

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Xác định vùng bóng mờ trong ảnh chụp CT của bệnh nhân sau khi nhiễm COVID-19

HOÀNG ANH ĐỨC
duc.ha183711@sis.hust.edu.vn

Ngành: Kỹ thuật máy tính

Giảng viên hướng dẫn: TS. Nguyễn Hồng Quang

Chữ kí GVHD

Khoa: Kỹ thuật máy tính

Trường: Công nghệ thông tin và Truyền thông

HÀ NỘI, 08/2023

LỜI CẢM ƠN

Em xin gửi lời cảm ơn đến thầy Nguyễn Hồng Quang đã hướng dẫn, định hướng và chỉ bảo em trong thời gian làm đồ án tốt nghiệp. Em cũng xin cảm ơn thầy Ngô Văn Linh, cô Nguyễn Thị Oanh là những người đã truyền dạy cho em những kiến thức nền tảng để có thể thực hiện tốt ĐATN này đồng thời em cũng xin cảm ơn các thầy cô đã đồng hành và dùn dắt em trong suốt quá trình học tại trường Đại học Bách Khoa Hà Nội.

Mình xin cảm ơn bạn Lê Việt Anh và những người bạn, người đồng đội trong đội bóng IT2++, các bạn đã giúp mình rất nhiều trong việc thực hiện ĐATN. Cảm ơn các bạn đã giúp mình cân bằng giữa đam mê và học tập cũng như đã hỗ trợ mình trong suốt quãng đời sinh viên.

Cuối cùng con xin cảm ơn bố mẹ và em trai đã luôn ở bên cạnh động viên, chăm sóc, luôn là nguồn động lực giúp con vượt qua mọi khó khăn và hoàn thành tốt mục tiêu của mình.

TÓM TẮT NỘI DUNG ĐỒ ÁN

Đại dịch Covid-19 đã gây ra một tác động vô cùng lớn đối với sức khỏe của con người trên toàn cầu. Việc phát hiện những triệu chứng của việc nhiễm virus trên phổi của bệnh nhân sau khi khỏi bệnh đóng một vai trò vô cùng quan trọng trong việc phòng ngừa và điều trị sớm cho bệnh nhân. Một trong những dấu hiệu phổ biến nhất đó là hình ảnh bóng mờ. Thủ thách phân đoạn tổn thương trên ảnh chụp CT phổi được tổ chức bởi nhóm chuyên gia và các nhà nghiên cứu đến từ Children's National Hospital và Viện Y tế Quốc gia Hoa Kỳ (NIH) nhằm mục đích đánh giá và so sánh những giải pháp hiện đại trong việc phân đoạn hình ảnh bóng mờ trên hình chụp CT phổi bệnh nhân sau COVID 19. Các phương pháp được đề xuất trong thử thách hầu hết đều dựa trên kiến trúc nnUnet. Framework này cung cấp một kiến trúc chung cho việc phân đoạn hình ảnh y tế, vì thế một số phương pháp chung trong nnUnet sẽ không phù hợp với dữ liệu bài toán. Trong đồ án tốt nghiệp này, tôi dựa trên phương pháp nnUnet đưa ra để xây dựng một thuật toán học máy thực hiện phân đoạn hình ảnh bóng mờ trên ảnh chụp CT phổi bệnh nhân sau khi mắc COVID 19. Tôi đề xuất một phương pháp chuẩn hóa hình ảnh dựa trên kết quả của mô hình phân đoạn phổi. Các hình ảnh bóng mờ sẽ được phân đoạn dựa trên sự kết hợp của hai mô hình 2D Unet và 3D Unet tích hợp cơ chế chú ý. Ngoài ra tôi tìm sự kết hợp tốt nhất của hai mô hình khi sử dụng các hàm lỗi kết hợp khác nhau. Đồ án này hướng đến một phương phân đoạn hiệu quả trong điều kiện phần cứng và bộ dữ liệu hạn chế đồng thời tạo ra một ứng dụng web triển khai kết quả của phương pháp trên.

Sinh viên thực hiện
(Ký và ghi rõ họ tên)

MỤC LỤC

| | |
|---|-----------|
| CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI..... | 1 |
| 1.1 Đặt vấn đề..... | 1 |
| 1.2 Các giải pháp hiện tại và hạn chế | 2 |
| 1.3 Mục tiêu và định hướng giải pháp | 3 |
| 1.4 Đóng góp của đồ án | 4 |
| 1.5 Bố cục đồ án | 4 |
| CHƯƠNG 2. NỀN TẢNG LÝ THUYẾT | 6 |
| 2.1 Ngữ cảnh của bài toán..... | 6 |
| 2.2 Các kết quả nghiên cứu tương tự | 7 |
| 2.3 Bài toán phân đoạn hình ảnh y tế..... | 11 |
| 2.4 Mạng nơ-ron tích chập | 11 |
| 2.5 Các kiến thức xử lý ảnh..... | 12 |
| 2.5.1 Tìm thành phần liên thông | 12 |
| 2.5.2 Các toán tử hình thái | 13 |
| 2.5.3 Thuật toán lấy ngưỡng OSTU..... | 13 |
| 2.6 nnUnet..... | 13 |
| 2.7 Tensorflow..... | 15 |
| 2.8 Flask | 16 |
| CHƯƠNG 3. PHƯƠNG PHÁP ĐỀ XUẤT..... | 17 |
| 3.1 Phương pháp chung..... | 17 |
| 3.2 Giai đoạn 1: Phân đoạn phổi | 20 |
| 3.3 Giai đoạn 2: Phân đoạn GGOs | 23 |
| 3.3.1 Tiền xử lý dữ liệu | 23 |
| 3.3.2 Mô hình 2D U-net..... | 26 |

| | |
|---|-----------|
| 3.3.3 Mô hình 3D U-net..... | 28 |
| 3.3.4 Hậu xử lý | 30 |
| 3.3.5 Hàm lỗi | 33 |
| 3.3.6 Website ứng dụng mô hình..... | 36 |
| CHƯƠNG 4. ĐÁNH GIÁ THỰC NGHIỆM..... | 38 |
| 4.1 Các tham số đánh giá | 38 |
| 4.1.1 Dice Coefficient..... | 38 |
| 4.1.2 Jaccard coefficient | 38 |
| 4.1.3 Hausdorff distance | 39 |
| 4.1.4 Surface Dice..... | 39 |
| 4.1.5 Sensitivity và specificity | 40 |
| 4.2 Thủ nghiệm mô hình trên các hàm lỗi..... | 40 |
| 4.3 Kết quả của phương pháp so với các mô hình phổ biến..... | 42 |
| 4.4 So sánh kết quả với các nhóm thực hiện trong thử thách COVID-19-20..... | 43 |
| 4.5 Kết quả dự đoán mô hình trên một số hình ảnh chụp CT | 44 |
| 4.6 Ứng dụng thử nghiệm..... | 46 |
| CHƯƠNG 5. KẾT LUẬN | 48 |
| 5.1 Kết luận | 48 |
| 5.2 Hướng phát triển trong tương lai | 48 |
| TÀI LIỆU THAM KHẢO..... | 50 |
| PHỤ LỤC..... | 50 |

DANH MỤC HÌNH VẼ

| | |
|--|----|
| Hình 2.1 Kiến trúc nnUnet *Nguồn: nnUnet (https://github.com/MIC-DKFZ/nnUNet) | 15 |
| Hình 3.1 Trực quan hóa bộ dữ liệu | 18 |
| Hình 3.2 Tổng quan phương pháp | 19 |
| Hình 3.3 Giai đoạn 1: Phân đoạn phổi sử dụng học bán giám sát | 20 |
| Hình 3.4 So sánh hình ảnh trước khi chuẩn hóa và sau khi chuẩn hóa bằng Z-score | 25 |
| Hình 3.5 2D Unet Architecture | 27 |
| Hình 3.6 Attention Block | 28 |
| Hình 3.7 3D Unet Architecture | 29 |
| Hình 3.8 Kiến trúc của Website ứng dụng | 37 |
| Hình 4.1 Một số kết quả phân đoạn | 45 |
| Hình 4.2 Giao diện Website ứng dụng | 47 |

DANH MỤC BẢNG BIỂU

| | |
|--|----|
| Bảng 4.1 Bảng so sánh sự kết hợp của các mô hình với các hàm lỗi khác nhau | 41 |
| Bảng 4.2 Bảng so sánh kết quả của phương pháp đề xuất với các phương pháp phân đoạn phổ biến | 42 |
| Bảng 4.3 Bảng so sánh kết quả của phương pháp đề xuất với các thuật toán hàng đầu trong thử thách COVID-19-20 | 44 |

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

| Thuật ngữ | Ý nghĩa |
|------------------|--|
| ĐATN | Đồ án tốt nghiệp |
| CCL | Đánh số thành phần liên thông (Connected Component Labeling) |
| CL | Hàm lỗi Combo loss |
| CNN | Mạng nơ ron tích chập (Convolution Neural Network) |
| COVID-19-20 | Thử thách phân đoạn hình ảnh bóng mờ trên ảnh chụp CT bệnh nhân sau COVID-19 |
| CT | Computed Tomography |
| DBL | Hàm lỗi Dice Boundary Loss |
| DFL | Hàm lỗi Dice Focal Loss |
| DSC | Dice similarity coefficient |
| GGOs | Hình ảnh bóng mờ (Ground Glass Opacities) |

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Bài toán phân đoạn hình ảnh bóng mờ trên ảnh chụp CT phổi bệnh nhân sau khi mắc COVID 19 (Ground Glass Opacity Segmentation) là một bài toán quan trọng trong lĩnh vực xử lý ảnh y tế. Sau khi một bệnh nhân mắc COVID 19 và khỏi bệnh, việc đánh giá sự tổn thương và phục hồi là rất quan trọng để đưa ra quyết định về điều trị và theo dõi tình trạng sức khỏe của bệnh nhân.

Bài toán GGOs nhằm mục đích xác định và phân đoạn các vùng bóng mờ trên hình ảnh CT phổi của bệnh nhân. Hình ảnh chụp CT phổi bệnh nhân có thể thu được từ nhiều thiết bị khác nhau, đồng nghĩa với kích thước mỗi hình ảnh sẽ khác nhau và độ phân giải hình ảnh cũng sẽ khác nhau ở mỗi bệnh nhân và mỗi thiết bị. Vì vậy chúng ta cần đưa ra một quy trình chuẩn hóa kích thước chung cho các hình ảnh. Mức độ biểu hiện GGOs trên ảnh chụp CT của các bệnh nhân khác nhau theo mức độ nghiêm trọng của bệnh và cả theo độ tuổi, GGOs có thể có kích thước từ rất nhỏ cho đến rất lớn và độ sáng so với phổi cũng khác nhau tùy theo tình trạng bệnh. Một vấn đề khác đó là độ tương phản giữa GGOs và các mô xung quanh thường thấp nên việc phân đoạn chính xác cạnh của GGOs trở nên khó khăn hơn nhiều. Việc phân đoạn chính xác vùng bóng mờ trên hình ảnh CT sẽ giúp các y bác sĩ đánh giá và theo dõi quá trình điều trị của bệnh nhân hiệu quả, từ đó đưa ra quyết định điều trị tốt hơn.

Lý do tôi chọn bài toán này đến từ sự cần thiết và tầm quan trọng của việc phân đoạn hình ảnh bóng mờ trong hình chụp CT phổi cũng như những bài toán phân đoạn hình ảnh y tế bằng các phương pháp hiện đại như xử lý hình ảnh và học máy sẽ giúp cải thiện quá trình chuẩn đoán, điều trị và theo dõi tình trạng bệnh nhân một cách hiệu quả hơn. Cho đến hiện tại, đã có nhiều cuộc thi được tổ chức và nhiều nghiên cứu được đưa ra để giải quyết bài toán này. Những phương pháp phổ biến cho bài toán bao gồm phân đoạn GGOs sử dụng mạng học sâu, phân đoạn GGOs sử dụng trí tuệ nhân tạo và phân đoạn GGOs không sử dụng mạng nơ-ron. Các phương pháp này đã giải quyết khá tốt vấn đề về độ phân giải và đạt những kết quả tích cực cho bài toán GGOs. Tuy nhiên việc phân đoạn chính xác ở những hình ảnh có độ tương phản thấp hay những ca bệnh có tình trạng bệnh nhẹ vẫn chưa được giải quyết một cách tối ưu. Đề án này đề xuất một phương pháp sử dụng học sâu kết hợp giữa mô hình phân đoạn 2D và 3D để xử lý những vấn đề nêu trên.

1.2 Các giải pháp hiện tại và hạn chế

Theo bản tổng hợp kết quả của thử thách COVID-19-20 [1] Một trong những phương pháp đạt điểm Dice Score cao nhất trong thử thách đến từ nhóm nghiên cứu thuộc Đại học Bách Khoa Tây Bắc (Trung Quốc) với phương pháp Học bán giám sát để phân đoạn tổn thương trên ảnh chụp CT ngực. Nhóm sử dụng nnUnet như một mạng xương sống để triển khai mô hình. Dữ liệu chưa có nhãn của nhóm được lấy từ bộ dữ liệu TCIA. Phương pháp này đạt điểm Dice score cao nhất trong thời gian diễn ra thử thách với 76,65%. Cũng trong báo cáo này, một phương pháp khác được đưa ra bởi Phòng Điện toán Hình ảnh Y tế thuộc Trung tâm Nghiên cứu Ung thư Đức là sử dụng Framework nnUnet với năm cấu hình 3D U-net với kiến trúc mạng và cách chuẩn hóa dữ liệu khác nhau. Các siêu tham số được lựa chọn cho từng cấu hình là mặc định và không áp dụng các phương pháp hậu xử lý. Kết quả của phương pháp này đạt 75.43 % Dice Score. Nhóm nghiên cứu đến từ Lynbrook High School, USA đề xuất một phương pháp dựa trên sự kết hợp của mô hình phân đoạn 2D và 3D. Các mô hình Unet được đào tạo có đầu vào khác nhau và sử dụng làm lỗi khác nhau, sau đó kết hợp 2 trong số các mô hình đã đào tạo để thực hiện phân đoạn hình ảnh. Các cấu hình Unet bao gồm: Unet 2D, Unet 3D độ phân giải thấp, Unet 3d độ phân giải cao, Unet 3D phân tầng. Có 3 hàm lỗi được sử dụng cho từng cấu hình bao gồm DiceCE loss, DiceTopK loss và DiceFocalLoss. Nhóm nghiên cứu đến từ đại học Nagoya, Nhật Bản cũng lựa chọn nnUNet để thực hiện huấn luyện trên 2 loại cấu hình 2D và 3D độ phân giải thấp. Patch size cho từng cấu hình lần lượt là 512x512 cho mô hình 2D và 28x256x256 cho mô hình 3D độ phân giải thấp. Hai mô hình được hợp nhất bằng cách tính trung bình cộng của 2 kết quả dự đoán từ hai mô hình và thực hiện hậu xử lý bằng một số phương pháp xử lý ảnh. Hầu hết các phương pháp được sử dụng trong thử thách này đều sử dụng framework nnUnet [2] và thường áp dụng các siêu tham số mặc định của framework được cấu hình để phù hợp với nhiều bài toán phân đoạn ảnh y tế tuy nhiên không phù hợp với bài toán này.

Daryl L. X. Fung và các cộng sự đề xuất một mô hình tự học có giám sát dựa trên mạng Inf-Net [3], [4]. Tác giả kết hợp mô hình học sâu với những phương pháp xử lý ảnh tiên tiến như phục hồi hình ảnh bằng mạng GAN, focal loss, tối ưu hóa lookahead. Phương pháp này chỉ sử dụng ảnh đầu vào 2D và không sử dụng các phương pháp tiền xử lý dữ liệu hay tăng cường dữ liệu dẫn đến việc phân đoạn hình ảnh không đạt kết quả tốt dù mô hình tương đối phức tạp.

Cũng trong năm 2021, D. Muller và các cộng sự đề xuất một giải pháp phân đoạn GGOs trên hình ảnh CT phổi bệnh nhân để chuẩn đoán sớm bệnh COVID 19 [5]. Giải pháp được giới thiệu nhằm giả quyết bài toán GGOs trong điều kiện dữ

liệu huấn luyện hạn chế, D. Muller và các cộng sự phát triển một quy trình học sâu dựa trên quy trình MIScnn [6]. Dữ liệu ban đầu sẽ được tiền xử lý trước khi đưa vào huấn luyện bằng các phương pháp lấy mẫu lại và chuẩn hóa theo thang màu Hounsfield(HU). Dữ liệu sau đó được tăng cường bằng nhiều phương pháp khác nhau và được đưa vào huấn luyện với mô hình 3D Unet tiêu chuẩn. Tuy kết quả của phương pháp tốt hơn những phương pháp được đề xuất trước đó tuy nhiên phương pháp này vẫn chưa giải quyết tối ưu những trường hợp GGOs có kích thước nhỏ và mờ hay những trường hợp GGOs có độ tương phản thấp so với thành phổi.

Một nghiên cứu gần đây của T. Ben-Haim và các cộng sự[7], đề xuất một mô hình học sâu sử dụng hai kiến trúc mạng Unet để phân đoạn GGOs. L-Unet được xây dựng với mạng xương sống VGG là encoder, sử dụng đầu vào là ảnh 2D và đầu ra là mặt nạ phân đoạn phổi của từng ảnh. Tiếp đó, mạng C-Unet có kiến trúc tương tự mạng L-Unet nhận đầu vào là hình ảnh 2D đã phân đoạn phổi và đầu ra là mặt nạ phân đoạn GGOs. Nghiên cứu này cũng đề xuất một hàm loss kết hợp giữa dice loss và cross-entropy loss có sử dụng thêm trọng số cho các nhãn. Tuy nhiên do chỉ sử dụng hình ảnh 2D, giải pháp này vẫn chưa đưa ra được một kết quả tốt và cần cải thiện hơn trong tương lai.

Ngoài những nghiên cứu phân đoạn hình ảnh sử dụng mạng nơ-ron nhân tạo, có một số phương pháp phi mạng được đề xuất và đạt được những kết quả rất ấn tượng. Một phương pháp được đề xuất gần đây đến từ Y. Rao và các cộng sự [8]. Các tác giả đề xuất phương pháp phân đoạn GGOs sử dụng ngưỡng cơ chế chú ý kết hợp với cân bằng đường viền và phân đoạn vùng phổi. Đây là một phương pháp có độ chính xác rất cao trên tập dữ liệu tuy nhiên việc chọn các tham số phù hợp như các ngưỡng trên các hình ảnh khác nhau là khó khăn. Hơn nữa phương pháp này sử dụng cho hình ảnh 2D vì thế cần thêm những giải pháp để kết hợp các lớp liên tiếp để tối ưu các kết quả khi phân đoạn trên hình ảnh 3D.

1.3 Mục tiêu và định hướng giải pháp

Mục tiêu của đồ án tốt nghiệp này là đưa ra một phương pháp phân đoạn hình bóng mờ trong ảnh chụp CT có độ chính xác cao và có thể thích ứng linh hoạt trên các trường hợp bệnh khác nhau đồng thời tạo một sản phẩm đơn giản để áp dụng mô hình này vào thực tế.

Trong đồ án này tôi đã phát triển một quy trình phân đoạn hình ảnh bao gồm 3 giai đoạn. Trong giai đoạn một, tôi huấn luyện một mô hình phân đoạn phổi đơn giản sử dụng kiến trúc U-net 2D theo phương pháp học bán giám sát. Dữ liệu là hình ảnh 2D được lấy từ bộ dữ liệu huấn luyện. Tôi lựa chọn khoảng 100 hình ảnh và gán nhãn thủ công. Sau đó dữ liệu đã gán nhãn thủ công được đưa vào mô hình

huấn luyện, tiếp theo sử dụng mô hình đã đào tạo để phân đoạn những hình ảnh chưa có nhãn, lựa chọn những hình ảnh được gán nhãn giả tốt để tiếp tục đưa vào huấn luyện mô hình.

Ở giai đoạn 2, tôi phát triển hai mô hình học sâu 2D và 3d dựa trên kiến trúc của Framework nnUnet. Đây là một công cụ phân đoạn hình ảnh mạnh mẽ trong lĩnh vực xử lý ảnh y tế, nó cung cấp những quy tắc và siêu tham số để hỗ trợ phân đoạn hình ảnh đặc biệt là hình ảnh 3D. Tuy nhiên như đã nhắc đến ở phần trước, những siêu tham số mặc định và mô hình mặc định của nnUnet không hoàn toàn phù hợp với bộ dữ liệu này. Mô hình 3D được đào tạo để phát hiện những trường hợp GGOs dễ phát hiện và có độ tương phản cao sao với nền. Trong khi đó mô hình 2D được đào tạo để phân đoạn những mẫu phó phát hiện hơn như những vùng tổn thương bé và có độ tương phản thấp. Dữ liệu huấn luyện được tiền xử lý bằng các phương pháp resampling và Intensity Normalization với những siêu tham số phù hợp. Ở giai đoạn này, tập huấn luyện được chia thành 5 phần và thực hiện đánh giá theo phương pháp k-fold cross validation.

Ở giai đoạn một, hàm lỗi được sử dụng là hàm Dice Loss. Ở giai đoạn hai, tôi sử dụng một vài hàm lỗi để đánh giá hiệu quả huấn luyện trên những hàm lỗi này, bao gồm: Combo loss là sự kết hợp giữa Dice loss và Cross-entropy loss, DiceFocal là sự kết hợp giữa Dice Loss và Focal loss đồng thời thử nghiệm một hàm lỗi mới là Boundary loss. 1.2.

1.4 Đóng góp của đồ án

Đồ án này có 3 đóng góp chính như sau:

- Đồ án đề xuất một phương pháp tiền xử lý bao gồm lấy mẫu lại và chuẩn hóa độ sáng có thể thích ứng linh hoạt với nhiều tình trạng bệnh nhân khác nhau đồng thời giúp tối ưu thời gian huấn luyện với lượng dữ liệu huấn luyện và tài nguyên phần cứng hạn chế.
- Đồ án đưa ra một phương pháp học sâu kết hợp giữa hai mô hình mạng 2D và 3D đồng thời so sánh hiệu quả của một số hàm lỗi sử dụng trong trường hợp phân đoạn hình ảnh có nhãn không đồng đều.
- Từ phương pháp và mô hình có được ở trên, tôi phát triển một website đơn giản có chức năng tải hình ảnh CT phổi và thực hiện phân đoạn hình ảnh bóng mờ trên hình ảnh đó.

1.5 Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau.

Chương 2 trình bày về nền tảng lý thuyết và những công nghệ liên quan được

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

sử dụng trong đồ án này. Ngoài ra chương này cũng giới thiệu về ngữ cảnh của bài toán GGOS và một số kết quả nghiên cứu tương tự cho bài toán. Những kiến thức chính được sử dụng trong đồ án này bao gồm Tổng quan về bài toán phân đoạn hình ảnh y tế và mạng nơ-ron tích chập CNN, một kiến trúc mạng nơ ron được sử dụng phổ biến trong việc giải quyết bài toán phân đoạn hình ảnh. Tiếp theo là tổng quan chung về Framework nnUnet, một framework được thiết kế đặc biệt cho các nhiệm vụ phân đoạn hình ảnh y tế, nnUnet được thiết kế dựa trên mô hình Unet và cung cấp một mô đun được thiết kế cho việc xác định cấu hình và huấn luyện mạng cho từng bài toán cụ thể, đồng thời nnUnet cũng cung cấp một tập hợp các phương thức tiền xử lý, tăng cường dữ liệu và hậu xử lý để người dùng có thể sử dụng hoặc tùy chỉnh theo nhu cầu. Phần cuối cùng của chương sẽ giới thiệu khái quát về Framework Flask, một trong những Web framework phổ biến nhất của Python.

Trong Chương 3, tôi trình bày đóng góp chính của đồ án, là đề xuất một phương pháp phân đoạn hình ảnh bóng mờ trên ảnh chụp CT phổi bệnh nhân sau khi nhiễm COVID 19, phương pháp này gồm 2 giai đoạn chính bao gồm phân đoạn phổi và phân đoạn hình ảnh bóng mờ. Chương này đề cập đến các phương pháp tiền xử lý dữ liệu dựa trên cơ sở framework nnUnet nêu ở chương 2 đồng thời phân tích những cải tiến cho phù hợp với bài toán này. Nội dung chương này giới thiệu chi tiết về các kiến trúc Unet kết hợp với cơ chế Attention được sử dụng trong huấn luyện. Ngoài ra, chương này còn giới thiệu chi tiết các thuật toán sử dụng trong tiền xử lý dữ liệu cũng như hậu xử lý kết quả dự đoán như Ensemble Learning, phương pháp đánh trọng số Gaussian và một số hàm lỗi được sử dụng trong đồ án này.

Trong chương 4, tôi đưa ra kết quả đánh giá của phương pháp đã đề xuất trong chương 3 dựa trên những chỉ số đánh giá phổ biến trong bài toán phân đoạn hình ảnh như Dice Coefficient, Jaccard Coefficient, Hausdorff distance và một số chỉ số khác. Trước hết tôi thực hiện so sánh kết quả của các mô hình trên nhiều làm lỗi khác nhau và chọn hàm lỗi có kết quả cao nhất dựa trên Dice Score. Sau đó tôi chứng minh sự hiệu quả của thuật toán với những mô hình phân đoạn hình ảnh y tế phổ biến như Unet, nnUnet, ... Tôi cũng so sánh kết quả của phương pháp đã đề xuất với những phương pháp top đầu trong thử thách COVID-19-20.

Ở chương cuối cùng, tôi tóm tắt lại những vấn đề của bài toán và những vấn đề đã giải quyết được trong đồ án này và đề xuất hướng phát triển trong tương lai.

CHƯƠNG 2. NỀN TẢNG LÝ THUYẾT

Như đã giới thiệu từ chương 1, chương này sẽ giới thiệu về ngữ cảnh của bài toán GGOS và một số kết quả tương tự. Chương này cũng trình bày về nền tảng lý thuyết của bài toán phân đoạn hình ảnh và một mô hình mạnh mẽ cho việc phân đoạn hình ảnh y tế là nnUnet. Ngoài ra chương này sẽ giới thiệu về một số công nghệ được sử dụng trong đồ án này.

2.1 Ngữ cảnh của bài toán

Sau khi đại dịch Covid 19 bùng nổ vào năm 2019, những tác động của nó gây nên với sức khỏe của con người trên toàn thế giới là vô cùng nghiêm trọng. Dịch bệnh đã lây nhiễm trên hàng triệu người và gây tử vong hàng nghìn người mắc phải và những triệu chứng bệnh sau khi bệnh nhân khỏi COVID 19 cũng ảnh hưởng rất nhiều đến chất lượng cuộc sống của các bệnh nhân. Sau khi mắc COVID 19, các bệnh nhân có thể bị mắc một số biến chứng như nhiễm virus ở phổi, những hình ảnh bóng mờ trên ảnh chụp CT phổi bệnh nhân là một trong những biểu hiện phổ biến của tình trạng hậu COVID. Vì thế bài toán phân đoạn hình ảnh bóng mờ trên ảnh chụp CT phổi bệnh nhân sau khi nhiễm COVID 19 là một bài toán quan trọng trong lĩnh vực xử lý hình ảnh y tế. Việc phân đoạn hình ảnh bóng mờ là quan trọng để xác định chính xác các vùng bị tổn thương và xác định mức độ nghiêm trọng của bệnh. Phân đoạn này cung cấp thông tin quan trọng cho việc phỏng đoán, đánh giá và theo dõi tình trạng bệnh của nhân.

Hình ảnh CT cung cấp hình ảnh cắt ngang chi tiết của phổi giúp dễ dàng phát hiện những thay đổi và bất thường cho dù rất nhỏ. Tuy nhiên, việc phân đoạn hình ảnh mờ trên CT đòi hỏi sự chính xác và độ tin cậy cao. Để giải bài toán này, nhiều phương pháp học máy và học sâu đã được đưa ra. Các mô hình học sâu như U-Net, nnU-Net và các biến thể khác đã được áp dụng để tự động phân vùng các vùng mờ trên CT. Các mô hình này sử dụng các kiến trúc chuyên sâu, kết hợp với các kỹ thuật như mạng nơ-ron tích chập (CNN) và các phương pháp tăng cường dữ liệu để đạt được hiệu quả tốt trong quá trình phân đoạn các vùng bị tổn thương.

Vì thế thử thách Phân đoạn hình ảnh bóng trên ảnh chụp CT phổi bệnh nhân sau khi nhiễm COVID 19 (COVID-19-20 Challenge) [9] được tổ chức để đánh giá những phương pháp hiện đại nhằm phân đoạn và xác định vùng tổn thương phổi của bệnh nhân sau khi nhiễm COVID 19.

2.2 Các kết quả nghiên cứu tương tự

Ở thử thách COVID-19-20, rất nhiều thuật toán được đưa ra dựa trên biến thể của mạng Unet được đưa ra để phân đoạn hình ảnh GGOs [1]. S. Hu và các cộng sự [10] đề xuất một thuật toán học bán giám sát để phân đoạn tổn thương ở phổi sau khi nhiễm COVID 19. Một mạng học sâu bán giám sát dựa trên mạng xương sống là mạng nnUnet [2] được sử dụng để thực hiện đào tạo một mạng phân đoạn đối với những dữ liệu đã dán nhãn ban đầu. Nhóm đã thu thập dữ liệu chưa gắn nhãn trong bộ cơ sở dữ liệu TCIA để thực hiện phân đoạn và tạo mặt nạ tổn thương giả trên những hình ảnh này. Tiếp theo, một mô hình phân đoạn thực hiện theo phương pháp học có giám sát được huấn luyện trên bộ dữ liệu được dán nhãn giả ở trên. Phương pháp này giúp tăng tính khái quát hóa của mô hình phân đoạn và kết quả đã chứng minh nó hiệu quả khi có điểm Dice score cao hơn 4,27% so với mạng học sâu có giám sát thông thường. F. Isensee và các cộng sự [10] cũng sử dụng nnUNet để triển khai 5 cấu hình 3D Unet với các độ phân giải và lựa chọn chuẩn hóa khác nhau. Bao gồm: residual U-net độ phân giải thấp sử dụng tăng cường dữ liệu và chuẩn hóa hàng loạt (Batch Normalization - BN); U-net độ phân giải cao sử dụng tăng cường dữ liệu và chuẩn hóa theo từng mẫu (Instance Normalization - IN); residual U-net độ phân giải cao; U-net đơn giản với độ phân giải cao sử dụng tăng cường dữ liệu và IN; U-net đơn giản với độ phân giải cao sử dụng tăng cường dữ liệu và BN. Nhóm sử dụng các siêu tham số mặc định của nnUnet cho từng cấu hình mạng trên và cũng không sử dụng các phương pháp hậu xử lý cho kết quả. Kết quả tốt nhất nhóm đạt được với điểm số Dice Score 75,43% cho cấu hình thứ 2. C. Tang đến từ trường Lynbrook High School, USA [10] phát triển một phương pháp phân đoạn GGOs bằng cách kết hợp 2 trong 4 mô hình cung cấp bởi nnUnet bao gồm: mô hình Unet 2D sử dụng từng lát cắt ảnh CT làm tập huấn luyện, mô hình 3D U-net độ phân giải thấp, , mô hình 3D U-net độ phân giải cao và mô hình 3D U-net phân tầng. Với mỗi mô hình, nhóm sử dụng 3 hàm loss khác nhau để đánh giá kết quả bao gồm: DiceCe kết hợp giữa Dice loss và Cross entropy loss, DiceTopK kết hợp giữa Dice loss và TopK loss, DiceFocal kết hợp giữa Dice loss và Focal loss. Kết quả đạt được trên sự kết hợp tốt nhất của nhóm đạt điểm Dice Score 73,63%. T. Zheng và các cộng sự [10] thực hiện huấn luyện 2 mô hình phân đoạn GGOs trên 2 cấu hình U-net 2d và U-net 3D độ phân giải thấp và kết hợp hai kết quả của chúng bằng cách lấy giá trị trung bình. Mô hình 2D sử dụng đầu vào 512x512 và mô hình 3D sử dụng đầu vào 28x256x256. Tôi không tìm thấy những bài báo hay nghiên cứu chi tiết cho các thuật toán được sử dụng trong thử thách COVID-19-20, tuy nhiên như trong bản tổng hợp kết quả [8], phần lớn các nhóm đều sử dụng framework nnUnet như một mạng xương sống cho phương pháp

của mình và sử dụng các siêu tham số mặc định của nnUnet. Tôi nhận thấy những siêu tham số này không phù hợp với đặc điểm của bộ dữ liệu khi hình dạng, độ sáng, kích thước của GGOs trong từng mẫu là khác nhau tùy tình trạng bệnh và độ tuổi của bệnh nhân. Vì thế cần có một phương pháp chuẩn hóa và lấy mẫu lại linh hoạt hơn cho từng hình ảnh.

Vào năm 2021, D. Muller và các cộng sự đưa ra một giải pháp phân đoạn GGOs dựa trên mô hình học sâu nhằm hỗ trợ cho việc chuẩn đoán sớm COVID 19 [5]. Các tác giả hướng đến một hệ thống phân đoạn hình ảnh dựa trên framework MIScnn [6], một frame work được thiết kế cho tác vụ phân đoạn hình ảnh y tế sử dụng mạng CNN và học sâu. Quy trình này được thiết kế để phát triển một giải pháp phân đoạn GGOs trong ảnh CT phổi bệnh nhân trong điều kiện dữ liệu huấn luyện hạn chế. Dựa trên MIScnn, quy trình huấn luyện mô hình được thực thiện qua những bước cụ thể bao gồm: tiền xử lý dữ liệu, tăng cường dữ liệu, lấy mẫu dữ liệu và huấn luyện mô hình. Bộ dữ liệu được sử dụng ở bài báo này là bộ dữ liệu phân đoạn tổn thương trên ảnh CT phổi của Ma và các cộng sự [11] gồm 20 ảnh chụp CT phổi của các bệnh nhân COVID 19 có tỉ lệ nhiễm trùng phổi từ rất thấp đến cao. Giai đoạn tiền xử lý được thực hiện qua bốn bước bao gồm các bước chuẩn hóa và lấy mẫu lại dữ liệu. Dữ liệu gốc được chuẩn hóa bằng cách cắt các giá trị độ sáng theo độ đo Hounsfield với ngưỡng -1250 đến +250 sau đó chuẩn hóa về thang đo đa mức xám với dải màu 0-255. Sau khi chuẩn hóa về thang đo đa mức xám, dữ liệu tiếp tục được chuẩn hóa theo z-score. Do khoảng cách voxel trong từng hình ảnh khác nhau nên nhóm tác giả thực hiện lấy mẫu lại các hình ảnh về khoảng cách voxel tiêu chuẩn là $1,58 \times 1,58 \times 2,7$. Dữ liệu sau khi chuẩn hóa và lấy mẫu lại tiếp tục được tăng cường bằng các phương pháp tăng cường theo không gian (lật, biến đổi đàm hồi, xoay, co dãn), tăng cường theo màu sắc (độ sáng, độ tương phản) và tăng cường sử dụng nhiễu Gauss. Dữ liệu đưa vào huấn luyện được cắt với Patch size $160 \times 160 \times 80$ và bước nhảy là $80 \times 80 \times 40$. Mô hình huấn luyện là mô hình Unet 3D tiêu chuẩn và sử dụng Tversky Loss kết hợp với Cross entropy Loss để giảm sự mất cân bằng trên nhãn. Tác giả thực hiện huấn luyện nhiều lần với kỹ thuật k-fold cross validation với k từ 2 đến 5. Phương pháp này đạt được kết quả tốt nhất trên tập dữ liệu kiểm tra đạt Dice score 0.661 (theo số liệu từ bài báo). Các tác giả đã đưa ra được một quy trình tổng quát cho tác vụ phân đoạn hình ảnh y tế tuy nhiên với việc chuẩn hóa hàng loạt với bộ dữ liệu này là chưa phù hợp với bài toán này.

Một kiến trúc mạng được D.-P. Fan và các cộng sự đề xuất vào năm 2020 nhằm giải quyết bài toán phân đoạn tổn thương phổi trên ảnh chụp CT ngực là Infnet [4]. Infnet được thiết kế dựa trên Unet [12], một kiến trúc CNN rất phổ biến trong phân đoạn ảnh y tế. Infnet được điều chỉnh để có thể phát hiện nhiều lớp như GGOs,

vùng đông đặc (consolidation) hay vùng bình thường. Infnet gồm 2 phần chính: Single Infnet được sử dụng để dự đoán các vùng tổn thương trên phổi mà chưa phân loại chúng, đầu ra của Single Infnet là dự đoán về những tổn thương và đường viền của chúng; Multiple Infnet được sử dụng để phân loại các tổn thương trong phổi, Multiple Infnet nhận đầu vào là kết quả của Single Infnet và hình ảnh gốc để đưa ra phân loại cụ thể cho các vùng tổn thương. Kiến trúc Infnet sử dụng cơ chế chú ý (Attention Mechanism) giúp mô hình tập trung hơn vào các vùng quan trọng của hình ảnh giúp cải thiện hiệu suất phát hiện tổn thương, ngoài ra ,InfNet sử dụng các lớp tích chập phân tách độc lập (depth separable convolution) để giảm số lượng tham số của mô hình. Daryl L. X. Fung và các cộng sự đã phát triển một phương pháp học tự giám sát dựa trên kiến trúc mạng Infnet và cơ chế inpainting [3]. Mô hình được phát triển trên cơ sở mạng Infnet có giám sát (SInfnet) tương tự như Infnet bao gồm 2 phần gồm Single SInfnet thực hiện dự đoán cạnh biên và 4 mặt nạ phân vùng tổn thương của phổi với các kích thước khác nhau, Multi SInfnet thực hiện phân loại các loại tổn thương với đầu vào là đầu ra của Single SInfnet. InfNet tự giám sát (SSInfNet) tương tự như SInfnet tuy nhiên cơ chế inpainting sẽ được thực hiện trong phần Single SSInfnet. SSInfNet học các đặc trưng của ảnh qua quá trình phục hồi hình ảnh (inpainting). Đầu vào của quá trình inpainting là hình ảnh phổi ban đầu và một mặt nạ được tạo ra để che một phần hình ảnh, mục tiêu của SSInfnet là học cách tái tạo hình ảnh bị che khuất bởi mặt nạ dựa trên thông tin từ các hình ảnh còn lại. Để hỗ trợ quá trình inpainting, một mạng huấn luyện bổ trợ (coach network) được sử dụng để tạo ra mặt nạ che phủ hình ảnh mặt nạ sẽ tăng dần độ phức tạp trong quá trình huấn luyện. Dù có nhiều điểm mạnh như sử dụng cơ chế chú ý giúp mô hình tăng sự tập trung vào các vùng quan trọng hay cơ chế inpainting tuy nhiên độ chính xác của mô hình chỉ đạt tối đa 0.63 điểm Dice score trên tập kiểm thử (số liệu từ bài báo) cho thấy phương pháp vẫn cần cải thiện trong tương lai như áp dụng thêm các phương pháp tiền xử lý dữ liệu và tăng cường dữ liệu phù hợp.

T. Ben-Haim và các cộng sự đề xuất một phương pháp sử dụng hai mạng U-net phân cấp để phân đoạn GGOs [7]. Mô hình học sâu này gồm 2 mạng U-net nối tiếp nhau, mạng L-net phía bên trái thực hiện phân đoạn phổi, C-net ở phía còn lại thực hiện phân đoạn GGOs và đông đặc. Hai mạng L-net và C-net có encoder được học chuyển tiếp từ mạng VGG16 [13] đã được pre-train với bộ dữ liệu Imagenet [14]. Để tăng cường dữ liệu cho việc huấn luyện, các patch được cắt một cách ngẫu nhiên từ tập dữ liệu. Hai loss được sử dụng trong phương pháp này là Dice loss và Cross entropy loss. L-net sử dụng hàm binary cross-entropy loss sử dụng trọng số như nhau ở background và foreground. C-net sử dụng hàm loss kết hợp

giữa weighted cross entropy và weighted Dice loss với trọng số cho mỗi nhãn khác nhau. Các tác giả sử dụng phương pháp Ensemble Learning để tăng độ chính xác của kết quả. K mô hình được huấn luyện với trọng số ở decoder được khởi tạo khác nhau và các phương thức tăng cường dữ liệu khác nhau. Kết quả tốt nhất của phương pháp này có được khi kết hợp $k = 6$ mô hình và sử dụng phương pháp lấy giá trị trung bình. Tuy nhiên trong phương pháp này vẫn chưa đưa ra một giải pháp cụ thể để giải quyết vấn đề khó khăn đã nêu trên của dữ liệu.

I. Ahmed và các cộng sự đề xuất một giải pháp phân đoạn GGOs có độ chính xác cao tập trung vào những vùng nhỏ và khó dự đoán bằng cách kết hợp mô hình U-net với cơ chế chú ý và hàm lỗi boundary loss [15]. Dữ liệu ban đầu được tăng cường bằng 3 phương pháp không gian, màu sắc và nhiễu theo quy trình MIScnn. Do mô hình U-net cơ bản trích xuất tính năng không có nhiều giá trị nên tác giả đã kết hợp cơ chế chú ý vào mô hình mạng U-net để giúp mô hình trích xuất tính năng tốt hơn. Tuy nhiên việc phát hiện những vùng tổn thương nhỏ trong hình ảnh vẫn rất khó khăn vì vậy tác giả đề xuất sử dụng hàm lỗi boundary loss. Boundary loss được giới thiệu lần đầu vào năm 2018 bởi H. Kervadec và các cộng sự[16], được thiết kế để giải quyết những bài toán có độ mất cân bằng dữ liệu lớn và khó xác định biên của vùng phân đoạn. Mô hình được huấn luyện bằng phương pháp k-fold cross validation, kết quả thu được trên tập test đạt 0,763 cao hơn nhiều so với những mô hình thông dụng như U-net, Unet++(Số liệu từ bài báo).

Ngoài những nghiên cứu sử dụng học sâu, Y. Rao và các cộng sự đề xuất một phương pháp ngưỡng cơ chế chú ý (ACL) [8]. Phương pháp này được đưa ra để giải quyết bốn vấn đề chính trong bài toán GGOs: đặc điểm và hình dạng của các tổn thương GGOs là khác nhau và khó tìm ra một đặc điểm cố định để thực hiện phân đoạn; độ tương phản của vùng GGOs và các mô xung quanh thấp nên khó xác định đúng cạnh; việc sử dụng một ngưỡng cố định để phân đoạn là rất khó khăn; việc chọn ngưỡng giá trị và lựa chọn một cách phù hợp là một bài toán khó. Dựa vào những thách thức nêu trên nhóm tác giả đã đề xuất phương pháp ngưỡng cơ chế chú ý bao gồm 3 chức năng là tìm ngưỡng cơ chế thích ứng với từng trường hợp khác nhau, thứ 2 là cân bằng các đường viền và cuối cùng là phân đoạn phổi. Thuật toán tìm ngưỡng được tóm tắt như sau, hình ảnh lát cắt ảnh CT phổi bệnh nhân được cắt vừa với 2 bên phổi sau đó được chia thành 4 phần bằng nhau. Ở 4 phần nhỏ này, thuật toán lấy ngưỡng OSTU được thực hiện xác định ngưỡng T_{i1} với i là phần ảnh tương ứng và 1 là ngưỡng thứ nhất. Ngưỡng thứ 2 được lấy thấp hơn so với ngưỡng thứ nhất bằng cách tiếp tục sử dụng phương pháp OSTU cho những pixel dưới ngưỡng thứ nhất và được đặt là T_{i2} nhằm mục đích phân tách những phần đáng kể hơn so với ngưỡng T_{i1} . Ngưỡng thứ 3 T_{i3} được tính tương tự bằng OSTU trong

tập các pixel trên ngưỡng T_{i2} . Các ngưỡng T_{i1} và T_{i2} được hợp nhất và lựa chọn giá trị Global T_1 và T_2 bằng cách lấy giá trị lớn nhất. Các pixel có giá trị nhỏ hơn Global T_2 được xác định là mặt nạ phổi của hình ảnh. Giá trị T_2 và T_{i3} chia hình ảnh thành 3 miền giá trị khác nhau từ S_1 đến S_3 trong đó S_2 và S_3 được xác định là vùng tổn thương và cần thực hiện cân bằng đường viền trên các vùng này để đưa ra cách cạnh phân tách mô phổi và GGOs chính xác. Tác giả đưa ra một công thức tính giá trị độ tương phản của hình ảnh để đưa ra những thuật toán khác nhau cho từng độ phân giải khác nhau. Các phương pháp được sử dụng bao gồm phép mở rộng, ăn mòn và làm mờ. Phương pháp này cho thấy được độ chính xác rất cao so với những mô hình sử dụng mạng học sâu như Gate U-net, Infnet hay JCS.

2.3 Bài toán phân đoạn hình ảnh y tế

Bài toán phân đoạn ảnh (image segmentation) là một trong những tác vụ quan trọng trong lĩnh vực xử lý ảnh. Có nhiều ứng dụng quan trọng của bài toán phân đoạn xử lý ảnh như nhận dạng đối tượng, phân đoạn đối tượng chuyển động, xử lý ảnh y tế ... Bài toán phân đoạn ảnh y tế là một bài toán đặc biệt quan trọng trong lĩnh vực y học và hình ảnh y tế. Trong bài toán này, chúng ta cần phân chia hình ảnh y tế (ví dụ: hình chụp MRI, CTscan) thành các vùng nhỏ hơn, biểu thị các cấu trúc y tế khác nhau trong cơ thể. Trong các ứng dụng y tế, việc phân đoạn ảnh 3D là cực kỳ quan trọng, một số ứng dụng tiêu biểu của phân đoạn ảnh y tế 3D bao gồm: phân đoạn óc và não, đây là một bài toán phân đoạn ảnh y tế quan trọng để xác định các cấu trúc và vùng khác nhau của não, giúp phát hiện các bất thường và bệnh lý; phân đoạn các cơ quan nội tạng; phân đoạn các khối u và u nang ...

Ảnh chụp CT (Computed Tomography) là một phương pháp hình ảnh y học không xâm lấn được sử dụng để tạo hình ảnh cắt lớp của cơ thể bằng cách sử dụng tia X. Ảnh CT được sử dụng trong nhiều lĩnh vực y học như chuẩn đoán ung thư, chẩn thương, nhi khoa ... Định dạng file ảnh được sử dụng trong bài toán này là định dạng NIfTI, một định dạng file hình ảnh được sử dụng rộng rãi trong cộng đồng nghiên cứu hình ảnh y tế và thị giác máy tính. Mỗi file hình ảnh chứa thông tin chi tiết về kích thước hình ảnh, không gian voxel và mô tả cấu trúc hình ảnh. Trong python định dạng này có thể dễ dàng đọc bằng thư viện nibabel.

2.4 Mạng nơ-ron tích chập

Mạng nơ-ron tích chập (Convolution Neural Network - CNN) là một loại mạng nơ-ron được thiết kế đặc biệt cho xử lý dữ liệu hình ảnh và video. CNN được lấy cảm hứng từ cách xử lý thông tin thị giác của não người, đây là một công cụ mạnh mẽ trong các nhiệm vụ xử lý ảnh phức tạp bao gồm phân đoạn hình ảnh. Kiến trúc của mạng CNN thường bao gồm một hoặc nhiều lớp tích chập (convolutional

layer), lớp pooling (pooling layer), và các lớp kết nối đầy đủ (fully connected layer) cuối cùng. Trong bài toán phân đoạn hình ảnh mạng CNN đã chứng minh được khả năng xuất sắc trong việc xử lý hình ảnh phức tạp và phân chia hình ảnh thành các vùng nhỏ hơn. Trong y học, mạng CNN được ứng dụng để phân đoạn trong nhiều tác vụ khác nhau như phân đoạn tế bào và mô mềm hay phân đoạn hình ảnh y tế 3D.

Một trong những kiến trúc mạng CNN phổ biến nhất trong tác vụ phân đoạn ảnh y tế chính là mạng U-net. Nó được giới thiệu bởi Olaf Ronneberger, Philipp Fischer và Thomas Brox vào năm 2015 [12]. Mạng U-Net có kiến trúc encoder-decoder, trong đó thông tin hình ảnh được giảm chiều qua các lớp tích chập (encoder) và sau đó được khôi phục lại kích thước ban đầu thông qua các lớp transposed convolution (decoder). Mạng U-Net có khả năng xử lý hiệu quả các hình ảnh y tế 3D và phân đoạn các cấu trúc phức tạp trong dữ liệu y tế. Với kiến trúc U-Net, thông tin chi tiết có thể được giữ lại trong quá trình phân đoạn và giúp cải thiện độ chính xác của mô hình.

2.5 Các kiến thức xử lý ảnh

2.5.1 Tìm thành phần liên thông

Thuật toán tìm thành phần liên thông (Connected Component Labeling, CCL) là một thuật toán được xuyên suốt trong đồ án này. Vùng liên thông là một nhóm các pixel có cùng giá trị và được kết nối với nhau qua các điểm ảnh kề cạnh hoặc đỉnh. Trong quá trình CCL, mỗi vùng liên thông được gán một nhãn định danh duy nhất.

Đối với ảnh 2D, có loại CCL phổ biến là 4-Connexity CCL (duyệt qua những pixel ở trên dưới, trái, phải) và 8-Connexity CCL (duyệt theo cả 8 hướng bao gồm đường chéo). Để đánh nhãn có các thành phần liên thông với trường hợp 4-Connexity CCL ta cần duyệt qua từng pixel trên ảnh từ trái qua phải và trên xuống dưới, tại mỗi pixel nếu là foreground, kiểm tra xem pixel này có liên kết với pixel phía bên trái và phía trên không, nếu có thì đánh số theo pixel liên kết với nó ưu tiên phía trên, nếu không ta đánh số mới cho pixel này. Sau khi duyệt hết các pixel, thực hiện vòng lặp thứ hai với chiều ngược lại vòng lặp đầu tiên để chỉnh sửa lại các nhãn với hàng xóm ở phía dưới và bên phải. Thực hiện lại quy trình này vài lần đến khi không có sự thay đổi về nhãn nữa thì kết thúc. Đối với trường hợp 8-Connexity hay trường hợp tìm thành phần liên thông trên ảnh 3D, quy trình cũng tương tự và có một chút thay đổi đối với những hàng xóm được duyệt.

2.5.2 Các toán tử hình thái

Trong xử lý ảnh và xử lý tín hiệu, các toán tử hình thái là các phép toán giữa một cấu trúc quét (scanning structuring element) và ảnh nhị phân được sử dụng để làm sạch kết quả từ việc phân ngưỡng ảnh (thresholding) và thực hiện các phép toán tiền xử lý để cải thiện chất lượng và phân đoạn đối tượng trong ảnh. Các phép toán này dựa vào việc di chuyển cấu trúc quét trên ảnh và thực hiện các phép tác động lên vùng được quét. Một số phép toán phổ biến trong xử lý ảnh bao gồm: Dilation(giãn nhở) sử dụng để mở rộng các vùng trắng trong ảnh nhị phân, nối liền các đối tượng; Erosion (co) có tác dụng thu nhỏ các vùng trắng, tách rời các vật thể hoặc loại bỏ các biên; Opening là phép toán kết hợp của Erosion và sau đó là Dilation có tác dụng xóa các liên kết nhỏ hoặc các nhiễu nhỏ trong ảnh nhị phân; Closing ngược lại với Opening, phép toán này sẽ lấp đầy các lỗ trống và kết nối các đối tượng không đầy đủ trong ảnh.

2.5.3 Thuật toán lấy ngưỡng OSTU

Thuật toán OSTU là một trong những thuật toán phân ngưỡng phổ biến nhất được phát triển bởi Nobuyuki Otsu vào năm 1979. Cơ chế phân ngưỡng của thuật toán này dựa trên việc tìm ngưỡng tối ưu sao cho giá trị intra-class variance nhỏ nhất. Intra-class variance là giá trị đo lường sự phân tán của dữ liệu trong một lớp, khi giá trị này càng nhỏ, các dữ liệu trong lớp càng gần nhau. Thuật toán OSTU thực hiện duyệt qua các giá trị mức xám trong biểu đồ histogram của ảnh, tại mỗi mức xám, tính toán giá trị intra-class variance của hình ảnh khi sử dụng ngưỡng là mức xám này. Dựa vào các giá trị intra-class variance tìm ngưỡng tối ưu cho hình ảnh.

2.6 nnUnet

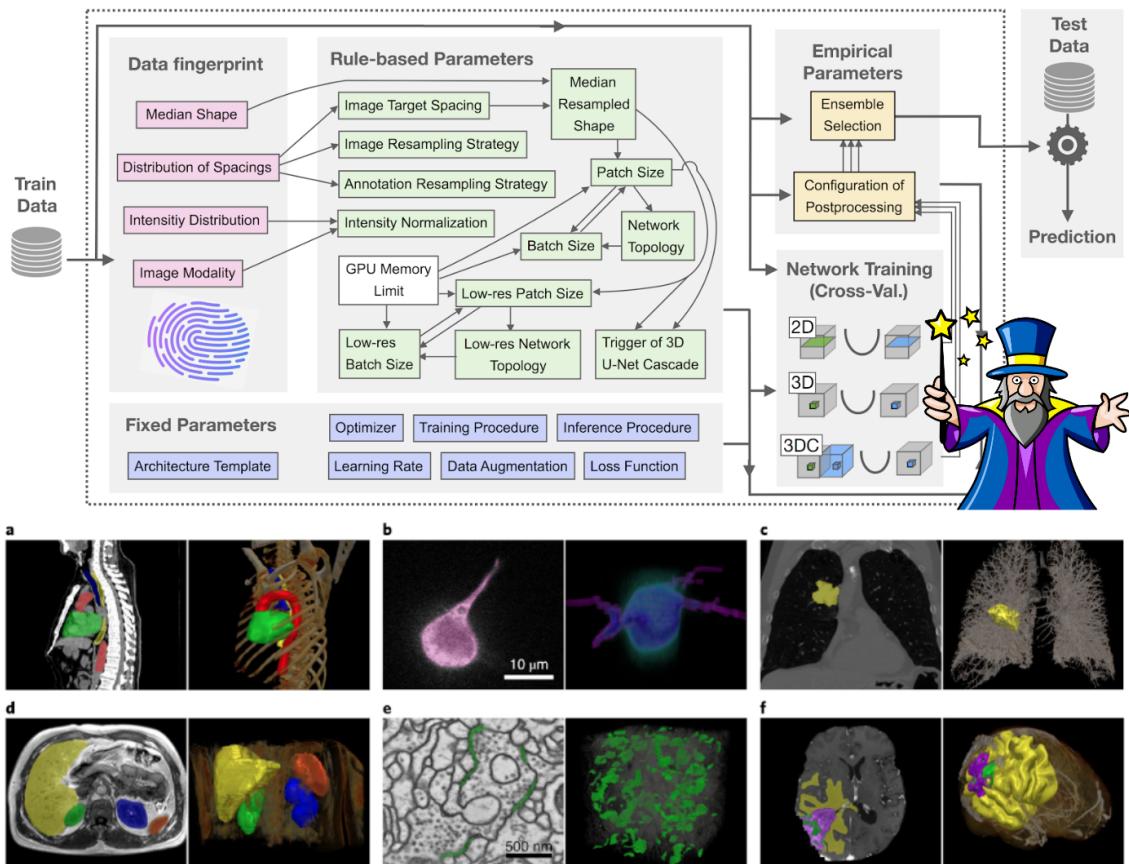
nnU-net là một framework học sâu được thiết kế đặc biệt cho các nhiệm vụ phân đoạn ảnh y tế được mở rộng từ kiến trúc U-net. nnUnet được giới thiệu vào năm 2020 bởi Fabian Isensee và các cộng sự [2] Điểm mạnh của nnU-net nằm ở sự linh hoạt và tích hợp sẵn của nó, cho phép người dùng dễ dàng triển khai và thử nghiệm các mô hình phân đoạn tùy chỉnh cho các ứng dụng y tế. nnUnet cung cấp cho người dùng một loạt tính năng và tiện ích để dễ dàng huấn luyện một mạng nơ-ron học sâu cho tác vụ phân đoạn hình ảnh. Cụ thể những tính năng này được mô tả trong hình 2.1.

Dataset fingerprint trong nnU-Net là quá trình tính toán và lưu trữ các siêu tham số liên quan đến việc chuẩn bị dữ liệu. Những siêu tham số này bao gồm các thông số như kích thước ảnh, kiểu dữ liệu, số lượng lớp, phạm vi giá trị của các voxel, thông số phép biến đổi dữ liệu Khi huấn luyện mô hình trên một tập dữ liệu

mới, nnU-Net sẽ tính toán và lưu trữ dataset fingerprint cho tập dữ liệu đó. Trong các lần huấn luyện tiếp theo, nnU-Net sẽ kiểm tra dataset fingerprint của tập dữ liệu đã được lưu trữ trước đó và so sánh nó với dataset fingerprint của tập dữ liệu hiện tại. Nếu dataset fingerprint không thay đổi, nnUnet sẽ tái sử dụng các siêu tham số cho việc chuẩn bị dữ liệu trước đó và điều này sẽ giúp tăng tốc quá trình chuẩn bị dữ liệu.

Rule-based Parameters là một tập hợp các thông số được tính toán tự động từ dữ liệu và kiến trúc mô hình trong quá trình chuẩn bị dữ liệu và huấn luyện. Các thông số này được tính toán và sử dụng một cách tự động để đảm bảo rằng mô hình được huấn luyện và áp dụng một cách hiệu quả và chính xác trên dữ liệu. Cụ thể quá trình này bao gồm những quy tắc để chuẩn hóa và resampling dữ liệu đồng thời xác định kiến trúc của các mạng học sâu như 2D, 3D, 3D phân tầng. Các siêu tham số của từng kiến trúc mạng (batch size, số lượng epoch, patch size,) cũng sẽ được tự động tính toán để phù hợp với điều kiện GPU của người dùng.

Fixed parameter là những tham số cố định của nnUnet bao gồm các thuật toán tối ưu (SGD và Nesterov momentum), hàm lỗi là hàm Combo loss, quy trình tăng cường dữ liệu: xoay, scale, thêm nhiễu Gaussian, thay đổi độ tương phản, độ sáng, ... Quy trình dự đoán trong nnUnet được áp dụng phương pháp cửa sổ trượt và quy trình thêm trọng số Gaussian để giảm hiện tượng nỗi ảnh không đồng đều. Empirical Parameters là tập hợp các phương thức hợp nhất và hậu xử lý kết quả của các mô hình với nhau. Các mô hình được hợp nhất dựa trên Dice score của các mô hình nhỏ khi sử dụng thuật toán k-fold cross validation. Các mô hình được kết hợp lại bằng cách lấy trung bình của mặt nạ xác suất. nnUnet sử dụng phương pháp thành phần liên thông để hậu xử lý kết quả.



Hình 2.1: Kiến trúc nnUnet *Nguồn: nnUnet (<https://github.com/MIC-DKFZ/nnUNet>)

2.7 Tensorflow

Các mô hình và mã nguồn huấn luyện trong ĐATN này được xây dựng với thư viện Tensorflow. Tensorflow được ra mắt vào năm 2015 là một thư viện mã nguồn mở được phát triển bởi nhóm Google Brain nhằm mục đích hỗ trợ việc học tập và nghiên cứu học máy một cách nhanh chóng và hiệu quả. Tensorflow cung cấp một khung học sâu bao gồm các chức năng tạo, huấn luyện, triển khai khác mạng nơ-ron trong nhiều tác vụ khác nhau như phân loại, phân đoạn hình ảnh, xử lý ngôn ngữ tự nhiên, nhận dạng văn bản Tensorflow tích hợp Keras để hỗ trợ cho việc xây dựng và huấn luyện các mô hình mạng nơ-ron dễ dàng, đơn giản hơn. Keras là một thư viện mã nguồn mở cho học sâu được viết bằng ngôn ngữ Python, được ra mắt vào năm 2015 bởi François Chollet. Keras thiết kế với triết lý dễ sử dụng, linh hoạt và dễ mở rộng, giúp cho việc tạo ra các mô hình deep learning trở nên đơn giản và nhanh chóng. Từ Tensorflow 2.0, Keras đã được tích hợp chặt chẽ vào TensorFlow làm thành API chính thức cho việc xác định và huấn luyện các mô hình học sâu.

Tensorflow dataset(TFDS) là một phần của hệ sinh thái Tensorflow, đây là một thư viện mã nguồn mở cung cấp các bộ dữ liệu chuẩn và sẵn sàng sử dụng trong các tác vụ học sâu khác nhau. TFDS cũng cung cấp một API dễ sử dụng và linh hoạt trong việc chuẩn bị một bộ dữ liệu mới. Các hàm và tiện ích trong TFDS giúp

người dùng có thể dễ dàng lưu trữ và định dạng dữ liệu bằng nhiều kiểu dữ liệu khác nhau như tensor, ảnh, audio, text, scalar Ngoài ra, TFDS hỗ trợ việc tự động xử lý và tiền xử lý dữ liệu như trộn, định dạng lại dữ liệu, chia lô và các tác vụ xử lý ảnh khác giúp việc tải dữ liệu huấn luyện nhanh chóng và sẵn sàng giúp việc huấn luyện nhanh hơn. Ta có thể dễ dàng khởi tạo một bộ dữ liệu mới bằng câu lệnh "`tfds new <dataset_name>`" và dễ dàng định dạng dữ liệu hay phân chia bộ dữ liệu dễ dàng với API được cung cấp bởi TFDS.

2.8 Flask

Flask là một framework web Python nhẹ nhàng và linh hoạt, được tạo ra để giúp xây dựng ứng dụng web nhanh chóng và dễ dàng. Với thiết kế nhỏ gọn và hướng tối tính năng tối giản, Flask cho phép tùy chỉnh và linh hoạt trong việc lựa chọn các công nghệ và tích hợp dữ liệu. Nó được ứng dụng rộng rãi trong việc xây dựng các trang web tĩnh, ứng dụng RESTful API, blog, diễn đàn, và các ứng dụng web dựa trên cơ sở dữ liệu. Flask là lựa chọn phổ biến cho các nhà phát triển muốn xây dựng ứng dụng web đơn giản và hiệu quả.

Trong đồ án này, tôi sử dụng Flask và python tạo một Website đơn giản để tải ảnh CT phổi lên và thực hiện phân đoạn GGOs và hiển thị hình ảnh 3D đã được phân đoạn.

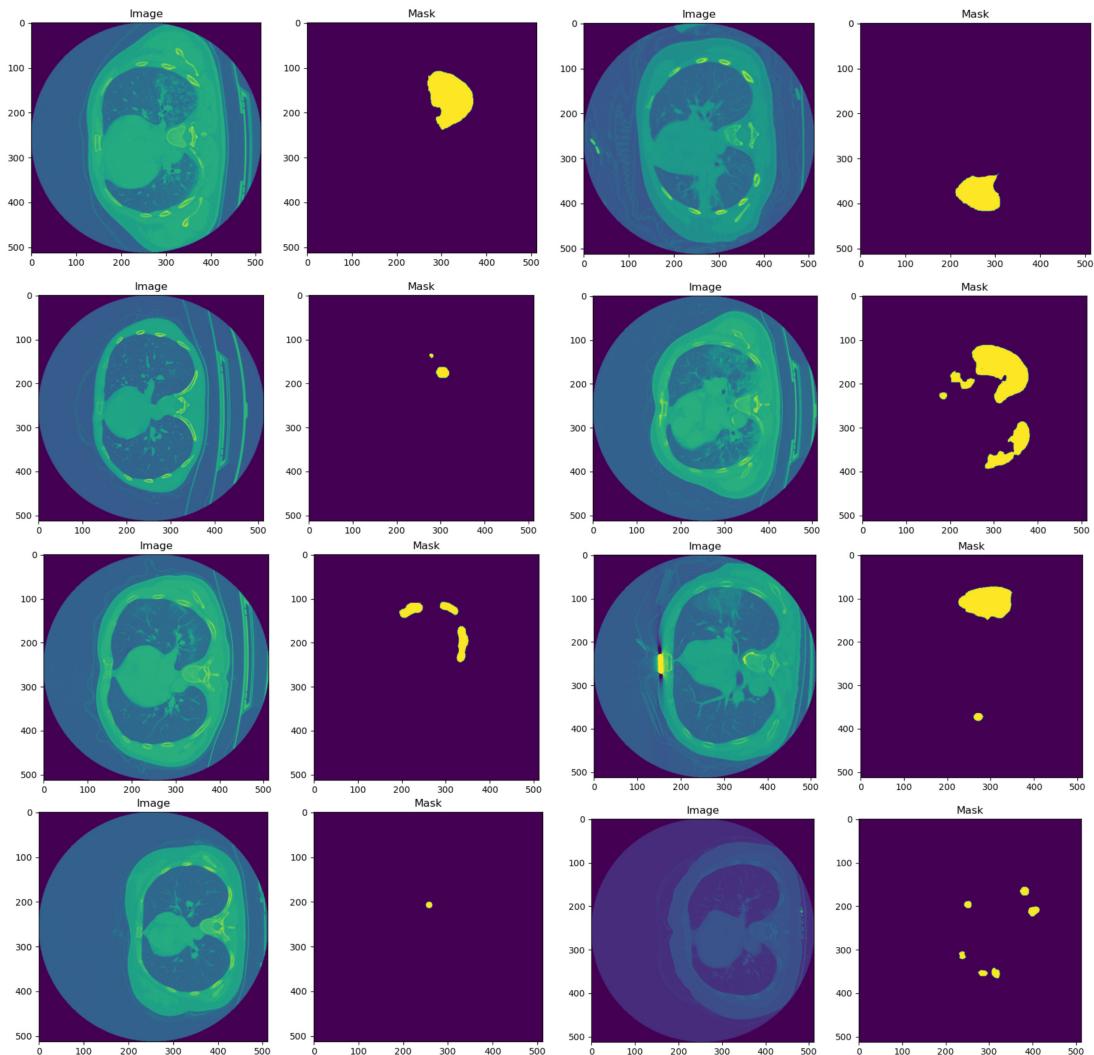
CHƯƠNG 3. PHƯƠNG PHÁP ĐỀ XUẤT

Chương 2 đã giới thiệu về ngữ cảnh bài toán GGOs và những nghiên cứu liên quan. Dựa vào những kiến thức cơ sở đã đề cập ở chương 2, chương này sẽ giới thiệu và phân tích giải pháp của tôi trong việc giải quyết bài toán GGOs. Chương này được chia thành 2 phần chính tương ứng với 2 giai đoạn chính đã được nêu trong chương 1, đồng thời chương này cũng giới thiệu về bộ dữ liệu sử dụng và những hàm lỗi được thử nghiệm trong các mô hình của phương pháp này.

3.1 Phương pháp chung

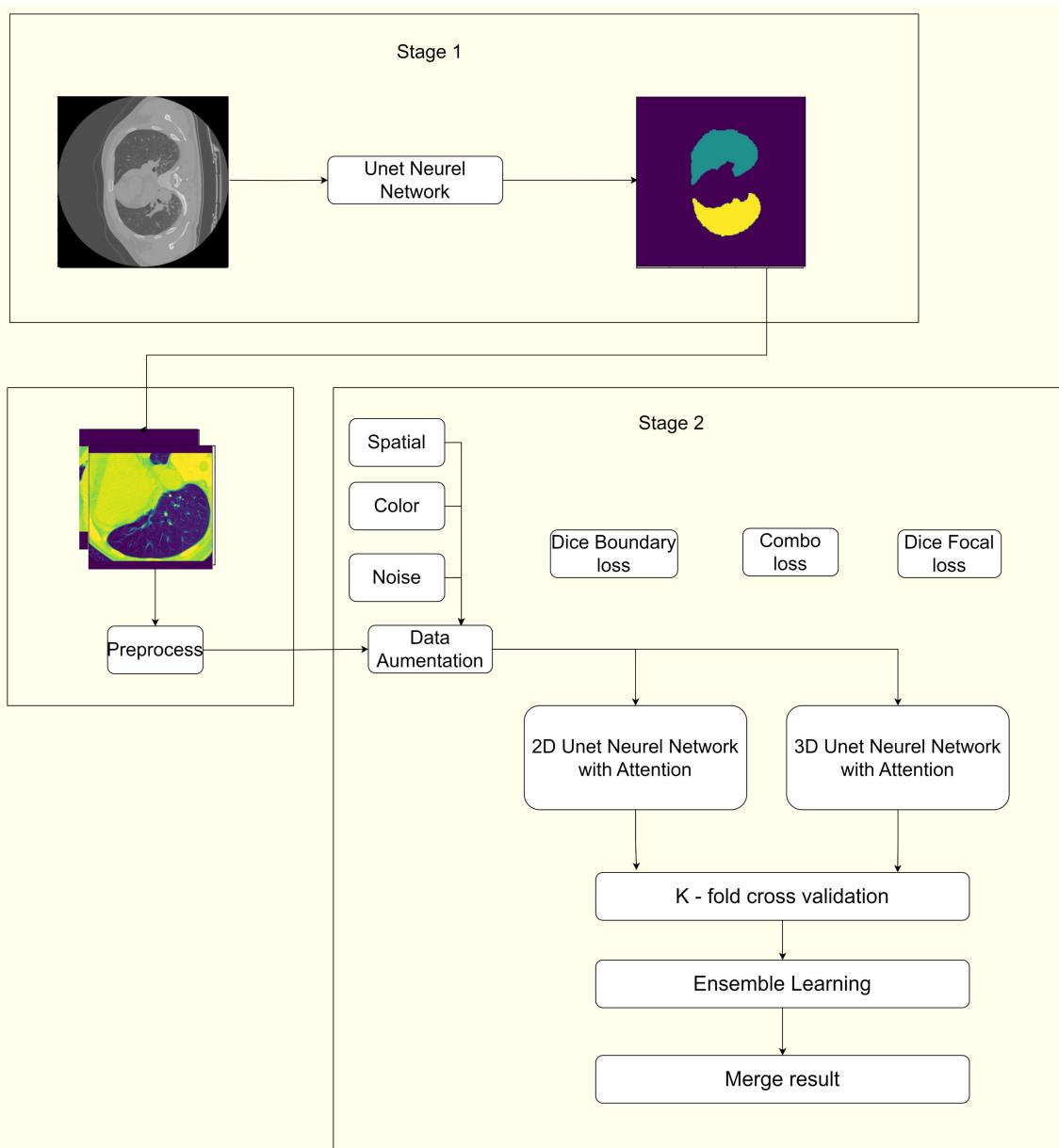
Dữ liệu được sử dụng trong đồ án này được cung cấp bởi thử thách COVID-19-20. Bộ dữ liệu bao gồm 199 ảnh huấn luyện và 50 ảnh thử nghiệm. Dữ liệu được thu thập từ 2 nguồn là CT Images in COVID-19 và COVID-19-AR [10], hai bộ dữ liệu này có sẵn trên The Cancer Imaging Archive (TCIA) [17]. Dữ liệu này được thu thập từ những bệnh nhân đã có kết quả dương tính với COVID-19 bằng phương pháp RP-PCR. Nhãn của tập dữ liệu được gán bằng một mô hình phân đoạn phát triển bởi NVIDIA cùng với các chuyên gia bệnh viện Nhi Quốc gia Hoa Kỳ và NIH, mặt nạ GGOs sau đó được điều chỉnh thủ công bởi các bác sĩ X-quang để đưa ra mặt nạ chính xác nhất. Dữ liệu được lưu dưới định dạng NifTi.

Phương pháp tôi đề xuất trong đồ án này gồm 2 giai đoạn chính: phân đoạn phổi và phân đoạn GGOs. Mục đích của việc phân đoạn phổi nhằm xác định một 3D bounding box bao quanh vùng phổi để giảm kích thước đầu vào cho giai đoạn tiếp theo đồng thời việc phân đoạn phổi đóng vai trò quan trọng trong việc tiền xử lý dữ liệu và hậu xử lý kết quả trong giai đoạn 2. Giai đoạn 2 phân đoạn GGOs được thực hiện bởi 2 mô hình U-net riêng biệt kết hợp với cơ chế chú ý. Mô hình 3D U-net được thiết kế để phân đoạn chính xác những vùng GGOs có kích thước lớn hơn và độ tương phản lớn hơn tận dụng mối liên kết không gian của những voxel. Mô hình 2D U-net tập trung vào việc phát hiện những mẫu khó phân đoạn hơn như những mẫu có độ tương phản thấp so với nền hoặc những mẫu nhỏ nằm rải rác ở trong một lớp của hình ảnh 3D. Các mô hình này được huấn luyện bằng k-fold cross validation với $k=5$ và thực hiện ensemble learning để kết hợp. Kết quả của 2 mô hình 2D và 3D được hợp nhất lại dựa trên mặt nạ phổi ở giai đoạn 1. Hình 3.2 mô tả tổng quan của phương pháp trên.



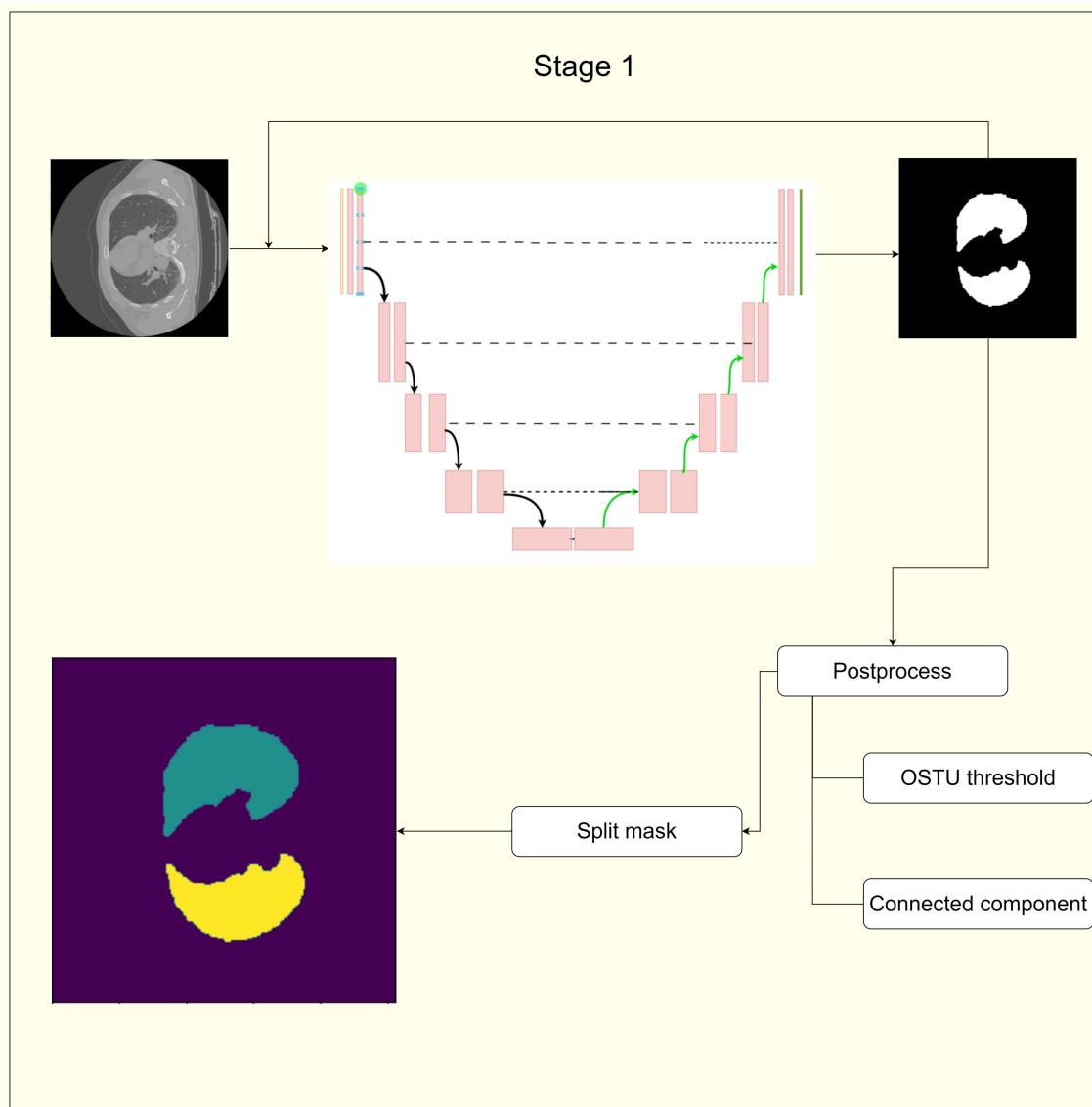
Hình 3.1: Trực quan hóa bộ dữ liệu: Nhãn của các hình ảnh có độ đa dạng rất lớn. Độ sáng của vùng tổn thương trong mỗi hình ảnh không đồng đều, kích thước các vùng GGOs trong toàn bộ tập dữ liệu khác nhau tùy vào tình trạng bệnh.

CHƯƠNG 3. PHƯƠNG PHÁP ĐỀ XUẤT



Hình 3.2: Tổng quan phương pháp: Tại giai đoạn 1, hình ảnh 2D được đưa vào một mô hình học bán giám sát với kiến trúc mô hình Unet để phân đoạn phổi. Dựa vào mặt nạ phổi đã phân đoạn, hình ảnh được tiền xử lý trước khi đưa vào huấn luyện bằng các phương pháp chuẩn hóa hình ảnh. Giai đoạn hai thực hiện phân đoạn GGOs với hai mô hình 2D và 3D Unet kết hợp với cơ chế chú ý và các hàm lỗi kết hợp.

3.2 Giai đoạn 1: Phân đoạn phổi



Hình 3.3: Giai đoạn 1: Phân đoạn phổi sử dụng học bán giám sát. Một phần dữ liệu được gán nhãn thủ công sau đó đưa qua một mô hình Unet đơn giản. Thực hiện phân đoạn trên các hình ảnh chưa có nhãn tạo thành các nhãn giả, sau đó chọn nhãn giả có kết quả tốt để tiếp tục huấn luyện mô hình nhiều lần. Kết quả cuối cùng được hậu xử lý để tạo ra mặt nạ gồm 2 phần phổi với hai nhãn khác nhau

Vì bộ dữ liệu không có nhãn cho vùng phổi vì thế giai đoạn này tôi thực hiện quá trình phân đoạn phổi bằng phương pháp học bán giám sát. Dữ liệu ban đầu được gán nhãn bằng tay trên 100 hình ảnh 2D được lấy ngẫu nhiên trong tất cả các mẫu của bộ dữ liệu huấn luyện. Các hình ảnh này sau đó được cắt (clip) trong khoảng từ -1200 đến 150 theo đơn vị Hounsfield (Hounsfield Unit - HU) và chuẩn hóa về ảnh đa mức xám với giá trị từ 0-255. 100 hình ảnh ban đầu này được đưa vào huấn luyện bằng một mô hình 2D U-net đơn giản sau khoảng 15 epoch sau đó thực hiện dự đoán trên những hình ảnh chưa được gán nhãn. Tôi lựa chọn những hình ảnh có

kết quả tốt trên những ảnh đã được gán nhãn giả bởi mô hình sau đó hợp nhất với những hình ảnh đã gán nhãn bằng tay và tiếp tục đưa vào huấn luyện.

Quá trình này được thực hiện lại nhiều lần đến khi đạt được hiệu quả phân đoạn phổi như mong muốn. Sau khi có được mô hình huấn luyện, tôi thực hiện dự đoán trên tất cả các hình ảnh có trong tập huấn luyện và tập thử nghiệm. Để loại bỏ các nhiễu có trong kết quả cuối cùng tôi thực hiện hậu xử lý kết quả bằng cách sử dụng kỹ thuật tìm thành phần liên thông trong hình ảnh 3D. Hình ảnh phân đoạn phổi của mô hình thường xuất hiện thêm nhãn của những bộ phận không phải phổi, vì thế tôi thực hiện tìm các thành phần liên thông trong các hình ảnh và giữ lại 1 đến 2 thành phần liên thông có khối lượng lớn nhất dựa vào kích thước 3D bounding box thu được, thuật toán trên được thực hiện theo mã giả 1 (thuật toán thực hiện trên từng mẫu)

Algorithm 1 Lung mask Post-Process Function

```

1: function LUNG_MASK_POST_PROCESS(mask)
2:   labels, num_labels  $\leftarrow$  ndimage.label(mask)
3:   sizes  $\leftarrow$  ndimage.sum(mask, labels, range(1, num_labels + 1))
4:   largest_components  $\leftarrow$  argsort_descending(sizes)[: 2]
5:   components_mask  $\leftarrow$  zeros_like(mask)
6:   for label in largest_components do
7:     components_mask[labels == (label + 1)]  $\leftarrow$  1
8:     box_indices  $\leftarrow$  find_indices(components_mask)
9:     y_min, x_min, z_min  $\leftarrow$  min_values(box_indices)
10:    y_max, x_max, z_max  $\leftarrow$  max_values(box_indices)
11:    if y_min < 150 and y_max > 350 then
12:      break
13:    end if
14:   end for
15:   return components_mask
16: end function

```

Đầu vào của hàm này là mặt nạ nhị phân phổi đã phân đoạn bằng mô hình trên. Tôi sử dụng hàm `ndimage.label()` được cung cấp bởi thư viện `scipy` để thực hiện tìm thành phần liên thông trên mặt nạ này. Để chỉ giữ lại mặt nạ phổi, các thành phần liên thông được sắp xếp theo thứ tự giảm dần về khối lượng. Duyệt qua các thành phần liên thông với khối lượng từ lớn đến bé, nếu bounding box của tổng các thành phần liên thông đã duyệt qua có tọa độ $y_{min} < 150$ và $y_{max} > 350$ (các hệ số này tùy chỉnh theo kích thước của hình ảnh) thì dừng vòng lặp và trả về mặt nạ phổi.

Sau khi có mặt nạ của phổi, tôi thực hiện chia mặt nạ này thành 2 phần phổi trái

và phải để chia nhỏ dữ liệu cho giai đoạn sau (Mã giả 2, 3). Tôi tiếp tục sử dụng thuật toán tìm thành phần liên thông kết hợp với phân ngưỡng OSTU để thực hiện quá trình này. Trên mỗi lát cắt của mặt nạ phân đoạn phổi, thực hiện tìm các thành phần liên thông sau đó tính tổng diện tích của vùng nằm phía trên và phía dưới đường thẳng chia hai nửa 3D bounding box theo trục Oy. Nếu vùng nằm trên có diện tích lớn hơn rất nhiều so với vùng bên dưới, thành phần liên thông này được gán nhãn 10 ngược lại gán nhãn 20. Trong trường hợp độ lệch diện tích không lớn, thành phần liên thông này sẽ được đưa vào hàm `split_component()` để phân tách thành 2 thành phần liên thông. Hàm này sẽ thực hiện nhân mặt nạ của thành phần liên thông này với ảnh gốc và áp dụng thuật toán OSTU để lấy ngưỡng và loại đi những điểm ảnh nằm trên ngưỡng (là những vị trí nằm giữa 2 phổi có độ sáng lớn hơn), trong trường hợp 2 phổi quá gần nhau và chưa thể phân tách thành 2 thành phần liên thông bằng thuật toán OSTU (loop ≥ 1) hàm `split_component()` sẽ tìm tọa độ y có tổng số pixel nhỏ nhất và gán các giá trị này bằng 0. Sau bước này, thực hiện gọi đệ quy hàm `split_mask()` để thực hiện gán nhãn cho từng thành phần liên thông.

Algorithm 2 Split Lung Function

```

1: function SPLIT_MASK(image, mask, local_box, loop=0)
2:   new_mask  $\leftarrow$  zeros_like(mask)
3:   center_y  $\leftarrow$  (local_box[0] + local_box[1])/2
4:   if any(mask) then
5:     labeled_mask, num_cc  $\leftarrow$  label(mask)
6:     for i  $\leftarrow$  1 to num_cc do
7:       label_mask  $\leftarrow$  (labeled_mask == i)
8:       area  $\leftarrow$  sum(label_mask)
9:       upper_area  $\leftarrow$  sum(label_mask[: center_y, :])
10:      lower_area  $\leftarrow$  sum(label_mask[center_y :, :])
11:      bbox  $\leftarrow$  bbox_2D(label_mask)
12:      height  $\leftarrow$  bbox[3] - bbox[2]
13:      if upper_area > 0.8 * area and height < 256 then
14:        new_mask  $\leftarrow$  new_mask + label_mask  $\times$  10
15:      else if lower_area > 0.8 * area and height < 256 then
16:        new_mask  $\leftarrow$  new_mask + label_mask  $\times$  20
17:      else
18:        _label_mask  $\leftarrow$  split_component(image, label_mask, bbox, loop)
19:        new_mask  $\leftarrow$  new_mask + _label_mask
20:      end if
21:    end for
22:  end if
23:  return new_mask
24: end function

```

Algorithm 3 Split Component Function

```

1: function SPLIT_COMPONENT(image, labeled_mask, bbox, loop=0)
2:   center_w ← (bbox[1] + bbox[0])/2
3:   delta ← (bbox[3] – bbox[2]) × 0.4
4:   indence ← ([bbox[2] + delta : bbox[3] – delta, bbox[0] : bbox[1]])
5:   tmp ← labeled_mask[indence]
6:   if loop = 0 then
7:     img ← image[indence]
8:     img ← normalize(img)
9:     binary_img ← ostu_threshold(img)
10:    labeled_mask[indence] ← tmp ∧ ¬binary_img
11:   else
12:     min_val ← 1e6
13:     index ← 0
14:     _bbox ← bbox_2D(tmp_1)
15:     for i ← _bbox[2] + 10 to _bbox[3] – 10 do
16:       if sum(tmp[i, :]) < min_val then
17:         min_val ← sum(tmp[i, :])
18:         index ← i
19:       end if
20:       if sum(tmp[i, :]) = min_val then
21:         index ← (index + i)/2  $\triangleright$  Nếu có nhiều vị trí nhỏ nhất thì gán 0
          ở điểm trung tâm giữa các vị trí
22:       end if
23:     end for
24:     tmp[index, :] ← 0
25:   end if
26:   return labeled_mask
27: end function

```

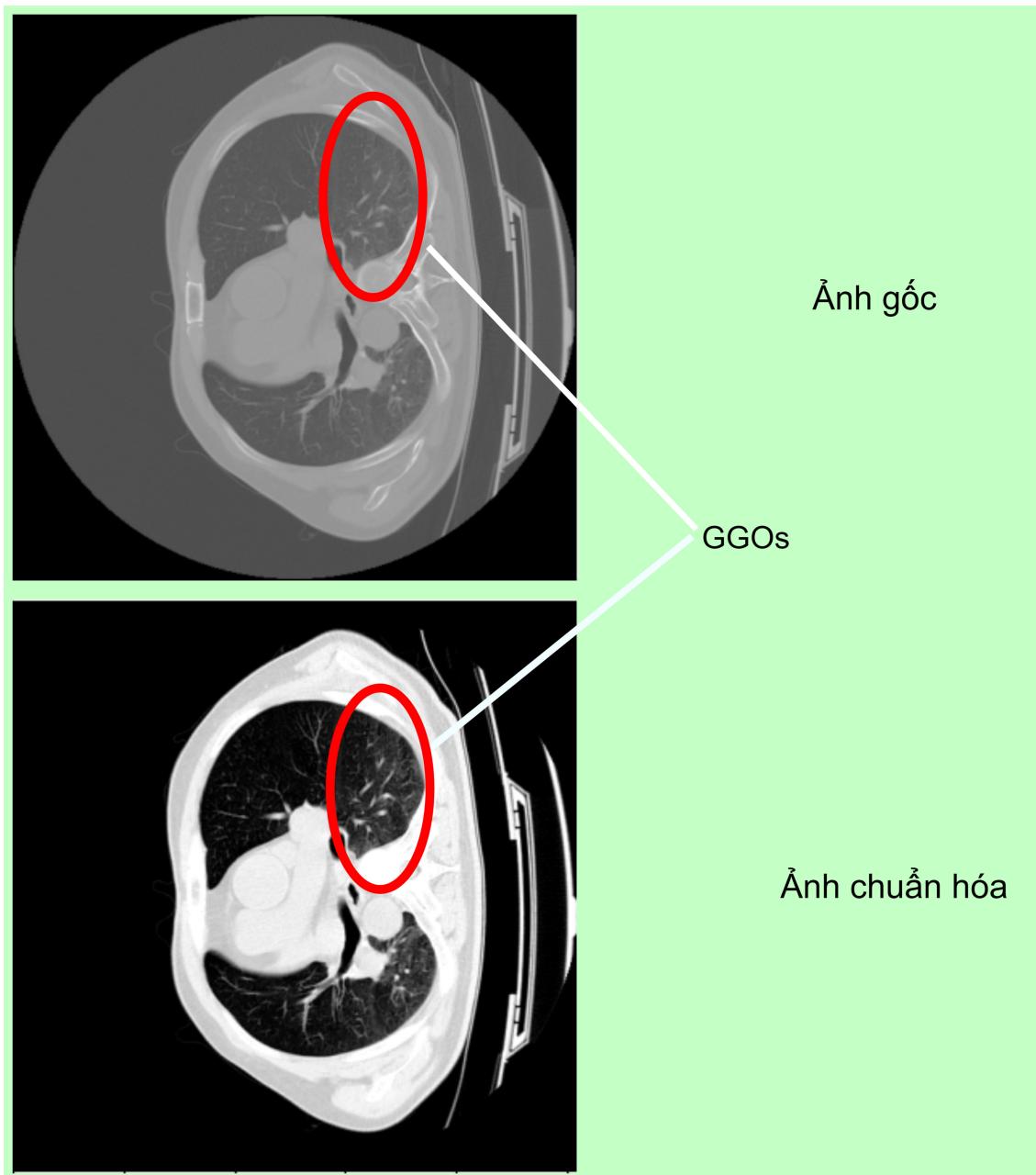
3.3 Giai đoạn 2: Phân đoạn GGOs

Như đã giới thiệu ở trên, giai đoạn này sẽ thực hiện tiền xử lý dữ liệu sau đó 2 mô hình Unet 2D và 3D được huấn luyện để phân đoạn GGOs

3.3.1 Tiền xử lý dữ liệu

Do tình trạng bệnh của bệnh nhân và độ tuổi của họ đều ảnh hưởng đến độ sáng và kích thước của vùng tổn thương dẫn đến việc khó có thể đưa ra một phương thức chuẩn hóa hàng loạt trên tất cả các mẫu, vì thế tôi đề xuất một phương án chuẩn hóa độ sáng của hình ảnh theo từng mẫu dựa vào kết quả phân đoạn phổi của giai đoạn 1. Trước tiên tôi thực hiện phân tích độ sáng trên toàn bộ các điểm ảnh được gán nhãn bị bệnh, kết quả cho thấy 90% giá trị điểm ảnh có giá trị từ -961 đến +86 theo đơn vị HU. Tuy nhiên vì mức sáng và kích thước trên mỗi hình ảnh lại khác nhau dẫn đến miền vừa rồi bỏ qua khá nhiều điểm ảnh ở các mẫu có mức sáng trung bình

thấp và kích thước nhỏ. Vì vậy tôi chọn khoảng chuẩn hóa từ -1000 đến +100, đây là khoảng có mức mốc míc trên các mẫu nhỏ và phù hợp với dữ liệu (Trên thang đo HU, -1000 biểu diễn cho vùng không khí và khoảng -100 đến +100 biểu diễn cho các mô). Tôi thực hiện chuẩn hóa theo phương pháp Z-score với kỳ vọng và trung vị được tính theo mặt nạ phổi có được ở giai đoạn 1. Lý do tôi chọn sử dụng mặt nạ phổi để chuẩn hóa thay vì mặt nạ GGOs bởi vì tình trạng và độ sáng của những mô không bị bệnh ở tất cả các hình ảnh là không đều và rất khó để sử dụng một giá trị kỳ vọng và phương sai chung (như sử dụng mặt nạ GGOs) để chuẩn hóa. Vùng nhiễm bệnh thường có độ sáng lớn hơn vùng bình thường vì thế khi sử dụng chuẩn hóa Z-score bằng mặt nạ phổi, vùng tổn thương sẽ nổi bật hơn. Hình 3.4 thể hiện hiệu quả của phương pháp chuẩn hóa này với hình ảnh có vùng tổn thương mờ và khó quan sát.



Hình 3.4: So sánh hình ảnh trước khi chuẩn hóa và sau khi chuẩn hóa bằng Z-score: Ảnh trên là ảnh trước khi chuẩn hóa, hình ảnh phía dưới cho thấy phần tổn thương GGOs đã hiển thị nổi bật hơn so với nền xung quanh.

Một khó khăn khác của bài toán là độ phân giải của hình ảnh là không đồng đều. Tôi đã thử nghiệm trên mô hình 3D với hai tập dữ liệu đầu vào bao gồm dữ liệu gốc và dữ liệu chuẩn hóa độ phân giải. Với dữ liệu chuẩn hóa độ phân giải, tôi thực hiện resampling dữ liệu để đưa giá trị voxel spacing của dữ liệu về giá trị trung bình là $5 \times 0.74 \times 0.74$. Phương pháp này tỏ ra khá hiệu quả trong giai đoạn huấn luyện khi hội tụ rất nhanh sau khoảng 15 epoch với điểm $f1_score = 0.84$ tuy nhiên điểm Dice score trên tập kiểm thử lại thấp hơn khá nhiều so với việc sử dụng dữ liệu gốc, nguyên nhân có thể do một số hình ảnh có độ phân giải cao hơn khi đưa về độ phân giải nhỏ đã làm mất thông tin dẫn đến việc phân đoạn thiếu

chính xác.

Sau khi chuẩn hóa, dữ liệu sẽ được cắt theo từng patch trước khi đưa vào huấn luyện ở 2 mô hình. Dựa vào mặt nạ phân đoạn phổi đã chia thành 2 phần phổi trái và phổi phải tôi thực hiện cắt hình ảnh dựa theo 2 mặt nạ phổi này, các patch sẽ được cắt sát với 3d bounding box tương ứng với từng mặt nạ sau đó sẽ được padding và resize về đúng kích thước 256x256.

Với từng patch tôi cũng thực hiện gán nhãn lại cho mặt nạ GGOs của tập dữ liệu huấn luyện. Giá trị của điểm ảnh tỉ lệ nghịch với kích thước thành phần liên thông chứa nó. Quá trình gán nhãn lại được thực hiện theo mã giả 4

Algorithm 4 generate_label(mask)

```

1: function GENERATE_MASK(image, labeled_mask, bbox, loop=0)
2:   shape ← get_shape(mask)           ▷ Get the shape of the input mask
3:   new_mask ← create_array(shape, dtype=float16)
4:   labels, num_components ← connected_components(mask)
5:   for i in 1 to num_components do
6:     label_mask ← (labels == i)
7:     area ← sum(label_mask)
8:     if area > 100 then
9:       value ← 1 - √{area/sum(ones_like(mask))}
10:      new_mask ← new_mask + label_mask * value
11:    end if
12:   end for
13:   return new_mask                  ▷ Return the new mask
14: end function

```

Với từng thành phần liên thông, tôi tính khối lượng của nó so với tổng khối lượng của patch (với 2D là 256x256, 3D là 256x256x12) và gán nhãn bằng hiệu của 1 và căn bậc hai của tỉ lệ này. Điều này sẽ giúp việc phạt nặng hơn cho những vùng tổn thương nhỏ trong các hàm lõi.

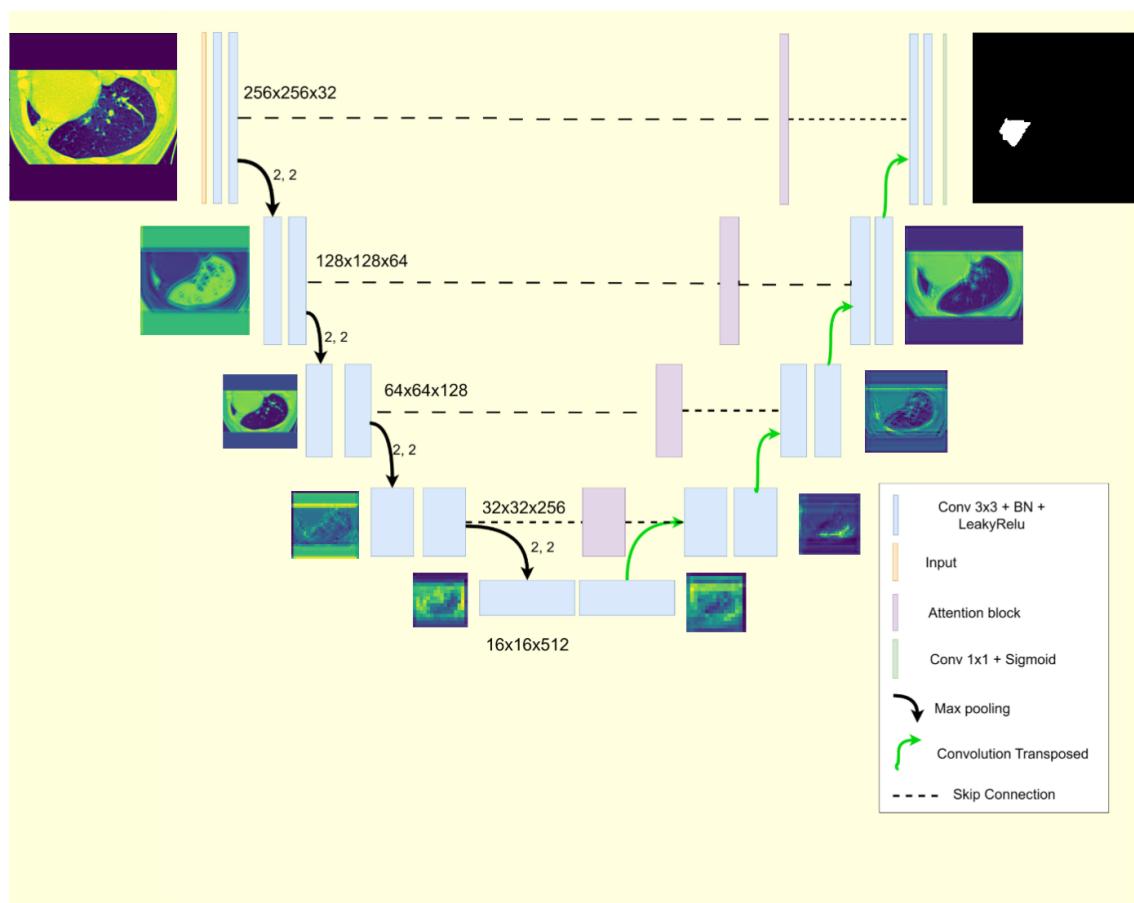
Để giảm hiện tượng overfitting tôi thực hiện tăng cường dữ liệu bằng các phương pháp phổ biến như lật, xoay, thêm nhiễu gaussian, thay đổi cường độ ánh sáng, độ tương phản.

3.3.2 Mô hình 2D U-net

Kích thước và độ sáng của GGOs trên từng bệnh nhân là khác nhau tuy nhiên đối với bệnh nhân có tình trạng bệnh càng nặng, đồng nghĩa với việc kích thước vùng GGOs lớn hơn thì độ sáng của vùng GGOs cũng rõ ràng hơn. Với những tổn thương có kích thước càng nhỏ thì nó càng nằm trên ít lát cắt. Hai mô hình 2D và 3D được huấn luyện để phân đoạn trong 2 trường hợp cụ thể khác nhau. Mô hình

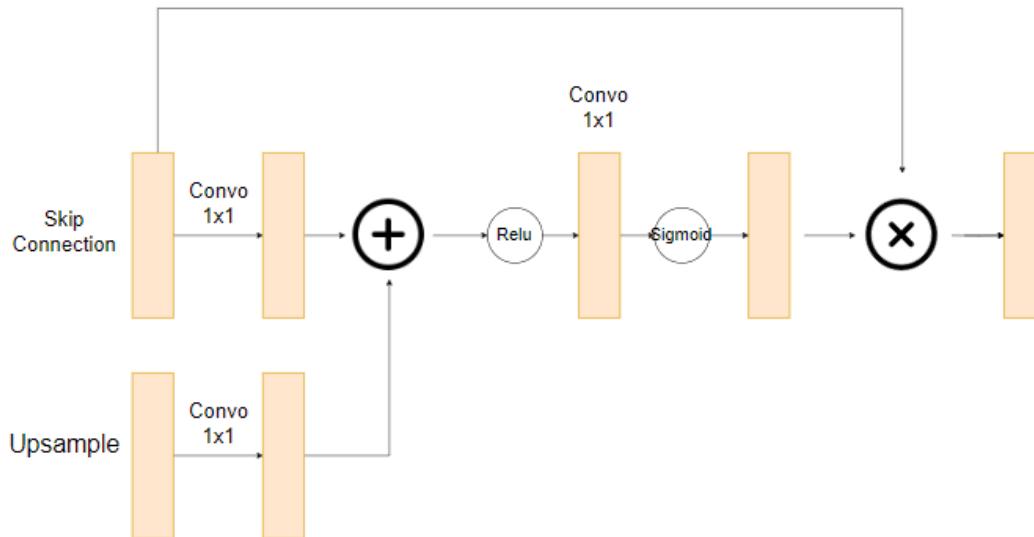
3D đóng vai trò phân đoạn những vùng tổn thương lớn hơn và rõ ràng hơn đồng thời đóng vai trò liên kết trên miền không gian trong giai đoạn kết hợp 2 mô hình. Mô hình 2D đóng vai trò phát hiện và phân đoạn những tổn thương nhỏ và mờ hơn hoặc những tổn thương lớn nhưng nằm rải rác trên các lớp do số lượng lát cắt nhỏ và không phát hiện hiệu quả trên mô hình 3D.

Kiến trúc 2D U-net được thiết kế tương tự như mô hình U-net cơ bản tuy nhiên tại mỗi kết nối skip connection có một cơ chế chú ý được thêm vào để tăng hiệu quả trích xuất tính năng. Mỗi block attention nhận đầu vào từ lớp tích chập ở encoder và lớp upsample cùng cấp ở decoder. Hai đầu vào này được cho qua một tích chập với kernel = 1 với số channel không đổi và lớp chuẩn hóa Batch Normalization. Hai kết quả này được cộng lại và đưa qua một hàm kích hoạt Leaky RELU. Tensor này tiếp tục được đưa qua một lớp tích chập với filter và kernel = 1 sau đó được chuẩn hóa bằng Batch Normalization. Kết quả cuối cùng được đưa qua hàm kích hoạt Sigmoid và nhân element-wise với skip connection để bỏ đi những tính năng không quan trọng và tập trung vào các tính năng quan trọng hơn.



Hình 3.5: 2D Unet Architecture: Mô hình Unet kết hợp với cơ chế chú ý gồm 5 tầng. Ảnh đầu vào có kích thước $256 \times 256 \times 1$, mỗi tầng bao gồm 2 convolution block (Conv 3x3, BatchNorm, Leaky Relu), số lượng channel ở mỗi tầng bắt đầu từ 32. Các attention block được đặt ở kết nối skip connection của mô hình

Mô hình 2D U-net 3.5 nhận đầu vào là ảnh 2D có kích thước 256x256. Kiến trúc mô hình gồm 5 tầng với mỗi tầng bao gồm 2 block Convolution, Batch Normalization, LeakyRelu Activation. Số channel lần lượt là 32, 64, 128, 256, 512. Có 4 Attention block 3.6 được đặt giữa skip connection và block Upsampling cùng cấp.



Hình 3.6: Attention Block: Nhận đầu vào là đầu ra của lớp Upsample(Transposed Convolution) và đầu ra của lớp Convolution cùng cấp (Skip Connection). Các lớp này được đưa qua các lớp Convolution 1x1 và các hàm kích hoạt để tính trọng số cho các tính năng.

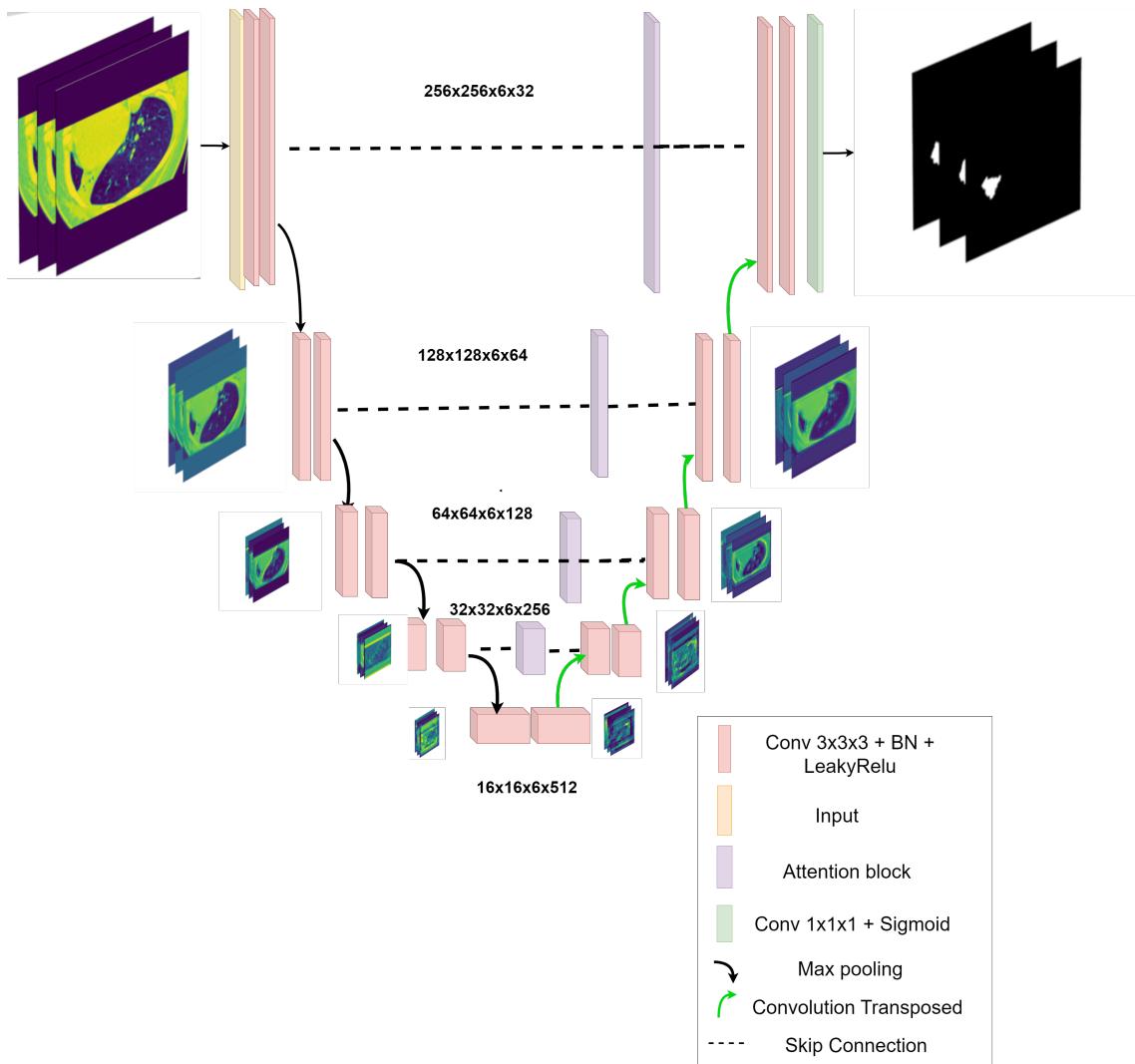
Tôi thực hiện phương pháp k-fold cross validation với $k = 5$ trên tập dữ liệu huấn luyện. Thuật toán tối ưu được sử dụng để huấn luyện mô hình này là Adam với learning rate ban đầu là 10^{-4} , learning rate sẽ tự động giảm một nửa sau 10 epoch nếu giá trị loss trên tập xác nhận không giảm.

3.3.3 Mô hình 3D U-net

Dữ liệu đầu vào của mô hình này có kích thước 256x256x12 được thu thập theo mặt nạ từng bên phổi với bước nhảy theo trục z = 6 và bước nhảy theo trục x bằng 0.2 chiều rộng của 3D bounding box mặt nạ phổi.

Tương tự mô hình 2D U-net, mô hình 3D U-net (Hình 3.7) cũng được thiết kế theo mô hình U-net thông thường và có thêm các block attention ở các kết nối skip connection. Mô hình này gồm 4 tầng với số channel lần lượt là 32, 63, 128, 256, 512. Các lớp tích chập 3D được thay thế cho tích chập 2D với kernel size = (3,3,3), MaxPooling 3D được thay thế cho MaxPooling 2D với pool_size lần lượt là (2,2,1), (2,2,1), (2,2,1), (2,2,1). Các lớp Convolution Transpose 3D có strides tương tự Max

Pooling 3D cùng cấp.



Hình 3.7: 3D Unet Architecture: Mô hình Unet kết hợp với cơ chế chú ý gồm 5 tầng. Ảnh đầu vào có kích thước $256 \times 256 \times 6 \times 1$, mỗi tầng bao gồm 2 convolution block (Conv $3 \times 3 \times 3$, BatchNorm, Leaky Relu), số lượng channel ở mỗi tầng bắt đầu từ 32. Các attention block được đặt ở kết nối skip connection của mô hình

Tương tự mô hình 2D, tôi sử dụng thuật toán tối ưu Adam với learning rate ban đầu là 10^{-4} và giảm 50% sau 10 epoch không cải thiện lỗi. Tôi cũng thực hiện k-fold cross validation với $k=5$ trên mô hình này và thực hiện Ensemble learning trên các kết quả.

Để tăng tốc độ huấn luyện, tôi sử dụng kỹ thuật mixed precision. Đây là một kỹ thuật học sâu sử dụng sự kết hợp giữa dấu phẩy động có độ chính xác thấp và số cố định với độ chính xác cao. Phần lớn các thiết bị GPU hiện hay có hỗ trợ tính toán dấu phẩy động 16 bit (FP16), khi tính toán trên FP16, GPU sẽ tính toán nhanh hơn so với dấu phẩy động 32 bit (FPP32) tuy nhiên điều này sẽ làm giảm độ chính xác của các tham số. Vì thế khi sử dụng phương pháp này, tôi sử dụng kiểu dữ liệu

float16 cho các lớp trích xuất tính năng và kiểu dữ liệu float32 cho lớp đầu ra cuối cùng để đảm bảo tính chính xác mà vẫn có thể tăng tốc độ tính toán. Tensorflow hỗ trợ tính năng này bằng cách gọi câu lệnh:

```
mixed_precision.set_global_policy('mixed_float16')
```

3.3.4 Hậu xử lý

Các thuật toán sử dụng trong phần này được mô tả bằng các mã giả 5, 6. Trong mô hình 3D U-net tôi sử dụng phương pháp đánh trọng số Gaussian để cải thiện độ chính xác của mô hình. Phương pháp đánh trọng số Gaussian là một kỹ thuật thường được sử dụng trong phân đoạn hình ảnh để tăng trọng số các điểm gần trung tâm và giảm ảnh hưởng của những điểm nằm xa trung tâm. Quy trình chung của phương pháp này bao gồm xác định điểm trung tâm, tính khoảng cách của các điểm trong mặt nạ dự đoán so với điểm trung tâm, tính trọng số Gaussian và gán các trọng số cho những điểm trong mặt nạ. Cụ thể với bài toán này, thuật toán sẽ được thực hiện như sau. Với mặt nạ dự đoán của từng patch, thực hiện tìm những thành phần liên thông với ngưỡng giá trị xác suất lớn hơn 0.4 và tổng khối lượng lớn hơn 100, tâm của bounding box của từng thành phần liên thông này được xác định là các điểm trọng. Duyệt qua từng điểm trọng tâm, thực hiện tính mặt nạ gaussian bằng hàm gaussian_filter của thư viện "scipy.ndimage". Cụ thể, với mỗi một điểm trọng tâm, một mảng có kích thước bằng mặt nạ được khởi tạo và gán điểm trọng tâm bằng 1, sau đó sử dụng hàm gaussian_filter để tính toán giá trị gaussian cho từng điểm tính từ điểm trọng tâm. Đối với mô hình 3D, phụ thuộc vào voxel_spacing(giá trị khoảng cách các điểm ảnh trong không gian) khoảng cách trên trục Oz sẽ được nhân với tỉ lệ voxel tương ứng để phù hợp với độ phân giải trong không gian của hình ảnh. Sau khi có mặt nạ gaussian, chuẩn hóa mặt nạ về khoảng từ 0-1 và gán các giá trị bằng 0 bằng với giá trị nhỏ nhất để tránh trường hợp chia cho 0. Khi thực hiện dự đoán sẽ có những hình ảnh chồng chéo nhau, mặt nạ cuối cùng là tổng các mặt nạ đã gán trọng số chia cho tổng giá trị trọng số.

Các mô hình được huấn luyện trên bộ dữ liệu có kích thước khá nhỏ và mô hình huấn luyện không quá phức tạp vì thế tôi sử dụng kỹ thuật Ensemble Learning để kết hợp những mô hình đã huấn luyện nhằm giảm tình trạng quá khớp đồng thời tăng độ chính xác và hiệu suất dự đoán cho mô hình. Hai mô hình 2D và 3D đều sử dụng k-fold cross validation nên tôi tận dụng cả 10 mô hình huấn luyện để áp dụng Ensemble Learning. Chiến lược sử dụng trên từng mô hình 2D và 3D là Majority voting.

Sau khi tổng hợp được hai mô hình 2D và 3D, do hai mô hình này có vai trò phân đoạn khác nhau nên cần một phương pháp kết hợp chúng lại. Dựa vào tính

Algorithm 5 Gaussian Weighting Algorithm

```

1: function GAUSSIAN_WEIGHTING(image,  $\sigma$ , threshold, min_region_size, voxel_ratio)
2:     labeled_image, num_labels  $\leftarrow$  label(image  $>$  threshold)     $\triangleright$  Tìm các thành
   phần liên thông
3:     regions  $\leftarrow$  regionprops(labeled_image)       $\triangleright$  Tính khôi lượng và trung tâm
   các TPLT
4:     weighted_image  $\leftarrow$  zeros_like(image, dtype=float)
5:     sigmas  $\leftarrow$  [256 *  $\sigma$ , 256 *  $\sigma$ , 256 *  $\sigma$  / voxel_spacing]
6:     for region in regions do
7:         if region.area  $\geq$  min_region_size then
8:             center  $\leftarrow$  region.centroid
9:             tmp  $\leftarrow$  zeros_like(image)
10:            tmp[center]  $\leftarrow$  1
11:            gaussian_map  $\leftarrow$  gaussian_filter(tmp, sigmas)
12:            gaussian_map  $\leftarrow$  norm(gaussian_map)
    $\triangleright$  Chuẩn hóa về khoảng 0-1
13:            gaussian_map[gaussian_map == 0]            $\leftarrow$ 
   min(gaussian_map[gaussian_map != 0])
    $\triangleright$  Gán các giá trị = 0 thành giá trị khác 0 nhỏ nhất
14:            weighted_image  $\leftarrow$  weighted_image + gaussian_map
15:        end if
16:    end for
17:    return weighted_image
18: end function

```

chất của vùng tổn thương phổi là có độ sáng lớn hơn những mô phổi không bị bệnh, tôi đề xuất một phương án kết hợp hai mô hình dựa trên việc loại bỏ các vùng dự đoán có độ sáng trung bình bé hơn độ sáng trung bình của mặt nạ phổi có được từ giai đoạn 1. Tôi tiếp tục sử dụng thuật toán tìm thành phần liên thông để kết hợp hai mô hình này. Trước tiên kết quả của hai mô hình được thực hiện phép AND với nhau để tìm ra mặt nạ giao nhau giữa 2 mặt nạ. Tìm các thành phần liên thông trên 2 mặt nạ và giữ lại các thành phần liên thông giao nhau, các thành phần này được gọi là strong GGOs và sẽ được giữ lại. Những thành phần liên thông còn lại được gọi là weak GGOs được lọc bằng cách so sánh độ sáng trung bình với độ sáng của mặt nạ phổi. Mặt nạ phổi được sử dụng để tính toán độ sáng trung bình là mặt nạ phổi ở giai đoạn 1 được trừ đi các mặt nạ dự đoán của hai mô hình. Tôi lựa chọn ngưỡng độ sáng là $\mathbb{E}[L] - 2 * \text{Var}[L]$ với $\mathbb{E}[L]$ và $\text{Var}[L]$ lần lượt là giá trị trung bình và phương sai của các điểm ảnh trong mặt nạ phổi.

Algorithm 6 Combine Model Algorithm

```

1: function COMBINE_MODEL(image, _2d_mask, lung_mask, _3d_mask)
2:   intersect_mask  $\leftarrow$  logical_and(_2d_mask, _3d_mask)
3:   slabels, snum_components  $\leftarrow$  label(_2d_mask)
4:   llabels, lnum_components  $\leftarrow$  label(_3d_mask)
5:   strong_ggo  $\leftarrow$  zeros(shape(_2d_mask))
6:   weak_ggo  $\leftarrow$  zeros(shape(_2d_mask))
7:   for i  $\leftarrow$  1 to snum_components do
8:     _2d_label  $\leftarrow$  slabels == i
9:     if sum(_2d_label  $\wedge$  intersect_mask) > 0 then
10:       strong_ggo  $\leftarrow$  _2d_label  $\vee$  strong_ggo
11:     end if
12:     if sum(_2d_label  $\wedge$  lung_mask) > 20 then
13:       weak_ggo  $\leftarrow$  weak_ggo + _2d_label
14:     end if
15:   end for
16:   for i  $\leftarrow$  1 to lnum_components do
17:     _3d_label  $\leftarrow$  llabels == i
18:     if sum(_3d_label  $\wedge$  intersect_mask) > 0 then
19:       strong_ggo  $\leftarrow$  _3d_label  $\vee$  strong_ggo
20:     end if
21:     if sum(_3d_label  $\wedge$  lung_mask) > 20 then
22:       weak_ggo  $\leftarrow$  weak_ggo + _3d_label
23:     end if
24:   end for
25:   filtered_lung_mask  $\leftarrow$  (lung_mask  $\wedge$   $\neg$ (_2d_mask  $\vee$  _3d_mask)
26:   mean_value  $\leftarrow$  mean(image[filtered_lung_mask])
27:   std  $\leftarrow$  std(image[filtered_lung_mask])
28:   labels, num_components  $\leftarrow$  label(weak_ggo)
29:   for i  $\leftarrow$  1 to num_components do
30:     weak_label  $\leftarrow$  labels == i
31:     if mean(image[nonzero(weak_label)]) > mean_value - std then
32:       strong_ggo  $\leftarrow$  logical_or(strong_ggo, weak_label)
33:     end if
34:   end for
35:   return strong_ggo
36: end function

```

3.3.5 Hàm lỗi

Như đã đề cập ở các chương trước, một trong những khó khăn gập phải của bài toán GGOs là vấn đề mâu thuẫn giữa dữ liệu. Các tổn thương gập phải có thể có kích thước lớn cũng có thể có kích thước rất nhỏ. Vì vậy cần có một hàm lỗi phù hợp để giải quyết vấn đề này. Trong đồ án này, trừ mô hình phân đoạn phổi ở giai đoạn 1 tôi chỉ sử dụng hàm lỗi Dice Loss để huấn luyện, các mô hình ở giai đoạn 2 đều

được sử dụng một hàm lỗi kết hợp giữa hàm lỗi Dice loss và những hàm lỗi khác nhau nhằm giải quyết vấn đề mất cân bằng dữ liệu nêu trên.

Dice loss là một hàm lỗi được sử dụng phổ biến trong tác vụ phân đoạn hình ảnh. Đây là hàm loss đo lường độ tương đồng giữa nhãn và kết quả dự đoán. Công thức của Dice loss được tính dựa trên chỉ số Dice coefficient (còn được gọi là F1-score) giữa hai tập hợp. Dice coefficient là một phép đo sự giống nhau giữa hai tập hợp, được tính bằng cách so sánh số lượng phần tử chung giữa hai tập hợp với tổng số phần tử của chúng. Công thức của Dice loss được biểu diễn như sau (3.1)

$$DiceLoss(y, \hat{y}) = 1 - \frac{2 * y \cap \hat{y}}{|y| + |\hat{y}|} \quad (3.1)$$

Trong đó:

y là nhãn thực tế.

\hat{y} là dự đoán.

$y \cap \hat{y}$ là số lượng phần tử chung giữa hai tập hợp y và $|\hat{y}|$.

$|y|$ là số lượng phần tử của tập hợp y .

$|\hat{y}|$ là số lượng phần tử của tập hợp \hat{y} .

Hàm Dice loss thường cho kết quả tốt trong bài toán phân đoạn hình ảnh vì nó tập trung vào đánh giá sự tương đồng giữa các vùng được phân đoạn thay vì tập trung vào việc phân loại từng pixel độc lập, Dice loss thường cho kết quả tốt với những bài toán có dữ liệu mất cân bằng, tuy nhiên Dice loss không giải quyết triệt để vấn đề dữ liệu mất cân bằng đối với những trường hợp dữ liệu bị mất cân bằng quá lớn. Vì lý do đó tôi đã sử dụng hàm Dice loss kết hợp cùng với một số hàm lỗi sau đây.

Weighted Cross-entropy (WCE) loss là một biến thể của hàm Cross-entropy (CE) loss, là một hàm lỗi thường được sử dụng trong bài toán phân loại và phân đoạn hình ảnh. Đối với bài toán phân đoạn hình ảnh nhị phân hàm lỗi thường sử dụng là Binary Cross Entropy (BCE) có công thức là :

$$BCE(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^N \cdot y_i \cdot \log(\hat{y}_i) \quad (3.2)$$

Tuy nhiên, BCE có một nhược điểm là xử lý kém trong điều kiện dữ liệu mất cân bằng, vì thế hàm lỗi WCE sử dụng thêm trọng số cho từng điểm ảnh để giải quyết vấn đề mất cân bằng của CE. Công thức hàm WCE như sau (3.3):

$$\text{WCE}(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^N w_i \cdot y_i \cdot \log(\hat{y}_i) \quad (3.3)$$

Trong đó N là số lượng điểm dữ liệu trong tập huấn luyện.

y_i là nhãn thực tế của điểm dữ liệu thứ i (0 hoặc 1).

\hat{y}_i là dự đoán của mô hình cho điểm dữ liệu thứ i.

\log là hàm logarithm tự nhiên.

Các trọng số w_i của hàm WCE được gán theo nhãn của hình ảnh đã được gán nhãn lại trong bước tiền xử lý dữ liệu. Hàm lỗi kết hợp giữa Dice loss và WCE gọi là Combo loss.

Focal loss được giới thiệu lần đầu vào năm 2017 bởi nhóm tác giả Tsung-Yi Lin, Priya Goyal, ... [18], hàm này được đề xuất để giải quyết vấn đề mất cân bằng dữ liệu trong bài toán object detection. Dựa trên những nhược điểm của hàm CE nêu ở trên, hàm Focal loss được đề xuất có thêm nhân tử... giúp tăng tác động của những class khó dự báo và giảm tác động của class dễ dự báo. Focal Loss trong bài toán binary segmentation là một biến thể của Focal Loss trong trường hợp này chúng ta có 2 lớp là positive class và negative class. Công thức của Focal Loss trong bài toán binary segmentation giữa kết quả dự đoán và nhãn thực tế tại pixel i (3.4):

$$Focal_loss(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^N (\alpha \cdot (1 - \hat{y}_i)^\gamma \cdot \log(\hat{y}_i)) \quad (3.4)$$

Trong đó:

Trong đó N là số lượng điểm dữ liệu trong tập huấn luyện.

y_i là nhãn thực tế của điểm dữ liệu thứ i (0 hoặc 1).

\hat{y}_i là dự đoán của mô hình cho điểm dữ liệu thứ i.

\log là hàm logarithm tự nhiên.

α và γ là hai tham số điều chỉnh của Focal Loss.

Tham số α giúp giảm trọng số cho lớp đa số và tăng trọng số cho lớp thiểu số, γ giúp tăng mức độ chú trọng vào các pixel khó phát hiện. Với những pixel khó phát hiện hay có tần suất xuất hiện thấp hơn sẽ có p_i nhỏ và $1 - p_i$ càng lớn dẫn đến trọng số của các pixel này được tăng lên và ngược lại, các pixel dễ phát hiện sẽ bị giảm ảnh hưởng lên hàm loss. Tôi sử dụng hàm kết hợp giữa Dice loss và Focal loss được gọi là DiceFocalLoss (DFL). Tuy hàm Focal loss đã giải quyết khá tốt

vẫn đề mất cân bằng dữ liệu, tuy nhiên với bài toán mất cân bằng dữ liệu khá lớn như bài toán GGOs này, trọng số của lớp thiểu số vẫn có ảnh hưởng không đủ lớn đến hàm lỗi và hàm Focal Loss có thể sẽ không hoạt động và dễ bị overfit hay hoặc đánh giá không công bằng.

Boundary loss [16] là một hàm lỗi được đề xuất vào năm 2021 bởi H. Kervadec và các cộng sự. Các tác giả nhận ra được điểm yếu của các hàm lỗi thông dụng được sử dụng trong tác vụ phân đoạn hình ảnh, như Dice loss thường gặp khó khăn trong việc tối ưu những bài toán có mặt nạ rất nhỏ, khi số pixel là positive rất nhỏ so với negative, chỉ cần một thay đổi nhỏ trong dự đoán cũng sẽ làm dice loss biến động rất nhiều và điều này dẫn đến việc tối ưu hóa mô hình rất khó khăn. Thay vì sử dụng thông tin về vùng như Dice loss, Boundary loss sử dụng thông tin về biên. Để xây dựng hàm boundary loss, ban đầu cần xác định các vùng mục tiêu trên nhãn và tính toán boundary (biên) của các vùng này. Ở giai đoạn tiền xử lý dữ liệu, tôi đã gán nhãn cho từng thành phần liên thông trong mỗi patch với trọng số khác nhau dựa vào khối lượng của thành phần đó so với tổng khối lượng của từng patch. Tiếp theo các điểm ảnh trong từng vùng sẽ được tính khoảng cách so với biên của nó, sau bước này ta có được một distance map để thực hiện xây dựng hàm lỗi. Boundary loss bằng tổng các giá trị xác suất trong mặt nạ dự đoán nhân với distance map có được. Khi chỉ sử dụng hàm boundary loss riêng biệt cho việc học sâu các tác giả có nhận xét rằng nó không hiệu quả khi sử dụng một mình mà cần sử dụng kết hợp với các hàm lỗi khác để đạt được hiệu quả tốt nhất. Hàm lỗi kết hợp giữa Dice loss và Boundary loss được gọi là DiceBoundary Loss.

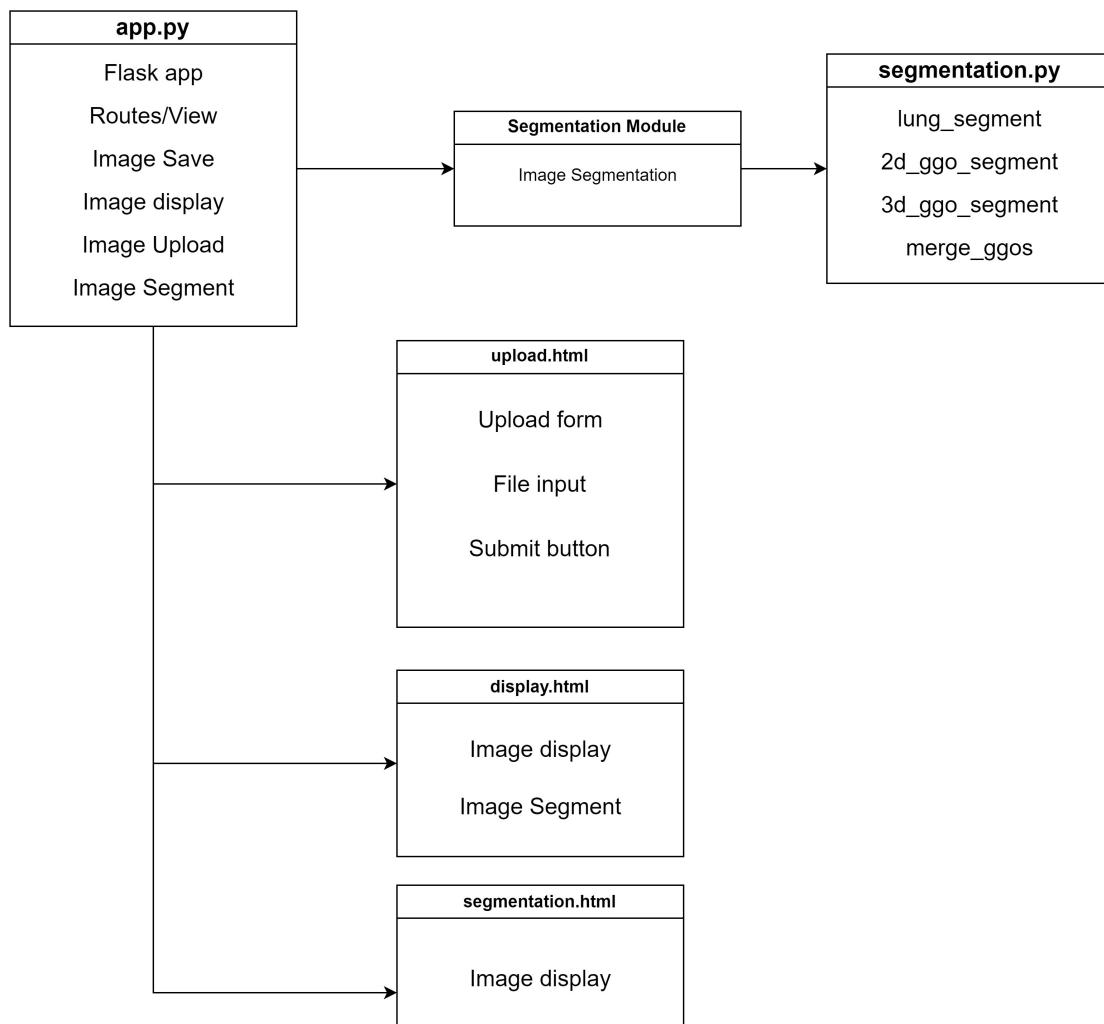
3.3.6 Website ứng dụng mô hình

Tôi sử dụng Framework Flask để tạo một trang web đơn giản để ứng dụng kết quả của mô hình này. Website này có chức năng upload hình ảnh 3D có định dạng NIfTI sau đó hiển thị hình ảnh 3D bằng cách hiển thị các lớp qua nút bấm và thao tác lăn chuột. Tôi cũng thực hiện chuyển vị và tăng độ phân giải bằng phương pháp nội suy bậc ba để hiển thị thêm hình ảnh phổi theo hình chiếu vuông góc với cơ thể. Kiến trúc của website được biểu diễn trong hình 3.8

Luồng hoạt động của ứng dụng như sau: Sau khi upload hình ảnh định dạng NIfTI ở giao diện Upload, hình ảnh được lưu vào 2 thư mục, thư mục thứ nhất chứa hình ảnh các lát cắt của hình ảnh 3D, thư mục thứ 2 chứa lát cắt của hình ảnh 3D đã chuyển vị. Sau khi lưu trữ các hình ảnh, giao diện Display được gọi và trực quan hóa hình ảnh 3D với 2 hình chiếu bao gồm lát cắt ngang ở hình gốc và hình chiếu vuông góc với cơ thể, các hình ảnh được hiển thị từng lát cắt và dùng nút bấm hoặc lăn chuột để chuyển sang lát cắt khác. Một luồng chạy trong nền được gọi khi giao

CHƯƠNG 3. PHƯƠNG PHÁP ĐỀ XUẤT

diện display được hiển thị, luồng này thực hiện chức năng phân đoạn hình ảnh và trả về thư mục lưu trữ hình ảnh. Tại giao diện này, tôi thêm một chức năng đếm để ước tính thời gian chờ đợi phân đoạn dựa vào số lát cắt của hình ảnh gốc. Khi luồng trong nền hoàn tất, nút bấm Show result trên màn hình sẽ chuyển trạng thái sang Enable. Giao diện Segmentation sẽ hiển thị sau khi bấm vào nút Show result, giao diện này hiển thị các hình ảnh đã phân đoạn tương tự như cơ chế của giao diện Display.



Hình 3.8: Kiến trúc của Website ứng dụng: Website sử dụng Flask Framework bao gồm file app.py là nơi tạo và cấu hình các chức năng của web, 3 file HTML và các file backend Python chứa các chức năng phân đoạn hình ảnh.

CHƯƠNG 4. ĐÁNH GIÁ THỰC NGHIỆM

Với đề xuất đã đưa ra ở chương 3, trong chương này tôi trình bày về các phương pháp đánh giá mô hình và so sánh sự hiệu quả của phương pháp với những phương pháp đã có và những phương pháp được đưa ra trong thử thách COVID-19-20

4.1 Các tham số đánh giá

4.1.1 Dice Coefficient

Dice Coefficient hay Dice similarity coefficient (DSC) là một phương pháp đánh giá sự tương đồng giữa hai tập hợp, phổ biến trong các bài toán đo lường hiệu suất với các mô hình phân đoạn hình ảnh. DSC được tính dựa trên hai tập hợp A và B. A là mặt nạ dự đoán còn B là nhãn (ground truth). Công thức của DSC như sau (4.1):

$$DSC = \frac{2 \cdot |A \cap B|}{|A| + |B|} \quad (4.1)$$

Trong đó: A là số phần tử trong tập hợp A (số lượng điểm dự đoán thuộc kết quả dự đoán của mô hình).

B là số phần tử trong tập hợp B (số lượng điểm thuộc ground truth).

$A \cap B$ là số phần tử chung giữa tập hợp A và tập hợp B (số lượng điểm dự đoán đúng).

4.1.2 Jaccard coefficient

Tương tự DSC, Jaccard coefficient (hay Intersection over Union - IOU) là một phương pháp đánh giá sự tương đồng giữa hai tập hợp. DSC được tính dựa trên tổng các phần tử của hai tập hợp, còn Jaccard coefficient được tính bằng hợp của hai tập hợp. Công thức của Jaccard coefficient như sau (4.2):

$$IOU = \frac{|A \cap B|}{|A \cup B|} \quad (4.2)$$

Trong đó:

A là số phần tử trong tập hợp A (số lượng điểm dự đoán thuộc kết quả dự đoán của mô hình).

B là số phần tử trong tập hợp B (số lượng điểm thuộc ground truth).

$A \cap B$ là số phần tử chung giữa tập hợp A và tập hợp B (số lượng điểm dự đoán đúng).

$A \cup B$ là số phần tử chung giữa tập hợp A và tập hợp B (số lượng điểm dự đoán đúng).

4.1.3 Hausdorff distance

Hausdorff distance là một phương pháp đo lường khoảng cách giữa hai tập hợp điểm trong không gian nhiều chiều, đặc biệt phổ biến trong lĩnh vực xử lý hình ảnh và đồ họa máy tính. Trong ngữ cảnh phân đoạn hình ảnh, Hausdorff distance được sử dụng để đánh giá độ tương đồng giữa hai phân đoạn hình ảnh, thường là phân đoạn dự đoán so với ground truth. Hausdorff distance giữa hai tập hợp A và B được tính bằng cách lấy khoảng cách xa nhất giữa các điểm trong tập hợp A và điểm gần nhất trong tập hợp B, sau đó lấy giá trị lớn nhất của các khoảng cách này (4.3):

$$d_H(A, B) = \max \left(\sup_{a \in A} \inf_{b \in B} d(a, b), \sup_{b \in B} \inf_{a \in A} d(b, a) \right) \quad (4.3)$$

Trong đó:

$d_H(A, B)$ là Hausdorff distance giữa hai tập hợp A và B.

$d(a, b)$ là khoảng cách giữa hai điểm a và b.

\sup là toán tử supremum (giá trị lớn nhất).

\inf là toán tử infimum (giá trị nhỏ nhất).

\max là toán tử lấy giá trị lớn nhất trong các giá trị bên trong ngoặc đơn.

Trong thử thách COVID-19-20, hai biến thể của Hausdorff distance được sử dụng là Hausdorff95 và Hausdorff100 lần lượt là giá trị khoảng cách tương ứng với phần trăm thứ 95% và 100% trong tập giá trị khoảng cách giữa hai mặt nạ. Hausdorff95 giúp quá trình đánh giá tập trung hơn đến những vùng quan trọng và ít nhạy cảm với nhiễu hơn.

4.1.4 Surface Dice

Surface Dice (còn gọi là Surface Dice Coefficient hoặc Surface Dice Overlap) là một phương pháp đo lường độ tương đồng giữa hai mặt nạ phân đoạn hình ảnh 3D dựa trên biên thay vì dựa vào vùng. Chỉ số này rất có ý nghĩa trong những bài toán phân đoạn ảnh y tế với biên của mục tiêu phân đoạn không rõ ràng. Công thức của Surface Dice như sau (4.4)

$$\text{SurfaceDice}(A, B) = \frac{2 \cdot |S_A \cap S_B|}{|S_A| + |S_B|} \quad (4.4)$$

Trong đó: $|S_A|$ là số lượng voxel trên biên giới hạn của phân đoạn A.

$|S_B|$ là số lượng voxel trên biên giới hạn của phân đoạn B.

$|S_A \cap S_B|$ là số lượng voxel chung trên biên giới hạn của phân đoạn A và phân đoạn B.

Thử thách COVID-19-20 sử dụng Surface Dice at 1mm là một biến thể của Surface Dice, chỉ số này mở rộng biên giới xem xét lên 1mm nhằm tăng tính ổn định và giảm sai số vì biên của GGOs không rõ ràng.

4.1.5 Sensitivity và specificity

Sensitivity và specificity là hai độ đo thường được sử dụng trong bài toán phân đoạn để đánh giá hiệu suất của mô hình. Sensitivity (độ nhạy, độ nhận dạng) hay còn gọi là true positive rate (TPR) đo lường các điểm positive được mô hình dự đoán đúng so với các điểm positive trong ground truth, nó đặc trưng cho khả năng nhận dạng đúng các điểm positive của mô hình. Công thức tính Sensitivity là:

$$Sensitivity = \frac{TP}{TP + FN} \quad (4.5)$$

Specificity (độ đặc hiệu) hay còn gọi là true negative rate (TNR) đo lường tỷ lệ các điểm thuộc vùng negative (nền) mà mô hình dự đoán đúng so với tổng số điểm negative trong ground truth. Nó cho biết khả năng của mô hình trong việc phát hiện các vùng negative. Công thức của Specificity là:

$$Specificity = \frac{TN}{TN + FP} \quad (4.6)$$

Trong 2 công thức trên:

TP (True Positive) là số lượng điểm positive được mô hình dự đoán đúng.

FN (False Negative) là số lượng điểm positive bị mô hình dự đoán sai

TN (True Negative) là số lượng điểm negative được mô hình dự đoán đúng.

FP (False Positive) là số lượng điểm negative bị mô hình dự đoán sai.

4.2 Thủ nghiệm mô hình trên các hàm lỗi

Tôi đã thử nghiệm ba hàm combo loss, dicefocal loss và diceboundary loss trên mô hình 2D, với mô hình 3D tôi sử dụng hai hàm lỗi combo loss, dicefocal loss. Các mô hình được huấn luyện với 5-fold cross validation sau 50 epoch, các mô hình nhỏ được kết hợp với nhau bằng phương pháp majority voting. Bảng sau là kết quả so sánh kết quả của các mô hình kết hợp từ 1 mô hình 2D và một mô hình 3D. Tôi thực hiện đánh giá các mô hình này trên bộ dữ liệu của Ma và các cộng sự

[11]. Bộ dữ liệu này bao gồm 20 hình ảnh CT có số lớp từ vài chục đến vài trăm.

| Mô hình | DSC | Hausdorff | Surface Dice | Sensitivity | Specificity |
|--------------------------------|--------|-----------|--------------|-------------|-------------|
| 2D Combo loss (2D CL) | 0.6528 | 64.107 | 0.5772 | 0.8401 | 0.9952 |
| 2D Dice Focal loss (2D DFL) | 0.6451 | 63.485 | 0.5701 | 0.8386 | 0.9951 |
| 2D Dice Boundary loss (2D DBL) | 0.6634 | 55.915 | 0.5933 | 0.8007 | 0.9963 |
| 3D Combo loss (3D CL) | 0.6058 | 58.455 | 0.5359 | 0.6875 | 0.9967 |
| 3D Dice Focal loss (3D DFL) | 0.6094 | 55.312 | 0.5394 | 0.6813 | 0.9968 |
| 2D CL + 3D CL | 0.6340 | 62.028 | 0.5729 | 0.8777 | 0.9943 |
| 2D CL + 3D DFL | 0.6327 | 61.157 | 0.5707 | 0.8836 | 0.9941 |
| 2D DFL + 3D CL | 0.6301 | 59.123 | 0.5679 | 0.8787 | 0.9941 |
| 2D DFL + 3D DFL | 0.6280 | 57.714 | 0.5652 | 0.8845 | 0.9939 |
| 2D DBL + 3D CL | 0.6431 | 54.289 | 0.5802 | 0.8638 | 0.9949 |
| 2D DBL + 3D DFL | 0.6416 | 54.382 | 0.5782 | 0.8697 | 0.9948 |

Bảng 4.1: Bảng so sánh sự kết hợp của các mô hình với các hàm lỗi khác nhau

Dựa vào bảng 4.1 ta có thể thấy được hiệu quả khác nhau của các hàm lỗi kết hợp trong bài toán này. Ở mô hình 2D, hai hàm lỗi CL và DFL đều là sự kết hợp của Dice loss và các biến thể của BCE cho thấy kết quả Sensutivity cao hơn mô hình 2D DBL cho thấy khi sử dụng các hàm lỗi này, việc phát hiện đủ các vùng tổn thương hiệu quả hơn sử dụng hàm lỗi DBL. Tuy nhiên các chỉ số DSC, Hausdorff distance và Surface Dice của mô hình 2D khi sử dụng Dice loss kết hợp Boundary loss cao hơn so với việc sử dụng hai hàm lỗi kết hợp còn lại cho thấy hiệu quả phát hiện chính xác biên của hàm lỗi Boundary loss là cao hơn so với Focal loss và Weighted Cross-entropy loss.

Với mô hình 3D, do mục đích của mô hình này hướng đến việc phát hiện các vùng tổn thương lớn và các vùng tổn thương quá nhỏ sẽ được bỏ qua nên kết quả của các chỉ số trong bảng là hợp lý. So sánh kết quả hai hàm lỗi được sử dụng với mô hình 3D không có quá nhiều khác biệt trong cả hai trường hợp.

Với 6 mô hình kết hợp, tôi nhận thấy ngoại trừ chỉ số DSC, tất cả các chỉ số còn lại đều có kết quả tốt hơn so trung bình của hai mô hình thành phần. Đặc biệt là chỉ số Sensitivity ở tất cả 6 trường hợp đều cao hơn từng mô hình thành phần, điều này cho thấy sự kết hợp của hai mô hình có khả năng phát hiện đầy đủ hơn những vùng tổn thương mà các mô hình 2D hoặc 3D không phát hiện được. Với các chỉ số liên quan đến biên và khoảng cách (Hausdorff và Surface Dice) hai mô hình kết hợp của 2D DBL có kết quả tốt hơn so với 4 mô hình còn lại, điều này một lần nữa khẳng định sự hiệu quả của hàm Boundary Loss trong việc xác định biên trong bài toán có biên khó phân biệt.

Các mô hình hoạt động khá tốt trên những hình ảnh có độ phân giải cao, tuy nhiên các hình ảnh có mức độ tổn thương rất nhỏ vẫn chưa đạt độ chính xác tốt.

4.3 Kết quả của phương pháp so với các mô hình phổ biến

Tôi thực hiện so sánh kết quả của phương pháp này với một số mô hình phổ biến trong phân đoạn hình ảnh y tế như 2D U-net, nnUnet 2D, nnUnet 3D, nnUnet 3D kết hợp với Feature Pyramid Net (FPN). Mô hình nnUnet-FPN được thiết kế bằng cách upsample các đầu ra của các lớp phía decoder của nnUnet, các lớp này được kết hợp với nhau bằng phép nối(concat). Kết quả so sánh này dựa trên kết quả huấn luyện các mô hình trên tập huấn luyện đã cho và kiểm thử trên trang chủ của thử thách COVID-19-20. Kết quả so sánh chi tiết ở bảng 4.2

| Phương pháp | DSC | Jardard | Hausdorff95 | Surface Dice | Sensitivity | Specificity |
|-----------------|---------------|---------------|---------------|---------------|---------------|---------------|
| Unet | 0.5546 | 0.4127 | 226,286 | 0.4607 | 0.6411 | 0.5689 |
| nnUnet 2D | 0.6520 | 0.5137 | 2069,907 | 0.6075 | 0.6477 | 0.7644 |
| nnUnet 3D | 0.6820 | 0.5426 | 99.92 | 0.6362 | 0.7041 | 0.7299 |
| nnUnet 3D + FPN | 0.7062 | 0.5812 | 83.378 | 0.6731 | 0.7145 | 0.7665 |
| Phương pháp này | 0.7522 | 0.6209 | 54.566 | 0.7193 | 0.7528 | 0.7749 |

Bảng 4.2: Bảng so sánh kết quả của phương pháp đề xuất với các phương pháp phân đoạn phổ biến

Phương pháp tôi đề xuất đã có kết quả rất tốt so với những phương pháp được nêu trên, với DSC cao hơn 5% so với mô hình kết hợp giữa nnUnet 3d và FPN. Chỉ số Hausdorff95 đạt 54.566 thấp hơn gần 35% so với mô hình có kết quả tốt nhất ở trên là nnUnet 3D + FPN với chỉ số Hausdorff95 là 83.378. Điều này cho thấy việc

tính toán hàm lỗi với trọng số gán theo thành phần liên thông đã có hiệu quả hơn so với hàm lỗi mặc định của nnUnet là hàm Combo loss. Hiệu quả của việc sử dụng thêm cơ chế chú ý và phương pháp chuẩn hóa mới cũng tỏ ra hiệu quả hơn so với phương pháp chuẩn hóa hàng loạt mặc định của nnUnet khi các chỉ số Sensitivity và Specificity cao hơn lần lượt 4,8% và 4,55% so với mô hình 3D nnUnet mặc định.

4.4 So sánh kết quả với các nhóm thực hiện trong thử thách COVID-19-20

Ở phần này, tôi thực hiện so sánh kết quả của mình với 8 trên top 10 nhóm đạt kết quả tốt nhất trong thời gian thử thách COVID 19-20 diễn ra. Các kết quả được tham khảo từ bài báo [10] và trang chủ của thử thách COVID-19-20. Các chỉ số chi tiết được thu thập tại bảng xếp hạng giai đoạn 1 của thử thách. Kết quả chi tiết có trong bảng 4.3

Dựa vào bảng kết quả có thể thấy được, dù điểm DSC chỉ đứng thứ 5 trong các kết quả được so sánh tuy nhiên chỉ số Hausdorff95 có kết quả tốt thứ 2 chỉ sau nhóm sử dụng phương pháp học bán giám sát sử dụng thêm dữ liệu ngoài, điều này cho thấy phương pháp của tôi đã phân đoạn khá chính xác các vùng nhiễm bệnh. Phương pháp của tôi có giá trị Sensitivity cao thứ 2 trong các phương pháp, điều này cho thấy phương pháp của tôi có tỉ lệ phân đoạn đúng cao hơn hầu hết các phương pháp trên.

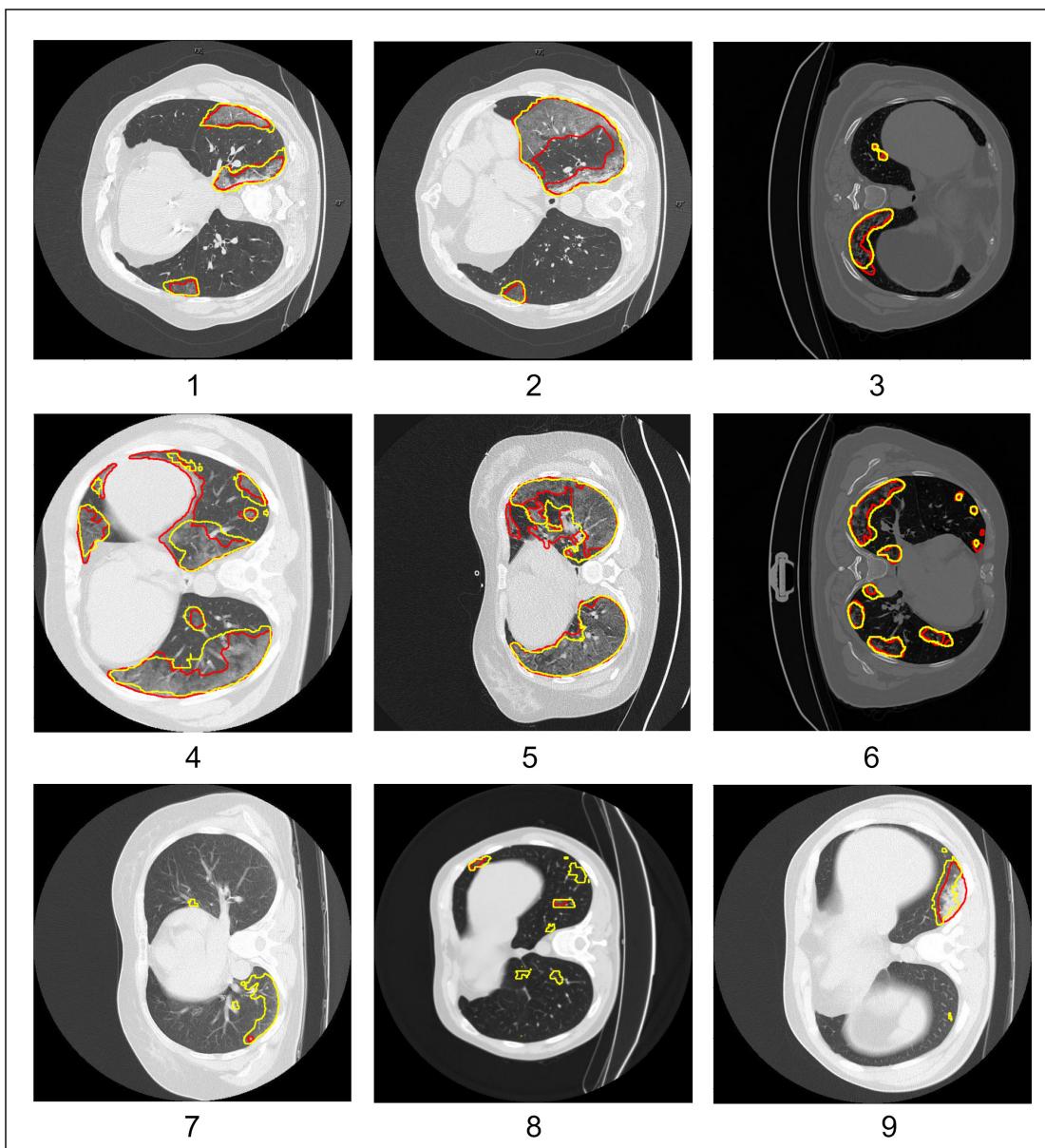
CHƯƠNG 4. ĐÁNH GIÁ THỰC NGHIỆM

| Nhóm | Phương pháp | Dữ liệu thêm | DSC | Hausdorff95 | Surface dice | Sensitivity | Specificity |
|---------------------------|--|-----------------|---------------|---------------|-----------------|---------------|---------------|
| S. Hu và công sự | Học bán giám sát | Có | 0.7606 | 51.533 | 0.7308 | 0.7267 | 0.8365 |
| C. Tang | nnUnet với các hàm loss kết hợp | Không | 0.7363 | 101.747 | 0.71139 | 0.7138 | 0.8409 |
| A. Husch và công sự | Kết hợp phân đoạn phổi và GGOS | Không | 0.7687 | 54.781 | 0.7457 | 0.7706 | 0.8171 |
| T. Zeng và công sự | Kết hợp 2D và 3D nnUnet | Không | 0.6213 | 414258 | 0.5731 | 0.6321 | 0.7100 |
| V. Liauchuk | Semi-3D CNN với MONAI | Không | 0.7035 | 2093.76 | 0.6735 | 0.6971 | 0.8063 |
| Z. Zhou và công sự | Kết hợp các mô hình 3D có độ phân giải khác nhau | Không | 0.7583 | 65.297 | 0.7352 | 0.7373 | 0.8365 |
| J. H. Moltz và công sự | nnUnet 3D | Không | 0.7089 | 63.067 | 0.6861 | 0.6684 | 0.8348 |
| B. Oliveira và công sự | nnUnet phân tầng | Không | 0.7573 | 78.335 | 0.7328 | 0.7521 | 0.8196 |
| Phương pháp này | | Không | 0.7522 | 54.566 | 0.7193 | 0.7528 | 0.7749 |

Bảng 4.3: Bảng so sánh kết quả của phương pháp đề xuất với các thuật toán hàng đầu trong thử thách COVID-19-20

4.5 Kết quả dự đoán mô hình trên một số hình ảnh chụp CT

Một số kết quả phân đoạn GGOs trên bộ dữ liệu thử nghiệm.



Hình 4.1: Một số kết quả phân đoạn: Viền màu đỏ là ground truth, viền màu vàng là kết quả dự đoán.

Hình 1 đến hình 3 là các hình ảnh có tổn thương GGOs dễ phân đoạn.

Hình 4 đến hình 6 là các hình ảnh có tình trạng bệnh ở phổi khá phức tạp, tổn thương trên diện rộng và nhiều vùng tổn thương khác nhau.

Hình 7 đến hình 9 là những trường hợp khó phân đoạn, GGOs quá nhỏ hoặc độ tương phản so với mô xung quanh quá thấp.

Theo như kết quả có được trong hình ảnh 4.1. Mô hình đề xuất trong ĐATN này đã có kết quả phân đoạn rất tốt trên hình ảnh từ 1 đến 3, đây là những hình ảnh có tổn thương khá rõ ràng và có độ tương phản cao so với các mô xung quanh. Đối với các hình từ 4 đến 6, mức độ tổn thương trên phổi là khá nhiều và có nhiều ổ tổn thương, mô hình đã phân đoạn khá chính xác và đầy đủ trên các hình này. Các hình 7, 8, 9 là những hình có độ chính xác thấp nhất. Với hình 7, vùng tổn thương là rất nhỏ và có độ sáng rất gần với các vùng xung quanh. Tương tự với hình 8, dù đã

phát hiện chính xác vùng tổn thương tuy nhiên lại xuất hiện khá nhiều nhiễu, đây là kết quả đến từ việc kết hợp 2 mô hình 2D và 3D chưa đạt được tối ưu. Đối với hình 9, vùng tổn thương rất sáng và có kích thước lớn tuy nhiên vùng tổn thương nằm ở ngoại vi phổi và có độ tương phản rất thấp so với mô bên phải, vì thế mô hình vẫn chưa thể phân đoạn chính xác được vùng biên này.

4.6 Ứng dụng thử nghiệm

Tôi sử dụng framework Flask tạo một website đơn giản để áp dụng phương pháp đã đề xuất. Website có chức năng tải hình ảnh lên và lựa chọn phương pháp phân đoạn là sử dụng ensemble hay không. Tôi thiết kế một giao diện hiển thị hình ảnh 3D sử dụng nút bấm hoặc lăn chuột để xem hình ảnh. Có hai hình ảnh được hiển thị bao gồm lát ngắt ngang qua phổi và lát cắt theo hình chiết vuông góc với cơ thể.

Giao diện và kết quả của website có trong hình 4.2.

Thời gian xử lý hình ảnh trung bình khi sử dụng Ensemble Learning là 100s/50 layer và khi không sử dụng Ensemble Learning là 75s/50 layer. Chương trình được chạy trên thiết bị có thông số RAM 16GB, GPU NVIDIA GeForce 1650 4GB.



Hình 4.2: Giao diện Website ứng dụng

CHƯƠNG 5. KẾT LUẬN

5.1 Kết luận

Như vậy trong ĐATN này, tôi đã đề xuất một phương pháp phân đoạn hình ảnh bóng mờ trong hình chụp CT phổi bệnh nhân sau khi nhiễm Covid-19. Tôi đã phát triển quy trình học máy dựa trên cơ chế của mô hình nnUnet, đề xuất một phương pháp chuẩn hóa hình ảnh để giải quyết vấn đề tình trạng hình ảnh khác nhau bằng cách kết hợp với mô hình phân đoạn phổi. Hai mô hình phân đoạn 2D và 3D Unet được thiết kế dựa trên mô hình Unet cơ bản kết hợp với cơ chế Attention. Đồng thời với việc kết hợp nhiều hàm lỗi khác nhau, kết quả của phương pháp đã cải thiện được vấn đề mất cân bằng dữ liệu của bài toán, tuy nhiên với trường hợp tổn thương quá nhỏ, mô hình vẫn chưa phân đoạn chính xác. Việc sử dụng hàm lỗi Boundary loss đã cho thấy phương pháp này có khả năng phân biệt biên tốt hơn so với các phương pháp trước đây tuy nhiên vẫn đề phân biệt chính xác biên trong bài toán GGOs vẫn còn là vấn đề mà phương pháp này chưa thể giải quyết triệt để.

5.2 Hướng phát triển trong tương lai

Phương pháp kết hợp hai mô hình 2D và 3D vẫn chưa tối ưu, trong tương lai có thể sử dụng áp dụng các phương pháp xử lý hình ảnh để kết hợp hai mô hình. Do ĐATN này được thực hiện trên cơ sở điều kiện phần cứng có hạn và dữ liệu huấn luyện không quá nhiều, phương pháp có thể cải thiện độ chính xác nếu được huấn luyện với nhiều mô hình và kiến trúc mô hình lớn hơn.

TÀI LIỆU THAM KHẢO

- [1] H. R. Roth, Z. Xu, C. Tor-Díez **and others**, “Rapid artificial intelligence solutions in a pandemic—the covid-19-20 lung ct lesion segmentation challenge,” *Medical image analysis*, **jourvol** 82, **page** 102 605, 2022.
- [2] F. Isensee, J. Petersen, A. Klein **and others**, “Nnu-net: Self-adapting framework for u-net-based medical image segmentation,” *ArXiv*, **jourvol** abs/1809.10486, 2018.
- [3] D. L. Fung, Q. Liu, J. Zammit, C. K.-S. Leung **and** P. Hu, “Self-supervised deep learning model for covid-19 lung ct image segmentation highlighting putative causal relationship among age, underlying disease and covid-19,” *Journal of Translational Medicine*, **jourvol** 19, **number** 1, 2021.
- [4] D.-P. Fan, T. Zhou, G.-P. Ji **and others**, “Inf-net: Automatic covid-19 lung infection segmentation from ct images,” *IEEE Transactions on Medical Imaging*, **jourvol** 39, **number** 8, **pages** 2626–2637, 2020.
- [5] D. Müller, I. Soto-Rey **and** F. Kramer, “Robust chest ct image segmentation of covid-19 lung infection based on limited data,” *Informatics in Medicine Unlocked*, **jourvol** 25, **page** 100 681, 2021.
- [6] D. Müller **and** F. Kramer, “Miscnn: A framework for medical image segmentation with convolutional neural networks and deep learning,” *BMC Medical Imaging*, **jourvol** 21, **number** 1, 2021.
- [7] T. Ben-Haim, R. M. Sofer, G. Ben-Arie, I. Shelef **and** T. R. Raviv, “A deep ensemble learning approach to lung ct segmentation for covid-19 severity assessment,” **in**2022 *IEEE International Conference on Image Processing (ICIP)* 2022, **pages** 151–155.
- [8] Y. Rao, Q. Lv, S. Zeng **and others**, “Covid-19 ct ground-glass opacity segmentation based on attention mechanism threshold,” *Biomedical Signal Processing and Control*, **jourvol** 81, **page** 104 486, 2023.
- [9] *Covid-19 lung ct lesion segmentation challenge - 2020*. **url:** <https://covid-segmentation.grand-challenge.org/>.
- [10] H. Roth, Z. Xu, C. T. Diez **and others**, *Rapid Artificial Intelligence Solutions in a pandemic - the COVID-19-20 lung CT lesion segmentation challenge*, 2021. DOI: 10.21203/rs.3.rs-571332/v1.
- [11] M. Jun, G. Cheng, W. Yixin **and others**, *Covid-19 ct lung and infection segmentation dataset*, 2020. **url:** <https://zenodo.org/record/3757476>.

- [12] O. Ronneberger, P. Fischer **and** T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” *Lecture Notes in Computer Science*, 234–241, 2015. DOI: 10.1007/978-3-319-24574-4_28.
- [13] K. Simonyan **and** A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv 1409.1556*, september 2014.
- [14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li **and** L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
- [15] I. Ahmed, A. Chehri **and** G. Jeon, “A sustainable deep learning-based framework for automated segmentation of covid-19 infected regions: Using u-net with an attention mechanism and boundary loss function,” *Electronics*, **jourvol** 11, **number** 15, **page** 2296, 2022. DOI: 10.3390/electronics11152296.
- [16] H. Kervadec, J. Bouchtiba, C. Desrosiers, E. Granger, J. Dolz **and** I. Ben Ayed, “Boundary loss for highly unbalanced segmentation,” *Medical Image Analysis*, **jourvol** 67, **page** 101851, **january** 2021. DOI: 10.1016/j.media.2020.101851.
- [17] **url:** <https://doi.org/10.7937/tcia.2020.gqry-nc81>.
- [18] T.-Y. Lin, P. Goyal, R. Girshick, K. He **and** P. Dollar, “Focal loss for dense object detection,” *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.