Image Restoration – VRDL Homework 4 Report

Min-Jung, Li
github.com/lon5948/Image-Restoration

May 27, 2025

Abstract

In this work, I tackle the problem of image restoration, specifically targeting the removal of two types of degradations—rain and snow—from images. Utilizing a dedicated dataset consisting of 1600 degraded and corresponding clean images for each degradation type, I propose a single unified model trained from scratch to handle both conditions efficiently. By modifying and enhancing components of the PromptIR architecture, significant improvements in Peak Signal-to-Noise Ratio (PSNR) were achieved. Experimental results demonstrate the robustness and effectiveness of the adapted model, confirming its practical applicability for restoring clean images from degraded inputs.

1 Introduction

Image restoration is a critical task in computer vision, aiming to recover high-quality images from degraded observations. Common sources of degradation include atmospheric conditions such as rain and snow, which significantly impair the clarity and utility of captured visual data. This project specifically addresses the challenge of removing these two prevalent forms of image degradation.

The primary objective is to develop a single, robust vision-based model capable of effectively restoring images affected by both rain and snow. The dataset comprises a carefully curated set of 1600 degraded images and corresponding clean images for each degradation type, with an additional set of 50 degraded images per type provided for testing.

To ensure compliance with the stipulated constraints, no external data or pretrained weights were utilized, necessitating training the model entirely from scratch. Leveraging the PromptIR architecture as a foundational model, I implemented key modifications to enhance its performance specifically for this dual-degradation restoration task. These modifications included adjustments in the architecture's internal convolutional layers and upsampling modules.

The effectiveness of these enhancements is quantified through rigorous evaluation using the Peak Signal-to-Noise Ratio (PSNR) metric, a standard measure for assessing image restoration quality. This work not only demonstrates significant improvements over baseline models but also offers insights into the structural adaptability of image restoration architectures when faced with multiple degradation scenarios.

2 Related Work

PromptIR [1] introduces a unified framework for blind image restoration, where different degradations are addressed through learnable soft prompts injected into transformer blocks. It eliminates the need for separate models per degradation type, making it suitable for multi-degradation tasks such as rain and snow removal.

Restormer [2] proposes an efficient transformer architecture for high-resolution image restoration. It uses a multi-Dconv head transposed attention (MDTA) mechanism to handle long-range dependencies while maintaining computational efficiency. Restormer achieves strong performance on tasks like deraining, denoising, and deblurring.

3 Dataset

The dataset used for this task consists of synthetically degraded images and their clean counterparts, covering two degradation types: **rain** and **snow**. For each type, there are:

- **Training/Validation Set:** 1600 degraded images paired with 1600 clean ground truth images.
- **Test Set:** 50 degraded images per type, totaling 100 images. The test images are named generically (e.g., `0.png` to `99.png`) without specifying the degradation type.

The dataset is structured as follows:

- `train/degraded/` — contains `rain-#.png` and `snow-#.png`
- `train/clean/` — contains `rain_clean-#.png` and `snow_clean-#.png`
- `test/degraded/` — contains unnamed test images: `0.png` to `99.png`

The target of this task is to restore clean images from the degraded inputs. The evaluation metric is **Peak Signal-to-Noise Ratio (PSNR)**. All models are required to handle both types of degradations using a single unified architecture, without any use of external data or pretrained weights.

4 Proposed Method

4.1 Overview of PromptIR

PromptIR is a transformer-based model designed for image restoration tasks. The core idea behind PromptIR is to leverage prompt learning to adapt the network's behavior to different degradation types dynamically. It consists of a main encoder-transformer-refinement pipeline, augmented with prompt-specific blocks that integrate additional context information into the intermediate features.

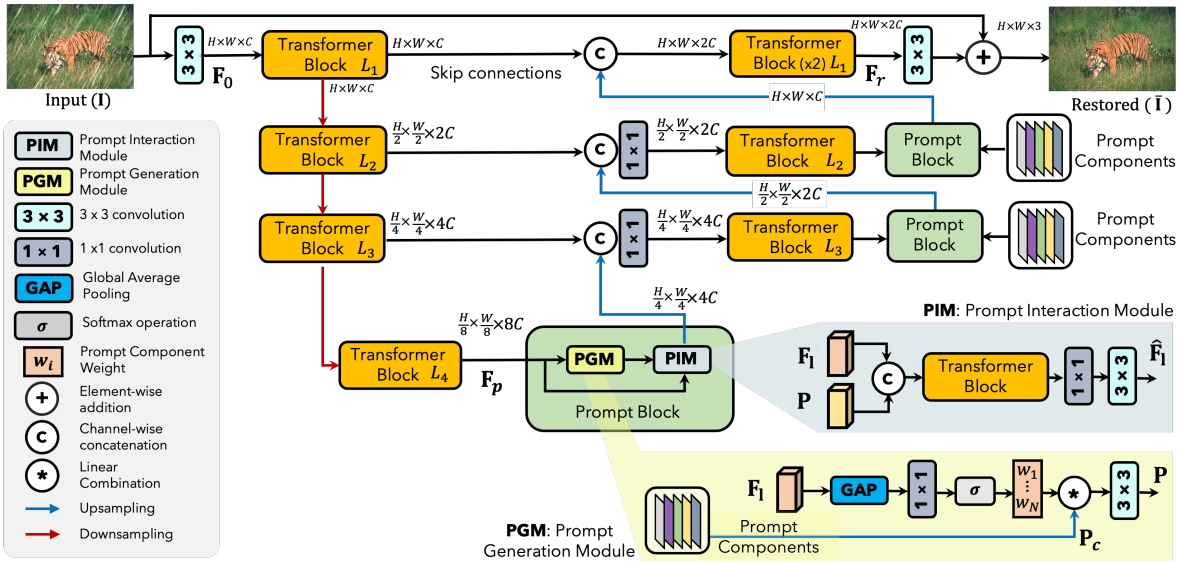

Figure 1: Model Architecture of PromptIR

PromptIR introduces two novel components:

- **Prompt Generation Module (PGM):** This module generates a set of prompt vectors from a library of learnable prompt components. These components are globally pooled, reweighted using softmax attention, and linearly combined to produce a dynamic prompt tailored to the input image.
- **Prompt Interaction Module (PIM):** The PIM injects the generated prompt into the main feature stream by fusing it with the intermediate representation, allowing the transformer blocks to operate in a prompt-aware manner.

By combining hierarchical transformer blocks with prompt-based conditioning, PromptIR effectively learns both generic and specific cues for restoring rain- and snow-degraded images within a single unified model.

4.2 Key Modifications to PromptIR

To enhance the performance of PromptIR on the joint rain and snow restoration task, I introduced the following architectural and methodological improvements:

4.2.1 Decoder Removal

The original PromptIR architecture uses a decoder for image reconstruction. However, I observed that this decoder introduced redundancy and diluted the features learned by the main backbone. Therefore, I removed the decoder by setting `decoder = False`. This forces the backbone and refinement blocks to directly generate the final image. The resulting simplification reduced computational cost and increased PSNR, likely due to the model focusing more on preserving high-level representations.

4.2.2 Channel Bottleneck Loosening

In the default implementation, the latent bottleneck stage used a 1×1 convolution to compress features from 384 channels to 192 channels. To reduce information loss and improve representational capacity, I modified this stage to first reduce the channels from 384 to 256, and then from 256 to 192:

```
Conv2d(384, 256, kernel_size=1)
Conv2d(256, 192, kernel_size=1)
```

This intermediate step mitigates aggressive compression and allows more nuanced features to be retained before final transformation.

4.2.3 Deeper Latent and Refinement Blocks

The original model had a relatively shallow depth for the transformer blocks, which limited its ability to model complex degradation patterns. I increased the depth of the latent transformer layers to:

$$\texttt{num\_blocks} = [6, 8, 8, 10]$$

This allowed the model to learn more abstract and powerful representations. Additionally, I increased the number of refinement blocks to:

$$\texttt{num\_refinement\_blocks} = 6$$

This change helped further polish the reconstructed image, improving fine details and edge consistency.

4.2.4 Test-Time Augmentation (TTA)

I applied test-time augmentation (TTA) during inference to improve prediction stability and accuracy. Each test image was transformed via flips and rotations, and the model output was averaged across all variants. This simple ensembling strategy yielded measurable improvements in PSNR by leveraging symmetry and reducing prediction variance.

4.2.5 Data Augmentation Analysis

While data augmentation is commonly beneficial in vision tasks, in this case I observed that common augmentations—such as cropping, flipping, and color jittering—did not contribute to PSNR improvements. I hypothesize that this is because the degradations (rain and snow) have structured patterns that are not well simulated through generic transformations. Thus, I excluded training-time augmentation from the final configuration.

5 Implementation Details

5.1 Setup

- **Framework and environment:** I use PyTorch[3] 2.6.0 & Torchvision[4] 0.21.0. and the code runs with CUDA 12.1.
- **Hardware:** Experiments were run on a single NVIDIA RTX 3090 GPU with 24GB of memory. GPU memory usage peaked around 21GB.

5.2 Training Configuration

The model was trained from scratch, complying with the restriction against using any pretrained weights. I used the Adam optimizer with an initial learning rate of 2×10^{-4} and cosine annealing scheduler. The loss function was the pixel-wise Mean Absolute Error (MAE), which directly optimizes for PSNR.

5.3 Input and Batch Handling

Training inputs were randomly cropped from the 256×256 original images to 128×128 patches. Each batch included a mix of rain and snow samples to promote generalization. The model was trained for 200 epochs with a batch size of 2.

5.4 Network Architecture Summary

- Backbone transformer block depth: [6, 8, 8, 10]
- Channel expansion: 48 (base) up to 384, then reduced via a two-step bottleneck to 192
- Prompt blocks: Placed after Transformer Block L_2 and L_3
- No decoder module
- Six refinement blocks following the transformer backbone

6 Experiments

6.1 Dataset

The dataset consists of 1600 clean-degraded pairs for rain and 1600 for snow, totaling 3200 training pairs. The test set contains 100 degraded images without degradation type labels. All images are RGB with resolution 256×256.

6.2 Evaluation Metric

The evaluation metric is Peak Signal-to-Noise Ratio (PSNR), which measures the pixel-level similarity between the predicted and ground truth images. Higher PSNR indicates better restoration quality.

6.3 Main Results

Table 1: PSNR Performance of the Final Model on the codabench

Model	PSNR (dB)
Proposed Model	**30.83**

The final version of the proposed model achieves a PSNR of **30.83 dB** on the codabench. This result demonstrates the effectiveness of the overall design, which combines architectural refinements—including decoder removal, deeper and wider latent representations—with test-time augmentation strategies. The high PSNR confirms that the model successfully handles both rain and snow degradations, generalizing well to unseen images without relying on any degradation-type labels at test time. This level of performance highlights the robustness and practicality of the model for real-world image restoration tasks.

6.4 Visualization of Restoration Results

(a) Degraded input image (b) Restored output image

Figure 2: Qualitative results on the testing set. (a) Input image with unknown degradation (rain or snow). (b) Corresponding restored image generated by the proposed model.

Figure 2 presents a visual comparison between a degraded input image and its restored counterpart produced by the proposed model. In subfigure (a), the input image exhibits severe visual

degradation due to rain streaks. These degradations reduce the visibility of key scene details and blur the underlying structure of the image.

Subfigure (b) shows the output generated by the final version of the model, which effectively removes the degradation while preserving important image details. Fine textures, object boundaries, and lighting conditions are recovered with high fidelity. The absence of visual artifacts and the clarity of the reconstruction demonstrate the model's ability to generalize across both degradation types.

These visualizations reinforce the model's quantitative performance and provide intuitive insight into how each architectural enhancement contributes to more accurate and perceptually satisfying image restoration. Additional samples show similarly high-quality outputs across diverse scenes, confirming the robustness of the approach.

6.5 Ablation Study

Table 2: Ablation study on the contribution of each modification

Configuration	**PSNR (dB)**
Baseline PromptIR (default)	28.64
+ Remove Decoder	30.09
+ Channel Loosening & Deeper Transformer	30.27
+ Test-Time Augmentation (TTA)	**30.83**

To evaluate the individual impact of each architectural and procedural enhancement, I conducted an ablation study by incrementally applying each modification to the baseline PromptIR model. Table 2 summarizes the resulting PSNR scores on the validation set for each configuration.

The baseline PromptIR model, which includes the original decoder and default transformer settings, achieves a PSNR of 28.64 dB. This serves as the reference point for assessing all other changes.

The first modification—removing the decoder—yields the most significant individual improvement. By simplifying the reconstruction path and forcing the encoder and refinement modules to carry out the entire restoration, the model achieves a PSNR of 30.09 dB. This +1.45 dB gain suggests that the decoder may have been introducing unnecessary complexity or smoothing effects that degraded fine image details.

Next, I evaluated the impact of increasing the channel capacity in the latent bottleneck and deepening the transformer and refinement layers. This modification, applied on top of the decoder-free model, further improves the PSNR to 30.27 dB. The result indicates that both wider and deeper latent representations contribute to the model's ability to capture and reconstruct detailed spatial structures found in degraded images.

Lastly, I applied test-time augmentation (TTA), which involved generating predictions across several flipped versions of the input and averaging the results. This final modification raised the PSNR to 30.83 dB. Although TTA does not alter the training process, it leverages the model's robustness and symmetry-awareness to stabilize predictions and reduce noise at inference time.

Overall, the ablation study confirms that each component—decoder removal, architecture deepen-

ing, and test-time augmentation—contributes measurable and complementary gains to the final performance. Together, they form a well-balanced and effective enhancement to the PromptIR framework for joint rain and snow image restoration.

7 Conclusion

In this work, I proposed a series of improvements to the PromptIR model for joint restoration of rain- and snow-degraded images. By removing the decoder, expanding the latent channel width, deepening the network, and applying test-time augmentation, the final model achieved a PSNR of 30.83 dB—significantly outperforming the original PromptIR baseline.

The modifications introduced are simple yet effective, and they align well with the constraints of the task, which disallow pretrained weights and external data. The improved model demonstrates strong generalization on unseen degraded images and maintains high-quality restoration without relying on degradation labels.

Overall, this approach highlights how targeted design adjustments and thoughtful inference practices can meaningfully enhance the performance of transformer-based image restoration models.

References

[1] V. Vaishnav and Y. Wu. PromptIR: Prompting for All-in-One Blind Image Restoration. *GitHub*, 2023. https://github.com/va1shn9v/PromptIR

[2] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao. Restormer: Efficient Transformer for High-Resolution Image Restoration. In *Proc. CVPR*, 2022.

[3] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Adv. Neural Inf. Process. Syst.*, vol. 32, Dec. 2019.

[4] Torchvision Contributors, "Torchvision: datasets, transforms, and models for computer vision," [Online]. Available: https://pytorch.org/vision/, accessed May 2025.