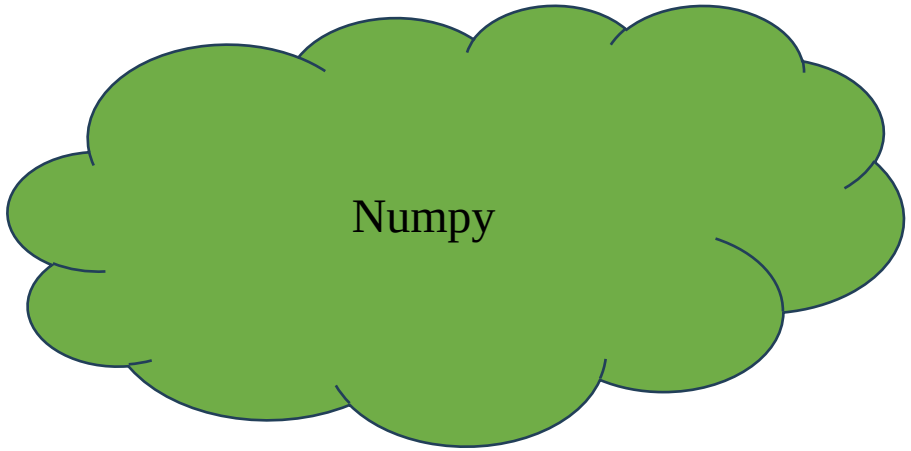
AI VIETNAM
All-in-One Course

REVIEW NUMPY
IMAGE PROCESSING

Review-Numpy array

Ôn tập kiến thức về numpy



Outline

- Numpy Examples
- Application 1: Image Brightness Change
- Application 2: Background Replacing

Numpy Example

❖ Read and save image

Grayscale Image

(Height, Width)

Color Image

(Height, Width, Channel)

Numpy Example

❖ Read and save image

```
1  import cv2
2  import numpy as np
3
4  # Đọc hình ảnh
5  image = cv2.imread('nature.jpg')
6
7
8  # Hiển thị hình ảnh
9  cv2.imshow("Image", image)
10 cv2.waitKey(0)  # Chờ người dùng nhấn một phím bất kỳ để đóng cửa sổ hiển thị
11 cv2.destroyAllWindows()  # Đóng tất cả các cửa sổ hiển thị
12
13 #Lưu hình ảnh
14 cv2.imwrite("new_nature.jpg", image)
```

Numpy Example

❖ Cắt ảnh với slicing

```
1  img_rgb_crop = image[50:150, 500:700]
```

Numpy Example

❖ Xoay ảnh – np.rot90

```
img_rot = np.rot90(image)
```

Rotate

Numpy Example

❖ Xoay ảnh - transpose

```
img_transpose = img.transpose(1, 0, 2)
```

Rotate

Numpy Example

❖ Plus

IMG1

IMG2

plus

IMG1 + IMG2

Applications

❖ Image

Grayscale Image

(Height, Width)

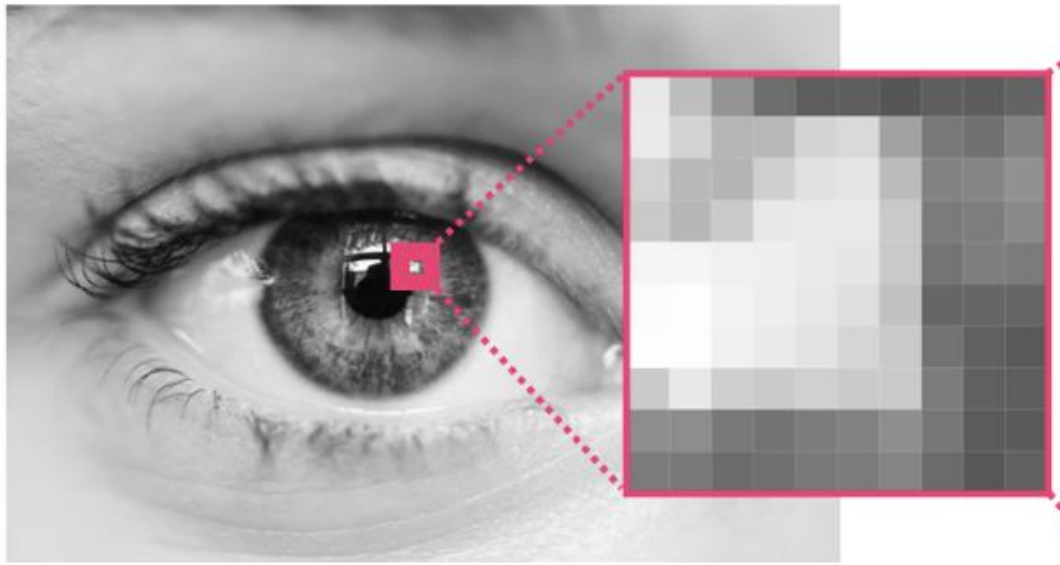
Color Image

(Height, Width, Channel)

Applications

❖ Grayscale images

Height



Width

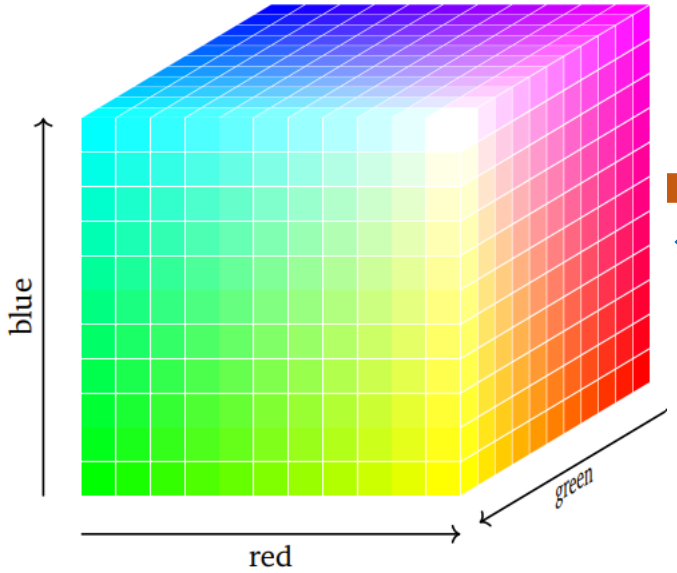
230	194	147	108	90	98	84	96	91	101
237	206	188	195	207	213	163	123	116	128
210	183	180	205	224	234	188	122	134	147
198	189	201	227	229	232	200	125	127	135
249	241	237	244	232	226	202	116	125	126
251	254	241	239	230	217	196	102	103	99
243	255	240	231	227	214	203	116	95	91
204	231	208	200	207	201	200	121	95	95
144	140	120	115	125	127	143	118	92	91
121	121	108	109	122	121	134	106	86	97

Pixel p = scalar

$0 \leq p \leq 255$

Resolution: #pixels
Resolution = HeightxWidth

Applications

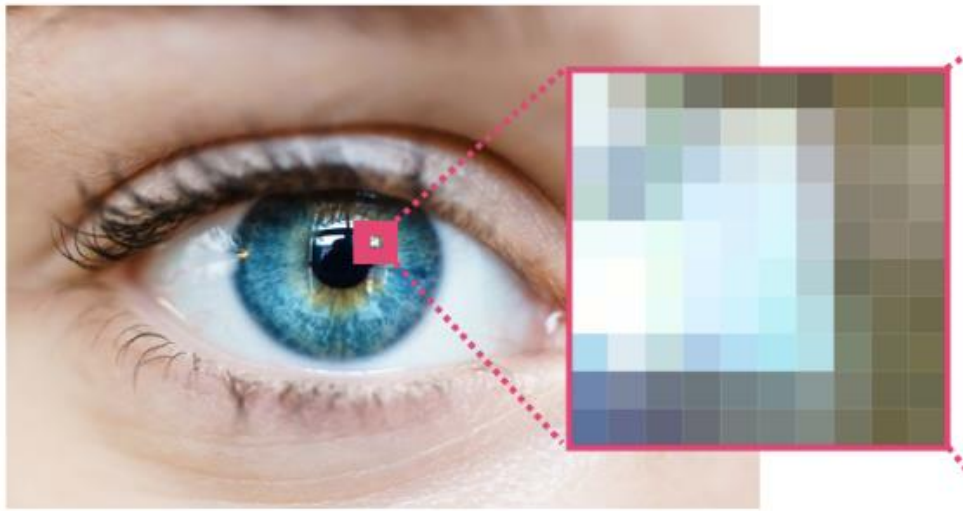


❖ Color images

RGB color image

Pixel p= $\begin{bmatrix} r \\ g \\ b \end{bmatrix}$

$0 \leq r,g,b \leq 255$



(Height, Width, channel)

	233	188	137	96	90	95	63	73	73	82	
237	202	159	120	105	110	88	107	112	121	109	
226	191	147	110	101	112	98	123	110	119	142	131
221	191	176	182	203	214	169	144	133	145	155	122
185	160	161	184	205	223	186	137	147	161	140	115
181	174	189	207	206	215	194	136	142	151	133	87
246	237	237	231	208	206	192	122	143	144	111	74
254	254	241	224	199	192	181	99	122	117	107	74
239	248	232	207	187	182	184	110	114	110	113	74
193	215	193	167	158	164	181	114	112	111	105	82
113	119	110	111	113	123	135	120	108	106	113	
93	97	91	103	107	111	122	112	104	114		

Resolution: #pixels

Resolution = HeightxWidth

https://textbooks.math.gatech.edu/ila/ila.pdf

Applications

❖ **Increase the brightness of a grayscale image**

Idea
For each pixel
    Increase pixel value
    by a value **v**

Increase brightness

Applications

❖ **Increase the brightness of a grayscale image**

Idea
For each pixel in each channel
    Decrease pixel value
    by a value **v**

Decrease brightness

Applications

❖ Load an image

```python
import cv2

# read a grayscale image
img = cv2.imread('nature.jpg', 0)

# save the image
cv2.imwrite('processed_image.jpg', img)
```

```python
# get image info

import numpy as np
import cv2

# read a grayscale image
img = cv2.imread('nature.jpg', 0)

shape = img.shape
print(shape)
```

(400, 650)

(Height, Width)

Applications

❖ Load an image

```python
import cv2

# read a color image
img = cv2.imread('nature.jpg', 1)

# save the image
cv2.imwrite('processed_image.jpg', img)
```

```python
# get image info

import numpy as np
import cv2

# read a grayscale image
img = cv2.imread('nature.jpg', 1)

shape = img.shape
print(shape)
```

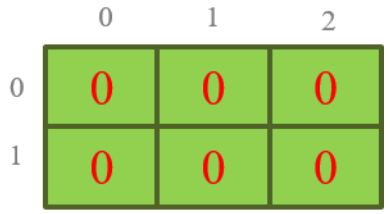
(400, 650, 3)

(Height, Width, Channel)

Applications

❖ zeros() and where() functions

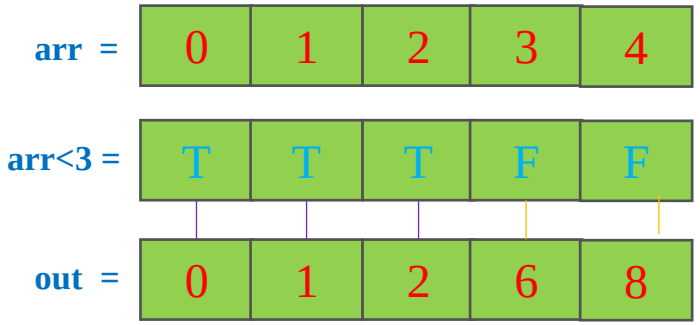
zeros() function




```
# Tạo một numpy array
# với tất cả phần tử là 0

import numpy as np

# shape: 2 dòng, 3 cột
arr = np.zeros((2,3))
print(arr)
```

```
[[0. 0. 0.]
 [0. 0. 0.]]
```

where() function



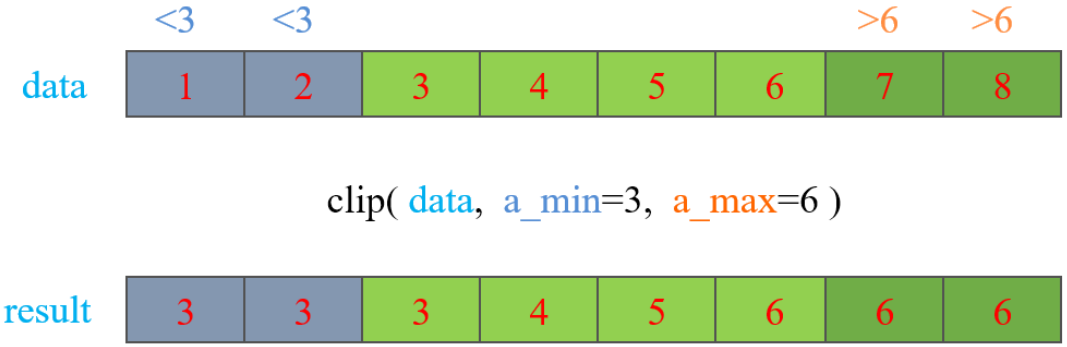

```
# create an array
arr = np.arange(5)
print(arr)

# condition
condition = arr < 3
out = np.where(condition, arr, arr*2)

print(condition)
print(out)
```

```
[0 1 2 3 4]
[ True  True  True False False]
[0 1 2 6 8]
```

Applications

❖ clip() function

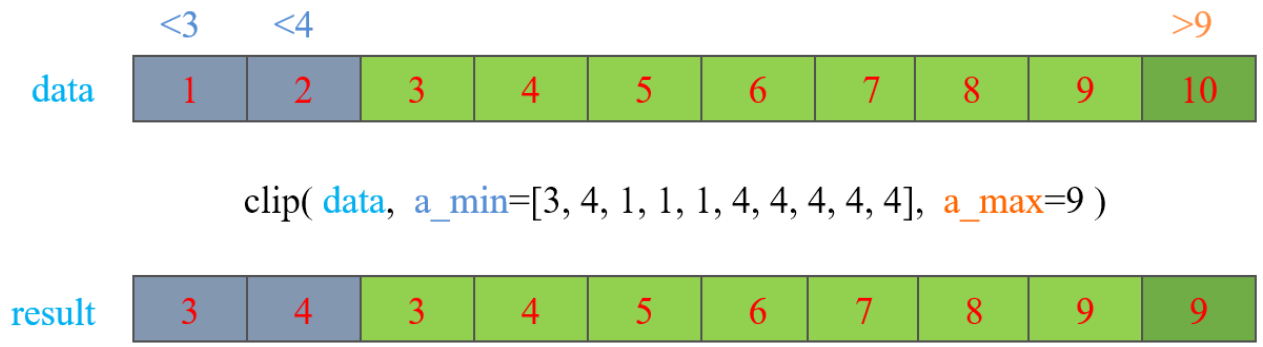
numpy.clip()




```
4  import numpy as np
5
6  data = np.array([1, 2, 3, 4, 5, 6, 7, 8])
7  print("data: ", data)
8
9  # element < 3 sẽ gán bằng 3
10 # element > 6 sẽ gán bằng 6
11 result = np.clip(data, a_min=3, a_max=6)
12 print("result: ", result)
```

```
data:   [1 2 3 4 5 6 7 8]
result:   [3 3 3 4 5 6 6 6]
```

numpy.clip()




```
4  import numpy as np
5
6  # create data
7  data = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
8  print ("data: ", data)
9
10 # clip values
11 min_list = [3, 4, 1, 1, 1, 4, 4, 4, 4, 4]
12 result = np.clip(data, a_min=min_list, a_max=9)
13 print("result: ", result)
```

```
data:   [ 1  2  3  4  5  6  7  8  9 10]
result:   [3 4 3 4 5 6 7 8 9 9]
```

Applications

❖ Increase brightness - Implementation - 1

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

img = cv2.imread('nature.jpg', 1)
img = img.astype(float)

h, w, c = img.shape
for i in range(h):
    for j in range(w):
        for k in range(c):
            img[i, j, k] = img[i, j, k] + 50
            if img[i, j, k] <0:
                img[i, j, k] = 0
            elif img[i, j, k] >255:
                img[i, j, k] = 255

img = img.astype('uint8')
cv2.imwrite('increase50_loop.jpg', img)
cv2.imshow("img", img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Applications

❖ Increase brightness - Implementation - 2

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

img = cv2.imread('nature.jpg', 1)
img = img.astype(float)

img = img + 50
img = np.clip(img, 0, 255)

img = img.astype(np.uint8)
cv2.imwrite('increase50_clip.jpg', img)
cv2.imshow("img", img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Applications

❖ Increase brightness - Implementation - 3

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

img = cv2.imread('nature.jpg', 1)
img = img.astype(float)

img = img + 50
img = np.where(img<0., 0., img)
img = np.where(img>255., 255., img)

img = img.astype(np.uint8)
cv2.imwrite('increase50_where.jpg', img)
```

Applications

❖ Decrease brightness - Implementation - 1

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

img = cv2.imread('nature.jpg', 1)
img = img.astype(float)

h, w, c = img.shape
for i in range(h):
    for j in range(w):
        for k in range(c):
            img[i, j, k] = img[i, j, k] -80
            if img[i, j, k] <0:
                img[i, j, k] = 0
            elif img[i, j, k] >255:
                img[i, j, k] = 255

img = img.astype('uint8')
cv2.imwrite('decrease80_loop.jpg', img)
cv2.imshow("img", img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Applications

❖ Decrease brightness - Implementation - 2

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

img = cv2.imread('nature.jpg', 1)
img = img.astype(float)

img = img - 80
img = np.clip(img, 0, 255)

img = img.astype(np.uint8)
cv2.imwrite('decrease80_clip.jpg', img)
cv2.imshow("img", img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Applications

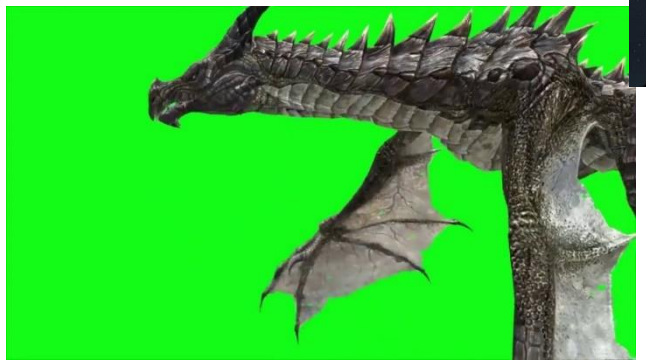
❖ Decrease brightness - Implementation - 3

```
1  import cv2
2  import numpy as np
3  import matplotlib.pyplot as plt
4
5  img = cv2.imread('nature.jpg', 1)
6  img = img.astype(float)
7
8  img = img - 80
9  img = np.where(img<0., 0., img)
10 img = np.where(img>255., 255., img)
11
12 img = img.astype(np.uint8)
13 cv2.imwrite('decrease80_where.jpg', img)
14 cv2.imshow("img", img)
15 cv2.waitKey(0)
16 cv2.destroyAllWindows()
```

Outline

- Numpy Examples
- Application 1: Image Brightness Change
- Application 2: Background Replacing

Background Subtraction



INPUT

OUTPUT

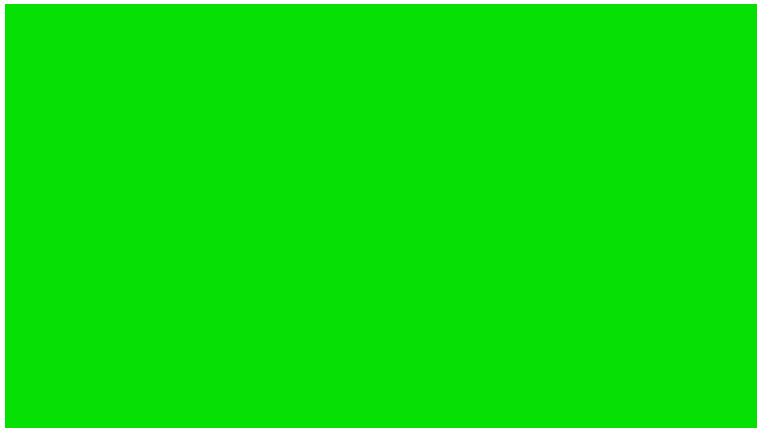
Background Subtraction

Background

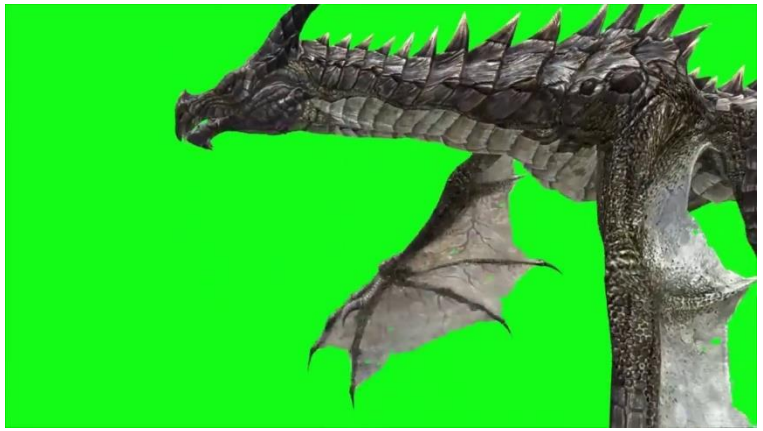
coming image

Background Subtraction

Background



coming image

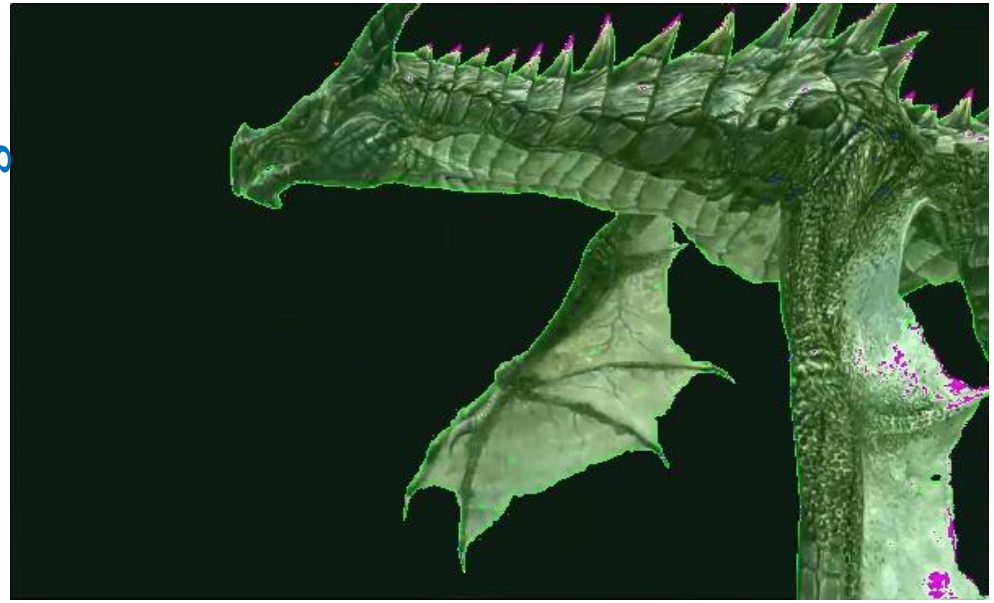


Subtraction

different image

Background Subtraction

Different image



New background

Addition

Output