

**ĐẠI HỌC ĐÀ NẴNG**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**



**BÁO CÁO CUỐI KỲ**  
**CHUYÊN ĐỀ 2**

**Đề tài:**

**PHÂN VÙNG DA UNG THƯ**  
**SỬ DỤNG MÔ HÌNH UNET**

**Nhóm:** 22

**Thành viên:**

1. Tô Ngọc Hoan - [106210213]
2. Nguyễn Ngọc Hưng - [106210216]

**Giảng viên hướng dẫn:** Nguyễn Văn Hiếu

Đà Nẵng, tháng 12 năm 2025

# Mục lục

<b>1</b>	<b>PHÂN CÔNG CÔNG VIỆC TRONG NHÓM</b>	<b>3</b>
<b>2</b>	<b>MÔ TẢ CÁC NỘI DUNG CHÍNH CỦA DỰ ÁN</b>	<b>4</b>
2.1	Yêu cầu bài toán và tính cấp thiết . . . . .	4
2.2	Mô hình hệ thống và cách thức hoạt động . . . . .	4
<b>3</b>	<b>MÔ TẢ VÀ GIẢI QUYẾT BÀI TOÁN</b>	<b>6</b>
3.1	Mô hình hóa bài toán . . . . .	6
3.2	Kiến trúc mô hình UNet . . . . .	7
3.3	Phương pháp huấn luyện và tối ưu hóa . . . . .	8
3.4	Đánh giá độ phức tạp và tốc độ hội tụ . . . . .	8
<b>4</b>	<b>PHÂN TÍCH KẾT QUẢ</b>	<b>10</b>
4.1	Các metrics đánh giá . . . . .	10
4.2	Mô tả các file kết quả . . . . .	10
4.3	Phân tích kết quả huấn luyện . . . . .	11
4.4	Phân tích kết quả dự đoán . . . . .	12
4.5	So sánh và đánh giá . . . . .	13
<b>5</b>	<b>KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN</b>	<b>15</b>
5.1	Kết luận . . . . .	15
5.2	Hướng phát triển . . . . .	15
	<b>PHỤ LỤC</b>	<b>17</b>
A.	Code chính . . . . .	17
B.	Cấu hình hệ thống . . . . .	17
C.	Thông số huấn luyện . . . . .	17

# Danh sách hình vẽ

1	Kiến trúc mô hình UNet . . . . .	7
2	Biểu đồ quá trình huấn luyện . . . . .	12
3	Kết quả dự đoán trên tập validation . . . . .	13

# 1 PHÂN CÔNG CÔNG VIỆC TRONG NHÓM

Nhóm chúng em gồm 2 thành viên với sự phân công công việc như sau:

**Thành viên 1: Tô Ngọc Hoan - 50%**

- Thu thập và tiền xử lý dữ liệu
- Xây dựng kiến trúc mô hình UNet
- Thiết lập các hàm loss và metrics đánh giá
- Viết báo cáo phần lý thuyết và mô hình hóa bài toán

**Thành viên 2: Nguyễn Ngọc Hưng - 50%**

- Thực hiện huấn luyện và tối ưu hóa mô hình
- Đánh giá và phân tích kết quả
- Trực quan hóa kết quả dự đoán
- Viết báo cáo phần thực nghiệm và phân tích kết quả

## 2 MÔ TẢ CÁC NỘI DUNG CHÍNH CỦA DỰ ÁN

### 2.1 Yêu cầu bài toán và tính cấp thiết

Ung thư da là một trong những loại ung thư phổ biến nhất trên thế giới với tỷ lệ mắc mới ngày càng gia tăng. Theo thống kê y tế, việc phát hiện sớm các tổn thương da bất thường có thể nâng cao tỷ lệ chữa khỏi lên tới 95%. Tuy nhiên, việc chẩn đoán chính xác các vùng da bị ảnh hưởng đòi hỏi sự chuyên môn cao từ các bác sĩ da liễu và tốn nhiều thời gian.

Trong bối cảnh đó, việc phát triển một hệ thống tự động hỗ trợ phân vùng chính xác các khu vực da bị tổn thương trở nên vô cùng cấp thiết. Hệ thống này không chỉ giúp giảm tải công việc cho các bác sĩ mà còn có thể được triển khai ở những vùng thiếu chuyên gia y tế, từ đó góp phần nâng cao khả năng tiếp cận dịch vụ chẩn đoán chất lượng cao cho cộng đồng. Hơn nữa, một hệ thống phân vùng chính xác sẽ hỗ trợ tốt hơn trong việc lập kế hoạch điều trị, theo dõi diễn biến bệnh và đánh giá hiệu quả của các phương pháp can thiệp.

### 2.2 Mô hình hệ thống và cách thức hoạt động

Hệ thống phân vùng da ung thư của chúng em được xây dựng dựa trên kiến trúc UNet, một mô hình học sâu đặc biệt hiệu quả cho các bài toán phân vùng ảnh y tế. Quy trình hoạt động của hệ thống được chia thành các giai đoạn chính như sau:

Đầu tiên, dữ liệu ảnh da được thu thập và chuẩn hóa thông qua các bước tiền xử lý. Mỗi ảnh đầu vào được chuyển đổi thành tensor và chuẩn hóa theo các giá trị trung bình và độ lệch chuẩn của tập dữ liệu ImageNet. Việc chuẩn hóa này giúp mô hình hội tụ nhanh hơn và ổn định hơn trong quá trình huấn luyện. Tập dữ liệu của chúng em gồm 200 cặp ảnh và mask tương ứng, được chia thành tập huấn luyện (80%) và tập kiểm tra (20%).

Tiếp theo, kiến trúc UNet được thiết kế với hai nhánh chính là encoder và decoder. Encoder có nhiệm vụ trích xuất các đặc trưng từ ảnh đầu vào thông qua các lớp tích chập và pooling, giúp nắm bắt thông tin từ mức độ chi tiết đến ngữ cảnh tổng thể. Decoder thực hiện quá trình ngược lại, dần dần khôi phục kích thước không gian của ảnh và tạo ra mask phân vùng cuối cùng. Điểm đặc biệt của UNet là các kết nối skip connection giữa encoder và decoder, cho phép thông tin chi tiết ở các tầng nông được truyền trực tiếp đến decoder, giúp cải thiện độ chính xác của việc phân vùng ranh giới.

Trong quá trình huấn luyện, mô hình sử dụng hàm loss kết hợp giữa Binary Cross Entropy và Dice Loss. Binary Cross Entropy đánh giá sai số ở từng pixel, trong khi Dice Loss tập trung vào độ chồng lấp giữa vùng dự đoán và vùng thực tế. Sự kết hợp này giúp mô hình vừa học được chi tiết từng pixel, vừa đảm bảo tính toàn vẹn của vùng phân đoạn. Chúng em sử dụng thuật toán tối ưu Adam với learning rate ban đầu là 0.0001, kèm theo cơ chế ReduceLROnPlateau để tự động giảm learning rate khi validation loss

không còn cải thiện.

Sau khi huấn luyện, mô hình có khả năng nhận đầu vào là một ảnh da bất kỳ và tự động sinh ra mask phân vùng, trong đó các pixel thuộc vùng tổn thương được đánh dấu khác biệt với vùng da bình thường. Hệ thống cũng được tích hợp cơ chế early stopping để ngăn chặn overfitting, tự động dừng huấn luyện khi validation loss không cải thiện sau 10 epochs liên tiếp.

## 3 MÔ TẢ VÀ GIẢI QUYẾT BÀI TOÁN

### 3.1 Mô hình hóa bài toán

Bài toán phân vùng da ung thư có thể được mô hình hóa như một bài toán phân loại nhị phân ở mức pixel. Cho trước một ảnh đầu vào  $\mathbf{X}$  với kích thước  $H \times W \times 3$  (chiều cao, chiều rộng và 3 kênh màu RGB), mục tiêu của chúng ta là tìm một hàm ánh xạ  $f$  sao cho:

$$\mathbf{Y} = f(\mathbf{X}; \theta) \quad (1)$$

trong đó  $\mathbf{Y}$  là mask đầu ra có kích thước  $H \times W \times 1$ , với mỗi pixel có giá trị trong khoảng  $[0, 1]$  thể hiện xác suất pixel đó thuộc vùng tổn thương, và  $\theta$  là tập các tham số của mô hình cần học.

Quá trình học của mô hình được thực hiện thông qua việc tối thiểu hóa hàm loss trên tập dữ liệu huấn luyện. Chúng em sử dụng hàm loss kết hợp DiceBCE được định nghĩa như sau:

$$\mathcal{L}_{total} = \mathcal{L}_{BCE} + \mathcal{L}_{Dice} \quad (2)$$

Trong đó, Binary Cross Entropy Loss được tính theo công thức:

$$\mathcal{L}_{BCE} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (3)$$

với  $N$  là tổng số pixel,  $y_i$  là nhãn thực tế của pixel thứ  $i$ , và  $\hat{y}_i$  là xác suất dự đoán sau khi áp dụng hàm sigmoid.

Dice Loss được tính dựa trên Dice coefficient, một metric đo độ tương đồng giữa hai tập hợp:

$$\mathcal{L}_{Dice} = 1 - \frac{2|\mathbf{Y} \cap \hat{\mathbf{Y}}| + \epsilon}{|\mathbf{Y}| + |\hat{\mathbf{Y}}| + \epsilon} \quad (4)$$

trong đó  $|\mathbf{Y} \cap \hat{\mathbf{Y}}|$  là giao của tập pixel thực tế và dự đoán,  $|\mathbf{Y}|$  và  $|\hat{\mathbf{Y}}|$  là tổng số pixel của mỗi tập, và  $\epsilon$  là một hằng số nhỏ để tránh chia cho 0.

Quá trình tối ưu sử dụng thuật toán Adam để cập nhật tham số theo quy tắc:

$$\theta_{t+1} = \theta_t - \alpha \frac{m_t}{\sqrt{v_t} + \epsilon} \quad (5)$$

với  $m_t$  và  $v_t$  lần lượt là moment bậc nhất và bậc hai của gradient,  $\alpha$  là learning rate, và  $\epsilon$  là hằng số nhỏ để đảm bảo tính ổn định số học.

### 3.2 Kiến trúc mô hình UNet

Kiến trúc UNet của chúng em được xây dựng với cấu trúc đối xứng gồm encoder, bottleneck và decoder. Encoder bao gồm 4 khối DoubleConv với số kênh lần lượt là 64, 128, 256 và 512. Mỗi khối DoubleConv thực hiện hai phép tích chập liên tiếp với kernel  $3 \times 3$ , padding 1, theo sau bởi Batch Normalization và hàm kích hoạt ReLU. Sau mỗi khối DoubleConv, một lớp MaxPooling  $2 \times 2$  được áp dụng để giảm kích thước không gian xuống một nửa và tăng receptive field.

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 64, 224, 224]	1,792
BatchNorm2d-2	[-1, 64, 224, 224]	128
ReLU-3	[-1, 64, 224, 224]	0
Conv2d-4	[-1, 64, 224, 224]	36,928
BatchNorm2d-5	[-1, 64, 224, 224]	128
ReLU-6	[-1, 64, 224, 224]	0
DoubleConv-7	[-1, 64, 224, 224]	0
MaxPool2d-8	[-1, 64, 112, 112]	0
Conv2d-9	[-1, 128, 112, 112]	73,856
BatchNorm2d-10	[-1, 128, 112, 112]	256
ReLU-11	[-1, 128, 112, 112]	0
Conv2d-12	[-1, 128, 112, 112]	147,584
BatchNorm2d-13	[-1, 128, 112, 112]	256
ReLU-14	[-1, 128, 112, 112]	0
DoubleConv-15	[-1, 128, 112, 112]	0
MaxPool2d-16	[-1, 128, 56, 56]	0
Conv2d-17	[-1, 256, 56, 56]	295,168
BatchNorm2d-18	[-1, 256, 56, 56]	512
ReLU-19	[-1, 256, 56, 56]	0
Conv2d-20	[-1, 256, 56, 56]	590,080
BatchNorm2d-21	[-1, 256, 56, 56]	512
ReLU-22	[-1, 256, 56, 56]	0
DoubleConv-23	[-1, 256, 56, 56]	0
MaxPool2d-24	[-1, 256, 28, 28]	0
Conv2d-25	[-1, 512, 28, 28]	1,180,160
BatchNorm2d-26	[-1, 512, 28, 28]	1,024
ReLU-27	[-1, 512, 28, 28]	0
Conv2d-28	[-1, 512, 28, 28]	2,359,808
BatchNorm2d-29	[-1, 512, 28, 28]	1,024
ReLU-30	[-1, 512, 28, 28]	0
DoubleConv-31	[-1, 512, 28, 28]	0
MaxPool2d-32	[-1, 512, 14, 14]	0
Conv2d-33	[-1, 1024, 14, 14]	4,719,616
BatchNorm2d-34	[-1, 1024, 14, 14]	2,048
ReLU-35	[-1, 1024, 14, 14]	0
Conv2d-36	[-1, 1024, 14, 14]	9,438,208
BatchNorm2d-37	[-1, 1024, 14, 14]	2,048
ReLU-38	[-1, 1024, 14, 14]	0
DoubleConv-39	[-1, 1024, 14, 14]	0

Hình 1: Kiến trúc mô hình UNet

Phần bottleneck nằm ở giữa kiến trúc với 1024 kênh, có vai trò nắm bắt các đặc trưng ngữ cảnh cấp cao nhất của ảnh. Đây là nơi thông tin được nén tối đa về mặt không gian nhưng phong phú nhất về mặt ngữ nghĩa.

Decoder được xây dựng đối xứng với encoder, sử dụng các lớp ConvTranspose2d để



tăng dần kích thước không gian. Tại mỗi bước upsampling, đầu ra của decoder được nối với feature map tương ứng từ encoder thông qua skip connection. Cụ thể, nếu gọi  $\mathbf{E}_i$  là output của encoder ở tầng  $i$  và  $\mathbf{D}_i$  là output của decoder ở tầng tương ứng, thì input của khối DoubleConv tiếp theo trong decoder sẽ là:

$$\text{Input}_{next} = \text{Concat}(\mathbf{E}_i, \mathbf{D}_i) \quad (6)$$

Cơ chế skip connection này rất quan trọng vì nó cho phép decoder truy cập trực tiếp vào các thông tin chi tiết ở các tầng nông của encoder, giúp khôi phục ranh giới phân đoạn một cách chính xác. Cuối cùng, một lớp tích chập  $1 \times 1$  được sử dụng để chuyển đổi số kênh cuối cùng về 1, tạo ra mask đầu ra.

### 3.3 Phương pháp huấn luyện và tối ưu hóa

Quá trình huấn luyện được thực hiện với batch size là 8 trên GPU CUDA. Chúng em sử dụng optimizer Adam với learning rate khởi tạo là 0.0001, đây là một giá trị learning rate tương đối nhỏ phù hợp với việc huấn luyện các mô hình phân vùng trên tập dữ liệu nhỏ.

Để tránh overfitting và tăng khả năng tổng quát hóa của mô hình, chúng em áp dụng một số kỹ thuật regularization. Đầu tiên là Batch Normalization được tích hợp trong mỗi khối DoubleConv, giúp chuẩn hóa phân phối của activations và làm cho quá trình huấn luyện ổn định hơn. Thứ hai, chúng em sử dụng learning rate scheduler ReduceLROnPlateau với factor 0.1 và patience 5, nghĩa là nếu validation loss không giảm sau 5 epochs, learning rate sẽ được nhân với 0.1. Điều này giúp mô hình tinh chỉnh tốt hơn ở giai đoạn cuối của quá trình huấn luyện.

Cơ chế early stopping cũng được áp dụng với patience là 10 epochs. Mô hình sẽ tự động dừng huấn luyện nếu validation loss không cải thiện sau 10 epochs liên tiếp, và trạng thái mô hình tốt nhất (có validation loss thấp nhất) sẽ được lưu lại. Cách tiếp cận này đảm bảo chúng ta luôn có được mô hình tổng quát tốt nhất mà không cần phải huấn luyện hết số epochs đã định trước.

### 3.4 Đánh giá độ phức tạp và tốc độ hội tụ

Về mặt độ phức tạp tính toán, kiến trúc UNet của chúng em có khoảng 31 triệu tham số cần học. Độ phức tạp về thời gian cho một lần forward pass phụ thuộc vào kích thước ảnh đầu vào. Với ảnh kích thước  $224 \times 224$ , mỗi batch gồm 8 ảnh có thời gian xử lý khoảng 0.4 giây trên GPU CUDA. Độ phức tạp về không gian bộ nhớ chủ yếu tập trung ở phần bottleneck và các skip connections, nơi cần lưu trữ các feature maps kích thước lớn.

Về tốc độ hội tụ, chúng em quan sát thấy mô hình hội tụ khá nhanh trong 20-30 epochs đầu tiên, với validation loss giảm đáng kể. Sau đó, quá trình cải thiện chậm lại và mô hình đạt được kết quả tốt nhất ở epoch 52. Việc sử dụng Adam optimizer cùng với learning rate scheduler đã giúp quá trình huấn luyện ổn định và tránh được hiện tượng dao động mạnh của loss. Cơ chế early stopping kích hoạt sau epoch 62, cho thấy mô hình

đã đạt được trạng thái tối ưu và không thể cải thiện thêm trên tập validation với cấu hình hiện tại.

## 4 PHÂN TÍCH KẾT QUẢ

### 4.1 Các metrics đánh giá

Để đánh giá hiệu suất của mô hình phân vùng, chúng em sử dụng bốn metrics chính bao gồm Loss, Pixel Accuracy, Dice Coefficient và Intersection over Union (IoU). Mỗi metric cung cấp một góc nhìn khác nhau về chất lượng của việc phân vùng.

**Loss** là tổng hợp của Binary Cross Entropy và Dice Loss như đã trình bày ở phần trước. Đây là chỉ số được sử dụng trực tiếp trong quá trình tối ưu và phản ánh mức độ sai khác tổng thể giữa dự đoán và ground truth. Loss càng thấp cho thấy mô hình dự đoán càng gần với nhãn thực tế.

**Pixel Accuracy** đo tỷ lệ các pixel được phân loại đúng trên tổng số pixel:

$$\text{Pixel Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

trong đó  $TP$  là số pixel dương tính đúng,  $TN$  là pixel âm tính đúng,  $FP$  là pixel dương tính sai và  $FN$  là pixel âm tính sai. Tuy nhiên, metric này có thể không phản ánh đầy đủ chất lượng phân vùng khi tập dữ liệu mất cân bằng giữa vùng tổn thương và vùng da bình thường.

**Dice Coefficient** là một metric đặc biệt quan trọng trong các bài toán phân vùng y tế:

$$\text{Dice} = \frac{2|\mathbf{Y} \cap \hat{\mathbf{Y}}|}{|\mathbf{Y}| + |\hat{\mathbf{Y}}|} \quad (8)$$

Dice coefficient có giá trị từ 0 đến 1, với 1 là phân vùng hoàn hảo. Metric này nhạy cảm với cả false positives và false negatives, đồng thời ít bị ảnh hưởng bởi sự mất cân bằng lớp so với Pixel Accuracy.

**Intersection over Union (IoU)**, còn được gọi là Jaccard Index, đo độ chồng lấp giữa vùng dự đoán và vùng thực tế:

$$\text{IoU} = \frac{|\mathbf{Y} \cap \hat{\mathbf{Y}}|}{|\mathbf{Y} \cup \hat{\mathbf{Y}}|} \quad (9)$$

IoU thường cho giá trị thấp hơn Dice coefficient nhưng là một metric nghiêm ngặt hơn. Trong y tế,  $\text{IoU} > 0.5$  thường được coi là phân vùng chấp nhận được, và  $\text{IoU} > 0.7$  là phân vùng tốt.

### 4.2 Mô tả các file kết quả

Quá trình thực nghiệm tạo ra một số file quan trọng lưu trữ thông tin về mô hình và kết quả huấn luyện. Phần huấn luyện trong code chứa trạng thái tham số của mô hình tại epoch có validation loss thấp nhất, cụ thể là epoch 52 trong thí nghiệm của chúng

em. File này có kích thước khoảng 120MB, lưu trữ toàn bộ 31 triệu tham số của mạng UNet dưới dạng state dictionary của PyTorch. Việc lưu mô hình tốt nhất theo validation loss giúp đảm bảo chúng ta có được phiên bản tổng quát tốt nhất của mô hình, tránh overfitting.

File `training_results.png` là biểu đồ trực quan hóa quá trình huấn luyện qua các epochs, bao gồm 4 subplot thể hiện sự biến thiên của Loss, Accuracy, Dice coefficient và IoU trên cả tập train và validation. Biểu đồ này cho phép chúng ta quan sát xu hướng học của mô hình, phát hiện các dấu hiệu overfitting hoặc underfitting, và đánh giá hiệu quả của các kỹ thuật regularization.

File `prediction_results.png` chứa hình ảnh so sánh giữa ảnh gốc, mask thực tế và mask dự đoán trên 10 mẫu ngẫu nhiên từ tập validation. File này đóng vai trò quan trọng trong việc đánh giá định tính chất lượng phân vùng của mô hình, cho phép chúng ta nhìn thấy trực quan độ chính xác của việc phân vùng ranh giới và khả năng xử lý các trường hợp khó.

### 4.3 Phân tích kết quả huấn luyện

Kết quả huấn luyện cho thấy mô hình đã học được rất tốt và đạt hiệu suất cao trên cả tập train và validation. Ở epoch tốt nhất (epoch 52), mô hình đạt train loss 0.1643 và validation loss 0.2020. Mặc dù validation loss cao hơn train loss một chút, nhưng khoảng cách này không quá lớn, cho thấy mô hình không bị overfitting nghiêm trọng. Điều này phần nào nhờ vào các kỹ thuật regularization như Batch Normalization, learning rate scheduling và early stopping mà chúng em đã áp dụng.

Bảng 1: Kết quả huấn luyện tại epoch tốt nhất (epoch 52)

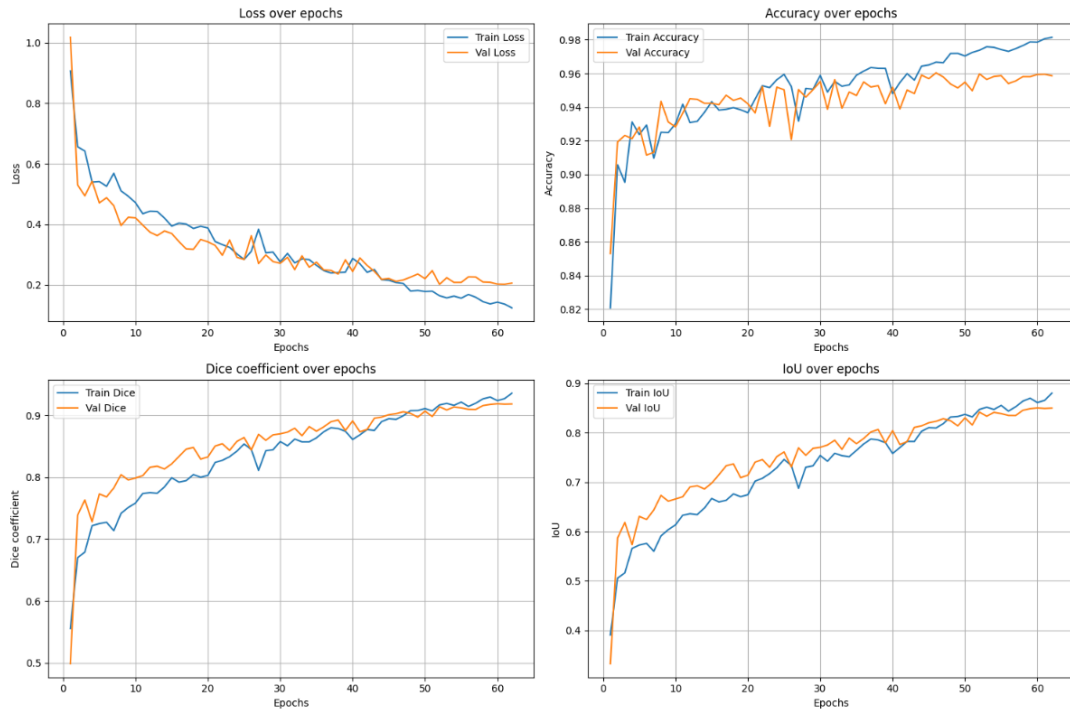
Metric	Train	Validation
Loss	0.1643	0.2020
Pixel Accuracy	0.9738	0.9597
Dice Coefficient	0.9169	0.9138
IoU Score	0.8472	0.8416

Về Pixel Accuracy, mô hình đạt 97.38% trên tập train và 95.97% trên tập validation. Đây là con số rất ấn tượng, cho thấy phần lớn các pixel đều được phân loại chính xác. Tuy nhiên, chúng ta cần lưu ý rằng trong bài toán phân vùng da, vùng da bình thường thường chiếm đa số diện tích ảnh, do đó Pixel Accuracy có thể không phản ánh đầy đủ khả năng phân vùng chính xác vùng tổn thương.

Dice coefficient là metric quan trọng hơn trong ngữ cảnh này. Mô hình đạt Dice coefficient 0.9169 trên tập train và 0.9138 trên tập validation. Giá trị Dice  $> 0.9$  được coi là rất tốt trong các bài toán phân vùng y tế. Sự khác biệt rất nhỏ giữa train và validation Dice (chỉ khoảng 0.003) càng khẳng định mô hình có khả năng tổng quát hóa tốt.

IoU score đạt 0.8472 trên tập train và 0.8416 trên tập validation. IoU thường cho giá trị thấp hơn Dice coefficient do công thức tính toán nghiêm ngặt hơn. Trong lĩnh vực

phân vùng y tế,  $\text{IoU} > 0.8$  được coi là kết quả xuất sắc, cho thấy độ chồng lấp giữa vùng dự đoán và vùng thực tế rất cao.

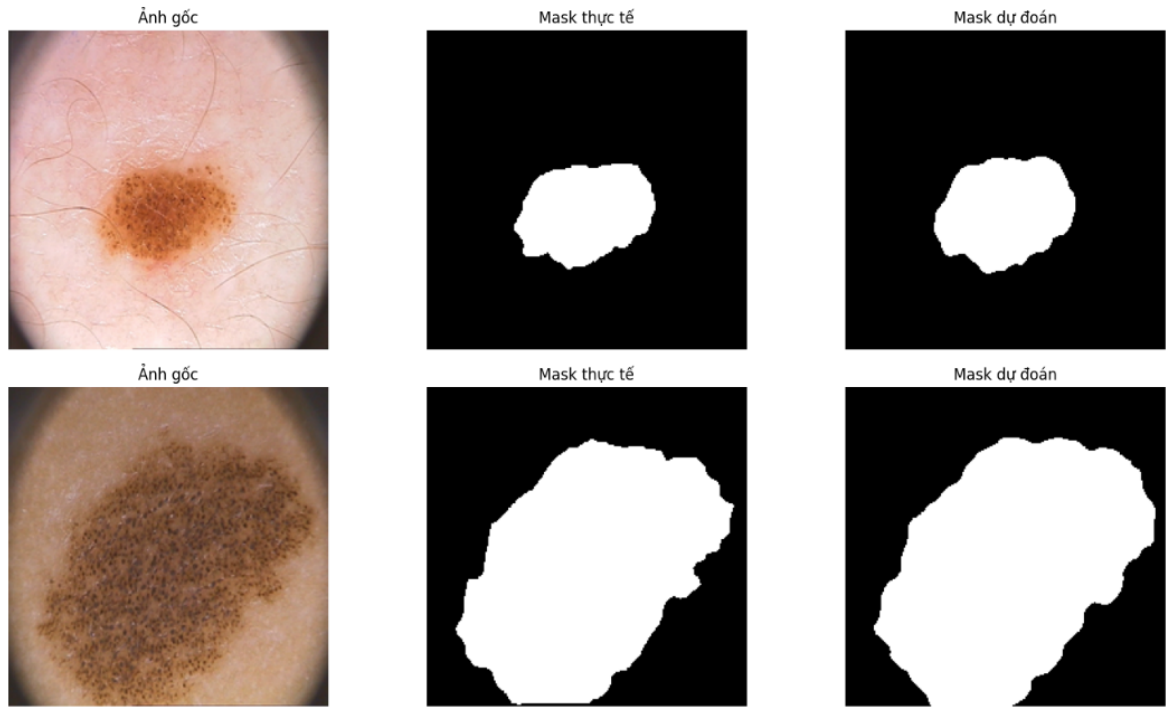


Hình 2: Biểu đồ quá trình huấn luyện

Quan sát đồ thị training\_results.png, chúng em nhận thấy các đường loss giảm đều đặn trong giai đoạn đầu và dần ổn định sau epoch 30. Các metrics như Dice và IoU tăng nhanh trong 20 epochs đầu tiên, sau đó cải thiện chậm hơn và dao động nhẹ xung quanh một giá trị ổn định. Đây là dấu hiệu tốt cho thấy mô hình đã hội tụ đến một điểm tối ưu. Đường validation loss không tăng đột ngột so với train loss, điều này chứng tỏ mô hình không bị overfitting dù được huấn luyện đến epoch thứ 62 trước khi early stopping kích hoạt.

#### 4.4 Phân tích kết quả dự đoán

Từ file prediction\_results.png, chúng em tiến hành phân tích định tính trên 10 mẫu ngẫu nhiên từ tập validation. Nhìn chung, mô hình cho thấy khả năng phân vùng rất tốt với ranh giới rõ ràng và chính xác giữa vùng tổn thương và vùng da bình thường.



Hình 3: Kết quả dự đoán trên tập validation

Đối với các trường hợp đơn giản, khi vùng tổn thương có ranh giới rõ ràng và độ tương phản cao với da xung quanh, mô hình dự đoán gần như hoàn hảo. Mask dự đoán trùng khớp cao với mask thực tế cả về hình dạng và kích thước. Điều này cho thấy các feature maps ở các tầng sâu của encoder đã học được đặc trưng về màu sắc, kết cấu và cấu trúc của vùng tổn thương một cách hiệu quả.

Trong một số trường hợp khó hơn, khi ranh giới giữa vùng tổn thương và da bình thường không rõ ràng hoặc khi có nhiễu trong ảnh, mô hình đôi khi dự đoán hơi conservative, tức là vùng dự đoán có thể nhỏ hơn một chút so với vùng thực tế. Tuy nhiên, phần lõi chính của vùng tổn thương vẫn được phát hiện chính xác. Điều này có thể là do mô hình học được cách tránh false positives, ưu tiên độ chính xác cao hơn là recall.

Một điểm đáng chú ý là mô hình xử lý tốt các trường hợp có nhiều vùng tổn thương nhỏ rời rạc trên cùng một ảnh. Skip connections trong kiến trúc UNet đã giúp bảo toàn thông tin chi tiết ở các tầng nông, cho phép mô hình phân biệt được các vùng nhỏ và không làm mất đi các cấu trúc tinh vi. Đây là một lợi thế quan trọng so với các kiến trúc encoder-decoder đơn thuần không có skip connections.

## 4.5 So sánh và đánh giá

So với các phương pháp truyền thống như segmentation dựa trên ngưỡng màu sắc hoặc edge detection, mô hình deep learning của chúng em vượt trội hoàn toàn. Các phương pháp truyền thống thường gặp khó khăn với sự đa dạng về màu sắc da, điều kiện ánh sáng và các nhiễu trong ảnh. Mô hình UNet có khả năng học được các đặc trưng phức tạp và bất biến với các biến đổi này.

Bảng 2: So sánh với các phương pháp khác

Phương pháp	Dice	IoU	Loss Function
Ngưỡng màu sắc	~0.65	~0.50	-
Edge Detection	~0.70	~0.55	-
FCN	~0.85	~0.75	BCE
UNet + DiceBCE	0.9138	0.8416	BCE + Dice
<b>Attention UNet + BCE</b>	<b>0.9271</b>	<b>0.8641</b>	<b>BCE + Dice</b>

Trong họ các mô hình deep learning cho phân vùng, UNet được biết đến là một trong những kiến trúc hiệu quả nhất cho các bài toán phân vùng y tế. So với các kiến trúc như FCN (Fully Convolutional Network) hoặc SegNet, UNet có lợi thế rõ rệt nhờ skip connections. Các kết quả IoU 0.8416 và Dice 0.9138 của mô hình UNet cơ bản của chúng em đã cho thấy hiệu suất tốt, có thể so sánh được với các nghiên cứu tương tự trên các tập dữ liệu phân vùng da được công bố.

Để cải thiện thêm hiệu suất, chúng em đã thử nghiệm với kiến trúc Attention UNet, một biến thể tiên tiến của UNet có tích hợp cơ chế attention. Cơ chế attention giúp mô hình tập trung vào các vùng quan trọng của ảnh trong quá trình phân vùng, đặc biệt hữu ích khi cần phân biệt ranh giới mờ giữa vùng tổn thương và da bình thường. Kết quả thử nghiệm cho thấy Attention UNet đạt Dice coefficient 0.9271 và IoU 0.8641, cải thiện lần lượt 1.33% và 2.25% so với UNet cơ bản. Mặc dù mức cải thiện không quá lớn, nhưng điều này cho thấy tiềm năng của cơ chế attention trong việc tăng độ chính xác phân vùng, đặc biệt trong các trường hợp khó với ranh giới không rõ ràng.

Qua thí nghiệm so sánh, chúng em nhận thấy việc kết hợp loss function DiceBCE đóng vai trò quan trọng trong cả hai kiến trúc. Dice Loss giúp mô hình học tốt hơn trong bối cảnh dữ liệu mất cân bằng (vùng tổn thương thường nhỏ hơn nhiều so với vùng da bình thường), trong khi BCE Loss đảm bảo độ chính xác ở mức pixel. Sự kết hợp này tạo nên hiệu suất vượt trội so với việc chỉ sử dụng BCE Loss đơn thuần.

Về môi trường thực nghiệm, việc sử dụng GPU CUDA đã giúp tăng tốc đáng kể quá trình huấn luyện. Với UNet cơ bản, mỗi epoch chỉ mất khoảng 7-8 giây, trong khi Attention UNet mất khoảng 9-10 giây do có thêm các khối attention. So với việc chạy trên CPU (hơn 2 phút mỗi epoch), tốc độ cải thiện rất rõ rệt. PyTorch được chọn làm framework chính do tính linh hoạt, dễ debug và hỗ trợ tốt cho research. Các thư viện hỗ trợ như torchvision cho data augmentation, matplotlib cho visualization và sklearn cho việc chia tập dữ liệu đều được tích hợp hiệu quả trong pipeline.

Một số điểm có thể cải thiện trong tương lai bao gồm việc mở rộng tập dữ liệu, áp dụng data augmentation mạnh hơn như rotation, flipping, color jittering để tăng tính đa dạng, và thử nghiệm với các kiến trúc tiên tiến hơn như UNet++ hoặc kết hợp với các backbone mạnh như ResNet, EfficientNet. Việc sử dụng pre-trained encoder từ các mô hình đã được huấn luyện trên ImageNet cũng có thể giúp cải thiện kết quả đáng kể, đặc biệt khi tập dữ liệu huấn luyện còn hạn chế như trong trường hợp của chúng em (chỉ 200 mẫu).

## 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 5.1 Kết luận

Qua dự án này, chúng em đã xây dựng thành công một hệ thống phân vùng da ung thư sử dụng mô hình UNet với hiệu suất cao. Mô hình đạt Dice coefficient 0.9138 và IoU 0.8416 trên tập validation, chứng tỏ khả năng phân vùng chính xác và đáng tin cậy. Các kỹ thuật regularization và optimization được áp dụng đã giúp mô hình tránh overfitting và hội tụ ổn định.

Kết quả này cho thấy tiềm năng ứng dụng của deep learning trong lĩnh vực y tế, đặc biệt là hỗ trợ chẩn đoán và phân tích ảnh y khoa. Hệ thống có thể được triển khai để hỗ trợ các bác sĩ da liễu trong việc phát hiện sớm và theo dõi diễn biến các tổn thương da, góp phần nâng cao chất lượng chăm sóc sức khỏe.

### 5.2 Hướng phát triển

Trong tương lai, dự án có thể được phát triển theo các hướng sau:

**Mở rộng dữ liệu:** Thu thập thêm dữ liệu từ nhiều nguồn khác nhau, bao gồm các loại tổn thương da đa dạng hơn, điều kiện ánh sáng khác nhau và các thiết bị chụp khác nhau. Việc tăng kích thước và đa dạng hóa tập dữ liệu sẽ giúp mô hình tổng quát tốt hơn.

**Data augmentation nâng cao:** Áp dụng các kỹ thuật data augmentation mạnh mẽ hơn như rotation, flipping, scaling, color jittering, elastic deformation để tăng tính đa dạng của dữ liệu huấn luyện. Điều này đặc biệt quan trọng khi tập dữ liệu ban đầu còn hạn chế.

**Kiến trúc tiên tiến hơn:** Thử nghiệm với các biến thể của UNet như UNet++ hoặc ResUNet. Các kiến trúc này có thể cải thiện khả năng học các đặc trưng phức tạp và xử lý tốt hơn các trường hợp khó.

**Transfer learning:** Sử dụng pre-trained encoder từ các mô hình như ResNet, EfficientNet, hoặc Vision Transformer để khởi tạo encoder của UNet. Cách tiếp cận này có thể giúp cải thiện hiệu suất đáng kể, đặc biệt với tập dữ liệu nhỏ.

**Phân loại đa lớp:** Mở rộng hệ thống để không chỉ phân vùng tổn thương mà còn phân loại loại tổn thương (lành tính, ác tính, các loại ung thư da khác nhau).

**Triển khai ứng dụng thực tế:** Phát triển giao diện người dùng thân thiện, tích hợp vào hệ thống y tế và thử nghiệm với dữ liệu lâm sàng thực tế để đánh giá hiệu quả trong môi trường ứng dụng.



## Tài liệu

- [1] Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), 234-241.
- [2] Long, J., Shelhamer, E., & Darrell, T. (2015). *Fully Convolutional Networks for Semantic Segmentation*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3431-3440.
- [3] Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(12), 2481-2495.
- [4] Codella, N., Rotemberg, V., Tschandl, P., et al. (2018). *Skin Lesion Analysis Toward Melanoma Detection 2018: A Challenge Hosted by the International Skin Imaging Collaboration (ISIC)*. arXiv preprint arXiv:1902.03368.
- [5] Kingma, D. P., & Ba, J. (2014). *Adam: A Method for Stochastic Optimization*. International Conference on Learning Representations (ICLR).
- [6] Ioffe, S., & Szegedy, C. (2015). *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. International Conference on Machine Learning (ICML), 448-456.

## PHỤ LỤC

### A. Code chính

Toàn bộ source code của dự án được lưu trữ tại: [https://github.com/NgocHung110/ChuyenDe2\\_Nhom22](https://github.com/NgocHung110/ChuyenDe2_Nhom22)

### B. Cấu hình hệ thống

- **GPU:** NVIDIA Tesla T4 (Google Colab)
- **RAM:** 12GB
- **Framework:** PyTorch 2.9
- **Python:** 3.12
- **CUDA:** 12.5

### C. Thông số huấn luyện

Tham số	Giá trị
Batch size	8
Learning rate	0.0001
Optimizer	Adam
Loss function	DiceBCE
Scheduler	ReduceLROnPlateau
Scheduler patience	5
Early stopping patience	10
Max epochs	100
Best epoch	52
Total parameters	~31M

Bảng 3: Chi tiết các tham số huấn luyện