산업컴퓨터비전 실제 – HW1

산업인공지능학과
2023254009
최현동

1. 히스토그램 평탄화

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

color = cv2.imread( filename: '../../1.Data/Lena.png', cv2.IMREAD_COLOR)
color_Hist = color.copy()
color_hsv = color.copy()

channel = input("Enter the channel to perform histogram equalization (R/G/B): ").upper()

if channel == "R":
    hist, bins = np.histogram(color[..., 2], bins: 256, range: [0, 255])
    color_Hist[..., 2] = cv2.equalizeHist(color[..., 2])
    plt.title('Red Histogram')
elif channel == "G":
    hist, bins = np.histogram(color[..., 1], bins: 256, range: [0, 255])
    color_Hist[..., 1] = cv2.equalizeHist(color[..., 1])
    plt.title('Green Histogram')
elif channel == "B":
    hist, bins = np.histogram(color[..., 0], bins: 256, range: [0, 255])
    color_Hist[..., 0] = cv2.equalizeHist(color[..., 0])
    plt.title('Blue Histogram')

plt.fill_between(range(256), hist, y2: 0)
plt.xlabel('pixel value')
plt.show()

cv2.imshow( winname: 'original color', color)

if channel == "R":
    cv2.imshow( winname: 'equalized Red color', color_Hist)
elif channel == "G":
    cv2.imshow( winname: 'equalized Green color', color_Hist)
elif channel == "B":
    cv2.imshow( winname: 'equalized Blue color', color_Hist)

hsv = cv2.cvtColor(color_hsv, cv2.COLOR_BGR2HSV)

hsv[..., 2] = cv2.equalizeHist(hsv[..., 2])
color_eq = cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR)

cv2.imshow( winname: 'equalized HSV V-channel', color_eq)
cv2.waitKey()
cv2.destroyAllWindows()
```

1. 히스토그램 평탄화

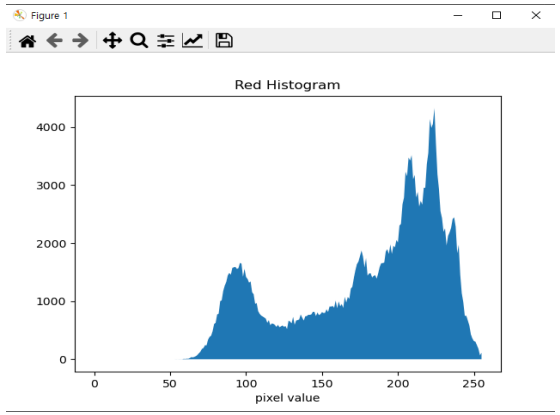
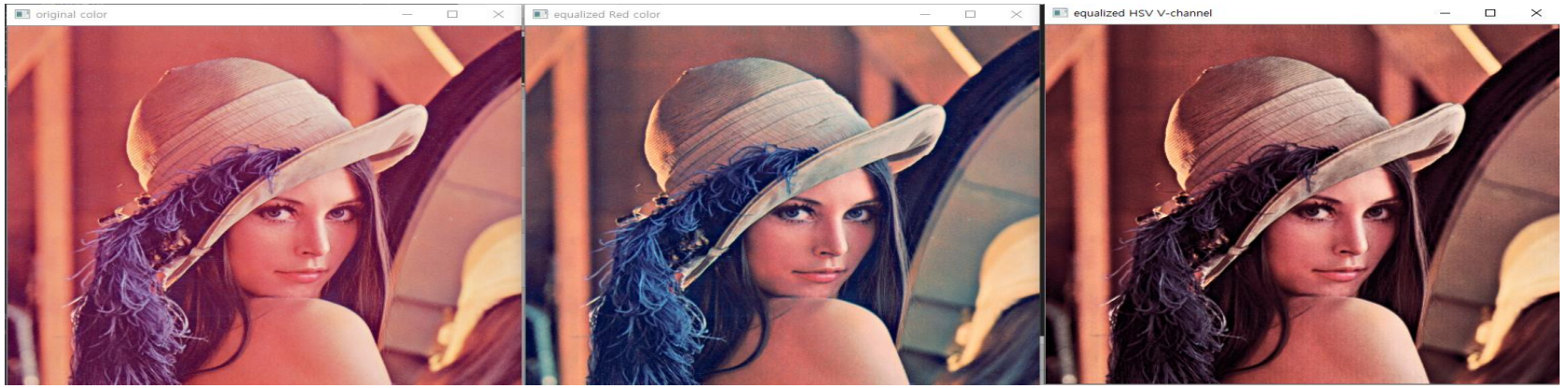


Figure 1
Red Histogram
4000
3000
2000
1000
0
0
50
100
150
200
250
pixel value


original color
equalized Red color
equalized HSV V-channel

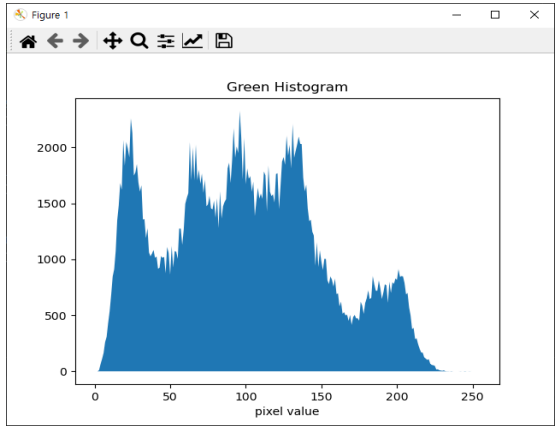
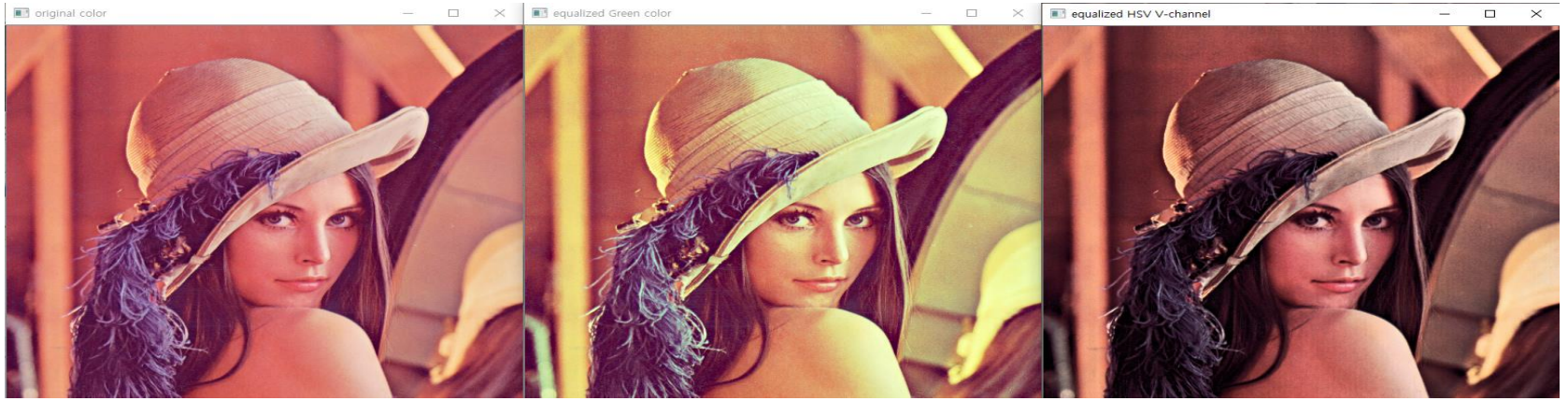


Figure 1
Green Histogram
2000
1500
1000
500
0
0
50
100
150
200
250
pixel value


original color
equalized Green color
equalized HSV V-channel

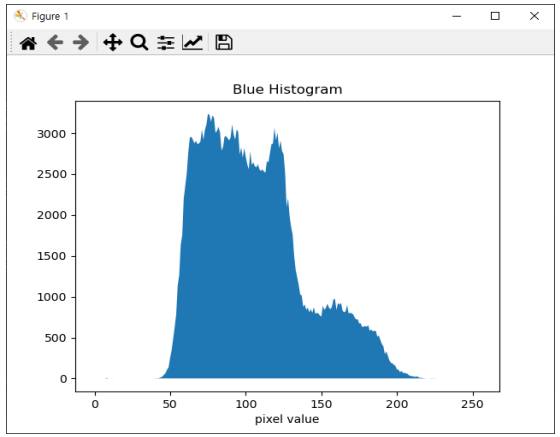
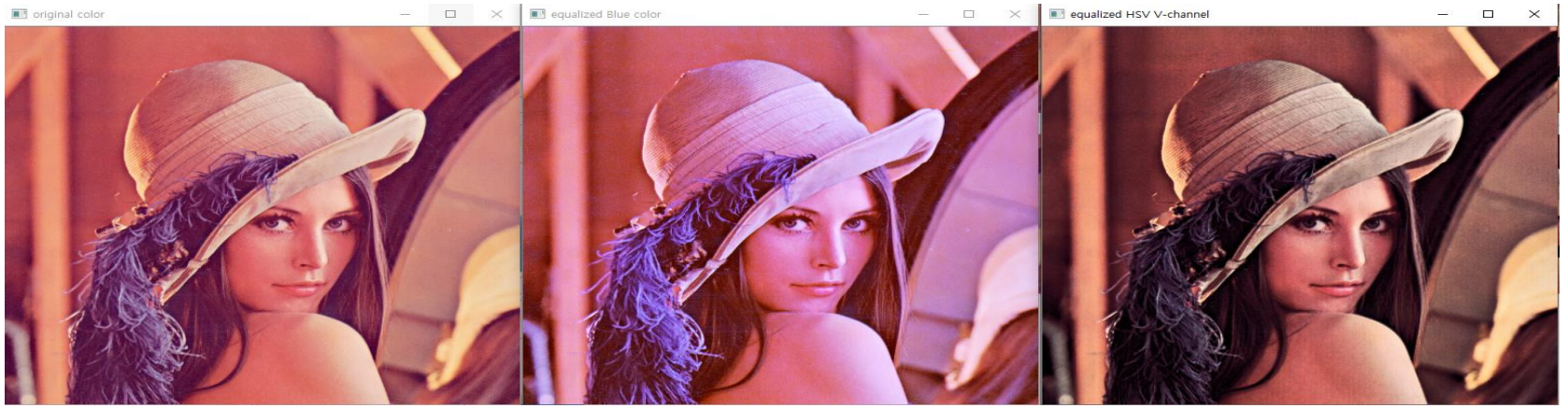


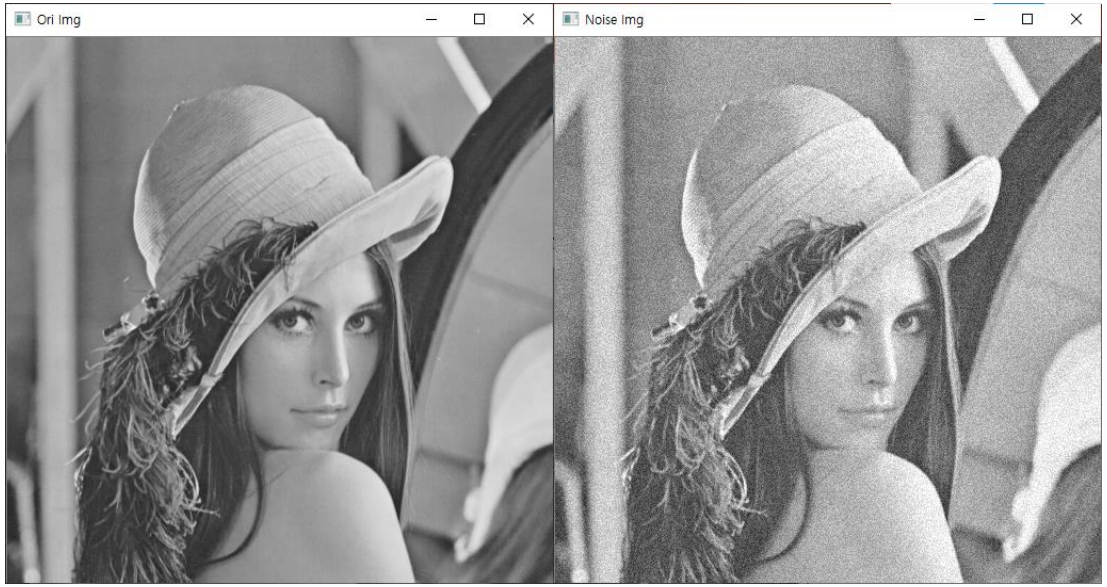
Figure 1
Blue Histogram
3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250
pixel value

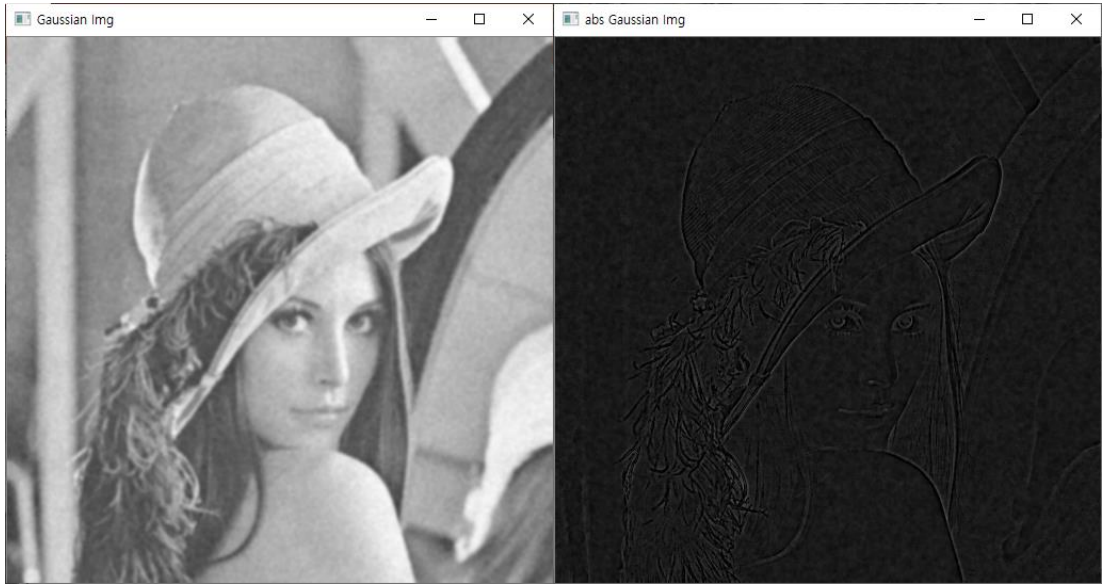

original color
equalized Blue color
equalized HSV V-channel

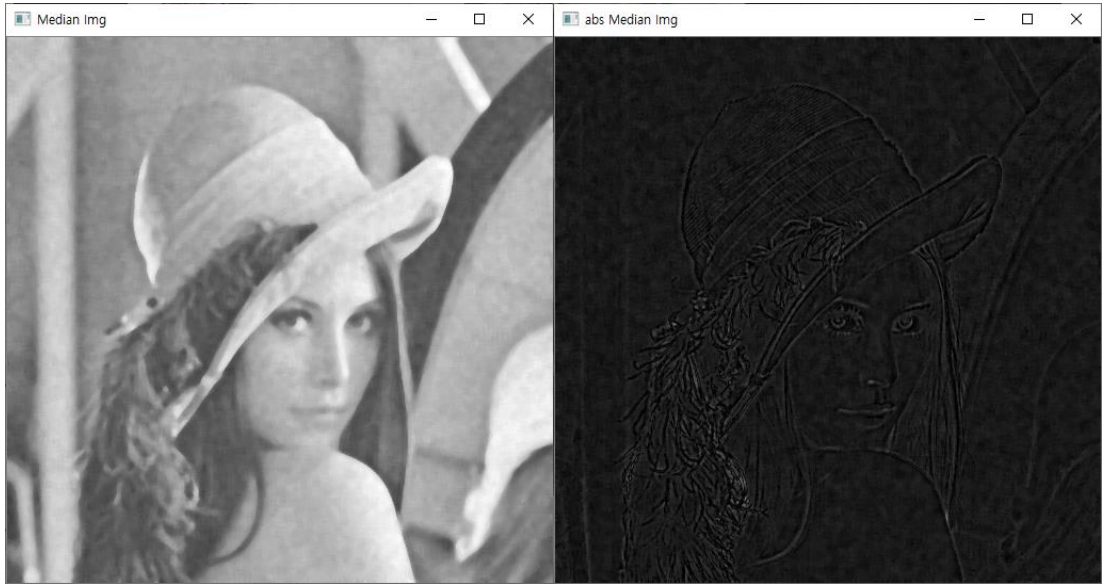
2. 공간 도메인 필터링

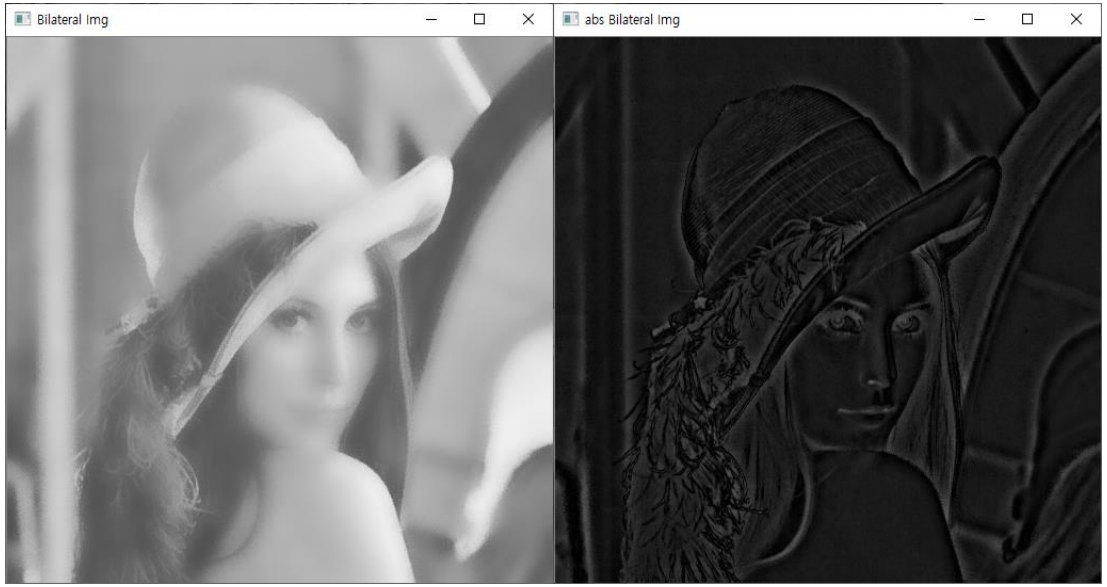
```python
import cv2
import numpy as np

image = cv2.imread( filename: '../../1.Data/Lena.png',  flags: 0).astype(np.float32) / 255

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

gauss_blur = cv2.GaussianBlur(noised,  ksize: (7, 7),  sigmaX: 0)

median_blur = cv2.medianBlur((noised * 255).astype(np.uint8),  ksize: 7)

bilateral = cv2.bilateralFilter(noised, -1,  sigmaColor: 0.3,  sigmaSpace: 10)

_abs_Gaussian = cv2.absdiff(image, gauss_blur)
_abs_Median = cv2.absdiff(image, median_blur.astype(np.float32) / 255)
_abs_Bilateral = cv2.absdiff(image, bilateral)

cv2.imshow( winname: 'Ori Img', image)
cv2.imshow( winname: 'Noise Img', noised)
cv2.imshow( winname: 'Gaussian Img', gauss_blur)
cv2.imshow( winname: 'Median Img', median_blur)
cv2.imshow( winname: 'Bilateral Img', bilateral)
cv2.imshow( winname: 'abs Gaussian Img', _abs_Gaussian)
cv2.imshow( winname: 'abs Median Img', _abs_Median)
cv2.imshow( winname: 'abs Bilateral Img', _abs_Bilateral)
cv2.waitKey()
```

2. 공간 도메인 필터링


Ori Img
Noise Img


Gaussian Img
abs Gaussian Img


Median Img
abs Median Img

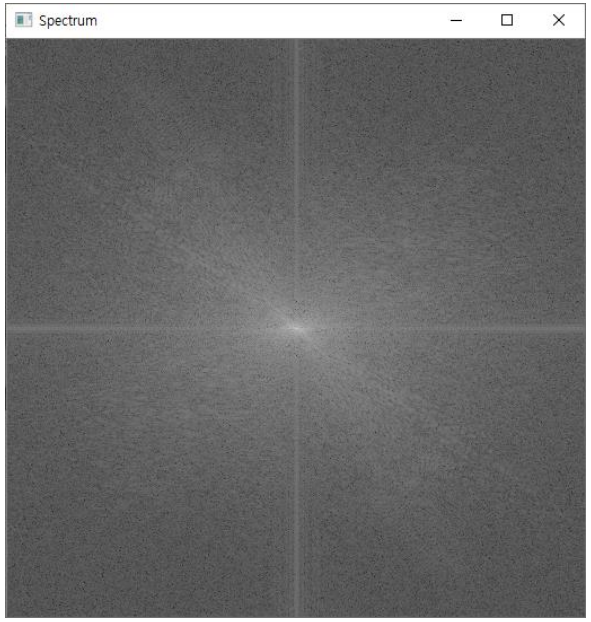

Bilateral Img
abs Bilateral Img

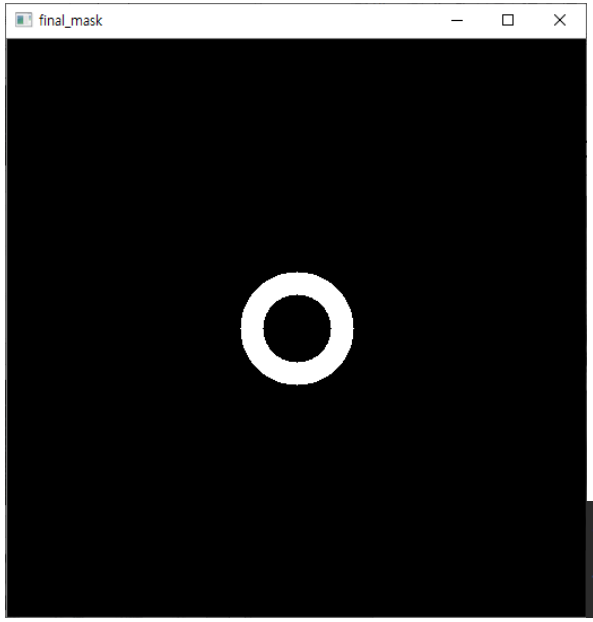
3. 주파수 도메인 필터링

```python
import cv2
import numpy as np

radius1 = int(input("첫 번째 원의 반지름을 입력하세요: "))
radius2 = int(input("두 번째 원의 반지름을 입력하세요: "))

# read input and convert to grayscale
img = cv2.imread( filename: '../../1.Data/Lena.png', cv2.IMREAD_GRAYSCALE)

# do dft saving as complex output
dft = np.fft.fft2(img, axes=(0,1))

# apply shift of origin to center of image
dft_shift = np.fft.fftshift(dft)

# generate spectrum from magnitude image (for viewing only)
mag = np.abs(dft_shift)
spec = np.log(mag) / 20

if radius1 > radius2:
    temp = radius1
    radius2 = radius1
    radius1 = temp
mask1 = np.zeros_like(img)
cy = mask1.shape[0] // 2
cx = mask1.shape[1] // 2
cv2.circle(mask1, center: (cx,cy), radius1, color: (255,255,255), -1)[0]

mask2 = np.zeros_like(img)
cy = mask2.shape[0] // 2
cx = mask2.shape[1] // 2
cv2.circle(mask2, center: (cx,cy), radius2, color: (255,255,255), -1)[0]

final_mask = mask2 - mask1

dft_shift_masked_final = np.multiply(dft_shift, final_mask) / 255

back_ishift = np.fft.ifftshift(dft_shift)
back_ishift_masked_final = np.fft.ifftshift(dft_shift_masked_final)

img_back = np.fft.ifft2(back_ishift, axes=(0,1))
img_filtered_final = np.fft.ifft2(back_ishift_masked_final, axes=(0,1))

img_back = np.abs(img_back).clip(0,255).astype(np.uint8)
img_filtered_final = np.abs(img_filtered_final).clip(0,255).astype(np.uint8)
```

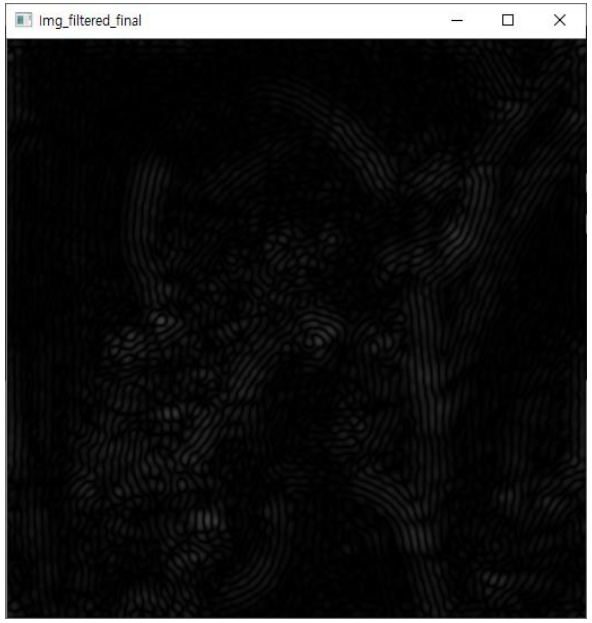
```python
cv2.imshow( winname: "Original", img)
cv2.imshow( winname: "Spectrum", spec)
cv2.imshow( winname: "final_mask", final_mask)
cv2.imshow( winname: "Original DFT/IFT trans", img_back)
cv2.imshow( winname: "Img_filtered_final", img_filtered_final)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

3. 주파수 도메인 필터링





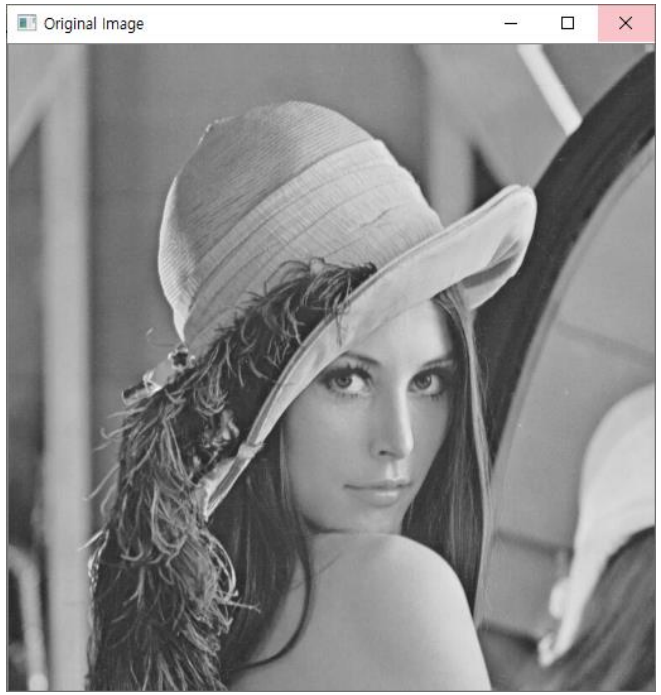

```
첫 번째 원의 반지름을 입력하세요: 30
두 번째 원의 반지름을 입력하세요: 50
```



4. 모폴로지 필터

```python
import cv2
import numpy as np

image = cv2.imread( filename: '../../1.Data/Lena.png', cv2.IMREAD_GRAYSCALE)

binary_method = input("이진화 방법 (otsu : OT / adaptive median : AM): ").upper()
morphology_operation = input("모폴로지 연산 (erosion : E, dilation : D, opening : O, closing : C): ").upper()
iterations = int(input("모폴로지 연산 적용 횟수 : "))

# 이진화
if binary_method == 'OT':
    _, binary_image = cv2.threshold(image, thresh: 0, maxval: 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
elif binary_method == 'AM':
    binary_image = cv2.adaptiveThreshold(image, maxValue: 255, cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY, blockSize: 11, C: 2)

# 모폴로지 연산 수행
kernel = np.ones( shape: (3, 3), np.uint8)
if morphology_operation == 'E':
    result = cv2.erode(binary_image, kernel, iterations=iterations)
elif morphology_operation == 'D':
    result = cv2.dilate(binary_image, kernel, iterations=iterations)
elif morphology_operation == 'O':
    result = cv2.morphologyEx(binary_image, cv2.MORPH_OPEN, kernel, iterations=iterations)
elif morphology_operation == 'C':
    result = cv2.morphologyEx(binary_image, cv2.MORPH_CLOSE, kernel, iterations=iterations)


# 결과 출력
cv2.imshow( winname: 'Original Image', image)
cv2.imshow( winname: 'Binary Image', binary_image)
cv2.imshow( winname: 'Morphology Result', result)
cv2.waitKey()
cv2.destroyAllWindows()
```

4. 모폴로지 필터




```
이진화 방법 (otsu : OT / adaptive median : AM): OT
모폴로지 연산 (erosion : E, dilation : D, opening : O, closing : C): E
모폴로지 연산 적용 횟수 : 5
```