



Seminar cuối kỳ

Selective Frequency Network for Image Restoration

Yung Cui et al. (ICLR2023)

Giảng viên lý thuyết:

TS. Nguyễn Ngọc Thảo

Team ID: KHMT08 – Team name: Nhóm 8

24C11051 Lưu Thiện Đức

24C11061 Phạm Phú Hoàng Sơn

24C11067 Nguyễn Anh Tuấn

24C11071 Phạm Quốc Vương

Nội dung

1. Thông tin bài báo
2. Abstract & Introduction
3. Methodology
4. Mức 1: Pre-trained model + Dữ liệu từ paper
5. Mức 2: Pre-trained model + Dữ liệu ngoài paper
6. Mức 3: Dữ liệu ngoài paper + Fine-tuning

1. Thông tin bài báo

Tên bài báo: Selective Frequency

Network for Image Restoration (SFNet)

Tác giả: Yuning Cui, Yi Tao, Zhenshan
Bing, Wenqi Ren, Xinwei Gao, Xiaochun
Cao, Kai Huang, Alois Knoll

Hội nghị: ICLR 2023

Lĩnh vực: Image Restoration
(thuộc nhóm bài toán Image-to-Image
Translation)

Link GitHub: github.com/c-yn/SFNet

Từ khóa chính: Frequency Domain,
Deep Learning, Image Restoration

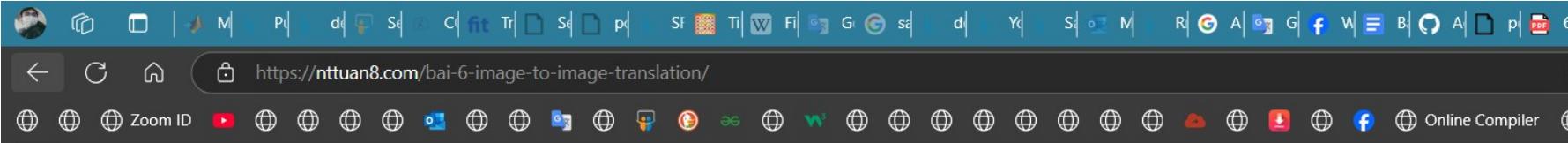
The figure illustrates the SFNet architecture and its performance across various image restoration tasks. It features three main sections: Motivation, Method, and Experimental Results.

Motivation: Compares Spatial Domain Network (using Attention), Spectral Domain Network (using FFT), and the proposed Selective Frequency Network (using Frequency Decomposition and Modulation).

Method: Details the Local Adaptive Selective Frequency Module (using Frequency Separation, GAP, Conv, BN, Softmax, Invert, and Concat) and the Global Efficient Selective Frequency Module (using Frequency Separation, Input Feature Pipeline, Frequency Modulation, Windows Partition, Windows Reverse, and Concat).

Experimental Results: Shows results for Image Dehazing, Image Motion, Deblurring, Image Defocus Deblurring, and Image Desnowing. It includes plots of Variance vs Iterations, Density vs Variance difference, and bar charts comparing methods like MAXIM, FSDGNet, and Ours across datasets like SOTS-Outdoor, CSD, and SRRS.

Below the diagram are two blurred images of cars, likely demonstrating the restoration quality of SFNet on real-world datasets.

A screenshot of a web browser window. The address bar shows the URL <https://nttuan8.com/bai-6-image-to-image-translation/>. The page content is in Vietnamese and discusses image-to-image translation, specifically GAN-based methods for mapping between different image styles.

Bài toán image to image translation

Image to image translation là bài toán rất phổ biến trong GAN với mục tiêu là học được mapping giữa ảnh input và ảnh output.

Thường thì input và output sẽ là ảnh cùng kích thước nhưng thay đổi các thuộc tính trong ảnh, ví dụ: không màu -> có màu (ứng dụng tô màu), ảnh thẻ -> ảnh anime, ảnh mờ -> ảnh nét (làm nét ảnh), ảnh màu -> sang các style khác như Monet, Van gogh, etc (style transfer).

A 3x5 grid of images demonstrating image-to-image translation. The first row shows various objects and scenes: wooden sticks, colorful balls, hands holding cameras, a landscape, and a forest. The second row shows a child with a white bird, a boy with a white swan, a ladybug on a leaf, a fish, and coral. The third row shows two roosters, a dog, a butterfly, and a flower.

Input ảnh xám, output ảnh màu

Two pairs of images illustrating style transfer. The top pair shows a boat on a lake as input and output. The bottom pair shows a landscape with a bridge as input and output. The outputs have a painterly, Impressionistic style.

Input ảnh thường. output: làm nét ảnh

2. Abstract & Introduction - Bài toán

Khôi phục hình ảnh: là quá trình tái tạo hoặc cải thiện chất lượng hình ảnh bị hư hỏng, nhiễu hoặc mất thông tin dựa trên các kỹ thuật xử lý ảnh và học máy.

Bao gồm 5 tác vụ chính:



Defocus image

Restored image

Ground truth

Single-image
defocus
deblurring

2. Abstract & Introduction - Bài toán

Khôi phục hình ảnh: là quá trình tái tạo hoặc cải thiện chất lượng hình ảnh bị hư hỏng, nhiễu hoặc mất thông tin dựa trên các kỹ thuật xử lý ảnh và học máy.

Bao gồm 5 tác vụ chính:

Image
dehazing



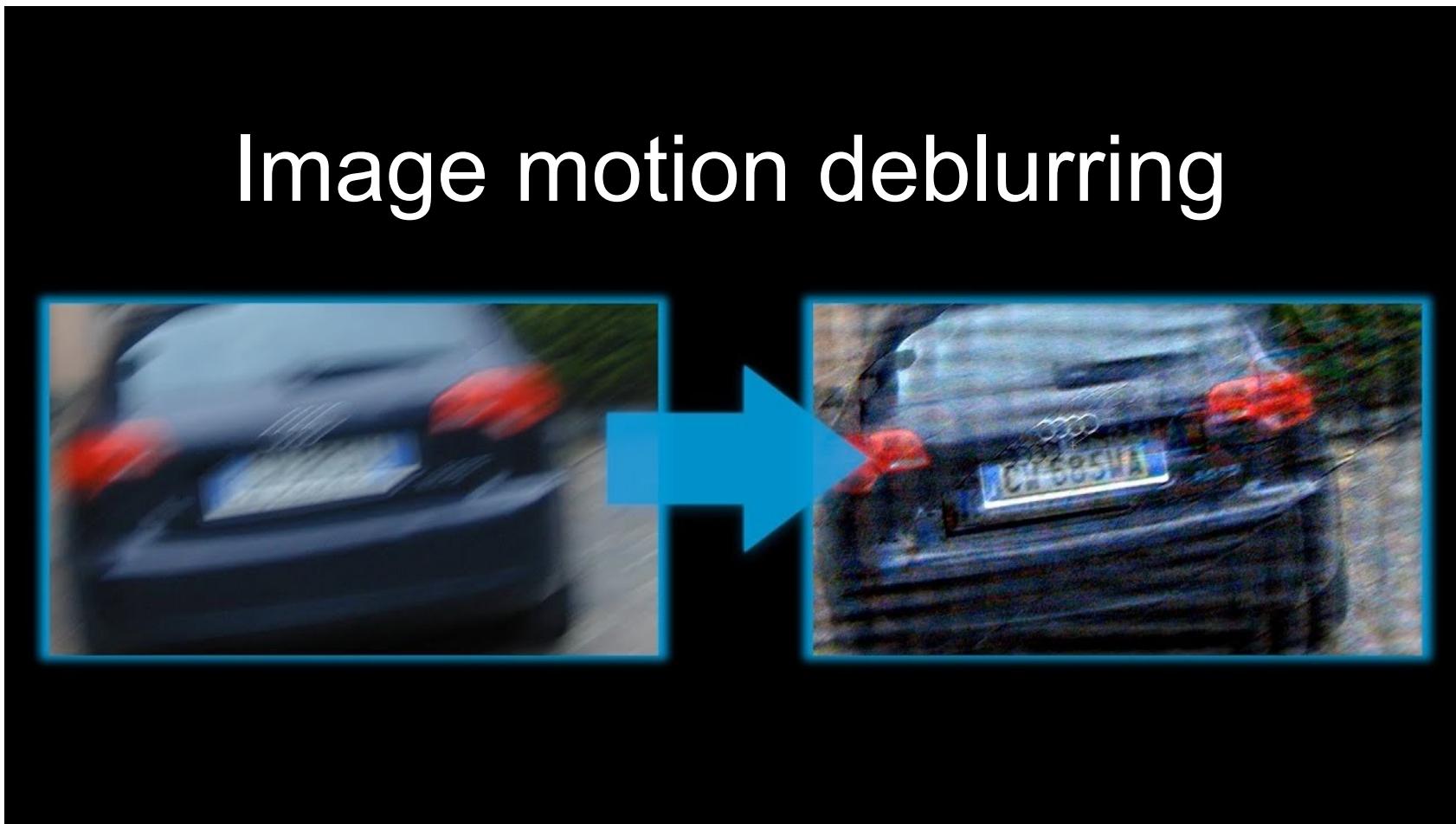
(a)

(b)

2. Abstract & Introduction - Bài toán

Khôi phục hình ảnh: là quá trình tái tạo hoặc cải thiện chất lượng hình ảnh bị hư hỏng, nhiễu hoặc mất thông tin dựa trên các kỹ thuật xử lý ảnh và học máy.

Bao gồm 5 tác vụ chính:



2. Abstract & Introduction - Bài toán

Khôi phục hình ảnh: là quá trình tái tạo hoặc cải thiện chất lượng hình ảnh bị hư hỏng, nhiễu hoặc mất thông tin dựa trên các kỹ thuật xử lý ảnh và học máy.

Bao gồm 5 tác vụ chính:



Image
desnowing

2. Abstract & Introduction - Bài toán

Khôi phục hình ảnh: là quá trình tái tạo hoặc cải thiện chất lượng hình ảnh bị hư hỏng, nhiễu hoặc mất thông tin dựa trên các kỹ thuật xử lý ảnh và học máy.

Bao gồm 5 tác vụ chính:

Image
deraining

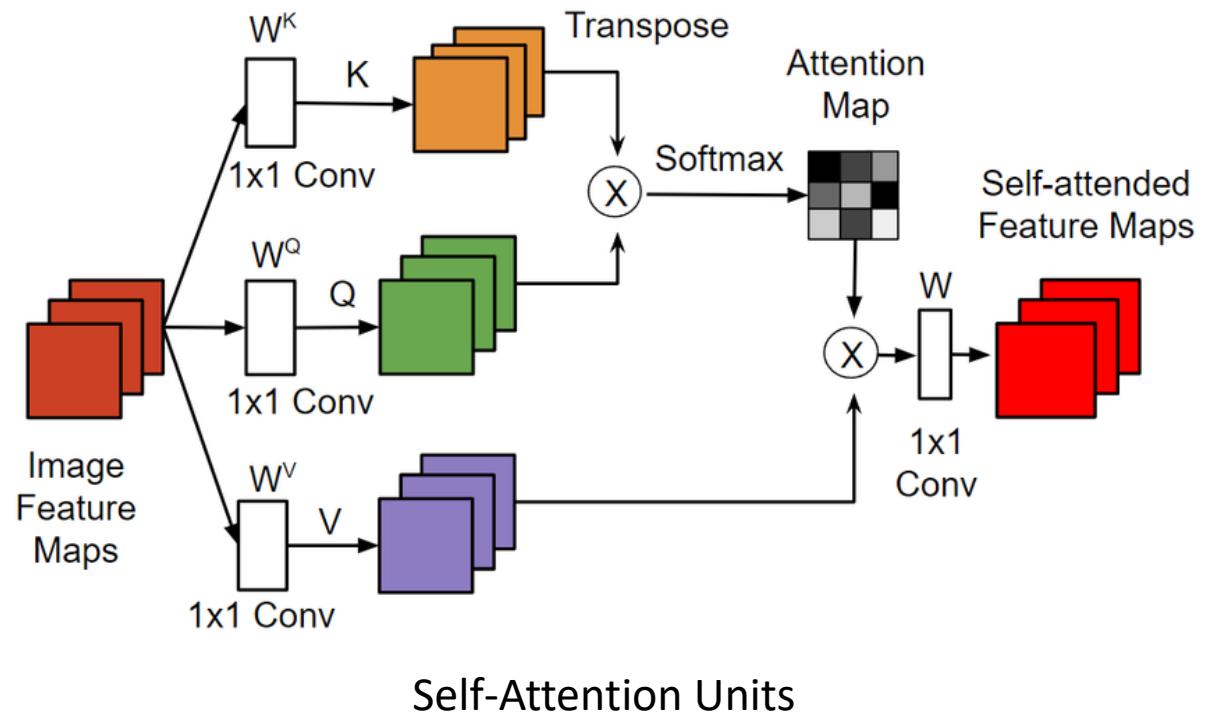
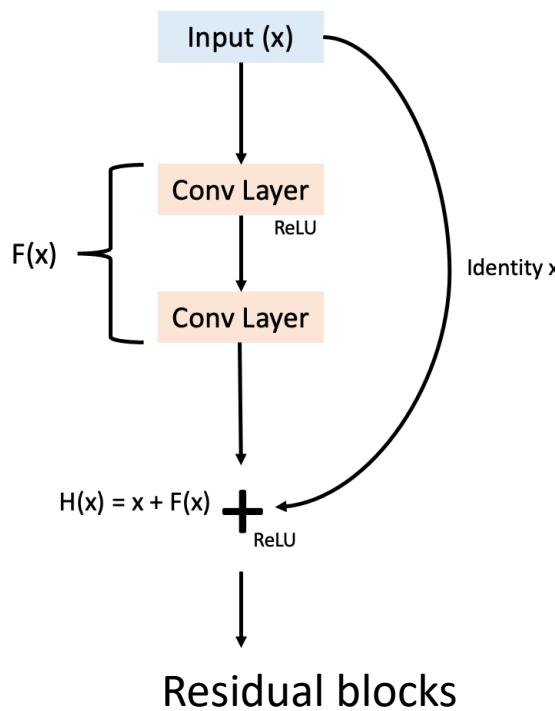


2. Abstract & Introduction - Các vấn đề đặt ra

Trước đây, hầu hết các mô hình deep learning thực hiện phục hồi ảnh trực tiếp trong miền không gian (spatial domain), sử dụng các thành phần như:

- Residual Blocks
- Self-Attention Units
- Various Spatial Attention Modules

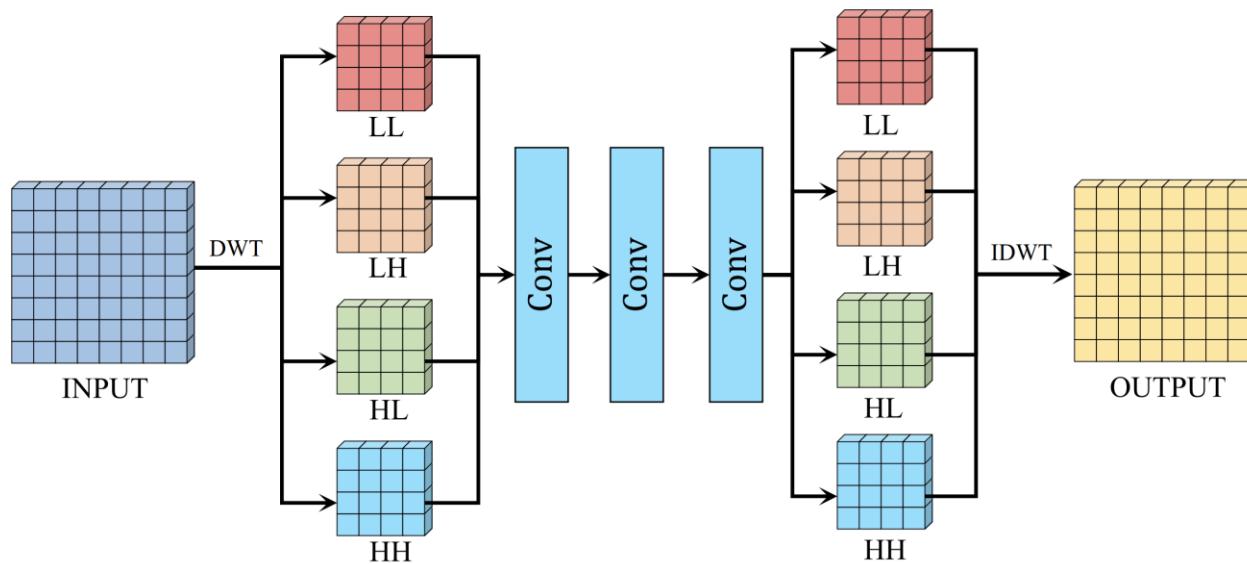
-> Chưa tận dụng hiệu quả sự khác biệt trong miền tần số giữa ảnh sắc nét và ảnh bị suy giảm, dẫn đến hạn chế trong việc phục hồi chi tiết ảnh.



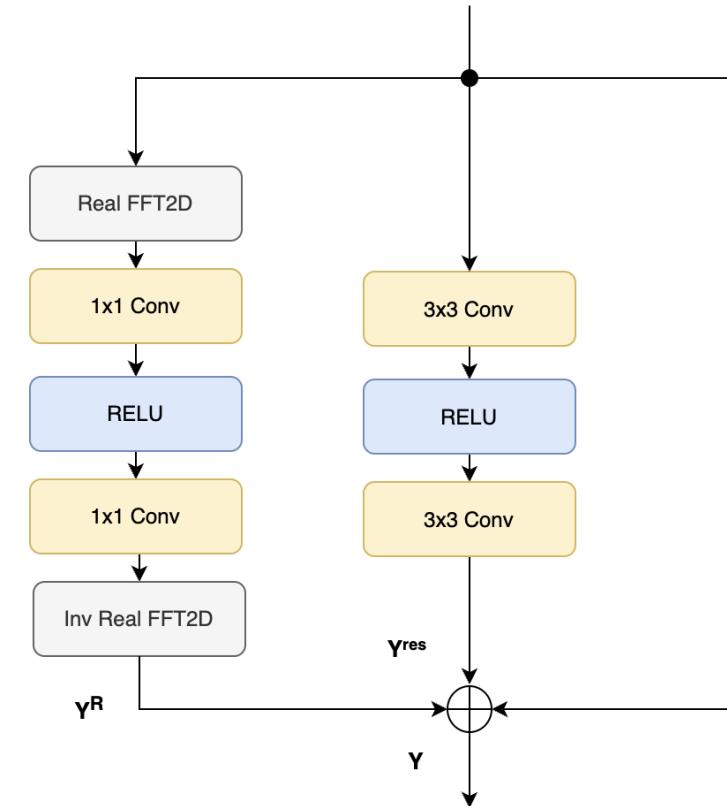
2. Abstract & Introduction - Các vấn đề đặt ra

Một số nghiên cứu sử dụng biến đổi wavelet/Fourier để phân tách và xử lý ảnh theo tần số (frequency):

- Wavelet transform
- Fourier transform



Wavelet transform

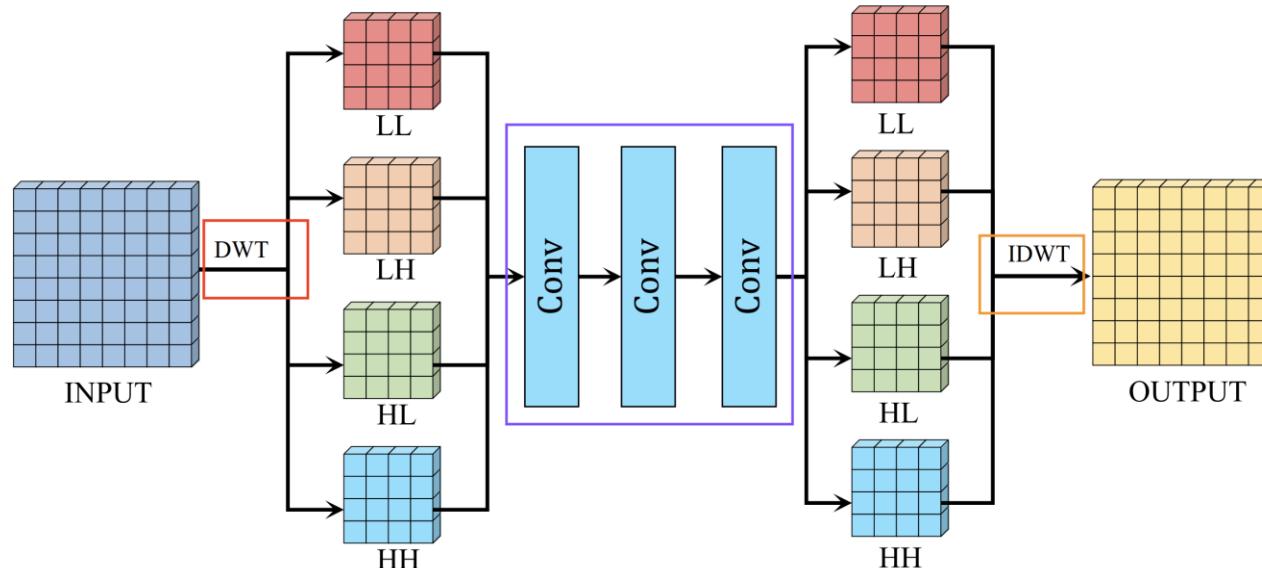


Fourier transform

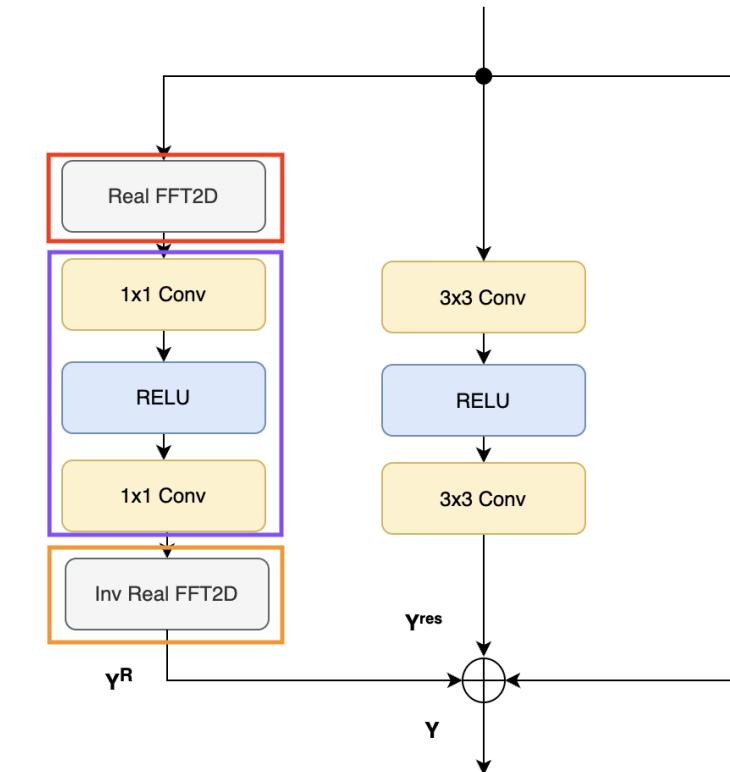
2. Abstract & Introduction - Các vấn đề đặt ra

Quy trình của các phương pháp biến đổi tần số

- **Phân tách tần số (Frequency Decomposition)** -> Không thể chọn lọc chính xác các tín hiệu tần số quan trọng để khôi phục
- **Xử lý đặc trưng (Feature Processing - CNN)** -> Chi phí tính toán cao
- **Phép biến đổi ngược (Inverse operation - IFFT, IDWT)**



Wavelet transform



Fourier transform

2. Abstract & Introduction - Mục tiêu đề ra

Mục tiêu đặt :

- Tập trung vào bước phân tách tần số để cải thiện hiệu suất khôi phục: Linh hoạt hơn trong việc chọn lọc và nhấn mạnh các tần số quan trọng.
- Giúp hiệu quả hơn, giảm chi phí tính toán của các bước xử lý và biến đổi ngược.



2. Abstract & Introduction - Các phương pháp đề xuất

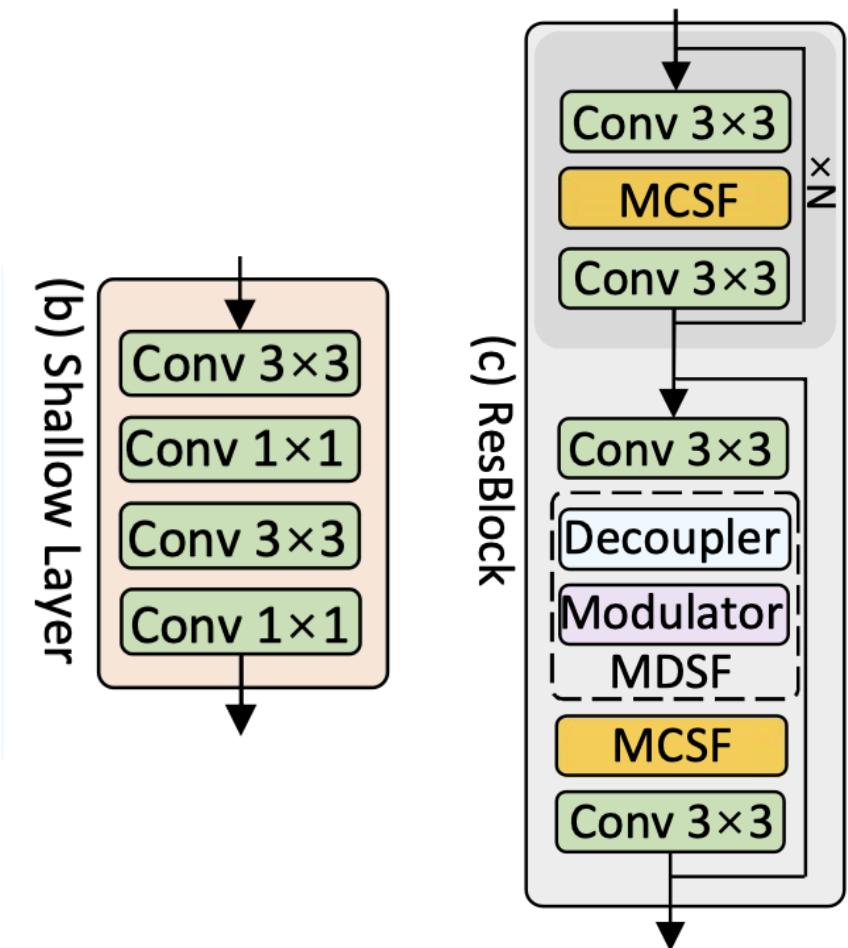
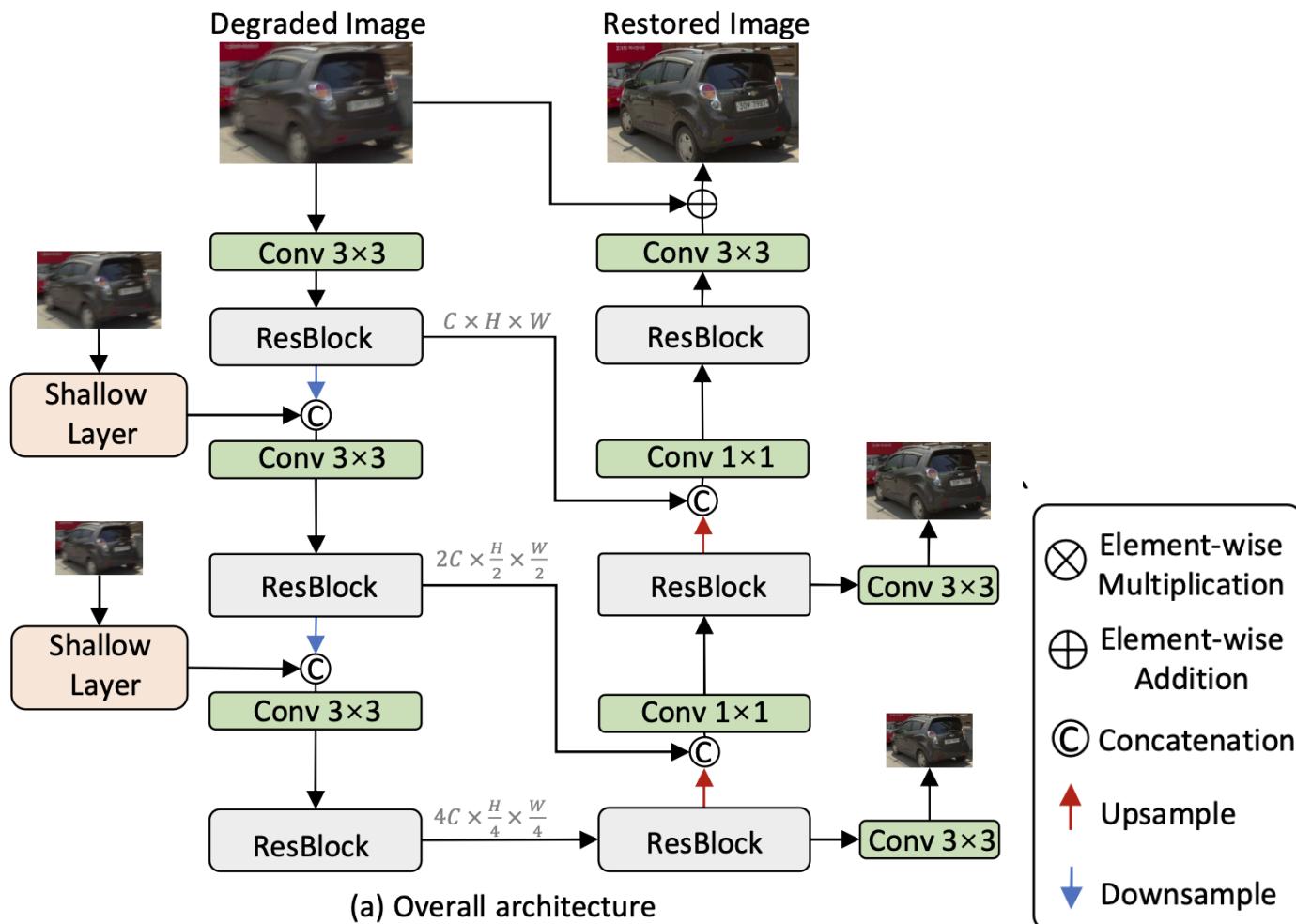
Đề xuất 2 mô đun mới:

- Multi-brand Dynamic Selective Frequency module (MDSF):
 - Tách và lựa chọn các thành phần tần số quan trọng
 - Chọn lọc tần số quan trọng với cơ chế channel-wise attention
 - Không cần bước biến đổi ngược
- Multi-brand Compact Selective Frequency module (MCSF)
 - Thực hiện phân tách và điều chỉnh tần số bằng multi-scale average pooling
 - Giúp mở rộng receptive field, giúp nắm bắt được các long-range dependencies

MDSF + MCSF + Kiến trúc U-Net --> **Selective Frequency Network (SFNET)**

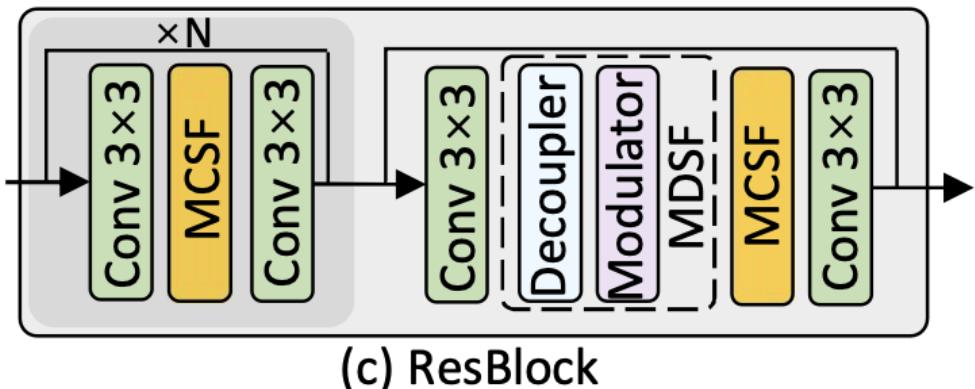
3. Methodology – Kiến trúc tổng quan

- Dựa trên **Unet**: (*three-scale*) Encoder-Decoder
 - Mỗi scale chứa một **ResBlock c**
 - ResBlock c**: Đè xuất module **MDSF** và **MCSF** xử lý ảnh dựa trên tần số
 - Shallow layer b**: Trích xuất shallow feature từ ảnh input đã giảm kích thước

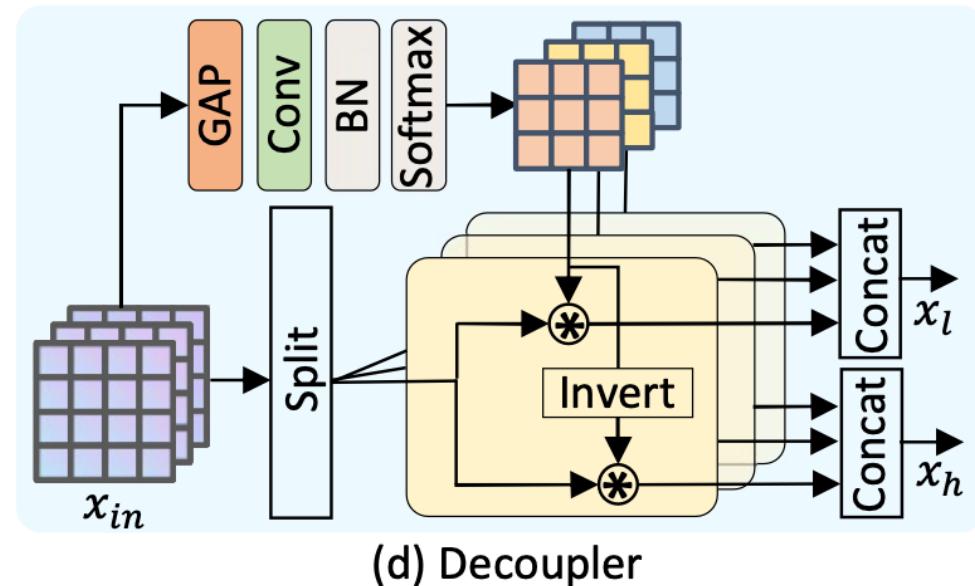


3. Methodology – MDSF (Multi-branch Dynamic Selective Frequency Module)

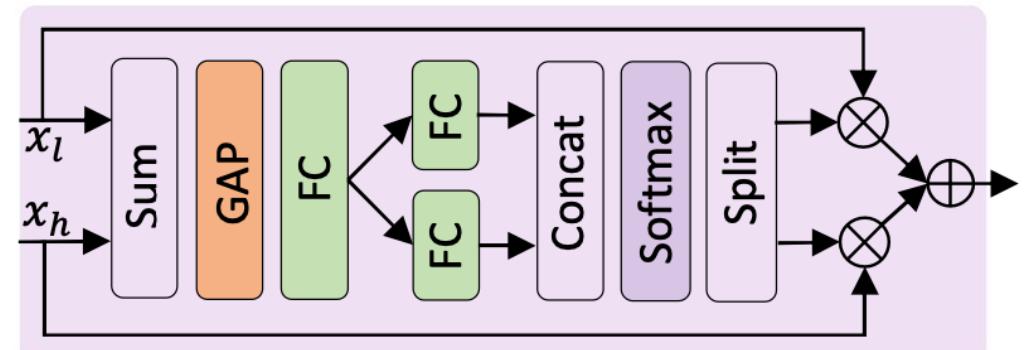
- **Chức năng của MDSF:** Chọn lọc và tăng cường các tần số quan trọng cho quá trình khôi phục hình ảnh.
- Hai thành phần chính:
 - **Decoupler d:** Phân tách đặc trưng thành các thành phần tần số cao và thấp dựa trên các bộ lọc học được (Low-freq filters và High-freq filters).
 - **Modulator e:** Dùng channel-wise attention để lựa chọn thành phần tần số quan trọng nhất.



(c) ResBlock



(d) Decoupler



(e) Modulator

3. Methodology – MDSF (Multi-branch Dynamic Selective Frequency Module)

Low-pass filter hay bộ lọc thông thấp chỉ cho phép các tín hiệu có tần số thấp hơn một mức cụ thể đi qua. Bộ lọc này có kích thước kernel $k \times k$ và được chia thành g nhóm kênh để giảm số lượng tham số, giúp mô hình gọn nhẹ hơn.

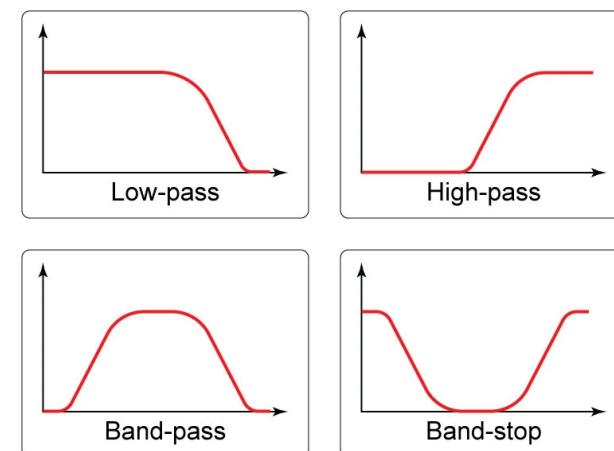
High-pass filter hay bộ lọc thông cao chỉ cho phép các tín hiệu có tần số cao hơn một mức cụ thể đi qua, được tính bằng cách trừ Low-pass filter với một identity kernel.

Công thức biểu diễn Low-pass filter:

$$F^l = \text{Softmax}((\mathcal{B}(W(\text{GAP}(X)))))$$

Trong đó:

- F^l : Low-pass filter
- X : Feature map
- \mathcal{B} : Batch Normalization
- W : Parameters of Convolution.
- GAP : Global Averaging Pooling



Một số loại filter (Nguồn: allaboutcircuits.com)

3. Methodology – MDSF (Multi-branch Dynamic Selective Frequency Module)

Sau khi có bộ lọc F^l và F^h , feature map đầu vào được tách thành hai phần:

- Thành phần tần số thấp X^l chứa các thông tin tổng quát (cạnh, góc, màu sắc trong ảnh).
- Thành phần tần số thấp X^h chứa các thông tin chi tiết (hình dạng, kết cấu).

Công thức tính các thành phần tần số thấp và cao:

$$X_{i,c,h,w}^l = \sum_{p,q} F_{i,p,q}^L X_{i,c,h+p,w+q}$$

$$X_{i,c,h,w}^h = \sum_{p,q} F_{i,p,q}^H X_{i,c,h+p,w+q}$$

Trong đó:

- $X_i \in R^{C_i \times H \times W}$: đặc trưng nhóm
- i : chỉ mục nhóm
- c : chỉ số kênh
- h, w : tọa độ trong không gian.
- p, q : thuộc các giá trị {-1, 0, 1}.

3. Methodology – MDSF (Multi-branch Dynamic Selective Frequency Module)

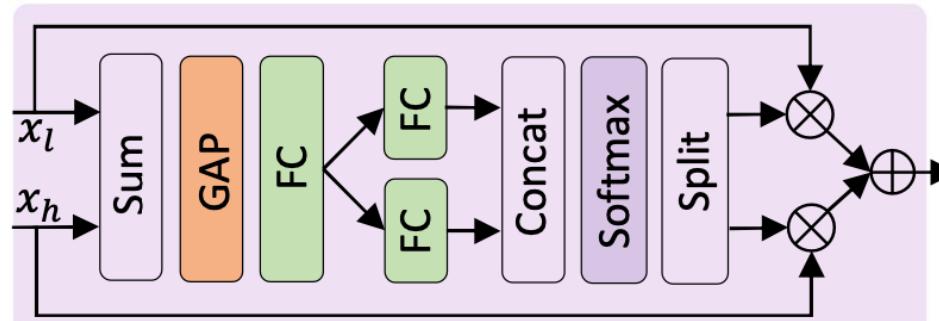
Sau khi phân tách, Bộ điều chỉnh tần số (Modulator) sẽ nhấn mạnh phần thực sự hữu ích cho việc tái tạo ảnh.

Bộ điều chỉnh tần số đầu tiên sẽ hợp nhất hai thành phần tần số cao và thấp bằng công thức:

$$Z = W_{fc}(\text{GAP}(X^l + X^h))$$

Trong đó:

- W_{fc} : tham số của fully connected layer (FC).
- GAP : Global Averaging Pooling.



(e) Modulator

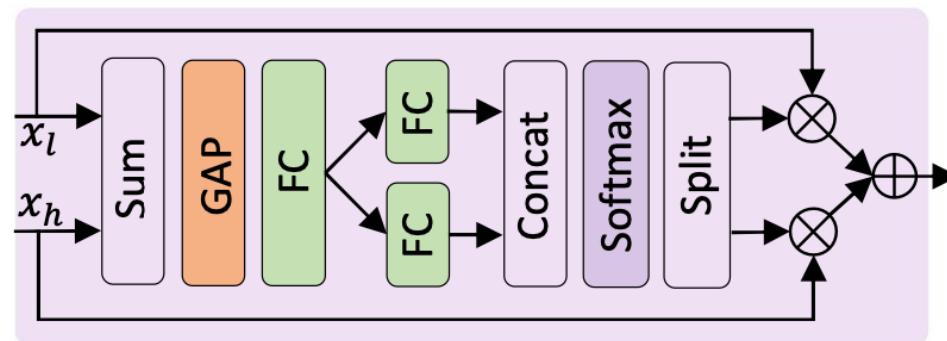
3. Methodology – MDSF (Multi-branch Dynamic Selective Frequency Module)

Sau khi tính toán Z , hai fully connected layer được sử dụng để tính toán trọng số của channel-wise attention cho hai thành phần tần số trên bằng công thức:

$$[W^l, W^h]_c = \frac{e^{[W_l(Z), W_h(Z)]_c}}{\sum_j^{2C} e^{[W_l(Z), W_h(Z)]_j}}$$

Trong đó:

- W^l : trọng số của attention cho đặc trưng tần số thấp.
- W^h : trọng số của attention cho đặc trưng tần số cao.
- $[\cdot, \cdot]$: biểu diễn phép nối.
- c: chỉ số kênh của các đặc trưng được nối.



(e) Modulator

3. Methodology – MDSF (Multi-branch Dynamic Selective Frequency Module)

Sử dụng kiến trúc đa nhánh (multi-branch structure) để tận dụng đặc trưng từ nhiều mức độ chi tiết khác nhau. Đặc trưng cuối cùng được tổng hợp từ các nhánh này, được tính như sau:

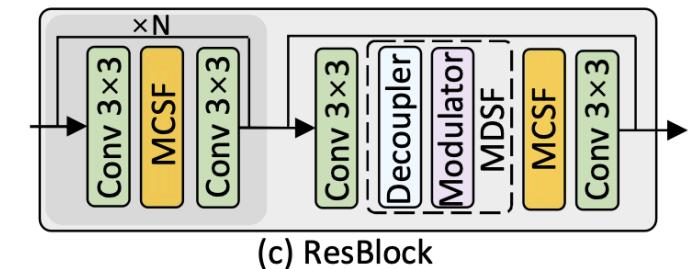
$$\hat{X} = [\mathcal{M}_1(\mathcal{D}_1(X_1)), \dots, \mathcal{M}_m(\mathcal{D}_m(X_m))]$$

Trong đó:

- D : bộ phân tách tần số (Decoupler)
- M : bộ điều chỉnh tần số (Modulator)
- X_m : đặc trưng đã được phân chia đều.

3. Methodology – MCSF (Multi-branch Compact Selective Frequency Module)

- ❑ Vấn đề của phục hồi ảnh: Các vết mờ/hỗng trên mỗi mẫu ảnh có kích thước khác nhau
 - Vai trò của Receptive Field (**RF**)
 - Mở rộng **RF** với **MCSF**
- ❑ **MCSF** = Nhánh **global RF** + Nhánh **window-based RF** (dựa trên window-based attention)
- ❑ Nhánh **window-based RF**:
 - ❑ Chia input thành 4 window
 - ❑ Lấy thông tin tần số **thấp**: Dùng **Global Average Pooling**
 - ❑ Lấy thông tin tần số **cao**: Lấy các window trừ **low-frequency map**
 - ❑ Chọn các dãy tần hữu ích: Rescale các **low-freq** và **high-freq map** bằng learnable weights
 - ❑ Chuyển các frequency map về kích thước input
- ❑ Nhánh **global RF**: Áp dụng trực tiếp quá trình trên lên input (không chia window)
- ❑ **MCSF** vs. **MDSF**:
 - ❑ **MCSF** Mở rộng **RF**
 - ❑ Không dùng **Conv Freq-Decoupler** và **Conv Freq-Modulator**
 - Giảm chi phí tính toán + Đặt được nhiều lần trên **ResBlock c**



(c) ResBlock

3. Methodology – Loss Function

- Sử dụng hàm mất mát L1 trong miền không gian và miền tần số.

- $L_{spatial} = \sum_{r=1}^3 \frac{1}{S_r} \|\widehat{X}_r - Y_r\|_1$

- $L_{frequency} = \sum_{r=1}^3 \frac{1}{S_r} \|\mathcal{F}(\widehat{X}_r) - \mathcal{F}(Y_r)\|_1$

Với r biểu thị chỉ số của các ảnh đầu vào/đầu ra với các độ phân giải khác nhau; \mathcal{F} đại diện cho phép biến đổi Fourier nhanh (Fast Fourier Transform - FFT); S_r là số phần tử dùng để chuẩn hóa; và \widehat{X}_r , Y_r lần lượt là ảnh đầu ra và ảnh mục tiêu.

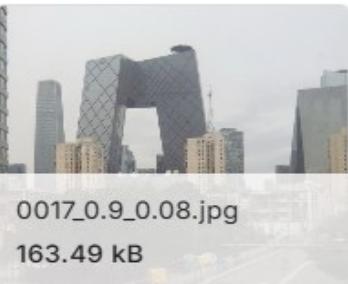
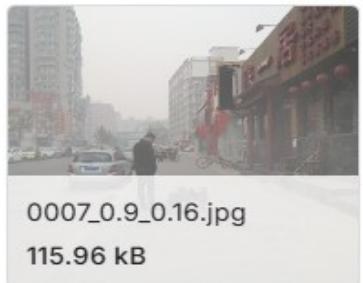
Hàm mất mát cuối cùng được xác định là:

$$L = L_{spatial} + \lambda L_{frequency}$$

trong đó λ được đặt là 0.1.

4. Mức 1: Pre-trained model + Dữ liệu từ paper

Khử sương mù



Khử tuyết

Image Desnowing	CSD	38.41	0.99
Số liệu bài báo	SRRS	32.40	0.98
	Snow100K	33.79	0.95

Method	SOTS-Indoor		SOTS-Outdoor	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
SFNet (Paper)	41.24	0.996	40.05	0.996
SFNet (Pretrained)	41.24	0.9962	39.26	0.9956

Bảng so sánh số liệu đánh giá mô hình SFNet trong bài báo và chạy thực nghiệm trên hai tập dữ liệu SOTS-Indoor và SOTS-Outdoor [8].
Bảng cho thấy hai kết quả điểm đánh giá giống nhau.

=====

The average PSNR is 32.51 dB
The average SSIM is 0.9824

Kết quả
chạy lại
xấp xỉ với
bài báo
(SRRS)

=====

The average PSNR is 33.83 dB
The average SSIM is 0.9541

Kết quả
chạy lại
xấp xỉ với
bài báo
(Snow100K)

4. Mức 1: Pre-trained model + Dữ liệu từ paper

Khử mờ

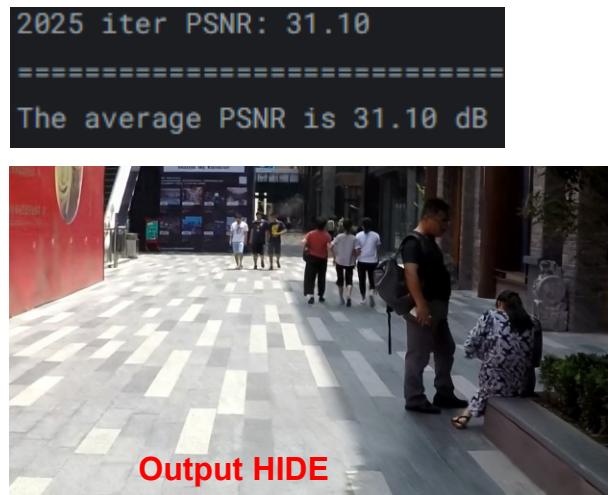
Datasets:

HIDE

```
Namespace(model_name='SFNet', data_dir='/kaggle/input/hide-dataset/HIDE', mode='test', data='HIDE', batch_size=4, learning_rate=0.0001, weight_decay=0, num_epoch=3000, print_freq=100, num_worker=8, save_freq=100, valid_freq=100, resume='', test_model='/kaggle/input/gopro/pytorch/default/1/GOPRO.pkl', save_image=True, model_save_dir='results/SFNet/HIDE/Training-Results/', result_dir='results/SFNet/HIDE')
```

Table 3: Image motion deblurring results on GoPro (Nah et al., 2017) and HIDE (Shen et al., 2020) datasets.

Method	GoPro		HIDE	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑
DeblurGAN-v2	29.55	0.934	26.61	0.875
DBGAN	31.10	0.942	28.94	0.915
DMPHN	31.20	0.940	29.09	0.924
SPAIR	32.06	0.953	30.29	0.931
MIMO-UNet+	32.45	0.957	29.99	0.930
IPT	32.52	-	-	-
MPRNet	32.66	0.959	30.96	0.939
HINet	32.71	0.959	30.32	0.932
Restormer	32.92	0.961	31.22	0.942
Stripformer	<u>33.08</u>	<u>0.962</u>	31.03	0.940
SFNet	33.27	0.963	<u>31.10</u>	<u>0.941</u>



Khử mưa

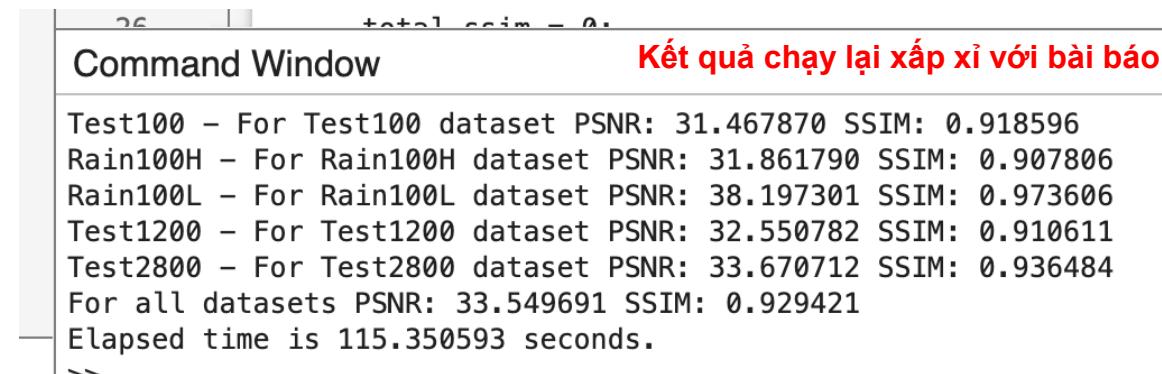
Datasets:

Test100, Rain100H, Test100L,
Test2800, Test1200

Table 7: Deraining comparisons with previous methods on five deraining datasets: Rain100H (Yang et al., 2017), Rain100L (Yang et al., 2017), Test100 (Zhang et al., 2019a), Test1200 (Zhang & Patel, 2018) and Test2800 (Fu et al., 2017).

Số liệu bài báo

Method	Test100		Rain100H		Rain100L		Test2800		Test1200		Average	
	PSNR	SSIM										
DerainNet	22.77	0.810	14.92	0.592	27.03	0.884	24.31	0.861	23.38	0.835	22.48	0.796
SEMI	22.35	0.788	16.56	0.486	25.03	0.842	24.43	0.782	26.05	0.822	22.88	0.744
DIDMDN	22.56	0.818	17.35	0.524	25.23	0.741	28.13	0.867	29.65	0.901	24.58	0.770
UMRL	24.41	0.829	26.01	0.832	29.18	0.923	29.97	0.905	30.55	0.910	28.02	0.880
RESCAN	25.00	0.835	26.36	0.786	29.80	0.881	31.29	0.904	30.51	0.882	28.59	0.857
PreNet	24.81	0.851	26.77	0.858	32.44	0.950	31.75	0.916	31.36	0.911	29.42	0.897
MSPFN	27.50	0.876	28.66	0.860	32.40	0.933	32.82	0.930	32.39	0.916	30.75	0.903
MAXIM-2S	<u>31.17</u>	0.922	<u>30.81</u>	<u>0.903</u>	<u>38.06</u>	0.977	33.80	0.943	32.37	0.922	<u>33.24</u>	0.933
SFNet	31.47	<u>0.919</u>	31.90	0.908	38.21	<u>0.974</u>	<u>33.69</u>	<u>0.937</u>	32.55	<u>0.911</u>	33.56	<u>0.929</u>



5. Mức 2: Pre-trained model + Dữ liệu ngoài paper



Input



Result

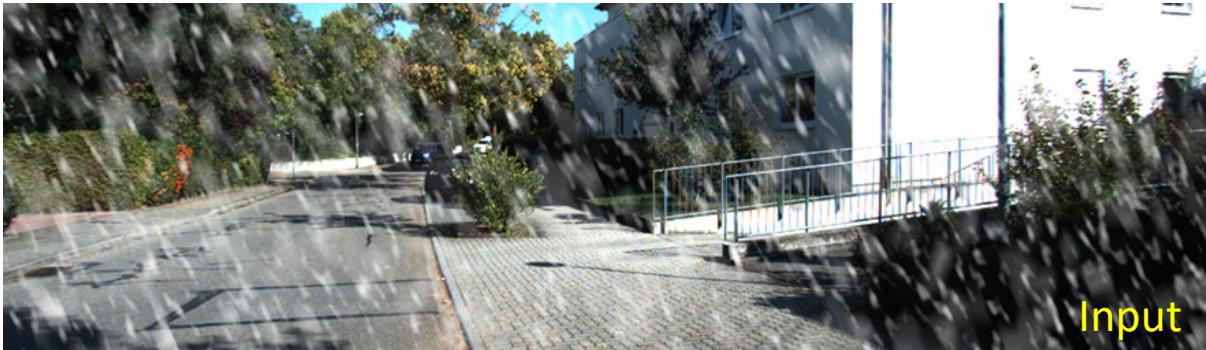


Ground truth

1000 iter PSNR: 30.65 SSIM: 0.940320

The average PSNR is 28.14 dB
The average SSIM is 0.9308

5. Mức 2: Pre-trained model + Dữ liệu ngoài paper



Input



Result



Ground truth

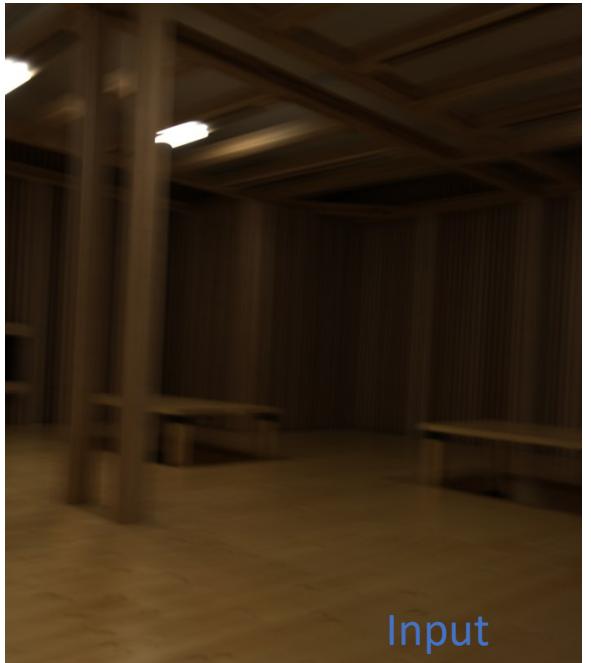
1000 iter PSNR: 23.50 SSIM: 0.821

=====

The average PSNR is 22.01 dB

The average SSIM is 0.8116

5. Mức 2: Pre-trained model + Dữ liệu ngoài paper



Input



Output



Input



Output

The average PSNR is 33.46 dB

Khử mờ – RealBlur_R Dataset

The average PSNR is 26.24 dB

Khử mờ – RealBlur_J Dataset

5. Mức 2: Pre-trained model + Dữ liệu ngoài paper

Dataset:

Rui Qian et al – DeRaindrop

(Attentive Generative Adversarial Network for Raindrop Removal from A Single Image - CVPR2018)



Khử mưa – Dataset 1

5. Mức 2: Pre-trained model + Dữ liệu ngoài paper

input (13.7k files) ⤒ ⤓ >

About this directory
This file does not have a description yet.

⤒

 10029.jpg 39.37 kB	 1003.jpg 46.42 kB	 10030.jpg 42.07 kB	 10031.jpg 47.43 kB	 10032.jpg 47.59 kB	 10033.jpg 49.54 kB	 10034.jpg 47.16 kB
 10035.jpg 49.58 kB	 10036.jpg 46.59 kB	 10037.jpg 48.89 kB	 10038.jpg 47.77 kB	 10039.jpg 49.7 kB	 1004.jpg 45.74 kB	 10040.jpg 46.71 kB
 10041.jpg 49.16 kB	 10042.jpg 49.74 kB	 10043.jpg 47.6 kB	 10044.jpg 49.84 kB	 10045.jpg 46.5 kB	 10046.jpg 46.7 kB	 10047.jpg 47.98 kB
 10048.jpg 46.66 kB	 10049.jpg 47.96 kB	 1005.jpg 54.12 kB	 10050.jpg 46.03 kB			

Data Explorer

Version 1 (2.69 GB)

- Rain13K
 - deraining_testset
 - ↳ Rain100H
 - ↳ Rain100L
 - ↳ Test100
 - ↳ Test1200
 - ↳ Test2800
 - input
 - ↳ 1.jpg
 - ↳ 10.jpg
 - ↳ 100.jpg
 - ↳ 1000.jpg
 - ↳ 10000.jpg
 - ↳ 10001.jpg
 - ↳ 10002.jpg
 - ↳ 10003.jpg
 - ↳ 10004.jpg
 - ↳ 10005.jpg
 - ↳ 10006.jpg
 - ↳ 10007.jpg
 - ↳ 10008.jpg
 - ↳ 10009.jpg

Summary

- ↳ 36.0k files

+ New Version

Những hạt mưa lì lùng trên tập train của bài báo

5. Mức 2: Pre-trained model + Dữ liệu ngoài paper

Dataset:

Rui Qian et al – DeRaindrop (CVPR2018)



Input (Phóng to)



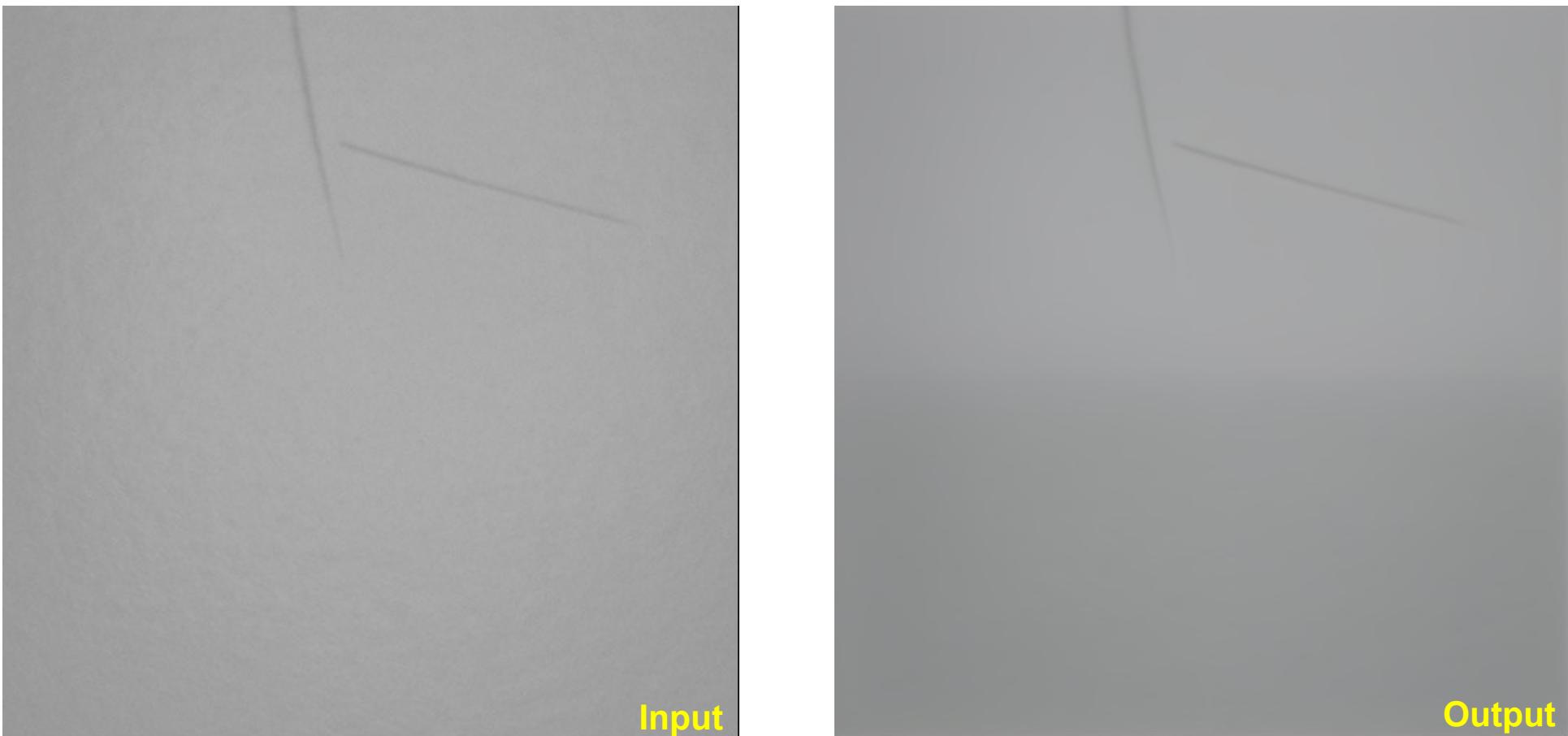
Output (Phóng to)

Khử mưa – Dataset 1

5. Mức 2: Pre-trained model + Dữ liệu ngoài paper

Dataset:

PudPawat - Scratch Defect on Surface (Kaggle)

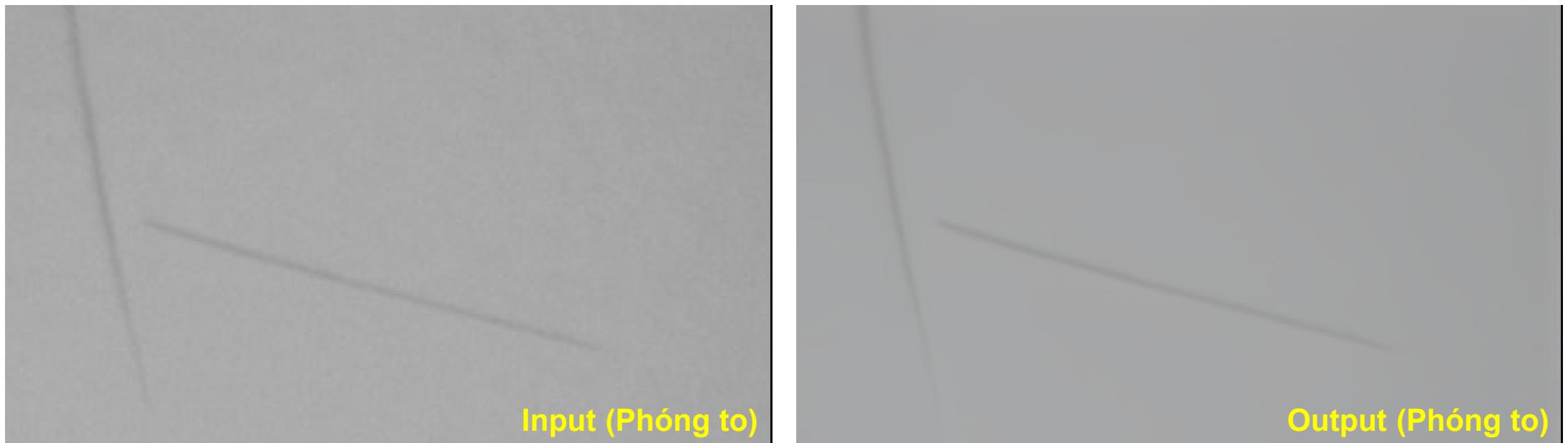


Khử mưa – Dataset 2

5. Mức 2: Pre-trained model + Dữ liệu ngoài paper

Dataset:

PudPawat - Scratch Defect on Surface (Kaggle)



6. Mức 3: Dữ liệu ngoài paper + Fine-tuning

Dataset:

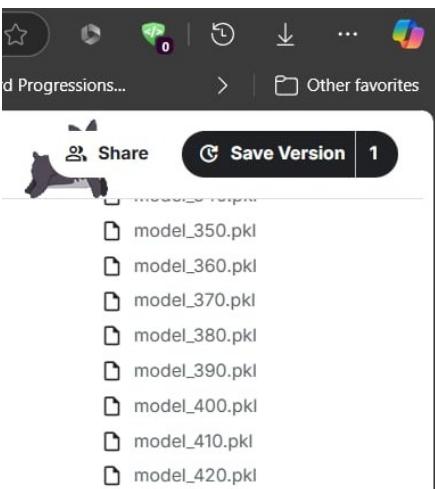
Rui Qian et al – DeRaindrop

(Attentive Generative Adversarial Network for Raindrop Removal from A Single Image - CVPR2018)

Số epoch đã chạy của bộ weight từ **bài báo**: 270

Số epoch **nhóm** đạt đến sau khi fine-tune: 420 (thêm 150 epoch)

```
n't have full control of the loaded file. Please open a
state = torch.load(args.resume)
Resume from 270
Time: 1.6326 Epoch: 271 Iter: 100/ 287 LR: 0.00003333
Time: 1.6350 Epoch: 271 Iter: 200/ 287 LR: 0.00003333
EPOCH: 271
Elapsed time: 4.70 Epoch Pixel Loss: 0.1919 Epoch FFT
```



Input



Output



Ground truth

6. Mức 3: Dữ liệu ngoài paper + Fine-tuning

Dataset:

Rui Qian et al – DeRaindrop (CVPR2018)



Ground truth (Phóng to)



Output (Phóng to)

Fine-tune với task Khử mưa trên Dataset 1

*. Thử nghiệm khử mưa thực hiện thêm với video từ mạng xã hội



Input (id: 10)



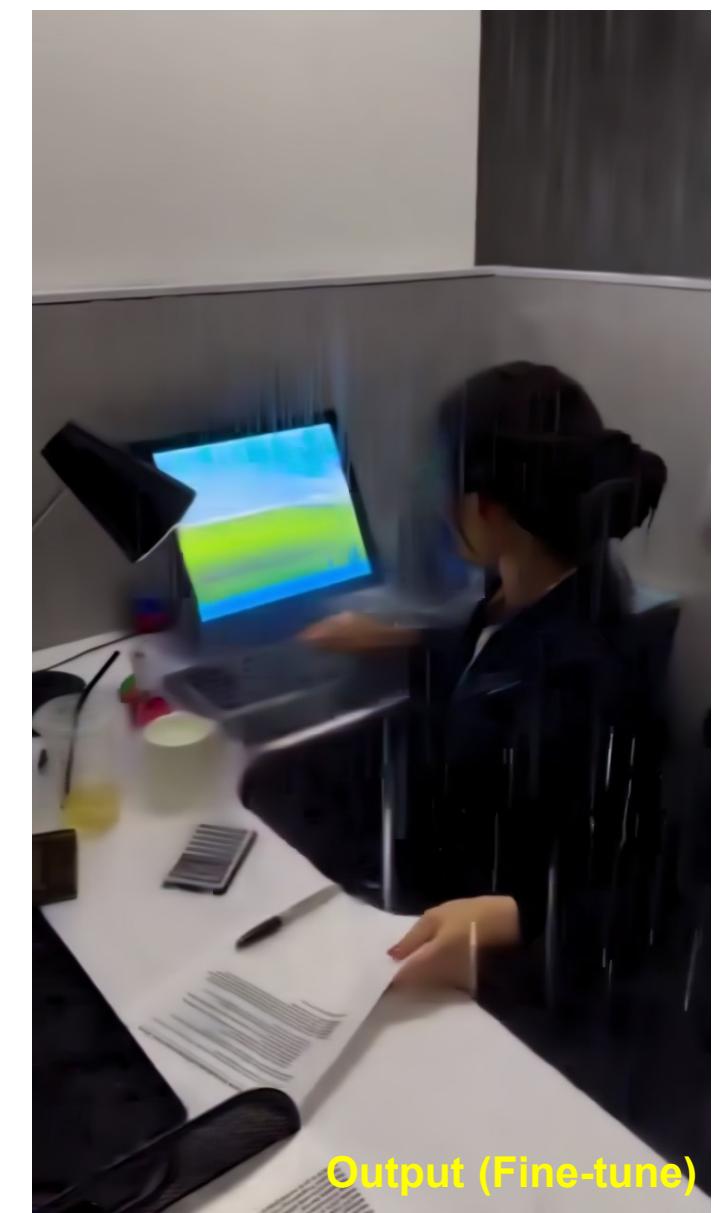
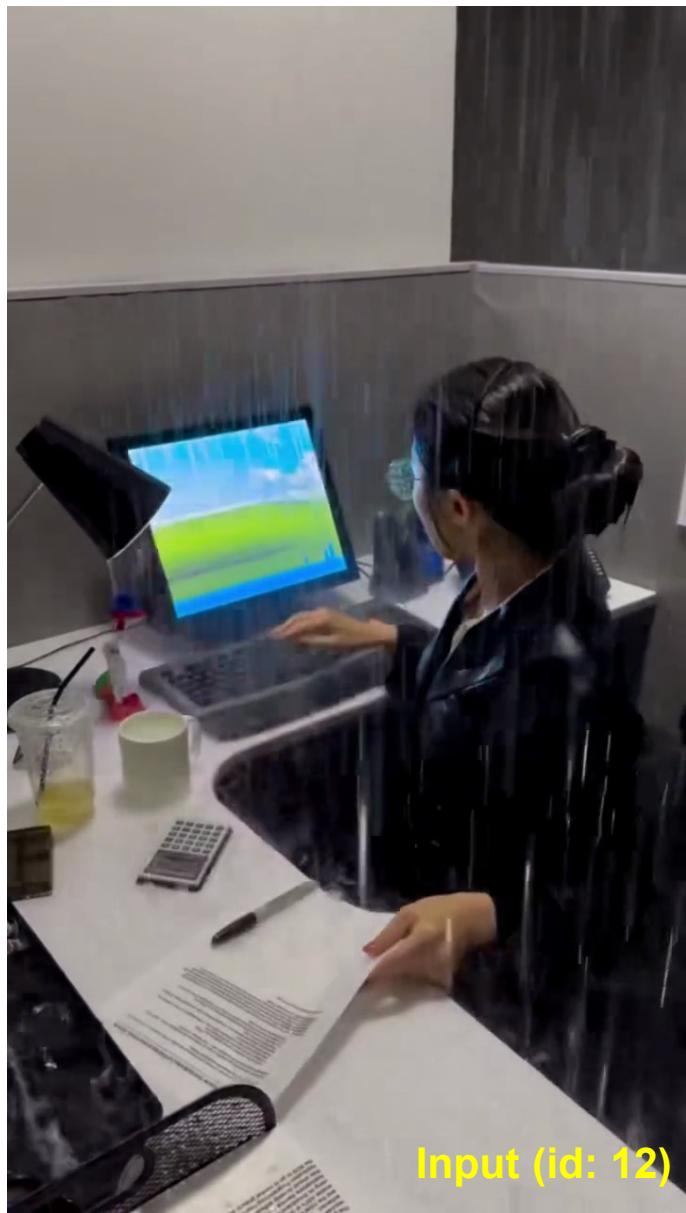
Output (Pre-trained)



Output (Fine-tune)

Video 1: <https://www.youtube.com/watch?v=QKk6IWuduSU>

*. Thử nghiệm khử mưa thực hiện thêm với video từ mạng xã hội



Video 2: https://www.instagram.com/reel/DFdCxeST7t6/?utm_source=ig_web_copy_link&igsh=MzRIODBiNWFIZA==

*. Thử nghiệm khử mưa thực hiện thêm với video từ mạng xã hội



*. Thử nghiệm khử mưa thực hiện thêm với video từ mạng xã hội



Video 4: <https://www.youtube.com/shorts/51OCVGigWLQ>

*. Thử nghiệm khử mưa thực hiện thêm với video từ mạng xã hội



Dữ liệu từ bài báo (Rain100H)

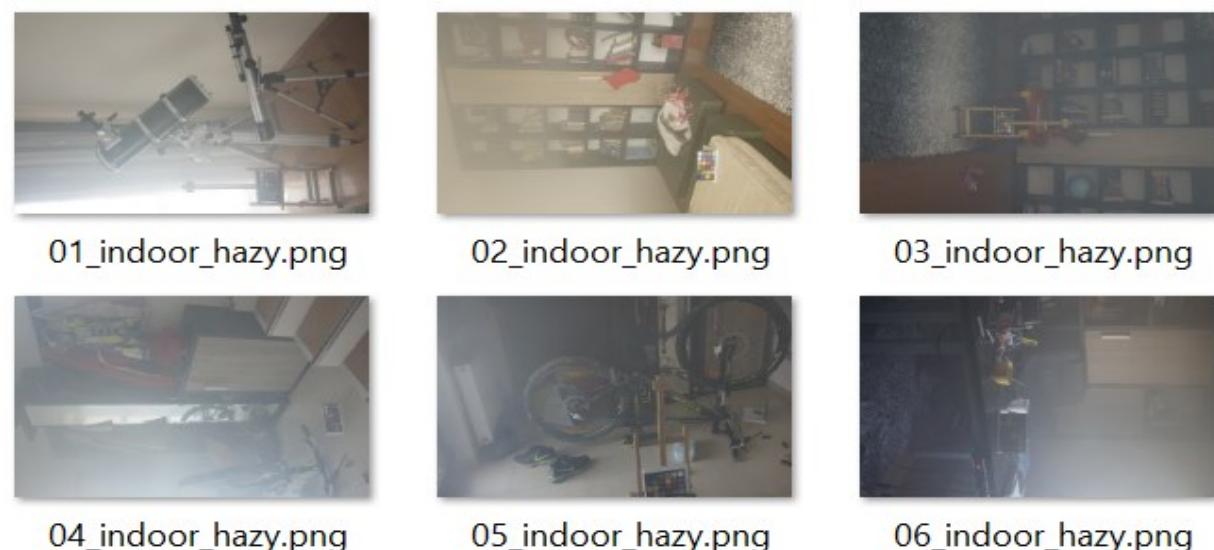
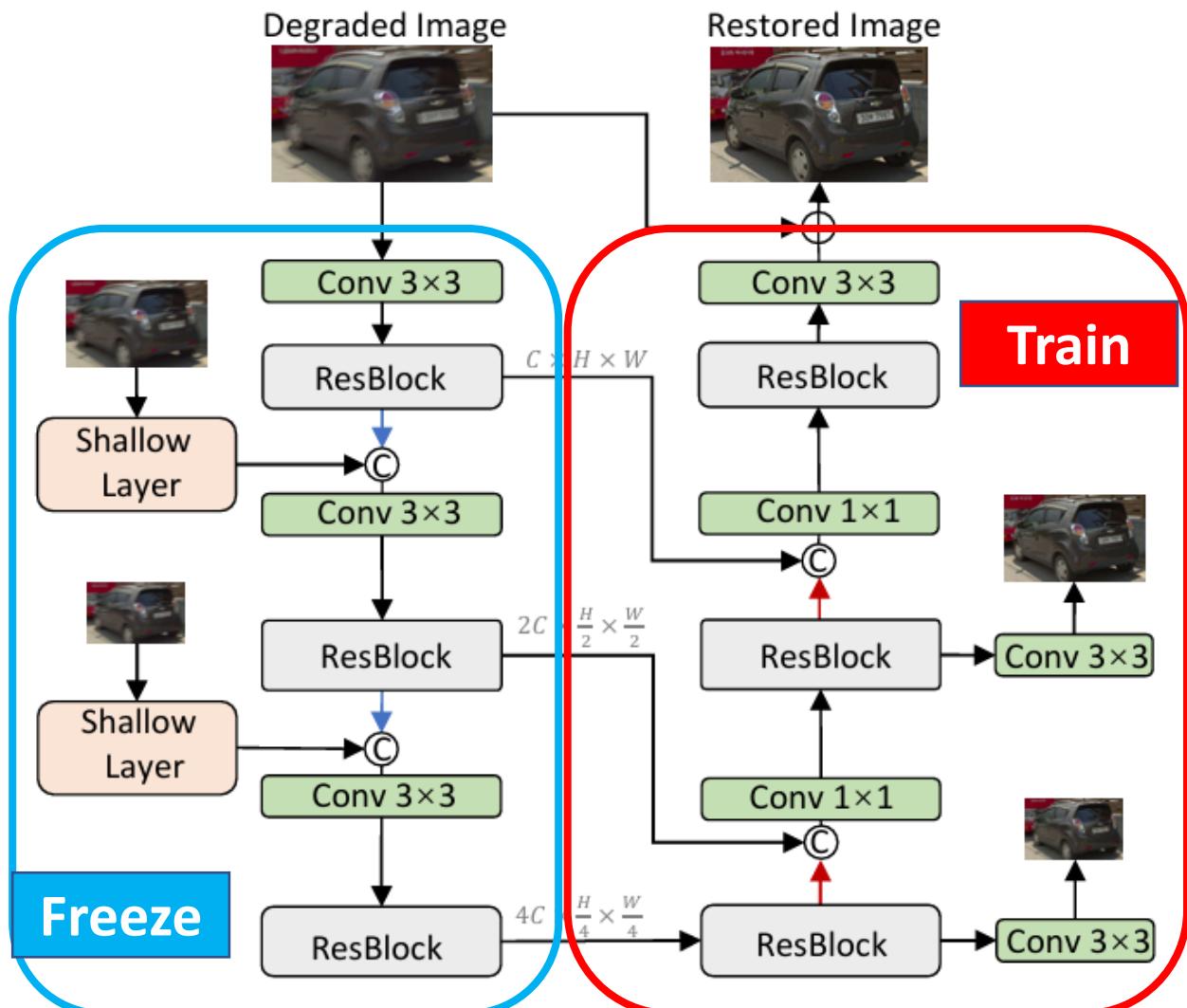


6. Mức 3: Dữ liệu ngoài paper + Fine-tuning

Tác vụ: **Khử sương mù trong ảnh**

Tập dữ liệu huấn luyện: **I-Haze [9]**

Tập dữ liệu tiền huấn luyện: **SOTS-Indoor [8]**



Hình minh họa một số ảnh trong tập dữ liệu I-Haze

I-Haze		
Method	PSNR↑	SSIM↑
w/o fine-tune	12.54	0.588
Fine-tune	18.44	0.8422

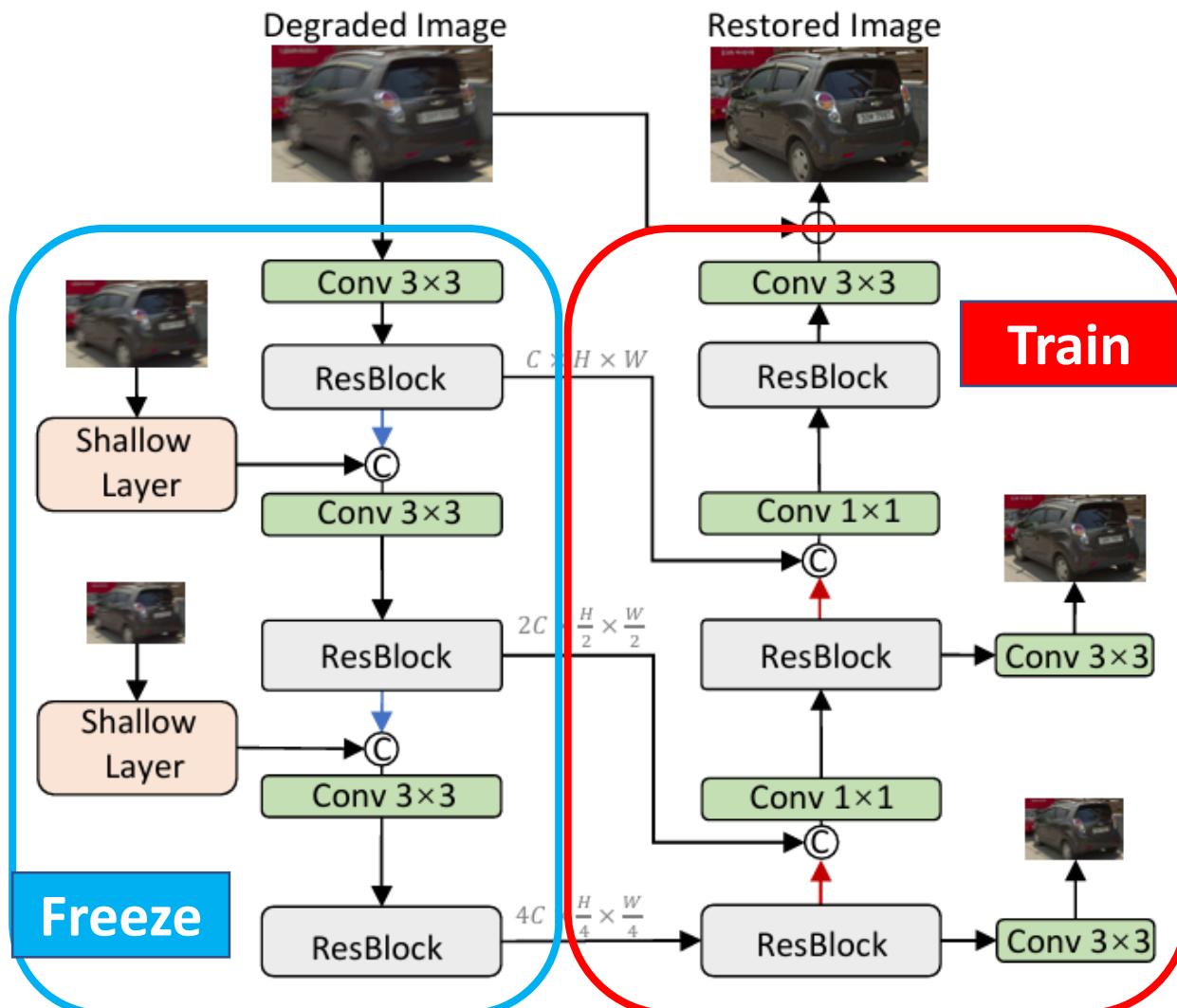
Bảng so sánh định lượng đánh giá mô hình SFNet trên tập dữ liệu I-Haze không fine-tune (w/o fine-tune) và có fine-tune.

6. Mức 3: Dữ liệu ngoài paper + Fine-tuning

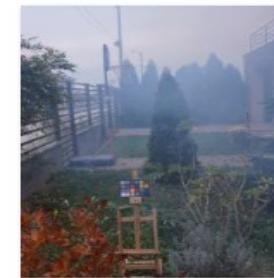
Tác vụ: Khử sương mù trong ảnh

Tập dữ liệu huấn luyện: O-Haze [10]

Tập dữ liệu tiền huấn luyện: SOTS-Outdoor [8]



01_outdoor_hazy.png



02_outdoor_hazy.png



03_outdoor_hazy.png



04_outdoor_hazy.png



05_outdoor_hazy.png



06_outdoor_hazy.png

Hình minh họa một số ảnh trong tập dữ liệu O-Haze

O-Haze

Method	PSNR↑	SSIM↑
w/o fine-tune	17.93	0.7652
Fine-tune	22.51	0.9078

Bảng so sánh định lượng đánh giá mô hình SFNet trên tập dữ liệu O-Haze không fine-tune (w/o fine-tune) và có fine-tune.

6. Mức 3: Dữ liệu ngoài paper + Fine-tuning

Tác vụ: **Khử sương mù** trong ảnh

Tập dữ liệu huấn luyện: **O-Haze**

Tập dữ liệu tiền huấn luyện: **SOTS-Outdoor**



Input



Output (Fine-tune)



Output (w/o fine-tune)

Trực quan kết quả khử sương mù của ảnh số 43 trong tập dữ liệu O-Haze (tập test)

*. Identify scientific, nuisance, and fixed hyperparameters

Scientific h-params: Siêu tham số mà ta muốn đo đạc sự ảnh hưởng của nó đến hiệu suất mô hình.

Nuisance h-params: Siêu tham số phiền toái; Cần tối ưu để có thể so sánh công bằng các giá trị khác nhau của **S**.

Fixed h-params: Hằng số trong thử nghiệm hiện tại; Muốn so sánh hai giá trị **S**, chúng phải có cùng **F**, vì hai thí nghiệm dùng các giá trị **F** khác nhau có thể dẫn đến kết luận khác nhau.

Ví dụ: Thí nghiệm để trả lời câu hỏi “*Số hidden layer của một mô hình tăng có làm giảm validation error không?*”

S: Số hidden layer

N: Learning rate (số layer khác nhau cần LR khác nhau để so sánh công bằng)

F: Activation function (Giả định: Activation function được chọn không bị ảnh hưởng bởi số lớp)

Chiến lược tiếp cận khi thực hiện một thí nghiệm:

- Xác định các giá trị **S** của bài toán
- Xem các siêu tham số còn lại là **N**
- Thực nghiệm và chuyển dần một số giá trị **N** về **F**

Tài liệu tham khảo

- [1] <https://viblo.asia/p/gioi-thieu-mang-resnet-vyDZOa7R5wj>
- [2] <https://www.kaggle.com/code/andradaolteanu/how-i-taught-myself-deep-learning-vanilla-nns>
- [3] <https://viblo.asia/p/paper-explain-hieu-ve-skip-connection-mot-ki-thuat-nho-ma-co-vo-trongcac-kien-truc-residual-networks-3Q75w7bQ5Wb>
- [4] <https://viblo.asia/p/shallow-neural-networks-Do7546y0ZM6>
- [5] <https://machinelearningmastery.com/introduction-to-1x1-convolutions-to-reduce-the-complexity-of-convolutional-neural-networks/>
- [6] <https://www.linkedin.com/pulse/vanilla-neural-networks-introduction-shashank-r-shankar/>
- [7] David L. Poole and Alan K. Mackworth - Artificial Intelligence Foundations of Computational Agents, 3rd Edition
- [8] B. Li, W. Ren, D. Fu, D.Tao, D. Feng, W.Zeng, and Z.Wang. Benchmarking single image dehazing and beyond. *IEEE Transactions on Image Processing*, PP:1–1, 2018.
- [9] Ancuti, C., Ancuti, C. O., Timofte, R., & De Vleeschouwer, C. (2018). I-HAZE: A dehazing benchmark with real hazy and haze-free indoor images. In *Advanced Concepts for Intelligent Vision Systems: 19th International Conference, ACIVS 2018, Poitiers, France, September 24–27, 2018, Proceedings* 19 (pp. 620-631). Springer International Publishing.

Tài liệu tham khảo

[10] Ancuti, C. O., Ancuti, C., Timofte, R., & De Vleeschouwer, C. (2018). O-haze: a dehazing benchmark with real hazy and haze-free outdoor images. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 754-762).