Learning Sequential Image Enhancement in Bilateral Space

Zongnan Bao

***Abstract*—Image enhancement plays a crucial role in improving the visual quality and interpretability of digital images, benefiting various domains such as photography, medical imaging, surveillance, and computer vision. This work proposed a novel approach to combine sequential image processing with bilateral space image processing methods, achieving competitive results while maintaining a small memory consumption and fast runtime speed. We also investigated numerous loss functions to recover color and pixel intensity of unprocessed raw images.** [1]

***Index Terms*—computer vision, image processing, image retouching, memory efficient, bilateral space**

I. Introduction

The rise of digital photography has greatly transformed how people capture and edit images. Software like Adobe Photoshop and Lightroom offer photographers a wide range of easy-to-use tools for editing. Additionally, digital cameras utilizes camera image signal processors (ISP) to automatically adjust images before they are displayed to humans. Therefore, it has become increasingly important for digital cameras to be able to adapt to different scenes and perform automatic image processing effectively.

Most of displayed images are 8-bit sRGB images, but the raw data captured by image sensors usually contains more information in different format and needs multiple processing steps to make them visually appealing for display. These steps involve demosaicing, auto-white balance, auto-exposure, auto-focusing, and others. With the recent progress in deep learning, image processing methods have emerged that can automatically convert low-quality images into high-quality ones. These methods enable tasks like image enhancement, image denoising, super resolution, style transfer, and more. In our investigation, we examined different image enhancement methods. For image enhancement task, the computations are typically performed on the device itself, such as digital cameras and phone cameras. Therefore, it is crucial for the image enhancement model to be capable of real-time processing, ideally providing immediate results on the viewfinder or within a short timeframe. This is essential for delivering a seamless and satisfying user experience.

II. Existing Methods

There are numerous existing methods ranging from traditional image processing techniques like histogram equalization, and more recent approaches based on deep learning. In this context, our emphasis lies on the deep learning methods due to their superior capabilities and broader applicability. This work is innovated by Neural Ops [1] and heavily based on their ideas.

A. Sequential Processing Based Methods

Neural Operator [1] is an innovative approach for image enhancement that aims to mimic the sequential processing performed by human photographers. By learning to apply operators in a sequential manner, the model gains interpretability, allowing users to understand the image processing steps. The trend of performing image restoration tasks in a stage-by-stage fashion has been observed in recent papers, including works like HINet [9] and MAXIM [10]. These approaches adopt a multi-stage architecture, where each stage focuses on addressing specific aspects of the restoration process. However, it's worth noting that the original Neural Operator implementation may encounter runtime performance issues when dealing with large input images. Processing larger images could potentially result in slower execution times or resource limitations.

B. GAN Based Methods

GAN-based approaches like Pix2Pix [8] are highly powerful for tasks such as image generation and style transfer. However, one limitation is their tendency to produce images with visible artifacts, resulting in a lack of visual smoothness. This drawback makes them generally unsuitable for applications in photography where high-quality results are required. Furthermore, generative models, including Pix2Pix [8], often have large model sizes that hinder real-time performance, making them less practical for certain real-time applications.

C. Look-up Tables (LUT) Based Methods

LUT-based approaches, such as AdaInt [6] and 3D-LUT [5], provide a fast and visually smooth method for image enhancement. These approaches utilize 3D Look-up Tables (LUTs) to efficiently transfer colors from one color space to another. The learned LUTs serve as a compact representation of color transfer, similar to bilateral grids. However, one limitation of 3D LUT-based methods is their lack of interpretability regarding how the model modifies the photo. Additionally, these methods often have relatively larger model sizes compared to Neural Operators.

D. Bilateral Space Based Methods

Bilateral grid comes from the paper HDRNet [2]. It is a data structure that represents an image transformation. Comparing to processing full resolution image, processing in bilateral space is faster and edge-aware. A bilateral grid is predicted using downsampled image (thus faster), and final output is sliced from the predicted grid using the full-resolution image, without the need to feed the full-resolution image into neural network. Because of this, it consumes less memory space compared to other methods processing using full resolution image.

[1] Codes available at: https://github.com/bznick98/bilateral-image-enhance

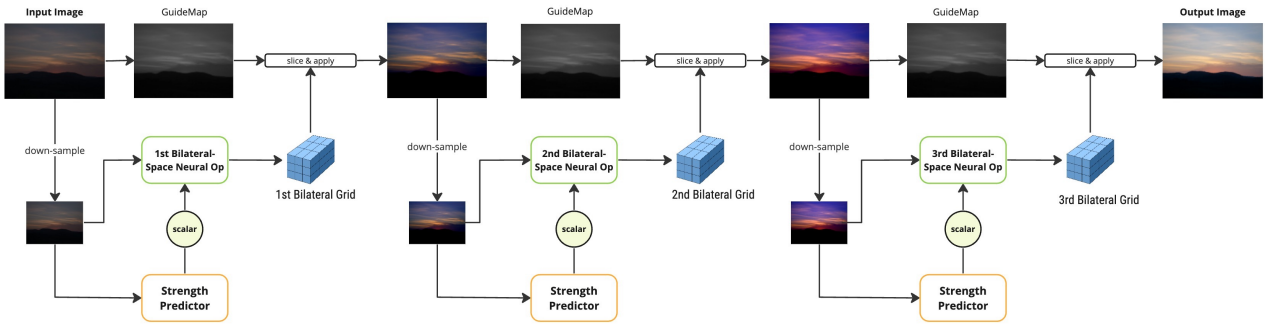

Fig. 1. Model Architecture

III. Dataset

A. MIT-Adobe-5K

MIT-Adobe-5K [3] dataset provides images that are shot and post-processed by expert photographers. This dataset consists of 5,000 raw images in 16-bit TIFF format, and corresponding expert-tuned images in 8-bit jpeg format. In order to make a fair comparison with the Neural Ops [1] paper, we decided to use the exact-same dataset they used, which was originally provided by CSRNet [12] (referred to as **FiveK-Dark**).

FiveK-Dark adopted professional-tuned images by *Expert C* in MIT-Adobe-5K dataset, and cropped both input and target images to spatial resolution of 500×300.

To initialize individual neural operators before training the entire model, we also generated 500 scenes of data for each type of image operator. Each scene contains 20 images retouched using Adobe Lightroom operator with intensity from 0% to 100%. The reason that we are generating this initialization dataset is because the original author of Neural Ops [1] did not provide the entire initialization dataset they used and we can't reproduce the exact same results.

B. VE-LOL

Vision **E**nhancement in the **LO**w-**L**ight condition (VE-LOL) [11] is an image enhancement dataset specially collected from low-light environment instead of photo adjustment like MIT-Adobe-5K [3]. Each low-light input image is paired with a normally-exposed image. We plan to train and test our model on this dataset in the future if time permits to compare with other low-light enhancement methods.

IV. Proposed Method

A. Motivation

Rather than relying on a black-box image transformation approach, adopting a more interpretable solution similar to the Neural Operator [1] is a viable approach for tackling the image restoration task. By applying sequential operations to the input image, the restoration process can be broken down into stages, allowing for better interpretability and control

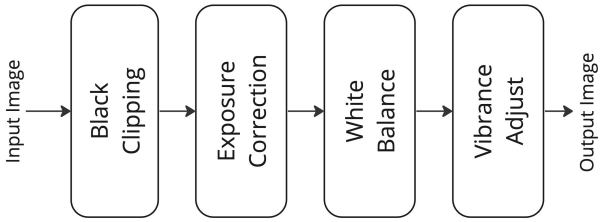

Fig. 2. Order of Bilateral Neural Operators

over the transformation. To mimic how human photographers edit photos, the intermediate operators can be constrained to approximate common photographic operations such as *black clipping*, *exposure correction*, *white balance adjustment*, *vibrance adjustment*, and more. In this work, we choose order to be the one mentioned above, but the order is subject to change for other use scenarios. This approach not only provides a clearer understanding of how the image is being modified but also allows for fine-grained adjustments to achieve the desired restoration outcome.

The original Neural Operator [1] method has shown impressive results on the MIT-Adobe-5K [3] dataset while maintaining a relatively small model size. However, its model architecture directly operates on the full-resolution image, leading to a constraint on the input image size. This limitation becomes particularly challenging when dealing with high-resolution inputs. (e.g. 4K resolution)

B. Our Solution - Bilateral Operator

To address the challenge of processing high-resolution images in the original Neural Operator [1] method, one possible approach is to replace each neural operator with a modified lightweight HDRNet [2], named as Bilateral Operator as shown in the figure 1. This alternative method avoids processing the full-resolution image "in network." Instead, it operates on a significantly down-sampled version of the image for most of

its computation, reducing the computational burden. The full-resolution image is only needed when predicting a simple map and applying the color transformation (slicing), which is a relatively low-cost operation and less dependent on the input resolution. Slicing [7] operation is parameter-free and can be implemented in CUDA. By adopting this strategy, the computational and memory requirements for processing high-resolution images can be alleviated, making the method more practical and efficient.

Each Bilateral Operator (simplified HDRNet) resizes the input image into $3 \times 256 \times 256$, and learns a bilateral grid of size $8 \times 64 \times 64$. Each grid cell has a 12-dim affine color transformation vector to transform the input RGB value. Each Bilateral Operator also learns to predict a 1-channel full resolution guidemap, which will be used for slicing the color transformation coefficients out from the grid, and apply on the input full resolution image.

C. Operator Strength Predictor

The strength predictors in our model work similar to the ones in Neural Operator [1], which it consists of 2 convolution layers, a global pooling layer and FC layers. Each predictor predicts a scalar value for each of the Bilateral Operator, controlling the strength of each operator. The benefits of using a scalar to control the strength of the operator is that the model can learn more effectively how operators are affecting the image.

V. Results

A. Quantitative Results

We evaluated our method along with the results from the paper NeuralOps [1]. To be fairly compared with Neural Ops [1], we used our own initialization dataset instead of the one provided by them for evaluating their model (marked as NeuralOps*). We also reported their claimed performance for reference. Detailed quantitative image metric reports are shown in I. The reason that there is a gap between the reproduced result and the officially claimed results is that they did not provide the full initialization dataset (only half was provided). All other methods are reported as-is from the NeuralOps [1] paper.

We also measured runtime and memory consumption for Neural Operator [1] and our method Bilateral Operator. The results II shows that our methods achieves similar image enhancement results while maintaining a much smaller memory consumption and faster inference speed, allowing for higher resolution input and more potential on mobile devices.

B. Visual Comparison

To make it easier to understand the strengths and weaknesses of the model, we show the visualization of our model for randomly selected four scenes in figure 3. The images are displayed from left to right, starting with the input image, followed by the output image, the target image, and the L2 difference between the output and target images. By examining these visuals and analyzing our measurements, we find that our model produces excellent results in terms of pixel intensity (SSIM). However, there are occasional instances where the colors in the output image differ from the target image, which cannot be fully captured by the SSIM metric alone.

VI. Implementation

A. Loss Function

We experimented numerous loss functions for model training, ranging from classical L1/L2 loss to recently proposed SSIM-L1 [4] and Deep Histogram Match Loss [13]. Our experimentations shows that SSIM-L1 [4] helps model achieve superior results in terms of intensity, while Deep Histogram Match Loss [13] helps achieve great color consistent results than simply using L1 loss.

B. Training Scheme

For MIT-Adobe-5K dataset, we used two-phase training strategy. In the Phase-I, only the bilateral neural operators are trained individually by randomly selecting pairs of enhanced image with the corresponding value. Given intensity values and desired outputs, our model is trained to reproduce the results as much as possible. In the Phase-II, we incorporated an encoder and a operator intensity predictor for each pre-trained bilateral operator and jointly train them together.

C. Hyperparameter Setting

We used Adam [14] Optimizer with 10^{-8} weight decay, 0.9 beta1 and 0.99 beta2, using batch size of 1 trained 400 epochs for Phase-I and 134 epochs for Phase-II.

VII. Conclusion

In our study, we conducted an investigation into various model designs with the goal of enhancing the efficiency of the Neural Operator [1], a sequential image retouching method. Through our efforts, we successfully achieved competitive results, demonstrating faster runtime and reduced memory consumption compared to previous approaches. However, further work is needed to refine operator initialization schemes to better overcome color shifts problem in our results. For example, during Phase-II training, we could design specific losses for each operator's output, makeing them to closely approximate the image operators in Adobe Lightroom. Overall, this work contributes by reviewing recent methods, introducing a novel model architecture, and improving the efficiency of the Neural Operator [1] for more practical and more accurate image enhancement.

References

[1] Wang, Y., Li, X., Xu, K., He, D., Zhang, Q., Li, F., & Ding, E. (2022, November). Neural Color Operators for Sequential Image Retouching. In Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIX (pp. 38-55). Cham: Springer Nature Switzerland.

[2] Gharbi, M., Chen, J., Barron, J. T., Hasinoff, S. W., & Durand, F. (2017). Deep bilateral learning for real-time image enhancement. ACM Transactions on Graphics (TOG), 36(4), 1-12.

[3] Bychkovsky, V., Paris, S., Chan, E., & Durand, F. (2011, June). Learning photographic global tonal adjustment with a database of input/output image pairs. In CVPR 2011 (pp. 97-104). IEEE.

Model	PSNR ↑	SSIM ↑	ΔE ↓	LPIPS ↓	# Params
White-Box	18.59	0.797	17.42	-	8,561,762
Distort & Recover	19.54	0.800	15.44	-	259,263,320
HDRNet	22.65	0.880	11.83	-	482,080
DUPE	20.22	0.829	16.63	-	998,816
MIRNet	19.37	0.806	16.51	-	31,787,419
Pix2Pix	21.41	0.749	13.26	-	11,383,427
3D-LUT	23.12	0.874	11.26	-	593,516
CSRNet	23.86	0.897	10.57	-	36,489
NeuralOps	24.32	0.907	9.795	0.045	28,108
NeuralOps*	24.04	0.898	10.333	0.048	28,108
BilateralOps (Ours)	24.22	0.906	9.830	0.043	69,012

Results in the upper section are from [1].
* means self-reproduced results, can't match exactly with the original paper.
↑ means higher is better, ↓ means lower is better.

TABLE I
QUANTITATIVE COMPARISON USING MIT-ADOBE-5K DATASET.

	500×300	1200×900	2500×1200	4000×3000
NeuralOps	8.26ms (121 FPS) **262 MB**	44.41ms (22 FPS) **1.61 GB**	121.14ms (8 FPS) **4.43 GB**	N/A **N/A**
BilateralOps (Ours)	6.78ms (147 FPS) **55 MB**	32.86ms (30 FPS) **266 MB**	87.05ms (11 FPS) **702 MB**	350.22ms (2 FPS) **2.70 GB**

The upper number shows inference runtime speed in milliseconds (ms).
The lower number shows inference memory consumption in MB.
N/A means model runs *out-of-memory* for given input resolution.
Measured on NVIDIA Tesla T4 16GB GPU, Intel Xeon 8259 CL @ 2.50GHz 8-core CPU and 32GB RAM.

TABLE II
RUNTIME & MEMORY CONSUMPTION UNDER DIFFERENT INPUT RESOLUTIONS.

[4] H. Zhao, O. Gallo, I. Frosio and J. Kautz, "Loss Functions for Image Restoration With Neural Networks," in IEEE Transactions on Computational Imaging, vol. 3, no. 1, pp. 47-57, March 2017, doi: 10.1109/TCI.2016.2644865.

[5] Zeng, H., Cai, J., Li, L., Cao, Z., & Zhang, L. (2020). Learning Image-Adaptive 3D Lookup Tables for High Performance Photo Enhancement in Real-Time. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44, 2058-2073.

[6] Yang, C., Jin, M., Jia, X., Xu, Y., & Chen, Y. (2022). AdaInt: learning adaptive intervals for 3D lookup tables on real-time image enhancement. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 17522-17531).

[7] Jiawen Chen, Sylvain Paris, and Frédo Durand. 2007. Real-time edge-aware image processing with the bilateral grid. ACM Trans. Graph. 26, 3 (July 2007), 103–es. https://doi.org/10.1145/1276377.1276506

[8] Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Imagetoimage translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 11251134).

[9] Chen, L., Lu, X., Zhang, J., Chu, X., & Chen, C. (2021). Hinet: Half instance normalization network for image restoration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 182-192).

[10] Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A., & Li, Y. (2022). Maxim: Multi-axis mlp for image processing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 5769-5780).

[11] Liu, J., Xu, D., Yang, W., Fan, M., & Huang, H. (2021). Benchmarking low-light image enhancement and beyond. International Journal of Computer Vision, 129, 1153-1184.

[12] He, J., Liu, Y., Qiao, Y., & Dong, C. (2020). Conditional sequential modulation for efficient global image retouching. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16 (pp. 679-695). Springer International Publishing.

[13] Avi-Aharon, M., Arbelle, A., & Raviv, T. R. (2020). Deephist: Differentiable joint and color histogram layers for image-to-image translation. arXiv preprint arXiv:2005.03995.

[14] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

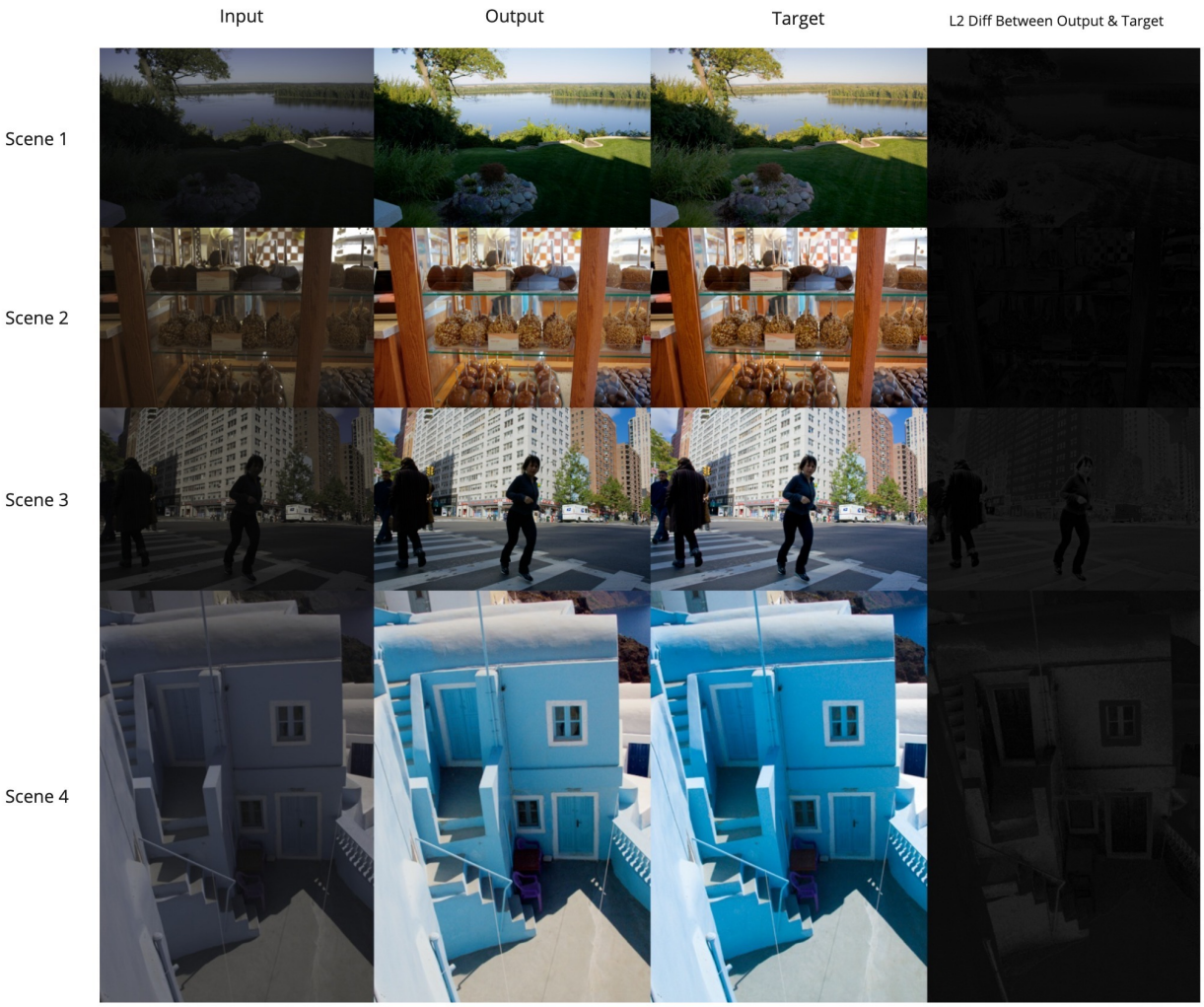

Fig. 3. Visual Results