차량비전시스템

Assignment 4

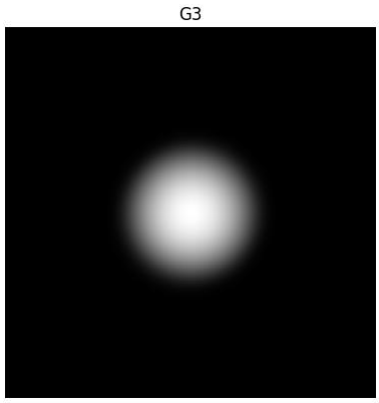
학번 5533082
전공 컴퓨터공학전공
이름 최승환

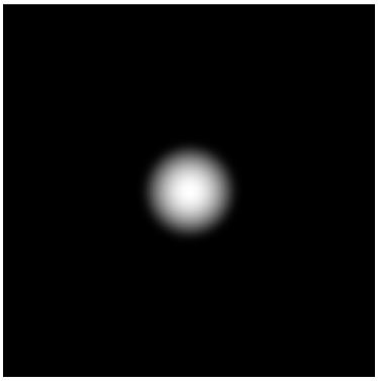
Topic. 주파수 영역에서의 Gaussian filter

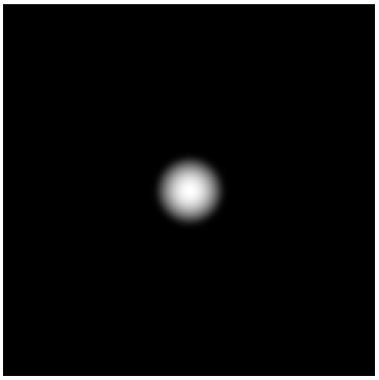
(1) 가우시안 필터 식에서의 σ을 3, 5, 7, 9, 11로 변화시키면서 주파수 영역에서 확인하시오.
(2) 서로 다른 σ값에 대한 필터 파일명을 각각 G3, G5, G7, G9, G11로 하시오.

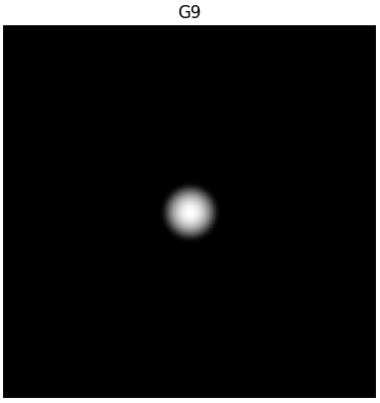
1. 가우시안 커널 정의: 가우시안 커널을 정의하기 위해서 scipy 라이브러리를 이용하였다.
2. 주파수 영역에서의 가우시안 필터: np.fft.fft2() 함수를 통해 2차원 가우시안 필터의 푸리에 변환.
3. 시각적 표현: 주파수 영역에서 필터링이 어떤 변화를 일으키는지 시각적으로 확인하기 위해 주파수 성분의 크기를 계산하여 나타낸다.

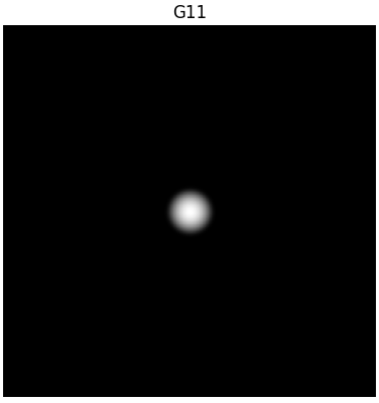
```
sig = [3, 5, 7, 9, 11]
G = {f"{s}": [] for s in sig}
for i, s in enumerate(sig):
    gauss = np.outer(signal.gaussian(x.shape[0], s), signal.gaussian(x.shape[1], s))
    G[f"{s}"] = fp.fft2(fp.ifftshift(gauss))
    G_mag = 20 * np.log10(0.1 + fp.fftshift(G[f"{s}"]))
    plt.imshow(G_mag.real, 'gray'), plt.title(f"G{s}"), plt.axis("off")
    plt.show()
```











(3) 이미지 파일(x)의 푸리에 변환 결과(X)를 나타내시오.

1. np.fft.fft2() 함수를 이용해 x를 주파수 영역으로 푸리에 변환
2. 시각적으로 표현을 위해 주파수 성분의 크기를 구한다.

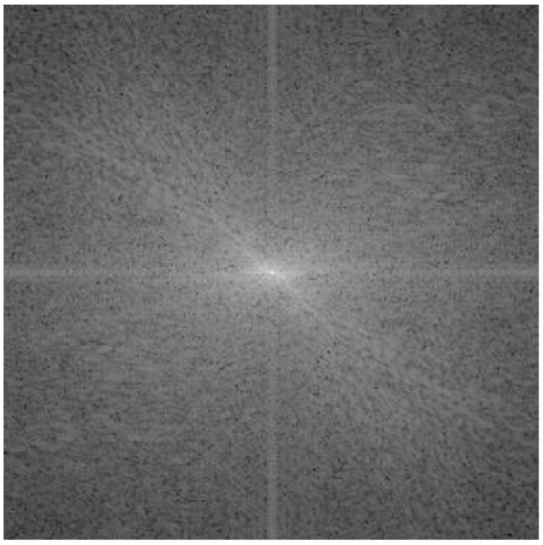
```
# 이미지 읽기
x = cv2.imread('./img/lena.jpg', cv2.IMREAD_GRAYSCALE)

# 주파수 영역으로 변환
X = fp.fft2(x)
X_mag = 20 * np.log10(0.1 + fp.fftshift(X))

# 원본
plt.figure(figsize=(10,20))
plt.subplot(121), plt.imshow(x, 'gray'), plt.title("x"), plt.axis("off")
plt.subplot(122), plt.imshow(X_mag.real, 'gray'), plt.title("X"), plt.axis("off")
plt.show()
```

x

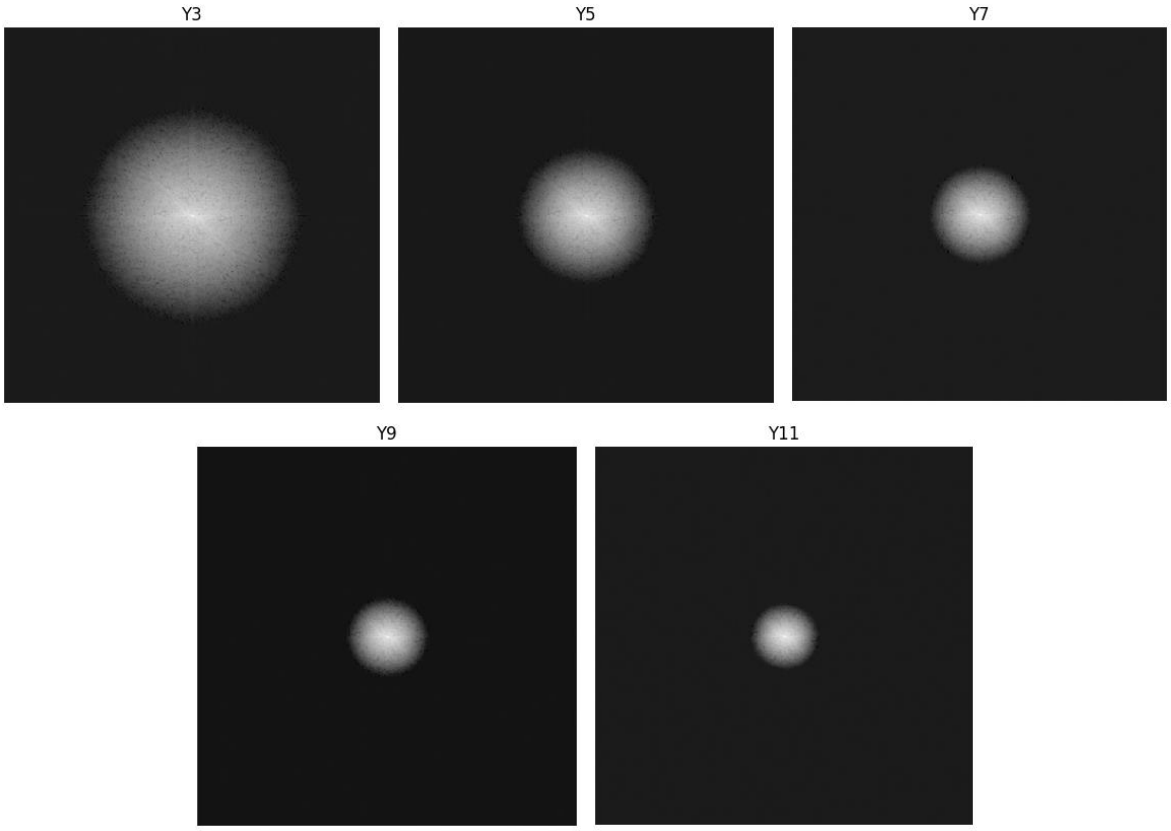
X



(4) X와 G3, G5, G7, G9, G11과 각각 곱한 후 그 결과를 확인하시오. 결과의 이름은 각각 Y3, Y5, Y7, Y9, Y11 이라 한다.

푸리에 변환은 주파수 영역에서의 연산을 공간 영역에서의 연산으로 변환해준다. 따라서 두 이미지의 곱셈 결과를 다시 공간영역으로 변환하면 두 이미지의 컨볼루션 결과와 동일한 결과를 얻을 수 있다.

```
Y = {f"{s}": [] for s in sig}
for i, s in enumerate(sig):
    Y[f"{s}"] = X * G[f"{s}"]
    Y_mag = 20 * np.log10(0.1 + fp.fftshift(Y[f"{s}"]))
    plt.imshow(Y_mag.real, 'gray'), plt.title(f"Y{s}"), plt.axis("off")
    plt.show()
```

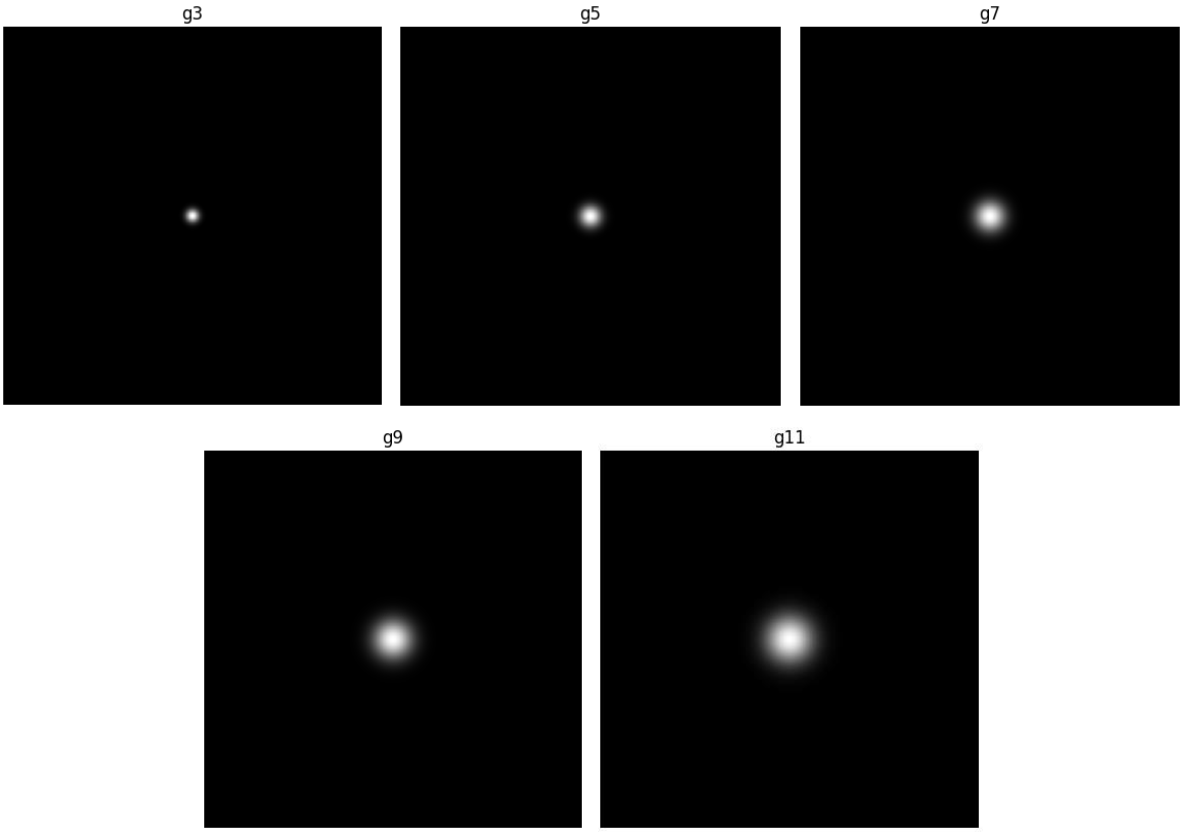


주파수 영역의 이러한 성질 덕분에 이미지 프로세싱에서는 주파수 영역에서의 연산이 빠르고 효율적이며 필터링에 유용하게 사용된다. 즉, 주파수 영역에서 필터링을 하고 다시 공간 영역으로 변환하여 이미지를 처리하는 것이 일반적인 방법이 된다.

(5) G3, G5, G7, G9, G11의 역푸리에 변환 결과를 각각 g3, g5, g7, g9, g11 파일명으로 저장하시오.

주파수 영역에서의 가우시안 필터인 G를 역푸리에 적용하여 공간 영역에서의 성분을 갖도록 한다.

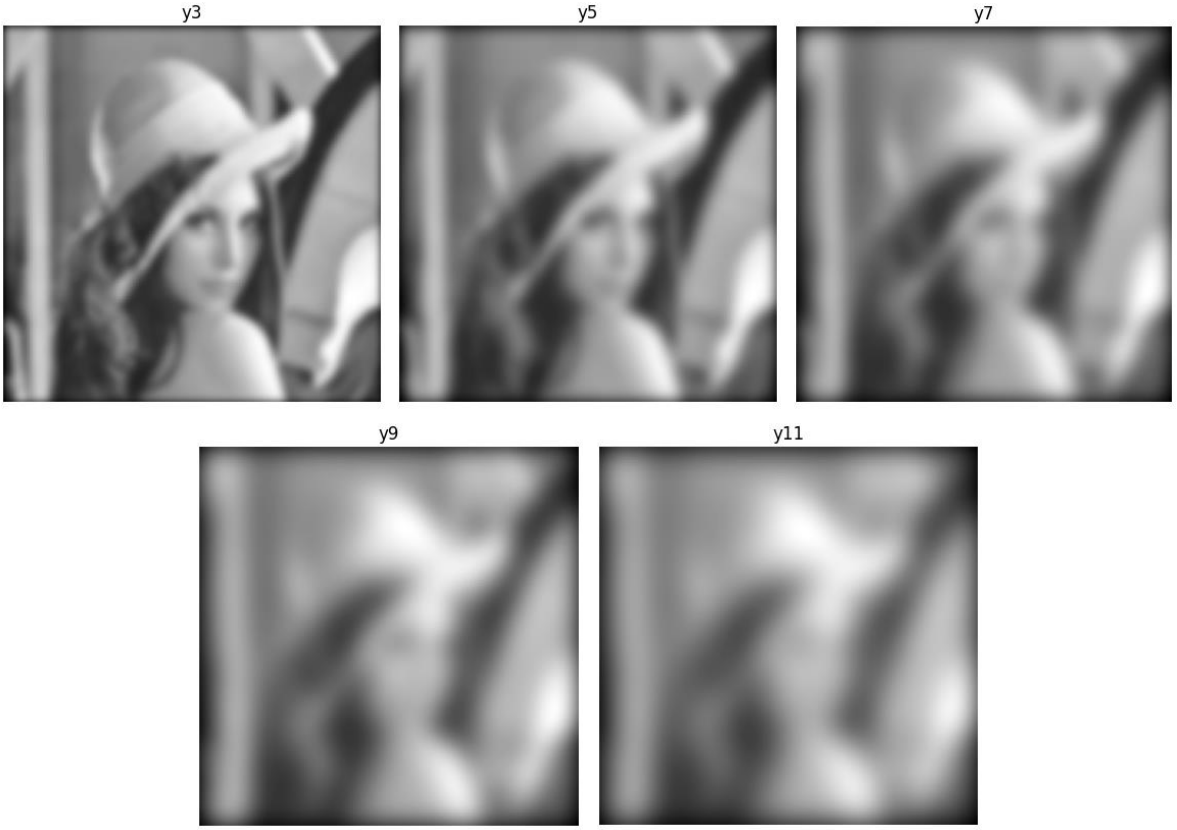
```
g = {f"{s}": [] for s in sig}
for i, s in enumerate(sig):
    g[f"{s}"] = fp.fftshift(fp.ifft2(G[f"{s}"])).real
    plt.imshow(g[f"{s}"], 'gray'), plt.title(f"g{s}"), plt.axis("off")
    plt.show()
```



(6) x와 g3, g5, g7, g9, g11 사이의 컨볼루션 결과를 구현하여 나타내고 이름은 각각 y3, y5, y7, y9, y11로 하시오.

주파수 영역에서의 가우시안 필터(G)의 역푸리에 적용한 g 필터를 이용하여 공간 영역에서의 컨볼루션 연산 수행했다.

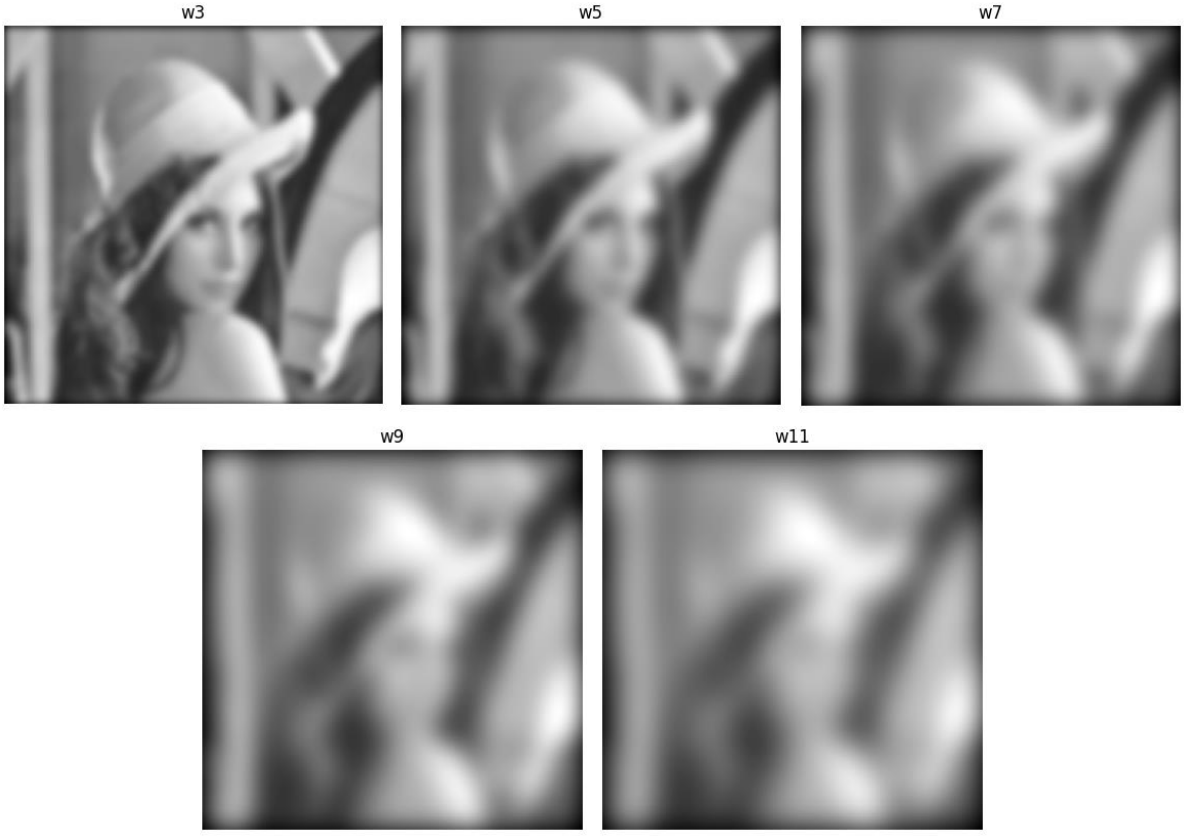
```
y = {f"{s}": [] for s in sig}
for i, s in enumerate(sig):
    y[f"{s}"] = signal.convolve2d(x, g[f"{s}"], mode='same')
    plt.imshow(y[f"{s}"], 'gray'), plt.title(f"y{s}"), plt.axis("off")
    plt.show()
```



(7) Opencv에서 제공하는 가우시안 필터에서 σ을 3, 5, 7, 9, 11 변환시켜서 x와 필터링한 결과를 각각 w3, w5, w7, w9, w11로 하시오.

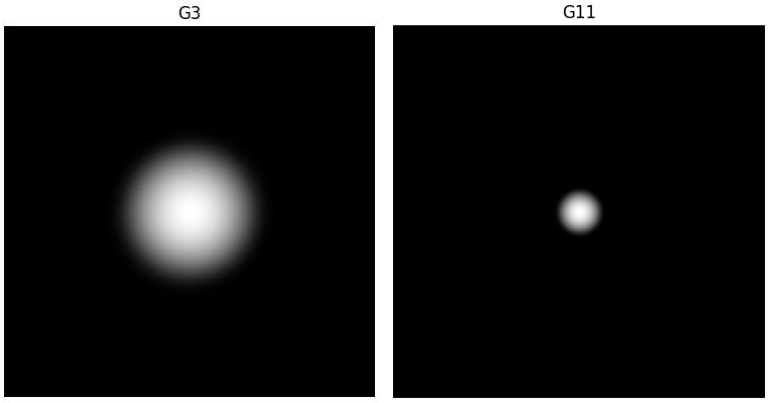
공간 영역에서의 가우시안 필터를 새로 정의하여 공간 영역에서의 원본 이미지(x)와 가우시안 필터의 컨볼루션 연산 수행했다.

```
w = {f"{s}": [] for s in sig}
for i, s in enumerate(sig):
    gauss = np.outer(signal.gaussian(x.shape[0], s), signal.gaussian(x.shape[1], s))
    w[f"{s}"] = signal.convolve2d(x, gauss, mode='same')
    plt.imshow(w[f"{s}"], 'gray'), plt.title(f"w{s}"), plt.axis("off")
    plt.show()
```

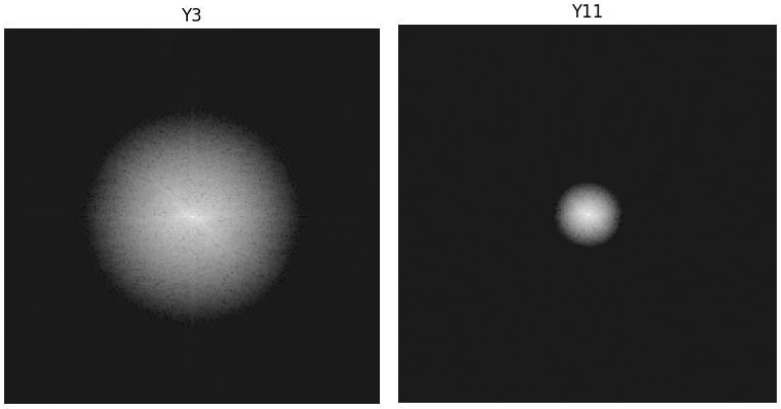


(8) (6)에서 구한 결과와 (7)에서 구한 결과를 비교한 후 분석 결과를 설명하시오.

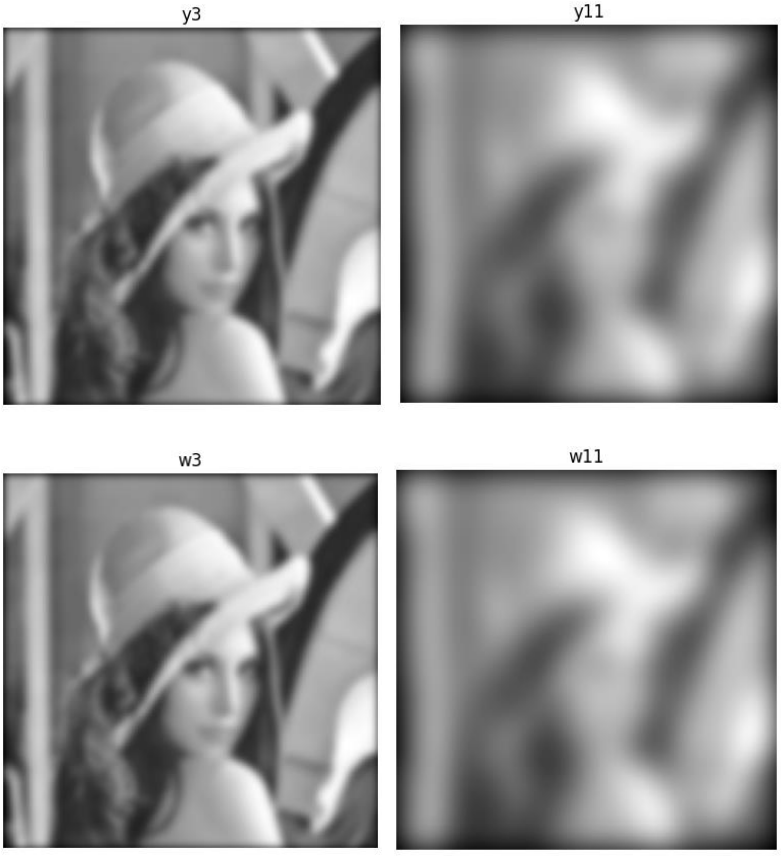
저주파 통과 필터(LPF)는 낮은 주파수 대역만 통과시키는 필터이다. 이미지에서 LPF를 사용한 결과는 낮은 주파수 대역만 남게 되어 결과적으로 블러 효과를 가진 이미지가 된다. 주파수 영역 이미지의 정중앙부분은 주파수 대역이 낮은 값들이 위치하고 있다. 아래 그림 G에서 흰색 부분은 통과되는 주파수 영역이고, 검정색 부분은 필터링 되는 주파수 영역이다. 가우시안 필터는 중앙 부분인 주파수가 낮은 영역만 통과시키는 LPF이다. 즉, 흰색 부분이 넓어질수록 이미지의 블러 효과는 적어지고, 반대로 검정색 부분이 넓어질수록 이미지 블러 효과는 커진다. 가우시안 분포 관점에서 말하자면 아래 그림과 같이 시그마 값이 커질수록 검정색 부분이 넓어지고, 시그마 값이 작아질수록 흰색 부분이 넓어진다고 말할 수 있다.



아래 그림 Y는 주파수 영역에서 x * G 필터링을 진행한 결과이다. 검정색 부분은 필터링 되었고, 흰색 부분만 통과되는 결과를 확인할 수 있다.



(6)문제에서 주파수 영역에서의 G의 역푸리에는 공간 영역에서의 g이다. g는 공간 영역에서의 가우시안 분포이고, 가우시안 필터링 과정은 컨볼루션 연산을 통해 진행된다. 반면에 (4)문제에서 진행했던 주파수 영역에서의 컨볼루션 연산은 단순 곱셈으로 진행된다. (6)과 (4)의 두 계산 결과 이미지는 동일하다. 또한, (6)과 (7)은 결국 공간 영역에서의 가우시안 필터링이기 때문에 아래 그림과 같이 서로 동일한 결과를 가지게 된다.


y3
y11
w3
w11