**TRƯỜNG ĐẠI HỌC BÁCH KHOA TP HCM**

**KHOA ĐIỆN – ĐIỆN TỬ**

**Bộ môn Viễn Thông**

****

**LUẬN VĂN TỐT NGHIỆP**

**SỬ DỤNG KỸ THUẬT HỌC SÂU KHÔI PHỤC HÌNH ẢNH MẤT MÁT THÔNG TIN DO BỊ BÔI XOÁ**

*Hướng dẫn:* **THS. ĐẶNG NGỌC HẠNH**

*Sinh viên thực hiện*: **NGUYỄN THÁI HỮU HUY**

MSSV **: 1810177**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021

NHẬN XÉT CỦA HỘI ĐỒNG XÉT DUYỆT

KẾT LUẬN: *(đánh dấu X vào ô chọn)*

Duyệt thông qua

Không thông qua

Ý kiến đề nghị:

Tp. HCM, ngày …… tháng …… năm …

Hội đồng xét duyệt

MỤC LỤC

[DANH MỤC CÁC HÌNH VẼ iv](#_Toc91101269)

[DANH MỤC CÁC BẢNG BIỂU v](#_Toc91101270)

[DANH MỤC CÁC TỪ VIẾT TẮT vi](#_Toc91101271)

[CHƯƠNG 1. GIỚI THIỆU 1](#_Toc91101272)

[1.1 đặt vấn đề 1](#_Toc91101273)

[1.2 tính cấp thiết của đề tài 1](#_Toc91101274)

[1.3 mục tiêu của đề tài 2](#_Toc91101275)

[1.4 nội dung nghiên cứu 3](#_Toc91101276)

[1.5 phương pháp luận và phương pháp nghiên cứu 3](#_Toc91101277)

[CHƯƠNG 2. TỔNG QUAN 4](#_Toc91101278)

[2.1 khái niệm 4](#_Toc91101279)

[2.2 lịch sử hình thành và phát triển 5](#_Toc91101280)

[2.3 các phương pháp trong inpainting hình ảnh 6](#_Toc91101281)

[2.3.1 Phương pháp truyền thống 7](#_Toc91101282)

[2.3.2 Kỹ thuật dựa trên học sâu 10](#_Toc91101283)

[2.3.3 Deep Convolutional Generative Adversarial Network 14](#_Toc91101284)

[2.4 các chỉ số đánh giá chất lượng mô hình sinh ảnh 18](#_Toc91101285)

[2.4.1 Mean Absolute Error (MAE): 18](#_Toc91101286)

[2.4.2 Frechet Inception Distance (FID) 19](#_Toc91101287)

[2.4.3 Structural Similarity Index Measurement (SSIM) 20](#_Toc91101288)

[2.4.4 Peak signal to noise ratio (PSNR) 20](#_Toc91101289)

[CHƯƠNG 3. THỰC HIỆN ĐỀ TÀI 22](#_Toc91101290)

[3.1 tiến độ thực hiện đề tài 22](#_Toc91101291)

[3.2 bố cục dự kiến của luận văn 23](#_Toc91101292)

[CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU SƠ KHỞI 24](#_Toc91101293)

[4.1 dữ liệu celebfaces attributes dataset (celeba) 24](#_Toc91101294)

[4.2 tiến hành 24](#_Toc91101295)

[4.3 kết luận và hướng phát triển 30](#_Toc91101296)

[TÀI LIỆU THAM KHẢO 32](#_Toc91101297)

[PHỤ LỤC A 34](#_Toc91101298)

# DANH MỤC CÁC HÌNH VẼ

Hình 2-1: Ví dụ về image inpainting. Trái: Ảnh gốc. Giữa: Ảnh đầu vào. Phải: Ảnh kết quả inpainting. Ảnh được lấy từ [4] 4

Hình 2-2: Các loại biến thể trong image inpainting. Từ trái sang: khối, văn bản, nhiễu, vật thể, mặt nạ ngẫu nhiên, vết xướt [5] 5

Hình 2-3-1-1: Ví dụ về inpainting sử dụng kỹ thuật dựa trên khuyếch tán [9]. 7

Hình 2-3-2-1: Inpainting hình ảnh bằng kỹ thuật dựa trên bản vá [10] 9

Hình 2-3-2-2: Cấu trúc mạng GAN 11

Hình 2-3-2-3: Cấu trúc của VAE [12] 14

Hình 2-3-3: Cấu trúc mạng DCGAN 14

Hình 2-3-3-1-a Ví dụ về strided convolutional layers 15

Hình 2-3-3-1-b: Ma trận 2x2 upsamples thành ma trận 5x5 16

Hình 4-2-a: Cấu trúc mạng sinh. 25

Hình 4-2-b: Cấu trúc mạng phân biệt 27

29

Hình 4-2-c: Đồ thị hàm mất mát của mô hình sau 87 epochs 29

Hình 4-2-d: Kết quả huấn luyện sau 40 epochs 30

# DANH MỤC CÁC BẢNG BIỂU

Bảng 3‑1: Tiến độ thực hiện đề tài 22

# DANH MỤC CÁC TỪ VIẾT TẮT

AI Artificial Intelligence

ML Machine Learning

DL Deep Learning

IDE Integrated Development Environment

CNNs Convolutional Neural Networks

MLP Multilayer Perceptron

RGB Red Green Blue

GAN Generative Adversarial Networks

VAE Variational Autoencoder

DCGAN Deep Convolutional Generative Adversarial Network

# GIỚI THIỆU

## Đặt vấn đề

Ngày nay, thuật ngữ trí tuệ nhân tạo, AI/ML không còn xa lạ với chúng ta. Trí tuệ nhân tạo nói chung thị giác máy tính nói riêng là lĩnh vực phát triển rất mạnh với nhiều ứng dụng thực tế giải quyết hầu hết các bài toán, các vấn đề then chốt trong đời sống xã hội hiện nay. Nhờ sự phát triển mạnh mẽ đó hàng loạt sản phẩm công nghệ cao ra đời như xe tự hành của Tesla, hệ thống chấm công, chương trình máy tính chơi cờ vây AlphaGo của Google Deepmind, hệ thống AI dự đoán bệnh COVID-19 qua tiếng ho, hệ thống dịch tự động, xoá chi tiết không mong muốn khỏi ảnh, … , là một trong những vô vàn ứng dụng hữu ích của trí tuệ nhân tạo.

Trí tuệ nhân tạo đã xuất hiện từ rất lâu nhưng chỉ những năm gần đây thì chúng mới thực sự bùng nổ nhờ sự phát triển mạnh mẽ của phần cứng, khi mà khả năng tính toán của máy tính được nâng lên một tầm cao mới và lượng dữ liệu khổng lồ được thu thập bởi các hãng công nghệ lớn, học máy đã tiến thêm một bước dài và một lĩnh vực mới được ra đời gọi là Deep Learning (DL) hay học sâu. Học sâu đã giúp máy tính thực thi những việc tưởng chừng như không thể vào 10 năm trước: phân loại cả ngàn vật thể khác nhau trong các bức ảnh, tự tạo chú thích cho ảnh, bắt chước giọng nói và chữ viết của con người, giao tiếp với con người hay thậm chí có thể sáng tác thơ [1]. Sự thành công to lớn của học sâu trong hầu hết các lĩnh vực là không thể phủ nhận, bên cạnh những thành tựu to lớn trong xử lý ngôn ngữ tự nhiên, xử lý tiếng nói, xử lý âm thanh thì xử lý hình ảnh luôn là một chủ đề nghiên cứu và ứng dụng gặt hái được rất nhiều thành công.

## Tính cấp thiết của đề tài

Lĩnh vực xử lý ảnh không phải là một lĩnh vực mới và được nghiên cứu rất rộng rãi trên thế giới và có nhiều ứng dụng thực tiễn. Một số ứng dụng của xử lý ảnh có thể kể đến như nén ảnh, phân đoạn ảnh, nâng cao chất lượng hình ảnh, phục hồi ảnh, … Trên thực tế, khi chúng ta đi du lịch, chụp những tấm ảnh để lưu giữ kỷ niệm và nghĩ rằng đó là một tấm ảnh đẹp, nhưng khi về đến nhà chúng ta mới phát hiện ra có người lạ đi ngang làm mờ nhoè thậm chí che cả mặt chúng ta hay có rất nhiều tài liệu hình ảnh được chụp từ điện thoại, máy tính tuy nhiên do nhiều nguyên nhân như giới hạn của thiết bị, mức độ ánh sáng, … làm cho ảnh bị nhiễu, mờ hay trong quá trình lưu trữ qua thời gian chất lượng ảnh sẽ giảm đi hay bị trầy xướt làm mất thông tin. Có thể nói thông tin dạng ảnh là một dạng dữ liệu quan trọng trong đời sống xã hội. Chính vì vậy, từ những vấn đề nhức nhói đó một câu hỏi được đặt ra phải có phương pháp nào đó để khôi phục hình ảnh sao cho ảnh tạo ra phải tự nhiên và chân thật nhất.

Từ những vấn đề nêu trên, em xin chọn đề tài: “**Kỹ thuật học sâu khôi phục hình ảnh bị bôi xoá mất thông tin**” để làm luận văn tốt nghiệp, nhằm áp dụng kỹ thuật học sâu vào bài toán khôi phục hình ảnh và phát triển ứng dụng để thử nghiệm với hình ảnh.

Với sự bùng nổ của các bộ dữ liệu lớn, đối tượng nghiên cứu của luận văn là hình ảnh với các chi tiết bị biến dạng, thiếu được tạo từ các bộ dữ liệu mặt nạ đã được công bố bởi Nvidia [2]. Trong quá trình nghiên cứu của luận văn dự kiến có sử dụng các ngôn ngữ lập trình như Python, Javascript, MySQL và các môi trường phát triển tích hợp (Intergrated Development Environment – IDE) cho các ngôn ngữ này.

Do giới hạn về mặt thời gian và phần cứng, phạm vi luận văn chỉ nghiên cứu các kỹ thuật đã được công bố trên các bài báo khoa học sẵn có trong chủ đề image inpainting theo hướng tiếp cận học sâu. Đầu vào mô hình là hình ảnh bị thiếu chi tiết, mất dạng, đầu ra của mô hình là hình ảnh đã được khôi phục các chi tiết bị mất. Áp dụng mô hình kết hợp với các kỹ thuật để xử lý dữ liệu cho những hình ảnh phù hợp. So sánh các kết quả và lựa chọn kỹ thuật sao cho đem lại kết quả tốt nhất.

## Mục tiêu của đề tài

- Mục tiêu tổng quát:nghiên cứu sử dụng mô hình encoder-decoder và những kỹ thuật liên quan để khôi phục hình ảnh.

- Mục tiêu cụ thể:

* Hoàn thành đề tài trong thời gian kế hoạch
* Tìm hiểu các kỹ thuật trong image inpainting theo hướng tiếp cận học sâu
* Nắm được kiến thức trong đề tài
* Phát triển được ứng dụng để thử nghiệm mô hình
* Đánh giá được kết quả về mức độ hiệu khôi phục của ảnh trước và sau khi sử dụng mô hình

## Nội dung nghiên cứu

Nội dung nghiên cứu chủ yếu của luận văn tập trung vào các kỹ thuật đã và đang được sử dụng trong image inpainting tập trung theo hướng tiếp cận học sâu, tìm hiểu về các bộ dữ liệu phổ biến được sử dụng trong chủ đề image inpainting; ứng dụng các ngôn ngữ lập trình để xây dựng mô hình khôi phục hình ảnh đồng thời xây dựng ứng dụng kiểm thử mô hình.

## Phương pháp luận và phương pháp nghiên cứu

- Khảo sát, phân tích và hệ thống hoá nội dung các tạp chí, bài báo khoa học về chủ đề image inpainting những năm gần đây.

- Tìm hiểu những kỹ thuật đã và đang được sử dụng để tìm ra hạn chế.

- Lựa chọn kỹ thuật, giải thuật phù hợp để thực thi mô hình.

- So sánh kết quả của mô hình lựa chọn với các mô hình khác để vừa đánh giá chất lượng vừa nghiên cứu tìm ra nội dung mới có thể cải tiến nếu có.

- Thiết kế ứng dụng để kiểm thử mô hình.

# TỔNG QUAN

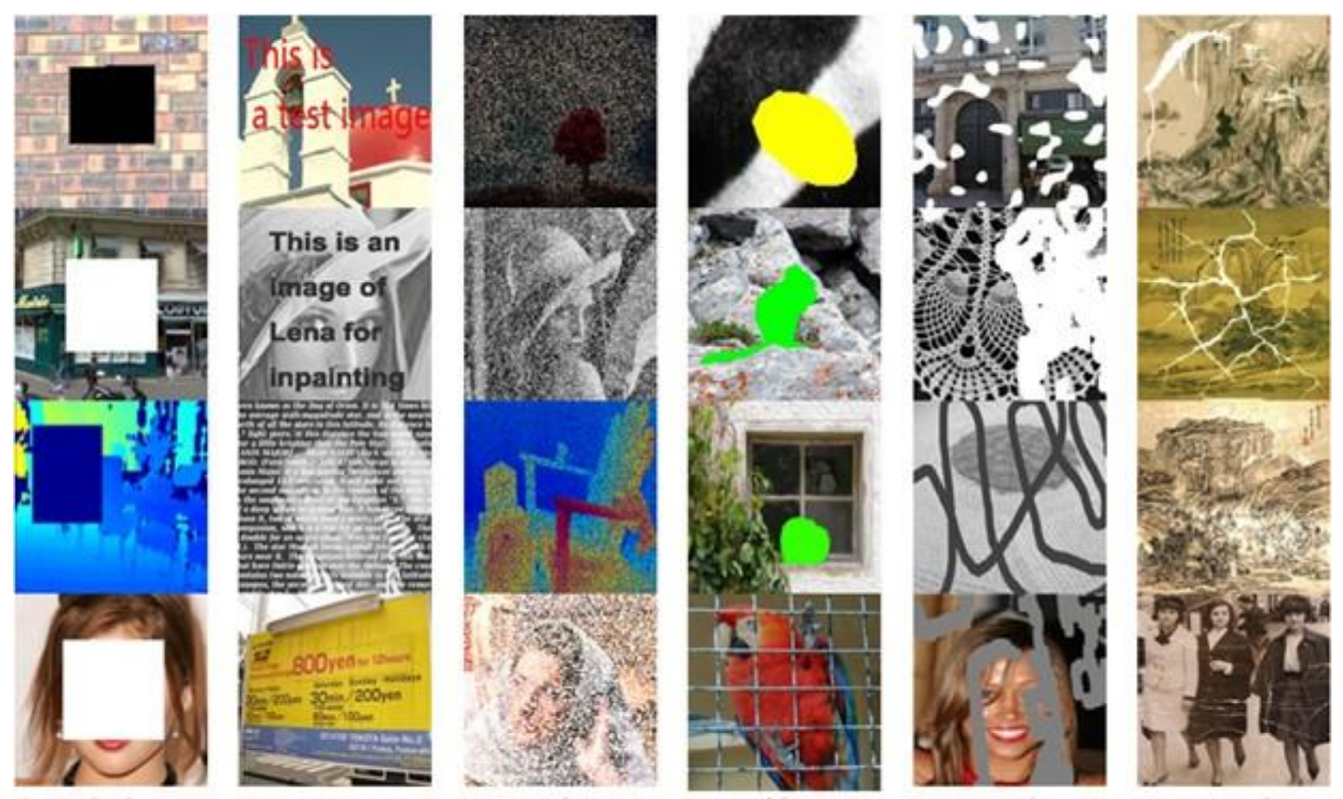
## Khái niệm

Inpainting hình ảnh là một nhánh của xử lý ảnh, là quá trình cho phép điền những dữ liệu bị thiếu trong một vùng của ảnh đầu vào bằng cách ước tính thông tin từ dữ liệu phụ trợ hoặc từ các vùng xung quanh, hoặc từ các vùng nằm ngoài [3]. Mục đích của việc inpainting có thể là để khôi phục các phần bị hư hỏng của hình ảnh (ví dụ: một tấm ảnh cũ có các nếp gấp và vết xướt) hoặc đơn giản là để loại bỏ những yếu tố không mong muốn có trong ảnh (ví dụ: micro xuất hiện trong khung hình) hoặc có thể do nhiều yếu tố khác. Xem hình 2-1.



Hình 2-1: Ví dụ về image inpainting. Trái: Ảnh gốc. Giữa: Ảnh đầu vào. Phải: Ảnh kết quả inpainting. Ảnh được lấy từ [4]

Hầu hết các ảnh đầu vào có các mặt nạ được cung cấp bởi người dùng, vì vậy vị trí của các mặt nạ không phải là vấn đề trong inpainting. Xoá các vật thể khỏi hình ảnh đang là vấn đề được công nhận rộng rãi và là một hướng đi trong nghiên cứu về thị giác máy tính. Ứng dụng của inpainting hình ảnh rất nhiều như khi chúng ta muốn chia sẻ ảnh lên mạng xã hội nhưng ảnh lại chứa các chi tiết thừa như ký tự hoặc biểu tượng cảm xúc việc đó dẫn đến ảnh có thể bị thay đổi ngữ nghĩa thì lúc này inpaiting là một giải pháp tuyệt vời. Inpainting hình ảnh cũng thường được sử dụng để phân biệt ảnh giả, loại bỏ nhiều biến thể khác nhau như văn bản, các khối, nhiễu, vết xướt, những đường thẳng hoặc bất kỳ loại mặt nạ nào, ... . Hình 2-2 biểu diễn các loại biến thể khác nhau đang tồn tại.



Hình 2-2: Các loại biến thể trong image inpainting. Từ trái sang: khối, văn bản, nhiễu, vật thể, mặt nạ ngẫu nhiên, vết xướt [5]

## Lịch sử hình thành và phát triển

Việc sử dụng kỹ thuật inpainting hiện đại có thể bắt nguồn từ Pietro Edwards (1744 – 1821), giám đốc phục hồi ảnh công cộng ở Venice, Ý. Sử dụng phương pháp tiếp cận khoa học, Edwards tập trung nỗ lực khôi phục theo chủ ý của những nghệ sĩ. [6]

Trong Hội nghị Quốc tế năm 1930 về Nghiên cứu các phương pháp khoa học để kiểm tra và bảo quản các tác phẩm nghệ thuật, phương pháp tiếp cận hiện đại đối với inpainting đã được thành lập. Helmut Ruhemann (1891 – 1973), một nhà phục chế và bảo tồn người Đức, đã dẫn đầu các cuộc thảo luận về việc sử dụng inpainting trong bảo tồn. Helmut Ruhemann là người dẫn đầu trong việc hiện đại hoá việc trùng tu và bảo tồn. Đóng góp lớn nhất của ông cho lĩnh vực bảo tồn “là sự kiên định của ông về việc tuân theo chính xác các phương pháp của hoạ sĩ gốc, và hiểu được ý định nghệ thuật của hoạ sĩ”. Sau hơn 40 năm làm nghề bảo quản, Ruhemann đã xuất bản chuyên luận Làm sạch tranh: Vấn đề và Tiềm năng vào năm 1968. Khi mô tả phương pháp của mình. Ruhemann nói rằng “Bề mặt nên thấp hơn một chút so vơi bề mặt của sơn xung quanh để tạo độ dày cho lớp sơn ... Phương tiện sơn phải trông và hoạt động giống như môi trường ban đầu, nhưng không được tối dần theo tuổi.” Cesare Brandi (1906 – 1988) đã phát triển một loại sơn in cách tiếp cận kết hợp giữa thẩm mỹ và tâm lý học. Tuy nhiên, cách tiếp cận này chủ yếu được sử dụng bởi các nhà phục chế và bảo tồn người Ý, với thuật ngữ này trở nên phổ biến vào những năm 1990. [6]

Tiến bộ của công nghệ đã dẫn đến các ứng dụng mới của inpainting. Việc sử dụng rộng rãi các kỹ thuật kỹ thuật số bao gồm từ phương pháp in máy tính hoàn toàn tự động đến các công cụ được được sử dụng để mô phỏng quy trình theo cách thủ công. Kể từ giữa những năm 1990, quy trình inpainting đã phát triển để bao gồm cả phương tiện kỹ thuật số. Thường được gọi là nội suy hình ảnh hoặc vídeo, một hình thức ước lượng, inpainting kỹ thuật số bao gồm việc sử dụng phần mềm máy tính dựa trên các thuật toán phức tạp để thay thế các phần dữ liệu hình ảnh bị mất hoặc bị hỏng [6].

Nhờ sự phát triển của máy tính trong thế kỷ 20, việc sử dụng hàng ngày và sự phát triển của các công cụ kỹ thuật số với khả năng thao tác hình ảnh đã giúp người dùng đánh giá cao việc chỉnh sửa hình ảnh, ví dụ: khôi phục và áp dụng hiển thị hình ảnh trên màn hình và các hiệu ứng đặc biệt cho hình ảnh. Do đó, việc sử dụng inpainting hình ảnh đã trở thành kỹ thuật phục chế hiện đại nhất.

## Các phương pháp trong inpainting hình ảnh

Bằng cách sử dụng các thuật toán được phát triển gần đây, hình ảnh inpainting có thể khôi phục một cách mạch lạc cả kết cấu và các thành phần cấu trúc của hình ảnh. Các kết quả thu được chứng minh rằng các phương pháp này có thể loại bỏ các đối tượng không mong muốn ra khỏi hình ảnh mà không để lại nhiều dấu vết [5]. Đã tồn tại nhiều phương pháp trong kỹ thuật inpainting hình ảnh nhưng đa phần chúng được chia thành 2 nhóm khác nhau gồm: phương pháp truyền thống và phương pháp dựa trên học sâu (DL) [7].

### Phương pháp truyền thống

Chúng ta có thể chia những phương pháp truyền thống thành 3 nhóm gồm: kỹ thuật dựa trên khuếch tán, kỹ thuật dựa trên bản vá và kỹ thuật dựa trên bộ lọc tích chập [7].

#### Kỹ thuật dựa trên khuếch tán

Những kỹ thuật này lấp đầy những vùng bị mất bằng cách lan truyền sự nội dung của hình ảnh cục bộ đến xung quanh vùng bị mất đó [5] [7]. Quá trình phân tán sử dụng kết quả vi phân từng phần. Ngoài ra, các phương pháp này cũng được sử dụng trong nén ảnh. Phương pháp inpainting hình ảnh dựa trên khuếch tán có thể được sử dụng để loại bỏ các đối tượng không mong muốn trong một hình ảnh hoặc để sửa chửa các phần bị thiếu, hư hỏng của nó. Tuy nhiên, nhược điểm của các phương pháp này là có thể để lại vết nhơ sau khi thực hiện xong quá trình inpainting điều này có thể giải thích do sự khác biệt phương sai cục bộ và mẫu nhiễu [8]. Thêm nữa các phương pháp này nhìn chung hoạt động mạnh mẽ với những hình ảnh đơn giản, nhưng khi ảnh phức tạp chứa nhiều kết cấu và cấu trúc thì việc tìm kiếm các bản vá tương tự có thể khó khăn.

Hình 2-3-1-1: Ví dụ về inpainting sử dụng kỹ thuật dựa trên khuyếch tán [9].

Hình 2-3-1-1 ở trên là một ví dụ về việc inpainting hình ảnh sử dụng kỹ thuật dựa trên khuếch tán. Vùng màu đen ở hình 2-3-1-1a là vùng chưa biết hay là vùng bị khuyết, hình 2-3-1-1b biểu diễn kết quả inpainting, vùng bao quanh đường nét đứt được khôi phục bằng cách inpainting. Ưu điểm của kỹ thuật này là có thể lưu giữ tốt thông tin ở các cạnh, thích hợp để khôi phục các đường thẳng, các đường cong và duy trì được cấu trúc của vùng được khôi phục.

Về cơ bản, quá trình inpainting không khác gì ngoài việc khôi phục thông tin các điểm ảnh bị thiếu trong một hình ảnh và có một số giả định phổ biến mà hầu hết các hình ảnh đều có chung. [8]

#### Kỹ thuật dựa trên bản vá

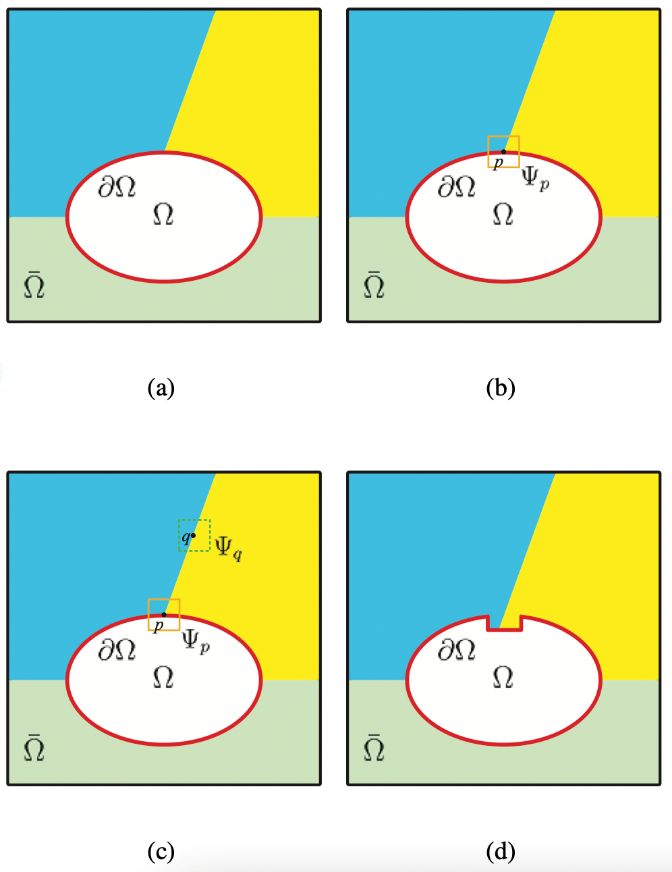
Phương pháp inpainting ảnh dựa trên bản vá bao gồm các phương pháp sử dụng tìm kiếm các vùng không bị phá huỷ trong những phần ảnh để tạo vùng ảnh được ghép vào sau đó sao chép chúng vào những vị trí tương ứng. Với cách tiếp cận này, nó nhằm mục đích tìm được bản vá lắp vào có chất lượng cao nhất. Nhiều phương pháp trong inpainting hình ảnh được đề xuất sử dụng phương pháp dựa trên bản vá.

Để thực hiện thuật toán này, cần phải có một thuật toán về cách chuyển các vùng theo lí thuyết sang các vùng được lên kế hoạch để định hướng. Một điểm quan trọng khác trong phương pháp này là việc lựa chọn thuật toán sẽ so sánh các vùng ở các phần khác nhau của ảnh với các vùng mong muốn được tạo.

Trong khi các phương pháp dựa trên bản vá hoạt động tốt, chúng giả định rằng thông tin trong phần thiếu sót của bức tranh nằm ở vị trí khác trong bức ảnh. Một vấn đề khác với các phương pháp khác là chúng đòi hỏi sức mạnh xử lý nhiều hơn vì là phương pháp dựa trên việc tìm kiếm và so sánh [8].

Hình 2-3-2-1 bên dưới là cách tiếp cận sử dụng kỹ thuật dựa trên bản vá. Với hình 2-3-2-1-a là ảnh gốc với vùng bị thiếu Ω, hình 2-3-2-1-b là ảnh có bản vá đích Ψp với mức độ ưu tiên cao nhất, hình 2-3-2-1-c là ảnh tìm kiếm bản vá phù hợp nhất Ψq cho Ψq trong vùng đã biết, hình 2-3-2-1-d: inpainting bản vá Ψq bằng Ψq và cập nhật δΩ bên trong. [10]

Các phương pháp tiếp cận dựa trên bản vá truyền bá các bản vá hình ảnh vào vùng không xác định bằng bản vá từ các vùng đã biết. Chúng thường mang lại kết quả tốt hơn, đặc biệt là khi sơn các vùng không xác định [10]. Phương pháp này cũng giống hầu hết các phương pháp khác chỉ tốt với trường hợp ảnh đơn giản, không có nhiều chi tiết, khi thực hiện inpainting với những ảnh có cấu trúc phức tạp như ảnh người, cảnh thì ảnh sau khi khôi phục cho kết quả tệ, có thể bị loè, mờ hoặc làm sai ngữ cảnh của bức ảnh.



Hình 2-3-2-1: Inpainting hình ảnh bằng kỹ thuật dựa trên bản vá [10]

#### Kỹ thuật dựa trên bộ lọc tích chập

Những kỹ thuật này lấp đầy vùng bị thiếu bằng cách xoay vùng lân cận của điểm ảnh bị hỏng với kernel thích hợp. Các thuật toán dựa trên bộ lọc tích chập, cung cấp một cách dễ dàng, nhanh chóng và chính xác để xâm nhập vào các khu vực nhỏ, tuy nhiên bị ảnh hưởng bởi cùng một vấn đề, khi khu vực bị nhiễm sơn tăng lên làm mờ được thêm vào kết quả. Những bộ lọc này cũng không thể tạo các đối tượng mới không được tìm thấy trong hình ảnh nguồn.

### Kỹ thuật dựa trên học sâu

#### Mạng thần kinh tích chập

Mạng thần kinh tích chập là một kiến trúc mạng thần kinh được sử dụng rộng rãi, đặc biệt nhờ vào sự bùng nổ của những bộ dữ liệu lớn, đạt thành công đáng kinh ngạc của nó trong các nghiên cứu về thị giác máy tính. Trong kiến trúc này sử dụng những tầng như tầng tích chập, tầng gộp lại, tầng làm phẳng. CNNs là các cấu trúc chỉ đạt được các đặc điểm quan trọng của hình ảnh bằng cách áp dụng quy trình tích chập sử dụng thước đo hai chiều và thu được mức độ thông tin khác nhau trong hình ảnh [8]. CNNs cung cấp cấu trúc chỉ đạt được các đặc điểm quan trọng của hình ảnh bằng cách áp dụng các tầng tích chập sử dụng thước đo hai chiều và thu được thông tin khác nhau trong hình ảnh. CNNs tận dụng lợi thế của các mẫu phân cấp trong dữ liệu bằng cách đi theo một đường dẫn khác với cấu trúc MLP. Một số phương pháp đã được đề xuất để inpainting hình ảnh bằng cách sử dụng CNNs giúp khôi phục những ảnh bị thiếu với các khối ở giữa ảnh cho độ chính xác cao trong điều kiện cấu trúc và kết cấu chi tiết tốt.

Trong những tác vụ của inpainting, nơi mà những ảnh màu RGB được sử dụng, những bộ lọc tích chập 2D sẽ được áp dụng riêng biệt cho từng kênh màu. Trong suốt quá trình xử lý tích chập sẽ áp dụng những bộ lọc thường sử dụng như 3x3, 5x5, 7x7 vào bên trong ma trận, quá trình padding được áp dụng trên những góc của ảnh.

Những lớp pooling thường được dùng giữa các lớp tích chập để làm giảm kích thước dữ liệu nhưng vẫn giữ được các thuộc tính quan trọng. Kích thước dữ liệu giảm giúp giảm quá trình tính toán của mô hình.

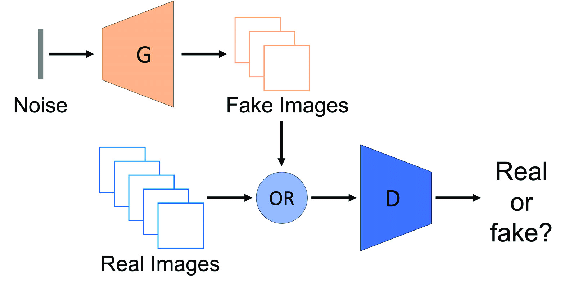
Lớp làm phẳng là lớp làm cho tensors trong cấu trúc CNN được vector hoá để sử dụng trong lớp fully connected thường được sử dụng trong cấu trúc MLP. Lớp này sẽ tuần tự hoá tensor đến và sẽ chuyển tất cả các giá trị cho các nơron trong lớp.

Có hai kiến trúc CNN phổ biến được sử dụng cho những nghiên cứu sâu về inpainting. Đầu tiên là cấu trúc encoder-decoder, và cái còn lại được gọi là cấu trúc autoencoder. Các kiến trúc này thực hiện tốt các tác vụ thị giác máy tính và cũng cho thấy kết quả đầy triển vọng đối với các vấn đề khác của thị giác máy tính như: inpainting, khử nhiễu, nâng cao chất lượng hình ảnh. Ứng dụng đa dạng như: phân tích cảm xúc, chú thích hình ảnh, dịch máy.

#### Mạng đối kháng sinh

Nhiều phương pháp học máy khác nhau đã hình thành và phát triển theo thời gian. Khi các phương pháp học sâu bắt đầu được sử dụng thường xuyên hơn và số lượng các lĩnh vực chúng được sử dụng tăng lên, các phương pháp khác nhau được sử dụng và phát triển. Một trong những hướng phát triển nổi bật là Mạng đối nghịch tạo sinh (GAN).

Với việc sử dụng và phát triển của GANs trong nhiều lĩnh vực khác nhau, cấu trúc GAN đã dẫn đầu trong thuật toán học không giám sát. Các phương pháp inpainting sử dụng GAN nhằm mục đích tạo ra một hình ảnh có điều kiện để nhận dạng cấp cao, dựa trên các điểm ảnh thấp tổng hợp lại thành một mạng tích chập (bộ encoder-decoder). Mạng đối thủ được đào tạo nâng cao sự kết hợp giữa những điểm ảnh của ảnh được tạo ra và ảnh gốc. Hình 2-3-2-2 biểu diễn cấu trúc mạng GAN.



Hình 2-3-2-2: Cấu trúc mạng GAN

GAN cấu tạo gồm 2 mạng là mạng sinh (generator network) và mạng phân biệt (discriminator network). Trong khi mạng sinh tìm cách sinh ra các dữ liệu giống thật thì mạng phân biệt cố gắng phân biệt đâu là dữ liệu do mạng sinh sinh ra đâu là dữ liệu gốc. Cả hai mạng sẽ cạnh tranh với nhau. Mạng sinh sẽ cố gắng đánh lừa mạng phân biệt. Đồng thời, mạng phân biệt sẽ thích nghi với dữ liệu giả vừa tạo ra. Thông tin thu được sẽ cải thiện mạng sinh, và cứ tiếp tục như vậy. [11]

Mạng phân biệt là một bộ phân loại nhị phân nhằm phân biệt xem đầu vào **x** là dữ liệu thật (dữ liệu từ datasets) hay là giả (được tạo ra từ mạng sinh). Đối với mạng sinh, trước tiên nó tạo ra vài tham số ngẫu nhiên . Ta thường gọi z như là một biến tiềm ẩn. Mạng sinh mục tiêu chính của nó là cố đánh lừa mạng phân biệt để phân biệt là dữ liệu thật. [11]

Hàm loss của mô hình GANs có dạng:

Trong đó:

* z là nhiễu đầu vào của mạng sinh
* x là dữ liệu thật
* G là ký hiệu của mạng sinh (generator network)
* D là ký hiệu của mạng phân biệt (discriminator network)
* G(z) là ảnh được sinh ra từ mạng sinh
* D(x) là giá trị dự đoán của mạng phân biệt để xem ảnh x là thật hay giả
* D(G(z)) là giá trị dự đoán xem ảnh sinh ra từ mạng sinh là thật hay giả
* E là kỳ vọng

Thông thường, đầu ra của mạng phân biệt là một số vô hướng dự đoán cho đầu vào x, chẳng hạn như sử dụng một tầng kết nối đầy đủ với kích thước ẩn và sau đó được đưa qua hàm sigmoid để nhận được xác suất dự đoán . Giả sử nhãn y cho dữ liệu thật là 1 và 0 cho dữ liệu giả. Ta sẽ huấn luyện mạng phân biệt để cực tiểu hoá hàm mất mát entropy chéo [11], nghĩa là

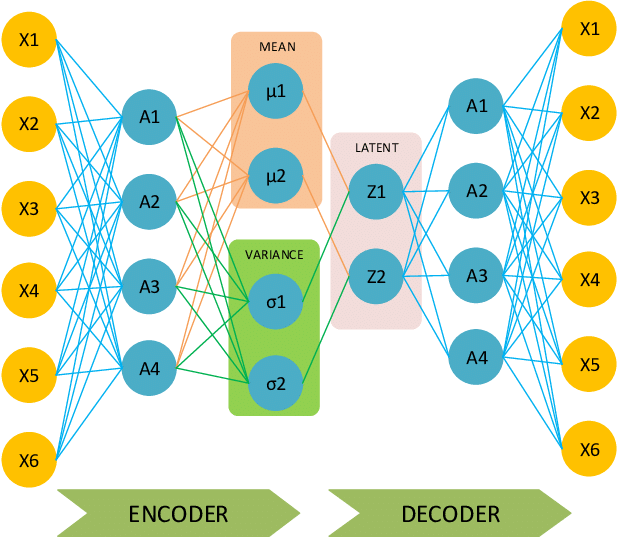
Mục tiêu của mạng sinh là đánh lừa mạng phân biệt để phân loại loại dữ liệu thật, hay là D(G(z)) ~ 1. Ta sẽ cập nhật thông số mạng sinh G nhằm cực đại hoá hàm mất mát entropy chéo, tương đương với minimize (1 – D(G(z))):

Từ hàm loss của GAN việc huấn luyện mạng sinh và mạng phân biệt là đối nghịch nhau, trong khi D cố gắng maximize hàm loss thì G lại muốn minimize hàm loss. Quá trình huấn luyện mạng GAN kết thúc khi mô hình đạt đến trạng thái cân bằng của 2 mô hình.

Các phương pháp dựa trên GAN cung cấp một bổ sung tốt cho hiệu suất của các thuật toán inpainting, nhưng tốc độ huấn luyện thấp và cần máy tính có hiệu suất cao do mô hình cần một số lượng tham số lớn để tính toán. Qua quá trình dài thử nghiệm, người ta đã phân tích được rằng các phương pháp dựa trên CNN và GANs cho kết quả tốt hơn trong việc đánh giá kết quả thực tế bằng hình ảnh.

#### Bộ biến thể mã hoá tự động (Variational Autoencoder)

Mạng GANs đã cho những kết quả tốt trong việc khôi phục hình ảnh phức tạp. Ngoài ra, còn một phương pháp mới gần đây được sử dụng là bộ biến thể mã hoá tự động – VAE. VAE giả định rằng dữ liệu nguồn có một số loại phân phối xác suất cơ bản (chẳng hạn như Gaussian), sau đó cố gắng tìm ra các tham số của phân phối. Việc triển khai một bộ biến thể mã hoá tự động khó hơn nhiều so với việc triển khai bộ mã hoá tự động. Một công dụng chính của trình biến thể mã hoá tự động là tạo dữ liệu mới có liên quan đến dữ liệu nguồn ban đầu. Một bộ biến thể mã hoá tự động (VAE) là một hệ thống tổng hợp và phục vụ một mục đích tương tự như mạng GAN.



Hình 2-3-2-3: Cấu trúc của VAE [12]

Mạng VAE gồm 1 cặp mạng kết nối với nhau, một mạng nhận đầu vào sau đó cắt giảm những đại diện nhỏ hơn (bộ mã hoá – encoder) và một mạng chuyển đổi từ những đại diện nhỏ hơn trở lại cấu trúc ban đầu của chúng là bộ giải mã – decoder nơi có có các không gian tiềm ẩn liên tục, dễ dàng lấy mẫu ngẫu nhiên và nội suy bởi vì đầu ra bộ mã hoá chứa 2 vectors: một vector trung vị (μ) và một vector độ lệch chuẩn/ phương sai (σ) [12].

Bộ mã hoá là một mạng nơron tạo ra một đại diện của dữ liệu . Theo thuật ngữ mô hình xác suất, mạng suy luận tham số phần sau gần đúng của các biến tiềm ẩn . Mạng suy luận tham số tới các phân phối .

Bộ giải mã là một mạng nơron học cách tái tạo lại dữ liệu x cho một biểu diễn z. Về mặt mô hình xác suất, khả năng dữ liệu x cho các biến tiềm ẩn z được tham số hoá bởi một mạng tổng hợp. Mạng tổng hợp xuất các tham số cho phân phối Đầu ra của bộ mã hoá cũng chính là đầu vào của bộ giải mã.

### Gated Convolution

Đa số các mô hình sinh ảnh sâu thường dựa trên lớp vanilla convolutions được sử dụng cho việc lấp đầy các lỗ hỏng trên ảnh vì các bộ lọc tích chập được chia sẻ tất cả không gian xử lý các pixel hoặc feature tương đồng nhau. Cho tác vụ lấp đầy lỗ, đầu vào giải quyết các pixel hợp lệ bên ngoài lỗ và những vùng mặt nạ không hợp lệ. Vanilla convolutions áp dụng same filters trên tất cả pixel hợp lệ, không hợp lệ và trộn dẫn đến việc ảnh tạo tác trực quan như ảnh bị lệch màu, nhoè, ... ngay vùng ảnh có lỗ hổng khi kiểm tra. Trong tác vụ image inpainting, các nhà nghiên cứu cân nhắc sử dụng một lớp tích chập như một chỗ chứa các bộ lọc được áp dụng vào feature map đầu vào như đầu ra. Nó học một cơ chế kiểm tra tính năng động cho từng kênh và từng vị trí không gian. Feature đầu vào được sử dụng đầu tiên để tính toán những giá trị gating ( trong đó là hàm kích hoạt sigmoid, là tham số học). Đầu ra cuối cùng là kết quả của phép nhân của tham số học và giá trị gating trong đó có thể là bất kỳ hàm kích hoạt nào.

Giả sử đầu vào là một kênh C, mỗi pixel nằm tại vị trí (y,x) trong kênh của đầu ra được tính theo công thức như sau:

Trong đó:

* x, y là toạ độ x và y của thành phần đầu ra.
* là kích thức kernel
* đại diện bộ lọc tích chập
* là inputs và outputs.

Để đơn giản hơn thì hệ số bias trong tích chập được bỏ qua. [13]

Phương trình trên chứng minh rằng trong tất cả vị trí không gian (y,x) , same filters được áp dụng để chế tạo đầu ra trong lớp vanilla convolutions. Điều này thật hữu ích cho những tác vụ như phân loại hình ảnh, nhận dạng vật thể, nơi mà tất cả pixels của ảnh đầu vào đều có giá trị, để trích xuất các đặc tính cục bộ, trong cửa sổ trượt. [13]

Gần đây, partial convlolution (tích chập từng phần) được đề xuất đáp ứng được mặt nạ và bước chuẩn hoá lại để làm tích chập bảo vệ những pixels có giá trị. Partial convolution được tính toán theo công thức như sau:

Trong đó :

* M tương ứng là mặt nạ nhị phân ( 0 là vùng pixel ảnh bị thiếu, 1 là vùng ảnh bình thường).
* là phép nhân element-wise.

Từ phương trình trên, ta có thể thấy được kết quả của phép tính partial convolution phụ thuộc vào giá trị đầu vào có giá trị (valid input). Phép tích chập được đề xuất học một cơ chế lựa chọn tính năng động cho từng kênh và từng không gian vị trí. Điều thú vị là hình ảnh hóa các giá trị kiểm soát trung gian cho thấy rằng nó học cách chọn đối tượng địa lý không chỉ theo nền, mặt nạ, phác thảo, nhưng cũng xem xét phân đoạn ngữ nghĩa trong một số kênh. Ngay cả trong sâu các lớp, tích chập gated học cách làm nổi bật các vùng bị che và phác thảo thông tin trong các kênh riêng biệt để tạo ra các kết quả inpainting tốt hơn.

### Batch normalization layers

Batch normalization là một trong những phương thức normalization được sử dụng phổ biến trong mô hình học sâu (deep learning). Nó cho phép đào tạo nhanh hơn và ổn định hơn các mạng nơron sâu bằng cách ổn định sự phân bố đầu vào trong các layers trong quá trình huấn luyện. Để cải thiện việc huấn luyện mô hình cần phải giảm Internal Covariate Shift bằng cách kiểm soát trung vị và phương sai của dữ liệu đầu vào các layer, điều chỉnh phân phối của dữ liệu cho đồng bộ. Batch normalization được áp dụng cho từng lớp riêng lẻ. Trong mỗi vòng lặp huấn luyện, tại mỗi lớp cần tính giá trị kích hoạt sau đó tính chuẩn hoá giá trị kích hoạt của mỗi nút bằng việc trừ đi giá trị trung bình và chia cho độ lệch chuẩn. [11] Batch normalization chuẩn hoá những giá trị tại lớp x theo công thức như sau:

Trong đó, là giá trị trung bình và là độ lệch chuẩn của các mẫu trong minibatch. Sau khi áp dụng batch normalization những giá trị kích hoạt của minibatch có giá trị trung bình bằng không và phương sai đơn vị.

Lợi ích của việc sử dụng Batch Normalization là:

* Làm giảm internal covariate shift và tăng tốc độ huấn luyện cho mô hình học sâu.
* Cách tiếp cận này làm giảm sự phụ thuộc của gradients vào tỉ lệ các tham số hoặc giá trị ban đầu của chúng dẫn đến hệ số học cao hơn mà không có nguy cơ phân kỳ.
* Batch Normalization có thể sử dụng các cơ chế phi tuyến bảo hoà bằng cách ngăn mạng khỏi bị kẹt trong chế độ bão hoà. [14]

## Spectral Normalization

Nếu mạng phân biệt là K-Lipschitz liên tục thì với tất cả x và y trong :

Trong đó: là chuẩn khoảng cách Euclide. Nếu L là nhỏ nhất thì được gọi là hằng số Lipschitz của f và có nghĩa là Lip(f). Trong trường hợp của chúng ta, f là một mạng thần kinh điều chúng ta muốn là nó trở thành một Lipschitz liên tục với Lip(f) nhỏ. Điều này cung cấp một giới hạn trên sự nhiễu loạn của các đầu ra. Tính liên tục Lipschitz cũng có thuộc tính sau:

Cho f=g ∘ h. Nếu g và h là Lipschitz liên tục, thì f cũng là Lipschitz liên tục với Lip(f) ≤ Lip(g)Lip(h)

Do đó, miễn là chúng ta làm cho mỗi thành phần của mạng nơron Lipschitz liên tục với hằng số Lipschitz nhỏ, thì toàn bộ mạng nơron cũng sẽ liên tục với hằng số Lipschitz nhỏ.

Hãy xét một một lớp kết nối đầy đủ. Để đơn giản chúng ta bỏ qua hệ số bias, do đó FC(x) = Wx đối với ma trận trọng số W. Có thể chứng minh rằng FC có hằng số Lipschitz Lip(FC) = định mức phổ của W, tương đương giá trị singular lớn nhất của W.

Chuẩn mức quang phổ hoạt động bằng cách chuẩn hoá định mức phổ của W:

Ma trận chuẩn hoá có định mức phổ bằng 1, do đó Lip(FC) cũng bằng 1.

Ý tưởng tương tự cũng có thể áp dụng cho các lớp tích chập. Lưu ý rằng tích chập là các phép toán tuyến tính, do đó có thể viết lại chúng dưới dạng ma trận có kích thước thích hợp và sau đó sử dụng lại ý tưởng trên.

## Các chỉ số đánh giá chất lượng mô hình sinh ảnh

Bài toán sinh ảnh nói chung và khôi phục hình ảnh bị bôi xoá do bị che mất thông tin (image inpainting) là một trong những bài toán cũng được quan tâm rất nhiều trong thời gian gần đây. Cùng với sự phát triển đó hàng loạt mô hình mới được công bố, vậy câu hỏi được đặt ra làm sao để so sánh các mô hình khác nhau khi thực hiện trên cùng một bộ dữ liệu hay đánh giá chất lượng mô hình. Để giải quyết vấn đề trên, đã có phương pháp khác nhau được đề xuất song nhìn chung, đánh giá ảnh sinh ra dựa trên hai phương pháp:

* Chất lượng ảnh: ảnh sinh ra có chất lượng cao, càng giống với dataset càng tốt
* Độ đa dạng của ảnh: bộ sinh dữ liệu cần sinh ra được càng nhiều ảnh khác nhau thuộc nhiều lớp khác nhau [15].

Dựa trên đó có nhiều chỉ số để đo chất lượng ảnh ra đời và thường được sử dụng như sau:

### Mean Absolute Error (MAE):

Mean absolute error (sai số tuyệt đối trung bình) là phương pháp đo lường sự khác biệt giữa hai biến liên tục. Giả sử X và Y là là hai biến liên tục thể hiện kết quả dự đoán của mô hình và kết quả thực tế. MAE có công thức như sau:

Trong đó :

* yi  là giá trị thực tế
* xi là giá trị dự đoán
* n là tổng số mẫu

Tóm lại, giá trị của MAE trong bài toán image inpainting càng nhỏ càng tốt.

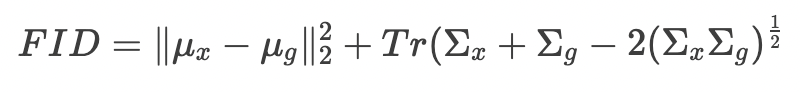
### Frechet Inception Distance (FID)

Frechet Inception Distance, gọi tắt là FID, là một số liệu dùng để đánh giá chất lượng ảnh được tạo ra, thường được dùng trong các mô hình sinh ảnh và image inpainting,...

Inception score ước tính chất lượng của một tập hợp các hình ảnh tổng hợp dựa trên mức độ hiệu quả của mô hình phân loại hình ảnh hoạt động hàng đầu Inceptionv3. Điểm số này kết hợp cả độ tin cậy của các lớp dự đoán có điều kiện cho mỗi hình ảnh tổng hợp và xác suất cận biên của các lớp dự đoán. Inception score không cho biết hình ảnh tổng hợp so với hình ảnh thực tế như thế nào. Mục tiêu khi phát triển FID score là để đánh giá các hình ảnh tổng hợp được dựa trên số liệu thống kê của một tập hợp các hình ảnh tổng hợp so với số liệu thống kê của tập hợp các hình ảnh thực tế.

Giống như inception score, FID score sử dụng mô hình inception v3, cụ thể lớp mã hoá của mô hình (lớp tổng hợp cuối cùng trước khi phân loại đầu ra của hình ảnh) để nắm bắt các features của ảnh đầu vào. Nếu chỉ số FID thấp chứng tỏ hình ảnh chất lượng thấp hơn, điểm cao hơn chứng tỏ hình ảnh chất lượng thấp hơn.

Công thức FID được tính như sau



Việc tính toán FID phụ thuộc vào Inception network, giữ đến lớp AvgPool, phần sau của mạng được lược bỏ đi.

Ta sẽ lấy tất cả các ảnh trong dataset, mỗi ảnh cho qua Inception network sẽ được 1 vector 2048x1 sau đó tiến hành tìm một multivariate gaussian distribution fit nhất tất cả vector này với mean = μx, std = x. Bên cạnh đó lấy các ảnh sinh ra bởi mạng sinh, mỗi ảnh cũng cho qua Inception network để được một vector 2048x1 sau đó tiến hành tìm một multivarite gaussian distribution fit nhất với mean = μy, std = y [16]. Để các ảnh sinh ra giống các ảnh trong dataset thì ta mong muốn hai multivariate gaussian distribution giống nhau, tức là mean và variance gần nhau.

Tóm lại, FID không âm và FID càng thấp thì hai distribution càng gần nhau tức là ảnh sinh ra càng giống với ảnh gốc.

### Structural Similarity Index Measurement (SSIM)

Chỉ số SSIM để ước lượng mức độ giống nhau giữa ảnh gốc và ảnh sinh ra. Chỉ số này tính toán dựa trên tính toán của ba độ đo là độ chói, độ tương phản và thời hạn kết cấu. Một ảnh sinh ra là tốt nếu những điểm ảnh có mức độ sáng tối khác nhau, và càng có nhiều mức độ sáng tối càng có nhiều chi tiết ảnh tạo nên chất lượng ảnh tốt, bên cạnh đó cần có sự hài hoà và cân đối tạo nên độ đa dạng.

Công thức tính toán SSIM như sau:

Trong đó:

* , là trung bình của x và y
* là phương sai của x và y
* là hiệp phương sai của x và y
* là hệ số ổn định
* là thang xám

SSIM có giá trị từ -1 đến 1, đạt giá trị bằng 1 khi hai bộ dữ liệu giống nhau. Chỉ số này càng lớn tức là mô hình càng tốt

### Peak signal to noise ratio (PSNR)

Sự tương ứng giữa ảnh gốc và ảnh được sinh ra được đánh giá thông qua giá trị của tỷ số tín hiệu lớn nhất trên nhiễu (Peak signal to noise ratio – PSNR). PSNR được định nghĩa thông qua sai số toàn phương trung bình (Mean squared error – MSE). MSE là một khái niệm trong thống kê học, nghĩa là sai số toàn phương trung bình của một phép ước lượng là trung bình của bình phương các sai số, nghĩa là sự khác nhau giữa giá trị ước lượng và giá trị thực tế.

Công thức của MSE được định nghĩa như sau:

Thông thường, nếu PSNR >= 40 dB thì ảnh sinh ra gần giống với ảnh gốc, mắt thường dường như không phân biệt được. PSNR càng cao thì chất lượng ảnh sinh ra càng tốt, khi hai ảnh giống hệt thì MSE = 0, và PSNR đi đến bằng vô cùng.

# THỰC HIỆN ĐỀ TÀI

## Tiến độ thực hiện đề tài

Bảng 3‑1: Tiến độ thực hiện đề tài

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tháng (năm 2022)**  **Dự kiến nội dung**  **thực hiện** | **1** | **2** | **3** | **4** | **5** | **6** |
| Tìm hiểu về image inpainting, ngôn ngữ lập trình Python, các thuật toán xử lý ảnh liên quan. | 1/1 – 20/1 |  |  |  |  |  |
| Tìm hiểu về các kỹ thuật đã và đang được sử dụng trong image inpainting. Tìm hiểu về học sâu, khảo sát các tập dữ liệu được sử dụng cho bài toán image inpainting | 21/1 – 31/1 | 1/2 – 10/2 |  |  |  |  |
| Đọc các bài báo khoa học về image inpainting, thử nghiệm các mô hình GANs áp dụng cho bài toán. |  | 11/2 – 28/02 |  |  |  |  |
| Tìm hiểu về mô hình DeepFillv2. Tiến hành lập trình mô hình DeepFillv2 và huấn luyện mô hình. Kiểm tra và sửa chữa lỗi |  |  | 1/3 – 1/4 |  |  |  |
| Tìm hiểu và lập trình mô hình mới.  Đánh giá kết quả các mô hình thực hiện. Kế hoạch dự định mô hình phục hồi ảnh tốt, ít bị nhoè. |  |  |  | 1/4-30/4 |  |  |
| Thực hiện viết, chỉnh sửa báo cáo luận văn, thực hiện slide trình bày. |  |  |  |  | 1/5 – 20/5 |  |
| Hoàn thiện luận văn tốt nghiệp |  |  |  |  |  | 20/5 – 23/5 |

## Bố cục dự kiến của Luận văn

Báo cáo luận văn tốt nghiệp được chia thành 4 chương như sau:

Chương 1: Giới thiệu tổng quan về đề tài, lý do chọn đề tài, phương pháp tiếp cận, đối tượng nghiên cứu, nội dung nghiên cứu và giới hạn của đề tài.

Chương 2: Giới thiệu về image inpainting, lý thuyết tổng quan của đề tài, các kỹ thuật tiếp cận của đề tài.

Chương 3: Tiến độ thực hiện đề tài, bố cục luận văn.

Chương 4: Trình bày kết quả nghiên cứu sơ khởi.

# KẾT QUẢ NGHIÊN CỨU

## Dữ liệu CelebFaces Attributes Dataset (CelebA)

CelebA là bộ dữ liệu gồm thuộc tính khuôn mặt quy mô lớn với hơn 200 nghìn ảnh người nổi tiếng, mỗi hình ảnh có 40 chú thích thuộc tính. Các hình ảnh trong tập dữ liệu này bao gồm các biến thể tư thế lớn và sự lộn xộn của phông nền. CelebA có sự đa dạng lớn, số lượng lớn và chú thích phong phú bao gồm:

* 10,177 số lượng danh tính
* 202,599 số hình ảnh khuôn mặt và
* 5 vị trí landmarks, 40 thuộc tính nhị phân cho mỗi ảnh. [17]

## Tiến hành

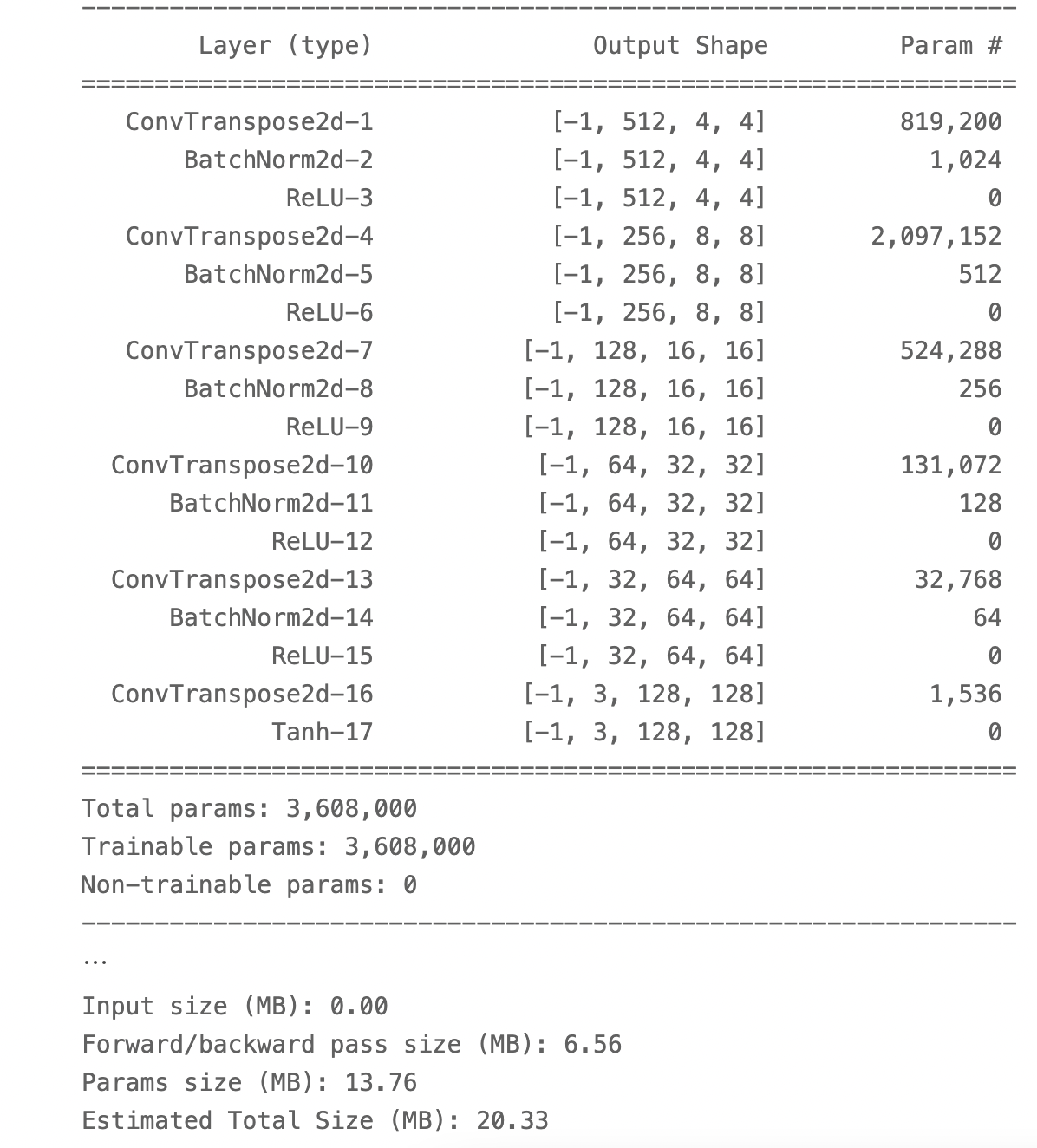
- Bước 1: Từ bài báo DCGAN, các tác giả chỉ định rằng tất cả các trọng số của mô hình sẽ được khởi tạo ngẫu nhiên từ phân phối chuẩn với giá trị trung bình = 0, độ lệch chuẩn = 0.02. [18] Hàm weights\_init bên dưới lấy một mô hình đã khởi tạo làm đầu vào và khởi động lại tất cả các lớp tích chập, tích chập chuyển vị và lớp batch normalization. Hàm này được áp dụng cho các mô hình sau khi đã khởi tạo.

- Bước 2: Xây dựng mạng sinh và mạng phân biệt.

- Bước 3: Giải thuật tối ưu Adam được chọn và hệ số học là 0.0002, hệ số beta1 cho hàm tối ưu Adam là 0.5 và hệ số beta2 là 0.999 được đặt như trong bài báo DCGAN.

- Bước 4: Huấn luyện mô hình. Chúng ta sẽ xây dựng các mini-batches cho hình ảnh thật và giả, đồng thời cũng điều chỉnh hàm mục tiêu G để tối đa hoá *log(D(G(z)).* Quá trình huấn luyện sẽ được chia thành 2 phần. Phần 1 cập nhật mạng phân biệt, phần 2 cập nhật mạng sinh.

* Phần 1: Huấn luyện mạng phân biệt, mục tiêu của mạng phân biệt là tối đa hoá xác suất phân loại chính xác đầu vào là thật hay giả. Thực tế, chúng ta muốn tối đa hoá hàm *log(D(x) + log(1-D(G(z))).* Đầu tiên, chúng ta sẽ xây dựng một loạt các thực từ tập huấn luyện chuyển tiếp qua D, tính toán hàm mất mát *log(D(x))*, sau đó tính gradients cho quá trình truyền ngược. Tiếp theo, chúng ta sẽ xây dựng ảnh giả từ mạng sinh hiện tại, chuyển tiếp những batch này qua D, tính toán hàm mất mát *(log(1 – D(G(z)))*
* Phần 2: Huấn luyện mạng sinh, mục tiêu là giảm thiểu hàm *(log(1 – D(G(z))),* để tạo những ảnh giả tốt hơn. [18]

Tổng quan cấu trúc mạng sinh như sau:

Hình 4-2-a: Cấu trúc mạng sinh.

Cấu trúc mạng sinh được lập trình gồm những lớp tích chập chuyển vị 2D theo sau đó là những lớp Batch Normalization và hàm kích hoạt ReLU và ở lớp cuối cùng được áp dụng hàm kích hoạt Tanh để trả kết quả đầu vào dữ liệu trong khoảng [-1,1].

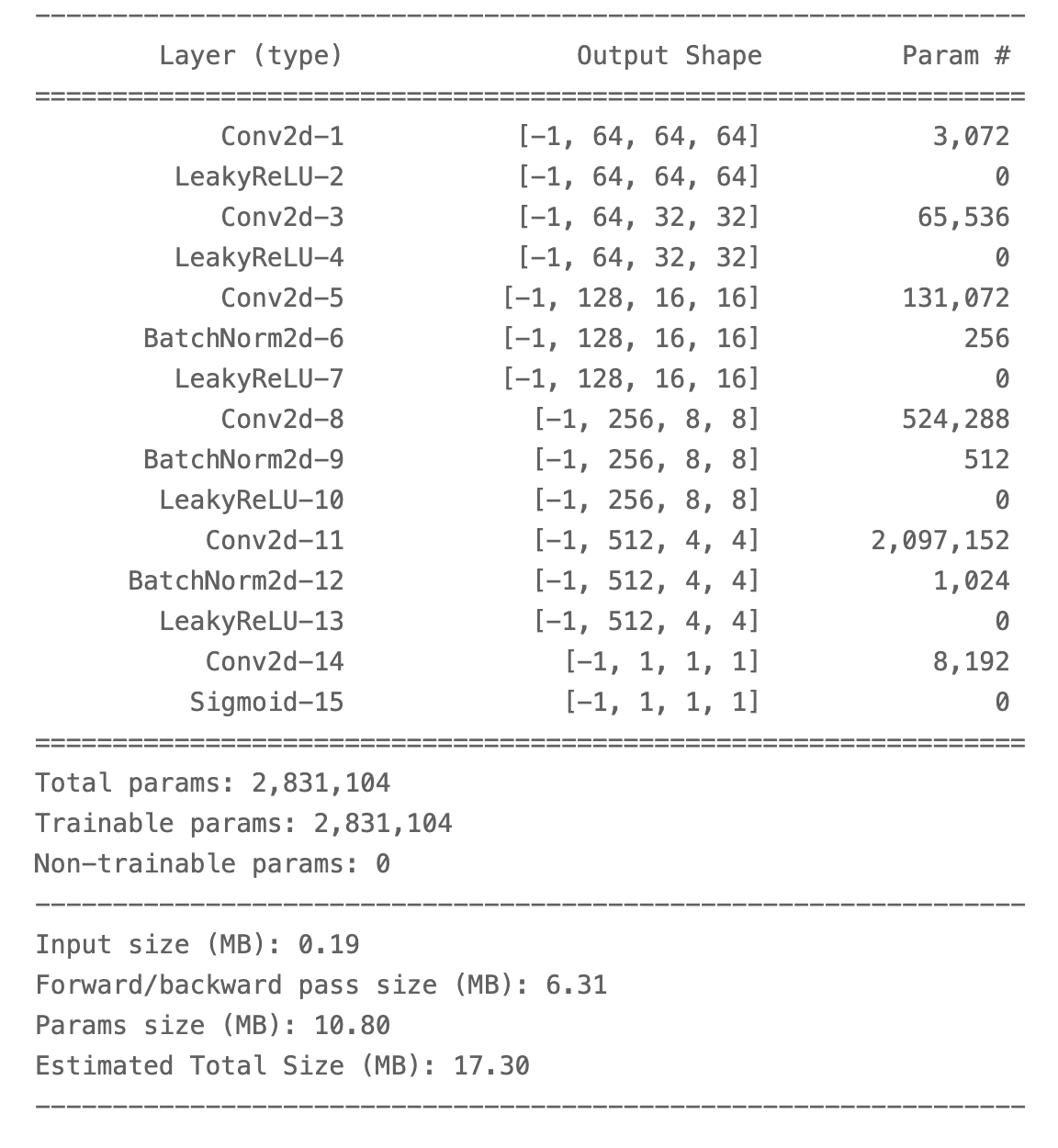
Mạng sinh là mạng fully-convolution nhận đầu vào là input noise và trả ra đầu ra là một ảnh có kích thước 3x128x128. Nó được xem như là một bộ giải mã, cung cấp cho nó một vector ẩn (latent vector) có kích thước 100 dimensions và sau đó nó được lấy mẫu ngược (upsamples) về kích thước 3x128x128.

Lớp tích chập chuyển vị 2D trong framework Pytorch có sáu 6 tham số như sau:

* Input channels
* Output channels
* Kernel of filter size
* Strides
* Padding
* Bias

Chúng ta bắt đầu với đầu ra là 512 kênh màu và chia các kênh đầu ra cho đến khối thứ 5 và ở khối cuối cùng đầu ra sẽ là 3 kênh màu. Hệ số stride bằng 2 được áp dụng tại mỗi lớp, nó giúp tăng gấp đôi đầu vào ở mỗi khối, từ 4x4 ở khối đầu tiên cho đến 128x128 ở khối cuối cùng. Cuối cùng, hàm kích hoạt Tanh được áp dụng ở lớp cuối cùng để đảm bảo rằng các giá trị pixel được đặt ở trong khoảng [-1,1].

Tổng quan cấu trúc mạng phân biệt như sau:



Hình 4-2-b: Cấu trúc mạng phân biệt

Ở hình 4-2-b, có thể thấy cấu trúc mạng phân biệt gồm những lớp tích chập theo sau là đó là hàm kích hoạt LeakyReLU. Ở khối 3, 4, 5 chứa lớp tích chập sau đó là batch norm và cuối cùng là hàm kích hoạt LeakyReLU. Ở khối cuối cùng sử dụng hàm kích hoạt Sigmoid do mạng phân biệt giống như bài toán phân loại nhị phân giúp đưa ra xác suất dự đoán ảnh được tạo ra là giả hay thật. Đầu vào của mạng phân biệt là ảnh có chiều 3x128x128, đầu ra là 0 (là ảnh giả) hoặc 1 (là ảnh thật). Hàm LeakyReLU (với độ dốc bằng 0.2) được áp dụng ở các khối

- Hàm loss Binary Cross-Entropy được sử dụng cho cả hai mạng sinh và mạng phân biệt.

- Hàm loss của mạng sinh G như sau:

# (2) Update G network: maximize log(D(G(z)))

self.G.zero\_grad()

label.fill\_(real\_label) # fake labels are real for generator cost

# Since we just updated D, perform another forward pass of all-fake batch through D

output = self.D(fake).view(-1)

# Calculate G's loss based on this output

errG = criterion(output, label)

Trong đó output là dự đoán đầu ra từ mạng phân biệt khi được cung cấp hình ảnh từ mạng sinh tạo ra, label là nhãn của ảnh bằng 1, cho mạng sinh G đánh lừa mạng phân biệt nhằm tạo ra ảnh chân thực hơn.

- Hàm loss của mạng phân biệt D như sau:

1. # Format batch
2. real\_cpu = data[0].to(self.device)
3. b\_size = real\_cpu.size(0)
4. label = torch.full((b\_size,), real\_label, device=self.device, dtype=torch.float)
5. # Forward pass real batch through D
6. # print(self.D(real\_cpu).size())
7. output = self.D(real\_cpu).view(-1)
8. # Calculate loss on all-real batch
9. # print(output.size(), label.size())
10. errD\_real = criterion(output, label)
11. # Calculate gradients for D in backward pass
12. errD\_real.backward()
13. D\_x = output.mean().item()
15. ## Train with all-fake batch
16. # Generate batch of latent vectors
17. noise = torch.randn(b\_size, self.nz, 1, 1, device=self.device)
18. # Generate fake image batch with G
19. fake = self.G(noise)
20. label.fill\_(fake\_label)
21. # Classify all fake batch with D
22. output = self.D(fake.detach()).view(-1)
23. # Calculate D's loss on the all-fake batch
24. errD\_fake = criterion(output, label)
25. # Calculate the gradients for this batch
26. errD\_fake.backward()
27. D\_G\_z1 = output.mean().item()
28. # Add the gradients from the all-real and all-fake batches
29. errD = errD\_real + errD\_fake

Hàm loss của mạng phân biệt dự đoán đầu ra thực tế (ảnh gốc) , nhãn ground truth là 1 và ảnh giả (ảnh được tạo), nhãn ground truth là 0. Khi huấn luyện mạng phân biệt thì giá trị hàm mất mát của mạng phân biệt D sẽ bằng tổng của ảnh thật và ảnh giả.

Cả 2 mạng sinh G và mạng phân biệt D cùng sử dụng hàm tối ưu Adam với hệ số học learning rate là 0.002, hệ số beta1 là 0.5 và hệ số beta2 là 0.99 để làm mượt mà quá trình huấn luyện.Với mỗi lần lặp đầu tiên ta cập nhật bộ phân biệt và sau đó đến bộ sinh.

Kết quả hàm loss của G và D sau 87 epochs huấn luyện:



Hình 4-2-c: Đồ thị hàm mất mát của mô hình sau 87 epochs

Hình 4-2-d bên dưới biểu diễn kết quả sau khi inpainting của mô hình. Hình bên trái là ảnh trước khi inpainiting đã được tạo các lớp mặt nạ, hình bên phải là ảnh sau khi tái tạo qua 40000 lần lặp. Kết quả tái tạo không được hoàn thiện do quá trình huấn luyện chỉ qua 10 epochs mô hình không học được nhiều. Để phát triển và hoàn thiện kết quả trong tương lai cần huấn luyện thêm nhiều epochs hơn nữa.

Kết quả huấn luyện sau 40 epochs:



Hình 4-2-d: Kết quả huấn luyện sau 40 epochs

## Kết luận và hướng phát triển

Chúng ta có thể thấy mạng GANs nói chung và DCGAN nói riêng được sử dụng khi sinh dữ liệu trên máy tính, có thể nói đây là một mạng nơ ron có nhiều tiềm năng khi sử dụng trong lĩnh vực xử lý ảnh và thị giác máy tính. Tuy nhiên, do tập dữ liệu học tập lớn và giới hạn về mặt phần cứng nên quá trình huấn luyện đòi hỏi rất nhiều thời gian và chi phí tính toán nên chất lượng ảnh và đường nét của ảnh sinh ra chưa được cao. Giới hạn của mô hình nằm ở việc nó chỉ học được những hình ảnh nhưng do chưa được học vị trí của các mặt nạ che khuất các vị trí thông tin trên ảnh nên khi khôi phục lại ảnh nó có thể dẫn đến tình trạng lắp vào những lỗ trống đó hình ảnh không hợp lý về mặt ngữ nghĩa, mô hình chỉ cố học cách mã hoá và giải mã đầu ra càng giống đầu vào càng tốt nên không quan tâm đến phân bố của các đặc trưng ẩn có hợp lý hay không.

Dựa trên những bất cập tồn tại ở trên, em xin đề xuất hướng phát triển của luận văn là sử dụng mô hình mới dựa trên phương pháp có thể huấn luyện kết hợp ảnh gốc sau khi đã áp dụng các mặt nạ để mạng vừa học được các thông tin trích xuất từ ảnh, vừa học được thông tin các vị trí của các mặt nạ che khuất thông tin trên các hình ảnh giúp việc khôi phục lại ảnh hợp lý hơn. Cụ thể hơn là nghiên cứu các phương pháp dựa trên mạng hai mạng sinh dữ liệu phổ biến hiện nay là GANs và VAEs.

# **TÀI LIỆU THAM KHẢO**

**Tiếng Việt:**

|  |  |
| --- | --- |
| [1] | V. H. Tiệp, "Machine Learning cơ bản," [Online]. Available: https://machinelearningcoban.com/2016/12/26/introduce/. |
| [2] | Guilin Liu and Fitsum A. Reda and Kevin J. Shih and Ting-Chun Wang and Andrew Tao and Bryan Catanzaro, "NVIDIA Irregular Mask Dataset," [Online]. Available: https://nv-adlr.github.io/publication/partialconv-inpainting. |
| [3] | M. Bertalmío, V. Caselles, . S. Masnou and G. Sapiro, Inpainting. |
| [4] | Y. Jiahui, L. Zhe, Y. Jimei, S. Xiaohui, L. Xin and H. Thomas S, "Generative Image Inpainting with Contextual Attention," *arXiv preprint arXiv:1801.07892,* 2018. |
| [5] | O. Elharrouss, N. Almaadeed, S. Al-Maadeed and Y. Akbari, "Image Inpainting: A Review," *Neural Processing Letters,* vol. 51, p. 2007–2028, Dec 2019. |
| [6] | [Online]. Available: https://en.wikipedia.org/wiki/Inpainting#History. |
| [7] | D. J. B. Rojas, B. J. T. Fernandes and S. M. M. Fernandes, "A Review on Image Inpainting Techniques and Datasets," *2020 33rd SIBGRAPI Conference on Graphics,* pp. 240-247, 2020. |
| [8] | E. Demirağ and H. Bengü, "Image Inpainting with Deep Learning," 2021. |
| [9] | H. Li, W. Luo and J. Huang, "Localization of Diffusion-Based Inpainting in Digital Images," *IEEE Transactions on Information Forensics and Security,* vol. 12, pp. 3050-3064, 2017. |
| [10] | X. Zhu, Y. Qian, X. Zhao, B. Sun and Y. Sun, "A deep learning approach to patch-based image inpainting forensics," *Signal Processing: Image Communication,* pp. 90-99, 2018. |
| [11] | Đắm mình vào học sâu, 2020. |
| [12] | H. Nugroho, M. Susanty, A. Irawan, M. Koyimatu and A. Yunita5, "Fully Convolutional Variational Autoencoder For Feature Extraction Of Fire Detection System," *Journal of Computer Science and Information,* pp. 9-15, 2020. |
| [13] | J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu and T. S. Huang, "Free-Form Image Inpainting with Gated Convolution," *arXiv preprint arXiv:1806.03589,* 2018. |
| [14] | [Online]. Available: https://viblo.asia/p/normalization-and-normalization-techniques-in-deep-learning-QpmleJyn5rd. |
| [15] | [Online]. Available: https://viblo.asia/p/cac-chi-so-danh-gia-duoc-su-dung-cho-bai-toan-image-generation-is-fid-psnr-ssim-XL6lA0zDZek. |
| [16] | https://nttuan8.com/bai-5-gan-evaluation/. [Online]. |
| [17] | Z. Liu, P. Luo, X. Wang and X. Tang, "Deep Learning Face Attributes in the Wild," *Proceedings of International Conference on Computer Vision (ICCV),* 2015. |
| [18] | "Pytorch," [Online]. Available: https://pytorch.org/tutorials/beginner/dcgan\_faces\_tutorial.html. |
| [19] | [Online]. Available: https://livebook.manning.com/book/gans-in-action/chapter-4/9. |

# PHỤ LỤC A