

Bài thực hành: Sử dụng thuật toán DFT để tách tin trong ảnh

1. Mục đích

Bài thực hành này hướng dẫn sử dụng thuật toán DFT để tách tin trong ảnh.

2. Yêu cầu đối với sinh viên

Có kiến thức cơ bản về thuật toán DFT.

Đã làm bài thực hành stego-image-code-dft.

3. Nội dung thực hành

3.1 Khởi động bài lab

Cài đặt bài lab tại: <https://github.com/dotrantrung2003/stego-image-code-dft>

Giãn nén và chuyển vào thư mục `/labtainer/trunk/labs`.

Trên terminal, gõ:

```
labtainer -r stego-image-code-dft-2
```

Sau khi khởi động xong, một terminal ảo xuất hiện.

3.2 Tiền xử lý

Trên terminal sử dụng lệnh: `find image.png` để xem ảnh được giấu tin ở bài thực hành stego-image-code-dft.

Trên terminal, gõ lệnh:

```
python3 preprocessing.py stego_image.png
```

Chương trình `preprocessing.py` sẽ chuyển đổi từ ảnh gốc ban đầu (`stego_image.png`) thành ma trận điểm ảnh của ảnh xám (`gray_matrix.txt`).

Quan sát ma trận điểm ảnh của ảnh xám, ta có nhận thấy các giá trị đều nằm trong đoạn $[0, 255]$, các giá trị này đại diện cho độ sáng của điểm ảnh tại vị trí đó.

Qua bước tiền xử lý ta biết được kích thước của ảnh là 1280×720 pixels. Ta chia ma trận điểm ảnh (`gray_matrix.txt`) thành các khối 8×8 pixels \Rightarrow Ta có thể coi ma trận điểm ảnh (`gray_matrix.txt`) là 1 ma trận khối có kích thước 160×90 (khối 8×8).

Tin sẽ được giấu vào các khối 8×8 này.

3.3 Tách tin

Trên terminal, gõ lệnh:

```
python3 block.py gray_matrix.txt
```

Chương trình `block.py` sẽ lấy khối 8×8 tại vị trí hàng i cột j (i, j nhập từ bàn phím) ở ma trận khối mà ta đã đề cập ở bước trên.

Khối 8×8 trích xuất được lưu tại `gray_block_<i>_<j>.txt`.

Để biết được các khối nào được sử dụng để giấu tin, sinh viên hãy xem file *instruction.txt*.

Trên terminal, gõ lệnh:

```
python3 DFT.py gray_block_<i>_<j>.txt
```

Chương trình DFT.py sẽ thực hiện biến đổi DFT và lượng tử hóa lên khối 8x8 vừa trích xuất.

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-i2\pi(\frac{ux}{M} + \frac{vy}{N})}$$

Công thức trên là phép biến đổi DFT.

f là ma trận cần biến đổi (miền không gian)

F là ma trận sau khi biến đổi (miền tần số)

M, N là kích thước của ma trận.

Ta có thể dễ dàng nhận thấy ma trận mới là một ma trận số phức. Do đó kết quả thu được sẽ được lưu thành 2 phần: phần thực (*dft_real.txt*) và phần ảo (*dft_imag.txt*).

Biết rằng phần thực (*dft_real.txt*) được sử dụng để giấu tin. Sinh viên xem file *instruction.txt* sẽ có 1 hình minh họa vị trí tin được giấu trong phần thực.

```
@ @ @ @ @ @ @ @
@ @ @ @ @ @ @ @
@ @ @ @ @ @ @ @
@ @ @ * @ @ @ @
@ @ @ @ * @ @ @
@ @ @ @ @ @ @ @
@ @ @ @ @ @ @ @
@ @ @ @ @ @ @ @
```

Với mỗi vị trí ta sẽ lấy được 1 bit. Ví dụ:

- Vị trí (3, 3) = 11 = 0000 1011 (nhị phân). Lấy 1 bit cuối cùng (Tách được bit thứ nhất).

- Vị trí $(4, 4) = 15 = 0000\ 1111$ (nhị phân). Lấy 1 bit cuối cùng. (Tách được bit thứ hai).

- Thu được: 11

Các bước trên là hướng dẫn để tách 2 bit được giấu vào 1 khối 8x8. Để hoàn thành bài thực hành này sinh viên cần tách được toàn bộ các bit được giấu tại các khối được đề cập trong file *instruction.txt*.

Kết quả thu được vào file *result.txt* gồm: Dạng nhị phân thu được và mã ASCII tương ứng.

Các bước tổng quan: Sử dụng *block.py* để lấy khối 8x8 tại vị trí hàng i cột j của *gray_matrix.txt* → DFT.py để biến đổi DFT → Tách tin từ phần thực → Ghi kết quả vào file *result.txt*.

3.4 Kết thúc bài lab

Trên terminal, gõ:

stoplab

Khi bài lab kết thúc, một tệp lưu kết quả được tạo và lưu vào một vị trí được hiển thị bên dưới.

Sinh viên cần nộp file *.lab* để chấm điểm.

Để kiểm tra kết quả khi trong khi làm bài thực hành sử dụng lệnh:

checkwork

Trong quá trình làm bài sinh viên cần thực hiện lại bài lab, sử dụng lệnh:

labtainer -r stego-image-code-dft-2