New Methods for Image Enhancement and Camera ISP Learning

Zhetong Liang

PhD

The Hong Kong Polytechnic University

2021

The Hong Kong Polytechnic University

Department of Computing

New Methods for Image Enhancement and Camera ISP Learning

Zhetong Liang

A thesis submitted in partial fulfilment of the requirements

for the degree of Doctor of Philosophy

April 2021

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____(Signed)

ZHETONG LIANG (Name of student)

Abstract

The captured images by modern camera sensor are color-mosaicked signals which contain incomplete color information, noise, less vivid colors and improper tones. To reconstruct a high-quality displayable image, an image signal processing (ISP) pipeline is employed onboard a camera to enhance the captured raw images by a cascade of image processing components, including demosaicking, white balance, noise removal, color space conversion, tone mapping and detail enhancement. However, there are two challenges in designing an ISP pipeline. First, the individual components in an ISP pipeline may have limited performance due to simple design. Second, there could be limitations on the whole ISP pipeline, which are designed in a divide-and-conquer manner with error accumulation. In this thesis, we leverage new optimization and learning methods to tackle the two challenges.

To address the first challenge, we make several improvements on the design of individual image processing components. In the first work, we propose a new method for tone mapping component, which aims to convert a high dynamic range (HDR) image to a standard dynamic range image with improved perceptual quality. We design a hybrid ℓ_1-ℓ_0 norm optimization approach for tone mapping, and address the halo artifacts and over-enhancement problem in existing methods in the literatures. In the second work, we propose a deep-learning-based approach for single image denoising. Unlike the common end-to-end architecture, we adopt a two-stage convolutional neural network (CNN) architecture with smooth-first and enhance-later strategy.

The proposed architecture removes the noise in the first stage and hallucinates high-frequency details back to the image in the second stage by adversarial learning. The proposed method can produce detail-enriched results and outperforms the existing denoising methods in terms of perceptual quality on both synthetic and real-world noisy images. In the third work, we propose a novel learning scheme for real-world burst denoising which leverages multiple images. To apply deep learning to burst denoising, it is difficult to construct a dataset for this purpose because of the object motions in a scene. We bypass this obstacle by designing a decoupled learning method to leverage two complementary datasets. With the designed network and the decoupled learning scheme, we achieve leading performance in real-world burst denoising without the need of a real-world burst dataset for training.

To address the second challenge, we propose a data-driven framework for camera ISP learning. Different from the existing camera ISPs that rely on manual design of individual image processing components, we design a deep CNN as an ISP and train it with pairwise datasets to reconstruct high-quality displayable images from raw counterparts. The challenge for this work is to properly characterize the diverse image processing components inside an ISP. We tackle this problem by designing a two-stage CNN architecture, where image restoration related subtasks are addressed in the first stage and image enhancement related subtasks in the second stage. The proposed ISP model achieves high image quality and outperforms the state-of-the-art ISP learning methods on several publicly available benchmark datasets.

In summary, in this thesis, we present a novel tone mapping algorithm, and two deep CNN-based methods for image denoising and burst denoising, respectively. In addition, we present a data-driven framework for the ISP pipeline design.

Keywords: Image enhancement, Image restoration, Tone mapping, Image denoising, Burst denoising, Camera ISP

Biography

Journal Papers

1. **Zhetong Liang**, Jianrui Cai, Zisheng Cao, Lei Zhang. CameraNet: A Two-Stage Framework for Effective Camera ISP Learning. IEEE Transactions on Image Processing, 30:2248-2262, 2021

2. Feida Zhu, **Zhetong Liang**, Xixi Jia, Lei Zhang, Yizhou Yu. A Benchmark for Edge-Preserving Image Smoothing. IEEE Transactions on Image Processing, 28:3556-3570, 2019.

Conference Papers

1. **Zhetong Liang**, Hongyi Zheng, Gaofeng Ren, Lei Zhang. Smooth and Enhance: A Two-stage Network for Detail Enriched Image Denoising. **Submitted to** IEEE International Conference on Computer Vision (ICCV), 2021

2. **Zhetong Liang**, Shi Guo, Hong Gu, Huaqi Zhang, Lei Zhang. A Decoupled Learning Scheme for Real-World Burst Denoising from Raw Images. European Conference on Computer Vision (ECCV), 2020

3. **Zhetong Liang**, Jun Xu, David Zhang, Zisheng Cao, Lei Zhang. A Hybrid ℓ_1-ℓ_0 Layer Decomposition Model for Tone Mapping. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018

Acknowledgements

I would like to express my sincere gratitude towards the following people.

First and foremost, I want to express my gratitude to my supervisor, Prof. Lei Zhang, for his valuable guidance and continuous support. During the past four years of research, he always encouraged me to think differently and critically on my research. When I was confused on my research, he always provided insightful advices and helped me through the difficulties. Without his dedicated assistance and insightful supervision, this thesis would have gone nowhere. It is my great pleasure to be a student of Prof. Zhang.

I would also like to express my gratitude to Dr. Zisheng Cao, who is my mentor when I was an intern in the camera imaging group in DJI Co., Ltd. He shared with me his valuable insight and experience on developing industry-level image processing algorithms. I would also thank other extraordinary engineers in the DJI group, including Zhiqiang Li, Robert and Pan Hu.

I would like to thank Gaofeng Ren, who is my mentor during my internship in Alibaba DAMO Academy. He taught me valuable skills on developing Camera ISP algorithms. I would also thank other outstanding group mates in Alibaba.

I want to express my gratitude to my office mate for their help in research and life. I have spent a meaningful time with them in the Hong Kong Polytechnic University.

Finally, I want to thank my parents for their support and encouragement.

Table of Contents

CERTIFICATE OF ORIGINALITY iii

Abstract iv

Biography vi

Acknowledgements vii

List of Figures xii

List of Tables xvii

1 Introduction 1

1.1 Motivation 1

1.2 Literature Review 5

1.2.1 Tone Mapping 5

1.2.2 Single Image Denoising 6

1.2.3 Burst Denoising 8

1.2.4 Camera ISP Pipeline 9

1.3 Contributions and Thesis Organization 11

2 Tone Mapping by Hybrid ℓ_1-ℓ_0 Layer Decomposition 13

2.1 Introduction 14

2.2 Tone Mapping by Hybrid ℓ_1-ℓ_0 Optimization 16

2.2.1 Hybrid ℓ_1-ℓ_0 Layer Decomposition 16

2.2.2 Solver 18

2.2.3 Extension to Multiscale Decomposition 20
2.2.4 Tone Mapping Process 22
2.3 Experiments and Analysis 24
2.3.1 Parameter Selection 24
2.3.2 The Decomposed Layers 26
2.3.3 Comparison of Tone Mapping 27
2.3.4 Discussion and Future Work 31
2.4 Summary 32
3 A Smooth-and-Enhance Strategy for Detail-enriched Single Image Denoising 34
3.1 Introduction 35
3.2 Image Denoising by Smooth-and-Enhance Strategy 37
3.2.1 Smooth and Enhance: A Two-stage Network 38
3.2.2 Training Strategy 41
3.3 Experiments 45
3.3.1 Experimental Settings 45
3.3.2 Results on AWGN Noise Removal 46
3.3.3 Results on Real-world Noise Removal 50
3.3.4 Ablation Study 54
3.4 Summary 55
4 A Decoupled Learning Scheme for Real-world Deep Burst Denoising 57
4.1 Introduction 58
4.2 Decoupled Learning Network for Burst Denoising 60
4.2.1 Training Data Preparation 61
4.2.2 Decoupled Network Design 62

4.2.3 Decoupled Learning Process 64
4.3 Experiments 67
4.3.1 Datasets 68
4.3.2 Results on Synthetic Noisy Sequences 69
4.3.3 Results on Real-world Noisy Sequences 71
4.3.4 Ablation Study 74
4.4 Summary 76
5 A Two-stage Framework for General and Effective Camera ISP Learning 78
5.1 Introduction 79
5.2 Two-stage Camera ISP Framework 81
5.2.1 Two-stage Grouping 82
5.2.2 Two-stage Network Design 84
5.2.3 Ground Truth Generation 87
5.2.4 Two-step Training Scheme 90
5.3 Experiments 91
5.3.1 Dataset Setting 92
5.3.2 Experimental Setting 93
5.3.3 Ablation Study 94
5.3.4 Comparison with Recent Learning-based ISP 99
5.3.5 Cross-dataset Testing 104
5.3.6 Comparison with Traditional ISP 105
5.3.7 Computational Complexity 106
5.3.8 Limitations 107
5.4 Summary 107

6 Conclusions and Future Work **109**

6.1 Conclusions 109

6.2 Future Work 111

Bibliography **114**

List of Figures

1.1 Major components in a traditional camera image signal processing pipeline. 9

1.2 Contributions and organizations of the thesis. 11

2.1 Results by the proposed layer decomposition. The 1-D analysis is performed on detail layers, which are illustrated on (b) and (c). The location is on the line drawn in (a). 16

2.2 The proposed two-scale tone mapping algorithm 20

2.3 Tone mapping results by one scales and two scales 21

2.4 The effect of λ_2 on the detail layer when λ_1 is fixed to 0.3. 24

2.5 The effect of λ_1 on the two layers when λ_2 is fixed to $0.01\lambda_1$. The upper three images are detail layers. The bottom three images are base layers. 25

2.6 Comparison of multiscale decomposition models 26

2.7 1-D analysis of multiscale decomposition 27

2.8 Comparison of tone mapping 28

2.9 Comparison of tone mapping 28

2.10 Comparison with Photomatix 29

2.11 Comparison of mean opinion score statistics 30

3.1 Network structure. 38

3.2 Fade-in layers. The green and orange blocks denote the modules from DE-Net and discriminator, respectively. ToRGB and FromRGB denotes 1×1 convolutions that reduce and expand the channel size, respectively. 41

3.3 The Gaussian denoising results ($\sigma = 50$) on an image from the McMaster testing set. 47

3.4 The Gaussian denoising results ($\sigma = 70$) on an image from the Urban100 testing set. 48

3.5 The Gaussian denoising results ($\sigma = 70$) on an image from the CBSD68 testing set. 49

3.6 Real-world denoising results of the compared methods on SIDD validation set. Best viewed on screen with zoom-in. 50

3.7 Real-world denoising results of the compared methods on vivo testing set. The images from left to right are (a) Noisy raw image. (b) Noisy patches. (c) UTR[15]. (d) CBDNet[52]. (e) RIDNet[4]. (f) SADNet[23]. (g) SAE-Net. All denoised raw images are transformed to sRGB space for display. Best viewed on screen with zoom-in. . . . 51

3.8 Real-world denoising results of the compared methods on vivo testing set. The images from left to right are (a) Noisy raw image. (b) Noisy patches. (c) UTR[15]. (d) CBDNet[52]. (e) RIDNet[4]. (f) SADNet[23]. (g) SAE-Net. All denoised raw images are transformed to sRGB space for display. Best viewed on screen with zoom-in. . . . 51

3.9 Comparison of human opinion scores on Real-dynamic test set. 53

3.10 The AWGN removal results ($\sigma = 50$) between different learning schemes on two images from CBSD68 testing set. The top and bottom rows are cropped patches of images "163085" and "105025" from CBSD68, respectively. 55

4.1 The decoupled learning framework for our burst denoising network (BDNet). 63

4.2 The structure of the PreP module M_p, TemP module M_t and PostP module M_o of the proposed BDNet. 63

4.3 The denoising results of the compared methods on Vimeo-200 test set with Gaussian noise σ=50. 69

4.4 The denoising results of the compared methods on Vimeo-200 test set with Poisson-Gaussian noise. 70

4.5 Denoising Results of the compared methods on Real-static test set. White balance gain and a gamma conversion with parameter 2.2 are applied for better visualization. 71

4.6 The denoising results of the compared methods on Real-static test set. 72

4.7 Denoising results of the compared methods on Real-dynamic test set. (a) Noisy reference frame. (b) Noisy patches. (c) M-RIDNet. (d) KPN. (e) INN. (f) BDNet. White balance gain and a gamma conversion with parameter 2.2 are applied for better visualization. Best viewed on screen with zoom-in. 73

4.8 The denoising results of the compared methods on Real-dynamic test set. 74

4.9 Illustration of different learning schemes for real-world burst denoising with dynamic scenes. Please refer to the text for detailed descriptions. 75

4.10 The results on a raw image sequence with large noise in Real-static test set by different learning schemes. (a) Noisy patch. (b) BDNet-ft. (c) BDN-at. (d) Default BDNet. (e) Ground-truth. White balance gain and a gamma conversion with parameter 2.2 are applied for better visualization. 76

4.11 The results on an raw image sequence with large noise by different learning schemes. (a) Noisy reference frame. (b) Noisy patch. (c) BDNet-ft. (d) BDN-at. (e) Defualt BDNet. White balance gain and a gamma conversion with parameter 2.2 are applied for better visualization. Best viewed on screen with zoom-in. 76

5.1 The image histogram changes caused by different image processing operations, including demosaicking, denoising (with $\sigma = 15$ and 25), $4\times$ super-resolution, local contrast enhancement [114] and global tone mapping [102], on images in the BSD100 dataset [85]. The vertical axis denotes the ℓ_1 norm of histogram differences, while the horizontal axis denotes the image index in BSD100. 83

5.2 The proposed CameraNet system for ISP learning. 84

5.3 The structure of UNet-like Restore-Net and Enhance-Net modules in the proposed CameraNet system. 85

5.4 The workflow of creating restoration and enhancement ground truths using Adobe software. The restoration ground truth is created in Adobe Camera Raw, while the enhancement ground truth is created in Lightroom. The dots in restoration operations refer to other possible restoration tasks such as aberration correction and deblurring, while the dots in enhancement operations refer to other possible adjustment of image features. 88

5.5 Illustration of the two-stage network outputs and ground truths. The columns from left to right are (a) Raw images. (b) Results by Restore-Net. (c) Restoration ground truth. (d) Results by Enhance-Net. (e) Enhancement ground truth. The image in the first row is from the HDR+ dataset [53], while the image in the second row is from the FiveK dataset [18]. A gamma transform with parameter 2.2 is applied to the raw images and restoration ground truths for display. 89

5.6 Results by one-stage, two-stage and three-stage CNN models. The two sets of images are from the SID dataset [24]. A gamma transform with parameter 2.2 is applied to the raw images and restoration ground truths for display. 94

5.7 Comparison between the default training setting and the setting without step 1. The image is from the HDR+ dataset [53]. 96

5.8 Comparison between the default training setting and the setting without step 2. The image is from the HDR+ dataset [53]. 96

5.9 Comparison between the results with and without perceptual loss (p.l.). We add the p.l. on the enhanced image in the fine-tuning step with weight 0.01. The image is from the HDR+ dataset [53]. 97

5.10 Comparison between SRGAN+CAN24 and CameraNet. A gamma transform with parameter 2.2 is applied to the raw images and restoration ground truths for display. 98

5.11 Results on a church image from the HDR+ dataset [53] by the competing methods. A gamma transform with parameter 2.2 is applied to the raw image for better visualization. 99

5.12 Results on a pavilion image from the SID dataset [24] by the competing methods. A gamma transform with parameter 2.2 is applied to the raw image for better visualization. 100

5.13 Results on a pavilion image from the SID dataset [24] by the competing methods. A gamma transform with parameter 2.2 is applied to the raw image for better visualization. 101

5.14 Results on a flower image from the FiveK dataset [18] by the competing methods. A gamma transform with parameter 2.2 is applied to the raw image for better visualization. 102

5.15 Cross-dataset testing. First row: the results of transferring the CameraNet trained on FiveK dataset to the testing image from HDR+ dataset. Second row: the results of transferring the CameraNet trained on HDR+ dataset to the testing image from FiveK dataset. "Full transfer" means transferring the whole CameraNet, while "Enhance-Net transfer" means transferring only the Enhance-Net. 104

5.16 Comparison with Sony A7S2 ISP in low-light scenarios. Both of the raw images are captured with aperture f3.5, exposure time 1/100s and ISO 12800. A gamma transform with parameter 2.2 is applied to the raw image for better visualization. 105

List of Tables

2.1 Comparison of mean opinion scores 29

2.2 Comparison of TMQI 30

2.3 Comparison of running time on a 1333×2000 image 31

3.1 Gaussian denoising results in PI/LPIPS. The values in bold letter indicate the best scores."-" means that the result is not available. . . 46

3.2 Gaussian denoising results in PSNR(dB). The values in bold letter indicate the best scores."-" means that the result is not available. . . 47

3.3 Quantitative results of Real-world denoising. For SIDD validation set with ground truths available we evaluate PSNR/PI/LPIPS, while for vivo testing set without ground truth we evaluate PI. The values in bold letter indicate the best results. "-" means the result is not available. 50

3.4 Quantitative results (PI/LPIPS) of different variants of SAE-Net on AWGN removal with $\sigma = 50$. 54

4.1 Quantitative results (PSNR/SSIM) on the synthetic test sets. G25, G50 and PG indicates Gaussian σ=25, Gaussian σ=50 and Poisson-Gaussian noise, respectively. 69

4.2 Quantitative evaluation on the Real-static test set. 71

4.3 Quantitative results (PSNR/SSIM) of different learning schemes on the Real-static test set. 75

5.1 Ablation study on the HDR+ and SID datasets. The best and second best scores are highlighted in red and blue for each column. 95

5.2 Objective comparison of different learning-based ISP methods. 99

5.3 Over-fitting evaluation. This table compares the training and testing losses of the last epoch for each training step on the three datasets. The "Gap" means the difference between the testing loss and the training loss. 103

5.4 Computational complexity of the compared CNN models. The GFLOPS and running time are evaluated on an image of resolution 4032×3024. 106

Chapter 1

Introduction

1.1 Motivation

Digital imaging devices become increasingly important in our daily life. Smartphone cameras are used by people to record the splendid moments of life, while surveillance cameras are used to guarantee the securities of the society. This phenomenon raises the needs to improve the quality of the captured images by these cameras. However, due to the limitations of the camera hardware design, the captured images could go through different levels of degradations. For example, most camera deploys a color filter array on top of the sensors, and thus, only one color channel can be recorded in one pixel location. In addition, the small aperture and small CMOS sensor limit the amount of collected light and lead to heavy noise corruption. As a result, the raw image data captured by camera sensors are typically color-mosaiced irradiance signals containing noise, incorrect colors, loss of details and high dynamic range [96, 62].

To reconstruct a high-quality displayable image for viewing, many cameras employ an image signal processing (ISP) pipeline to process and enhance the sensor raw data. Such pipeline is a cascade of image processing components, which typically include image demosiacking, noise removal, white balance, color space conversion, tone mapping and detail enhancement. Despite the many efforts to develop the ISP pipeline, there are some problems that remain unsolved. For one thing, in the

traditional ISP design, the individual processing components are hand-crafted simple algorithms. This arrangement could lead to unsatisfactory image quality, especially in challenging imaging scenarios such as nighttime and in high dynamic range (HDR) environment. For another, the whole ISP pipeline design could cause limitation on the image quality because the individual components are developed independently, which results in error accumulation and unnecessary long development period. In this thesis, we aim to develop new methods for some image processing components as well as the whole ISP pipeline.

As an image processing component in an ISP pipeline, tone mapping aims to reproduce a standard dynamic range (SDR) image for display from the sensor HDR image. Tone mapping operation plays an important role in the image perceptual quality. The traditional tone mapping component in an ISP pipeline usually adopts a global adjusting curve and results in loss of details in the tone mapped image. On the other hand, the tone mapping algorithms proposed in the literatures usually generate various artifacts, including halo artifacts and over-enhancement artifacts. One may improve tone mapping by adopting the deep learning technique, which has recently demonstrated good performance on several image processing tasks. However, generating the ground truth is a labor-intensive and highly subjective process, and this causes instability in the network training. Thus, we propose a hybrid ℓ_1-ℓ_0 optimization model for tone mapping. We use ℓ_1 sparsity to preserve the edge regions and prevent the halo artifacts, while adopting ℓ_0 sparsity to characterize the image details, which improves the overall naturalness of the image. We collect a large HDR image database and compare with several state-of-the-art tone mapping algorithms. Experiments show that our tone mapping algorithm can achieve excellent details reproduction while avoiding halo and over-enhancement artifacts.

As an important ISP component, single image denoising aims to remove various noise introduced in the imaging process. Nowadays, image denoising becomes in-

creasingly important for smartphone cameras which collect insufficient light under challenging environment due to the use of small lens and sensors. The noise removal components in commercial cameras usually adopt filter-based method and result in residual noise. To improve the denoising performance, there are many works in the literatures that train deep convolutional neural networks (CNN) on large scale datasets with noisy-clean image pairs. Though these deep-learning-based methods show notable advantages over traditional ones, they usually produce over-smooth results. This is because noise and image details are entangled in the high-frequency domain of an image, and the operation of removing noise inevitably leads to detail loss. To address this problem, we propose a two-stage denoising network with a smooth-first and enhance-later strategy. In the first stage, a noise reduction subnetwork conducts normal denoising on the noisy images and produces smooth but clean images. Then in the second stage a detail enhancement subnetwork hallucinates high-frequency details on the output of the first stage and produces detail-enriched results. A sophisticated adversarial training scheme is applied to train the network to generate realistic details. The proposed method outperforms the recently proposed deep-learning-based denoising methods by a large margin in terms of perceptual quality metrics on both synthetic Gaussian noise and real-world noise.

Burst denoising is an advanced imaging technique of modern smartphone cameras which captures multiple noisy images of a dynamic scene and combines them into a clean image. Typical algorithm procedures include pre-denoising, frame alignment and image fusion. In view of the recent success of deep learning technique, it is much desirable to apply deep-learning-based method to real-world burst denoising. Specifically, we can train a deep CNN to learn aligning images with object motions and removing real-world noise from data. However, it is difficult to construct a real-world burst denoising dataset for this purpose due to the presence of object motions. We propose a decoupled learning scheme which leverages two complementary

datasets to learn real-world burst denoising. The first dataset is a video dataset which contains dynamic sequences corrupted by synthetic noise. The other dataset is a burst dataset with static scenes and real-world noisy sequence. Our decoupled learning algorithm can learn the frame alignment from the dynamic sequences in the video dataset and learn the adaptation to real-world noise statistics in the static burst dataset. Experiments demonstrate that through our decoupled learning scheme, a burst denoising network can effectively tackle the real-world dynamic noise sequences without the need to construct a real-world burst denoising dataset.

Last but not least, we make improvement on the ISP pipeline as a whole. The traditional ISP pipeline is designed in a divide-and conquer manner where the image processing components are developed independently. Such a design method could lead to error accumulation since each component scarcely considers the previous and the following components. Moreover, the development of ISP pipeline will go through a long period because each image processing components requires painstaking parameter tuning. To address the above drawbacks of ISP design, we propose a general and effective framework which is based on deep-learning technique. Specifically, we first analyze the individual components in an ISP and divide them into two weakly correlated groups, i.e., image restoration and enhancement. Image restoration components aim to faithfully reconstruct the detail information of scene irradiance, while the objective of style enhancement components is to improve the visual appearance of images. We propose a CNN architecture for ISP pipeline which characterizes the two groups with two subnetworks, respectively. This arrangement allows collaborative processing of correlated ISP subtasks while avoiding mixed treatment of weakly correlated subtasks, leading to high quality image reconstruction in various imaging scenarios.

1.2 Literature Review

In this section, we review on the previous research on image tone mapping, single image denoising, burst denoising and ISP pipeline.

1.2.1 Tone Mapping

Existing tone mapping algorithms can be categorized into global methods and local methods. Global tone mapping methods reproduce a LDR image by applying a single nonlinear curve to the radiance map [110, 117, 101]. Because of the spatially invariant nature, global methods are lack of detail preservation. In contrast, local tone mapping algorithms with a spatially variant property are the main stream of tone mapping techniques [102, 37, 75, 86, 49, 41]. Local methods are commonly based on layer decomposition, where the base layer is first estimated by edge preserving filter and detail layer is the residual between base layer and the original image. Different local tone mapping algorithms mainly differ in the filter design. Reinhard *et al*. proposed to use a Gaussian-based filter with a spatially adaptive scale parameter [102]. Durand *et al*. adopted a bilateral filter to estimate the base layer [37]. Although this method can avoid halo artifacts to some extent, it over-enhances the image by boosting the small-scale details. Meylan proposed a Retinex-based adaptive filter tone mapping [86]. Gu *et al*. proposed a weighted guided filter for tone mapping [49]. Many local tone mapping methods also have halo artifact or over-enhancement problem due to the improper characterization of image structures.

Tone mapping task is also related to the research of edge-preserving filtering. The earliest edge-preserving filter is bilateral filter that considers local range variation of the image [39]. Min proposed a fast global smoother based on weighted least square [88]. Other representative filters are Xu's ℓ_0-based filter [120] and Bi's ℓ_1-based filter [12]. While most filters imposed strong edge-preserving prior to avoid halo

artifacts, they lack a naturalness prior on the visual appearance of the images and lead to over-enhancement artifact in the results.

Deep learning technique has been successfully applied to various image processing tasks, including image super-resolution and denoising. When applied to tone mapping, there needs a large-scale pairwise datasets with HDR images and the corresponding ground truth (GT) tone mapped images. The difficulty lies in the creation of GT, which requires labor-intensive tuning and subjective evaluation of image quality. Recently Rana *et al*. proposed a deep learning approach for tone mapping and created the GTs by selecting the results from several traditional tone mapping algorithms [99]. In such a compromised scheme, the tone mapping quality is bounded by the performance of the traditional algorithms. Since it is a complex process to construct an ideal tone mapping dataset, in this thesis we do not consider deep learning method and leave it to future work.

1.2.2 Single Image Denoising

The image denoising has been widely studied in the academia and can be divided into synthetic Gaussian denoising and real-world denoising. Additionally, the image denoising method proposed in our thesis is related to adversarial image restoration.

Gaussian denoising. As a classical and fundamental problem in image processing, image denoising has been widely studied in the academia, while most of the methods employ the additive Gaussian white noise (AWGN) model. Traditional methods heavily rely on hand-crafted image priors, including statistical prior [104, 94], sparsity prior [3, 40, 84], non-local self-similarity [17, 33, 50, 119, 82], etc. Recently, deep learning based methods [79, 132, 23, 4] have significantly improved the Gaussian denoising performance by learning image priors from data. These methods train deep CNN models with different architectures on large-scale datasets that contain noisy-clean image pairs. In the seminal work of DnCNN [131], Zhang *et al*. demonstrated

that a ResNet-like [55] model can outperform the traditional hand-crafted methods by a large margin. NLRN [79] and RNAN [137] models incorporate the self-similarity prior and achieve further improvement. RIDNet [4] incorporates the attention mechanism into the model, which adaptively controls the relative importance among feature maps.

Real-world Image Denoising. The research on real-world denoising has not been widely investigated until recently. The real-world noise in raw images has much more complex statistics than AWGN, and it is further complicated by the camera image signal processing (ISP) pipeline. Due to the lack of large-scale datasets of real-world noisy images and their noise-free counterparts, researchers proposed to synthesize noisy raw images by reversing the ISP pipeline on high quality clean sRGB images and adding Poisson-Gaussian noise on the raw images [15, 130, 52]. In this way, a large amount of noisy and clean image pairs can be synthesized to train deep denoising models. Different methods differ mainly on the modeling of ISP pipeline. The unprocessing-to-raw method [15] employs simple parametric models to characterize several important ISP operations, including demosaicking, white balance, color conversion and tone mapping. The CycleISP method [130] trains a CNN model to reverse the ISP pipeline, which can account for some complex ISP operations. Some researchers collected real-world datasets for denoising by using different approaches to generate ground truth clean images [1, 24]. The SID dataset captures long-exposed images as clean ground truths [24], while the SIDD dataset averages a large number of noisy raw images as ground truths [1].

Adversarial Image Restoration. Generative adversarial network (GAN) [47] is a generative modeling method which learns to sample realistic data by alternatively training a generator and a discriminator network with different objectives. The generator learns to sample real data to fool the discriminator, while the discriminator learns to differentiate between real and the synthetic data. Many subsequent works

of GAN focus on improving its training stability [5, 89, 61, 6, 64], e.g., by adopting different loss functions [6, 61], progressive learning [64] and regularizations [51].

Due to the capability of generating realistic data, GAN has been successfully applied to image translation [58, 142, 30, 139], super-resolution (SR) [69, 116, 111, 20, 29, 113], image inpainting [126] and raindrop removal [74], etc. Notable progress has been witnessed in SR where high-resolution details need to be recovered. Ledig *et al.* [69] trained a SR network with an additional GAN loss, where a discriminator network distinguishes real high-resolution images from the super-resolved ones. Though the SR results obtain lower objective metrics (e.g., PSNR), higher perceptual quality is obtained. Wang *et al.* [116] proposed to use relativistic GAN loss [61] for SR, where the discriminator estimates the relativistic rather than absolute authenticity between real and fake samples. Recently, GAN has been applied to noise modeling [25, 128]. However, there is little work that successfully applies adversarial training to improve the visual quality of denoised images, which is our goal in this work.

1.2.3 Burst Denoising

Burst denoising methods capture a noisy image sequence as input, and perform a series of operations, including frame alignment, temporal fusion and post-processing, to reproduce the underlying scene [143, 53, 32, 82, 78, 16]. The frame alignment operation aims to build the correspondence between the dynamic contents of the target and reference frames. Some works adopt block matching for alignment [53, 32, 82, 33, 143], while others use optical flow methods [78, 16]. The fusion operation aims to merge the outputs from multiple frames, which should be robust to alignment error. Representative approaches include collaborative filtering [33], non-local means [16] and frequency domain fusion [53].

Recently, a few works have been proposed to learn frame alignment and fusion from the input sequences for burst denoising. The KPN model proposed by Mildenhall

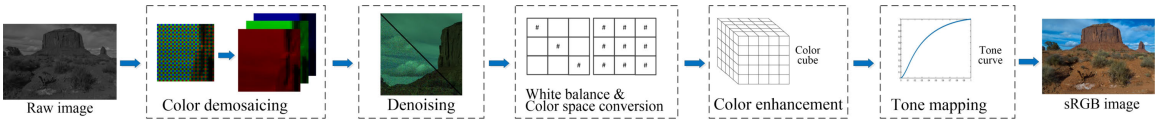


Figure 1.1: Major components in a traditional camera image signal processing pipeline.

et al. [87] predicts the convolutional kernels to selectively fuse a burst of images with object motion. Xue *et al*. [121] designed a CNN model that explicitly consists of frame alignment, fusion and post-processing modules. Godard *et al*. [46] proposed a recurrent architecture for burst denoising, which can increase the image quality by accumulating noisy images. These learning-based methods achieve better image quality than their non-learning counterparts.

Despite the recent success, it is difficult to apply deep-learning-based burst denoising method to real-world scenario where the noise statistics is intricate. This is because it is difficult to construct a real-world burst denoising dataset in the presence of scene motions. As a remedy, several works have been reported to synthesize realistic data for burst denoising [15, 52, 87, 38]. Tim *et al*. [15] and Guo *et al*. [52] proposed to reverse the ISP pipeline on the sRGB images and generate noisy training images that are close to the camera raw data. Mildenhall *et al*. [87] proposed to synthesize noise and motions for burst denoising. Ehret *et al*. [38] proposed to train a CNN model for synthetic noise, and then fine-tune it on the camera raw images without clean ground-truth based on the Noise2Noise principle [72]. However, these methods are compromised schemes which cannot cope with the real-world scenes with heavy noise corruption and object motion.

1.2.4 Camera ISP Pipeline

There exist various types of image processing components inside the ISP pipeline of a camera. The major ones include demosaicking, noise reduction, white balancing, color space conversion, tone mapping and color enhancement, as shown in Fig. 1.1.

The demosaicking operation interpolates the single-channel raw image with repetitive mosaic pattern (e.g., Bayer pattern) into a full color image [134, 97, 73], followed by a denoising step to enhance the signal-to-noise ratio [33, 50, 119, 82]. White balancing corrects the color that is shifted by illumination according to human perception [28, 57, 13, 10]. Color space conversion usually involves two steps of matrix multiplication [63]. It firstly transforms the raw image in camera color space to an intermediate color space (e.g., CIE XYZ) for processing and then transforms the image to sRGB space for display. Tone mapping compresses the dynamic range of the raw image and enhances the image details [125, 102, 37]. Color enhancement operation manipulates the color style of an image [45, 31, 76, 127], usually in the form of 3D lookup table (LUT) search. A detailed survey of the ISP components can be found in [96, 62].

In the design of a traditional ISP pipeline, each algorithm component is usually developed and optimized independently without knowing the effect to its successors. This may cause error accumulation along the algorithm flow in the pipeline [56]. Moreover, each step inside an ISP pipeline is characterized by simple algorithms which are not able to tackle the challenging imaging requirement by ubiquitous cellphone photography. Recently, there are a few works that apply the learning-based approach to the ISP pipeline design [100, 105]. One pioneering work of this type is Jiang *et al.* 's affine mapping framework [59]. In this work, the raw image patches are clustered based on simple features and then a per-class affine mapping is learned to map the raw patches to the sRGB patches. This learning-based approach has limited regression performance due to the use of simple parametric model. Chen *et al.* proposed a multiscale CNN for nighttime denoising [24]. They constructed a denoising dataset and the CNN model is trained to convert a noisy raw image to a clean sRGB image. Schwartz *et al.* proposed a CNN architecture called DeepISP that learns to correct the exposure [105]. One common limitation of Chen's and Schwartz's models

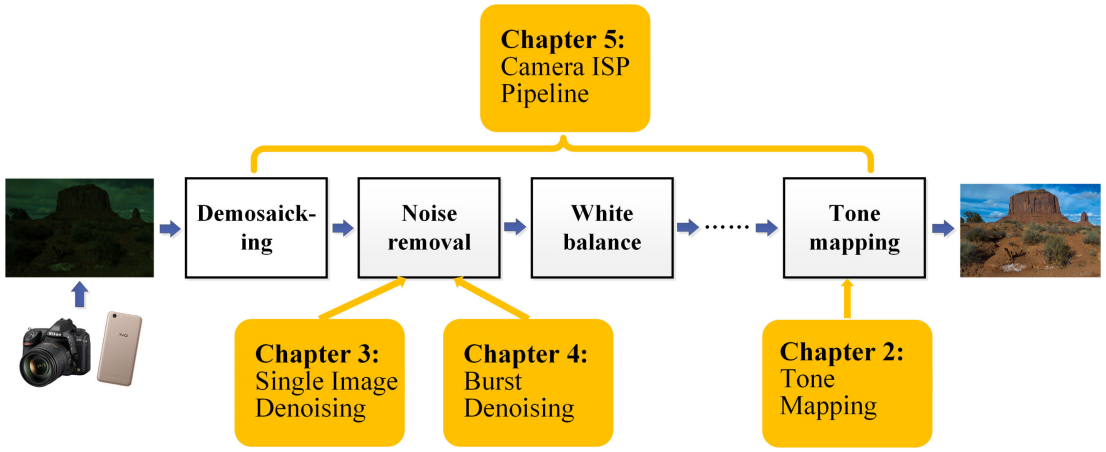


Figure 1.2: Contributions and organizations of the thesis.

is that they only considered one single aspect of ISP pipeline. Their task-specific architectures are not general enough to model the various components inside an ISP pipeline.

There exist a few datasets with different imaging scenarios that can be used for ISP pipeline learning [53, 24, 18]. These datasets contain raw images and the corresponding groundtruth sRGB images that are manually processed and retouched in a controlled setting. The HDR+ dataset is featured with burst denoising and sophisticated style retouching [53]; the SID dataset is featured with nighttime denoising [24]; and the FiveK dataset contains groundtruth images that are retouched by five photographers to have different color styles [18].

1.3 Contributions and Thesis Organization

The contributions and organizations of the thesis are summarized in Fig. 1.2, and are described in detail in the following.

In chapter 2, we propose a tone mapping algorithm which adopts a hybrid ℓ_1-ℓ_0 norm optimization to characterize different features of an image. Our tone mapping algorithm achieves visually compelling results and outperforms state-of-the-art tone

mapping algorithms in terms of artifact prevention.

In chapter 3, we propose a novel denoising network for detail-enriched image denoising. We adopt a smooth-and-enhance strategy where the network smooths the input noisy image first and then hallucinates high-frequency textures to enhance the perceptual quality. Adversarial training technique is applied for realistic details generation. Our denoising network is shown to outperform the recently proposed deep-learning-based denoising methods by a large margin in terms of perceptual quality.

In chapter 4, we propose a practical learning scheme for burst denoising in real-world scenarios. While it is difficult to construct a real-world dataset for burst denoising, our learning scheme leverages two existing datasets to learn the required operations in burst denoising, including frame alignment and real-world noise adaptation. Our burst denoising method trained by the proposed learning scheme can be applied to real-world burst denoising and outperforms other burst denoising methods in the literatures.

In chapter 5, we propose a general and effective framework for ISP pipeline learning. We analyze the typical image processing components in an ISP and divide them into two weakly correlated groups. A two-stage network is designed to learn the two groups of operations by two ground truths. Our two-stage framework is shown to have great advantages over the traditional ISP pipeline and outperforms several deep-learning-based ISPs in the literatures.

In chapter 6, we summarize our works and discuss the future research work.

Chapter 2

Tone Mapping by Hybrid ℓ_1-ℓ_0 Layer Decomposition

Tone mapping is an important image enhancement algorithm in the image signal processing (ISP) pipeline of a camera. It aims to transform the high dynamic range (HDR) raw image to a standard dynamic range (SDR) image with good visual quality for display. This is usually achieved by reducing the luminance dynamic range of an image while boosting the image details. Most existing tone mapping algorithms in the literatures are based on layer decomposition method, where the image is decomposed into a base layer and a detail layer. The base layer is compressed and the detail layer is enhanced, followed by a recombination of the two layers to obtain the tone mapped image. However, the tone mapped images usually contain halo artifacts or over-enhancement artifacts, due to poor characterization of the two layers. In this chapter, we propose a hybrid ℓ_1-ℓ_0 optimization model as the layer decomposition for tone mapping. The proposed optimization model uses ℓ_1 and ℓ_0 sparsities to characterize the base and detail layer, respectively, and leads to excellent tone mapping quality with minimal artifacts.

2.1 Introduction

The real-world scenes could span a luminance dynamic range that significantly exceeds the response range of most imaging devices. Thanks to the celebrated high dynamic range(HDR) technique in the past decade, the intact information of the scene can be recorded in a radiance map by bracketed exposure fusion technique [35, 9]. However, most of the display devices have a limited dynamic range and fail to reproduce the information in the radiance map faithfully. Therefore, an effective tone mapping algorithm is needed to transform the HDR radiance map into a standard dynamic range(SDR) image without sacrificing the main visual information.

In the past two decades, a large number of tone mapping techniques have been proposed in the literature. Despite the diversity in the design methodology, a large part of these tone mapping methods are based on layer decomposition [102, 37, 86, 49]. Specifically, an image is decomposed into a base layer and a detail layer and then processed separately. The detail layer with fine-grain details is preserved or boosted [37, 49], and the base layer with large spatial smoothness and high range variations is compressed. Although most layer-decomposition-based tone mapping algorithms could increase the visual interpretability of a radiance map to some extent, they have limitations in obtaining natural and visually pleasing results. A typical problem is over-enhancement, where small scale textural details dominate the image. This is because the existing works commonly ignore the spatial property of the detail layer, which has a significant impact on the tone mapped image. In addition, halo artifacts are still a problem in some tone mapping algorithms due to the lack of edge-preserving property for the base layer [49]. In order to obtain a natural and artifact-free reproduction of the radiance map, some proper priors should be incorporated into the layer decomposition framework.

Given the fact that a tremendous amount of information is recorded in a HDR

radiance map, which part of the information should be assigned a high priority for visual perception is an important question for tone mapping. In psychology, it was found that human vision is more sensitive to edges [7, 48]. This visual mechanism facilitates the capturing of the main semantic information of the scene. In intrinsic decomposition research [22, 12], it is commonly assumed that the edges in the reflectance layer(a concept similar to the detail layer) is sparse, which also indicates the high importance of the structural information in an image. In view of the above observations, a tone mapping operator should address the structural reproduction in the first place. Since the spatial property of the detail layer in the layer decomposition framework mostly affects the visual appearance of the tone mapped image, we consider to impose a structural sparsity prior on the detail layer.

While the use of prior on detail layer has not been reported in tone mapping research, the ℓ_1 sparsity prior has long been adopted in Retinex decomposition [90, 43] to model the structural sparsity of the reflectance layer. However, although the ℓ_1 term preserves edges in an image, its piecewise smoothness nature leads to weak structural prior. On the other hand, the ℓ_0 sparsity term has been shown to have great piecewise flattening property [120]. Thus, the ℓ_0 term seems to be a better choice for the structural prior.

We propose a hybrid ℓ_1-ℓ_0 layer decomposition model for tone mapping. Specifically, a ℓ_0 gradient sparsity term is imposed on detail layer to model the structural prior. In this way, the detail layer mostly contains structural information, which is enhanced later in the algorithm. Additionally, to prevent the halo artifacts, an ℓ_1 gradient sparsity term is imposed on the base layer to preserve edges. Then, we devise an effective multiscale tone mapping scheme based on our decomposition model. Due to the use of proper priors in our layer decomposition, our tone mapper outperforms state-of-the-art tone mapping algorithms in visual quality.

(a) Radiance map (b) Detail layer by ℓ_1-ℓ_0 model (c) Detail layer by ℓ_1-ℓ_1 model

(d) Result by ℓ_1-ℓ_0 model (e) Result by ℓ_1-ℓ_1 model (f) 1-D analysis

Figure 2.1: Results by the proposed layer decomposition. The 1-D analysis is performed on detail layers, which are illustrated on (b) and (c). The location is on the line drawn in (a).

2.2 Tone Mapping by Hybrid ℓ_1-ℓ_0 Optimization

2.2.1 Hybrid ℓ_1-ℓ_0 Layer Decomposition

To devise a suitable layer decomposition framework, we mainly model the structural prior of the detail layer and the edge-preserving prior on the base layer. Denote by $\boldsymbol{S}$, $\boldsymbol{B}$ and $\boldsymbol{S}-\boldsymbol{B}$ the original image, the base layer, and the detail layer, respectively. The proposed layer decomposition optimization model is as follows:

$$\begin{aligned}\min_{\boldsymbol{B}} \sum_{p=1}^{N} \Big\{ &(\boldsymbol{S}_p - \boldsymbol{B}_p)^2 + \lambda_1 \sum_{i=\{x,y\}} |\partial_i \boldsymbol{B}_p| \\ &+ \lambda_2 \sum_{i=\{x,y\}} F\big(\partial_i(\boldsymbol{S}_p - \boldsymbol{B}_p)\big) \Big\}\end{aligned} \tag{2.1}$$

where p is the pixel index, N is the number of pixels in the image. The first term $\sum_{p=1}^{N}(\boldsymbol{S}_p - \boldsymbol{B}_p)^2$ forces the base layer to be close to the original image. The spatial property of the base layer is formulated as a ℓ_1 gradient sparsity term $|\partial_i \boldsymbol{B}_p|$, $i = x, y$, where ∂_i is the partial derivative operation and $i = x, y$ denote the horizontal and

vertical axes, respectively. The spatial property of the detail layer is formulated as a ℓ_0 gradient sparsity term with an indicating function $F(x)$:

$$F(x) = \begin{cases} 1, & x \neq 0 \\ 0, & x = 0 \end{cases} . \tag{2.2}$$

The merits of our layer decomposition model lie in the hybrid usage of the ℓ_1 and the ℓ_0 terms. For one thing, due to the outlier-rejection nature of ℓ_1 sparsity term [80], the large gradient of the base layer is preserved. Thus, the base layer is piecewise smooth. For another, it has been shown that the ℓ_0 sparsity term yields flattening effects [120, 91]. Our model applies ℓ_0 term to force some small textural gradients of the detail layer to be zeros, while leaving the main structural gradients intact. This arrangement yields piecewise constant effect and successfully models the structural prior, as depicted in Figs 2.1(b).

Another possible choice for the detail layer is ℓ_1 gradient sparsity term, which has been reported in Retinex research [43, 90]. In Fu's model [43], the ℓ_1 term is imposed on the reflectance/detail layer to gain piecewise constant effect. However, the ℓ_1 term has two drawbacks. First, its nature is piecewise smoothness [81] and is not effective to produce piecewise constant result, as depicted in Figs 2.1(c). Second, under the same parameter setting, the ℓ_1 term does not strongly regularize the detail layer, which could lead to over-enhancement of the tone mapped image, as shown in Figs 2.1(e). To illustrate the difference between the ℓ_1 term and ℓ_0 term, the 1-D signals extracted from their resultant detail layers are shown in Figs 2.1(f). The position of the signal is the yellow line in Figs 2.1(a). We can see that the ℓ_0 term flattens the small trivial variations and preserves visually important edges, whereas the ℓ_1 term is not effective in such mechanism. As a result, the use of ℓ_0 term avoids the over-enhancement problem and increases the visual interpretability of an image, as shown in Figs 2.1(d).

2.2.2 Solver

The optimization model (2.1) is nonconvex due to the ℓ_0 norm. We adopt the ADMM framework to solve our objective function. For the sake of clarity, firstly the objective function (2.1) is rewritten in a matrix-vector form as:

$$\min_{\boldsymbol{b}} \tfrac{1}{2}\|\boldsymbol{s}-\boldsymbol{b}\|_2^2 + \lambda_1\|\nabla\boldsymbol{b}\|_1 + \lambda_2\mathbf{1}^\top F(\nabla(\boldsymbol{s}-\boldsymbol{b})), \tag{2.3}$$

where $\boldsymbol{s}, \boldsymbol{b} \in \mathbb{R}^N$ are the concatenated vector form of $\boldsymbol{S}, \boldsymbol{B}$ in (2.1), respectively, and $\mathbf{1} \in \mathbb{R}^{2N}$ is a vector of all ones. ∇ denotes the concatenation of two gradient operator matrices $\nabla = [\nabla_x^\top, \nabla_y^\top]^\top \in \mathbb{R}^{2N\times N}$. $F(\nabla(\boldsymbol{s}-\boldsymbol{b}))$ performs elementwise non-zero indication and outputs a binary vector. Now two auxiliary variables $\boldsymbol{c}_1$, $\boldsymbol{c}_2 \in \mathbb{R}^{2N}$ are introduced to replace $\nabla\boldsymbol{b}$, $\nabla(\boldsymbol{s}-\boldsymbol{b})$, respectively. The resultant augmented Lagrangian function of our model is written as

$$\begin{aligned}\mathcal{L}(\boldsymbol{b},\boldsymbol{c}_1,\boldsymbol{c}_2,\boldsymbol{y}_1,\boldsymbol{y}_2) &= \tfrac{1}{2}\|\boldsymbol{s}-\boldsymbol{b})\|_2^2 + \lambda_1\|\boldsymbol{c}_1\|_1 \\ &+ \lambda_2\mathbf{1}^\top \mathrm{E}_z(\boldsymbol{c}_2) + (\boldsymbol{c}_1 - \nabla\boldsymbol{b})^\top\boldsymbol{y}_1 \\ &+ (\boldsymbol{c}_2 - \nabla(\boldsymbol{s}-\boldsymbol{b}))^\top\boldsymbol{y}_2 \\ &+ \tfrac{\rho}{2}(\|\boldsymbol{c}_1 - \nabla\boldsymbol{b}\|_2^2 + \|\boldsymbol{c}_2 - \nabla(\boldsymbol{s}-\boldsymbol{b})\|_2^2),\end{aligned} \tag{2.4}$$

where $\boldsymbol{y}_i, i = 1, 2$ are the Lagrangian dual variables. At iteration k, the function (2.4) is optimized by minimizing primal sub-problems with respect to $\boldsymbol{b}$, $\boldsymbol{c_1}$, $\boldsymbol{c_2}$ and maximizing the dual problems with respect to $\boldsymbol{y_1}$, $\boldsymbol{y_2}$ alternatively.

(1) Solving $\boldsymbol{b}^{k+1}$:

Firstly we split vector $\boldsymbol{c}_1^k$ into two equal-length pieces, i.e., $\boldsymbol{c}_1^k = [\boldsymbol{c}_{1,1}^{k\,T}, \boldsymbol{c}_{1,2}^{k\,T}]^T$, where $\boldsymbol{c}_{1,i}^{k\,T} \in \mathbb{R}^N, i = 1, 2$. In the same fashion, $\boldsymbol{c}_2^k = [\boldsymbol{c}_{2,1}^{k\,T}, \boldsymbol{c}_{2,2}^{k\,T}]^T$, $\boldsymbol{y}_1^k = [\boldsymbol{y}_{1,1}^{k\,T}, \boldsymbol{y}_{1,2}^{k\,T}]^T$ and $\boldsymbol{y}_2^k = [\boldsymbol{y}_{2,1}^{k\,T}, \boldsymbol{y}_{2,2}^{k\,T}]^T$. Then the objective function with respect to $\boldsymbol{b}^{k+1}$ is a quadratic programming problem

$$\begin{aligned}\boldsymbol{b}^{k+1} = arg\min_{\boldsymbol{b}} \Bigg\{ &\frac{1}{2}\|\boldsymbol{s}-\boldsymbol{b}\|_2^2 + \frac{1}{2}\|\boldsymbol{c}_{1,1}^k - \nabla_x\boldsymbol{b} + \frac{\boldsymbol{y}_{1,1}^k}{\rho^k}\|_2^2 + \frac{1}{2}\|\boldsymbol{c}_{1,2}^k - \nabla_y\boldsymbol{b} + \frac{\boldsymbol{y}_{1,2}^k}{\rho^k}\|_2^2 \\ &+ \frac{1}{2}\|\boldsymbol{c}_{2,1}^k - \nabla_x\boldsymbol{b} + \frac{\boldsymbol{y}_{2,1}^k}{\rho^k}\|_2^2 + \frac{1}{2}\|\boldsymbol{c}_{2,2}^k - \nabla_y\boldsymbol{b} + \frac{\boldsymbol{y}_{2,2}^k}{\rho^k}\|_2^2 \Bigg\},\end{aligned} \tag{2.5}$$

which can be efficiently solved in Fourier domain

$$\boldsymbol{b}^{k+1} = \mathrm{fft}^{-1}\left(\frac{\mathrm{fft}(s) + \mathrm{fft}^*(\nabla_x \cdot \boldsymbol{fx}^k) + \mathrm{fft}^*(\nabla_y \cdot \boldsymbol{fy}^k)}{1 + 2\rho^k\big(\mathrm{fft}^*(\nabla_x)\cdot\mathrm{fft}(\nabla_x) + \mathrm{fft}^*(\nabla_y)\cdot\mathrm{fft}(\nabla_y)\big)}\right), \tag{2.6}$$

where

$$\begin{aligned}
\boldsymbol{fx}^k &= \mathrm{fft}\left(\rho^k(\boldsymbol{c}_{1,1}^k + \frac{\boldsymbol{y}_{1,1}^k}{\rho^k} + \nabla_x \boldsymbol{s} - \boldsymbol{c}_{2,1}^k - \frac{\boldsymbol{y}_{2,1}^k}{\rho^k})\right), \\
\boldsymbol{fy}^k &= \mathrm{fft}\left(\rho^k(\boldsymbol{c}_{1,2}^k + \frac{\boldsymbol{y}_{1,2}^k}{\rho^k} + \nabla_y \boldsymbol{s} - \boldsymbol{c}_{2,2}^k - \frac{\boldsymbol{y}_{2,2}^k}{\rho^k})\right).
\end{aligned} \tag{2.7}$$

The denotations fft, fft^* and fft^{-1} are the 2-D FFT, conjugate FFT and inverse FFT, respectively.

(2) Solving $\boldsymbol{c}_1^{k+1}$:

The objective function with respect to $\boldsymbol{c}_1^{k+1}$ is

$$\boldsymbol{c}_1^{k+1} = arg\min_{\boldsymbol{c}_1}\left\{\frac{2\lambda_1}{\rho^k}\|\boldsymbol{c}_1\|_1 + \|\boldsymbol{c1} - \nabla\boldsymbol{b}^{k+1} + \frac{\boldsymbol{y}_1^k}{\rho^k}\|_2^2,\right\}, \tag{2.8}$$

which can be solved by soft-shrinkage operation:

$$\boldsymbol{c}_1^{k+1} = \mathcal{T}_{\lambda_1/\rho^k}(\nabla\boldsymbol{b}^{k+1} - \boldsymbol{y}_1^k/\rho^k), \tag{2.9}$$

where $\mathcal{T}_\alpha(x) = \mathrm{sign}(x)\cdot\max(|x| - \alpha, 0)$ is the soft-thresholding function.

(3) Solving $\boldsymbol{c}_2^{k+1}$:

The objective function with respect to $\boldsymbol{c}_2^{k+1}$ is:

$$\boldsymbol{c}_2^{k+1} = arg\min_{\boldsymbol{c}_2}\left\{\frac{2\lambda_2}{\rho^k}F(\boldsymbol{c}_2) + (\boldsymbol{c}_2 - \boldsymbol{q}^k)^2\right\}, \quad \text{where} \quad \boldsymbol{q}^k = \nabla(\boldsymbol{s} - \boldsymbol{b}^{k+1}) - \frac{\boldsymbol{y}_2^k}{\rho^k}. \tag{2.10}$$

This objective function can be solved in an element-wise manner

$$\sum_{j=1}^{2N}\min_{\boldsymbol{c}_{2,j}}\left\{\frac{2\lambda_2}{\rho^k}F(\boldsymbol{c}_{2,j}) + (\boldsymbol{c}_{2,j} - \boldsymbol{q}_j^k)^2\right\}, \tag{2.11}$$

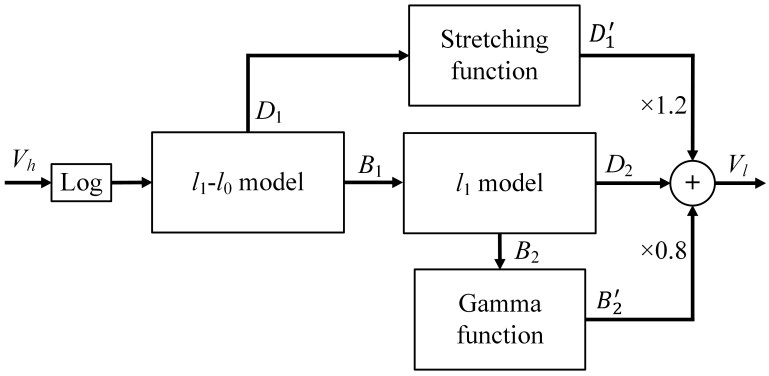


Figure 2.2: The proposed two-scale tone mapping algorithm

where j is the entry index of a vector.

According to the analysis of [120], the solution of $\boldsymbol{c}_2^{k+1}$ at entry j is

$$\boldsymbol{c}_{2,j}^{k+1} = \begin{cases} 0, & \text{if } (\boldsymbol{q}_j^k)^2 \leqslant \frac{\lambda_2}{\rho^k} \\ \boldsymbol{q}_j^k, & \text{Otherwise} \end{cases} . \tag{2.12}$$

(4) Dual ascent for Lagrangian multipliers.

$$\begin{aligned} \boldsymbol{y}_1^{k+1} &= \boldsymbol{y}_1^k + \rho^k(\boldsymbol{c}_1^{k+1} - \nabla \boldsymbol{b}^{k+1}), \\ \boldsymbol{y}_2^{k+1} &= \boldsymbol{y}_2^k + \rho^k(\boldsymbol{c}_2^{k+1} - \nabla(\boldsymbol{s} - \boldsymbol{b}^{k+1})). \end{aligned} \tag{2.13}$$

(5) Update ρ^{k+1} as $\rho^{k+1} = 2\rho^k$.

The ADMM is efficient to find the approximate solution for the base layer $\boldsymbol{B}$ variable within a few iterations(15 in our case). Lastly, after the estimation for $\boldsymbol{B}$, the detail layer is obtained by $\boldsymbol{S} - \boldsymbol{B}$.

2.2.3 Extension to Multiscale Decomposition

By applying the hybrid ℓ_1-ℓ_0 decomposition model (2.1) to the radiance map, we can produce a piecewise constant detail layer and a piecewise smooth base layer. While this single-scale scheme is a standard framework for tone mapping, applying the decomposition repeatedly to create multiple scales can further improve the tone mapping algorithm. In this way, different attributes of an image, represented by

(a) One scale (b) Two scales

Figure 2.3: Tone mapping results by one scales and two scales

different scale layers, can be manipulated, which leads to more flexible and effective tone reproduction. By leveraging the efficiency and effectiveness, we adopt a two-scale decomposition scheme for tone mapping, as depicted in Fig 2.2. This will produce the first scale detail layer $\boldsymbol{D}_1$, the second scale detail layer $\boldsymbol{D}_2$ and the second scale base layer $\boldsymbol{B}_2$.

As discussed in Section 2.2.1, the spatial property of $\boldsymbol{D}_1$ mostly affects the tone mapped image. We apply the proposed ℓ_1-ℓ_0 model (2.1) to the first scale decomposition:

$$\begin{aligned} \boldsymbol{B}_1 &= \text{model}_{\ell_1,\ell_0}(\boldsymbol{S}), \\ \boldsymbol{D}_1 &= \boldsymbol{S} - \boldsymbol{B}_1, \end{aligned} \tag{2.14}$$

where $\text{model}_{\ell_1,\ell_0}(\cdot)$ is the optimization model in (2.1). After the first level decomposition, the structural information remains in the detail layer $\boldsymbol{D}_1$ and the main textural information is transfered to the base layer $\boldsymbol{B}_1$.

For the second scale decomposition model, model (2.1) is applied to $\boldsymbol{B}_1$, but the weight λ_2 of the ℓ_0 term is set to 0, leading to a total variation problem:

$$\begin{aligned} \boldsymbol{B_2} &= arg\min_{\boldsymbol{B}} \sum_{p=1}^{N} \left\{ (\boldsymbol{B}_{1,p} - \boldsymbol{B}_p)^2 + \lambda_3 \textstyle\sum_{i=\{x,y\}} |\partial_i \boldsymbol{B}_p| \right\}, \\ \boldsymbol{D}_2 &= \boldsymbol{B}_1 - \boldsymbol{B}_2, \end{aligned} \tag{2.15}$$

This arrangement is due to the strategy that we preserve the textural information of

the image in the 2nd scale detail layer $\boldsymbol{D}_2$. Thus, the ℓ_0-based structural prior is not applicable in this decomposition. As a result, the layer $\boldsymbol{D}_2$ stores the majority of the textural information, and the layer $\boldsymbol{B}_2$ contains local mean brightness.

To summarize, our two-scale decomposition scheme produces three layers in addition relationship:

$$\boldsymbol{S} = \boldsymbol{D}_1 + \boldsymbol{D}_2 + \boldsymbol{B}_2, \tag{2.16}$$

Figs 2.3 shows the difference between our tone mappers with 1 scale and 2 scales(The details of our algorithm will be discussed in Section 2.2.4). It can be seen that while the one-scale result is acceptable, the two-scale result preserves the medium frequency of an image and achieves more natural appearance.

Acceleration. The accuracy of the second scale decomposition (2.15) is not strictly required. Thus, we adopt a acceleration scheme. First, we linearly downsample the $\boldsymbol{B}_1$ by a factor of 4. Then the decomposition model in (2.15) is performed on a low resolution of $\boldsymbol{B}_2$, followed by a linear upsampling to the original resolution. Because the boundary regions in the image are slightly blurred due to the sampling scheme, we finally perform a rapid guided filtering of $\boldsymbol{B}_2$ with the original $\boldsymbol{B}_1$ as the guidance image to recover the sharp boundary information [54].

2.2.4 Tone Mapping Process

The processing steps of the proposed tone mapping algorithm mainly include color transformation, multiscale decomposition, detail layer boosting, base layer compression, and recombination of the layers. While this algorithm framework is common in the tone mapping research, our approach mainly differs in two aspects. First, our suit of layer decomposition models is discriminative in the spatial attributes of an image. As described in Section 2.2.3, our multiscale decomposition deploys the structural information, textural information and local mean brightness separately

into different layers, whereas existing multiscale models merely perform a progressive smoothing [41, 49]. Second, in our multiscale manipulation approach, we perform a layer-selective nonlinear processing, whereas other works only perform linear intensity scaling [41].

Since the dynamic range of an image is mostly embedded in the brightness domain, our core algorithm only processes the luminance channel and preserves the chromaticity components. Specifically, the input RGB radiance map is transformed to HSV space and only the V channel is tone mapped. At the reverse transformation stage, the saturation channel is multiplied by 0.6 to prevent oversaturation.

Our tone mapping algorithm on the luminance channel of a radiance map is depicted in Fig 2.2. The V channel $\boldsymbol{V}_h$ of the radiance map is firstly converted to log domain and normalized to the range of (0, 1). This approach mimics the response of human vision to the luminance and preliminarily reduces the dynamic range. Then our two-scale decomposition scheme using (2.14) and (2.15) is applied, yielding three layers $\boldsymbol{D}_1$, $\boldsymbol{D}_2$, and $\boldsymbol{B}_2$. Since the base layer $\boldsymbol{B}_2$ can be considered as the local brightness level of the image, we compress it by a gamma function:

$$\boldsymbol{B}_2' = L \cdot (\frac{\boldsymbol{B}_2}{L})^{\frac{1}{\gamma}}, \tag{2.17}$$

where L is the largest brightness level($L = 1$ in our case, due to the normalization). For the first detail layer $\boldsymbol{D}_1$, we use a nonlinear stretching function to boost it:

$$\boldsymbol{D}_1' = \operatorname{sign} \boldsymbol{D}_1 \cdot \left(\frac{|\boldsymbol{D}_1|}{\max(|\boldsymbol{D}_1|)} \right)^{\alpha} \cdot \max(|\boldsymbol{D}_1|), \tag{2.18}$$

This function with the parameter α has a stretching effect for signals centering at 0. Smaller α yields larger stretching degree and vice versa. Since the structural prior is imposed in $\boldsymbol{D}_1$ by decomposition model (2.1), the structural residual of the original image is boosted by the stretching function. This arrangement would result in a more visually appealing image. Then, a luminance SDR image is reconstructed by

$$\boldsymbol{V}_l = 1.2\boldsymbol{D}_1' + \boldsymbol{D}_2 + 0.8\boldsymbol{B}_2'. \tag{2.19}$$

(a) The log radiance map (b) $\lambda_2 = 0.0008$

(c) $\lambda_2 = 0.003$ (d) $\lambda_2 = 0.008$

Figure 2.4: The effect of λ_2 on the detail layer when λ_1 is fixed to 0.3.

Finally, the values of $\boldsymbol{V}_l$ at 0.5% and 99.5% intensity level are mapped to 0 and 1, respectively. Values out of this range are clipped.

2.3 Experiments and Analysis

This section presents several experiments to verify the performance of our hybrid ℓ_1-ℓ_0 layer decomposition model (2.1) and the proposed tone mapping algorithm. A HDR database with 40 radiance maps is collected for evaluation.

2.3.1 Parameter Selection

The parameters that affect our ℓ_1-ℓ_0 decomposition model (2.1) are λ_1, λ_2. They balance the fidelity term, the ℓ_1 gradient sparsity term on $\boldsymbol{B}_1$ and the ℓ_0 gradient sparsity term on $\boldsymbol{D}_1$. Figs 2.4 shows the effects of λ_2 on the detail layer when λ_1 is fixed. It can be seen that different values of λ_2 lead to different degrees of flattening effect on $\boldsymbol{D}_1$. When λ_2 is excessively large, some structures are totally flattened. In contrast, when λ_2 is overly small, some small texture gradients appear in the $\boldsymbol{D}_1$, and

(a) $\lambda_1 = 0.06$ (b) $\lambda_1 = 0.3$ (c) $\lambda_1 = 1$

(d) $\lambda_1 = 0.06$ (e) $\lambda_1 = 0.3$ (f) $\lambda_1 = 1$

Figure 2.5: The effect of λ_1 on the two layers when λ_2 is fixed to $0.01\lambda_1$. The upper three images are detail layers. The bottom three images are base layers.

the structural prior is less modeled. We performed an exhaustive experiment with our database and found that when λ_2 is set to $0.01\lambda_1$ the decomposition is consistently satisfactory. Figs 2.5 presents the effect of parameter λ_1 when λ_2 is fixed to $0.01\lambda_1$. It can be seen that λ_2 mainly controls the degree of piecewise smoothness of $\boldsymbol{B}_1$ and the signal magnitude of $\boldsymbol{D}_1$. We fix λ_1 to a moderate value of 0.3. In summary, λ_1 is empirically fixed at 0.3 while λ_2 is fixed at 0.003. Both parameters are not dependent on image contents and we find them satisfactory for most HDR images.

Other parameters that are left to be determined are λ_3 in (2.15), γ in (2.17) and α in (2.18). λ_3 controls the degree of smoothness in the final base layer $\boldsymbol{B}_2$. We found that except some extreme settings, λ_3 does not considerably affect the tone mapped image. Hence λ_3 is fixed to 0.1. α mainly controls the stretching degree of the first detail layer $\boldsymbol{D}_1$. To prevent over-boosting effect, we set it to a moderate value of 0.8. Finally, the γ is set to 2.2 as a common practice in Retinex decomposition research [66, 90, 43].

(a) $\boldsymbol{D}_1$ by Gu's[49] (b) $\boldsymbol{D}_2$ by Gu's[49] (c) $\boldsymbol{B}_2$ by Gu's[49] (d) Results by Gu's[49]

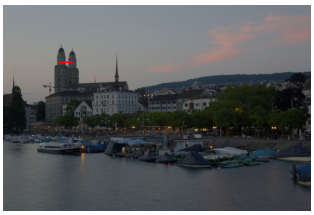
(e) $\boldsymbol{D}_1$ by Ours (f) $\boldsymbol{D}_2$ by Ours (g) $\boldsymbol{B}_2$ by Ours (h) Results by ours

Figure 2.6: Comparison of multiscale decomposition models

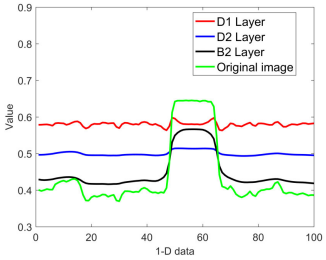
2.3.2 The Decomposed Layers

To verify the multiscale decomposition performance of our tone mapping algorithm, we compare with Gu's multiscale tone mapper [49]. In Gu's model, a local guided filter weighted by gradient function is repeatedly applied to the original image to obtain 2 scale layers. Note that although Gu's model is claimed to have 3 scales(4 layers), the last scale base layer is a constant image. Thus the valid scale number is two. We merge the last two layers of Gu's model to one, resulting in 2 scales(3 layers) in total. Gu's model enforces the edge-preserving property on the base layer without imposing any prior on the detail layer.

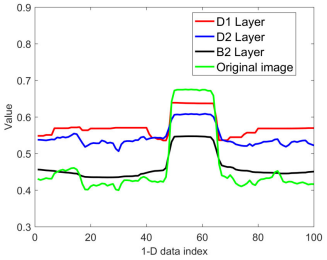
In Figs 2.6, the multiscale decomposition results by Gu's model and our methods are compared. A 1-D auxiliary analysis is shown in Figs 2.7, where a piece of 1-D signal(the position is the red line in Fig 2.7(a)) is extracted from the decomposed layers of each method. It can be seen from Fig 2.7(b)) that Gu's model simply performs progressive smoothing without considering the spatial property of the detail layer. Thus, the first detail layer(the red curve in 2.7(b)) is full of small fluctuations and the tone mapped image is over-enhanced, as depicted in Fig 2.6(d). What's



(a) Log radiance map(with line for (b) and (c))



(b) Gu's model[49]



(c) Ours

Figure 2.7: 1-D analysis of multiscale decomposition

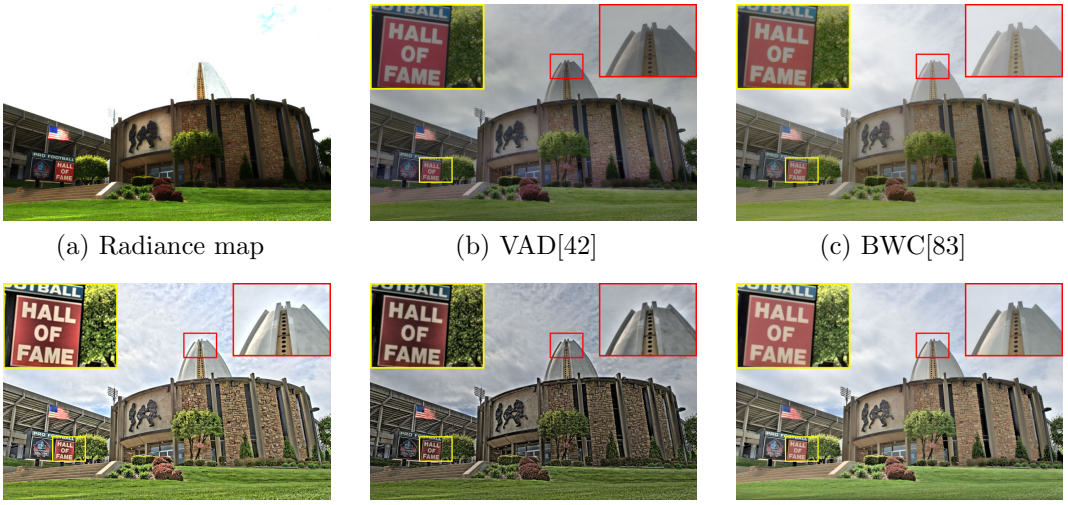
worse, Gu's model does not strictly preserve edges due to the nature of local filter. Thus the tone mapped result has halo artifact, see the zoom-in in Fig 2.6(d). In contrast, due to the structural prior, our method distributes the small-scale variations in the second layer $\boldsymbol{D}_2$, and enforces the first layer $\boldsymbol{D}_1$ to be piecewise constant, as shown in Fig 2.7(c). In addition, our method is also edge-preserving. Therefore, our model not only avoids halo artifacts but also achieves visually compelling results, as shown in Fig 2.6(h).

2.3.3 Comparison of Tone Mapping

We compare our tone mapper with 4 state-of-the-art tone mapping algorithms(TMO). These 4 TMOs are visual adaptation method(VAD) [42], backward-compatible method (BWC) [83], guided filter method(GF) [49], and gradient reconstruction method(GR) [107]. GF is implemented by us since the source code is not available. BWC is implemented with pfstool[1]. VAD and GR are implemented by the authors' source codes. All these tone mapping methods use the default parameters as provided in the original papers.

Subjective evaluation. Figs 2.8, 2.9 show the comparison of tone mapping results. We can see that our method achieves a strong balance between detail

[1]http://pfstools.sourceforge.net/



(a) Radiance map (b) VAD[42] (c) BWC[83]

(d) GF[49] (e) GR[107] (f) Ours

Figure 2.8: Comparison of tone mapping

(a) Radiance map (b) VAD[42] (c) BWC[83]

(d) GF[49] (e) GR[107] (f) Ours

Figure 2.9: Comparison of tone mapping

(a) Radiance map (b) Photomatix (c) Ours

Figure 2.10: Comparison with Photomatix

Table 2.1: Comparison of mean opinion scores

	VAD[42]	BWC[83]	GF[49]	GR[107]	Ours
Mean	4.68	5.11	5.31	4.60	6.43
Std	1.48	1.21	1.45	1.60	1.20

enhancement and naturalness preservation. In contrast, other TMOs suffer from different types of distortions. GR and GF have over-enhancement problem and halo artifacts. VAD has color shift problem and BWC overly softens the images. In Figs 2.10, our tone mapper is compared with the default tone mapper of Photomatix[2]. We can see that while the two methods can obtain satisfactory results, our method achieves higher visual interpretability on the image due to the highlighting of structural information.

To further qualitatively verify the performance of our tone mapper, we perform a subjective experiment on our HDR database. Specifically, 6 subjects, 3 males and 3 females, are requested to give scores to 40 images tone-mapped by each method. The score ranges from 1(the worst) to 8(the best) spaced with 0.5. 2 of the subjects

[2]https://www.hdrsoft.com/

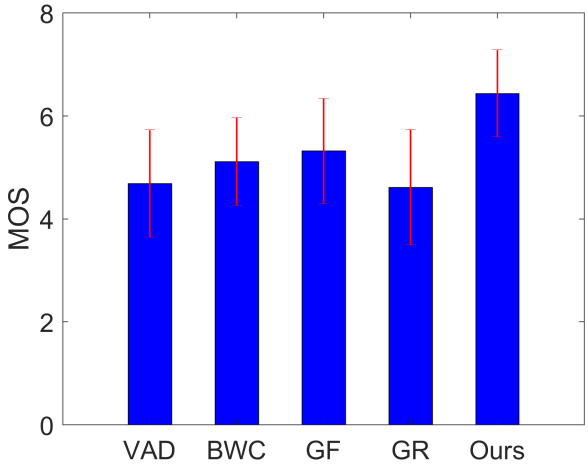


Figure 2.11: Comparison of mean opinion score statistics

Table 2.2: Comparison of TMQI

	TMQI	Structural fidelity	Naturalness
WLS[41]	0.8703	0.8513	0.4540
VAD[42]	0.8695	**0.8614**	0.4320
BWC[83]	0.8633	0.8498	0.4213
GF[49]	0.8692	0.8446	0.4508
GR[107]	0.8746	0.8303	0.5147
Ours	**0.8851**	0.8334	**0.5547**

are researchers in image processing, while the others major in other fields. The tone mapped images are shown in random order on a PA328 display with 32 inch (7680×4320), controlled by a Mac Pro PC with 2.9 GHz CPU. The subjects are not acknowledged of the tone mapping algorithms involved in the experiment. The subjects are taught how to use the programs before the evaluation. The mean opinion score statistics are illustrated in Fig 2.11 and table 2.1. We can see that our tone mapper achieves the highest mean scores(6.43) and a tolerable variation(1.20).

Objective evaluation. Aside from subjective evaluation, we use the Tone

Table 2.3: Comparison of running time on a 1333 × 2000 image

	VAD[42]	BWC[83]	GF[49]	GR[107]	Ours
Code	C++	C++	Matlab	Matlab	Matlab
Time	18.1s	0.7s	1.7s	77.6s	8.6s

Mapped Image Quality Index (TMQI) to perform an objective evaluation on 6 TMOs. [124]. TMQI first evaluates the structural fidelity and naturalness of the tone mapped images. Then the two measures are adjusted by power function and averaged to give a final score in the range from 0 to 1. Larger values of TMQI indicate better quality of the tone mapped image, and vice versa. Table 2.2 illustrates the mean TMQI score of each TMO performed on our database with 40 HDR images. We can see that our method achieves not only the highest TMQI score (0.8851), but also the highest naturalness measure (0.5547). These excellent marks objectively indicate the high performance of our algorithm.

Efficiency. The proposed tone mapper has a relatively low computational complexity. The most complicated part is the FFT operation in the ADMM-based solver, which cost $\mathcal{O}(N\Delta log(N))$. Table 2.3 compares the running time of the 5 TMOs on a 1333 × 2000 sized image(Fig 2.8(a)). The testing environment is a PC with i7 6850k CPU, 16G RAM. It can be seen that our tone mapper has a moderate running time.

2.3.4 Discussion and Future Work

Given the fact that deep learning technique has been successfully applied to various image processing tasks, i.e., super-resolution and denoising, it is a natural idea to adopt it to improve the tone mapping quality. One can construct a pairwise dataset with HDR images and their ideal tone mapped counterparts and train a deep convolutional neural network (CNN) to perform the tone mapping process. However,

the generation of ground truths, i.e., the tone mapped images, is a challenging problem. It requires labor-intensive adjusting of various image attributes, including brightness, local contrast, color and details. In addition, the agreement on ground truth quality is highly subjective as an HDR image could have multiple style of tone mapped images. Therefore, there are little work in the academia that adopts deep learning approach for tone mapping. Rana *et al.* [99] adopted a compromised scheme, where the ground truth tone mapped images are generated by manually selecting the best results from several traditional tone mapping algorithms. This compromised approach will degrade the performance of the learned network.

In the future work, we plan to address the above problem by constructing a dedicated tone mapping dataset. We will collect a large number of HDR images and hire several photographers to create the tone mapped images by using HDR softwares like Photoshop or Aurora HDR. Each HDR image will correspond to multiple styles of tone mapped images which are generated by different photographers. We believe this dataset will not only provide high quality ground truths for network training, but also offer wide range of choices for tone mapped results.

2.4 Summary

We propose a novel hybrid ℓ_1-ℓ_0 layer decomposition model to address the overenhancement and halo artifact problem of tone mapping. This decomposition model effectively realizes a structural prior of the detail layer and the edge-preserving prior of the base layer. The ADMM algorithm is adopted to solve the optimization model efficiently. Then, based on this ℓ_1-ℓ_0 layer decomposition, a multiscale tone mapping algorithm is proposed. It performs dynamic range reduction in the base layer and structure boosting in the detail layer. Due to the proper use of the two priors, our multiscale tone mapping algorithm not only avoids halo artifact but also achieve more

visually compelling tone mapping results than existing works.

Chapter 3

A Smooth-and-Enhance Strategy for Detail-enriched Single Image Denoising

Single image denoising is a fundamental research topic in the academia. Though the recently proposed deep learning based denoising methods have achieved a great success, the trade-off between noise removal and texture preservation often leads to over-smoothed results, sacrificing the perceptual quality of denoised images. One may train a denoising network with adversarial training technique to synthesize image details. However, mixing the tasks of noise removal and detail synthesis tends to generate much visual artifact. In this chapter, we propose a two-stage network for detail-enriched image denoising by using a smooth-first (noise removal) and enhance-later (detail enhancement) strategy, with one subnetwork designed for each of the two stages. Firstly, the noise removal and detail enhancement subnetworks are pre-trained for conventional denoising and super-resolution tasks, respectively, to learn image priors for smoothing and enhancement. Then a joint adversarial training is performed in a scale-progressive manner to output detail enriched results. Experiments on synthetic and real-world noisy images demonstrate that the proposed smooth-and-enhance method can significantly improve the perceptual quality of denoised images

with richer textures and details over the recent state-of-the-arts.

3.1 Introduction

Noise introduction is inevitable in the digital imaging process. The photonic nature of light causes fluctuation in the light collection of a camera, while the imperfection of electronic hardware adds further disturbance on the analog to digital conversion [118]. The noise degrades the image quality by impairing the image details and structures, and hence denoising is an indispensable step in the camera image signal processing (ISP) pipeline [98] to remove the noise and recover the image details.

A variety of image denoising algorithms [104, 40, 17, 33, 50, 119] have been developed in the past decades. Most of the successful methods exploit the prior knowledge of natural image statistics, including statistical prior [104, 94], sparsity prior [3, 40, 84], self-similarity prior [17, 33, 50, 119, 82], etc. The recently developed deep learning based methods [131, 132, 71, 123] significantly boost the denoising performance by training deep convolutional neural networks (CNNs) on large-scale datasets of noisy-clean image pairs. Instead of using hand-crafted image priors, CNNs can learn priors from data and they demonstrate much better results on reconstructing image structures.

In spite of the great success, CNN-based denoising methods still need to trade-off between noise removal and texture preservation, leading to over-smoothed results, which deviate from the natural image statistics and have low perceptual quality. Although some methods can recover more details by decreasing the denoising strength via extra network input such as noise map [132], the residual noise impairs the image quality. The aforementioned problem is caused by the fact that image details and noise are tangled with each other in high-frequency domain. It is difficult for the denoisers to tell noise from image details. Thus, much textures and fine details are

lost during the noise removal process.

To enrich the image details and improve the perceptual quality of denoising results, one intuitive idea is to adopt the popular adversarial training techniques [47, 116]. Specifically, one could fine-tune the pre-trained denoising network with the supervision of a discriminator network, which differentiates between original natural images and denoised ones. Under such a supervision, the denoising network can learn to synthesize images that approximate the statistics of natural images. However, directly introducing adversarial training to image denoising could lead to unstable results with inferior details, because removing noise and synthesizing details are two contradicting subtasks: the former reduces the high-frequency image components while the latter increases. Mixing these two subtasks will mislead the CNN to take noise as source of details and amplify them. A more deliberate strategy is needed to synthesize image details without amplifying noise.

To solve the aforementioned problem, we propose a two-stage network for detail-enriched image denoising by using a smooth-first and enhance-later strategy. We explicitly split the denoising process into a noise removal process and a detail enhancement process, and characterize them by two subnetworks, respectively. The noise removal subnetwork conducts normal denoising, which yields smooth but overall clean results. Based on the clean semantic content of the first stage, the detail enhancement subnetwork then hallucinates image details via adversarial training. We call the proposed network SAE-Net (smooth and enhance network) and design an effective training scheme of it. First, the noise removal subnetwork is pre-trained via a denoising task, while the detail enhancement subnetwork is pre-trained via a super-resolution task for initial detail synthesis. The pre-training process aims to learn image priors for noise smoothing and detail enhancement. Then the whole network is jointly fine-tuned with the supervision of a discriminator network. To stabilize the adversarial training, we adopt a progressive scheme where the learning

gradually moves from low to high resolution scales. Experiments on synthetic and real-world noisy images show that our SAE-Net can achieve visually compelling results by simultaneously removing noise and synthesizing high-quality details. It outperforms the recently proposed deep-learning-based denoising methods by a large margin in terms of perceptual quality metrics (e.g., PI [14] and LPIPS [135]).

3.2 Image Denoising by Smooth-and-Enhance Strategy

Image denoising aims to recover the latent clean image I from its degraded version I_n. In this work we consider the AWGN in sRGB space and real-world noise in raw domain. Given a training dataset $\mathcal{S}$ with noisy-clean image pairs (I_n,I_{gt}), we aim to train a denoising CNN $G(\cdot;\theta)$ parameterized by θ to restore the clean images.

The key challenge to restore a visually pleasant image is to remove the noise while preserving the image fine details. However, it is difficult for a denoiser to address these two subtasks simultaneously because image details and noise are entangled with each other in the high-frequency domain of an image. When a network is trained to remove noise, like the many existing works [131, 4, 137], it may treat some of the image details as noise and suppress them, resulting in piece-wise smooth denoising output. It is always a high demand for a denoiser to reproduce image fine-scale details so that the denoised images could look more natural.

Intuitively, one may fine-tune the pre-trained denoiser with the recently proposed adversarial training methods [116] by using a discriminator network to guide the denoising network to synthesize realistic details. However, the process of synthesizing details conflicts with removing noise because the former enhances the high-frequency components while the latter suppresses them. Mixing the two subtasks in a one-stage network structure could mistreat the noise as source of image details and amplify

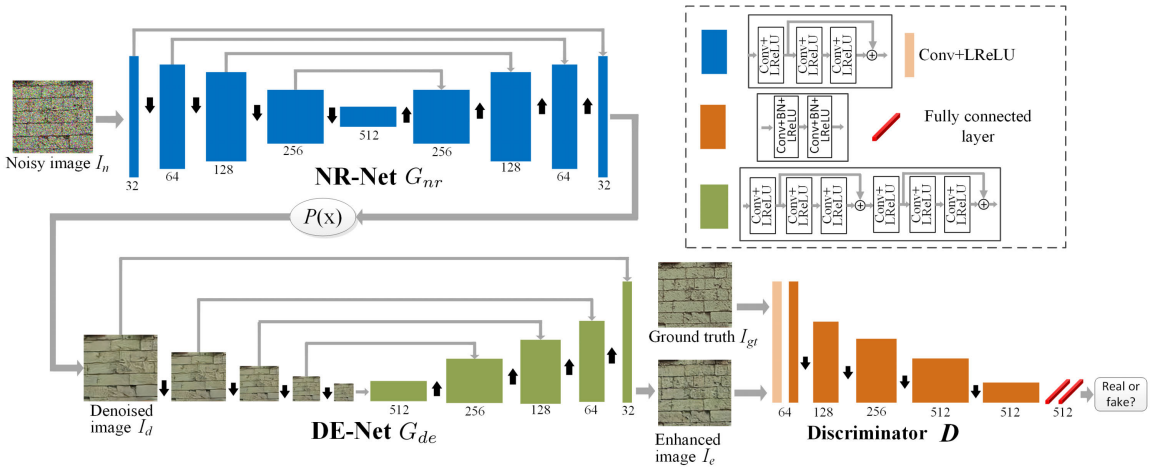


Figure 3.1: Network structure.

them, leading to results with much artifacts and inferior quality. Therefore, we need to carefully treat the noise removal and detail synthesis subtasks to better leverage the adversarial training technique.

3.2.1 Smooth and Enhance: A Two-stage Network

Instead of using a one-stage network for denoising, we proposed a two-stage network structure with a smooth-first and enhance-later strategy. We split the denoising process into a noise removal stage and a detail enhancement stage, which are characterized by two subnetworks, respectively. The noise removal subnetwork, called NR-Net, focuses on smoothing image noise, while the detail enhancement subnetwork, called DE-Net, focuses on hallucinating high-frequency textures for detail-enriched results. In order to effectively train the two subnetworks, which have conflicting objectives, we have the following considerations.

First, the two subnetworks should be trained to accomplish their specific objectives, i.e., smoothing noise and synthesizing details. To this end, we pre-train the NR-Net as a normal denoising network and pre-train DE-Net as a super-resolution network which recovers certain image details. Second, the whole network should produce

clean images with realistic details that match natural image statistics. To achieve this goal, we perform adversarial joint fine-tuning of the two subnetworks after the pre-training. We describe the network structures in this section while discussing the details of training strategy in Section 3.2.2.

Network structure. We call the proposed network SAE-Net (smooth and enhance network), while its two stage structure is shown in Fig. 3.1. The noise removal subnetwork NR-Net, denoted as $G_{nr}(\cdot;\theta_{nr})$, takes I_n as input and produces an initially denoised image I_d:

$$I_d = G_{nr}(I_n;\theta_{nr}) \tag{3.1}$$

Then, I_d goes through an operation $P(\cdot)$ before being further processed. In the case of AWGN removal (in sRGB color space), $P(\cdot)$ is simply an identity mapping. In the case of real-world noise removal in raw domain, $P(\cdot)$ is a set of differentiable ISP operations that convert I_d from raw domain to sRGB domain, including demosaicking $M(\cdot)$, white balance $W(\cdot)$, color space conversion $C(\cdot)$ and gamma correction $G(\cdot)$:

$$P(I_d) = G(C(W(M(I_d))) \tag{3.2}$$

In the nonlinear sRGB space, image details in dark regions are more prominent, which facilitates the detail synthesis in dark regions.

The detail enhancement subnetwork DE-Net, denoted as $G_{de}(\cdot;\theta_{de})$, takes $P(I_d)$ as input and outputs an enhanced clean image I_e with richer textures and details:

$$I_e = G_{de}(P(I_d);\theta_{de}) \tag{3.3}$$

A discriminator network $D(\cdot;\theta_d)$ is employed to differentiate the authenticity of the enhanced image. Specifically, $D(\cdot;\theta_d)$ takes I_e and the ground truth image as input, and outputs a probability to indicate the naturalness of the input image. Although many sophisticated CNN architectures can be employed to implement the two subnetworks and the discriminator network, we implement them with simple yet effective multi-scale structures for simplicity.

Noise removal network. We adopt a 5-level UNet-like structure for NR-Net [103], as shown in Fig. 3.1. The NR-Net has a contracting path that progressively decreases the resolution of feature maps, followed by an expanding path to progressively expand the resolution back. The detail information in the feature map is preserved by the skip connections between the contracting path and the expanding path at the same resolution level. To facilitate information propagation, we deploy residual convolutional blocks in the two paths at each resolution level. The UNet has advantages of scale-adaptive operation, where finer scales focus on removing high-frequency noise and coarser scales focus on restoring overall image structures.

Detail enhancement network. We also adopt a multi-scale structure for DE-Net but with different settings from NR-Net. We keep the expanding path but remove the learnable layers in the contracting path. Instead, the contraction is performed by downsampling the input image with bilinear method to each resolution level. This setting facilitates the progressive GAN training, which will be explained in Section 3.2.2. At each resolution level, a convolutional layer is deployed to process the input image, followed by two residual blocks which hallucinate details on the previous feature maps. Skip connections are deployed between the input image and the expanding path at the same resolution level. With the multi-scale structure, DE-Net can hallucinate large-scale structures on coarse scales, while synthesize high-frequency textures on finer scales.

Discriminator network. As shown in the right-bottom corner of Fig. 3.1, the discriminator works at 5 resolution levels by progressively downsampling the input image with stride-2 convolutions. Two fully connected layers are deployed at the lowest resolution and output the probability value. Batch normalization is employed between convolutions and activations, except the first convolution.

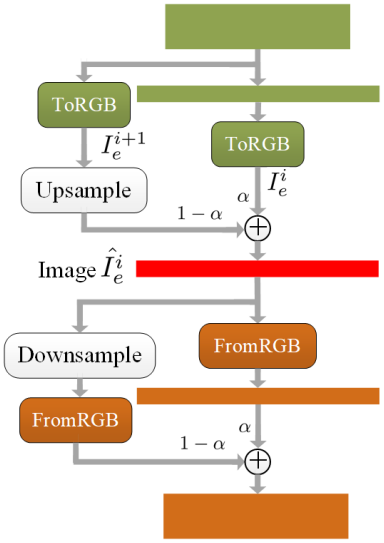


Figure 3.2: Fade-in layers. The green and orange blocks denote the modules from DE-Net and discriminator, respectively. ToRGB and FromRGB denotes 1×1 convolutions that reduce and expand the channel size, respectively.

3.2.2 Training Strategy

The training of the proposed SAE-Net is divided into two steps. The first step independently pre-trains the two subnetworks to learn denoising and super-resolution priors from natural images. The second step performs joint fine-tuning of NR-Net and DE-Net in an adversarial manner to enrich image details in the final results. For simplicity of expression, we use I_{gt} to denote the ground truths in both sRGB and raw domains. When calculating a loss, we assume that the ground truth is already transformed to the same domain as the corresponding network output.

Step 1: Pre-training NR-Net and DE-Net. For NR-Net, we pre-train it as a normal deep denoiser with an ℓ_1 loss function such as $\mathcal{L}_{s1,nr} = \mathcal{L}_{\ell_1}(I_d, I_{gt})$. The results may be piece-wise smooth but contain overall clean contents.

For DE-Net, we pre-train it as a super-resolution network because super-resolution is highly correlated to detail hallucination. We use the same dataset but use different settings to generate training image pairs. Specifically, we use the original clean image

I_{gt} as the ground truth. To obtain the input image, we randomly downsample I_{gt} by factors of 2, 4 or 8 and then resize them back to the original resolution. A set of ℓ_1 losses are imposed on the outputs at all resolution levels to train DE-Net, which can be depicted as:

$$\mathcal{L}_{s1,de} = \sum_{i=0}^{4} \mathcal{L}_{\ell_1}(I_e^i, I_{gt}^i) \tag{3.4}$$

where I_e^i denotes the output image at the ith resolution level of DE-Net ($I_e^0 = I_e$ is the original resolution). For level $i \neq 0$, I_e^i is obtained by adding a 1×1 convolution to the output feature map of DE-Net to reduce the channel size to 3. Correspondingly, I_{gt}^i is the ground truth image which is bilinearly downsampled to the ith resolution level. Such a multi-scale arrangement is to let DE-Net generate a detail-enriched image at every resolution level, and facilitate the progressive adversarial training in the second step.

Step 2: Joint adversarial fine-tuning. The second step performs joint adversarial training of NR-Net and DE-Net. The loss is set as $\mathcal{L}_{s2} = \lambda\mathcal{L}_{s2,nr} + (1-\lambda)\mathcal{L}_{s2,de}$. The first term $\mathcal{L}_{s2,nr} = \mathcal{L}_{\ell_1}(I_d, I_{gt})$ regularizes NR-Net with ℓ_1 loss to maintain its pre-denoising functionality. The second term $\mathcal{L}_{s2,de}$ is imposed on DE-Net, which is composed of an ℓ_1 loss $\mathcal{L}_{\ell_1}$, a perceptual loss $\mathcal{L}_{per}$ and an adversarial loss $\mathcal{L}_{ad}^G$:

$$\mathcal{L}_{s2,de} = \lambda_1\mathcal{L}_{\ell_1} + \lambda_2\mathcal{L}_{per} + \lambda_3\mathcal{L}_{ad}^G \tag{3.5}$$

The perceptual loss $\mathcal{L}_{per}$ in (3.5) measures the ℓ_1 distance between the feature representations in the pre-trained VGG-19 network of two images, which can improve the perceptual quality of important structures. Similar to [116], we use the convolutional layers before activations to compute the loss, which can provide dense supervision[1].

For the adversarial loss $\mathcal{L}_{ad}^G$ in (3.5), we adopt a relativistic GAN loss [116], which has the following formulation:

$$\begin{aligned}\mathcal{L}_{ad}^G(x_f, x_r) = &- \mathbb{E}_{x_r}\left[1 - log(D(x_r, x_f)\right] \\ &- \mathbb{E}_{x_f}\left[log(D(x_f, x_r))\right]\end{aligned} \tag{3.6}$$

[1]We use both low and high level features "22" and "54" for the loss computation, where "xy" denotes the xth convolution before the yth max-pooling.

where x_f and x_r denote the fake and real data, respectively, and $D(\cdot)$ denotes the estimated relative authenticity between x_f and x_r:

$$\begin{cases} D(x_r, x_f) = \sigma(D_c(x_r) - \mathbb{E}_{x_f}[C(x_f)]) \\ D(x_f, x_r) = \sigma(D_c(x_f) - \mathbb{E}_{x_r}[C(x_r)]) \end{cases} \tag{3.7}$$

where $\sigma(\cdot)$ denotes the sigmoid function, $D_c(\cdot)$ denotes the non-transformed discriminator output, and $\mathbb{E}_{x_f}[\cdot]$ denotes the operation of averaging the fake images in a mini-batch. The loss for the discriminator network is simply the symmetric form of the generator loss in (3.6):

$$\begin{aligned} \mathcal{L}_{ad}^{D}(x_f, x_r) = &- \mathbb{E}_{x_r}\left[log(D(x_r, x_f))\right] \\ &- \mathbb{E}_{x_f}\left[1 - log(D(x_f, x_r))\right] \end{aligned} \tag{3.8}$$

Different from the standard GAN [47] where the discriminator predicts the samples as absolutely real or fake, the discriminator in relativistic GAN predicts whether a real sample is more realistic than a fake one. This arrangement can leverage both real and fake samples and provide better supervision for the training of the generator.

Directly optimizing the adversarial losses in (3.5) and (3.8) at the original image resolution is difficult and can be unstable. We adopt a progressive training scheme [64], where the generator and discriminator are trained synchronously from low to high resolutions. Specifically, starting from the lowest (i.e., the 4th) resolution level of the DE-Net and the discriminator, we train their modules at the $(i+1)$th resolution level for T_{i+1} iterations with the pair of adversarial losses (3.5)(3.8), and then go to the ith level until reaching the original resolution. When moving from one resolution level to another, we adopt a "fade-in" layer [64], as shown in Fig. 3.2, which avoids degrading the learned module for previous resolution level. Specifically, the loss for DE-Net minimizes $\mathcal{L}_{s2,de}(\hat{I}_e^i, I_{gt}^i)$ from resolution level $i = 4$ to $i = 0$, where I_{gt}^i is the ground truth bilinearly resized to ith resolution level. $\hat{I}_e^i$ denotes the output image of DE-Net, which is combined by the image at the current resolution level I_e^i and the upsampled image at previous level I_e^{i+1}:

Algorithm 3.1 Adversarial fine-tuning of NR-Net and DE-Net.

Markers: $i \in \{0, 1, 2, 3, 4\}$: resolution level of DE-Net; T_i: number of training iterations at resolution i; α: weight in fade-in layer; I_e^i: the image reconstructed at the ith level of DE-Net; I_{gt}^i: the ground truth resized to the ith resolution level.

Optimization:

for $i = 4$ to 0 **do**
  Initialize $\alpha = 0$
  if $i = 4$ **then**
    $\hat{I}_e^i = I_e^i$
  else
    $\hat{I}_e^i = \alpha \cdot I_e^i + (1 - \alpha) \cdot up(I_e^{i+1})$
  end if
  for $iter = 0$ to T_i **do**
    1. Update from the $i + 1$th to the ith level modules of SAE-Net with the gradient of
      $\lambda \mathcal{L}_{\ell_1}(I_d, I_{gt}) + (1 - \lambda)\mathcal{L}_{s2,de}(\hat{I}_e^i, I_{gt}^i)$
    2. Update from the $i + 1$th to the ith level modules of the discriminator with the gradient of
      $\mathcal{L}_{ad}^D(\hat{I}_e^i, I_{gt}^i)$
    3. $\alpha = min(1, \frac{2}{T_i} iter)$
  end for
end for

$$\hat{I}_e^i = \begin{cases} \alpha \cdot I_e^i + (1 - \alpha) \cdot up(I_e^{i+1}), & i \in \{0, 1, 2, 3\} \\ I_e^i, & i = 4 \end{cases} \tag{3.9}$$

The parameter α is the weight that grows from 0 to 1 in the training, gradually switching the training from the i+1th to the ith level module of DE-Net (i.e., increasing resolution). Meanwhile, the discriminator loss minimizes $\mathcal{L}_{ad}^D(\hat{I}_e^i, I_{gt}^i)$. Similar to the DE-Net, the training of the discriminator, which also has a multi-scale structure, progresses from the $i + 1$th to the ith resolution level. The discriminator takes $\alpha \hat{I}_e^i$ as the input for its ith level module, while taking the downsampled $(1 - \alpha)\hat{I}_e^i$ for its $i + 1$th level. The discrimination on the input image switches from $i + 1$th to the ith

resolution level as α grows from 0 to 1. The above training procedure is summarized in Algorithm 3.1.

3.3 Experiments

We conduct experiments to verify the effectiveness of the proposed SAE-Net on both synthetic and real-world noisy images quantitatively and qualitatively. Three metrics are used for objective evaluation, including the peak signal-to-noise ratio (PSNR), the perceptual index (PI) used in the PIRM-SR Challenge [14] and the LPIPS metric [135]. While PSNR simply measures the pixel-wise errors between two images, PI and LPIPS are metrics designed for human perception. In particular, PI is a no-reference metric and a lower PI indicates better perceptual quality. LPIPS compares the feature representations of two images in a pre-trained classification network. A smaller LPIPS means the image is visually closer to the ground truth, hence better perceptual quality.

3.3.1 Experimental Settings

Datasets. For the experiments on AWGN removal, we use the combination of DIV2K [2], Flickr2K [109] and OST [111] datasets as training data. The three datasets have 800, 2,650 and 10,324 high quality images, respectively. Following previous works [131, 23], we choose CBSD68, Kodak24, McMaster and Urban100 as the testing sets.

For the experiments on real-world noise removal, we select 3,600 high-quality images with low ISO from the FiveK dataset [18] to create the training data because they have high resolution and rich details. We adopt the ISP unprocessing method [15] to synthesize the training pairs in raw or sRGB domain. We also include the SIDD medium set and Renoir dataset for training as a complement because they contain real-world noisy-clean image pairs. The testing set is composed of two parts. The first is the SIDD validation set which contains 1280 noisy sRGB patches from

Table 3.1: Gaussian denoising results in PI/LPIPS. The values in bold letter indicate the best scores."-" means that the result is not available.

Datasets	σ	DnCNN	FFDNet	IRCNN	RNAN	RIDNet	SADNet	SMNet	SAE-Net
CBSD68	30	2.67/0.148	2.85/0.147	2.47/0.117	2.91/0.119	3.11/0.152	2.95/0.111	2.76/0.144	**2.56**/**0.096**
	50	2.79/0.210	3.30/0.241	2.69/0.203	3.24/0.200	3.43/0.229	3.21/0.180	3.12/0.188	**2.50**/**0.147**
	70	3.26/0.316	3.75/0.313	-	3.52/0.264	4.07/0.319	3.41/0.234	3.15/0.306	**3.11**/**0.181**
Kodak24	30	2.63/0.172	2.81/0.168	2.34/0.139	2.60/0.132	2.86/0.173	2.63/0.124	2.51/0.169	**2.34**/**0.110**
	50	2.70/0.237	3.30/0.259	2.57/0.225	2.94/0.210	3.22/0.247	2.84/0.187	2.93/0.205	**2.35**/**0.154**
	70	3.32/0.338	3.81/0.331	-	3.23/0.271	4.03/0.329	3.05/0.234	3.12/0.334	**3.04**/**0.184**
McMaster	30	3.49/0.159	3.75/0.156	3.07/0.138	3.51/0.138	3.74/0.167	3.50/0.133	3.28/0.143	**2.91**/**0.124**
	50	3.38/0.207	4.18/0.215	3.20/0.194	4.00/0.196	4.28/0.222	4.04/**0.148**	3.86/0.183	**2.95**/0.156
	70	4.12/0.268	4.59/0.268	-	4.35/0.240	4.93/0.275	4.41/0.228	**3.42**/0.254	3.69/**0.181**
Urban100	30	3.99/0.088	4.32/0.084	3.84/0.073	4.25/0.066	4.27/0.085	4.16/0.062	4.00/0.083	**3.74**/**0.062**
	50	3.80/0.136	4.39/0.136	3.76/0.123	4.27/0.107	4.29/0.131	4.20/0.095	4.32/0.108	**3.71**/**0.094**
	70	3.80/0.224	4.44/0.186	-	4.26/0.144	4.42/0.188	4.25/0.127	**3.75**/0.200	4.07/**0.116**

indoor images. To test the performance in challenging outdoor scenario, we capture 20 noisy raw images with vivo NEX 3S in nighttime scenario as the second test part, which is called vivo testing set.

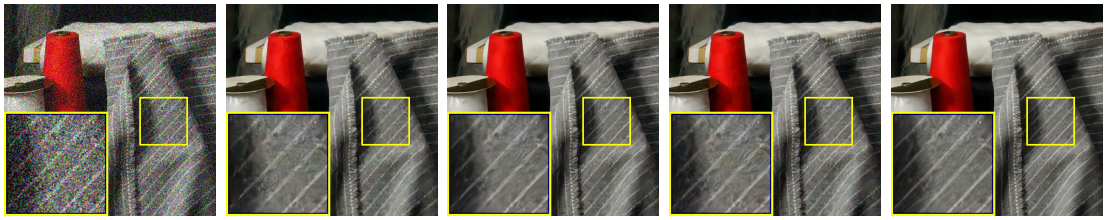
Training details. The kernel sizes of SAE-Net and the discriminator are set to 3×3. We crop non-overlapping 256×256 patches from each training image and exclude those smooth patches, yielding 133,854 patches in total for training. The batch size is set to 16 and the number of epochs is set to 25 for each training step. We use the Adam optimizer ($\beta_1 = 0.5, \beta_2 = 0.99$) [67] for network optimization. The initial learning rate is set to 1e-4 and it is exponentially decayed by 0.1 at the $\frac{3}{4}$th epoch for step 1, and is fixed at step 2. At step 2, the update iterations for the resolution levels$\{T_0, T_1, T_2, T_3, T_4\}$ are $\{80000, 40000, 40000, 40000, 10000\}$. The weights λ, λ_1, λ_2,λ_3 are set to 0.5, 0.01, 1 and 0.005, respectively.

3.3.2 Results on AWGN Noise Removal

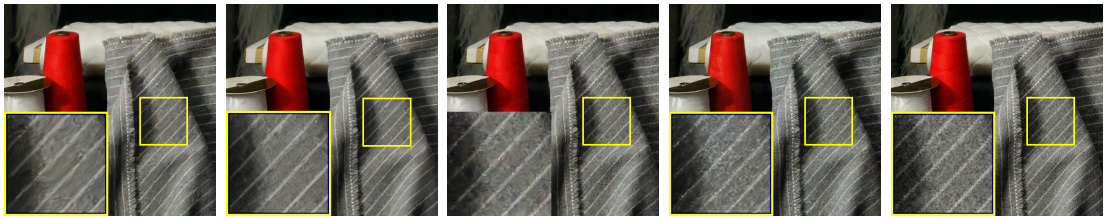
We firstly evaluate the performance of SAE-Net on images corrupted by AWGN with noise levels σ =30, 50 and 70. We compare with the state-of-the-art denoising

Table 3.2: Gaussian denoising results in PSNR(dB). The values in bold letter indicate the best scores."-" means that the result is not available.

Datasets	σ	DnCNN	FFDNet	IRCNN	RNAN	RIDNet	SADNet	SMNet	SAE-Net
CBSD68	30	29.9	30.2	30.1	30.7	30.3	**30.7**	30.1	28.8
	50	27.9	27.9	27.9	28.2	28.0	**28.3**	28.1	27.1
	70	26.1	26.5	-	26.8	26.5	**26.9**	26.1	25.3
Kodak24	30	31.1	31.4	31.2	**31.9**	31.4	31.8	31.1	29.5
	50	28.8	28.9	28.9	29.5	29.1	**29.6**	29.4	28.0
	70	27.1	27.61	-	28.2	27.6	**28.3**	27.2	26.3
McMaster	30	31.1	31.5	31.3	32.1	31.4	**32.0**	31.3	29.4
	50	28.6	29.2	28.9	29.7	29.1	**29.7**	29.6	28.0
	70	26.9	27.6	-	28.2	27.6	**28.3**	27.1	26.0
Urban100	30	29.9	30.5	30.3	**31.5**	30.4	31.3	30.3	28.7
	50	27.6	28.1	27.7	**29.1**	28.1	29.0	28.8	26.8
	70	25.3	26.4	-	27.4	26.2	**27.5**	25.8	24.9



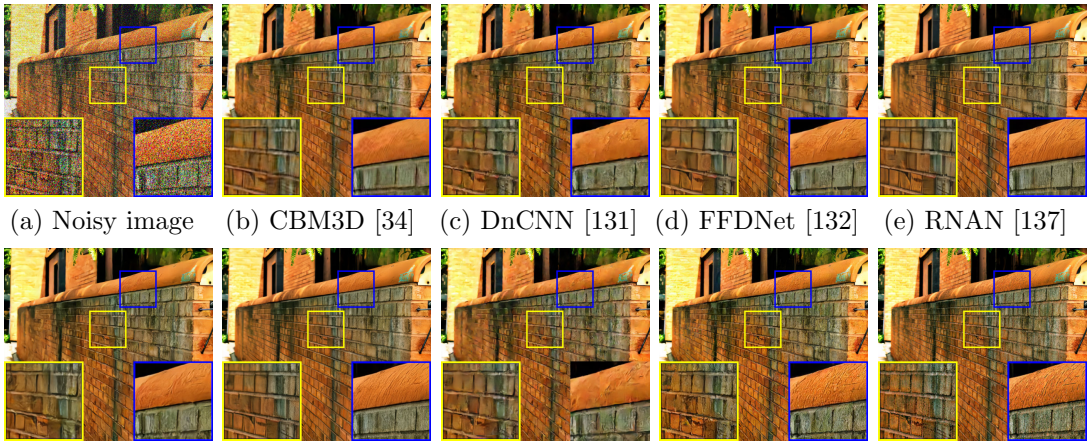
(a) Noisy image (b) DnCNN [131] (c) FFDNet [132] (d) IRCNN [133] (e) RNAN [137]



(f) RIDNet [4] (g) SADNet [23] (h) SMNet [93] (i) SAE-Net (j) Ground truth

Figure 3.3: The Gaussian denoising results ($\sigma = 50$) on an image from the McMaster testing set.

methods, including CBM3D [34], DnCNN [131], FFDNet [132], IRCNN [133], RNAN [137], RIDNet [4], SADNet [23] and SMNet [93]. Except CBM3D, all the compared methods are deep CNN-based denoisers. The detailed configurations of the compared methods are described as follows.



(a) Noisy image (b) CBM3D [34] (c) DnCNN [131] (d) FFDNet [132] (e) RNAN [137]

(f) RIDNet [4] (g) SADNet [23] (h) SMNet [93] (i) SAE-Net (j) Ground truth

Figure 3.4: The Gaussian denoising results ($\sigma = 70$) on an image from the Urban100 testing set.

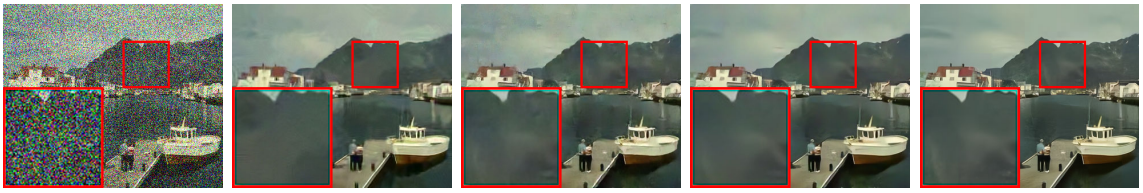
1) For FFDNet [132], RNAN [137] and SADNet [23], we use the available pre-trained models by the original authors for testing on σ=30, 50 and 70.

2) For DnCNN [131] and SMNet [93], pre-trained models for σ =50 are available. For σ =30 and 70, we train DnCNN and SMNet from scratch by using the original codes and select the models with the best testing PSNR.

3) For IRCNN [133], the pre-trained models for σ =30 and 50 are available. Since the training code is not publically available and is difficult to implement, we do not report its performance on σ =70.

4) For RIDNet [4] which has testing code only, we train it on our training dataset for σ = 30, 50 and 70 with the same setting described in the original paper. The models with the best testing PSNR are selected.

Table 3.2 and Table 3.2 illustrates the quantitative results of the compared methods in terms of PSNR and PI/LPIPS, respectively. We have several observations. First, SAE-Net obtains the best PI and LPIPS scores on almost all testing sets and noise



(a) Noisy image (b) CBM3D [34] (c) DnCNN [131] (d) FFDNet [132] (e) RNAN [137]

(f) RIDNet [4] (g) SADNet [23] (h) SMNet [93] (i) SAE-Net (j) Ground truth

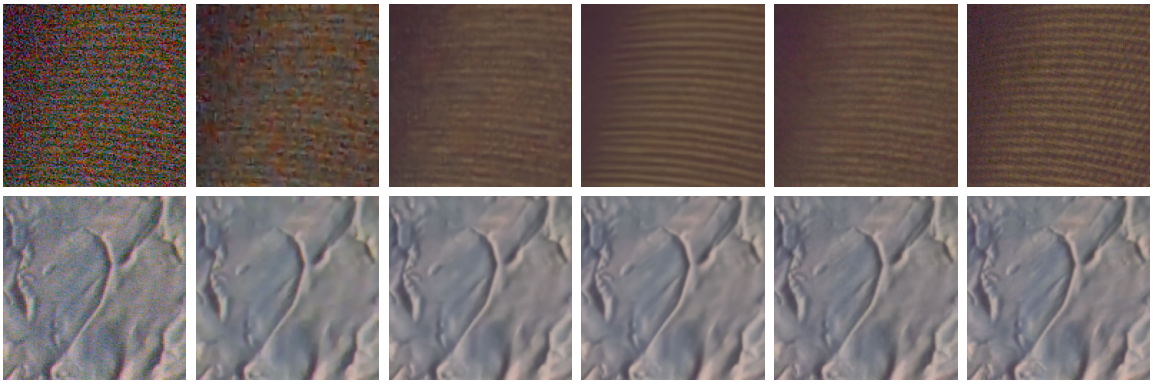
Figure 3.5: The Gaussian denoising results (σ = 70) on an image from the CBSD68 testing set.

levels. This indicates that the results of SAE-Net are perceptually closer to the ground truths (in term of LPISP) and they have very high perceptual quality (in term of PI). Second, as the noise level increases from σ=30 to 70, the gap between SAE-Net and the competing methods becomes more significant in terms of LPIPS. This indicates that SAE-Net can better maintain the image perceptual quality. Third, the proposed SAE-Net has the lowest PSNR scores in all cases. This is because our network is to minimize the perceptual distortion by using adversarial training, rather than optimize the PSNR index.

Fig. 3.3 shows the denoising results of SAE-Net and the compared methods on σ=50. It can be seen that SAE-Net achieves the best perceptual quality by synthesizing realistic textures on the cloth surface, whereas the competing methods produce piecewise smooth results (e.g., RNAN, RIDNet) or irregular visual artifacts (e.g., DnCNN, FFDNet, RIDNet). The advantage of SAE-Net becomes larger as noise level increases. Figs. 3.4 and 3.5 show the results on noise level σ=70. We can see that under such a high noise level, all the competing methods blur severely the image details, while SAE-Net can still successfully recover part of the missing details on the

Table 3.3: Quantitative results of Real-world denoising. For SIDD validation set with ground truths available we evaluate PSNR/PI/LPIPS, while for vivo testing set without ground truth we evaluate PI. The values in bold letter indicate the best results. "-" means the result is not available.

	CBDNet[52]	UTR[15]	RIDNet[4]	SADNet[23]	SAE-Net
SIDD	14.46/13.12.11/0.321	-	38.71/11.81/0.221	**39.45**/11.98/0.207	37.22/**11.24**/**0.147**
vivo test set	6.52	7.94	6.36	6.34	**6.16**



(a) Noisy image (b) CBDNet[52] (c) RIDNet[4] (d) SADNet[23] (e) SAE-Net (f) Ground truth

Figure 3.6: Real-world denoising results of the compared methods on SIDD validation set. Best viewed on screen with zoom-in.

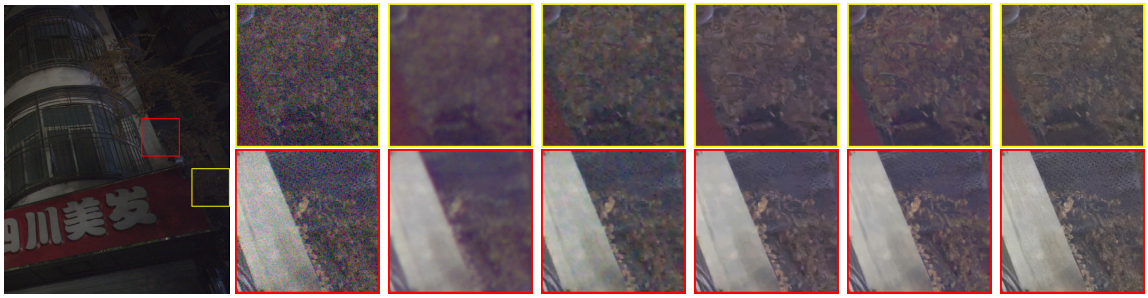
mountain areas. This validates that our smooth and enhance strategy is robust to noise level and it can effectively introduce extra details in the enhancement stage.

3.3.3 Results on Real-world Noise Removal

We compare SAE-Net with state-of-the-art real-world denoising methods, including CBDNet [52], UTR [15], RIDNet [4] and SADNet [23]. Other denoising methods in Section 3.3.2 are not included because they are not designed for real-world noise removal. The evaluations are performed on SIDD validation set and the vivo testing set.

(a) (b) (c) (d) (e) (f) (g)

Figure 3.7: Real-world denoising results of the compared methods on vivo testing set. The images from left to right are (a) Noisy raw image. (b) Noisy patches. (c) UTR[15]. (d) CBDNet[52]. (e) RIDNet[4]. (f) SADNet[23]. (g) SAE-Net. All denoised raw images are transformed to sRGB space for display. Best viewed on screen with zoom-in.



(a) (b) (c) (d) (e) (f) (g)

Figure 3.8: Real-world denoising results of the compared methods on vivo testing set. The images from left to right are (a) Noisy raw image. (b) Noisy patches. (c) UTR[15]. (d) CBDNet[52]. (e) RIDNet[4]. (f) SADNet[23]. (g) SAE-Net. All denoised raw images are transformed to sRGB space for display. Best viewed on screen with zoom-in.

SIDD validation set. Following some previous works [4, 23], the denoising and evaluation are both conducted in sRGB space. We use the publicly available pre-trained models of CBDNet, RIDNet and SADNet, for testing. For SAE-Net, we first train it on the FiveK subset (with 3600 sRGB images) with our two-step training strategy because FiveK subset contains diverse images for training a GAN. The original sRGB images in FiveK are treated as the ground truth. To acquire the noisy sRGB images, we firstly transform the ground truth I to raw images I_r by

reversing several ISP operations with the method in [15], including gamma correction $G(\cdot)$, color space conversion $C(\cdot)$, white balance $W(\cdot)$ and demosaicking $D_m(\cdot)$, which can be expressed as

$$I_r = D_m^{-1}(W^{-1}(C^{-1}((G^{-1}(I))))) \tag{3.10}$$

Then we add Poisson-Gaussian noise to the raw images, denoted as

$$I_{nr} = n(I_r) \tag{3.11}$$

where $n(\cdot)$ denotes the noise corruption. Lastly we transform the noisy raw images back to sRGB space as the input for SAE-Net:

$$I_r = G(C(W((D_m(I_{nr}))))) \tag{3.12}$$

After training on FiveK, we perform adversarial fine-tuning without progressive scheme on the SIDD median set and Renoir dataset to adapt to the real-world noisy sRGB images. From Fig. 3.6 we can see that SAE-Net can recover high-frequency details in the object surface, while other methods produce slightly over-smoothed results.

vivo testing set. For the vivo testing set, the denoising is performed in raw space and the results are converted to sRGB space for evaluations. We train UTR, RIDNet, SADNet and SAE-Net on the FiveK subset. We apply the ISP unprocessing operations in (3.2) and (3.3) to the sRGB images in FiveK to synthesize the noisy raw images, which are then initially demosaicked by $D_m(\cdot)$ as network inputs. We have different configurations on the network outputs, which are described as follows:

1) For UTR, RIDNet and SADNet, the network outputs are in raw RGB space and are transformed to sRGB space using forward ISP operation in (3.4) (except demosaicking $D_m(\cdot)$) for loss calculation.

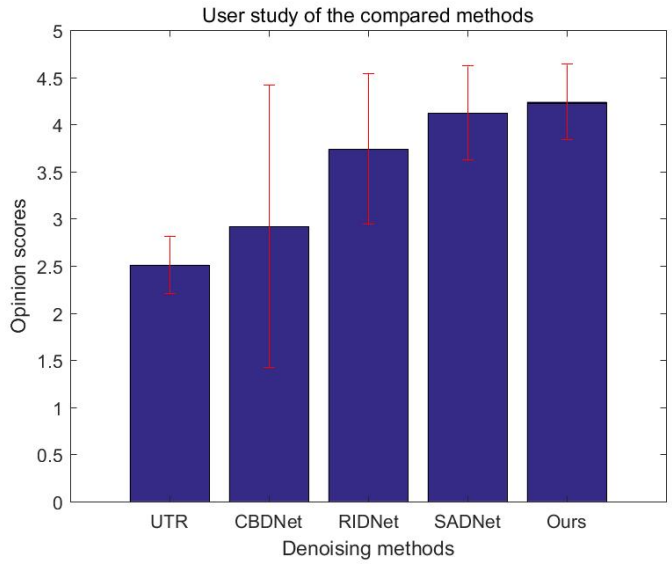


Figure 3.9: Comparison of human opinion scores on Real-dynamic test set.

2) For SAE-Net, the output of NR-Net is in raw space and is transformed by the operations in (3.4) (except demosaicking $D_m(\cdot)$) to sRGB space as the input for DE-Net. Finally, DE-Net outputs an enhanced sRGB image.

3) For CBDNet which operates in sRGB space, we transform the noisy raw images to sRGB space before denoising.

Since there is no ground truths for this testing set, we compare the PI scores and conduct visual comparison. From Table 3.3, we can see that SAE-Net obtains the best PI score. From the visual comparison in Fig. 3.7 and 3.8, we can see that SAE-Net can recover many realistic details on the tree trunk region. In contrast, while the other methods can remove most of the noise, they blur the image details to different degrees and lead to lower visual quality.

To further evaluate the visual quality on vivo testing set, we perform a user study, where 10 subjects (5 males and 5 females) are requested to rate the quality of the denoised images. The score ranges from 1 (the worst) to 5 (the best) with

Table 3.4: Quantitative results (PI/LPIPS) of different variants of SAE-Net on AWGN removal with $\sigma = 50$.

	OS	OS-GAN	OS-ProGAN	TS-GAN	SAE-Net
CBSD68	3.47/0.246	3.16/0.189	3.14/0.171	3.02/0.160	**2.50/0.147**
Kodak24	3.26/0.257	3.04/0.212	3.04/0.184	2.82/0.180	**2.35/0.154**

decimal interval. The denoised image by each denoiser is shown in random order by a MATLAB program on a PA328 display with 32 inch (7680×4320), controlled by a Mac Pro PC with 2.9 GHz CPU. The program allows zooming in to the same regions simultaneously for all the compared results. The subjects are not acknowledged of the denoising algorithms involved in the experiment. Before the evaluation, the subjects are taught how to use the programs, especially in zooming the image to inspect local image details. The mean opinion score statistics are illustrated in Fig 3.9 and table 2.1. We can see that our tone mapper achieves the highest mean scores(4.24) and a tolerable variation(1.1).

3.3.4 Ablation Study

To better validate the effectiveness of our smooth and enhance strategy, we compare SAE-Net with its variants trained by different training strategies. In particular, OS denotes a one-stage network trained with pixel-wise loss[2]. OS-GAN denotes the OS model fine-tuned with normal adversarial loss. OS-ProGAN denotes the OS model fine-tuned with scale progressive adversarial loss. TS-GAN denotes the two-stage SAE-Net fine-tuned with normal adversarial loss in the second training step. All the compared methods are trained on the AWGN training set with $\sigma = 50$ and evaluated on CBSD68 and Kodak24 testing sets. For each setting, we select the model with the best LPIPS for evaluation.

[2]We double the convolutions blocks in NR-Net to make OS have comparable amount of parameters to SAE-Net.

(a) OS (b) OS-GAN (c) OS-ProGAN (d) TS-GAN (e) SAE-Net (f) GT

Figure 3.10: The AWGN removal results ($\sigma = 50$) between different learning schemes on two images from CBSD68 testing set. The top and bottom rows are cropped patches of images "163085" and "105025" from CBSD68, respectively.

Table 3.4 illustrates the PI and LPIPS scores. We can see that SAE-Net obtains the best PI/LPIPS scores. Meanwhile, the two-stage structures (TS-GAN, SAE-Net) outperform the one-stage structures (OS-ProGAN, OS-GAN, OS), indicating that the smooth and enhance strategy is more effective than the one-stage strategy for improving the perceptual quality. Fig. 3.10 shows the results of the compared schemes. We can see that the two-stage structures (TS-GAN and SAE-Net) can generate more details than the one-stage ones. In contrast, the one-stage OS-GAN produces many noise-like artifacts and color errors.

3.4 Summary

In image denoising, it is a challenging problem to remove the noise while reconstructing the image details since noise and image details are both high-frequency components. In this chapter, we propose a smooth and enhance strategy for detail-enriched image denoising. We explicitly divide the denoising process into a noise removal and a detail enhancement stages, which are characterized by two subnetworks. The noise removal subnetwork focuses on removing noise while the detail enhancement

subnetwork hallucinates the lost image details. We developed a training scheme which first performs task-specific pre-training of the two subnetworks and then performs adversarial fine-tuning of the whole network to approximate the manifold of natural images. The trained network, call SAE-Net, can effective remove the noise and generate realistic image details. Experiments on Gaussian and real-world noise removal demonstrated that SAE-Net has leading performance on improving the perceptual quality of the denoised images.

Chapter 4

A Decoupled Learning Scheme for Real-world Deep Burst Denoising

The recently developed burst denoising approach, which reduces noise by using multiple frames captured in a short time, has demonstrated much better denoising performance than its single-frame counterparts. However, existing learning based burst denoising methods are limited by two factors. On one hand, most of the models are trained on video sequences with synthetic noise. When applied to real-world raw image sequences, visual artifacts often appear due to the different noise statistics. On the other hand, there lacks a real-world burst denoising benchmark of dynamic scenes because the generation of clean ground-truth is very difficult due to the presence of object motions. In this chapter, a novel multi-frame CNN model is carefully designed, which decouples the learning of motion from the learning of noise statistics. Consequently, an alternating learning algorithm is developed to learn how to align adjacent frames from a synthetic noisy video dataset, and learn to adapt to the raw noise statistics from real-world noisy datasets of static scenes. Finally, the trained model can be applied to real-world dynamic sequences for burst denoising. Extensive experiments on both synthetic video datasets and real-world dynamic sequences demonstrate the leading burst denoising performance of our proposed method.

4.1 Introduction

The imaging quality of smartphone cameras is much affected by the small aperture and small CMOS sensor, which limit the amount of collected light and result in heavy noise in the raw images. Denoising is a crucial step in the camera image processing pipeline (ISP) to remove the noise and reveal the latent image details. The denoising algorithms can be divided into single-frame denoising methods [33, 131, 52, 4] and burst denoising methods [143, 53, 87, 46]. While the former ones take a single-frame image as input for processing and are easier to implement, their denoising performance is limited, especially under the low-light environment. The recently developed burst denoising methods capture multiple frames in a short time as input, and thus they can leverage more redundant information for noise removal, leading to much better denoising quality.

The burst denoising problem can be addressed by hand-crafted methods [53, 32, 82, 33, 143] or learning-based methods [87, 121, 46]. The traditional hand-crafted algorithms are often manually designed to exploit the spatio-temporal similarities. For example, the well-known VBM3D method [32] denoises an image patch by finding and fusing its similar patches in the adjacent frames. In contrast, the learning-based methods train a denoising model by using pairwise datasets with a noisy image sequence as input and a clean image as ground-truth. In particular, the rapid development of deep convolutional neural networks (CNNs) [87, 121, 46] largely facilitate the research of learning based burst denoising. The CNN model is powerful to learn a set of nonlinear transformations from the noisy input to the clean output, including frame alignment, fusion and post processing, achieving superior performance to traditional burst denoising methods.

Despite the great progress, the learning-based burst denoising methods are limited by two factors. On one hand, the current multi-frame CNN models are mostly trained

on video datasets with synthetic noise, e.g., Gaussian or Poisson-Gaussian noises. When the learned models are applied to real-world raw image sequences, whose noise distribution and statistics are more complex, unpleasant visual artifacts such as color shift and residual noise will appear. One the other hand, there lacks a real-world dataset for learning burst denoising models of dynamic sequence. This is mainly because in the presence of scene motion (e.g., hand shake motion and object motion), it is difficult to craft a clean ground-truth frame by using existing ground-truth generation techniques, such as using low ISO setting [24] or averaging multiple frames [1]. Misalignment problem will occur, which significantly degrades the quality of ground-truth. It is highly desirable to develop a burst denoising CNN model that can adapt to the real-world noise statistics without the need of a real-world pairwise burst image dataset.

There are two key issues in designing such a burst denoising CNN model. Firstly, to enable multi-frame processing, the CNN model should be able to align input frames to compensate the scene motion caused by hand shake and object movement in real scenarios. Second, the CNN model should be able to adapt to real-world noise for better generalization to real-world burst images. Based on the above considerations, we propose a decoupled learning framework for real-world burst denoising. First, a novel multi-frame CNN model is carefully designed with modular architecture which decouples the learning of motion from the learning of noise adaption. Second, an alternative learning algorithm is developed to leverage the complementary information from two datasets we prepared. One is a video dataset with synthetic noise, where the model learns to perform frame alignment, while the other is a real-world burst image dataset of static scenes, from which the model learns to adapt to raw noise statistics. With the designed CNN model and our decoupled learning algorithm, the learned CNN model achieves leading performance in real-world burst denoising without the need of a pairwise real-world burst dataset for training.

4.2 Decoupled Learning Network for Burst Denoising

Given a sequence of N noisy raw images (e.g., in the Bayer color filter array (CFA) pattern [11]) captured by a handheld camera, denoted by $\boldsymbol{I} = \{I_1, I_2, ..., I_N\}$, our goal is to estimate a clean RGB image O from $\boldsymbol{I}$, i.e., $O = f(\boldsymbol{I}; \theta)$, where $f(\cdot; \theta)$ denotes the denoising model (e.g., a CNN model in our work) parameterized by θ. We consider one frame from $\boldsymbol{I}$ as the reference frame, denoted by I_r, and denoise it by aligning and fusing it with other frames $I_i, i \neq r$.

To denoise real-world burst image sequences of dynamic scenes, the CNN model should learn to simultaneously align frames and adapt to real-world noise from some training dataset. Considering the fact that there lacks a real-world burst image dataset of dynamic scenes with ground-truth clean images, we propose to use two types of datasets for training, which can be generated by using the publically accessible data. One is a synthetic noisy video dataset of dynamic scenes, denoted by $\mathcal{D}_d$ (subscript "d" for dynamic). Each data pair ($\boldsymbol{I}_d$,G_d) in $\mathcal{D}_d$ consists of a noisy video sequence $\boldsymbol{I}_d$ and a clean ground-truth frame G_d. The other is a real-world burst image dataset of static scenes, denoted by $\mathcal{D}_s$ (subscript "s" for static). Each data pair ($\boldsymbol{I}_s$,G_s) in $\mathcal{D}_s$ consists of a noisy raw image sequence $\boldsymbol{I}_s$ and a ground-truth clean RGB image G_s.

$\mathcal{D}_d$ can be easily built by using the many high quality video sequences [121], while $\mathcal{D}_s$ can be built by the existing frame averaging method [1]. These two datasets have complementary information. The video dataset $\mathcal{D}_d$ contains rich dynamic scene motions, but the noise is synthetic and not real. In contrast, the static burst dataset $\mathcal{D}_s$ does not contain scene motion, but can provide information of real noise statistics. We investigate how to learn a CNN model $f(\cdot; \theta)$ from $\mathcal{D}_d$ and $\mathcal{D}_s$, and present a decoupled learning scheme to achieve this goal.

4.2.1 Training Data Preparation

Before we present the CNN model architecture and the decoupled learning scheme, the two required datasets, $\mathcal{D}_d$ and $\mathcal{D}_s$, must be prepared. We present how to use the existing data to build these two datasets in this section.

Preparation of $\mathcal{D}_d$. We collect high quality video sequences from the Vimeo-90k dataset [121] to prepare $\mathcal{D}_d$. Specifically, we extract 5 consecutive frames as a burst sequence from the many videos in this dataset, yielding a total of 20000 sequences with different contents. However, directly adding noise to those sequences will make $\mathcal{D}_d$ deviate too much from the real-world dynamic noisy image sequences. Inspired by the work of [15], we propose to reverse the ISP pipeline and add noise to the reversed raw images so that the synthesized noisy sequences can be more realistic.

Specifically, we reverse four key ISP operations, including gamma correction, color space conversion, white balance and demosaicking, together with realistic noise synthesis, for building $\mathcal{D}_d$. A reverse gamma conversion with parameter γ is applied on a video frame L, where γ is sampled from a uniform distribution within range [2.0,2.6]. Then, a reverse color space conversion C is applied, with the color matrix randomly interpolated by the color matrices given in static real-world dataset $\mathcal{D}_s$. Next, a reverse white balance gain of $W = 1/(r_g, 1, b_g)$ is applied with r_g and b_g matched to the statistics in $\mathcal{D}_s$. Finally, we obtain the synthetic clean RGB image of a frame as $G = WCL^{\gamma}$.

To synthesize the noisy input, a mosaicking mask M is applied to G, yielding a Bayer CFA pattern image, denoted by G_M. Then Poisson-Gaussian noise which is approximated by heteroscedastic Gaussian [87] is added to the CFA image to synthesize noisy raw image I:

$$I = G_M + n(G_M) \tag{4.1}$$

where noise n is dependent on the signal intensity g at each location:

$$n(g) \sim \mathcal{N}(\mu = g, \sigma^2 = \lambda_{shot} g + \lambda_{read}^2) \tag{4.2}$$

where $\mathcal{N}(\mu, \sigma^2)$ is Gaussian distribution. λ_{shot} and λ_{read} are the shot noise and readout noise, which are uniformly sampled in the range (0.00001,0.01) and (0,0.058), respectively.

By the above described process, we can synthesize a sequence of noisy raw images I and take them as $\boldsymbol{I}_d$. The clean RGB image G of the center frame is taken as the ground-truth G_d. A data pair ($\boldsymbol{I}_d$,G_d) is then constructed for $\mathcal{D}_d$.

Preparation of $\mathcal{D}_s$. We use the static burst image datasets in [24, 1] to prepare dataset $\mathcal{D}_s$. We extract 140 and 162 groups of data pairs in [24] and [1], respectively. Each group contains a static noisy sequence of 5 raw images and a clean RGB ground-truth. We propose to add simple motions to the static burst sequences to facilitate the learning of frame alignment. Specifically, for a raw noisy image sequence, we add vertical and horizontal global shifts to its frames (except for its reference frame I_r):

$$\hat{I}_i = I_i(x + x_i, y + y_i), \quad for \quad i \neq r \tag{4.3}$$

where the shift x_i and y_i are uniformly sampled from the range [-4,4].

The ground-truth image G_s is already available in the static noisy image datasets [24, 1]. After adding simple motions to its adjacent noisy frames and taking them as $\boldsymbol{I}_s$, a data pair ($\boldsymbol{I}_s$,G_s) for the dataset D_s can be generated.

4.2.2 Decoupled Network Design

To achieve the goal of decoupled learning with $\mathcal{D}_d$ and $\mathcal{D}_s$, we design a modular CNN which is explicitly divided into a pre-processing (PreP) module M_p, a temporal processing module (TemP) M_t and a post-processing module (PostP) M_o. We call the proposed CNN model BDNet (burst denoising network), whose learning framework

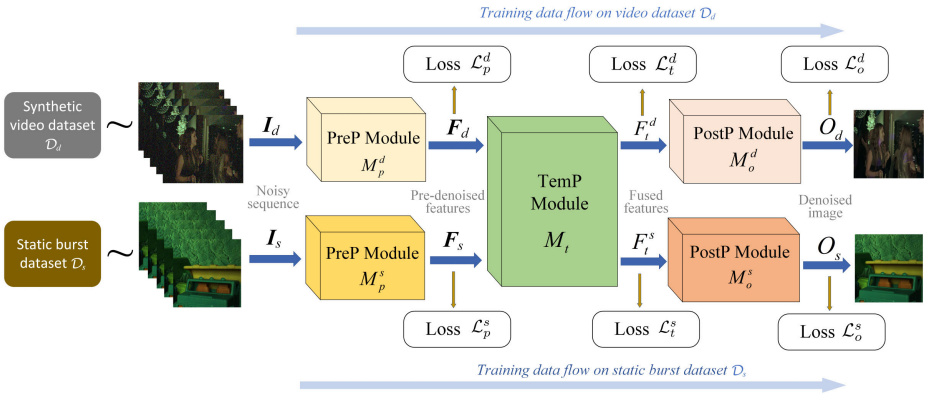
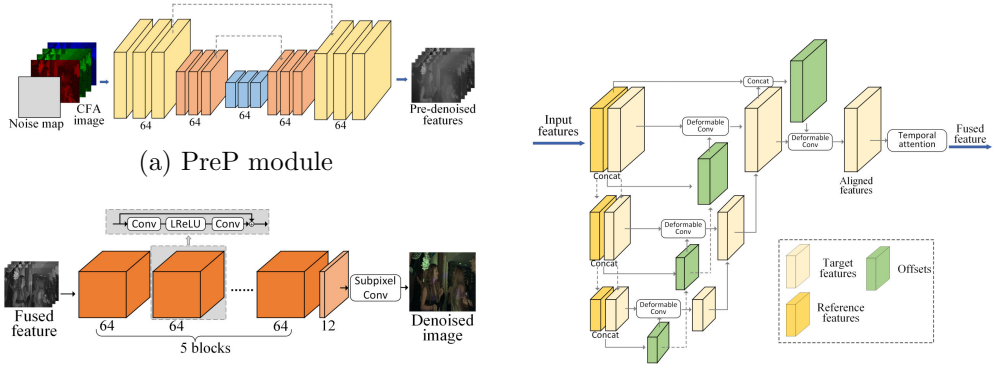


Figure 4.1: The decoupled learning framework for our burst denoising network (BDNet).



(a) PreP module

(b) TemP module

(c) PostP module

Figure 4.2: The structure of the PreP module M_p, TemP module M_t and PostP module M_o of the proposed BDNet.

is illustrated in Fig. 4.1. The detailed structures of modules M_p, M_t and M_o are illustrated in Fig. 4.2.

Pre-processing module. The PreP module M_p is constructed to perform single-frame denoising on the noisy CFA sequence $\boldsymbol{I} = \{I_1, I_2, ..., I_N\}$ and output pre-denoised features $\boldsymbol{F} = \{F_1, F_2, ..., F_N\}$. In addition, we add a noise level as input, which is obtained by $\sqrt{\lambda_{shot} + \lambda_{read}^2}$. We adopt a multi-scale (three scales) UNet [103] with 15 convolutional layers for single image denoising for its simplicity and good performance. As shown in Fig. 4.2(a), the adopted UNet consists of a contracting path

which continuously downsamples the image features with stride convolutions, and an expanding path that gradually upsamples the features to the original resolution. Skip connections are added between the contracting and expanding paths at the same scale level. The PreP module M_p not only helps to reduce the noise but also increases the robustness in the subsequent frame alignment operation.

Temporal processing module. The TemP module M_t is constructed to align and fuse the pre-denoised features $\boldsymbol{F} = \{F_1, F_2, ..., F_N\}$ and output a single feature map F_t. It has been shown that accurate alignment can be obtained with deformable convolutions [115]. Thus, we adopt the Pyramid, Cascading and Deformable alignment (PCD) model and temporal attention methods in [115] as the alignment and fusion components in our TemP module, respectively. As shown in Fig. 4.1(b), the PCD takes a pair of reference and target features as input, and progressively warps the target feature to the reference feature in a multi-scale and cascading manner. The temporal attention component fuses all the aligned features according to their similarities to the reference feature.

Post-processing module. The PostP module M_o takes the fused feature F_t as input and conducts some refinement operations to reconstruct a clean image. As shown in Fig. 4.1(c), we deploy 5 residual blocks to build M_o, each containing two convolutional layers. Then a 1×1 convolutional and a sub-pixel convolutional layer are applied to output the denoised RGB image O.

4.2.3 Decoupled Learning Process

Given the BDNet model in Section 4.2.2 and the two prepared datasets 4.2.1 $\mathcal{D}_d$ and $\mathcal{D}_s$ in Section 3.2, the remaining question is how to effectively learn frame alignment and real-world noise adaptation for burst denoising. We propose a decoupled learning method to this end, which is illustrated in Fig. 4.1.

First, considering that the noise statistics in the dynamic video dataset $\mathcal{D}_d$

(synthetic noise) and static burst dataset $\mathcal{D}_s$ (real-world noise) are different, different CNN modules should be deployed for each case to avoid mixed learning. Therefore, we train and deploy two instances of the PreP module M_p with the same architecture but different parameters. These two module instances, denoted by M_p^d and M_p^s, transform the synthetic noisy sequences $\boldsymbol{I}_d$ (from $\mathcal{D}_d$) and real-world noisy sequences $\boldsymbol{I}_s$ (from $\mathcal{D}_s$) to pre-denoised feature sequences $\boldsymbol{F}_d$ and $\boldsymbol{F}_s$, respectively. We assign a pair of sub-losses, denoted by $\mathcal{L}_p^d$ and $\mathcal{L}_p^s$, for the pre-denoising modules

$$\begin{cases} \min \mathcal{L}_p^d\big(G_d, \mathit{Recon}_1(F_{d,r})\big) \\ \min \mathcal{L}_p^s\big(G_s, \mathit{Recon}_1(F_{s,r})\big)\big) \end{cases} \tag{4.4}$$

where $F_{d,r}$ and $F_{s,r}$ are the reference feature maps in the pre-denoised feature sequences $\boldsymbol{F}_d$ and $\boldsymbol{F}_s$, respectively. This pair of sub-losses $\mathcal{L}_p^d$ and $\mathcal{L}_p^s$ (e.g., ℓ_1 loss) calculate the errors between the ground-truths G_d, G_s from the two datasets and the images reconstructed from the pre-denoised reference features $F_{d,r}$, $F_{s,r}$, respectively. The reconstruction operation Recon_1 is performed by a shared 1×1 convolution that reduces the channel size, followed by a sub-pixel convolution to expand to the original resolution. Since the features are initially denoised, they are in a relatively clean signal space, which facilitates the subsequent frame alignment learning.

Second, we deploy one TemP module M_t to receive the feature sequences $\boldsymbol{F}_d$ and $\boldsymbol{F}_s$, perform frame alignment and fusion, and output the fused features F_t^d and F_t^s, respectively. Since both $\boldsymbol{F}_d$ and $\boldsymbol{F}_s$ are in a relatively clean latent space, the learned frame alignment capability of $\boldsymbol{F}_d$ can be transfered to $\boldsymbol{F}_s$. A pair of sub-losses, denoted by $\mathcal{L}_t^d$ and $\mathcal{L}_t^s$, are deployed on M_t:

$$\begin{cases} \min \mathcal{L}_t^d\big(G_d, \mathit{Recon}_2(F_t^d)\big) \\ \min \mathcal{L}_t^s\big(G_s, \mathit{Recon}_2(F_t^s)\big) \end{cases} \tag{4.5}$$

The sub-losses compare the ground-truths G_d and G_s with the images reconstructed

from the fused features F_t^d and F_t^s, respectively. The reconstruction operation $Recon_2$ consists of a shared 1×1 convolution followed by a sub-pixel convolution.

Third, considering that the dataset $\mathcal{D}_d$ is generated by reversing the ISP, while the images in dataset $\mathcal{D}_s$ are collected in the real raw image domain, the ground-truth images of the two datasets may have some appearance differences. In particular, the ground-truth images in $\mathcal{D}_s$ have genuine image structures, whereas the ones in $\mathcal{D}_d$ may have artifacts caused by reversing ISP. Therefore, different CNN modules should be deployed to learn different types of ground-truths. We assign two instances of PostP module M_o, denoted by M_o^d and M_o^s, to transform the fused features F_t^d and F_t^s to the final denoised images O_d and O_s, respectively. A pair of sub-losses, denoted by $\mathcal{L}_o^d$ and $\mathcal{L}_o^s$, are deployed to compare G_d and G_s with the denoised images O_d and O_s, respectively:

$$\begin{cases} \min \mathcal{L}_o^d\big(G_d, O_d\big)\big) \\ \min \mathcal{L}_o^s\big(G_s, O_s\big)\big) \end{cases} \tag{4.6}$$

Finally, in the training process, we have two sets of loss functions $\mathcal{L}^d$ and $\mathcal{L}^s$ to update the BDNet on $\mathcal{D}_d$ and $\mathcal{D}_s$, respectively, which are as follows:

$$\begin{cases} \mathcal{L}^d = w_p(k) \cdot \mathcal{L}_p^d + w_t(k) \cdot \mathcal{L}_t^d + w_o(k) \cdot \mathcal{L}_o^d \\ \mathcal{L}^s = w_p(k) \cdot \mathcal{L}_p^s + w_t(k) \cdot \mathcal{L}_t^s + w_o(k) \cdot \mathcal{L}_o^s \end{cases} \tag{4.7}$$

where $w_p(k)$, $w_t(k)$ and $w_o(k)$ are the weights assigned on the sub-losses, which are variables dependent on the global epochs k in the training. We adopt an adaptive weighting scheme to train the modules progressively by setting:

$$\begin{cases} w_p(k) = 0.1^{\frac{k}{K}}, & 1 \leqslant k \leqslant K, \quad else \quad 0.1 \\ w_t(k) = 0.1 \cdot 10^{\frac{k}{K}}, & K \leqslant k \leqslant 2K, \quad else \quad 0.1 \\ w_o(k) = 0.1 \cdot 10^{\frac{k}{K}}, & 2K \leqslant k \leqslant 3K, \quad else \quad 0.1 \end{cases} \tag{4.8}$$

Under this weighting scheme, the three pairs of sub-losses in Eq. (4.7) dominate the training process in turn. In the first K epochs, $w_p(k)$ gradually decreases from 1 to 0.1, while the others remain at 0.1. This setting emphasizes the sub-losses $\mathcal{L}_p^d$ and $\mathcal{L}_p^s$ that optimize the PreP module. Then, during the epochs from K to $2K$, the weight $w_t(k)$ gradually ascends from 0.1 to 1, with the others remain at 0.1. At this stage, the sub-loss $\mathcal{L}_t^d$ and $\mathcal{L}_t^s$ dominate the training, focusing on the TemP module. Lastly, during the epoch from $2K$ to $3K$, the weight $w_o(k)$ on sub-losses $\mathcal{L}_o^d$ and $\mathcal{L}_o^s$ ascends from 0.1 to 1, with the other weights remaining at 0.1. This stage focuses on the training of the PostP module.

We adopt ℓ_1 loss for all the sub-losses involved in Eq. (4.7). An alternative training scheme is adopted to assign J_1 iterations for loss $\mathcal{L}^d$ and J_2 iterations for loss $\mathcal{L}^s$ in one cycle. In the testing stage, the modules M_p^d and M_o^d are removed, and only the M_p^s, M_t and M_o^s modules are used to form the final BDNet model.

4.3 Experiments

In this section, we conduct experiments to verify the effectiveness of proposed decoupled learning approach for burst denoising. We evaluate our BDNet on both synthetic noisy video dataset and real-world noisy sequences quantitatively and qualitatively. The peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [141] are used as the quantitative metrics. Metrics like PI and LPIPS are not employed in our experiments since we emphasize on the fidelity of the denoised image rather than the perceptual aspect.

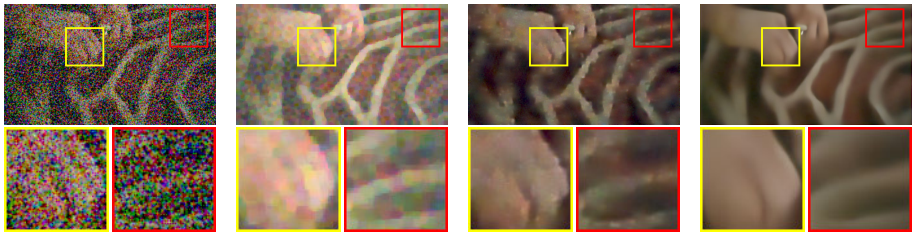
The kernel size of the convolutional layers of our BDNet is set to 3×3. Leaky ReLU is used as the activation function. The number of input frames N of a burst sequence is set to 5 for all multi-frame methods in the comparison. In all experiments, we use the Adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.99$) [67] to train BDNet and other

competing CNN models. The initial learning rate is set to 10^{-4}, and it exponentially decays by 0.1 at 3/4 of the total epochs. The parameter K in Eq. (4.8) is set to 30. In the decoupled training, we update the model for $J_1 = 3$ iterations on $\mathcal{D}_d$ and $J_2 = 1$ on $\mathcal{D}_s$ in one cycle. In all training, the batch size is set to 2 and the patch size is set to 128×128. Random rotations, vertical and horizontal flippings are applied for data augmentation.

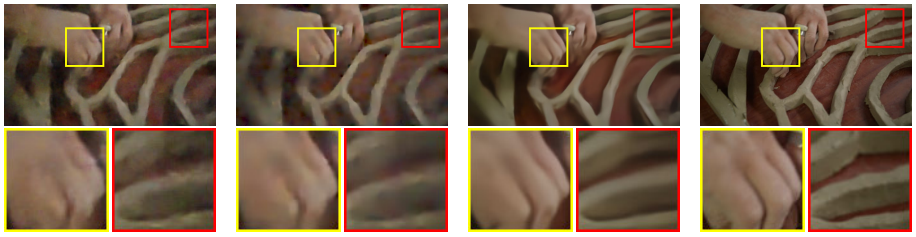
4.3.1 Datasets

Training set. For dynamic video dataset $\mathcal{D}_d$, we extract 20,000 image sequences from the Vimeo-90K video dataset [121], each containing 5 consecutive frames. As for $\mathcal{D}_s$, we leverage the SIDD [1] and SID datasets [24] to build it. Specifically, we combine the Sony training set of SID (162 image sequences) and 140 image sequences selected from SIDD training set as our static burst dataset $\mathcal{D}_s$.

Testing set. Our testing set consists of a synthetic test set and a real-world test set. For the synthetic test set, we extract another 200 image sequences (different from the training sequences in scene and content) from the Vimeo-90k dataset [121], denoted by Vimeo-200. For the real-world test set, we build a static test set, denoted by Real-static, for quantitative evaluation, as well as a dynamic test set, denoted by Real-dynamic, for qualitative perceptual evaluation because the ground-truths are hard to generate for dynamic scenes. The Real-static set is composed of the Sony test set (50 image sequences) in SID dataset [24] and 20 image sequences selected from the SIDD dataset [1]. For the Real-dynamic test set, we use iPhone 7 to capture 20 dynamic noisy image sequences in low-light environment. All the images are stored in raw format.



(a) Ref noisy frame (b) VBM4D (c) DNCNN (d) RIDNet



(e) KPN (f) TOFlow (g) BDNet (h) ground-truth

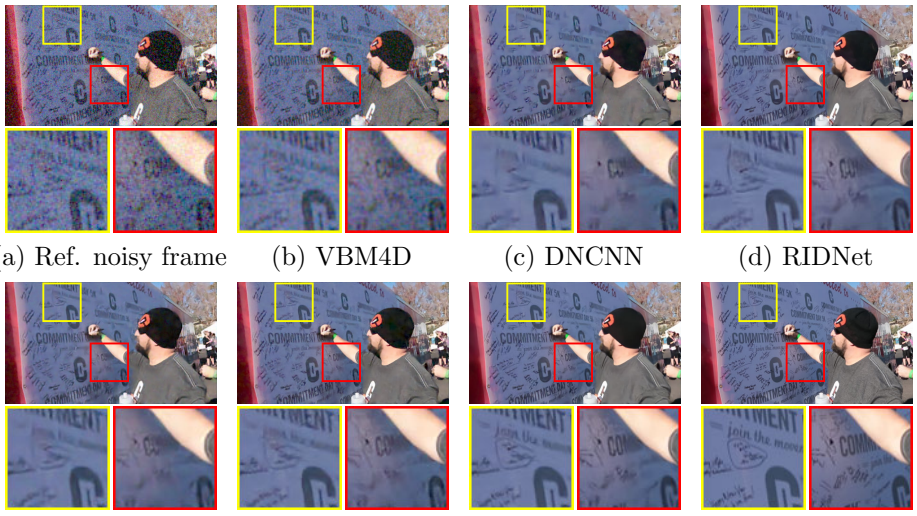
Figure 4.3: The denoising results of the compared methods on Vimeo-200 test set with Gaussian noise σ=50.

Table 4.1: Quantitative results (PSNR/SSIM) on the synthetic test sets. G25, G50 and PG indicates Gaussian σ=25, Gaussian σ=50 and Poisson-Gaussian noise, respectively.

	VBM4D	DNCNN	RIDNet	KPN	TOFlow	BDNet
G25	28.30/0.735	32.60/0.870	34.74/0.908	34.84/0.907	34.99/0.902	36.78/0.937
G50	25.92/0.621	29.32/0.776	31.47/0.821	32.44/0.862	31.95/0.829	34.03/0.900
PG	30.48/0.845	35.79/0.934	38.34/0.954	37.77/0.940	37.90/0.951	39.45/0.965

4.3.2 Results on Synthetic Noisy Sequences

We firstly evaluate the burst denoising performance of our BDNet on synthetic noisy data. We compare BDNet with several representative and state-of-the-art methods which are popularly used for synthetic noisy video denoising, including VBM4D [82], DnCNN [131], RIDNet [4], KPN [87] and TOFlow [121]. Among them, VBM4D is a classical patch based video denoising method; DnCNN and RIDNet are single-frame denoising CNN models; and KPN and TOFlow are CNN based multi-frame denoising models. We train all the CNN based models, including BDNet, until convergence on



(a) Ref. noisy frame (b) VBM4D (c) DNCNN (d) RIDNet

(e) KPN (f) TOFlow (g) BDNet (h) ground-truth

Figure 4.4: The denoising results of the compared methods on Vimeo-200 test set with Poisson-Gaussian noise.

the dataset $\mathcal{D}_d$. We add three types of noises, including Gaussian noise with σ=25 (G25), Gaussian noise with σ=50 (G50) and Poisson-Gaussian noise (PG) defined in Eq. (4.2), to the Vimeo-200 test set, and apply the competing models to these synthetic noisy sequences.

Table 4.1 shows the PSNR/SSIM results of the compared methods. We can see that the proposed BDNet achieves the highest PSNR and SSIM scores in all cases. While TOFlow performs well in the cases of low noise levels, i.e., G25 and PG, its performance heavily degrades in the case of higher noise level, i.e., G50. This is because it performs frame alignment in the image domain, but the alignment accuracy is affected by the heavy image noise. While the single-frame models, DnCNN and RIDNet, have relatively lower PSNR/SSIM scores, RIDNet performs well on PG noise, which may be attributed to its robust feature attention modules. Figs. 4.3 and 4.4 present the visual comparison of different denoising methods on Gaussian noise of $\sigma = 50$ and Poisson-Gaussian noise, respectively. We can clearly see that BDNet can reconstruct some subtle image structures and deliver very good visual quality, whereas other methods over-smooth much the details.

(a) Noisy frame (b) UTR (c) M-UNet (d) M-RIDNet

(e) KPN (f) INN (g) BDNet (h) ground-truth

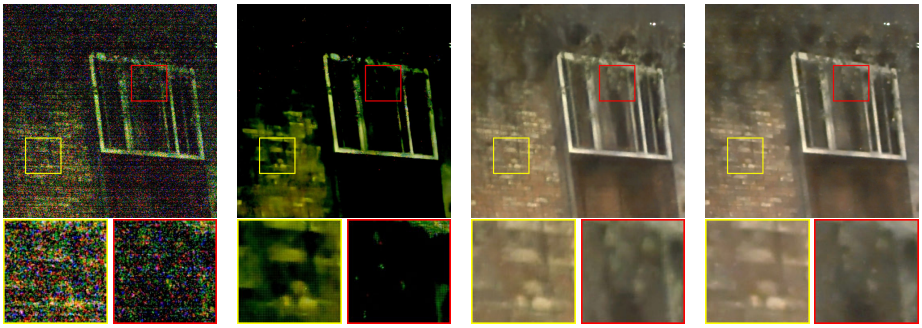
Figure 4.5: Denoising Results of the compared methods on Real-static test set. White balance gain and a gamma conversion with parameter 2.2 are applied for better visualization.

Table 4.2: Quantitative evaluation on the Real-static test set.

	VBM4D	UTR	UNet	M-UNet	RIDNet	M-RIDNet	KPN	INN	BDNet
PSNR	40.49	42.02	43.85	44.23	44.17	44.54	39.68	43.95	45.31
SSIM	0.901	0.897	0.954	0.964	0.960	0.968	0.867	0.964	0.971

4.3.3 Results on Real-world Noisy Sequences

We use the "Real-static" (for quantitative evaluation) and "Real-dynamic" (for qualitative evaluation) test sets to evaluate the performance of BDNet on real-world burst noisy sequences. We compare BDNet with those methods popularly used for real-world image denoising in literature, including VBM4D [82], UNet [24], RIDNet [4], Unprocess-to-raw (UTR) [15], KPN [87] and INN [68]. Both UTR and KPN methods learn real-world denoising by synthesizing data that resembles raw noisy images. In particular, UTR reverses the ISP pipeline, while KPN adds motion and noise to clean images to synthesize a burst of noisy images. In addition, the INN



(a) Ref. noisy frame (b) UTR (c) M-UNet (d) M-RIDNet

(e) KPN (f) INN (g) BDNet (h) ground-truth

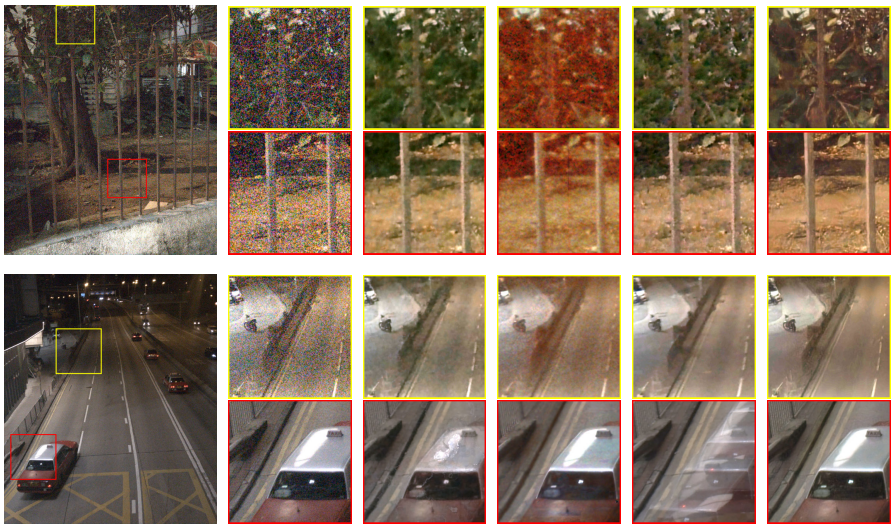
Figure 4.6: The denoising results of the compared methods on Real-static test set.

method uses global affine transformation to align frames and performs burst denoising by learning a trainable proximal operation. For fair comparison, we make the following configurations.

1) First, for the single-frame denoising methods UNet and RIDNet, we build a multi-frame version for them, denoted by M-UNet and M-RIDNet, respectively. M-UNet and M-RIDNet first denoise each frame in the noisy sequence, and then apply optical flow alignment [129] to fuse the denoised frames by average fusion, resulting in the finally denoised sequences.

2) Second, the UTR method learns a single-frame CNN. For fair comparison with UTR, we replace its single-frame CNN by our multi-frame BDNet structure and re-train it on $\mathcal{D}_d$.

3) Third, we re-train KPN on dataset $\mathcal{D}_d$ using the same data synthesis setting as



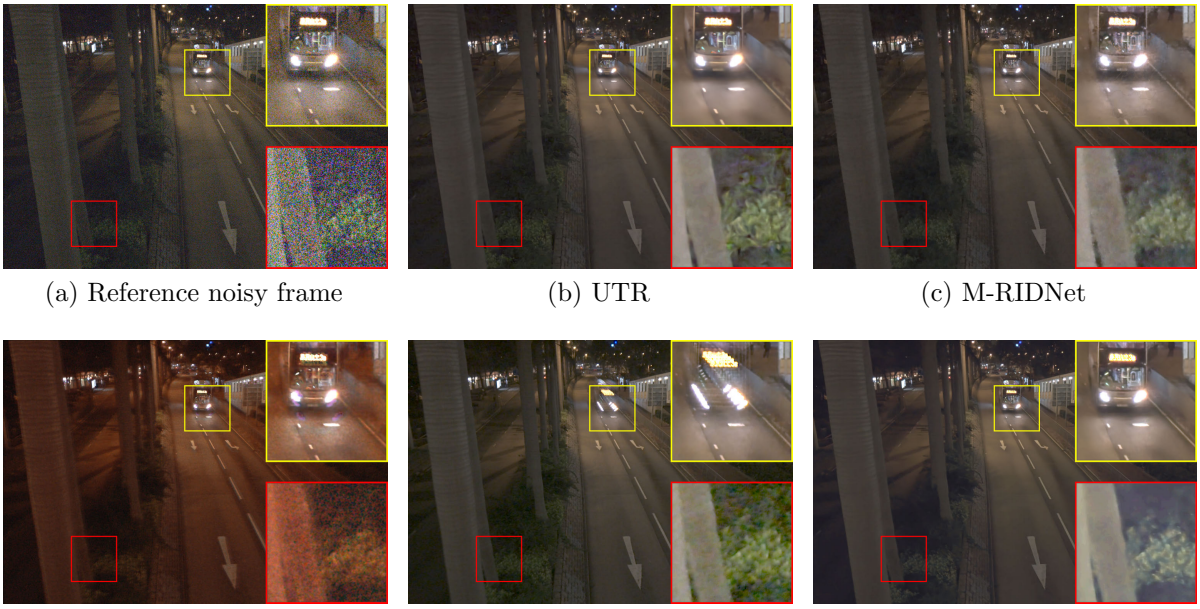
(a) (b) (c) (d) (e) (f)

Figure 4.7: Denoising results of the compared methods on Real-dynamic test set. (a) Noisy reference frame. (b) Noisy patches. (c) M-RIDNet. (d) KPN. (e) INN. (f) BDNet. White balance gain and a gamma conversion with parameter 2.2 are applied for better visualization. Best viewed on screen with zoom-in.

the original paper [87], including ISP pipeline reversing and noise generation.

4) At last, we train UNet, RIDNet and INN models on dataset $\mathcal{D}_s$ until convergence, and use the models with the best testing performance.

Table 4.2 shows the quantitative evaluation results on the "Real-static" test set. It is clear that the proposed BDNet obtains the highest PSNR and SSIM scores. UTR and KPN have low objective scores since they are not able to adapt to the real-world static test data. The two multi-frame models, M-UNet and M-RIDNet, obtain higher scores than their single-frame counterparts, which proves that the multi-frame fusion helps for realistic noise removal. However, their PSNR/SSIM results are still lower than the proposed BDNet. Figs. 4.5 and 4.6 compare visually the denoising results of the compared methods on images in the Real-static test set. One can see that the proposed BDNet is able to remove the noise without blurring the details, whereas the other methods tend to over-smooth the image details. In addition, the UTR



(a) Reference noisy frame (b) UTR (c) M-RIDNet

(d) KPN (e) INN (f) BDNet

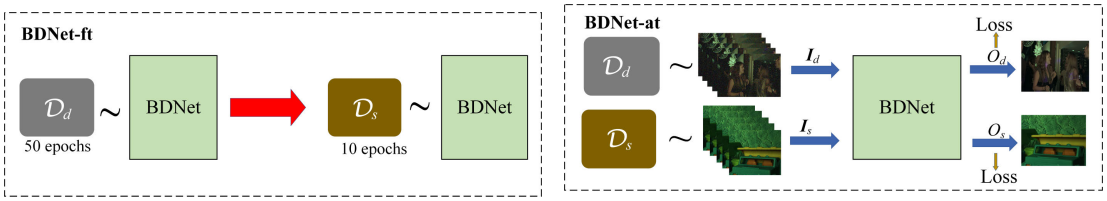
Figure 4.8: The denoising results of the compared methods on Real-dynamic test set.

method leaves residual noise in the image (Fig. 4.5(b)) because it is not adapted to the real-world dataset.

We then compare the competing models on the Real-dynamic test set. Since no ground-truths are available, we can only make qualitative comparisons on them. Figs. 4.7 and 4.8 show the results, where we can see that those competing methods have residual noise or artifacts caused by scene motion. In particular, the KPN method has severe color shift on image with large noise (the plant area in Fig. 4.7(d)). The M-RIDNet and INN methods encounter motion artifacts in the car area in Fig. 4.7(c)(e)). This is because optical flow and global affine alignment cannot effectively account for the local object motion. In contrast, the proposed BDNet is able to compensate for scene motion and restore the clean details.

4.3.4 Ablation Study

To better validate the effectiveness of our decoupled learning strategy, we make some ablation studies here by comparing it with two other intuitive training strategies



(a) Train on $\mathcal{D}_d$ and fine-tune on on $\mathcal{D}_s$ (b) Direct alternating training

Figure 4.9: Illustration of different learning schemes for real-world burst denoising with dynamic scenes. Please refer to the text for detailed descriptions.

Table 4.3: Quantitative results (PSNR/SSIM) of different learning schemes on the Real-static test set.

BDNet-ft	BDNet-at	Default setting
45.17/0.968	44.67/0.967	**45.31/0.971**

using $\mathcal{D}_d$ and $\mathcal{D}_s$, which are illustrated in Fig. 4.9. The first scheme, denoted by BDNet-ft, trains BDNet on dataset $\mathcal{D}_d$ and fine-tunes it on $\mathcal{D}_s$ till convergence. The second scheme, denoted by BDNet-at, directly alternates the training on $\mathcal{D}_d$ and $\mathcal{D}_s$ without deploying two instances of the PreP module M_p and the PostP module M_o.

Table 4.3 shows the quantitative results of the compared schemes on the Real-static test set. It can be seen that BDNet-at has much lower PSNR/SSIM scores than BDNet, which validates the importance of using two instances for M_p and M_o. BDNet-ft achieves similar PSNR/SSIM scores to BDNet. This is mainly because it utilizes $\mathcal{D}_s$ in the training while this quantitative test is also on static scenes. However, the perceptual quality of BDNet-ft and BDNet-na is much worse than BDNet for both Real-static and Real-dynamic scenarios. Fig. 4.10 shows the denoising results of three schemes on a static low-light sequence. One can see that BDNet-ft and BDNet-na generate visual artifacts in the street lamp area due to insufficient adaption to real-world noise. Fig. 4.11 shows the denoising results on dynamic scenes. It can be seen that BDNet-ft causes ghost artifacts in the car area with large motion, because

(a) (b) (c) (d) (e)

Figure 4.10: The results on a raw image sequence with large noise in Real-static test set by different learning schemes. (a) Noisy patch. (b) BDNet-ft. (c) BDN-at. (d) Default BDNet. (e) Ground-truth. White balance gain and a gamma conversion with parameter 2.2 are applied for better visualization.

(a) (b) (c) (d) (e)

Figure 4.11: The results on an raw image sequence with large noise by different learning schemes. (a) Noisy reference frame. (b) Noisy patch. (c) BDNet-ft. (d) BDN-at. (e) Defualt BDNet. White balance gain and a gamma conversion with parameter 2.2 are applied for better visualization. Best viewed on screen with zoom-in.

its fine-tuning on static dataset corrupts the learned alignment ability. In contrast, the decoupled learning scheme can achieve both merits of aligning dynamic sequences and revealing fine details in real-world scenes.

4.4 Summary

It is a challenging problem to learn a burst denoising network for real-world dynamic noisy sequences because of the lack of a pairwise training dataset. In this chapter, we proposed to leverage two types of existing datasets, a synthetic noisy video dataset and a static real-world burst dataset, to address this issue. We designed a modular CNN model, and proposed a decoupled learning approach, which learns to align

adjacent frames from the synthetic video dataset and learns to adapt to raw noise statistics from the static burst dataset. The trained CNN model, namely BDNet, can be well applied to real-world dynamic noisy sequences and it obtains compelling detail reconstruction quality with little motion blur. BDNet achieves leading performance, both quantitatively and qualitatively, on the task of burst image sequence denoising in real-world scenes.

Chapter 5

A Two-stage Framework for General and Effective Camera ISP Learning

Traditional image signal processing (ISP) pipeline consists of a set of cascaded image processing modules onboard a camera to reconstruct a high-quality sRGB image from the sensor raw data. Recently, some methods have been proposed to learn a convolutional neural network (CNN) to improve the performance of traditional ISP. However, in these works usually a CNN is directly trained to accomplish the ISP tasks without considering much the correlation among the different components in an ISP. As a result, the quality of reconstructed images is barely satisfactory in challenging scenarios such as low-light imaging. To address this problem, we firstly analyze the correlation among the different tasks in an ISP, and categorize them into two weakly correlated groups: restoration and enhancement. Then we design a two-stage network, called CameraNet, to progressively learn the two groups of ISP tasks. In each stage, a ground truth is specified to supervise the subnetwork learning, and the two subnetworks are jointly fine-tuned to produce the final output. Experiments on three benchmark datasets show that the proposed CameraNet achieves consistently compelling reconstruction quality and outperforms the recently proposed ISP learning methods.

5.1 Introduction

The raw image data captured by camera sensors are typically red, green and blue channel-mosaiced irradiance signals containing noise, less vivid colors and improper tones [96, 62]. To reconstruct a displayable high-quality sRGB image, an in-camera image signal processing (ISP) pipeline is generally required, which consists of a set of cascaded components, including color demosaicking, denoising, white balance, color space conversion, tone mapping and color enhancement, etc. The performance of an ISP plays the key role to improve the quality of sRGB images output from a camera.

The traditional ISP is usually designed as a set of handcrafted modules, each of which addresses a specific task [96]. For instance, a 3D lookup table is typically employed for the color enhancement task [62]. In most traditional ISP models, the modules are designed in a divide-and-conquer manner (i.e., splitting the ISP into a set of modules and developing them independently), while little attention has been paid to design them as a whole [56]. Moreover, it is time-consuming to tune each module for high image quality since the best output of one module may not result in the desired quality of the final output. Besides the standard ISP pipeline, there are also some ISP methods designed for burst imaging in the literature [53, 143]. However, these methods are subject to the effectiveness of image alignment techniques [8], which may generate ghost artifacts caused by object motion.

Recently, it has been shown that the performance of some image processing tasks, such as denoising [132, 131], white balance [57, 13], color demosaicking [108, 44], color enhancement [27, 45, 19], etc, can be significantly improved by deep learning techniques. In these methods, a convolutional neural network (CNN) is trained with a task-specific dataset that contains image pairs for supervised learning. Inspired by these methods, an intuitive idea is that we can train a subnetwork for each subtask of the ISP pipeline, and then chain them together as a whole ISP network. However, this

is still a divide-and-conquer strategy as used in the traditional ISP design, which is cumbersome and ineffective. First, it is difficult and expensive to construct a dataset which has a ground truth for each subtask in the ISP. If we use different task-specific datasets to train different subnetworks separately, errors will be accumulated as in traditional ISP. Second, training a subnetwork for each subtask will make the whole network very heavy and complex. Third, some subtasks in an ISP are not independent but correlated. It has been verified that for correlated tasks, it is more effective to treat them jointly and train a shared network for them [44, 112, 95].

Instead of learning a subnetwork for each subtask, some works have been reported to directly train a CNN model for all ISP subtasks as a whole [105, 100, 24]. Like the many CNN methods for image denoising and super-resolution [131, 65, 36], in these works a single-stage network is straightforwardly trained as an ISP in an end-to-end manner. However, an ISP is a composition of multiple image processing tasks, some of which may not be correlated too much with each other. Directly training them as a whole may make the network difficult to optimize, and lead to unsatisfactory learning performance.

In this chapter, we propose a new framework for deep-learning-based ISP pipeline design, which includes a two-stage CNN and the associated training scheme. We firstly analyze the relationships of individual subtasks of an ISP and group them into two weakly correlated clusters, namely, the restoration group and the enhancement group. Then a CNN model called CameraNet is proposed with two subnetworks to address the two groups of subtasks, respectively. Accordingly, a restoration and an enhancement ground truths are specified and used to train CameraNet in a progressive manner. With this arrangement, the two-stage CameraNet allows collaborative processing of correlated ISP subtasks while avoiding mixed treatment of weakly correlated subtasks, leading to high quality sRGB image reconstruction in various imaging scenarios. In our experiments, CameraNet outperforms the state-of-the-art ISP learning methods

and obtains consistently compelling results on three publically available benchmark datasets, including HDR+ [53], SID [24] and FiveK datasets [18].

5.2 Two-stage Camera ISP Framework

Suppose there are N essential subtasks in an ISP pipeline, including but not restricted to demosaicking, white balance, denoising, tone mapping and color enhancement. The traditional ISP pipeline employs N cascaded hand-crafted modules to address these subtasks. Let I_{cfa} be the raw CFA image and I_o be the output sRGB image. The traditional ISP can be represented as $I_o = f_N(f_{N-1}(\dots(f_1(I_{cfa})\dots))$, where $f_i, 1 \leqslant i \leqslant N$, denotes the ith algorithm component. The main drawback of such traditional ISP design is that each algorithm component is hand-crafted and it is difficult to optimize the pipeline as a whole, which limits the quality of output sRGB images.

In contrast to the traditional ISP design, we adopt the data-driven approach and model an ISP as a deep CNN system to address the N subtasks as a whole:

$$I_o = F_{isp}(I_{cfa}, \omega; \theta), \tag{5.1}$$

where $F_{isp}(.;\theta)$ refers to the CNN model with parameters θ to be optimized, and ω denotes the optional camera metadata (e.g., noise level, shutter speed) that can be used to help the network training and inference. We leverage a dataset S to train F_{isp} in a supervised manner. The dataset contains a set of input raw images I_{cfa}, and for each I_{cfa} there are K associated ground truth images $G_k, 1 \leqslant k \leqslant K$. In the case that $K = 1$, there is only one final ground truth output. In the case that $K > 1$, there are several intermediate ground truths $G_k, k < K$, leveraged to train the network, while G_K is the final ground truth for sRGB image reconstruction.

In the design of $F_{isp}(.;\theta)$, it is desirable that the CNN model can explicitly address

the different ISP subtasks while keeping the network as compact and simple as possible. To this end, we propose an effective two-stage CNN architecture and the associated learning scheme, which are described in the following sections.

5.2.1 Two-stage Grouping

As discussed in the previous subsection, F_{isp} is expected to address the ISP subtasks explicitly. One possible approach is to deploy a CNN subnetwork for each ISP subtask and chain them in sequence [77, 21]. As we discussed in the introduction section, however, such a divide-and-conquer strategy is cumbersome and ineffective, and it will make the whole network too heavy and complex. On the other hand, it has been demonstrated that some ISP subtasks, e.g., demosaicking and denoising, are correlated and they can be jointly addressed [44, 140]. Therefore, we propose to group the ISP subtasks into several weakly correlated clusters, while each cluster consists of several correlated subtasks. A CNN module is deployed for each cluster to allow joint learning of correlated subtasks, and then all the CNN modules are jointly fine-tuned to reduce the possible accumulated errors.

Based on the existing works in low-level vision, we group the ISP subtasks into two clusters: image restoration and enhancement. The goal of image restoration is to faithfully reconstruct the linear scene irradiance which contains genuine image structures and colors from raw image data. Typical restoration operations include color demosaicking, white balance, noise removal, deblurring, super-resolution, etc. They usually maintain the image distribution without largely changing the contrast and color style of an image. In contrast, the enhancement operations often nonlinearly change the image contrast and color distribution to make the image visually more appealing to human observers. Image enhancement operations are mainly located at the rear part of an ISP, such as tone mapping, color transform and contrast enhancement.

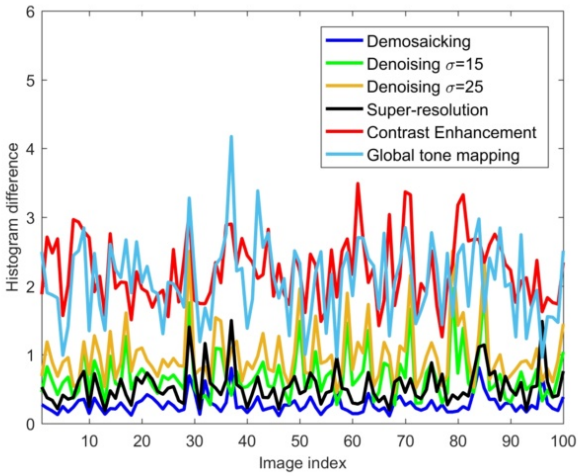


Figure 5.1: The image histogram changes caused by different image processing operations, including demosaicking, denoising (with $\sigma = 15$ and 25), $4\times$ super-resolution, local contrast enhancement [114] and global tone mapping [102], on images in the BSD100 dataset [85]. The vertical axis denotes the ℓ_1 norm of histogram differences, while the horizontal axis denotes the image index in BSD100.

Let's perform a test to evaluate the influences of several typical restoration and enhancement operators on image distribution. The restoration operators, including demosaicking, denoising ($\sigma = 15, 25$) and super-resolution, and the enhancement operators, including local enhancement [114] and global tone mapping [102][1], are employed in the test. White balance is excluded because it can be simply accounted for by per-channel global scaling. We denote the image before and after an operation $f(\cdot)$ as I and $f(I)$, respectively. Then, the ℓ_1 difference between the histogram vectors (with 256 bins) of I and $f(I)$ are computed to measure the amount of change on image intensity distribution. The BSD100 images are employed in the test [85]. For the restoration operations, we use the original images as $f(I)$ and degrade them to obtain I. Fig. 5.1 shows the ℓ_1 norms of histogram differences of the BSD100 images. We can see that the enhancement operators produce much higher changes on the image histogram than the restoration operators. This phenomenon validates that

[1]We firstly apply a reverse gamma conversion with parameter 2.4 to synthesize a linear raw image before applying the tone mapper.

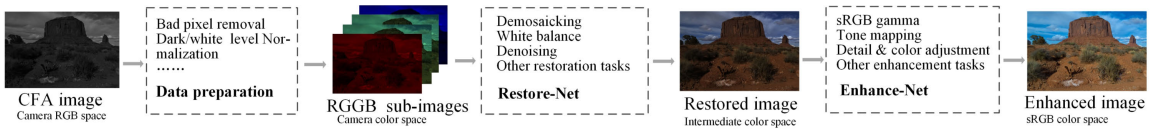


Figure 5.2: The proposed CameraNet system for ISP learning.

the enhancement and restoration operators have substantially different algorithm behaviors, which motivates us to employ a two-stage network design for ISP learning.

5.2.2 Two-stage Network Design

According to the discussions in the above subsection, we categorize the ISP subtasks into two groups (restoration and enhancement), and propose a two-stage CNN system, namely CameraNet, which is illustrated in Fig. 5.2. It is composed of a data preparation module, a restoration module called Restore-Net, and an enhancement module called Enhance-Net.

The role of the data preparation module is to separate some simple operations from the training since they can be well performed beforehand. The pre-processing operations applied on the CFA image I_{cfa} include bad pixel repairing, dark and white level normalization and pixel rearrangement. Bad pixel repairing interpolates the pixels where there are no response due to manufacturing imperfection. We use the python package Rawpy for this operation, which replaces the bad pixels by their neighboring pixels. Dark and white level normalization normalizes the dynamic range to [0,1]. Pixel rearrangement repacks the channel interlaced CFA image I_{cfa} to several single channel sub-images. Without loss of generality, we suppose that Bayer pattern is adopted. Then the CFA image I_{cfa} is rearranged as four sub-images (R, G, G, B) of the same size, and we denote by I_{rggb} the four sub-images for simplicity of expression.

Then Restore-Net, denoted by F_r, applies restoration-related operations, such as demosaicking, white balance and denoising, on the output of data preparation

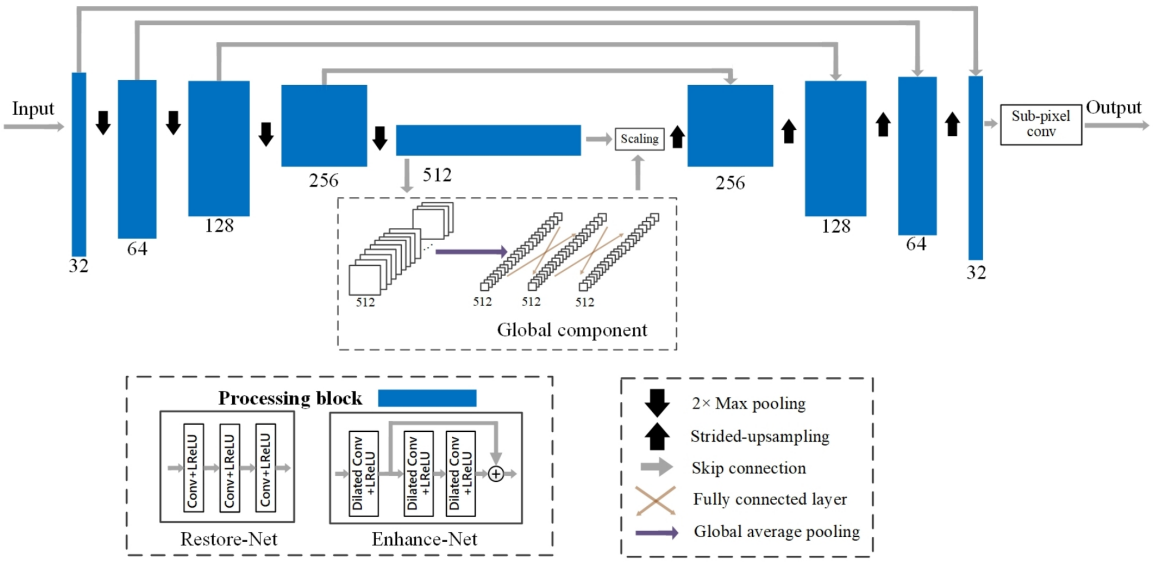


Figure 5.3: The structure of UNet-like Restore-Net and Enhance-Net modules in the proposed CameraNet system.

module, i.e., I_{rggb}. The output of Restore-Net is:

$$I_r = F_r(I_{rggb}, \omega_n; \theta_r) \tag{5.2}$$

where θ_r denotes the parameters of Restore-Net, and $\omega_n = \lambda_{shot} + \lambda_{read}^2$ is the input noise level to facilitate the denoising subtask. λ_{shot} and λ_{read} are the shot and readout noise parameters that can be obtained from the camera metadata. The restored image I_r is in an intermediate color space. The CIE XYZ space is considered here because it is designed to match human vision [92].

The Enhance-Net, denoted by F_e, takes the restored image I_r as input for processing. It first clips the intensity values below 0 and above 1, and applies an sRGB gamma function to the clipped image to account for the fixed nonlinear transformation from CIE XYZ space to sRGB space. Then the Enhance-Net learns to perform enhancement operations, such as tone mapping, detail enhancement and color manipulation, on I_r to produce the final output image I_o in sRGB color space:

$$I_o = F_e(I_r; \theta_e) \tag{5.3}$$

where θ_e denotes the parameters of Enhance-Net.

CNN architecture. There could be many possible designs for the Restore-Net and Enhance-Net modules. We consider a simple yet effective one, where two 5-level UNet like subnetworks are employed for Restore-Net and Enhance-Net, respectively. The architecture of the two subnetworks is shown in Fig. 5.3. A UNet has a contracting path to progressively reduce the resolution of feature maps, followed by an expanding path to progressively expand the resolution back [103]. Image structures are preserved by the skip connections from the contracting path to the expanding path at the same level. We adopt UNet for three reasons. First, the multi-scale processing nature of UNet can result in good image quality by learning adaptive operations for each scale. The finer scales focus on reproducing image local details and textures, while the coarser scales focus on enhancing image global colors and tones. Second, with UNet the main computations are deployed on the coarse image scales (lower resolution), resulting in relatively lower computational complexity. In addition, UNet can well solve multiple restoration subtasks by extracting common multiscale features to all subtasks and adopting a similar set of operations. Each subtask is flexibly accounted for in the network rather than rigidly treated.

Since the full color images I_r and I_o have twice the spatial resolution of the input sub-images I_{rggb} in each channel, a sub-pixel convolutional layer [106] is deployed at the end of Restore-Net and Enhance-Net to expand the resolution. In addition, to account for the global transformations in both modules (white balance in Restore-Net and global enhancement in Enhance-Net), we deploy an extra global transform block in the UNet modules, as shown in Fig. 5.3. This block first applies global averaging pooling to the input feature maps on the 5th level (lowest resolution), followed by 2

fully connected layers to obtain the globally scaled features as a 1-D vector. Finally, the global features are multiplied to the output feature maps on the 5th level in a per-channel manner. This process can be described as:

$$H_{5,out} = U_5(H_{5,in}) \otimes L_{fc}(L_{fc}(L_p(H_{5,in}))), \tag{5.4}$$

where $H_{5,out}$ and $H_{5,in}$ denote the output and input feature maps on the 5th level, respectively. $U_5(.)$, $L_{fc}(.)$ and $L_p(.)$ denote the 5th level operation block of UNet, the fully connected layer and the global pooling layer, respectively. Symbol "$\otimes$" denotes per-channel multiplication.

To further promote the learning performance of Enhance-Net, we deploy two extra settings that are found helpful for enhancement tasks. First, the convolution dilation rates of Enhance-Net are set to 1,2,2,4,8 from the 1st level to the 5th level to enlarge the receptive field. By this setting, the network can refer to a larger context to enhance an image, which avoids halo artifacts around the edges. Second, Enhance-Net deploys a residual connection within a convolutional block, as shown in the specification of Fig. 5.3. The residual connection predicts and adds features upon the previous feature maps in the network, which is helpful for detail boosting.

5.2.3 Ground Truth Generation

Most existing datasets [24, 53, 18] contain only the final ground truth G_o of the network output. For example, the HDR+ [53] and FiveK [18] datasets provide the sRGB ground truths that are created by HDR+ algorithm and human retouching, respectively. However, for our proposed two-stage CameraNet system, it is expected that we could have a restoration ground truth G_r and an enhancement ground truth G_o, which are corresponding to the intermediately restored image I_r and the finally enhanced image I_o, for network training.

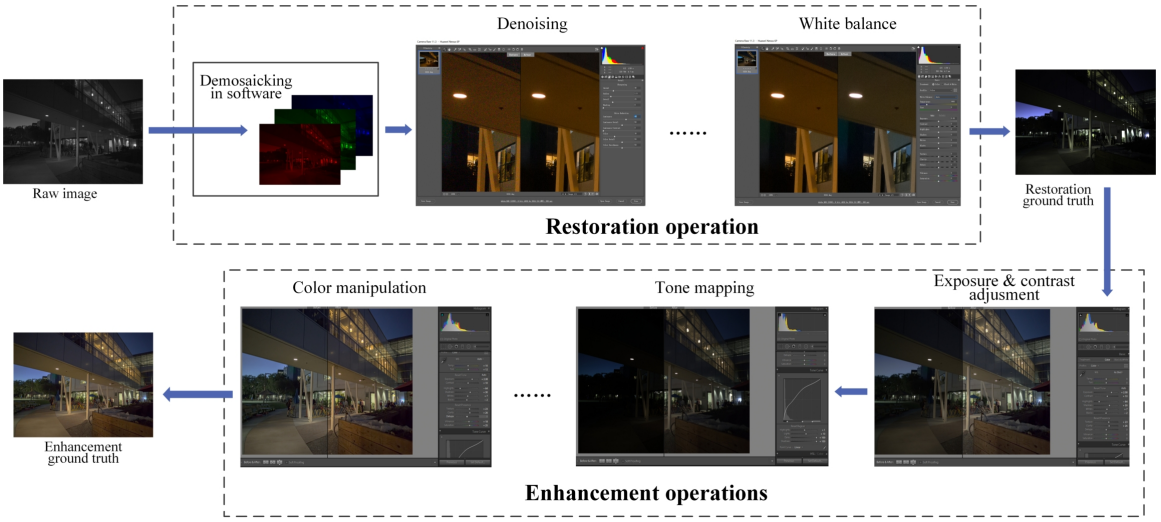


Figure 5.4: The workflow of creating restoration and enhancement ground truths using Adobe software. The restoration ground truth is created in Adobe Camera Raw, while the enhancement ground truth is created in Lightroom. The dots in restoration operations refer to other possible restoration tasks such as aberration correction and deblurring, while the dots in enhancement operations refer to other possible adjustment of image features.

The ground truths G_r and G_o can be generated by using photo editing software, e.g., Adobe software. An example procedure is shown in Fig. 5.4. In the first step, the restoration ground truth G_r is created by performing restoration-related operations on the raw image I_{cfa}, including demosaicking, denoising and white balance. In our experiments on the FiveK dataset, the G_r is generated in this way. On some datasets (e.g., HDR+ dataset) where a raw image sequence is available for each scene, one can adopt additional operations to boost the quality of the restoration ground truth. For example, the sequence of raw images can be fused into one raw image to suppress noise, followed by other restoration operations. We use this method to generate the restoration ground truths on the HDR+ dataset. In the second step, the enhancement ground truth G_o is created by applying enhancement-related operations on the restoration ground truth G_r, including contrast adjustment, tone mapping, color manipulation and color conversion. This can be easily done by using photo

(a) (b) (c) (d) (e)

Figure 5.5: Illustration of the two-stage network outputs and ground truths. The columns from left to right are (a) Raw images. (b) Results by Restore-Net. (c) Restoration ground truth. (d) Results by Enhance-Net. (e) Enhancement ground truth. The image in the first row is from the HDR+ dataset [53], while the image in the second row is from the FiveK dataset [18]. A gamma transform with parameter 2.2 is applied to the raw images and restoration ground truths for display.

retouching software, e.g., Adobe Lightroom.

Since the goal of restoration tasks is to objectively reconstruct genuine image structures and colors, the styles of the generated ground truths G_r from raw images are generally similar. In contrast, the enhancement tasks are subjective to human observers, which may result in various styles of enhancement ground truths G_o. Fig. 5.5 shows the image triplets from the HDR+ dataset and the FiveK dataset, including the raw image (demosaicked for better visualization), the restoration and the enhancement ground truths. We also show the reconstructed images in the two stages of our CameraNet for reference. One can see that the two restoration ground truths exhibit similar visual attributes, whereas the two enhancement ground truths are of very different styles. The enhancement ground truth in HDR+ dataset emphasizes on detail enhancements while that in FiveK dataset focuses on color style manipulation.

5.2.4 Two-step Training Scheme

Based on the two-stage structure of CameraNet, we propose a two-step training scheme of it. In the first step, the Restore-Net and Enhance-Net are independently trained in parallel, while in the second step, the two subnetworks are jointly fine-tuned. We adopt a set of ℓ_1 losses in the training of CameraNet because the ℓ_1 loss is simple to calculate and tends to converge to a visually good local minimum [138].

In the first step, the Restore-Net is trained with a restoration loss calculated between the restored image I_r and the ground truth G_r in linear and logarithmic space:

$$\begin{aligned}\mathcal{L}_r(I_r, G_r) = \|I_r - G_r\|_1 + \\ \|log(max(I_r, \epsilon)) - log(max(G_r, \epsilon))\|_1,\end{aligned} \tag{5.5}$$

where ϵ is a small value to avoid infinity. The use of the log sub-loss is based on the fact that the restored image I_r is in a linear space where the image intensity is proportional to scene radiance but not human visual response. Thus, to penalize the error in terms of human perception, we introduce this nonlinear term in the loss computation.

Meanwhile, the Enhance-Net is trained in parallel to Restore-Net. The restoration ground truth G_r is input to the Enhance-Net, and the output is denoted as $I_{o,r} = F_e(G_r; \theta_e)$. The enhancement loss is calculated as the ℓ_1 difference between $I_{o,r}$ and the ground truth G_o:

$$\mathcal{L}_o(I_{o,r}, G_o) = \|I_{o,r} - G_o\|_1, \tag{5.6}$$

It can be seen that the training of Enhance-Net does not rely on the output of Restore-Net. In addition, there is not a nonlinear term in the loss because the enhanced image is already in a nonlinear color space, i.e., sRGB space.

Once the parallel training of Restore-Net and Enhance-Net is finished in the first step, in the second step the two subnetworks are jointly fine-tuned with the following joint loss:

$$\mathcal{L}_{joint} = \lambda \cdot \mathcal{L}_r(I_r, G_r) + (1 - \lambda) \cdot \mathcal{L}_o(I_o, G_o) \tag{5.7}$$

Note that in this step, the enhancement sub-loss takes I_o rather than $I_{o,r}$ in Eq. (5.6) as input for loss calculation. The joint fine-tuning has two roles. First, the Enhance-Net receives the gradients from the enhancement sub-loss, while the Restore-Net receives the gradients from both restoration and enhancement sub-losses, weighted by λ and $1 - \lambda$, respectively. Thus, this step allows the Restore-Net to contribute to the final sRGB image reconstruction. Second, since the two subnetworks are trained independently in the first step, cumulative errors may occur due to the gap in the intermediate results. Joint fine-tuning can reduce such cumulative errors by facilitating the interaction between the two modules. The setting of parameter λ is scenario-specific. If the restoration subtasks dominate the ISP pipeline, e.g., in the low-light imaging scenario, λ should be set larger to emphasize the restoration functionality of Restore-Net, and vice versa.

While the adopted ℓ_1-based loss functions yield good results, our training scheme is open to other advanced loss design, e.g., adversarial loss [70] and perceptual loss [60]. Actually, we find that employing the perceptual loss in the fine-tuning step can slightly improve the visual appearance of the reconstructed images, which will be discussed in the experiment section.

5.3 Experiments

In this section we perform extensive experiments to verify the learning capability and image reconstruction performance of our CameraNet system both quantitatively and

qualitatively. Three objective indices, including PSNR, SSIM [141] and S-CIELAB [136], are employed in the quantitative evaluation. PSNR calculates the ratio of the peak signal power to the power of reconstruction errors, while SSIM measures the structural similarity between reconstructed and ground truth images. S-CIELAB measures the perceptual errors of two colors in the Lab space (the smaller the measure, the better the color fidelity). We use the code from [59] for calculating S-CIELAB. Without loss of generality, Bayer CFA pattern is used in all our experiments. However, it is not difficult to adapt CameraNet to a new CFA design. One can simply retrain the Restore-Net with the input data of the new CFA pattern, e.g., RGBW or RGBG.

5.3.1 Dataset Setting

Three publically available datasets that can be used for ISP learning are employed in our experiments, including the HDR+ dataset [53], the SID dataset [24] and the FiveK dataset [18]. These datasets have different features and they can be used to validate the performance of an ISP learning method from different aspects.

The HDR+ dataset [53] focuses on burst denoising and detail enhancement. For each scene, a burst of underexposed raw images are captured. Those images are firstly aligned and fused into one raw image to suppress noise, and then the HDR+ algorithm is applied to the fused raw image to produce the sRGB image. For each scene, the fused raw image and the corresponding sRGB image are provided in the dataset. We use DCraw to perform demosaicking, white balance and color conversion on the fused raw image to obtain the restoration ground truth, and treat the provided sRGB images as the enhancement ground truth. The Nexus 6P subset, which includes 665 scenes as training data and 240 scenes as testing data[2], is used in the experiment. We take a single raw image (the reference frame in alignment) as the input of CameraNet.

[2]The other subsets are not used because there are some misalignments between the input images and the ground truths.

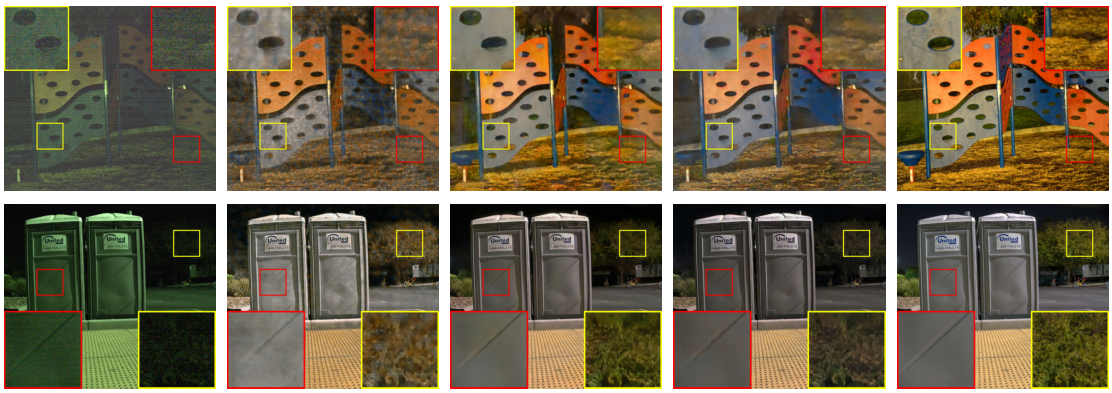
The data in the testing set are sampled according to the distribution of ISO values, which are rough indicators of noise level.

The SID dataset [24] focuses on denoising in low-light environment. For each scene, it provides a noisy raw image with short exposure and a relatively clean raw image with long exposure. To obtain the restoration ground truths, we use the DCRaw to perform restoration operations on the long-exposed raw images. Since the SID dataset does not involve any enhancement operation, we further process the restoration ground truth by the auto-enhancement tool in Photoshop to obtain the enhancement ground truth. We use the Sony A7S2 subset for experiments, which includes 181 and 50 scenes for training and testing, respectively. The data in the testing set are sampled according to the distribution of ISO values.

The FiveK dataset [18] is featured with strong manual retouching on image tone and color style. For each raw image of the 5,000 scenes, five photographers are employed to adjust various visual attributes of the image by using the Lightroom software and generate five images with different photographic styles. As in many previous works [45, 122], we take the set of images retouched by expert-C as the enhancement ground truths. Since the FiveK dataset does not contain restoration ground truth, we process the input raw image using DCRaw to obtain the restoration ground truth. The Nikon D700 subset with 500 training images and 150 uniformly sampled testing images is used in the experiments.

5.3.2 Experimental Setting

We use the Adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.99$) to train CameraNet and all the competing CNN models. In the first training step, Restore-Net is trained for 2000, 4000 and 1000 epochs on the FiveK, HDR+, and SID datasets, respectively, depending on the task complexity on the three datasets. The Enhance-Net is trained with 500 epochs on all the three datasets since the enhancement tasks on these datasets have



(a) Raw image (b) One-stage (c) Two-stage (d) Three-stage (e) Ground truth

Figure 5.6: Results by one-stage, two-stage and three-stage CNN models. The two sets of images are from the SID dataset [24]. A gamma transform with parameter 2.2 is applied to the raw images and restoration ground truths for display.

comparable complexity. In the second fine-tuning step, 200 epochs are used for all the datasets.

The initial learning rate for the first training step is set to 10^{-4}, and exponentially decays by 0.1 at 3/4 epochs. The learning rate for the fine-tuning step is fixed to 10^{-5}. Considering the importance of the restoration subtask on each dataset, the parameter λ in Eq. (5.7) is set to 0.1, 0.5 and 0.9 on the FiveK, HDR+, and SID datasets, respectively. In both training steps, the batch size is set to 1 and the patch size is set to 1024×1024. Random rotations, vertical and horizontal flippings are applied for data augmentation.

5.3.3 Ablation Study

We use the HDR+ and SID datasets for ablation study on the proposed two-stage network design, the training scheme, and the network architecture. All the evaluated models in this subsection are trained until convergence with the best testing performance.

Table 5.1: Ablation study on the HDR+ and SID datasets. The best and second best scores are highlighted in red and blue for each column.

	HDR+ dataset			SID dataset		
	PSNR	SSIM	S-CIELAB	PSNR	SSIM	S-CIELAB
Default two-stage setting	25.01	0.854	5.32	22.44	0.742	7.58
One-stage setting	21.53	0.816	6.48	19.04	0.692	9.48
Three-stage setting	23.57	0.834	5.99	21.56	0.735	8.40
Training without step 1	22.02	0.824	5.48	21.89	0.719	7.76
Training without step 2	23.98	0.843	5.34	22.21	0.738	7.63
Fine-tuning with perceptual loss	24.05	0.839	5.64	21.06	0.713	8.13
One-stage SRGAN+CAN24	21.72	0.801	7.18	19.85	0.682	9.50
Two-stage SRGAN+CAN24	22.31	0.815	6.93	20.96	0.714	8.85

The effectiveness of two-stage network design. To verify the effectiveness of the proposed two-stage design of CameraNet, we compare it with a one-stage and a three-stage counterparts. In the one-stage setting, a UNet with the same number of parameters as the two-stage CameraNet (i.e., double the number of processing blocks at each resolution level) is employed, and it is trained with the final enhancement ground truth. In the three-stage setting, three UNets are employed to progressively learn the ISP pipeline in three stages, i.e., demosaicking, denoising/white balance and enhancement. The number of parameters are maintained the same by reducing 1/3 the number of processing blocks at each resolution level.

The PSNR/SSIM/S-CIELAB results of the three competing networks on the HDR+ and SID datasets are shown in Table 5.1. One can see that the default two-stage network works significantly better than the one-stage network, and much better than the three-stage network. Some visual comparison results are shown in Fig. 5.6. It can be seen that the one-stage network produces various visual artifacts, the three-stage network performs much better, while the two-stage network delivers the best visual quality. This experiment validates that it is difficult to use a single

(a) Without step 1 (b) Defualt setting (c) Ground truth

Figure 5.7: Comparison between the default training setting and the setting without step 1. The image is from the HDR+ dataset [53].

(a) Without step 2 (b) Defualt setting (c) Ground truth

Figure 5.8: Comparison between the default training setting and the setting without step 2. The image is from the HDR+ dataset [53].

network to handle all ISP tasks together, while it is less effective to process correlated subtasks (e.g., demosaicking and denoising) using different networks. By grouping the ISP subtasks into two groups of correlated subtasks and deploying one network for each group, our two-stage CameraNet demonstrates highly effective ISP learning performance.

The two-step training scheme. We then evaluate the effectiveness of the proposed two-step training scheme. Firstly, we compare it with two variants. The first variant skips the first training step and directly goes to the second joint training step, i.e., we directly train the whole CameraNet with the loss in Eq. (5.7). The

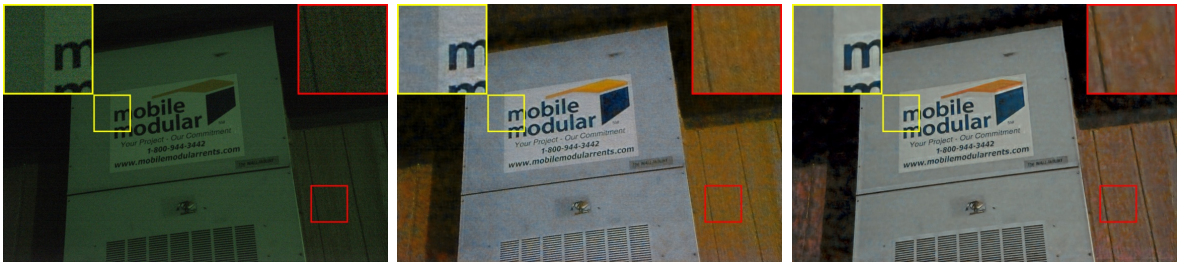
(a) Without p.l. (b) With p.l. (c) ground truth

Figure 5.9: Comparison between the results with and without perceptual loss (p.l.). We add the p.l. on the enhanced image in the fine-tuning step with weight 0.01. The image is from the HDR+ dataset [53].

second variant keeps the first step but removes the second joint fine-tuning step. The results are shown in Table 5.1. We can see that without the first step in training, the PSNR/SSIM/S-CIELAB scores become significantly worse. One visual example is presented in Fig. 5.7. We can see that some noises remain in the reconstructed image. This indicates the importance of progressive training of restoration and enhancement modules. On the other hand, from Table 5.1 we can see that without the joint fine-tuning step, the results are not that bad but still far behind our default two-step training scheme. One visual example is shown in Fig. 5.8. We can see that without the joint fine-tuning, the sky area has a sudden color change and has unnatural appearance.

Perceptual loss. The perceptual loss [60] has been widely used in many image restoration and enhancement networks to improve the image visual quality. It is interesting to evaluate whether the perceptual loss can bring additional benefit to our CameraNet. We apply the perceptual loss (weighted by 0.01) on the enhanced images in the fine-tuning step[3]. The quantitative results are presented in Table 5.1, and one visual example is shown in Fig. 5.9. We can see that the perceptual loss

[3]We use the "relu2_2" and "relu5_4" layers in the VGG-19 network to calculate the loss.



(a) Raw image (b) One-stage SRGAN+CAN24 (c) Two-stage SRGAN+CAN24

(d) CameraNet (e) Ground truth

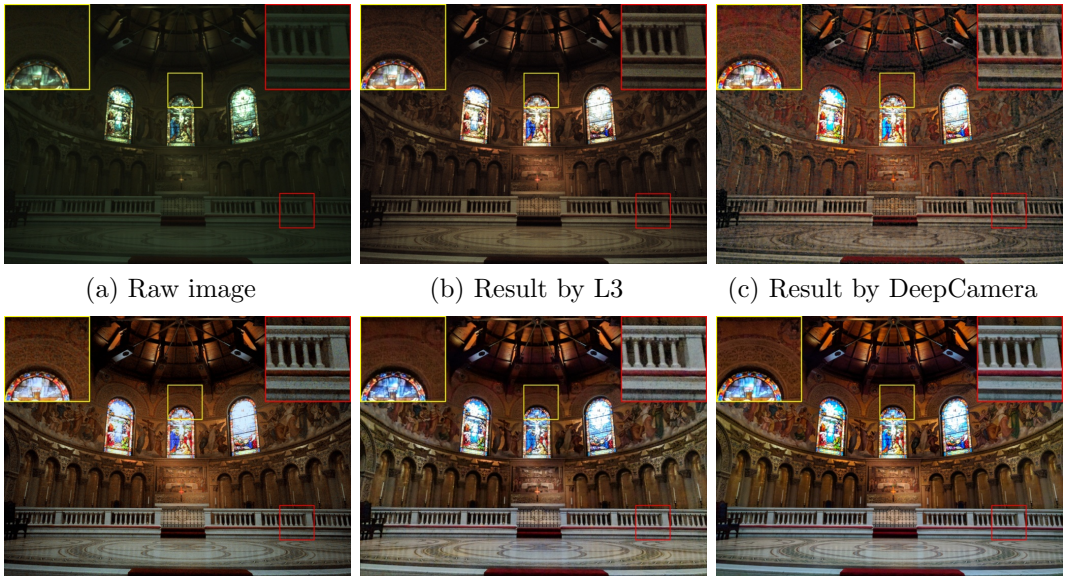
Figure 5.10: Comparison between SRGAN+CAN24 and CameraNet. A gamma transform with parameter 2.2 is applied to the raw images and restoration ground truths for display.

slightly improves the visual quality by reducing some subtle artifacts, while it leads to a moderate drop in the quantitative metrics since it penalizes the error in feature domain rather than the image domain.

Other CNN architectures. To verify whether the proposed two-stage framework can be generalized to other CNN architectures, we further compare the one-stage and two-stage settings by using a different CNN architecture. We use SRGAN [70] with 10 layers as the restoration subnetwork and CAN24 [26] as the enhancement subnetwork. The PSNR/SSIM/S-CIELAB scores are shown in Table 5.1, and one visual example is shown in Fig. 5.10. We can see that the two-stage setting of SRGAN+CAN24 outperforms its one-stage counterpart. Meanwhile, the two-stage SRGAN+CAN24 is not as effective as our CameraNet in noise removal. We think this is mainly because SRGAN+CAN24 lacks multiscale processing that facilitates the denoising task.

Table 5.2: Objective comparison of different learning-based ISP methods.

	HDR+ dataset			SID dataset			FiveK dataset		
	PSNR	SSIM	S-CIELAB	PSNR	SSIM	S-CIELAB	PSNR	SSIM	S-CIELAB
CameraNet	**25.01**	**0.854**	**5.32**	**22.44**	**0.742**	**7.58**	**23.57**	**0.849**	**6.74**
L3 algorithm [59]	19.23	0.682	9.84	16.47	0.462	12.64	20.00	0.797	10.70
DeepISP-Net [105]	22.88	0.818	6.78	18.26	0.649	10.18	22.59	0.845	7.38
DeepCamera [100]	20.65	0.738	8.81	18.19	0.587	10.97	20.67	0.776	8.86



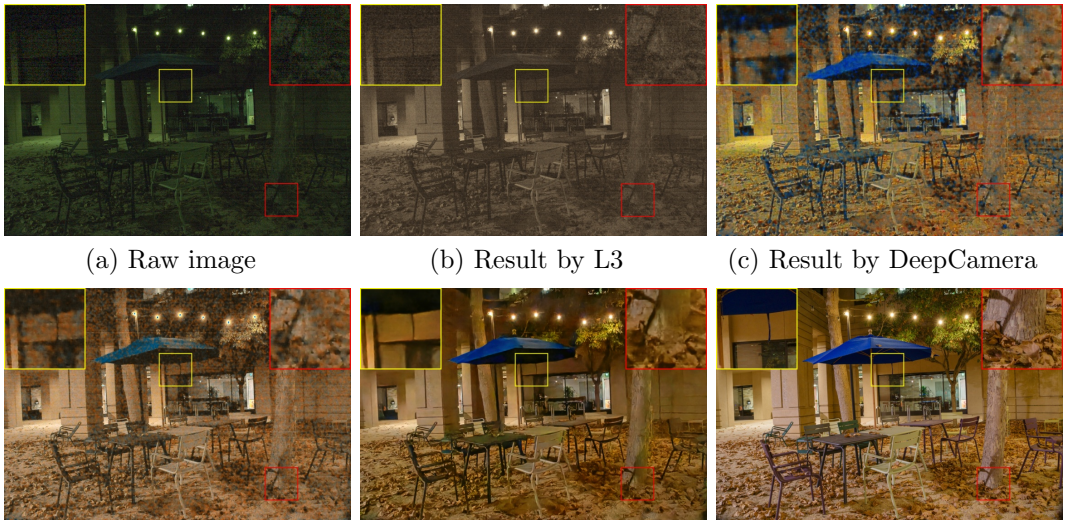
(a) Raw image (b) Result by L3 (c) Result by DeepCamera

(d) Result by DeepISP-Net (e) Result by CameraNet (f) Ground truth

Figure 5.11: Results on a church image from the HDR+ dataset [53] by the competing methods. A gamma transform with parameter 2.2 is applied to the raw image for better visualization.

5.3.4 Comparison with Recent Learning-based ISP

In this section, we compare our CameraNet with those recently developed learning-based ISP methods, including L3 algorithm [59], DeepISP-Net [105] and DeepCamera [100]. The L3 algorithm firstly groups the patches of the input raw images according to the intensity level and then learns a per-class filter to obtain the sRGB image. DeepISP-Net and DeepCamera are single-stage CNN models trained in an end-to-end



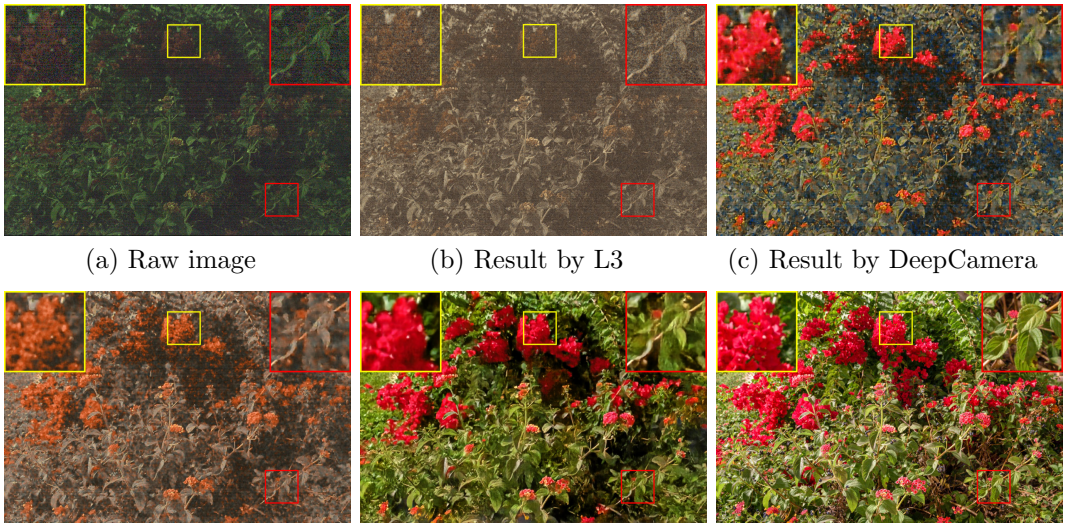
(a) Raw image (b) Result by L3 (c) Result by DeepCamera

(d) Result by DeepISP-Net (e) Result by CameraNet (f) Ground truth

Figure 5.12: Results on a pavilion image from the SID dataset [24] by the competing methods. A gamma transform with parameter 2.2 is applied to the raw image for better visualization.

manner. In particular, DeepISP-Net takes a pre-demosaicked image as input and process the image with a single scale. DeepCamera takes the mosaic CFA image as input and adopts a multi-scale architecture. We train all the compared methods on the HDR+, FiveK and SID datasets until convergence with their best testing results. The source codes of L3 is provided by the authors. Because the source codes of DeepISP-Net and DeepCamera are unavailable, we implement them based on the settings described in the original papers and train them using the original loss functions. The PSNR/SSIM/S-CIELAB results of the compared methods are shown in Table 5.2, while Figs. 5.11-5.14 present the visual results.

Results on the HDR+ dataset. The HDR+ dataset is featured with moderate denoising and strong detail enhancement. As can be seen from Table 5.2, the proposed CameraNet achieves significantly better objective scores than the other methods. This is because the two-stage nature of CameraNet can effectively account for the restoration and enhancement tasks involved in the HDR+ dataset. In contrast, the L3



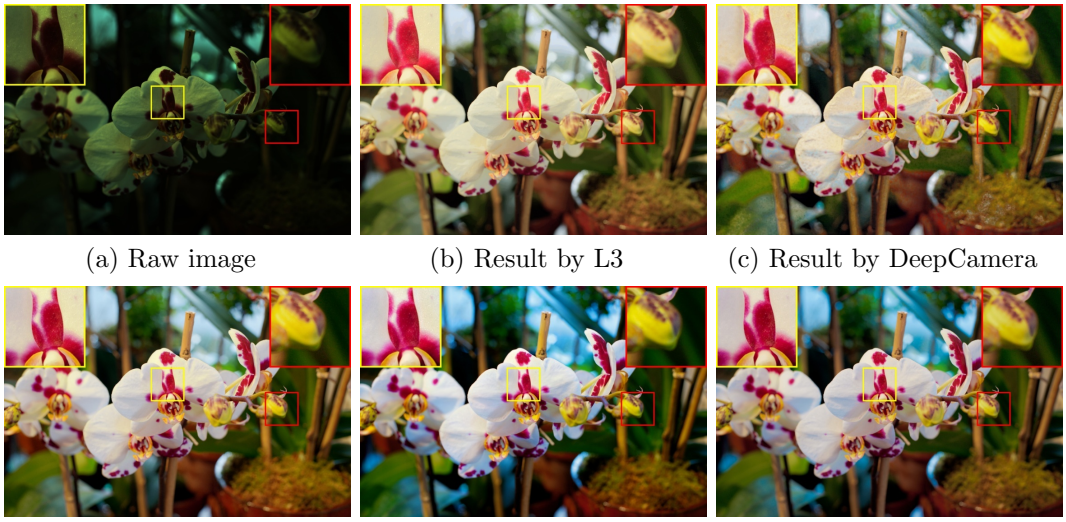
(a) Raw image (b) Result by L3 (c) Result by DeepCamera

(d) Result by DeepISP-Net (e) Result by CameraNet (f) Ground truth

Figure 5.13: Results on a pavilion image from the SID dataset [24] by the competing methods. A gamma transform with parameter 2.2 is applied to the raw image for better visualization.

method, DeepISP-Net and DeepCamera mix the restoration and enhancement tasks to train the filters or networks, making the learning process more difficult. Fig. 5.11 shows a visual example for comparison. The proposed CameraNet obtains visually pleasing results, while the L3 method, DeepISP-Net and DeepCamera produce visual artifacts. In particular, the L3 method barely performs denoising and produces false colors because the filter learning approach is too simple for the complex ISP tasks. DeepISP-Net and DeepCamera show better results, but they retain some noise-like artifacts. We suspect this is because DeepISP-Net and DeepCamera mix the denoising and color manipulation subtasks. As a result, the noise in the raw image is not effectively removed but amplified. In contrast, the proposed CameraNet produces visually appealing results with much less artifacts, which can be attributed to the two-stage treatment of different ISP subtasks.

Results on the SID dataset. On the SID dataset, from Table 5.2 we can see that CameraNet outperforms the other methods by a large margin. This is because



(a) Raw image (b) Result by L3 (c) Result by DeepCamera

(d) Result by DeepISP-Net (e) Result by CameraNet (f) Ground truth

Figure 5.14: Results on a flower image from the FiveK dataset [18] by the competing methods. A gamma transform with parameter 2.2 is applied to the raw image for better visualization.

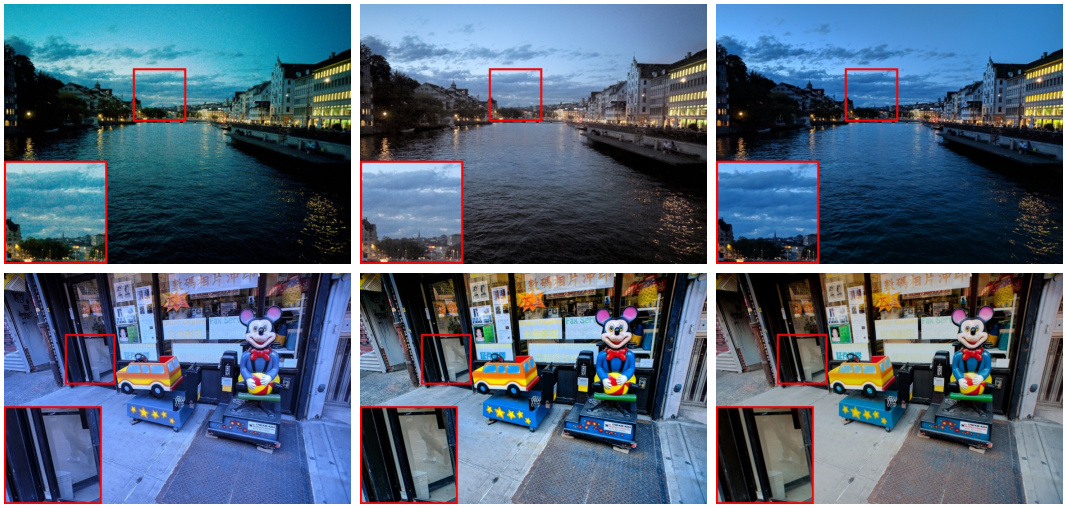
the noise level in the SID dataset is much higher than the HDR+ dataset, which requires the CNN model to have strong denoising capability. Our CameraNet meets this requirement by explicitly considering the denoising subtask in the restoration stage, whereas DeepISP-Net and DeepCamera mix all the ISP subtasks together in learning, leading to inferior performance. Figs. 5.12 and 5.13 show the results of the compared methods. We can see that the visual quality of the proposed CameraNet is significantly higher than DeepISP-Net and DeepCamera. Specifically, DeepISP-Net produces inaccurate colors, while DeepCamera remains serious noise in the reconstructed images. In comparison, CameraNet effectively reduces the noise and enhances the image structures. Moreover, the results by the L3 method largely deviate from the ground truth. This is because the filter-learning-based L3 model is not expressive enough to perform the ISP tasks in challenging conditions such as low-light imaging.

Table 5.3: Over-fitting evaluation. This table compares the training and testing losses of the last epoch for each training step on the three datasets. The "Gap" means the difference between the testing loss and the training loss.

	First step (Restore-Net)			First step (Enhance-Net)			Second step		
	Train	Test	Gap	Train	Test	Gap	Train	Test	Gap
HDR+ dataset	0.0043	0.0058	+0.0015	0.0348	0.0421	+0.0073	0.0370	0.0482	+0.0112
SID dataset	0.0078	0.0117	+0.0039	0.0387	0.0676	+0.0289	0.0400	0.0769	+0.0369
FiveK dataset	0.0034	0.0067	+0.0033	0.0326	0.0659	+0.0333	0.0345	0.0670	+0.0325

Results on the FiveK dataset. Compared with the HDR+ and SID datasets, the FiveK dataset is less challenging because it does not involve the denoising subtask. In fact, the major task on the FiveK dataset is the enhancement of colors and tones on images captured by high-end cameras with little noise. From Table 5.2, we can see that the advantage of CameraNet over DeepISP-Net is not as significant as that on the HDR+ and SID datasets because the dominant enhancement tasks can be well learned by DeepISP-Net. The results by DeepCamera and L3 model are much worse than CameraNet and DeepISP-Net. The inferior performance of DeepCamera may be caused by its use of mosaic CFA image as input to the network. In such case, the convolutional kernels at the early layers have extra burden to separate the color channels of CFA image, leading to less accurate results. Fig. 5.14 compares the results of different methods on a flower image. We can see that CameraNet and DeepISP-Net achieve satisfactory results, whereas the L3 method and DeepCamera generate some artifacts.

Over-fitting evaluation. Table 5.3 compares the training and testing losses of the last epoch of each training step on the three datasets. One can see that there is over-fitting on all datasets, especially on the SID and FiveK datasets since they have fewer training data than the HDR+ dataset. The over-fitting problem is mostly caused by the lack of training data on the three datasets. We believe it can be diluted if more data can be collected for training.

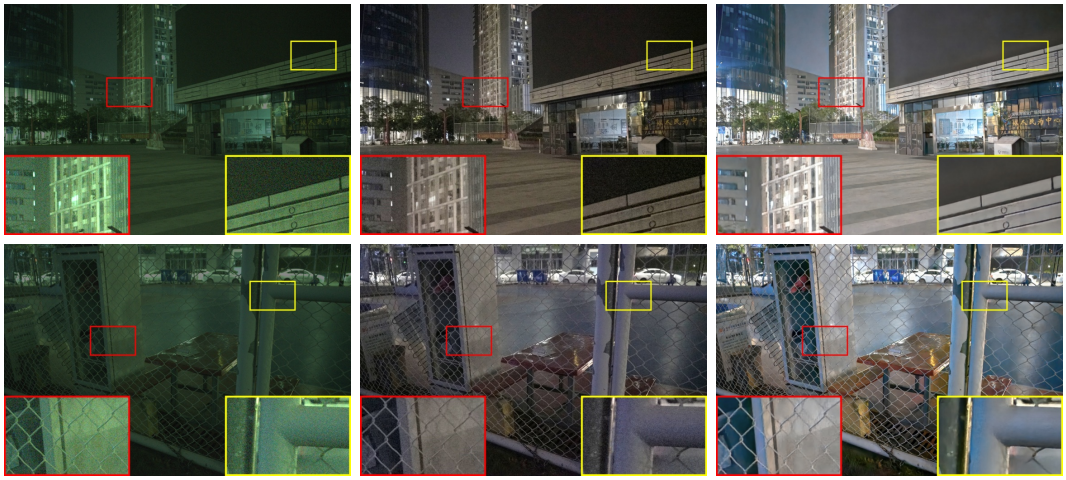


(a) Full transfer (b) Enhance-Net transfer (c) Ground truth

Figure 5.15: Cross-dataset testing. First row: the results of transferring the CameraNet trained on FiveK dataset to the testing image from HDR+ dataset. Second row: the results of transferring the CameraNet trained on HDR+ dataset to the testing image from FiveK dataset. "Full transfer" means transferring the whole CameraNet, while "Enhance-Net transfer" means transferring only the Enhance-Net.

5.3.5 Cross-dataset Testing

We use the HDR+ and FiveK datasets to test the cross-dataset performance of CameraNet. Specifically, we apply the network trained on one dataset to the testing set of another dataset. We only perform subjective evaluation because the two datasets have different types of ground truths, which makes the objective comparison less meaningful. Fig. 5.15 presents two cross-dataset testing examples, from which we can have two observations. On one hand, if we transfer the whole CameraNet trained on one dataset to the raw images of another dataset with a different sensor, the results have erroneous colors and details (see the left column of Fig. 5.15). For example, the result of "FiveK to HDR+" (top left image in Fig. 5.15) exhibits greenish color and noisy details. This is because the Restore-Net depends heavily on the camera sensor, and the mismatched sensor statistics will cause the inaccurate reconstruction of the sRGB image. On the other hand, if we only apply the Enhance-Net to the restored



(a) Raw image (b) Result by Sony ISP (c) Result by CameraNet

Figure 5.16: Comparison with Sony A7S2 ISP in low-light scenarios. Both of the raw images are captured with aperture f3.5, exposure time 1/100s and ISO 12800. A gamma transform with parameter 2.2 is applied to the raw image for better visualization.

images by Restore-Net in another dataset, the results are perceptually acceptable but with a different image style (see the middle column of Fig. 5.15). This is because the restored images are in the similar color space so that the Enhance-Net depend less on the camera sensor.

The above observations imply that when we develop an ISP for a new sensor (possibly with a new CFA pattern), we may not need to completely retrain the CameraNet. We could only retrain the Restore-Net and then refine the Enhance-Net a little. In addition, different Enhance-Nets can be trained for a sensor to obtain different enhancement styles, such as nighttime, portrait, landscape, objects, etc.

5.3.6 Comparison with Traditional ISP

Since there is not a traditional ISP publically available to use, we compare CameraNet with the ISP onboard a Sony A7S2 camera (the same model as the one used in the SID dataset [24]) to demonstrate the advantage of our method over traditional ISP pipeline. Specifically, we use the Sony A7S2 camera to collect several noisy raw images

Table 5.4: Computational complexity of the compared CNN models. The GFLOPS and running time are evaluated on an image of resolution 4032×3024.

	GFLOPS	Running time (sec.)	Number of parameters (mill.)
CameraNet	**3306.69**	**0.892**	26.53
DeepISP-Net [105]	12869.79	2.12	0.629
DeepCamera [100]	4460.35	1.62	**0.467**

in low-light environment with similar settings to those used in the construction of the SID dataset. The JPEG images output by the camera are collected as the results by Sony A7S2 ISP. The results by our approach are obtained by first applying CameraNet trained on the SID dataset to the collected noisy raw images, and then compressing the output sRGB images by JPEG. Fig. 5.16 shows the visual comparison between CameraNet and Sony A7S2 ISP on two raw images. One can see that in such low-light imaging scenario, the Sony A7S2 ISP produces results with residual noise and faded color, while the results by CameraNet exhibit clean structure, high local contrast and vivid color. This demonstrates the powerful image reconstruction capability of learning-based ISP methods in challenging scenarios.

5.3.7 Computational Complexity

In Table 5.4, we compare the computational complexity, running time and number of parameters of the competing CNN-based methods on Nvidia Quadro GV100. We can see that CameraNet has the lowest complexity and fastest speed. To produce an sRGB image of size 4032×3024, it consumes 3306.69 GFLOPS in 0.892s. DeepISP-Net consumes much more GFLOPS than CameraNet and DeepCamera and it runs the slowest. The lower computational complexity of CameraNet is mainly attributed to its multi-scale operations. However, CameraNet has 26.53 million parameters, which consumes much more memory than the other two CNN models. This is because the number of convolution channels grows exponentially in the contracting path of a

UNet module, yielding roughly 33% parameters at the lowest resolution level. Since UNet deploys most of the computations on the lowest resolution level, the proposed CameraNet still has a low GFLOPS consumption.

5.3.8 Limitations

The proposed CameraNet has two main limitations. First, the number of parameters (26.53M) and computational cost (3306.69 GLOPS) are relatively high for application to mobile devices. It is expected that the network can be trimmed and compressed to attain better compactness and efficiency. Second, our CameraNet is designed for single-frame photography. In recent years, burst imaging is becoming more and more popular in mobile cameras, where multiple raw images are captured and fused into one sRGB image. For burst imaging, some additional components should be added to our current CNN architecture, such as frame alignment and fusion. How to compress our network for mobile devices and how to extend it to burst photography will be our future work.

5.4 Summary

We proposed an effective two-stage CNN system, namely CameraNet, for data-driven ISP pipeline learning. We exploited the intrinsic correlations among the ISP subtasks and categorized them into two sets of weakly correlated operations, i.e., restoration and enhancement. Accordingly, a two-stage architecture was adopted in the proposed CameraNet to account for the two sets of operations, facilitating the learning capability while maintaining the model compactness. Two ground truths were specified to train the two-stage model, and a two-step training scheme was employed to train the whole model. Experiments showed that the proposed two-stage CNN framework significantly outperforms the commonly used one-stage framework in deep ISP learning. The proposed CameraNet outperforms state-of-the-art learning-based ISP models on three

benchmark ISP datasets in terms of both quantitative measures and visual perception quality.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

The image signal processing (ISP) pipeline of a camera is a cascade of multiple image processing components to transform the sensor raw data to high-quality displayable images. Typical image processing components include demosaicking, noise removal, white balance, tone mapping and color enhancement. The traditional ISPs employ a simple algorithm for each processing component, which causes various limitations on the results, including residual noise, loss of image details and visual artifacts. In this thesis, we propose several new designs to improve the camera ISP pipeline, including an optimization-based tone mapping algorithm, two deep-learning based image denoising algorithms and a deep-learning-based ISP framework.

As key component for the image perceptual quality, tone mapping is the process to reproduce a displayable standard dynamic range (SDR) image from the high dynamic range (HDR) senor data. The challenge of tone mapping is to compress the dynamic range, enhance the image details and preserve naturalness simultaneously. Many methods in academia have halo artifacts or over-enhancement problem due to improper treatment on image features. In chapter 2, we propose a hybrid ℓ_1-ℓ_0 optimization method for tone mapping. We employ ℓ_1 sparsity prior on image edges to prevent halo artifact, while imposing ℓ_0 sparsity prior on image details to

suppress over-enhancement artifacts. The experimental results demonstrate that the proposed algorithm could produce visually appealing results with minimal artifact and outperforms the state-of-the-art tone mapping methods in the academia.

Since noise corruption is a common and severe problem in camera imaging process, single image denoising is an important component in a camera ISP pipeline to improve recover the clean image details. While the recent deep-learning-based approaches achieve leading performance, they produce over-smooth results due to the trade-off between noise removal and detail reconstruction. In chapter 3, we address this problem by adopting a smooth-and-enhance strategy in designing a convolutional neural network (CNN) for denoising. The proposed denoiser performs normal denoising first and then hallucinates high-frequency details later to produce detail-enriched results. Adversarial training technique is adopted in the training to generate realistic image details. Experiments on synthetic Gaussian noise and real-world noise demonstrate that the proposed method can produce denoised images with notably better perceptual quality than the deep-learning-based denoisers in the literatures.

Burst denoising leverages multiple noisy images of the same scene to reduce noise. It is desirable to apply deep learning technique to real-world burst denoising, where a CNN can be employed to learn the burst denoising process, including adaptation to real-world noise and frame alignment. However, it is difficult to construct a dataset for this purpose because generating ground truths is difficult for dynamic scenes. In chapter 4, we propose a decoupled learning scheme to alleviate this problem. We leverage two complementary datasets, including a video dataset with synthetic noise, and a static burst dataset with real-world noise. We design a decoupled CNN architecture and a training scheme, which can learn the real-world noise adaptation from static burst dataset and learn frame alignment from the dynamic content in the video dataset. Experiments show that under the proposed decouple learning scheme,

our burst denoising CNN achieves leading performance in denoising real-world noisy sequences without the need of a real-world burst denoising dataset.

In the traditional camera ISP pipeline, the individual image processing components are designed independently with little considerations to design them as a whole. This not only leads to error accumulation in the result but also requires long development period on algorithm tuning. In chapter 5, we design a data-driven framework for ISP learning, where a two-stage network, dubbed CameraNet, is developed to replace the traditional ISP pipelines. CameraNet is divided into an image restoration stage and an enhancement stage, which demonstrates excellent capability in learning various types of ISP pipelines. Experiments show that the proposed CameraNet can achieve stably good performance in several benchmark datasets, and outperforms both traditional ISP pipelines and recently proposed deep-learning-based ISP designs in the literatures.

6.2 Future Work

We plan to expand our study in the following directions as the future work:

- **Learning tone mapping operation**. Deep learning technique has demonstrated notable advantages over traditional algorithms in some image processing task. It remains a question whether it can be applied to tone mapping task to make improvement. In the future, we plan to construct a large-scale pairwise dataset with HDR images and the ground truth tone mapped images to develop deep-learning-based tone mapping approach. We will hire several experienced photographers to use image editing software to generate multiple styles of ground truth tone mapped images for each HDR image.

- **Real-world denoising with adversarial learning**. While the proposed smooth-and-enhance denoiser in chapter 3 shows great advantages in detail richness, it could generate some small artifacts, including distorted textures

and color bias, in denoising some real-world noisy images. This is caused by the employment of adversarial training technique, which are known to be unstable. In the future, we will explore how to refine the adversarial learning technique to benefit the real-world image restoration application.

- **Real-time deep burst denoising**. While deep learning has shown great advantage on real-world burst denoising task, it has several limitations when applied to real-time scenario, including large model complexity and slow running time. This is caused by the manner of taking and processing multiple frames simultaneously. In the future, we will explore more runtime-efficient architecture design for burst denoising. For example, developing a recurrent architecture which takes one frame at a time could be a possible solution.

- **Deep ISP pipeline by burst imaging**. Currently the CameraNet proposed in chapter 5 only takes single raw image as input. It is non-trivial to extend it to burst imaging where a collection of raw images are captured as input. The input raw images may contain different exposures and dynamic contents, which brings both challenges and potential benefits to the ISP pipeline design.

- **Hardware deployment**. Although the proposed algorithms, including tone mapping, denoising and ISP pipeline, have high performance in laboratory environment, there are many factors that should be put to consideration when deploying them into hardware. First, computational complexity should be reduced by decreasing some of the network hyper parameters, e.g., number of channels and layers. Second, since the model size is reduced, knowledge distillation technique should be applied to maintain the same amount of learned knowledge. Third, hardware-friendly operations should be adopted in the network design to maximize the infer speed. For example, transposed convolution

should be avoided since it is not well supported by most of the hardware platforms. Finally, one should have a systematic set of knowledge when deploying the algorithms. This is our future work.

Bibliography

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, June 2018.

[2] Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *CVPR Workshops*, pages 1122–1131, 2017.

[3] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, November 2006.

[4] Saeed Anwar and Nick Barnes. Real image denoising with feature attention. In *ICCV*, October 2019.

[5] Martín Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *5th International Conference on Learning Representations, ICLR*, 2017.

[6] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML*, volume 70, pages 214–223, 2017.

[7] Rudolf Arnheim. *Art and visual perception*. Stockholms Universitet, Institutionen för Konstvetenskap, 2001.

[8] Simon Baker and Iain A. Matthews. Equivalence and efficiency of image alignment algorithms. In *IEEE Int. Conf. on Comput. Vis. and Pattern Recognit. (CVPR)*, pages 1090–1097, 2001.

[9] N. Barakat, A. N. Hone, and T. E. Darcie. Minimal-bracketing sets for high-dynamic-range image capture. *IEEE Transactions on Image Processing*, 17(10):1864–1875, October 2008.

[10] Jonathan T. Barron and Yun-Ta Tsai. Fast fourier color constancy. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6950–6958. IEEE Computer Society, 2017.

[11] Bryce E Bayer. Color imaging array, July 20 1976. US Patent 3,971,065.

[12] Sai Bi, Xiaoguang Han, and Yizhou Yu. An l1 image transform for edge-preserving smoothing and scene-level intrinsic decomposition. *ACM Trans. Graph.*, 34(4):78:1–78:12, July 2015.

[13] Simone Bianco, Claudio Cusano, and Raimondo Schettini. Color constancy using cnns. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2015, Boston, MA, USA, June 7-12, 2015*, pages 81–89, 2015.

[14] Yochai Blau, Roey Mechrez, Radu Timofte, Tomer Michaeli, and Lihi Zelnik-Manor. The 2018 PIRM challenge on perceptual image super-resolution. In *ECCV Workshops*, volume 11133, pages 334–355, 2018.

[15] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. In *CVPR*, 2019.

[16] A. Buades, J. Lisani, and M. Miladinović. Patch-based video denoising with optical flow estimation. *IEEE Transactions on Image Processing*, 25(6):2573–2586, June 2016.

[17] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *CVPR*, volume 2, pages 60–65. IEEE, 2005.

[18] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input / output image pairs. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 97–104, 2011.

[19] Jianrui Cai, Shuhang Gu, and Lei Zhang. Learning a deep single image contrast enhancer from multi-exposure images. *IEEE Transactions on Image Processing*, 27(4):2049–2062, 2018.

[20] Jie Cai, Zibo Meng, and Chiu Man Ho. Residual channel attention generative adversarial network for image super-resolution and noise reduction. In *CVPR Workshops*, pages 1852–1861, 2020.

[21] A. Chakrabarti, D. Scharstein, and T. Zickler. An empirical camera model for internet color vision. In *BMVC*, volume 1, page 4. Citeseer, 2009.

[22] Jason Chang, Randi Cabezas, and John W. Fisher. *Bayesian Nonparametric Intrinsic Image Decomposition*. Springer International Publishing, 2014.

[23] Meng Chang, Qi Li, Huajun Feng, and Zhihai Xu. Spatial-adaptive network for single image denoising. In *ECCV*.

[24] C. Chen, Q. Chen, J. Xu, and V. Koltun. Learning to see in the dark. In *CVPR*, pages 3291–3300, June 2018.

[25] Jingwen Chen, Jiawei Chen, Hongyang Chao, and Ming Yang. Image blind denoising with generative adversarial network based noise modeling. In *CVPR*, pages 3155–3164. IEEE Computer Society, 2018.

[26] Qifeng Chen, Jia Xu, and Vladlen Koltun. Fast image processing with fully-convolutional networks. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[27] Yu-Sheng Chen, Yu-Ching Wang, Man-Hsin Kao, and Yung-Yu Chuang. Deep photo enhancer: Unpaired learning for image enhancement from photographs with gans. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6306–6314, June 2018.

[28] D. Cheng, B. Price, S. Cohen, and M. S. Brown. Beyond white: Ground truth colors for color constancy correction. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 298–306, Dec 2015.

[29] Manri Cheon, Jun-Hyuk Kim, Jun-Ho Choi, and Jong-Seok Lee. Generative adversarial network-based image super-resolution using perceptual content losses. In *ECCV*, volume 11133, pages 51–62, 2018.

[30] Yunjey Choi, Min-Je Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 8789–8797. IEEE Computer Society, 2018.

[31] D. Coltuc, P. Bolon, and J. . Chassery. Exact histogram specification. *IEEE Transactions on Image Processing*, 15(5):1143–1152, 2006.

[32] K. Dabov, A. Foi, and K. Egiazarian. Video denoising by sparse 3D transform-domain collaborative filtering. In *Proc. 15th European Signal Processing Conf*, pages 145–149, September 2007.

[33] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, August 2007.

[34] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen O. Egiazarian. Color image denoising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space. In *Proceedings of the International Conference on Image Processing, ICIP*, pages 313–316. IEEE, 2007.

[35] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 369–378, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[36] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *Eur. Conf. on Comput. Vis. (ECCV)*, volume 8692, pages 184–199, 2014.

[37] Frédo Durand and Julie Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Trans. Graph.*, 21(3):257–266, July 2002.

[38] Thibaud Ehret, Axel Davy, Pablo Arias, and Gabriele Facciolo. Joint demosaicking and denoising by fine-tuning of bursts of raw images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8868–8877, 2019.

[39] M. Elad. On the origin of the bilateral filter and ways to improve it. *IEEE Transactions on Image Processing*, 11(10):1141–1151, October 2002.

[40] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, December 2006.

[41] Zeev Farbman, Raanan Fattal, Dani Lischinski, and Richard Szeliski. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Trans. Graph.*, 27(3):67:1–67:10, August 2008.

[42] S. Ferradans, M. Bertalmio, E. Provenzi, and V. Caselles. An analysis of visual adaptation and contrast perception for tone mapping. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):2002–2012, October 2011.

[43] X. Fu, D. Zeng, Y. Huang, X. P. Zhang, and X. Ding. A weighted variational model for simultaneous reflectance and illumination estimation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 2782–2790, June 2016.

[44] Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédo Durand. Deep joint demosaicking and denoising. *ACM Transactions on Graphics (TOG)*, 35(6):191:1–191:12, November 2016.

[45] Michaël Gharbi, Jiawen Chen, Jonathan T. Barron, Samuel W. Hasinoff, and Frédo Durand. Deep bilateral learning for real-time image enhancement. *ACM Transactions on Graphics (TOG)*, 36(4):118:1–118:12, 2017.

[46] Clément Godard, Kevin Matzen, and Matt Uyttendaele. Deep burst denoising. In *ECCV*, pages 560–577, Cham, 2018.

[47] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. pages 2672–2680, 2014.

[48] Cosmin Grigorescu, Nicolai Petkov, and Michel A. Westenberg. Contour and boundary detection improved by surround suppression of texture edges. *Image and Vision Computing*, 22(8):609 – 622, 2004.

[49] B. Gu, W. Li, M. Zhu, and M. Wang. Local edge-preserving multiscale decomposition for high dynamic range image tone mapping. *IEEE Transactions on Image Processing*, 22(1):70–79, January 2013.

[50] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. In *CVPR*, pages 2862–2869, June 2014.

[51] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5767–5777, 2017.

[52] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[53] Samuel W. Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T. Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 35(6), 2016.

[54] K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, June 2013.

[55] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition,CVPR*, pages 770–778, 2016.

[56] Felix Heide, Markus Steinberger, Yun-Ta Tsai, Mushfiqur Rouf, Dawid Pajak, Dikpal Reddy, Orazio Gallo, Jing Liu, Wolfgang Heidrich, Karen Egiazarian, Jan Kautz, and Kari Pulli. Flexisp: A flexible camera image processing framework. *ACM Transactions on Graphics (TOG)*, 33(6):231:1–231:13, November 2014.

[57] Yuanming Hu, Baoyuan Wang, and Stephen Lin. Fc4: Fully convolutional color constancy with confidence-weighted pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4085–4094, 2017.

[58] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

[59] H. Jiang, Q. Tian, J. Farrell, and B. A. Wandell. Learning the image processing pipeline. *IEEE Transactions on Image Processing*, 26(10):5032–5042, Oct 2017.

[60] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II*, pages 694–711, 2016.

[61] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard GAN. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

[62] Hakki Can Karaimer and Michael S. Brown. A software platform for manipulating the camera imaging pipeline. In *European Conference on Computer Vision (ECCV)*, 2016.

[63] Hakki Can Karaimer and Michael S. Brown. Improving color reproduction accuracy on cameras. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6440–6449. IEEE Computer Society, 2018.

[64] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *6th International Conference on Learning Representations, ICLR*, 2018.

[65] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 1646–1654, 2016.

[66] Ron Kimmel, Michael Elad, Doron Shaked, Renato Keshet, and Irwin Sobel. A variational framework for retinex. *International Journal of Computer Vision*, 52(1):7–23, Apr 2003.

[67] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[68] Filippos Kokkinos and Stamatios Lefkimmiatis. Iterative residual cnns for burst photography applications. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 5929–5938. Computer Vision Foundation / IEEE, 2019.

[69] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image

super-resolution using a generative adversarial network. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 105–114, July 2017.

[70] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 105–114, 2017.

[71] S. Lefkimmiatis. Non-local color image denoising with convolutional neural networks. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 5882–5891, July 2017.

[72] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2971–2980. PMLR, 2018.

[73] B. Leung, G. Jeon, and E. Dubois. Least-squares luma-chroma demultiplexing algorithm for bayer demosaicking. *IEEE Transactions on Image Processing*, 20(7):1885–1894, 2011.

[74] Ruoteng Li, Loong-Fah Cheong, and Robby T. Tan. Heavy rain image restoration: Integrating physics model and conditional adversarial learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 1633–1642, 2019.

[75] Yuanzhen Li, Lavanya Sharan, and Edward H. Adelson. Compressing and companding high dynamic range images with subband architectures. *ACM Trans. Graph.*, 24(3):836–844, July 2005.

[76] Z. Liang, W. Liu, and R. Yao. Contrast enhancement by nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, 25(2):673–686, February 2016.

[77] Hai Ting Lin, Seon Joo Kim, Sabine Süsstrunk, and Michael S. Brown. Revisiting radiometric calibration for color computer vision. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 129–136, 2011.

[78] Ce Liu and William T. Freeman. A high-quality video denoising algorithm based on reliable motion estimation. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, pages 706–719, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[79] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S. Huang. Non-local recurrent network for image restoration. In *NeurIPS*, pages 1680–1689, 2018.

[80] C. Lu, J. Shi, and J. Jia. Online robust dictionary learning. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 415–422, June 2013.

[81] Wenye Ma and Stanley Osher. A tv bregman iterative model of retinex theory. *Ucla Cam Report*, pages 10–13, 2010.

[82] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian. Video denoising, deblocking, and enhancement through separable 4-D nonlocal spatiotemporal transforms. *IEEE Transactions on Image Processing*, 21(9):3952–3966, September 2012.

[83] Z. Mai, H. Mansour, R. Mantiuk, P. Nasiopoulos, R. Ward, and W. Heidrich. Optimizing a tone curve for backward-compatible high dynamic range image and video compression. *IEEE Transactions on Image Processing*, 20(6):1558–1571, June 2011.

[84] Julien Mairal, Francis R. Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Non-local sparse models for image restoration. In *ICCV*, pages 2272–2279, 2009.

[85] David R. Martin, Charless C. Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the Eighth International Conference On Computer Vision (ICCV-01), Vancouver, British Columbia, Canada, July 7-14, 2001 - Volume 2*, pages 416–425, 2001.

[86] L. Meylan and S. Susstrunk. High dynamic range image rendering with a retinex-based adaptive filter. *IEEE Transactions on Image Processing*, 15(9):2820–2830, September 2006.

[87] B. Mildenhall, J. T. Barron, J. Chen, D. Sharlet, R. Ng, and R. Carroll. Burst denoising with kernel prediction networks. In *CVPR*, pages 2502–2510, June 2018.

[88] D. Min, S. Choi, J. Lu, B. Ham, K. Sohn, and M. N. Do. Fast global image smoothing based on weighted least squares. *IEEE Transactions on Image Processing*, 23(12):5638–5653, December 2014.

[89] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *6th International Conference on Learning Representations, ICLR*, 2018.

[90] Michael K. Ng and Wei Wang. A total variation model for retinex. *SIAM Journal on Imaging Sciences*, 4(1):345–365, 2011.

[91] R. M. H. Nguyen and M. S. Brown. Fast and effective l0 Gradient minimization by region fusion. In *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, pages 208–216, December 2015.

[92] Rang Ho Man Nguyen and Michael S. Brown. Why you should forget luminance conversion and do something better. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5920–5928, 2017.

[93] Yali Peng, Lu Zhang, Shigang Liu, Xiaojun Wu, Yu Zhang, and Xili Wang. Dilated residual networks with symmetric skip connection for image denoising. *Neurocomputing*, 345:67–76, 2019.

[94] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, 12(11):1338–1351, November 2003.

[95] Guocheng Qian, Jinjin Gu, Jimmy S. Ren, Chao Dong, Furong Zhao, and Juan Lin. Trinity of pixel enhancement: a joint solution for demosaicking, denoising and super-resolution. *CoRR*, abs/1905.02538, 2019.

[96] R. Ramanath, W. E. Snyder, Y. Yoo, and M. S. Drew. Color image processing pipeline. *IEEE Signal Processing Magazine*, 22(1):34–43, Jan 2005.

[97] Rajeev Ramanath and Wesley Snyder. Adaptive demosaicking. *Journal of Electronic Imaging*, 12:633–642, 2003.

[98] Rajeev Ramanath, Wesley E. Snyder, Youngjun Yoo, and Mark S. Drew. Color image processing pipeline. *IEEE Signal Process. Mag.*, 22(1):34–43, 2005.

[99] Aakanksha Rana, Praveer Singh, Giuseppe Valenzise, Frédéric Dufaux, Nikos Komodakis, and Aljosa Smolic. Deep tone mapping operator for high dynamic range images. *IEEE Trans. Image Process.*, 29:1285–1298, 2020.

[100] Sivalogeswaran Ratnasingam. Deep camera: A fully convolutional neural network for image signal processing. In *IEEE Int. Conf. on Comput. Vis. Workshops (ICCVW)*, pages 3868–3878, 2019.

[101] E. Reinhard and K. Devlin. Dynamic range reduction inspired by photoreceptor physiology. *IEEE Transactions on Visualization and Computer Graphics*, 11(1):13–24, January 2005.

[102] Erik Reinhard, Michael Stark, Peter Shirley, and James Ferwerda. Photographic tone reproduction for digital images. *ACM Trans. Graph.*, 21(3):267–276, July 2002.

[103] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[104] S. Roth and M. J. Black. Fields of experts: a framework for learning image priors. In *CVPR*, volume 2, pages 860–867 vol. 2, June 2005.

[105] Eli Schwartz, Raja Giryes, and Alexander M. Bronstein. Deepisp: Toward learning an end-to-end image processing pipeline. *IEEE Transactions on Image Processing*, 28(2):912–923, 2019.

[106] Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *IEEE Conf. on Comput. Vis. and Pattern Recognit. (CVPR)*, pages 1874–1883, 2016.

[107] T. Shibata, M. Tanaka, and M. Okutomi. Gradient-domain image reconstruction framework with intensity-range and base-structure constraints. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 2745–2753, June 2016.

[108] Runjie Tan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Color image demosaicking via deep residual learning. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 793–798. IEEE, 2017.

[109] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang et.al. NTIRE 2017 challenge on single image super-resolution: Methods and results. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1110–1121. IEEE Computer Society, 2017.

[110] J. Tumblin and H. Rushmeier. Tone reproduction for realistic images. *IEEE Computer Graphics and Applications*, 13(6):42–48, November 1993.

[111] Rao Muhammad Umer, Gian Luca Foresti, and Christian Micheloni. Deep generative adversarial residual convolutional networks for real-world super-resolution. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops*, pages 1769–1777, 2020.

[112] Patrick Vandewalle, Karim Krichane, David Alleysson, and Sabine Süsstrunk. Joint demosaicing and super-resolution imaging from a set of unregistered aliased images. In *Digital Photography III, San Jose, CA, USA, January 29-30, 2007*, page 65020A, 2007.

[113] Jianyi Wang, Xin Deng, Mai Xu, Congyong Chen, and Yuhang Song. Multi-level wavelet-based generative adversarial network for perceptual quality enhancement of compressed video. In *ECCV*, volume 12359, pages 405–421, 2020.

[114] Shuhang Wang, Jin Zheng, Hai-Miao Hu, and Bo Li. Naturalness preserved enhancement algorithm for non-uniform illumination images. *IEEE Transactions on Image Processing*, 22(9):3538–3548, 2013.

[115] Xintao Wang, Kelvin C.K. Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *The IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2019.

[116] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. ESRGAN: enhanced super-resolution generative adversarial networks. In *ECCV Workshops*, volume 11133, pages 63–79, 2018.

[117] Greg Ward. A contrast-based scale factor for luminance display. *Graphics gems IV*, pages 415–421, 1994.

[118] Kaixuan Wei, Ying Fu, Jiaolong Yang, and Hua Huang. A physics-based noise formation model for extreme low-light raw denoising. In *CVPR*, pages 2755–2764, 2020.

[119] J. Xu, L. Zhang, D. Zhang, and X. Feng. Multi-channel weighted nuclear norm minimization for real color image denoising. In *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, pages 1105–1113, October 2017.

[120] Li Xu, Cewu Lu, Yi Xu, and Jiaya Jia. Image smoothing via l0 gradient minimization. *ACM Trans. Graph.*, 30(6):174:1–174:12, December 2011.

[121] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision (IJCV)*, 127(8):1106–1125, 2019.

[122] Z. Yan, H. Zhang, B. Wang, S. Paris, and Y. Yu. Automatic photo adjustment using deep neural networks. *ACM Trans. on Graph. (TOG)*, 35(2):11, May 2016.

[123] D. Yang and J. Sun. Bm3d-net: A convolutional neural network for transform-domain collaborative filtering. *IEEE Signal Processing Letters*, 25(1):55–59, January 2018.

[124] H. Yeganeh and Z. Wang. Objective quality assessment of tone-mapped images. *IEEE Transactions on Image Processing*, 22(2):657–667, February 2013.

[125] Lu Yuan and Jian Sun. Automatic exposure correction of consumer photographs. In *Computer Vision – ECCV 2012*, pages 771–785, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[126] Zefeng Yuan, Hengyu Li, Jingyi Liu, and Jun Luo. Multiview scene image inpainting based on conditional generative adversarial networks. *IEEE Trans. Intell. Veh.*, 5(2):314–323, 2020.

[127] Huanjing Yue, Jingyu Yang, Xiaoyan Sun, Feng Wu, and Chunping Hou. Contrast enhancement based on intrinsic image decomposition. *IEEE Trans. Image Process.*, 26(8):3981–3994, 2017.

[128] Zongsheng Yue, Qian Zhao, Lei Zhang, and Deyu Meng. Dual adversarial network: Toward real-world noise removal and noise generation. In *ECCV*, volume 12355, pages 41–58. Springer, 2020.

[129] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-L^1 optical flow. In Fred A. Hamprecht, Christoph Schnörr, and Bernd Jähne, editors, *Pattern Recognition, 29th DAGM Symposium, Heidelberg, Germany, September 12-14, 2007, Proceedings*, volume 4713 of *Lecture Notes in Computer Science*, pages 214–223. Springer, 2007.

[130] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Cycleisp: Real image restoration via improved data synthesis. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 2693–2702. IEEE, 2020.

[131] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, July 2017.

[132] K. Zhang, W. Zuo, and L. Zhang. Ffdnet: Toward a fast and flexible solution for CNN-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, September 2018.

[133] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep CNN denoiser prior for image restoration. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2808–2817. IEEE Computer Society, 2017.

[134] Lei Zhang, Xiaolin Wu, Antoni Buades, and Xin Li. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *Journal of Electronic imaging*, 20:023016, 2011.

[135] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

[136] Xuemei Zhang and Brian A Wandell. A spatial extension of cielab for digital color-image reproduction. *J. of the Soc. for Inf. Display*, 5(1):61–63, 1997.

[137] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual non-local attention networks for image restoration. In *7th International Conference on Learning Representations, ICLR*, 2019.

[138] H. Zhao, O. Gallo, I. Frosio, and J. Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging*, 3(1):47–57, March 2017.

[139] Yihao Zhao, Ruihai Wu, and Hao Dong. Unpaired image-to-image translation using adversarial consistency loss. In *ECCV*, volume 12354, pages 800–815, 2020.

[140] Ruofan Zhou, Radhakrishna Achanta, and Sabine Süsstrunk. Deep residual network for joint demosaicing and super-resolution. *CoRR*, abs/1802.06573, 2018.

[141] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004.

[142] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.

[143] Xiaoou Tang Matt Uyttendaele Ziwei Liu, Lu Yuan and Jian Sun. Fast burst images denoising. *ACM Transactions on Graphics (TOG)*, 33(6), 2014.