

Hướng dẫn thực hành bài lab dct_tool trên nền tảng Ubuntu

1. Mục đích

- Hiểu nguyên lý nhúng thông điệp vào ảnh sử dụng kỹ thuật **DCT (Discrete Cosine Transform)** trong watermarking.
- Nắm được quy trình mã hóa và giải mã thông điệp trong ảnh dựa trên miền tần số, bao gồm chuyển đổi ảnh sang miền DCT và trích xuất thông điệp.
- Thực hành nhúng thông điệp vào ảnh và trích xuất thông điệp từ ảnh bằng công cụ Python.
- Làm quen với môi trường dòng lệnh, các thư viện Python và công cụ hỗ trợ xử lý ảnh trên Ubuntu.
- Đánh giá tính hiệu quả và độ bền của kỹ thuật DCT trong watermarking so với các phương pháp khác như LSB.

2. Lý thuyết thuật toán

- **DCT (Discrete Cosine Transform):**
 - Là một kỹ thuật chuyển đổi ảnh từ miền không gian (spatial domain) sang miền tần số (frequency domain), thường được sử dụng trong các chuẩn nén ảnh như JPEG.
 - Trong watermarking, DCT cho phép nhúng thông điệp vào các hệ số tần số của ảnh, giúp thông điệp ít bị ảnh hưởng bởi các phép biến đổi như nén, cắt xén hoặc thay đổi kích thước.
 - Các hệ số tần số trung bình thường được chọn để nhúng thông điệp, đảm bảo cân bằng giữa độ bền của watermark và chất lượng hình ảnh.
- **Các bước chính:**
 - **Mã hóa:**
 - Chuyển ảnh sang miền DCT bằng cách chia ảnh thành các khối (thường là 8x8 pixel).
 - Nhúng thông điệp vào các hệ số DCT (thay đổi các hệ số tần số trung bình).
 - Chuyển ngược từ miền DCT về miền không gian để tạo ảnh đã nhúng thông điệp.
 - **Giải mã:**
 - Chuyển ảnh đã nhúng thông điệp sang miền DCT.
 - Trích xuất thông điệp từ các hệ số DCT đã được thay đổi.
 - Khôi phục thông điệp gốc dưới dạng văn bản.

- **Ưu điểm:**
 - Độ bền cao: Watermark dựa trên DCT chịu được các phép biến đổi ảnh như nén JPEG, thay đổi kích thước hoặc thêm nhiễu.
 - Khó phát hiện: Thông điệp được nhúng vào miền tần số, ít gây thay đổi trực quan trên ảnh.
- **Nhược điểm:**
 - Phức tạp hơn LSB, yêu cầu tính toán nặng hơn (chuyển đổi DCT/IDCT).
 - Chất lượng ảnh có thể bị ảnh hưởng nếu nhúng thông điệp với cường độ cao.

3. Nội dung thực hành

3.1 Chuẩn bị môi trường

Khởi động bài lab:

Truy cập đường dẫn: https://github.com/vuongnguyen168/dct_tool.git tải bài lab về và lưu trong thư mục labtainer/trunk/labs

Cài đặt lab: `rebuild -b dct_tool`

Khởi động bài lab: `labtainer -r dct_tool`

Cài đặt các thư viện cần thiết:

`sudo apt update`

`pip install -r requirements.txt`

`sudo apt install libgl1`

Kiểm tra các file ảnh có sẵn trong thư mục:

`ls`

Các file ảnh khả dụng: `lenna.png`, `pepper.png`, `babylon.png`.

3.2 Chạy chương trình

- Khởi động chương trình Python:
`python3 watermarking.py`

3.3 Mã hóa thông điệp

- Chọn tùy chọn 1 trong giao diện chương trình.
- Nhập tên file ảnh (chọn 1 trong 3 file sau):
 - `lenna.png`
 - `pepper.png`
 - `babylon.png`
- Nhập thông điệp cần mã hóa, ví dụ:
`This is the secret message`
- Chương trình sẽ chuyển ảnh sang miền DCT, nhúng thông điệp vào các hệ số tần số và tạo file ảnh mới (nếu có).

3.4 Giải mã thông điệp

- Chọn tùy chọn 2 trong giao diện chương trình.
- Nhập tên file ảnh đã nhúng thông điệp (file được tạo từ bước mã hóa).
- Chương trình sẽ trích xuất thông điệp từ các hệ số DCT và lưu vào một file văn bản.
- Thoát chương trình:
 - Nhấn bất kỳ phím nào ngoài 1, 2, 3.
- Kiểm tra danh sách file trong thư mục để tìm file chứa thông điệp đã giải mã:
ls
- Xem nội dung thông điệp đã giải mã:
cat <tên_file>

3.5 Kiểm tra kết quả

- So sánh thông điệp gốc (This is the secret message) với thông điệp được giải mã.
- Kiểm tra chất lượng ảnh đã nhúng tin bằng cách quan sát trực quan hoặc sử dụng công cụ phân tích ảnh (nếu có).

4. Kết quả

- **Kết quả thực hành:**
 - Hoàn thành toàn bộ quy trình: mã hóa thông điệp → nhúng vào ảnh → trích xuất và giải mã thông điệp.
 - Thông điệp khôi phục khớp hoàn toàn với thông điệp gốc.
 - Ảnh sau khi nhúng thông điệp có sự thay đổi chất lượng không đáng kể, phù hợp cho các ứng dụng watermarking.
- **Nhận xét:**
 - **Độ bền:** Kỹ thuật DCT cung cấp độ bền cao hơn so với LSB, đặc biệt khi ảnh bị nén hoặc thay đổi kích thước. Điều này làm cho DCT phù hợp cho các ứng dụng như bảo vệ bản quyền hoặc xác thực nội dung.
 - **Chất lượng ảnh:** Thay đổi chất lượng ảnh là tối thiểu, nhưng có thể nhận thấy khi nhúng thông điệp dài hoặc sử dụng cường độ nhúng cao.
 - **Hiệu suất:** Quy trình DCT yêu cầu tính toán phức tạp hơn LSB, nhưng vẫn khả thi trên các hệ thống hiện đại.
 - **Hạn chế:** Kỹ thuật DCT vẫn có thể bị tấn công bởi các phương pháp phân tích tần số nâng cao, đòi hỏi các cải tiến như nhúng thông điệp vào nhiều khối DCT hoặc kết hợp với các kỹ thuật mã hóa khác.