

중간고사 대체 프로그래밍

# 제출 방식

- 번호.py 로 제출 (1.py, 2.py,~)
- 04/26 자정까지
- 이미지들은 과제에 같이 첨부되어 있음.
  - 미 첨부시 본인이 임의로 첨부하여 사용(+ 제출시 같이 제출)
- 코드 공유 금지
  - 적발시 F

# 1. Nearest resize 함수 완성하기

- 다음 함수를 완성하여, opencv의 resize 함수와 똑같은 효과를 낼 수 있도록 하여라.

```
import numpy as np, cv2
```

```
def my_resize(image, dsize, fx = 0.0, fy = 0.0):
```

```
    |  
    return image
```

```
image = cv2.imread("images/write_test.jpg", cv2.IMREAD_COLOR)    # 원본 영상 읽기
```

```
if image is None: raise Exception("영상 파일 읽기 오류 ")
```

```
img2 = cv2.resize(image, (400,300), fx=0.0, fy = 0.0, interpolation = cv2.INTER_NEAREST)
```

```
img3 = my_resize(image, (400,300), fx = 0.0, fy = 0.0)
```

```
cv2.imshow("openceresize", img2)
```

```
cv2.imshow("userresize", img3)
```

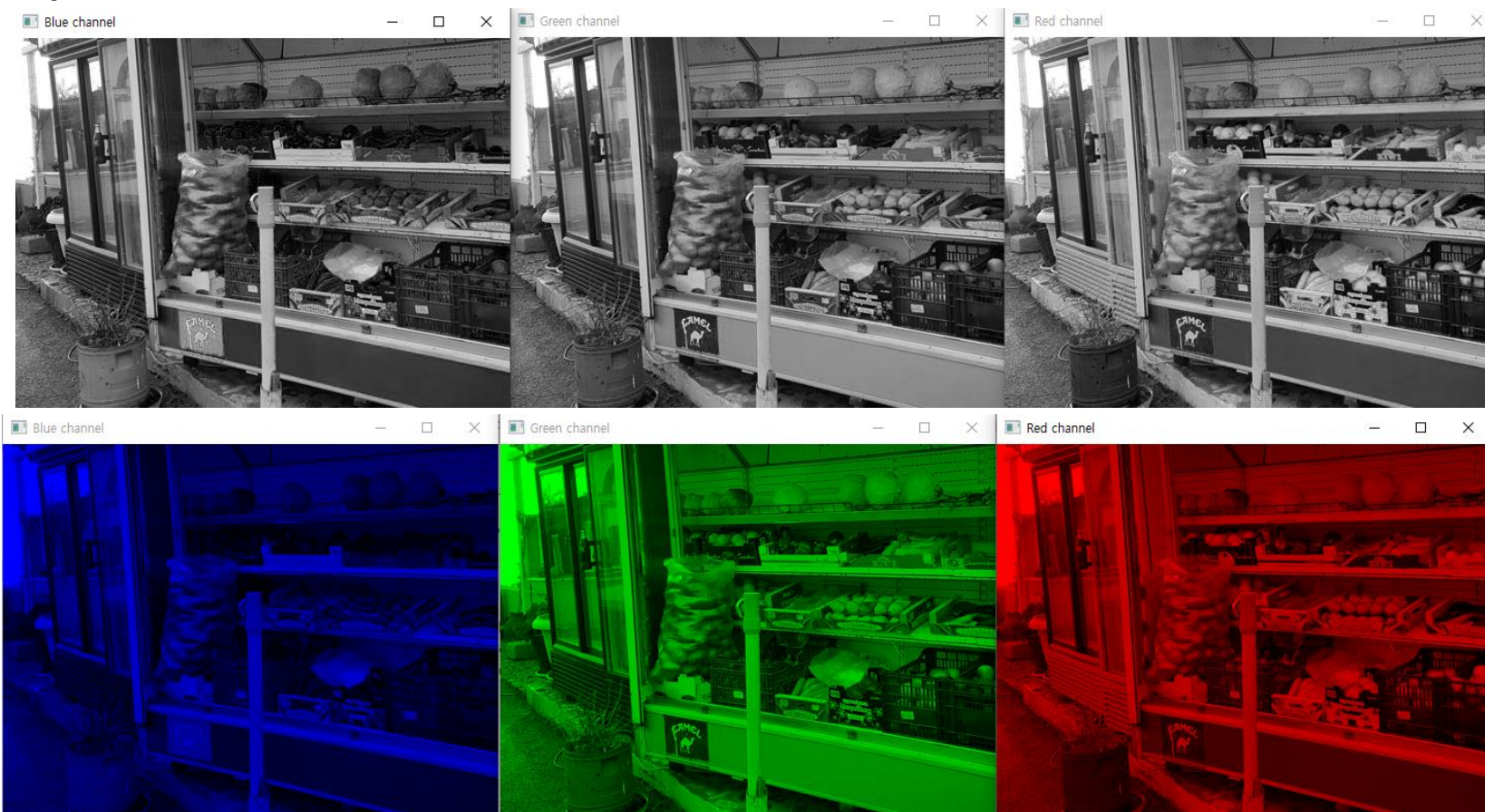
```
cv2.waitKey()
```

Def 함수를 정의하여라.

- img2와 img3에 같은 파라미터가 들어갔을 경우 같은 결과가 나오도록 해야 한다.
- dsize는 전체 해상도 값으로 resize하는 것이고, (fx,fy)는 비율 대비로 resize하는 것이다.
- (fx,fy) 비율을 기반으로 resize할 때는 dsize 값이 꼭 (0,0)인지 체크하여라.
- Interpolation 은 생략한다.(무조건 nearest)
- 실제 테스트 할 때는 파라미터의 값을 바꿔 가며 실행해볼 예정

## 2. R, G, B 채널 별 밝기 표현하기

- 기존의 사진에서 R, G, B 채널의 밝기 정도를 Display하고자 한다. 아래와 같이 Grayscale 형태가 아닌 R, G, B 형태로 출력하여라.



### 3. 트랙바를 이용하여 이미지 블렌딩(blending)

- 왼쪽 이미지(image 1)과 오른쪽 이미지(image 2)를 두 개의 트랙바를 추가해서, 각 영상의 반영 비율을 조절하여 합성 하는 프로그램 제작하여라. (가운데 이미지가 영상이 합성된 이미지 이다.)





## 4. 워터마크

- 숫자 두개를 입력받아서, 로고를 개수만큼 합성하는 워터마크를 제작하여라.

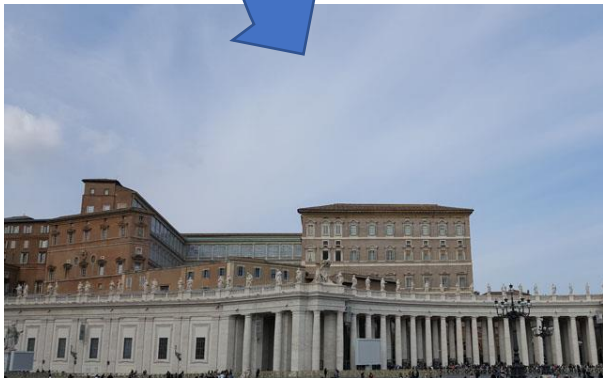
가로에 들어갈 개수를 입력하시오. 6

세로에 들어갈 개수를 입력하시오. 4

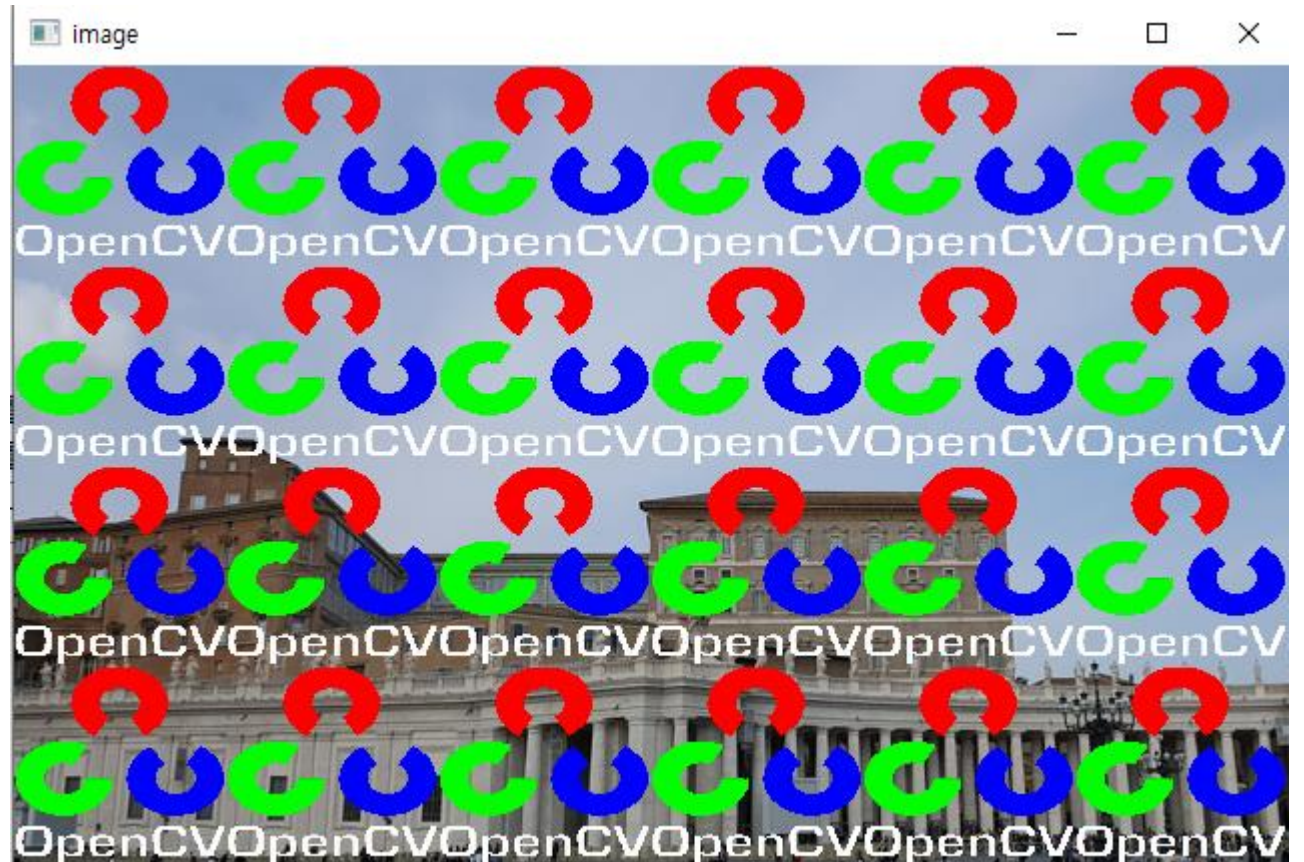
Input을 통한 입력



합성 로고



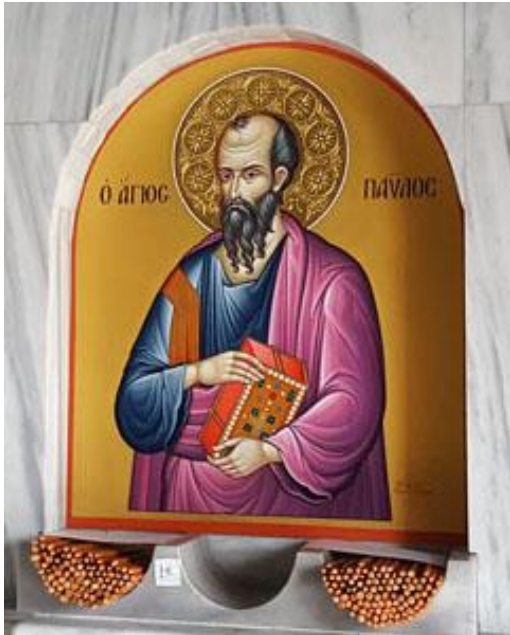
배경 이미지



결과 이미지

## 5.사각형 그리기

- 이미지 한 장을 Grayscale로 불러온다. 마우스를 두 번 클릭하여, 두 점을 양 끝으로 하는 사각형을 그리고, 그 사각형을 마스크로 하여, 이미지 내부의 픽셀 값에 대한 히스토그램을 그려라.(다음 슬라이드에 설명) - 히스토그램은 총 32개의 단계(계급)으로 출력한다.



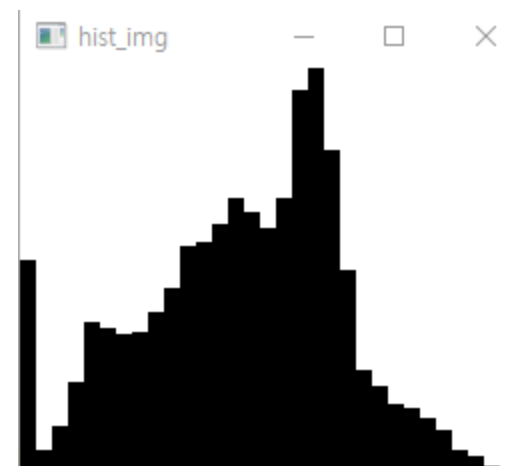
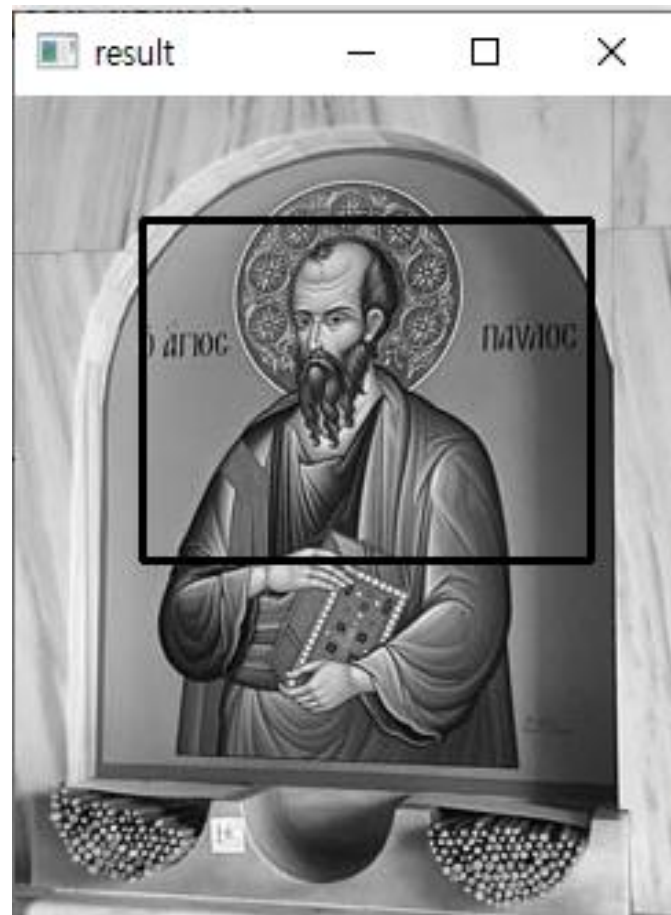
색상 이미지를 흑백으로 출력



첫번째 마우스 클릭



두번째 마우스 클릭



사각형을 그리고, 그 부분에 대한  
명암도를 히스토그램으로 그려라.