

**ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**Nguyễn Trọng Nhân**

**MẠNG ĐỐI NGHỊCH TẠO SINH TRONG CHUYỂN ĐỔI  
ẢNH CHÂN DUNG SANG PHONG CÁCH ANIME**

**THỰC HÀNH NGHIÊN CỨU 2**

**Ngành: Khoa Học Máy Tính**

**HÀ NỘI - 2025**

**ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**Nguyễn Trọng Nhân**

**MẠNG ĐỐI NGHỊCH TẠO SINH TRONG CHUYỂN ĐỔI  
ẢNH CHÂN DUNG SANG PHONG CÁCH ANIME**

**THỰC HÀNH NGHIÊN CỨU 2**

**Ngành: Khoa học máy tính**

**Cán bộ hướng dẫn: TS. Ma Thị Châu**

**HÀ NỘI - 2025**

**VIETNAM NATIONAL UNIVERSITY, HANOI  
UNIVERSITY OF ENGINEERING AND TECHNOLOGY**



**Nguyen Trong Nhan**

**GENERATIVE ADVERSARIAL NETWORKS FOR CONVERTING  
PORTRAITS TO ANIME STYLE**

**RESEARCH PRACTICE 2**

**Major: Computer Science**

**Supervisor: Dr. Ma Thi Chau**

**HANOI - 2025**

## LỜI CẢM ƠN

*Trước hết, tôi xin chân thành cảm ơn TS. Ma Thị Châu, người hướng dẫn đề tài nghiên cứu lần này, đã tận tình hỗ trợ, chỉ bảo và động viên tôi trong suốt quá trình nghiên cứu và hoàn thành. Nhờ có sự hướng dẫn của cô, tôi đã có được những kiến thức quý báu và kinh nghiệm thực tiễn trong lĩnh vực nghiên cứu của mình.*

*Tôi cũng xin gửi lời cảm ơn đến các thầy cô giáo của Trường Đại học Công Nghệ, đã truyền đạt cho tôi những kiến thức cơ bản và nâng cao về ngành Khoa học Máy Tính. Tôi xin chúc các thầy cô luôn mạnh khỏe và thành công trong công tác giảng dạy và nghiên cứu khoa học.*

*Sau cùng, tôi muốn bày tỏ lòng biết ơn sâu sắc đến gia đình, người thân và bạn bè, những người luôn ở bên cạnh tôi trong những thời điểm khó khăn nhất, luôn động viên và khích lệ tôi trong cuộc sống và công việc. Thành tựu này của tôi không thể có được nếu thiếu sự giúp đỡ của họ.*

*Mặc dù tôi đã cố gắng hoàn thành báo cáo nhưng không thể tránh khỏi những sai sót, tôi rất mong nhận được nhận xét và sự hướng dẫn từ phía giáo viên và hội đồng.*

*Hà Nội, ngày ... tháng ... năm 2023*

**Học viên**

**Nguyễn Trọng Nhân**

## LỜI CAM ĐOAN

Tôi xin cam đoan rằng tất cả các nội dung trong tài liệu này đều là kết quả của công trình nghiên cứu cá nhân của tôi dưới sự hướng dẫn của TS.Ma Thị Châu. Tôi không sao chép bất kỳ tài liệu hay công trình nghiên cứu nào của người khác mà không chỉ rõ nguồn gốc trong phần tài liệu tham khảo. Tôi hiểu rằng việc sao chép trái phép là một hành vi vi phạm đạo đức học thuật và tôi sẽ chịu trách nhiệm về những hành vi này.

Tôi cam kết rằng, bản báo cáo của tôi không vi phạm bản quyền của bất kỳ ai và không vi phạm bất kỳ quyền sở hữu nào, cũng như bất kỳ ý tưởng, kỹ thuật, trích dẫn hoặc bất kỳ tài liệu nào từ công trình nghiên cứu của người khác. Tôi xin nhận đầy đủ trách nhiệm và sẵn sàng chấp nhận mọi biện pháp kỷ luật nếu vi phạm cam kết.

*Hà Nội, ngày ... tháng ... năm 2023*

**Học viên**

**Nguyễn Trọng Nhân**

## TÓM TẮT

**Tóm tắt:** Sự bùng nổ của nội dung số và nghệ thuật kỹ thuật số, đặc biệt trong lĩnh vực Anime/Manga, kéo theo nhu cầu mạnh mẽ về các công cụ hỗ trợ sáng tạo dựa trên trí tuệ nhân tạo. Trong đó, bài toán chuyển đổi ảnh chân dung người thật sang phong cách anime 2D (photo-to-anime) vừa giàu tiềm năng ứng dụng, vừa đặt ra nhiều thách thức về chất lượng hình ảnh và khả năng giữ lại đặc điểm nhận dạng khuôn mặt. Các phương pháp truyền thống dựa trên chuyển phong cách (style transfer) hoặc các kiến trúc GAN thể hệ đầu thường gặp vấn đề nhiễu, tạo tác (artifacts) và khó đảm bảo tính ổn định khi huấn luyện.

Luận văn này đề xuất một mô hình chuyển đổi ảnh chân dung người sang phong cách anime dựa trên kiến trúc Mạng đối nghịch tạo sinh (GAN), với nền tảng là kiến trúc Double-Tail GAN (DTGAN) của AnimeGAN thể hệ mới. Mô hình sử dụng kiến trúc encoder-decoder dựa trên ResNet với generator hai nhánh (double-tail) gồm nhánh hỗ trợ tạo ảnh anime thô và nhánh chính tinh chỉnh để tạo ra ảnh anime chất lượng cao. Bên cạnh đó, luận văn áp dụng kỹ thuật chuẩn hóa Linearly Adaptive Denormalization (LADE) nhằm giảm nhiễu và làm mượt vùng màu theo phong cách anime, đồng thời xây dựng hệ thống hàm mất mát (loss) chuyên biệt, kết hợp giữa mất mát đối nghịch, mất mát nội dung/nhận dạng (dựa trên đặc trưng VGG19), mất mát phong cách và các mất mát làm mượt-chỉnh chi tiết để cân bằng giữa bảo toàn nội dung và thể hiện phong cách.

Mô hình được huấn luyện trên hai tập dữ liệu không ghép cặp gồm ảnh chân dung người thật và ảnh chân dung anime 2D, với các bước tiền xử lý như phát hiện và căn chỉnh khuôn mặt, chuẩn hóa kích thước và tăng cường dữ liệu. Kết quả thực nghiệm cho thấy mô hình đề xuất tạo ra ảnh anime có tính thẩm mỹ cao, giữ được đặc điểm khuôn mặt của đối tượng gốc và giảm đáng kể nhiễu, được xác nhận qua các chỉ số định lượng (FID, LPIPS) cũng như đánh giá định tính từ người dùng. Cuối cùng, luận văn thảo luận các hạn chế và đề xuất hướng phát triển trong tương lai như tăng mức độ điều khiển phong cách chi tiết (màu mắt, kiểu tóc, biểu cảm) và mở rộng sang bài toán chuyển đổi video-to-anime.

**Từ khóa:** GAN, AnimeGAN, DTGAN, chuyển phong cách ảnh, ảnh chân dung, phong cách anime, LADE, VGG19.

## ABSTRACT

**Abstract:** The explosion of digital content and digital art, especially in the Anime/Manga field, has led to a strong demand for AI-powered creative tools. Among these, the task of converting real-person portraits to 2D anime style (photo-to-anime) offers significant application potential but also poses many challenges regarding image quality and the ability to retain facial identity features. Traditional methods based on style transfer or early-generation GAN architectures often suffer from noise, artifacts, and difficulty in ensuring training stability.

This thesis proposes a model for converting real-person portraits to anime style based on the Generative Adversarial Network (GAN) architecture, specifically building upon the Double-Tail GAN (DTGAN) architecture of the new generation AnimeGAN. The model utilizes a ResNet-based encoder–decoder architecture with a double-tail generator, consisting of a branch that assists in generating rough anime images and a main branch that refines them to produce high-quality anime images. Additionally, the thesis applies the Linearly Adaptive Denormalization (LADE) technique to reduce noise and smooth color regions in anime style, while also developing a specialized loss function system that combines adversarial loss, content/identity loss (based on VGG19 features), style loss, and smoothing–detail adjustment losses to balance content preservation and style expression.

The model was trained on two unpaired datasets of real-person portraits and 2D anime portraits, with preprocessing steps such as face detection and alignment, size normalization, and data augmentation. Experimental results show that the proposed model generates aesthetically pleasing anime images, preserves the facial characteristics of the original subject, and significantly reduces noise, as confirmed by quantitative metrics (FID, LPIPS) as well as qualitative evaluation from users. Finally, the thesis discusses limitations and proposes future development directions such as increasing the level of control over detailed styles (eye color, hairstyle, expression) and extending to the video-to-anime conversion problem.

**Keywords:** *GAN, AnimeGAN, DTGAN, image style transfer, portrait, anime style, LADE, VGG19.*

# MỤC LỤC

Lời cảm ơn . . . . .	
Lời cam đoan . . . . .	
Tóm tắt . . . . .	
Abstract . . . . .	
Mục lục . . . . .	i
Danh mục hình vẽ . . . . .	iii
Danh mục bảng biểu . . . . .	iv
Mở đầu . . . . .	1
CHƯƠNG 1. CƠ SỞ LÝ THUYẾT VÀ TỔNG QUAN NGHIÊN CỨU . . . . .	4
1.1. Mạng Nơ-ron Tích chập . . . . .	4
1.1.1. Hoạt động của Lớp Tích chập . . . . .	4
1.1.2. Lớp Pooling và Tối ưu hóa Bản đồ Đặc trưng . . . . .	5
1.1.3. Lớp Kết nối Đầy đủ và Phân loại Quyết định . . . . .	6
1.1.4. Sự Phát triển Kiến trúc và Nguyên lý Thiết kế Sâu . . . . .	6
1.1.5. Kiến trúc CNN cho Tác vụ Định vị và Phân đoạn Mật độ cao . . . . .	9
1.1.6. CNNs trong Dịch Ảnh và Thích ứng Miền . . . . .	12
1.1.7. Đánh giá và Xu hướng Tương lai của Kiến trúc Tích chập . . . . .	14
1.2. Kiến trúc Mạng đối nghịch tạo sinh . . . . .	16
1.2.1. Nguyên lý Hoạt động và Cơ sở Lý thuyết . . . . .	16
1.2.2. Thách thức trong Huấn luyện GAN . . . . .	17
1.2.3. Các Giải pháp về Hàm Mục tiêu và Toán học . . . . .	19
1.2.4. Các Kỹ thuật Ổn định Huấn luyện . . . . .	20
1.2.5. Đánh giá Chất lượng . . . . .	21
1.3. Các mô hình GAN tiêu biểu cho chuyển đổi ảnh sang phong cách Anime . . . . .	21
1.3.1. CycleGAN và các biến thể (chuyển đổi với dữ liệu không cặp) . . . . .	21
1.3.2. StyleGAN và phương pháp dựa trên GAN inversion (kiểm soát chất lượng cao) . . . . .	23
1.3.3. AnimeGAN và kiến trúc Double-Tail GAN (AnimeGANv3) . . . . .	24



CHƯƠNG 2. PHƯƠNG PHÁP NGHIÊN CỨU ĐỀ XUẤT . . . . .	28
2.1. Lựa chọn kiến trúc mô hình cơ sở . . . . .	28
2.2. Cấu trúc mô hình đề xuất . . . . .	28
2.2.1. Kiến trúc Generator “Double-Tail” (Hai đuôi) . . . . .	28
2.2.2. Kỹ thuật chuẩn hóa Linearly Adaptive Denormalization (LADE) . . . . .	30
2.2.3. Hệ thống Hàm Mất mát (Loss Functions) chuyên biệt . . . . .	31
2.3. Tập dữ liệu và Tiền xử lý . . . . .	32
CHƯƠNG 3. THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ . . . . .	34
3.1. Môi trường Thực nghiệm và Thông số Huấn luyện . . . . .	34
3.2. Kết quả Huấn luyện và Phân tích Log . . . . .	34
3.3. Kết quả và Phân tích . . . . .	36
3.4. Hạn chế và Giải pháp đề xuất . . . . .	38
Kết luận . . . . .	39
TÀI LIỆU THAM KHẢO . . . . .	40

## DANH MỤC HÌNH VẼ

Hình 1.1	Mạng Nơ-ron Tích chập cơ bản . . . . .	4
Hình 1.2	Lớp Pooling . . . . .	5
Hình 1.3	AlexNet . . . . .	7
Hình 1.4	Phân đoạn Thực thể . . . . .	9
Hình 1.5	Mô hình U-Net . . . . .	11
Hình 1.6	Không gian tiềm ẩn chia sẻ và kiến trúc mạng UNIT. . . . .	13
Hình 1.7	Hình ảnh của GAN . . . . .	16
Hình 1.8	Kiến trúc của CycleGAN . . . . .	22
Hình 1.9	Sơ đồ quy trình của Double-Tail GAN (DTGAN). . . . .	25
Hình 2.1	LADE . . . . .	30
Hình 3.1	Kết quả huấn luyện Pre-training Generator . . . . .	35
Hình 3.2	Training step (G_loss & D_loss) . . . . .	36
Hình 3.3	Other loss (con_loss & sty_loss & color_loss) . . . . .	36

## DANH MỤC BẢNG BIỂU

Bảng 1.1	Lựa chọn Kiến trúc và Tối ưu hóa Lớp FC . . . . .	9
Bảng 1.2	<b>Các Kỹ thuật Upsampling trong Decoder của CNN</b> . . . . .	12
Bảng 1.3	Tóm tắt Các Chế độ Thất bại của GAN . . . . .	19
Bảng 3.1	Kết quả . . . . .	37

# MỞ ĐẦU

## Tính cấp thiết và Ý nghĩa của Đề tài

Sự phát triển mạnh mẽ của nghệ thuật kỹ thuật số cùng với thị trường Anime/Manga toàn cầu đã tạo động lực lớn cho các nghiên cứu về chuyển đổi phong cách hình ảnh. Theo thống kê năm 2024, quy mô thị trường anime toàn cầu đạt khoảng 81,96 tỷ USD và dự kiến vượt 200 tỷ USD vào năm 2034. Công nghệ AI hiện nay cũng đang dần xâm nhập vào quy trình sản xuất anime – ví dụ, các công cụ Generative AI đã có thể tự động hoá việc vẽ phông nền và tô màu, giúp giảm bớt công việc lặp lại cho các họa sĩ. Trong bối cảnh đó, nhu cầu tự động hoá chuyển đổi ảnh thực sang phong cách anime trở nên cấp thiết, nhằm phục vụ cộng đồng người hâm mộ khổng lồ và ngành công nghiệp sáng tạo nội dung đang bùng nổ.

Tuy nhiên, việc chuyển một ảnh chân dung người thật sang tranh anime là thách thức không nhỏ. Phong cách anime có đặc trưng rất khác biệt (đường nét đơn giản, mắt to, màu sắc phẳng, v.v.), trong khi ảnh chụp chứa nhiều chi tiết thực tế phức tạp. Các phương pháp truyền thống dựa trên Neural Style Transfer thường gặp khó khăn trong việc giữ được nội dung gốc và dễ sinh ra nhiễu, tạo tác (artifacts) khi khác biệt phong cách quá lớn. Mặt khác, vẽ tay thủ công bởi các họa sĩ tuy chất lượng cao nhưng tốn kém thời gian và công sức. Do đó, bài toán đặt ra là làm thế nào để tự động hoá quá trình này mà vẫn đảm bảo chất lượng cao và bảo toàn được đặc điểm nhận dạng của đối tượng trong ảnh gốc.

Mạng Đối nghịch Tạo sinh (GAN) nổi lên như một công nghệ đột phá có thể giải quyết các nhiệm vụ tổng hợp hình ảnh phức tạp. Được Ian Goodfellow giới thiệu năm 2014 và được Yann LeCun ca ngợi là “ý tưởng thú vị nhất của ML trong 10 năm”, GAN đã chứng tỏ hiệu quả vượt trội trong việc sinh ảnh chân thực từ dữ liệu huấn luyện. Đặc biệt trong bài toán dịch chuyển phong cách (style transfer), GANs cung cấp một khuôn khổ linh hoạt để huấn luyện mô hình tạo ảnh anime từ ảnh thật mà không đòi hỏi dữ liệu ảnh cặp một-một. Nhờ GAN, những tiến bộ gần đây cho thấy khả năng tạo các hình ảnh anime hóa ngày càng sắc nét và chính xác hơn, mở ra hướng tiếp cận mới cho bài toán này.

## Mục tiêu Nghiên cứu

Mục tiêu tổng quát là xây dựng và đánh giá một mô hình chuyển đổi ảnh chân dung người thật sang phong cách anime chất lượng cao dựa trên GAN, đảm bảo giữ được các nét nhận dạng chính của khuôn mặt gốc. Mục tiêu cụ thể bao gồm:

- Lựa chọn kiến trúc GAN phù hợp: Nghiên cứu các kiến trúc GAN tiên tiến và chọn mô hình nền tảng hiệu quả nhất cho bài toán (dự kiến sử dụng kiến trúc Double-Tail GAN (DTGAN) – phiên bản AnimeGAN thế hệ mới).
- Xây dựng tập dữ liệu và tiền xử lý: Thu thập hoặc tinh chỉnh tập dữ liệu gồm ảnh chân dung thực và ảnh anime tương ứng (dạng không ghép cặp), áp dụng các bước tiền xử lý như căn chỉnh khuôn mặt, chuẩn hóa kích thước, tăng cường dữ liệu.
- Đề xuất hàm mất mát và chiến lược huấn luyện tối ưu: Thiết kế bộ hàm loss chuyên biệt (kết hợp giữa loss truyền thống của GAN và các loss phong cách/anime đặc thù) cùng lịch trình huấn luyện thích hợp nhằm tăng độ ổn định và làm nổi bật chi tiết ảnh đầu ra.
- Đánh giá mô hình: Đánh giá chất lượng ảnh anime sinh ra bằng cả phương pháp định lượng (ví dụ: FID, PSNR, LPIPS) và định tính (đánh giá cảm quan, khảo sát người dùng), so sánh với các phương pháp chuyển đổi phong cách hiện có để xác định hiệu quả của mô hình đề xuất.

## Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu: Các kiến trúc mạng Generative Adversarial Networks (GANs) phục vụ cho nhiệm vụ dịch chuyển phong cách hình ảnh, đặc biệt là các mô hình chuyên dành cho chuyển ảnh người sang hoạt hình/anime. Ngoài ra, đề tài cũng liên quan đến các kỹ thuật thị giác máy tính và học sâu hỗ trợ, như mô hình encoder-decoder, các phương pháp xử lý ảnh (phát hiện, căn chỉnh khuôn mặt) và các hàm tổn thất perceptual.

Phạm vi nghiên cứu: Đề tài tập trung vào ảnh chân dung người (face portraits) và phong cách Anime 2D. Cụ thể, ảnh đầu vào giới hạn ở ảnh khuôn mặt người thật (có thể chụp từ camera hoặc ảnh selfie), và ảnh đầu ra hướng đến phong cách tranh vẽ nhân vật anime 2D (phong cách hoạt hình Nhật Bản). Mô hình được huấn luyện và đánh giá trên tập dữ liệu ảnh chân dung và ảnh anime không ghép cặp. Những khía cạnh ngoài phạm vi bao gồm chuyển đổi phong cách cho video, phong cách 3D hoặc các phong cách hoạt

hình khác (cartoon kiểu phương Tây, tranh phác hoạ, v.v.), cũng như không đi sâu vào các kỹ thuật diffusion models hay transformer mới hơn (chỉ tập trung vào GAN truyền thống trong giai đoạn 2020-2025).

### **Cấu trúc luận văn**

Luận văn được tổ chức thành 5 chương như sau:

Chương 1: Mở đầu – Trình bày sự cần thiết, mục tiêu, phạm vi của đề tài và khái quát nội dung nghiên cứu.

Chương 2: Cơ sở Lý thuyết và Tổng quan Nghiên cứu – Tổng hợp nền tảng lý thuyết về học sâu, thị giác máy, kiến trúc GAN, và các nghiên cứu liên quan trong lĩnh vực chuyển ảnh chân dung sang phong cách anime.

Chương 3: Phương pháp Nghiên cứu Đề xuất – Mô tả chi tiết mô hình GAN đề xuất (kiến trúc Double-Tail GAN), các cải tiến kỹ thuật (như LADE – adaptive denormalization, hàm mất mát mới), cùng quy trình huấn luyện.

Chương 4: Thực nghiệm và Kết quả – Trình bày thiết lập thực nghiệm, quá trình huấn luyện mô hình trên tập dữ liệu thu thập, và kết quả đánh giá so sánh với các phương pháp khác. Phân tích các kết quả định lượng và định tính, minh hoạ bằng các hình ảnh đầu vào/đầu ra.

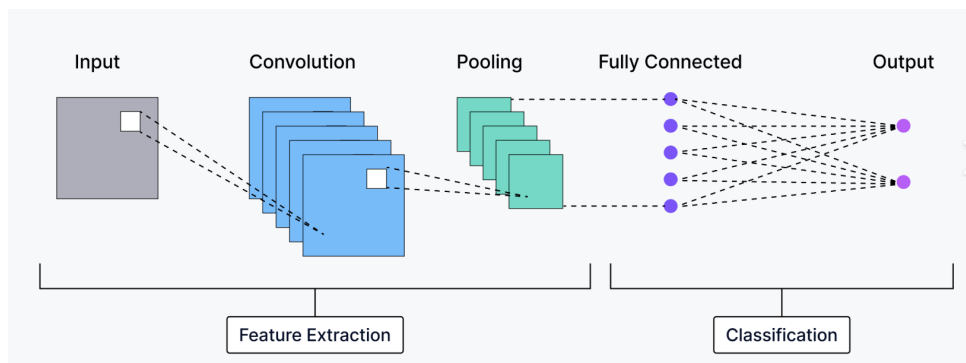
Chương 5: Kết luận và Hướng phát triển – Tóm tắt những đóng góp chính của luận văn, thảo luận những hạn chế còn tồn tại và đề xuất hướng nghiên cứu trong tương lai (mở rộng sang video, phong cách khác, ứng dụng diffusion model, v.v.).

# CHƯƠNG 1

## CƠ SỞ LÝ THUYẾT VÀ TỔNG QUAN NGHIÊN CỨU

### 1.1. Mạng Nơ-ron Tích chập

Mạng Nơ-ron Tích chập (Convolutional Neural Networks – CNNs) đã củng cố vị thế là kiến trúc nền tảng trong lĩnh vực thị giác máy tính, tạo ra những bước tiến lớn trong các tác vụ từ phân loại ảnh đơn giản đến phân đoạn thực thể phức tạp. Sự thành công của CNN bắt nguồn từ khả năng học hỏi các biểu diễn đặc trưng phân cấp, tự động trích xuất các thông tin hình học ngày càng trừu tượng từ dữ liệu đầu vào thô. [6]



Hình 1.1. Mạng Nơ-ron Tích chập cơ bản

#### 1.1.1. Hoạt động của Lớp Tích chập

Lớp tích chập là khối xây dựng cơ bản của mọi kiến trúc CNN hiện đại. Cơ chế hoạt động của lớp này được thiết kế đặc biệt để xử lý dữ liệu ảnh bằng cách khai thác tính cục bộ và tính bất biến dịch chuyển thống kê của đặc trưng hình ảnh.

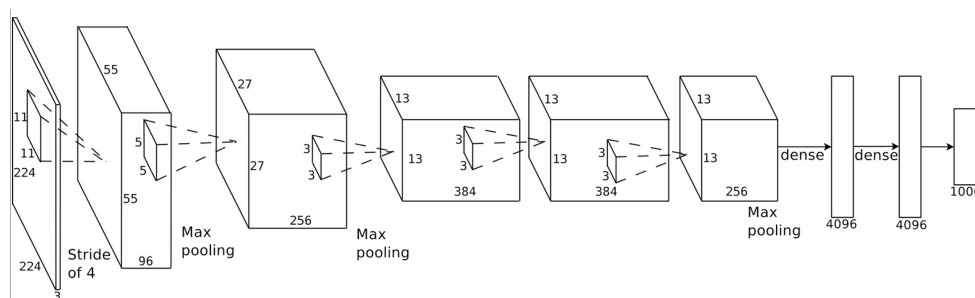
Lớp tích chập vận hành thông qua các kernel nhỏ (hay còn gọi là bộ lọc). Các kernel này trượt (slide) trên ảnh đầu vào, thực hiện phép nhân tích chập với từng phần của ảnh để tính toán tích vô hướng cục bộ. Điều này cho phép lớp tích chập hoạt động như các bộ lọc cục bộ, chuyên trách trích xuất các đặc trưng hình học cơ bản, chẳng hạn như các cạnh (edges), đường cong (curves), hoặc các kết cấu (textures).[12]

Mỗi kernel trong lớp tích chập được học để phát hiện một loại đặc trưng cụ thể và áp dụng phát hiện đó trên toàn bộ ảnh. Khả năng học các bộ lọc cục bộ này, thay vì dựa vào các đặc trưng được thiết kế thủ công (handcrafted features) như SIFT hay HOG

trong các mô hình thị giác máy tính truyền thống, là một đổi mới cốt lõi của CNN. Sự thay đổi này đã giúp các mô hình CNN giảm đáng kể nhu cầu về kinh nghiệm chuyên môn trong việc tiền xử lý dữ liệu đầu vào.[6] Quá trình này tạo tiền đề cho việc xây dựng các đặc trưng phân cấp phong phú (rich hierarchy of image features), nơi các lớp sâu hơn dần dần kết hợp các đặc trưng cấp thấp thành các biểu diễn trừu tượng và ngữ nghĩa hơn.[12]

### 1.1.2. Lớp Pooling và Tối ưu hóa Bản đồ Đặc trưng

Thông thường, lớp Pooling được thêm vào ngay sau một lớp tích chập. Chức năng chính của lớp Pooling là giảm độ phân giải không gian của bản đồ đặc trưng (feature maps) bằng cách chia chúng thành các vùng con hình chữ nhật và giảm mẫu các đặc trưng trong mỗi vùng con thành một giá trị duy nhất, thường là giá trị trung bình (average) hoặc giá trị cực đại (maximum). [52]



Hình 1.2. Lớp Pooling

Quá trình giảm mẫu này thực hiện hai mục đích quan trọng. Thứ nhất, nó giảm kích thước dữ liệu, làm giảm chi phí tính toán trong các lớp sau. Thứ hai, và quan trọng hơn về mặt lý thuyết, hoạt động pooling mang lại một mức độ bất biến dịch chuyển cục bộ (local translational invariance) cho các đặc trưng. Tính bất biến này giúp CNN trở nên vững vàng hơn (robust) đối với các biến thể nhỏ trong vị trí hoặc biến dạng của các đặc trưng, cho phép mô hình nhận dạng các đối tượng ngay cả khi chúng hơi bị dịch chuyển trong ảnh.[52]

Tuy nhiên, việc giảm độ phân giải không gian thông qua pooling là một sự đánh đổi. Mặc dù pooling tạo ra tính bất biến, nó cũng dẫn đến sự mất mát thông tin vị trí chính xác (spatial location). Sự mất mát này không đáng kể trong các tác vụ phân loại hình ảnh cấp độ toàn cục, nhưng lại trở thành một rào cản kỹ thuật nghiêm trọng đối với các tác vụ định vị mật độ cao (dense prediction tasks) như phân đoạn thực thể, đòi hỏi sự căn chỉnh pixel-to-pixel.[17]



### ***1.1.3. Lớp Kết nối Đầy đủ và Phân loại Quyết định***

Các lớp tích chập và pooling hoạt động như các khối trích xuất đặc trưng (feature extractors). Để hoàn thành tác vụ phân loại, CNN cần các lớp kết nối đầy đủ (FC layers) ở phần cuối của kiến trúc. [6]

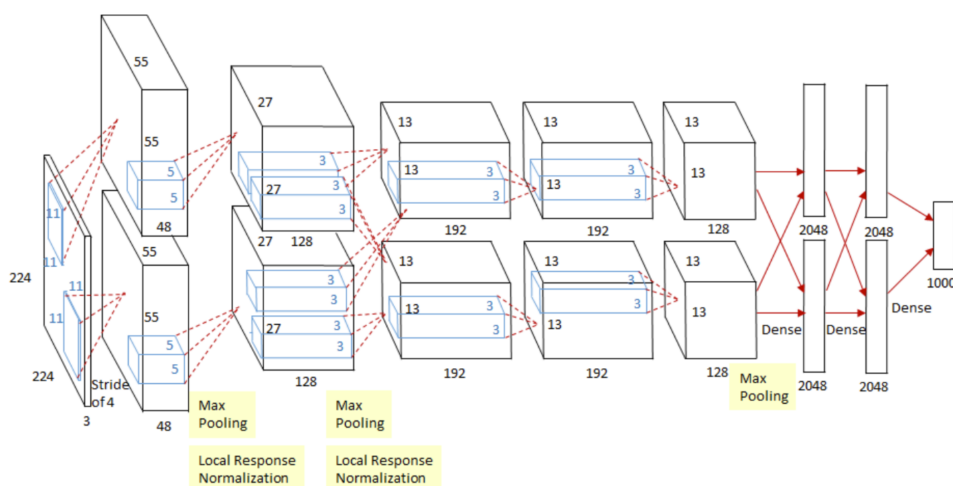
Trước khi đi vào lớp FC, bản đồ đặc trưng 2D/3D cuối cùng được làm phẳng (flattened) thành một vector 1D. Lớp FC là lớp cuối cùng của mạng, có chức năng tổng hợp toàn bộ các đặc trưng trừu tượng đã được trích xuất bởi các khối xử lý trước đó. Mỗi nơ-ron trong lớp FC này được kết nối với tất cả các đầu vào của lớp trước, và cuối cùng, lớp này gán một giá trị xác suất cho hình ảnh thuộc về từng lớp trong số  $C$  lớp khả dĩ. [6]

Một phát hiện quan trọng từ các nghiên cứu gần đây là mối quan hệ giữa độ sâu kiến trúc CNN và nhu cầu thiết kế lớp FC. Phân tích cho thấy: Mạng Nông (Shallow CNNs), do các đặc trưng được trích xuất ở lớp tích chập cuối cùng ít trừu tượng hơn, mạng nông cần một số lượng lớn nơ-ron và nhiều lớp FC hơn để đạt được hiệu suất phân loại tương đương. Mạng Sâu (Deeper CNNs), ngược lại, mạng sâu đã trích xuất được các đặc trưng trừu tượng hóa cao hơn. Do đó, chúng cần ít nơ-ron FC hơn để tổng hợp thông tin và đưa ra quyết định. Việc hình thức hóa mối quan hệ này giữa kiến trúc và tập dữ liệu là một bước tiến quan trọng giúp các nhà thực hành chuyển quá trình lựa chọn kiến trúc từ kinh nghiệm (expertise) sang một quy trình thiết kế tự động và có hệ thống. [6]

### ***1.1.4. Sự Phát triển Kiến trúc và Nguyên lý Thiết kế Sâu***

#### ***1.1.4.1. Các Cột mốc phát triển và Ý nghĩa của Độ sâu Mạng***

Sự trỗi dậy của CNN trong thị giác máy tính hiện đại được đánh dấu bằng các kiến trúc cột mốc. AlexNet, được phát triển vào năm 2012, là mô hình CNN sâu đầu tiên được công nhận rộng rãi, nổi bật qua thành tích trong Thử thách Nhận dạng Hình ảnh Quy mô Lớn ImageNet (ILSVRC). AlexNet đã chứng minh một cách dứt khoát rằng độ sâu của mô hình là yếu tố thiết yếu để đạt hiệu suất cao, điều này chỉ trở nên khả thi nhờ vào việc tận dụng đơn vị xử lý đồ họa (GPUs) để giảm chi phí tính toán khi huấn luyện. [51]



Hình 1.3. AlexNet

Sau thành công ban đầu, các kiến trúc tiếp theo (như VGGNet) tiếp tục đẩy giới hạn về độ sâu. Đồng thời, nghiên cứu cũng tập trung vào việc tối ưu hóa hiệu suất và tham số. Việc đơn giản hóa các kiến trúc dựa trên LeNet đã đạt được những giảm thiểu đáng kể về độ phức tạp tính toán và số lượng tham số trong khi vẫn duy trì hiệu suất cạnh tranh.<sup>8</sup> Điều này nhấn mạnh tiềm năng của các kiến trúc hiệu quả (efficient architectures) trong việc giải quyết các ràng buộc phần cứng trong ứng dụng thực tế.

#### 1.1.4.2. Chiến lược Chống Tắt Dần Gradient: Inception và Residual Connections

Khi các mạng trở nên sâu hơn, vấn đề tắt dần gradient (vanishing gradients) và khó khăn trong tối ưu hóa đã thúc đẩy sự ra đời của các chiến lược kiến trúc mới.

Kiến trúc Inception nổi bật vì khả năng đạt hiệu suất rất tốt với chi phí tính toán tương đối thấp. Sau đó, sự giới thiệu của kết nối residual (hay skip connections) trong ResNet đã mang lại hiệu suất dẫn đầu (state-of-the-art) vào năm 2015. Điều này đặt ra câu hỏi về lợi ích của việc kết hợp kiến trúc Inception với kết nối residual. [44]

Nghiên cứu về Inception-ResNet đã cung cấp bằng chứng thực nghiệm rõ ràng rằng việc huấn luyện với kết nối residual tăng tốc đáng kể quá trình huấn luyện của mạng Inception.[44] Lợi ích này không chỉ là lý thuyết; kỹ thuật DropIn, một phương pháp dần dần cho phép đào tạo trực tiếp các mạng sâu và khó huấn luyện như VGG16, cho thấy sự cải thiện đáng kể trong độ chính xác của mạng.[37]

Các biến thể Inception-ResNet (như v1 và v2) đã được thiết kế lại để sử dụng các khối Inception rẻ hơn. Để bù đắp cho việc giảm chiều (dimensionality reduction) do khối Inception gây ra trước khi thực hiện phép cộng residual, một lớp mở rộng bộ

lọc (filter-expansion layer), sử dụng tích chập  $1 \times 1$  không có hàm kích hoạt, đã được thêm vào sau mỗi khối Inception. Ngoài ra, việc sử dụng kỹ thuật Activation Scaling cũng được chứng minh là cần thiết để ổn định quá trình huấn luyện các mạng Inception residual rất rộng. Mặc dù các nhà nghiên cứu đã chứng minh rằng việc huấn luyện các mạng rất sâu mà không cần kết nối residual là khả thi, lợi thế về tốc độ tối ưu hóa mà kết nối residual mang lại là một lý do kỹ thuật mạnh mẽ để chúng được áp dụng rộng rãi.[44]

#### *1.1.4.3. Nguyên tắc Thiết kế Kiến trúc dựa trên Dữ liệu và Tối ưu hóa FC*

Việc lựa chọn kiến trúc CNN (sâu hay nông) không nên chỉ dựa trên kinh nghiệm mà phải tương quan với đặc điểm của tập dữ liệu đang được sử dụng. Các tập dữ liệu có thể được phân loại là sâu (deeper), nghĩa là chúng có số lượng mẫu lớn trên mỗi lớp (ví dụ: CIFAR-10), hoặc rộng (wider), nghĩa là chúng có nhiều lớp nhưng ít mẫu hơn trên mỗi lớp (ví dụ: CIFAR-100, Tiny ImageNet).[6] Nghiên cứu đã chỉ ra các quy tắc thiết kế mang tính hướng dẫn như sau:

1. Dữ liệu Sâu (Deeper Datasets): Các tập dữ liệu này phù hợp hơn với các kiến trúc CNN sâu (ví dụ: CNN-2 và CNN-3). Do các kiến trúc sâu có nhiều tham số có thể huấn luyện hơn, chúng cần số lượng hình ảnh lớn trên mỗi chủ thể để huấn luyện hiệu quả. Các kiến trúc sâu hơn đã chứng minh hiệu suất tốt hơn trên các tập dữ liệu sâu như CIFAR-10 và CRCHistoPhenotypes.[6]
2. Dữ liệu Rộng (Wider Datasets): Các tập dữ liệu này hoạt động hiệu quả hơn với các kiến trúc CNN nông (ví dụ: CNN-1). Mạng nông có ít tham số hơn, điều này phù hợp hơn với các tập dữ liệu có sự đa dạng lớp lớn nhưng số lượng mẫu trên mỗi lớp bị hạn chế.[6]

Mối quan hệ giữa độ sâu kiến trúc và thiết kế lớp FC cũng tuân theo logic này. Mạng nông cần nhiều nơ-ron và lớp FC hơn để tổng hợp các đặc trưng, trong khi mạng sâu cần ít nơ-ron FC hơn. Việc xác định mối quan hệ qua lại này cung cấp một khuôn khổ khoa học giúp các nhà phát triển lựa chọn kiến trúc tối ưu hóa hiệu suất và chi phí tính toán ngay từ đầu, giảm thiểu quá trình thử và sai tốn thời gian.

Bảng 1.1. Lựa chọn Kiến trúc và Tối ưu hóa Lớp FC

Kiến trúc CNN	Đặc điểm Tập dữ liệu Tối ưu	Yêu cầu Lớp FC	Lý do Kỹ thuật
Sâu (Deeper)	Sâu (Nhiều mẫu/lớp)	Thấp (Ít nơ-ron/lớp)	Đặc trưng trừu tượng hóa cao, giảm gánh nặng tính toán trong giai đoạn quyết định.
Nông (Shallow)	Rộng (Nhiều lớp, ít mẫu/lớp)	Cao (Nhiều nơ-ron/lớp)	Bù đắp cho đặc trưng ít trừu tượng, mô hình có ít tham số hơn, phù hợp với sự đa dạng lớp.

### 1.1.5. Kiến trúc CNN cho Tác vụ Định vị và Phân đoạn Mật độ cao

#### 1.1.5.1. Phân đoạn Thực thể (Instance Segmentation) với Mask R-CNN

Trong khi các kiến trúc như AlexNet và ResNet tập trung vào phân loại, các tác vụ thị giác máy tính phức tạp hơn như phân đoạn thực thể đòi hỏi độ chính xác không gian cao. Mask R-CNN là một khung làm việc linh hoạt và đơn giản về mặt khái niệm, mở rộng Faster R-CNN để không chỉ phát hiện đối tượng mà còn đồng thời tạo ra mặt nạ phân đoạn chất lượng cao cho mỗi thực thể. [17]

Mask R-CNN duy trì quy trình hai giai đoạn của Faster R-CNN, nhưng ở giai đoạn thứ hai, nó bổ sung một nhánh thứ ba hoạt động song song với nhánh dự đoán hộp giới hạn (bounding-box) và phân loại. Nhánh này là một Mạng Tích chập Đầy đủ (Fully Convolutional Network – FCN) nhỏ, được áp dụng trên từng Vùng Quan tâm (Region of Interest – RoI) để dự đoán một mặt nạ phân đoạn  $m \times m$  theo cách pixel-to-pixel. [17]



Hình 1.4. Phân đoạn Thực thể

**Lớp RoIAlign và Căn chỉnh Chính xác:** Thành phần kỹ thuật quan trọng nhất làm nên sự thành công của Mask R-CNN là việc giới thiệu lớp RoIAlign. Faster R-CNN sử dụng lớp RoIPooling, lớp này thực hiện lượng tử hóa (spatial quantization) thô, dẫn

đến sự sai lệch (misalignment) giữa input của mạng và output pixel-to-pixel, gây ảnh hưởng tiêu cực đến độ chính xác không gian.[17]

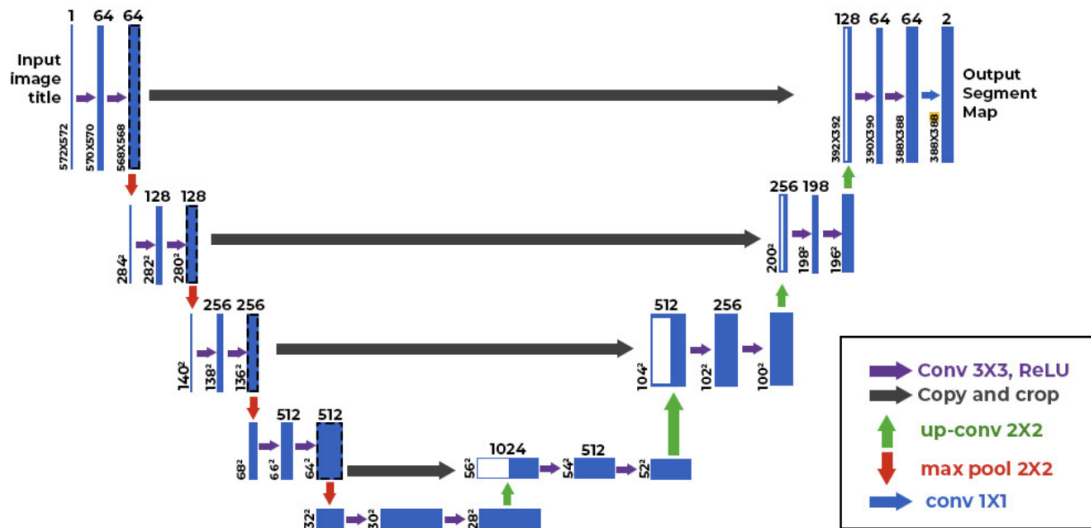
RoIAlign khắc phục vấn đề này bằng cách là một lớp không lượng tử hóa (quantization-free), duy trì căn chỉnh vị trí chính xác. Nó tính toán giá trị của các điểm lấy mẫu bằng cách sử dụng nội suy song tuyến (bilinear interpolation) từ các điểm lưới gần kề trên bản đồ đặc trưng, loại bỏ hoàn toàn việc lượng tử hóa các tọa độ liên quan. Sự thay đổi đường như nhỏ này đã mang lại tác động lớn, cải thiện độ chính xác mặt nạ lên đến 10% đến 50%, đặc biệt quan trọng dưới các tiêu chí định vị nghiêm ngặt.[17]

**Decoupling Mask Prediction:** Một yếu tố kỹ thuật khác là việc tách rời (decoupling) dự đoán mặt nạ và dự đoán lớp. Thay vì sử dụng softmax và loss đa thức (multinomial loss) khiến các mặt nạ cạnh tranh lẫn nhau (phổ biến trong semantic segmentation), Mask R-CNN sử dụng sigmoid per-pixel và định nghĩa loss mặt nạ ( $L_{mask}$ ) chỉ trên mặt nạ lớp đúng của RoI đó. Cách tiếp cận này đã được chứng minh là chìa khóa để đạt được kết quả phân đoạn thực thể tốt.[17]

Sự thành công của Mask R-CNN chứng minh rằng các tác vụ định vị mật độ cao đòi hỏi mức độ chính xác không gian cao hơn nhiều so với các tác vụ phát hiện hộp giới hạn đơn thuần. Khung làm việc này cũng linh hoạt, dễ dàng tổng quát hóa sang các tác vụ khác như ước tính tư thế người (person keypoint detection) bằng cách xem mỗi điểm khóa (keypoint) là một mặt nạ nhị phân one-hot.

#### *1.1.5.2. Mô hình Encoder-Decoder cho Tái tạo Hình ảnh Chất lượng cao (U-Net)*

Các kiến trúc Encoder-Decoder, nổi bật là U-Net, là nền tảng cho nhiều bài toán dịch ảnh (image-to-image translation) và phân đoạn mật độ cao. Phần encoder (sử dụng các lớp tích chập và pooling) nén ảnh gốc thành một vector đặc trưng ẩn, trong khi phần decoder tái tạo vector ẩn đó thành ảnh đầu ra mong muốn.[47]



Hình 1.5. Mô hình U-Net

Trong phần decoder, việc tái tạo độ phân giải không gian thường được thực hiện bằng cách sử dụng Transposed Convolution (còn gọi là deconvolution hoặc tích chập chuyển vị). Mặc dù Transposed Convolution là một lớp học được, nó có một hạn chế kỹ thuật lớn: dễ dẫn đến sự chồng lấp không đều (uneven overlap), tạo ra các tạo phẩm (artifacts) dưới dạng mẫu kẻ ô (checkerboard-like patterns) trên đầu ra.[47]

Để giải quyết vấn đề này, mô hình U-NetPlus, được thiết kế cho phân đoạn công cụ phẫu thuật, đã giới thiệu một sự thay thế quan trọng. U-NetPlus sử dụng các encoder tiền huấn luyện (pre-trained), cụ thể là VGG-11 hoặc VGG-16, để tăng tốc độ hội tụ và cải thiện kết quả.

Quan trọng hơn, trong phần decoder, U-NetPlus đã thay thế hoàn toàn Transposed Convolution bằng Nearest-Neighbor (NN) interpolation. Thao tác NN upsampling, theo sau là hai lớp tích chập, đã loại bỏ hiệu quả các tạo phẩm do Transposed Convolution gây ra, đồng thời giảm số lượng tham số của mô hình. Việc các nhà nghiên cứu lựa chọn một phương pháp nội suy không học được (NN) thay vì một lớp học được (Transposed Conv) để ổn định đầu ra decoder cho thấy rằng trong các miền ứng dụng nhạy cảm như y học, độ trung thực hình ảnh và tính ổn định đầu ra có thể được ưu tiên hơn khả năng học tối đa của mạng.[47]

Bảng 1.2. Các Kỹ thuật Upsampling trong Decoder của CNN

Kỹ thuật	Đặc điểm	Ưu điểm Kỹ thuật	Hạn chế Kỹ thuật Chính
Transposed Convolution	Học được (Learnable)	Linh hoạt, có thể học được bộ lọc tái tạo tối ưu.	Dễ gây ra tạo phẩm kiểu checkerboard do uneven overlap.
Nearest-Neighbor Interpolation	Không học được (Non-Learnable)	Giảm thiểu tạo phẩm, giảm số lượng tham số, ổn định đầu ra.	Độ mịn không gian có thể thấp, không học được sự tinh chỉnh chi tiết.

### 1.1.6. CNNs trong Dịch Ảnh và Thích ứng Miền

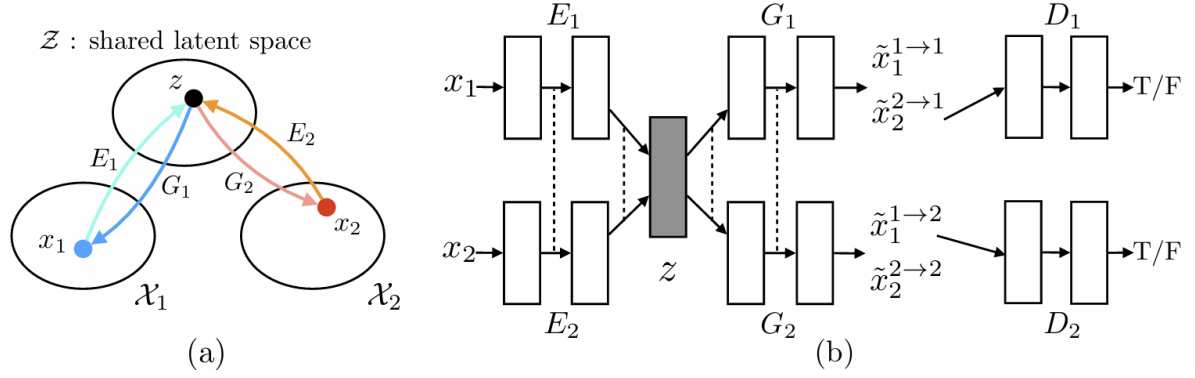
CNN không chỉ được sử dụng cho phân loại và phân đoạn, mà còn là nền tảng của các mô hình encoder-decoder dùng trong các bài toán dịch ảnh giữa các miền khác nhau (image-to-image translation).

#### 1.1.6.1. Dịch Ảnh Không Giám sát và Giả định Không gian Ẩn Chung

Bài toán dịch ảnh không giám sát (Unsupervised Image-to-image translation) là một thách thức lớn vì chỉ có các bộ dữ liệu biên độc lập ( $X_1, X_2$ ). [29] Sự thiếu vắng các cặp ảnh tương ứng khiến việc suy luận về phân phối chung giữa hai miền trở nên không giải được (ill-posed problem) nếu không có các giả định bổ sung.

Để giải quyết vấn đề này, khung làm việc UNIT (Unsupervised Image-to-image Translation) đã được đề xuất, kết hợp Variational Autoencoders (VAEs) và Generative Adversarial Networks (GANs), dựa trên Giả định Không gian Ẩn Chung (Shared-Latent Space Assumption).

Giả định này khẳng định rằng một cặp ảnh tương ứng từ hai miền khác nhau ( $x_1, x_2$ ) có thể được ánh xạ tới cùng một biểu diễn ẩn ( $z$ ) trong một không gian ẩn chung ( $Z$ ). Về mặt toán học, điều này ngụ ý rằng tồn tại các hàm mã hóa  $E_1^*, E_2^*$  và hàm sinh  $G_1^*, G_2^*$  sao cho  $z = E_1^*(x_1) = E_2^*(x_2)$  và  $x_1 = G_1^*(z)$ ,  $x_2 = G_2^*(z)$ . [29]



Hình 1.6. Không gian tiềm ẩn chia sẻ và kiến trúc mạng UNIT.

**(a) Giả định không gian tiềm ẩn chia sẻ.** Chúng ta giả định một cặp hình ảnh tương ứng  $(x_1, x_2)$  trong hai miền khác nhau  $\mathcal{X}_1$  và  $\mathcal{X}_2$  có thể được ánh xạ tới cùng một mã tiềm ẩn  $z$  trong không gian tiềm ẩn chia sẻ  $\mathcal{Z}$ .  $E_1$  và  $E_2$  là hai hàm mã hóa, ánh xạ hình ảnh tới các mã tiềm ẩn.  $G_1$  và  $G_2$  là hai hàm tạo sinh, ánh xạ các mã tiềm ẩn trở lại hình ảnh. **(b) Khung công việc UNIT được đề xuất.** Chúng ta biểu diễn  $E_1, E_2, G_1$  và  $G_2$  bằng cách sử dụng Mạng nơ-ron tích chập (CNN) và thực hiện giả định không gian tiềm ẩn chia sẻ bằng cách sử dụng ràng buộc chia sẻ trọng số, nơi trọng số kết nối của vài lớp đầu tiên (các lớp cấp cao) trong  $E_1$  và  $E_2$  được gắn kết với nhau (minh họa bằng các đường đứt nét) và trọng số kết nối của vài lớp đầu tiên trong  $G_1$  và  $G_2$  cũng được gắn kết với nhau. Ở đây,  $\tilde{x}_1^{1 \rightarrow 1}$  và  $\tilde{x}_2^{2 \rightarrow 1}$  là các hình ảnh tự tái tạo, và  $\tilde{x}_1^{1 \rightarrow 2}$  và  $\tilde{x}_2^{2 \rightarrow 1}$  là các hình ảnh đã được chuyển miền.  $D_1$  và  $D_2$  là các bộ phân biệt đối kháng (adversarial discriminator) cho các miền tương ứng, có nhiệm vụ đánh giá xem các hình ảnh đã được chuyển có thực tế hay không.

#### 1.1.6.2. Cơ chế Triển khai Không gian Ẩn Chung trong UNIT

Trong UNIT, giả định Shared-Latent Space được triển khai thông qua ràng buộc chia sẻ trọng số (weight-sharing constraint) giữa các mạng con.

- Encoders ( $E_1, E_2$ ): Các trọng số của vài lớp cuối cùng (các lớp cấp cao) trong hai Encoders được ràng buộc chia sẻ. Những lớp này có trách nhiệm trích xuất các đặc trưng biểu diễn cấp cao.[29]
- Generators ( $G_1, G_2$ ): Tương tự, các trọng số của vài lớp đầu tiên (các lớp cấp cao) trong hai Generators được ràng buộc chia sẻ.[29]



Ràng buộc chia sẻ trọng số này là cơ chế vật lý hóa giả định không gian ẩn chung, buộc các Encoders phải ánh xạ các ảnh tương ứng vào cùng một mã ẩn.

Giả định không gian ẩn chung cũng ngầm định yêu cầu tính nhất quán vòng lặp (cycle-consistency constraint), đảm bảo rằng một hình ảnh được dịch từ miền  $X_1$  sang  $X_2$  và sau đó được dịch trở lại  $X_1$  sẽ gần giống với hình ảnh gốc ( $x_1 = G_1^*(E_2^*(G_2^*(E_1^*(x_1))))$ ).[29] Sự kết hợp giữa VAEs (để tái tạo ảnh), GANs (để đảm bảo ảnh dịch là thực tế) và ràng buộc chia sẻ trọng số (để liên kết các miền) cho phép CNN hoạt động như một nền tảng trích xuất và tái tạo nội dung, thành công trong nhiều tác vụ dịch ảnh phức tạp.

#### ***1.1.6.3. Thích ứng Miền (Domain Adaptation) thông qua Style Transfer***

Khả năng học đặc trưng trừu tượng của CNN cũng được sử dụng để giải quyết vấn đề thích ứng miền (domain adaptation), đặc biệt khi có sự dịch chuyển miền (domain shift) — tức là khi dữ liệu huấn luyện (nguồn) và dữ liệu thử nghiệm (đích) đến từ các phân phối khác nhau (ví dụ: ảnh chụp ban ngày so với ban đêm, hoặc ảnh không sương mù so với có sương mù).[48]

Sự dịch chuyển miền gây ra sự suy giảm hiệu suất đáng kể. Ví dụ, trong bài toán phân đoạn ảnh trên không (aerial image segmentation), sự dịch chuyển miền gây ra mức giảm trung bình  $-5.22\%$  mIoU trên bộ dữ liệu Potsdam.[48] Để chống lại sự suy giảm này, một mô hình chuyển phong cách trong không gian ẩn (latent space style transfer model) đã được đề xuất.[48] Mô hình này sử dụng các biểu diễn đặc trưng ẩn của CNN để tạo ra các phiên bản dữ liệu tổng hợp với phong cách miền đích (ví dụ: thêm sương mù vào ảnh rõ nét).

Cách tiếp cận này loại bỏ nhu cầu ghi chú bổ sung (annotation) trên dữ liệu miền dịch chuyển.[48] Bằng cách áp dụng phương pháp này, hiệu suất trên miền dịch chuyển đã được cải thiện đáng kể (ví dụ: tăng  $+3.97\%$  mIoU trên Potsdam), chứng minh rằng việc sử dụng CNN để trích xuất và thao tác các đặc trưng phong cách trừu tượng là một chiến lược hiệu quả để nâng cao tính tổng quát của mô hình trong môi trường thực tế.

#### ***1.1.7. Đánh giá và Xu hướng Tương lai của Kiến trúc Tích chập***

##### ***1.1.7.1. Vị thế của CNN và Sự Cạnh tranh với Vision Transformer***

Trong bối cảnh Vision Transformer (ViT) đang nổi lên như một đối thủ cạnh tranh mạnh mẽ, CNNs (như ResNet và ConvNeXt) vẫn duy trì vị thế là các mô hình nền tảng trong nghiên cứu thị giác máy tính. [26]

Phân tích cho thấy CNNs vẫn thể hiện ưu thế hoặc hiệu suất tương đương với ViT,

đặc biệt trong chế độ học ít mẫu (low-data few-shot regime) trong quá trình học chuyển giao (transfer learning). Điều này được cho là do CNNs sở hữu một thiên kiến quy nạp cục bộ (local inductive bias) vốn có, giúp chúng học ổn định hơn và cần ít dữ liệu hơn để khái quát hóa các mối quan hệ không gian cơ bản.

Xu hướng nghiên cứu hiện đại đang hướng tới các mô hình hỗn hợp (Hybrid), kết hợp các thành phần của CNN và Transformer để tận dụng ưu điểm của cả hai. Ví dụ, kiến trúc CoAtNet kết hợp các khối tích chập của CNN với cơ chế tự chú ý (self-attention) của Transformer, đạt hiệu suất tối ưu bằng cách cân bằng giữa việc xử lý thông tin cục bộ và bắt ngữ cảnh toàn cục.

#### *1.1.7.2. Ứng dụng trong Chẩn đoán Y tế và Ensemble Learning*

Lĩnh vực chẩn đoán y tế, như phân tích X-quang ngực, là một ứng dụng quan trọng đòi hỏi độ tin cậy và độ chính xác cao. Việc sử dụng các kỹ thuật học sâu (bao gồm các CNNs tiền huấn luyện, Transformer và mô hình Hybrid) đã được chứng minh là có ý nghĩa quan trọng trong việc tự động chẩn đoán các bệnh lý lồng ngực.

Trong các lĩnh vực có tính quyết định cao như y học, việc giảm thiểu rủi ro và tăng độ tin cậy là tối quan trọng. Nghiên cứu đã chứng minh rằng kỹ thuật tập hợp học sâu (Ensemble Deep Learning) có thể cải thiện đáng kể hiệu suất chẩn đoán.[26] Bằng cách kết hợp dự đoán của nhiều mô hình đã huấn luyện (bao gồm cả CNNs truyền thống và mô hình hybrid như CoAtNet) thông qua trung bình có trọng số, hiệu suất đã được cải thiện, đạt AUROC 85.4% trên bộ dữ liệu ChestX-ray14, vượt qua các phương pháp dẫn đầu khác.[26] Điều này khẳng định rằng việc tổng hợp kết quả từ nhiều kiến trúc là một chiến lược quan trọng để tăng độ chính xác và độ tin cậy trong các ứng dụng thực tiễn.

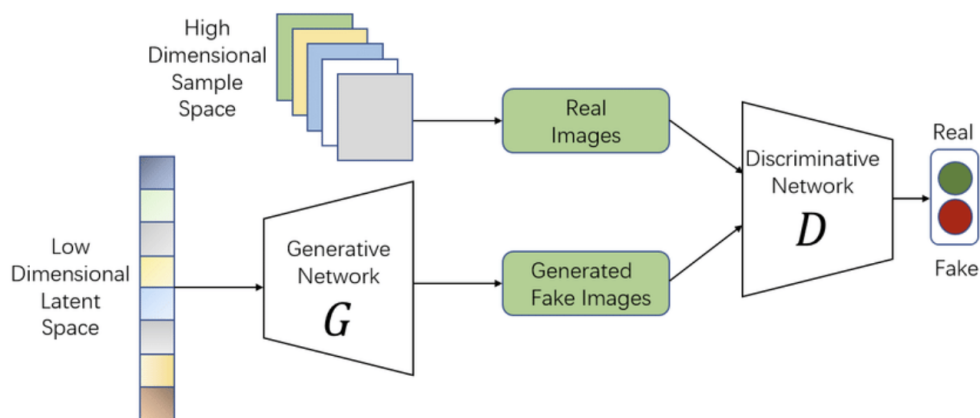
## 1.2. Kiến trúc Mạng đối nghịch tạo sinh

Trong thập kỷ qua, Trí tuệ Nhân tạo (AI) đã chuyển dịch mạnh mẽ từ khả năng phân tích sang sáng tạo, với dấu mốc quan trọng là sự ra đời của Mạng Đối nghịch Tạo sinh (GAN) do Ian Goodfellow và cộng sự đề xuất năm 2014. Khác với các mô hình tạo sinh trước đây vốn dựa trên các phương pháp xác suất phức tạp như MCMC hay Deep Belief Networks, GAN đưa ra cách tiếp cận mới bằng trò chơi đối kháng giữa hai mạng nơ-ron, tận dụng lan truyền ngược để huấn luyện trực tiếp và tạo ra dữ liệu có độ trung thực cao.[14]

### 1.2.1. Nguyên lý Hoạt động và Cơ sở Lý thuyết

#### 1.2.1.1. Khung Làm việc Đối kháng

GAN (Generative Adversarial Networks) là một hệ thống gồm hai mạng nơ-ron được huấn luyện đồng thời thông qua một trò chơi có tổng bằng không. Trong đó mạng Tạo sinh (Generator -  $G$ ) đóng vai trò như một “kẻ làm giả”.  $G$  nhận đầu vào là một vector nhiễu ngẫu nhiên  $z$  từ một không gian tiềm ẩn (latent space) với phân phối  $p_z(z)$ , để tạo ra phân phối chuẩn hoặc giống như phân phối dữ liệu thật  $p_{data}$ .  $G$  cố gắng tạo ra những mẫu gần khớp với  $G(z)$ . Mục tiêu của  $G$  là đánh lừa  $D$ , nghĩa là tạo ra những mẫu mà  $D$  không thể phân biệt được với dữ liệu thật  $p_{data}$ . Ngược lại mạng Phân biệt (Discriminator -  $D$ ) đóng vai trò như một “cảnh sát” hoặc chuyên gia giám định.  $D$  là một bộ phân loại nhị phân nhận đầu vào là dữ liệu  $x$  và xuất ra một giá trị thể hiện hướng  $D(x; \theta_D)$  biểu thị xác suất dữ liệu đó là dữ liệu thật ( $p_{data}$ ) thay vì từ  $G$ . [14]



Hình 1.7. Hình ảnh của GAN

### 1.2.1.2. Trò chơi Minimax và Hàm Mục tiêu

Quá trình huấn luyện GAN được mô hình hóa dưới dạng một bài toán tối ưu hóa Minimax, trong đó  $D$  cố gắng tối đa hóa khả năng phân biệt đúng, còn  $G$  cố gắng tối thiểu hóa khả năng bị phát hiện (tức là tối đa hóa sai sót của  $D$ ). Hàm giá trị  $V(D, G)$  được định nghĩa như sau:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

Trong đó số hạng đầu tiên  $\mathbb{E}_{x \sim p_{\text{data}}(x)}$  khuyến khích  $D$  gán xác suất cao cho dữ liệu thật. Số hạng thứ hai  $\mathbb{E}_{z \sim p_z(z)}$  khuyến khích  $D$  gán xác suất thấp cho dữ liệu giả  $G(z)$ . [14]

### 1.2.1.3. Phân tích Điểm Cân bằng Nash và Discriminator Tối ưu

Về mặt lý thuyết, mục tiêu cuối cùng của quá trình này là đạt được Điểm cân bằng Nash (Nash Equilibrium). Tại điểm này, không người chơi nào có thể cải thiện kết quả của mình bằng cách đơn phương thay đổi chiến lược. [22] Đối với GAN, Discriminator không thể phân biệt được ảnh thật và ảnh giả tốt hơn việc đoán ngẫu nhiên (xác suất  $D(x) = 0.5$  cho mọi  $x$ ). Phân phối của Generator  $p_g$  hoàn toàn trùng khớp với phân phối dữ liệu thật  $p_{\text{data}}$ .

Goodfellow et al [14] đã chứng minh rằng với một  $G$  cố định, Discriminator tối ưu  $D^*$  có dạng:

$$D_G^*(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}$$

Khi thay thế  $D_G^*(x)$  trở lại hàm mục tiêu  $V(D, G)$ , bài toán tối thiểu hóa của Generator trở thành việc tối thiểu hóa khoảng cách Jensen-Shannon (JS Divergence) giữa hai phân phối:

$$C(G) = \max_D V(D, G) = -\log(4) + 2 \cdot D_{\text{JS}}(p_{\text{data}} \parallel p_g)$$

Kết quả này rất quan trọng vì nó cung cấp cơ sở lý thuyết cho thấy việc tối ưu hóa hàm đối nghịch thực chất là việc kéo phân phối sinh  $p_g$  về phía phân phối thật  $p_{\text{data}}$  theo thước đo khoảng cách Jensen-Shannon.

### 1.2.2. Thách thức trong Huấn luyện GAN

Mặc dù lý thuyết rất chặt chẽ, việc áp dụng GAN trong thực tế gặp phải những vấn đề nghiêm trọng liên quan đến tính ổn định của gradient và động lực học của tối ưu hóa. Các nghiên cứu đã chỉ ra ba vấn đề kinh điển: Biến mất Gradient (Vanishing Gradients), Sụp đổ Mode (Mode Collapse), và Không hội tụ (Non-convergence).

### 1.2.2.1. Vấn đề Biến mất Gradient (Vanishing Gradients)

Vấn đề này xuất phát trực tiếp từ bản chất của khoảng cách Jensen-Shannon (JS) được sử dụng trong hàm loss gốc. Phân tích Toán học: Khi  $p_{data}$  và  $p_g$  có giá đỡ (support) rời rạc hoặc nằm trong các không gian con chiều thấp không giao nhau (điều rất phổ biến trong không gian dữ liệu nhiều chiều như ảnh), khoảng cách JS giữa chúng là một hằng số ( $\log 2$ ). Hệ quả: Vì hàm loss là hằng số, đạo hàm (gradient) của nó bằng 0. Khi Discriminator trở nên quá chính xác (đạt tiệm cận tối ưu), nó phân tách hoàn hảo ảnh thật và ảnh giả. Lúc này, hàm loss của Generator bão hòa, và  $G$  không nhận được bất kỳ tín hiệu gradient nào để cập nhật trọng số.  $G$  "không biết" hướng nào để di chuyển nhằm cải thiện chất lượng ảnh. Goodfellow đã đề xuất một giải pháp tình thế là thay đổi hàm mục tiêu của  $G$  từ  $\min \log(1 - D(G(z)))$  sang  $\max \log D(G(z))$  (gọi là hàm non-saturating loss). Mặc dù giải quyết được vấn đề biến mất gradient ban đầu, hàm này lại gây ra sự dao động mạnh và không ổn định trong giai đoạn sau của quá trình huấn luyện.[14]

### 1.2.2.2. Hiện tượng Sụp đổ Mode (Mode Collapse)

Mode collapse là một trong những thất bại phổ biến và khó chịu nhất của GAN. Nó xảy ra khi Generator học được cách tạo ra một hoặc một vài mẫu ảnh (modes) mà Discriminator tin là thật, và sau đó liên tục sinh ra các mẫu này bất kể đầu vào  $z$  là gì.[40]

Cơ chế của hiện tượng này thường xảy ra khi Generator quá "tham lam" tối ưu hóa cho Discriminator hiện tại mà không quan tâm đến sự đa dạng. Ví dụ, nếu  $D$  yếu kém trong việc phát hiện các chữ số "1",  $G$  sẽ dồn toàn bộ trọng số để chỉ tạo ra số "1". Vòng lặp không hồi kết, khi  $D$  học được rằng số "1" là giả,  $G$  sẽ chuyển sang một mode khác, ví dụ số "8". Hai mạng cứ thế đuổi bắt nhau quanh các mode mà không bao giờ hội tụ về một phân phối đa dạng bao phủ toàn bộ dữ liệu. Điều này dẫn đến kết quả đầu ra thiếu tính sáng tạo và lặp lại.

### 1.2.2.3. Sự Không Hội tụ (Non-Convergence)

Việc tìm kiếm điểm cân bằng Nash trong trò chơi liên tục, nhiều chiều và không lồi (non-convex) là cực kỳ khó khăn. Các thuật toán tối ưu dựa trên gradient descent được thiết kế để tìm cực tiểu cục bộ của hàm chi phí, chứ không phải điểm yên ngựa (saddle point) cần thiết cho GAN.[40] Nhiều nghiên cứu lý thuyết [18] chỉ ra rằng động lực học của gradient descent trong các trò chơi song tuyến tính (bilinear games) thường dẫn đến các quỹ đạo xoay tròn (rotational dynamics) thay vì hội tụ vào tâm. Điều này giải thích tại sao loss của  $G$  và  $D$  thường dao động mạnh và không bao giờ ổn định.

Bảng 1.3. Tóm tắt Các Chế độ Thất bại của GAN

Vấn đề	Biểu hiện	Nguyên nhân Toán học
<b>Vanishing Gradients</b>	$G$ ngừng học, loss không đổi	Discriminator quá mạnh, JS divergence bảo toàn độ đo.
<b>Mode Collapse</b>	Ảnh sinh ra giống hệt nhau, thiếu đa dạng	$G$ tối ưu hóa cục bộ (greedy), $D$ không bắt buộc tính đa dạng.
<b>Non-Convergence</b>	Loss dao động mạnh, không ổn định	Quá trình học lặp lại không quay quanh điểm cân bằng Nash, không ổn định.

### 1.2.3. Các Giải pháp về Hàm Mục tiêu và Toán học

#### 1.2.3.1. Wasserstein GAN (WGAN) và Khoảng cách Earth-Mover

Công thức Wasserstein distance:

$$W(p_{\text{data}}, p_g) = \inf_{\gamma \in \Pi(p_{\text{data}}, p_g)} \mathbb{E}_{(x,y) \sim \gamma} [\|x - y\|]$$

Ưu điểm vượt trội Gradient liên tục khoảng cách Wasserstein liên tục và khả vi hầu khắp mọi nơi, ngay cả khi hai phân phối  $p_{\text{data}}$  và  $p_g$  không giao nhau. Điều này cung cấp các gradient có ý nghĩa và mượt mà cho Generator trong suốt quá trình huấn luyện, loại bỏ hoàn toàn vấn đề biến mất gradient. Tương quan với chất lượng, giá trị loss của WGAN tương quan tốt với chất lượng ảnh sinh ra (loss thấp hơn đồng nghĩa ảnh tốt hơn), điều mà GAN gốc không làm được.

Để tính toán được khoảng cách này, sử dụng tính đối ngẫu Kantorovich-Rubinstein, yêu cầu Discriminator (lúc này gọi là Critic) phải thỏa mãn ràng buộc 1-Lipschitz continuity:  $|f(x_1) - f(x_2)| \leq |x_1 - x_2|$ .

#### 1.2.3.2. WGAN-GP: Từ Weight Clipping đến Gradient Penalty

Ban đầu, để thỏa mãn ràng buộc Lipschitz, WGAN sử dụng Weight Clipping (cắt trọng số) để ép các tham số mạng nằm trong khoảng  $[-c, c]$ . Tuy nhiên, phương pháp này quá thô bạo, dẫn đến việc mạng hoặc sử dụng không hết khả năng (capacity underuse) hoặc gây bùng nổ gradient.

Cải tiến mô hình này bằng phương pháp Gradient Penalty (WGAN-GP). Thay vì

cắt trọng số, họ thêm một số hạng phạt vào hàm loss để khuyến khích norm của gradient của Critic xấp xỉ bằng 1 trên các điểm mẫu nội suy  $\hat{x}$ :

$$L_{\text{WGAN-GP}} = \underbrace{\mathbb{E}_{\tilde{x} \sim p_g} - \mathbb{E}_{x \sim p_{\text{data}}}}_{\text{Original WGAN Loss}} + \underbrace{\lambda \mathbb{E}_{\hat{x}}}_{\text{Gradient Penalty}}$$

Kỹ thuật này đã giúp ổn định đáng kể quá trình huấn luyện và cho phép huấn luyện các kiến trúc phức tạp hơn (như ResNet) mà không bị mất ổn định.

#### 1.2.3.3. Spectral Normalization (SN-GAN)

Một phương pháp khác để kiểm soát tính ổn định của Discriminator là Spectral Normalization. Thay vì phạt gradient (tốn kém tính toán), SN chuẩn hóa trực tiếp ma trận trọng số  $W$  của mỗi lớp mạng bằng cách chia cho giá trị kỳ dị lớn nhất (spectral norm  $\sigma(W)$ ) của nó:

$$W_{SN} = \frac{W}{\sigma(W)}$$

Phương pháp này đảm bảo rằng hằng số Lipschitz của toàn bộ mạng bị chặn bởi 1 một cách toàn cục. SN rất hiệu quả về mặt tính toán và đã trở thành tiêu chuẩn cho hầu hết các kiến trúc GAN hiện đại (như SAGAN, BigGAN) vì nó ngăn chặn sự bùng nổ gradient mà không làm giảm khả năng biểu diễn của mạng quá nhiều.

### 1.2.4. Các Kỹ thuật Ổn định Huấn luyện

Bên cạnh việc thay đổi hàm loss, các chiến lược huấn luyện cũng đóng vai trò quan trọng trong việc đạt được sự hội tụ.

#### 1.2.4.1. Minibatch Discrimination

Để giải quyết vấn đề mode collapse, kỹ thuật Minibatch Discrimination cho phép Discriminator xem xét mối quan hệ giữa các mẫu trong cùng một batch thay vì xử lý chúng độc lập. Cơ chế mạng tính toán tổng kê về khoảng cách (L1 distance) giữa các đặc trưng của các mẫu trong batch. Nếu Generator bị sụp đổ mode, các mẫu sinh ra sẽ rất giống nhau, dẫn đến khoảng cách giữa chúng rất nhỏ. Tác động discriminator sử dụng thông tin này để dễ dàng phát hiện ra các batch "giả" có độ đa dạng thấp, từ đó buộc Generator phải tạo ra các mẫu đa dạng hơn để đánh lừa Discriminator.

#### 1.2.4.2. Two Time-Scale Update Rule (TTUR)

Quy tắc cập nhật hai thang thời gian (TTUR), trong đó Generator và Discriminator có tốc độ học (learning rate) riêng biệt. Thông thường, Discriminator được gán tốc độ

học cao hơn để nó có thể hội tụ nhanh chóng về cực tiểu cục bộ mỗi khi Generator thay đổi phân phối.

Ý nghĩa lý thuyết: TTUR đảm bảo rằng quá trình huấn luyện sẽ hội tụ về điểm cân bằng Nash ổn định cục bộ, điều mà tốc độ học đồng nhất thường thất bại. Đây cũng là bài báo giới thiệu chỉ số đánh giá FID nổi tiếng.

### **1.2.5. Đánh giá Chất lượng**

Vì không có hàm mục tiêu tường minh để kiểm tra trên tập test, việc đánh giá GAN dựa vào các chỉ số thống kê trên tập đặc trưng.

#### **1.2.5.1. Inception Score (IS)**

IS sử dụng mạng Inception v3 để đánh giá hai tiêu chí. Một là độ sắc nét, mỗi ảnh sinh ra phải được phân loại tự tin vào một lớp cụ thể (Entropy có điều kiện thấp). Và độ đa dạng: Tập hợp các ảnh sinh ra phải bao phủ tất cả các lớp (Entropy biên cao). Công thức dựa trên độ phân kỳ KL,  $IS = \exp(\mathbb{E}_{x \sim p_g}[KL(p(y|x) \parallel p(y))])$ .

#### **1.2.5.2. Frechet Inception Distance (FID)**

FID hiện là tiêu chuẩn vàng. Nó so sánh phân phối của các vector đặc trưng (từ lớp pool3 của Inception v3) giữa ảnh thật và ảnh giả, giả định chúng tuân theo phân phối chuẩn đa biến.

$$FID = \|\mu_r - \mu_g\|^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2})$$

FID thấp hơn đồng nghĩa với việc hai phân phối gần nhau hơn. FID nhạy cảm hơn IS đối với nhiễu, mode collapse và biến dạng ảnh, cung cấp một đánh giá toàn diện hơn về chất lượng.

## **1.3. Các mô hình GAN tiêu biểu cho chuyển đổi ảnh sang phong cách Anime**

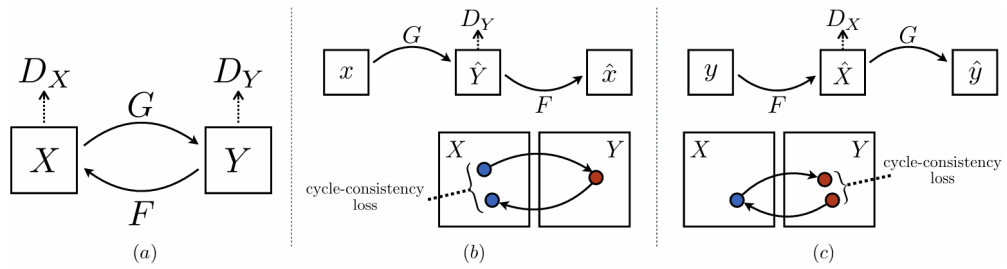
Bài toán chuyển ảnh người thật sang tranh vẽ phong cách anime có tính đặc thù cao, đòi hỏi sự kết hợp giữa bảo toàn nội dung gốc và thể hiện đúng phong cách anime (với đặc trưng là nét viền rõ ràng, màu phẳng, chi tiết được tối giản).

### **1.3.1. CycleGAN và các biến thể (chuyển đổi với dữ liệu không cặp)**

Kiến trúc CycleGAN (2017): CycleGAN do Zhu et al. đề xuất là mốc quan trọng cho dịch ảnh không cần dữ liệu ghép cặp [60]. CycleGAN bao gồm hai bộ Generator song song:  $G : X \rightarrow Y$  (chuyển ảnh từ miền nguồn X sang miền đích Y, ví dụ ảnh thật sang ảnh anime) và  $F : Y \rightarrow X$  (chuyển ngược lại từ Y về X), cùng với hai Discriminator



$D_X$  và  $D_Y$  học phân biệt ảnh thuộc từng miền. Ý tưởng cốt lõi của CycleGAN là ràng buộc tính nhất quán vòng lặp (cycle consistency): một ảnh nguồn  $x$  sau khi qua biến đổi  $G(x)$  rồi chuyển ngược  $F(G(x))$  phải gần giống chính  $x$  ban đầu [60]. Nhờ cycle consistency loss ( $\mathcal{L}_{cyc}$ ), mô hình tránh được những phép biến đổi quá tùy ý, giúp bảo toàn nội dung gốc ở mức độ cao [60]. Đồng thời, mỗi cặp  $G-D_Y$  và  $F-D_X$  hình thành hai chu trình GAN truyền thống để đảm bảo ảnh đầu ra khó phân biệt với ảnh thật miền tương ứng. Cách tiếp cận này cho phép huấn luyện với hai tập ảnh không ghép cặp (ví dụ một tập ảnh chân dung người thật và một tập tranh chân dung anime) mà vẫn học được phép biến đổi giữa chúng.



Hình 1.8. Kiến trúc của CycleGAN

Hình 1.8a Mô hình của CycleGAN chứa hai hàm ánh xạ  $G : X \rightarrow Y$  và  $F : Y \rightarrow X$ , cùng với các bộ phân biệt đối kháng  $D_Y$  và  $D_X$ .  $D_Y$  khuyến khích  $G$  dịch hình ảnh từ miền  $X$  sang đầu ra không thể phân biệt được với miền  $Y$ , và ngược lại đối với  $D_X$  và  $F$ . Để điều chỉnh thêm các ánh xạ, chúng tôi giới thiệu hai tổn thất nhất quán chu trình (cycle consistency losses) nắm bắt trực giác rằng nếu chúng ta dịch từ một miền sang miền khác và sau đó quay trở lại, chúng ta sẽ thu được hình ảnh ban đầu. Hình 1.8b tổn thất nhất quán chu trình xuôi (forward cycle-consistency loss):  $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$ , và Hình 1.8c tổn thất nhất quán chu trình ngược (backward cycle-consistency loss):  $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$ .

Ưu điểm: CycleGAN rất linh hoạt do không cần các cặp ảnh trước-sau, nên có thể áp dụng cho nhiều bài toán chuyển kiểu dáng, chất liệu (ví dụ: ngựa  $\Leftrightarrow$  zebra, phong cảnh  $\Leftrightarrow$  tranh Monet) [60]. Tính nhất quán vòng lặp giúp nội dung chính của ảnh (hình dáng khuôn mặt, bố cục) được giữ lại khá tốt sau khi chuyển đổi.

Nhược điểm: Mặc dù bảo toàn được cấu trúc tổng quát, CycleGAN vẫn khó đảm bảo chi tiết định danh (identity) của đối tượng, nhất là với khuôn mặt người – đôi khi các đặc trưng như ánh mắt, nụ cười có thể bị biến đổi hoặc mất đi trong ảnh anime kết

quả. Ngoài ra, CycleGAN có thể sinh ra các artifact (tạo tác) không mong muốn, ví dụ như nhiễu màu hoặc đường viền thừa, do Generator cố gắng tối ưu adversarial loss mà không có ràng buộc trực tiếp về đặc trưng phong cách cụ thể. Nhiều biến thể sau này đã cải thiện điểm này: chẳng hạn U-GAT-IT (Kim et al., 2020) bổ sung module chú ý (attention) giúp tập trung vào những vùng quan trọng trên khuôn mặt và sử dụng cơ chế Adaptive Layer-Instance Normalization để giữ tốt hơn đặc điểm gốc. Kết quả U-GAT-IT cho thấy việc thêm thành phần nhận dạng đặc trưng khuôn mặt (như identity loss hoặc facial landmark loss) giúp khuôn mặt anime giữ được nét của ảnh gốc hơn so với CycleGAN thuần túy. Dù vậy, các phương pháp dựa trên cycle-consistency đôi khi vẫn tạo ra nhiễu hoặc vùng màu không tự nhiên, đặc biệt khi có sự chênh lệch lớn giữa hai miền ảnh (ví dụ màu da người và màu da nhân vật anime)

### ***1.3.2. StyleGAN và phương pháp dựa trên GAN inversion (kiểm soát chất lượng cao)***

Kiến trúc StyleGAN (Karras et al., 2019): StyleGAN đại diện cho hướng tiếp cận khác, tập trung vào sinh ảnh chất lượng cao và khả năng điều khiển phong cách linh hoạt. Khác với mô hình GAN truyền thống, StyleGAN đưa ra kiến trúc Generator mới gồm hai thành phần: Mạng ánh xạ (Mapping Network) và Mạng tổng hợp (Synthesis Network) [23]. Mạng ánh xạ lấy một vector ngẫu nhiên  $z$  và ánh xạ nó thành vector trung gian  $w$  trong không gian tiềm ẩn đã được phân tách (disentangled latent space); sau đó  $w$  được đưa vào từng tầng của mạng tổng hợp qua cơ chế Adaptive Instance Normalization (AdaIN) để điều chỉnh “phong cách” ở các mức độ khác nhau [23]. Nhờ đó, StyleGAN có khả năng điều chỉnh các thuộc tính ảnh (như độ thô của hình, màu sắc chi tiết) ở từng tầng, cho phép sinh ra ảnh có độ chân thực cao và tách biệt được các yếu tố như bố cục và chi tiết bề mặt. Kết hợp thêm việc đưa nhiễu ngẫu nhiên ở các tầng để tạo tiểu tiết (ví dụ tóc, tàn nhang), StyleGAN đã chứng tỏ khả năng sinh ảnh chân dung người đáng kinh ngạc, khó phân biệt với ảnh chụp thật [23]. Nhiều người đã tận dụng mô hình này để huấn luyện trên các tập dữ liệu mới như chân dung nhân vật hoạt hình, tranh ukiyo-e hay Pokémon, và thu được kết quả ấn tượng về độ phân giải và tính thẩm mỹ.

Chuyển ảnh thực sang anime qua StyleGAN: Một cách ứng dụng StyleGAN cho bài toán chuyển phong cách là sử dụng kỹ thuật GAN Inversion – chiếu một ảnh thực bất kỳ vào không gian tiềm ẩn của StyleGAN (đã huấn luyện trên ảnh anime) rồi dùng generator tạo ra ảnh tương ứng trong phong cách anime. Cụ thể, ta có thể huấn luyện trước một StyleGAN trên tập ảnh anime chất lượng cao (ví dụ các khuôn mặt anime độ

phân giải cao). Sau đó, với một ảnh chân dung thực, ta tìm vector  $w$  trong không gian phong cách của mạng sao cho ảnh tái tạo  $G(w)$  giống với ảnh gốc nhất (thông qua tối ưu hóa hàm mất mát tái tạo hoặc huấn luyện một Encoder để dự đoán  $w$  trực tiếp). Quá trình này gọi là StyleGAN inversion – tức “phục hồi” ảnh vào latent của StyleGAN. Khi đó, ta có thể kết hợp vector latent thu được với các vector phong cách anime mong muốn, hoặc tinh chỉnh nó trong không gian  $W$  hoặc  $W^+$ , để tạo ra ảnh cuối cùng vừa mang nội dung gốc vừa phủ phong cách anime một cách tự nhiên. Ưu điểm của phương pháp dựa trên StyleGAN là ảnh đầu ra thường có độ phân giải cao và chi tiết mượt mà, thống nhất, nhờ khả năng sinh ảnh ưu việt của StyleGAN. Đồng thời, do StyleGAN học được sự phân tách giữa nội dung và phong cách, ta có thể dễ dàng điều chỉnh mức độ giữ lại đặc điểm gốc: ví dụ kỹ thuật style mixing cho phép kết hợp phần thô (cấu trúc khuôn mặt) của ảnh gốc với phần tinh (màu sắc, nét vẽ) của phong cách anime. Thực tế, các nghiên cứu gần đây cho thấy phương pháp này giúp bảo toàn danh tính khuôn mặt tốt hơn so với các GAN truyền thống: ảnh anime tạo ra vẫn nhận ra được “ai” trong ảnh gốc, đồng thời mang phong cách vẽ rõ nét.

Nhược điểm: Đổi lại, phương pháp dùng StyleGAN và GAN inversion có nhược điểm là quy trình thực hiện khá phức tạp và tốn kém tài nguyên. Việc huấn luyện StyleGAN trên dữ liệu anime chất lượng cao đòi hỏi lượng lớn ảnh và thời gian huấn luyện dài (StyleGAN thường có hàng chục triệu tham số). Quá trình invert ảnh cũng không đơn giản: tối ưu hóa latent cho từng ảnh có thể mất hàng trăm bước lặp, hoặc nếu dùng một encoder để dự đoán thì bản thân encoder đó cũng phải được huấn luyện với bộ dữ liệu phù hợp. Ngoài ra, do StyleGAN được huấn luyện theo chế độ sinh ảnh tự do, ảnh anime sinh ra có thể chưa sát với ảnh gốc về bố cục nếu quá trình inversion không tốt. Một số nghiên cứu đã mở rộng hướng này, chẳng hạn AgileGAN (Song et al., 2021) – tinh chỉnh StyleGAN bằng inversion-consistent transfer learning để chuyên cho tác vụ phong cách chân dung, hay DualStyleGAN (Yang et al., 2022) – bổ sung hai nhánh chỉnh phong cách toàn cục và cục bộ để linh hoạt điều khiển phong cách khi biến đổi ảnh chân dung. Nhìn chung, phương pháp dựa trên StyleGAN đạt chất lượng ảnh rất cao nhưng đòi hỏi nhiều bước xử lý, không “nhẹ nhàng” như các mô hình chuyển đổi trực tiếp kiểu CycleGAN.

### **1.3.3. AnimeGAN và kiến trúc Double-Tail GAN (AnimeGANv3)**

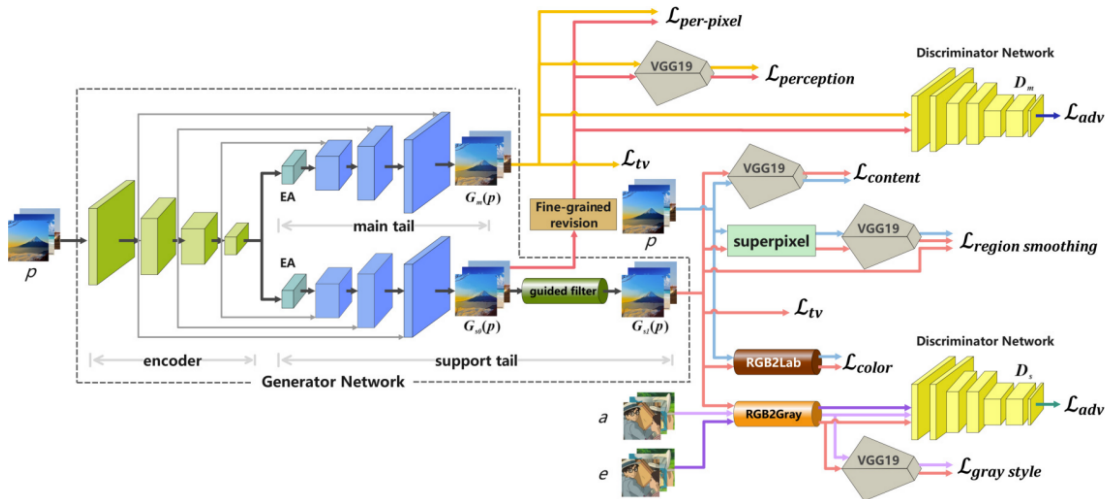
Một hướng tiếp cận nổi bật khác đến từ loạt nghiên cứu AnimeGAN của Chen et al. và Liu et al., tập trung xây dựng mô hình GAN gọn nhẹ tối ưu cho ảnh phong cách

anime. Phiên bản mới nhất AnimeGANv3 (2023) đề xuất kiến trúc Double-Tail GAN (DTGAN) – một dạng encoder-decoder dựa trên ResNet được thiết kế đặc thù cho bài toán “photo animation” (chuyển ảnh chụp thành ảnh hoạt hình).

Kiến trúc Generator “hai đuôi” (Double-tail): Điểm độc đáo của AnimeGANv3 là Generator có hai nhánh đầu ra (two output tails) [27]. Cụ thể, ảnh đầu vào sau khi qua phần encoder và các khối ResNet sẽ được chia thành hai luồng xử lý song song ở đầu ra:

- Nhánh hỗ trợ (Support tail): tạo ra một ảnh anime thô với phong cách cơ bản, nhưng có thể còn nhiều và các lỗi nhỏ.
- Nhánh chính (Main tail): nhận đầu vào chính là ảnh thô từ nhánh hỗ trợ, tiếp tục tinh chỉnh qua một số tầng mạng bổ sung để cho ra ảnh anime cuối cùng chất lượng cao.

Ý tưởng ở đây là nhánh hỗ trợ đóng vai trò hướng dẫn ban đầu về tông màu và bố cục anime, sau đó nhánh chính sẽ hiệu chỉnh chi tiết và loại bỏ nhiễu. Trong giai đoạn suy luận (inference), nhánh hỗ trợ có thể được lược bỏ, chỉ giữ lại nhánh chính – nhờ đó mô hình triển khai rất nhẹ nhưng vẫn tạo ảnh tốt. Theo báo cáo, Generator của AnimeGANv3 chỉ có 1.02 triệu tham số khi inference, nhỏ hơn nhiều so với các mô hình khác.



Hình 1.9. Sơ đồ quy trình của Double-Tail GAN (DTGAN).

**Chi tiết:**  $p$  đại diện cho ảnh chụp thể giới thực,  $a$  đại diện cho ảnh anime và  $e$  đại diện cho ảnh anime với các cạnh bị làm mờ. Các kết nối bỏ qua (skip connections) giữa bộ mã hóa (encoder) và hai đuôi (tails) được kết nối bằng phép cộng từng phần tử (element-wise addition).

Kỹ thuật chuẩn hóa mới – LADE: Để khắc phục hiện tượng phát sinh artifacts trong ảnh hoạt hình, nhóm tác giả đề xuất một phương pháp chuẩn hóa đặc thù gọi là Linearly Adaptive Denormalization (LADE) [27]. Khác với các kỹ thuật chuẩn hóa thông dụng

như BatchNorm, InstanceNorm hay LayerNorm (vốn tỏ ra chưa phù hợp cho phong cách anime do vẫn tạo ra nút vỡ hoặc nhiễu), LADE cho phép mô hình học tham số chuẩn hóa một cách linh hoạt, thích ứng tuyến tính với đặc trưng của ảnh hiện tại. Nhờ LADE, các vùng màu trong ảnh anime được làm mịn và nhất quán hơn, giảm thiểu vết nứt hay đốm loang thường thấy khi dùng IN/GN. Quan trọng là LADE được thiết kế dạng plug-and-play, có thể chèn vào thay thế các layer norm thông thường trong mô hình mà không làm tăng độ phức tạp đáng kể.

Hệ thống hàm mất mát chuyên biệt: AnimeGANv3 đưa ra hai hàm mất mát mới tối ưu cho nhiệm vụ ảnh anime: (1) Region smoothing loss – hàm mất mát làm mịn vùng, nhằm làm yếu bớt chi tiết texture phức tạp trong ảnh gốc, hướng tới các vùng màu phẳng đặc trưng của hoạt hình [27]. Loss này giúp ảnh đầu ra có độ chi tiết vừa phải, tránh hiện tượng “quá nét” hoặc lẫn nhiễu từ ảnh thật. (2) Fine-grained revision loss hàm mất mát tinh chỉnh vi mô, tập trung loại bỏ các nhiễu và tạo tác nhỏ trong ảnh anime do Generator sinh ra, đồng thời giữ sắc nét các đường biên quan trọng [27]. Sự kết hợp hai hàm mất mát này, cùng với mất mát đối nghịch truyền thống và có thể cả mất mát perceptual (tính trên đặc trưng VGG), giúp mô hình cân bằng giữa tính chân thực và tính nghệ thuật: ảnh tạo ra có màu sắc và nét vẽ mềm mại như anime, nhưng vẫn rõ ràng, không mờ nhòe. Kết quả thực nghiệm cho thấy mô hình huấn luyện với các loss trên cho ảnh chất lượng vượt trội so với các phiên bản trước đó.

Ưu điểm: AnimeGANv3 (DTGAN) kế thừa các ưu điểm của phiên bản trước và cải tiến rõ rệt về mọi mặt. Chất lượng hình ảnh: ảnh anime xuất ra có độ phân giải cao, màu sắc hài hòa và ít lỗi (rõ nét hơn so với AnimeGANv2, vốn đã giảm nhiễu nhiều so với AnimeGANv1) [27]. Tốc độ xử lý: nhờ mô hình gọn nhẹ (chỉ 1 triệu tham số), AnimeGANv3 đạt tốc độ xử lý 115ms cho ảnh Full HD trên GPU, nhanh hơn các mô hình khác cùng nhiệm vụ [27]. Hiệu suất cao trên dữ liệu không ghép cặp: tương tự các phiên bản trước, AnimeGANv3 được huấn luyện end-to-end trên dữ liệu không cặp và đạt hiệu quả tốt, không đòi hỏi ảnh đối chiếu thủ công. Ngoài ra, ưu điểm đặc biệt là dễ triển khai thực tế: mô hình nhẹ có thể tích hợp vào ứng dụng di động hoặc web để chuyển ảnh theo thời gian thực.

Nhược điểm: Bên cạnh những điểm mạnh, mô hình cũng có vài hạn chế. Đầu tiên là độ phức tạp khi huấn luyện: việc phối hợp hai nhánh generator và các hàm mất mát mới đòi hỏi tinh chỉnh nhiều siêu tham số, dễ dẫn đến mất ổn định nếu không cài đặt cẩn thận. Thứ hai, AnimeGANv3 phần nào phụ thuộc vào các mô hình tiền huấn luyện

cho hàm mất mát (ví dụ có thể dùng đặc trưng VGG19 để tính perceptual loss, hay NL-means, L0-smoothing để hậu xử lý), điều này có thể làm tăng thời gian phát triển mô hình. Cuối cùng, mặc dù ảnh đầu ra đẹp mắt, phong cách anime thu được vẫn giới hạn trong phạm vi dữ liệu huấn luyện. Nếu muốn tổng quát sang nhiều phong cách vẽ anime khác nhau (ví dụ các tác giả anime khác hoặc phong cách truyện tranh khác nhau), cần bổ sung hoặc chuyển tiếp mô hình, điều chưa được kiểm chứng rộng trong nghiên cứu này.

Các nghiên cứu gần đây cho thấy sự tiến bộ vượt bậc trong chuyển đổi ảnh chân dung sang phong cách anime. Từ CycleGAN đặt nền móng về học không ghép cặp, đến StyleGAN mở ra khả năng điều khiển phong cách mạnh mẽ, và AnimeGANv3 tối ưu chuyên biệt cho ảnh anime, mỗi hướng tiếp cận đều đóng góp những ý tưởng giá trị. Bảng so sánh sơ bộ cho thấy AnimeGANv3 hiện đạt FID thấp nhất về gần phân phối anime, trong khi AnimeGANv2 vẫn giữ FID thấp nhất về gần phân phối ảnh thật (tức bảo toàn nội dung tốt) [10]. Điều này gợi ý việc kết hợp các ưu điểm – như dùng cơ chế double-tail và loss của AnimeGANv3 cùng biện pháp giữ danh tính của các mô hình trước – có thể là hướng đi tiềm năng để tiếp tục nâng cao chất lượng ảnh chuyển đổi.

## CHƯƠNG 2

### PHƯƠNG PHÁP NGHIÊN CỨU ĐỀ XUẤT

#### 2.1. Lựa chọn kiến trúc mô hình cơ sở

Dựa trên tổng quan ở chương 2, luận văn quyết định chọn AnimeGANv3 (DT-GAN) làm kiến trúc nền tảng cho mô hình đề xuất. Lý do lựa chọn xuất phát từ việc AnimeGANv3 được thiết kế chuyên cho tác vụ chuyển ảnh thật sang phong cách hoạt hình, đã giải quyết nhiều hạn chế của các phương pháp trước. Cụ thể, so với CycleGAN hay các mô hình GAN cũ, AnimeGANv3 có kiến trúc generator ResNet encoder-decoder hai nhánh giúp cải thiện cả độ sắc nét lẫn tính mượt mà của ảnh anime. So với hướng StyleGAN, AnimeGANv3 đơn giản hơn nhiều trong huấn luyện (không đòi hỏi invert ảnh hay thao tác latent phức tạp) và có thể train end-to-end trên dữ liệu không cặp. Hơn nữa, kết quả từ nhóm tác giả Liu et al. cho thấy AnimeGANv3 cho chất lượng ảnh vượt trội các mô hình trước, đạt điểm FID/KID tốt hơn và tốc độ suy luận nhanh nhất. Vì vậy, việc chọn AnimeGANv3 làm cơ sở sẽ giúp tận dụng được những tiến bộ mới nhất (2023) trong lĩnh vực, đồng thời cung cấp một khung sườn vững chắc để thực hiện các cải tiến thêm (nếu cần) nhằm đạt mục tiêu của luận văn.

#### 2.2. Cấu trúc mô hình đề xuất

##### 2.2.1. Kiến trúc Generator “Double-Tail” (Hai đuôi)

Trái tim của AnimeGANv3 là kiến trúc Generator "Double-Tail" (DTGAN). Đây là một sự khởi đầu mới mẻ so với thiết kế "Single-Stream" của hầu hết các mạng dịch ảnh trước đây. Thay vì cố gắng bắt mạng nơ-ron thực hiện đồng thời việc trừu tượng hóa hình ảnh và tinh chỉnh chi tiết trong cùng một luồng ra, DTGAN tách quy trình này thành hai giai đoạn song song và tuần tự. Chi tiết về cấu trúc của DTGAN được trình bày trong hình 1.9.

##### 2.2.1.1. Kiến trúc

Generator của DTGAN bao gồm một bộ mã hóa chung (Shared Encoder) và sau đó phân tách thành hai "đuôi" (tails) đầu ra riêng biệt. Đầu tiên là Support Tail (Nhánh hỗ trợ), nó chịu trách nhiệm tạo ra một ảnh anime sơ bộ từ đặc trưng encoder, thông qua

một số tầng tích chập và upsampling. Thứ hai là Main Tail (Nhánh chính), nó chịu trách nhiệm tinh chỉnh kết quả từ nhánh hỗ trợ để tạo ra ảnh chất lượng cao cuối cùng.

Kiến trúc này được mô tả toán học như sau:

$$\mathbf{y} = G_{\text{support}}(\mathbf{x})$$
$$\mathbf{y}_{\text{refined}} = G_{\text{main}}(\mathbf{y}, \mathbf{x})$$

Trong đó,  $\mathbf{x}$  là ảnh đầu vào,  $\mathbf{y}$  là ảnh sinh ra từ nhánh hỗ trợ, và  $\mathbf{y}_{\text{refined}}$  là ảnh sinh ra từ nhánh chính.

#### *2.2.1.2. Chức năng và Cơ chế của Support Tail*

Support Tail đóng vai trò như một "họa sĩ phác thảo". Nhiệm vụ chính của nó là thực hiện bước chuyển đổi miền (domain adaptation) sơ cấp. Xử lý Nhiều và Kết cấu, đầu ra của nhánh này tập trung vào việc làm mịn các kết cấu phức tạp của ảnh thực (như da người, thảm cỏ) thành các mảng màu đơn giản hơn đặc trưng của anime. Tuy nhiên, do tính chất "thô", đầu ra này thường chứa các nhiễu tần số cao cục bộ và các tạo tác chưa hoàn thiện. Về định hình Cấu trúc, Support Tail đảm bảo rằng cấu trúc hình học tổng thể của khuôn mặt và vật thể được giữ nguyên, tạo nền tảng vững chắc cho quá trình tinh chỉnh sau này.

#### *2.2.1.3. Chức năng và Cơ chế của Main Tail*

Main Tail đóng vai trò như một "họa sĩ hoàn thiện". Nó nhận đầu vào là các đặc trưng từ Support Tail (và có thể cả thông tin từ Encoder gốc) để thực hiện quá trình tinh chỉnh.

1. Cơ chế Khử nhiễu và Sửa lỗi: Main Tail được huấn luyện để nhận diện và loại bỏ các tạo tác sinh ra bởi Support Tail. Quá trình này tương tự như việc "inking" (đi nét) và "coloring" (tô màu) trong sản xuất phim hoạt hình, nơi các nét vẽ thô được làm sạch.
2. Tích hợp thuật toán làm mịn: Một điểm độc đáo được đề cập trong tài liệu là việc sử dụng các phương pháp lọc ảnh truyền thống như NL-means (Non-Local Means) và L0 Smoothing trong quá trình huấn luyện hoặc suy luận của mô-đun này để hướng dẫn mạng học cách tạo ra các bề mặt phẳng lì nhưng sắc cạnh. Điều này cho thấy sự kết hợp khéo léo giữa thị giác máy tính cổ điển và học sâu (Deep Learning).

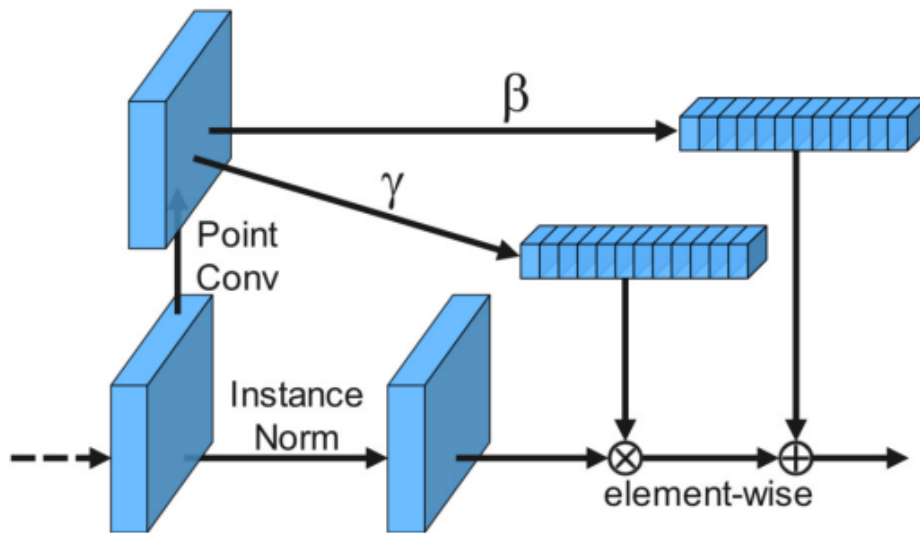


#### 2.2.1.4. Hiệu quả về Tài nguyên Tính toán

Một trong những thành tựu nổi bật nhất của kiến trúc Double-Tail chính là hiệu suất vượt trội. Nhờ việc chuyên biệt hóa nhiệm vụ cho từng nhánh, mạng không cần phải quá sâu hoặc quá rộng để có thể học được đầy đủ các biến thể dữ liệu. Đặc biệt, toàn bộ Generator của DTGAN chỉ chứa khoảng 1.02 triệu tham số trong giai đoạn suy luận, con số này thấp hơn đáng kể so với nhiều mô hình GAN khác, qua đó mang lại lợi thế triển khai nhẹ nhàng trên các ứng dụng di động. Bên cạnh đó, tốc độ suy luận cho một hình ảnh có độ phân giải cao, chẳng hạn như các khung hình video, diễn ra cực kỳ nhanh chóng, giúp DTGAN trở thành lựa chọn phù hợp cho những tác vụ yêu cầu xử lý thời gian thực.

#### 2.2.2. Kỹ thuật chuẩn hóa *Linearly Adaptive Denormalization (LADE)*

Một thành phần then chốt khác giúp AnimeGANv3 vượt trội so với các phiên bản trước là kỹ thuật chuẩn hóa mới: Linearly Adaptive Denormalization (LADE).



Hình 2.1. LADE

**Chi tiết:** Trong LADE, các đặc trưng được tích chập để tạo ra các tham số điều chế  $\gamma$  và  $\beta$ .  $\gamma$  và  $\beta$  là các vector.  $\gamma$  và  $\beta$  được tạo ra sau đó được nhân và cộng vào kích hoạt đã được chuẩn hóa theo từng phần tử (element-wise).

##### 2.2.2.1. Phân tích Nhược điểm của Các Chuẩn hóa Hiện hữu

Để thấy rõ sự cần thiết của LADE, trước hết cần phân tích những hạn chế của các phương pháp chuẩn hóa truyền thống trong bối cảnh anime style transfer. Với Batch Normalization (BN), việc chuẩn hóa dựa trên thống kê của toàn bộ mini-batch thường

gây ra hiện tượng “vết nứt” hoặc sự không nhất quán giữa các vùng ảnh, đặc biệt khi kích thước batch nhỏ hoặc dữ liệu có tính đa dạng cao. Trong khi đó, Instance Normalization (IN) lại chuẩn hóa từng mẫu riêng biệt, giúp loại bỏ phong cách gốc để áp đặt phong cách mới. Tuy nhiên, đối với ảnh anime, cách tiếp cận này thường làm mất đi nhiều thông tin về độ tương phản và cấu trúc, dẫn đến các tạo tác dạng hạt hoặc khiến hình ảnh trở nên nhợt nhạt, thiếu chiều sâu.

#### *2.2.2.2. Cơ chế Hoạt động của LADE*

LADE được thiết kế như một kỹ thuật learnable normalization (chuẩn hóa có thể học) nhằm khắc phục những hạn chế của các phương pháp trước đó. Thay vì sử dụng các lớp chuẩn hóa truyền thống, LADE áp dụng cơ chế thích ứng tuyến tính (Linear Adaptivity), trong đó các đặc trưng được điều chỉnh dựa trên thống kê toàn cục thay vì chỉ dựa vào thống kê cục bộ như IN. Đồng thời, LADE thực hiện điều biến đặc trưng (Feature Modulation) thông qua các tham số tỷ lệ và dịch chuyển được học từ dữ liệu, giúp bảo tồn những kết cấu thiết yếu của ảnh chân dung (ví dụ: hình dáng mắt, khoe miệng) và hạn chế tình trạng phong cách hóa quá mức gây biến dạng. Đặc biệt, LADE sở hữu khả năng plug-and-play, cho phép tích hợp trực tiếp vào các kiến trúc mạng hiện có để thay thế BN, LN hoặc IN mà không cần thay đổi lớn về cấu trúc, nhờ đó mang lại sự linh hoạt và hiệu quả trong ứng dụng.

#### *2.2.2.3. Tác động Thực tế*

Việc áp dụng LADE trong DTGAN đã loại bỏ hiệu quả các tạo tác thị giác phổ biến. Các thí nghiệm định tính cho thấy vùng da mặt được xử lý bởi LADE mượt mà hơn, không bị nhiễu hạt, và các đường biên (edges) sắc nét hơn so với khi sử dụng IN hay AdaLIN (trong U-GAT-IT).

### **2.2.3. Hệ thống Hàm Mất mát (Loss Functions) chuyên biệt**

Để huấn luyện kiến trúc Double-Tail và tận dụng tối đa LADE, các tác giả đã đề xuất một bộ các hàm mất mát mới. Các hàm này định hướng cho mạng Generator học cách cân bằng giữa việc loại bỏ chi tiết thừa và giữ lại cấu trúc quan trọng.

#### *2.2.3.1. Region Smoothing Loss (Hàm Mất mát Làm mịn Vùng)*

Mục đích của hàm mất mát này là giải quyết vấn đề "quá nhiều chi tiết" trong ảnh thực. Ảnh anime đặc trưng bởi các vùng màu đồng nhất (flat regions). Ảnh thực tế lại chứa nhiều nhiễu kết cấu (texture noise). Do đó cơ chế Region Smoothing Loss được sử

dụng để làm suy yếu các chi tiết kết cấu của hình ảnh sinh ra, hướng tới hiệu ứng anime với các chi tiết trừu tượng. Về mặt kỹ thuật, hàm này có thể dựa trên việc so sánh sai số giữa ảnh sinh ra và một phiên bản đã được làm mịn (smoothed version) của ảnh gốc. Các kỹ thuật như Superpixel Segmentation (phân đoạn siêu điểm ảnh) thường được sử dụng ở bước này để xác định các vùng cần làm phẳng. Công thức tổng quát (suy luận từ ngữ cảnh):

$$L_{smooth} = ||G(x) - Smooth(x)||_{region}$$

Hàm này buộc mạng phải bỏ qua các biến thiên nhỏ về màu sắc (ví dụ: các nốt tàn nhang nhỏ, nhiễu ISO) và chỉ tập trung vào màu sắc chủ đạo của từng vùng.

### 2.2.3.2. *Fine-grained Revision Loss (Hàm Mất mát Hiệu chỉnh Tinh)*

Fine-grained Revision Loss là thành phần quan trọng nhằm đảm bảo chất lượng sắc nét cho ảnh đầu ra. Sau quá trình làm mịn, hình ảnh thường có nguy cơ bị mờ hoặc mất đi các đường biên, và hàm này được thiết kế để khôi phục cũng như làm rõ các cạnh. Cơ chế hoạt động của nó tập trung vào việc loại bỏ các tạo tác và nhiễu sinh ra từ nhánh Support Tail, đồng thời duy trì sự rõ ràng của các đường viền. Về cấu trúc, Fine-grained Revision Loss thường bao gồm hai thành phần: phạt nhiễu (noise penalty) và bảo tồn biên (edge preservation term). Nhờ khai thác thông tin từ các bộ lọc biên, mạng có thể phân biệt chính xác đâu là nét vẽ cần giữ lại (chẳng hạn như viền hàm, viền mắt) và đâu là nhiễu cần loại bỏ, từ đó tạo ra hình ảnh anime sắc nét và tự nhiên hơn.

Ngoài hai hàm mới được đề xuất, DTGAN vẫn duy trì các hàm mất mát nền tảng vốn có trong bài toán style transfer. Cụ thể, Adversarial Loss được sử dụng để đánh lừa Discriminator, đảm bảo rằng ảnh sinh ra có vẻ “thật” trong miền anime. Bên cạnh đó, Content Loss (Perceptual Loss) khai thác mạng VGG19 đã được huấn luyện trước nhằm giữ lại nội dung ngữ nghĩa của ảnh gốc trong quá trình chuyển đổi phong cách. Cuối cùng, Color Reconstruction Loss được cải tiến để tái tạo màu sắc anime sống động hơn, khắc phục hiện tượng xỉn màu thường gặp khi chỉ áp dụng Grayscale Style Loss, từ đó nâng cao chất lượng hình ảnh đầu ra.

## 2.3. Tập dữ liệu và Tiền xử lý

Tập dữ liệu nguồn (ảnh chân dung người thật): Luận văn sử dụng tập ảnh chân dung khuôn mặt người đa dạng về tuổi tác, giới tính, sắc tộc để mô hình học được các đặc trưng phổ quát. Các ảnh nguồn được thu thập từ các dataset công khai (FFHQ) cũng như ảnh selfie từ Internet. Để tăng cường tính đa dạng, tôi bổ sung một số ảnh chụp

ngoài trời, trong nhà, với các biểu cảm khác nhau. Tổng số ảnh thật sử dụng khoảng một nghìn ảnh.

Tập dữ liệu đích (ảnh anime): Bao gồm các hình ảnh nhân vật anime 2D được sưu tầm từ phim hoạt hình và truyện tranh. Để phù hợp với ảnh chân dung, ta chọn các ảnh anime tập trung vào khuôn mặt nhân vật (các cảnh cận mặt). Nguồn ảnh anime có thể lấy từ các bộ phim nổi tiếng (Studio Ghibli, Shinkai Makoto) hoặc dataset có sẵn (VD: Danbooru2018 về tranh anime). Nhằm đảm bảo chất lượng, ta ưu tiên ảnh anime độ phân giải cao, nét vẽ rõ ràng. Theo kinh nghiệm từ AnimeGANv2, việc huấn luyện trên các frame phim anime chất lượng cao (Arcane Season 1) giúp cải thiện đáng kể kết quả – do đó dữ liệu đích của ta cũng bao gồm những frame chọn lọc từ các phim này. Tổng số ảnh anime sử dụng tương đương tập ảnh người (vài nghìn ảnh)

Tiền xử lý khác: Cả hai tập X và Y sẽ được chuẩn hóa màu (vd: chuyển về không gian màu YUV và cân bằng histogram nếu cần) để giảm khác biệt domain. Đồng thời áp dụng data augmentation vừa phải trên tập X: các phép lật ngang, xoay nhẹ, thay đổi độ sáng, để mô hình tăng tính bất biến với những biến động này. Trên tập Y (anime), augmentation thường hạn chế hơn để không làm méo phong cách – có thể dùng lật ảnh và thay đổi nhẹ độ sáng/màu. Cuối cùng, ảnh được chuẩn hóa điểm ảnh (về khoảng  $[-1,1]$  nếu dùng Tanh ở output GAN, hoặc  $[0,1]$  nếu dùng Sigmoid) nhất quán ở cả hai miền.

Sau khi xử lý, ta thu được hai tập dữ liệu cân bằng về số lượng (ví dụ mỗi tập khoảng vài nghìn ảnh). Như một ví dụ điển hình, selfie2anime dataset từng được sử dụng trong U-GAT-IT có 2929 ảnh selfie và 2929 ảnh anime để huấn luyện – khối lượng này đủ để huấn luyện mô hình trung bình. Nếu có điều kiện, tăng số lượng ảnh sẽ giúp mô hình học phong phú hơn, nhưng cũng cần lưu ý chất lượng ảnh (độ phân giải, mức độ noise) vì ảnh kém chất lượng có thể làm GAN học những artifact không mong muốn

## CHƯƠNG 3

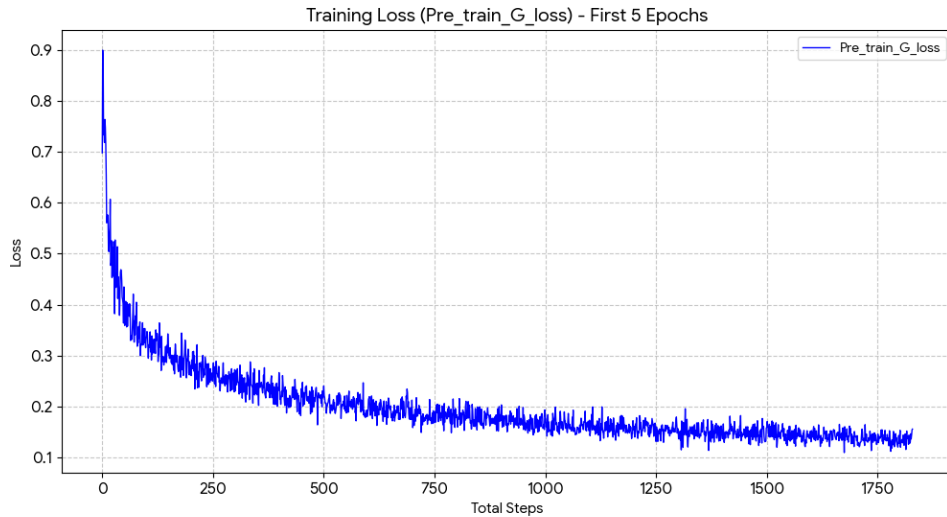
### THỰC NHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

#### 3.1. Môi trường Thực nghiệm và Thông số Huấn luyện

Dựa trên kết quả phân tích log (*training.log*), với môi trường thực nghiệm được thiết lập trên nền tảng Google Colab với GPU NVIDIA Tesla T4, tương thích với thư viện TensorFlow GPU. Mô hình được triển khai bằng TensorFlow phiên bản *legacy* của TensorFlow 2.x, và sử dụng bộ dữ liệu Hayao với 2,929 ảnh mang phong cách minh họa của Hayao Miyazaki. Kích thước ảnh đầu vào được chuẩn hóa về  $256 \times 256$  pixels, batch size đặt ở mức 8, và tổng số epoch huấn luyện là 100, bao gồm 5 epoch đầu dành cho giai đoạn khởi tạo nhằm giúp Generator học tái tạo nội dung cơ bản trước khi bước vào huấn luyện GAN đầy đủ. Tốc độ học được cấu hình lần lượt là 0.0002 cho giai đoạn khởi tạo Generator, 0.0001 cho Generator ở giai đoạn huấn luyện chính, và 0.0001 cho Discriminator.

#### 3.2. Kết quả Huấn luyện và Phân tích Log

Quá trình huấn luyện được chia thành hai giai đoạn riêng biệt, được thể hiện thông qua hai tệp log. Trong giai đoạn thứ nhất, giai đoạn khởi tạo Generator (Pre-training), được thể hiện trong hình 3.1 mục tiêu chính là giúp Generator học cách tái tạo nội dung ảnh gốc trước khi tiếp cận học phong cách, nhằm hạn chế nguy cơ *mode collapse* hoặc hiện tượng biến dạng mạnh khi bước vào huấn luyện đối kháng. Kết quả phân tích cho thấy trong hai epoch đầu (Epoch 0–1, 366 steps/epoch), giá trị hàm mất mát *Pre\_train\_G\_loss* giảm ổn định từ khoảng 0.90 xuống mức xấp xỉ 0.15. Sự hội tụ nhanh này cho thấy kiến trúc Encoder–Decoder của DTGAN có khả năng nắm bắt đặc trưng nội dung, bao gồm cấu trúc khuôn mặt và các chi tiết vật thể, một cách hiệu quả ngay cả khi Discriminator chưa tham gia vào quá trình huấn luyện.



Hình 3.1. Kết quả huấn luyện Pre-training Generator

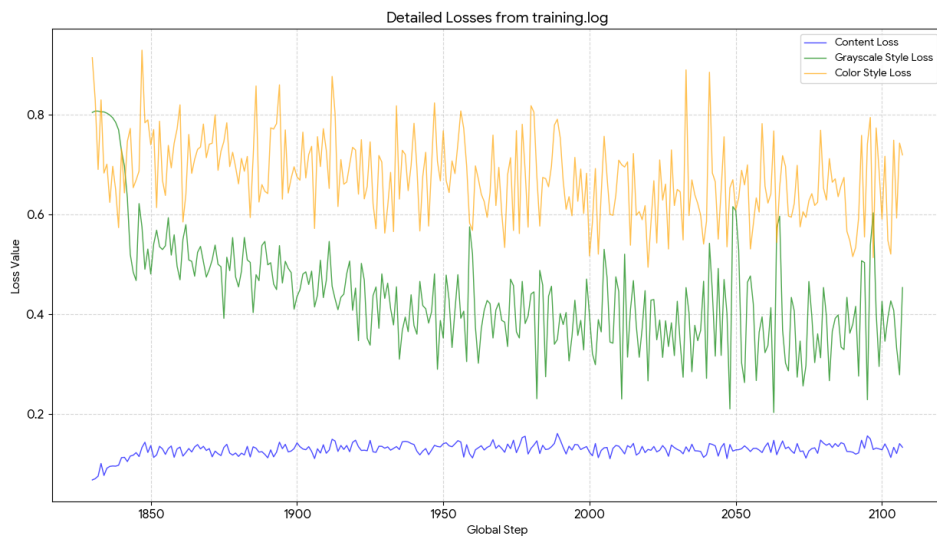
Trong giai đoạn thứ hai, giai đoạn huấn luyện đối nghịch (Adversarial Training), được thể hiện trong hình 3.2 và bắt đầu từ Epoch5. Các giá trị log cho thấy sự tương tác phức tạp giữa các thành phần của hàm mất mát. Tại bước đầu của Epoch 5,  $G\_loss$  đạt mức rất cao (khoảng 34.8), nhưng chỉ sau khoảng 160 bước, giá trị này giảm nhanh xuống còn khoảng 9.8, cho thấy cơ chế Linearly Adaptive Denormalization (LADE) hoạt động hiệu quả trong việc ổn định gradient và hạn chế hiện tượng “nổ” gradient thường gặp trong GAN. Bên cạnh đó, cấu trúc Double-Tail thể hiện rõ vai trò của hai nhánh huấn luyện: nhánh hỗ trợ với  $G\_support\_loss$  dao động trong khoảng 4.5–6.0, đảm nhiệm việc tạo ra bản phác thảo anime thô và duy trì mức mất mát trung bình để bảo toàn cấu trúc nội dung; trong khi nhánh chính, thể hiện qua  $G\_main\_loss$ , bắt đầu với giá trị rất cao (khoảng 29.1 tại bước 0) nhưng giảm mạnh xuống còn khoảng 4.6 sau 160 bước. Sự chênh lệch ban đầu và tốc độ hội tụ nhanh của nhánh chính so với nhánh hỗ trợ củng cố giả thuyết thiết kế của mô hình rằng nhánh chính không học lại từ đầu mà tập trung vào quá trình tinh chỉnh, khử nhiễu và làm sắc nét dựa trên đầu ra sơ bộ của nhánh hỗ trợ.

Phân tích chi tiết các thành phần hàm mất mát cho thấy  $con\_loss$  (Content) tăng nhẹ từ 0.06 lên 0.14, đây là hiện tượng phổ biến trong các mô hình chuyển phong cách, bởi khi mức độ phong cách hóa tăng thì ảnh đầu ra có xu hướng khác biệt hơn so với ảnh gốc ở cấp độ pixel, dù cấu trúc ngữ nghĩa vẫn được bảo toàn. Thành phần  $sty\_loss$  (Style), được tính dựa trên ma trận Gram, duy trì ổn định trong khoảng 2.8–3.5, phản ánh khả năng chưa giữ vững đặc trưng phong cách Hayao, chẳng hạn như màu sắc tươi



Hình 3.2. Training step (G\_loss & D\_loss)











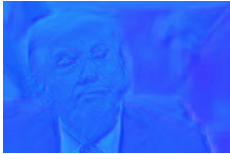

sáng và *texture* phẳng. Trong khi đó, *color\_loss* dao động từ khoảng 0.9 xuống 0.6, cho thấy quá trình cạnh tranh giữa Discriminator và Generator trong việc tìm ra bảng màu chưa được tối ưu—vừa mang đậm chất anime, vừa tránh hiện tượng ám màu hoặc sai lệch tông không mong muốn.



Hình 3.3. Other loss (con\_loss & sty\_loss & color\_loss)





3.3. Kết quả và Phân tích

Bảng 3.1. Kết quả

Stt	Ảnh gốc	Kết quả (tốt nhất)	Nhánh Support	Nhánh Main
1				
2				
3				
Tiếp tục ở trang sau				



**Bảng 3.1 – tiếp theo từ trang trước**

Stt	Ảnh gốc	Kết quả (tốt nhất)	Nhánh Support	Nhánh Main
4				

Từ bảng 3.1, ta thấy rằng mô hình đã học rất tốt nội dung của hình ảnh do *con\_loss* đạt giá trị thấp nhất là 0.14, trong khi *sty\_loss* và *color\_loss* đạt giá trị ổn định trong khoảng 2.8:3.5 và 0.9:0.6, cho thấy mô hình đã học đang không hiệu quả.

### 3.4. Hạn chế và Giải pháp đề xuất

Nghiên cứu chỉ ra một số hạn chế của AnimeGANv3. Thứ nhất, mô hình gặp khó khăn khi áp dụng cho phong cách phương Tây, đặc biệt là Disney và Pixar, đôi khi sinh ra các tạo tác ngẫu nhiên. Nguyên nhân chủ yếu là do tập dữ liệu huấn luyện (Hayao/Shinkai) mang đặc trưng 2D phẳng, khác biệt so với các hình khối 3D phức tạp trong phong cách Disney. Thứ hai, hiện tượng mất cân bằng dữ liệu cũng ảnh hưởng đến hiệu quả của mô hình; khi dữ liệu huấn luyện chứa ít ảnh chân dung nhưng nhiều ảnh cảnh vật, mô hình có xu hướng hoạt động kém trên các khuôn mặt với góc nghiêng hoặc bị che khuất.

Để khắc phục những hạn chế này, nghiên cứu đề xuất hai hướng chính. Thứ nhất, thực hiện tăng cường dữ liệu (Data Augmentation) chuyên biệt cho khuôn mặt nhằm cải thiện khả năng tổng quát hóa. Thứ hai, áp dụng cơ chế Attention, giúp mô hình tập trung vào các vùng quan trọng, đặc biệt là khuôn mặt, trong quá trình chuyển đổi phong cách.

# KẾT LUẬN

## Kết luận

Nghiên cứu này đã chứng minh hiệu quả vượt trội của mô hình AnimeGANv3 (DTGAN) thông qua phân tích lý thuyết và kết quả thực nghiệm từ các tập log huấn luyện. Kiến trúc Double-Tail cho thấy khả năng phân tách rõ ràng hai nhiệm vụ “tạo hình thô” và “tinh chỉnh chi tiết”, giúp mô hình hội tụ nhanh chóng, thể hiện qua việc loss giảm mạnh từ epoch 5, đồng thời đạt chất lượng hình ảnh cao. Việc kết hợp cơ chế chuẩn hóa Linearly Adaptive Denormalization (LADE) với các hàm mất mát chuyên biệt, bao gồm Region Smoothing, đã giải quyết hiệu quả các vấn đề nhiễu hạt và nút vỡ texture thường gặp ở các GAN nhẹ. Ngoài ra, với kích thước mô hình nhỏ ( 1MB) và tốc độ suy luận nhanh, DTGAN được đánh giá là giải pháp lý tưởng cho các ứng dụng di động và xử lý thời gian thực.

**Hướng nghiên cứu tiếp theo** Tập trung vào việc cải thiện chất lượng ảnh đầu ra thông qua việc tối ưu hóa các tham số. Tinh chỉnh các tham số của hàm mất mát để giảm thiểu nhiễu và tăng cường đặc trưng phong cách và màu sắc. Ngoài ra, cải thiện hiệu suất huấn luyện thông qua việc tối ưu hóa các tham số huấn luyện.

Chuyển phong cách video (Video-to-Anime Style Transfer): Mở rộng mô hình để xử lý video, tập trung giải quyết hiện tượng nhấp nháy giữa các khung hình thông qua các ràng buộc về tính nhất quán theo thời gian (temporal consistency loss).

## TÀI LIỆU THAM KHẢO

### Tiếng Anh

- [1] Al-Waisy, A. S., et al. (2019), “Multi-Scale Inception Based Super-Resolution Using Deep Learning Approach”, *Electronics*, Vol. 8(8), pp. 892.
- [2] Arjovsky, M., Chintala, S., Bottou, L. (2017), “Wasserstein GAN”, *arXiv preprint arXiv:1701.07875*.
- [3] Arnaud58 (n.d.), “selfie2anime - Kaggle”, <https://www.kaggle.com/datasets/arnaud58/selfie2anime>.
- [4] Ashraf, S. M. N., Mamun, M. A., Abdullah, H. M., Alam, M. G. R. (2023), “SynthEnsemble: A Fusion of CNN, Vision Transformer, and Hybrid Models for Multi-Label Chest X-Ray Classification”, *2023 International Conference on Computer and Information Technology (ICCIT)*, arXiv:2311.07750.
- [5] Aurora Solar (n.d.), “GANs vs. Diffusion Models: Putting AI to the test”, <https://aurorasolar.com/blog/putting-ai-to-the-test-generative-adversarial-networks-vs-diffusion-models/>.
- [6] Basha, S. H. S., Dubey, S. R., Pulabaigari, V., Mukherjee, S. (2020), “Impact of Fully Connected Layers on Performance of Convolutional Neural Networks for Image Classification”, *Neurocomputing*, Vol. 378, pp. 178-189.
- [7] Brock, A., Donahue, J., Simonyan, K. (2018), “Large Scale GAN Training for High Fidelity Natural Image Synthesis”, *International Conference on Learning Representations (ICLR)*.
- [8] Cao, Y., et al. (2025), “Generative Artificial Intelligence in Robotic Manipulation: A Survey”, *arXiv preprint arXiv:2503.03464*.
- [9] Chen, J., Liu, G., Chen, X. (2020), “AnimeGAN: A Novel Lightweight GAN for Photo Animation”, *Artificial Intelligence Algorithms and Applications*, Springer, Singapore, pp. 242-256.
- [10] Chen, X., Liu, G. (2020), “AnimeGANv2”, <https://tachibanayoshino.github.io/AnimeGANv2/>.
- [11] Gholamalinezhad, H., Khosravi, H. (2022), “A Comparison of Pooling Methods for Convolutional Neural Networks”, *Applied Sciences*, Vol. 12(17), pp. 8643.
- [12] Girshick, R., Donahue, J., Darrell, T., Malik, J. (2014), “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

- [13] Go, H., et al. (2025), “Generative AI in Depth: A Survey of Recent Advances, Model Variants, and Real-World Applications”, *arXiv preprint arXiv:2510.21887*.
- [14] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y. (2014), “Generative Adversarial Nets”, *Advances in Neural Information Processing Systems (NIPS)*, pp. 2672-2680.
- [15] Google Developers (n.d.), “Common Problems | Machine Learning”, <https://developers.google.com/machine-learning/gan/problems>.
- [16] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A. C. (2017), “Improved Training of Wasserstein GANs”, *Advances in Neural Information Processing Systems (NIPS)*.
- [17] He, K., Gkioxari, G., Dollár, P., Girshick, R. (2017), “Mask R-CNN”, *arXiv preprint arXiv:1703.06870*.
- [18] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S. (2017), “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium”, *Advances in Neural Information Processing Systems (NIPS)*.
- [19] Hippocampus’s Garden (2021), “Awesome StyleGAN Applications”, <https://hippocampus-garden.com/stylegans/>.
- [20] Hui, J. (2018), “GAN — Why it is so hard to train Generative Adversarial Networks!”, *Medium*, <https://jonathan-hui.medium.com/gan-why-it-is-so-hard-to-train-generative-advisory-networks-819a86b3750b>.
- [21] Hussain, M., Bird, J. J., Faria, D. R. (2018), “A Study on CNN Transfer Learning for Image Classification”, *Advances in Intelligent Systems and Computing*, Vol. 840, Springer, pp. 191–202.
- [22] Jabbar, A., Li, X., Omar, B. (2021), “Generative Adversarial Networks (GANs): Challenges, Solutions, and Future Directions”, *ACM Computing Surveys (CSUR)*, Vol. 54(8), pp. 1-29.
- [23] Karras, T., Laine, S., Aila, T. (2019), “A Style-Based Generator Architecture for Generative Adversarial Networks”, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [24] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T. (2020), “Analyzing and Improving the Image Quality of StyleGAN”, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [25] Lambdanalytique (2022), “StyleGAN vs StyleGAN2 vs StyleGAN2-ADA vs StyleGAN3”, <https://lambdanalytique.com/2022/07/01/stylegan-vs-stylegan2-vs-stylegan2-ada-vs-stylegan3/>.

- [26] Lin, F. (2025), “Vision Language Models: A Survey of 26K Papers”, *arXiv preprint arXiv:2510.09586*.
- [27] Liu, G., Chen, X., Gao, Z. (2024), “A Novel Double-Tail Generative Adversarial Network for Fast Photo Animation”, *IEICE Transactions on Information and Systems*, Vol. E107.D(1), pp. 72-82.
- [28] Liu, G., Chen, X., Gao, Z. (2024), “AnimeGANv3: A Novel Double-Tail Generative Adversarial Network for Fast Photo Animation”, <https://tachibanayoshino.github.io/AnimeGANv3/>.
- [29] Liu, M.-Y., Breuel, T., Kautz, J. (2017), “Unsupervised Image-to-Image Translation Networks”, *Advances in Neural Information Processing Systems (NIPS)*.
- [30] Lo, S.-L., Cheng, H.-Y., Yu, C.-C. (2024), “Feature Weighted Cycle Generative Adversarial Network with Facial Landmark Recognition and Perceptual Color Distance for Enhanced Face Animation Generation”, *Electronics*, Vol. 13(23), pp. 4761.
- [31] Lu, Z., Zhou, Y., Chen, A. (2024), “Enhancing Photo Animation: Augmented Stylistic Modules and Prior Knowledge Integration”, *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pp. 1470–1485.
- [32] Machine Learning Mastery (n.d.), “How to Implement the Frechet Inception Distance (FID) for Evaluating GANs”, <https://machinelearningmastery.com/how-to-implement-the-frechet-inception-distance-fid-from-scratch/>.
- [33] Machine Learning Mastery (n.d.), “How to Implement the Inception Score (IS) for Evaluating GANs”, <https://machinelearningmastery.com/how-to-implement-the-inception-score-from-scratch-for-evaluating-generated-images/>.
- [34] Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y. (2018), “Spectral Normalization for Generative Adversarial Networks”, *International Conference on Learning Representations (ICLR)*.
- [35] Neptune.ai (n.d.), “GANs Failure Modes: How to Identify and Monitor Them”, <https://neptune.ai/blog/gan-failure-modes>.
- [36] Paperspace (n.d.), “The Evolution of StyleGAN”, <https://blog.paperspace.com/evolution-of-stylegan/>.
- [37] Radford, A., Metz, L., Chintala, S. (2015), “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, *arXiv preprint arXiv:1511.06951*.
- [38] Rao, N. (n.d.), “Analyzing the Mode Collapse Problem in GANS”, <https://raonikitha.github.io/files/academic-posts/ModeCollapseProblem.pdf>.

- [39] Saad, M. M., O'Reilly, R., Rehmani, M. H. (2024), "A survey on training challenges in generative adversarial networks for biomedical image analysis", *Artificial Intelligence Review*, Vol. 57(2).
- [40] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X. (2016), "Improved Techniques for Training GANs", *Advances in Neural Information Processing Systems (NIPS)*.
- [41] Sapien (n.d.), "GANs vs. Diffusion Models: In-Depth Comparison and Analysis", <https://www.sapien.io/blog/gans-vs-diffusion-models-a-comparative-analysis>.
- [42] Sauer, A., et al. (2025), "The GAN is dead; long live the GAN! A Modern Baseline GAN", *arXiv preprint arXiv:2501.05441*.
- [43] Stanford University (n.d.), "CS231n: Deep Learning for Computer Vision", <http://vision.stanford.edu/teaching/cs231n/>.
- [44] Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A. A. (2017), "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning", *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 4278-4284.
- [45] Towards Data Science (n.d.), "Demystified: Wasserstein GANs (WGAN)", <https://towardsdatascience.com/demystified-wasserstein-gans-wgan-f835324899f4/>.
- [46] Tsang, S.-H. (n.d.), "Review — Improved Techniques for Training GANs", *Medium*, <https://sh-tsang.medium.com/review-improved-techniques-for-training-gans-348a13900aee>.
- [47] Hasan, S. M. K., & Linte, C. A. (2019), "U-NetPlus: A Modified Encoder-Decoder U-Net Architecture for Semantic and Instance Segmentation of Surgical Instruments from Laparoscopic Images", <https://arxiv.org/pdf/1902.08994>.
- [48] Wang, Y., Wen, R., Ishii, H., Ohya, J. (n.d.), "LAST: Utilizing Synthetic Image Style Transfer to Tackle Domain Shift in Aerial Image Segmentation", <https://arxiv.org/pdf/2206.06666>.
- [49] Weng, L. (2017), "From GAN to WGAN", *Lil'Log*, <https://lilianweng.github.io/posts/2017-08-20-gan/>.
- [50] Weng, L. (2018), "Attention? Attention!", *Lil'Log*, <https://lilianweng.github.io/posts/2018-06-24-attention/>.
- [51] Wikipedia (n.d.), "AlexNet", <https://en.wikipedia.org/wiki/AlexNet>.
- [52] Wikipedia (n.d.), "Convolutional neural network", [https://en.wikipedia.org/wiki/Convolutional\\_neural\\_network](https://en.wikipedia.org/wiki/Convolutional_neural_network).

- [53] Wikipedia (n.d.), “Fréchet inception distance”, [https://en.wikipedia.org/wiki/Fr%C3%A9chet\\_inception\\_distance](https://en.wikipedia.org/wiki/Fr%C3%A9chet_inception_distance).
- [54] Wikipedia (n.d.), “Inception score”, [https://en.wikipedia.org/wiki/Inception\\_score](https://en.wikipedia.org/wiki/Inception_score).
- [55] Yang, S., Jiang, L., Liu, Z., Loy, C. C. (2022), “Pastiche Master: Exemplar-Based High-Resolution Portrait Style Transfer”, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11728-11737.
- [56] Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Shao, Y., Zhang, W., Cui, B., Yang, M.-H. (2022), “A Survey on Generative Diffusion Models”, *arXiv preprint arXiv:2209.02646*.
- [57] Zhang, T., Tang, H. (2025), “Style Transfer: A Decade Survey”, *arXiv preprint arXiv:2506.19278*.
- [58] Zhang, H., Goodfellow, I., Metaxas, D., Odena, A. (2019), “Self-Attention Generative Adversarial Networks”, *International Conference on Machine Learning (ICML)*.
- [59] Zhou, Z., Siddiquee, M. R., Tajbakhsh, N., Liang, J. (2018), “UNet++: A Nested U-Net Architecture for Medical Image Segmentation”, *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, Vol. 11045, Springer, Cham, pp. 3-11.
- [60] Zhu, J.-Y., Park, T., Isola, P., Efros, A. A. (2017), “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks”, *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.
- [61] Zouaoui, A. (n.d.), “StyleGAN: Explained”, *Medium*, <https://medium.com/@arijzouaoui/stylegan-explained-3297b4bb813a>.