2022-1 영상정보처리 3주차 과제 템플리트

마감: 2021년 3월 23일 오후 11시 59분

이름: 이현정
학번: 32203660

구글 드라이브를 연결하고 자신의 노트북이 저장되어 있는 폴더로 이동하시오

```
from google.colab import drive
drive.mount('/gdrive')
%cd /gdrive/MyDrive/ImageProcClass/Notebook-Week3
```

```
Mounted at /gdrive
/gdrive/MyDrive/ImageProcClass/Notebook-Week3
```

다음에 이미지 경로 '../Dongkeun-OpenCV-ImgData/logo.png' 를 변경하지 말고 이미지를 읽고, 해당 이미지에 대한 type, size, dtype 등의 속성을 체크하시오.

```
import cv2
img = cv2.imread('../Dongkeun-OpenCV-ImgData/data/opencv_logo.png')
print("type =", type(img))
print("size =", img.size)
print("dtype =", img.dtype)
```

```
type = <class 'numpy.ndarray'>
size = 102684
dtype = uint8
```

다음의 셀에 읽은 이미지에서 좌표 y = 100, x = 50 에 있는 화소의 각 색요소 값을 한번에 읽어 출력하시오. 또한 각 색요소의 값을 별도로 가져오는 예를 작성하시오.

```
# 화소의 각 색요소 값을 한번에 읽기
(b, g, r) = img[100, 50]
print("blue =", b)
print("green =", g)
print("red =", r)

# 색요소의 값을 별도로 가져오는 예
blue = img[10, 20, 0]
green = img[10, 20, 1]
red = img[10, 20, 2]
print()
print("blue =", blue)
print("green =", green)
print("red =", red)
```

```
blue = 0
green = 255
red = 0

blue = 255
green = 255
red = 255
```

다음의 셀에 읽은 이미지에서 좌표 y = 100, x = 50 의 화소를 순수 녹색(green) 으로 세팅하는 코드를 작성하고 해당 분야를 ROI 를 이용해 확대해서 출력하시오.

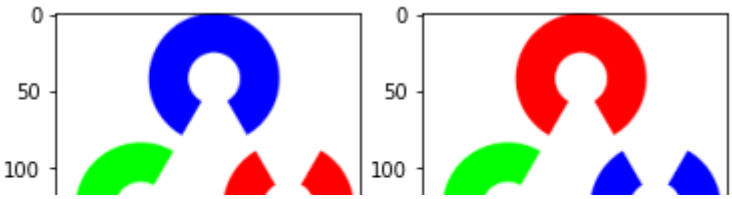
```
import matplotlib.pyplot as plt

img[100,50] = (0,255,0) # 주위 색이 순수 녹색이라서 보이지 않음
plt.imshow(img[95:105, 45:55])
plt.show()
```

Accessing and Manipulating pixels in opencv with grayscale images

위에서 사용한 동일 입력 이미지를 읽어서 matplotlib.pyplot에 맞는 채널 순서로 변환하고 본래 이미지와 변환된 이미지를 matplotlib.pyplot 을 이용하여 출력하시오.

```
img_origin = cv2.imread('../Dongkeun-OpenCV-ImgData/data/opencv_logo.png')
(b,g,r) = cv2.split(img_origin)
img_matplotlib = cv2.merge([r,g,b])
plt.subplot(121)
plt.imshow(img_origin)
plt.subplot(122)
plt.imshow(img_matplotlib)
plt.show()
```



In Depth Example

강의노트 'In Depth' 부분에 있는 예제를 임의의 값을 이용하여 예시하고 간단한 설명을 붙여 자기 노트를 완성하시오.

```
import numpy as np

# 0차원 배열
x1 = np.array(5)
# 1차원 배열
x2 = np.array([1, 0.1, -0.2])
# 2차원 배열
x3 = np.array([[1,2],[3,4]])
# 3차원 배열
x4 = np.array([[[1,2],[3,4]],[[5,6],[7,8]]])

print('x1 =', x1)
print('x2 =',x2)
print('x3 =',x3)
print('x4 =',x4)
```

```
x1 = 5
x2 = [ 1.   0.1 -0.2]
x3 = [[1 2]
 [3 4]]
x4 = [[[1 2]
  [3 4]]

 [[5 6]
  [7 8]]]
```

```
# 1차원 배열
n = np.array([1,2,3,4,5])
print('n =', n)

# 행렬 형태
print('n.shape = ', n.shape)
print()

# 인덱싱
print('n[1] =', n[1]) # 2번째
print('n[-1] =', n[-1]) # 마지막 = 5번째
print()

print('n[1:4] =', n[1:4]) # 2~4번째
print('n[1:-1] =', n[1:-1]) # 2~4번째
print()
```

```
print('n[0::2]=:', n[0::2]) # 1, 3 ,5번째
```

```
n = [1 2 3 4 5]
n.shape =  (5,)

n[1] = 2
n[-1] = 5

n[1:4] = [2 3 4]
n[1:-1] = [2 3 4]

n[0::2]=: [1 3 5]
```

```
# 2차원 np 배열
n2 = np.array([[1,2,3],[4,5,6]])
print('n2 =', n2)

# 행렬 형태
print('n2.shape = ', n2.shape)
print()

# 인덱싱
print('n2[1,0] =', n2[1,0]) # 2행 1열
print('n2[-1,1] =', n2[-1,1]) # 2행 2열
```

```
n2 = [[1 2 3]
 [4 5 6]]
n2.shape =  (2, 3)

n2[1,0] = 4
n2[-1,1] = 5
```

```
m = np.array([[[1,2],[3,4]],[[5,6],[7,8]]])
print('m :', m)

# 행렬 형태
print('m.shape :',m.shape)
print()

print('m[0,1:,::-1] :', m[0,1:,::-1]) # 1장 2행의 원소들이 역순으로 출력
```

```
m : [[[1 2]
  [3 4]]

 [[5 6]
  [7 8]]]
m.shape : (2, 2, 2)

m[0,1:,::-1] : [[4 3]]
```

✓ 0초 오후 2:01에 완료됨