Image Processing
실습 2주차

신 동 헌

Department of Computer Science and Engineering

Chungnam National University, Korea

실습 소개

- **과목 홈페이지**
  - 충남대학교 사이버 캠퍼스 (http://e-learn.cnu.ac.kr)

- **TA 연락처**
  - 공대 5호관 531호 컴퓨터비전 연구실
  - 과제 질문은 [IP]를 제목에 붙여 메일로 주세요.
  - 00반
    - 안준혁
    - ajh99345@gmail.com
  - 01반
    - 신동헌
    - doghon85@naver.com

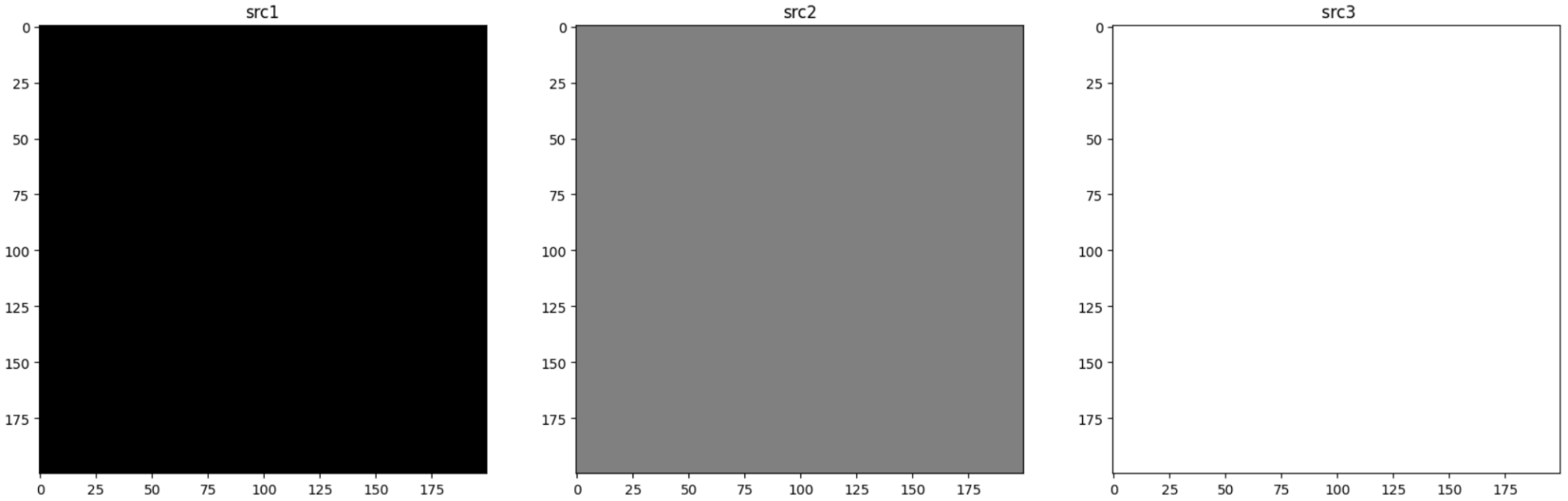
목차

- **영상 기초**
- **실습**
  - OpenCV 기초
- **과제**
  - myVideo_BGR2GRAY

영상 기초

- **영상 자료형 이해**
  - 8bit 부호가 없는 정수형(uint8)은 0 ~ 255 값을 가짐
  - 값이 0에 가까울 수록 어둡고, 255에 가까울 수록 밝음

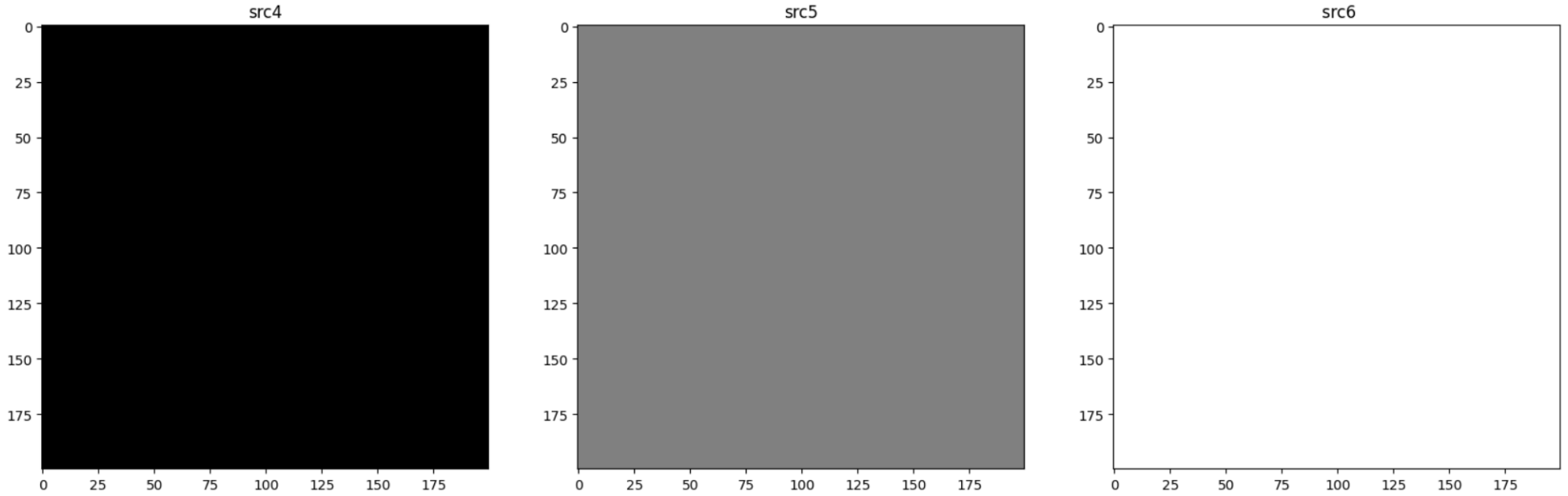
```
src1 = np.zeros((200, 200), dtype=np.uint8) # 값이 모두 0
src2 = np.full((200, 200), 128, dtype=np.uint8) # 값이 모두 128
src3 = np.full((200, 200), 255, dtype=np.uint8) # 값이 모두 255
```



영상 기초

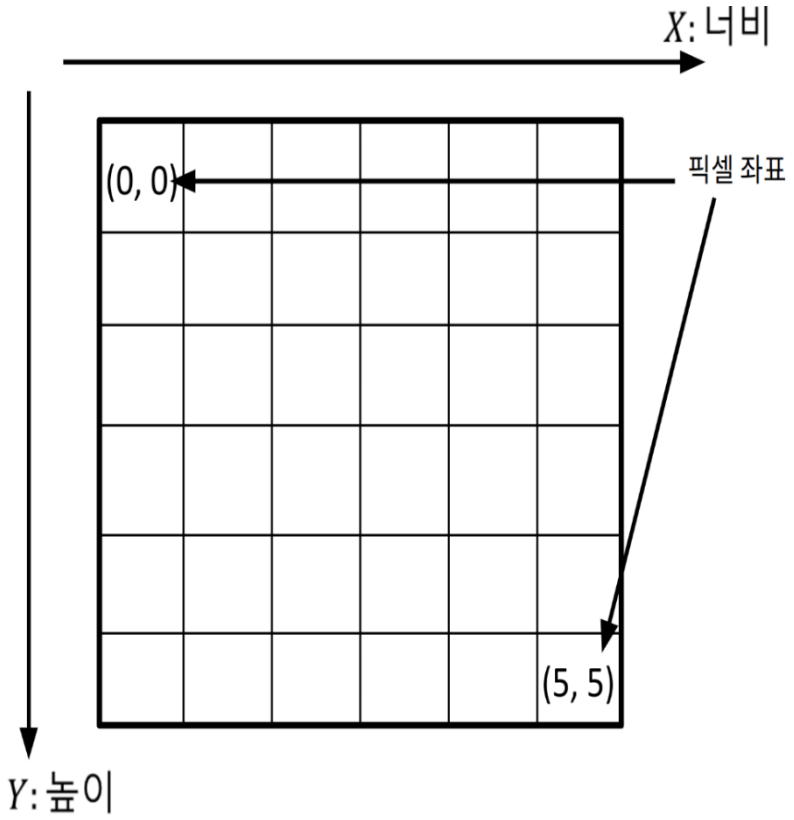
- **영상 자료형 이해**
  - 정규화된 실수형(float)은 0. ~ 1. 값을 가짐
  - 값이 0.에 가까울 수록 어둡고, 1.에 가까울 수록 밝음

```
src4 = np.zeros((200, 200), dtype=np.float32) # 값이 모두 0
src5 = np.full((200, 200), 0.5, dtype=np.float32) # 값이 모두 0.5
src6 = np.full((200, 200), 1, dtype=np.float32) # 값이 모두 1
```



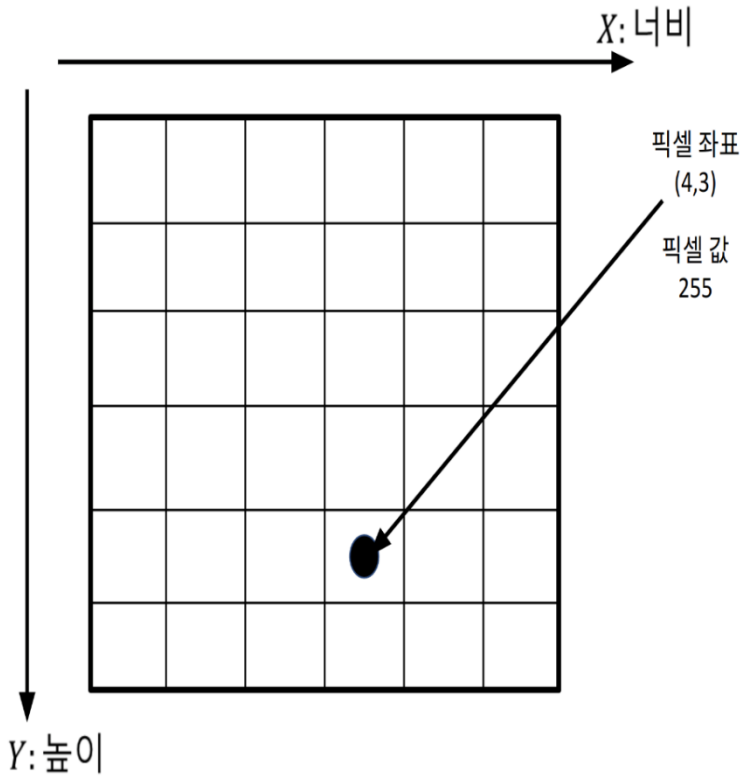
영상 기초

- **영상 구조 이해**
  - 흑백 이미지는 (높이, 너비), 컬러 이미지는 (높이, 너비, 채널) 구조
  - 각 픽셀들은 좌표 값과 해당 좌표에서의 픽셀 값을 가짐
  - 픽셀 값은 밝기의 정도를 나타냄



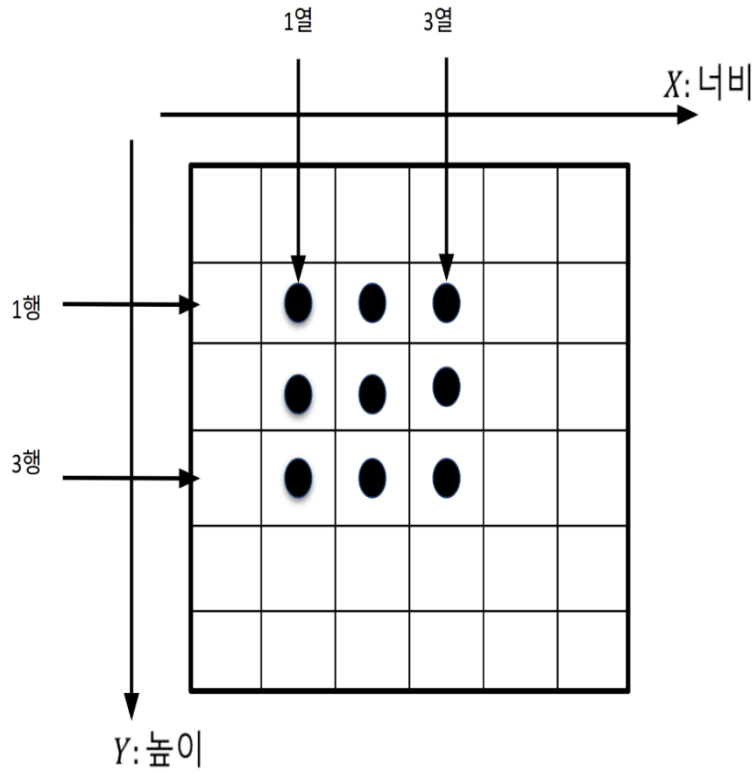
영상 기초

- **영상의 특정 좌표 값 접근**
  - 영상의 특정 좌표에 대한 픽셀 값의 접근은 인덱싱을 사용
    - Image[행, 열]: 행과 열은 0을 포함하는 양의 정수
    - 예를 들어, 영상의 (4, 3) 좌표에서의 값은 Image[4, 3]과 같이 표현
    - 주의: 영상 좌표는 (0, 0)부터 시작



영상 기초

- **영상의 특정 영역의 값들 접근**
  - 영상의 특정 영역에 대한 픽셀 값들의 접근은 슬라이싱을 사용
    - Image[m:n+1, i:j+1]: m행 ~ n행, i열 ~ j열
    - 예를 들어, 1행 ~ 3행, 1열 ~ 3열까지의 영역의 좌표 값은 Image[1:4, 1:4]



영상 기초

- **영상 저장**
  - OpenCV 라이브러리의 cv2.imwrite()를 사용해서 이미지 저장
    - cv2.imwrite(영상 경로, 저장할 영상)
  - 영상 저장시 dtype은 uint8 이어야함
  - 저장할 영상이 정규화된 실수형이면 아래와 같은 단계로 저장
    - 영상에 255를 곱함
    - 영상을 반올림 해줌
    - 영상의 픽셀 범위를 0 ~ 255로 조정(clipping 진행)
    - 데이터 타입을 uint8로 설정

```
# 저장할 때, 확장자는 일반적으로 .png을 쓴다
uint8_image = np.clip(np.round(gray_img * 255), 0, 255).astype(np.uint8)
cv2.imwrite('gray_image_save.png', uint8_image)
```

실습

- **OpenCV 함수**
  - 이미지 프로세싱 관련 라이브러리
    - import cv2로 사용
  - cv2.imread(file_path, flag=cv2.IMREAD_COLOR)
    - file_path: 이미지 경로(String)
    - flag: 이미지 읽기 플래그(Color 형식)
      - cv2.IMREAD_GRAYSCALE: 흑백으로 이미지를 읽음(정수값 0)
      - cv2.IMREAD_COLOR: BGR로 이미지를 읽음(정수값 1)
      - cv2.IMREAD_UNCHANGED: 원본 그대로 이미지 읽음(정수값 -1)
  - cv2.imwrite(file_path, img)
    - file_path: 이미지를 저장할 경로(String) / .png, .jpg 등 확장자 필요
    - img: 저장될 이미지
  - cv2.imshow(window_name, img)
    - window_name: 이미지가 표시 될 윈도우 이름
    - img: 윈도우에 표시 될 이미지

실습

- **OpenCV 함수**
  - cv2.cvtColor(img, flag)
    - flag에 따라 색상 변경(cv2.COLOR_[type1]2[type2]와 같이 사용)
    - type: GRAY, BGR, HSV, YCrCb, YUV, Lab 등
  - img.shape
    - img의 shape를 반환
  - cv2.waitKey(t)
    - t millisecond만큼 키 입력 대기
    - t가 0이면 key입력이 있을 때 까지 무한 대기
  - cv2.destroyAllWindows()
    - 모든 윈도우 종료

실습

- **OpenCV 함수**
  - cv2.add(src1, src2)
    - src1 + src2
  - cv2.subtract(src1, src2)
    - src1 – src2
  - cv2.multiply(src1, src2)
    - src1 * src2
  - cv2.divide(src1, src2)
    - src1 / src2
  - cv2.pow(src, power)
    - src ** power
  - 위 함수의 계산 결과는 0 ~ 255 사이의 값을 가짐

실습

- **OpenCV 기초 1**

```python
import cv2
import numpy as np

src = np.full( shape: (2, 2), fill_value: 100, dtype=np.uint8)
print('<original>')
print(src)
print('<add>')
print(cv2.add(src, src2: 3))
print(cv2.add(src, src2: 200))
print('<subtract>')
print(cv2.subtract(src, src2: 10))
print(cv2.subtract(src, src2: 150))
print('<multiply>')
print(cv2.multiply(src, src2: 2))
print(cv2.multiply(src, src2: 5))
print('<divide>')
print(cv2.divide(src, 10))
print(cv2.divide(src, 200))
```

```
<original>
[[100 100]
 [100 100]]
<add>
[[103 103]
 [103 103]]
[[255 255]
 [255 255]]
<subtract>
[[90 90]
 [90 90]]
[[0 0]
 [0 0]]
```

```
<multiply>
[[200 200]
 [200 200]]
[[255 255]
 [255 255]]
<divide>
[[10 10]
 [10 10]]
[[0 0]
 [0 0]]
```

<결과>

실습

- **OpenCV 기초 2**
  - float와 uint8의 차이

<float>
black : 0.
white : 1.

<uint8>
black : 0
white : 255

```python
import cv2
import numpy as np

src1 = np.zeros((200, 200))
src2 = np.ones((200, 200))

src3 = np.zeros( shape: (200, 200), dtype=np.uint8)
src4 = np.ones( shape: (200, 200), dtype=np.uint8)
src5 = np.full( shape: (200, 200),  fill_value: 255, dtype=np.uint8)


cv2.imshow( winname: 'src1', src1)
cv2.imshow( winname: 'src2', src2)
cv2.imshow( winname: 'src3', src3)
cv2.imshow( winname: 'src4', src4)
cv2.imshow( winname: 'src5', src5)

cv2.waitKey()
cv2.destroyAllWindows()
```

실습

- **OpenCV 기초 2**

<float>
black : 0.
white : 1.

<uint8>
black : 0
white : 255

```
<float>
src1.shape: (200, 200)
src2.shape: (200, 200)
src1[0, 0]: 0.0, src2[0, 0]: 1.0
<uint8>
src3.shape: (200, 200)
src4.shape: (200, 200)
src5.shape: (200, 200)
src3[0, 0]: 0, src4[0, 0]: 1, src5[0, 0]: 255
```

<결과>

```
cv2.imshow( winname: 'src1', src1)
cv2.imshow( winname: 'src2', src2)
cv2.imshow( winname: 'src3', src3)
cv2.imshow( winname: 'src4', src4)
cv2.imshow( winname: 'src5', src5)
```

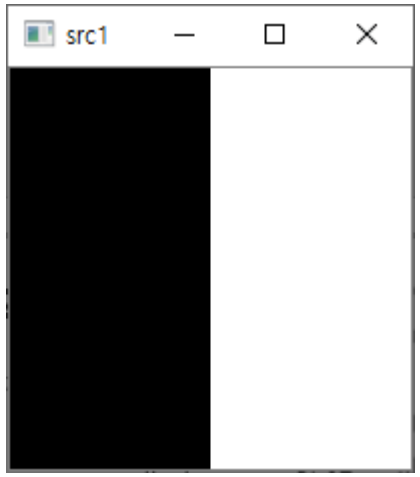
코드 추가

```
print('<float>')
print(f'src1.shape: {src1.shape}')
print(f'src2.shape: {src2.shape}')
print(f'src1[0, 0]: {src1[0, 0]}, src2[0, 0]: {src2[0, 0]}')
print('<uint8>')
print(f'src3.shape: {src3.shape}')
print(f'src4.shape: {src4.shape}')
print(f'src5.shape: {src5.shape}')
print(f'src3[0, 0]: {src3[0, 0]}, src4[0, 0]: {src4[0, 0]}'
      f', src5[0, 0]: {src5[0, 0]}')
```

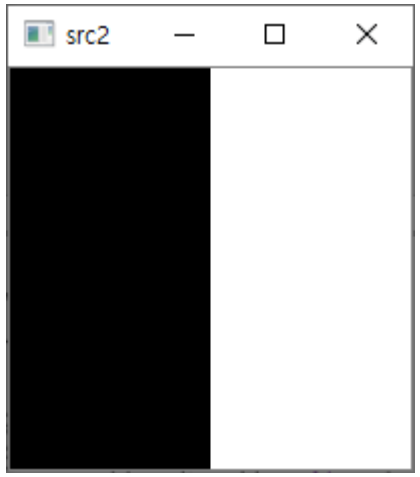
```
cv2.waitKey()
cv2.destroyAllWindows()
```

실습

- **OpenCV 기초 3**
  - 특정 pixel 값 접근하기



<float>

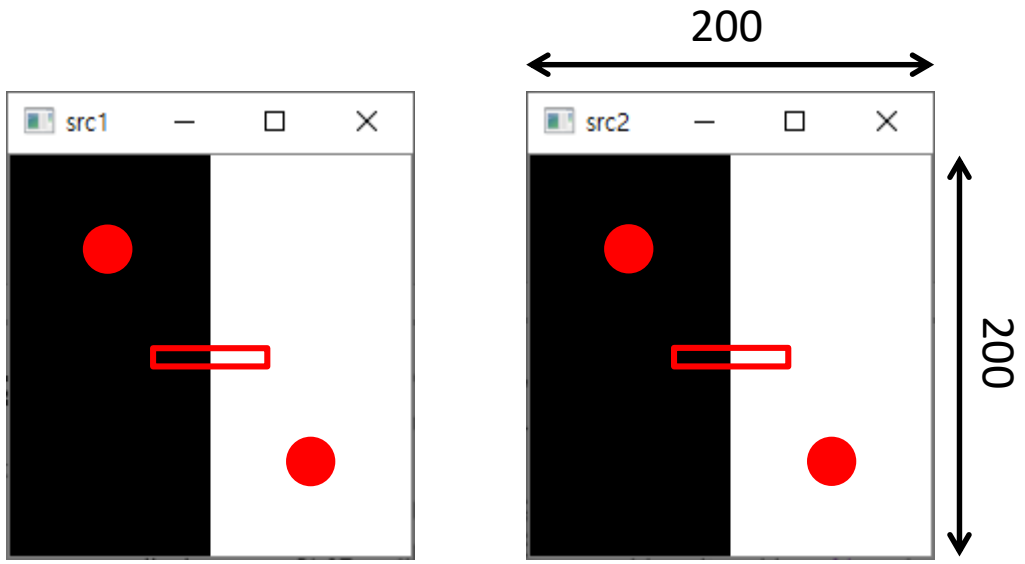


<uint8>

```python
import cv2
import numpy as np

src1 = np.zeros((200, 200))
src2 = np.zeros( shape: (200, 200), dtype=np.uint8)

src1[:, 100:200] = 1.
src2[:, 100:200] = 255

cv2.imshow( winname: 'src1', src1)
cv2.imshow( winname: 'src2', src2)

cv2.waitKey()
cv2.destroyAllWindows()
```

실습

- **OpenCV 기초 3**
  - 특정 pixel 값 접근하기



<float> <uint8>

```
0.0 1.0
0 255
[0. 0. 0. 0. 0. 1. 1. 1. 1. 1.]
[  0   0   0   0   0 255 255 255 255 255]
```

<결과>

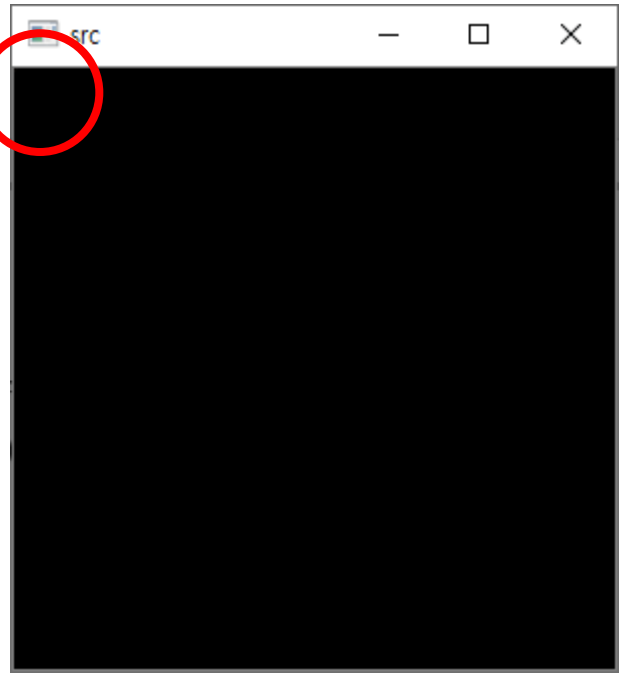
```
cv2.imshow( winname: 'src1', src1)
cv2.imshow( winname: 'src2', src2)
```

코드 추가

```
print(src1[50, 50], src1[150, 150])
print(src2[50, 50], src2[150, 150])

print(src1[100, 95:105])
print(src2[100, 95:105])

cv2.waitKey()
cv2.destroyAllWindows()
```

실습

- **OpenCV 기초 4**
  - 컬러 이미지 기초(행, 열, 채널)



1	4	0	0
7	0	0	0
0	0	0	0
0	0	0	0

Blue channel

2	5	0	0
8	0	0	0
0	0	0	0
0	0	0	0

Green channel

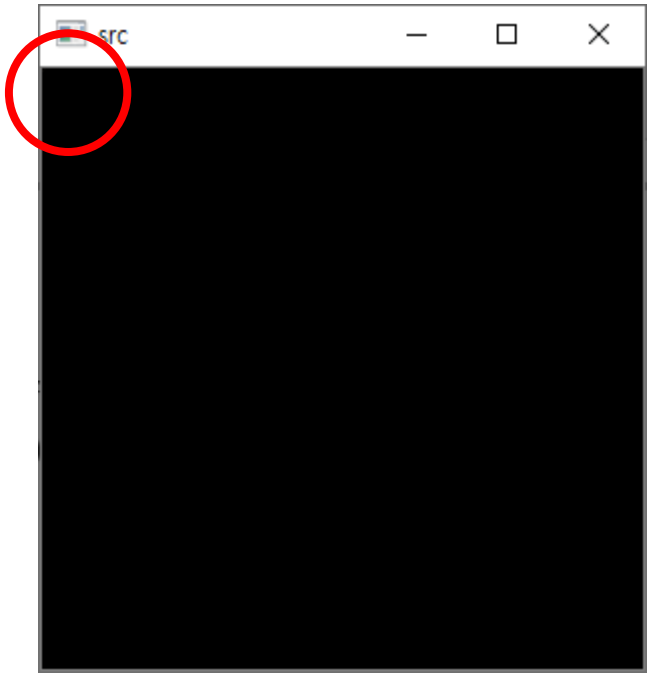
3	6	0	0
9	0	0	0
0	0	0	0
0	0	0	0

Red channel

```python
import cv2
import numpy as np

src = np.zeros( shape: (300, 300, 3), dtype=np.uint8)

src[0, 0] = [1, 2, 3]
src[0, 1] = [4, 5, 6]
src[1, 0] = [7, 8, 9]

print(src.shape)
print(src[0, 0, 0], src[0, 0, 1], src[0, 0, 2])
print(src[0, 0])
print(src[0])
print(src)

cv2.imshow( winname: 'src', src)
cv2.waitKey()
cv2.destroyAllWindows()
```

실습

- **OpenCV 기초 4**
  - 컬러 이미지 기초(행, 열, 채널)



1	4	0	0
7	0	0	0
0	0	0	0
0	0	0	0

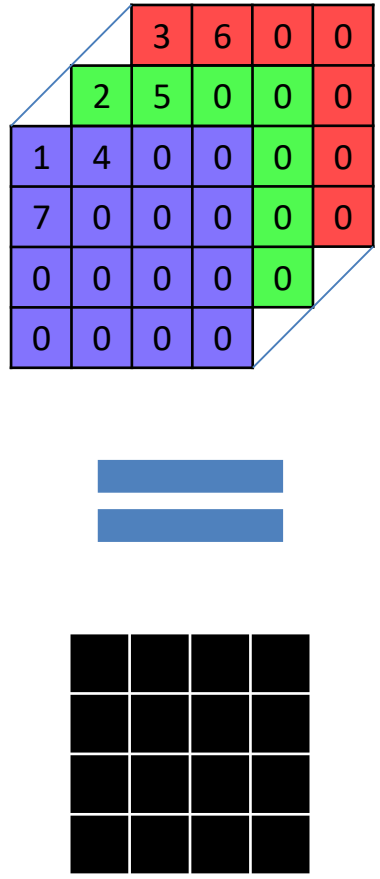
Blue channel

2	5	0	0
8	0	0	0
0	0	0	0
0	0	0	0

Green channel

3	6	0	0
9	0	0	0
0	0	0	0
0	0	0	0

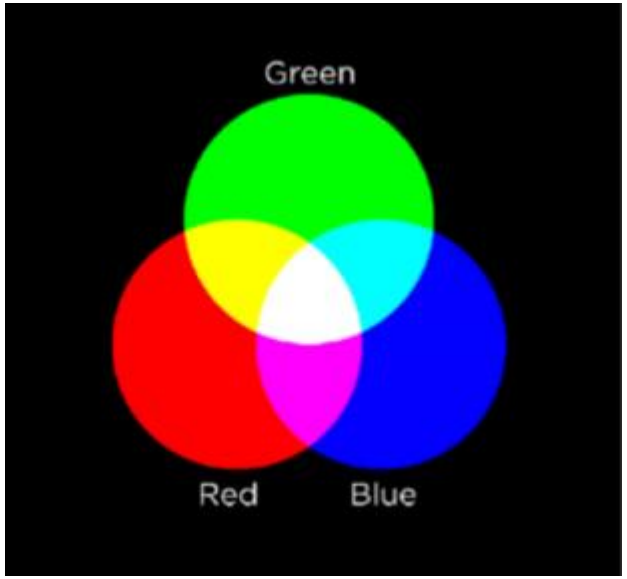
Red channel



우리한테 보이는 이미지

실습

- **OpenCV 기초 4**
  - 컬러 이미지 기초(행, 열, 채널)
  - 만약
    - src[0, 2]=[255, 0, 0]
    - src[1, 1]=[255, 255, 255]



빛의 3원색

1	4	255	0
7	255	0	0
0	0	0	0
0	0	0	0

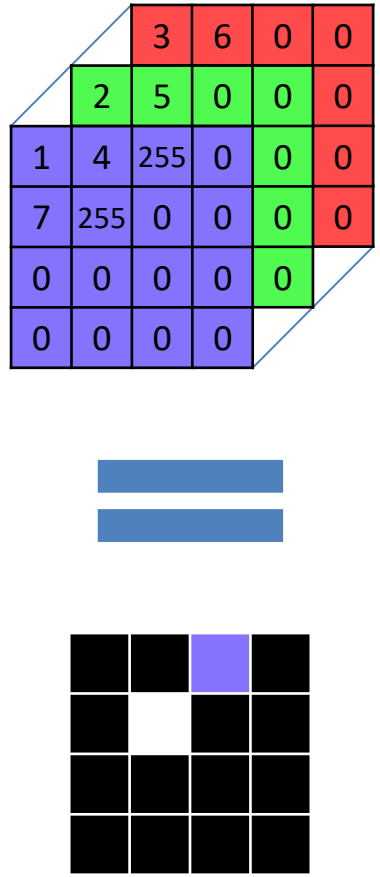
Blue channel

2	5	0	0
8	255	0	0
0	0	0	0
0	0	0	0

Green channel

3	6	0	0
9	255	0	0
0	0	0	0
0	0	0	0

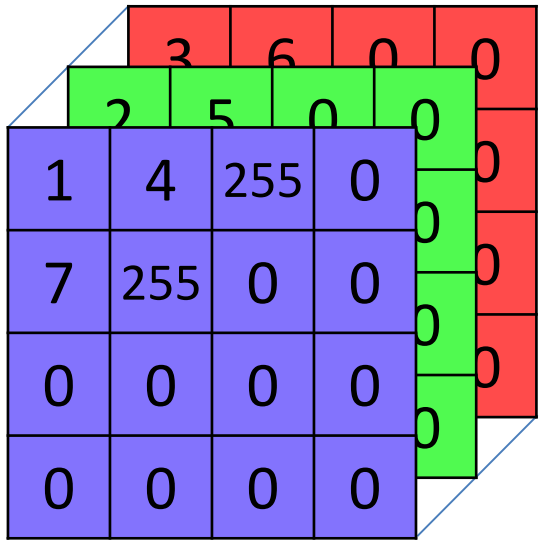
Red channel

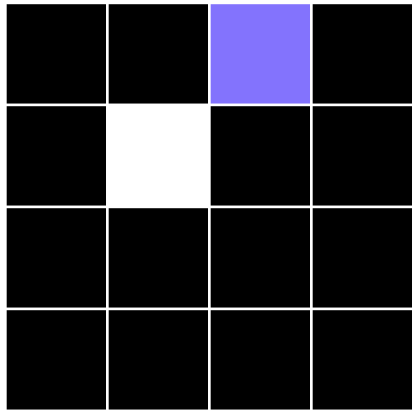


우리한테 보이는 이미지

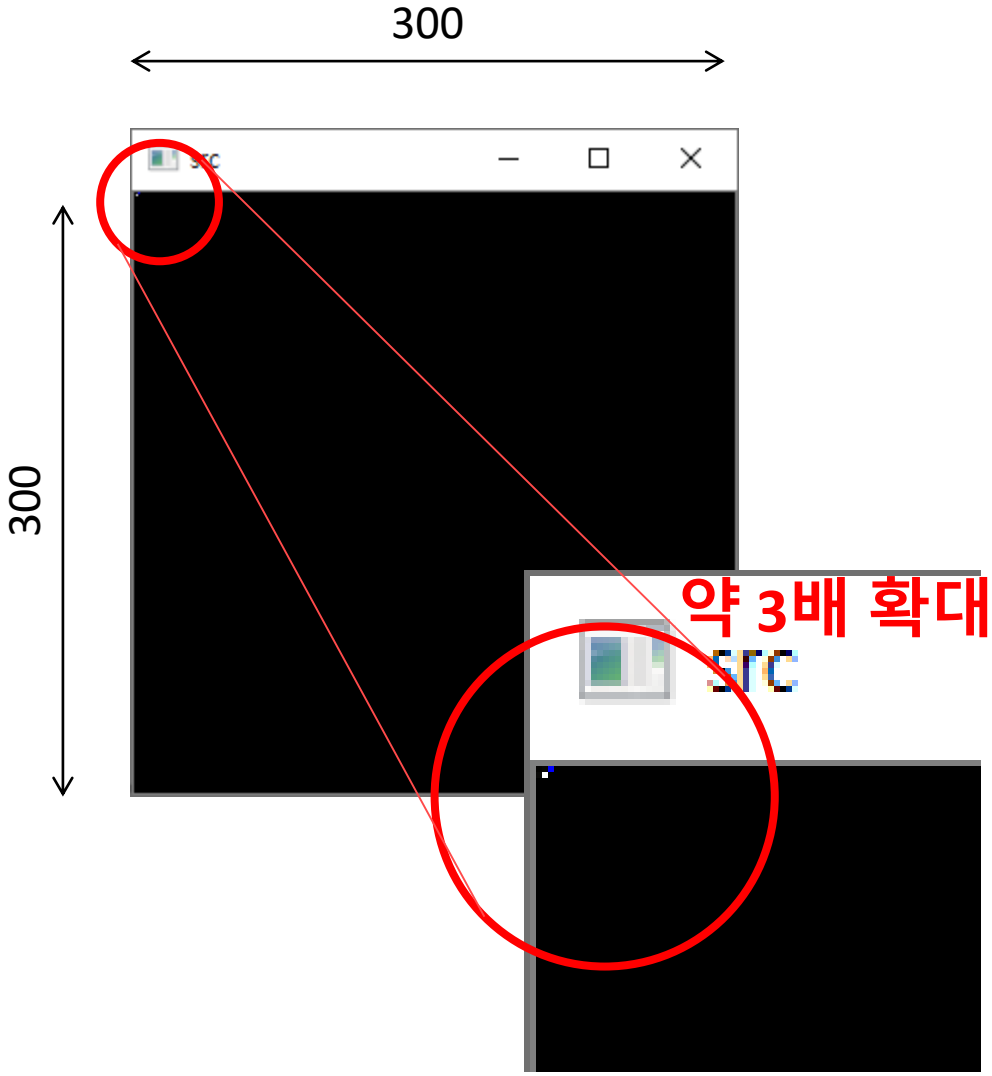
실습

- **OpenCV 기초 4**
  - 1 pixel의 크기





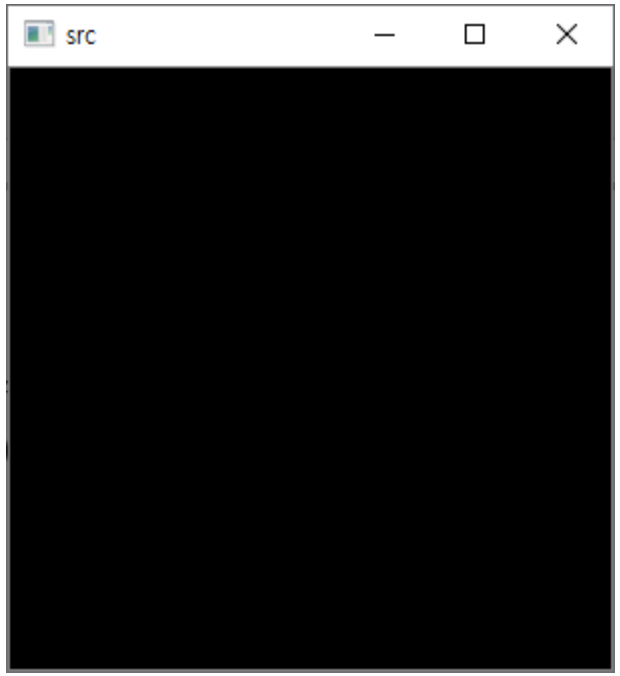
우리한테 보이는 이미지



실제 우리가 보는 이미지

실습

- **OpenCV 기초 4**
  - 컬러 이미지 기초(행, 열, 채널)



<결과>

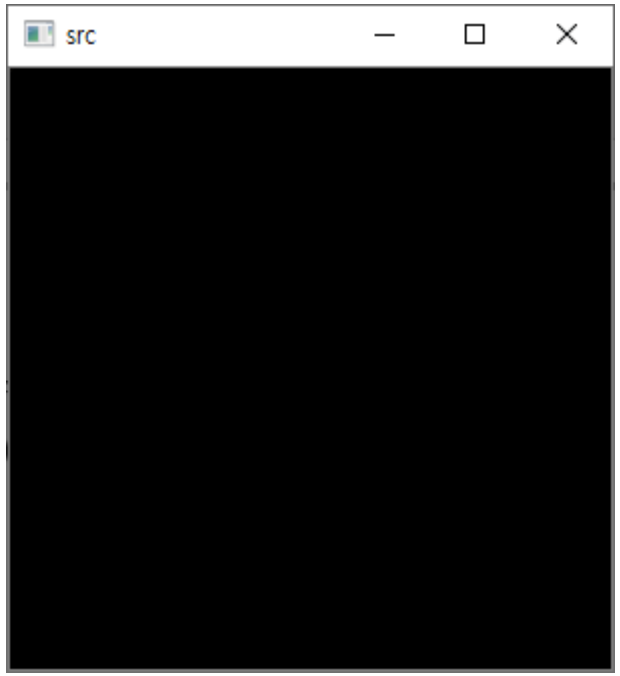
```
(300, 300, 3)
1 2 3
[1 2 3]
[[1 2 3]
 [4 5 6]
 [0 0 0]
 [0 0 0]
```

...생략...

```python
import cv2
import numpy as np

src = np.zeros( shape: (300, 300, 3), dtype=np.uint8)

src[0, 0] = [1, 2, 3]
src[0, 1] = [4, 5, 6]
src[1, 0] = [7, 8, 9]

print(src.shape)
print(src[0, 0, 0], src[0, 0, 1], src[0, 0, 2])
print(src[0, 0])
print(src[0])
print(src)

cv2.imshow( winname: 'src', src)
cv2.waitKey()
cv2.destroyAllWindows()
```

실습

- **OpenCV 기초 4**
  - 컬러 이미지 기초(행, 열, 채널)



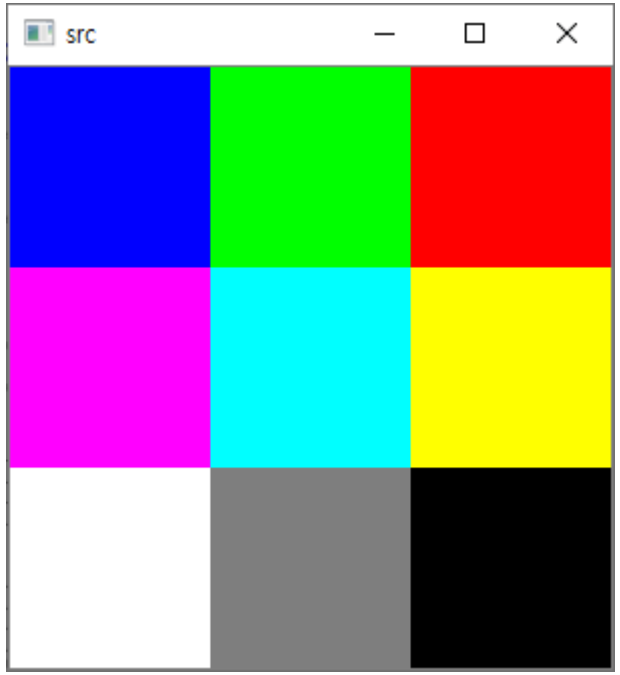
<결과>

```
...생략...
 [0 0 0]
 [0 0 0]
 [0 0 0]]
[[[1 2 3]
  [4 5 6]
  [0 0 0]
  ...
  [0 0 0]
  [0 0 0]
  [0 0 0]]

 [[7 8 9]
  [0 0 0]
  [0 0 0]
  ...
...생략...
```

```python
import cv2
import numpy as np

src = np.zeros( shape: (300, 300, 3), dtype=np.uint8)

src[0, 0] = [1, 2, 3]
src[0, 1] = [4, 5, 6]
src[1, 0] = [7, 8, 9]

print(src.shape)
print(src[0, 0, 0], src[0, 0, 1], src[0, 0, 2])
print(src[0, 0])
print(src[0])
print(src)

cv2.imshow( winname: 'src', src)
cv2.waitKey()
cv2.destroyAllWindows()
```

실습

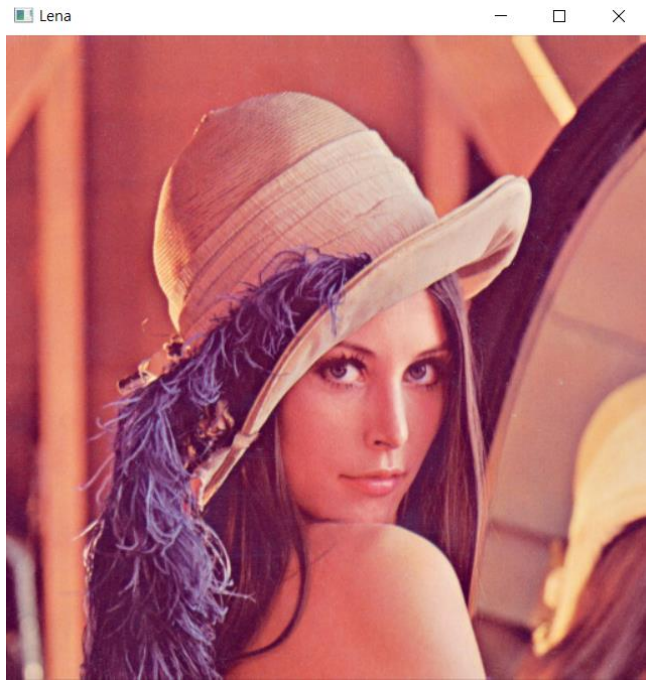
- **OpenCV 기초 5**
  - 컬러 이미지(행, 열, 채널)




```python
import cv2
import numpy as np

src = np.zeros( shape: (300, 300, 3), dtype=np.uint8)

# b = 255   g, r = 0
src[0:100, 0:100, 0] = 255
# g = 255   b, r = 0
src[0:100, 100:200, 1] = 255
# r = 255   b, g = 0
src[0:100, 200:300, 2] = 255

# b + r
src[100:200, 0:100, 0] = 255
src[100:200, 0:100, 2] = 255

# b + g
src[100:200, 100:200, 0] = 255
src[100:200, 100:200, 1] = 255

# g + r
src[100:200, 200:300, 1] = 255
src[100:200, 200:300, 2] = 255

# b + g + r
src[200:, :100, 0] = 255
src[200:, :100, 1] = 255
src[200:300, :100, 2] = 255

# b/2 + g/2 + r/2
src[200:, 100:200, 0] = 128
src[200:, 100:200, 1] = 128
src[200:300, 100:200, 2] = 128

cv2.imshow( winname: 'src', src)
cv2.waitKey()
cv2.destroyAllWindows()
```

실습

- **OpenCV 기초 6**
  - OpenCV의 imread 함수는 이미지를 불러올 때 사용
    이때, 불러온 이미지는 numpy이고, uint8 (0 ~ 255)의 데이터 타입을 가짐



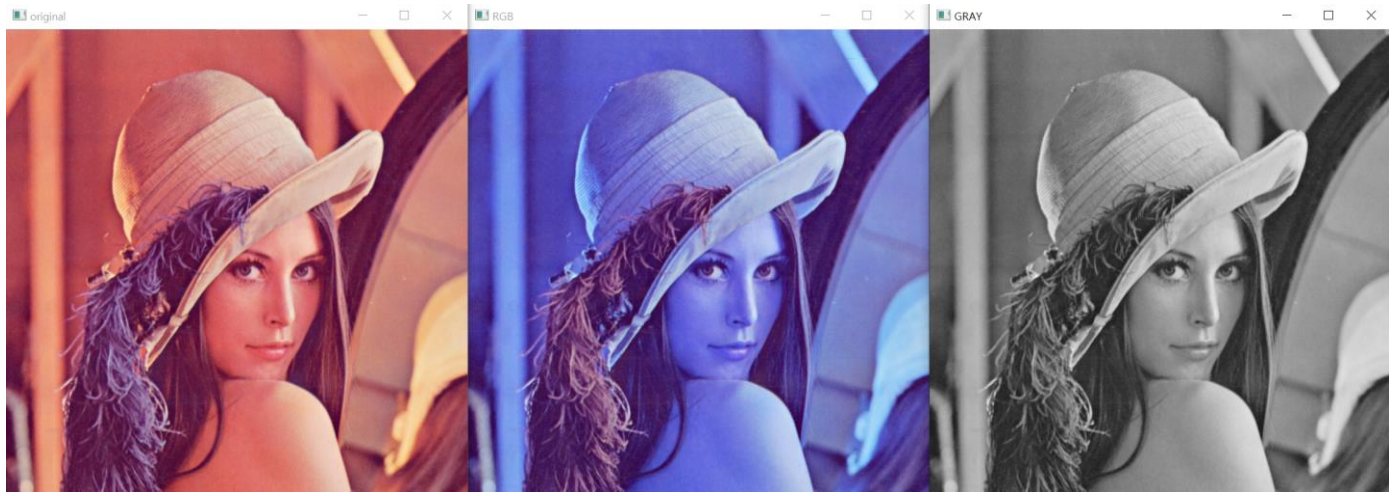

```python
import cv2

src = cv2.imread('./Lena.png')

print(f'type(src): {type(src)}')
print(f'src.dtype: {src.dtype}')
print(f'src.shape: {src.shape}')

cv2.imshow( winname: 'Lena', src)
cv2.waitKey()
cv2.destroyAllWindows()
```

```
type(src): <class 'numpy.ndarray'>
src.dtype: uint8
src.shape: (512, 512, 3)
```

<결과>

실습

- **OpenCV 기초 7**
  - OpenCV의 imread 함수는 이미지를 BGR로 읽음



Original RGB GRAY

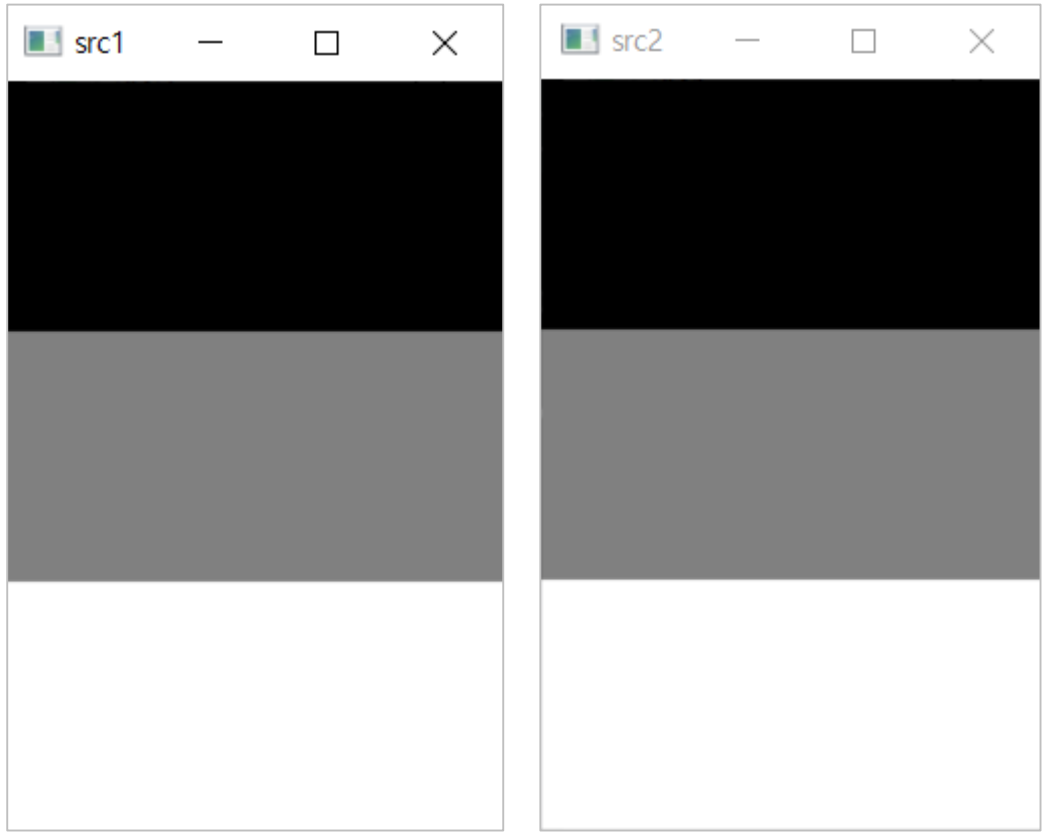
```
[BGR]: [128 138 225]
[RGB]: [225 138 128]
[GRAY]: 163
```

<결과>

```python
import cv2

src = cv2.imread('./Lena.png')
rgb = cv2.cvtColor(src, cv2.COLOR_BGR2RGB)
gray = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY)

cv2.imshow( winname: 'original', src)
cv2.imshow( winname: 'RGB', rgb)
cv2.imshow( winname: 'GRAY', gray)

print(f'[BGR]: {src[0, 0]}')
print(f'[RGB]: {rgb[0, 0]}')
print(f'[GRAY]: {gray[0, 0]}')

cv2.waitKey()
cv2.destroyAllWindows()
```

실습

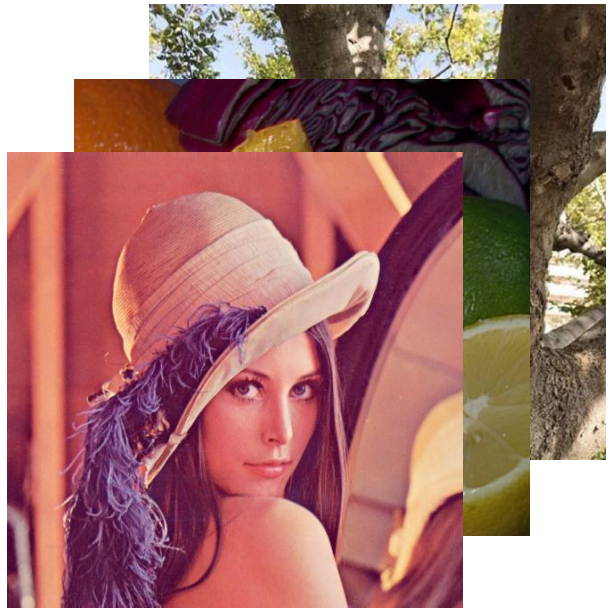
- **OpenCV 기초 8**
  - 아래 사진처럼 코드 완성하기




```python
import cv2
import numpy as np

src1 = np.zeros((300, 200))
src2 = np.zeros( shape: (300, 200), dtype=np.uint8)

src1[:100] = [redacted]
src1[100:200] = [redacted]
src1[200:] = [redacted]

src2[:100] = [redacted]
src2[100:200] = [redacted]
src2[200:] = [redacted]

cv2.imshow( winname: 'src1', src1)
cv2.imshow( winname: 'src2', src2)

cv2.waitKey()
cv2.destroyAllWindows()
```

실습

- **OpenCV 기초 9**
  - Tensorflow: 머신 러닝 라이브러리와 python을 결합한 오픈 소스 딥러닝 프레임워크
    - (batch, height, width, channel) 형태로 이미지 사용

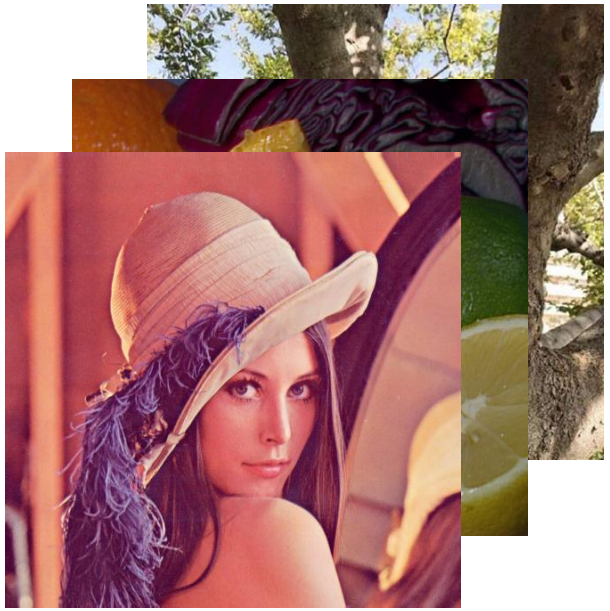


```
tf 1::src1.shape: (512, 512, 3), src2.shape: (512, 512, 3), src3.shape:(512, 512, 3)
tf 2::src1.shape: (1, 512, 512, 3), src2.shape: (1, 512, 512, 3), src3.shape:(1, 512, 512, 3)
tf 3::batch.shape: (3, 512, 512, 3)
```

```python
import cv2
import numpy as np

### tensorflow
src1 = cv2.imread('./Lena.png')
src2 = cv2.imread('./cat.png')
src3 = cv2.imread('./fruits.png')
print(f'tf 1::src1.shape: {src1.shape}, src2.shape: {src2.shape}, src3.shape:{src3.shape}')

# batch dimension 생성 (batch, height, width, channel)
src1 = np.expand_dims(src1, axis=0)
src2 = np.expand_dims(src2, axis=0)
src3 = np.expand_dims(src3, axis=0)
print(f'tf 2::src1.shape: {src1.shape}, src2.shape: {src2.shape}, src3.shape:{src3.shape}')

# batch dim으로 연결 (batch, height, width, channel)
tf_batch = np.concatenate( arrays: [src1, src2, src3], axis=0)
print(f'tf 3::batch.shape: {tf_batch.shape}\n')

cv2.imshow( winname: 'Lena', tf_batch[0])
cv2.imshow( winname: 'Cat', tf_batch[1])
cv2.imshow( winname: 'Fruits', tf_batch[2])

cv2.waitKey()
cv2.destroyAllWindows()
```

실습

- **OpenCV 기초 9**
  - Pytorch: 머신 러닝 라이브러리와 python을 결합한 오픈 소스 딥러닝 프레임워크
    - (batch, channel, height, width) 형태로 이미지 사용



```
torch 1::src1.shape: (512, 512, 3), src2.shape: (512, 512, 3), src3.shape:(512, 512, 3)
torch 2::src1.shape: (1, 512, 512, 3), src2.shape: (1, 512, 512, 3), src3.shape:(1, 512, 512, 3)
torch 3::src1.shape: (1, 3, 512, 512), src2.shape: (1, 3, 512, 512), src3.shape:(1, 3, 512, 512)
torch 4::src1.shape: (1, 3, 512, 512), src2.shape: (1, 3, 512, 512), src3.shape:(1, 3, 512, 512)
batch.shape: (3, 3, 512, 512)
```

```python
### pytorch
src1 = cv2.imread('./Lena.png')
src2 = cv2.imread('./cat.png')
src3 = cv2.imread('./fruits.png')
print(f'torch 1::src1.shape: {src1.shape}, src2.shape: {src2.shape}, src3.shape:{src3.shape}')

# batch dimension 생성 (batch, height, width, channel)
src1 = np.expand_dims(src1, axis=0)
src2 = np.expand_dims(src2, axis=0)
src3 = np.expand_dims(src3, axis=0)
print(f'torch 2::src1.shape: {src1.shape}, src2.shape: {src2.shape}, src3.shape:{src3.shape}')

# height와 channel 전환 (batch, channel, width, height)
src1 = np.swapaxes(src1, axis1: 1, axis2: 3)
src2 = np.swapaxes(src2, axis1: 1, axis2: 3)
src3 = np.swapaxes(src3, axis1: 1, axis2: 3)
print(f'torch 3::src1.shape: {src1.shape}, src2.shape: {src2.shape}, src3.shape:{src3.shape}')

# height와 width 전환 (batch, channel, height, width)
src1 = np.swapaxes(src1, axis1: 2, axis2: 3)
src2 = np.swapaxes(src2, axis1: 2, axis2: 3)
src3 = np.swapaxes(src3, axis1: 2, axis2: 3)
print(f'torch 4::src1.shape: {src1.shape}, src2.shape: {src2.shape}, src3.shape:{src3.shape}')

# batch dim으로 연결 (batch, channel, height, width)
batch = np.concatenate( arrays: [src1, src2, src3], axis=0)
print(f'batch.shape: {batch.shape}')
```

실습

- **OpenCV 기초 10**
  - 컬러 이미지 흑백 전환
  - 직접 계산시 반올림, uint8 전환 진행


```python
import cv2
import numpy as np

if __name__ == '__main__':
    src = cv2.imread('./Lena.png')
    dst1 = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY)
    # dst2 = (B+G+R)/3
    dst2 = (src[:, :, 0] * 1/3 +
            src[:, :, 1] * 1/3 +
            src[:, :, 2] * 1/3)
    # dst3 = 0.0721*B + 0.7154*G + 0.2125*R
    dst3 = (src[:, :, 0] * 0.0721 +
            src[:, :, 1] * 0.7154 +
            src[:, :, 2] * 0.2125)

    dst2 = np.round(dst2).astype(np.uint8)
    dst3 = np.round(dst3).astype(np.uint8)

    cv2.imshow( winname: 'Original', src)
    cv2.imshow( winname: 'Gray (cvtColor)', dst1)
    cv2.imshow( winname: 'Gray (1/3)', dst2)
    cv2.imshow( winname: 'Gray (formula)', dst3)

    cv2.waitKey()
    cv2.destroyAllWindows()
```

실습

- **OpenCV 기초 11**
  - 동영상 load 및 save
    - cv2.VideoWriter_fourcc(): 인코딩 포맷 설정
    - cv2.VideoWriter(path, fourcc, fps, size, isColor)
      - path: 저장 경로
      - fourcc: 저장 포멧
      - fps: 초당 프레임수
      - size: (w, h) 크기
      - isColor: 영상의 컬러 유무
    - cv2.namedWindow(win_name, flags)
      - 영상의 Window 크기를 맞춰주는 함수
      - win_name: imshow와 동일한 str
      - flags: 자동, 수동 선택

```python
import cv2

video_file = './jin.avi' # 동영상 경로
cap = cv2.VideoCapture(video_file) # 동영상 캡처 객체 생성

if cap.isOpened(): # 객체 초기화 정상
    fps = 30.0  # FPS, 초당 프레임 수
    fourcc = cv2.VideoWriter_fourcc(*'DIVX')  # 인코딩 포맷 문자
    h, w = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT)), int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))

    # 영상 저장을 위핸 객체 생성
    out = cv2.VideoWriter('./record_color.avi', fourcc, fps, (w, h), isColor=True)
    while True:
        ret, frame = cap.read() # 다음 프레임 읽기
        if ret: # 프레임 읽기 정상
            cv2.namedWindow(video_file, flags=cv2.WINDOW_NORMAL)
            cv2.imshow(video_file, frame) # win_name, img
            out.write(frame) # frame 저장
            if cv2.waitKey(int(1000/fps)) != -1: # 지연시간 = 1000/fps
                break
        else: # 프레임 읽기 비정상
            break
    out.release() # 저장 객체 소멸
else: # 객체 초기화 비정상
    print("can't open video")

cap.release() # 캡쳐 객체 소멸
cv2.destroyAllWindows()
```

과제

- **tqdm**
  - 터미널에 ‘pip install tqdm’ 입력하여 라이브러리 설치

```
(.venv) PS C:\Users\DONGHUN'S\PycharmProjects\ImageProcessing> pip install tqdm
```

  - tqdm 라이브러리는 현재 프로그램의 진행상황을 보여줌

```
from tqdm import tqdm
for i in tqdm(range(1000000)):
    print(i, end=' ')
```

```
 13%|          | 129171/1000000 [00:00<00:04, 177677.57it/s]
```

과제

- **myVideo_bgr2gray**
  - 컬러 비디오를 흑백 비디오로 변환
    - cv2.cvtColor() 사용 금지
    - 30p의 dst3 참고
  - myVideo_bgr2gray.py 파일의 빈칸을 완성시켜 제출
    - 입력: (h, w, 3)인 BGR 영상
    - 출력: (h, w)인 grayscale 영상
  - 과제 보고서를 함께 제출
  - 제공된 동영상을 흑백으로 처리한 결과 영상 포함
    - 파일명: record_gray.avi
    - 영상을 흑백으로 전환하는데 약 10~20분 정도 소요
  - .py 파일 이름 수정하지 않고 그대로 제출

과제

- **보고서**
  - 내용
    - 학과, 학번, 이름
    - 구현 코드: 구현한 코드에 대한 간단한 설명
    - 이미지: 동영상 일부분 1 ~ 2장 캡처
    - 분석: 반복문을 사용한 것과 사용하지 않은 것에 대한 시간, 연산량에 대한 분석
    - 느낀 점: 구현 결과를 보고 느낀 점, 혹은 어려운 점 등
    - 과제 난이도: 개인적으로 느낀 난이도 및 이유(과제가 쉽다, 어렵다 등)
  - .pdf 파일로 제출(이외의 파일 형식일 경우 감점)
  - 보고서 명
    - [IP]20xxxxxxx_이름_x주차_과제.pdf

과제

- **과제 요약**
  - myVideo_bgr2gray.py 완성
  - 채점 기준
    - 구현을 못하거나 잘못 구현한 경우
    - 보고서 내용이 빠진 경우
    - 다른 사람의 코드 copy 적발시 보여준 사람, copy한 사람 둘 다 0점
    - 내장 함수 사용시 감점(내장 함수를 사용해도 된다고 한 것, 제공한 것 제외)
  - 제출 파일
    - 아래의 파일을 압축해서 [IP]20XXXXXXX_이름_X주차_과제.zip 으로 제출
      - .py 파일
      - .pdf 보고서 파일
      - .avi 영상 파일
  - 제출 기한
    - 2024년 3월 28일 23시 59분까지

Q & A