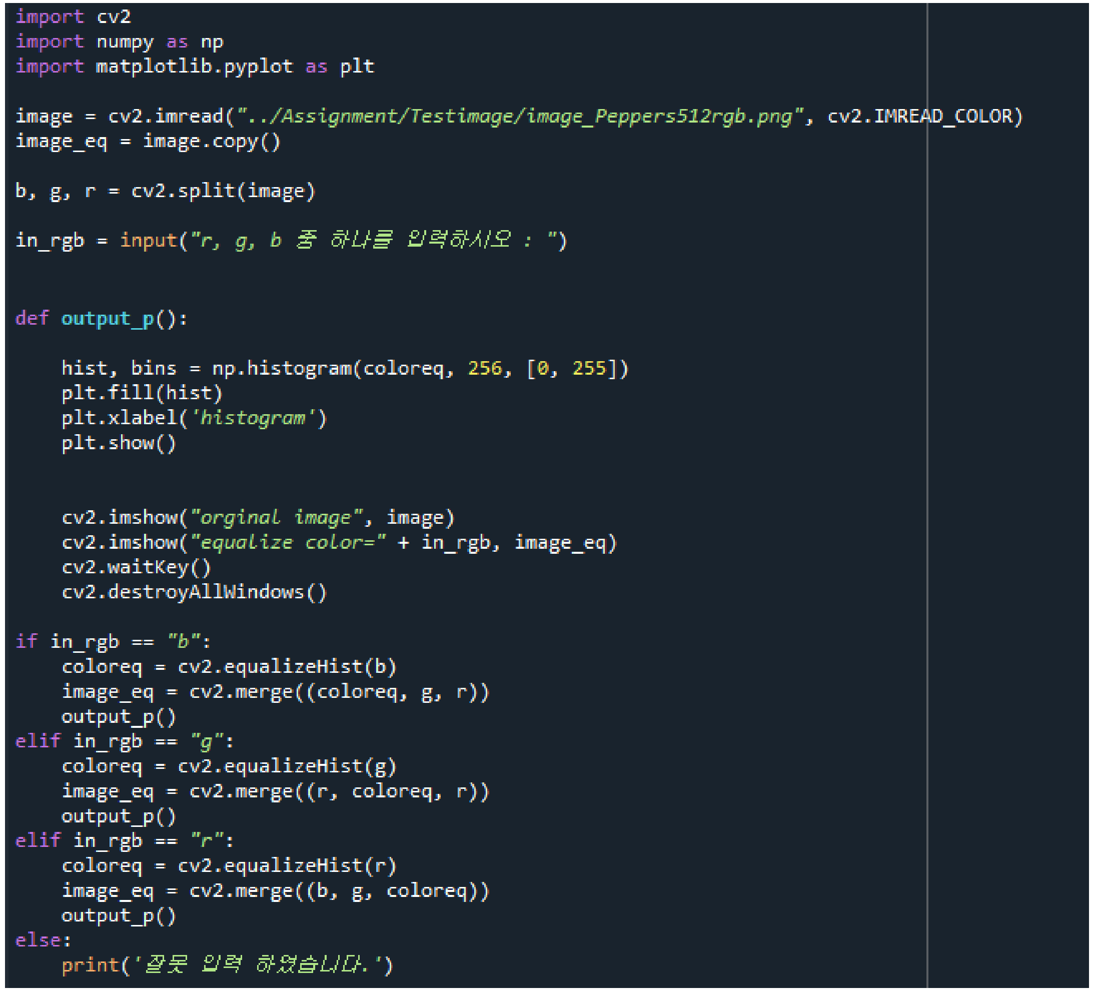
산업 컴퓨터비젼 실제 프로그래밍 과제 #1

산업인공지능학과

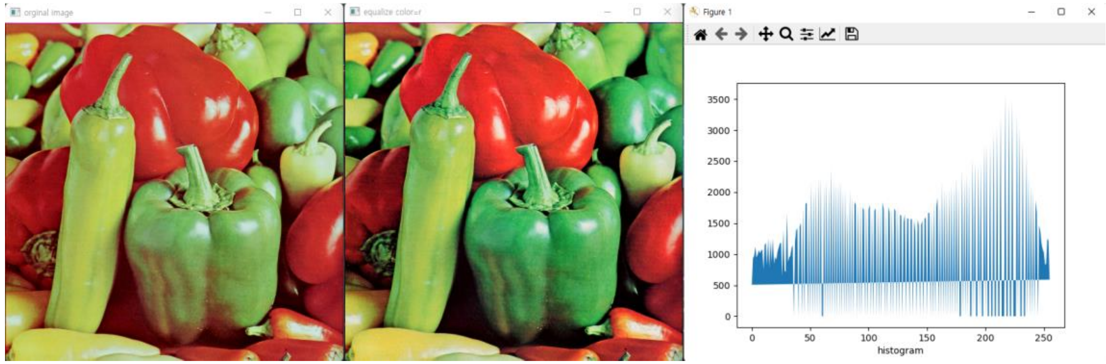
2020254018 강윤구

1) 히스토그램 평탄화

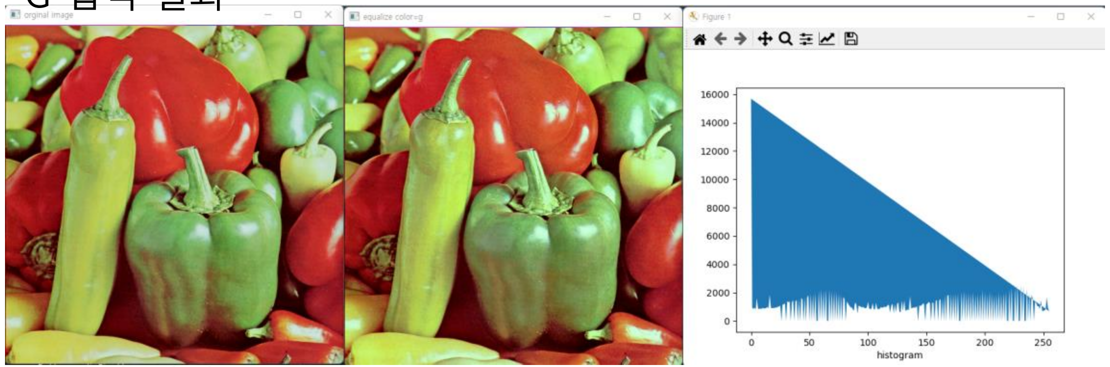


```python
import cv2
import numpy as np
import matplotlib.pyplot as plt

image = cv2.imread("../Assignment/Testimage/image_Peppers512rgb.png", cv2.IMREAD_COLOR)
image_eq = image.copy()

b, g, r = cv2.split(image)

in_rgb = input("r, g, b 중 하나를 입력하시오 : ")


def output_p():

    hist, bins = np.histogram(coloreq, 256, [0, 255])
    plt.fill(hist)
    plt.xlabel('histogram')
    plt.show()


    cv2.imshow("orginal image", image)
    cv2.imshow("equalize color=" + in_rgb, image_eq)
    cv2.waitKey()
    cv2.destroyAllWindows()

if in_rgb == "b":
    coloreq = cv2.equalizeHist(b)
    image_eq = cv2.merge((coloreq, g, r))
    output_p()
elif in_rgb == "g":
    coloreq = cv2.equalizeHist(g)
    image_eq = cv2.merge((r, coloreq, r))
    output_p()
elif in_rgb == "r":
    coloreq = cv2.equalizeHist(r)
    image_eq = cv2.merge((b, g, coloreq))
    output_p()
else:
    print('잘못 입력 하였습니다.')
```

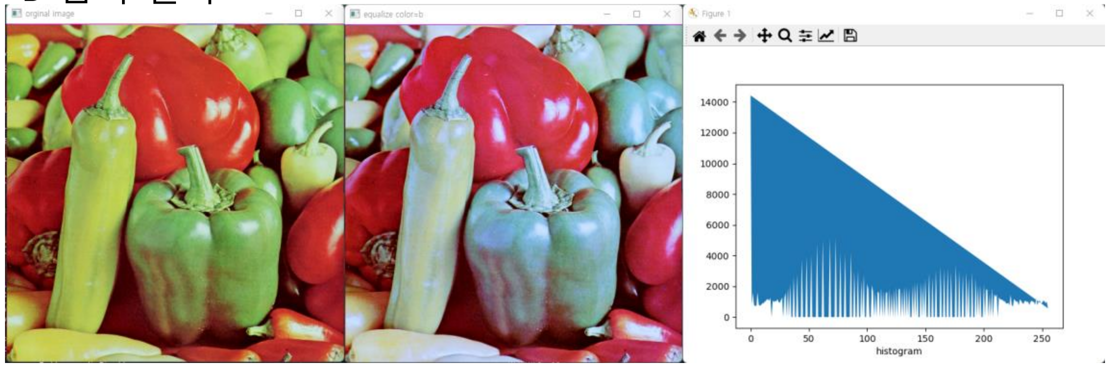
R 입력 결과



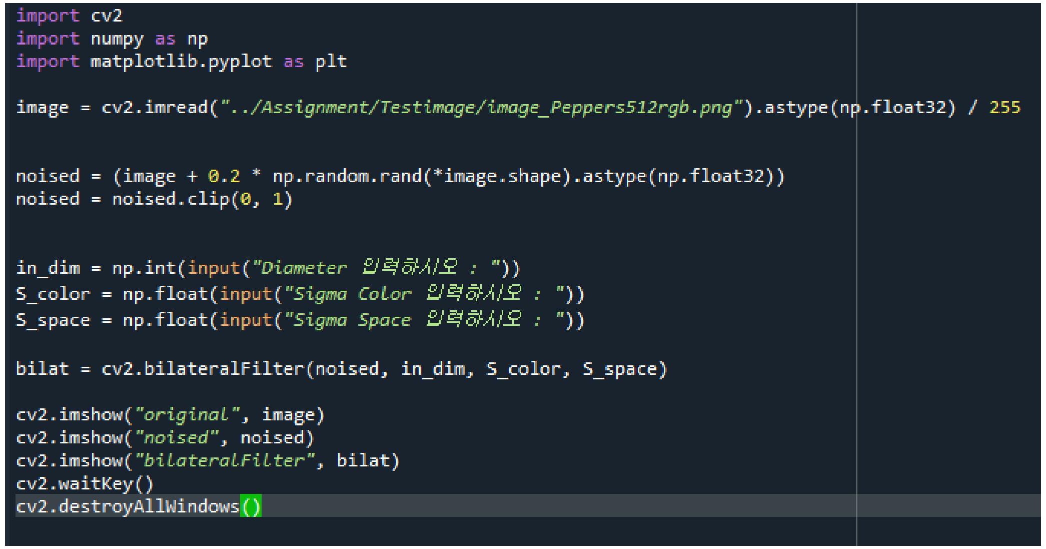
G 입력 결과



B 입력 결과

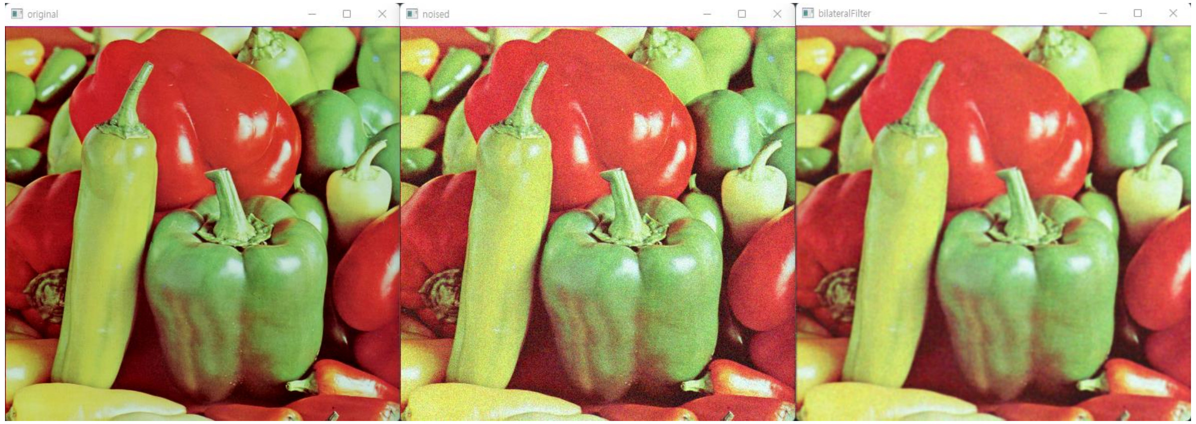


2) 공간도메인 필터링

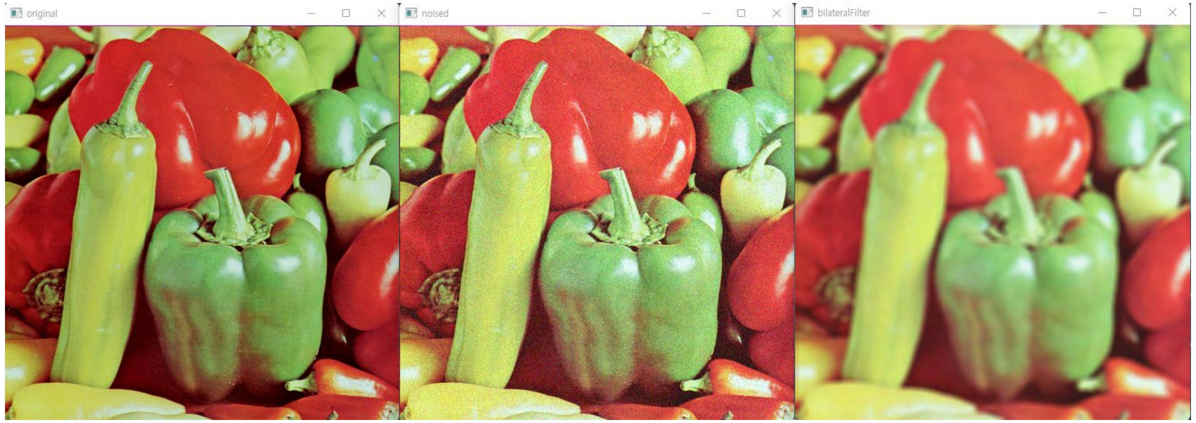


```
import cv2
import numpy as np
import matplotlib.pyplot as plt

image = cv2.imread("../Assignment/Testimage/image_Peppers512rgb.png").astype(np.float32) / 255


noised = (image + 0.2 * np.random.rand(*image.shape).astype(np.float32))
noised = noised.clip(0, 1)


in_dim = np.int(input("Diameter 입력하시오 : "))
S_color = np.float(input("Sigma Color 입력하시오 : "))
S_space = np.float(input("Sigma Space 입력하시오 : "))

bilat = cv2.bilateralFilter(noised, in_dim, S_color, S_space)

cv2.imshow("original", image)
cv2.imshow("noised", noised)
cv2.imshow("bilateralFilter", bilat)
cv2.waitKey()
cv2.destroyAllWindows()
```

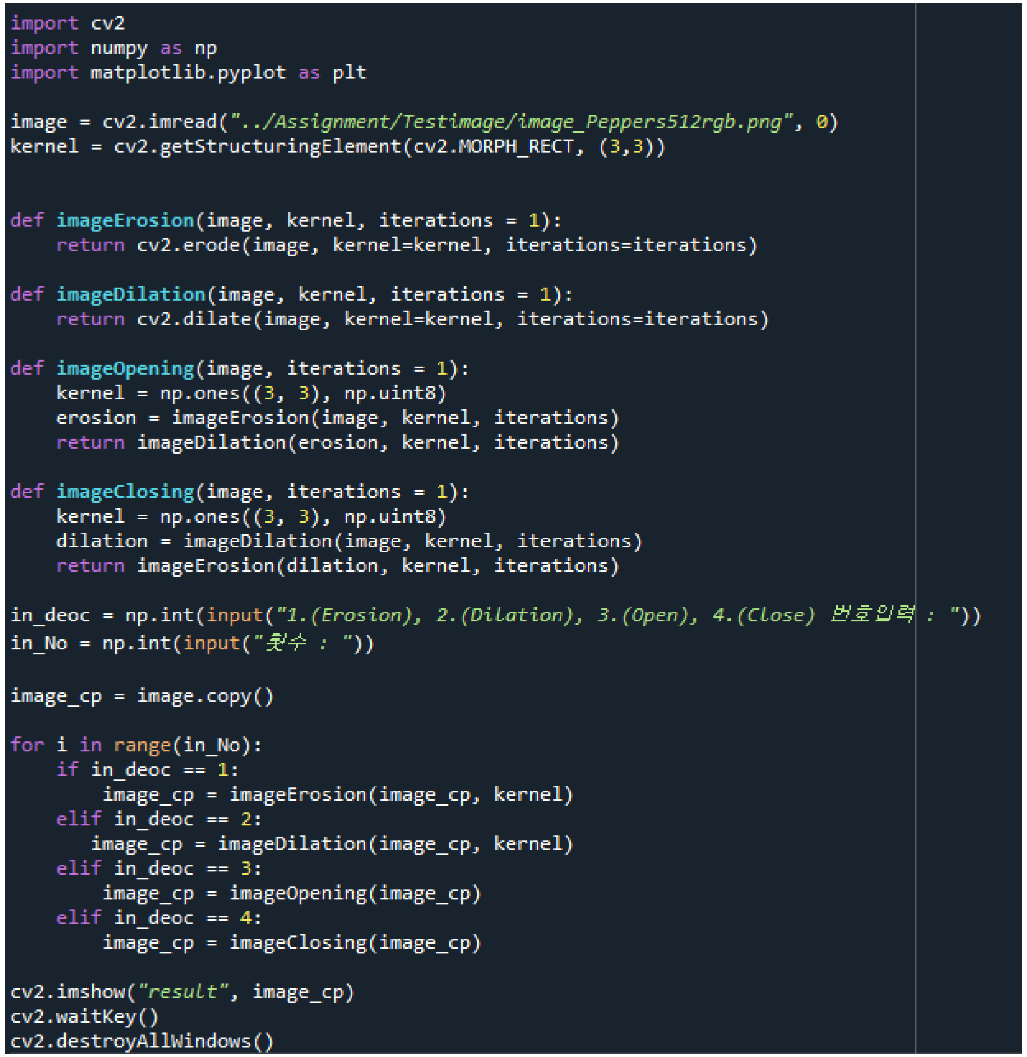
Diameter : 4, Sigma Color : 2, Sigma Space : 15 입력 결과



Diameter : 10, Sigma Color : 15, Sigma Space : 30 입력 결과

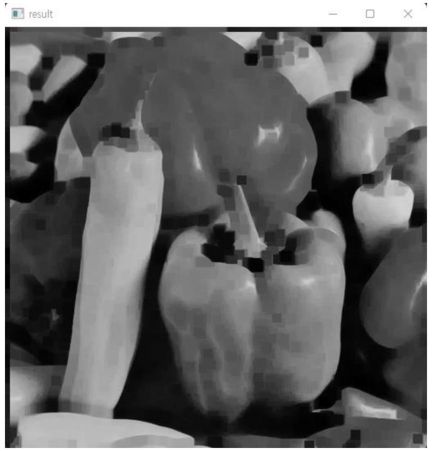


4) 모폴로지 필터

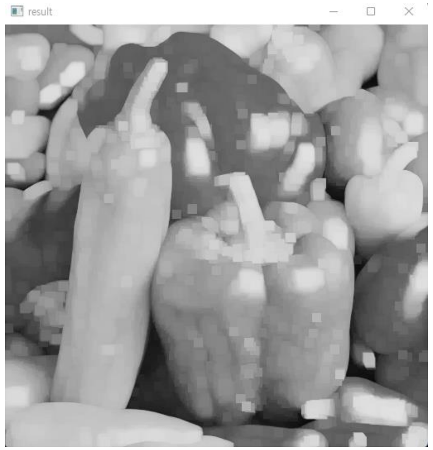


```
import cv2
import numpy as np
import matplotlib.pyplot as plt

image = cv2.imread("../Assignment/Testimage/image_Peppers512rgb.png", 0)
kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (3,3))


def imageErosion(image, kernel, iterations = 1):
    return cv2.erode(image, kernel=kernel, iterations=iterations)

def imageDilation(image, kernel, iterations = 1):
    return cv2.dilate(image, kernel=kernel, iterations=iterations)

def imageOpening(image, iterations = 1):
    kernel = np.ones((3, 3), np.uint8)
    erosion = imageErosion(image, kernel, iterations)
    return imageDilation(erosion, kernel, iterations)

def imageClosing(image, iterations = 1):
    kernel = np.ones((3, 3), np.uint8)
    dilation = imageDilation(image, kernel, iterations)
    return imageErosion(dilation, kernel, iterations)

in_deoc = np.int(input("1.(Erosion), 2.(Dilation), 3.(Open), 4.(Close) 번호입력 : "))
in_No = np.int(input("횟수 : "))

image_cp = image.copy()

for i in range(in_No):
    if in_deoc == 1:
        image_cp = imageErosion(image_cp, kernel)
    elif in_deoc == 2:
       image_cp = imageDilation(image_cp, kernel)
    elif in_deoc == 3:
        image_cp = imageOpening(image_cp)
    elif in_deoc == 4:
        image_cp = imageClosing(image_cp)

cv2.imshow("result", image_cp)
cv2.waitKey()
cv2.destroyAllWindows()
```

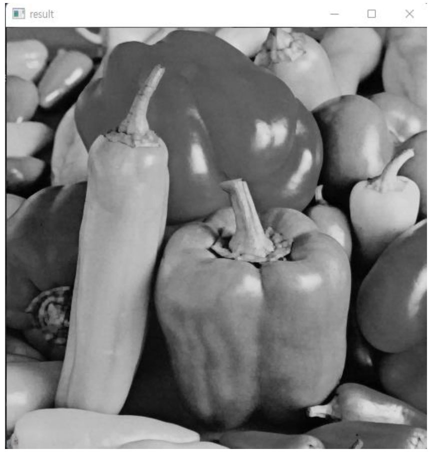
Erosion, 횟수 : 5



Dilation, 횟수 : 5



Opening, 횟수 : 5



Colsing, 횟수 : 5