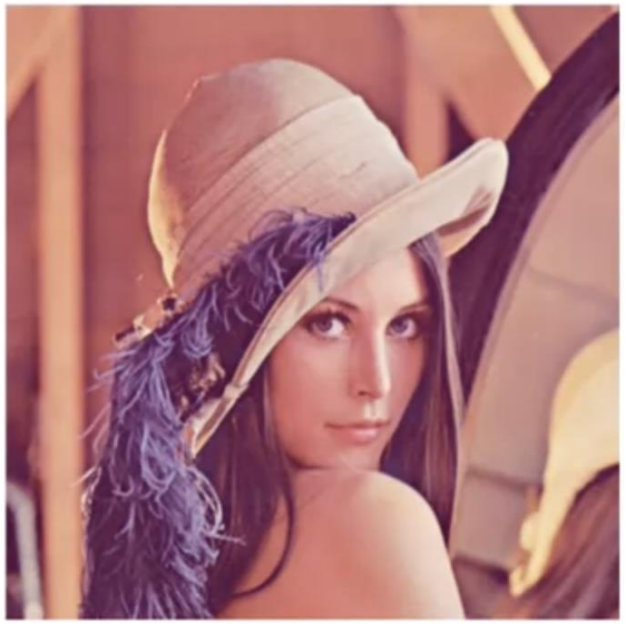
# Week 2
# Image Processing with Opencv & Pillow

ThS. Cáp Phạm Đình Thăng

# Digital Image - Ảnh số

- Ảnh số có thể được hiểu là một ma trận các số.
- Ảnh grayscale: ảnh được tạo thành từ nhiều mức xám khác nhau. Ảnh xám thường được sử dụng trong các bài toán xử lý ảnh hoặc thị giác máy tính



# Intensity Values – Giá trị cường độ

- Trong thế giới thực, một hình ảnh có thể có số lượng giá trị gần như không giới hạn, nhưng hình ảnh kỹ thuật số có giá trị cường độ từ 0 đến 255.


- Thanh bar thể hiện mối quan hệ giữa các sắc thái khác nhau của mức xám và các giá trị số. Các mức xám đậm hơn có giá trị thấp gần với 0 là màu đen và các mức xám nhạt hơn có giá trị cao gần với 255 là màu trắng.

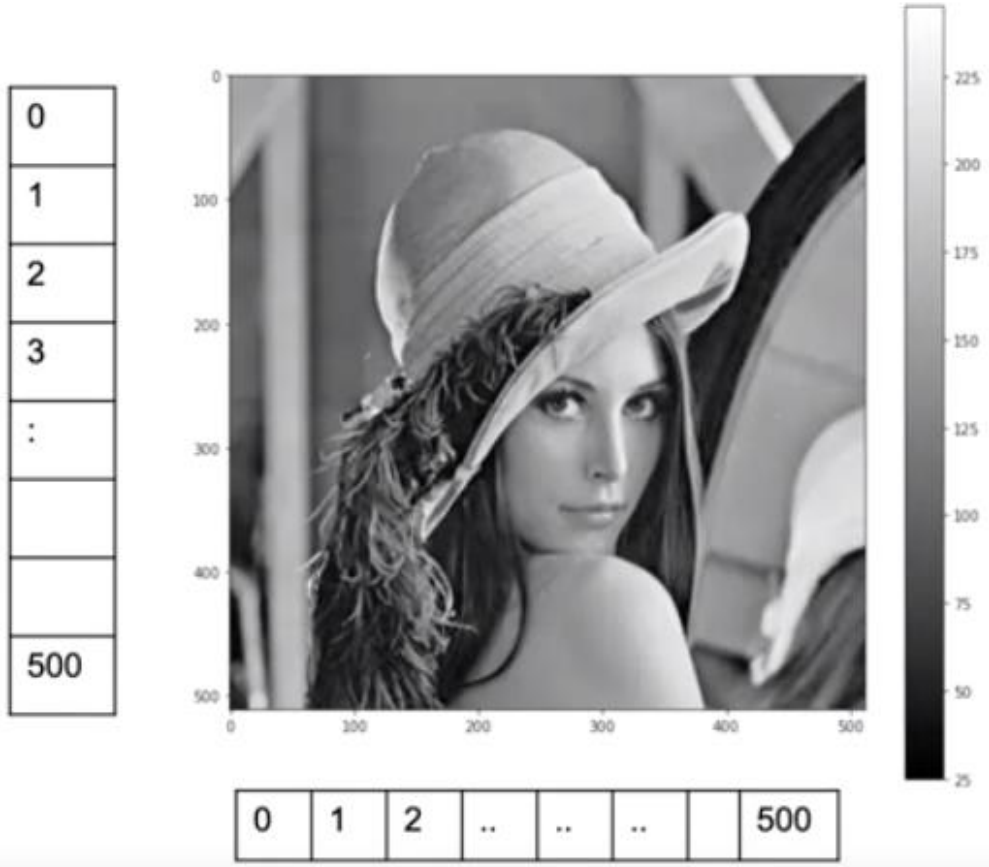
# Biểu diễn ảnh số:

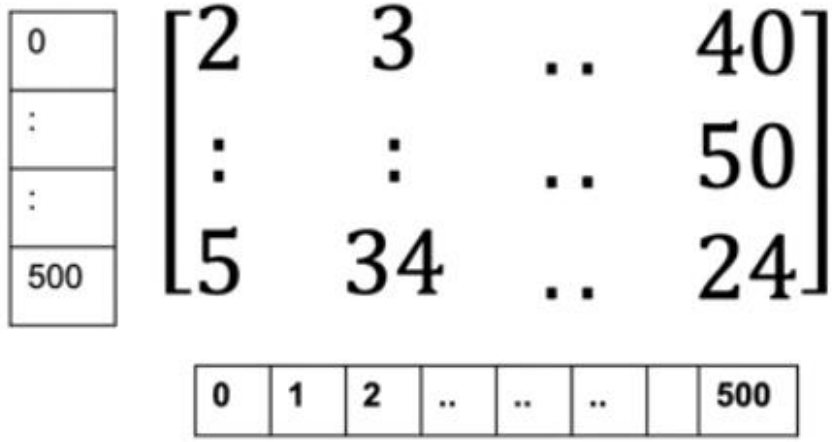

Columns

rows

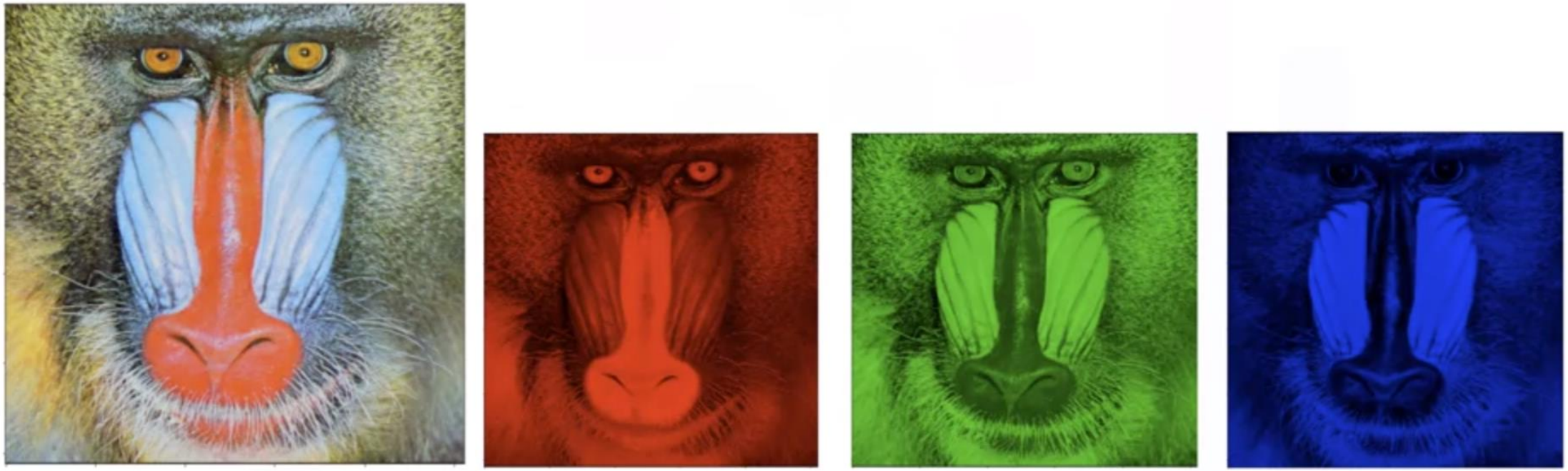
$$\begin{bmatrix} 2 & 3 & .. & 40 \\ : & : & .. & 50 \\ 5 & 34 & .. & 24 \end{bmatrix}$$

# Biểu diễn ảnh số



# Ảnh RGB



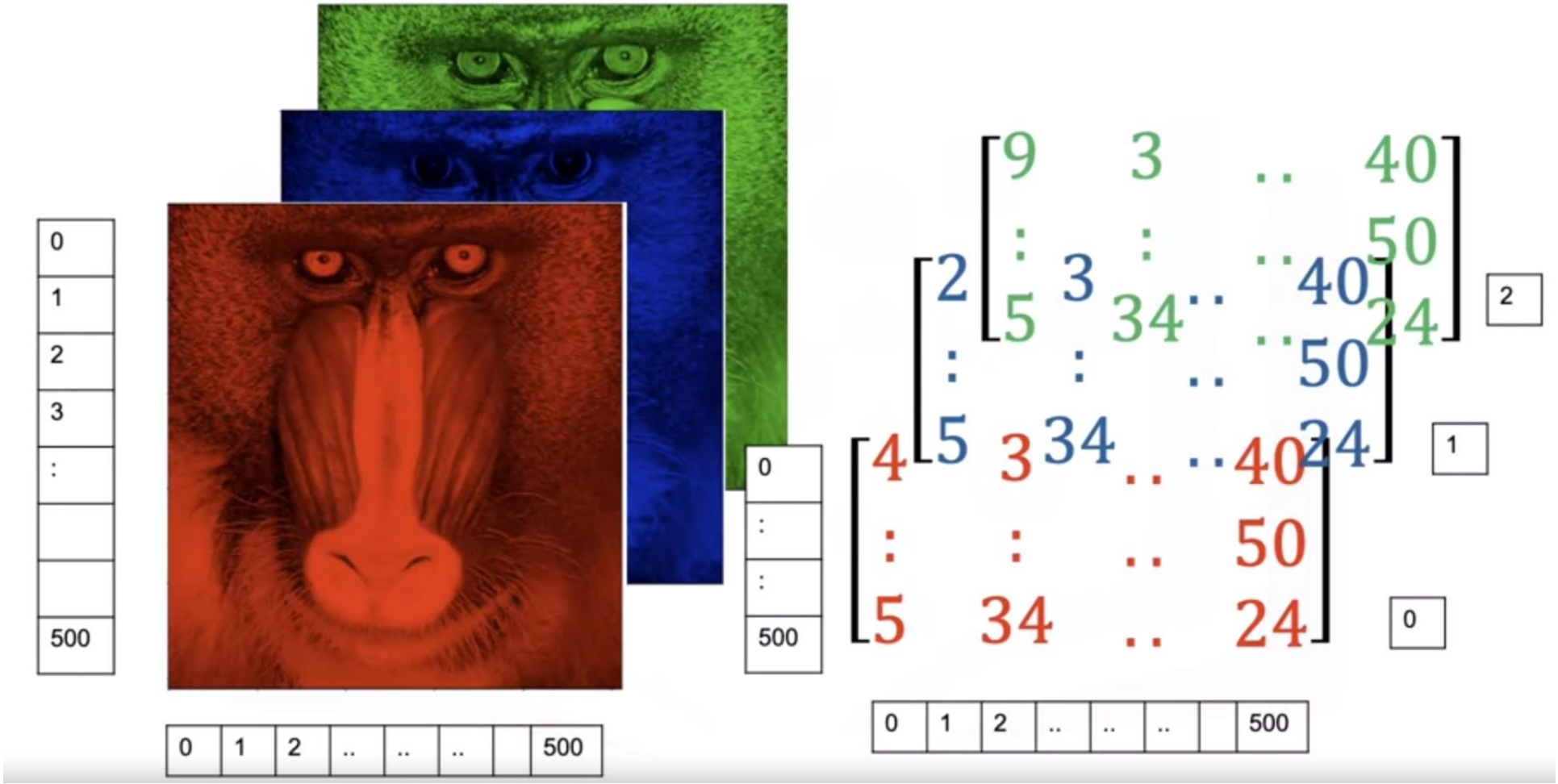
# Ảnh RGB

$$\begin{bmatrix} 9 & 3 & .. & 40 \\ : & : & .. & 50 \\ 5 & 34 & .. & 24 \end{bmatrix}$$

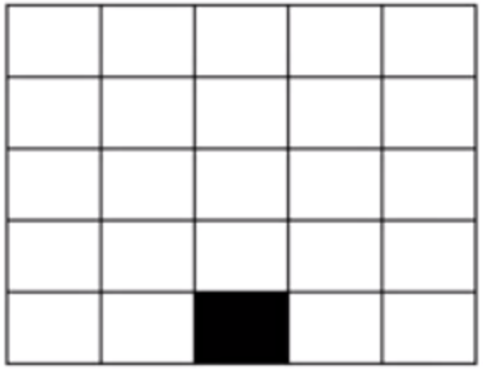
$$\begin{bmatrix} 2 & 3 & .. & 40 \\ : & : & .. & 50 \\ 5 & 34 & .. & 24 \end{bmatrix}$$

$$\begin{bmatrix} 4 & 3 & .. & 40 \\ : & : & .. & 50 \\ 5 & 34 & .. & 24 \end{bmatrix}$$

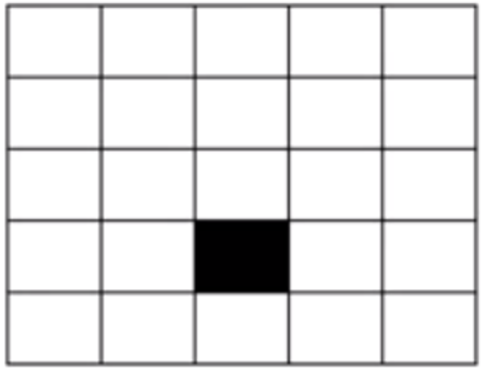
# Biểu diễn ảnh RGB



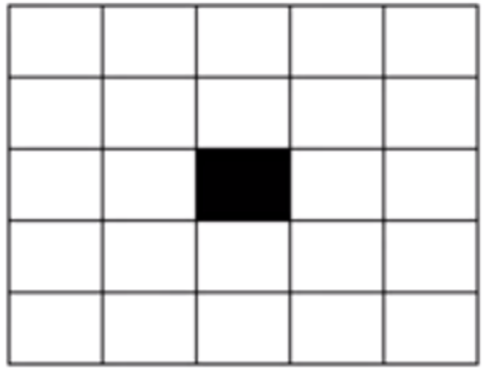
# Video:



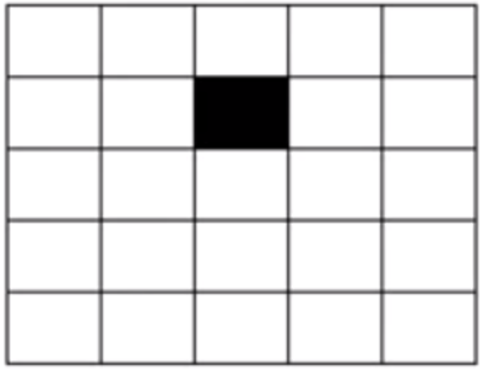
$t = 0$



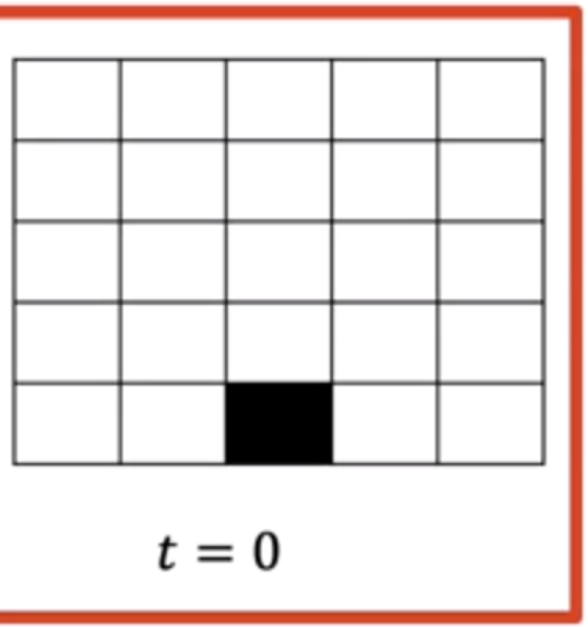
$t = 1$

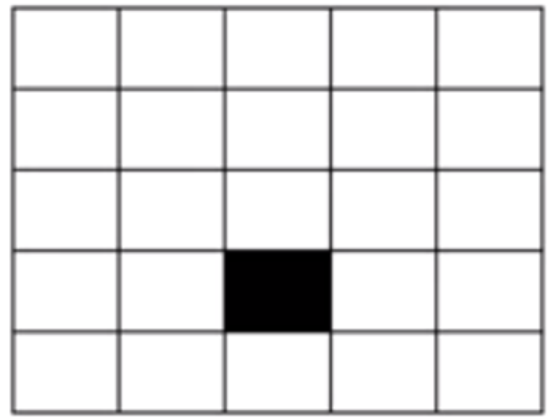


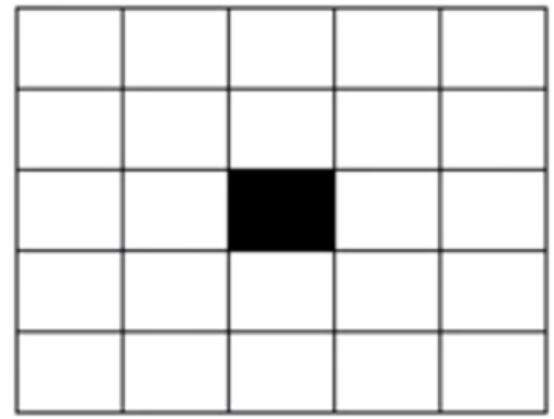
$t = 2$

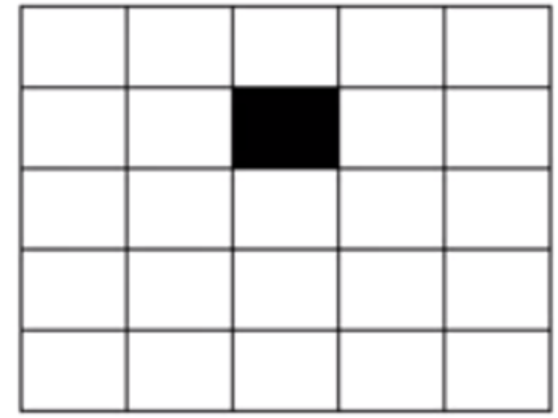


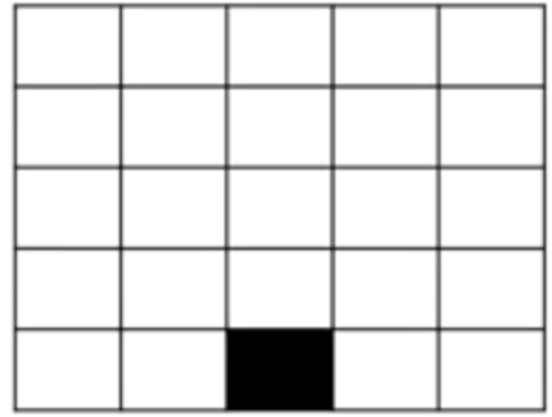
$t = 3$

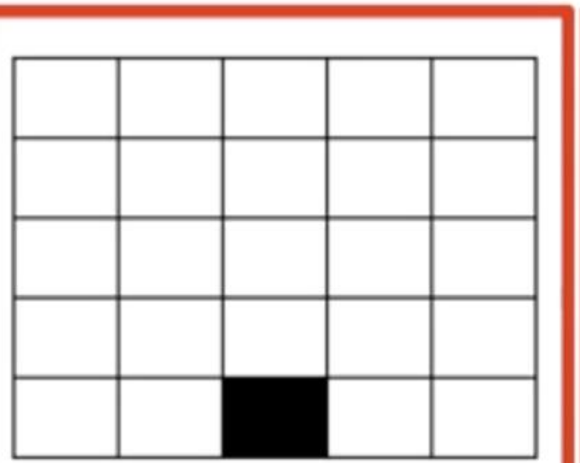

$t = 0$


$t = 1$

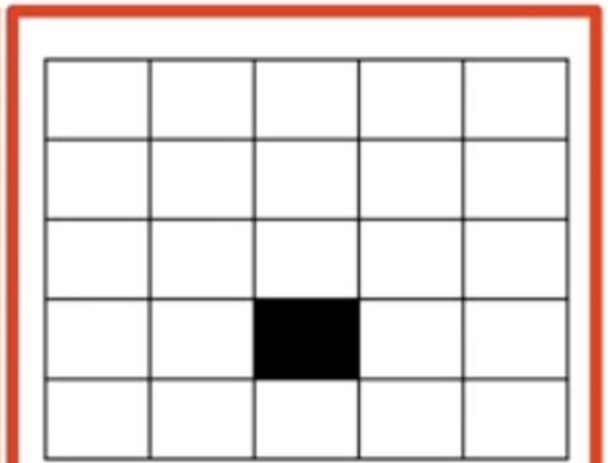

$t = 2$


$t = 3$

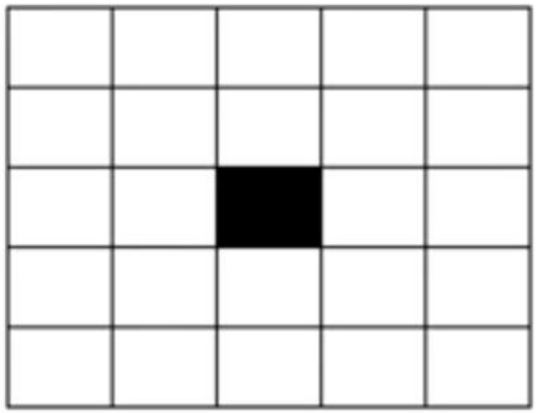




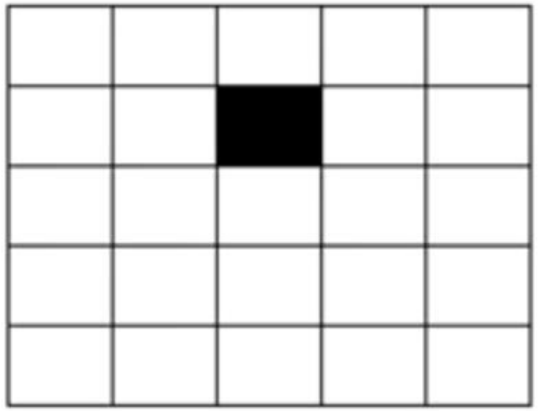
$t = 0$



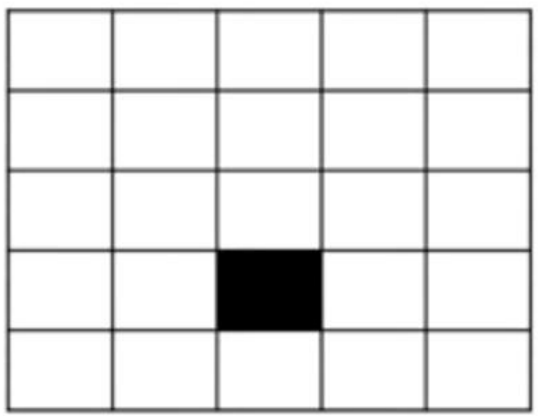
$t = 1$

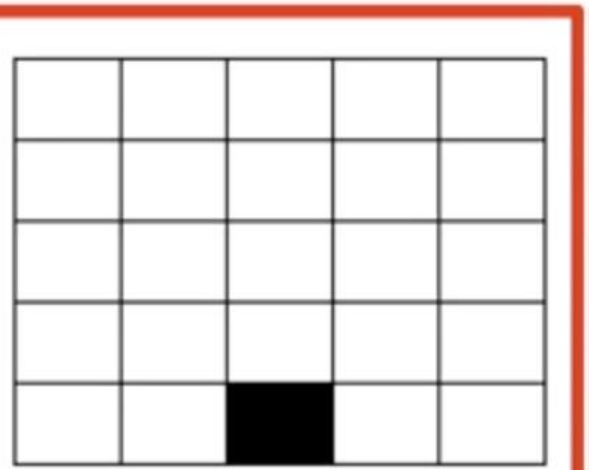


$t = 2$

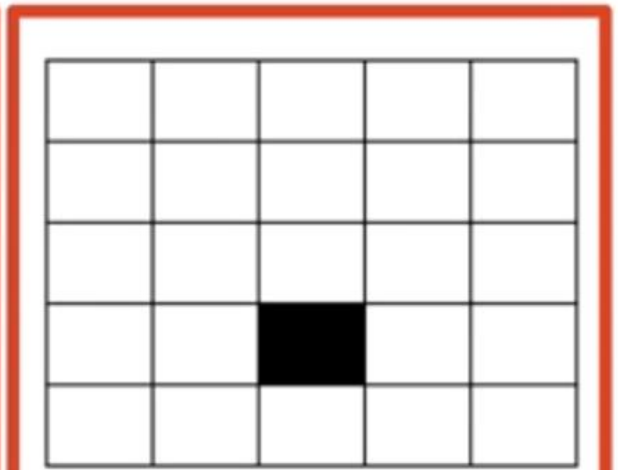


$t = 3$

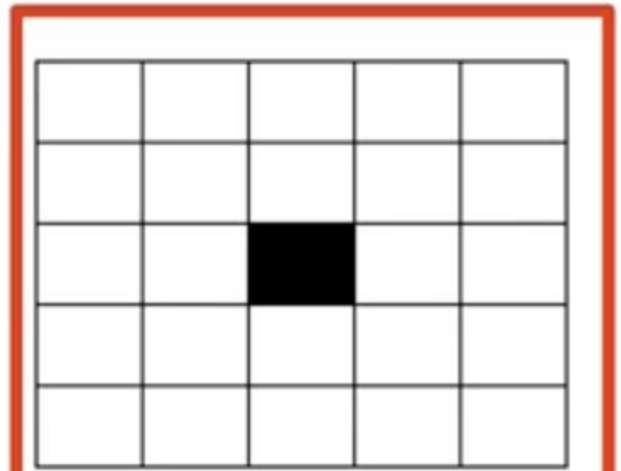




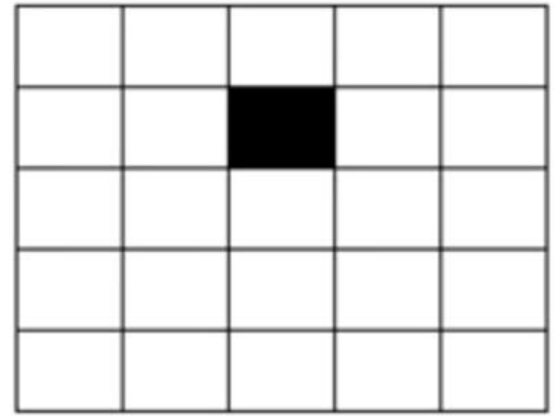
$t = 0$



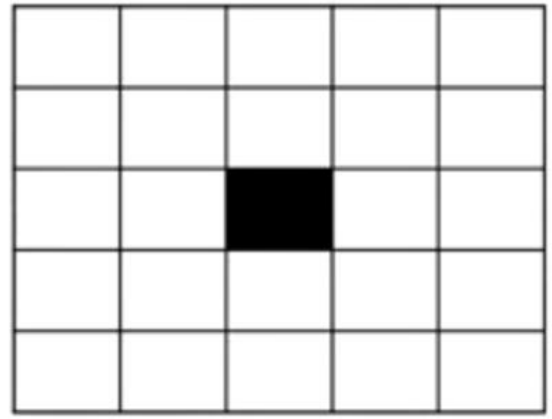
$t = 1$



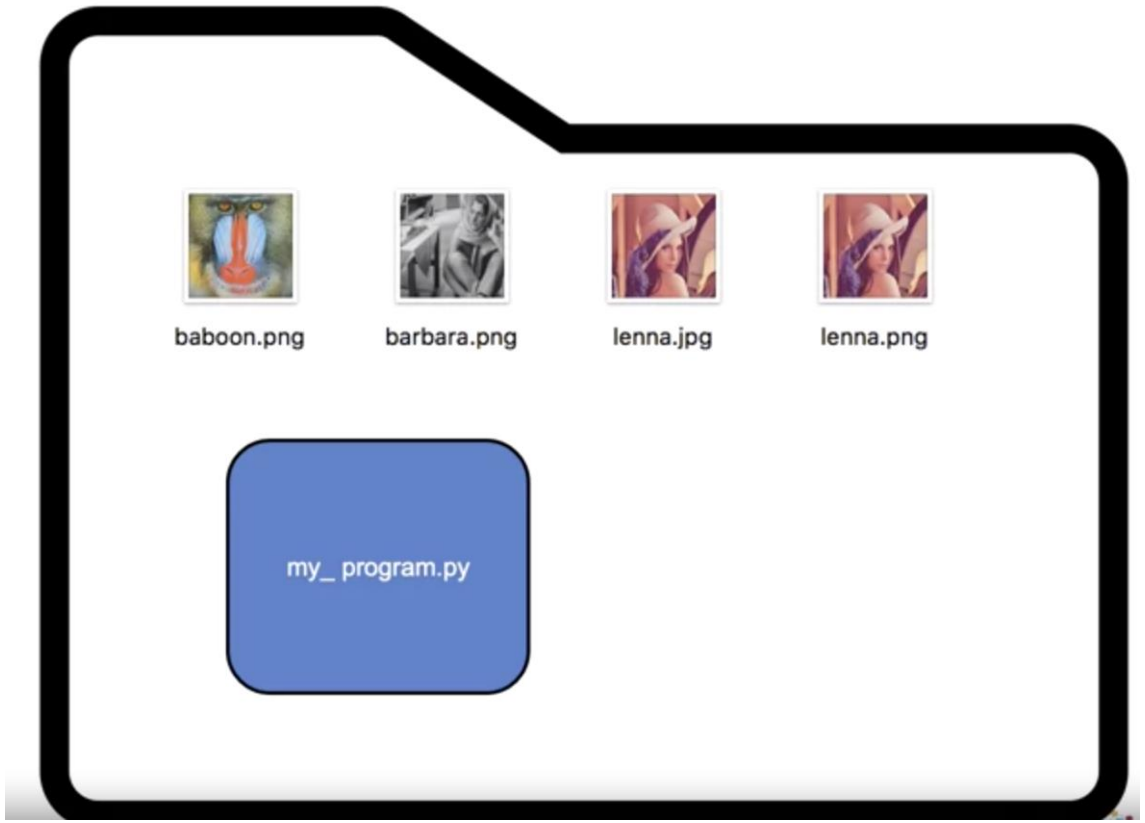
$t = 2$



$t = 3$



```
my_image="lenna.png"
```



image_path='/Users/…..labs/Module2/lenna.png'

my_ program.py

# Pillow (PIL Library)

```
from PIL import Image

image = Image.open(my_image)

image.show(title="Lena")
```

```
PIL.PngImagePlugin.PngImageFile
```

```
from PIL import Image

image = Image.open(my_image)

import matplotlib.pyplot as plt
plt.imshow(image)
```

PIL.PngImagePlugin.PngImageFile

image.format:PNG

image.size:(512, 512)

image.mode:RGB

```
from PIL import ImageOps

image_gray = ImageOps.grayscale(image)

  image_gray.mode:L


 image_gray.save("lenna.jpg")
```

```
image_gray = Image.open("barbara.png")
```

image_gray.mode:L

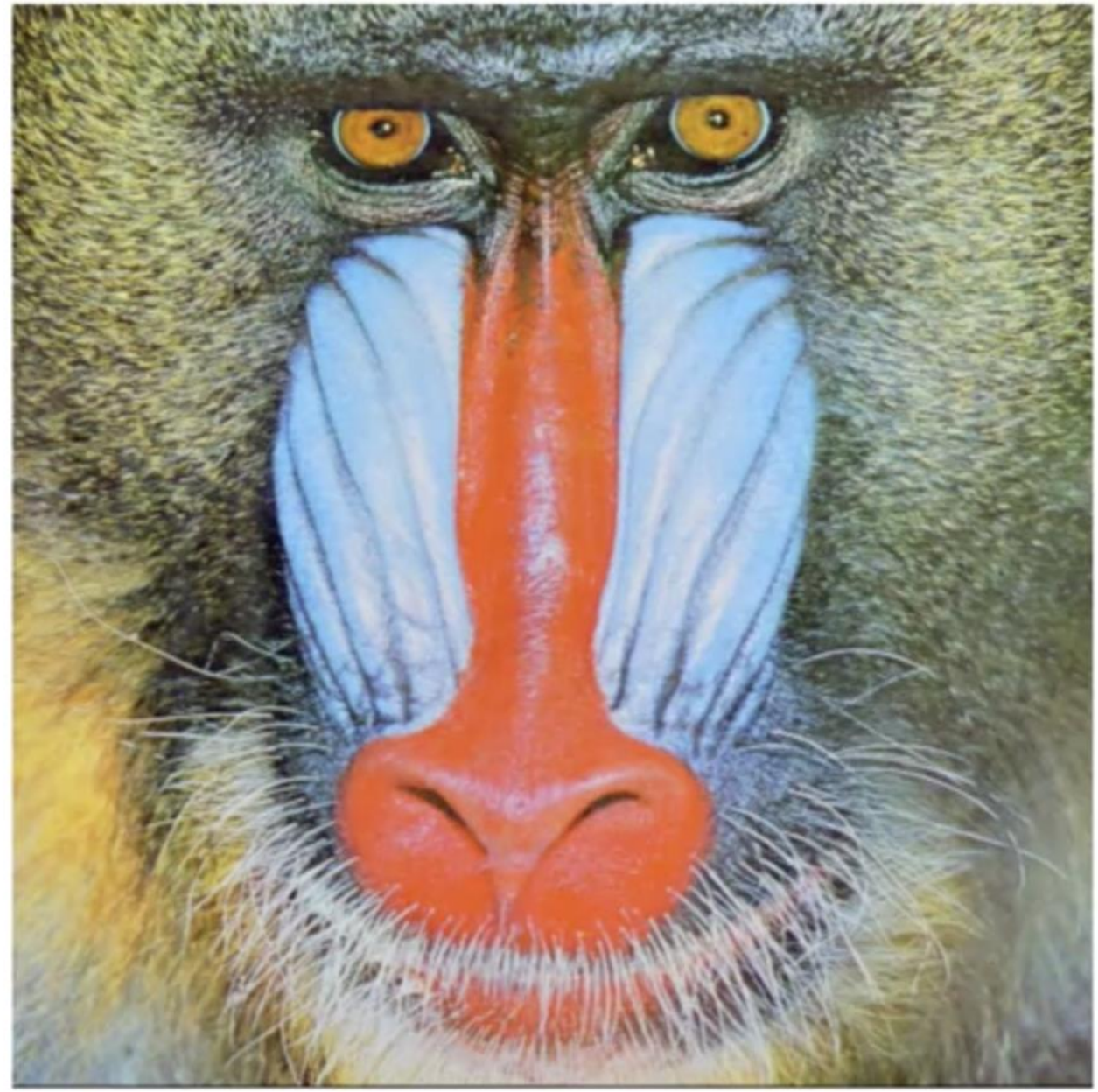
```
image_gray = Image.open("barbara.png")
```

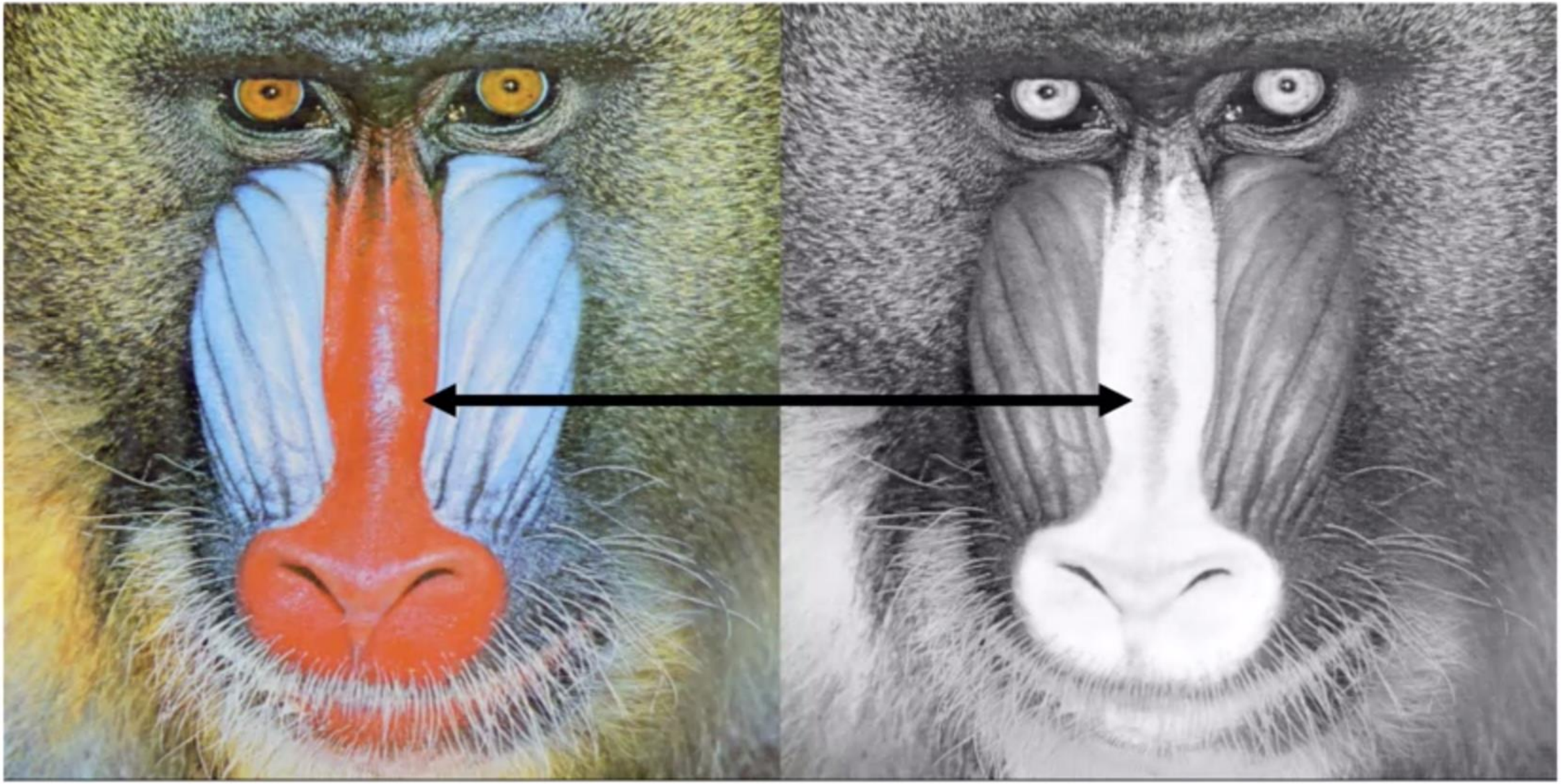
```
image_gray.mode:L
```

```
image_gray. quantize(2)
```

*Image.quantize(colors)*
*Colors <=256*

```
baboon= Image.open("baboon.png")

red, green, blue = baboon.split()
```





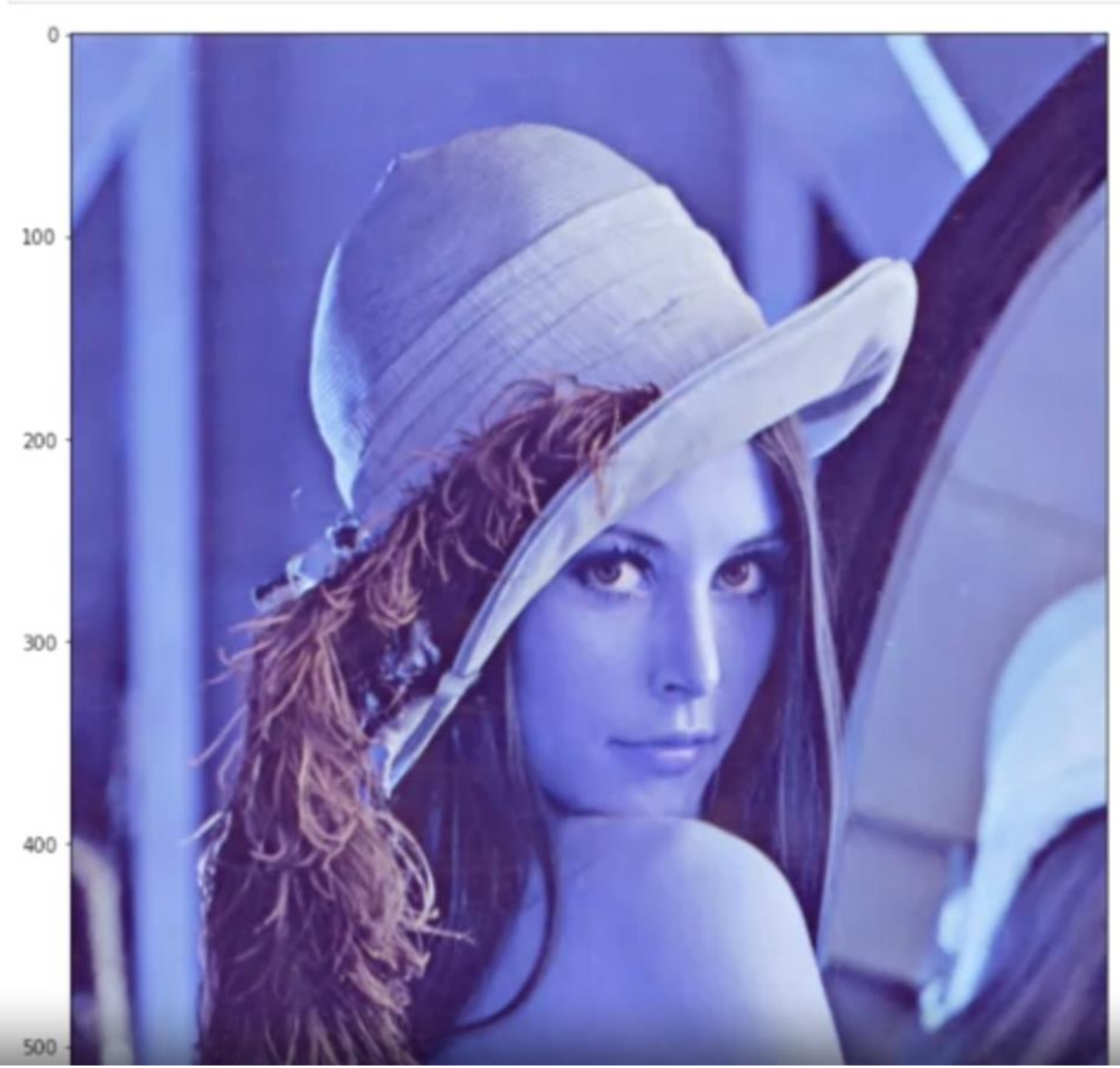
```
import numpy as np

array= np.array(image)

print(array)
```

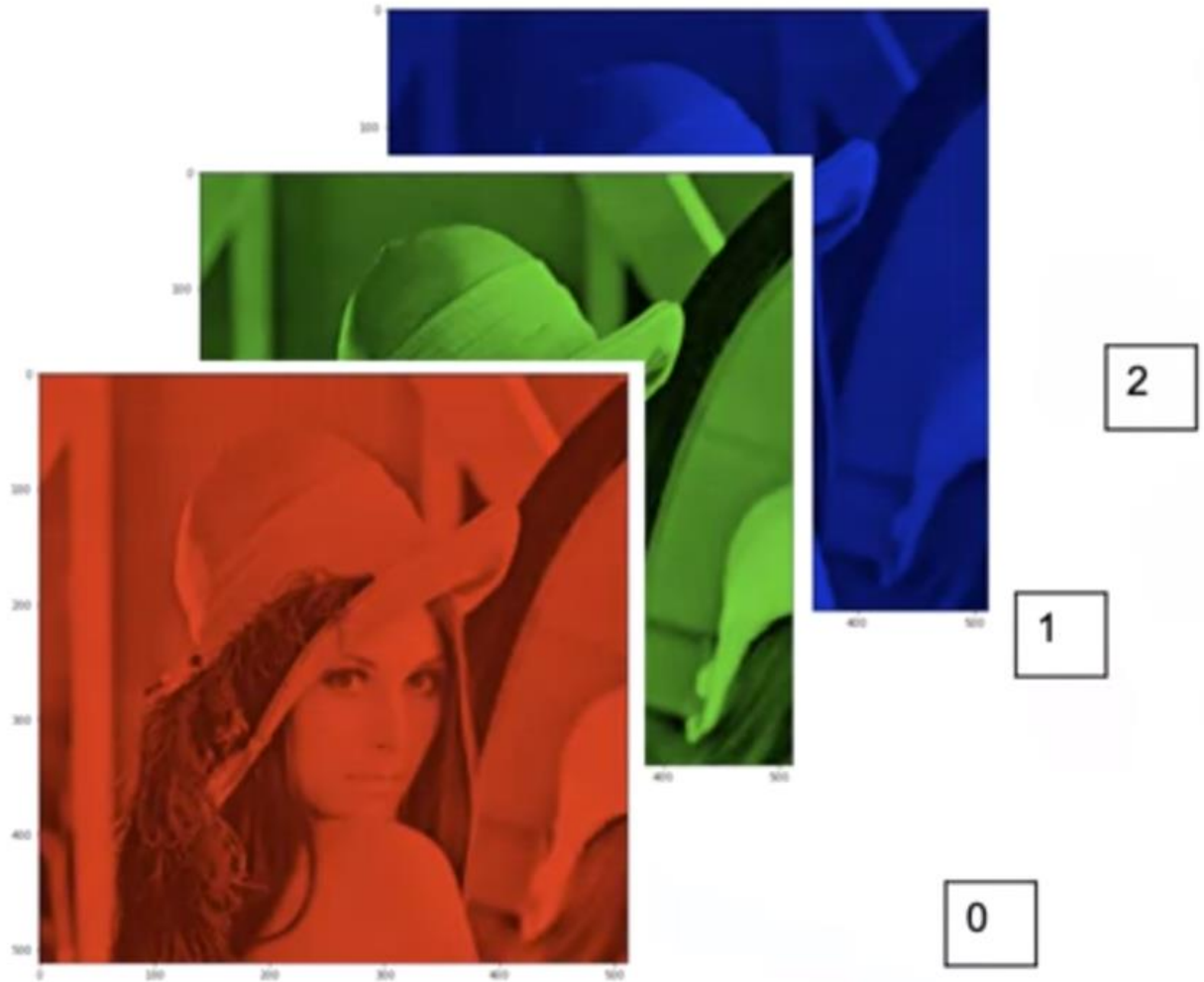
```
[[[226 137 125] [226
137 125] [223 137 133]
... [230 148 122] [221
130 110] [200 99 90]]
[[226 137 125] [226 137
125] [223 137 133] ...
[230 148 122] [221 130
110] [200 99 90]] [[226
137 125] [226 137 125]
[223 137 133] ... [230
148 122] [221 130 110]
[200 99 90]] ... [[ 84 18
60] [ 84 18 60] [ 92 27
58] ... [173 73 84] [172
68 76] [177 62 79]] [[ 82
22 57] [ 82 22 57] [ 96
32 62] ... [179 70 79]
[181 71 81] [185 74 81]]
[[ 82 22 57] [ 82 22 57] [
96 32 62] ... [179 70 79]
[181 71 81] [185 74
81]]]
```

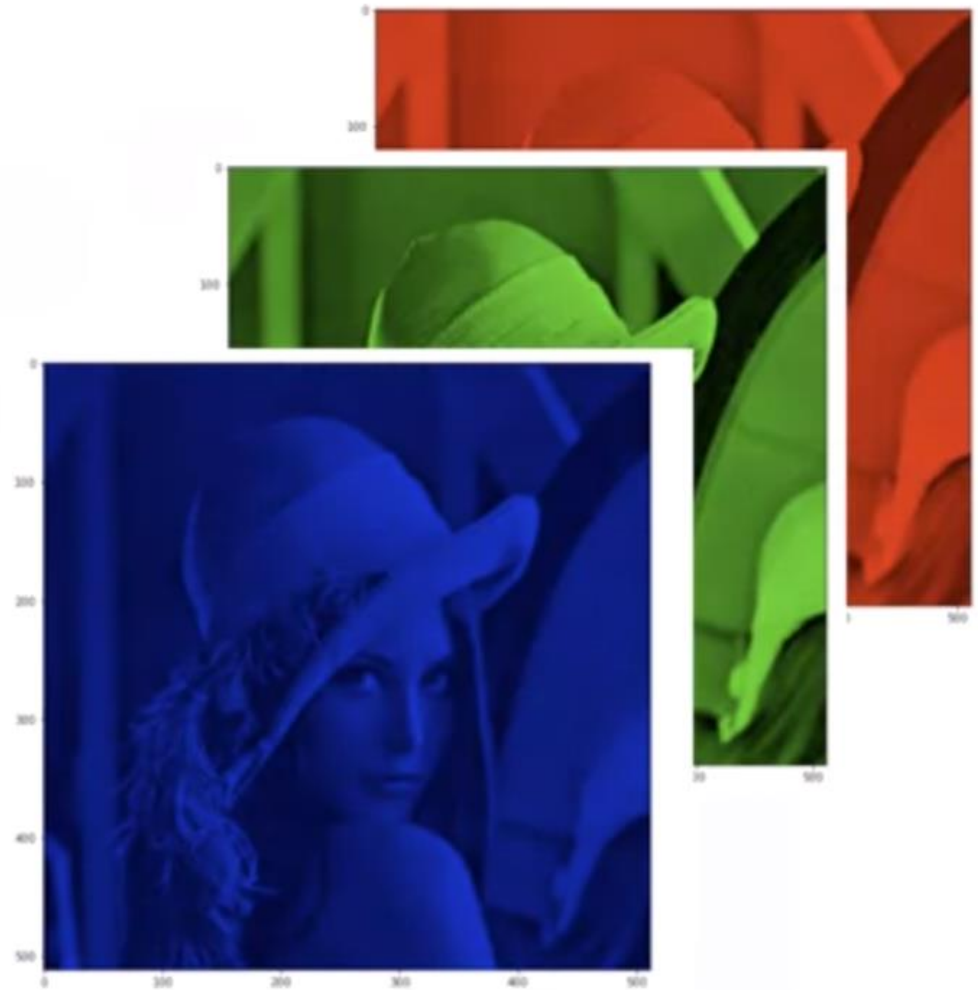
# OpenCV

```
import cv2

image = cv2.imread(my_image)

type(image):numpy.ndarray

image.shape:(512, 512, 3)

import matplotlib.pyplot as plt
plt.imshow(image)
```

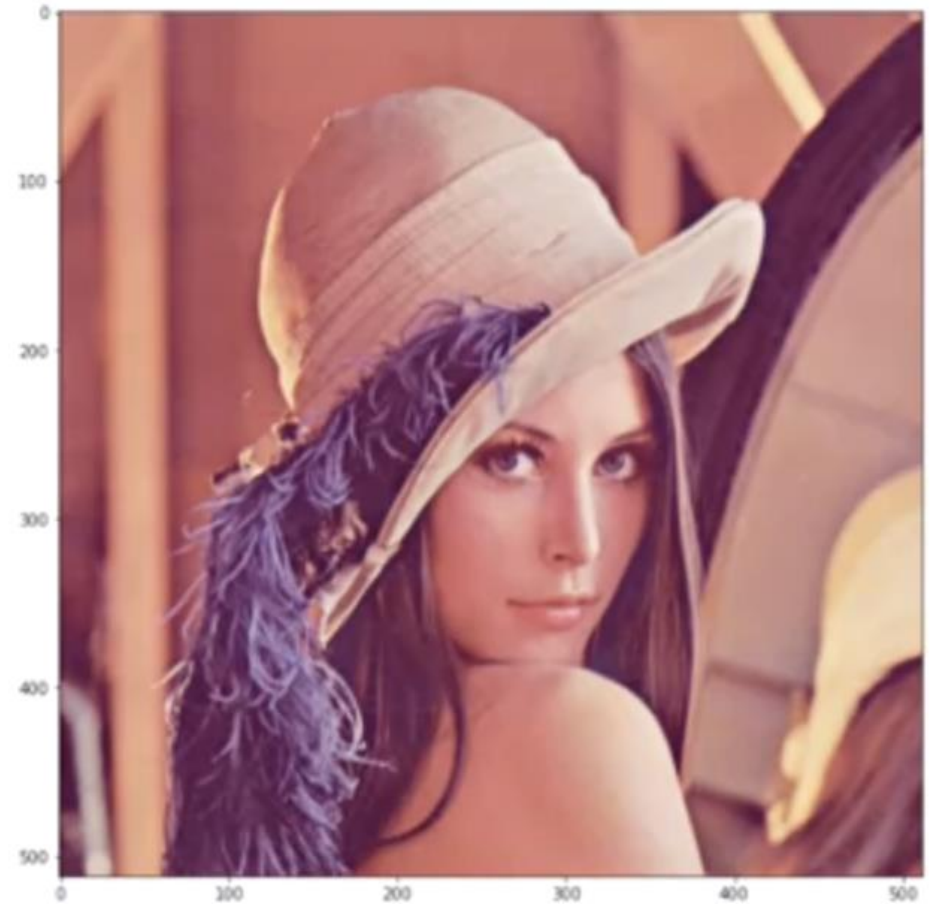
PIL that is RGB. OpenCV is BGR




2
1
0



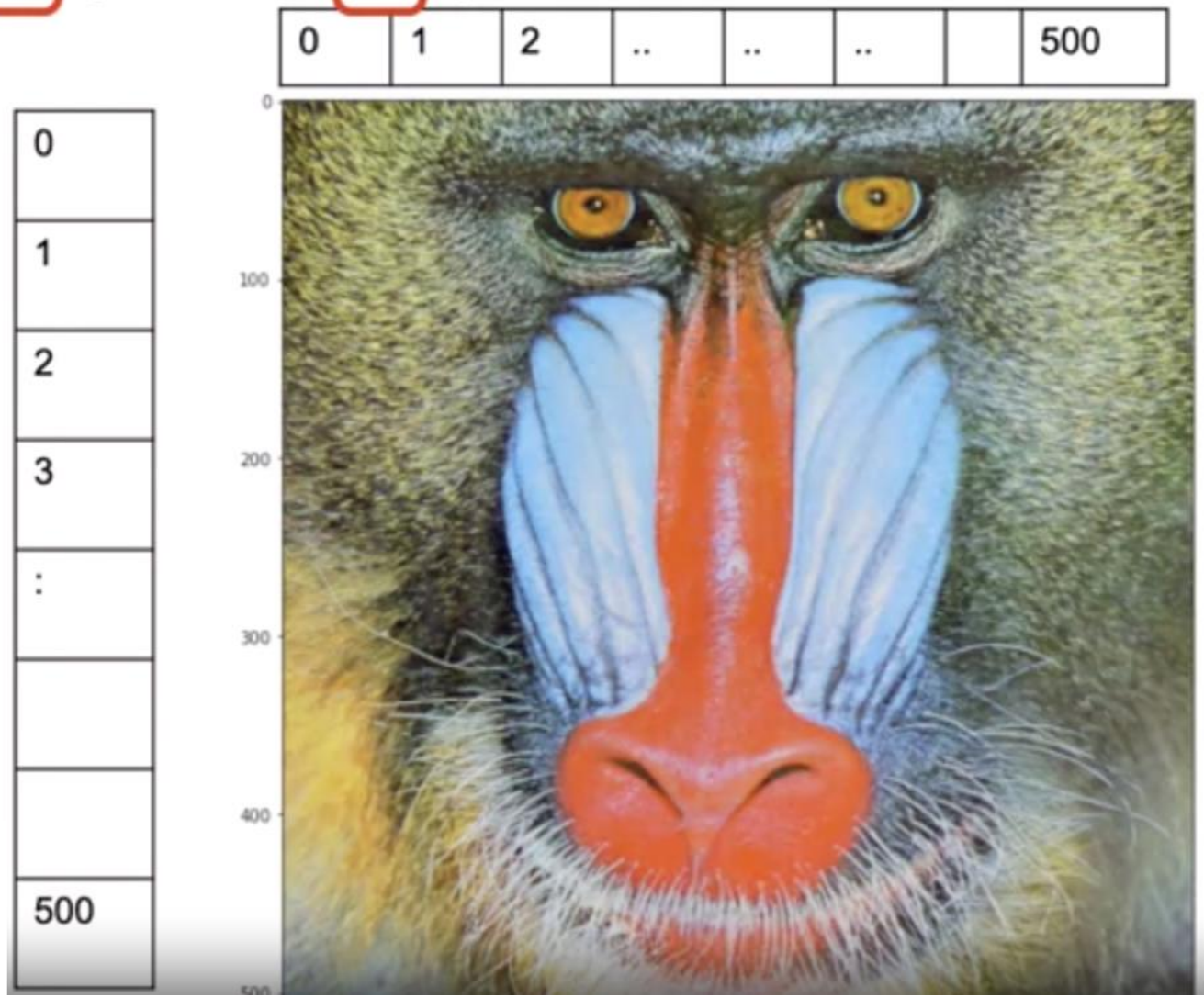
```
new_image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

plt.imshow(new_image )
```



```
image_gray = cv2. cvtColor(image, cv2.COLOR_BGR2GRAY)

plt.imshow(image_gray )

 cv2.imwrite('lena_gray_cv.jpg', image_gray)
```

```
im_gray = cv2.imread('barbara.png', cv2.IMREAD_GRAYSCALE)
```

```
baboon=cv2.imread('baboon.png')

blue, green,red = baboon[:,:,0],baboon[:,:,1],baboon[:,:,2]
```

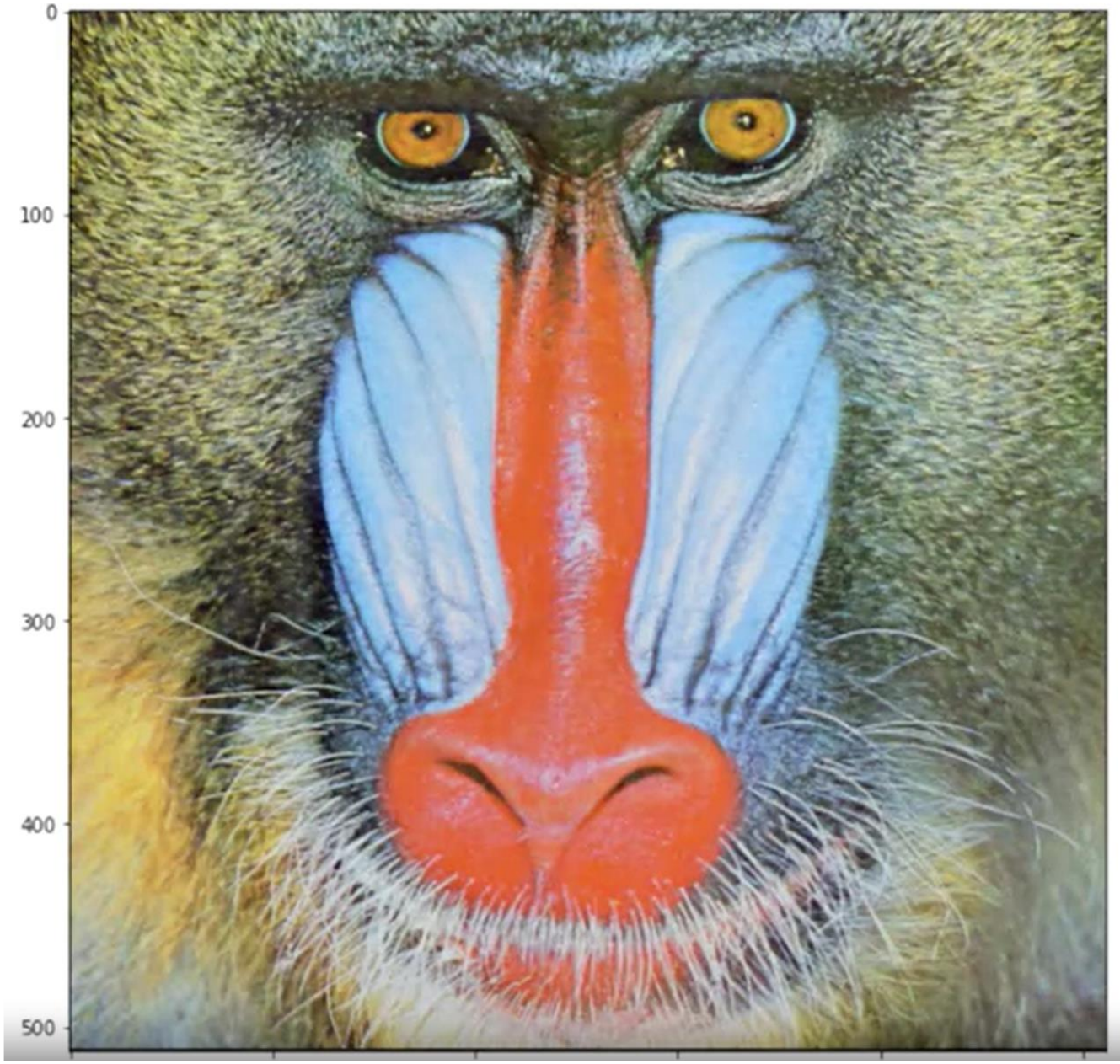


# Manipulating Images

Thao tác trên ảnh số

# Copying an image

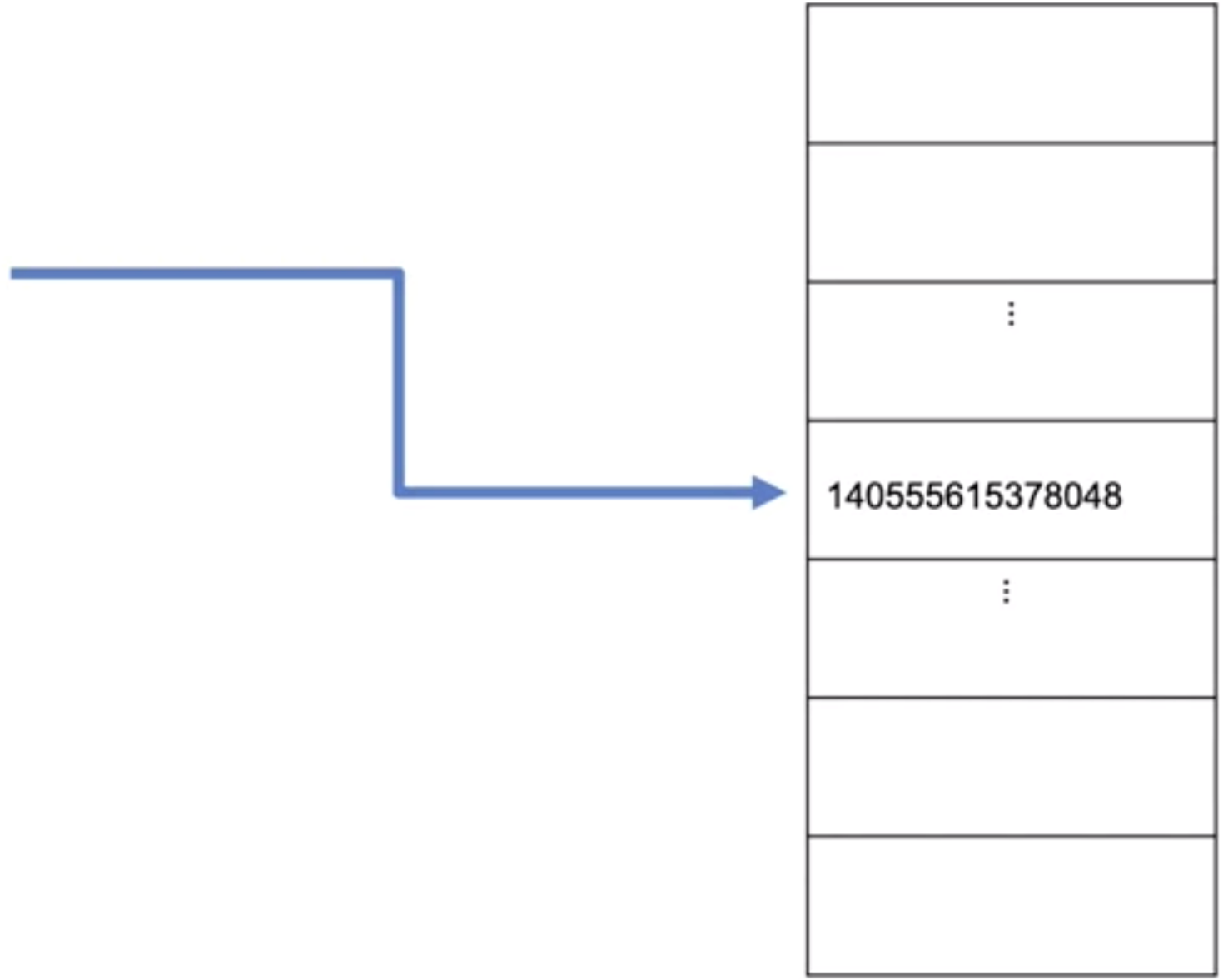
```
from PIL import Image
import cv2
import numpy as np

baboon = cv2.imread("baboon.png")

baboon = np.array(Image.open( “baboon.png”))
```



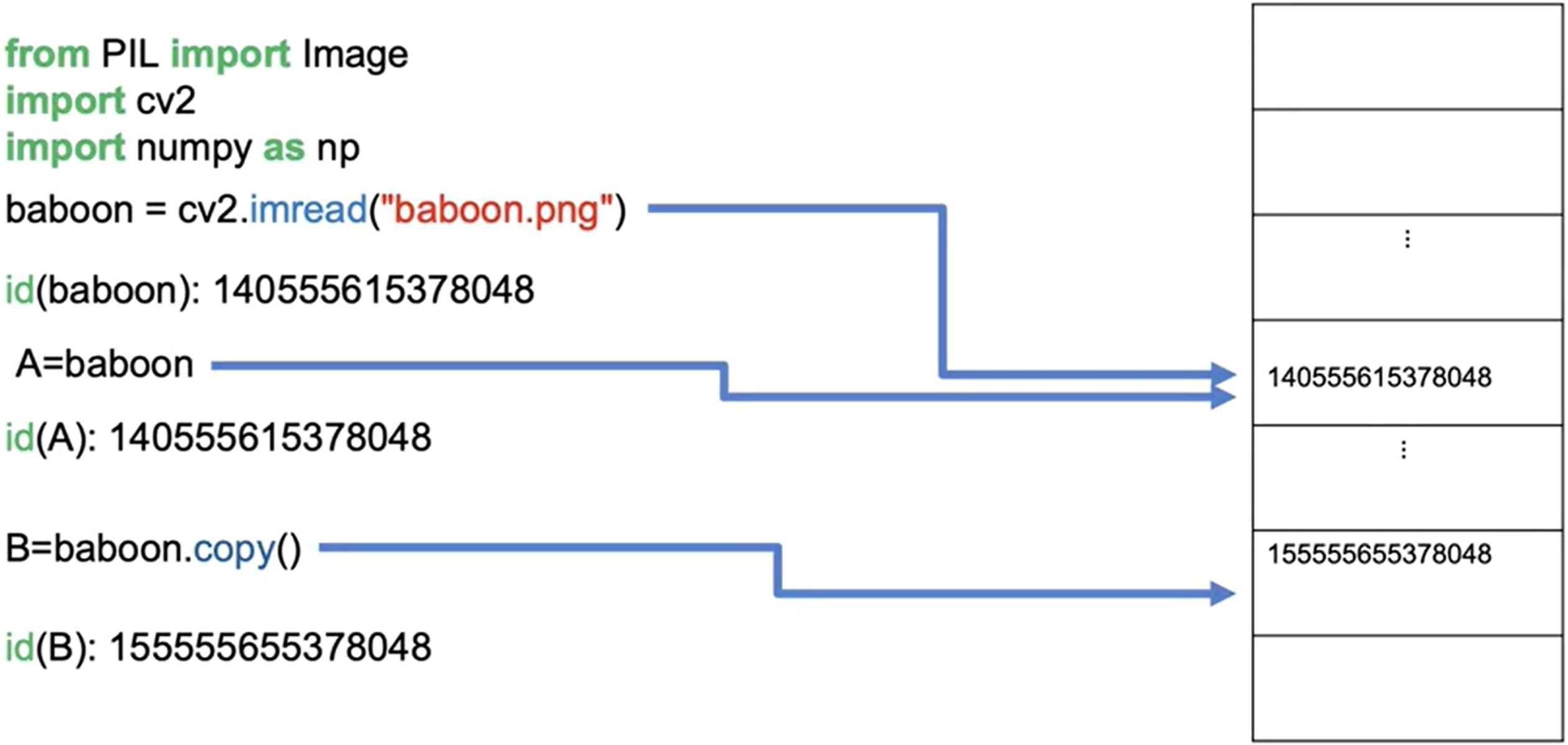
```
from PIL import Image
import cv2
import numpy as np
baboon = cv2.imread("baboon.png")
```

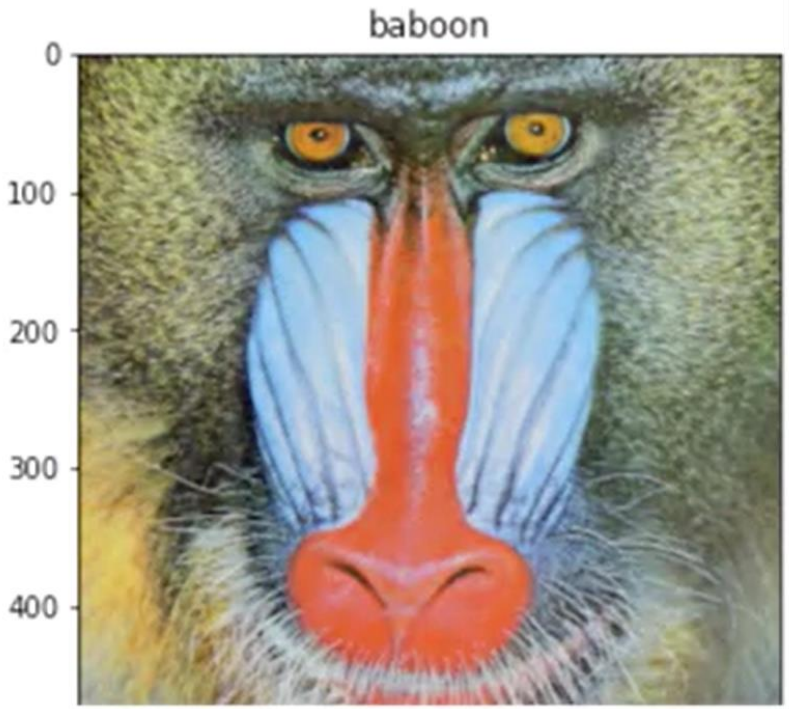
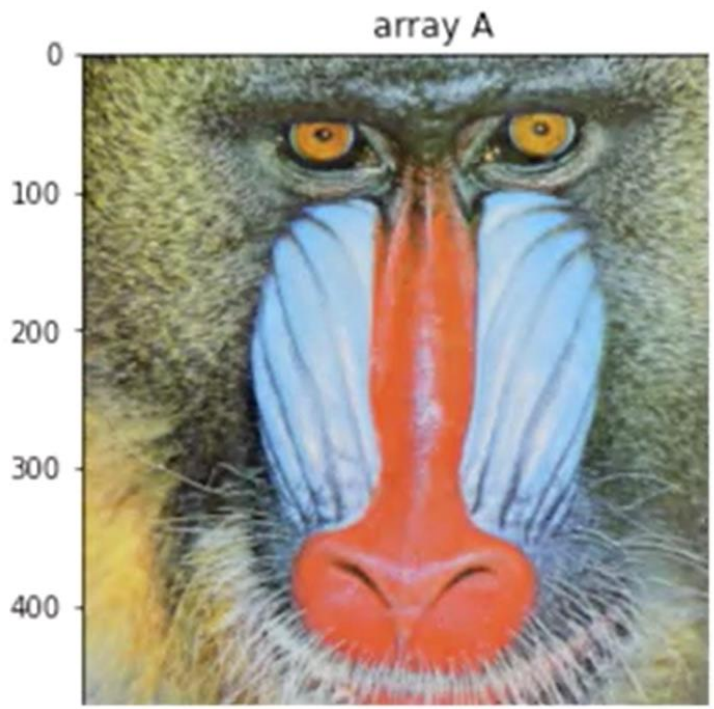
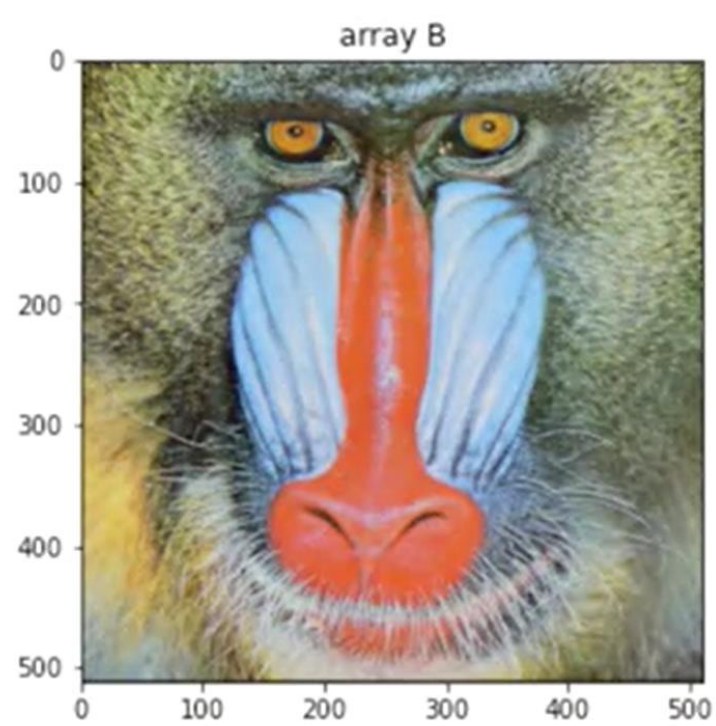
id(baboon): 140555615378048

A=baboon

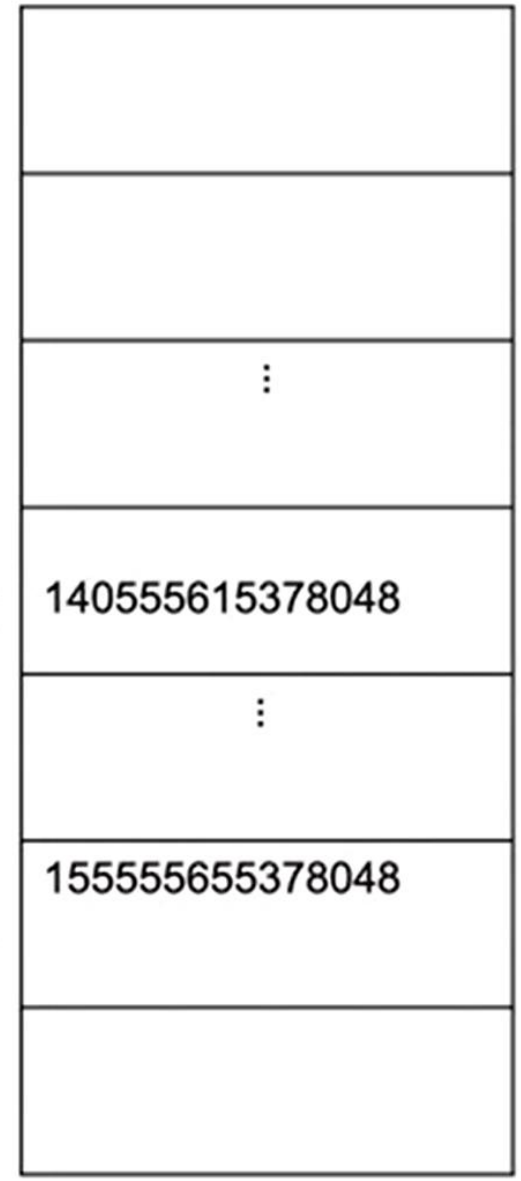


```
from PIL import Image
import cv2
import numpy as np
baboon = cv2.imread("baboon.png")

id(baboon): 140555615378048

 A=baboon

id(A): 140555615378048


B=baboon.copy()

id(B): 155555655378048
```

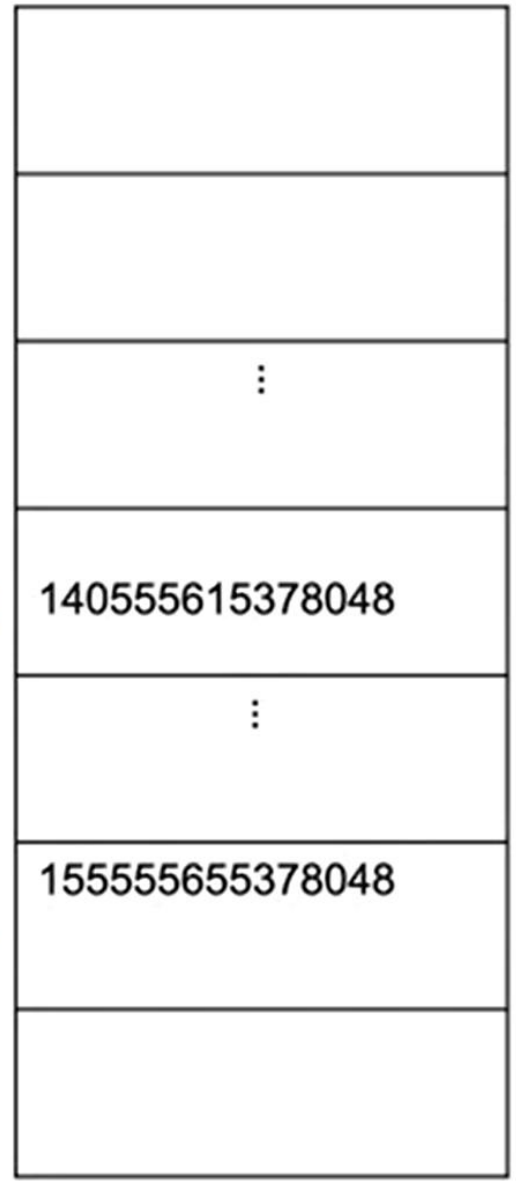


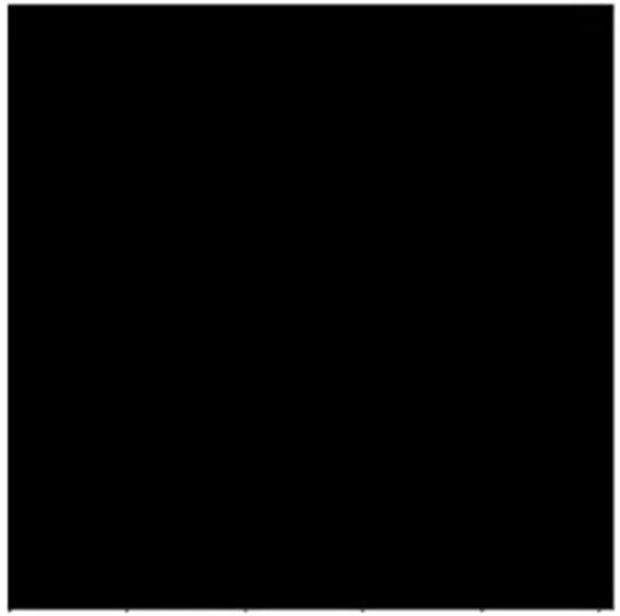
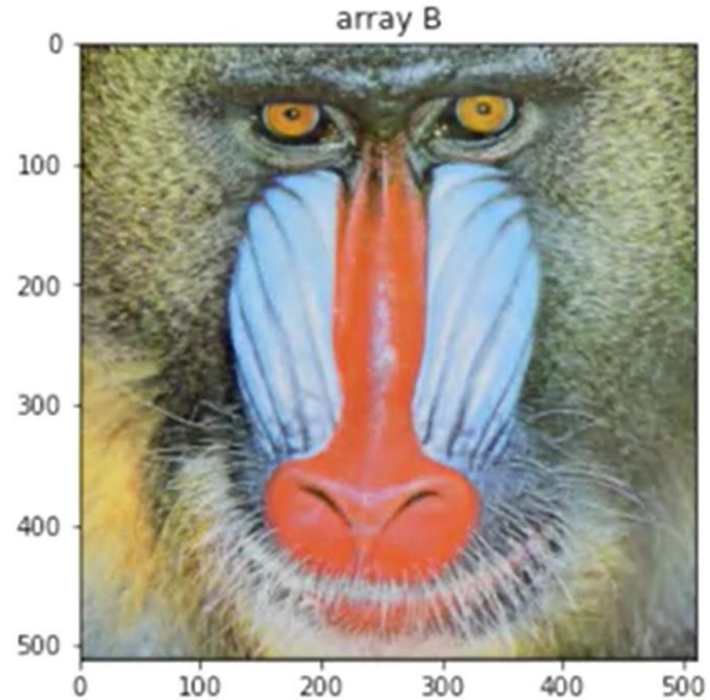
| baboon | A | B |
| --- | --- | --- |
|  |  |  |
| id:140555615378048 | id:140555615378048 | id:155555655378048 |

```
from PIL import Image
import cv2
import numpy as np
baboon = cv2.imread("baboon.png")

id(baboon): 140555615378048

 A=baboon

id(A): 140555615378048


B=baboon.copy()

id(B): 155555655378048


 baboon[:,:,]=0
```



```
from PIL import Image
import cv2
import numpy as np
baboon = cv2.imread("baboon.png")

id(baboon): 140555615378048

 A=baboon

id(A): 140555615378048


B=baboon.copy()

id(B): 155555655378048


 baboon[:,:,]=0
```



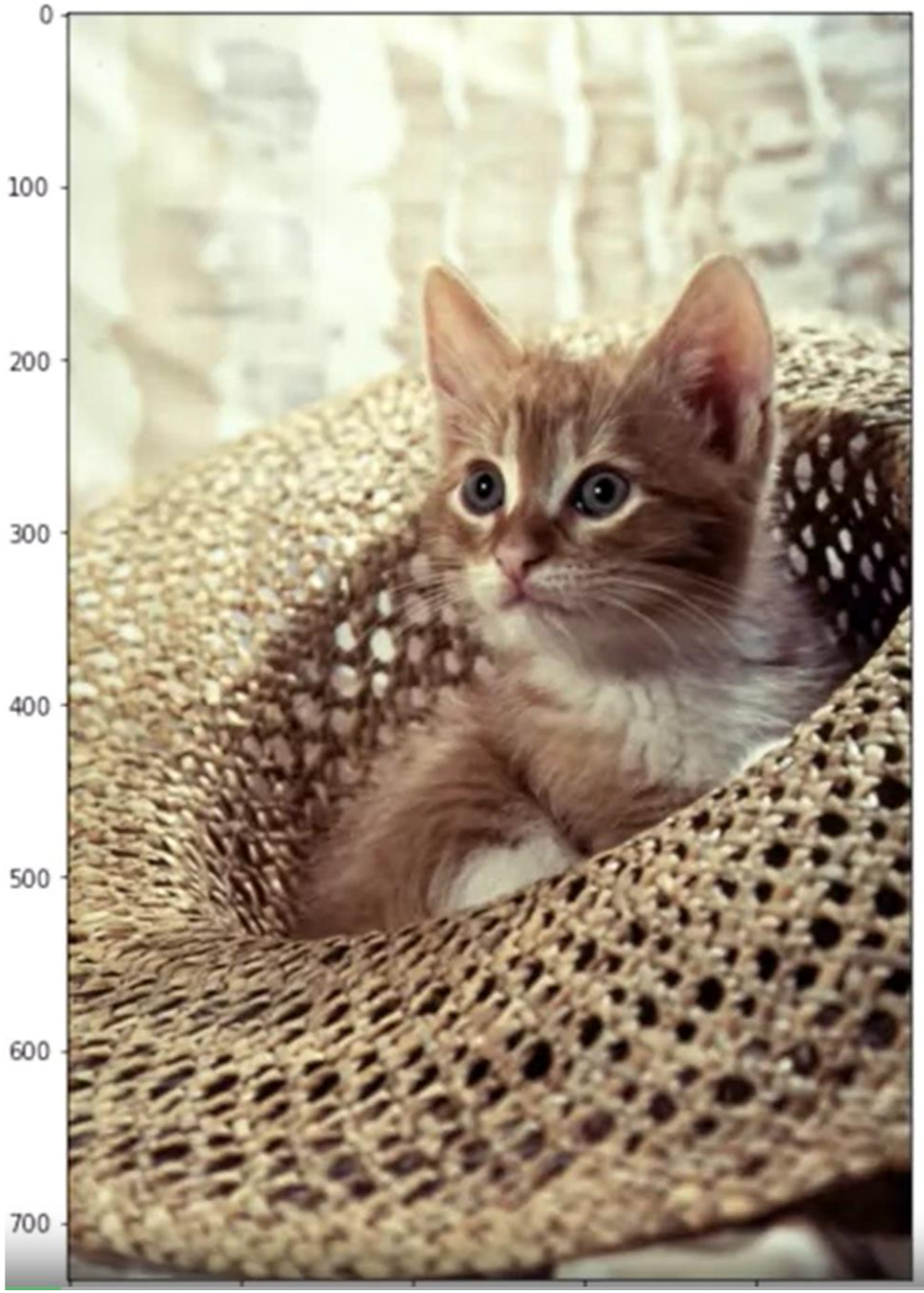
| baboon | A | B |
| --- | --- | --- |
|  |  |   |
| id:140555615378048 | id:140555615378048 | id:155555655378048 |

# Flipping images

| | | | | | |
|---|---|---|---|---|---|
| I[0,0] | I[0,1] | I[0,2] | I[0,3] | I[0,4] | I[0,5] |
| I[1,0] | I[1,2] | I[1,2] | I[1,3] | I[1,4] | I[1,5] |
| I[2,0] | I[2,2] | I[2,2] | I[2,3] | I[2,4] | I[2,5] |
| I[3,0] | I[3,1] | I[3,2] | I[3,3] | I[3,4] | I[3,5] |
| I[4,0] | I[4,1] | I[4,2] | I[4,3] | I[4,4] | I[4,5] |
| I[5,0] | I[5,1] | I[5,2] | I[5,3] | I[5,4] | I[5,5] |

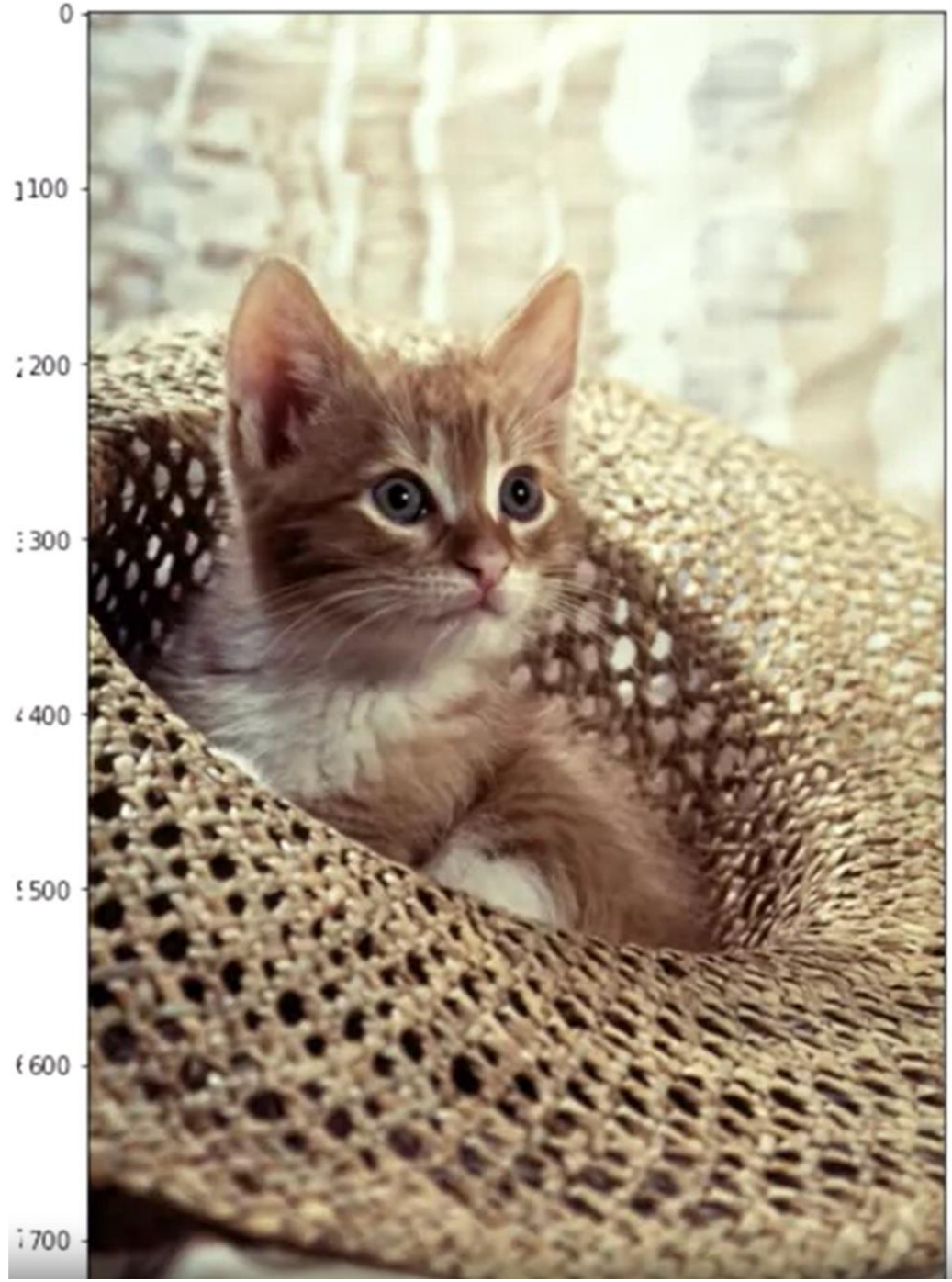
| | | | | | |
|---|---|---|---|---|---|
| I[0,0] | I[1,0] | I[2,0] | I[3,0] | I[4,0] | I[5,0] |
| I[0,1] | I[1,1] | I[2,1] | I[3,1] | I[4,1] | I[5,1] |
| I[0,2] | I[1,2] | I[2,2] | I[3,2] | I[4,2] | I[5,2] |
| I[0,3] | I[1,3] | I[2,3] | I[3,3] | I[4,3] | I[5,3] |
| I[0,4] | I[1,4] | I[2,4] | I[3,4] | I[4,4] | I[5,4] |
| I[0,5] | I[1,5] | I[2,5] | I[3,5] | I[4,5] | I[5,5] |

```
from PIL import ImageOps
```

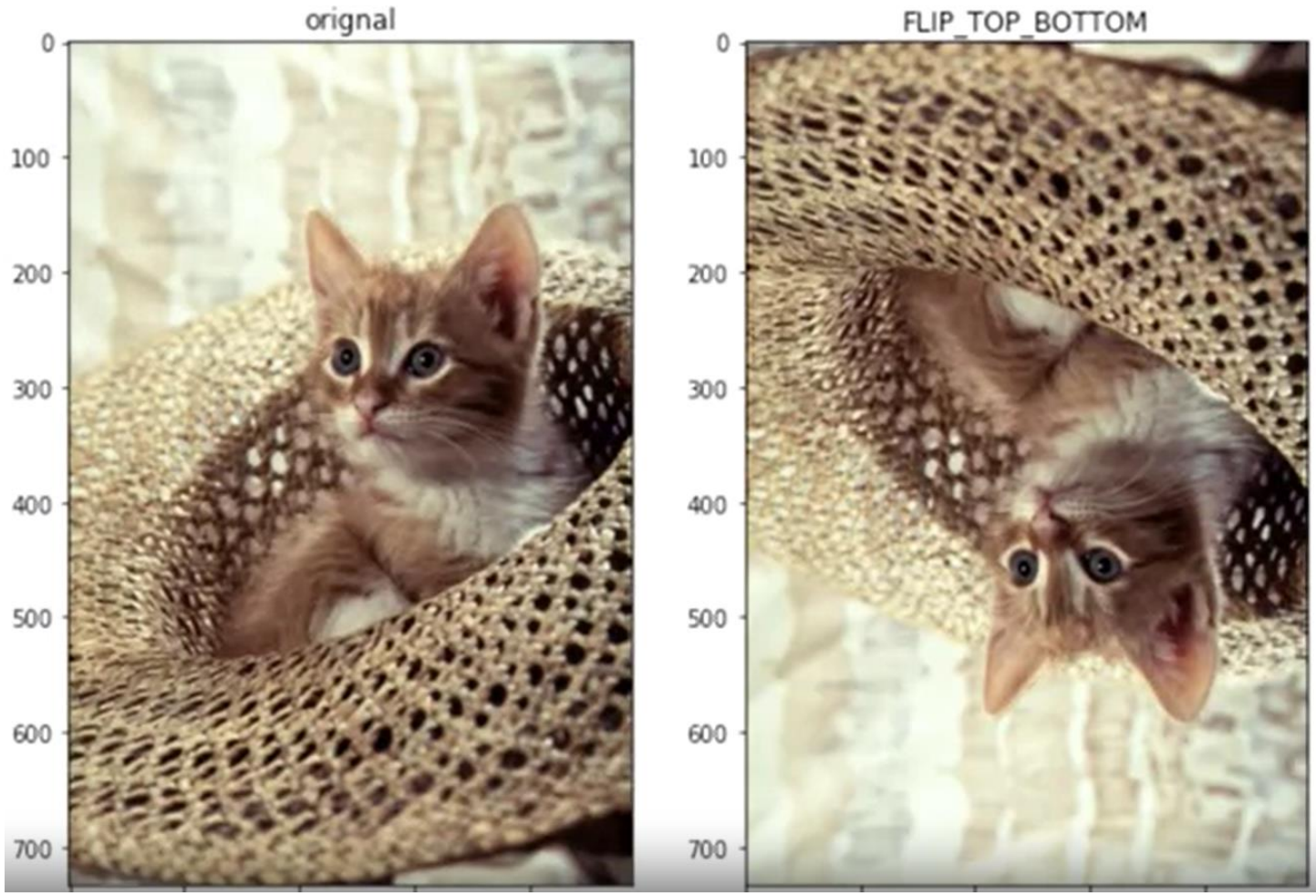


```
from PIL import  ImageOps

im_flip = ImageOps.flip(image)
```

```
from PIL import  ImageOps

im_flip = ImageOps.flip(image)

im_mirror = ImageOps.mirror(image)
```



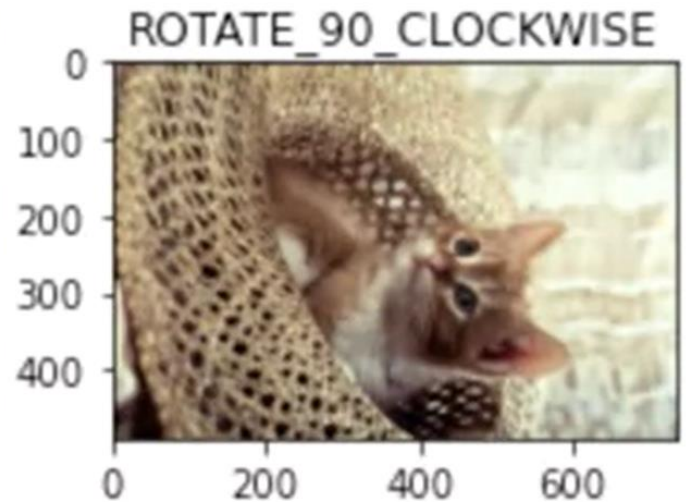
```
image.transpose(Image.FLIP_TOP_BOTTOM)
```



# Flipping images using OpenCV

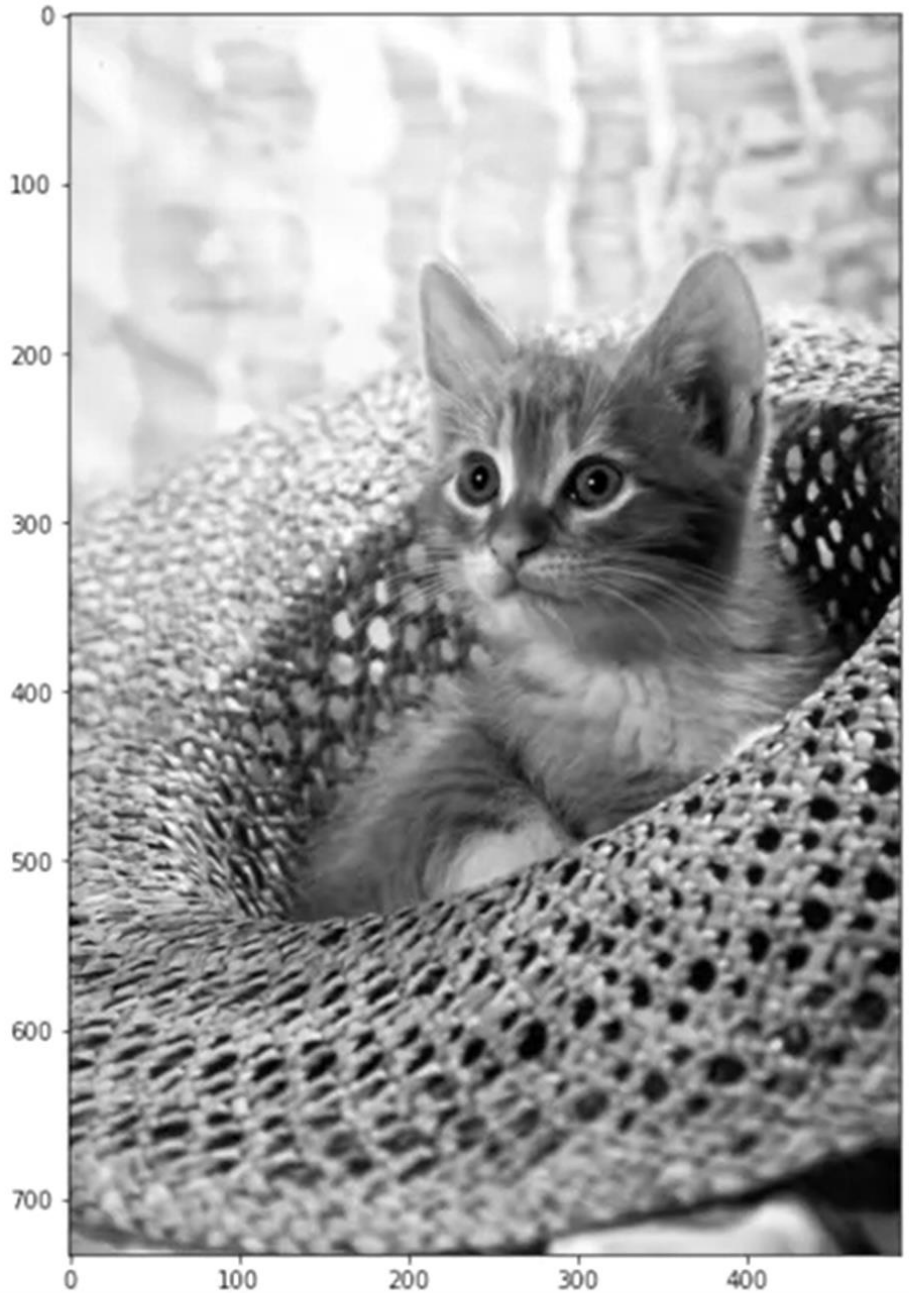
```
import cv2

im_flip =  cv2.flip(image,0 )
```

```
im_flip=cv2.rotate(image, cv2.ROTATE_90_CLOCKWISE)
```

# Cropping

| |
|---|
| 0 |
| 1 |
| 2 |
| 3 |
| : |
| |
| |
| 700 |




| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | .. | .. | .. | 499 | 500 |

| | | | | | |
|---|---|---|---|---|---|
| 225 | 134 | .. | .... | 142 | 14 |
| 254 | 232 | | | | 232 |
| 122 | 213 | | | : | : |
| : | : | : | : | : | : |
| : | : | : | : | : | : |
| : | : | : | : | : | |
| 24 | 23 | 44 | 24 | 32 | 12 |
| 123 | 232 | .. | … | 34 | 53 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 225 | 134 | 123 | 12 | 142 | 14 |
| 1 | 254 | 232 | 222 | 232 | 212 | 232 |
| 2 | 122 | 213 | 123 | 12 | 142 | 132 |
| 3 | 254 | 232 | 222 | 232 | 212 | 122 |
| 4 | 122 | 213 | 44 | 24 | 32 | 123 |
| 5 | 123 | 33 | 122 | 122 | 34 | 123 |

| 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | I[0,0] | I[0,1] | I[0,2] | I[0,3] | I[0,4] | I[0,5] |
| 1 | I[1,0] | I[1,2] | I[1,2] | I[1,3] | I[1,4] | I[1,5] |
| 2 | I[2,0] | I[2,2] | I[2,2] | I[2,3] | I[2,4] | I[2,5] |
| 3 | I[3,0] | I[3,1] | I[3,2] | I[3,3] | I[3,4] | I[3,5] |
| 4 | I[4,0] | I[4,1] | I[4,2] | I[4,3] | I[4,4] | I[4,5] |
| 5 | I[5,0] | I[5,1] | I[5,2] | I[5,3] | I[5,4] | I[5,5] |

| | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 |

I =

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | I[0,0] | I[0,1] | I[0,2] | I[0,3] | I[0,4] | I[0,5] |
| 1 | I[1,0] | I[1,2] | I[1,2] | I[1,3] | I[1,4] | I[1,5] |
| 2 | I[2,0] | I[2,2] | I[2,2] | I[2,3] | I[2,4] | I[2,5] |
| 3 | I[3,0] | I[3,1] | I[3,2] | I[3,3] | I[3,4] | I[3,5] |
| 4 | I[4,0] | I[4,1] | I[4,2] | I[4,3] | I[4,4] | I[4,5] |
| 5 | I[5,0] | I[5,1] | I[5,2] | I[5,3] | I[5,4] | I[5,5] |

| 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|

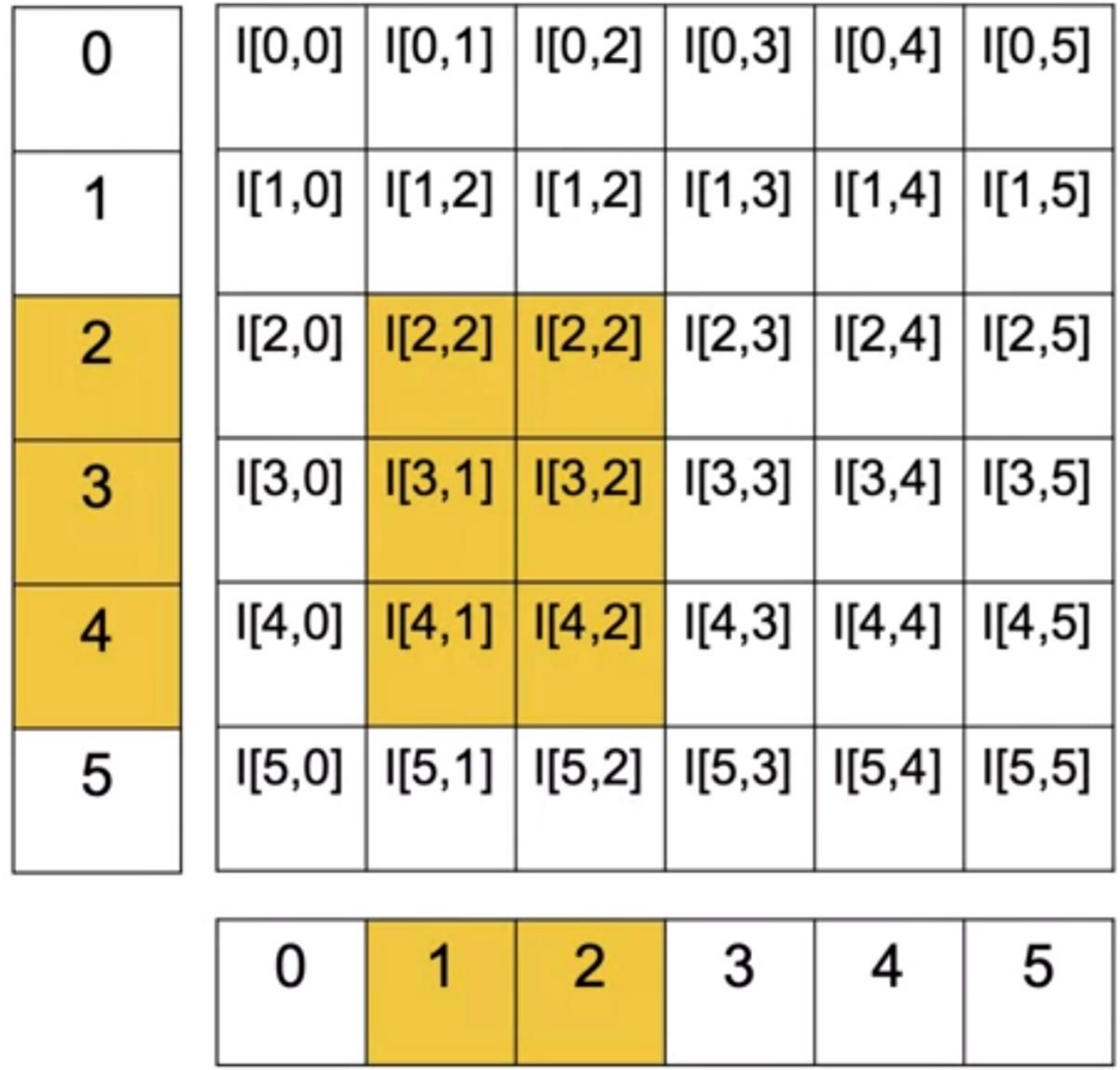
I[2:5,:]

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | I[0,0] | I[0,1] | I[0,2] | I[0,3] | I[0,4] | I[0,5] |
| 1 | I[1,0] | I[1,2] | I[1,2] | I[1,3] | I[1,4] | I[1,5] |
| 2 | I[2,0] | I[2,2] | I[2,2] | I[2,3] | I[2,4] | I[2,5] |
| 3 | I[3,0] | I[3,1] | I[3,2] | I[3,3] | I[3,4] | I[3,5] |
| 4 | I[4,0] | I[4,1] | I[4,2] | I[4,3] | I[4,4] | I[4,5] |
| 5 | I[5,0] | I[5,1] | I[5,2] | I[5,3] | I[5,4] | I[5,5] |

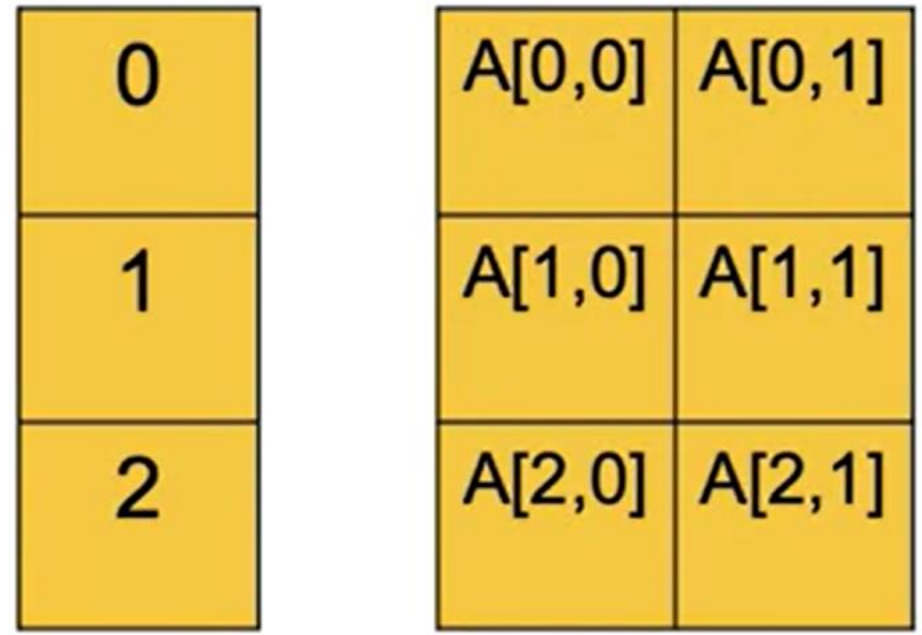
| 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|

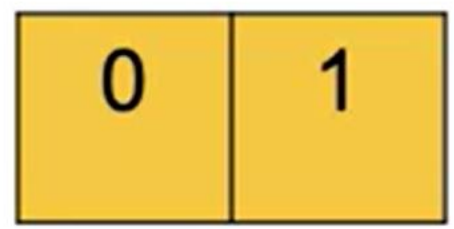
I[2:5,1:2]

A=I[2:5,1:2]

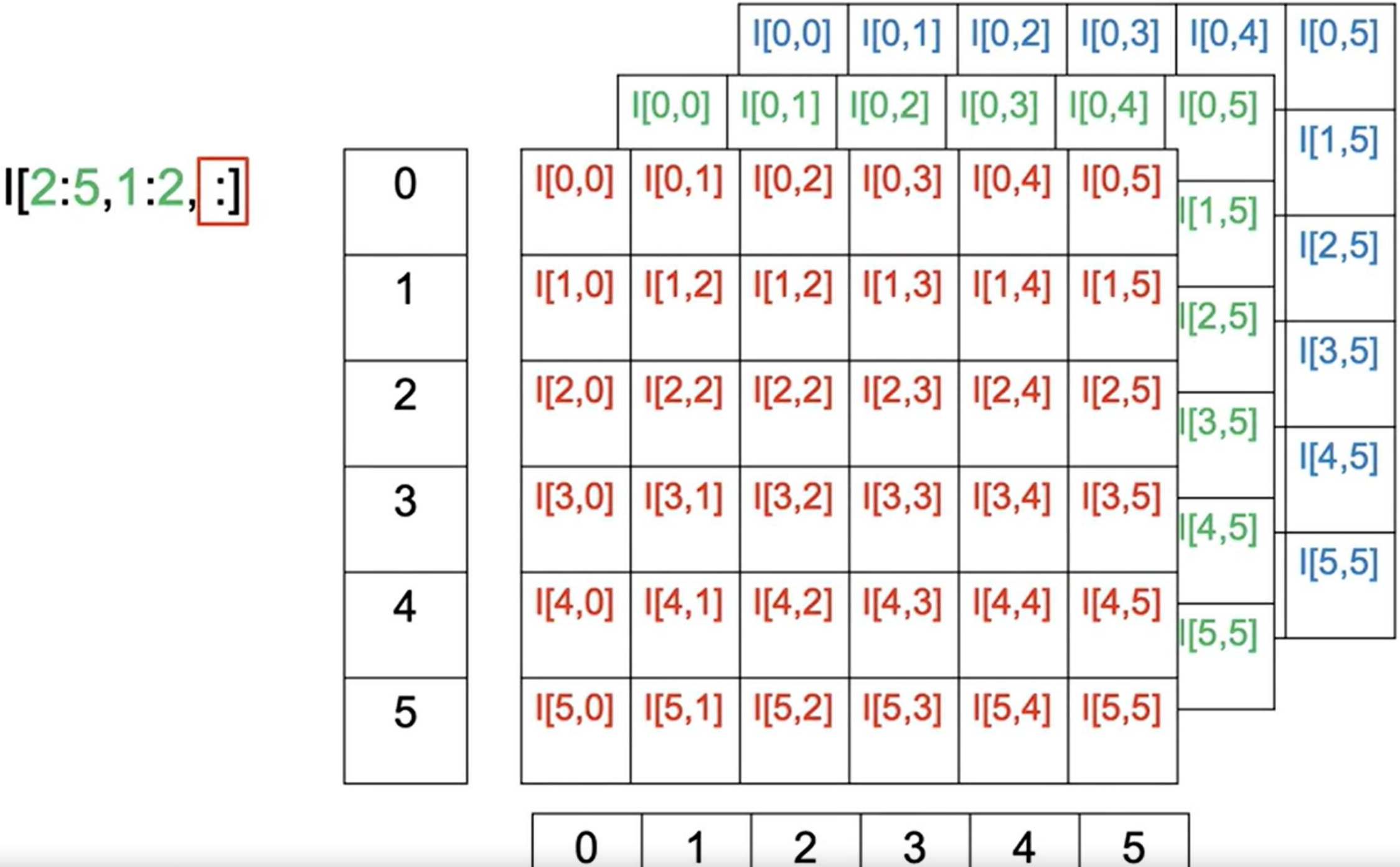


A=I[2:5,1:2]



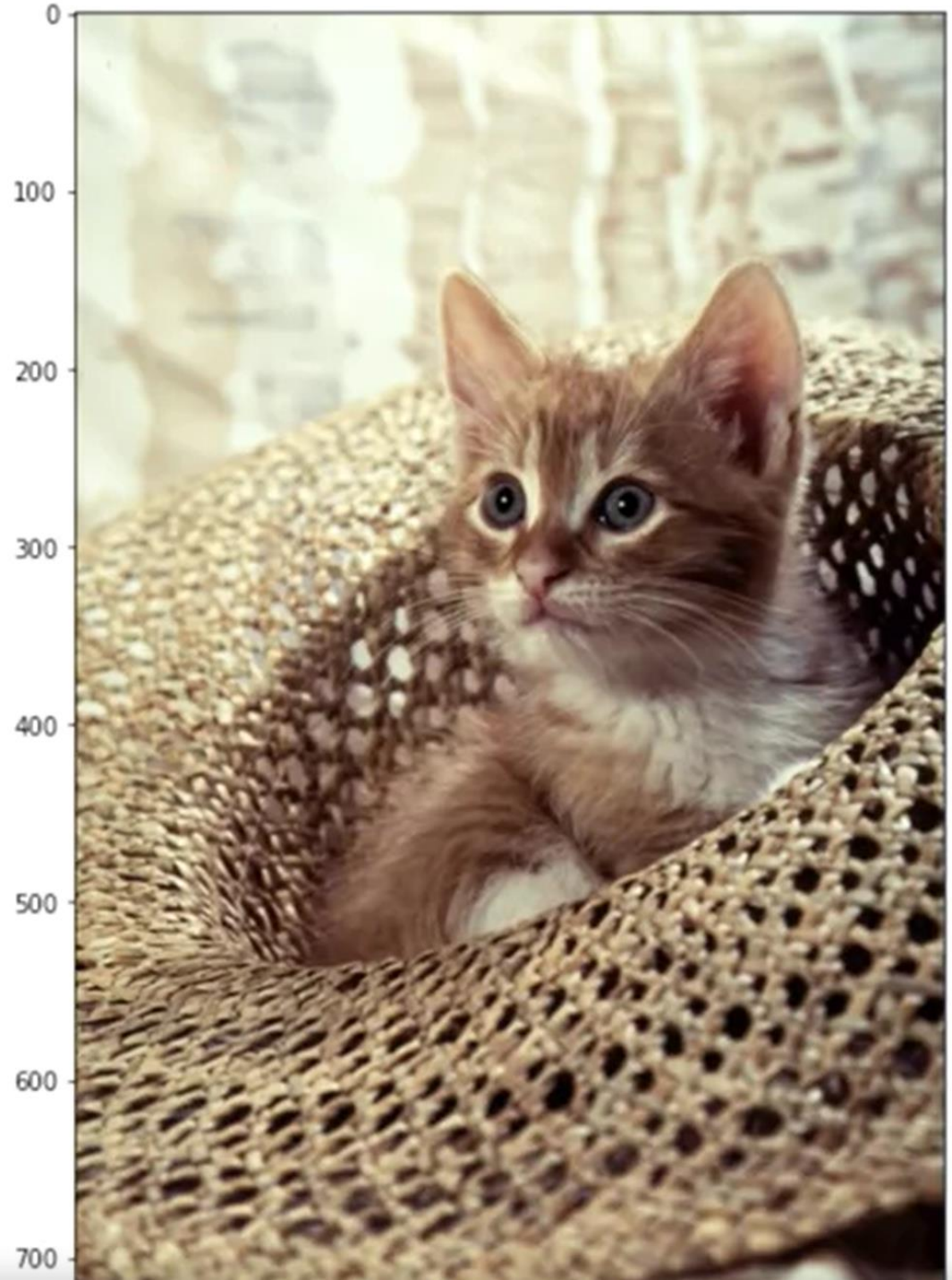



I[2:5,1:2,:]


I[0,0] I[0,1] I[0,2] I[0,3] I[0,4] I[0,5]
I[0,0] I[0,1] I[0,2] I[0,3] I[0,4] I[0,5]
I[1,5]
I[2,5]
I[3,5]
I[4,5]
I[5,5]
0
1
2
3
4
5
I[0,0] I[0,1] I[0,2] I[0,3] I[0,4] I[0,5]
I[1,0] I[1,2] I[1,2] I[1,3] I[1,4] I[1,5]
I[2,0] I[2,2] I[2,2] I[2,3] I[2,4] I[2,5]
I[3,0] I[3,1] I[3,2] I[3,3] I[3,4] I[3,5]
I[4,0] I[4,1] I[4,2] I[4,3] I[4,4] I[4,5]
I[5,0] I[5,1] I[5,2] I[5,3] I[5,4] I[5,5]
0 1 2 3 4 5

```
from PIL import Image
import numpy as np

image = Image.open("cat.png")
image =np.array(image)
```

```
import cv2

image =cv2.imread("cat.png")
```



```
upper= 150
lower=400
crop_top=image[upper: lower,:,:]
```

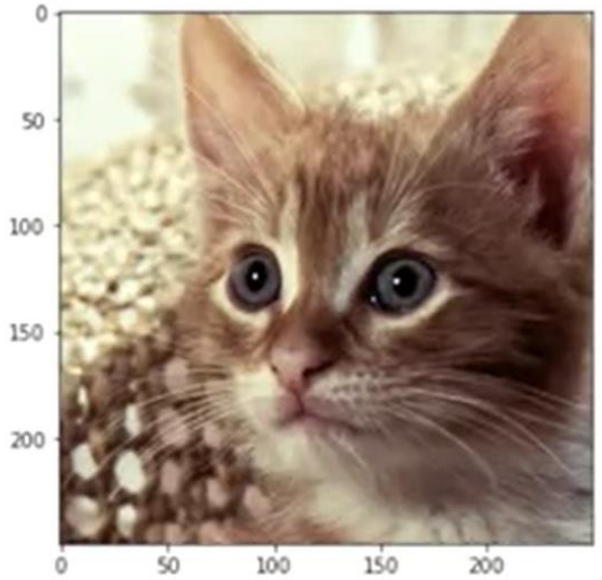
the variable “upper” is the first row,
the variable “lower” is the last row

```
upper= 150
lower=400
crop_top=image[upper: lower,:,:]
```

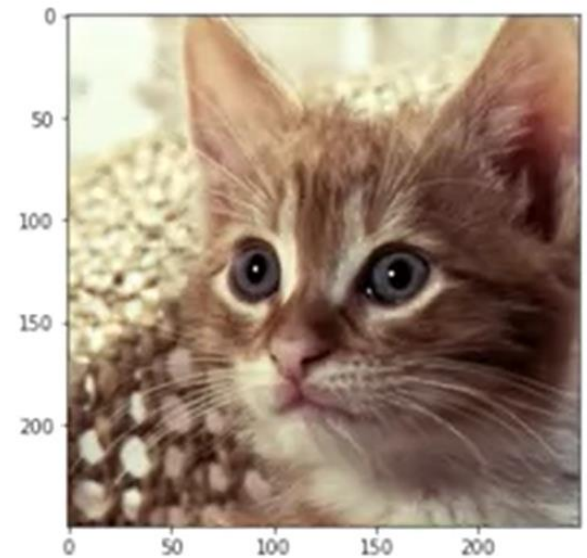
the variable “upper” is the first row,
the variable “lower” is the last row

```
left = 150
right=400
crop_horizontal=crop_top[: ,left:right,:]
```

the variable right is the first column
the variable left is the last column



```
image = Image.open("cat.png")

crop_image=image.crop((left, upper, right, lower))
```

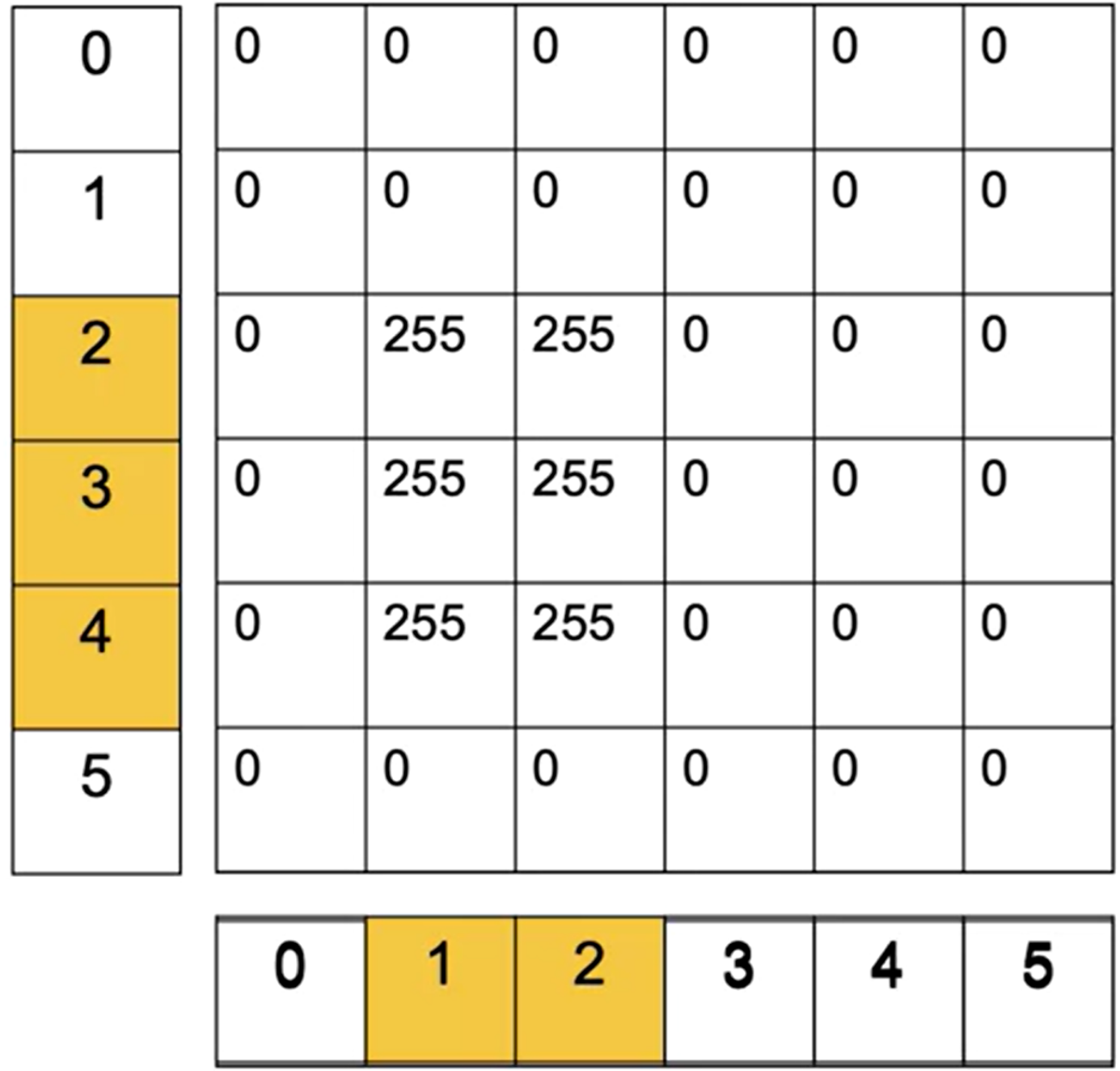


# Changing image pixels

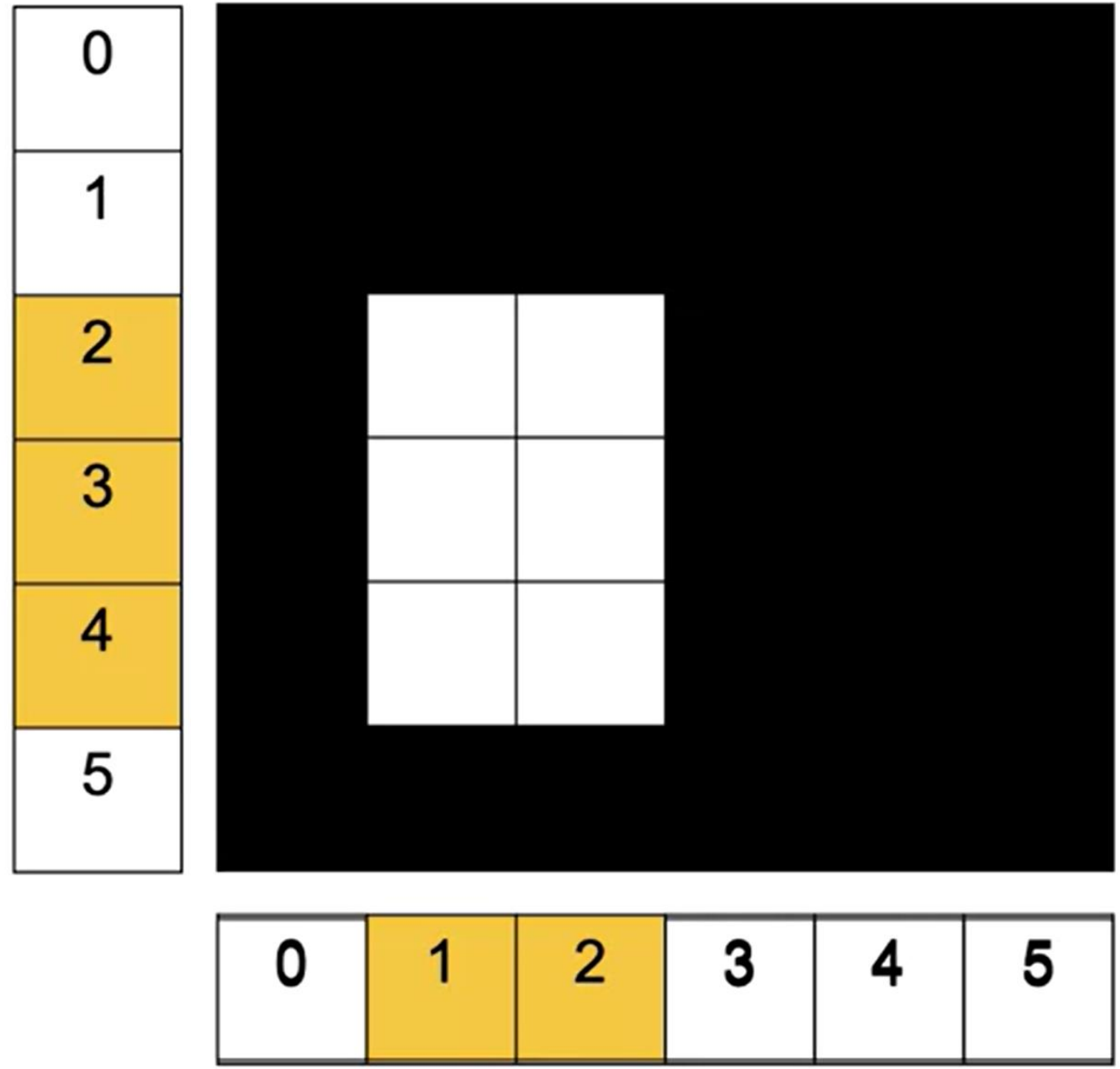
| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 |

| 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|

I[2:5,1:2]=255


0
1
2
3
4
5
0 0 0 0 0 0
0 0 0 0 0 0
0 255 255 0 0 0
0 255 255 0 0 0
0 255 255 0 0 0
0 0 0 0 0 0
0
1
2
3
4
5

I[2:5,1:2]=255



I[4,1:5]=255

I:

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 255 | 255 | 255 | 255 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 |

| 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|

I[4,1:5]=255

I[1,1]=255

I[1,4]=255

I:

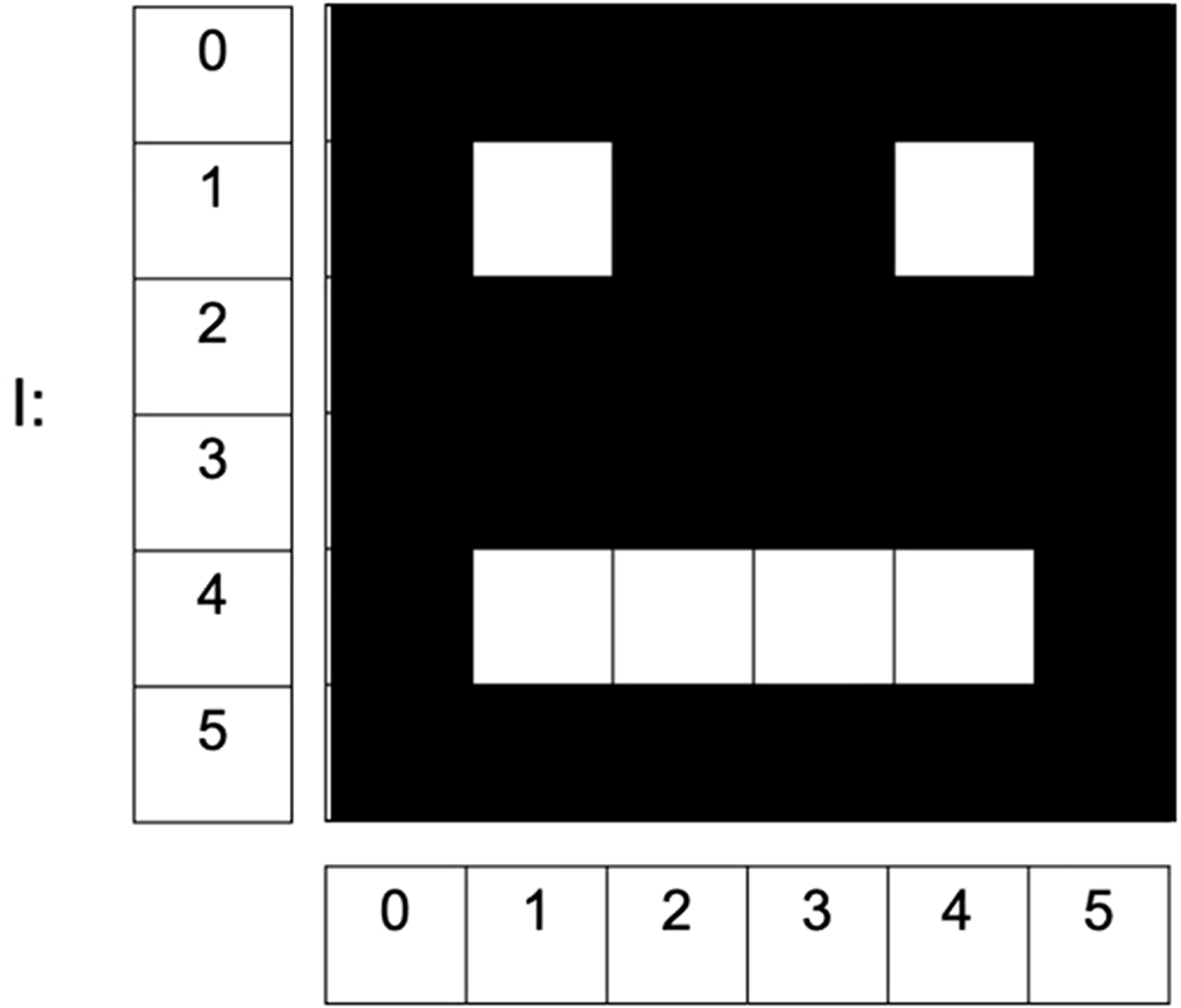
| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 255 | 0 | 0 | 255 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 255 | 255 | 255 | 255 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 |

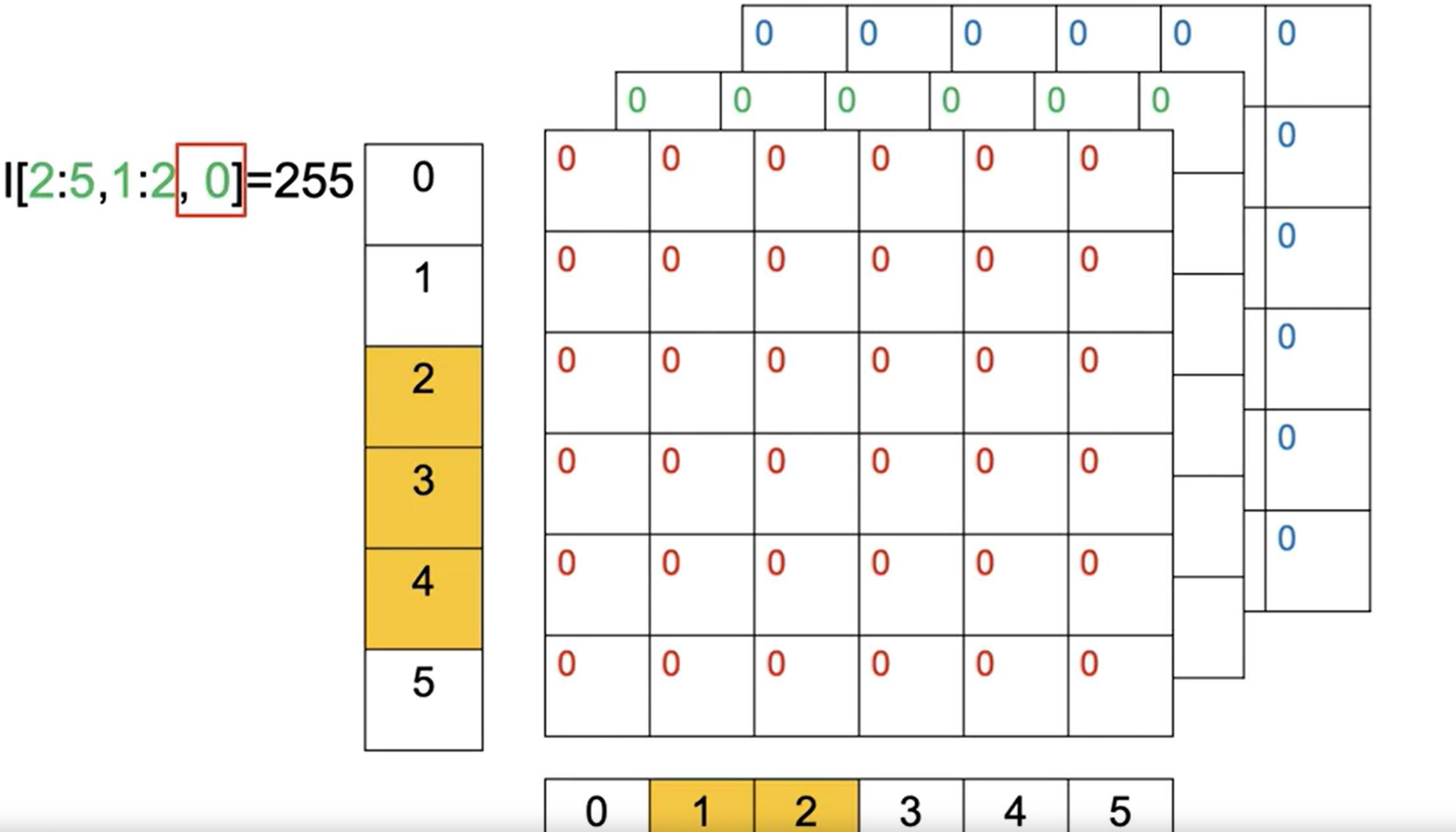
| 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|

I[4,1:5]=255

I[1,1]=255

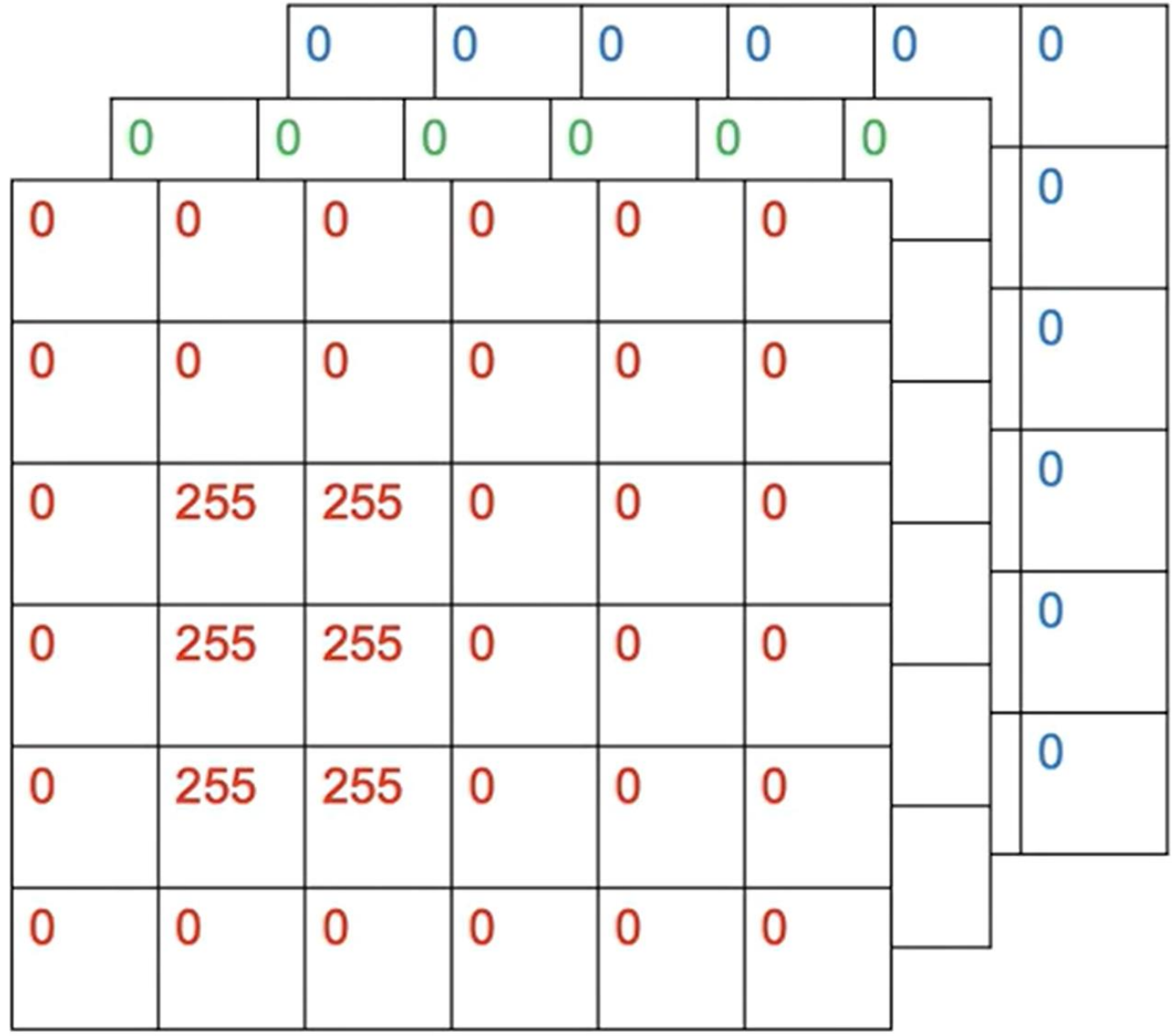
I[1,4]=255



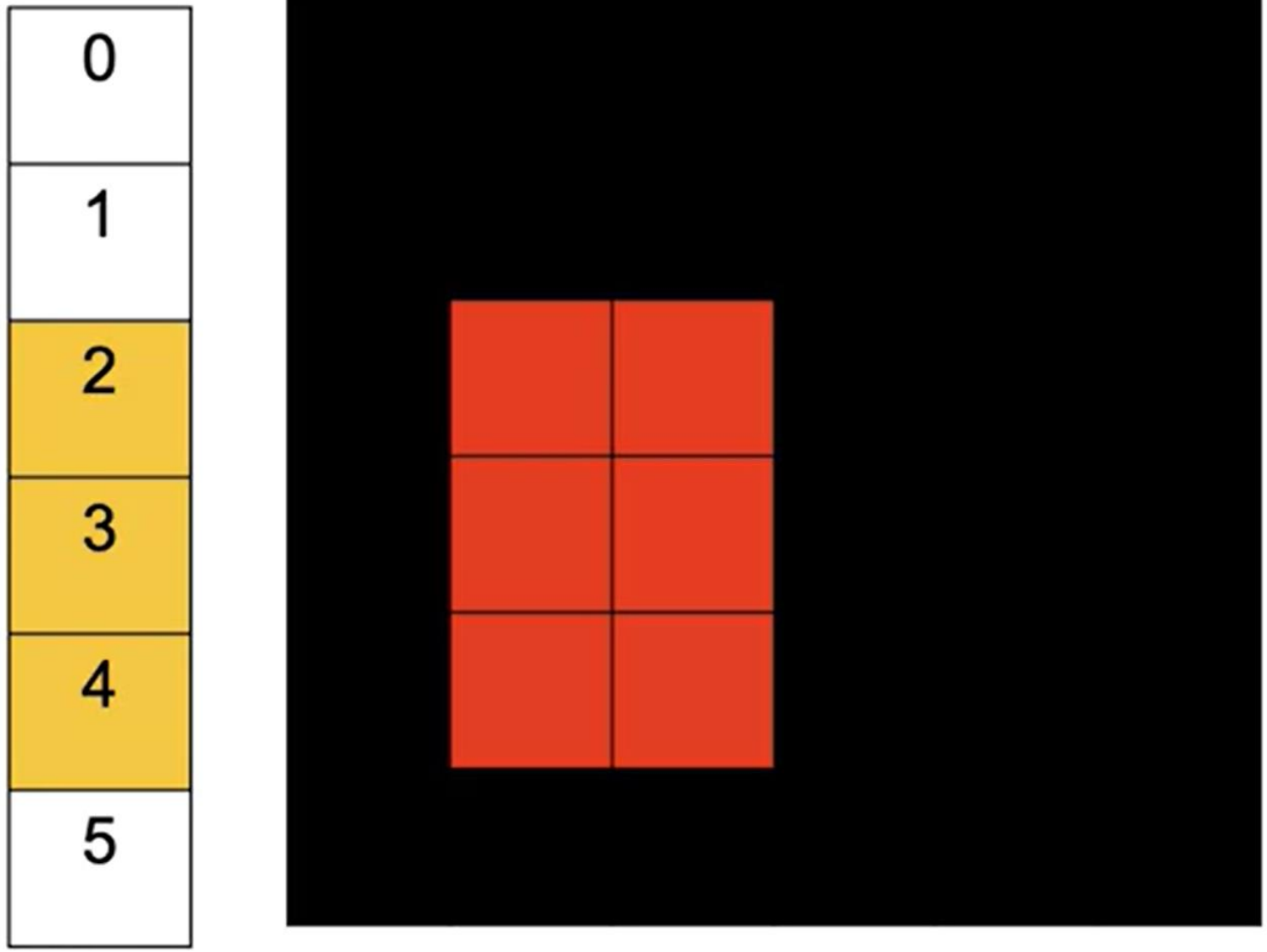

I[2:5,1:2, 0]=255
0
1
2
3
4
5
0 1 2 3 4 5

I[2:5,1:2, 0]=255


0
1
2
3
4
5

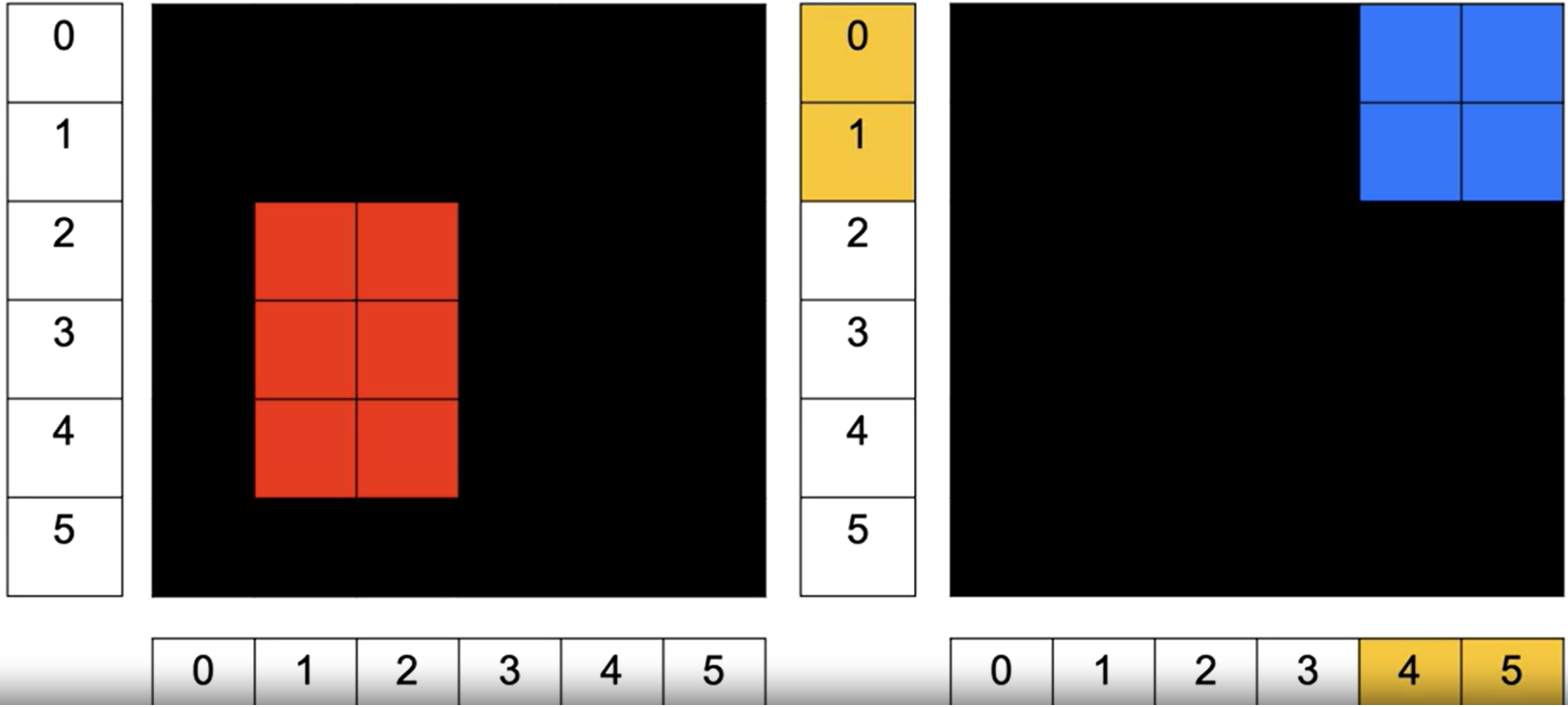

0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 255 255 0 0 0 0
0 255 255 0 0 0 0
0 255 255 0 0 0 0
0 0 0 0 0 0 0
0 1 2 3 4 5

I[2:5,1:2, 0]=255



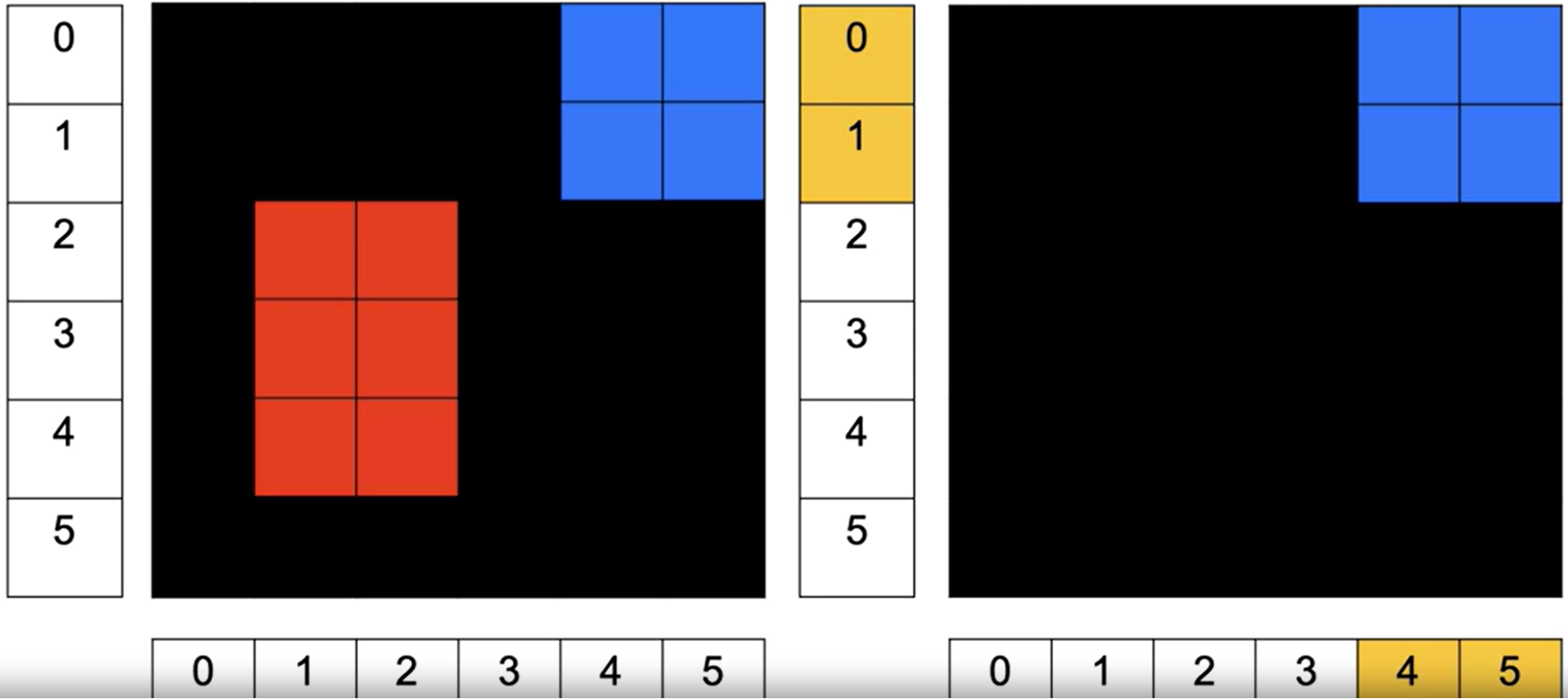
I [0:1,4:6,:] =A[0:1,4:6,:]

A

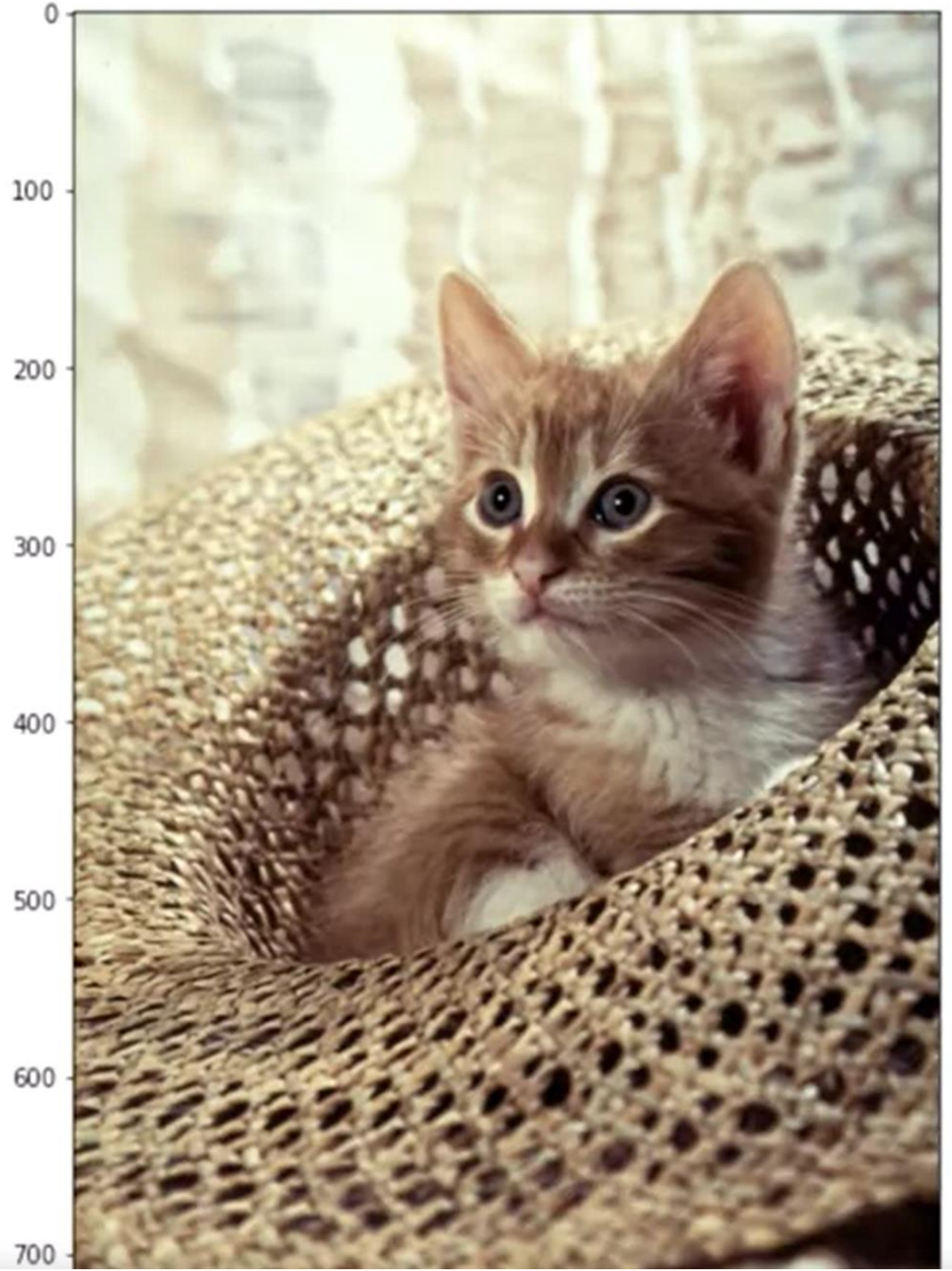


I [0:1,4:6,:] =A[0:1,4:6,:]

A

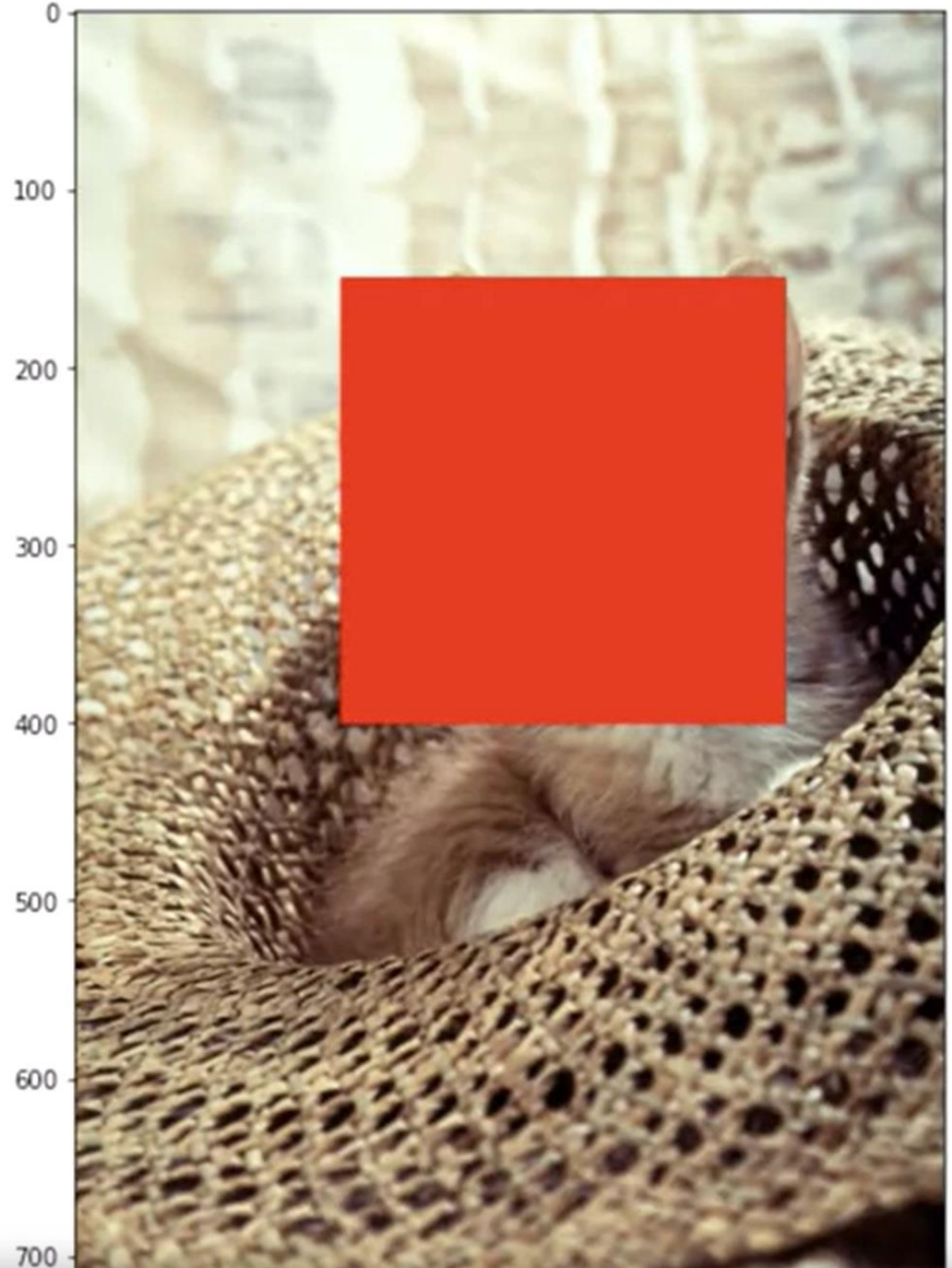


```
from PIL import  ImageDraw

image_draw=image.copy()

image_fn= ImageDraw.Draw(im=image_draw)

shape = [left, upper, right, lower]
image_fn.rectangle(xy=shape,fill="red")
```



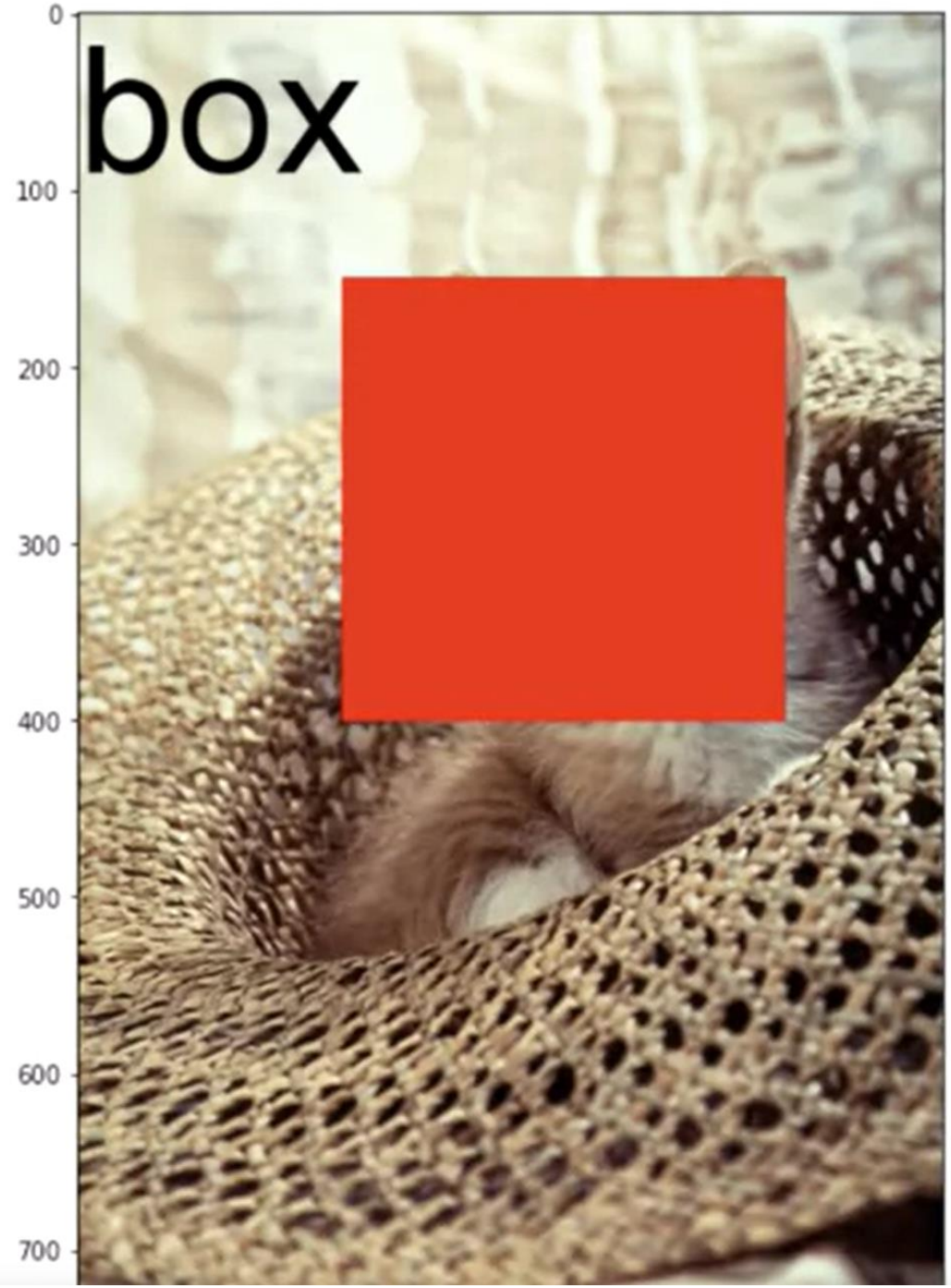
```
from PIL import  ImageDraw

image_draw=image.copy()

image_fn= ImageDraw.Draw(im=image_draw)

shape = [left, upper, right, lower]
image_fn.rectangle(xy=shape,fill="red")
```

image_draw:



```
from PIL import ImageFont

fnt = ImageFont.truetype( '/Library/Fonts/Arial.ttf' , 100)

image_fn.text(xy=(0,0),text="box",font=fnt,fill=(0,0,0))
```

image_draw:

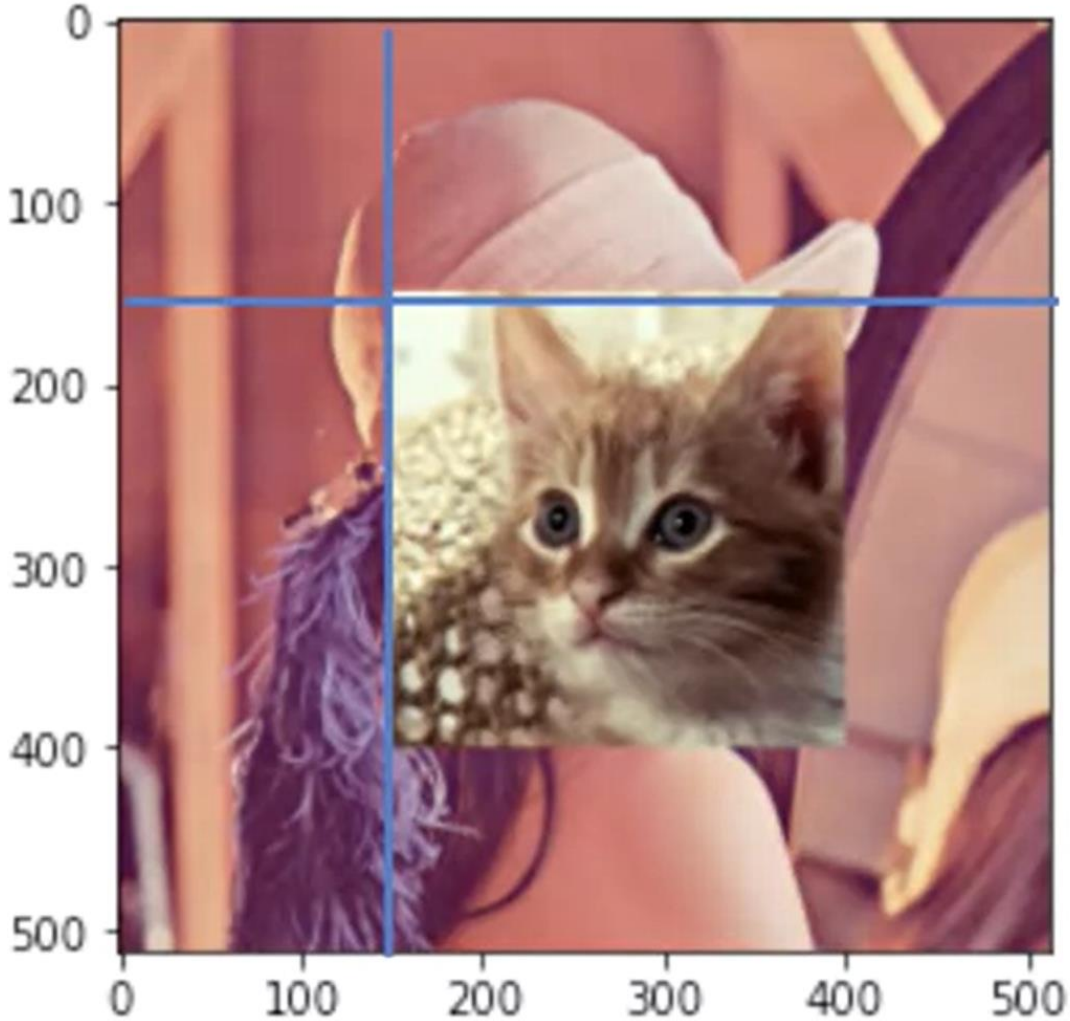


```
image_lenna = Image.open( "lenna.png" )
```

crop_image

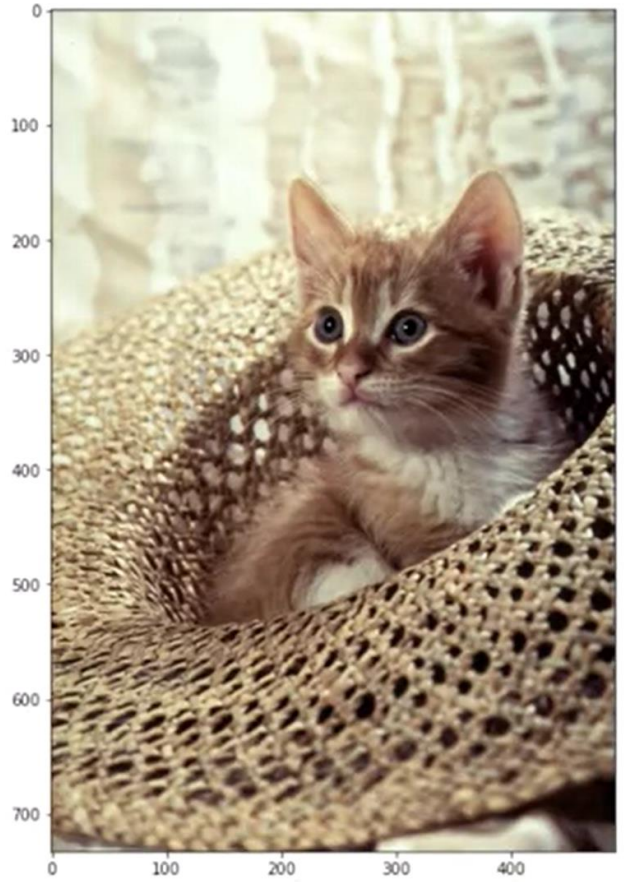
```
image_lenna = Image.open( "lenna.png" )

left = 150
upper= 150

image_lenna.paste(crop_image, box=(left,upper))
```

crop_image

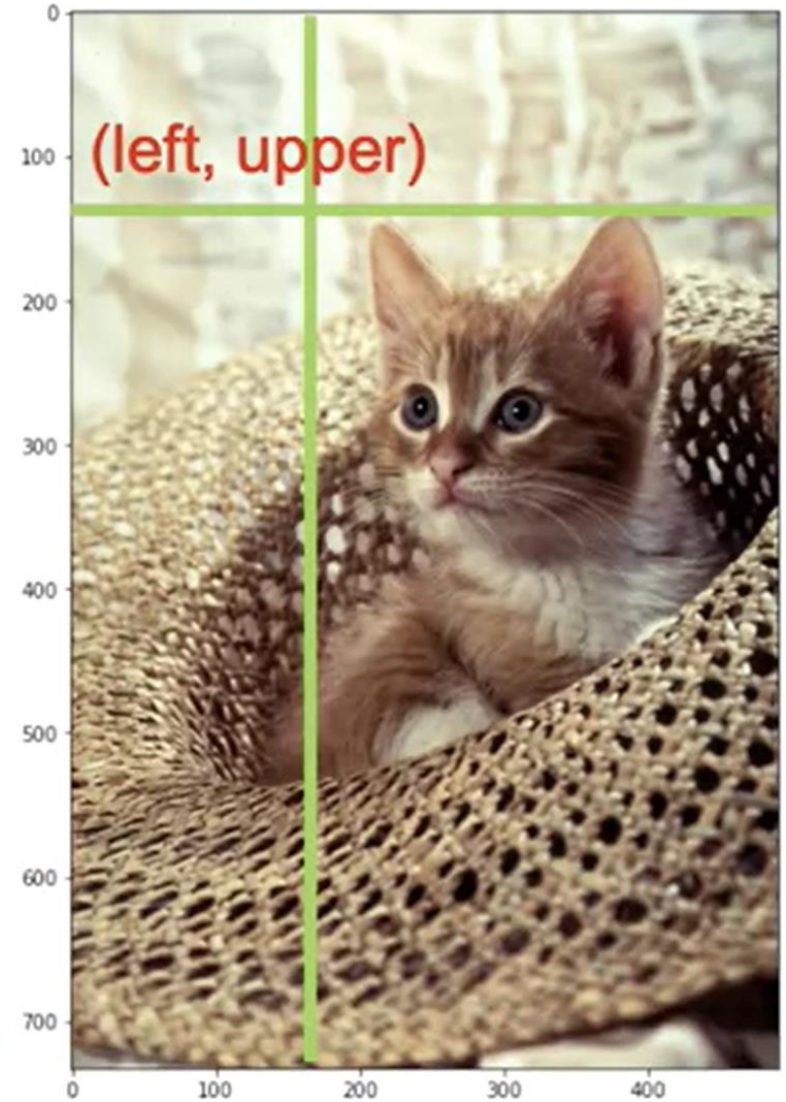
# Using OpenCV

```
image_draw=np.copy(image)

left = 150
upper= 150

right = 400
lower=400

start_point, end_point=(left, upper),(right, lower)
image_draw=np.copy(image)
```



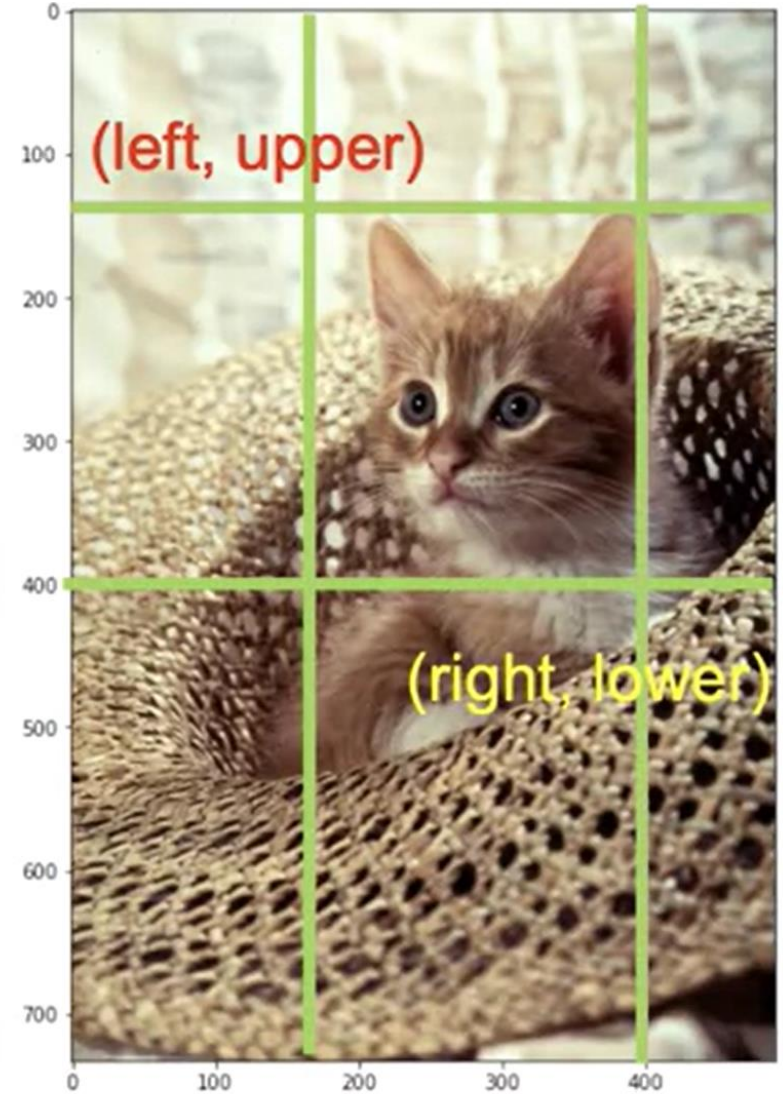

```
cv2.rectangle(image_draw, pt1=start_point, pt2=end_point, color = (0, 255, 0), thickness=3)
```

```
image_draw=np.copy(image)

left = 150
upper= 150

right = 400
lower=400

start_point, end_point=(left, upper),(right, lower)
image_draw=np.copy(image)
```



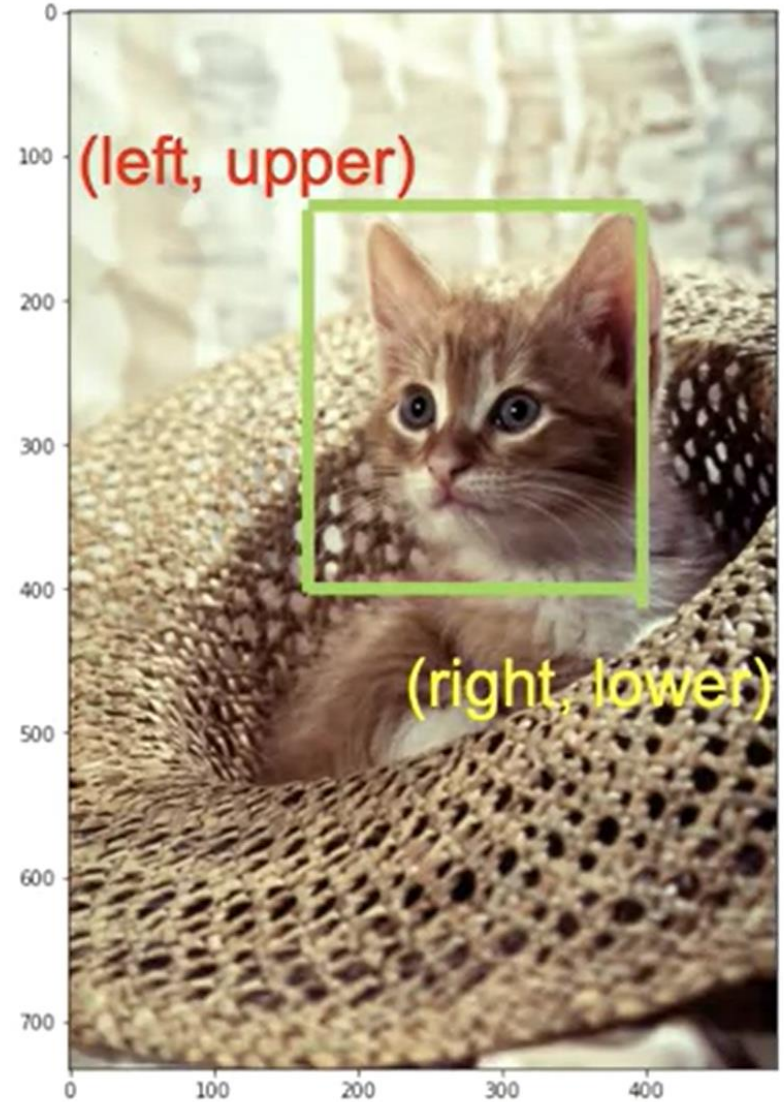

```
cv2.rectangle(image_draw, pt1=start_point, pt2=end_point, color = (0, 255, 0), thickness=3)
```

```
left = 150
upper= 150

right = 400
lower=400

start_point, end_point=(left, upper),(right, lower)
image_draw=np.copy(image)
```




```
cv2.rectangle(image_draw, pt1=start_point, pt2=end_point, color = (0, 255, 0), thickness=3)
```

```
upper= 150
lower=400

left = 150
right=400

start_point, end_point=(left, upper),(right, lower)

image_draw=np.copy(image)
```




```
cv2.rectangle(image_draw, pt1=start_point, pt2=end_point, color = (0, 255, 0), thickness=3)
```


```
cv2.putText(img=image,text='Stuff',org=(10,500),color=(255,255,255),fontFace=4,fontScale=5,thickness=2)
```

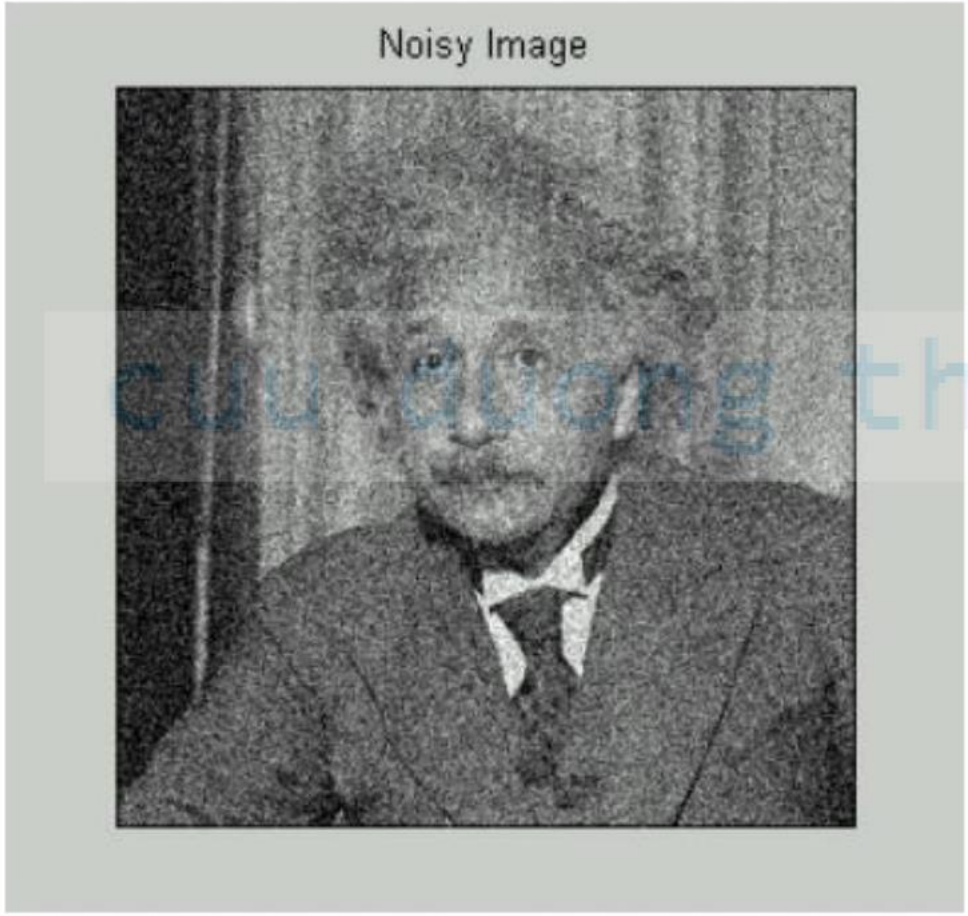
# Một số ứng dụng trong xử lý ảnh

- Cải thiện ảnh

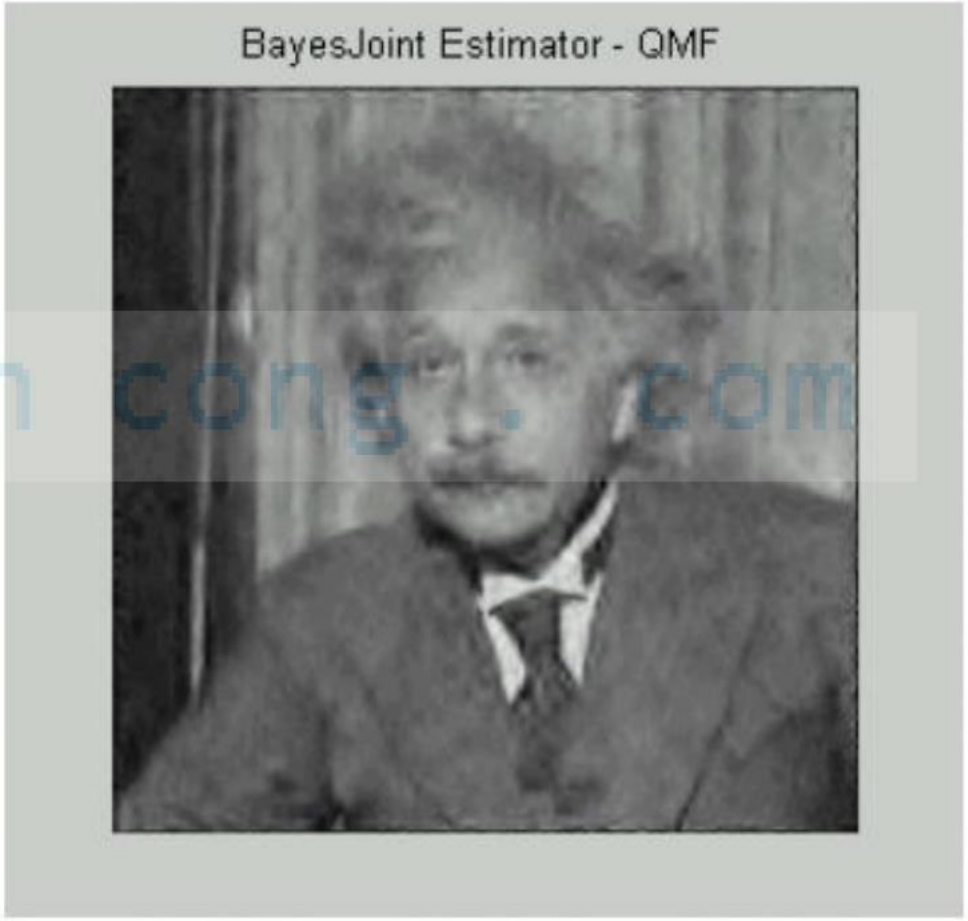
Original

Automatic Enhancement

# Giảm nhiễu

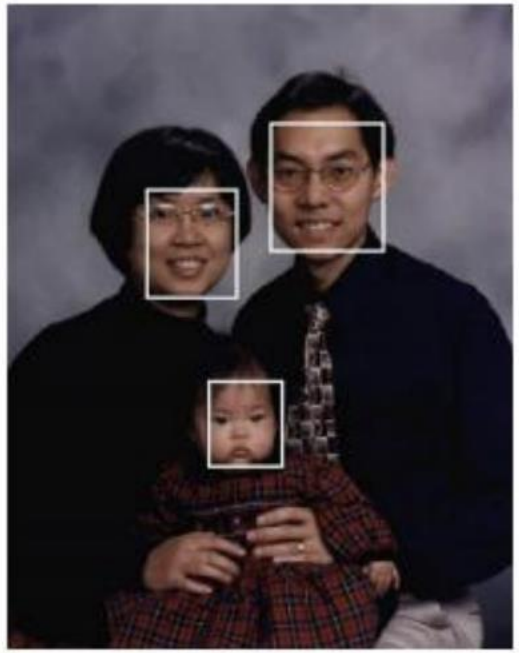



Degraded image

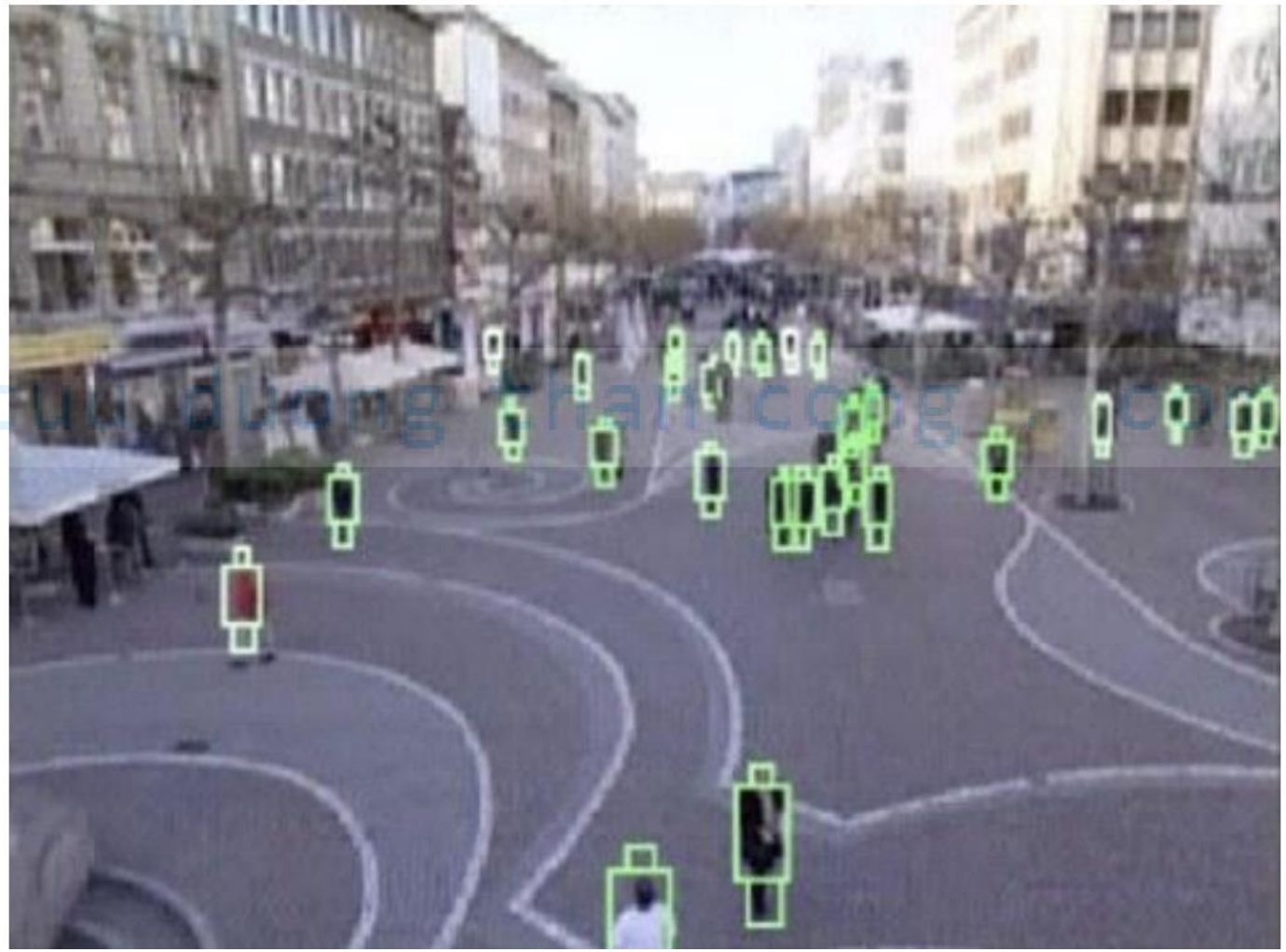



Noise-reduced image

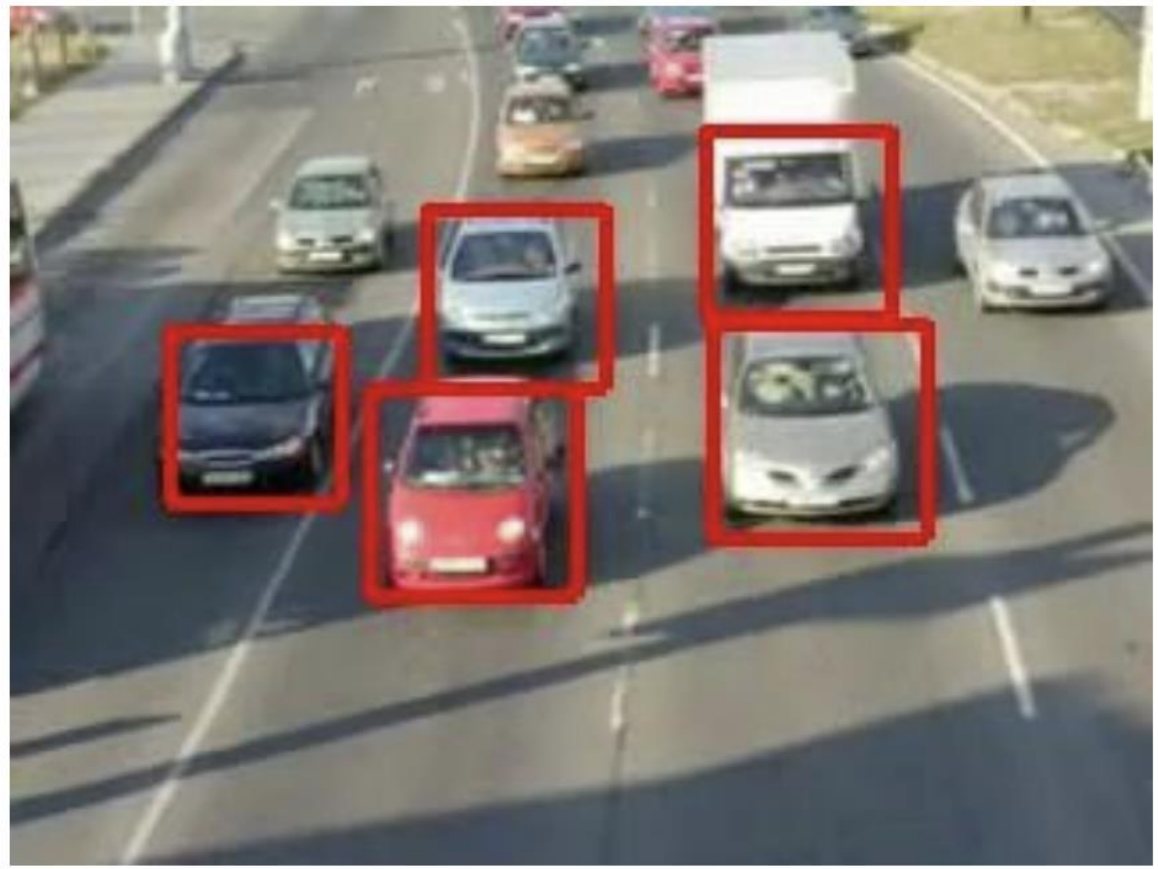
# Face Detection



# Các ứng dụng của xử lý ảnh (tiếp)

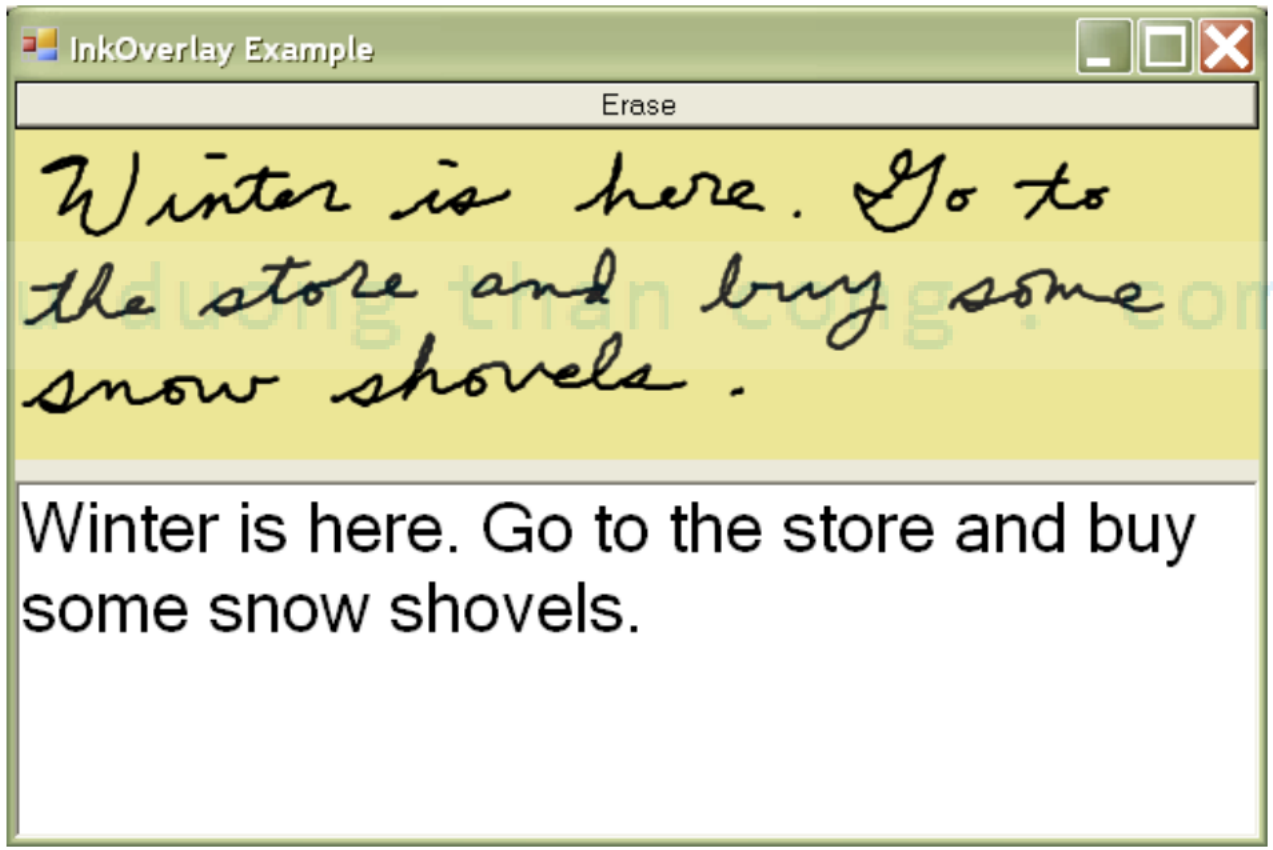


# Các ứng dụng của xử lý ảnh (tiếp)



# Các ứng dụng của xử lý ảnh (tiếp)

- Nhận dạng chữ viết

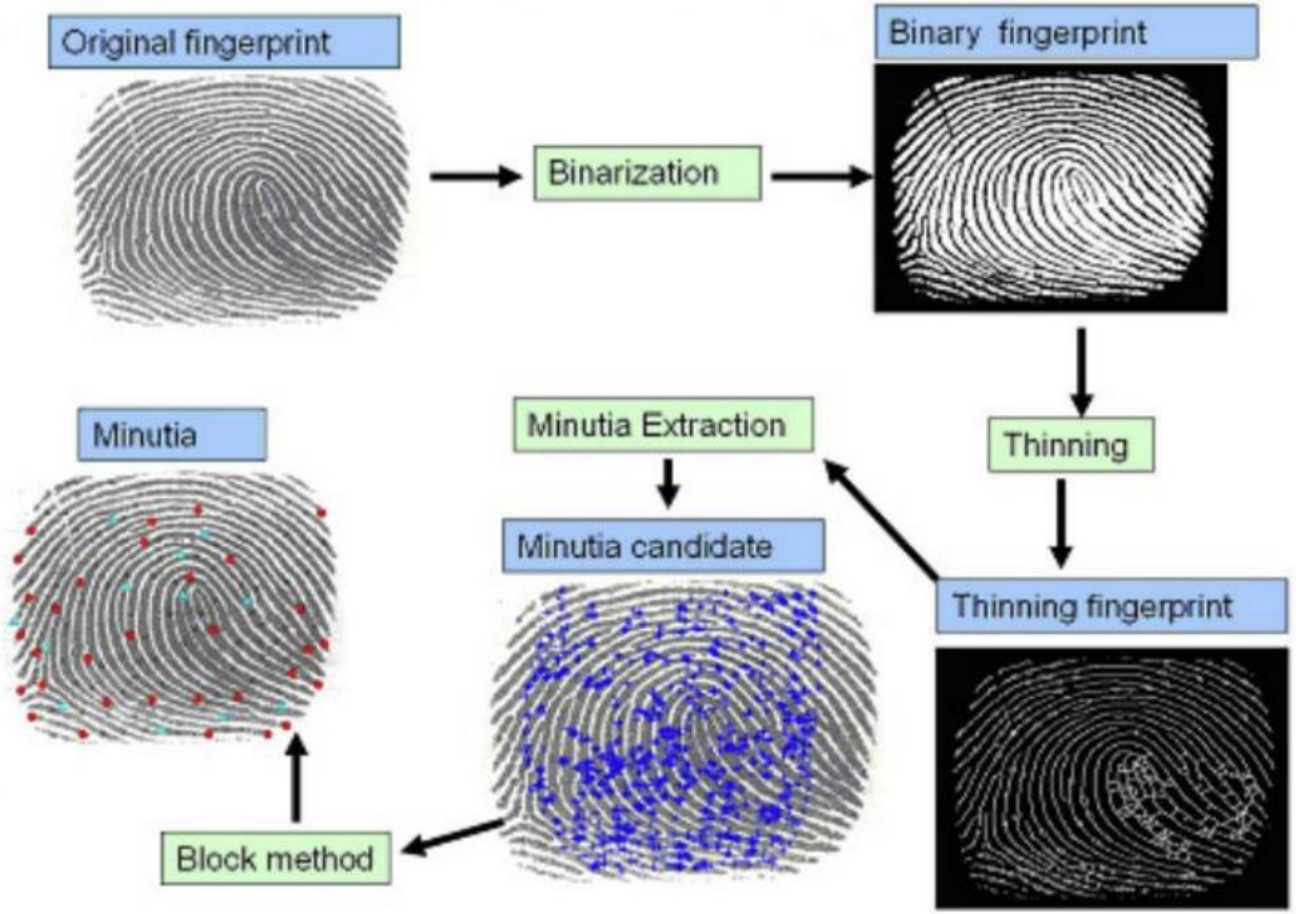


# Các ứng dụng của xử lý ảnh (tiếp)

Biometrics: Fingerprint recognition

FBI's Integrated Automated Fingerprint Identification System

IAFIS



# Các ứng dụng của xử lý ảnh (tiếp)

Biometrics: Iris recognition