**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP HỒ CHÍ MINH KHOA**

**CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN CUỐI KÌ**

**THỊ GIÁC MÁY TÍNH**

*Người thực hiện*: **NGUYỄN QUANG BẢO – MSHV:20083601**

Lớp **: 420301411801**

Khoá  **: 16**

*Người hướng dẫn*: **THẦY LƯU GIANG NAM**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN CUỐI KÌ**

**THỊ GIÁC MÁY TÍNH**

Người thực hiện: **NGUYỄN QUANG BẢO- MSSV: 20083601**

Lớp **: 420301411801**

Khoá  **: 16**

Người hướng dẫn: **THẦY LƯU GIANG NAM**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

LỜI CẢM ƠN

Tôi muốn bày tỏ lòng biết ơn sâu sắc đến người thầy đã dành thời gian và kiến thức để hướng dẫn và hỗ trợ tôi trong quá trình hoàn thành dự án này. Sự chỉ dẫn của người đã giúp tôi vượt qua những thách thức và đạt được kết quả tốt nhất có thể. Xin chân thành cảm ơn!

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP HỒ CHÍ MINH**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi và được sự hướng dẫn của thầy Lưu Giang Nam . Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Công nghiệp TP Hồ Chí Minh không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Nguyễn Quang Bảo*

PHẦN ĐÁNH GIÁ CỦA GIẢNG VIÊN

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Bài toán này tập trung vào việc chuyển đổi hình ảnh tranh vẽ tay thành hình ảnh thực tế, sử dụng mô hình GAN (Generative Adversarial Network). Mục tiêu là tạo ra một mô hình có khả năng sinh ra hình ảnh thực tế từ hình ảnh tranh vẽ tay, sao cho hình ảnh thực tế tạo ra có chất lượng và độ tương đồng cao với hình ảnh tranh vẽ tay. Cách tiếp cận này sử dụng một mạng GAN, trong đó bao gồm một mạng sinh (Generator) và một mạng phân biệt (Discriminator). Mạng sinh nhận vào hình ảnh tranh vẽ tay và cố gắng tạo ra hình ảnh thực tế, trong khi mạng phân biệt cố gắng phân biệt giữa hình ảnh thực tế được tạo ra và hình ảnh thực tế thật.

Quá trình huấn luyện được thực hiện thông qua việc tối ưu hóa hai hàm mất mát: hàm mất mát GAN (đo lường sự giả mạo của hình ảnh thực tế) và hàm mất mát L1 (đo lường sự khác biệt giữa hình ảnh thực tế và hình ảnh tranh vẽ tay). Đồng thời, kiểm soát chất lượng hình ảnh được thực hiện thông qua việc sử dụng chỉ số SSIM (Structural Similarity Index) để đánh giá sự tương đồng giữa hình ảnh thực tế và hình ảnh tranh vẽ tay.

Kết quả đã đạt được sau khi huấn luyện mô hình được đánh giá thông qua việc so sánh hàm mất mát và chỉ số SSIM trên cả tập huấn luyện và tập kiểm tra qua nhiều epochs. Điều này giúp đánh giá hiệu suất của mô hình và đảm bảo rằng nó có khả năng tổng quát hóa tốt đối với dữ liệu mới.

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc166674897)

[PHẦN ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc166674898)

[TÓM TẮT iv](#_Toc166674899)

[MỤC LỤC 1](#_Toc166674900)

[DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT 3](#_Toc166674901)

[DANH MỤC CÁC HÌNH VẼ 4](#_Toc166674902)

[DANH MỤC CÁC BẢNG 5](#_Toc166674903)

[CHƯƠNG 1: GIỚI THIỆU VỀ BÀI TOÁN 6](#_Toc166674904)

[1.1 Giới thiệu về bài toán 6](#_Toc166674905)

[1.2 Ý nghĩa của bài toán 6](#_Toc166674906)

[CHƯƠNG 2: PHÂN TÍCH YÊU CẦU CỦA BÀI TOÁN 7](#_Toc166674907)

[2.1 Yêu cầu của bài toán 7](#_Toc166674908)

[2.2 Các phương pháp giải quyết bài toán 8](#_Toc166674909)

[2.3 Phương pháp đề xuất giải quyết bài toán 9](#_Toc166674910)

[CHƯƠNG 3: PHƯƠNG PHÁP ĐỀ XUẤT 10](#_Toc166674911)

[3.1. Mô hình 10](#_Toc166674912)

[3.1.1 Mô hình Pix2pix 10](#_Toc166674913)

[3.1.2 Mô hình CycleGAN 13](#_Toc166674914)

*[3.2. Phương pháp huấn luyện](#_Toc166674915)* [16](#_Toc166674915)

[3.2.1. Dữ liệu 16](#_Toc166674916)

[3.2.2 Phương pháp huấn luyện 17](#_Toc166674917)

[CHƯƠNG 4: THỰC NGHIỆM 19](#_Toc166674918)

[4.1 Dữ liệu 19](#_Toc166674919)

[4.2 Xử lý dữ liệu 19](#_Toc166674920)

[4.3 Công nghệ sử dụng 20](#_Toc166674921)

[4.4 Cách đánh giá 20](#_Toc166674922)

[4.5 Kết quả đạt được 21](#_Toc166674923)

[CHƯƠNG 5: KẾT LUẬN 24](#_Toc166674924)

[5.1 Kết luận 24](#_Toc166674925)

[TÀI LIỆU THAM KHẢO 25](#_Toc166674926)

[LÀM VIỆC NHÓM](#_Toc166674927) **[.](#_Toc166674927)**

[TỰ ĐÁNH GIÁ](#_Toc166674928) **[.](#_Toc166674928)**

**DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT**

**CÁC CHỮ VIẾT TẮT**

GAN Generative Adversarial Network

Pix2pix Pixels to pixels

CycleGAN Cycle-Consistent Generative Adversarial Network

DANH MỤC CÁC HÌNH VẼ

[Hình 1: Mô hình pix2pix 10](#_Toc166675106)

[Hình 2: Cấu trúc Discriminator của Pix2Pix 11](#_Toc166675107)

[Hình 3: Cấu trúc Generator của Pix2Pix 13](#_Toc166675108)

[Hình 4: Mô hình Discriminator và Generator 14](#_Toc166675109)

[Hình 5: Kết quả predict của mô hình pix2pix 21](#_Toc166675110)

[Hình 6: Biểu đồ độ đo của CycleGAN 22](#_Toc166675111)

[Hình 7: Kết quả predict của CycleGAN 23](#_Toc166675112)

DANH MỤC CÁC BẢNG

[Bảng 1: Kết quả đánh giá độ đo SSIM của 2 mô hình Pix2pix và CycleGAN 23](#_Toc166681934)

CHƯƠNG 1: GIỚI THIỆU VỀ BÀI TOÁN

* 1. Giới thiệu về bài toán

Bài toán chuyển đổi hình ảnh từ hình ảnh vẽ tay (sketch) sang hình ảnh khuôn mặt thực (real face) là một trong những thách thức quan trọng trong lĩnh vực xử lý hình ảnh và trí tuệ nhân tạo. Trong nghiên cứu này, chúng tôi tập trung vào việc áp dụng các phương pháp học sâu để giải quyết bài toán này.Bài toán được đặt ra nhằm tạo ra một mô hình có khả năng chuyển đổi hình ảnh vẽ tay sang hình ảnh khuôn mặt thực một cách tự động và hiệu quả. Mục tiêu là tạo ra các hình ảnh khuôn mặt có chất lượng và độ tương đồng cao với hình ảnh khuôn mặt thực tế tương ứng.Chúng tôi sử dụng mô hình học sâu, cụ thể là mạng GAN (Generative Adversarial Network) và các biến thể của nó, để giải quyết bài toán này. Mô hình được huấn luyện để nhận vào hình ảnh vẽ tay và tạo ra hình ảnh khuôn mặt thực tương ứng. Quá trình huấn luyện này thường bao gồm việc tối ưu hóa các hàm mất mát, bao gồm cả hàm mất mát GAN.Trong quá trình huấn luyện, chúng tôi sử dụng dữ liệu huấn luyện bao gồm 188 cặp hình ảnh vẽ tay và hình ảnh khuôn mặt thực tế. Mô hình được điều chỉnh thông qua việc điều chỉnh các tham số và trọng số của nó để tối ưu hóa chất lượng của hình ảnh kết quả.

1.2 Ý nghĩa của bài toán

Trong lĩnh vực trí tuệ nhân tạo và xử lý hình ảnh, bài toán chuyển đổi hình ảnh từ sketch (bản vẽ tay) sang real face (khuôn mặt thực) đang thu hút sự quan tâm của nhiều nhà nghiên cứu. Vấn đề này không chỉ mang lại những triển vọng trong lĩnh vực công nghệ, mà còn có ảnh hưởng sâu rộng đến nhiều lĩnh vực khác nhau như nghệ thuật, giáo dục và phát triển nhân lực.

Ở mức độ kỹ thuật, việc giải quyết bài toán này đòi hỏi sự phát triển và ứng dụng các phương pháp học sâu và xử lý hình ảnh. Mục tiêu của bài toán là tạo ra một mô hình có khả năng tự động chuyển đổi các bức vẽ tay thành hình ảnh khuôn mặt thực với chất lượng và độ tương đồng cao.

CHƯƠNG 2: PHÂN TÍCH YÊU CẦU CỦA BÀI TOÁN

2.1 Yêu cầu của bài toán

Bài toán chuyển đổi hình ảnh từ sketch thành real face đặt ra một số yêu cầu cụ thể như sau:

- Chất lượng Hình ảnh: Một trong những yêu cầu chính là đảm bảo chất lượng cao của hình ảnh kết quả sau khi chuyển đổi. Hình ảnh khuôn mặt thực phải được tái tạo một cách chân thực và rõ ràng từ các bức vẽ tay, đảm bảo độ phân giải và chi tiết.

- Độ Tương đồng: Một yêu cầu quan trọng khác là độ tương đồng giữa hình ảnh kết quả và hình ảnh khuôn mặt thực tế. Điều này đảm bảo rằng hình ảnh sau khi chuyển đổi giữ lại được các đặc điểm và đặc tính quan trọng của khuôn mặt người.

- Tính Tổng quát hóa: Mô hình phải có khả năng tổng quát hóa để có thể chuyển đổi các loại sketch khác nhau thành real face một cách hiệu quả. Điều này đòi hỏi mô hình có khả năng nhận biết và tái tạo các đặc điểm chung của khuôn mặt người mà không phụ thuộc vào các điều kiện cụ thể của hình ảnh.

- Tính Tự động và Hiệu quả: Một yêu cầu khác là mô hình phải hoạt động tự động và hiệu quả, tức là có khả năng chuyển đổi các bức vẽ tay thành hình ảnh khuôn mặt thực một cách nhanh chóng và mạnh mẽ mà không cần sự can thiệp nhiều từ con người.

- Tính Tổng quát hóa và Tính Linh hoạt: Mô hình phải có khả năng hoạt động với nhiều loại sketch và real face khác nhau, từ những bức vẽ đơn giản đến những hình ảnh khuôn mặt phức tạp, đồng thời có khả năng hoạt động trên nhiều nền tảng và môi trường khác nhau.

2.2 Các phương pháp giải quyết bài toán

Có một số phương pháp đã được đề xuất để giải quyết bài toán chuyển đổi hình ảnh từ sketch thành real face, trong đó mỗi phương pháp mang lại các ưu điểm và hạn chế riêng. Dưới đây là một số phương pháp tiêu biểu đã được thảo luận trong các nghiên cứu trước đó: Phillip Isola và đồng nghiệp đã giới thiệu phương pháp Pix2Pix trong bài báo "Image-to-Image Translation with Conditional Adversarial Networks" (2017)[1]. Pix2Pix sử dụng một mạng GAN có điều kiện để học ánh xạ từ một loại hình ảnh (ví dụ: sketch) sang một loại hình ảnh khác (ví dụ: real face). Mô hình được huấn luyện trên các cặp dữ liệu hình ảnh sketch và real face, và tạo ra những kết quả ấn tượng với chất lượng hình ảnh cao. Tuy nhiên, mô hình có thể gặp khó khăn khi xử lý các trường hợp phức tạp hoặc hình ảnh chất lượng thấp.Jun-Yan Zhu và đồng nghiệp đã giới thiệu phương pháp CycleGAN trong bài báo "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks" (2017)[2]. CycleGAN không yêu cầu các cặp dữ liệu huấn luyện ghép nối, mà tập trung vào việc học một ánh xạ từ miền nguồn sang miền đích và ngược lại. Mô hình này cho phép chuyển đổi hình ảnh giữa các miền một cách hiệu quả mà không cần dữ liệu huấn luyện ghép nối. Tuy nhiên, việc không có sự ghép nối có thể dẫn đến việc mô hình không hiệu quả trong một số trường hợp cụ thể.

Rui Chen và đồng nghiệp đã giới thiệu phương pháp FaceShop trong bài báo "FaceShop: A Deep Sketch-to-Photo Image Synthesis System" (2018)[3]. FaceShop sử dụng một mạng GAN có điều kiện để thực hiện chuyển đổi từ sketch sang hình ảnh khuôn mặt thực. Mô hình này được thiết kế đặc biệt để tạo ra hình ảnh khuôn mặt chất lượng cao từ các bức vẽ tay. FaceShop đạt được kết quả ấn tượng với chất lượng hình ảnh cao, mặc dù mô hình có thể gặp khó khăn khi xử lý các trường hợp phức tạp hoặc các sketch không chính xác.

2.3 Phương pháp đề xuất giải quyết bài toán

Hướng giải quyết bài toán là sử dụng mạng Generative Adversarial Network (GAN)[4] và các biến thể cho việc chuyển đổi ảnh (image-to-image translation).

Đề xuất phương pháp giải quyết bài toán là sử dụng hai kiến trúc mạng học sâu, đó là pix2pix[5] và CycleGAN[6], để thực hiện việc chuyển đổi hình ảnh từ sketch sang real face.

Pix2pix là một mô hình mạng học sâu được thiết kế cho các bài toán chuyển đổi ảnh từ một miền ảnh này sang miền ảnh khác, thông qua việc học ánh xạ từ đầu vào (sketch) sang đầu ra (real face). Mạng này bao gồm hai thành phần chính:

* Generator: Nhận vào hình ảnh sketch và tạo ra hình ảnh khuôn mặt thực.
* Discriminator: Phân biệt giữa hình ảnh khuôn mặt thực và hình ảnh được tạo ra từ Generator.

CycleGAN là một mô hình mạng học sâu được sử dụng cho việc chuyển đổi hình ảnh giữa hai miền ảnh mà không cần cặp dữ liệu đào tạo điều chỉnh. Với bài toán của chúng tôi, miền ảnh đầu vào là sketch và miền ảnh đầu ra là real face. CycleGAN bao gồm hai bộ Generator[7] và hai bộ Discriminator[8]:

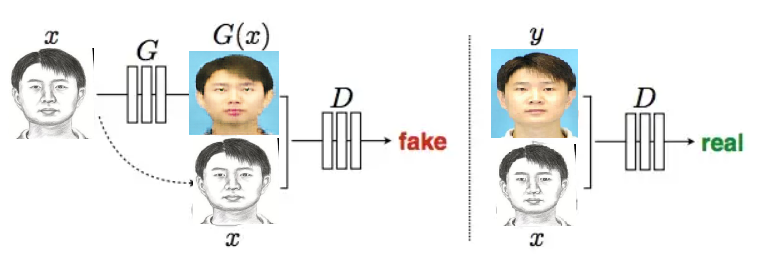
* Hai Generator: Một để chuyển đổi từ sketch sang real face và một để chuyển đổi ngược lại.
* Hai Discriminator: Mỗi bộ để phân biệt giữa ảnh thực và ảnh được tạo ra từ Generator tương ứng.

Cả hai phương pháp đều có khả năng học các ánh xạ phức tạp từ sketch sang real face và ngược lại, đồng thời đảm bảo chất lượng và tính thực tế của hình ảnh kết quả. Điều này làm cho pix2pix và CycleGAN là lựa chọn phù hợp cho bài toán chuyển đổi hình ảnh trong nghiên cứu của chúng tôi.

CHƯƠNG 3: PHƯƠNG PHÁP ĐỀ XUẤT

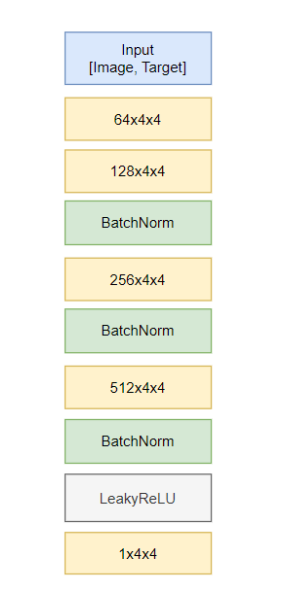
3.1. Mô hình

3.1.1 Mô hình Pix2pix



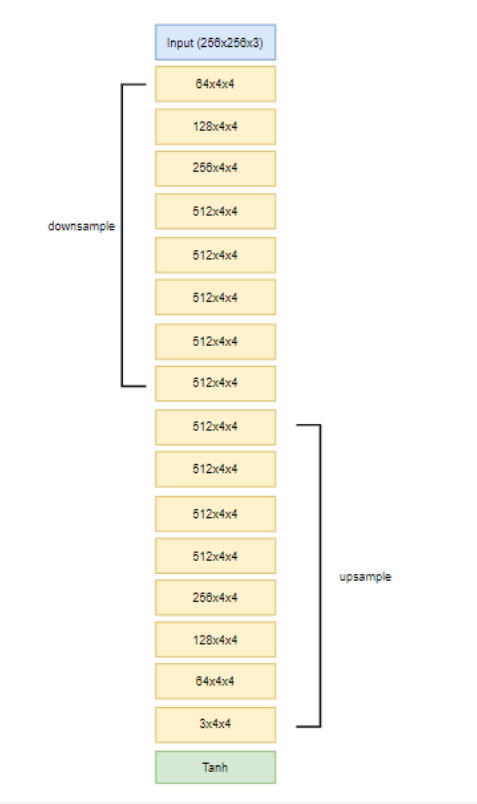
Hình 1: Mô hình pix2pix

Pix2pix là một mạng GAN nên cũng có 2 phần Generator (G)để sinh ảnh fake và Discriminator(D) để phân biệt ảnh thật vàảnh fake. Tuy nhiên khác với GAN bình thường khi input của Generator là nhiễu, thì trong pix2pix của generator là ảnh. Và output cũng là ảnh. Input của disciminator là ảnh x (input của gennerator) và G(x) (output của generator). Hai ảnh này cũng kích thước được xếp lên nhau rồi cho vào discriminator. Discriminator học bằng cách phân biệt x và G(x) là ảnh fake, x và y là ảnh thật.



Hình 2: Cấu trúc Discriminator của Pix2Pix

Ngược lại generator sẽ học bằng cách cho x và G(x) là ảnh thật.



Hình 3: Cấu trúc Generator của Pix2Pix

Generator input là ảnh và output cũng là ảnh nên kiến trúc cũng gần giống mạng U-net, dạng encoder-decoder. Kiến trúc generator chúng tôi xây dựng gồm 8 lớp Conv2d với hàm kích hoạt LeakyReLU[9], 8 lớp Conv2dTranspose[10] với hàm kích hoạt ReLU. Từ layer thứ 2 trở đi chúng tôi xây dựng kết hợp với BatchNorm chuẩn hóa dữ liệu. Generator loss sử dụng binary cross entropy kết hợp với L1(MAE)[11] bằng tham số lambda.

Công thức Binary cross entropy:

Trong đó:

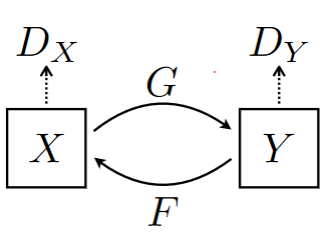
• λ: là hệ số, thường là 100

Mô hình này sử dụng SSIM làm thông số đánh giá.

3.1.2 Mô hình CycleGAN

CycleGAN được thiết kế dựa trên Generative Adversarial Network (GAN). CycleGAN là một mở rộng của kiến trúc GAN cổ điển bao gồm 2 Generator và 2 Discriminator. Generator đầu tiên gọi là G, nhận đầu vào là ảnh từ domain X (ảnh đầu vào) và convert nó sang domain Y (ảnh mục tiêu. Generator còn lại gọi là Y, có nhiệm vụ convert ảnh từ domain Y sang X. Mỗi mạng Generator có 1 Discriminator tương ứng với nó:

* DY : phân biệt ảnh lấy từ domain Y và ảnh được dịch qua G(x).
* DX: phân biệt ảnh lấy từ domain X và ảnh được dịch qua F(y).



Hình 4: Mô hình Discriminator và Generator

1. Mạng CycleGAN sử dụng để giải quyết các bài toán unsupervised uni-model image to image translation[12], để giải quyết bài toán này, nhóm chúng tôi đã xây dựng mô hình với kiến trúc CycleGAN như sau:

a. Bộ tạo (Generator):

1. Phần mã hóa (Encoder):

Mỗi bộ tạo bắt đầu với một chuỗi các lớp tích chập để mã hóa đặc trưng từ ảnh đầu vào. Lớp tích chập đầu tiên (self.conv1) có kernel size là 4 và stride là 2, được áp dụng trực tiếp lên ảnh đầu vào để tạo ra các đặc trưng sơ bộ. Lớp tích chập thứ hai (self.conv2) có kernel size là 4 và stride là 2, để tiếp tục trích xuất các đặc trưng với độ phức tạp cao hơn. Lớp tích chập thứ ba (self.conv3) có kernel size là 4 và stride là 2, để tiếp tục giảm kích thước của đặc trưng và tăng cường tính hiệu quả của mã hóa.

2. Phần mạng nơ-ron chồng chất (ResNet):

Sau khi mã hóa đặc trưng từ ảnh đầu vào, mỗi bộ tạo sử dụng một số khối Residual Block để học ánh xạ từ miền dữ liệu nguồn sang miền dữ liệu đích. Mỗi khối Residual Block bao gồm hai lớp tích chập với kernel size là 3 và padding là 1, cùng với các lớp batch normalization để tăng cường hiệu suất học của mô hình.

3. Phần giải mã (Decoder):

Sau khi đi qua các khối Residual Block[13], đặc trưng đã được mã hóa được chuyển đến phần giải mã để tạo ra ảnh đích.

Phần giải mã bao gồm một chuỗi các lớp tích chập chuyển ngược (deconvolutional layers) để tăng kích thước của đặc trưng trở lại gần với kích thước ban đầu của ảnh.

Lớp tích chập chuyển ngược cuối cùng không có batch normalization[14] để đảm bảo rằng đầu ra cuối cùng của bộ tạo nằm trong phạm vi giá trị mong muốn (ví dụ: -1 đến 1 cho ảnh RGB).

b. Bộ phân biệt (Discriminator):

1. Phần mã hóa đặc trưng (Feature Encoding):

Mỗi bộ phân biệt bắt đầu bằng một chuỗi các lớp tích chập để học các đặc trưng của ảnh.

Lớp tích chập đầu tiên (self.conv1) có kernel size là 4 và stride là 2, được sử dụng để trích xuất các đặc trưng cơ bản từ ảnh đầu vào.

Tiếp theo, mỗi bộ phân biệt có thể có một hoặc nhiều lớp tích chập khác nhau để tiếp tục tăng cường việc học các đặc trưng với độ phức tạp cao hơn.

2. Lớp phân loại (Classification Layer):

Cuối cùng, mỗi bộ phân biệt kết thúc bằng một lớp phân loại để đưa ra dự đoán về độ thật của ảnh.

Lớp phân loại này thường là một lớp tích chập với kernel size là 4 và stride là 1, được áp dụng lên các đặc trưng đã được mã hóa để đưa ra dự đoán cuối cùng.

Các độ đo được tính trong mô hình này là :

1. MSE Loss (Mean Squared Error Loss):

Được sử dụng để đo lỗi giữa đầu ra của bộ phân biệt và các nhãn mong muốn (1 cho ảnh thật, 0 cho ảnh giả).

Trong đó là đầu ra của bộ phân biệt và là nhãn mong muốn và n là số lượng mẫu

Mô hình này cũng sử dụng SSIM[15] làm thông số đánh giá.

*3.2. Phương pháp huấn luyện*

3.2.1. Dữ liệu

Bộ dữ liệu CUHK (Chinese University of Hong Kong) là một tập dữ liệu phổ biến được sử dụng trong nhiều nghiên cứu về nhận dạng khuôn mặt và các ứng dụng liên quan đến xử lý ảnh. Bộ dữ liệu bao gồm cả hình ảnh khuôn mặt thật và các bản phác thảo (sketch) tương ứng với các khuôn mặt đó.Bộ dữ liệu CUHK có thể được sử dụng để huấn luyện và đánh giá hiệu suất của các mô hình trong việc thực hiện chuyển đổi từ sketch sang hình ảnh khuôn mặt.

Để tiền xử lý dữ liệu cho mô hình Đầu tiên, chúng tôi sử dụng một hàm tùy chỉnh để sắp xếp các tệp hình ảnh trong thứ tự số tăng dần, đảm bảo rằng các hình ảnh sẽ được tải theo thứ tự đúng đắn từ thư mục.

Tiếp theo, chúng tôi định nghĩa kích thước mong muốn cho hình ảnh, sau đó tải các hình ảnh từ hai thư mục khác nhau (photos và sketches). Mỗi hình ảnh sau khi được tải sẽ được chuyển đổi sang không gian màu RGB và điều chỉnh kích thước của nó thành kích thước mong muốn. Cuối cùng, các giá trị pixel của hình ảnh được chuẩn hóa về phạm vi [0, 1] và chúng được chuyển đổi thành mảng numpy. Các hình ảnh được thêm vào các danh sách tương ứng để sử dụng trong quá trình huấn luyện mô hình. Đoạn mã này giúp chuẩn bị dữ liệu từ hai tập dữ liệu khác nhau để sử dụng trong quá trình huấn luyện mô hình.

3.2.2 Phương pháp huấn luyện

Mô hình pix2pix

Để huấn luyện mô hình Pix2Pix, chúng tôi sử dụng một quá trình lặp lại qua các epochs để cải thiện chất lượng của mô hình. Mỗi epochs bao gồm hai giai đoạn chính: giai đoạn huấn luyện và giai đoạn kiểm tra.

Trong giai đoạn huấn luyện, chúng tôi sử dụng hai bộ dữ liệu: bộ dữ liệu đầu vào (sketch\_dataset) và bộ dữ liệu mục tiêu (photo\_dataset). Chúng tôi lặp lại qua từng cặp hình ảnh trong hai bộ dữ liệu này và áp dụng hàm train\_step để cập nhật trọng số của mô hình dựa trên gradient của hai hàm mất mát: hàm mất mát của bộ sinh (generator loss) và hàm mất mát của bộ phân biệt (discriminator loss).

Tương tự, trong giai đoạn kiểm tra, chúng tôi sử dụng hai bộ dữ liệu kiểm tra (sketch\_dataset\_t và photo\_dataset\_t) để đánh giá hiệu suất của mô hình. Chúng tôi lặp lại qua từng cặp hình ảnh trong hai bộ dữ liệu kiểm tra này và áp dụng hàm test\_step để tính toán hàm mất mát và độ đo SSIM (Structural Similarity Index) của mô hình trên dữ liệu kiểm tra.

Quá trình huấn luyện diễn ra qua một số epochs được xác định trước và trong mỗi epochs, chúng tôi theo dõi sự thay đổi của các hàm mất mát và độ đo SSIM. Kết quả của quá trình huấn luyện, bao gồm các giá trị hàm mất mát của bộ sinh và bộ phân biệt, cũng như độ đo SSIM, được ghi lại và hiển thị sau mỗi epochs.

Cuối cùng, sau khi quá trình huấn luyện hoàn thành, chúng tôi sử dụng mô hình được huấn luyện để tạo ra các hình ảnh dự đoán từ các hình ảnh đầu vào trong tập kiểm tra. Các hình ảnh đầu vào, hình ảnh mục tiêu và hình ảnh dự đoán được hiển thị để đánh giá chất lượng của mô hình được huấn luyện.

Mô hình CycleGAN

Để huấn luyện mô hình CycleGAN, chúng tôi thực hiện một vòng lặp huấn luyện qua các epochs. Mỗi epochs bao gồm ba giai đoạn chính: huấn luyện các bộ phân biệt (Discriminators), huấn luyện các bộ sinh (Generators) và ghi lại các thống kê.

Trong giai đoạn huấn luyện các bộ phân biệt, chúng tôi lặp lại qua từng batch của dữ liệu từ cả hai miền (Domain X và Domain Y). Đối với mỗi miền, chúng tôi tính toán hai thành phần của hàm mất mát của bộ phân biệt: mất mát cho các ảnh thật (real) và mất mát cho các ảnh giả (fake) được tạo ra bởi bộ sinh. Sau đó, chúng tôi tính toán tổng hợp hàm mất mát và cập nhật trọng số của các bộ phân biệt thông qua các bước lan truyền ngược (backpropagation).

Trong giai đoạn huấn luyện các bộ sinh, chúng tôi cũng lặp lại qua từng batch của dữ liệu từ cả hai miền. Đối với mỗi miền, chúng tôi tính toán hai thành phần của hàm mất mát của các bộ sinh: mất mát "real" của bộ phân biệt và mất mát do tính nhất quán của các ảnh đã tạo ra. Sau đó, chúng tôi tổng hợp các hàm mất mát này và cập nhật trọng số của các bộ sinh thông qua các bước lan truyền ngược.

Các thông tin chi tiết và đánh giá về hiệu suất của mô hình, bao gồm các giá trị hàm mất mát và các thước đo như SSIM (Structural Similarity Index), được ghi lại và hiển thị sau mỗi epochs để đánh giá và theo dõi quá trình huấn luyện.

Đồng thời, chúng tôi thực hiện lưu trữ các mẫu hình ảnh được tạo ra sau mỗi số epochs nhất định để kiểm tra chất lượng và sự học tập của mô hình.

CHƯƠNG 4: THỰC NGHIỆM

4.1 Dữ liệu

Dữ liệu của Bộ dữ liệu CUHK (Chinese University of Hong Kong) được thu thập từ các nguồn khác nhau như cơ sở dữ liệu hình ảnh công cộng trực tuyến, các tài liệu nghiên cứu và các bộ dữ liệu khác có sẵn. Đây là một bộ dữ liệu công cộng, nơi mà các nhà nghiên cứu và các nhà phát triển có thể truy cập và sử dụng miễn phí cho mục đích nghiên cứu và phát triển.

Bộ dữ liệu CUHK bao gồm các hình ảnh khuôn mặt thật và các bản phác thảo tương ứng với các khuôn mặt đó. Mỗi bản ghi trong bộ dữ liệu có thể chứa một cặp hình ảnh, bao gồm một hình ảnh khuôn mặt thật và một bản phác thảo tương ứng. Các hình ảnh trong bộ dữ liệu có thể được thu thập từ nhiều nguồn khác nhau và có thể bao gồm đa dạng về độ phân giải, chiều sâu màu sắc và điều kiện ánh sáng.

4.2 Xử lý dữ liệu

Tiền xử lý dữ liệu là bước quan trọng trong quá trình huấn luyện mô hình. Trước khi đưa dữ liệu vào mô hình, chúng ta cần đảm bảo rằng dữ liệu được chuẩn bị và biến đổi một cách chính xác để phù hợp với yêu cầu của mô hình.

Quá trình tiền xử lý dữ liệu bao gồm các bước sau:

Sắp xếp các tệp hình ảnh trong thứ tự số tăng dần: Điều này đảm bảo rằng các hình ảnh sẽ được tải theo thứ tự đúng đắn từ thư mục, giúp quản lý và xử lý dữ liệu một cách hiệu quả hơn.

Chuyển đổi kích thước và chuẩn hóa hình ảnh: Mỗi hình ảnh sau khi được tải được chuyển đổi sang không gian màu RGB và điều chỉnh kích thước của nó thành kích thước mong muốn. Sau đó, các giá trị pixel của hình ảnh được chuẩn hóa về phạm vi [0, 1] để tối ưu hóa hiệu suất của mô hình trong quá trình huấn luyện.

Lưu trữ dữ liệu: Các hình ảnh được chuyển đổi thành mảng numpy và được thêm vào các danh sách tương ứng để sử dụng trong quá trình huấn luyện mô hình.

Quá trình tiền xử lý này giúp chuẩn bị dữ liệu từ hai tập dữ liệu khác nhau (hình ảnh thật và bản phác thảo) để sử dụng trong quá trình huấn luyện mô hình, đồng thời đảm bảo rằng dữ liệu đầu vào được chuẩn hóa và phù hợp với yêu cầu của mô hình.

4.3 Công nghệ sử dụng

Ngôn ngữ lập trình: Python.

Thư viện sử dụng: TensorFlow, Keras, PyTorch.

Công cụ giải quyết bài toán: Mô hình mạng nơ-ron sâu (deep neural network), các thuật toán tối ưu hóa như Adam optimizer.

4.4 Cách đánh giá

Đánh giá được thực hiện bằng độ đo SSIM (Structural Similarity Index), được tính bằng cách so sánh cấu trúc và nội dung giữa hai hình ảnh. Công thức tính SSIM bao gồm sự tương đồng về độ sáng, độ tương phản và cấu trúc của hình ảnh.

Trong đó:

và là giá trị trung bình của x và y tương ứng.

và là phương sai của x và y tương ứng.

là phương sai tương ứng của x và y.

và là hằng số để đảm bảo tính ổn định và tránh việc chia cho 0.

Kích thước cửa sổ N×N được sử dụng để tính toán các giá trị trên toàn bộ hình ảnh.

## 4.5 Kết quả đạt được

Mô hìnhPix2pix chúng tôi huấn luyện với 20 epoch và huấn luyện trên từng batch và thu được kết quả như sau:

Kết quả predict của mô hình pix2pix:



Hình 5: Kết quả predict của mô hình pix2pix

Đối với mô hình này chúng tôi thực hiện huấn luyện với 50 epoch và batch size bằng 1, learning rate của G và D đều bằng 0.0002, kết quả thu được được trình bày như sau:



Hình 6: Biểu đồ độ đo của CycleGAN

Kết quả predict của mô hình CycleGAN:



Hình 7: Kết quả predict của CycleGAN

Kết quả đánh giá độ đo SSIM của hai mô hình:

|  |  |  |
| --- | --- | --- |
| Thông số | Pix2pix | CycleGAN |
| SSIM | 0.590926965 | 0.384734 |

Bảng 1: Kết quả đánh giá độ đo SSIM của 2 mô hình Pix2pix và CycleGAN

Qua kết quả so sánh phía trên ta thấy được mô hình pix2pix cho ra kết quả tốt hơn CycleGAN.

CHƯƠNG 5: KẾT LUẬN

5.1 Kết luận

Việc chuyển đổi phác họa - ảnh khuôn mặt giờ đây đã là một vấn đề giúp ích trong việc nhận dạng các đối tượng tình nghi, việc chuyển đổi từ ảnh qua phác họa cũng góp phần vào việc giải trí khi nhiều người muốn biết khuôn mặt mình khi chuyển qua ảnh phác họa sẽ như thế nào. Trong đề tài này, chúng tôi sử dụng các mô hình như Pix2Pix, CycleGAN và cho kết quả khả quan. Bằng việc đánh giá và so sánh giữa các mô hình, chúng tôi nhận thấy mô hình Pix2pix cho kết quả tạo ảnh tốt nhất.

Hướng phát triển tương lai:

Để nâng cao hiệu suất của mô hình chuyển đổi phác họa - ảnh khuôn mặt, các hướng phát triển tiếp theo bao gồm nghiên cứu về kiến trúc mạng sâu hơn, áp dụng các kỹ thuật mới nhất trong học sâu, tối ưu hóa hàm mất mát, tăng cường dữ liệu và đa dạng hóa, cũng như đánh giá và so sánh kỹ lưỡng giữa các mô hình. Bằng cách thực hiện các nghiên cứu này, chúng ta có thể đạt được một mô hình chất lượng cao hơn, có khả năng tổng quát hóa tốt và mang lại kết quả tạo ảnh chính xác và sinh động hơn.

**TÀI LIỆU THAM KHẢO**

1. Phillip Isola và đồng nghiệp đã giới thiệu phương pháp Pix2Pix trong bài báo *"Image-to-Image Translation with Conditional Adversarial Networks"* (2017).
2. Jun-Yan Zhu và đồng nghiệp đã giới thiệu phương pháp CycleGAN trong bài báo *"Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks"* (2017).
3. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
4. G. W. Taylor, I. Spiro, C. Bregler, and R. Fergus. Learning invariance through imitation. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pages 2729–2736. IEEE, 2011
5. G. Wang, D. Hoiem, and D. Forsyth. Learning image similarity from flickr groups using stochastic intersection kernel machines. In Computer Vision, 2009 IEEE 12th International Conference on, pages 428–435 IEEE, 2009.
6. J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1386–1393, 2014.
7. Q. Liu, X. Tang, H. Jin, H. Lu, and S. Ma. A nonlinear approach for face sketch synthesis and recognition. In CVPR 2005, volume 1, pages 1005–1010. IEEE, 2005.
8. X. Gao, J. Zhong, J. Li, and C. Tian. Face sketch synthesis algorithm based on e-hmm and selective ensemble. IEEE CSVT, 18(4):487–496, 2008.
9. B. Xiao, X. Gao, D. Tao, and X. Li. A new approach for face recognition by sketches in photos. Signal Processing, 89(8):1576–1588, 2009.
10. X. Wang and X. Tang. Face photo-sketch synthesis and recognition. TPAMI, 31(11):1955–1967, 2009.
11. H. Zhou, Z. Kuang, and K.-Y. K. Wong. Markov weight fields for face sketch synthesis. In CVPR, pages 1091–1097. IEEE, 2012.
12. C. Peng, X. Gao, N. Wang, D. Tao, X.Li, and J. Li.Multiple representations-based face sketch-photo synthesis. IEEE NNLS, 27(11):2201–2215, 2016.
13. N. Wang, X. Gao, L. Sun, and J. Li. Bayesian face sketch synthesis. IEEE TIP, 26(3):1264–1274, 2017.
14. F. Gao, S. Shi, J. Yu, and Q. Huang. Composition-aided sketch-realistic portrait generation. arXiv:1712.00899, 2017.
15. C. Chen, X. Tax, and K. Wong. Face sketch synthesis with style transfer using pyramid column feature. In IEEE WACV, 2018.