

Image Super-Resolution

Tuan Phan, Tung Pham

AI For Everyone

Ngày 14 tháng 3 năm 2022

Outline

1. Bài toán image super-resolution

2. Image Interpolation

3. CNN for Super-Resolution

3.1 SRCNN

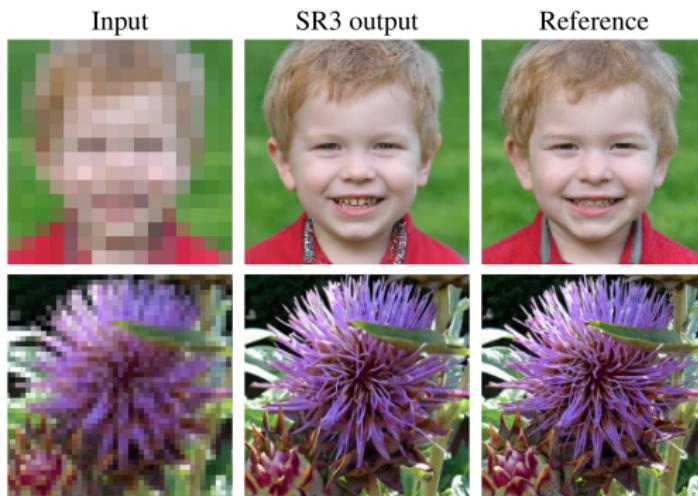
3.2 ResNet Super Resolution

3.3 Autoencoder Super Resolution

4. Tổng kết

Bài toán image super-resolution

Cho ảnh X là một ảnh high-resolution (HR), Y là ảnh low-resolution (LR) tương ứng của X . Gọi hàm dùng để upscale ảnh LR thành HR là F . Bài toán image super-resolution hướng đến việc tìm F sao cho $F(Y) \approx X$. Hay ảnh upscale từ Y giống X nhất có thể



Hình 1: Ví dụ về bài toán image super-resolution

Mục tiêu project

Upscale ảnh với độ phân giải gấp đôi và tăng chi tiết của ảnh



Hình 2: Upscale ảnh (169, 255, 3) thành (338, 510, 3)

Dataset

Bộ dữ liệu sử dụng: DIV2K

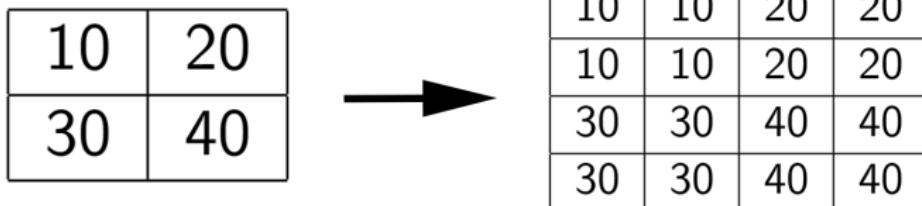
DIV2K bao gồm 1000 ảnh 2K trong đó:

- 800 ảnh cho training
- 100 ảnh cho validation
- 100 ảnh cho testing

Image Interpolation

Image Interpolation

- Nearest-neighbor interpolation
- Bilinear interpolation
- Bicubic interpolation
-



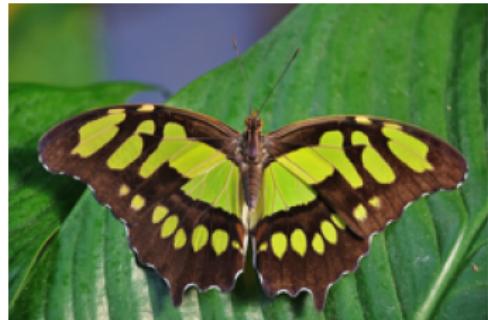
Hình 3: Nearest-neighbor interpolation

Image Interpolation



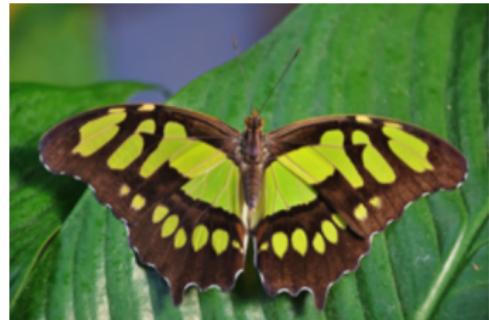
Hình 4: Ảnh high-resolution và low-resolution

Image Interpolation



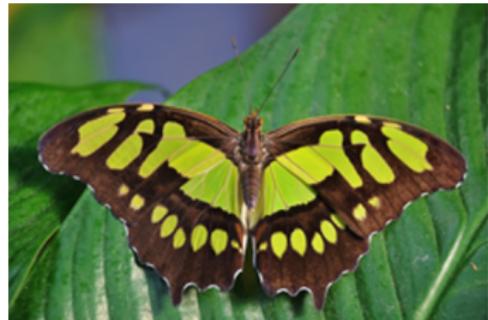
Hình 5: Ảnh high-resolution và Nearest-neighbor interpolation

Image Interpolation



Hình 6: Ảnh high-resolution và Bilinear interpolation

Image Interpolation

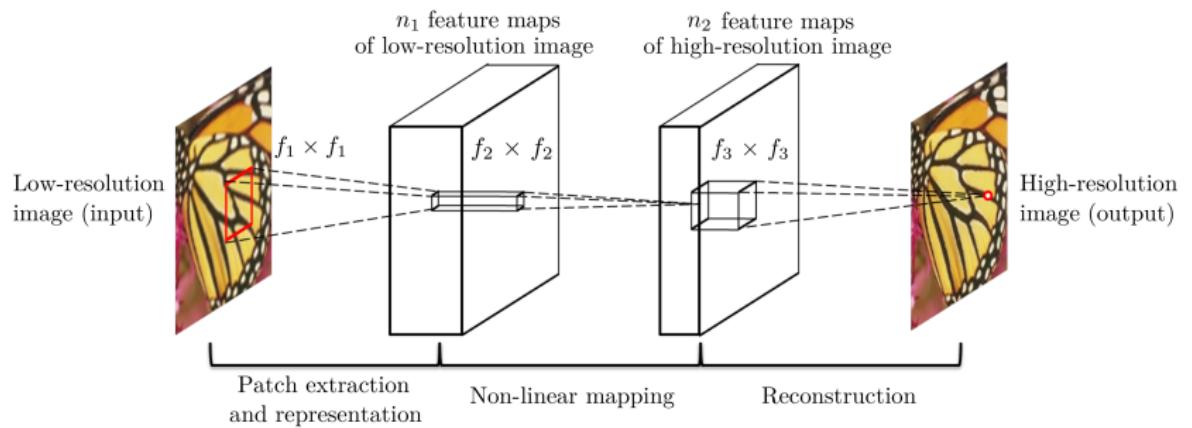


Hình 7: Ảnh high-resolution và Bicubic interpolation

CNN for Super-Resolution

SRCNN

Kiến trúc mạng



Hình 8: Kiến trúc mạng SRCNN

Training

Loss function:

$$L(\Theta) = \frac{1}{n} \sum_{i=1}^n \|F(\mathbf{Y}_i; \Theta) - \mathbf{x}_i\|^2$$

Với n là số sample được dùng để train

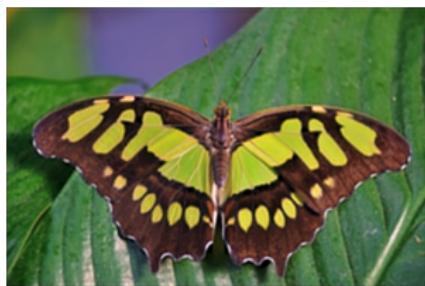
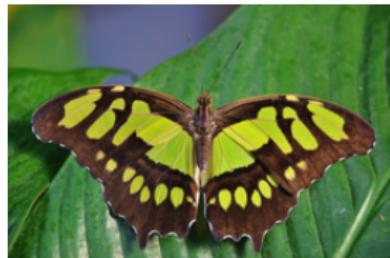
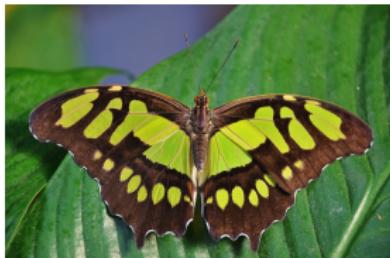
Kiến trúc SRCNN

Block	Đầu vào	Phép toán	Số channel
X_1	(Input)	Interpolation	3
X_2	AxBx3	Conv2D	64
X_3	AxBx64	Batch Normalization	
X_4	AxBx64	Conv2D	32
X_5	AxBx32	Batch Normalization	
X_6	AxBx32	Conv2D	3

Bảng 1: Kiến trúc SRCNN

Tổng tham số: 20,483

Kết quả



Hình 9: Ảnh high-resolution, low-resolution và output SRCNN

ResNet Super Resolution

Residual block

Block	Đầu vào	Phép toán	Số channel
X_0	(Input)		
X_1	AxBx3	Conv2D	64
X_2	AxBx64	Batch Normalization	
X_3	AxBx64	Conv2D	64
X_4	AxBx64	Batch Normalization	
X_5	AxBx64	Add[X_0, X_4]	

Bảng 2: Residual block

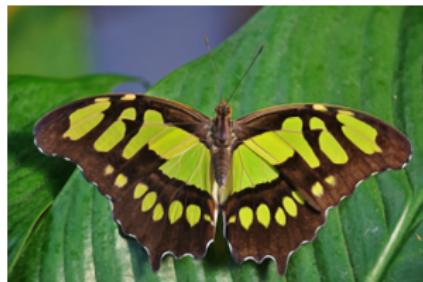
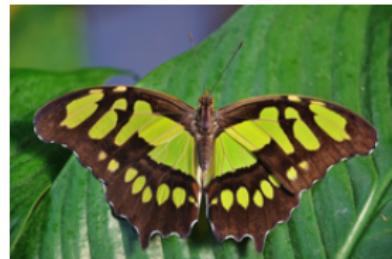
Kiến trúc SRResNet

Block	Đầu vào	Phép toán	Số channel
X_1	(Input)	Interpolation	3
X_2	AxBx3	Conv2D	64
X_3	AxBx64	residual_block	64
X_4	AxBx64	residual_block	64
X_5	AxBx64	residual_block	64
X_6	AxBx64	residual_block	64
X_7	AxBx64	residual_block	64
X_8	AxBx64	residual_block	64
X_9	AxBx64	Add[X_2, X_8]	
X_{10}	AxBx64	Conv2D	3

Bảng 3: Kiến trúc SRResNet

Kết quả

Tổng tham số: 449,731



Hình 10: Ảnh high-resolution, low-resolution và output SRResNet

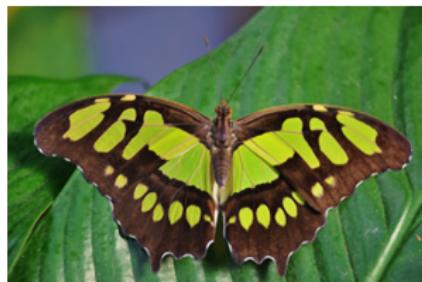
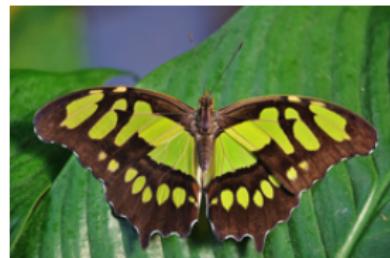
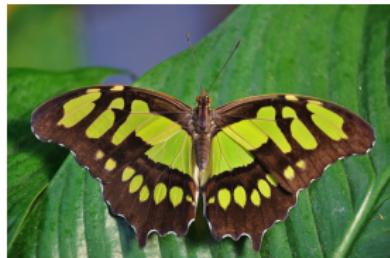
Kiến trúc SRResNet(lite)

Block	Đầu vào	Phép toán	Số channel
X_1	(Input)	Interpolation	3
X_2	AxBx3	Conv2D	32
X_3	AxBx32	Conv2D	32
X_4	AxBx32	Batch Normalization	
X_5	AxBx32	Conv2D	32
X_6	AxBx32	Batch Normalization	
X_7	AxBx32	Add[X_6, X_2]	
X_8	AxBx32	Add[X_7, X_2]	
X_9	AxBx32	Conv2D	64
X_3	AxBx64	Conv2D	3

Bảng 4: Kiến trúc SRResNet(lite)

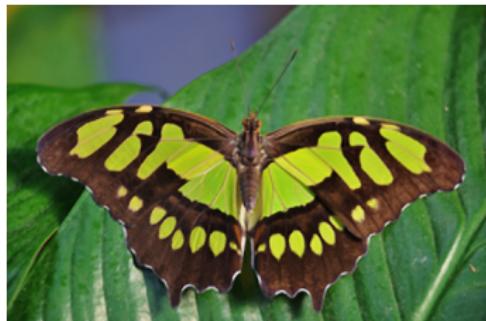
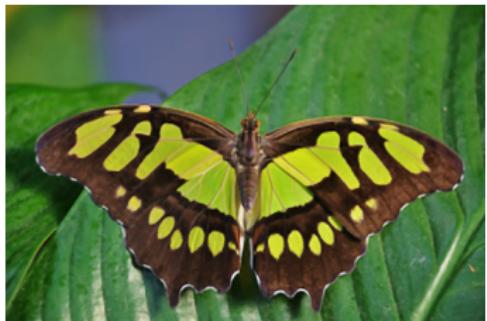
Kết quả

Tổng tham số: 39,875



Hình 11: Ảnh high-resolution, low-resolution và output SRResNet(lite)

SRResNet và SRResNet(lite)



Hình 12: Ảnh SRResnet và SRResNet(lite)

Autoencoder Super Resolution

Kiến trúc SRAutoencoder

Block	Đầu vào	Phép toán	Số channel
X_1	(Input)	Interpolation	3
X_2	AxBx3	Conv2D	64
X_3	AxBx64	Conv2D	64
X_4	AxBx64	MaxPooling2D	
X_5	$(A/2) \times (B/2) \times 64$	Conv2D	128
X_6	$(A/2) \times (B/2) \times 128$	Conv2D	128
X_7	$(A/2) \times (B/2) \times 128$	MaxPooling2D	
encoder	$(A/4) \times (B/4) \times 128$	Conv2D	256

Bảng 5: Kiến trúc encoder

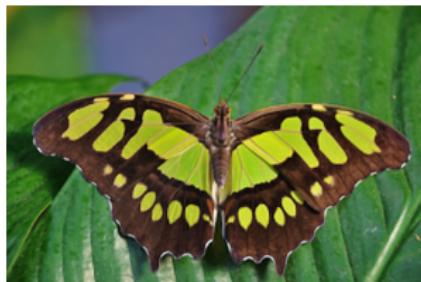
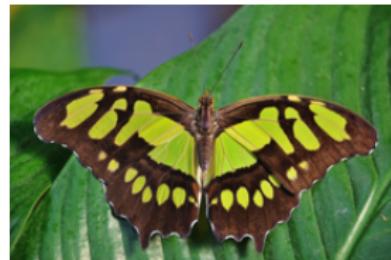
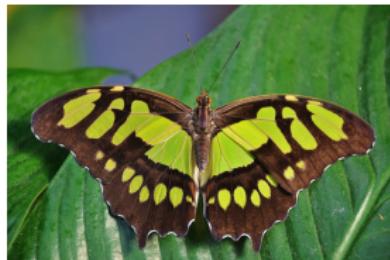
Kiến trúc SRAutoencoder

Block	Đầu vào	Phép toán	Số channel
X_8	$(A/4) \times (B/4) \times 256$	Conv2DTranspose	256
X_9	$(A/2) \times (B/2) \times 256$	Conv2D	128
X_{10}	$(A/2) \times (B/2) \times 128$	Conv2D	128
X_{11}	$(A/2) \times (B/2) \times 128$	Add[X_6, X_{10}]	
X_{12}	$(A/2) \times (B/2) \times 128$	Conv2DTranspose	128
X_{13}	AxBx128	Conv2D	64
X_{14}	AxBx64	Conv2D	64
X_{15}	AxBx64	Add[X_3, X_{14}]	
decoder	AxBx64	Conv2D	3

Bảng 6: Kiến trúc decoder

Kết quả

Tổng tham số: 1,848,067



Hình 13: Ảnh high-resolution, low-resolution và output Autoencoder

Tổng kết

Tổng kết

Peak signal-to-noise ratio (PSNR): dùng để tính tỉ lệ giữa giá trị năng lượng tối đa của một tín hiệu và năng lượng nhiễu ảnh hướng đến độ chính xác của thông tin

PSNR được sử dụng để đo chất lượng tín hiệu khôi phục của các thuật toán nén có mất mát dữ liệu

Công thức:

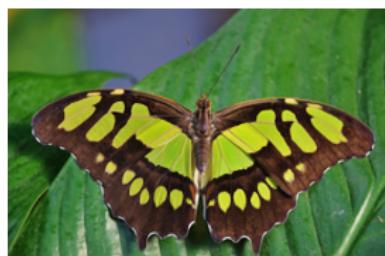
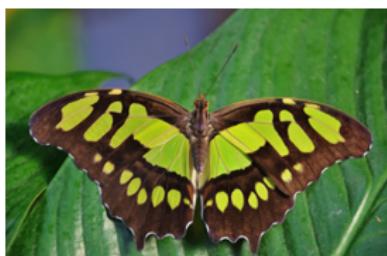
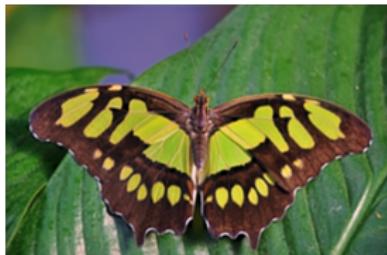
$$PSNR = 10 * \log_{10} \left(\frac{MAX^2}{MSE} \right)$$

Tổng kết

Kết quả đo được từ tập validation

Tên	Parameters	Loss (MSE)	PSNR	Thời gian
SRCNN	20,483	581.2148	21.2076	0.155355s
SRResnet(lite)	39,875	397.9567	23.1751	0.193009s
SRResnet	449,731	396.3909	23.3171	0.352079s
Autoencoder	1,848,067	382.0012	23.4283	2.352550s

Tổng kết



Hình 14: SRCNN, SRResNet, SRResNet(lite) và SRAutoencoder

Một số ảnh khác



Hình 15: Ảnh high-resolution và low-resolution

Một số ảnh khác



Hình 16: SRCNN, SRResNet, SRResNet(lite) và Autoencoder

Một số ảnh khác



Hình 17: Ảnh high-resolution và low-resolution

Một số ảnh khác



Kết thúc