

# Hướng dẫn thực hành bài lab DWT Coding trên nền tảng Labtainer

## 1. Mục đích

- Hiểu nguyên lý giấu tin trong ảnh sử dụng phép biến đổi Wavelet rời rạc (DWT)
- Thực hành giấu và giải mã thông tin trong ảnh số
- Làm quen với các công cụ xử lý ảnh trong Python

## 2. Lý thuyết thuật toán

- **Biến đổi Wavelet rời rạc (DWT):** Phân tách ảnh thành các thành phần tần số khác nhau
- **Giấu tin trong miền Wavelet:** Thông tin được giấu trong các hệ số Wavelet của ảnh
- **Ưu điểm:** Độ bền cao với các tác động như nén ảnh, nhiễu
- **Các bước chính:**
  1. Phân tách ảnh bằng DWT
  2. Nhúng thông tin vào các hệ số Wavelet
  3. Tái tạo ảnh đã giấu tin bằng biến đổi ngược IDWT

## 3. Nội dung thực hành

### 3.1 Khởi động bài lab

Truy cập đường dẫn: [https://github.com/vuongnguyen168/my\\_designed\\_lab](https://github.com/vuongnguyen168/my_designed_lab) tải bài lab về và cop vào thư mục labtainer/trunk/labs

Build bài lab

```
rebuild -b dwt_coding
```

Khởi động bài lab

```
labtainer -r dwt_coding
```

### 3.2 Chuẩn bị môi trường

```
sudo apt update
```

```
sudo apt install git
```

### 3.3 Cài đặt công cụ

git init

git pull https://github.com/vuongnguyen168/lab\_1\_dwt\_code.git main--allow-unrelated-histories

- Cài các thư viện cần thiết:  
pip install PyWavelets opencv-python Pillow

### 3.4 Thực hiện giấu tin

- Tìm kiếm và chỉnh sửa thông điệp được giấu trong file dwt\_encrypt.py thành “secret message”
- Tiến hành giấu tin:  
python3 dwt\_encrypt.py
- Kiểm tra file encrypted.png đã được tạo thành công hay chưa:  
ls

### 3.5 Giải mã tin đã giấu

- Chạy chương trình để giải mã tin:  
python3 dwt\_decrypt.py >> plaintext.txt
- Xem nội dung thông điệp đã giải mã  
cat plaintext.txt
- Kiểm tra dung lượng file ảnh đã giấu:  
ls -lh <tên file>

=> So sánh với dung lượng ảnh gốc trong /picture, có gì khác biệt ?

## 4. Kết quả

- Chạy được tất cả các bước như yêu cầu, đạt được hết các checkwork của bài lab.
- Kết thúc bài lab: stoplab dwt\_coding
- Khởi động lại bài lab: labtainer -r dwt\_coding