

# Homework 1

## 1 — Làm việc với file .bin (256 × 256, 8 bits/pixel)

### Mục tiêu

- (a) Đọc và hiển thị hai ảnh lena.bin và peppers.bin.
  - (b) Tạo ảnh J có nửa trái từ Lena và nửa phải từ Peppers.
  - (c) Tạo ảnh K bằng cách hoán đổi nửa trái/phải của J.
  - (d) Nộp mã và printouts (ảnh gốc, J, K).
- File .bin chứa giá trị gray-level liên tiếp, 1 byte/pixel. Đọc toàn bộ file vào 1D numpy array rồi reshape thành (256,256).
  - Kiểm tra kích thước file: phải bằng  $256 \times 256 = 65536$  bytes.

### Các bước chi tiết

#### 1. Đọc file .bin

- Dùng `np.fromfile(path, dtype=np.uint8)` để đọc tất cả byte.
- Kiểm tra `arr.size == 256*256` — nếu khác, báo lỗi.
- Chuyển `arr.reshape((256,256))`.

#### 2. Tạo ảnh J

- Khởi tạo `J = np.zeros_like(image1)` (cùng kích thước, dtype).
- Gán `J[:, :128] = lena[:, :128]` (cột 0..127)
- Gán `J[:, 128:] = peppers[:, 128:]` (cột 128..255)
- Lưu ý: indexing của NumPy 0-based, cột 128 là cột thứ 129 theo đề.

#### 3. Tạo ảnh K (swap halves)

- Khởi tạo `K = np.zeros_like(J)`
- `K[:, :128] = J[:, 128:]`
- `K[:, 128:] = J[:, :128]`

#### 4. Hiển thị ảnh

- Dùng `cv2.imshow(window_name, image)` cho mỗi ảnh.

- cv2.waitKey(0) để đợi phím, cv2.destroyAllWindows() để đóng cửa sổ.
- **Để in vào báo cáo:** lưu ảnh bằng cv2.imwrite("lena.png", lena) v.v. (PNG giữ chất lượng, tốt hơn JPEG cho ảnh đồ họa).

## 5. Ghi chú về lưu file

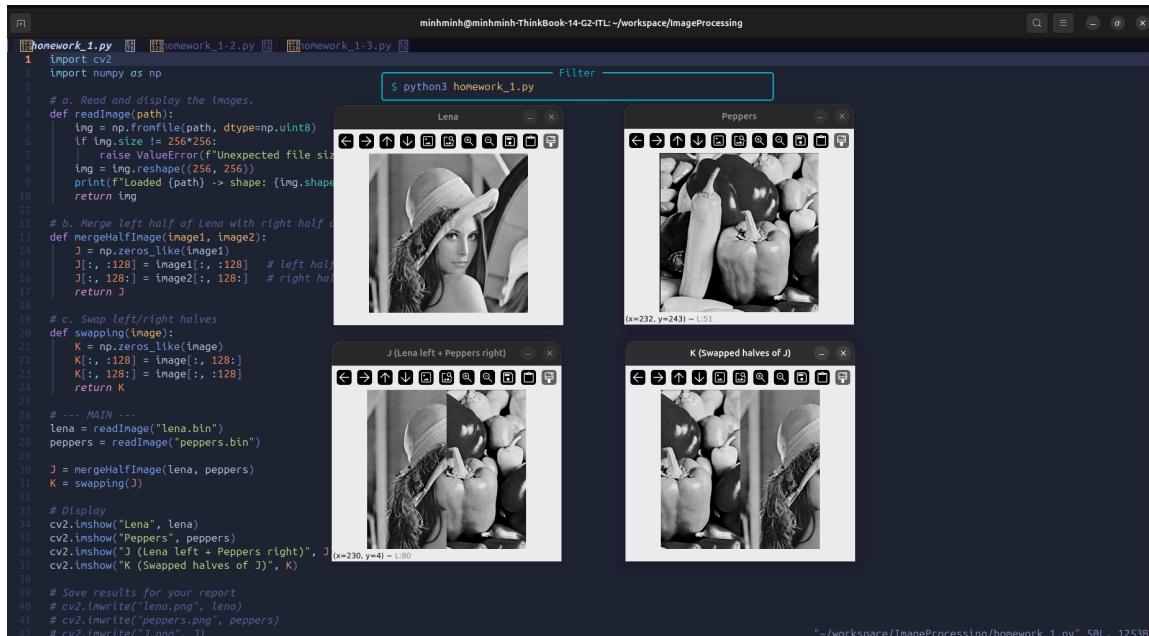
- np.fromfile đọc theo byte — không cần lo về endianness vì mỗi pixel 1 byte.
- Nếu file .bin có header, phương pháp này sai — kiểm tra kích thước.

## Giải thích đoạn mã chính (ý chính)

- `readImage(path):`
  - đọc file, kiểm tra kích thước, reshape.
- `mergeHalfImage(image1, image2):`
  - tạo ảnh mới J bằng nửa trái image1 và nửa phải image2.
- `swapping(image):`
  - hoán đổi hai nửa của ảnh.

## Kiểm tra & lưu kết quả

- Lưu cả 4 ảnh: lena.png, peppers.png, J.png, K.png.



The screenshot shows a terminal window with the command \$ python3 homework\_1.py. Below the terminal are four image windows. The first window is titled 'Lena' and shows a woman wearing a hat. The second window is titled 'Peppers' and shows several bell peppers. The third window is titled 'J (Lena left + Peppers right)' and shows the left half of Lena's face and the right half of the peppers. The fourth window is titled 'K (Swapped halves of J)' and shows the right half of Lena's face and the left half of the peppers. The code in the terminal is as follows:

```

#homework_1.py
# homework_1-2.py
#homework_1-3.py
minhminh@minhminh-ThinkBook-14-G2-ITL:~/workspace/ImageProcessing$ python3 homework_1.py
# a. Read and display the images.
1 import cv2
2 import numpy as np
3 # a. Read and display the images.
4 def readImage(path):
5     img = np.fromfile(path, dtype=np.uint8)
6     if img.size != 256*256:
7         raise ValueError(f"Unexpected file size {img.size}.")
8     img = img.reshape((256, 256))
9     print(f"Loaded {path} -> shape: {img.shape}")
10    return img
11
12 # b. Merge left half of Lena with right half of Peppers
13 def mergeHalfImage(image1, image2):
14     J = np.zeros_like(image1)
15     J[:, :128] = image1[:, :128] # left half
16     J[:, 128:] = image2[:, 128:] # right half
17     return J
18
19 # c. Swap left/right halves
20 def swapping(image):
21     K = np.zeros_like(image)
22     K[:, :128] = image[:, 128:]
23     K[:, 128:] = image[:, :128]
24     return K
25
26 # --- MAIN ---
27 lena = readImage("lena.bin")
28 peppers = readImage("peppers.bin")
29
30 J = mergeHalfImage(lena, peppers)
31 K = swapping(J)
32
33 # Display
34 cv2.imshow("Lena", lena)
35 cv2.imshow("Peppers", peppers)
36 cv2.imshow("J (Lena left + Peppers right)", J)
37 cv2.imshow("K (Swapped halves of J)", K)
38
39 # Save results for your report
40 # cv2.imwrite("lena.png", lena)
41 # cv2.imwrite("peppers.png", peppers)
42 # cv2.imwrite("J.png", J)
43 # cv2.imwrite("K.png", K)

```

## 2 — Ảnh grayscale lenagray.jpg (phóng ảnh âm bản)

### Mục tiêu

- Đọc ảnh grayscale (J1) và tạo ảnh âm bản J2 = 255 - J1.
- Hiển thị và lưu kết quả.
- Với ảnh 8-bit grayscale, đảo sáng = 255 - pixel.
- Đảm bảo ảnh đọc bằng flag cv2.IMREAD\_GRAYSCALE.

### Các bước chi tiết

#### 1. Đọc ảnh:

- J1 = cv2.imread("lenagray.jpg", cv2.IMREAD\_GRAYSCALE)
- Kiểm tra if J1 is None: raise FileNotFoundError(...)

#### 2. Tạo negative:

- J2 = 255 - J1 (NumPy broadcast, dtype uint8 -> kết quả vẫn uint8)

#### 3. Hiển thị & lưu:

- cv2.imshow("Original J1", J1)
- cv2.imshow("Negative J2", J2)
- cv2.imwrite("lenagray\_negative.jpg", J2) (hoặc png nếu muốn không nén)

#### 4. Ghi chú:

- Nếu J1 không phải uint8, convert J1 = J1.astype(np.uint8) trước khi tính toán.

- o 255 - J1 giữ nguyên dtype là uint8 (trừ khi J1 là khác).

```

minhminh@minhminh-ThinkBook-14-G2-ITL:~/workspace/ImageProcessing
$ python3 homework_1-2.py

```

```

1 #homework_1.py | homework_1-2.py | homework_1-3.py
2
3 import cv2
4 import numpy as np
5
6 # (c) Read the grayscale image
7 J1 = cv2.imread('lenagray.jpg', cv2.IMREAD_GRAYSCALE)
8
9 if J1 is None:
10     raise FileNotFoundError("Image not found. Make sure lenagray.jpg is in your folder.")
11
12 # Show original
13 cv2.imshow("Original J1", J1)
14
15 # (d) Create negative
16 J2 = 255 - J1
17
18 # Show negative
19 cv2.imshow("Negative J2", J2)
20
21 # Save negative
22 cv2.imwrite("lenagray_negative.jpg", J2)
23
24 cv2.waitKey(0)
25 cv2.destroyAllWindows()

```

## 3 — Ảnh màu lena512color.jpg (hoán đổi kênh màu)

### Mục tiêu

- Đọc ảnh màu 512×512, tạo ảnh J2 bằng cách:
  - o Red(J2) = Blue(J1)
  - o Green(J2) = Red(J1)
  - o Blue(J2) = Green(J1)

### Các bước chi tiết

#### 1. Đọc ảnh:

- o `J1 = cv2.imread("lena512color.jpg", cv2.IMREAD_COLOR)`
- o Kiểm tra `if J1 is None: raise FileNotFoundError(...)`

#### 2. Khởi tạo J2:

- o `J2 = np.zeros_like(J1)` (giữ shape và dtype)

#### 3. Gán kênh:

- o Theo đề:  $\text{Red}(J2)=\text{Blue}(J1) \rightarrow$  trong OpenCV: `J2[:, :, 2] = J1[:, :, 0]`
- o  $\text{Green}(J2)=\text{Red}(J1) \rightarrow$  `J2[:, :, 1] = J1[:, :, 2]`
- o  $\text{Blue}(J2)=\text{Green}(J1) \rightarrow$  `J2[:, :, 0] = J1[:, :, 1]`

#### 4. Hiển thị & lưu:

- o `cv2.imshow("Original J1", J1)`

- o `cv2.imshow("Modified J2 (Swapped Bands)", J2)`
- o `cv2.imwrite("lena512color_swapped.jpg", J2)` (lưu JPEG hoặc PNG)

## 5. Ghi chú:

- o Nếu muốn xuất file mà giữ màu chuẩn khi mở bằng các công cụ khác (Pillow, trình duyệt), dùng PNG.
- o Kiểm tra kích thước `J1.shape == (512, 512, 3)`.

The screenshot shows a terminal window with the command `$ python3 homework_1-3.py` running. Below the terminal are two image windows. The left window is titled "Original J1" and shows a woman wearing a hat. The right window is titled "Modified J2 (Swapped Bands)" and shows the same woman with her colors swapped, appearing mostly green. A status bar at the bottom of each image window displays color coordinates and values.

```

minhminh@minhminh:~/workspace/ImageProcessing$ python3 homework_1-3.py
minhminh@minhminh:~/workspace/ImageProcessing$ python3 homework_1-3.py
# (c) Read the grayscale image
J1 = cv2.imread("lenagray.jpg", cv2.IMREAD_GRAYSCALE)

if J1 is None:
    raise FileNotFoundError("Image not found. Make sure lenagray.jpg is in your folder.")

# Show original
cv2.imshow("Original J1", J1)

# (d) Create negative
J2 = 255 - J1

# Show negative
cv2.imshow("Negative J2", J2)

# Save negative
cv2.imwrite("lenagray_negative.j
cv2.waitKey(0)
cv2.destroyAllWindows()

```