

BMED318 Chapter 7 Hands-on Report

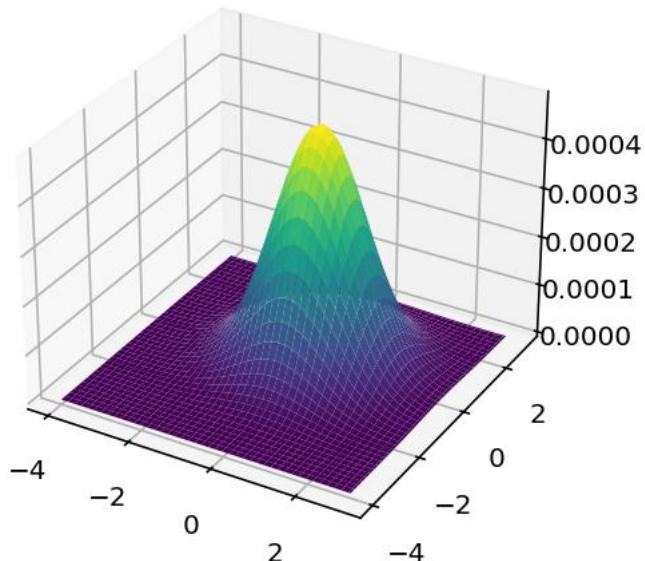
2023250054 윤현주

1. 입력 이미지 준비 (Prepare input image)

- a) matplotlib.pyplot.imread 를 사용하여 'skull.jpg' 파일을 읽어 im 배열에 저장 후
- b) skimage.transform.resize 를 사용하여 원본 이미지를 과제에서 지정한 226x187 크기로 리사이즈했습니다. 입력 이미지가 226x187 크기의 2D 배열 im 으로 준비되었습니다.

2. 공간 도메인 필터 준비 (Prepare a spatial domain filter)

- a) make_gaussian_mask2d 함수를 사용하여 $\sigma=1.0$ 을 갖는 8x8 크기의 가우시안 필터를 생성하여 lpf 배열에 저장했습니다.
- b) 8x8 필터는 3D 표면 플롯으로 표현하기에 너무 작아 보이지 않습니다. 따라서 $\sigma = 1.0$ 특성을 유지하고 128x128 크기로 정밀하게 샘플링한 lpf_128 배열을 생성하여 3D Surface Plot 으로 시각화했습니다.



3. 푸리에 스펙트럼 얻기 (Get Fourier spectrums)

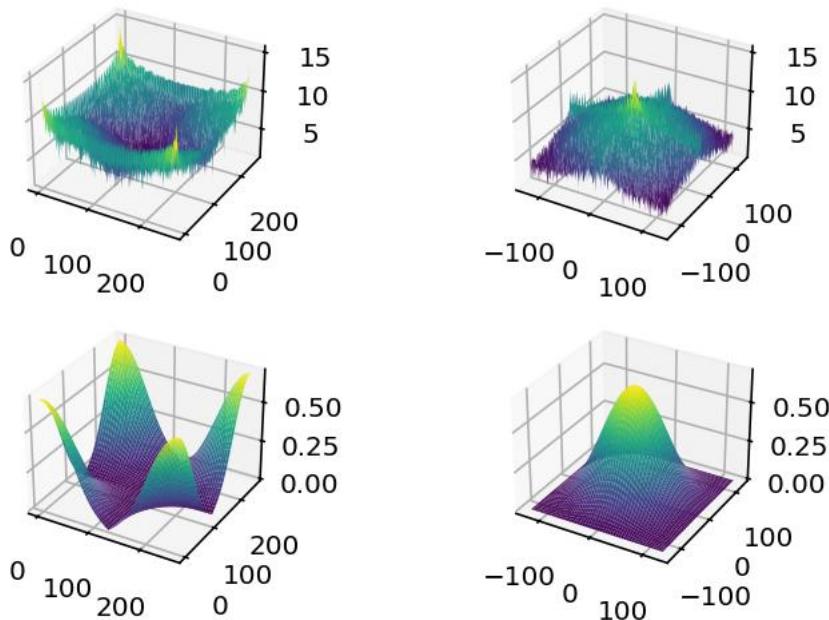
- a) 256x256 크기의 zero matrix imz 를 생성하고 좌측 상단에 226x187 크기의 원본 이미지 im 을 복사하여 제로 패딩을 수행했습니다.
- b) 256x256 크기의 zero matrix lpfz 를 생성하고, 좌측 상단에 8x8 크기의 가우시안 필터 lpf 를 복사하여 제로 패딩을 수행했습니다.

c) 주파수 영역에서의 곱셈은 공간 영역에서의 circular convolution에 해당합니다. 우리가 원하는 linear convolution 결과를 얻기 위해서는 신호가 순환하며 발생하는 time aliasing을 방지해야 합니다. 이를 위해 두 신호를 제로 패딩해야 합니다. 이미지 크기가 $(M, N) = (226, 187)$ 이고 필터 크기가 $(m, n) = (8, 8)$ 일 때 선형 컨볼루션 결과의 크기는 $(M+m-1, N+n-1) = (226+8-1, 187+8-1) = (233, 194)$ 가 됩니다. 따라서 순환 컨볼루션이 선형 컨볼루션과 같아지려면 최소한 $(233, 194)$ 크기 이상으로 두 신호를 제로 패딩해야 합니다. FFT 알고리즘이 2의 거듭제곱 크기일 때 가장 효율적으로 동작하므로 과제에서 사용한 $(256, 256)$ 가 가장 적절합니다.

d) numpy.fft.fft2 함수를 사용하여 제로 패딩된 이미지(imz)와 필터(lpfz)를 2D 푸리에 변환하여 각각 IMZ 와 LPFZ 에 저장했습니다.

e) numpy.fft.fftshift 함수를 사용하여 IMZ 와 LPFZ 의 [0,0] 인덱스에 위치한 DC 성분을 스펙트럼의 중앙으로 이동시켜 IMZ_CS 와 LPFZ_CS 에 저장했습니다.

f) 4 개의 스펙트럼(IMZ, LPFZ, IMZ_CS, LPFZ_CS)의 magnitude 를 $\text{np.log}(1 + \text{np.abs(array)})$ 를 통해 log scale 로 변환한 후 3D 표면 플롯으로 시각화했습니다.

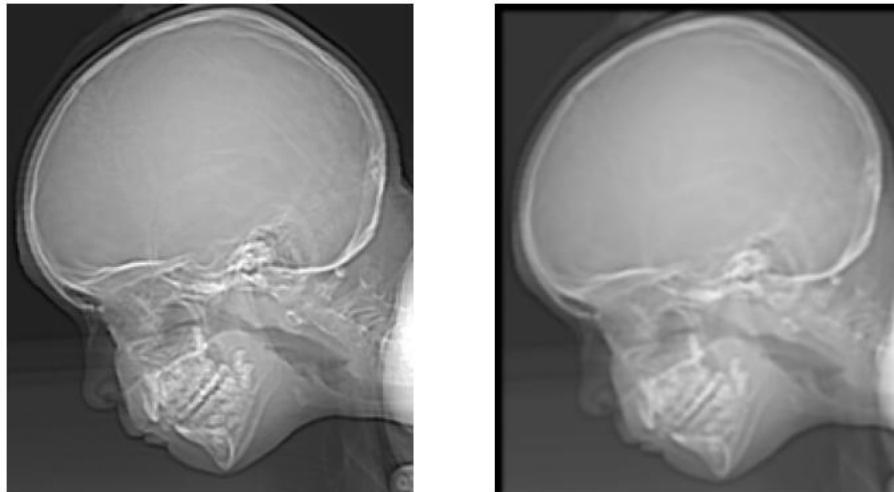


4. 주파수 도메인 필터링 - LPF

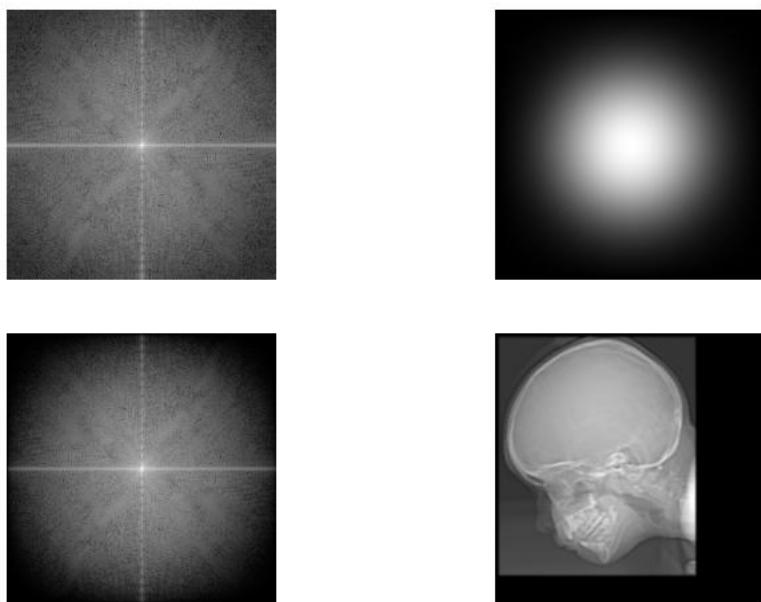
- 이미지 스펙트럼(IMZ)과 필터 스펙트럼(LPFZ)을 주파수 도메인에서 요소별로 곱셈했습니다.
- 곱셈 결과(IMZ_LPF)를 역푸리에 변환(ifft2)하여 공간 도메인으로 되돌렸습니다.
- IDFT 결과는 복소수이므로 np.real 을 사용해 실수부만 취했습니다.
- 제로 패딩되었던 256×256 크기에서 원본 이미지 크기인 226×187 만 잘라내어 최종 결과물 g_final 을 얻었습니다.

5. 두 개의 Figure로 결과를 시각화했습니다.

첫 번째 Figure: 원본 f 와 최종 결과 g



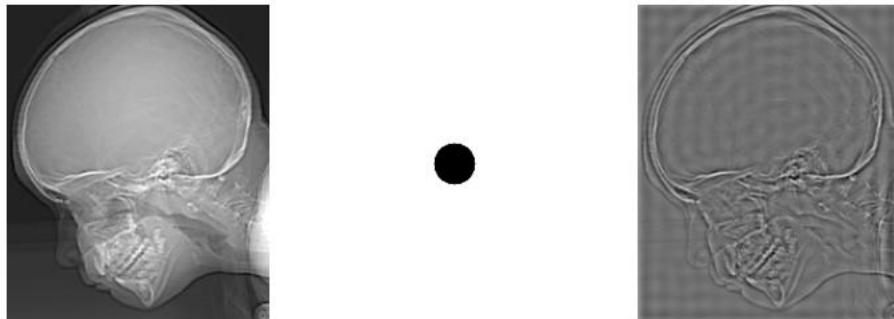
두 번째 Figure: 중간 단계 (F_p , H, HF_p , IDFT 결과)



5. 주파수 도메인 필터링: HPF (Frequency domain filtering: HPF)

4 번의 과정을 일반화한 freq_filter_H 함수를 정의했습니다. 이 함수는 입력 이미지와 주파수 도메인에서 직접 생성된 필터 마스크를 인자로 받아 필터링을 수행합니다. 필터 마스크 생성을 위해 np.meshgrid를 사용하여 256x256 크기의 중앙 정렬된 주파수 좌표계(U, V)와 중심에서의 거리(D)를 미리 계산했습니다.

a) Ideal HPF – Ideal HPF 마스크를 생성하여 freq_filter_H 함수에 적용했습니다. 여러 D_0 입니다.



D_0 가 10 이하로 너무 작으면 대부분의 저주파 성분이 차단되어 외곽선이 강해지지만 Ideal 필터 특유의 ringing artifact 가 이미지 전체에 심하게 나타납니다. D_0 가 50 처럼 너무 크면 저주파 성분이 너무 많이 통과되어 외곽선이 흐릿해지고 원본 이미지와 비슷해져 HPF의 의미가 약해집니다. $D_0 = 20$ 은 ringing artifact 가 다소 존재하지만 두개골과 얼굴의 주요 외곽선을 가장 선명하게 잘 추출한다고 판단했습니다.

b) Gaussian HPF – Gaussian HPF 마스크를 생성하여 적용했습니다. 여러 σ 값을 테스트한 후 선택한 Standard Deviation $\sigma = 15$ 입니다.



가우시안 필터는 Ideal 필터와 달리 경계가 부드러워 링잉 현상이 발생하지 않습니다.

$\sigma = 10$ 처럼 너무 작으면 필터의 차단 경계가 너무 넓고 부드러워 필요한 edge 성분까지 일부 제거되어 이미지가 전반적으로 어두워집니다. $\sigma = 40$ 처럼 너무 크면 저주파 차단 효과가 약해져 외곽선이 선명하지 않고 원본의 음영이 많이 남게 됩니다. $\sigma = 15$ 는 링잉 현상 없이 두개골과 얼굴의 외곽선을 비교적 선명하고 자연스럽게 강조한다고 판단했습니다.