

Chuyển Đổi Ảnh-Phác Thảo Khuôn Mặt Sử Dụng GAN

Nguyễn Thị Thanh Hòa* and Ngô Quốc Hoàng* and Quách Trọng Nghĩa*

*Industrial University of Ho Chi Minh City

Computer Vision Course
Ngày 20 tháng 12 năm 2022



Nội dung

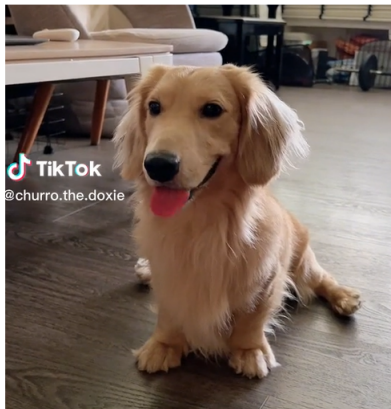
- 1 Giới thiệu
- 2 Dữ liệu
- 3 Autoencoder
- 4 Pix2Pix
- 5 CycleGAN
- 6 So sánh kết quả
- 7 Hướng nghiên cứu tương lai

Nội dung

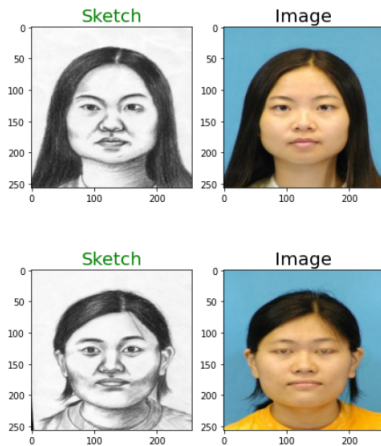
- 1 Giới thiệu
- 2 Dữ liệu
- 3 Autoencoder
- 4 Pix2Pix
- 5 CycleGAN
- 6 So sánh kết quả
- 7 Hướng nghiên cứu tương lai

Giới thiệu

Hiện nay có nhiều công cụ AI chuyển đổi hình ảnh sang hình ảnh. Ta có thể kể đến một ví dụ như AI Manga trên mạng xã hội tiktok:



Hình: Ví dụ

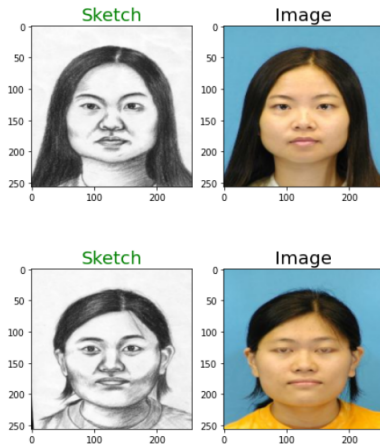


Hình: Dữ liệu

Nội dung

- 1 Giới thiệu
- 2 Dữ liệu**
- 3 Autoencoder
- 4 Pix2Pix
- 5 CycleGAN
- 6 So sánh kết quả
- 7 Hướng nghiên cứu tương lai

Dữ liệu



Hình: Dữ liệu

Xử lý dữ liệu

Mỗi hình ảnh sẽ được resize về kích thước $256 \times 256 \times 3$, sau đó tiến hành lật, xoay góc 90 độ và tạo thành 8 ảnh.

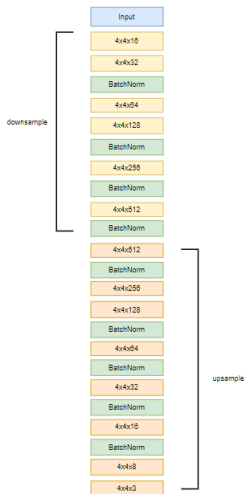
Với bộ data gồm 188 cặp ảnh, sau xử lý được 1504 cặp ảnh.

Tập dữ liệu chia thành train test với tỉ lệ 8:2

Nội dung

- 1 Giới thiệu
- 2 Dữ liệu
- 3 Autoencoder**
- 4 Pix2Pix
- 5 CycleGAN
- 6 So sánh kết quả
- 7 Hướng nghiên cứu tương lai

Mô hình Autoencoder



Hình: Kiến trúc mô hình Autoencoder, 6 lớp Conv2d, 8 lớp Conv2dTranspose

Đánh giá

Sử dụng độ lỗi MAE, độ tương đồng SSIM:

Công thức MAE:

$$MAE = \frac{abs(\hat{y} - y)}{n} \quad (1)$$

SSIM (Structural Similarity Index Measure) được sử dụng để đo độ giống nhau giữa hai ảnh dựa trên độ chói (μ), độ tương phản (c) và kết cấu (s). Công thức:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1) + (\sigma_x^2 + \sigma_y^2 + c_2)} \quad (2)$$

Trong đó:

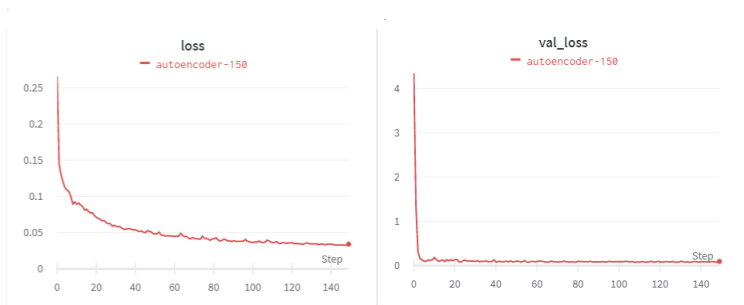
- $\mu_x = \frac{1}{N} \sum_{i=1}^N x_i$: Trung bình
- $\sigma_x = (\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x))^2$: Phương sai
- $c_1 = (k_1 L)^2, c_2 = (k_2 L)^2$
- $k_1 = 0.01, k_2 = 0.03$
- L : Phạm vi dao động của giá trị pixel

Huấn luyện với epoch = 150, learningrate = 0.001

Kết quả mô hình Autoencoder

Loss: 0.033

Val Loss: 0.091

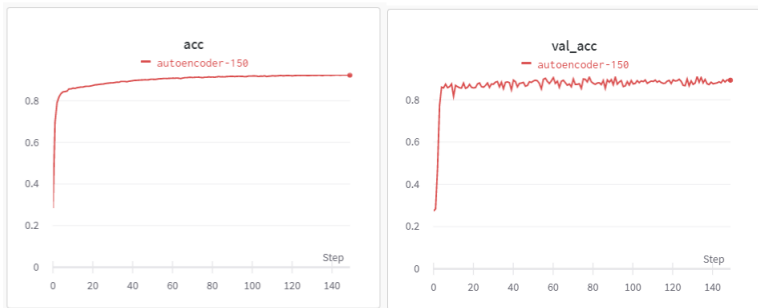


Hình: Loss Autoencoder

Kết quả mô hình Autoencoder

Acc : 0.92

Val acc: 0.89

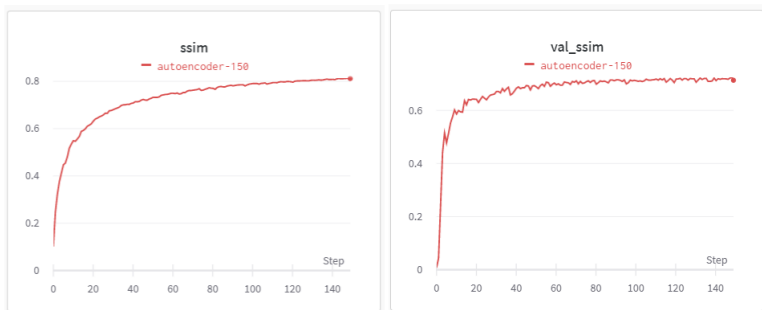


Hình: Accuracy Autoencoder

Kết quả mô hình Autoencoder

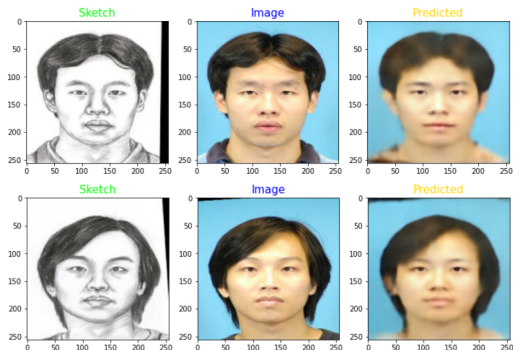
SSIM: 0.81

Val SSIM: 0.71



Hình: SSIM Autoencoder

Kết quả mô hình Autoencoder

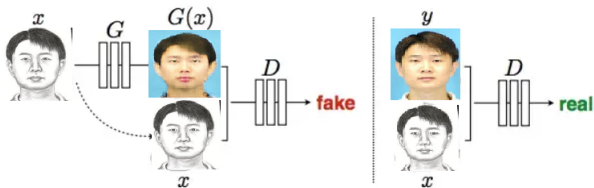


Hình: Kết quả của mô hình autoencoder

Nội dung

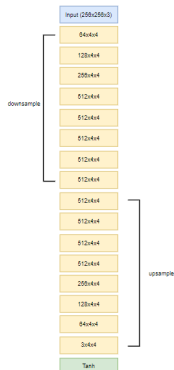
- 1 Giới thiệu
- 2 Dữ liệu
- 3 Autoencoder
- 4 Pix2Pix**
- 5 CycleGAN
- 6 So sánh kết quả
- 7 Hướng nghiên cứu tương lai

Mô hình Pix2Pix



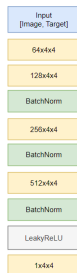
Hình: Pix2Pix

Mô hình Pix2Pix



Hình: Cấu trúc Generator của Pix2Pix, 8 lớp Conv2d, 8 lớp Conv2dTranspose

Mô hình Pix2Pix



Hình: Cấu trúc Discriminator của Pix2Pix, 5 lớp Conv2D

Đánh giá

Generator loss sử dụng binary cross entropy kết hợp với L1(MAE) bằng tham số lambda.

Công thức Binary cross entropy:

$$Loss_{BCE} = -\frac{1}{m} \sum_{i=1}^m y_i \cdot \log \hat{y}_i + (1 - y_i) \cdot \log(1 - \hat{y}_i) \quad (3)$$

$$Loss_G = Loss_{BCE} + \lambda * MAE \quad (4)$$

Trong đó:

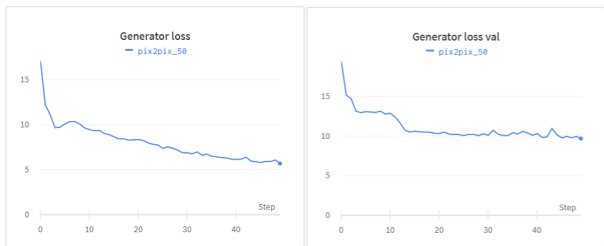
- λ : là hệ số, ở đây là 100

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

Kết quả

Loss_G : 5.6

Val Loss_G : 9.68

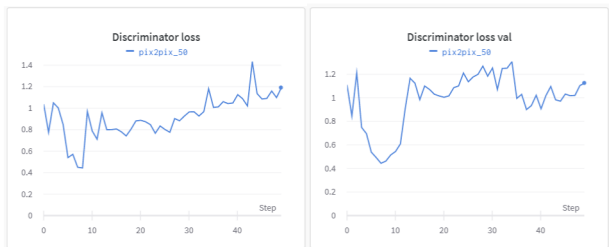


Hình: Generator Loss của Pix2Pix

Kết quả

$\text{Loss}_D : 1.19$

Val $\text{Loss}_D : 1.012$

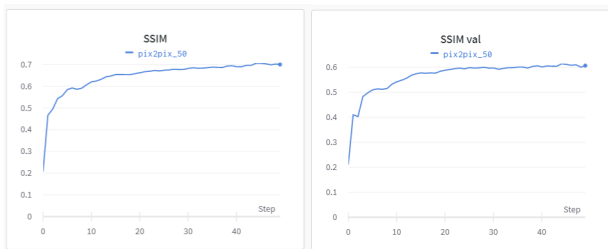


Hình: Discriminator Loss của Pix2Pix

Kết quả

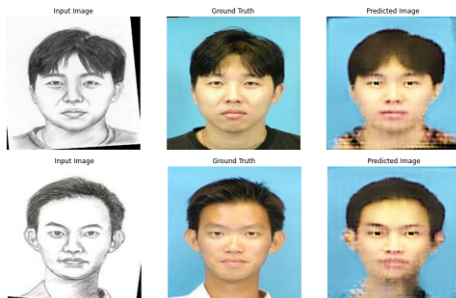
SSIM: 0.7

Val SSIM: 0.6



Hình: SSIM của Pix2Pix

Kết quả



Hình: Kết quả predict Pix2Pix

Nội dung

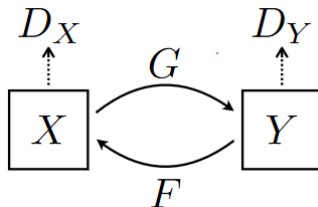
- 1 Giới thiệu
- 2 Dữ liệu
- 3 Autoencoder
- 4 Pix2Pix
- 5 CycleGAN**
- 6 So sánh kết quả
- 7 Hướng nghiên cứu tương lai

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 F , có nhiệm vụ convert ảnh từ domain Y sang X . Mỗi mạng Generator có 1 Discriminator tương ứng với nó:

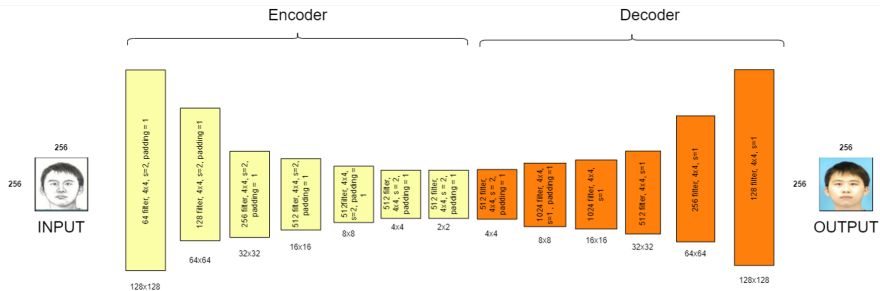
D_Y : phân biệt ảnh lấy từ domain Y và ảnh được dịch qua $G(x)$.

D_X : phân biệt ảnh lấy từ domain X và ảnh được dịch qua $F(y)$.



Hình: CycleGAN

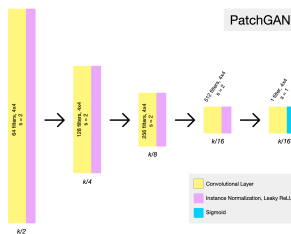
Mô hình CycleGAN



Hình: Kiến trúc của generator CycleGAN

Mô hình CycleGAN

PatchGAN có sự khác biệt với CNN, khi ta đưa một ảnh vào CNN, chúng sẽ trả về xác suất của toàn bộ kích thước hình ảnh đầu vào thành một vector đầu ra duy nhất. Còn với PatchGAN, nó cũng trả về xác suất là thật hay giả, nhưng không phải của toàn bộ kích thước ảnh mà là xác suất trên từng vùng ảnh, do đó đầu ra sẽ là một ma trận $N \times N$.



Hình: Kiến trúc Discriminator của CycleGAN

Hàm mất mát của CycleGAN

a. Adversarial loss

$$L_{adv}(G, D_Y, X, Y) = \frac{1}{n}[\log D_Y(y)] + \frac{1}{n}[\log(1 - D_Y(G(x)))] \quad (5)$$

Trong quá trình huấn luyện, generator G cố gắng tối thiểu hóa hàm adversarial loss bằng cách translate ra ảnh $G(x)$ (với x là ảnh lấy từ domain X) sao cho giống với ảnh từ domain Y nhất, ngược lại Discriminator D_Y cố gắng cực đại hàm adversarial loss bằng cách phân biệt ảnh $G(x)$ và ảnh thật y từ domain.

Hàm mất mát của CycleGAN

a. Adversarial loss

Adversarial loss được áp dụng tương tự đối với generator F và Discriminator:

$$L_{adv}(F, D_X, Y, X) = \frac{1}{n}[\log D_X(x)] + \frac{1}{n}[\log(1 - D_X(F(y)))] \quad (6)$$

Hàm mất mát của CycleGAN

b. Cycle consistency loss

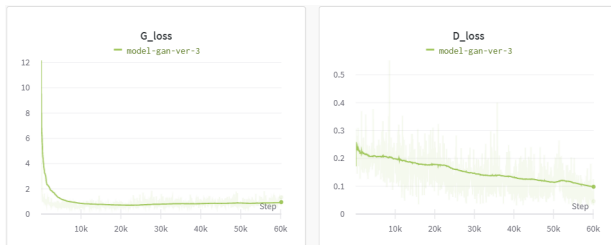
Chỉ riêng adversarial loss là không đủ để mô hình cho ra kết quả tốt. Nó sẽ lai generator theo hướng tạo ra được ảnh output bất kỳ trong domain mục tiêu chứ không phải output mong muốn. Để giải quyết vấn đề này, cycle consistency loss được giới thiệu:

$$L_{cycle}(G, F) = \frac{1}{n} \sum |F(G(x_i)) - x_i| + |G(F(y + i)) - y_i| \quad (7)$$

Công thức tổng quát:

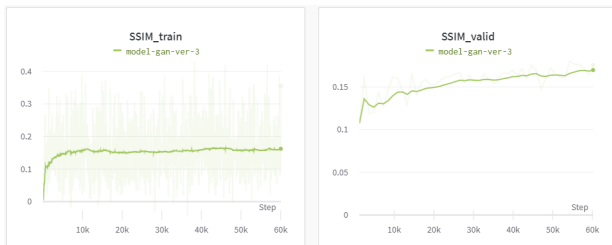
$$L = L_{adv}(G, D_Y, X, Y) + L_{adv}(F, D_X, Y, X) + \lambda L_{cycle}(G, F) \quad (8)$$

Kết quả



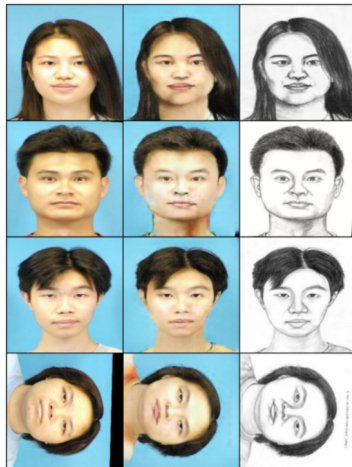
Hình: Generator Loss và Discriminator của CycleGAN

Kết quả



Hình: Đánh giá sử dụng SSIM

Kết quả



Hình: Kết quả dự đoán sử dụng CycleGAN

Nội dung

- 1 Giới thiệu
- 2 Dữ liệu
- 3 Autoencoder
- 4 Pix2Pix
- 5 CycleGAN
- 6 So sánh kết quả**
- 7 Hướng nghiên cứu tương lai

So sánh kết quả giữa ba mô hình và bài báo đối sánh

Thông số	Autoencoder	Pix2Pix	CycleGAN
D_loss		1.1917	0.046
D_loss val		1.1264	
G_loss		5.6730	0.9478
G_loss val		9.6808	
SSIM	0.8118	0.7002	0.1698
SSIM val	0.7135	0.6064	0.1624

Bảng: Kết quả độ của các mô hình

Thông số	Pix2Pix	CycleGAN
SSIM	0.6606	0.7626

Bảng: Kết quả SSIM trên bài báo đối chiếu

Nội dung

- 1 Giới thiệu
- 2 Dữ liệu
- 3 Autoencoder
- 4 Pix2Pix
- 5 CycleGAN
- 6 So sánh kết quả
- 7 Hướng nghiên cứu tương lai**

Hướng nghiên cứu tương lai

Tìm hiểu sâu hơn về các mô hình GAN, huấn luyện với các bộ data tương tự.

Thank you!