ChatGPT

Tổng quan các nghiên cứu liên quan (Chương 1–3)

Chương 1: Mở đầu

1.1 Tính cấp thiết và ý nghĩa của đề tài

Sự phát triển nhanh chóng của **nội dung tạo bởi AI (AIGC)** đang làm thay đổi sâu sắc lĩnh vực sáng tạo nội dung số và nghệ thuật thị giác [1]. Trong bối cảnh đó, ngành công nghiệp Anime/Manga – một lĩnh vực văn hóa quan trọng được ứng dụng rộng rãi trong quảng cáo, giáo dục và giải trí – cũng chứng kiến nhu cầu ứng dụng AI để hỗ trợ quá trình sáng tác [2]. Việc tạo ra các nhân vật và hình ảnh anime chất lượng cao đòi hỏi nhiều thời gian và kỹ năng chuyên môn, do đó **tự động hóa quá trình chuyển ảnh thật sang phong cách anime** sẽ giúp **giảm rào cản kỹ thuật** và **tăng tính tự do sáng tạo** cho nghệ sĩ [2]. Tuy đã có những tiến bộ nhất định, **bài toán chuyển đổi ảnh chân dung người thật sang phong cách anime** vẫn là một thách thức lớn trong lĩnh vực **phong cách truyền (style transfer)**. Các phương pháp trước đây thường gặp vấn đề sinh ra nhiễu, mất chi tiết hoặc méo dạng trong ảnh kết quả [3]. Đặc biệt, việc **giữ được đặc điểm nhận dạng khuôn mặt** của người trong ảnh gốc khi chuyển sang phong cách anime là một yêu cầu quan trọng nhưng khó đạt được. Công nghệ **Mạng đối nghịch tạo sinh (GAN)** đã nổi lên như một công cụ mạnh mẽ cho các tác vụ sinh ảnh và biến đổi phong cách phức tạp, mở ra hướng tiếp cận đầy hứa hẹn để giải quyết bài toán này [4]. Tuy nhiên, áp dụng GAN cho ảnh phong cách anime không đơn giản do khác biệt lớn giữa ảnh thực và tranh hoạt hình, đòi hỏi những kiến trúc và kỹ thuật chuyên biệt mới [4]. Do đó, nghiên cứu một giải pháp GAN hiệu quả để chuyển ảnh thực sang phong cách anime có ý nghĩa cấp thiết cả về mặt khoa học lẫn ứng dụng thực tiễn.

1.2 Mục tiêu nghiên cứu

Mục tiêu tổng quát: Xây dựng và đánh giá một mô hình **GAN** hiệu quả để chuyển đổi ảnh chân dung người thật thành ảnh phong cách **anime 2D chất lượng cao**, trong đó ảnh đầu ra vừa mang đậm nét vẽ anime vừa **giữ được các đặc điểm nhận dạng chính** của đối tượng gốc.

Mục tiêu cụ thể:

- **Lựa chọn kiến trúc GAN phù hợp:** Nghiên cứu các mô hình GAN hiện có và lựa chọn kiến trúc nền tảng tối ưu (dự kiến là AnimeGAN thế hệ mới) làm cơ sở phát triển mô hình.
- **Xây dựng và xử lý dữ liệu huấn luyện:** Thu thập hoặc tinh chỉnh tập dữ liệu gồm ảnh chân dung người thật và ảnh anime, kèm theo các bước tiền xử lý như nhận dạng và căn chỉnh khuôn mặt, chuẩn hóa kích thước và tăng cường dữ liệu.
- **Đề xuất hàm mất mát và chiến lược huấn luyện tối ưu:** Thiết kế các hàm mất mát chuyên biệt (ví dụ: loss làm mịn vùng, loss chỉnh sửa chi tiết) và chiến lược huấn luyện để mô hình hội tụ ổn định, giảm hiện tượng nhiễu và **bảo toàn chi tiết nội dung**.
- **Đánh giá kết quả:** Đánh giá chất lượng ảnh anime tạo ra thông qua các chỉ số định lượng (FID, LPIPS, v.v.) và đánh giá định tính (khảo sát người dùng), so sánh với các phương pháp hiện có để xác định mức độ cải thiện của mô hình đề xuất.

1.3 Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu: Các kiến trúc **Mạng đối nghịch tạo sinh (GAN)** phục vụ cho bài toán chuyển đổi hình ảnh, đặc biệt là các mô hình GAN chuyên xử lý **ảnh chân dung** và **phong cách anime**. Ngoài ra, nghiên cứu liên quan đến các kỹ thuật thị giác máy tính và học sâu như mạng **CNN**, mô hình **encoder–decoder**, kỹ thuật **truyền phong cách neural**, và các hàm mất mát perceptual dựa trên **mạng VGG19**,...

Phạm vi nghiên cứu: Luận văn tập trung vào **bài toán chuyển đổi ảnh chân dung người** sang **phong cách tranh anime 2D**. Phạm vi không bao gồm các phong cách hoạt hình 3D hoặc chuyển đổi video động (video-to-anime). Dữ liệu huấn luyện chủ yếu là ảnh chân dung người (có thể từ nguồn ảnh khuôn mặt công khai) và ảnh anime (các khung hình trích xuất từ phim hoạt hình hoặc tranh vẽ phong cách anime). Nghiên cứu ưu tiên phương pháp học sâu với dữ liệu **không ghép cặp** (unpaired), tức là không đòi hỏi cặp ảnh trước-sau tương ứng, nhằm tăng tính linh hoạt khi thu thập dữ liệu.

1.4 Cấu trúc của luận văn

Luận văn dự kiến gồm 5 chương với nội dung như sau:

- **Chương 1: Mở đầu** – Trình bày sự cần thiết của đề tài, mục tiêu nghiên cứu, đối tượng và phạm vi, cùng cấu trúc tổng quan của luận văn.
- **Chương 2: Cơ sở lý thuyết và tổng quan nghiên cứu** – Giới thiệu nền tảng lý thuyết về học sâu và thị giác máy tính liên quan, khái quát về kiến trúc GAN, và tổng quan các nghiên cứu, mô hình tiêu biểu đã được đề xuất cho bài toán chuyển ảnh sang phong cách anime.
- **Chương 3: Phương pháp nghiên cứu đề xuất** – Trình bày chi tiết mô hình đề xuất dựa trên GAN (kiến trúc, thành phần Generator/Discriminator, kỹ thuật normalization, hàm mất mát, v.v.), cùng với quy trình xây dựng dữ liệu và các bước huấn luyện mô hình.
- **Chương 4: Thực nghiệm và đánh giá kết quả** – Mô tả cấu hình môi trường thực nghiệm, các siêu tham số huấn luyện, trình bày kết quả thu được của mô hình đề xuất so với các mô hình so sánh, phân tích định lượng và định tính chất lượng ảnh sinh ra, cũng như thảo luận các trường hợp thành công/thất bại.
- **Chương 5: Kết luận và hướng phát triển** – Tóm tắt những đóng góp chính của luận văn, đánh giá việc hoàn thành mục tiêu đề ra, và đề xuất các hướng nghiên cứu mở rộng trong tương lai (như cải thiện điều khiển phong cách chi tiết hơn, hoặc mở rộng sang bài toán video-to-anime).

Chương 2: Cơ sở lý thuyết và Tổng quan nghiên cứu

2.1 Cơ sở lý thuyết về học sâu và thị giác máy tính

Mạng nơ-ron tích chập (Convolutional Neural Networks – CNN): CNN là nền tảng của nhiều mô hình thị giác máy tính hiện đại. Kiến trúc CNN được thiết kế chuyên biệt cho dữ liệu ảnh, gồm chuỗi các **lớp tích chập (convolutional layers)** xen kẽ với các **lớp kích hoạt phi tuyến** (ví dụ: ReLU) và **lớp pooling** để giảm độ phân giải không gian [5] [6]. Lớp tích chập hoạt động như các bộ lọc cục bộ, trượt trên ảnh đầu vào và trích xuất các đặc trưng hình học như cạnh, đường cong,... Ứng với mỗi lớp tích chập, một **kernel** nhỏ được học để phát hiện một loại đặc trưng nhất định trên toàn ảnh. Sau tích chập, các lớp **Pooling** (như max-pooling) thực hiện giảm mẫu, gộp các giá trị đặc trưng trong vùng lân cận, qua đó giảm kích thước dữ liệu và tạo tính bất biến tương đối đối với dịch chuyển và biến dạng nhỏ [6]. Nhờ đan xen các tầng tích chập – phi tuyến – pooling, CNN dần xây dựng được các đặc trưng trừu tượng hơn ở các tầng sau. Với đầu ra cuối

cùng thường được làm phẳng (flatten) và đưa qua các **lớp kết nối đầy đủ (fully-connected)**, CNN có thể thực hiện nhiều tác vụ như phân loại ảnh, nhận dạng đối tượng, v.v. Kiến trúc CNN cũng là nền tảng của các mô hình **encoder–decoder** dùng trong bài toán **dịch ảnh (image-to-image translation)**: phần **encoder** (với các lớp tích chập và pooling) nén ảnh gốc thành vector đặc trưng ẩn, sau đó phần **decoder** (các lớp tích chập chuyển vị hoặc upsampling) tái tạo vector ẩn thành ảnh đầu ra mong muốn. Cách tiếp cận encoder–decoder kết hợp với học sâu đã chứng tỏ hiệu quả trong nhiều bài toán xử lý ảnh phức tạp, tạo tiền đề cho việc áp dụng vào truyền phong cách và **chuyển đổi ảnh** giữa các miền khác nhau.

2.2 Kiến trúc Mạng đối nghịch tạo sinh (GAN)

Nguyên lý hoạt động: Mạng đối nghịch tạo sinh (**GAN**) được đề xuất bởi Goodfellow et al. (2014) gồm hai thành phần chính huấn luyện song song: **Mạng tạo sinh (Generator)** và **Mạng phân biệt (Discriminator)** [7] [8]. Generator nhận một nguồn đầu vào (ví dụ vector nhiễu ngẫu nhiên hoặc một ảnh cần biến đổi) và tìm cách **sinh ra dữ liệu giả** (ảnh sinh) sao cho **giống với dữ liệu thật** mong muốn. Ngược lại, Discriminator đóng vai trò một bộ phân loại nhị phân, cố gắng phân biệt ảnh nào là **thật** (từ dữ liệu huấn luyện) và ảnh nào là **giả** do Generator tạo ra [9]. Hai mạng này được huấn luyện trong một trò chơi **minimax**: Generator cố gắng đánh lừa Discriminator (tối ưu để Discriminator phân loại sai ảnh sinh là thật), trong khi Discriminator tối ưu để phân biệt chính xác thật/giả [10]. Hàm mục tiêu ban đầu của GAN là hàm mất mát đối nghịch (adversarial loss) theo dạng minimax, trong đó Generator muốn **minimize** xác suất bị Discriminator phát hiện, còn Discriminator muốn **maximize** độ chính xác phân biệt. Quá trình huấn luyện lý tưởng đạt **Điểm yên ngựa Nash (Nash equilibrium)** khi Discriminator không thể phân biệt được thật/giả tốt hơn đoán ngẫu nhiên, đồng nghĩa Generator đã sinh ảnh rất sát với phân phối ảnh thật [11].

Hàm mất mát gốc và vấn đề huấn luyện: Hàm mất mát GAN cổ điển gồm hai phần: mất mát cho Discriminator (phân biệt đúng thật/giả) và mất mát cho Generator (khi ảnh giả bị phân biệt là giả). Công thức ban đầu sử dụng hàm cross-entropy, tuy nhiên trong thực tế việc tối ưu hóa trực tiếp thường dẫn đến tình trạng **mất cân bằng**: nếu Discriminator quá mạnh, Generator không nhận được gradient hiệu quả. Để khắc phục, Goodfellow đề xuất thủ thuật cho Generator thay vì maximize xác suất ảnh giả bị nhận là thật, thì minimize log(1 - D(G(z))), giúp gradient của Generator không bị triệt tiêu khi Discriminator mạnh. Mặc dù vậy, huấn luyện GAN thường gặp nhiều **vấn đề ổn định**. Các thách thức kinh điển gồm: **Mode collapse** – Generator chỉ học được một vài mẫu ảnh và lặp lại chúng cho mọi đầu vào (dẫn đến thiếu đa dạng) [12]; **Non-convergence (không hội tụ)** – Generator và Discriminator dao động, không đạt điểm cân bằng; và **Gradient vanishing/Exploding** – làm quá trình học trở nên không ổn định [8] [13]. Nhiều nghiên cứu đã đề xuất giải pháp cho các vấn đề này, ví dụ: thay đổi hàm mất mát (Wasserstein GAN sử dụng Earth-Mover distance để giảm vanishing gradient [14]), **Regularization** cho Discriminator, kỹ thuật **Normalizaion** (như Spectral Normalization để ổn định độ lợi của mạng) [15], và bổ sung các cơ chế như **minibatch discrimination**, **self-attention**... để giảm hiện tượng mode collapse [16]. Nhờ những cải tiến này, các biến thể GAN hiện đại đã ổn định hơn, tạo ra ảnh chất lượng cao và đa dạng hơn so với GAN nguyên bản.

2.3 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) [4]. Dưới đây là các mô hình GAN tiêu biểu (giai đoạn 2017–2024) đã được nghiên cứu cho nhiệm vụ này:

2.3.1 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** [17] [18]. CycleGAN bao gồm hai bộ Generator song song: $G: X \to Y$ (chuyển ảnh từ miền nguồn X sang miền đích Y, ví dụ ảnh thật sang ảnh anime) và $F: Y \to X$ (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 [18]. 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 [18] [19]. 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.

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 ⟷ zebra, phong cảnh ⟷ tranh Monet) [17]. 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 [20]. 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 [21] [22]. 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) [23].

2.3.2 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)** [24]. 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 [24]. 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) [25], StyleGAN đã chứng tỏ khả năng sinh ảnh chân dung người **đáng kinh ngạc**, khó phân biệt với ảnh chụp thật [26]. 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 [27].

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 [28] [29]. 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) [29] [30]. 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 [26]. 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 [31]. 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 [28] [29].

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 [32] [33]. 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 [34] [35]. Nhìn chung, phương pháp dựa trên StyleGAN đạt chất lượng ảnh rất cao nhưng đòi hỏi nhiều bước xử lý, không "nhanh gọn" như các mô hình chuyển đổi trực tiếp kiểu CycleGAN.

2.3.3 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) [36].

Kiến trúc Generator "hai đuôi" (Double-tail): Điểm độc đáo của AnimeGANv3 là Generator có **hai nhánh đầu ra (two output tails)** [36]. 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 nhiễu 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 [36].

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 [37].

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)** [38]. 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 nứt vỡ hoặc nhiễu [39]), 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 [39]. 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 [40].

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 [41]. 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 [41]. 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 [42].

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 nhiều nhiễu so với AnimeGANv1) [43]. **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 [43]. **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 [37]. 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 [44]), 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.

Tóm lại, 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) [43]. 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 3: Phương pháp nghiên cứu đề xuất

3.1 Lựa chọn kiến trúc mô hình cơ sở

Dựa trên tổng quan ở chương 2, luận văn quyết định chọn **AnimeGANv3 (DTGAN)** 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 [36]. 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 [43]. 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.

3.2 Cấu trúc mô hình đề xuất

3.2.1 Kiến trúc Generator "Double-Tail" (Hai đuôi)

Mô hình đề xuất sử dụng Generator theo kiến trúc **"Double-Tail"** giống AnimeGANv3 [36]. Về tổng thể, Generator gồm hai phần chính: **Encoder** và **Decoder hai nhánh**.

- **Encoder:** Được xây dựng từ các khối **Residual (ResNet blocks)** liên tiếp, kết hợp với các lớp tích chập và pooling để trích xuất đặc trưng của ảnh chân dung gốc. Encoder chịu trách nhiệm mã hóa thông tin nội dung khuôn mặt (hình dạng khuôn mặt, vị trí mắt, mũi, miệng) vào tensor ẩn, đồng thời trích rút các đặc điểm cần truyền sang ảnh anime (tư thế, bố cục).
- **Decoder hai nhánh:** Từ tensor đặc trưng chung, Decoder tách làm hai nhánh đầu ra:
- **Nhánh Hỗ trợ:** gồm một số tầng tích chập và upsampling, tạo ảnh anime sơ bộ. Ảnh này giữ bố cục và nét chính nhưng chưa được tinh lọc, có thể còn viền răng cưa hoặc nhiễu màu nhẹ.
- **Nhánh Chính:** nhận đầu vào là ảnh anime sơ bộ (đã qua một phép trộn/concat với đặc trưng encoder, tùy thiết kế cụ thể), sau đó qua thêm các khối tích chập Residual để **loại bỏ nhiễu và tăng nét**. Kết quả là ảnh anime cuối cùng với chi tiết được làm sạch và nét vẽ sắc sảo hơn. Trong quá trình huấn luyện, cả hai nhánh đều hoạt động; nhưng khi suy luận, **chỉ nhánh chính được sử dụng**, nhánh hỗ trợ có thể bỏ đi để tăng tốc độ [45].

Thiết kế hai đuôi này cho phép mô hình học tốt hơn: nhánh hỗ trợ học biểu diễn phong cách anime thô, giúp nhánh chính tập trung học phần sai khác tinh tế (fine-grained) cần hiệu chỉnh, thay vì phải học tất cả trong một lần. Điều này tương tự chiến lược "thô rồi tinh" giúp ảnh đầu ra đạt chất lượng cao mà không cần một generator quá lớn.

3.2.2 Kỹ thuật chuẩn hóa Linearly Adaptive Denormalization (LADE)

Mô hình áp dụng kỹ thuật **LADE** như đề xuất trong AnimeGANv3 [38]. Cụ thể, thay vì dùng Instance Normalization (IN) hay Layer Normalization (LN) sau các khối tích chập (vốn dễ gây mất tương phản cục bộ hoặc sinh nhiễu đường biên trong phong cách anime [39]), mô hình sử dụng **LADE layers**. Mỗi LADE layer sẽ học các tham số α và β để điều chỉnh phân phối kích hoạt theo dạng tuyến tính: $\text{LADE}(h) = \alpha \cdot \hat{h} + \beta$, trong đó $\hat{h}$ là feature map đã được chuẩn hóa (ví dụ trừ mean chia độ lệch chuẩn theo batch hoặc layer). Khác với AdaIN (dùng style code từ vector z), LADE học trực tiếp α, β như tham số của mạng, cho phép **tinh chỉnh phân phối đặc trưng** phù hợp với ảnh hoạt hình. Thực nghiệm cho thấy LADE giúp giảm hẳn hiện tượng **"nứt nẻ"** hay **"loang màu"** so với BN, IN, LN trong cùng điều kiện [39]. Do đó, trong kiến trúc đề xuất, các lớp normalization trong ResNet block và decoder sẽ được thay bằng LADE. Lưu ý rằng LADE không làm tăng nhiều tham số và hoạt động *plug-and-play*, nên mô hình tổng thể vẫn gọn nhẹ tương đương AnimeGANv2/3.

3.2.3 Hệ thống hàm mất mát chuyên biệt

Để đạt hiệu quả cao trong việc chuyển phong cách anime, mô hình sử dụng tổ hợp nhiều hàm mất mát:

- **Adversarial loss (mất mát đối nghịch):** thành phần cơ bản huấn luyện GAN, gồm mất mát cho Discriminator và Generator như thông lệ. Hàm mất mát này đảm bảo ảnh anime sinh ra **khó phân biệt** với ảnh anime thật. Mô hình sử dụng kiến trúc Discriminator tương tự AnimeGAN (dạng PatchGAN), và áp dụng hàm mất mát GAN phiên bản **hinge loss** (hoặc một biến thể ổn định hơn so với cross-entropy truyền thống) để cải thiện độ ổn định.

- **Content loss / Identity loss:** nhằm bảo toàn **nội dung và danh tính** khuôn mặt gốc, mô hình thêm một hàm mất mát nội dung. Cách tính: so sánh ảnh sinh $G(x)$ với ảnh gốc x trên không gian đặc trưng trích xuất bởi mạng **VGG19** đã tiền huấn luyện [46]. Cụ thể, sử dụng một số layer của VGG19 (ví dụ conv3_3, conv4_3) để đảm bảo cấu trúc khuôn mặt (hình dáng mắt, mũi, miệng) được giữ lại. Ngoài ra, có thể áp dụng thêm **identity loss** trực tiếp trên ảnh (L1 hoặc L2 giữa $G(x)$ và x) với trọng số nhỏ, nhất là khi huấn luyện với ảnh chân dung để khuôn mặt nhân vật anime giống người thật nhất có thể (theo gợi ý từ U-GAT-IT sử dụng identity loss cho ảnh cùng miền [21]).

- **Style loss (mất mát phong cách):** Để đảm bảo ảnh đầu ra mang đúng chất **anime**, mô hình áp dụng mất mát phong cách dựa trên **Gram matrix** của feature map (theo phương pháp Neural Style Transfer của Gatys). Sử dụng tập ảnh anime tham chiếu y (có thể chính là ảnh trong batch từ domain Y) và tính khoảng cách giữa các thống kê Gram của ảnh $G(x)$ và ảnh y trên các tầng VGG19. Mất mát này khuyến khích **texture và màu sắc tổng quát** của ảnh sinh ra tương đồng với phong cách anime thật (ví dụ mắt to tròn, màu da, màu tóc kiểu hoạt hình).

- **Region smoothing loss (mất mát làm mượt vùng):** Theo đề xuất của Liu *et al.* [41], mô hình thêm loss làm mượt để khuyến khích các vùng đồng nhất màu trong ảnh anime. Cách thực hiện: áp dụng bộ lọc làm mờ hoặc phép tính phương sai cục bộ trên ảnh $G(x)$, phạt những vùng có độ biến thiên texture cao. Điều này giúp **loại bỏ chi tiết thừa** từ ảnh thật (như lỗ chân lông, nếp nhăn nhỏ) hướng tới mảng màu phẳng của tranh vẽ.

- **Fine-grained revision loss (mất mát chỉnh chi tiết):** Cũng theo AnimeGANv3 [41], mô hình sử dụng một hàm loss để xử lý vi sai tạo tác. Chẳng hạn, so sánh $G(x)$ trước và sau khi qua bộ lọc làm mịn

(như **bilateral filter** hoặc **Non-Local Means**): nếu có nhiều khác biệt tức là ảnh $G(x)$ còn nhiễu hạt, cần phạt. Loss này tập trung bảo vệ đường viền quan trọng (vì đường viền sẽ không bị làm mờ quá mức bởi filter cạnh-bảo-toàn như bilateral), đồng thời loại bỏ điểm nhiễu. Kết quả là ảnh cuối cùng giữ được nét vẽ đen quanh các vùng ranh giới khuôn mặt, tóc, mắt rõ ràng, nhưng các đốm nhỏ vô nghĩa bị triệt tiêu.

Các hàm mất mát trên được gán trọng số thích hợp và kết hợp trong hàm mục tiêu tổng. Việc cân bằng các trọng số sẽ được tinh chỉnh qua thực nghiệm để đạt sự đánh đổi tối ưu giữa **giữ nội dung** và **thể hiện phong cách**.

3.4 Tập dữ liệu và tiền xử lý

Để huấn luyện mô hình, ta cần chuẩn bị hai tập dữ liệu **không ghép cặp**: **tập nguồn X** (ảnh chân dung người thật) và **tập đích Y** (ảnh phong cách anime).

Tập nguồn (ảnh người thật): Bao gồm các ảnh chân dung khuôn mặt người đa dạng về giới tính, độ tuổi, sắc tộc. Có thể sử dụng các bộ dữ liệu có sẵn như **CelebA-HQ**, **FFHQ** hoặc ảnh selfie thu thập từ Internet, đảm bảo chất lượng cao (độ phân giải ít nhất 256×256). Ảnh được **căn giữa khuôn mặt**, loại bỏ nền phức tạp nếu cần, để tập trung vào vùng mặt. Công cụ như **MTCNN** có thể được dùng để tự động phát hiện khuôn mặt, căn chỉnh mắt-mũi-miệng theo một vị trí chuẩn, và cắt ảnh vuông. Việc căn chỉnh giúp các đặc trưng khuôn mặt ở vị trí nhất quán, hỗ trợ Generator học mapping thuận lợi hơn.

Tập đích (ảnh anime): Gồm các hình ảnh phong cách anime 2D. Để đảm bảo tính thống nhất phong cách, ta chọn các ảnh từ phim hoạt hình Nhật Bản hoặc tranh minh họa kỹ thuật số có nét vẽ tương đồng. Chẳng hạn, **AnimeGANv2** đã sử dụng ba phong cách anime cụ thể (Miyazaki Hayao, Makoto Shinkai, Satoshi Kon) bằng cách trích xuất hàng nghìn khung hình chất lượng cao từ các bộ phim tương ứng [47]. Tương tự, trong luận văn này có thể thu thập khung hình từ nhiều phim anime khác nhau để phong phú dữ liệu, nhưng nên tiền xử lý chuyển về cùng **phong cách hình ảnh** (ví dụ, nếu có sự khác biệt lớn về bảng màu hoặc độ nét giữa các nguồn, cần cân bằng bằng thuật toán tiền xử lý). Mỗi ảnh anime cũng được cắt/thu phóng về độ phân giải 256×256 và tập trung vào khuôn mặt nhân vật. Trường hợp ảnh gốc có nhiều nhân vật hoặc cảnh nền phức tạp, có thể áp dụng mô hình tách nền và chỉ giữ lại nhân vật chính.

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 3400 ảnh selfie và 3400 ảnh anime để huấn luyện [48] – 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.

With the dataset prepared and the model architecture defined, we proceed to train the model end-to-end, which will be detailed in the experiment section. (Chuyển ý: Phần huấn luyện và đánh giá kết quả sẽ được trình bày ở Chương 4).

---

1 Style Transfer: A Decade Survey
https://arxiv.org/html/2506.19278v1

2 4 Enhancing Photo Animation: Augmented Stylistic Modules and Prior Knowledge Integration
https://openaccess.thecvf.com/content/ACCV2024/papers/Lu_Enhancing_Photo_Animation_Augmented_Stylistic_Modules_and_Prior_Knowledge_Integration_ACCV_2024_paper.pdf

3 36 37 38 39 40 41 42 43 44 45 AnimeGANv3: A Novel Double-Tail Generative Adversarial Network for Fast Photo Animation
https://tachibanayoshino.github.io/AnimeGANv3/

5 6 CS231n Deep Learning for Computer Vision
https://cs231n.github.io/convolutional-networks/

7 8 9 10 11 13 14 15 16 A survey on training challenges in generative adversarial networks for biomedical image analysis | Artificial Intelligence Review
https://link.springer.com/article/10.1007/s10462-023-10624-y

12 17 18 19 Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks
https://openaccess.thecvf.com/content_ICCV_2017/papers/Zhu_Unpaired_Image-To-Image_Translation_ICCV_2017_paper.pdf

20 21 22 23 Feature Weighted Cycle Generative Adversarial Network with Facial Landmark Recognition and Perceptual Color Distance for Enhanced Face Animation Generation | MDPI
https://www.mdpi.com/2079-9292/13/23/4761

24 25 26 27 28 29 30 31 32 33 Awesome StyleGAN Applications | Hippocampus's Garden
https://hippocampus-garden.com/stylegans/

34 williamyang1991/DualStyleGAN: [CVPR 2022] Pastiche Master
https://github.com/williamyang1991/DualStyleGAN

35 A Novel Double-Tail Generative Adversarial Network for Fast Photo ...
https://www.jstage.jst.go.jp/article/transinf/E107.D/1/E107.D_2023EDP7061/_article/-char/ja

46 AnimeGAN: A Novel Lightweight GAN for Photo Animation
https://www.researchgate.net/publication/341634830_AnimeGAN_A_Novel_Lightweight_GAN_for_Photo_Animation

47 AnimeGANv2
https://tachibanayoshino.github.io/AnimeGANv2/

48 selfie2anime - Kaggle
https://www.kaggle.com/datasets/arnaud58/selfie2anime