

**ĐẠI HỌC ĐÀ NẴNG  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA CÔNG NGHỆ THÔNG TIN**



# **PBL6: ĐỒ ÁN CHUYÊN NGÀNH**

## **Đề tài**

**Xây dựng trang web tái tạo ảnh CT từ ảnh Sinogram**

### **SINH VIÊN THỰC HIỆN:**

<b>Nguyễn Duy Hậu</b>	<b>LỚP: 20TCLC_Nhat2</b>
<b>Nguyễn Trọng Nghĩa</b>	<b>LỚP: 18TCLC_Nhat</b>
<b>Trần Chính Huy</b>	<b>LỚP: 20TCLC_Nhat2</b>

**GIẢNG VIÊN HƯỚNG DẪN: TS. Phạm Công Thắng**

**Đà Nẵng, 12/ 2023**

## MỤC LỤC

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT .....	5
1.1.        6	
1.2.        6	
CHƯƠNG 2. ỨNG DỤNG HỌC MÁY, KIỂM THỬ VÀ QUẢN TRỊ MẠNG ...	7
2.1.        8	
2.2.        12	
2.3.        13	
CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ .....	12
3.1 Các bước triển khai đề tài .....	12
3.2 Đánh giá kết quả .....	14
3.2.1 Kết quả mô hình AI .....	14
3.2.2 Kết quả trang web .....	17
CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	17
4.1 Kết luận.....	17
4.2 Hướng phát triển .....	17

---

**DANH SÁCH CÁC TỪ VIẾT TẮT**

Từ viết tắt	Diễn giải
CT	computed tomography
ReLU	rectified linear unit
FBP	Filtered Back Projection
IR	Iterative Reconstruction

## GIỚI THIỆU ĐỀ TÀI

Đề tài mà tôi đang nghiên cứu và phát triển tập trung vào việc xây dựng một hệ thống tái tạo ảnh CT từ dữ liệu sinogram. Trong lĩnh vực y học và xử lý hình ảnh, quá trình xây dựng hình ảnh CT từ dữ liệu sinogram đóng một vai trò quan trọng trong việc chẩn đoán và điều trị các bệnh lý.

Nghiên cứu này kết hợp sự sáng tạo của mô hình học máy, đặc biệt là mô hình Unet, để tối ưu hóa và cải thiện chất lượng hình ảnh tái tạo. Mô hình Unet được lựa chọn vì khả năng của nó trong việc kết hợp thông tin rộng lớn và giữ lại chi tiết cần thiết, đặc biệt là trong các ứng dụng y học.

Hệ thống được phát triển cũng tích hợp các công nghệ mới như TensorFlow để tạo ra một trải nghiệm tái tạo hình ảnh mạnh mẽ và linh hoạt. Đồng thời, quá trình nghiên cứu này cũng liên quan đến các khía cạnh như bảo mật thông tin y tế và tối ưu hóa hiệu suất của hệ thống trực tuyến thông qua việc tích hợp vào môi trường web.

Với mong muốn nâng cao chất lượng hình ảnh tái tạo từ dữ liệu sinogram và tạo ra một ứng dụng web hiệu quả, đề tài này không chỉ đóng góp vào lĩnh vực xử lý hình ảnh y tế mà còn mang lại giá trị ứng dụng trong việc cải thiện quy trình chẩn đoán và điều trị bệnh lý trong lĩnh vực y học.

**BẢNG PHÂN CÔNG CÔNG VIỆC**

Tên thành viên	Công việc
Nguyễn Duy Hậu	Tìm dataset, Train model, backend, hỗ trợ front end, viết báo cáo
Nguyễn Trọng Nghĩa	Front-end,web server, hỗ trợ back-end
Trần Chính Huy	Kiểm thử, hỗ trợ front end

---

## CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

### 1.1. Tái tạo ảnh CT<sup>[1]</sup> là gì

- Chụp cắt lớp vi tính (CT hoặc CAT scan) là kỹ thuật kết hợp máy tính và tia X để tạo ra hình ảnh các cơ quan, xương và các mô của cơ thể theo lát cắt ngang
- Máy quét CT và Sinogram:
  - + Máy quét CT sử dụng tia X để tạo ra hình ảnh của cơ thể bằng cách đo lường lượng tia X được hấp thụ hoặc đi qua mô cơ thể.
  - + Sinogram là biểu đồ 2D hoặc 3D của dữ liệu tia X được thu thập từ máy quét CT.
- Transform Radon và Inverse Radon Transform:
  - + Transform Radon chuyển đổi hình ảnh CT thành sinogram tương ứng.
  - + Inverse Radon Transform thực hiện quá trình ngược lại, chuyển đổi sinogram thành hình ảnh CT.
- Tái tạo ảnh CT từ sinogram là quá trình khôi phục hình ảnh y tế ban đầu từ dữ liệu sinogram thu được từ máy quét CT.

Các phương pháp tái tạo ảnh CT được biết đến phổ biến là : FBP, IR, Deep Learning Reconstruction. Chúng ta có thể sử dụng lọc và các thuật toán khác để cải thiện chất lượng và độ rõ nét của hình ảnh được tái tạo.

### 1.2. Áp dụng mô hình học máy vào phục hồi ảnh CT

Áp dụng mô hình học máy vào phục hồi ảnh CT từ ảnh sinogram là việc sử dụng các mô hình học máy để tái tạo hình ảnh CT từ dữ liệu sinogram. Mô hình học máy có thể là một mạng nơ-ron sâu (deep neural network) hoặc bất kỳ mô hình học máy nào khác có khả năng học từ dữ liệu và áp dụng kiến thức đó để thực hiện một công việc cụ thể, trong trường hợp này là phục hồi hình ảnh CT.

- Sử dụng thư viện **TensorFlow**<sup>[2]</sup> để tạo mô hình học máy :

TensorFlow là một thư viện mã nguồn mở được phát triển bởi Google dùng để xây dựng và huấn luyện mô hình học máy và học sâu. Được giới thiệu lần đầu tiên vào năm 2015, TensorFlow nhanh chóng trở thành một trong những thư viện phổ biến nhất và mạnh mẽ nhất trong lĩnh vực trí tuệ nhân tạo và học máy.

Ưu điểm của TensorFlow:

- Khả năng Tích hợp và Tương thích Rộng Rãi:

TensorFlow có thể tích hợp một cách tốt với nhiều công nghệ và hệ sinh thái khác nhau, giúp tạo ra các ứng dụng máy học phức tạp và đa dạng.

- Tính Linh hoạt và Mở Rộng:

TensorFlow hỗ trợ cả việc xây dựng và triển khai mô hình trên nhiều nền tảng khác nhau như máy tính cá nhân, di động, và đám mây.

- Cộng đồng Lớn và Hỗ trợ Mạnh mẽ:

Có một cộng đồng lớn của các nhà phát triển và người sử dụng TensorFlow, điều này giúp động viên, chia sẻ kiến thức và giải đáp vấn đề.

TensorFlow được sử dụng rộng rãi trong nhiều lĩnh vực như xử lý ngôn ngữ tự nhiên, thị giác máy tính, dự đoán chuỗi thời gian, và nhiều ứng dụng khác trong lĩnh vực trí tuệ nhân tạo.

### 1.3. JavaScript<sup>[3]</sup> và ReactJS<sup>[4]</sup>

- **JavaScript** là ngôn ngữ lập trình website phổ biến hiện nay, nó được tích hợp và nhúng vào HTML giúp website trở nên sống động
- **ReactJS** là một opensource được phát triển bởi Facebook, ra mắt vào năm 2013, bản thân nó là một thư viện Javascript được dùng để xây dựng các tương tác với các thành phần trên website. Một trong những điểm nổi bật nhất của ReactJS đó là việc render dữ liệu không chỉ thực hiện được trên tầng Server mà còn ở dưới Client nữa.
- **Ưu điểm của ReactJS**
  - **Tốc độ:** Về cơ bản, ReactJS cho phép các nhà phát triển sử dụng các phần riêng lẻ của ứng dụng của họ ở cả phía máy khách và phía máy chủ, điều này cuối cùng giúp tăng tốc độ của quá trình phát triển. Nói một cách dễ hiểu, các nhà phát triển khác nhau có thể viết các phần riêng lẻ và tất cả các thay đổi được thực hiện sẽ không gây ra tính logic của ứng dụng.
  - **Linh hoạt:** So với các frontend framework khác, mã React dễ bảo trì hơn và linh hoạt hơn do cấu trúc mô-đun của nó. Do đó, sự linh hoạt này giúp tiết kiệm rất nhiều thời gian và chi phí cho doanh nghiệp.
  - **Hiệu suất:** React JS được thiết kế để cung cấp hiệu suất cao. Cốt lõi của khung cung cấp chương trình DOM ảo và kết xuất phía máy chủ, giúp các ứng dụng phức tạp chạy cực nhanh.
  - **Khả năng sử dụng:** Việc triển khai React khá dễ thực hiện nếu bạn có một số kiến thức cơ bản về JavaScript. Trên thực tế, một nhà phát triển JavaScript chuyên nghiệp có thể dễ dàng tìm hiểu tất cả các thông tin chi tiết của React framework chỉ trong một hoặc hai ngày.
  - **Phát triển ứng dụng di động**

### 1.4. Flask<sup>[5]</sup>

- **Flask** là một framework web được viết bằng ngôn ngữ lập trình Python. Flask được thiết kế để đơn giản và linh hoạt, giúp những người phát triển xây dựng ứng dụng web một cách dễ dàng.

- **Flask** không có các yêu cầu đặc biệt về cấu trúc dự án, điều này cho phép người sử dụng tự do tổ chức mã nguồn theo cách họ muốn. Flask cung cấp các công cụ cơ bản để xử lý routing (định tuyến URL), kết nối với cơ sở dữ liệu, xử lý yêu cầu HTTP, và tạo giao diện người dùng web.
- **Các ưu điểm của Flask**
  - **Đơn giản và Nhẹ:** Flask được thiết kế để đơn giản và nhẹ, giúp người phát triển tập trung vào logic của ứng dụng mà không bị mất trong cấu trúc phức tạp.
  - **Dễ Học và Sử Dụng:** Với cấu trúc ngắn gọn và tài liệu chi tiết, Flask là một framework dễ học và sử dụng, đặc biệt là cho những người mới bắt đầu với phát triển web.
  - **Mở Rộng Tốt:** Flask là một microframework, cho phép bạn chọn lựa và tích hợp các thành phần cần thiết theo yêu cầu. Các extension giúp mở rộng chức năng của Flask một cách linh hoạt.
  - **Khả Năng Tùy Biến Cao:** Flask không áp đặt quy tắc nghiêm ngặt về cấu trúc dự án, cho phép người phát triển tự do tổ chức mã nguồn theo cách tốt nhất đối với dự án của họ.
  - **Phù Hợp cho Dự Án Nhỏ và Trung Bình:** Flask thích hợp cho việc phát triển các ứng dụng web nhỏ đến trung bình, dự án prototype, và các dự án có yêu cầu đặc biệt.

## CHƯƠNG 2. ỨNG DỤNG HỌC MÁY, KIỂM THỬ VÀ QUẢN TRỊ MẠNG

### 2.1. Ứng dụng học máy

#### a) Mô hình Unet<sup>[6]</sup>

- U-Net là một mạng nơ-ron tích chập ban đầu được phát triển để phân đoạn các hình ảnh y sinh. Kiến trúc của nó trông giống như chữ U và do đó có tên là U-Net. Kiến trúc của nó được tạo thành từ hai phần, phần bên trái – trích xuất đặc trưng (encoder) và phần bên phải – giải mã đặc trưng (decoder). Mục đích của encoder là cô đọng thông tin ngữ cảnh trong khi vai trò của decoder là giải mã thông tin, xác định chính xác các đối tượng.

- Đặc điểm Chính:

+ Kiến Trúc U-Shape:

Mô hình U-Net có kiến trúc hình chữ U, với phần đỉnh của U là mạng trích xuất đặc trưng (Encoder), phần dưới là mạng phục hồi đặc trưng (Decoder).

+ Khả Năng Segment hóa Ảnh:



U-Net được thiết kế đặc biệt cho nhiệm vụ segment hóa hình ảnh, nơi mục tiêu là tạo ra một ảnh đầu ra mà mỗi pixel được gán một nhãn.

- + Cửa sổ Diễn giải (Context) và Chi tiết (Localization):

Bằng cách sử dụng cửa sổ diễn giải lớn ở phần Encoder và chi tiết ở phần Decoder, U-Net có khả năng kết hợp thông tin rộng lớn và giữ lại chi tiết cần thiết.

- + Sử dụng Công Kết hợp (Skip Connections):

Công kết hợp giữa các lớp ở phần Encoder và Decoder giúp truyền thông tin chi tiết từ các lớp bậc cao sang các lớp bậc thấp, giúp mô hình học được đặc trưng cấp cao và cấp thấp.

- + Hàm Kích hoạt Sigmoid ở Lớp Đầu ra:

Lớp đầu ra thường sử dụng hàm kích hoạt sigmoid để tạo ra các giá trị xác suất cho mỗi pixel, thường được sử dụng trong các nhiệm vụ như phân đoạn đối tượng.

- **Chúng tôi sử dụng mô hình Unet trong nghiên cứu này là vì:**

- + Khả năng tái tạo chi tiết:

Mô hình U-Net có khả năng tái tạo chi tiết cao, đặc biệt là khi cần phục hồi các chi tiết nhỏ trong hình ảnh CT từ sinogram.

- + Xử lý thông tin không đồng nhất:

Sinogram là một biểu đồ trực quan của dữ liệu CT và thường chứa các thông tin không đồng nhất. Mô hình U-Net, với các lớp skip connection giữa encoder và decoder, giúp xử lý hiệu quả các đặc trưng không đồng nhất này.

- + Khả năng học tập chuyển giao:

Mô hình U-Net có thể được huấn luyện trên một lượng lớn dữ liệu hình ảnh y tế, giúp nó học tập chuyển giao và áp dụng kiến thức đã học vào các tác vụ tái tạo hình ảnh CT cụ thể.

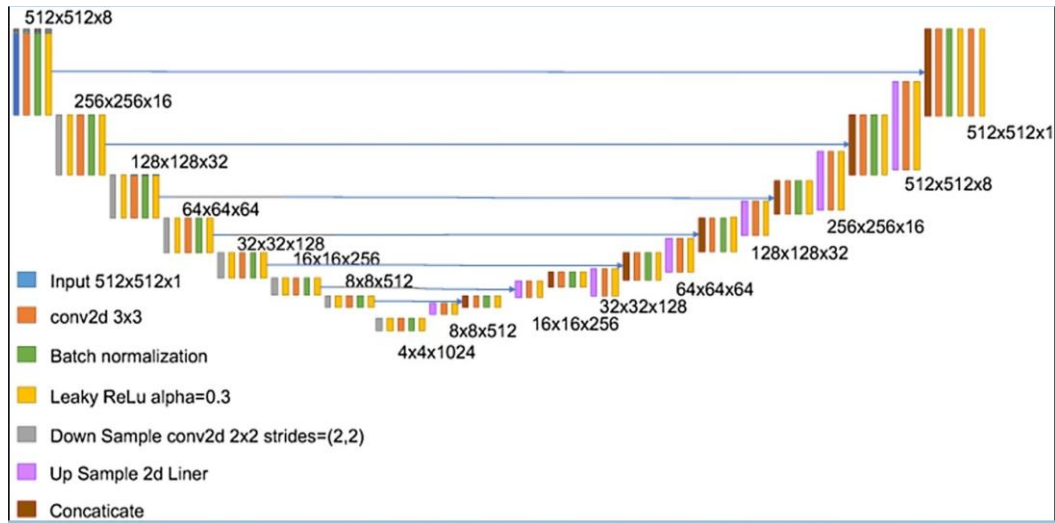
- + Hiệu suất và tốc độ:

Mô hình U-Net thường có hiệu suất cao và tốc độ tính toán tốt, đặc biệt là khi sử dụng trên phần cứng tăng tốc GPU. Điều này làm giảm thời gian tái tạo hình ảnh và tăng tính khả dụng của mô hình.

- + Linh hoạt và dễ triển khai:

Mô hình U-Net có thể được dễ dàng triển khai trong môi trường sản xuất và tích hợp vào các hệ thống y tế.

- Hình 1 cho thấy cấu trúc của U-Net block được sử dụng trong nghiên cứu này.



Hình 1. Cấu trúc của một unet block

Mô hình U-Net ban đầu được thiết kế để thực hiện phân đoạn hình ảnh (image segmentation) khác với việc tái tạo hình ảnh CT từ sinogram. Tôi đã sửa lại mô hình : đầu ra cuối cùng được đặt hành một kênh duy nhất , kết quả đầu ra được thể hiện dưới dạng hình ảnh.

#### Comparison of the applied methods.

Class	Method	Adoption	Validation loss
Downscaling	$2 \times 2$ stride 2 convolution	✓	0.0086
	Max pooling		0.0107
	Linear scaling		0.0099
Upscaling	$2 \times 2$ stride 2 deconvolution		0.0107
	Linear scaling	✓	0.0086
Generalization	Batch normalization	✓	0.0083
	Dropout		0.0141
Activate function	Sigmoid		0.0567
	ReLU		0.0132
	LeakyReLU	✓	0.0086
Skip connection	Previous U-Net input image to the next U-Net image		—
	First input image to all U-Net input images		—
Optimizer	Adam	✓	0.0071
	AdaBound		0.0142

Hình 2. So sánh các phương pháp khác nhau áp dụng cho mô hình Unet

- Từ bảng so sánh trên, chúng tôi đã sửa đổi quy trình downscaling từ max pooling sang tích chập  $2 \times 2$  sải bước 2 ( $2 \times 2$  stride 2 convolution). Chúng tôi cũng đã thay đổi quy trình upscaling từ deconvolution sang chia tỷ lệ tuyến tính (linear scaling) và hàm kích hoạt từ ReLU thành LeakyReLU. Chúng tôi cũng áp dụng batch normalization cho mô hình.

b ) **Stacked unet**<sup>[7]</sup>

- Mô hình Stacked U-Net là một biến thể của kiến trúc U-Net thông thường, trong đó nhiều lớp U-Net được xếp chồng lên nhau.
- Đặc điểm Chính của Stacked U-Net:
- + Nhiều Lớp U-Net:

Mô hình Stacked U-Net bao gồm nhiều lớp U-Net chồng lên nhau. Mỗi lớp U-Net được coi như một module độc lập với khả năng học và trích xuất đặc trưng từ dữ liệu.

- + Chia sẻ Thông tin Trong Quá trình Học:

Các lớp U-Net ở mỗi tầng đều chia sẻ thông tin với các tầng lân cận của chúng. Điều này giúp mô hình học được các mức độ biểu diễn đặc trưng ở cả các tầng thấp và cao.

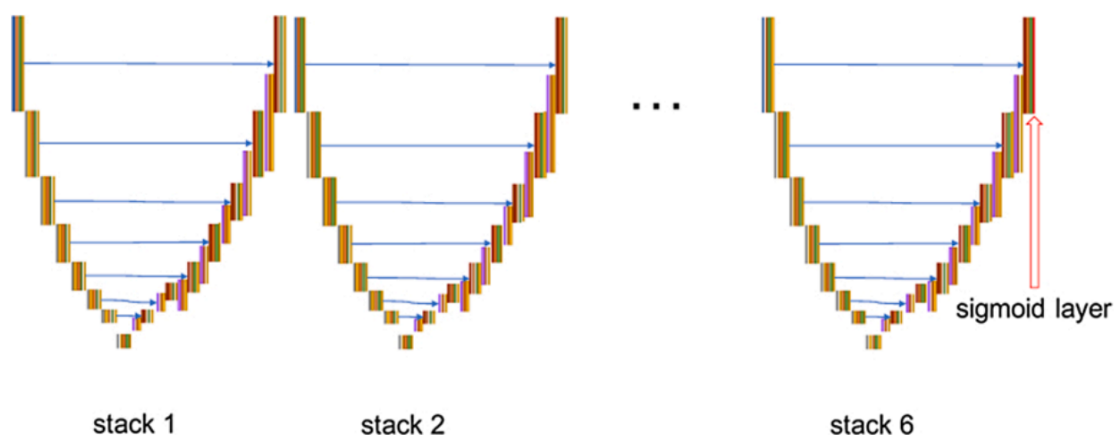
- + Tăng Cường Độ Phức Tạp của Mô Hình:

Stacked U-Net thường được sử dụng trong các nhiệm vụ đòi hỏi độ phức tạp và chi tiết cao, nơi các đặc trưng ở nhiều mức độ cần phải được học và tích hợp.

- + Tích hợp Các Phương Pháp Tăng cường:

Các phương pháp tăng cường như dropout và normalization thường được kết hợp để giảm quá mức học và cải thiện khả năng tổng quát hóa của mô hình.

- Hình 3 là model stacked unet được dùng trong nghiên cứu này :



Hình 3. Stacked unet

Relationship between the number of stacks and evaluation values for the 32 × 32 images.

Number of stacks	Training loss	Validation loss
1	0.0070	0.0073
2	0.0070	0.0072
3	0.0051	0.0053
4	0.0053	0.0054
5	0.0053	0.0301
6	0.0050	0.0173
7	0.0052	0.0232
8	0.0054	0.0479

Hình 4. Mối quan hệ giữa số stack và hiệu suất mô hình

Dựa vào Hình 4 trên chúng tôi lựa chọn số stack cho mô hình là 6 vì đó là số stack mà mô hình có training loss và validation loss tối ưu nhất. Sử dụng stacked unet giúp tăng cường khả năng học và hiệu suất của mô hình, đặc biệt là trong các nhiệm vụ phức tạp yêu cầu độ phân giải và độ chi tiết cao.

## 2.2. Ứng dụng kiểm thử

**Selenium IDE**<sup>[8]</sup> là công cụ giúp bạn phát triển và kiểm thử được xây dựng dưới dạng Add – ons của Firefox. Đây là cách tiện lợi nhất để xây dựng các ca kiểm thử, gồm các phần tử giao diện giúp bạn có thể lựa chọn thể hiện các thao tác, không chỉ tiết kiệm thời gian mà còn là cách thông minh để hiểu kịch bản Selenium. Bộ công cụ này cung cấp chức năng “thu và chạy lại” – Record and Playback. Nhờ đó, Tester có thể nhanh chóng tạo một bộ kịch bản kiểm tra (test script) bằng cách trực tiếp “thu” các thao tác của mình trên đối tượng cần kiểm tra thành 1 tập các câu lệnh “Selenese” (ngôn ngữ kịch bản được phát triển cho Selenium IDE và Selenium Core có dạng bản HTML). Sau đó chạy lại các câu lệnh này để kiểm tra. Chức năng này rất hữu dụng, cho phép tiết kiệm thời gian viết kịch bản kiểm tra. Selenium IDE cho phép lưu kịch bản đã thu dưới nhiều loại ngôn ngữ lập trình.

Dưới đây là một số đặc điểm và khả năng chính của Selenium IDE:

- Ghi và Chạy Kịch Bản Tự Động : Selenium IDE cho phép bạn ghi các hoạt động trên trình duyệt web và chuyển chúng thành các kịch bản thử nghiệm tự động mà bạn có thể chạy lại nhiều lần.
- Hỗ Trợ Đa Trình Duyệt : Selenium IDE có thể sử dụng với nhiều trình duyệt web phổ biến như Chrome, Firefox, và Edge.
- Dễ Dàng Sử Dụng : Giao diện người dùng của Selenium IDE thiết kế để làm cho việc tạo và quản lý các kịch bản thử nghiệm trở nên dễ dàng mà không đòi hỏi kỹ năng lập trình cao.

- Kiểm Thử Hiệu Suất : Selenium IDE có thể sử dụng để thực hiện kiểm thử hiệu suất bằng cách thực hiện đồng thời nhiều tác vụ trên trình duyệt.

Selenium IDE thường được sử dụng cho các dự án với quy mô nhỏ đến trung bình, trong khi dự án lớn và phức tạp hơn thì thường chuyển sang sử dụng Selenium WebDriver và các ngôn ngữ lập trình như Java, Python, hoặc C#.

### 2.3. Web Server

**Ngrok**<sup>[9]</sup> là công cụ tạo đường hầm (tunnel) giữa localhost của bạn và internet. Giúp người khác mạng có thể truy cập được localhost của bạn thông qua custom domain của ngrok Ví dụ: mydomain.ngrok.io => localhost:80

Các ứng dụng chính của **Ngrok**:

- Giúp bạn chạy demo dự án cho khách hàng xem từ chính máy của bạn mà không cần deploy lên server
- Bạn có thể test responsive trên mobile một cách dễ dàng thông qua URL mà ngrok cung cấp
- Xây dựng webhook tới localhost của bạn một cách dễ dàng
- Hỗ trợ http, https, tcp
- Hỗ trợ IP whitelist

## CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

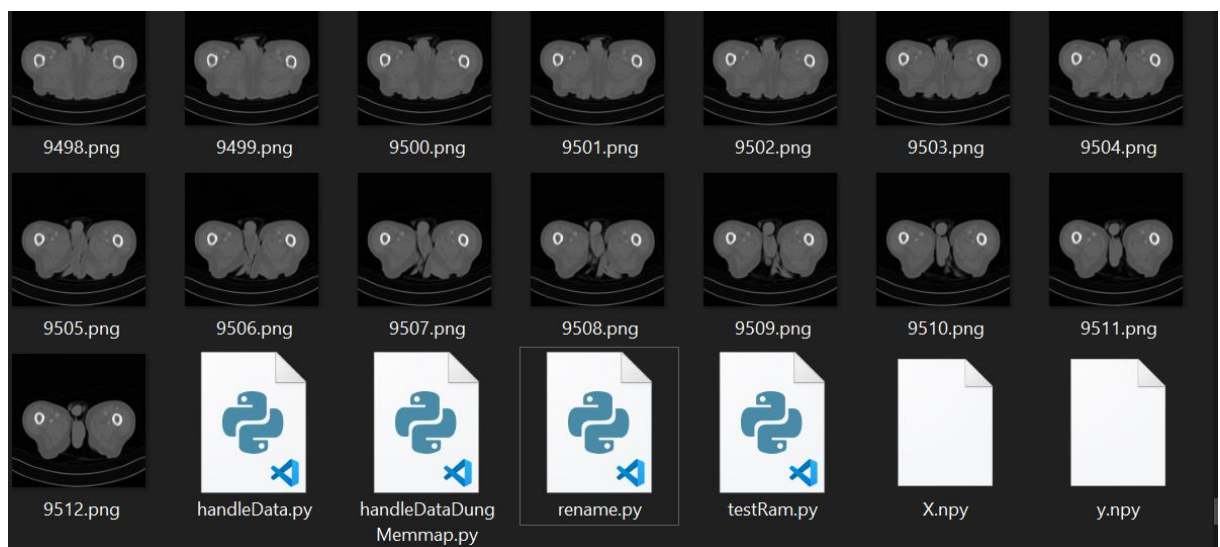
### 3.1 Các bước triển khai đề tài

- **B1 : Chuẩn bị Dữ liệu**
- + Dữ liệu gốc được lấy từ : CT Low Dose Reconstruction Dataset<sup>[10]</sup>



Hình ... Ảnh dataset CT Low Dose Reconstruction

- + Xây dựng tập dữ liệu huấn luyện gồm các cặp sinogram và ảnh CT tương ứng : ta sử dụng hàm Radon để chuyển ảnh CT thành ảnh Sinogram và lưu vào file y.npy, ảnh CT tương ứng được lưu vào file X.npy



- + Tập dữ liệu nguồn gồm 9512 cặp ảnh sinogram và ảnh CT được chia thành 3 tập dữ liệu nhỏ : Train\_data (5040 ảnh), Val\_data (1400 ảnh), Test\_data (560 ảnh).

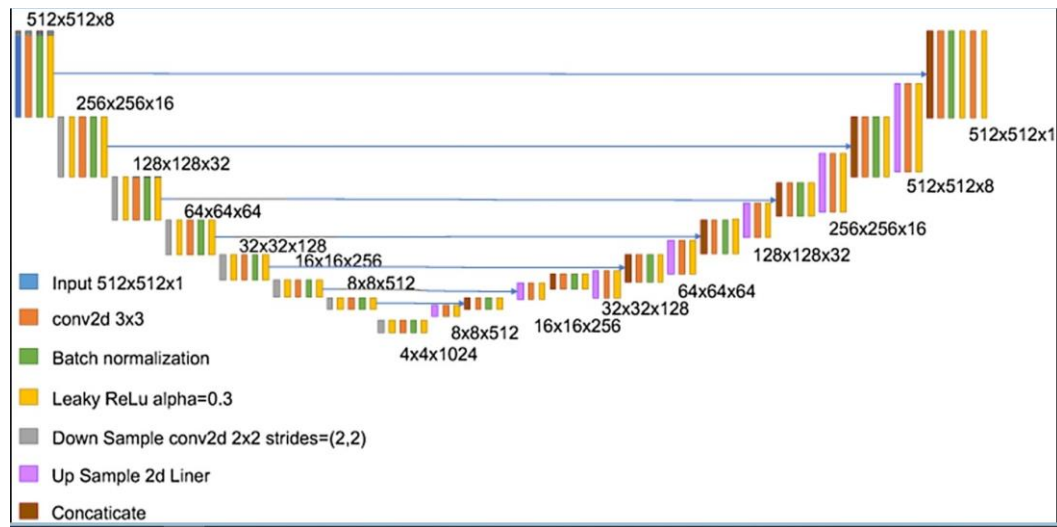
```
print(X_train.shape)
print(X_val.shape)
print(X_test.shape)
```

```
(5040, 128, 128, 1)
(1400, 128, 128, 1)
(560, 128, 128, 1)
```

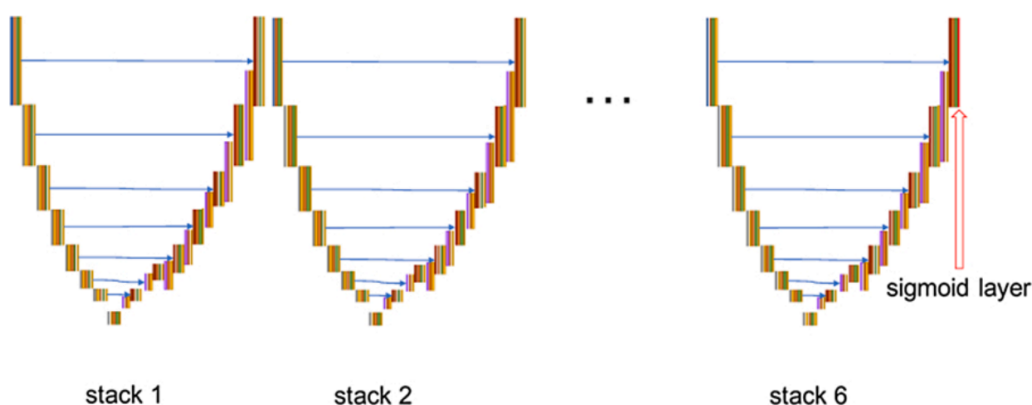
+ Sau đó ta chuẩn hóa dữ liệu về khoảng  $[0,1]$  để phù hợp với mô hình học máy.

## - B2 Xây dựng mô hình học máy:

+ Xây dựng mô hình Unet giống như hình sau



+ Xây dựng mô hình Stacked Unet gồm 6 block Unet được xếp chồng với nhau: Kết quả đầu ra trung gian giữa các block unet là kết quả của hàm LeakyReLU; kết quả đầu ra cuối cùng được chuẩn hóa trong khoảng từ 0 đến 1 bằng cách áp dụng hàm sigmoid.



## - B3 Huấn Luyện Mô hình:

Sử dụng tập dữ liệu huấn luyện để đào tạo mô hình học máy.

Điều chỉnh các tham số mô hình và theo dõi hiệu suất trên tập kiểm thử.



```

from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint

early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)
checkpointer = ModelCheckpoint('stacked_unet_29.h5', verbose=1, save_best_only=True)

callbacks = [early_stopping, checkpointer]

# Use pre-defined validation data
val_data = (X_val, y_val)

# Fit the model with early stopping and pre-defined validation data
history = model.fit(X_train, y_train, epochs=20, batch_size=64, validation_data=val_data, callbacks=callbacks)

Epoch 1/20
107/107 [=====] - ETA: 0s - loss: 0.0040 - mean_absolute_error: 0.0349
Epoch 1: val_loss improved from inf to 0.02024, saving model to stacked_unet_29.h5
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3103: UserWarning: You are saving your model as an HDF5 file via `model.save()`. This fi
saving_api.save_model()
107/107 [=====] - 101s 530ms/step - loss: 0.0040 - mean_absolute_error: 0.0349 - val_loss: 0.0202 - val_mean_absolute_error: 0.1027
Epoch 2/20
107/107 [=====] - ETA: 0s - loss: 0.0019 - mean_absolute_error: 0.0241
Epoch 2: val_loss improved from 0.02024 to 0.00308, saving model to stacked_unet_29.h5
107/107 [=====] - 48s 452ms/step - loss: 0.0019 - mean_absolute_error: 0.0241 - val_loss: 0.0031 - val_mean_absolute_error: 0.0359
Epoch 3/20
107/107 [=====] - ETA: 0s - loss: 0.0014 - mean_absolute_error: 0.0213
Epoch 3: val_loss improved from 0.00308 to 0.00184, saving model to stacked_unet_29.h5
107/107 [=====] - 56s 527ms/step - loss: 0.0014 - mean_absolute_error: 0.0213 - val_loss: 0.0018 - val_mean_absolute_error: 0.0260
Epoch 4/20
107/107 [=====] - ETA: 0s - loss: 0.0013 - mean_absolute_error: 0.0201
Epoch 4: val_loss improved from 0.00184 to 0.00135, saving model to stacked_unet_29.h5
107/107 [=====] - 56s 521ms/step - loss: 0.0013 - mean_absolute_error: 0.0201 - val_loss: 0.0014 - val_mean_absolute_error: 0.0216
Epoch 5/20
107/107 [=====] - ETA: 0s - loss: 0.0011 - mean_absolute_error: 0.0187
Epoch 5: val_loss improved from 0.00135 to 0.00124, saving model to stacked_unet_29.h5
107/107 [=====] - 63s 589ms/step - loss: 0.0011 - mean_absolute_error: 0.0187 - val_loss: 0.0012 - val_mean_absolute_error: 0.0212

```

#### - B4 Kiểm Thử và Đánh Giá:

Sử dụng tập dữ liệu kiểm thử để đánh giá hiệu suất của mô hình.

Đo lường các chỉ số chất lượng như độ PSNR (Peak Signal-to-Noise Ratio) hoặc SSIM (Structural Similarity Index) để đánh giá sự tương đồng giữa hình ảnh tái tạo và hình ảnh thực tế.

Kiểm tra trên các tập dữ liệu mới để đảm bảo tính tổng quát của mô hình.

#### - B5 Triển Khai vào Trang Web:

Nhúng mô hình học máy vào trang web để có thể tự động phục hồi ảnh CT từ sinogram khi người dùng tải lên dữ liệu.

#### - B6 Deploy trang Web lên server sử dụng Ngrok:

Các bước để deploy back-end bằng Ngrok:

- Vào trang <https://ngrok.com/download/> để tải Ngrok
- Sau khi tải Ngrok, vào trang <https://dashboard.ngrok.com/>, đăng ký và đăng
- Vào mục Your Authtoken để lấy token đăng nhập
- Sau khi có token đăng nhập, tiến hành gõ lệnh trên cmd hoặc chạy file ngrok.exe để kết nối đến tài khoản:

“ngrok config add-authtoken {your token}”



```
C:\Users\ntnga\Downloads\ngrok>ngrok -v
C:\Users\ntnga\Downloads\ngrok>ngrok authtoken 2aLVcyKG8jFXW0D9WJNciSTGynV_4ahd5nmJT8maUq74zwfPU
Authtoken saved to configuration file: C:\Users\ntnga\AppData\Local\ngrok\ngrok.yml
C:\Users\ntnga\Downloads\ngrok>ngrok http 5000
```

- Website AI đang chạy với địa chỉ truy cập là <http://localhost:5000/>, để truy cập được đến ứng dụng này từ bên ngoài internet, thực hiện bằng cách nhấn tiếp tục lệnh trên cmd:

Gõ lệnh để tạo một đường hầm ngrok với các tham số, loại giao thức http và cổng: “ngrok http 5000”

```
ngrok
Build better APIs with ngrok. Early access: ngrok.com/early-access

Session Status      online
Account             Nguyen Trong Nghia (Plan: Free)
Version             3.5.0
Region              Asia Pacific (ap)
Latency             -
Web Interface       http://127.0.0.1:4040
Forwarding           https://dc52-2402-800-629c-d792-ccb9-9e9b-69a-ae6e.ngrok-free.app -> http://localhost:5000

Connections
  ttl    opn    rt1    rt5    p50    p90
    0     0     0.00   0.00   0.00   0.00
```

Như hình, đường hầm được tạo ra, khi kết nối đang được duy trì, có thể truy cập ứng dụng bằng url do ngrok cung cấp như ví dụ trên hình là

<https://dc52-2402-800-629c-d792-ccb9-9e9b-69a-ae6e.ngrok-free.app/>

### Và kết quả deploy đưa code AI ra ngoài internet thành công

```

C:\Users\ntnga\Downloads\ngrok> ngrok
diagnose      diagnose connection issues
help          Help about any command
http         start an HTTP tunnel
service      run and control an ngrok service on a target operating system
start        start tunnels by name from the configuration file

ngrok (Ctrl+C to quit)

Build better APIs with ngrok. Early access: ngrok.com/early-access

Session Status
Account      online
Version     3.5.0
Region      Asia Pacific (ap)
Latency      47ms
Web Interface http://127.0.0.1:4040
Forwarding  https://dc52-2402-800-629c-d792-ccb9-9e9b-69a-ae6e.ngrok-free.app -> http://localhost:5000

Connections
ttl      opn      rt1      rt5      p50      p90
8         0         0.01     0.01     0.03     0.27

HTTP Requests
-----
GET /      200 OK
POST /convert 200 OK
GET /      200 OK
GET /aboutus 200 OK
GET /faq    200 OK
GET /      200 OK
GET /      200 OK

```

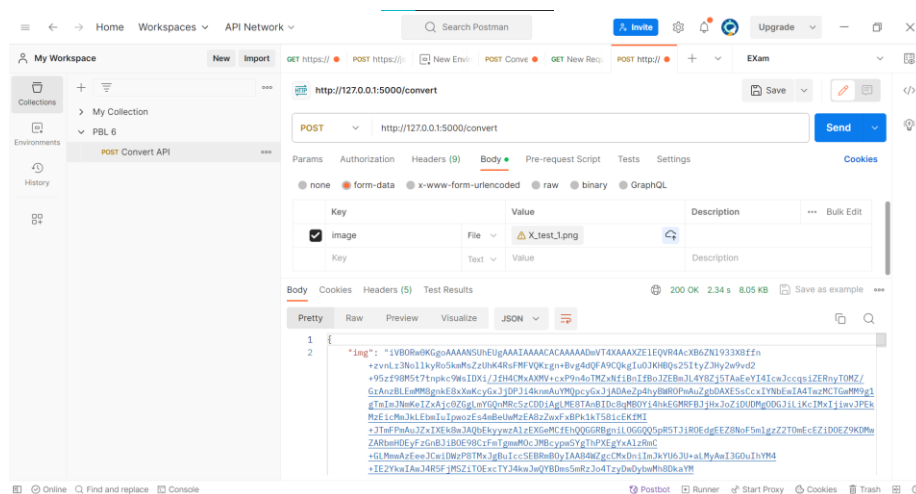
#### - B7 Kiểm thử trang Web trên server :

- Kiểm thử tự động

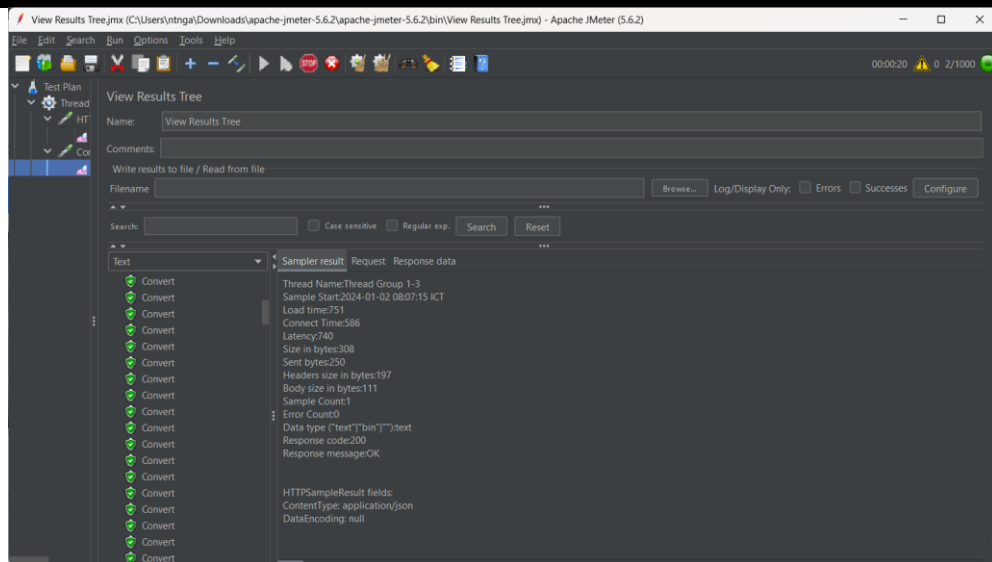
Sử dụng Selenium IDE để kiểm thử tự động :

- Kiểm thử API

Sau khi deploy server thành công ta sử dụng công cụ postman để kiểm tra API tái cấu trúc ảnh CT :



- Kiểm thử khả năng chịu tải của server



Kiểm thử phương thức POST với 100 user ảo và ramp-up period là 100s

- Nhận xét
  - Lượng yêu cầu HTTP (requests) và phản hồi (responses) chưa cao( 46.8 request per second): cho thấy hệ thống xử lý chưa tốt khi có một lượng lớn yêu cầu trong khoảng thời gian kiểm thử.
  - Thời gian phản hồi bình khá thấp ( trung bình là 175ms, thấp nhất là 154ms mà cao nhất là 351ms): cho thấy hệ thống xử lý các yêu cầu tương đối nhanh

### 3.2 Đánh giá kết quả

#### 3.2.1 Kết quả mô hình AI

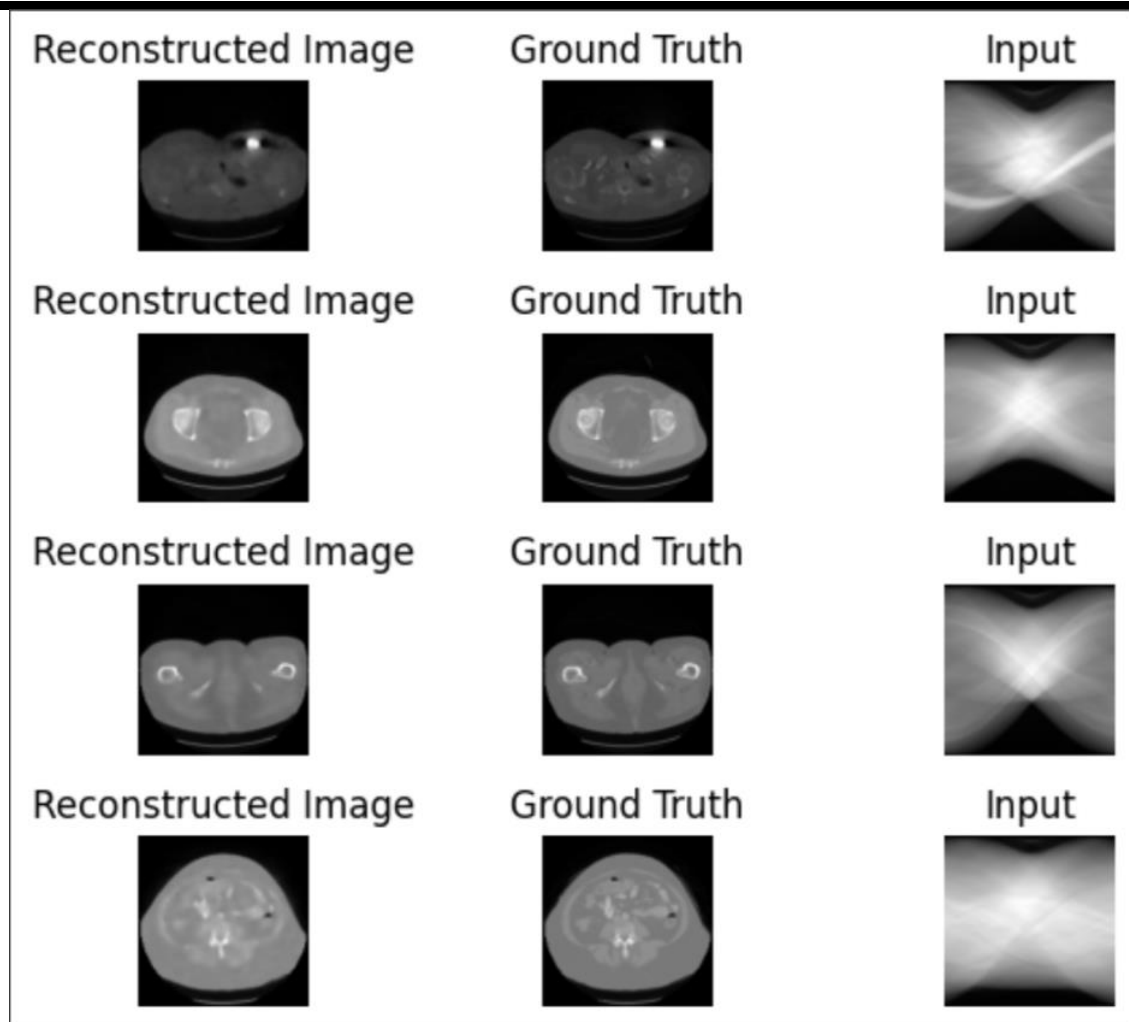
- Model được train với 137 epochs

```

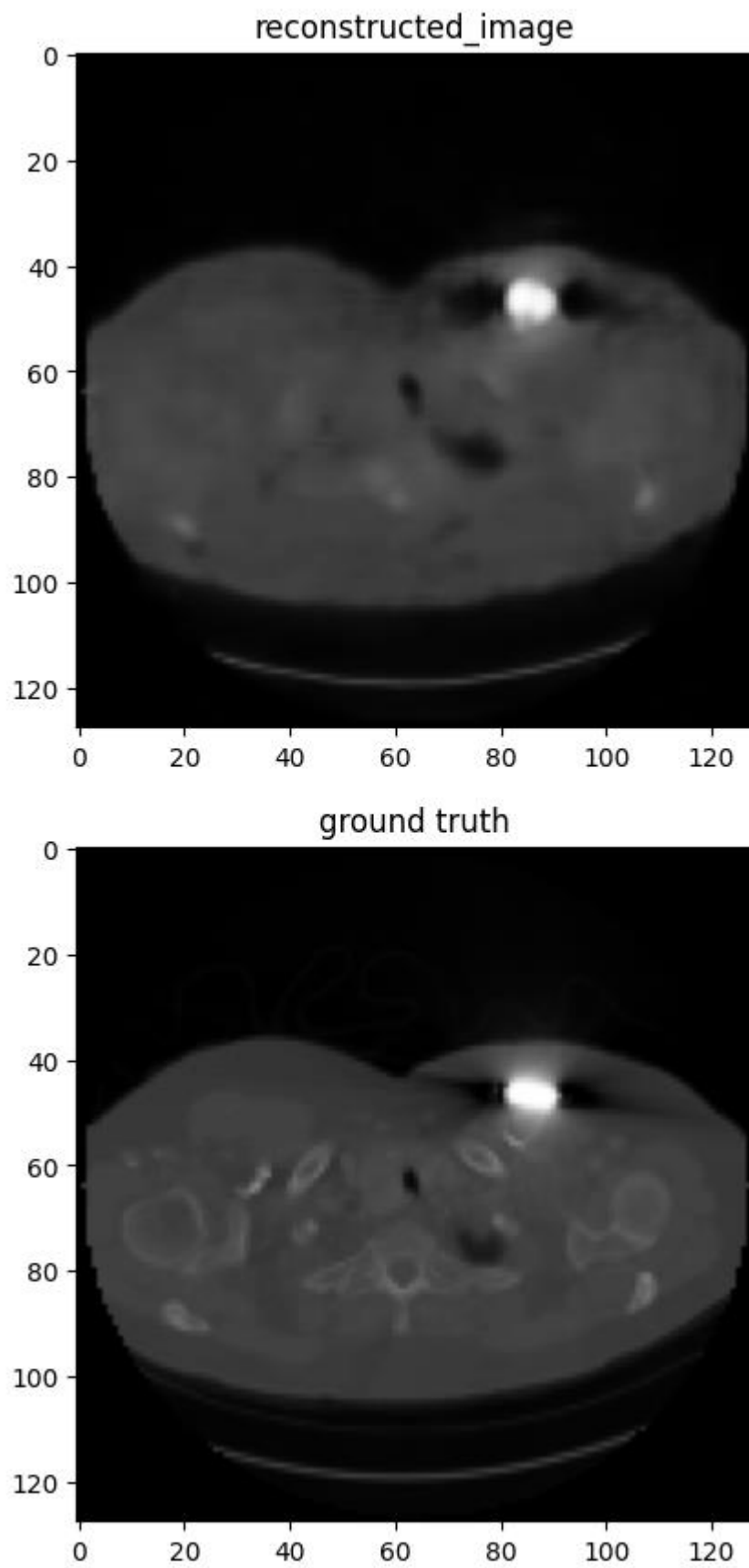
[ ] loss, mae = model.evaluate(X_val, y_val)
    loss_test, mae_test = model.evaluate(X_test, y_test)

60/60 [=====] - 6s 72ms/step - loss: 4.7169e-04 - mean_absolute_error: 0.0128
24/24 [=====] - 3s 115ms/step - loss: 4.6869e-04 - mean_absolute_error: 0.0128
    
```

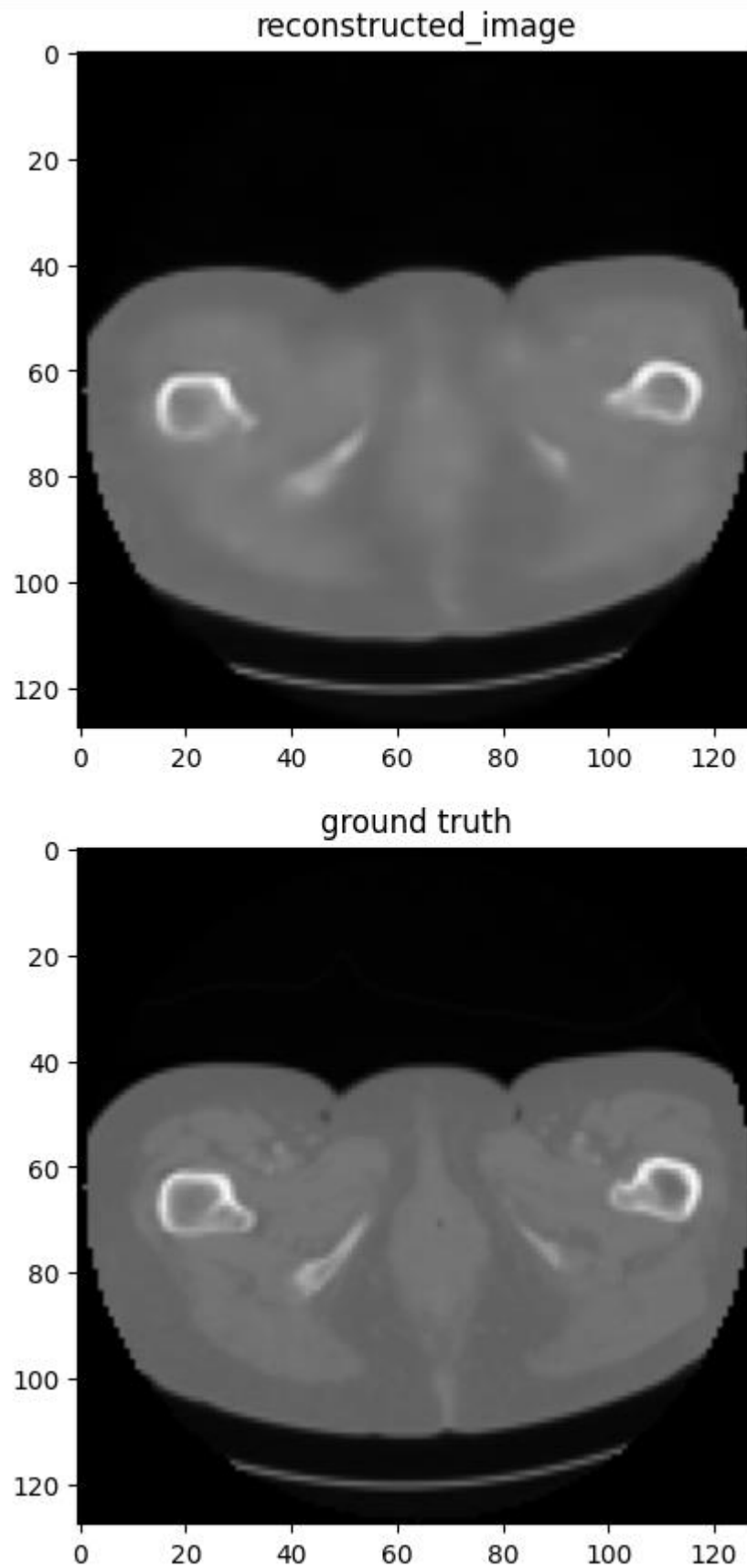
Hình 5. Test loss và mean absolute error



Hình 6. So sánh giữa ảnh tái tạo và ảnh gốc

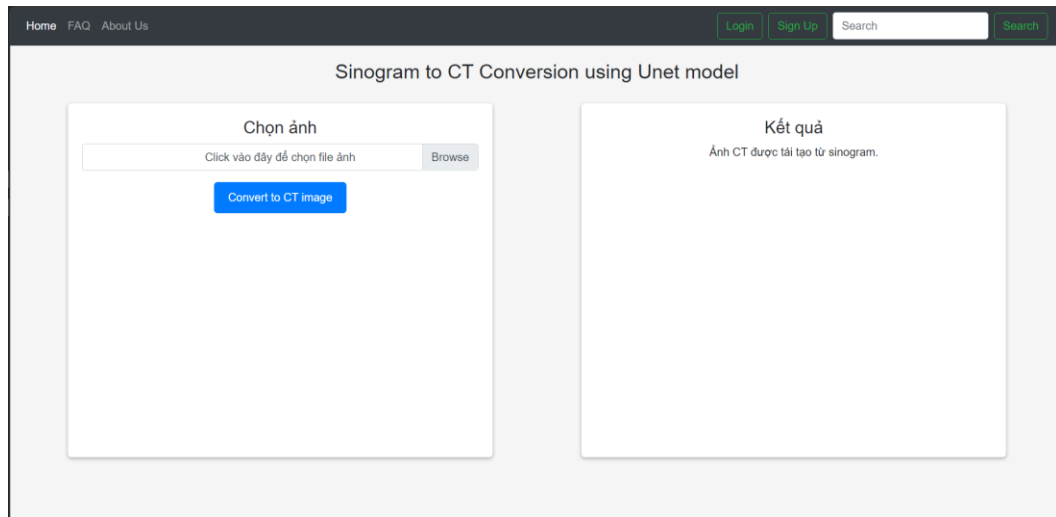


**Hình ...** So sánh chi tiết ảnh thực và ảnh tái tạo

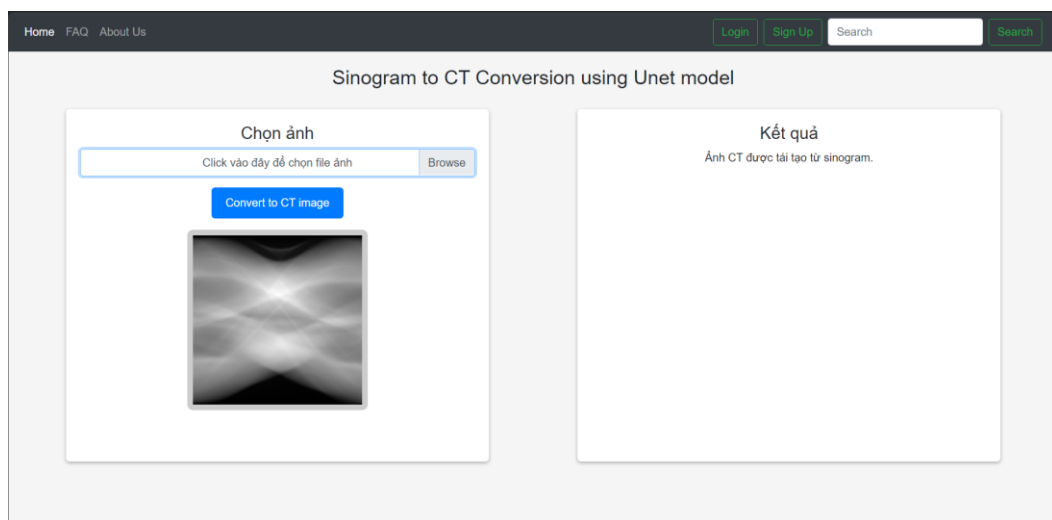


Hình ... So sánh chi tiết ảnh thực và ảnh tái tạo 2

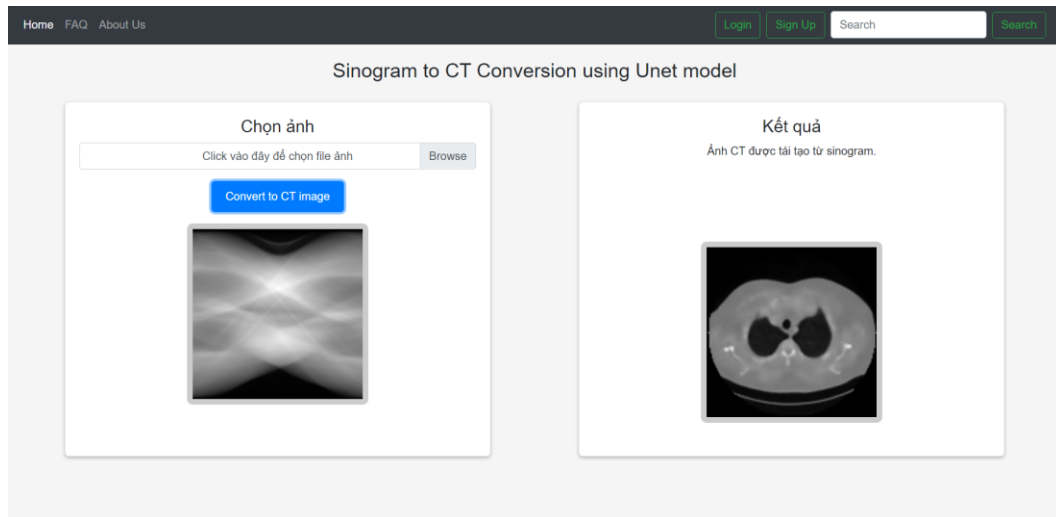
### 3.2.2 Kết quả trang web



Hình 8. Trang HomePage



Hình 9. Hiển thị hình ảnh trước xử lý từ nhập vào từ input



Hình 10. Kết quả thực thi chương trình

## CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 4.1 Kết luận

- Đã hoàn thành tương đối các yêu cầu ban đầu đề ra.
- Website nhận dạng đạt kết quả chính xác cao đối với các hình ảnh đầu vào thỏa mãn những điều kiện đã đặt ra, thời gian xử lý khá nhanh. Tuy nhiên, ứng dụng còn phụ thuộc vào các yếu tố như chất lượng của ảnh CT
  - Tạo sản phẩm gần như hoàn thiện đáp ứng nhu cầu của mục tiêu đề án đề ra cũng như nâng cao kỹ năng làm việc nhóm.
  - Có thêm nhiều kinh nghiệm trong việc cấu hình server.



## 4.2 Hướng phát triển

Website phục hồi ảnh CT từ ảnh Sinogram cho ra kết quả khả quan, tuy nhiên nó vẫn tồn tại một số hạn chế nhất định. Trong tương lai, nhóm sẽ cải tiến để trang web trở nên tối ưu hơn các mặt sau :

- Cải thiện thuật toán phục hồi ảnh để có độ chính xác cao hơn và không bị phụ thuộc vào các yếu tố bên ngoài
- Cải thiện là giao diện web để trở nên đẹp hơn, phù hợp với người dùng hơn
- Thêm tính năng đăng nhập, đăng xuất để tối ưu hóa cho người dùng, cho phép người dùng có thể xem lại lịch sử phục hồi ảnh cũng như ảnh đã được phục hồi của họ, cũng như thuận tiện cho các tính năng khác.

## TÀI LIỆU THAM KHẢO

- **Tái tạo ảnh CT<sup>[1]</sup>** là gì ? : [Chụp cắt lớp vi tính \(chụp CT\): Quy trình, ưu và nhược điểm \(tamanhhospital.vn\)](#)
- **Tensorflow<sup>[2]</sup>** <https://topdev.vn/blog/tensorflow-la-gi/>
- **JavaScript<sup>[3]</sup>** <https://tenten.vn/tin-tuc/javascript-la-gi/>
- **ReactJS<sup>[4]</sup>** <https://200lab.io/blog/reactjs-la-gi/>
- **Flask<sup>[5]</sup>** <https://vn.got-it.ai/blog/flask-trong-python-ly-do-nen-dung-flask>
- **Mô hình Unet<sup>[6]</sup>** : [Phân vùng ảnh: Các thuật toán và cơ sở dữ liệu mã nguồn mở hữu ích - VinBigData](#)
- **Mô hình stacked Unet<sup>[7]</sup>** : Computed tomography image reconstruction using stacked U-Net - written by Satoru Mizusawa, Yuichi Sei, Ryohei Orihara, Akihiko Ohsuga of The University of Electro - Communications, 1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan <https://www.sciencedirect.com/science/article/abs/pii/S0895611121000690>
- **SeleniumIDE<sup>[8]</sup>** <https://viblo.asia/p/gioi-thieu-ve-selenium-va-selenium-ide-mDYGDpVLGpx>
- **Ngrok<sup>[9]</sup>** <https://ngrok.com/>
- **CT Low Dose Reconstruction dataset<sup>[10]</sup>** : [CT Low Dose Reconstruction \(kaggle.com\)](#)