

VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



HỌC MÁY (CO3117)

IMAGE SUPER RESOLUTION

CÁC MẠNG NƠRON CẢI TIẾN

Giảng viên hướng dẫn: Nguyễn Đức Dũng

Sinh viên thực hiện: Phạm Như Thuần - 2213349

HO CHI MINH CITY, DECEMBER 2024



1 Giới thiệu đề tài

1.1 Động lực

Image Super Resolution (ISR) là một trong những bài toán quan trọng trong xử lý hình ảnh và thị giác máy, nhất là trong bối cảnh các ngành đòi hỏi hình ảnh có độ phân giải cao. Việc nâng cao độ phân giải giúp tăng cường chi tiết đem lại nhiều lợi ích trong y tế, truyền thông, giám sát an ninh, và nhiều ngành khác.

Ví dụ thực tiễn của việc ứng dụng ISR:

- Y tế: Nâng cao chất lượng hình ảnh y khoa như MRI, CT để phân tích chi tiết bệnh lý.
- Truyền thông: Cải thiện chất lượng hình ảnh, video cũ hoặc bị lỗi.
- An ninh: Tái tạo hình ảnh độ phân giải thấp phục vụ cho việc giám sát

Mục tiêu của bài toán này là để so sánh các mô hình đã được cải tiến, từ đó đưa ra những mô hình phù hợp cho bài toán cụ thể hơn.

1.2 Mô tả bài toán

Bài toán ISR nhấn mạnh việc chuyển đổi hình ảnh từ chất lượng thấp sang chất lượng cao. Đối với các phương pháp interpolation truyền thống như bicubic, việc chuyển đổi chỉ đơn thuần làm tăng kích thước hình ảnh mà không tăng cường chi tiết. Vì vậy trong bài toán này, ta sẽ dùng các kỹ thuật hiện đại hơn như sử dụng mạng học sâu giúp cải thiện chi tiết tốt hơn các phương pháp interpolation truyền thống.

1.3 Các giới hạn

Mặc dù ISR mang lại nhiều ứng dụng quan trọng, nhưng việc thực hiện vẫn gặp phải nhiều thách thức. Vì vậy bài toán này chỉ giải quyết các vấn đề trong giới hạn sau:

- Các chi tiết có thể không được rõ ràng hoặc đáp ứng được mong đợi từ người dùng.
- Chỉ áp dụng các mô hình nhỏ có thể huấn luyện được. Đối với các mô hình lớn cần nhiều tài nguyên để huấn luyện thì chỉ sử dụng mô hình đã được huấn luyện trước.
- Hình ảnh trong dữ liệu huấn luyện không thể đại diện cho tất cả tình huống thực tế, nên kết quả có thể không đạt yêu cầu.

2 Literature survey

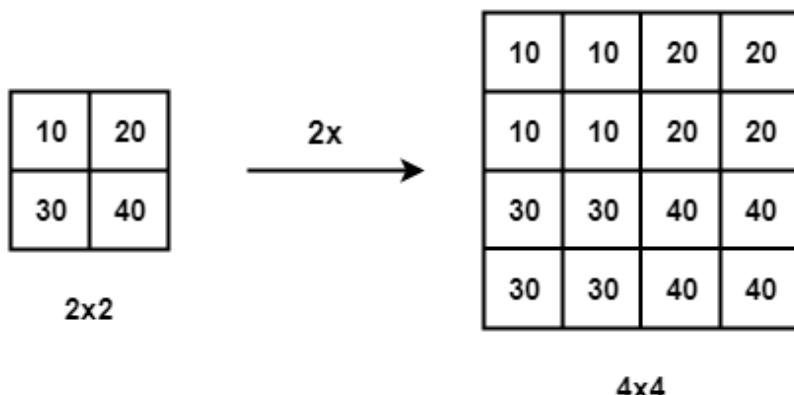
2.1 Interpolation

Interpolation là phương pháp làm tăng kích thước ảnh số và được sử dụng nhiều trong xử lý ảnh. Những phương pháp interpolation truyền thống bao gồm:

- Nearest-neighbor interpolation: chọn giá trị của pixel gần nhất cho mỗi vị trí cần nội suy mà không quan tâm đến bất kỳ pixel nào khác. Là một thuật toán đơn giản nhưng cho ra chất lượng ảnh thấp
- Bilinear interpolation: Bilinear interpolation đầu tiên thực hiện nội suy tuyến tính trên một trực của hình ảnh và sau đó thực hiện trên trực còn lại. Vì kết quả là một phép

interpolation bậc hai với vùng tiếp nhận (receptive field) kích thước 2×2 , phương pháp này cho hiệu năng tốt hơn nhiều so với nội suy hàng xóm gần nhất trong khi vẫn giữ được tốc độ tương đối nhanh

- Bicubic interpolation: Tương tự, Bicubic interpolation thực hiện nội suy trên từng trục của hình ảnh. So với BLI, BCI tính toán trên 4×4 pixel và cho ra kết quả mượt mà hơn với ít hiện tượng nhiễu (artifacts) hơn, nhưng tốc độ lại chậm hơn đáng kể



Hình 1: Nearest-neighbor Interpolation

2.2 Convolutional Neural Network

2.2.1 SRCNN

Super-Resolution Convolutional Neural Network (SRCNN) là mô hình đầu tiên áp dụng mạng nơ-ron tích chập (CNN) cho bài toán tăng cường độ phân giải ảnh, được đề xuất bởi Dong et al. vào năm 2014. Kiến trúc của SRCNN bao gồm ba thành phần chính: trích xuất đặc trưng, ánh xạ phi tuyến, và tái tạo ảnh. Mặc dù đạt hiệu quả tốt hơn các phương pháp truyền thống như nội suy bicubic, SRCNN có độ sâu mạng thấp và khả năng học các chi tiết phức tạp còn hạn chế.

2.2.2 ESPCN

Tiếp nối SRCNN, Efficient Sub-Pixel Convolutional Network (ESPCN) được Shi et al. giới thiệu năm 2016 nhằm cải thiện hiệu suất. Thay vì nội suy trước như SRCNN, ESPCN sử dụng kỹ thuật sub-pixel convolution để phóng to ảnh trực tiếp từ đặc trưng, mang lại tốc độ nhanh và tái cấu trúc ảnh hiệu quả hơn. Tuy nhiên, khả năng khôi phục chi tiết của ESPCN chưa bằng các mô hình hiện đại.

2.2.3 EDSR

Để cải thiện chất lượng hình ảnh, Enhanced Deep Super-Resolution Network (EDSR), được Lim et al. giới thiệu năm 2017, đã loại bỏ Batch Normalization và sử dụng các khối dư (Residual Blocks) để khai thác sâu hơn các đặc trưng ảnh. Kết quả là EDSR đạt hiệu suất cao, có thể tái tạo chi tiết phức tạp với chất lượng vượt trội, nhưng yêu cầu tài nguyên tính toán lớn



2.3 Generative Model

2.3.1 SRGAN

Super-Resolution Generative Adversarial Network (SRGAN) là một trong những mô hình tiên phong áp dụng mạng GAN (Generative Adversarial Network) cho bài toán tăng cường độ phân giải ảnh, được Ledig et al. giới thiệu vào năm 2017. Kiến trúc của SRGAN bao gồm hai thành phần chính: Generator (mạng sinh) và Discriminator (mạng phân biệt). Generator được thiết kế để tái tạo ảnh có độ phân giải cao từ ảnh đầu vào có độ phân giải thấp, trong khi Discriminator học cách phân biệt giữa ảnh thực và ảnh do Generator tạo ra. Để cải thiện chất lượng hình ảnh, SRGAN sử dụng Perceptual Loss, bao gồm Content Loss và Adversarial Loss. Trong đó, Content Loss đo lường sự khác biệt giữa các đặc trưng ảnh đầu ra và ảnh gốc thông qua một mạng pre-trained như VGG, giúp tối ưu hóa chi tiết ảnh.

SRGAN là bước tiến lớn so với các mô hình trước đó như SRCNN và ESPCN vì không chỉ cải thiện độ sắc nét mà còn tạo ra hình ảnh có chất lượng trực quan cao hơn, giống thực hơn. Tuy nhiên, SRGAN có thể gặp hiện tượng artifacts (nhiều không mong muốn) ở một số vùng ảnh và yêu cầu tài nguyên tính toán lớn do tính phức tạp của GAN.

2.3.2 ESRGAN

Cải tiến từ SRGAN, Enhanced Super-Resolution Generative Adversarial Network (ESRGAN) ra đời vào năm 2018 với các cải tiến như Residual-in-Residual Dense Blocks (RRDB) để khôi phục chi tiết tốt hơn và một Relativistic Discriminator giúp tăng tính chân thực của ảnh đầu ra. ESRGAN đã khắc phục các hạn chế của SRGAN và trở thành một trong những mô hình hàng đầu trong lĩnh vực Image Super-Resolution, tái tạo ảnh với độ chi tiết và chất lượng vượt trội.

2.3.3 Real-ERGAN

Real-ERGAN là một mô hình siêu phân giải ảnh được phát triển dựa trên ESRGAN, nhằm mục đích giải quyết các vấn đề phục hồi ảnh bị suy giảm chất lượng trong thế giới thực. Mô hình này được huấn luyện bằng dữ liệu tổng hợp, sử dụng quy trình suy giảm bậc cao để mô phỏng các yếu tố làm giảm chất lượng ảnh phức tạp như blur, noise, và nén JPEG. Real-ERGAN sử dụng bộ lọc sinc để tái tạo các hiện tượng nhiễu như ringing và overshoot. Ngoài ra, Real-ERGAN còn sử dụng kiến trúc bộ phân biệt U-Net với spectral normalization để tăng cường khả năng phân biệt và ổn định quá trình huấn luyện. Mục tiêu chính của Real-ERGAN là khôi phục chi tiết và loại bỏ các nhiễu ảnh trong các ảnh thực tế, vượt trội so với các phương pháp siêu phân giải trước đây trên các ảnh bị suy giảm chất lượng.

3 Phương pháp tiếp cận sử dụng Machine Learning

Trong bài toán này ta sẽ tập trung vào 3 mô hình là ESPCN, EDSR đã qua điều chỉnh và Real-ERGAN.

3.1 ESPCN

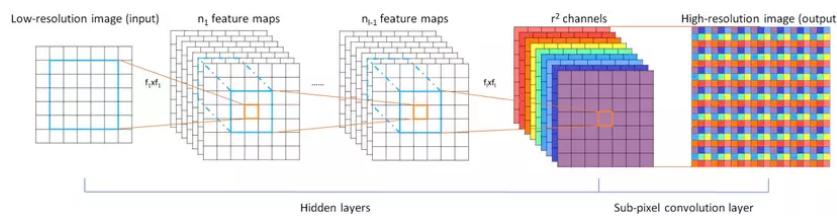
3.1.1 Vấn đề của SRCNN

Trong mạng SRCNN, để xử lý một ảnh input low-resolution (LR), tác giả đã dùng phương pháp bicubic interpolation để upsample ảnh lên sao cho nó có kích thước giống với ảnh high-resolution (HR). Điều này có hai điểm bất lợi sau:

- Upscale ảnh input và xử lý ảnh đã upsample trong model khiến chi phí tính toán tăng lên n^2 lần nếu kích thước ảnh tăng n lần. Điều này không phù hợp với ứng dụng real-time.
- Không cải thiện thông tin: Bicubic interpolation không cung cấp thêm thông tin hữu ích và còn làm ảnh hưởng đến kết quả model.

Do đó, tác giả bài báo Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network đã đề xuất ra một phương pháp mới để giải quyết 2 điểm yếu trên. Thay vì thực hiện việc upscale ngay ở input để khớp với kích thước high resolution của output, họ đề xuất thực hiện việc này ở cuối mạng để giảm chi phí tính toán của model.

3.1.2 Kiến trúc mạng ESPCN

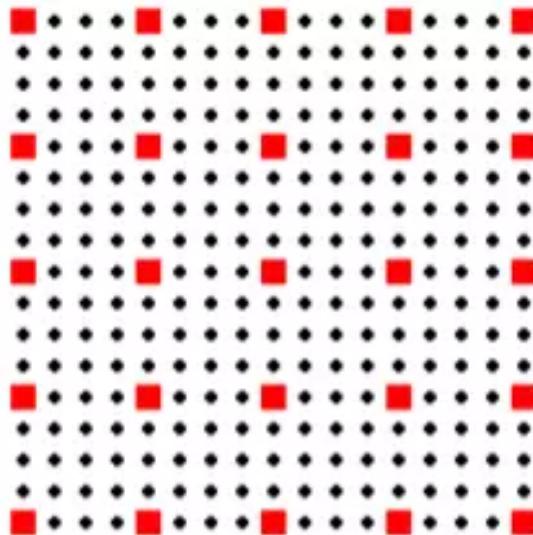


Hình 2: Kiến trúc ESPCN

Mạng ESPCN thực hiện feature extraction tương tự SRCNN nhưng không sử dụng bicubic interpolation để upsample ảnh. Ảnh LR được đưa trực tiếp qua các convolution layer để trích xuất feature map trong không gian LR. Sau đó, từ feature map LR $rW \times rH \times C$, mạng sẽ tái tạo ảnh HR $rW \times rH \times C$ (với rr là hệ số upscale).

3.1.3 Sub-Pixel

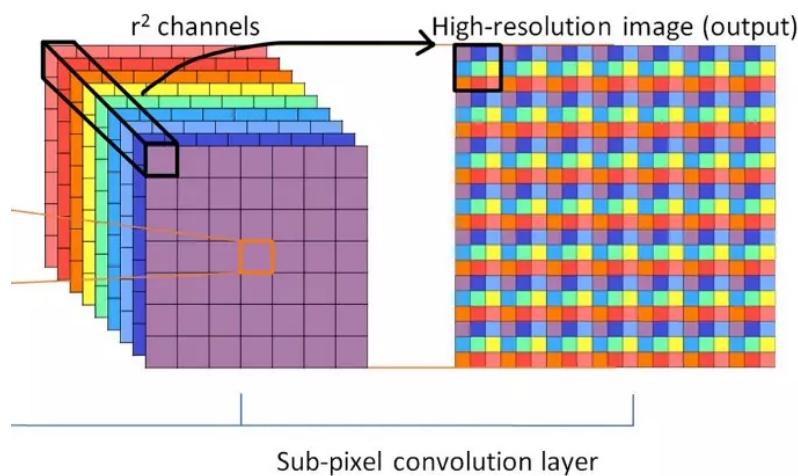
Khi chụp một tấm ảnh số, hệ thống hình ảnh (imaging system) của camera sẽ chiếu khung cảnh (scene) lên một image plane rồi thực hiện bước sampling và quantize để ra được một tấm ảnh số. Bước sampling ở đây sẽ dùng để số hoá (digitize) toạ độ lấy mẫu ra các pixel, còn bước quantize dùng để số hoá giá trị của từng pixel. Do giới hạn của sensor, ảnh sẽ thường bị giới hạn ở một độ phân giải nhất định. Vì thế, trên ảnh đó ta sẽ không có thêm thông tin gì ở giữa hai pixel cạnh nhau. Tuy nhiên, ở trong thế giới thực, ta có thể có rất nhiều các pixel nằm giữa 2 pixel đó. Các pixel nằm giữa đó sẽ được gọi là sub-pixel. Như hình ví dụ dưới đây, các điểm đỏ vuông là các điểm được lấy mẫu và sẽ xuất hiện ở trong bức ảnh, còn các điểm đen tròn nằm giữa sẽ không được lấy mẫu và đây chính là các sub-pixel.



Hình 3: Sub-Pixel

3.1.4 Efficient sub-pixel convolution layer

Trong paper này, tác giả giới thiệu một loại layer mới là sub-pixel convolution layer. Layer này gồm 2 bước, bước đầu tiên là convolution thông thường để đưa ra output là $H \times W \times r^2C$, bước còn lại là shuffle lại pixel để cho ra output là $rH \times rW \times C$, đúng với độ phân giải của I_{HR} . Bước pixel shuffle này được thực hiện bằng cách coi mỗi pixel trên r^2 feature map là các sub-pixel, ta sẽ sắp xếp lại chúng theo một thứ tự nhất định ở trên ảnh đầu ra. Hình dưới sẽ minh họa cách sắp xếp lại pixel để sinh ra ảnh output.



Hình 4: Sub-pixel convolution layer

Việc dùng layer này có hai điểm lợi chính:

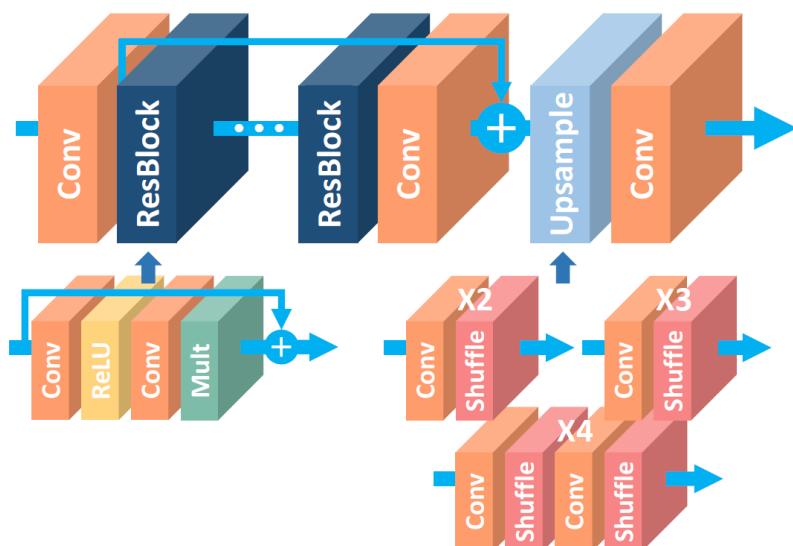
- Giúp ta tránh được việc phải dùng zero-padding làm ảnh hưởng đến kết quả output.
- Tiết kiệm chi phí tính toán, tránh việc thực hiện convolution trong không gian high-resolution.

3.2 EDSR

Nếu như ESPCN tập trung vào việc cải thiện hiệu quả tính toán, thì EDSR lại hướng tới việc tăng cường khả năng biểu diễn của mạng neural network để đạt được chất lượng siêu phân giải cao hơn.

Kiến trúc EDSR dựa trên kiến trúc SRResNet và bao gồm nhiều residual block. Nó sử dụng các lớp tỷ lệ cố định thay cho các lớp batch normalization để tạo ra kết quả nhất quán (input và output có phân phối tương tự, do đó chuẩn hóa các đặc trưng trung gian có thể là không cần thiết). Thay vì sử dụng hàm mất mát L2 (sai số bình phương trung bình), các tác giả đã sử dụng hàm mất mát L1 (sai số tuyệt đối trung bình), cho hiệu quả thực nghiệm tốt hơn.

3.2.1 Kiến trúc



Hình 5: Kiến trúc EDSR

Kiến trúc EDSR chủ yếu bao gồm các residual block. Mỗi block thường bao gồm một chuỗi các lớp convolution, mỗi lớp tiếp theo là một hàm kích hoạt ReLU. Số lượng các khối dư có thể thay đổi tùy thuộc vào việc triển khai cụ thể và độ phức tạp mong muốn của mô hình, nhưng thường thấy các mạng với hàng chục khối block. Để duy trì tính nhất quán của các đặc trưng trong suốt mạng, các lớp tỷ lệ cố định được sử dụng thay cho batch normalization thường thấy. Lựa chọn thiết kế này giúp bảo toàn phân phối của các đặc trưng, dẫn đến quá trình huấn luyện ổn định hơn và tiềm năng hiệu suất tốt hơn. Ở cuối mạng, một lớp convolution được sử dụng để tạo ra hình ảnh độ phân giải cao cuối cùng, thường với số lượng channel bằng với hình ảnh đầu vào. Độ sâu của các lớp convolution này thường được chọn để phù hợp với scale factor mong muốn.

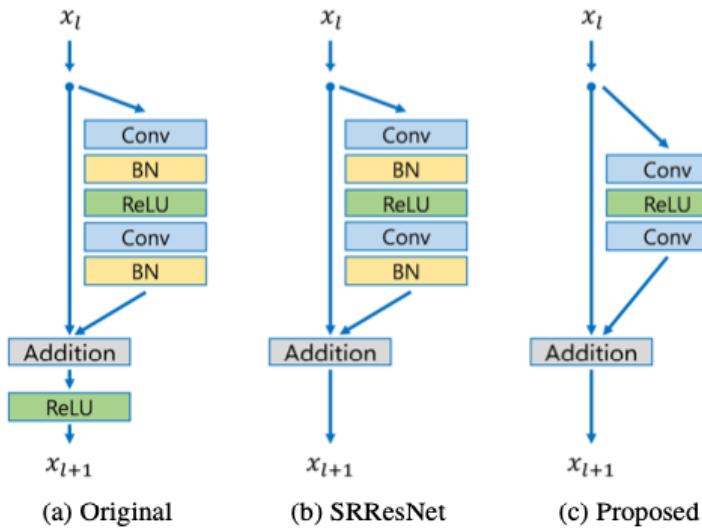
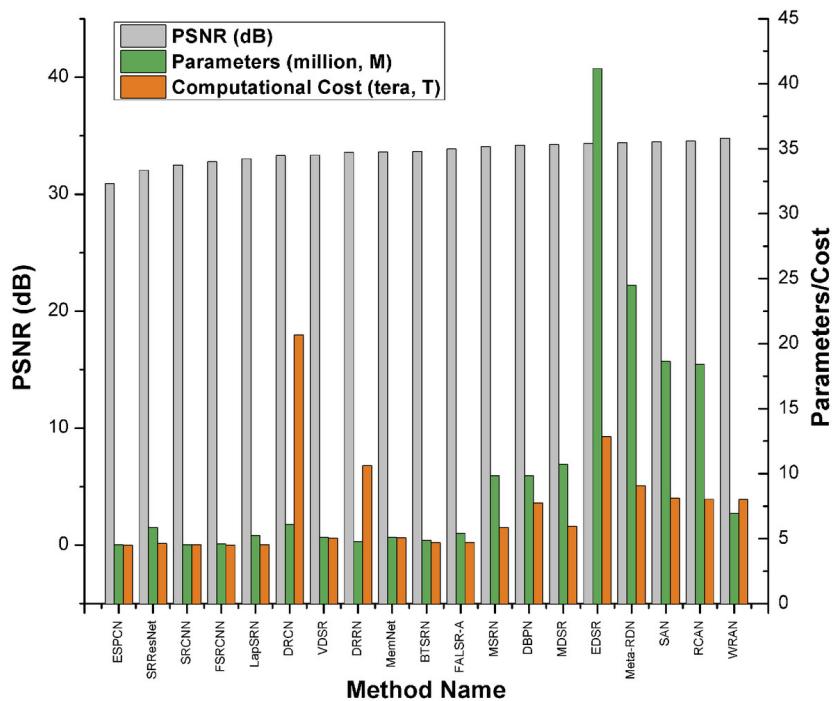


Figure 2: Comparison of residual blocks in original ResNet, SRResNet, and ours.

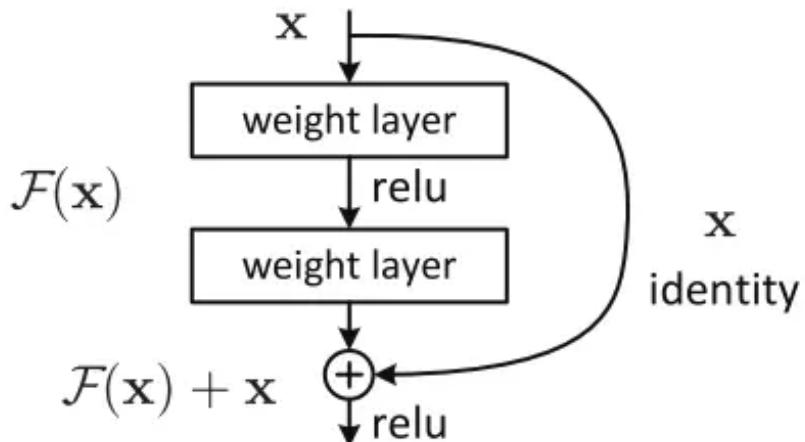
Hình 6: EDSR loại bỏ batch normalization layer của Residual block

Mạng EDSR thông thường sẽ bao gồm 128 filters cho mỗi lớp convolution và gồm 32 residual block. Ngoài ra một số implementation sử dụng số filter là 64 và số residual block là 16, giúp cho mạng được nhỏ gọn hơn.



Hình 7: So sánh EDSR và các mạng khác

3.2.2 Residual block



Hình 8: Skip connection

Residual block được giới thiệu lần đầu tiên ở trong mạng ResNet năm 2015, giúp giải quyết hiện tượng vanishing gradient. Ảnh bên trên hiển thị khối dư được sử dụng trong mạng. Xuất hiện



một mũi tên cong xuất phát từ đầu và kết thúc tại cuối khối dư. Hay nói cách khác là sẽ bổ sung Input X vào đầu ra của layer, hay chính là phép cộng mà ta thấy trong hình minh họa, việc này sẽ chống lại việc đạo hàm bằng 0, do vẫn còn cộng thêm X.

3.3 Real-ESRGAN

Với sự ra đời của các generative model (mô hình tạo sinh), việc tăng cường chi tiết hình ảnh dựa trên các model này rất được kỳ vọng. Một trong số generative model đầu tiên là GAN (generative adversarial network) hay gọi là mạng đối kháng tạo sinh. Đây là một mô hình tạo sinh đặc biệt gồm 2 thành phần chính:

- **Mạng sinh (Generator):** Nhiệm vụ của mạng này là tạo ra dữ liệu mới, giống như một họa sĩ tài ba, cố gắng vẽ nên những bức tranh chân thực nhất có thể. Trong trường hợp siêu phân giải ảnh, mạng sinh sẽ tạo ra những hình ảnh có độ phân giải cao từ những hình ảnh độ phân giải thấp.
- **Mạng phân biệt (Discriminator):** Có nhiệm vụ phân biệt giữa dữ liệu thật và dữ liệu do mạng sinh tạo ra. Mạng phân biệt đánh giá xem ảnh đầu vào là ảnh thật hay ảnh do mạng sinh tạo ra

Quá trình hoạt động của GAN:

1. Generator sinh tạo ra một hình ảnh giả.
2. Discriminator cố gắng phân loại hình ảnh này là thật hay giả.
3. Cập nhật: Cả hai mạng đều được cập nhật dựa trên kết quả phân loại của mạng phân biệt. Generator sẽ cố gắng tạo ra những hình ảnh ngày càng giống thật để đánh lừa discriminator, trong khi discriminator sẽ cố gắng trở nên khắt khe hơn để phân biệt chính xác hơn.

Real-ESRGAN được phát triển dựa trên ESRGAN để giải quyết vấn đề ISR trong thế giới thực, nơi các hình ảnh thường bị suy giảm chất lượng do nhiều yếu tố khác nhau. Những cải tiến chính của Real-ESRGAN so với ESRGAN bao gồm:

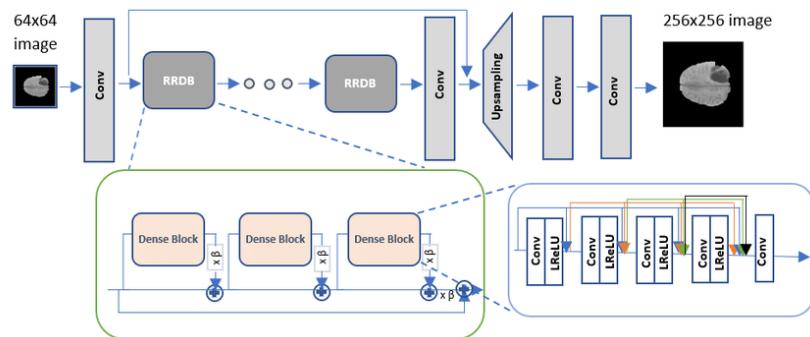
- **Degradation Model:** Sử dụng mô hình degradation "bậc cao" để mô phỏng các degradation phức tạp trong thế giới thực. Thay vì sử dụng một quy trình degradation duy nhất, Real-ESRGAN sử dụng nhiều quy trình degradation lặp đi lặp lại, mỗi quy trình là mô hình degradation cổ điển (blur, downsampling, noise, JPEG compression). Bộ lọc sinc được sử dụng để mô phỏng các hiện tượng nhiễu ảnh thường gặp như ringing và overshoot.
- **Mạng phân biệt (Discriminator):** Sử dụng kiến trúc U-Net thay vì bộ phân biệt VGG trong ESRGAN, điều này tăng khả năng phân biệt của bộ phân biệt, cung cấp phản hồi gradient chính xác hơn để tăng cường chi tiết cục bộ. Spectral normalization (SN) được sử dụng để ổn định quá trình huấn luyện.
- **Dữ liệu huấn luyện:** Real-ESRGAN được huấn luyện bằng dữ liệu tổng hợp hoàn toàn. Các cặp dữ liệu huấn luyện được tạo ra "ngay lập tức" (on the fly) dựa trên bộ ảnh gốc trong quá trình huấn luyện, sử dụng các mô hình degradation phức tạp.
- **Mục tiêu:** Real-ESRGAN có khả năng khôi phục các chi tiết và loại bỏ các nhiễu ảnh khó chịu trong các hình ảnh thực tế. Mô hình này có tính ứng dụng cao hơn so với các phương pháp siêu phân giải trước đó trên các hình ảnh bị suy giảm chất lượng trong thế giới thực.

Tóm lại, ESRGAN tập trung vào việc cải thiện chất lượng thị giác của hình ảnh siêu phân giải thông qua các cải tiến về kiến trúc mạng, hàm mất mát và bộ phân biệt. Trong khi đó, Real-

ESRGAN mở rộng khả năng của ESRGAN để giải quyết các vấn đề siêu phân giải thực tế bằng cách mô phỏng các suy giảm phức tạp và cải thiện mạng phân biệt để đạt được kết quả tốt hơn trên các hình ảnh bị suy giảm chất lượng trong thế giới thực.

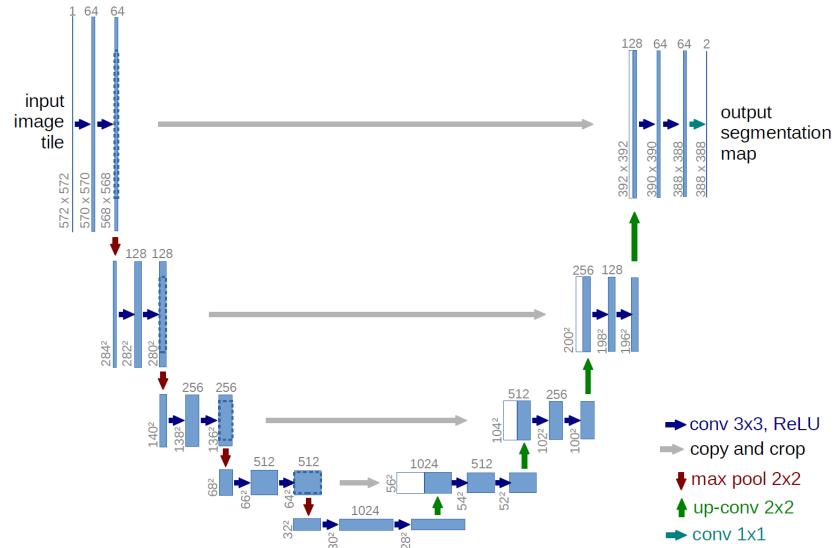
3.3.1 Kiến trúc Real-ESRGAN

ESRGAN giới thiệu RRDB (Residual in Residual Dense Block), là sự kết hợp của multi-level residual network và dense connection (fully connected) và bỏ đi batch normalization. Và cũng tương tự, Real-ESRGAN sử dụng kiến trúc trên cho Generator network. Đối với các hệ số phóng đại (scale factor) là $\times 2$ và $\times 1$, Real-ESRGAN sử dụng thêm thao tác pixel-unshuffle trước khi đưa ảnh vào mạng chính. Thao tác này giúp giảm kích thước không gian và tăng kích thước kênh, do đó giảm đáng kể chi phí tính toán.

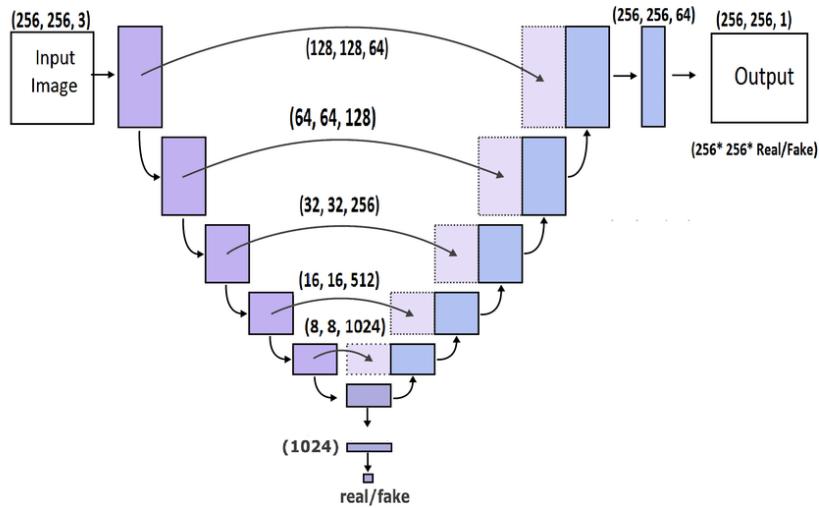


Hình 9: Real-ESRGAN architecture

Bộ phân biệt (Discriminator): Để phân biệt các hình ảnh được tạo ra từ dữ liệu huấn luyện phức tạp, Real-ESRGAN sử dụng bộ phân biệt U-Net thay vì bộ phân biệt VGG như trong ESRGAN. Kiến trúc U-Net có các kết nối tắt (skip connection), cho phép bộ phân biệt cung cấp phản hồi chi tiết ở mức pixel cho bộ tạo. Spectral Normalization (SN) được sử dụng để ổn định quá trình huấn luyện. SN cũng giúp giảm hiện tượng tạo ra các chi tiết quá sắc nét và các nhiễu ảnh khó chịu do quá trình huấn luyện GAN gây ra.



Hình 10: Kiến trúc UNet trong Image Segmentation



Hình 11: Kiến trúc UNet trong GAN

Ngoài ra, Real-ESRGAN cũng cho ra một mô hình degradation, dùng để mô phỏng các image degradation trong thực tế (blur, downsample, noise, jpeg compression, ...). Real-ESRGAN cũng sử dụng bộ lọc sinc để mô phỏng các hiện tượng nhiễu như ringing và overshoot

4 Thí nghiệm

Trong phần thí nghiệm, ta sẽ so sánh 3 mô hình cải tiến, bao gồm ESPCN, EDSR đã được điều chỉnh chỉnh để được train local, và pretrained ESRGAN.

Code có thể tìm thấy tại <https://github.com/chezzijr/image-super-resolution>

4.1 Tập dữ liệu

Tập dữ liệu sẽ là tập Div2K gồm 900 ảnh RGB chất lượng cao, 800 ảnh cho tập training (huấn luyện) và 100 ảnh cho tập validation (đánh giá). Upscale factor r trong bài toán này sẽ được chọn là 4, tức là $H_{out} = rH_{in} = 4H_{in}$ và $W_{out} = rW_{in} = 4W_{in}$. Các hình ảnh sẽ được thông qua tiền xử lý để đảm bảo rằng chiều cao và chiều rộng là bội của r .

Data Augmentation (Tăng cường dữ liệu): tạo ra những dữ liệu mới dựa trên tập dữ liệu hiện có, giúp cải thiện độ bao quát của tập dữ liệu. Các hình ảnh gốc sẽ được thông qua biến đổi để tạo ra hình ảnh mới, bao gồm:

- Ngẫu nhiên lật hình ảnh theo trục ngang hoặc dọc
- Ngẫu nhiên xoay hình ảnh 90 hoặc -90 độ
- Ngẫu nhiên cắt một phần của hình ảnh để làm input. Chiều dài và chiều rộng mới không được nhỏ hơn nửa chiều dài và chiều rộng ảnh gốc và phải là bội của r

4.2 Huấn luyện

Từng mô hình (trừ ESRGAN) sẽ được huấn luyện trên tập dữ liệu được tăng cường. Hình ảnh ban đầu sẽ được scale xuống, trong trường hợp này là scale xuống 4 lần (width và height giảm đi 4). Ảnh được scale sẽ là input của network, và output sẽ được so sánh với ảnh gốc ban đầu.

Mô hình sẽ được train trong 50 epochs, batch size là 1 (vì hạn chế của môi trường tính toán). Dùng early patience là 5 cho validation loss, tức là nếu sau 5 epoch validation loss không cải thiện sẽ ngừng huấn luyện, giúp giảm thời gian huấn luyện.

Metric để đánh giá mô hình là PSNR. PSNR được định nghĩa thông qua sai số toàn phương trung bình (MSE – Mean squared error). 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 biệt giữa các ước lượng và những gì đánh giá. Ở đây MSE được xác định cho ảnh hai chiều có kích thước mxn trong đó I và K là ảnh gốc và ảnh sau khi tổng hợp.

$$\text{MSE} = \frac{1}{m * n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2$$
$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}_I^2}{\text{MSE}} \right) = 20 \cdot \log_{10} \left(\frac{\text{MAX}_I^2}{\sqrt{\text{MSE}}} \right)$$

Hình 12: PSNR

Ở đây MAX_i là giá trị tối đa của pixel trên ảnh. Khi các pixels được biểu diễn bởi 8 bits, thì giá trị của nó là 255. Trường hợp tổng quát khi tín hiệu được biểu diễn bởi B bit trên một đơn vị mẫu MAX_i là 2^{B-1} .

Thông thường nếu $\text{PSNR} \geq 40$ dB thì hệ thống mắt thường gần như không phân biệt được ảnh gốc và ảnh sinh ra. PSNR càng cao thì chất lượng ảnh sinh càng tốt, khi 2 ảnh giống hệt nhau thì $\text{MSE}=0$ và PSNR đi đến vô hạn, đơn vị của PSNR là Decibel.

Môi trường huấn luyện:

- OS: Arch Linux (kernel 6.10.2-arch1-2)
- GPU: GeForce RTX 3050 Mobile 4GB
- CUDA: 12.5
- Driver: nvidia-dkms 555.58.02

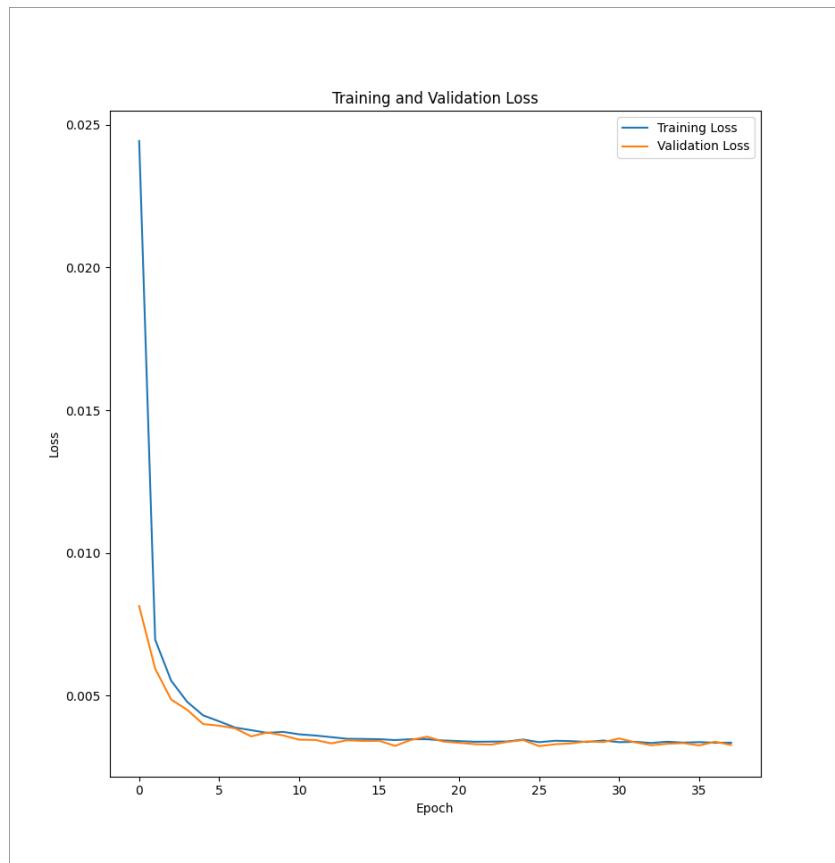
4.3 ESPCN

4.3.1 Thông số

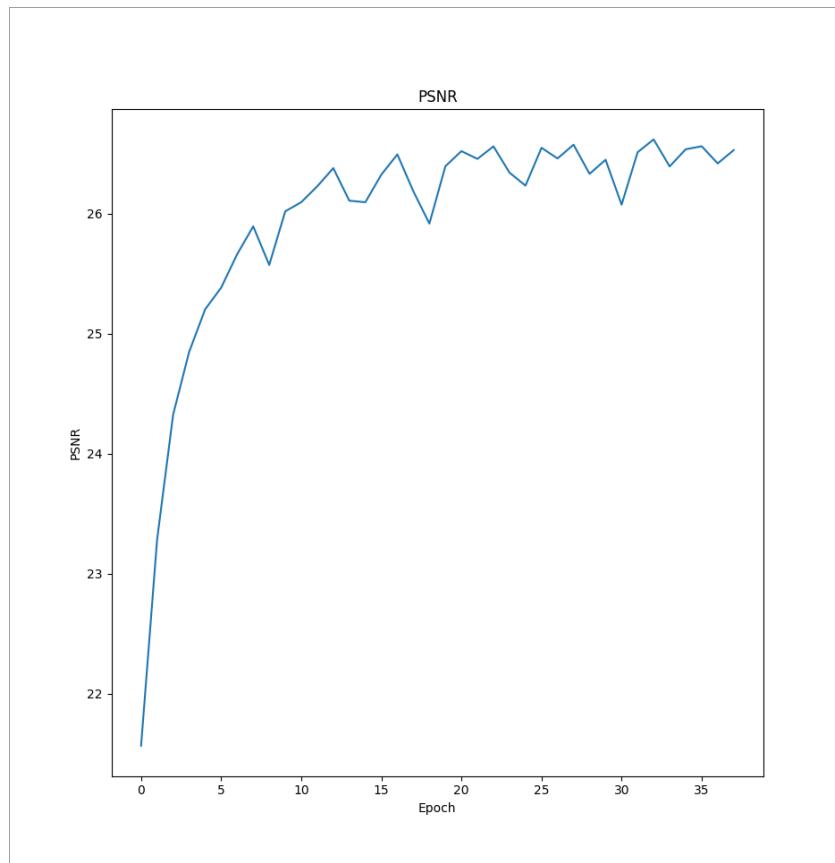
- Số tham số: 37,200
- Các layers:
 1. Conv2d(3, 64, kernel size=(5, 5), stride=(1, 1), padding=(2, 2))
 2. Tanh()
 3. Conv2d(64, 32, kernel size=(3, 3), stride=(1, 1), padding=(1, 1))
 4. Tanh()
 5. Conv2d(32, 48, kernel size=(3, 3), stride=(1, 1), padding=(1, 1))
 6. PixelShuffle(upscale factor=4)
- Loss function: Mean Squared Error
- Optimizer: Adam ($learningrate = 10^{-4}$)
- Tốc độ huấn luyện: 9 batches (images) / giây
- Thời gian huấn luyện: khoảng 1 giờ
- VRAM sử dụng: 1.4 GB

4.3.2 Kết quả

- Training loss: 0.0033
- Validation loss: 0.0032
- PSNR: 26.5291



Hình 13: ESPCN Training and Validation Loss



Hình 14: ESPCN PSNR

4.4 EDSR (được điều chỉnh)

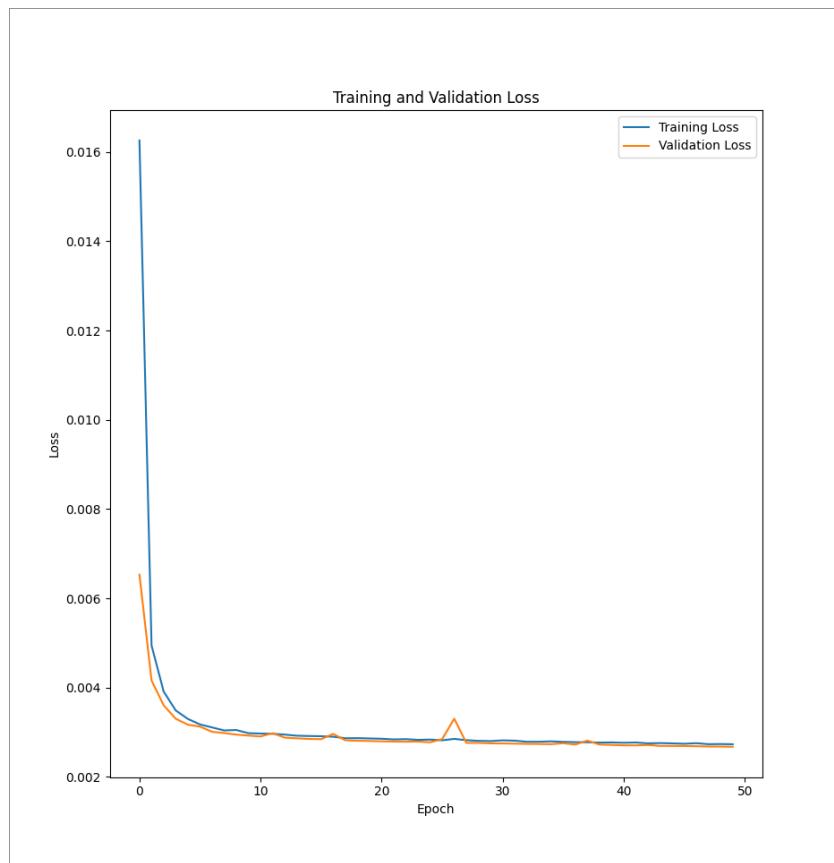
4.4.1 Thông số

- Số tham số: 56,563
- Các layers (sử dụng 4 residual blocks với 16 filters để có thể train):
 1. Conv2d(3, 16, kernel size=(3, 3), stride=(1, 1), padding=(1, 1))
 2. x4 ResidualBlock((block): Sequential((0): Conv2d(16, 16, kernel size=(3, 3), stride=(1, 1), padding=(1, 1)) (1): ReLU(inplace=True) (2): Conv2d(16, 16, kernel size=(3, 3), stride=(1, 1), padding=(1, 1))))
 3. Conv2d(16, 256, kernel size=(3, 3), stride=(1, 1), padding=(1, 1))
 4. PixelShuffle(upscale factor=4)
 5. Conv2d(16, 3, kernel size=(3, 3), stride=(1, 1), padding=(1, 1))
- Loss function: Mean Squared Error (Sử dụng MSE thay vì MAE để hội tụ nhanh hơn vì môi trường tính toán bị giới hạn)

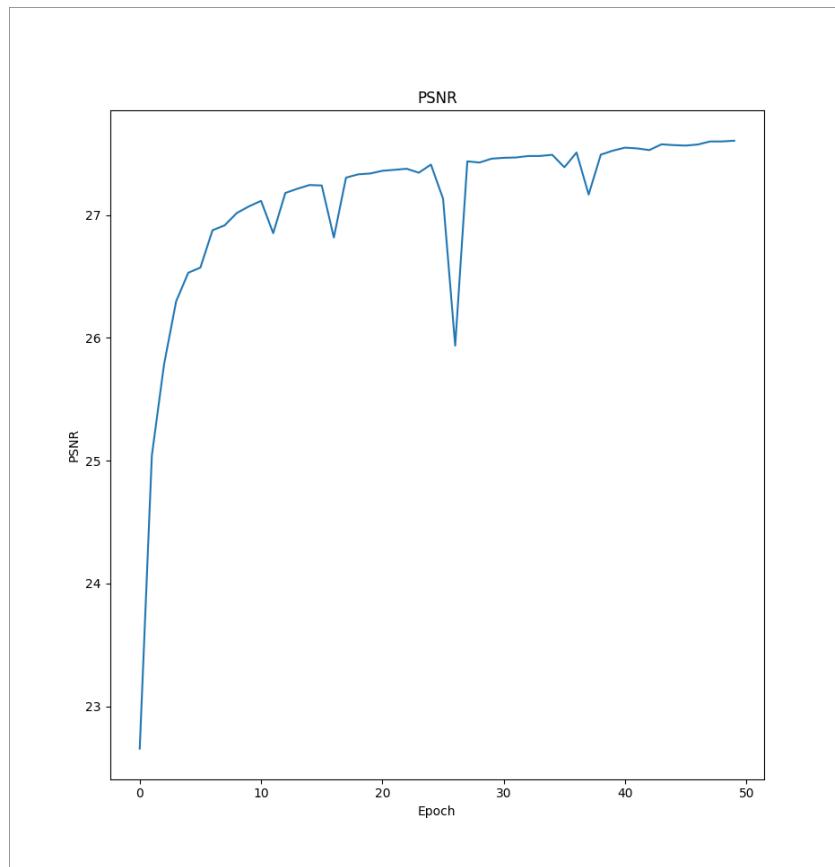
- Optimizer: Adam ($learningrate = 10^{-4}$)
- Tốc độ huấn luyện: 8 batches (images) / giây
- Thời gian huấn luyện: khoảng 2 giờ
- VRAM sử dụng: khoảng 3 GB

4.4.2 Kết quả

- Training loss: 0.0027
- Validation loss: 0.0026
- PSNR: 27.6051



Hình 15: EDSR Training and Validation Loss



Hình 16: EDSR PSNR

4.5 Real-ESRGAN (pretrained)

4.5.1 Thông số

- Số tham số: 16.7M
- Các layers của generator:
 1. Conv2d(3, 64, kernel size=(3, 3), stride=(1, 1), padding=(1, 1))
 2. x23 RRDB, mỗi block bao gồm x4 ResidualDenseBlock, mỗi ResidualDenseBlock sẽ kết hợp của input ban đầu và output của các cặp layer trước dưới dạng $x_n = \text{LeakyReLU}(\text{Conv2d}(\sum_{i=0}^{n-1} x_i))$ với x_0 là input ban đầu, output sẽ là $\text{Conv2d}(x_4) \times 0.2 + x_0$
 3. Conv2d(64, 64, kernel size=(3, 3), stride=(1, 1), padding=(1, 1))
 4. x2 Upsample Block
 5. Conv2d(64, 64, kernel size=(3, 3), stride=(1, 1), padding=(1, 1))
 6. Conv2d(64, 3, kernel size=(1, 1), stride=(1, 1))

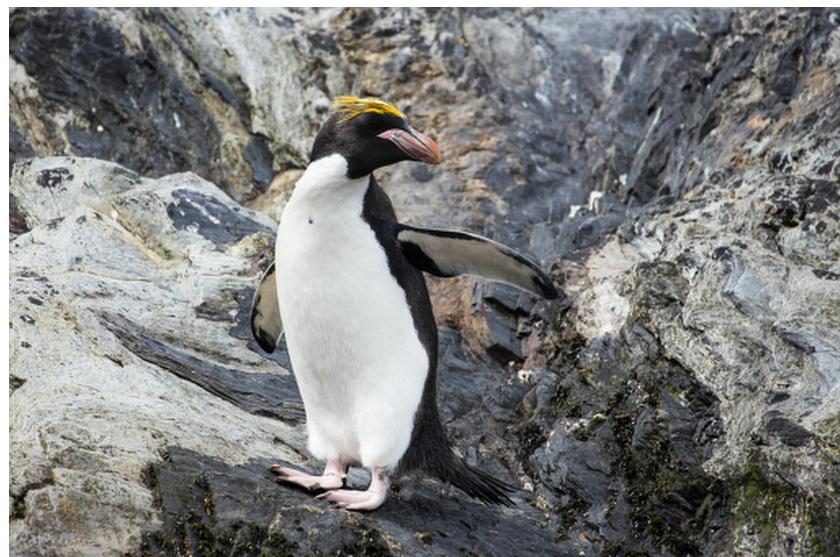
7. LeakyReLU(negative slope=0.2, inplace=True)
- Các layers của discriminator: tương tự U Net
- Loss function cho Generator:
 - Content loss: L1
 - Perceptual loss: khoảng cách euclidean giữa đặc trưng của output và target, trích xuất bởi VGG19 và được tính toán trước activation
 - GAN loss: sự khác biệt giữa bộ phân biệt dự đoán độ chân thực của ảnh thật và ảnh giả
- Loss function cho Discriminator: dùng Relativistic average Discriminator (RaD)
- Optimizer: Adam

4.6 So sánh

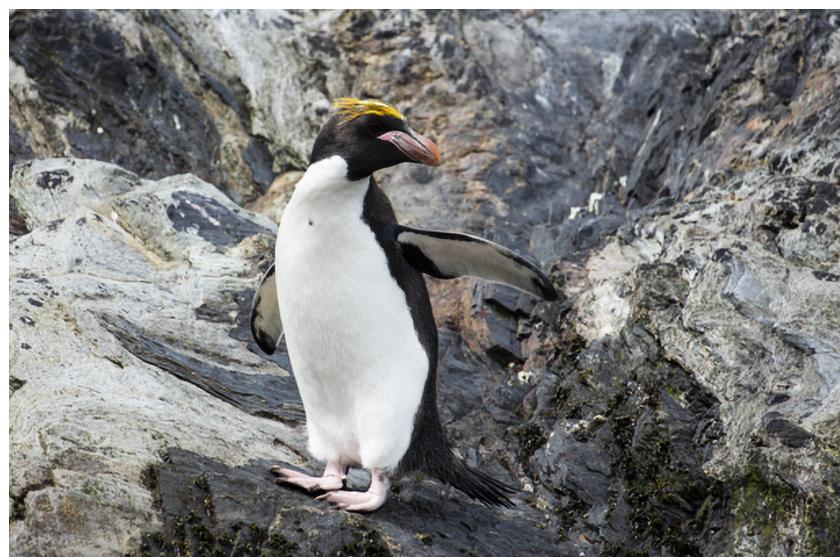
Ta sẽ so sánh giữa những model cải tiến. Trước hết ta sẽ dùng dữ liệu trong tập dữ liệu ban đầu để làm mẫu:



Hình 17: Low resolution image



Hình 18: ESPCN



Hình 19: EDSR



Hình 20: Real-ESRGAN

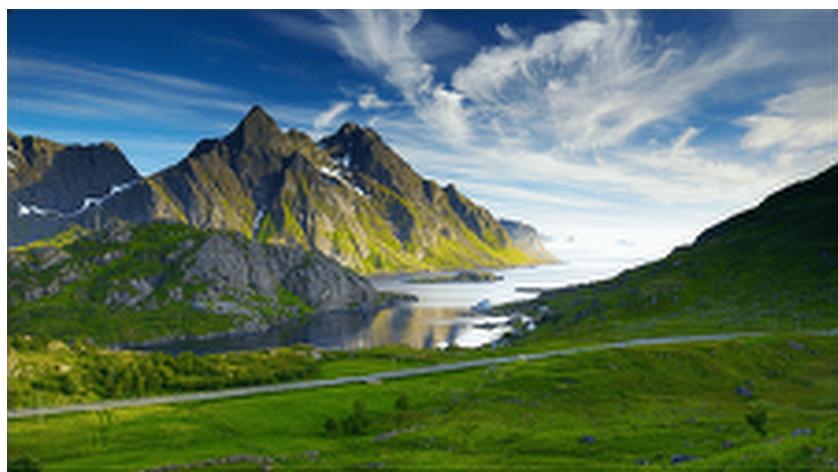


Hình 21: Ground truth

Sau đây là dữ liệu thực tế



Hình 22: Low resolution image



Hình 23: ESPCN



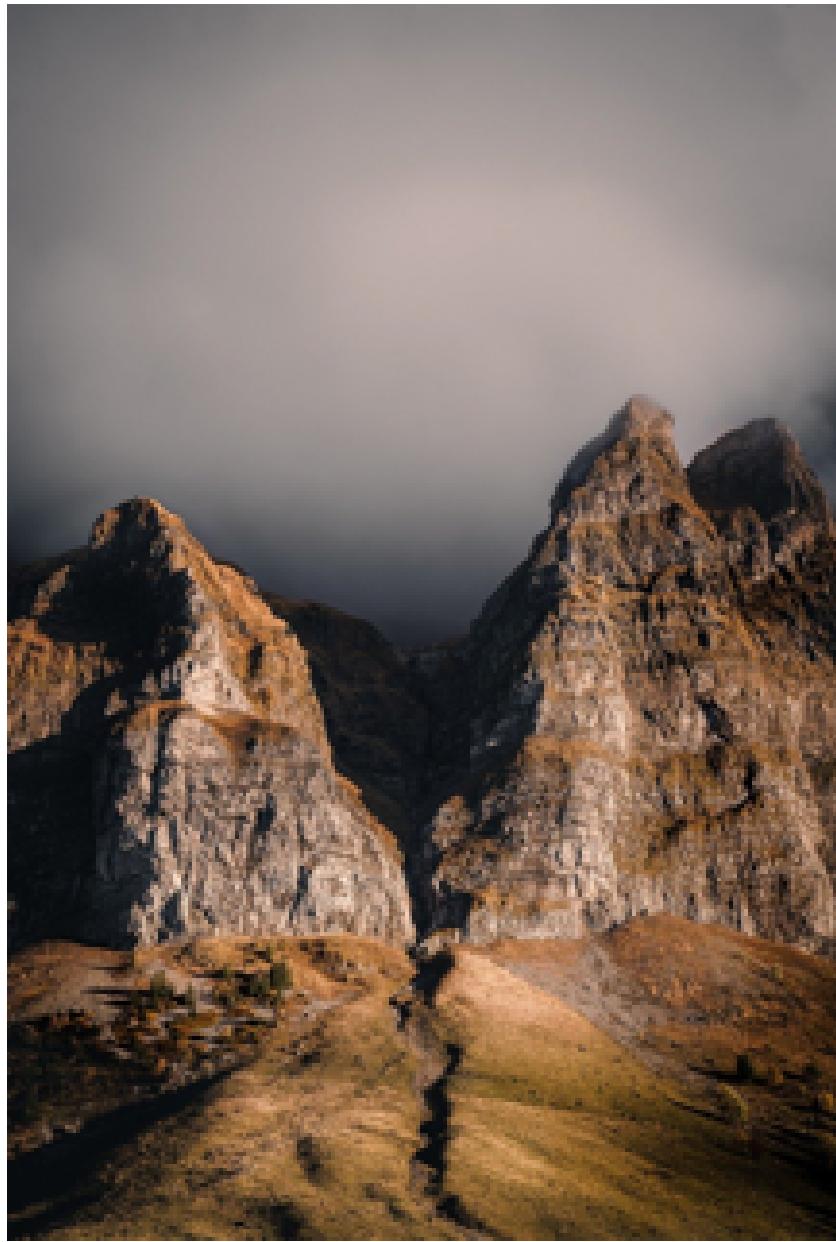
Hình 24: EDSR



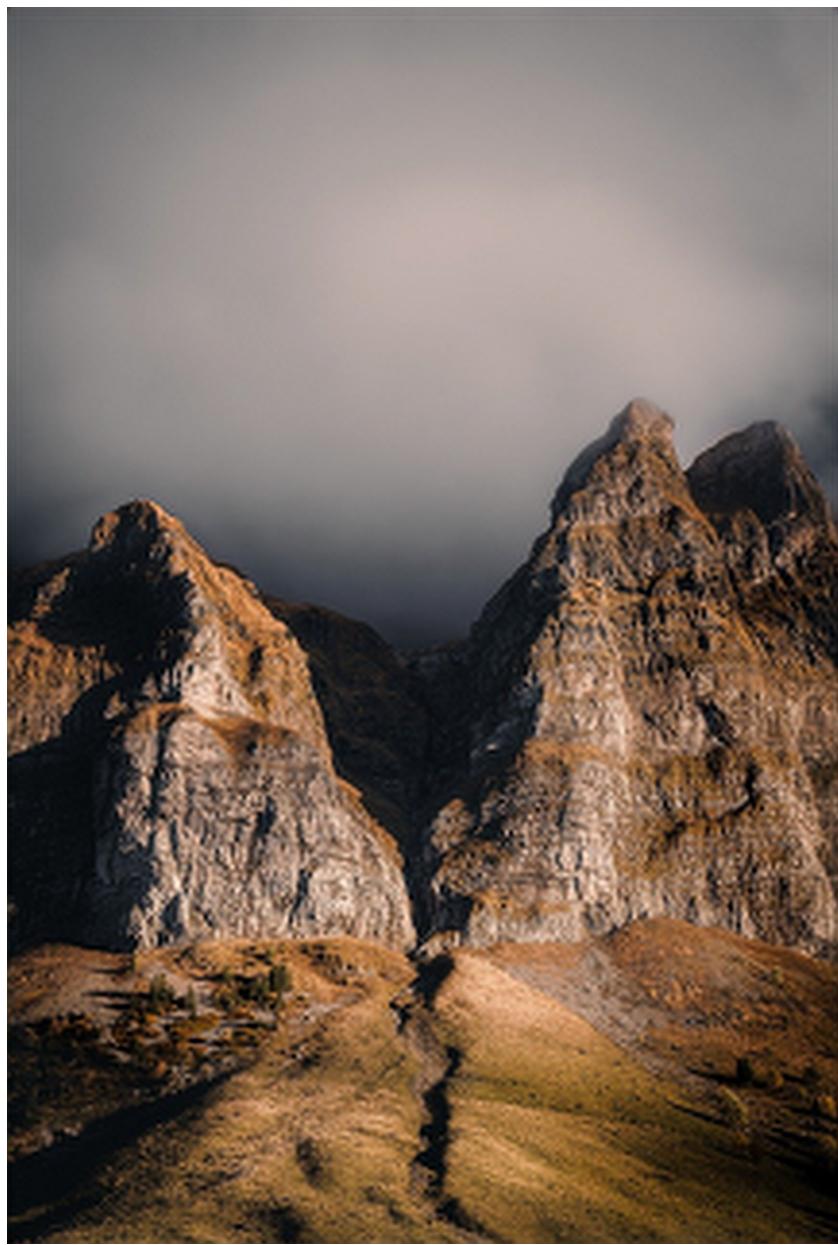
Hình 25: Real-ESRGAN



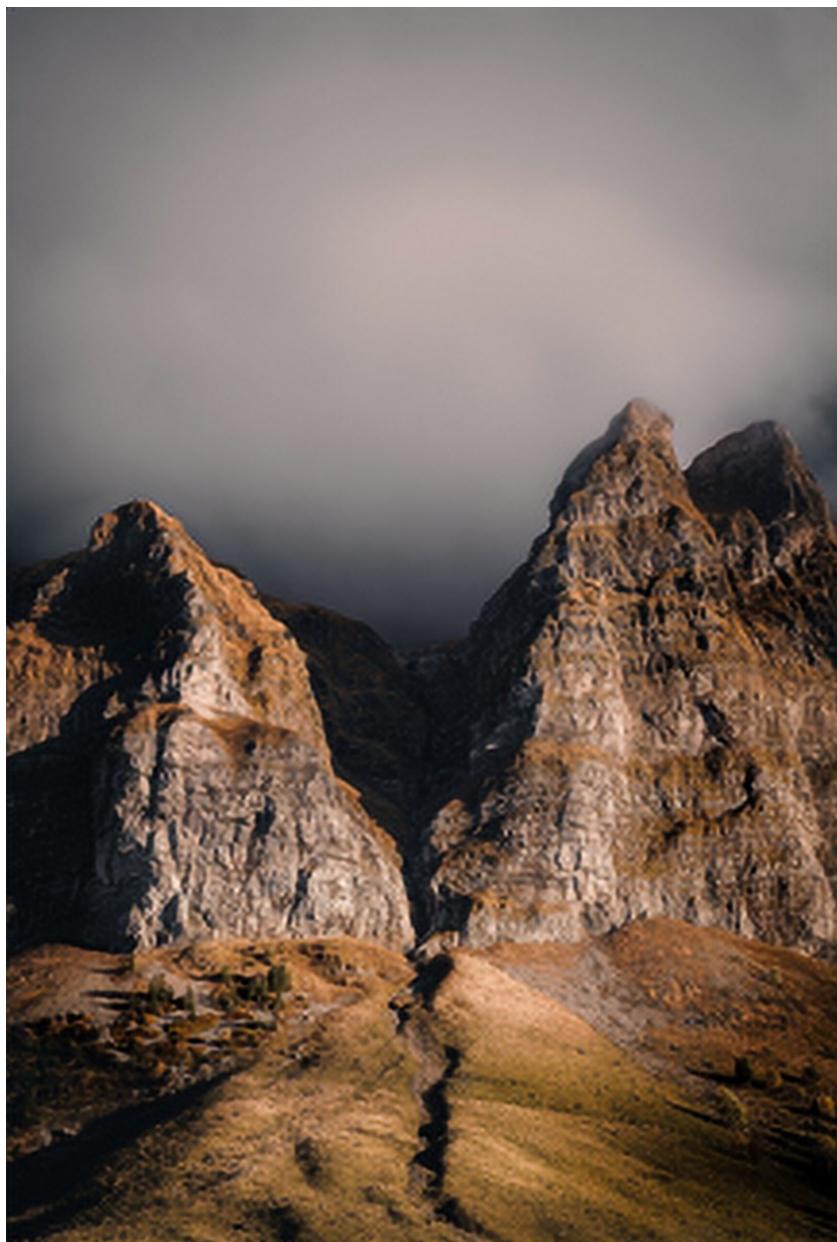
Hình 26: Ground truth



Hình 27: Low resolution image



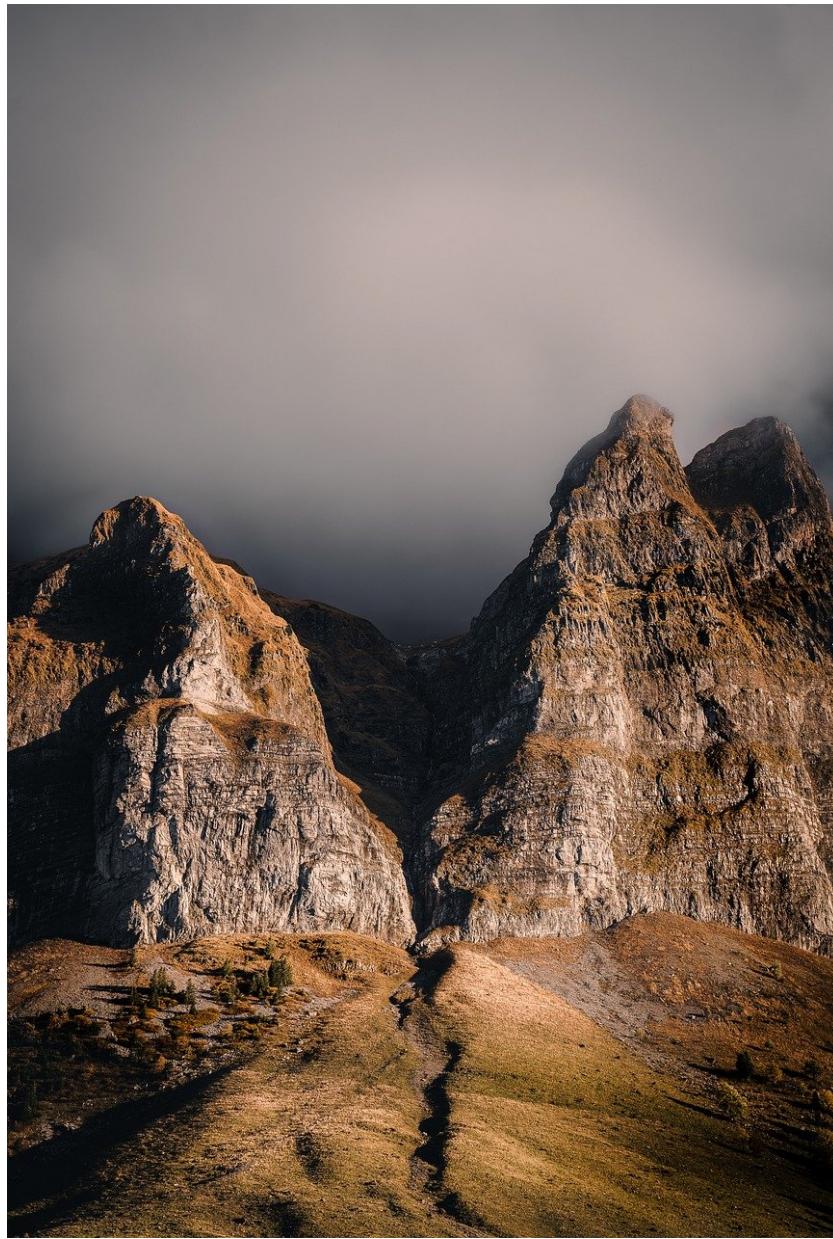
Hình 28: ESPCN



Hình 29: EDSR



Hình 30: Real-ESRGAN



Hình 31: Ground truth

4.7 Kết quả

Nhìn chung, cả 3 mô hình đều làm khá tốt trong ISR, tuy vậy vẫn có một vài vấn đề với các mô hình. Ta có thể liệt kê các ưu và nhược điểm của các mô hình như sau:

4.7.1 ESPCN

- Ưu điểm: nhanh, nhỏ gọn, dễ huấn luyện, huấn luyện nhanh



- Nhược điểm: mô hình không đủ phức tạp nên cho ra kết quả không bằng hai mô hình kia

4.7.2 EDSR (đã được điều chỉnh)

- Ưu điểm: cho ra kết quả tốt hơn ESPCN với số tham số không quá lớn hơn so với ESPCN, mô hình gốc với số lượng tham số lớn hơn sẽ cho ra kết quả tốt hơn nữa
- Nhược điểm: thời gian huấn luyện lâu (đối với mô hình gốc thì thời gian huấn luyện còn lâu hơn nữa) nhưng kết quả cải thiện không đến mức là đột phá

4.7.3 Real-ESRGAN (pretrained)

- Ưu điểm: sinh ra các chi tiết liền mạch (phù hợp cho digital art), khiến cho ảnh được thực sự rõ nét hơn hai mô hình còn lại.
- Nhược điểm: thời gian huấn luyện lâu nhất, số lượng tham số nhiều nhất, thời gian infer lâu nhất, các chi tiết sinh ra có thể quá mượt không phù hợp với một số loại ảnh, hoặc sinh ra các chi tiết không mong muốn.

5 Tổng kết

Chúng ta đã thảo luận qua 3 mô hình cải tiến để giải quyết bài toán ISR. Hiện nay, mô hình được sử dụng rộng rãi là Real-ESRGAN, thường được sử dụng trong việc tăng cường chân dung, phong cảnh hay ảnh vẽ số (digital art) với những chi tiết rất rõ ràng. Tuy vậy, với những hạn chế trên, việc hiện thực inference liên tục sẽ tốn nhiều thời gian và tài nguyên. Ví dụ, việc tăng cường video (gồm rất nhiều hình ảnh ghép lại với nhau) sử dụng Real-ESRGAN sẽ rất tốn thời gian, không những thế những chi tiết được thêm vào có thể sẽ không được liên tục và mượt mà do Real-ESRGAN chỉ hoạt động trên một bức ảnh.

Khi đó, các mô hình như ESPCN sẽ được sử dụng. Với thời gian inference nhanh, tuy không tăng cường chi tiết được như Real-ESRGAN, ESPCN cũng đã có thể tăng cường đáng kể được chất lượng mà video cũng được mượt mà hơn. Đặc biệt, trong việc cải thiện những video cũ, hạn chế trong việc tăng cường chi tiết của ESPCN có thể là một điểm cộng do giữ lại nét cổ kính. Ngoài ra, với ưu điểm tốc độ inference nhanh, ESPCN có thể sử dụng để tăng cường video streaming thời gian thực, đồng thời với độ nhỏ gọn của mô hình, ESPCN có thể được deploy trên client thay vì server, phục vụ việc tăng cường hình ảnh, video hoặc streaming ngay tại máy người dùng.

Đối với EDSR, tuy cho ra kết quả tốt hơn so với ESPCN, nhưng hiệu quả không đáng kể so với chi phí phải bỏ ra để train (xem hình 7). Việc này khiến cho finetuning model trở nên khó khăn hơn nhưng lại không đạt được hiệu quả rõ rệt như Real-ESRGAN. Hơn nữa, số lượng tham số lớn cũng khiến cho việc inference trở nên chậm hơn, khiến cho mô hình này khó được ứng dụng trong các hệ thống lớn, video hoặc streaming.

Mở rộng: Ngày nay, với sự bùng nổ của model diffusion, ta có thể ứng dụng công nghệ này vào bài toán ISR. Diffusion là một generative model ra đời sau và thừa hưởng các tiền bối như Auto-Encoder (AE), GAN và Flow-based model. Ta có thể dựa trên Stable Diffusion đã làm mưa làm gió giới công nghệ năm 2022, bằng cách prompt ảnh và phong cách mình muốn tăng cường ảnh, ta có thể được hình ảnh như mong muốn và có thể có chất lượng tốt hơn việc sử dụng Real-ESRGAN.



Tài liệu tham khảo

- [1] Wikipedia, *Neutral network (machine learning)*,
[https://en.wikipedia.org/wiki/Neural_network_\(machine_learning\)](https://en.wikipedia.org/wiki/Neural_network_(machine_learning))
- [2] Paper, *Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network* <https://arxiv.org/abs/1609.05158>
- [3] Paper, *Enhanced Deep Residual Networks for Single Image Super-Resolution* <https://arxiv.org/abs/1707.02921>
- [4] Paper, *Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data* <https://arxiv.org/abs/2107.10833>