

Homework 3

1. TẠO ẢNH PHÚC I_1 VÀ BIẾN ĐỔI FOURIER (DFT)

Mục đích

Xây dựng ảnh phức I_1 có tần số không gian xác định ($u_0 = v_0 = 2$ cpi), hiển thị phần thực và ảo của ảnh, và tính DFT để quan sát phổ tần số.

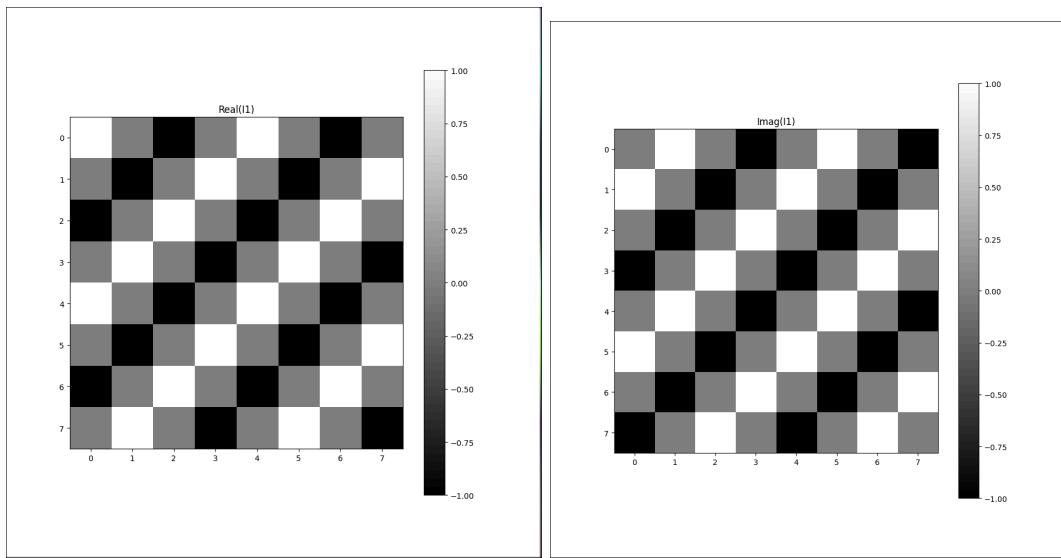
Lý thuyết

- Ảnh I_1 được định nghĩa:
$$I_1(m,n) = e^{j \frac{2\pi}{N}(u_0 m + v_0 n)} I_1(m,n) = e^{j \frac{2\pi}{N}(u_0 m + v_0 n)}$$
- Trong đó:
 - u_0, v_0 : tần số không gian theo hướng ngang và dọc (cycles per image).
 - $N=8N = 8N=8$: kích thước ảnh.
- Biến đổi Fourier rời rạc (DFT) được dùng để biểu diễn ảnh trong miền tần số:
$$F(u,v) = \sum_{m,n} I(m,n) e^{-j2\pi(um+vn/N)} F(u,v) = \sum_{m,n} I(m,n) e^{-j2\pi(um+vn/N)}$$
- Hàm `fftshift()` được dùng để đưa tần số gốc về trung tâm phổ.

Các bước thực hiện

1. Tạo ma trận tọa độ hàng – cột ($0 \rightarrow 7$) bằng `np.meshgrid`.
2. Tính ảnh phức I_1 với biểu thức $e^{j2\pi(u_0 m + v_0 n)/N}$ $e^{j2\pi(u_0 m + v_0 n)/N}$.
3. Hiển thị phần **thực** và **ảo** của I_1 bằng thang xám.
4. Tính **DFT** và **dịch tâm phổ** bằng
`np.fft.fftshift(np.fft.fft2(I1))`.

5. In ma trận phần thực và ảo của DFT dưới dạng mảng số thực.



Output:

```
@minhminh ➔ python3 homework4_1.py
Re[DFT(I1)]:
[[ 0. -0. -0. -0.  0. -0.  0. -0.]
 [-0. -0.  0.  0.  0.  0.  0.  0.]
 [-0.  0.  0. -0.  0. -0.  0.  0.]
 [-0. -0. -0.  0.  0. -0. -0.  0.]
 [ 0.  0.  0. -0.  0. -0.  0.  0.]
 [-0.  0. -0. -0.  0.  0. -0. -0.]
 [ 0.  0.  0. -0. -0. -0.  64.  0.]
 [-0.  0.  0.  0.  0.  0.  0. -0.]]
Im[DFT(I1)]:
[[ 0.  0.  0.  0. -0.  0. -0.]
 [ 0. -0. -0. -0.  0.  0.  0.]
 [-0. -0.  0. -0.  0. -0.  0.]
 [-0. -0. -0. -0.  0.  0.  0.]
 [ 0.  0.  0.  0. -0.  0. -0.]
 [-0. -0. -0.  0. -0.  0.  0.]
 [ 0.  0.  0.  0.  0. -0.  0.]
 [-0.  0. -0. -0. -0.  0.  0.]]
```

2. TẠO ẢNH PHỨC I_2 NGƯỢC PHA

Mục đích

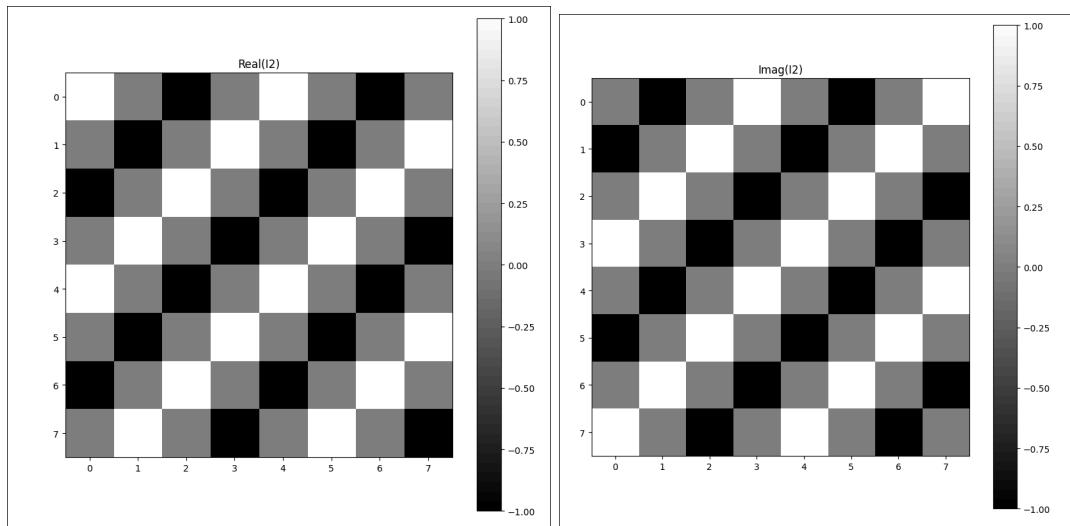
Tạo ảnh I_2 là ảnh có pha ngược với I_1 và phân tích DFT của nó.

Lý thuyết

- Ảnh được định nghĩa:
 $I_2(m,n) = e^{-j2\pi N(u_0m+v_0n)}$
 $I_2(m,n) = e^{-jN2\pi(u_0m+v_0n)}$
- I_2 là **liên hợp phức (complex conjugate)** của I_1 .
- Phổ tần số của I_2 là **phản chiếu** của I_1 quanh gốc tần số.

Các bước thực hiện: Tương tự bài 1 nhưng với biến đổi ngược pha

1. Khởi tạo giá trị $u_0=v_0=2$, $u_0=v_0=2$.
2. Tạo ma trận tọa độ (m,n) và tính $I_2=e^{-j2\pi(u_0m+v_0n)/N}$
 $I_2=e^{-j2\pi(u_0m+v_0n)/N}$.
3. Hiển thị phần thực và ảo của I_2 .
4. Tính và in DFT sau khi dịch tâm bằng `fftshift`.



3. ẢNH COSINE I_3

Mục đích

Tạo ảnh thực I_3 từ I_1 và I_2 , hiển thị và phân tích phổ Fourier.

Lý thuyết

- Ảnh định nghĩa:

$$I_3(m,n) = \cos(2\pi N(u_0m + v_0n))$$
$$I_3(m,n) = \cos\left(\frac{2\pi}{N}(u_0m + v_0n)\right)$$

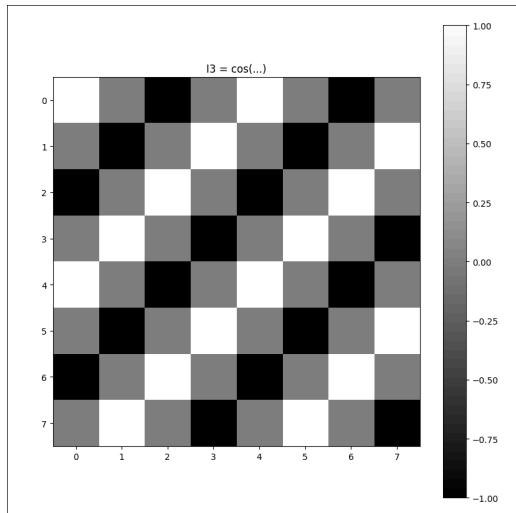
- Theo công thức Euler:

$$\cos(x) = e^{jx} + e^{-jx}$$
$$\cos(x) = \frac{e^{jx} + e^{-jx}}{2}$$
$$\cos(x) = \frac{1}{2} + \frac{1}{2} \cos(2x)$$
$$\rightarrow I_3 = (I_1 + I_2)/2$$

- Do đó, I_3 là **ảnh thực**, chỉ chứa thành phần **cosine**.

Các bước thực hiện

- Dùng công thức cosine để tạo ma trận I_3 .
- Hiển thị ảnh I_3 ở dạng thang xám (8 bpp).
- Tính DFT bằng **fft2** và hiển thị phần thực, ảo của phổ.
- Tính và in ra DFT (8×8).



Output:

```

Tue 23 Sep - 13:53 ~ ~/workspace/ImageProcess
@minhminh> python3 homework4_3.py
Re[DFT(I3)]:
[[ 0. -0. -0. -0.  0. -0. -0. -0.]
 [-0. -0.  0.  0.  0.  0.  0.  0.]
 [-0.  0. 32. -0.  0. -0.  0.  0.]
 [-0. -0. -0.  0.  0. -0. -0.  0.]
 [ 0.  0.  0. -0.  0.  0.  0.  0.]
 [-0.  0. -0. -0.  0.  0. -0. -0.]
 [-0.  0. -0.  0. -0. 32.  0.  0.]
 [-0.  0.  0.  0.  0.  0. -0. -0.]]
Im[DFT(I3)]:
[[ 0.  0.  0.  0. -0. -0. -0. -0.]
 [ 0. -0. -0. -0.  0.  0.  0.  0.]
 [ 0. -0.  0. -0.  0. -0.  0. -0.]
 [ 0. -0. -0. -0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0.  0. -0. -0. -0.]
 [-0. -0. -0.  0. -0.  0.  0.  0.]
 [-0.  0.  0. -0.  0. -0.  0.  0.]
 [-0.  0. -0. -0. -0.  0.  0.  0.]]

```

4. ẢNH SINE I₄

Mục đích

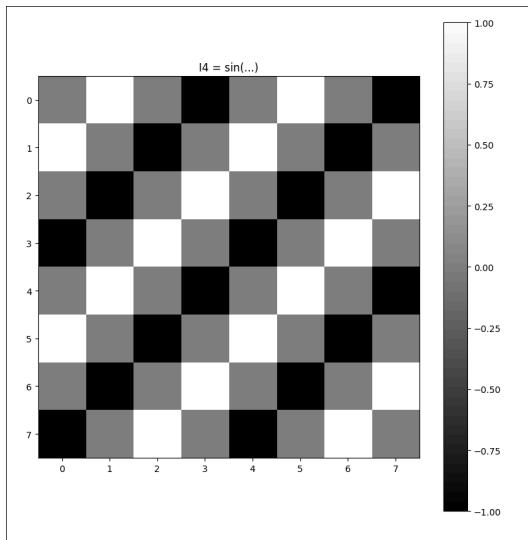
Tạo ảnh I₄ là dạng sóng sine và phân tích phổ Fourier.

Lý thuyết

- Ảnh định nghĩa:
 $I_4(m,n) = \sin(2\pi N(u_0m + v_0n))$
 $I_4(m,n) = \sin\left(\frac{2\pi}{N}(u_0m + v_0n)\right)$
- Dựa vào công thức Euler:
 $\sin(x) = e^{jx} - e^{-jx}$
 $\sin(x) = \frac{e^{jx} - e^{-jx}}{2j}$
 $\sin(x) = 2je^{jx}/2 - e^{-jx}/2$
 $I_4 = -j(I_1 - I_2)/2$
 $I_4 = -j(I_1 - I_2)/2$
- I₄ biểu diễn thành phần **pha lệch 90°** so với cosine.

Các bước thực hiện

- Tính I₄ bằng công thức sin.
- Hiển thị ảnh bằng thang xám.
- Thực hiện DFT và hiển thị phần thực và ảo.



Output:

```
@minhminh ➤ python3 homework4_4.py
Re[DFT(I4)]:
[[ 0.  0.  0.  0.  0.  0.  0.  0.]
 [-0. -0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0. -0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0. -0. -0.  0.  0.  0.]
 [-0.  0.  0.  0.  0.  0.  0.  0.]
 [ 0.  0.  0.  0. -0. -0.  0.  0.]
 [ 0.  0.  0.  0.  0.  0. -0.  0.]
 [-0.  0.  0.  0.  0.  0.  0. -0.]]
Im[DFT(I4)]:
[[ 0.  0.  0. -0.  0.  0. -0. -0.]
 [ 0. -0.  0.  0.  0.  0. -0. -0.]
 [ 0.  0. 32. -0. -0. -0.  0.  0.]
 [-0. -0. -0.  0. -0.  0.  0. -0.]
 [ 0.  0. -0. -0.  0.  0.  0. -0.]
 [ 0.  0. -0. -0.  0. -0.  0.  0.]
 [-0. -0. -0.  0.  0.  0. -32. -0.]
 [-0.  0.  0. -0. -0. -0. -0.  0.]]
```

5. ẢNH COSINE TẦN SỐ KHÁC I₅

Mục đích

Tạo ảnh cosine với tần số không nguyên $u_1=v_1=1.5$ $cpi_{u_1} = v_1 = 1.5$, $cpi_{u_1}=v_1=1.5cpi$ để quan sát hiện tượng nhiễu xạ (aliasing).

Lý thuyết

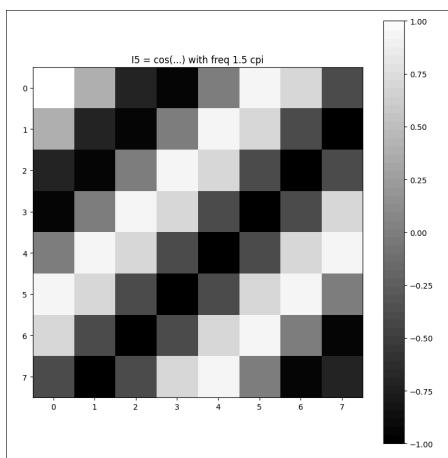
- Ảnh định nghĩa:

$$I_5(m,n)=\cos(2\pi N(u_1m+v_1n)) \quad I_5(m,n) = \cos\left(\frac{2\pi}{N}(u_1m + v_1n)\right)$$

- Khi tần số không phải số nguyên, dạng sóng không khớp hoàn toàn với kích thước ảnh → phô DFT **không tập trung** mà **bị trải rộng**.

Các bước thực hiện

1. Chọn $u_1=v_1=1.5$, $u_{-1}=v_{-1}=1.5$.
2. Tạo ma trận I_5 theo công thức cosine.
3. Hiển thị ảnh.
4. Tính DFT và in phần thực, ảo.
5. So sánh phô với các bài I_1-I_4 .



Output:

```

Re[DFT(I5)]: [
[ 0.5535  0.4335 -0.6131  2.9028  2.         2.9028 -0.6131  0.4335]
[ 0.4335 -0.1397 -2.7422  4.8284  2.2688  2.         0.6488  0.7023]
[ -0.6131 -2.7422 -11.6569 13.5706  4.6131  2.1796  2.         0.6488]
[ 2.9028  4.8284 13.5706 -11.8603 -3.262   -2.3592  2.1796  2.        ]
[ 2.         2.2688  4.6131 -3.262   -1.2398 -3.262   4.6131  2.2688]
[ 2.9028  2.         2.1796 -2.3592 -3.262   -11.8603 13.5706  4.8284]
[ -0.6131  0.6488  2.         2.1796  4.6131  13.5706 -11.6569 -2.7422]
[ 0.4335  0.7023  0.6488  2.         2.2688  4.8284 -2.7422 -0.1397]]
Im[DFT(I5)]: [
[ 0.         0.6488  2.6131 -2.1796  0.         2.1796 -2.6131 -0.6488]
[ 0.6488  1.2977  3.262   -1.5307  0.6488  2.8284 -1.9643  0.        ]
[ 2.6131  3.262   5.2263  0.4335  2.6131  4.7927  0.         1.9643]
[ -2.1796 -1.5307  0.4335 -4.3592 -2.1796  0.         -4.7927 -2.8284]
[ 0.         0.6488  2.6131 -2.1796  0.         2.1796 -2.6131 -0.6488]
[ 2.1796  2.8284  4.7927 -0.         2.1796  4.3592 -0.4335  1.5307]
[ -2.6131 -1.9643  0.         -4.7927 -2.6131 -0.4335 -5.2263 -3.262 ]
[ -0.6488 -0.         1.9643 -2.8284 -0.6488  1.5307 -3.262   -1.2977]]

```

6. PHÂN TÍCH DFT CỦA ẢNH GRayscale THỰC

Mục đích

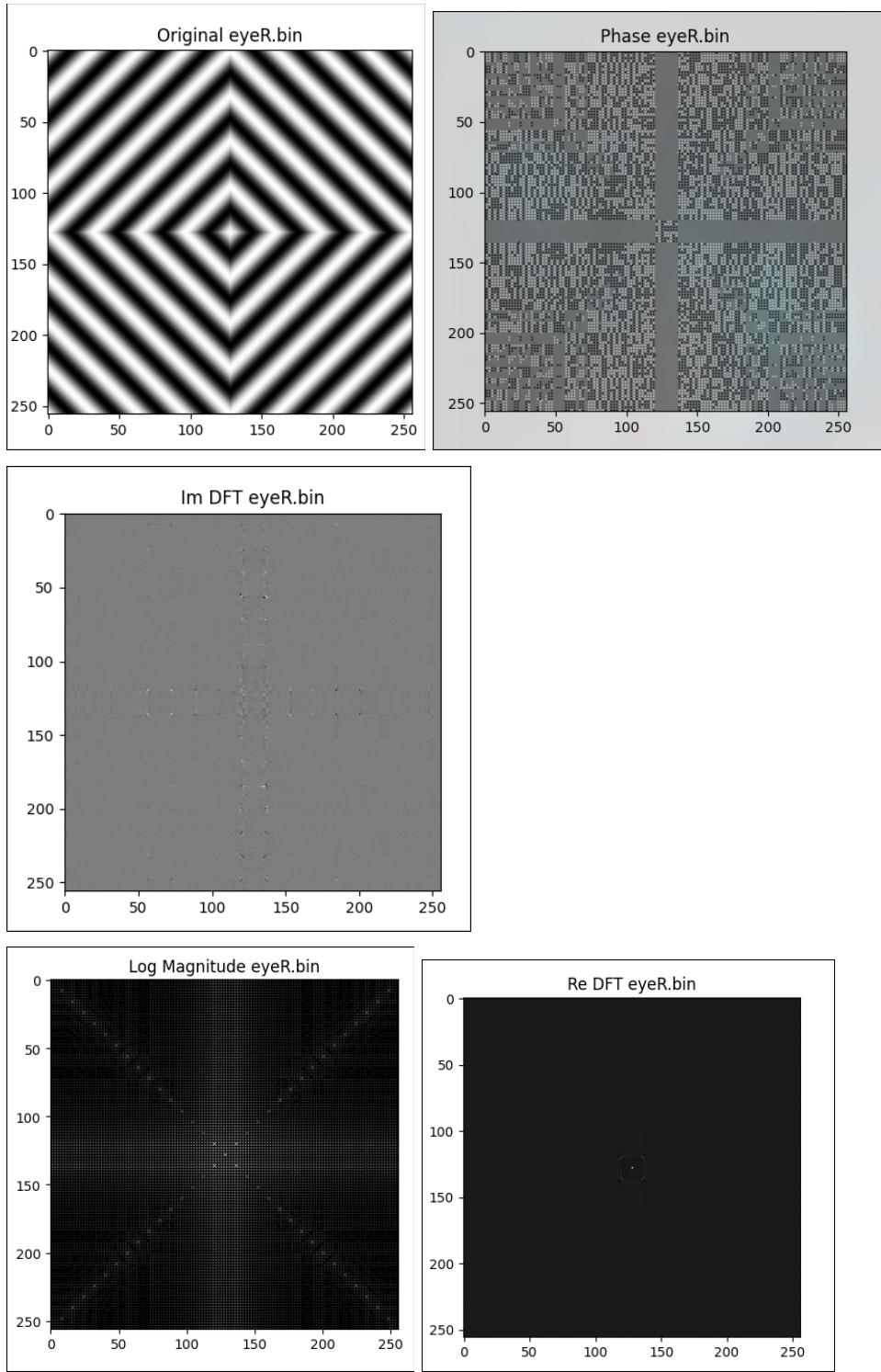
Tính và hiển thị các thành phần Fourier của ảnh thật: phần thực, phần ảo, biên độ logarit, và pha.

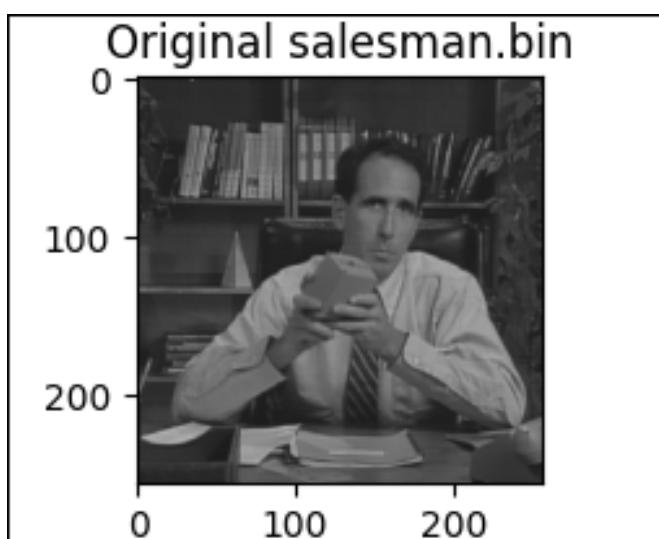
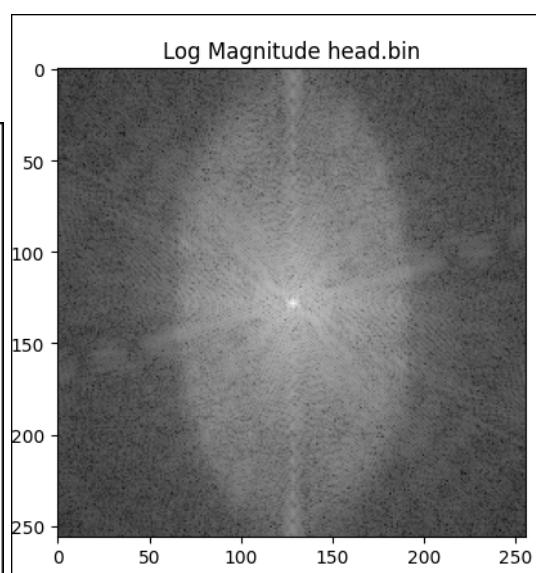
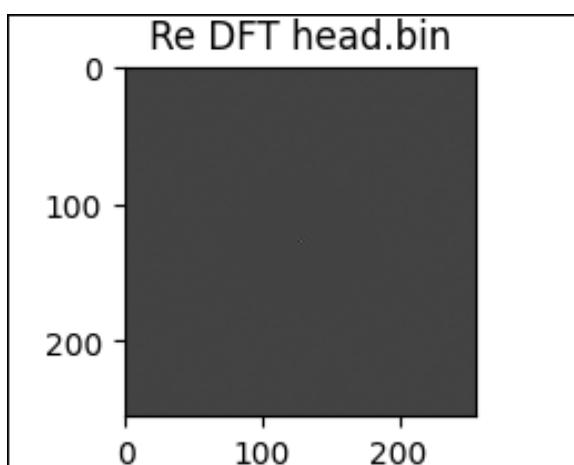
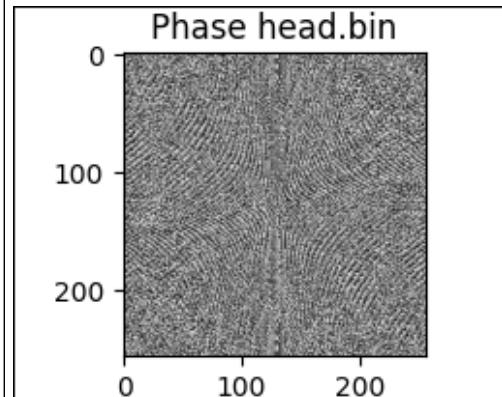
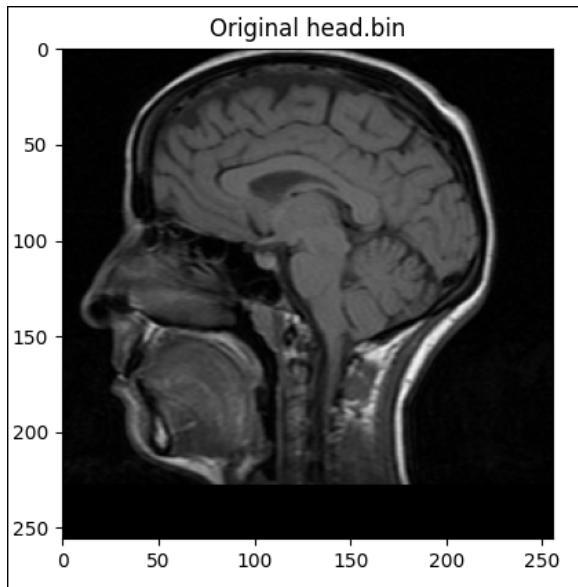
Lý thuyết

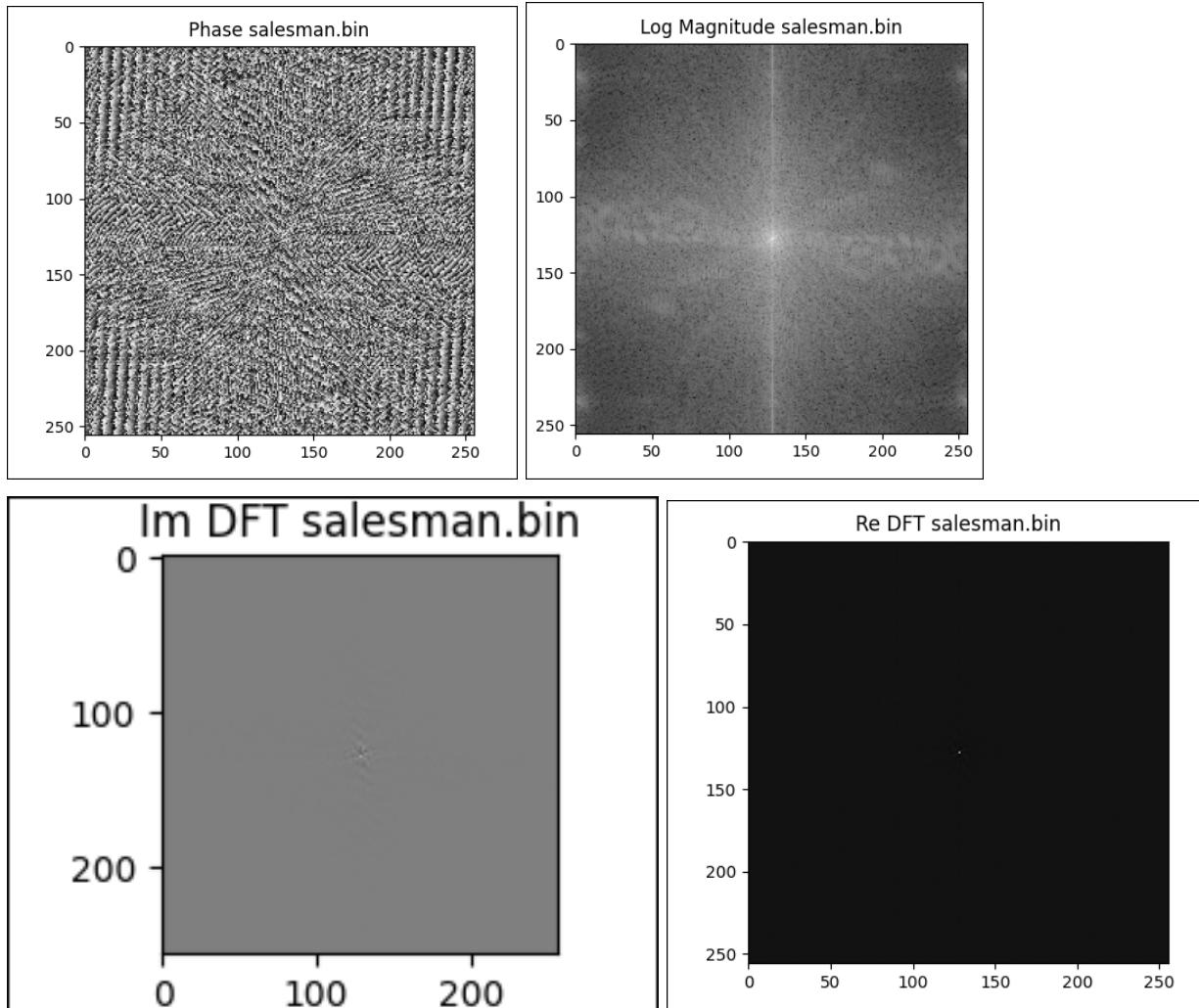
- Biến đổi Fourier 2D cho phép phân tách ảnh thành thành phần tần số.
- Thành phần:
 - **Real(F)**: phần chẵn của tín hiệu.
 - **Img(F)**: phần lẻ của tín hiệu.
 - **|F|**: biên độ tần số.
 - **arg(F)**: pha.
- Dùng **log(1+|F|)** để làm nổi bật chi tiết tần số nhỏ.

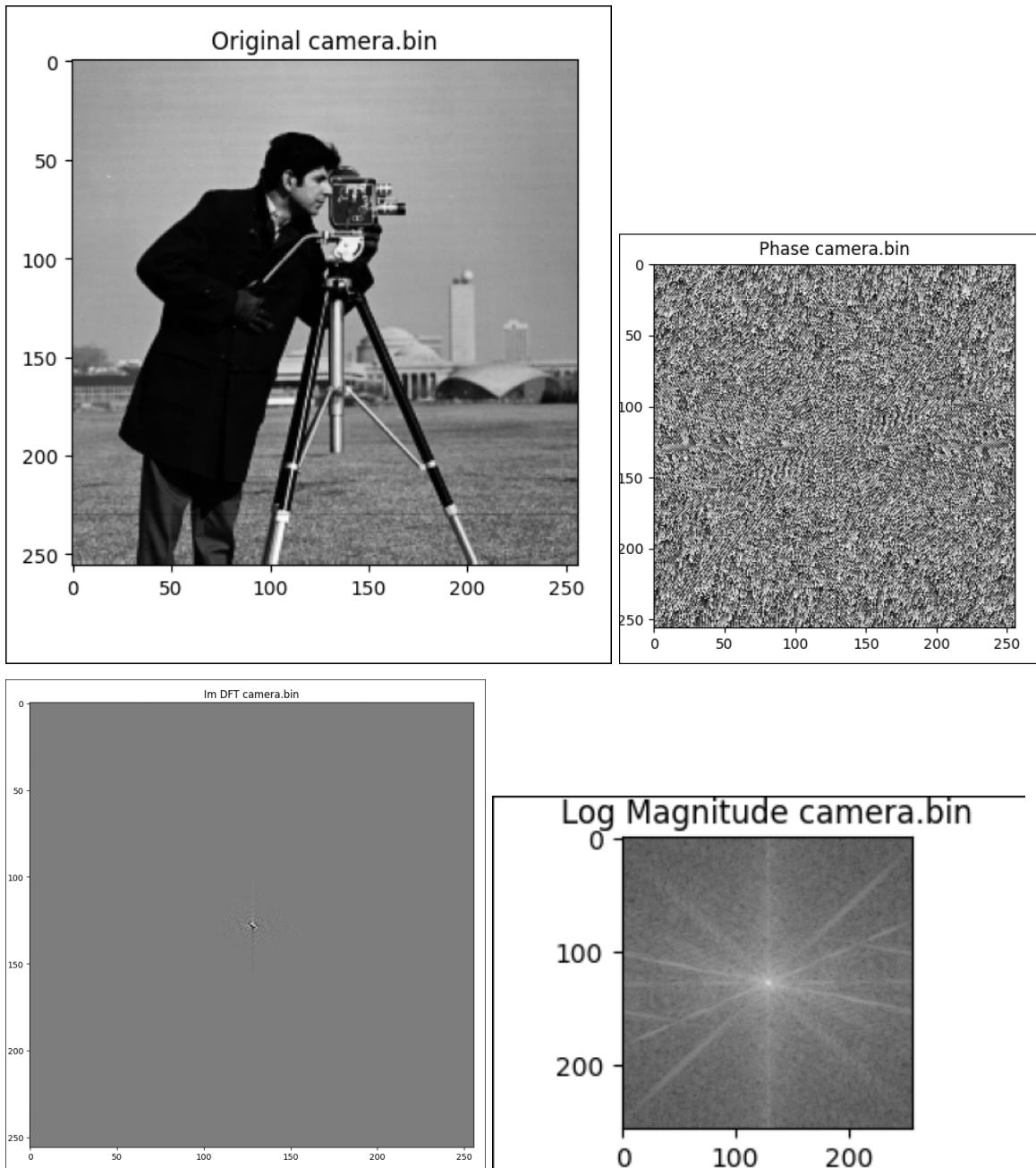
Các bước thực hiện

1. Đọc ảnh nhị phân `.bin` (256×256 , 8 bit) bằng `np.fromfile`.
2. Hiển thị ảnh gốc.
3. Tính DFT bằng `fft2` và dịch tâm bằng `fftshift`.
4. Hiển thị:
 - o Phần thực.
 - o Phần ảo.
 - o Log-magnitude spectrum.
 - o Pha (phase).









7. ẢNH PHỤC HỒI TỪ BIÊN ĐỘ VÀ PHA

Mục đích

Khảo sát tầm quan trọng của **biên độ** và **pha** trong việc tái tạo ảnh bằng cách xây dựng hai ảnh mới:

- J1J_1J1: chỉ giữ biên độ (magnitude), pha = 0.
J2J_2J2: chỉ giữ pha, biên độ = 1.

Lý thuyết

- Biểu diễn DFT dưới dạng cực:
 $F(u,v) = |F(u,v)| e^{j\theta(u,v)}$
 $|F(u,v)| e^{j\theta(u,v)} = |F(u,v)|$
- Khi:
 $|F| |F| |F|$ cố định và $\theta=0$: ảnh chỉ còn thông tin năng lượng.
 - $|F|=1$, $|F|=1$ và $\theta=0$: ảnh giữ lại cấu trúc (edges, hình dạng).
- Ảnh J2J_2J2 cho thấy **pha đóng vai trò chính trong việc bảo toàn hình dạng ảnh**.

Các bước thực hiện

1. Đọc ảnh **camera.bin** (256×256).
2. Tính DFT: $F = \text{fft2}(I)$.
3. Tạo:
 - $J1 = |F|$, $J1 = |F|$, sau đó lấy log để hiển thị: $JJ1 = \log(1 + J1)$.
 $J2 = e^{j \cdot \arg(F)}$, $J2 = e^{j \cdot \arg(F)}$, sau đó nghịch đảo Fourier để hiển thị ảnh thật.
4. Hiển thị ảnh **J2 (phase only)** và **log(J1) (magnitude only)**.
5. Quan sát rằng **J2 gần giống ảnh gốc**, chứng minh **pha chứa thông tin cấu trúc ảnh**.

