ĐỀ TÀI:

Phân đoạn Pneumothorax (tràn khí màn phổi)

Thành viên:

1. Lưu Minh Quân
2. Trần Hoàng Vũ
3. Trần Tấn Thành

Hanoi, 2-2021

MỤC LỤC

CHƯƠNG 1

GIỚI THIỆU

1.1. Giới thiệu vấn đề

Tràn khí màng phổi (Pneumothorax) có thể gây ra do vết thương ở ngực, tổn thương do bệnh phổi tiềm ẩn. Trong các trường hợp này, sự hư hại của phổi có thể đe dọa đến tính mạng của bệnh nhân.

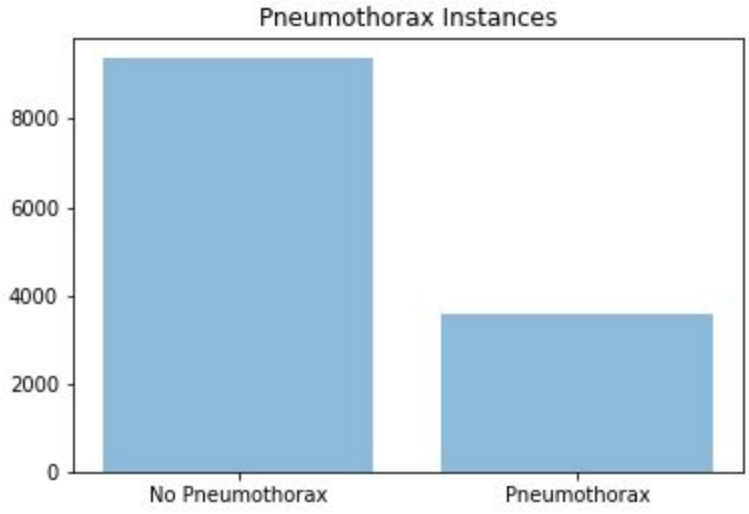
1.2. Mục tiêu

Mục tiêu chính của đề tài là phát triển thuật toán để phân đoạn tràn khí màng phổi từ một tập ảnh X-quang phổi để trợ giúp bác sĩ trong việc nhận dạng vùng tràn khí màng phổi. Cụ thể, đề tài sẽ cố gắng:

- Xây dựng các model phân đoạn vùng tràn khí màng phổi dựa trên ảnh X-quang ngực
- So sánh một số phương pháp học sâu để chọn ra mô hình phù hợp nhất với bộ dữ liệu
- Xây dựng trang web để người dùng có thể sử dụng mô hình được đào tạo nhằm phân đoạn ảnh

1.3. Dataset

Bộ dữ liệu được sử dụng trong đề tài gồm hơn 15,294 file X-quang dicom, trong đó có khoảng 12,089 trên cho tập Training và Validation, trong khi có hơn 3205 file cho tập Testing. Nhãn của bộ dataset là các ảnh phân đoạn ứng với từng ảnh trong bộ dataset.



Hình 1. Phân bố của độ dài các file âm thanh

Đa phần các file X-quang đều là ảnh không bị tràn dịch màng phổi chiếm tới 77.57% (9378).

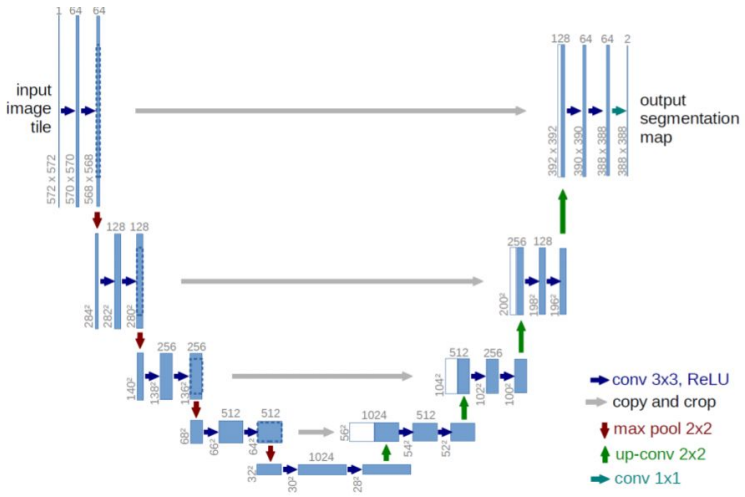
CHƯƠNG 2

CƠ SỞ LÝ THUYẾT

2.1. Giới thiệu

Kỹ thuật phân đoạn hình ảnh, nhằm phân vùng hình ảnh kỹ thuật số thành các phần hoặc đối tượng hình ảnh, cực kỳ hữu ích cho một số nhiệm vụ. Để xử lý hình ảnh y tế, cần phải xác định vị trí và phân đoạn các đối tượng hoặc vùng, chẳng hạn như u não, mô não, và chứng phình động mạch chủ bụng. Gần đây, Fully Convolutional Networks (FCN) đã cho thấy thành công lớn trong việc phân đoạn hình ảnh y tế. Thành công này chủ yếu liên quan đến kiến trúc hiệu quả, chẳng hạn như SegNet và U-Net.

2.2. Phương pháp sử dụng U-Net



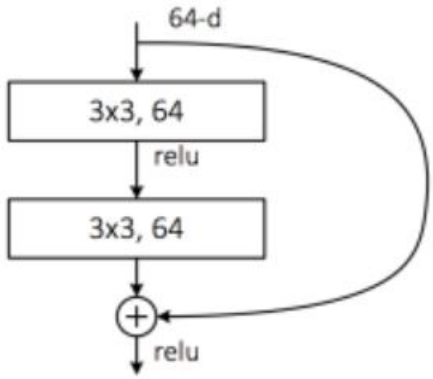
Hình 2. Kiến trúc U-Net

Biết rằng kỹ thuật phân đoạn nhằm mục đích phân vùng hình ảnh thành các phân đoạn có ý nghĩa tương tự, điều này đã giúp hiểu được các đối tượng trong hình ảnh. Học sâu đã cho thấy một bước tiến lớn trong hiệu suất mô hình cho các nhiệm vụ phân đoạn. Thành công lớn có thể được nhìn thấy trong nhiệm vụ phân đoạn ngữ nghĩa, nhằm mục đích để gắn nhãn và phân loại từng pixel trong ảnh. Điều này dẫn đến những cải tiến đặc biệt trong phân tích, chẩn đoán và điều trị trong lĩnh vực ảnh y tế, chẳng hạn như phát hiện vùng tổn thương. Một nghiên cứu [13] đã giới thiệu sự phân đoạn ngữ nghĩa end-to-end cho hình ảnh y sinh, được gọi là U-Net (Hình 2). Mạng lưới này đã đạt được đầy hứa hẹn kết quả cho phân đoạn phân đoạn tế bào. Do hiệu suất nâng cao của U-Net, nó đã trở thành một mạng phổ biến cho các nhiệm vụ phân đoạn hình ảnh y tế, chẳng hạn như phân đoạn hình ảnh X-quang phổi.

2.2.1. U-Net với encoder dựa trên ResNet

Nói chung, kiến trúc U-Net bao gồm việc học được biểu diễn của dữ liệu và sau đó mở rộng biểu diễn để cho phép phân đoạn được ảnh đầu vào. Encoder của U-Net chủ yếu dựa vào CNN

để downsample các feature map để sau đó decoder sẽ upsample những biểu diễn đã học được ở phần output của Encoder nhằm phân đoạn được ảnh
Ở phần Encoder của U-Net, ResNet được sử dụng trong đề xuất này vì sự phổ biến cũng như việc có thể tùy chỉnh nhiều layer khác nhau mang trọng số (phổ biến nhất là 34 hoặc 50 layer mang trọng số) kèm theo đó là việc học các biểu diễn ngày càng sâu. Để có thể huấn luyện mô hình sâu như trên, ResNet sử dụng Residual connection nhằm đưa kết hợp của cả output của một layer và input của layer đó vào layer tiếp theo. Nhờ đó, trong việc backpropagation, gradient của các lớp đầu tiên sẽ không bị giảm về 0 do việc nhân liên tiếp các đạo hàm liên quan trong chain rule



Hình 3. Residual Connection

ResNet34	ResNet50
$7 \times 7, 64$, stride 2	$7 \times 7, 64$, stride 2
3×3, max pool stride 2	3×3, max pool stride 2
$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$
$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
$9 \times 1, 512$, stride 1	$9 \times 1, 2048$, stride 1
$1 \times N$, avg pool stride 1	$1 \times N$, avg pool stride 1

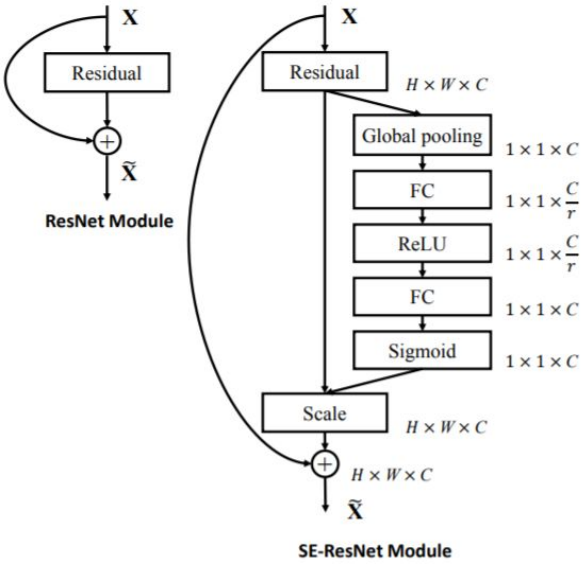
Hình 4. Kiến trúc ResNet-34 và ResNet-50

Để xây dựng Encoder, ta chỉ lấy từ những lớp đầu đến output của residual block cuối cùng trong cả ResNet-34 và ResNet-50 để đưa output thành đầu vào của decoder.
Trong Decoder, ta có những block cơ bản giúp upsample gấp đôi input đầu vào của mỗi lớp đồng thời giảm số channel của mỗi lớp xuống gấp đôi gồm các lớp transposed convolution. Cụ thể, ta sẽ có số block cơ bản này bằng với số residual block được sử dụng trong Encoder, và

output của mỗi block này sẽ được concatenate với residual block tương ứng với nó trong Encoder để trở thành input của block tiếp theo.

2.2.2. U-Net với encoder dựa trên SE-ResNext-50

Squeeze-and-Excitation Networks (SENets) giới thiệu một building block cho CNN giúp cải thiện sự phụ thuộc lẫn nhau của các kênh mà hầu như không có chi phí tính toán. Bên cạnh việc tăng hiệu suất một cách đáng kể, chúng có thể dễ dàng được thêm vào các kiến trúc hiện có. Ý tưởng chính là: Hãy thêm các tham số vào từng channel của một khối tích chập để mạng có thể điều chỉnh trọng số của các feature map.
CNNs sử dụng bộ kernel để trích xuất thông tin phân cấp từ hình ảnh. Các layer thấp hơn tìm thấy các phần nhỏ của ngữ cảnh như các cạnh, trong khi các layer cao hơn có thể phát hiện khuôn mặt, văn bản hoặc các hình dạng hình học phức tạp khác. Họ trích xuất bất cứ thứ gì cần thiết để giải quyết công việc một cách hiệu quả.
Tất cả điều này hoạt động bằng cách kết hợp thông tin không gian và channel của một ảnh. Các kernel khác nhau trước tiên sẽ tìm các đặc điểm không gian trong mỗi channel đầu vào trước khi thêm thông tin trên tất cả các kênh đầu ra có sẵn. Do đó, mạng sẽ đánh trọng số như nhau cho mỗi channel khi tạo ra output feature maps. SENet sẽ tạo ra cơ chế nhằm đánh trọng số khác nhau cho mỗi channel một cách thích hợp. Để thực hiện, mạng có được các thông tin của tất cả các channel bằng việc lấy trung bình các giá trị theo chiều dài và rộng của dữ liệu. Sau đó chúng được vào mạng nơ-ron 2 lớp để sinh ra vecto có cùng kích thước. Khi này, các giá trị này có thể được sử dụng làm trọng số trên feature map ban đầu, chia tỷ lệ từng kênh dựa trên tầm quan trọng của nó.



Hình 5. SE Module trong ResNet

Output size	SE-ResNeXt-50 (32 × 4d)
112 × 112	conv, 7 × 7, 64, stride 2
56 × 56	max pool, 3 × 3, stride 2
	$\begin{bmatrix} \text{conv}, 1 \times 1, 128 \\ \text{conv}, 3 \times 3, 128 \quad C = 32 \\ \text{conv}, 1 \times 1, 256 \\ fc, [16, 256] \end{bmatrix} \times 3$
28 × 28	$\begin{bmatrix} \text{conv}, 1 \times 1, 256 \\ \text{conv}, 3 \times 3, 256 \quad C = 32 \\ \text{conv}, 1 \times 1, 512 \\ fc, [32, 512] \end{bmatrix} \times 4$
14 × 14	$\begin{bmatrix} \text{conv}, 1 \times 1, 512 \\ \text{conv}, 3 \times 3, 512 \quad C = 32 \\ \text{conv}, 1 \times 1, 1024 \\ fc, [64, 1024] \end{bmatrix} \times 6$
7×7	$\begin{bmatrix} \text{conv}, 1 \times 1, 1024 \\ \text{conv}, 3 \times 3, 1024 \quad C = 32 \\ \text{conv}, 1 \times 1, 2048 \\ fc, [128, 2048] \end{bmatrix} \times 3$
1 × 1	global average pool, 1000-d fc, softmax

Hình 6. Kiến trúc SE-ResNeXt-50

Tương tự như phương pháp sử dụng U-net với encoder dựa trên ResNet, ta cũng chỉ lấy những lớp input đầu tới lớp residual block cuối cùng dùng lớp input của decoder.

2.3. Hàm mất mát và cách đánh giá

2.3.1. Cách đánh giá

Dice coefficient có thể được sử dụng để so sánh sự trùng khớp về pixel giữa phân đoạn được dự đoán và ground-truth tương ứng của nó.

$$\frac{2 * |X \cap Y|}{|X| + |Y|}$$

Trong đó, **X** là tập các pixel trong ảnh segment dự đoán và **Y** là tập các pixel trong ảnh segment ground-truth

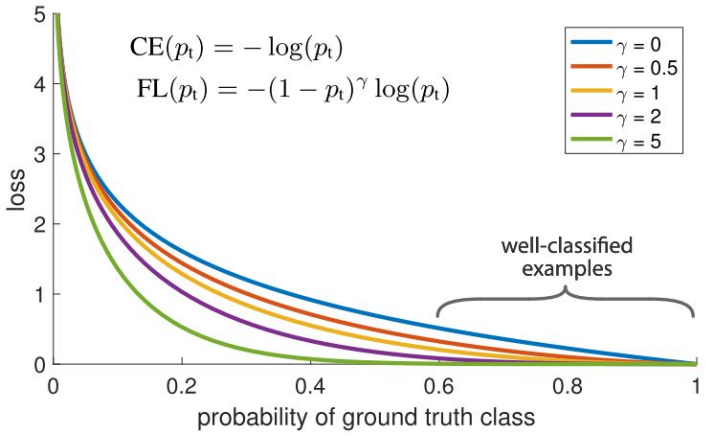
2.3.2. Hàm mất mát

Ngoài việc sử dụng Binary-Cross entropy (BCE) để tối thiểu hóa sự sai khác giữa ảnh ground-truth segment và ảnh segment dự đoán, ta còn sử dụng thêm Dice loss để tối đa hóa cách đánh giá dựa vào Dice coefficient. Cụ thể, Dice loss:

$$1 - \frac{2 \times |\mathrm{X} \bigcap \mathrm{Y}|}{|\mathrm{X}| + |\mathrm{Y}|}$$

Trong đó, **X** là tập các pixel trong ảnh segment dự đoán và **Y** là tập các pixel trong ảnh segment ground-truth

Ngoài BCE và Dice loss, focal loss còn được sử dụng trong đề xuất vì bộ dữ liệu có sự mất cân bằng lớn giữa các foreground, pixel chỉ ra bệnh, và background, pixel không chỉ ra bệnh. Cụ thể, Focal loss được sử dụng bằng việc thay đổi một chút hàm loss cross-entropy nhằm giảm trọng số mất mát đối với các instance được phân loại tốt



Hình 6. So sánh Cross entropy loss và Focal loss

2.3. Cơ chế xét ngưỡng cho ảnh phân đoạn

Đầu ra của mô hình là các xác suất để xác định pixel đang xét có là pixel chỉ ra bệnh hay không. Nghiên cứu đề xuất cách sử dụng 3 ngưỡng (*top score threshold*, *min contour area*, *bottom score threshold*) để xác định được pixel nào của output là pixel chứa bệnh.
Đầu tiên, ta đếm có bao nhiêu pixel lớn hơn *top score threshold* và ảnh không có bệnh khi số lượng pixel trên nhỏ hơn *min contour area*, nhưng khi số pixel trên lớn hơn *min contour area*, ta chỉ xét những pixel có xác suất lớn hơn *bottom score threshold*.

CHƯƠNG 3

PHƯƠNG PHÁP THỰC HIỆN VÀ KẾT QUẢ

3.1. Tiền xử lý

3.1.1. Data Augmentation

Vì bộ dataset chỉ có hơn 10,000 ảnh X-quang phổi và đa phần trong số đó là ảnh không bị bệnh nên việc sinh thêm ảnh nhờ vào các phương pháp data augmentation là một việc rất cần thiết. Các phương pháp mà nghiên cứu này đề xuất gồm:

- Horizontal Flip
- Random Contrast: ngẫu nhiên thay đổi độ tương phản của ảnh
- Random Brightness: ngẫu nhiên thay đổi độ sáng của ảnh
- ShiftScaleRotate: ngẫu nhiên dịch ảnh, phóng to/ thu nhỏ, và xoay ảnh

3.1.2. Sliding sample rate

Vì số ảnh không có bệnh vượt gấp khoảng 3 lần so với số ảnh không có bệnh, nên sự chênh lệch lớn đó có thể làm ảnh hưởng đến việc học của mạng. Vì thế, ta có thể lấy mẫu các ảnh sao cho số lượng ảnh có bệnh chiếm k% trong khi số ảnh không có bệnh chiếm phần còn lại (1 - k)%. Dựa vào các phần huấn luyện của các nghiên cứu, ta thấy k = 80% cho ra kết quả huấn luyện khá tốt.

3.2. Modeling

3.2.1. U-Net với encoder dựa trên ResNet-34

Qua thực nghiệm, ta tìm ra rằng tốc độ học là 1e-4, và weight decay là 5e-6 là những giá trị tốt nhất cho việc huấn luyện mô hình. Cùng với đó, ta sử dụng Adam optimizer với batch size là 2. Mô hình được huấn luyện trên GPU Nvidia RTX 2070. Đồng thời trọng số cho cả ba loss lần lượt là 3 cho BCE, 1 cho Dice loss và 4 cho Focal loss khi công các loss lại. Đồng thời bộ 3 ngưỡng tối ưu cho mô hình là (*top score threshold*, *min contour area*, *bottom score threshold*) = (0.75, 1000, 0.3)

Model	Validation accuracy	Test accuracy
U-Net with ResNet-34	78.6%	60%

3.2.2. U-Net với encoder dựa trên ResNet-50

Qua thực nghiệm, ta tìm ra rằng tốc độ học là 1e-4, và weight decay là 5e-6 là những giá trị tốt nhất cho việc huấn luyện mô hình. Cùng với đó, ta sử dụng Adam optimizer với batch size là 2. Mô hình được huấn luyện trên GPU Nvidia RTX 2070. Đồng thời trọng số cho cả ba loss lần lượt là 2 cho BCE, 1 cho Dice loss và 2 cho Focal loss khi công các loss lại. Đồng thời bộ 3 ngưỡng tối ưu cho mô hình là (*top score threshold*, *min contour area*, *bottom score threshold*) = (0.75, 2000, 0.4)

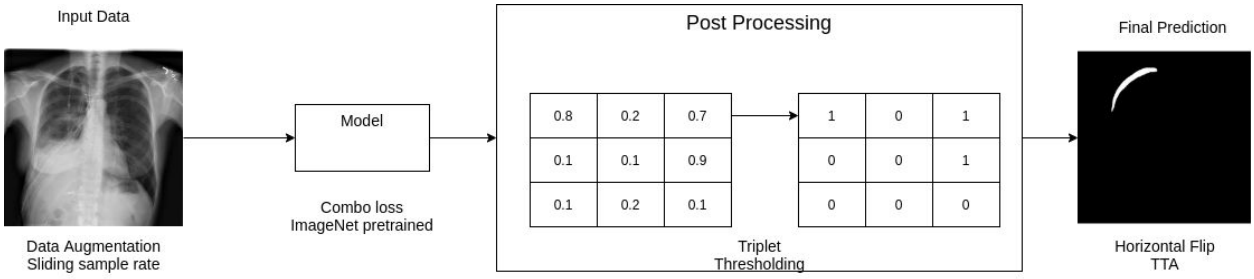
Model	Validation accuracy	Test accuracy
U-Net with ResNet-50	86.8%	88.2%

3.2.2. U-Net với encoder dựa trên SE-ResNext-50

Qua thực nghiệm, ta tìm ra rằng tốc độ học là 1e-4, và weight decay là 5e-6 là những giá trị tốt nhất cho việc huấn luyện mô hình. Cùng với đó, ta sử dụng Adam optimizer với batch size là 2. Mô hình được huấn luyện trên GPU Nvidia RTX 2070. Đồng thời trọng số cho cả ba loss lần lượt là 3 cho BCE, 1 cho Dice loss và 4 cho Focal loss khi công các loss lại. Đồng thời bộ 3 ngưỡng tối ưu cho mô hình là (*top score threshold*, *min contour area*, *bottom score threshold*) = (0.6, 3000, 0.4)

Model	Validation accuracy	Test accuracy
U-Net with SE-ResNext-50	87.2%	89.4%

3.3. Pipeline bài toán



Hình 7. Pipeline bài toán phân đoạn.

Sau khi các files X-quang được extract thành file ảnh png, data được sinh thêm nhờ vào data augmentation và sample sao cho mỗi epoch được 80% mẫu là có bệnh. Model được sử dụng là các mô hình được pretrained trên tập ImageNet và có tất cả 3 loss được cộng lại để tối ưu hóa mô hình. Sau khi train mỗi epoch, một tập các triplet được thử để tìm ra xem bộ triplet

thresholding nào đạt kết quả cao nhất. Khi inference, ngoài việc phân đoạn ảnh đầu vào dựa trên kết quả training, một ảnh horizontal flip dựa trên ảnh đầu vào được sử dụng để tăng thêm dữ liệu khi inference.

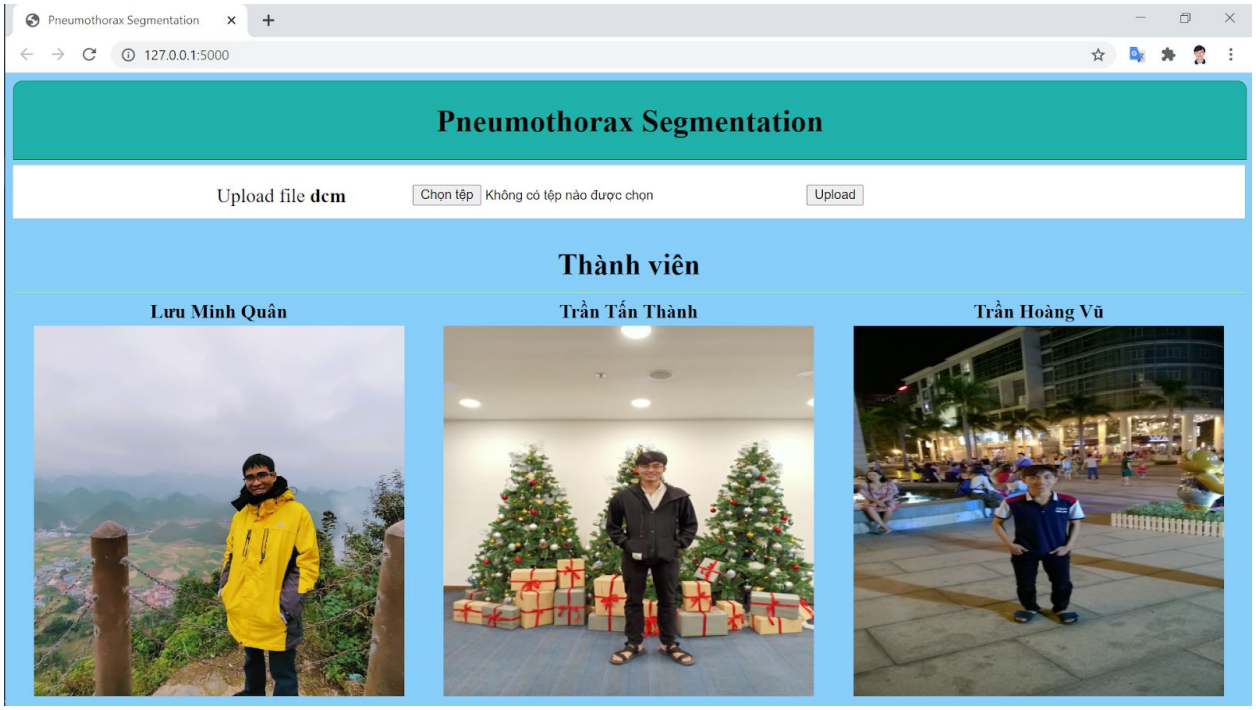
3.4. Nhận xét

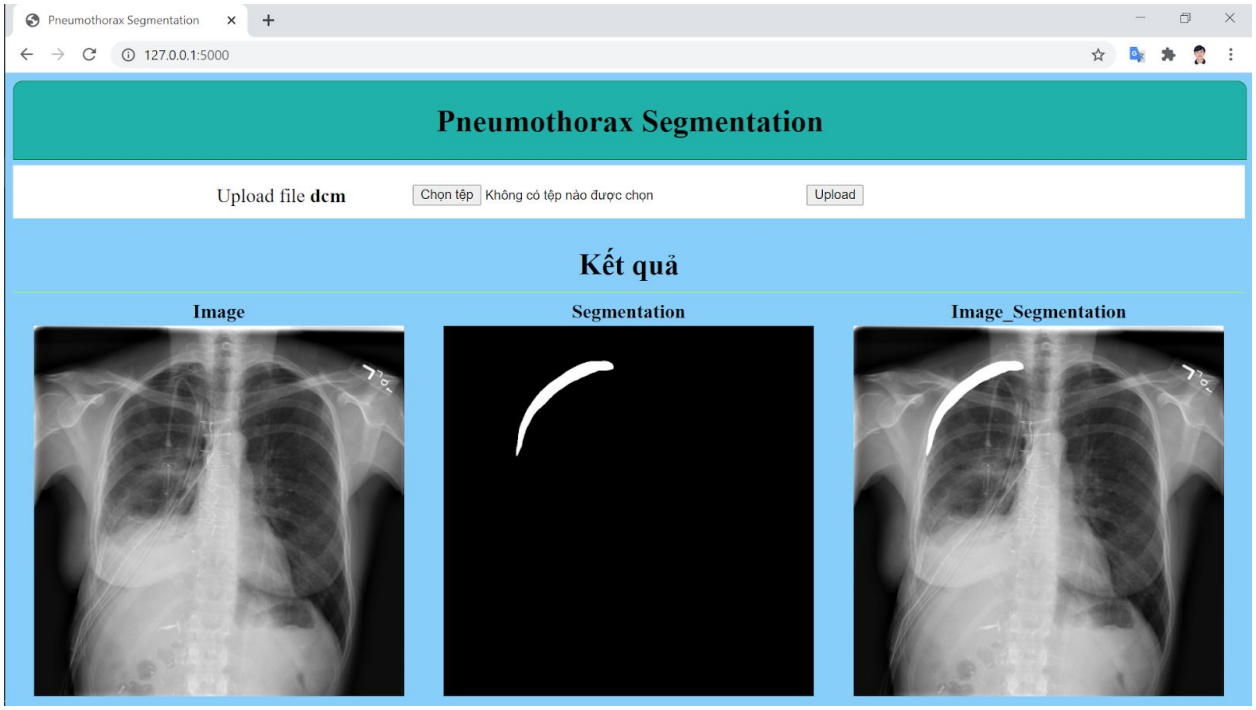
Hầu hết các phương pháp U-Net dựa trên ResNet đều cho kết quả tốt khi phân đoạn ảnh.

CHƯƠNG 4

DEMO

Model sau khi được huấn luyện được đưa lên một dịch vụ web để người dùng có thể dễ dàng đăng file X-quang phổi và sử dụng hệ thống phân đoạn





Sau đó hệ thống sẽ xuất ra hình X-quang phổi, hình ảnh phân đoạn kết quả và khi chèn ảnh phân đoạn lên ảnh gốc.