

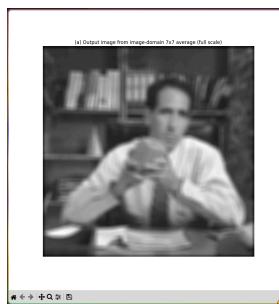
## Homework 4

### Bài 1.

#### (a): Tích Chập trong Miền Ảnh (Zero-Padding)

Phương pháp này sử dụng tích chập trực tiếp trong miền không gian để mô phỏng **tích chập tuyến tính**.

- **Zero-Padding:** Ảnh gốc `Xf` được đệm **3 hàng/cột zero** ở mỗi cạnh (`pad=3`) để thành kích thước  $262 \times 262$  (`Xpad`). Đây là độ đệm cần thiết cho kernel  $7 \times 7$  để kết quả có cùng kích thước  $256 \times 256$ .
- **Tích Chập:** Sử dụng `scipy.signal.convolve2d(Xpad, k, mode='valid')`. Tham số `mode='valid'` đảm bảo rằng kernel chỉ trượt qua các vùng đã được đệm, trả về kết quả  $256 \times 256$  của tích chập tuyến tính.
- **Kết quả:** `Y1a` là ảnh đã lọc.

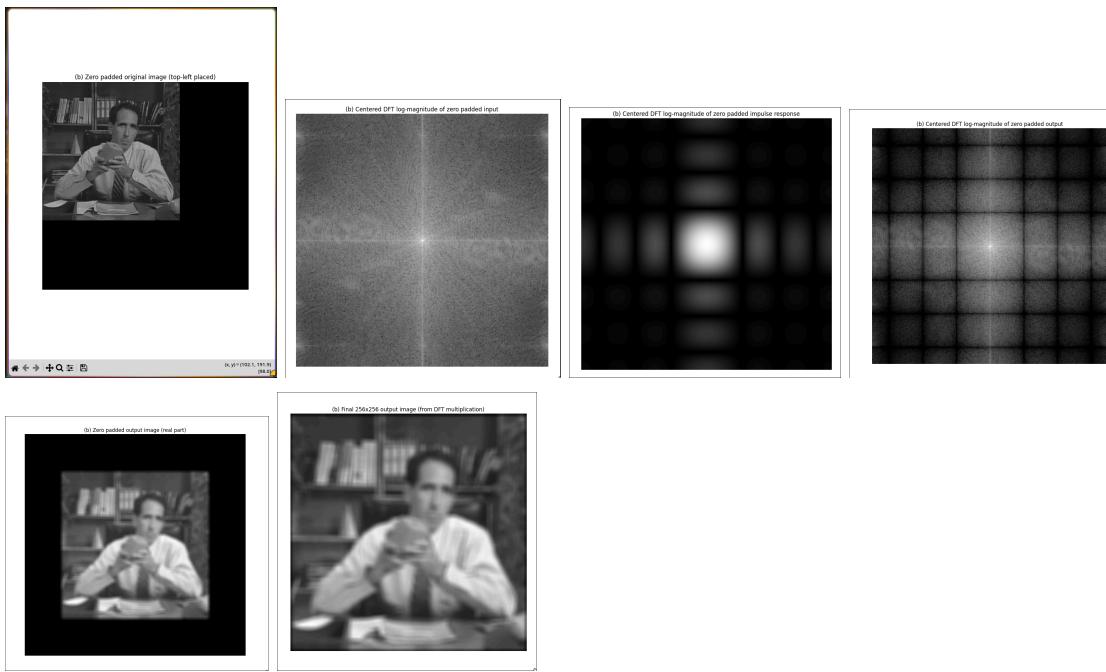


#### (b): Tích Chập Tuyến Tính DFT (Kiểu Example 3)

Phương pháp này thực hiện tích chập tuyến tính bằng cách nhân trong miền tần số, sử dụng **zero-padding** theo quy tắc: Kích thước  $\text{pad} \geq (\text{Nx} + \text{Nh} - 1)$ , với  $\text{Nx} = 256$  (ảnh) và  $\text{Nh} = 128$  (tổng kích thước kernel và padding). Mã sử dụng kích thước đệm **384x384**.

- **Tạo Đáp Ứng Xung H128:** Tạo ma trận  $128 \times 128$  với khối  $7 \times 7$  có giá trị  $1/49$  được đặt tại **góc trên bên trái** (chính xác là centered tại index  $p0=64$ ).

- **Zero-Padding:** Cả ảnh gốc (**Xf**) và đáp ứng xung (**H128**) đều được đệm zero vào ma trận  $384 \times 384$  (**ZPX**, **ZPH**) và đặt tại **góc trên bên trái** ( $0:N, 0:N$ ).
- **Lọc:**
  - Tính FFT:  $X \sim = \text{FFT2}(ZPX)$ ,  $H \sim = \text{FFT2}(ZPH)$ .
  - Nhân điểm:  $Y \sim = X \sim \times H \sim$ .
  - Tính IFFT:  $ZPY = \text{IFFT2}(Y \sim)$ .
- **Cắt Kết Quả (Crop):** Ảnh đã lọc  $256 \times 256$  (**Y1b**) được trích xuất từ vị trí  $64:320, 64:320$  của **ZPY** (tương đương với  $65:320$  theo MATLAB 1-based). Vị trí này đảm bảo lấy được phần kết quả tích chập tuyến tính không bị ảnh hưởng bởi cạnh zero.
- **So sánh:** So sánh kết quả **Y1b** với **Y1a** bằng cách tính độ lệch tối đa.

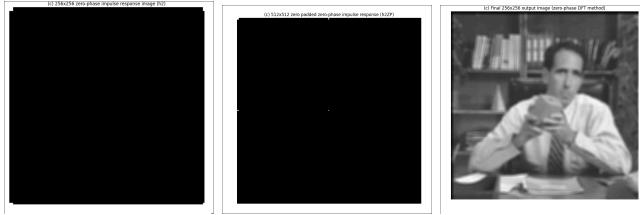


### (c): Zero-Phase DFT (Kiểu Example 5)

Phương pháp này cũng thực hiện tích chập tuyến tính nhưng sử dụng kỹ thuật **Zero-Phase Impulse Response** (Đáp ứng xung zero-phase) và kích thước đệm  $512 \times 512$ .

- **Tạo H:** Tạo ma trận  $256 \times 256$  với khối  $7 \times 7$  có giá trị  $1/49$  được đặt tại **vị trí dịch tâm** (center-shifted) tại index  $126:133$  (gần vị trí  $128, 128$ ).
- **Zero-Phase H2:** Sử dụng `np.fft.fftshift(H)` để dịch tâm kernel, tạo ra đáp ứng xung zero-phase H2.
- **Zero-Padding:** Cả ảnh gốc (**Xf**) và H2 đều được đệm zero vào ma trận  $512 \times 512$  (**ZPXc**, **ZPH2**) và đặt tại **góc trên bên trái**.

- **Lọc:**
  - Tính FFT:  $X \sim c = \text{FFT2}(ZP\bar{X}c)$ ,  $H \sim c = \text{FFT2}(ZPH2)$ .
  - Nhân điểm:  $Y \sim c = X \sim c \times H \sim c$ .
  - Tính IFFT:  $ZPYc = \text{IFFT2}(Y \sim c)$ .
- **Cắt Kết Quả (Crop):** Ảnh đã lọc  $256 \times 256$  (`Y1c`) được trích xuất từ vị trí **góc trên bên trái**  $0:256, 0:256$  của `ZPYc`.
- **So sánh:** So sánh kết quả `Y1c` với `Y1a` bằng cách tính độ lệch tối đa.



## Bài 2.

### 1. Utilities (Các hàm tiện ích)

Phần này định nghĩa các hàm cơ bản để đọc, hiển thị, và đánh giá ảnh:

- **`read_bin_image`:** Đọc ảnh từ tệp nhị phân (`.bin`) và định hình lại thành mảng NumPy  $256 \times 256$  kiểu `float64`.
- **`fullscale_display_im`:** Chuẩn hóa (scale) ảnh tuyến tính về phạm vi **0 đến 255** (kiểu `uint8`) để hiển thị tối ưu.
- **`show`:** Hiển thị ảnh bằng `matplotlib.pyplot.imshow`.
- **`log_mag_spectrum`:** Tính và hiển thị **phổ logarit biên độ (log magnitude spectrum)** đã dịch tâm (centered) của ảnh (sử dụng `np.fft.fftshift`).
- **`mse`:** Tính **Sai số Bình phương Trung bình (Mean Squared Error - MSE)** giữa hai ảnh.  

$$\text{MSE}(a,b) = M \times N \sum_{i,j} (a_{i,j} - b_{i,j})^2$$
- **`isnr`:** Tính **Tỉ số Tín hiệu trên Nhiều Cải thiện (Improvement in Signal-to-Noise Ratio - ISNR)**, đo lường mức độ bộ lọc giảm nhiễu so với ảnh nhiễu ban đầu.  

$$\text{ISNR} = 10 \log_{10} \left( \frac{\sum (\text{orig} - \text{filtered})^2}{\sum (\text{orig} - \text{noisy})^2} \right)$$

### (a) Load Images

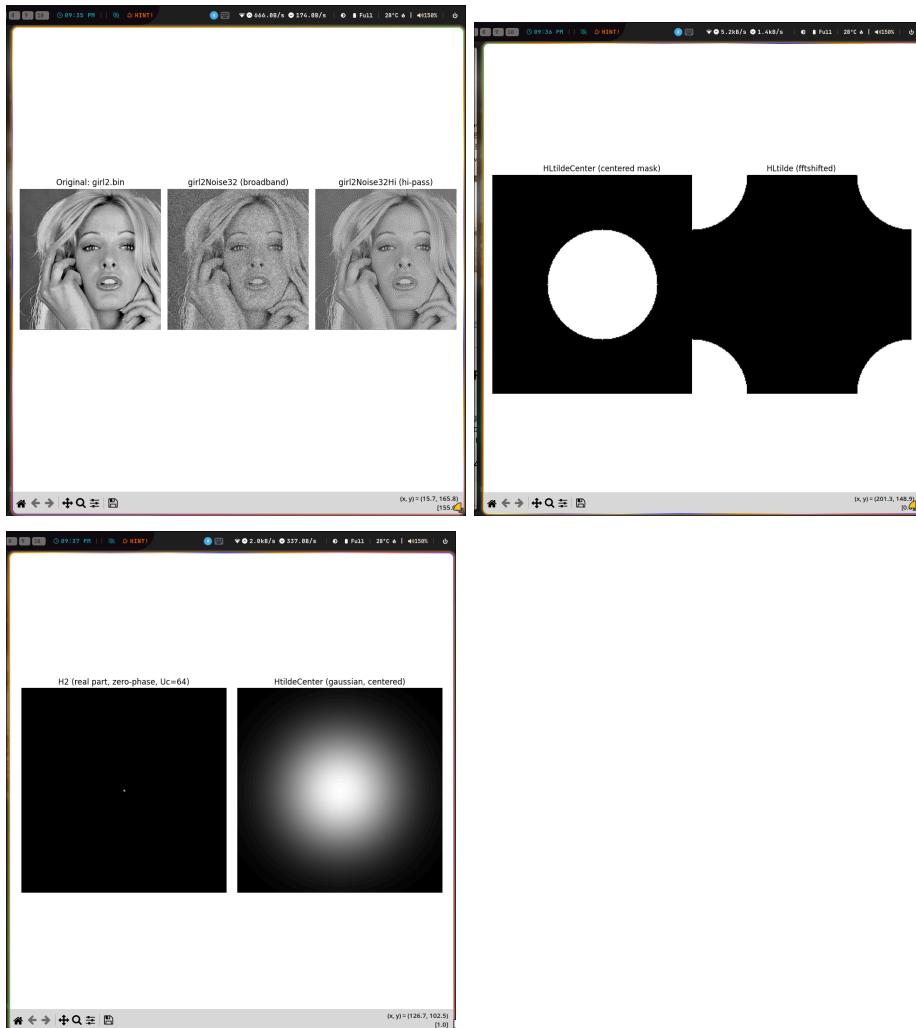
- Tải ba ảnh  $256 \times 256$  float:
  1. **f**: Ảnh gốc (`girl12.bin`).

2. **x\_broad**: Ảnh nhiễu băng rộng (Broadband Noise) (`girl2Noise32.bin`).
  3. **x\_high**: Ảnh nhiễu thông cao (High-Pass Noise) (`girl2Noise32Hi.bin`).
- **Hiển thị** các ảnh gốc đã được chuẩn hóa.
  - **Tính MSE** ban đầu giữa ảnh gốc (`f`) và hai ảnh nhiễu.
- 

## (b) Ideal Isotropic LPF (Bộ lọc thông thấp lý tưởng đằng hướng)

Phần này áp dụng Bộ lọc thông thấp lý tưởng  $U_{cutoff}=64$  bằng **phương pháp tích chập vòng** (**circular convolution**) trong miền tần số.

- **Tạo bộ lọc  $H \sim L$ :**
  1. Tạo lưới tọa độ tần số  $U, V$  dịch tâm từ  $-128$  đến  $127$ .
  2. Tạo mặt nạ **Hình tròn lý tưởng** centered  $H \sim L, Center$ : giá trị là  $1$  nếu  $U^2 + V^2 \leq U_{cutoff}$  ( $64$ ), ngược lại là  $0$ .
  3. Dịch chuyển tần số thấp về góc  $H \sim L = fftshift(H \sim L, Center)$ .
- **Hàm `apply_pointwise_dft_filter`**: Thực hiện tích chập vòng bằng cách:
  1. Tính FFT của ảnh ( $X = FFT2(image)$ ).
  2. Nhân điểm (pointwise multiply)  $Y = X \times H \sim L$ .
  3. Tính IFFT của kết quả ( $y = IFFT2(Y).real$ ).
- **Áp dụng và Đánh giá**: Bộ lọc được áp dụng cho cả ba ảnh. Kết quả được hiển thị, MSE và ISNR được tính toán.



## (c, d) Gaussian LPF (Bộ lọc thông thấp Gaussia)

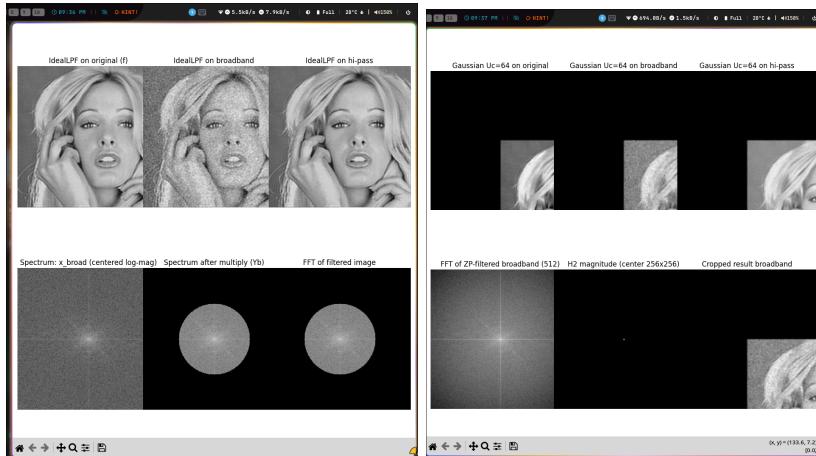
Phần này sử dụng Bộ lọc thông thấp Gaussian bằng **phương pháp tích chập tuyen tính** (**linear convolution**) thông qua **zero-phase filtering** với zero-padding kích thước  $512 \times 512$  (theo Example 4).

### Gaussian LPF Ucutoff=64 (c)

- **Hàm gaussian\_zero\_phase\_impulse:**
  1. Tính độ lệch chuẩn  $\Sigma H$  cho Gaussian LPF:  $\Sigma H = 0.19 \times N / U_{cutoff}$ .
  2. Tạo hàm truyền đạt Gaussian centered  $H \sim Center$ :  

$$H \sim Center(U, V) = \exp(-N^2 2 \pi^2 H^2 (U^2 + V^2))$$
  3. Tính đáp ứng xung **zero-phase**  $H2$ :  $H2 = \text{fftshift}(\text{IFFT2}(\text{fftshift}(H \sim Center)))$ .
- **Hàm apply\_zero\_phase\_filter:** Thực hiện tích chập tuyen tính:

1. Tạo ma trận  $512 \times 512$  cho ảnh (`ZP_img`) và đáp ứng xung (`ZP_H2`), đặt ảnh/ H2 vào góc trên bên trái ( $0:256, 0:256$ ).
  2. Tính FFT của cả hai (`Xzp, Hzp`).
  3. Nhân điểm  $Yzp = Xzp \times Hzp$ .
  4. Tính IFFT ( $yzp = \text{IFFT2}(Yzp).\text{real}$ ).
  5. **Cắt (crop)** kết quả về kích thước gốc  $256 \times 256$ .
- **Áp dụng và Đánh giá:** Bộ lọc được áp dụng và MSE, ISNR được tính toán.



## 4.2. Gaussian LPF Ucutoff=77.5 (d)

- Lặp lại quy trình ở Phần (c) với **Ucutoff=77.5** để tìm Ucutoff tối ưu hơn cho việc khử nhiễu, sau đó tính MSE và ISNR để so sánh.

