



과목명 : 컴퓨터 비전

과제명 : 영역처리 실습예제

학 과 : 소프트웨어학과

제 출 일 자 : 2022.05.22

학 번 : 1726088

이 름 : 최민수

[문제] 예제 7.1.1 회선이용 블러링

[소스코드]

```
import numpy as np,cv2

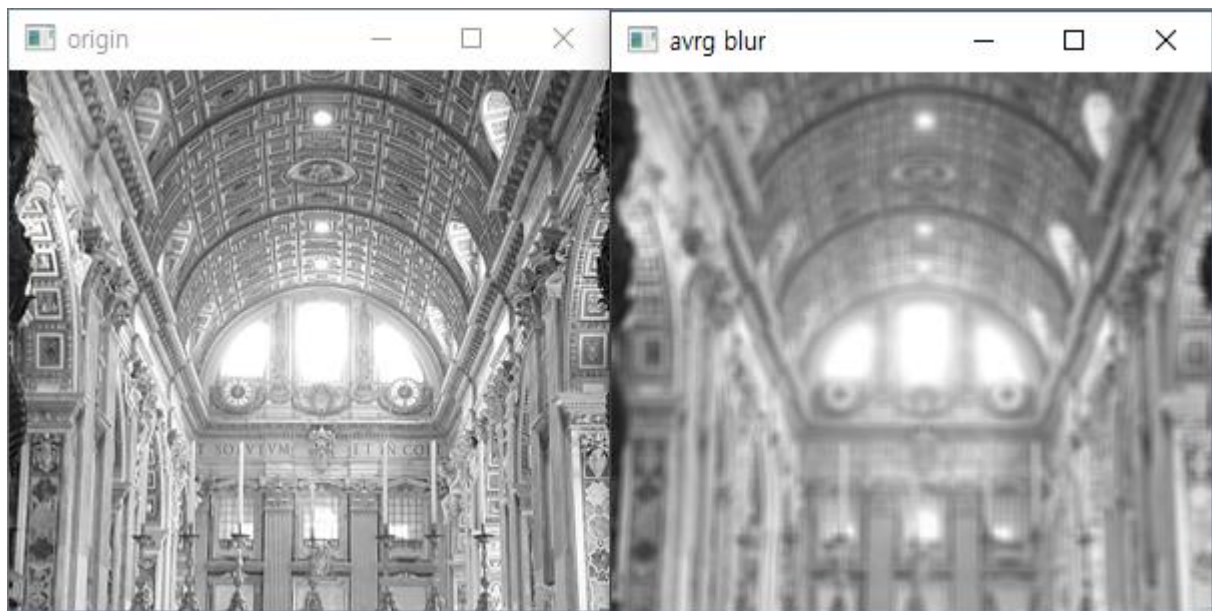
image = cv2.imread("C:/Users/chlasltn/Desktop/imagesss/filter_blur.jpg", cv2.IMREAD_GRAYSCALE) # 파일 읽어오기

kernel = np.array([[0.04, 0.04, 0.04, 0.04, 0.04],
                   [0.04, 0.04, 0.04, 0.04, 0.04],
                   [0.04, 0.04, 0.04, 0.04, 0.04],
                   [0.04, 0.04, 0.04, 0.04, 0.04],
                   [0.04, 0.04, 0.04, 0.04, 0.04]]) # 필터 마스크 행렬

kernel=np.ones((5,5))/5*2
blured=cv2.filter2D(image,-1,kernel) # 커널 값을 이용한 블러링

cv2.imshow('origin',image)
cv2.imshow('avrg blur',blured)
cv2.waitKey()
cv2.destroyAllWindows()
```

[결과]



[문제] 예제 7.1.2 회선이용 샤프닝

[소스코드]

```
import numpy as np,cv2

image = cv2.imread("C:/Users/chlasltn/Desktop/imagesss/filter_blur.jpg", cv2.IMREAD_GRAYSCALE) # 파일 읽어오기

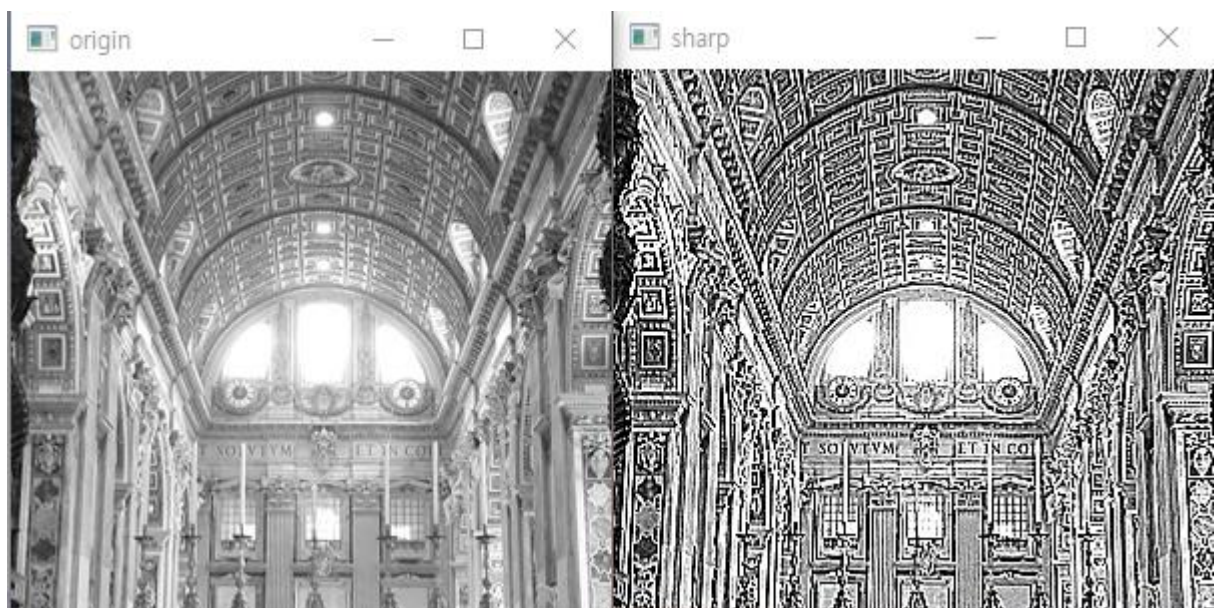
kernel=np.array([[ -1,-1,-1],[-1,9,-1],[-1,-1,-1]]) # 전체 값이 1 이 되게 커널 값 설정

sharpened=cv2.filter2D(image,-1,kernel) # 커널을 이용한 샤프닝

cv2.imshow('origin',image)
cv2.imshow('sharp',sharpened)

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

[결과]



[문제] 예제 7.3.1 최솟값 - 최댓값 필터링

[소스코드]

```
import numpy as np
import cv2

def minmax_filter(image, ksize, mode): # 최솟값 & 최댓값 필터링 함수
    rows, cols = image.shape[:2]
    dst = np.zeros((rows, cols), np.uint8)
    center = ksize // 2 # 마스크 절반 크기

    for i in range(center, rows - center): # 입력 영상 순회
        for j in range(center, cols - center):
            y1, y2 = i - center, i + center + 1 # 마스크 높이 범위
            x1, x2 = j - center, j + center + 1 # 마스크 너비 범위
            mask = image[y1:y2, x1:x2] # 마스크 영역
            dst[i, j] = cv2.minMaxLoc(mask)[mode]

    return dst

image = cv2.imread("C:/Users/chlasltn/Desktop/imagesss/min_max.jpg", cv2.IMREAD_GRAYSCALE)
# 파일 읽어오기

minfilter_img = minmax_filter(image, 3, 0) # 3 x 3 마스크 최솟값 필터링
maxfilter_img = minmax_filter(image, 3, 1) # 3 x 3 마스크 최댓값 필터링

cv2.imshow("image", image)
cv2.imshow("minfilter_img", minfilter_img)
cv2.imshow("maxfilter_img", maxfilter_img)

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

[결과]



[문제] 예제 7.3.2 평균값 필터링

[소스코드]

```
import numpy as np, cv2

def average_filter(image, ksize): # 평균값 필터링 함수
    rows, cols = image.shape[:2]
    dst = np.zeros((rows, cols), np.uint8)
    center = ksize // 2 # 마스크 절반 크기

    for i in range(rows): # 입력 영상 순회
        for j in range(cols):
            y1, y2 = i - center, i + center + 1 # 마스크 높이 범위
            x1, x2 = j - center, j + center + 1 # 마스크 너비 범위
            if y1 < 0 or y2 > rows or x1 < 0 or x2 > cols: # 입력 영상 벗어남
                dst[i, j] = image[i, j]
            else:
                mask = image[y1:y2, x1:x2]
                dst[i, j] = cv2.mean(mask)[0]
    return dst

image = cv2.imread("C:/Users/chlasltn/Desktop/imagesss/smoothing.jpg", cv2.IMREAD_GRAYSCALE)

kernel = np.array([[0.04, 0.04, 0.04, 0.04, 0.04],
                   [0.04, 0.04, 0.04, 0.04, 0.04],
                   [0.04, 0.04, 0.04, 0.04, 0.04],
                   [0.04, 0.04, 0.04, 0.04, 0.04],
                   [0.04, 0.04, 0.04, 0.04, 0.04]]) # 평균값을 위한 kernel 생성

kernel = np.ones((5, 5)) / 5 * 2
avg_img = average_filter(image, 5) # 사용자 정의 평균값 필터 함수
blur_img = cv2.filter2D(image, -1, kernel) # 블러링 함수
box_img = cv2.boxFilter(image, ddepth=-1, ksize=(5, 5)) # 박스 필터 함수

cv2.imshow("image", image)
cv2.imshow("avg_img", avg_img)
cv2.imshow("blur_img", blur_img)
cv2.imshow("box_img", box_img)

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

[결과]



[문제] 예제 7.3.3 미디언 필터링

[소스코드]

```
import numpy as np,cv2

def median_filter(image,ksize): # 미디언 필터링 함수
    rows,cols=image.shape[:2]
    dst=np.zeros((rows,cols),np.uint8)

    center=ksize//2 # 마스크 절반 크기

    for i in range(center,rows-center): # 입력 영상 순호너 |
        for j in range(center,cols-center):
            y1,y2=i-center,i+center+1 # 마스크 높이 범위
            x1,x2=j-center,j+center+1 # 마스크 너비 범위

            mask=image[y1:y2,x1:x2].flatten() # 관심 영역 지정 및 벡터 변환

            sort_mask=cv2.sort(mask,cv2.SORT_EVERY_COLUMN) # 정렬 수행
            dst[i,j]=sort_mask[sort_mask.size//2]
    return dst

def salt_pepper_noise(img,n): # 소금 후추 잡음 생성 함수
    h,w=img.shape[:2]

    x,y=np.random.randint(0,w,n),np.random.randint(0,h,n)
    noise=img.copy()

    for (x,y) in zip(x,y):
        noise[y,x]=0 if np.random.rand()<0.5 else 255

    return noise

image = cv2.imread("C:/Users/chlas1tn/Desktop/imagesss/bit_test.jpg", cv2.IMREAD_GRAYSCALE)
# 파일 읽어오기

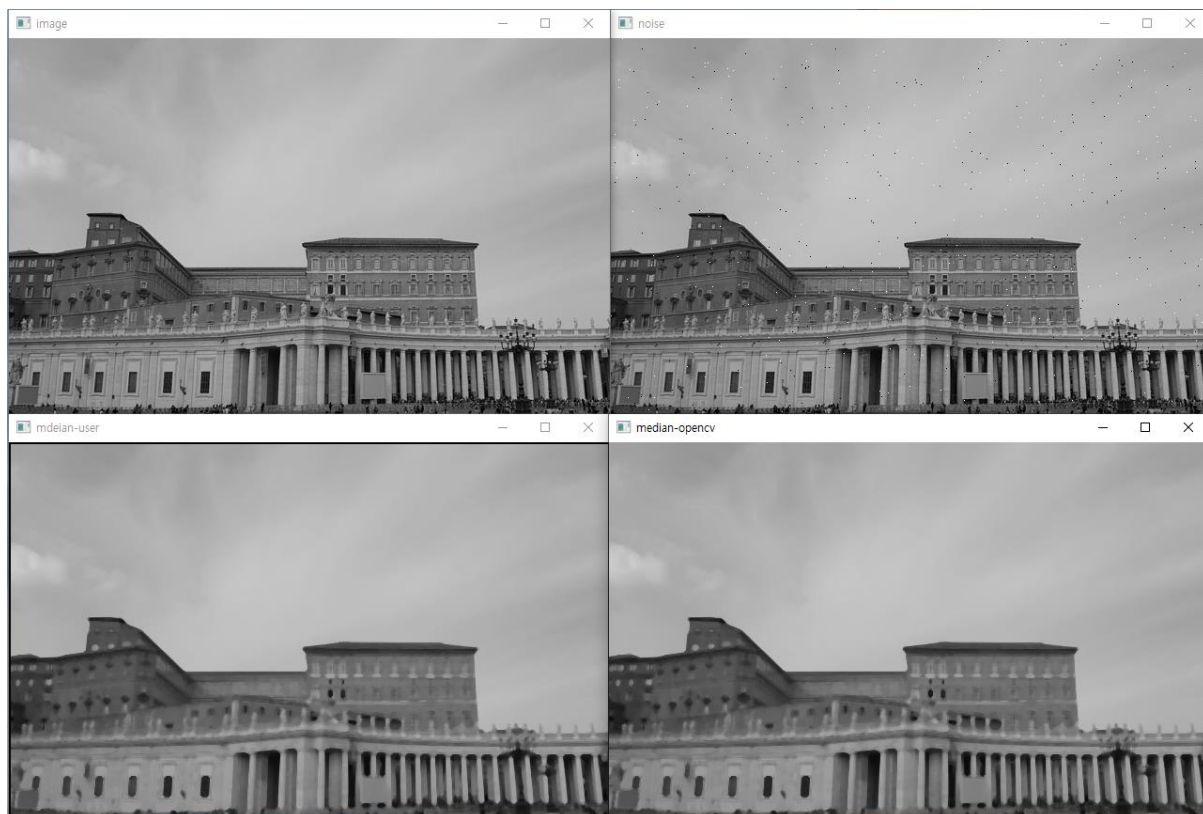
noise=salt_pepper_noise(image,500) # 소금 - 후추 잡음 영상 생성

med_img1=median_filter(noise,5) # 사용자 정의 함수
med_img2=cv2.medianBlur(noise,5) # 블러링 함수

cv2.imshow("image",image)
cv2.imshow("noise",noise)
cv2.imshow("mdeian-user",med_img1)
cv2.imshow("median-opencv",med_img2)

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

[결과]



[문제] 예제 7.3.4 가우시안 필터링

[소스코드]

```
import numpy as np, cv2

def getGaussianMask(ksize, sigmaX, sigmaY): # 가우시안 마스크 생성 함수
    sigma = 0.3 * ((np.array(ksize) - 1.0) * 0.5 - 1.0) + 0.8
    if sigmaX <= 0: sigmaX = sigma[0] # 표준편차가 양수가 아닐 때
    if sigmaY <= 0: sigmaY = sigma[1] # ksize 로 기본 표준편차 계산

    u = np.array(ksize) // 2 # 커널 크기 절반
    x = np.arange(-u[0], u[0] + 1, 1) # x 방향 범위
    y = np.arange(-u[1], u[1] + 1, 1) # y 방향 범위
    x, y = np.meshgrid(x, y) # 정방 행렬 생성

    ratio = 1 / (sigmaX * sigmaX * 2 * np.pi)
    v1 = x ** 2 / (2 * sigmaX ** 2)
    v2 = y ** 2 / (2 * sigmaY ** 2)
    mask = ratio * np.exp(-(v1 + v2)) # 2 차원 정규분포 수식

    return mask / np.sum(mask) # 원소 전체 합 1 유지

image = cv2.imread("C:/Users/chlasltn/Desktop/imagesss/smoothing.jpg", cv2.IMREAD_GRAYSCALE)
# # 파일 읽어오기

ksize = (17, 5) # 커널 크기
gaussian_2d = getGaussianMask(ksize, 0, 0)
gaussian_1dX = cv2.getGaussianKernel(ksize[0], 0, cv2.CV_32F) # 가로 방향 마스크
gaussian_1dY = cv2.getGaussianKernel(ksize[1], 0, cv2.CV_32F) # 세로 방향 마스크

gauss_img1 = cv2.filter2D(image, -1, gaussian_2d) # 사용자 생성 마스크
gauss_img2 = cv2.GaussianBlur(image, (3, 3), 0) # 가우시안 블러링
gauss_img3 = cv2.sepFilter2D(image, -1, gaussian_1dX, gaussian_1dY) # sepFilter2D 함수 사용

titles = ['image', 'gauss_img1', 'gauss_img2', 'gauss_img3']
for t in titles: cv2.imshow(t, eval(t))
cv2.waitKey()
cv2.destroyAllWindows()
```

[결과]



[문제] 예제 7.3.5 블러링과 캐니 에지를 이용한 에지 검출

[소스코드]

```
import cv2

def onTrackbar(th): # 트랙바 콜백 함수
    rep_edge=cv2.GaussianBlur(rep_gray,(5,5),0) # 가우시안 블러링
    rep_edge=cv2.Canny(rep_edge,th,th*2,5) # 캐니 에지 검출

    h,w=image.shape[:2]
    cv2.rectangle(rep_edge,(0,0,w,h),255,-1)
    color_edge=cv2.bitwise_and(rep_image,rep_image,mask=rep_edge)

    cv2.imshow("color edge",color_edge)

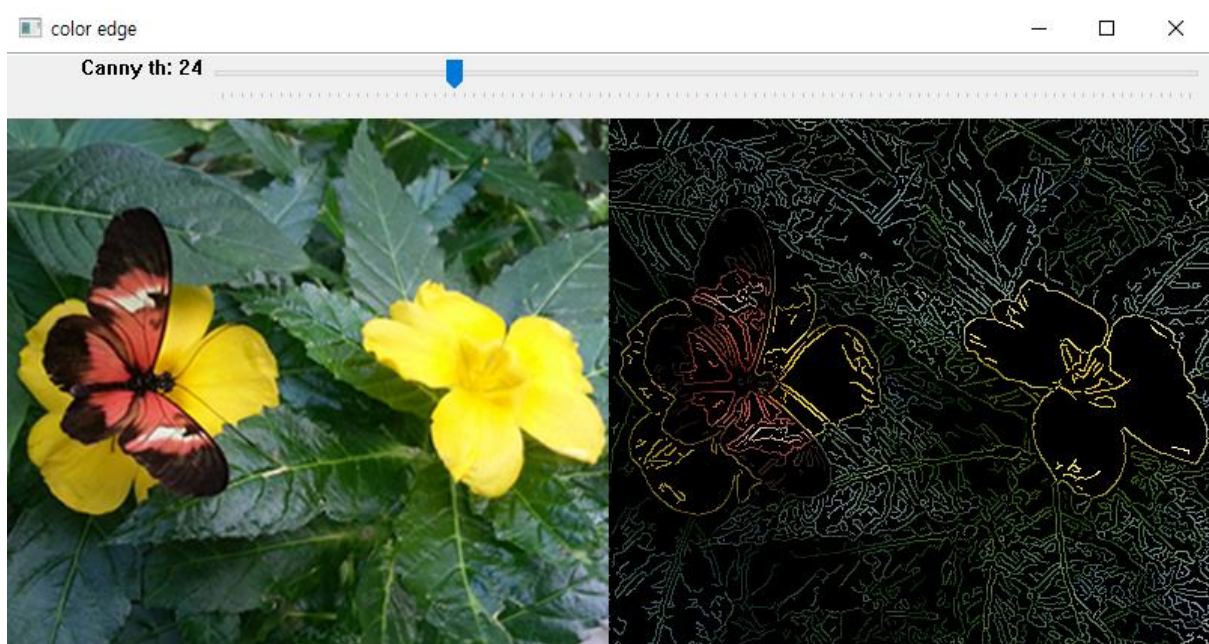
image = cv2.imread("C:/Users/chlasltn/Desktop/imagesss/color_edge.jpg", cv2.IMREAD_COLOR) #
파일 읽어오기

th=50
rep_image=cv2.repeat(image,1,2) # 가로 반복 복사
rep_gray=cv2.cvtColor(rep_image,cv2.COLOR_BGR2GRAY) # 명암도 영상 변환

cv2.namedWindow("color edge",cv2.WINDOW_AUTOSIZE)
cv2.createTrackbar("Canny th","color edge",th,100,onTrackbar)

onTrackbar(th)
cv2.waitKey()
cv2.destroyAllWindows()
```

[결과]



[문제] 예제 7.2.3 로버츠 에지 검출

[소스코드]

```
import numpy as np, cv2

image = cv2.imread("C:/Users/chlasltn/Desktop/imagesss/edge.jpg", cv2.IMREAD_GRAYSCALE) #
파일 읽어오기

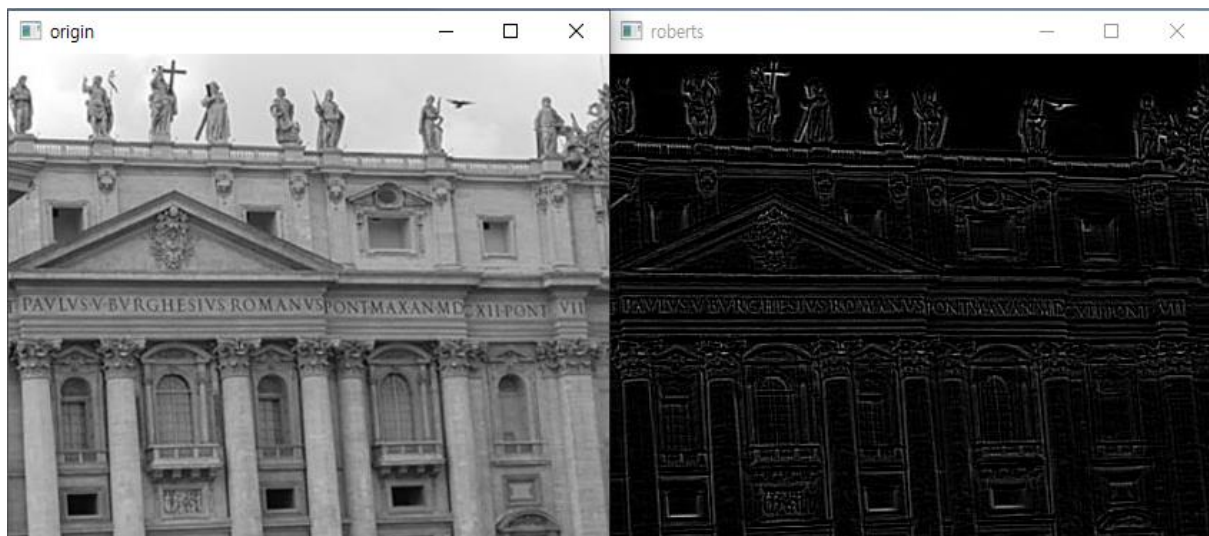
robertsX=np.array([[ -1, 0, 0],[ 0, 1, 0],[ 0, 0, 0]]) # x 방향 마스크 생성
robertsY=np.array([[ 0, 0, -1],[ 0, 1, 0],[ 0, 0, 0]]) # y 방향 마스크 생성

robertsX=cv2.convertScaleAbs(cv2.filter2D(image, -1, robertsX))
robertsY=cv2.convertScaleAbs(cv2.filter2D(image, -1, robertsY))

roberts=cv2.addWeighted(robertsX, 1, robertsY, 1, 0)

cv2.imshow("origin", image)
cv2.imshow("roberts", roberts)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

[결과]



[문제] 예제 7.2.4 프리윗 에지 검출

[소스코드]

```
import numpy as np, cv2

image = cv2.imread("C:/Users/chlas1tn/Desktop/imagesss/edge.jpg", cv2.IMREAD_GRAYSCALE) #
파일 읽어오기

prewittX=np.array([[ -1, -1, -1],[ 0, 0, 0],[ 1, 1, 1]]) # x 방향 마스크 생성
prewittY=np.array([[ 1, 0, -1],[ 1, 0, -1],[ 1, 0, -1]]) # y 방향 마스크 생성

prewittX=cv2.convertScaleAbs(cv2.filter2D(image, -1, prewittX)) # 절대값화 시키는 함수
prewittY=cv2.convertScaleAbs(cv2.filter2D(image, -1, prewittY))

prewitt=cv2.addWeighted(prewittX, 1, prewittY, 1, 0) # x 값과 y 마스크 값을 합침

cv2.imshow("origin", image)
cv2.imshow("prewittX", prewittX)
cv2.imshow("prewittY", prewittY)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

[결과]



[문제] 예제 7.2.5 소벨 에지 검출

[소스코드]

```
import numpy as np, cv2

image = cv2.imread("C:/Users/chlas1tn/Desktop/imagesss/edge.jpg", cv2.IMREAD_GRAYSCALE) #
파일 읽어오기

sobelX=np.array([[ -1, -2, -1],[ 0, 0, 0],[ 1, 2, 1]]) # 수평 소벨 마스크
sobelY=np.array([[ 1, 0, -1],[ 2, 0, -2],[ 1, 0, -1]]) # 수직 소벨 마스크

sobelX=cv2.convertScaleAbs(cv2.filter2D(image, -1, sobelX))
sobelY=cv2.convertScaleAbs(cv2.filter2D(image, -1, sobelY))

roberts=cv2.addWeighted(robertsX, 1, robertsY, 1, 0)

cv2.imshow("origin", image)
cv2.imshow("sobelX", sobelX)
cv2.imshow("sobelY", sobelY)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

[결과]



[문제] 예제 7.2.6 라플라시안 에지 검출

[소스코드]

```
import numpy as np,cv2

image = cv2.imread("C:/Users/chlas1tn/Desktop/imagesss/laplacian.jpg", cv2.IMREAD_GRAYSCALE)
) # 파일 읽어오기

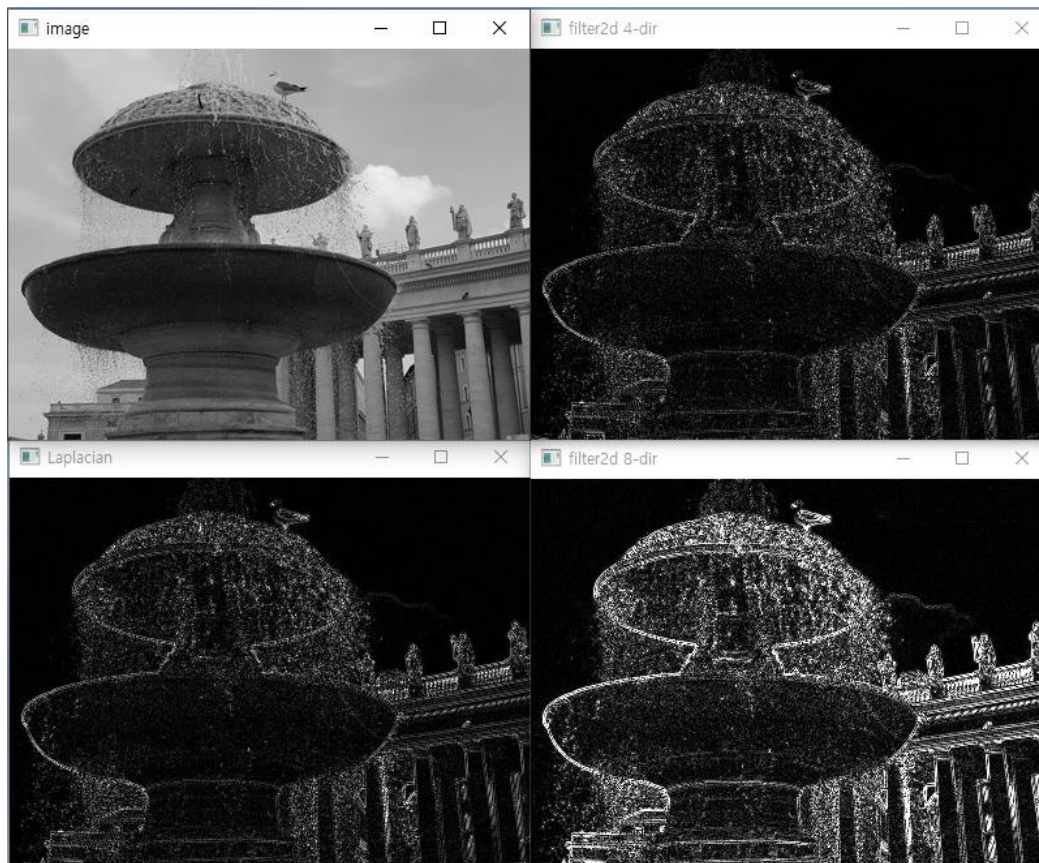
data1=[[0,1,0],[1,-4,1],[0,1,0]] # 4 방향 필터
data2=[[-1,-1,-1],[-1,8,-1],[-1,-1,-1]] # 8 방향 필터

mask4=np.array(data1,np.int16) # 음수로 인해 int16 행렬 선언
mask8=np.array(data2,np.int16)

dst1=cv2.filter2D(image,cv2.CV_16S,mask4) # OpenCV 회선 함수 호출
dst2=cv2.filter2D(image,cv2.CV_16S,mask8)
dst3=cv2.Laplacian(image,cv2.CV_16S,1) # 라플라시안 수행 함수

cv2.imshow("image",image)
cv2.imshow("filter2d 4-dir",cv2.convertScaleAbs(dst1)) # 형변환후 영상 표시
cv2.imshow("filter2d 8-dir",cv2.convertScaleAbs(dst2))
cv2.imshow("Laplacian",cv2.convertScaleAbs(dst3))
cv2.waitKey(0)
```

[결과]



[문제] 예제 7.2.7 LoG / DoG 에지 검출

[소스코드]

```
import numpy as np, cv2

image = cv2.imread("C:/Users/chlas1tn/Desktop/imageesss/dog.jpg", cv2.IMREAD_GRAYSCALE) # 파일 읽어오기

gaus=cv2.GaussianBlur(image,(7,7),0,0) # 가우시안 마스크 적용

dst1=cv2.Laplacian(gaus,cv2.CV_16S,7) # 라플라시안 수행
gaus1=cv2.GaussianBlur(image,(3,3),0) # 가우시안 블러링
gaus2=cv2.GaussianBlur(image,(9,9),0)

dst2=gaus1-gaus2 # DoG 수행

cv2.imshow("image",image)
cv2.imshow("dst1 Log",dst1.astype('uint8'))
cv2.imshow("dst2 Dog",dst2)
cv2.waitKey(0)
```

[결과]

