

TRƯỜNG ĐẠI HỌC XÂY DỰNG HÀ NỘI

Khoa Công Nghệ Thông Tin



BÁO CÁO ĐỒ ÁN

KHÔI PHỤC ẢNH

Sinh viên thực hiện:

Nhóm 4: Lớp 67CS1

- Hoàng Quang Duy 0190267
- Đỗ Văn Dũng 0300467
- Tô Tiến Dũng 0300567
- Đặng Hoàng Hải 0047167

Giảng viên hướng dẫn: Nguyễn Đình Quý

Hà Nội, tháng 11 năm 2024

Lời Cảm Ơn

Chúng em xin chân thành cảm ơn sự giúp đỡ tận tình của thầy Nguyễn Đình Quý, những góp ý quý báu của thầy đã giúp chúng em hoàn thiện đề án môn học xử lý ảnh một cách thuận lợi nhất.

Chúng em cũng xin gửi lời cảm ơn đến các thầy cô trong khoa công nghệ thông tin đã hỗ trợ, cung cấp tài liệu và môi trường học tập để đề án được hoàn thành đúng hạn.

Xin chân thành cảm ơn!

Mục lục

Lời Cảm Ơn	1
Danh sách hình vẽ	3
Danh mục từ viết tắt	4
1. Giới thiệu đề tài	5
2. Cơ sở lý thuyết	5
2.1. Lớp tích chập - Convolutional Layer	5
2.2. Hàm ReLU	6
2.3. Lớp tổng hợp - Pooling	6
2.4. Lớp Transposed Convolution	7
2.5. U-net	7
2.7. Chỉ số SSIM	9
2.8. Hàm Loss	9
2.9. Một số phương pháp làm sắc nét ảnh	9
3. Phương án đề xuất	10
3.1. Ý tưởng chung	10
3.2. Mô tả bộ dữ liệu sử dụng và tiền xử lý tạo dữ liệu đầu vào	10
3.3. Lựa chọn mô hình và hàm chi phí	11
3.4. Quy trình xử lý	12
4. Kết quả	13
5. Hạn chế và hướng phát triển	14
6. Kết luận	15

Danh sách hình vẽ

1	Phép tích chập với các loại kernel khác nhau	5
2	Hàm ReLU	6
3	Kiến trúc mạng U-net	8
4	Phân bố nhãn	10
5	Ảnh trước và sau xử lí	11
6	Kiến trúc U-net	11
7	Sự biến thiên của loss và SSIM trong quá trình huấn luyện	12
8	Quy trình khôi phục ảnh	12
9	Kết quả khôi phục trường hợp 1	13
10	Kết quả khôi phục trường hợp 2	13
11	Bảng kết quả so sánh	14
12	Ảnh khôi phục không giống vật thể gốc	14

Danh mục từ viết tắt

TC Tranposed Convolution

SSIM Structural Similarity Index Measurement

1. Giới Thiệu Đề Tài

Những bức ảnh xưa cũ luôn là chiếc cầu nối thời gian, lưu giữ những ký ức quý giá và giá trị đẹp đẽ của một thời đã qua. Tuy nhiên, không ít bức ảnh đã bị hư hỏng, khuyết hình do tác động của thời gian hoặc những sự cố ngoài ý muốn, để lại nỗi tiếc nuối sâu sắc cho người sở hữu.

Chính điều đó đã trở thành nguồn cảm hứng để chúng em thực hiện đề tài đồ án này: Khôi phục ảnh cũ, nhằm đem lại sức sống mới cho những khoảnh khắc tưởng chừng đã mất, giúp lưu giữ và trân trọng hơn những giá trị ký ức vô giá.

Phát biểu bài toán: Cho một hình ảnh X bị che khuất một phần ảnh, khôi phục X về trạng thái giống ban đầu nhất có thể.

- Input: Ảnh X bị che khuất.
- Output: Ảnh X_{pred} được khôi phục từ X

2. Cơ sở lý thuyết

2.1. Lớp tích chập - Convolutional Layer

Đối với hình ảnh, ta sử dụng tích chập 2 chiều. Ảnh X là ma trận đầu vào, và sử dụng một ma trận F khác làm bộ lọc. Ta dùng F trượt qua X , Tại mỗi vị trí, ta lấy tích các phần tử của F và phần con của X , sau đó cộng tất cả các kết quả này lại với nhau.

Bằng cách sử dụng các bộ lọc khác nhau, phép tích chập có thể ứng dụng trong việc phát hiện cạnh, làm mờ ảnh hoặc làm ảnh sắc nét...



Hình 1: Phép tích chập với các loại kernel khác nhau

Phép tích chập được thực hiện bằng cách sử dụng ma trận bộ lọc trượt trên ma trận đầu vào từ trái qua phải và từ trên xuống dưới theo một bước nhảy S . Để tránh bị mất thông tin ở vùng rìa ảnh, chúng ta cần thêm một vùng đệm P xung quanh 4 cạnh của ảnh, các giá trị pixel trong vùng đệm luôn bằng 0. Kết thúc phép tích chập, ta thu được trọn vẹn một ma trận đầu ra \mathbf{Y} .

Trong mạng Nơ ron tích chập, các tầng liên sau lấy đầu vào từ tầng liên trước nó, do đó chúng ta cần xác định được kích thước đầu ra ở mỗi tầng. Kích thước đầu ra được tính theo công thức:

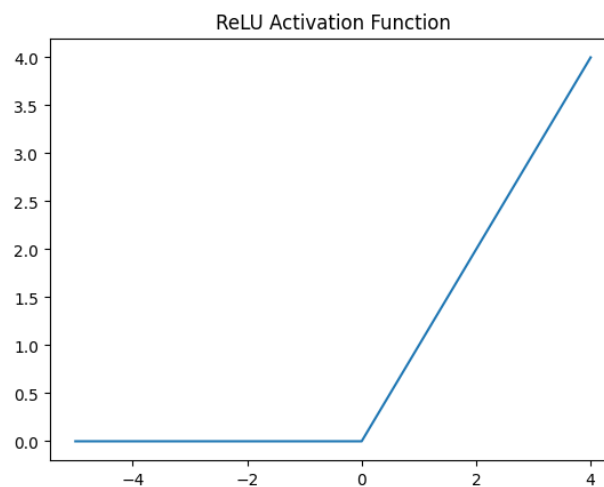
$$OutputShape = \frac{InputShape + 2 \times P - FilterShape}{S}$$

2.2. Hàm ReLU

Hàm kích hoạt là những hàm được sử dụng để tạo sự phi tuyến tính và giữ các giá trị output từ các layer trong khoảng nhất định.

Trong mạng nơ ron tích chập, hàm kích hoạt được sử dụng phổ biến nhất là hàm ReLU, công thức:

$$output = \max(0, output)$$



Hình 2: Hàm ReLU

Ưu điểm của hàm Relu là tính đơn giản của nó và nó đã được chứng minh là giúp tăng tốc quá trình training. Tiếp theo là nó không bị chặn cho nên sẽ không xảy ra hiện tượng vanishing gradient (khi mà độ dốc trở nên quá nhỏ để cập nhật tham số).

2.3. Lớp gộp - Pooling

Việc độ sâu của mạng tăng lên khiến cho số lượng tham số cũng tăng theo cấp số nhân, dẫn đến khối lượng tính toán cũng tăng mạnh. Để giảm tải tính toán, vấn đề bức thiết là cần phải giảm kích thước các chiều của khối ma trận đầu vào \mathbf{M} mà vẫn giữ được những đặc trưng cần thiết của \mathbf{M} .

Ta sử dụng một bộ lọc trượt từ trái sang phải, từ trên xuống dưới ma trận \mathbf{M} theo một bước nhảy \mathbf{S} , tìm ra giá trị đại diện của mỗi vùng để giữ lại. Kết quả của quá trình này là một ma trận mới nhỏ hơn nhưng vẫn giữ được các đường nét chính của ma trận \mathbf{M} . Tương tự như lớp tích chập, ta cũng có một công thức để tính kích thước ma trận đầu ra như sau:

$$OutputShape = \frac{InputShape - FilterShape}{S}$$

Có nhiều loại pooling khác nhau để ta có thể lựa chọn:

- **Max Pooling:** Lấy phần tử lớn nhất trong mỗi lần dịch chuyển ma trận bộ lọc.
- **Sum Pooling:** Lấy tổng các phần tử.
- **Average Pooling:** Lấy giá trị trung bình các phần tử.

2.4. Lớp Transposed convolution

Có thể coi là phép toán ngược của tích chập. Quy trình thực hiện cũng là sử dụng một bộ lọc trượt từ trái sang phải, từ trên xuống dưới ma trận đầu vào, tuy nhiên đầu ra lại là ma trận có kích thước lớn hơn so với Input.

Việc thực hiện TC cần xác định trước kích thước bộ lọc F , bước nhảy S và vùng đệm P , cách hoạt động được giải thích qua 4 bước sau:

- Bước 1: Tính giá trị $z = S - 1$ và $p' = F - P - 1, s' = 1$
- Bước 2: Giữa các hàng và các cột của input chúng ta chèn z hàng 0, z cột 0, Ví dụ nếu $z = 1$ Kích thước của input mới là $(2i - 1) * (2i - 1)$ trong đó i là kích thước input ban đầu.
- Bước 3: Thêm vùng đệm vào ảnh nhận được ở bước trên với độ dày bằng p' .
- Bước 4: Thực hiện phép tích chập thông thường với ảnh được tạo thành ở bước 3 với bước nhảy $s' = 1$

Kích thước của đầu ra cho TC được tính theo công thức:

$$OutputShape = (InputShape - 1) * s + FilterShape - 2 * p$$

2.5. U-net

Unet là một kiến trúc mạng nơ ron tích chập được phát triển bởi Olaf Ronneberger và các cộng sự phát triển nhằm phân vùng các cấu trúc nơ ron thần kinh trong não người. Kiến trúc U-net được thiết kế đặc biệt nhằm xử lý các tác vụ phân đoạn ảnh, và phục hồi các thông tin chi tiết từ ảnh.

Kiến trúc của mạng U-net có hình chữ U gồm 2 phần thu hẹp và mở rộng tương ứng với nhánh bên trái thực hiện downsampling và nhánh bên phải thực hiện upsampling. U-net triển khai các kết nối tắt đối xứng giữa các lớp tương ứng cùng tầng trong hai nhánh.

Downsampling(Giảm kích thước không gian): Sau vài lớp tích chập, U-net sử dụng max pooling để giảm kích thước của ảnh, quá trình này giúp mạng tập trung vào các đặc trưng hơn và giảm độ phức tạp tính toán, Thêm vào đó, mỗi lớp tích chập đi kèm với ReLU activation để cải thiện độ ổn định và khả năng học của mô hình.

Upsampling: tăng kích thước ảnh trở lại sau khi giảm qua phần mã hóa bằng cách sử dụng các lớp upsample, chủ yếu là Transposed convolution để tăng kích thước ảnh, sau đó kết hợp (concatenate) với các đặc trưng tương ứng từ nhánh trái. Quá trình này giúp mạng sử dụng lại các thông tin chi tiết từ phần mã hóa để tái tạo ảnh tốt hơn. Sau khi kết hợp, ảnh sẽ được xử lý thêm bằng các lớp convolution để học các đặc trưng phức tạp và cải thiện chất lượng kết quả.

Từ đó ta có thể rút ra chức năng của hai nhánh như sau:

Nhánh trái: Phần thu hẹp:

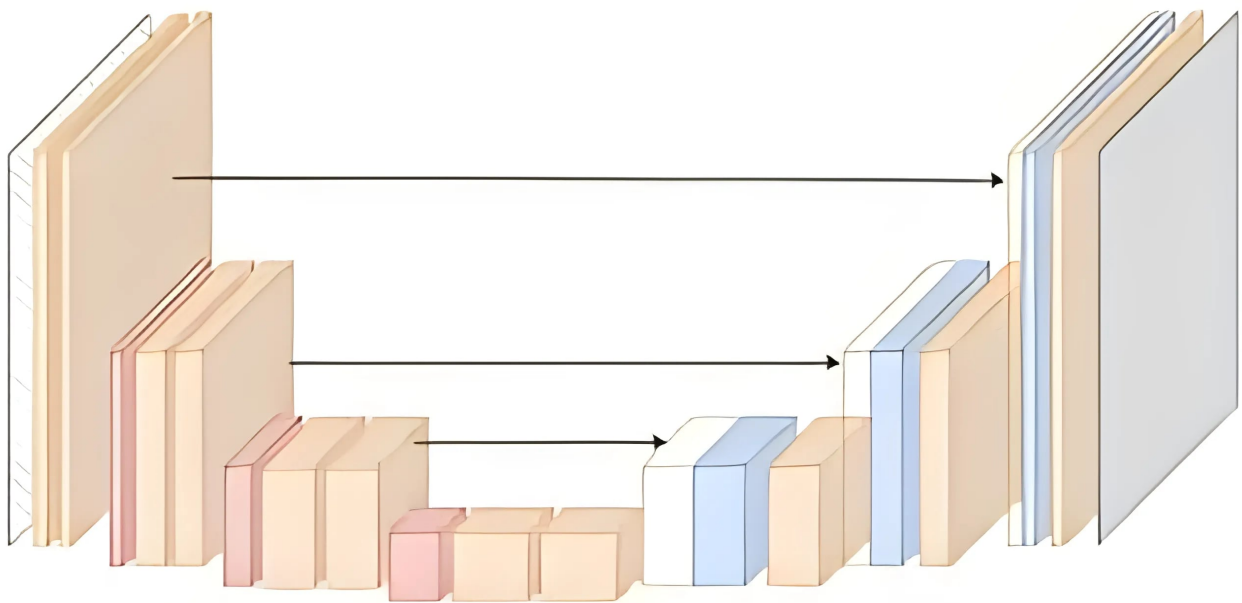
Làm nhiệm vụ trích lọc đặc trưng để tìm ra bối cảnh của hình ảnh. Lý do nhánh được gọi là thu hẹp vì kích thước dài và rộng của các lớp giảm dần.

Trích xuất đặc trưng từ ảnh đầu vào.

Nhánh phải: Phần mở rộng:

Có chức năng khôi phục kích thước không gian của ảnh sau khi đã trải qua quá trình mã hóa ở nhánh trái, giúp tái cấu trúc ảnh đầu ra thành một ảnh có kích cỡ tương tự như ảnh đầu vào, đồng thời giữ lại những đặc trưng đã học được trước đó.

U-Net có khả năng phục hồi thông tin không gian tốt nhờ vào các tầng giải mã, điều này giúp nó tái tạo các cấu trúc hình ảnh một cách mượt mà và chính xác trong các vùng bị thiếu, tuy nhiên tốc độ huấn luyện lại khá chậm do lượng tham số lớn hơn so với mô hình học thông thường.



Hình 3: Kiến trúc mạng U-net

2.7. Chỉ số SSIM

Chỉ số SSIM được sử dụng để đo mức độ giống nhau giữa hình ảnh đầu vào và ảnh sinh ra. Công thức SSIM dựa trên ba thông số để so sánh: độ chói (là giá trị trung bình của các điểm ảnh trong ảnh), tương phản (phương sai của giá trị các điểm ảnh) và cấu trúc (hiệp phương sai của hai ảnh).

Với x và y là hai ảnh cần so sánh, Công thức tính chỉ số này như sau:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

Trong đó:

- μ_x và μ_y là giá trị trung bình (luminance) của hình ảnh x và y .
- σ_x^2 và σ_y^2 là phương sai (variance) của hình ảnh x và y .
- σ_{xy} là hiệp phương sai (covariance) giữa x và y .
- C_1 và C_2 là các hằng số nhỏ để tránh chia cho 0

Chỉ số SSIM có giá trị càng lớn thì độ tương đồng càng cao (tối đa là 1), tức là ảnh được tạo ra càng giống ảnh gốc.

2.8. Hàm Loss

Hàm loss được sử dụng để đánh giá chất lượng của mô hình trong quá trình huấn luyện, hay nói cách khác, hàm loss định hướng quá trình huấn luyện. Đầu ra của hàm là một số không âm thể hiện sự chênh lệch giữa hai đại lượng y_{true} và y_{pred} , mục tiêu của huấn luyện là đưa đầu ra này về giá trị cực tiểu.

Một số hàm loss:

Mean Absolute Error: $loss = |y_{true} - y_{pred}|$ được dùng cho các bài toán hồi quy đơn giản.

Mean Square Error: $loss = (y_{true} - y_{pred})^2$ dùng cho những bài toán hồi quy có dạng số không quá lớn.

Cross-Entropy Loss: $loss = -\sum x \log(y)$, với x là xác suất trả ra của nhãn thật và y là xác suất nhãn được dự đoán, dùng cho bài toán phân loại.

SSIM Loss: $loss = 1 - SSIM(y, y_{pred})$ Loại loss này dựa trên chỉ số SSIM, nếu SSIM là độ giống nhau, vậy $1 - SSIM$ chính là độ khác nhau, mục tiêu là giảm giá trị này đi nhiều nhất có thể.

2.9. Một số phương pháp làm sắc nét ảnh

Làm sắc nét ảnh (image sharpening) là quá trình xử lý hình ảnh nhằm làm tăng độ tương phản và làm rõ các chi tiết trong ảnh, giúp cho các đường nét, kết cấu và các đối tượng trong ảnh trở nên sắc nét hơn. Quá trình này thường được sử dụng để cải thiện độ rõ ràng của ảnh, đặc biệt là trong các trường hợp ảnh bị mờ hoặc thiếu chi tiết.

Chúng em chọn được hai phương pháp làm sắc nét để hỗ trợ việc khôi phục ảnh đó là:

Unsharp Masking: Làm mờ ảnh ban đầu bằng cách sử dụng Gaussian Blur, sau đó trừ ảnh mờ khỏi ảnh gốc để tạo ra các chi tiết sắc nét hơn.

Kernel Sharping: Áp dụng một bộ lọc (kernel) để làm sắc nét ảnh. Bộ lọc thường thấy nhất là:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

3. Phương án đề xuất

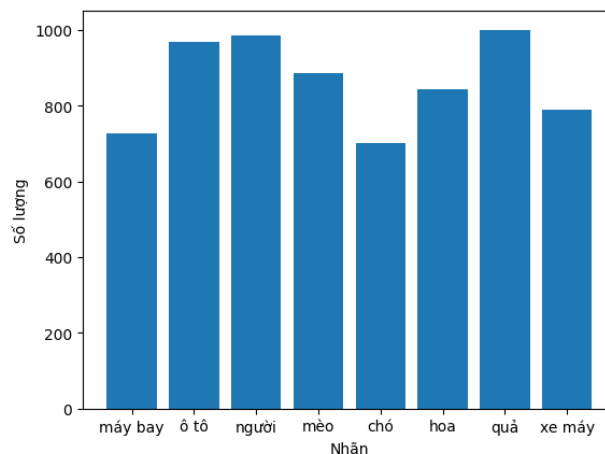
3.1. Ý Tưởng Chung

Khi đối mặt với những bức ảnh bị khuyết, cách tiếp cận tự nhiên của chúng ta thường là dựa vào hiểu biết sẵn có: từ cấu trúc vật thể, màu sắc xung quanh, cho đến những chi tiết liên kết trong bức ảnh. Tuy nhiên, việc hoàn thiện thủ công một bức ảnh bị mất đi những mảng lớn không chỉ đòi hỏi sự kiên nhẫn mà còn tiêu tốn rất nhiều thời gian và công sức.

Giải pháp của chúng em là áp dụng mạng nơ ron nhân tạo, cho phép máy tính học và hiểu các đặc điểm của vật thể, từ đó tái tạo và lấp đầy những phần bị thiếu trong bức ảnh một cách thông minh. Nói cách khác, ý tưởng của đề án này là sự mô phỏng cách suy nghĩ và tái tạo của con người thông qua sức mạnh của trí tuệ nhân tạo.

3.2. Mô tả bộ dữ liệu sử dụng và tiền xử lí tạo dữ liệu huấn luyện

Chúng em sử dụng bộ dữ liệu **Natural image** được công bố trên **Kaggle**, bộ dữ liệu bao gồm 6899 hình ảnh từ 8 lớp khác nhau: máy bay, ô tô, người, mèo, chó, hoa, quả, xe máy. Các lớp có số lượng mẫu khá đồng đều:



Hình 4: Phân bố nhãn

Việc xử lí dữ liệu gồm 2 bước:

Bước 1: Chỉnh sửa lại cỡ ảnh về 128×128

Bước 2: Tạo một mặt nạ và áp dụng nó lên ảnh thu được từ bước trước để mô phỏng ảnh bị hỏng.



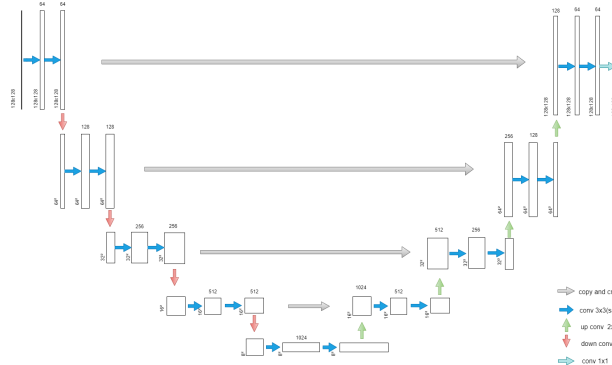
Hình 5: Ảnh trước và sau xử lý

Ảnh sau bước hai và ảnh sau bước một được nhóm với nhau thành một cặp mẫu-nhãn phục vụ việc huấn luyện.

Sau khi xử lý, tập dữ liệu ban đầu được chia thành hai tập huấn luyện và tập kiểm thử với tỉ lệ 8:2.

3.3. Lựa chọn mô hình và hàm chi phí

Mô hình: Chúng em chọn mô hình Unet với 4 lớp Upsample và 4 lớp Downsamle, với cấu trúc như hình phía dưới:



Hình 6: Kiến trúc U-net

Do Unet có lượng tham số nhiều hơn so với các mô hình thông thường, huấn luyện lượng tham số khổng lồ tốn rất nhiều thời gian và tài nguyên tính toán, thêm vào đó, việc xử lý ảnh cỡ lớn hơn yêu cầu độ sâu của mạng cũng phải tăng lên để đảm bảo độ chính xác, và điều đó làm lượng tham số tăng rất nhiều. Xem xét về mặt thời gian huấn luyện cũng như tài nguyên phần cứng, chúng em đã chọn cỡ ảnh 128x128 để thuận tiện cho quá trình thử nghiệm.

Hàm chi phí: Trong khuôn khổ đề án môn học, chúng em quyết định sử dụng kết hợp hai hàm loss là hàm Mean Square Error và SSIM loss. MSE sẽ đánh giá mức độ chính xác trên từng pixel, còn SSIM loss sẽ đánh giá độ chính xác về mặt cấu trúc tổng thể. Công thức cho hàm chi phí cuối cùng là:

$$loss = MSELoss + SSIMLoss = (y_{true} - y_{pred})^2 + 1 - SSIM(y, y_{pred})$$

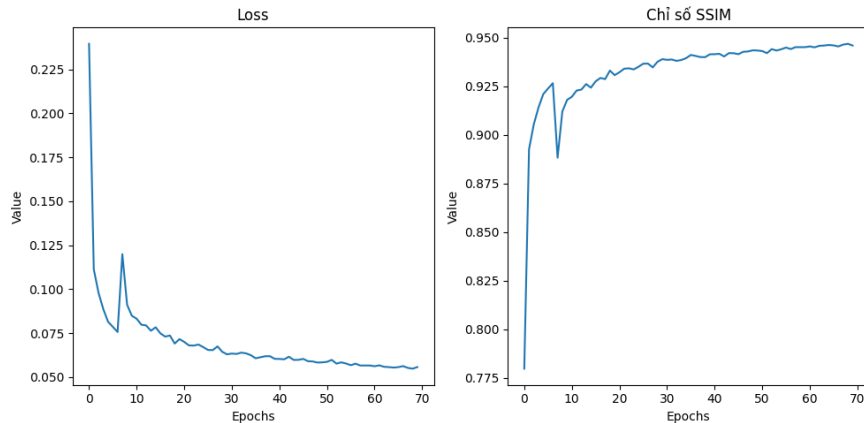
Thông số máy huấn luyện, lựa chọn tham số: 16 GB RAM, 2 GPU T4 (16 GB VRAM), tất cả được cung cấp bởi **Kaggle**. Các tham số huấn luyện được chọn như sau:

Learning rate: 0.001

Batch size (Số ảnh truyền vào trong mỗi lượt): 32 ảnh

Epoch (Số vòng huấn luyện): 120

Sự thay đổi của hàm chi phí trong quá trình học trong 70 epoch đầu được thể hiện trong hai biểu đồ dưới đây:



Hình 7: Sự biến thiên của loss và SSIM trong quá trình huấn luyện

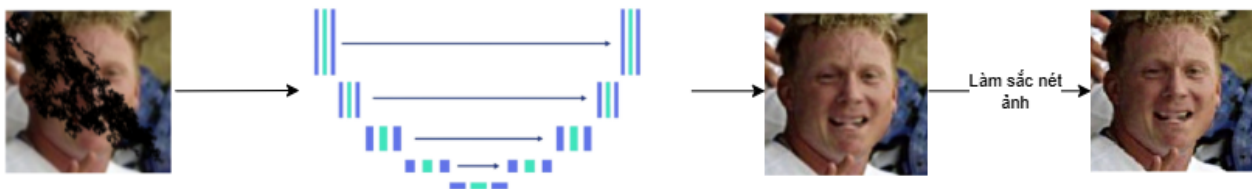
Ta có thể thấy cả hai chỉ số loss và SSIM đều được cải thiện sau mỗi vòng huấn luyện, điều này cho thấy mô hình đang học một cách có hiệu quả.

3.4. Quy trình xử lý

Quy trình khôi phục bao gồm hai bước: khôi phục và làm sắc nét, cụ thể như sau:

Khôi phục: Ảnh được đưa vào mô hình học, cho ra ảnh đã xóa những mảng bị che khuất.

Làm sắc nét: Ảnh được làm net bằng một quy trình gồm 2 phương pháp làm sắc nét: sử dụng lần lượt unsharp masking và nhân tích chập với một kernel làm sắc nét ảnh.



Hình 8: Quy trình khôi phục ảnh

4. Kết Quả

Chúng em sẽ tiến hành hai kiểm tra hai trường hợp: trường hợp 1 kiểm tra mô hình trên các ảnh giống với tập huấn luyện, thí nghiệm 2 kiểm tra khả năng phục hồi trên những ảnh không có trong tập huấn luyện.

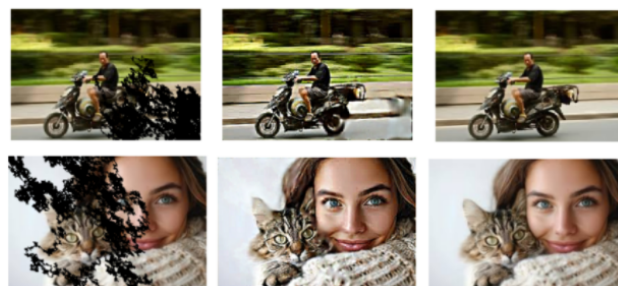
Trong các ảnh minh họa kết quả, kết quả các ảnh sẽ được trình bày theo một cách duy nhất: từ trái qua phải, ảnh đầu tiên là ảnh đầu vào, ảnh ở giữa là ảnh đã khôi phục, và ảnh bên phải là ảnh gốc.

Trường hợp 1: Các ảnh thử nghiệm được lấy từ tập test đã chia ra từ trước, đảm bảo không có trong tập huấn luyện.



Hình 9: Kết quả khôi phục trường hợp 1

Kết quả khôi phục Ảnh trong phần này được lấy từ trên internet, những vật thể trong ảnh sẽ nằm trong tập 10 nhân trong tập huấn luyện.



Hình 10: Kết quả khôi phục trường hợp 2

Trong cả hai trường hợp, bước đầu có thể thấy các ảnh được phục hồi gần như giống hệt với ảnh gốc và chất lượng ảnh được tăng cường. điều này chứng tỏ mô hình học đã được huấn luyện tốt, có khả năng xử lí những bức ảnh thực tế và bộ lọc làm net đã hoạt động hiệu quả.

Để cung cấp một góc nhìn khách quan hơn, chúng em đã tiến hành thử nghiệm trên 4 bộ dữ liệu khác nhau bao gồm: **Natural Image**, **Human faces dataset** bao gồm các bức ảnh về khuôn mặt người, **Cats and Dogs image dataset** Gồm toàn các bức ảnh chó và mèo, **Flower image dataset** là bộ dữ liệu ảnh của các loài hoa. Lấy chỉ số SSIM làm thước đo độ hiệu quả, kết quả được thể hiện trong bảng dưới đây:

	Chỉ số SSIM trung bình
Test set - Natural Dataset	0.9499
Full dataset - Natural Dataset	0.9504
Full dataset - Human faces dataset	0.9551
Cats and Dogs image dataset	0.9457
Flower image dataset	0.9102

Hình 11: Bảng kết quả so sánh

Có thể thấy rằng, mô hình hoạt động tốt với cả 4 bộ dữ liệu với chỉ số SSIM đều trên 0.9. Bộ dữ liệu có chỉ số cao nhất là **Human faces dataset** với gần 0.96 điểm và thấp nhất là **Flower image dataset** với 0.91 điểm. Từ đó ta có thể thấy rằng với những cấu trúc có tính đồng dạng (như khuôn mặt ở người đều có 2 mắt, 1 mũi...) thì mô hình có thể học tập và khôi phục rất hiệu quả, còn đối với những cấu trúc có tính đa dạng cao như các bông hoa (số lượng cánh, chủng loại...) thì mô hình vẫn chưa thực sự tốt.

5. Hạn chế và hướng phát triển

Tuy mô hình đã hoạt động hiệu quả trong đa phần các trường hợp, tuy nhiên cũng còn tồn tại những điểm cần cải tiến thêm, ví dụ như trường hợp dưới đây:



Hình 12: Ảnh khôi phục không giống vật thể gốc

Ta có thể nhìn thấy hai vấn đề lớn đang hiện hữu từ trường hợp này:

Thứ nhất: Những chi tiết nhỏ và phức tạp như mắt thường bị phục hồi sai ở trên những mẫu được thể hiện khá khác nhau mặc dù nằm chung một nhãn, mô hình có xu hướng lấy màu sắc xung quanh đắp vào phần mắt.

Thứ hai: Đối với ảnh bị che khuất một mảng lớn liên tục, khu vực đó thường bị phục hồi sai, hoặc bị nhòe hình.

Nhóm em đã nghiên cứu và đưa ra kết luận rằng hai vấn đề trên xảy ra do các mẫu trong cùng một nhãn thường không có cùng một phân bố pixel và do lượng dữ liệu huấn luyện chưa đủ nhiều.

Hướng phát triển trong tương lai:

Hướng thứ nhất: Có thể phân cụm các ảnh trong cùng một nhãn theo phân bố các điểm ảnh và trong mỗi lần lấy mẫu huấn luyện sẽ lấy từ một cụm.

Hướng thứ hai: Bổ sung thêm dữ liệu huấn luyện.

Do lượng tài nguyên phần cứng và nhân lực có hạn, hai hướng trên chúng em đều chưa thể triển khai ở thời điểm hiện tại.

6. Kết Luận

Đồ án khôi phục ảnh đã hoàn thành mục tiêu đề ra, phục hồi các bức ảnh bị hỏng về trạng thái ban đầu. Mặc dù trong quá trình thực hiện còn gặp phải một số sai sót, chúng em đã nỗ lực và tự giải quyết được các vấn đề phát sinh. Tuy nhiên, đồ án không thể tránh khỏi những hạn chế, và nếu có bất kỳ điểm nào chưa hoàn thiện, chúng em rất mong nhận được sự góp ý để có thể cải thiện và phát triển thêm trong tương lai.