312540021 / Kezia Nathania (林紀霞)

HW4 Report

GitHub: https://github.com/Kezia-Nathania/NYCU-Visual-Recognition-Spring-2025-HW4.git

Introduction

The task of this homework is image restoration for two types of degradation, rain and snow. The dataset is split into training/validation with 1600 pairs of clean and degraded images for each degradation type, and the testing set consists of 50 degraded images for each type. This restoration task is evaluated using PSNR (Peak Signal-to-Noise Ratio).

Method

1. Train/Validation Split and Augmentation
    a. Splitting

    To enable model evaluation on unseen data, the dataset is programmatically divided into training and validation subsets during initialization. The split is based on a user-defined ratio (defaulting to 80/20):

    - Shuffling: All valid (degraded, clean) image pairs are first collected and then randomly shuffled to avoid ordering bias.
    - Partitioning: The shuffled list is then divided into two subsets:
        - Training Set (split='train'): Contains the first (1 - val_ratio) portion of the data.
        - Validation Set (split='val'): Contains the remaining val_ratio portion.

    This ensures that training and validation sets are mutually exclusive and collectively exhaustive subsets of the full dataset.

    b. Augmentation

    To improve the generalization of the model, a lightweight data augmentation strategy is applied to training samples:

- Horizontal Flip: Applied with 50% probability.

To ensure spatial consistency, flipping is done on both degraded and clean images simultaneously.

2. PromptIR

a. Key Design

PromptIR uses prompts to encode degradation-specific information. These prompts are a set of tunable parameters that encode crucial discriminative information about various types of image degradation. The method utilizes a plug-in prompt module, known as the “prompt block”. This block is generic and can be easily integrated into existing restoration networks, such as a UNet-style network with transformer blocks used in the PromptIR framework. The prompt block works by implicitly predicting degradation-conditioned prompts based on the input image with unknown degradation. The prompt block consists of two core components:

- Prompt Generation Module (PGM):
  This module generates input-conditioned prompts by dynamically predicting attention-based weights from the input features and applying them to learnable prompt components. This dynamic approach is preferred over static methods.
- Prompt Interaction Module (PIM): This module facilitates interaction between the generated prompts and the input features for guided restoration. It concatenates the prompts and features and passes them through a Transformer block, which uses the degradation information from the prompts to transform the input features.

The guidance from prompts is injected into the network at multiple decoding stages with few learnable parameters, enabling the network to learn a unified model [1].

b. Contribution

PromptIR introduces a prompting-based, all-in-one blind image restoration framework that recovers clean images solely from the input without requiring any prior knowledge of the degradation. This overcomes the limitations of previous approaches that depended on prior information or multiple specialized models. The prompt block, designed as a generic and architecture-agnostic plug-in module, can be seamlessly integrated into various restoration networks. PromptIR dynamically adapts to diverse restoration tasks—including image denoising, deraining, and dehazing—achieving state-of-the-art performance with a single unified model. It demonstrates substantial improvements over prior state-of-the-art all-in-one methods such as AirNet across different tasks and noise levels, especially when trained in the all-in-one setting. Furthermore, PromptIR exhibits significantly better generalization to unseen degradation levels and produces more disentangled representations of corruption types compared to the contrastive learning embeddings employed by AirNet. The framework is based on a single-stage training pipeline, which is conceptually simpler than conventional two-stage methods [1].

c. Implementation (Tuning)

In a preliminary comparison over two initial training epochs, a configuration with prompt_len = 5 achieved a validation PSNR of 22.16, whereas increasing it to prompt_len = 10 improved the validation PSNR to 22.51. Based on these findings, the final training was conducted using prompt_len = 10, indicating that a higher prompt length may enhance the network's capacity, though it also comes with a slight addition of computational cost.

3. CBAM

The Convolutional Block Attention Module (CBAM) is a lightweight and effective attention mechanism designed to improve convolutional neural networks by sequentially applying channel and spatial attention. It first infers attention maps along the channel dimension to emphasize important feature channels and then along the spatial dimension to highlight informative regions within feature maps. This two-step attention process helps the network focus on more relevant features while suppressing less useful information, leading to enhanced representation and improved performance in various vision tasks. The CBAM module is lightweight and can be easily integrated into existing architectures, as demonstrated in the original work by Woo et al. [2].

In the modified PromptIR network, the Convolutional Block Attention Module (CBAM) is incorporated at multiple key stages throughout both the encoder and decoder to enhance feature representation by emphasizing informative channels and spatial locations. Specifically, CBAM modules are applied after the outputs of the first three encoding levels (cbam1, cbam2, cbam3) as well as after critical decoder blocks (cbam_dec3, cbam_dec2, and cbam_dec1). Each CBAM sequentially performs channel attention through adaptive average pooling followed by a bottleneck convolutional structure and sigmoid gating, and spatial attention via convolution on concatenated max and average pooled feature maps across channels. This dual attention strategy allows the network to dynamically recalibrate feature responses, improving the focus on relevant details while suppressing noise and less useful information. The integration of CBAM complements the transformer-based PromptIR backbone by refining features at multiple resolutions, ultimately boosting restoration quality with minimal computational overhead. The application of CBAM at both encoding and decoding phases is expected to support more effective noise reduction and detail preservation.

4. Charbonnier Loss

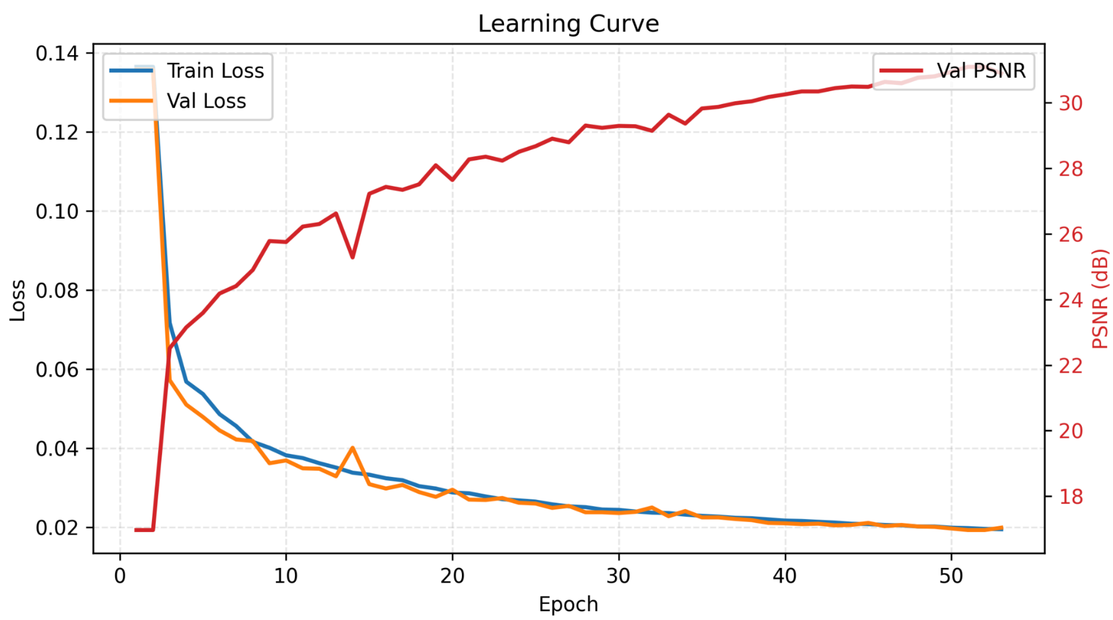
In this work, the Charbonnier loss is adopted as the primary pixelwise loss function to supervise image restoration. It serves as a differentiable variant of the L1 loss, offering a robust and smooth approximation that is especially effective at handling outliers and preserving structural details.

Charbonnier loss has been shown in prior work—particularly in GAN-based CT denoising frameworks—to combine the strengths of both L1 and L2 (MSE) losses. It maintains high performance in low-variance regions (e.g., smooth surfaces or soft tissue) while preserving edges and fine details in high-variance areas. This property addresses the common issue of over-smoothing or blurring encountered when using MSE alone and the potential instability of pure L1 loss [3].

Empirically, incorporating Charbonnier loss in restoration models has been demonstrated to improve both Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM), making it highly suitable for tasks that require both fidelity and perceptual quality [3]. In our setting, it enables the network to reconstruct clean images with enhanced textural accuracy and reduced artifacts, particularly around object boundaries and edges.

Result

Base PromptIR model is from github: https://github.com/va1shn9v/PromptIR.git. Presented below is the learning curve from the modified PromptIR training. The loss values indicate a good fit, without signs of overfitting or underfitting. Meanwhile, the validation PSNR steadily improves over time.



Here are several restored images from the test dataset. The restorations effectively preserve fine details and appear visually accurate. (left: degraded, right: restored)

The modified PromptIR (with CBAM and Charbonnier Loss) seems to effectively restore the images in the dataset provided.

Reference

[1] V. Potlapalli, S. W. Zamir, S. Khan, and F. Khan, “PromptIR: Prompting for all-in-one image restoration,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Dec. 2023.

[2] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “CBAM: Convolutional block attention module,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 3–19.

[3] B. Gajera, S. R. Kapil, D. Ziaei, J. Mangalagiri, E. Siegel and D. Chapman, "CT-Scan Denoising Using a Charbonnier Loss Generative Adversarial Network," in IEEE Access, vol. 9, pp. 84093-84109, 2021, doi: 10.1109/ACCESS.2021.3087424

[4] PromptIR github: https://github.com/va1shn9v/PromptIR.git