Learning Based Methods for Color Constancy and Image Enhancement

Jin Xiao

PhD

The Hong Kong Polytechnic University

2021

The Hong Kong Polytechnic University

Department of Computing

Learning Based Methods for Color Constancy and Image Enhancement

Jin Xiao

A thesis submitted in partial fulfilment of the requirements

for the degree of Doctor of Philosophy

January 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)

_____Jin XIAO_____(Name of student)

Abstract

With the fast development of camera devices and social media, images are nowadays one of the most widely used media in our daily life. However, during the acquisition, formation and transmission processes, images are prone to various types of corruptions, leading to degradation in image quality. The on-camera image signal processing (ISP) algorithms and the image enhancement methods are too crucial to ensure and improve the quality of camera output images. Plenty of efforts have been devoted to the research of ISP and image enhancement, and the recently developed deep learning technique has achieved prominent results in these areas. In this thesis, we leverage deep learning for several fundamental tasks in camera ISP pipeline and image enhancement.

Color constancy is the foremost unit in ISP to correct the color bias of the captured images to cater to the human vision system. In chapter 2, we introduce a multi-domain learning strategy for color constancy to relief from lacking training data by leveraging cross-device datasets. Our method achieves state-of-art performance on the commonly used benchmark datasets. Particularly, our model is capable of transferring to a new device with merely a few training samples, which largely reduces the cost of time-consuming data acquisition stage for camera manufacturers when developing color constancy models for new devices.

Image diffraction blurring is another type of deterioration which blurs the image and degrades the image quality. In chapter 3, we conduct a pioneer work by

constructing a real-world diffraction blur dataset. With the constructed real-world dataset, we further design a progressive learning strategy and a robust loss function to train a deep convolutional neural network for diffraction blur removal. Our model can effectively recover more textures and details from images with diffraction blur than the general image deblurring methods.

Single image super-resolution (SISR) is a fundamental task in image enhancement, which aims to increase the resolution of given images. In this thesis, we focus on the more challenging real-world SISR task, where the image degradation process is much more complicated and unknown. In chapter 4, we learn the degradation model from existing real-world SISR datasets, and use the learned degradation model to synthesize large scale realistic training image pairs. By using the generated realistic SISR image pairs, more robust SISR models can be trained, which exhibit higher generalization performance than previous SISR models, presenting promising visual quality for real-world images.

In chapter 5, we further investigate the real-world SISR problem. We work from another perspective, i.e., designing blind super-resolution models. Specifically, we first estimate the pixel-wise degradation map of the given image, and then utilize a deep CNN whose local filters are dependent on estimated degradation to achieve super-resolution. Our method is able to handle complex non-uniform image degradations in real-world scenarios and achieves leading performance on a wide variety of real-world images with good runtime efficiency.

In summary, in this thesis we tackle several important tasks in camera ISP and image enhancement by leveraging deep learning techniques. Our methods demonstrate state-of-art performances on these tasks.

Keywords: Color constancy, Single image super-resolution, Image diffraction removal, Deep convolutional neural network

Biography

Conference Papers

1. **Jin Xiao**, Shuhang Gu, and Lei Zhang. "Multi-Domain Learning for Accurate and Few-Shot Color Constancy." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

2. **Jin Xiao**, Hongwei Yong, and Lei Zhang. "Degradation Model Learning for Real-World Single Image Super-resolution." Proceedings of 15^{th} Asian Conference on Computer Vision. 2020.

Paper Submitted

1. **Jin Xiao**, Hui Zeng, Jianrui Cai, and Lei Zhang. "Learning to Remove Diffraction Blur in Real-world Photography." Submitted to Pattern Recognition.

2. **Jin Xiao**, Shuhang Gu, and Lei Zhang. "Blind Super-Resolution for Real-World Images." Submitted to CVPR, 2021.

Acknowledgements

First and foremost, I want to express my sincere gratitude to my supervisor, Prof. Lei Zhang, for his continuous guidance, endless support, encouragement and patience throughout the entire duration of my research. It is my great pleasure to be a student of Prof. Zhang, who is always showing me the right direction. He taught me how best to identify a research problem and how to recognize viable solutions hidden in the thorough analysis of the problem. His critical way of thinking will influence me throughout my industry journey in the coming years.

I want to express my gratitude to Dr. Zisheng Cao. During my internship at the imaging group of DJI Co., Ltd, he gave me valuable suggestions and help me getting familiar with the industrial thinking. I also want to thank Dr. Shuhang Gu, Dr. Jianrui Cai, and Dr. Hui Zeng for hosting me in my research. Their valuable suggestions have greatly help me to do good research. I am very thankful for their very constructive suggestions and feel fortunate to have the chance to learn from them.

I want to express my gratitude to my lab and office mates, for their assistance, for their encouragement, and for the wonderful time in The Hong Kong Polytechnic University, I have shared with them.

Finally, I want to thank my family for inspiring me to pursue this route. I want to thank them for their endless love, support, and encouragement.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the fast development and increasing popularity of digital camera devices, storage device, and social media, images are becoming the main media in our daily life with the mission of recording commemorable moments, information dissemination, and etc. However, during the acquisition, formation and transmission of images, they inevitably undergo various kinds of deteriorations, leading to the degradation in visual quality and loss of information. As a result, it is highly desired to develop camera image processing pipeline and image enhancement methods to improve the image quality to present better perceptual experience and recover information for the subsequent image understanding tasks.

The camera processing pipeline takes the signal captured by camera sensor as input, and aims to output the captured image which well adapt for human vision system. It embeds a series of modules, such as de-mosaicking, white balance, color transform, color rendering and etc. On the other hand, due to the limited in-camera computational budget, images produced by camera pipeline can still be in bad quality. Consequently, image enhancement methods are developed to complement the camera pipeline to deliver high quality images. The common practical image enhancement applications includes super-resolution (increase the image resolution), denoising (recover the clean image from noisy one) and deblurring (restore the sharp

image from blurry one).

Plenty of efforts had been done toward developing advanced camera pipeline algorithms, and effective and efficient image enhancement methods. Very recently, the deep learning based methods have achieved prominent results in various computer vision applications. In this thesis, inspired by the powerful representation ability of deep learning, we leverage deep convolutional neural networks for color constancy and several image enhancement tasks. In the remaining of this chapter, we first introduce the principle of image formation in camera system in 1.1. We then introduce the background of image enhancement in Section 1.2. We finally summarize the contributions and organization of this thesis in Section 1.3.

1.1 Image Formation in Camera System

The formation of a digital image in camera consists of roughly three steps: first camera lens focuses light, then sensor captures convert light to digital signal, and finally image signal processor (ISP) processes the captured signal to an image subject to human vision system (HVS). The overview of such a process is illustrated in Fig. 1.1. In the following, we introduce the details of these three steps, and illustrate how they influence the quality of the captured images.

1.1.1 Camera Lens

Camera lens is an optical lens or an assemble of lenses used to focus light to the sensor. There are two fundamental parameters of the camera lens: focal length and aperture size.

Focal length measures how strongly the lens converges light. Shorter focal length has a wider angle of view and captures a greater extent of the scene; longer focal length has a narrower angle of view and therefore shows less of the scene. Meanwhile, the object size in the captured image (i.e., resolution of the captured image)

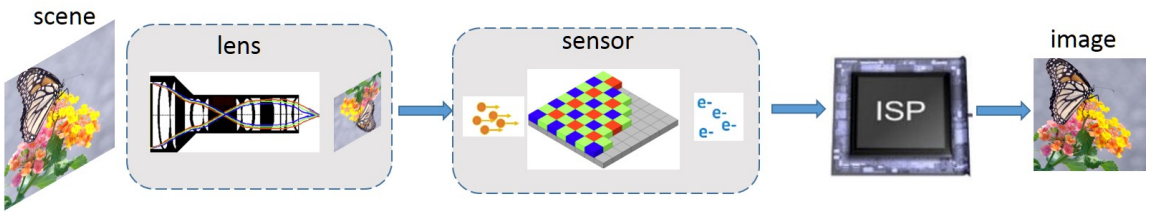


Figure 1.1: An illustration of image formation process in digital camera.

decreases as the focal length gets shorter and increases as the focal length gets longer. This feature is exploited by researchers to constitute the pairwise dataset for super-resolution by collecting a series of images with different resolutions by using varying focal lengths.

Aperture measures how big the opening is for the optical lens to let light in, and is denoted using f-number. The larger the f-number, the smaller the aperture size is. Aperture affects the depth of field (DoF) and the number of photons hit the sensor. In some circumstance, small aperture is necessary, e.g., to endow large DoF for landscape or macro photography, to enable less light for long exposure and to produce starburst effect. However small aperture triggers the light diffraction, leading to unpleasant diffraction blur of captured images. In chapter 3 we introduce our framework for diffraction blur removal.

Despite the significant technological advances in the manufacture and design of camera lens over several decades, there remains imperfections in lens when focusing light to a single point on the image sensor. For example, chromatic aberration usually exists in the high contrast region and causes color fringing; vignetting appears as a gradual variation in intensity along the radial direction from the image center; and distortion appears as straight lines bending inwards or outwards. Accordingly, the camera ISP usually embeds lens correction module to correct some types of distortions.

1.1.2 Sensor

Digital camera uses sensor to convey the captured photons into electronic signal and further the digital signal to make an image. The two main types of electronic image sensors are the charge-coupled device (CCD) and the complementary metal oxide semiconductor sensor (CMOS sensor). One of the main feature of sensor is the spectral sensitivity. Spectral sensitivity characterizes the relative efficiency of detection of light as a function of the light wavelength. We use $\mathbf{C}(\lambda_n)$ to denote the sensor spectral sensitivity, and λ_n for $n = 1, 2, ...N$ represents the discrete sample of wavelength λ. Denote $\mathbf{R}(\lambda_n)$ as the surface reflectance of the captured scene, and denote $\mathbf{I}(\lambda_n)$ as the spectral power distribution of scene illuminant, the image formation can be formulated as:

$$\mathbf{Y} = \Sigma_{n=1}^{N} \mathbf{C}(\lambda_n)\mathbf{I}(\lambda_n)\mathbf{R}(\lambda_n) \tag{1.1}$$

where $\mathbf{Y}$ is the captured image on the camera raw color space.

The human retina uses three kinds of cone cells to conceive color. Theoretically to form a colorimetric image, sensors with at least three spectral bands $\mathbf{C}_c$ with $c \in \{r, g, b\}$ at each pixel are required. In practice, for the consideration of cost and portability, most digital cameras place color filter array (CFA) on top of the sensor pixels, to capture a spatially undersampled images where each pixel contains only one color component. To recover the full color image with color triples at each pixel, the de-mosaicking algorithm is further embedded in ISP.

Sensor Noise Sensor noise is inevitable during the capturing of light photons. There are several different types of sensor noise. Among these, read noise and shot noise are the most distinct ones. The sensor read noise is created during the readout process of the captured electrons, e.g., the analog to digital conversion, the amplification and etc. It is independent of signal level or temperature of the sensor, and

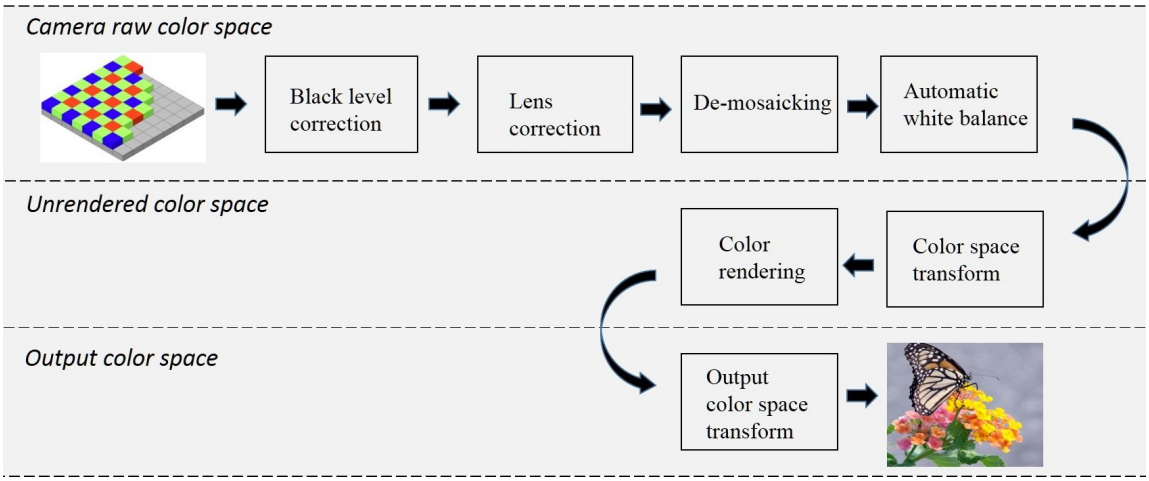


Figure 1.2: An illustration of the typical camera ISP pipeline (adapted from [68, 101]). The detailed implementation of different manufactures may vary.

is larger for faster readout rates. Shot noise originates from the discrete nature of electrons, caused by the arrival process of light photons on the sensor. The shot noise can be modeled by Poisson distribution, with the standard deviation equals the square root of the mean signal level. Apart from these two sensor noise, other types of noise exist. For example the dark current noise and fixed-pattern noise.

Image noise leads to deteriorated visual image quality and loss of image details, adversely affecting the subsequent tasks such as classification and segmentation. Modern digital camera usually embeds noise reduction unit in their ISP to remove sensor noise from raw image signal. Image denoising is also a classic yet active topic in image enhancement area to compensate for the limited in-camera computational resource for noise removal.

1.1.3 Camera Image Processing Pipeline

Different camera manufacturer employs different image signal processor (ISP) to process the raw signal from camera sensor to generate the final image. A typical flowchart of camera ISP is illustrated in Fig. 1.2.

Black Level Correction Most sensors employ a offset when converting photons to voltage and further the digital value, aiming to improve the converting precision when dealing with small signals. As a result, black level correction is important in ISP to remove the non-zeros black level since it would hamper the subsequent processing. Another reason to use black level correction is due to the dark current noise. It is a thermal phenomenon and exists even when no photons are incident on the sensor.

Lens Correction Lens correction aims to correct the distortions presented in the captured images due to lens imperfections. Many cameras provide a calibration matrix to compensate for lens distortion. The calibration matrix is scaled or interpolated to the image size, and then multiplied to the captured image to achieve spatially variant correction.

De-mosaicking As mentioned in section 1.1.2, digital cameras usually employ a single sensor with a color filter array (CFA) to capture color information. The CFA is a single and alternating color filter at each pixel in sensor, and therefore results color subsampling in the captured image, namely the mosaic image. Among all the CFA settings, Bayer pattern is the most common one. It has twice as many green filters as red or blue ones, catering to the human vision system which has higher sensitivity to green light over red and blue. To restore a full color image from the mosaic one, de-mosaicking algorithm is embedded in camera ISP to interpolate the missing colors at each pixel according to the neighborhood information.

Automatic White Balance From the image formation process in Eq. (1.1), one can find out that the color of the image captured by camera is biased due to the scene illumination $\mathbf{I}$. Without the white balance algorithm, the captured image

would appear "blueish" under sunlight and "yellowish" under indoor incandescent light. However, human vision system naturally has the ability to compensate for different illuminants to a scene, named color constancy. To cater to our human vision, automatic white balance is developed and is an important unit in camera ISP aiming at estimating the scene illuminant from the captured image and further correcting the color bias of captured images.

Color Space Transform When a scene is captured by a digital camera, it is represented using the device and scene specific color space, relating to the sensor spectral sensitively as illustrated in Eq. (1.1). To enable the usage of common color manipulation algorithms, the sensor space images are required to transform to the unrendered color space, to represent the scene's color under a device-independent color space. Examples of such unrendered color spaces are CIE XYZ, CIE RGB and etc. The transformation is usually achieved by applying a 3×3 transformation matrix, correlating to the scene illumination. Typical ISP calibrates two matrixes under two extreme illuminations respectively. And the transformation matrix of a given image is derived by interpolating the two pre-defined matrixes according to the illuminant estimated in white balance unit.

Color Rendering This procedure usually applies non-linear function to the color image intensity to improve the color quality of images. The color rendering methods vary among different manufacturers. It generally involves the modification of hue, saturation and exposure, and is usually implemented using look up table (LUT).

Output Color Space Transform After a series of color manipulation modules, the images are transformed to the final output color space for display. Similarly, the transformation is achieved by applying a 3×3 matrix which correlates to the

unrendered color space used. The standard RGB (sRGB) and Adobe RGB are widely used output device color spaces. Usually, the sRGB images will be compressed, e.g., using Jpeg, to save cost for storage and transmission.

1.2 Image Enhancement

Above we introduce the image formation process in a typical camera system, and how the image signal processor (ISP) generates an image from the raw signal. Despite the fast development of camera systems and the ISP, the generated images can still be in bad quality, originating from the complex scene conditions (e.g., extreme low light, moving objects), imperfections in camera hardware, and limited in-camera computational budget. As a result, image enhancement algorithms are developed to further enhance the captured images to deliver pleasant visual quality. In the following, we introduce several important tasks in image enhancement area, i.e., image deblurring, super-resolution and denoising.

1.2.1 Image Deblurring

Image deblurring is an important task in image processing, which aims to recover a sharp latent image with clear edges and details from the given blurry one.

Existing image deblurring algorithms [128, 35, 127, 94, 125, 79, 144, 110, 107] are mainly focused on removing motion-blur or focal-blur, caused by camera shake, object motion or out-of-focus. Traditional methods focus on designing various priors to model the characteristics of blur kernel and natural image priors to regularize the solution space. The recent learning-based methods implicitly exploit priors from external training dataset, and learn a mapping function (either in an end-to-end manner or following the traditional framework) from the blurry images to the clear one. These methods achieve promising results in image deblurring.

Apart from these two classic blurring types, the diffraction blur also severely

deteriorates the image quality however achieves relatively less attention in the community, as mentioned in Sec. 1.1. Moreover, the characteristic of diffraction blur kernel is significantly different from those in motion and out-of-focus blur, making these deblurring models ineffective on removing diffraction blur. To study the problem of diffraction blurring, in chapter 3 we construct a real-world dataset with paired images and leverage the deep learning technique for diffraction blur removal.

1.2.2 Image Super-Resolution

With the fast growth of display devices, memory and network bandwidth, high resolution (HR) images, e.g., 1080p and even 4K, become increasingly prevalent in our daily life. The HR images offer higher pixel density and thereby present vivid details about the original scene. Unfortunately, there is still a great number of low resolution images, due to the limitation of camera devices or compression effects during transmission. Therefore, single image super-resolution (SISR) is developed to recover the high resolution (HR) image from its low resolution (LR) observation. It is a highly valuable technique for improving the quality of images, and is widely used in many practical applications, e.g., the recovery of old pictures, the surveillance, medical and satellite imaging systems.

SISR is a classic yet still active topic in low-level vision, and a plenty of works have been proposed in the past several decades. The traditional methods [85, 34, 119, 57, 132, 36] generally utilize powerful image priors, e.g., the total variation prior, sparse model and nonlocally self-similarity prior, to regularize the solution space for SISR. These methods have made remarkable progresses. While when it comes to complex scenes, these prior based methods generally have limited performance. Besides, the optimization process involved in these methods is time-consuming and infeasible for practical usage.

Recently, the deep learning based SISR methods have shown great advantages

in learning image representations and leveraging external datasets, and consequently improved much the SISR performance [33, 70, 77, 81, 80, 141]. Various SISR networks with specially designed architectures have been proposed. Despite the great success, the task of real-world SISR is still challenging, due to the fact that real-world scenario is much more complicated than the widely used benchmark SISR training datasets. In this thesis, we proposed two learning based methods for the complex real-world SISR task. Compared to existing SISR methods, our methods generate more vivid details with less artifacts on real-world applications.

1.2.3 Image Denoising

Due to the physical limitations of digital cameras, images are prone to various types of noise, e.g., read noise, shot noise and dark current noise, as mentioned in Sec. 1.1.2. The image noise not only deteriorates the image quality, but also hampers the subsequent image understanding tasks, e.g., classification and detection. Image denoising aims to recover the underlying clean image $\mathbf{x}$ from its noisy observation $\mathbf{y}$, where the image degradation process is modeled as $\mathbf{y} = \mathbf{x} + \mathbf{v}$.

Similar to SISR, traditional image denoising methods [86, 38, 126, 56] employ the Bayesian framework and work on designing natural image priors. The nonlocally self-similarity prior, total variation prior and sparse prior have also been widely used for image denoising. Recently with the overall great success of convolutional neural networks (CNNs) in computer vision community, the image denoising performance has also been largely improved by employing CNNs [135, 54, 58, 65].

Before training a denoising-CNN, large scale clean-noisy image pairs are required for supervised learning. The noisy images with real sensor noise can be easily collected, however the construction of corresponding clean images is not straightforward. Consequently, early learning based methods assume the noise distribution as additive white Gaussian noise (AWGN) and synthesize clean-noisy image pairs by adding syn-

thetic noise to clean images. However the task of real-world denoising is still difficult due to the domain gap between AWGN and real sensor noise. To remedy this issue, sophisticated sensor noise model has been proposed. Several works [58, 91] use the Poission-Gaussian distribution to model sensor shot and read noise. There are also efforts focusing on constructing real-world clean-noisy image pairs. They proposed to use digital cameras to capture noisy images, and collect the clean counterparts by capturing the same static scene using low-ISO [98], long exposure time or the averaging of multiple frames. These methods have largely bridge the gap between the distribution of synthetic and real-world noise.

1.3 Organization and Contributions of the Thesis

This thesis consists of four works I have done in my PhD career. During this period, I focus on designing deep convolutional neural networks (CNNs) for improving the image quality. Specifically, we investigate several topics in camera ISP and image enhancement area, i.e., color constancy, diffraction blur removal and single image super-resolution. The organization of the thesis is illustrated in Figure 1.3.

In the first work, we focus on leveraging deep learning technique for color constancy, also referred to as automatic white-balance (AWB) in camera industry. One of the challenges in employing deep CNN for color constancy lies in the costly data acquisition process for each camera device. In chapter 2, we start a pioneer work by introducing the idea of multi-domain learning to color constancy area to leverage the cross-device training data. For different camera devices, we train a branch of networks with shared feature extractor and illuminant estimator, and only employ a camera-specific channel re-weighting module to adapt to the camera-specific characteristics. Our method achieved state-of-the-art performance on the commonly used benchmark datasets. Moreover, given a new unseen device with limited number

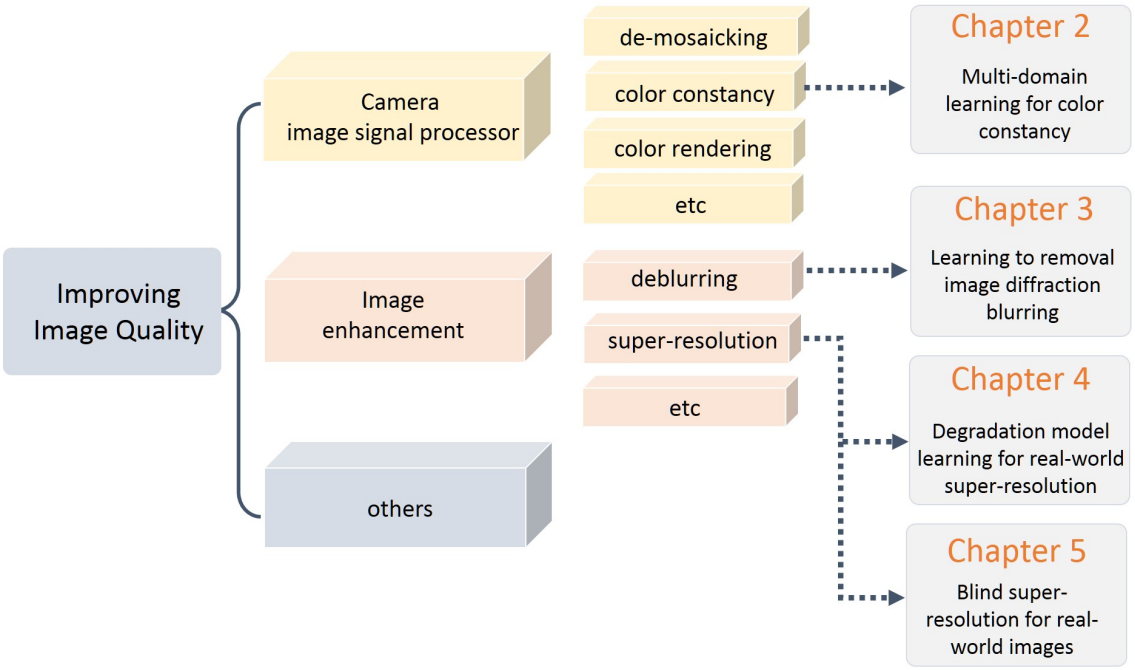


Figure 1.3: The organization of this thesis.

of training samples, our method is capable of delivering accurate color constancy performance.

In our second work, we focus on the task of image diffraction blur removal. little attention has been paid to investigating this practical problem using a learning based method. In chapter 3, we first discuss in detail the characteristics of diffraction blur. To facilitate the research on this important real-world problem, we construct the first real-world diffraction blur dataset. It provides a good benchmark for studying the problem of diffraction blur removal. Moreover, we design a progressive learning architecture and an effective loss function to train a CNN model for diffraction blur removal. Our trained models can recover more textures and details from diffraction blurred image than existing image deblurring methods.

In our third work, we focus on the task of real-world single image super-resolution (SISR). Despite the fast growth of CNN based SISR methods, the real-

world SISR task remains challenging. One of the reason in degrading most SISR methods for real-world low-resolution (LR) images lies in the domain gap between the synthetic LR images used for training, and those real-world LR images during the testing stage. To remedy this issue, In chapter 4, we propose to learn the degradation model from the existing real-world SISR datasets, and use the learned degradation model to synthesize realistic training image pairs. By using the learned degradation model to generate realistic SISR image pairs, more robust SISR models can be trained, which exhibit higher generalization performance than previous SISR models and produce promising visual quality for real-world images.

In our last work, we work on the task of real-world SISR from another perspective. Different from the degradation model learning method in chapter 4, in chapter 5 we design a novel blind super-resolution method toward real-world SISR. we propose to first estimate the pixel-wise degradations in a one-step manner, and then perform super-resolution using a deep CNN, whose local filters are adaptive to the estimated degradations. Specifically, we leverage the image edge map to guide the degradation estimation, and design a pyramid U-shaped sub-network to constrain the smoothness of estimated degradation map, with which a hyper-parameter network is trained to generate the adaptive filters to perform blind SR. Our method is able to handle complex non-uniform image degradations in real-world scenarios and achieves leading results on benchmark datasets as well as real-world LR images with good runtime efficiency.

Chapter 2

Multi-Domain Learning for Color Constancy

In this chapter, we leverage the learning based method to investigate the foremost unit in camera processing pipeline: color constancy, also referred to as automatic white balance in camera industry. Color constancy aims to remove the color bias of captured image caused by scene illumination. Recently, with the great success of deep learning in various applications, significant improvements have also been achieved in color constancy accuracy by using deep neural networks (DNNs). However, existing DNN-based color constancy methods learn distinct mappings for different cameras, which require a costly data acquisition process for each camera device. In this chapter, we start a pioneer work to introduce multi-domain learning to color constancy area. For different camera devices, we train a branch of networks which share the same feature extractor and illuminant estimator, and only employ a camera-specific channel re-weighting module to adapt to the camera-specific characteristics. Such a multi-domain learning strategy enables us to take benefit from cross-device training data. The proposed multi-domain learning color constancy method achieved state-of-the-art performance on three commonly used benchmark datasets. Furthermore, we also validate the proposed method in a few-shot color constancy setting. Given a new unseen device with limited number of training samples, our method is ca-

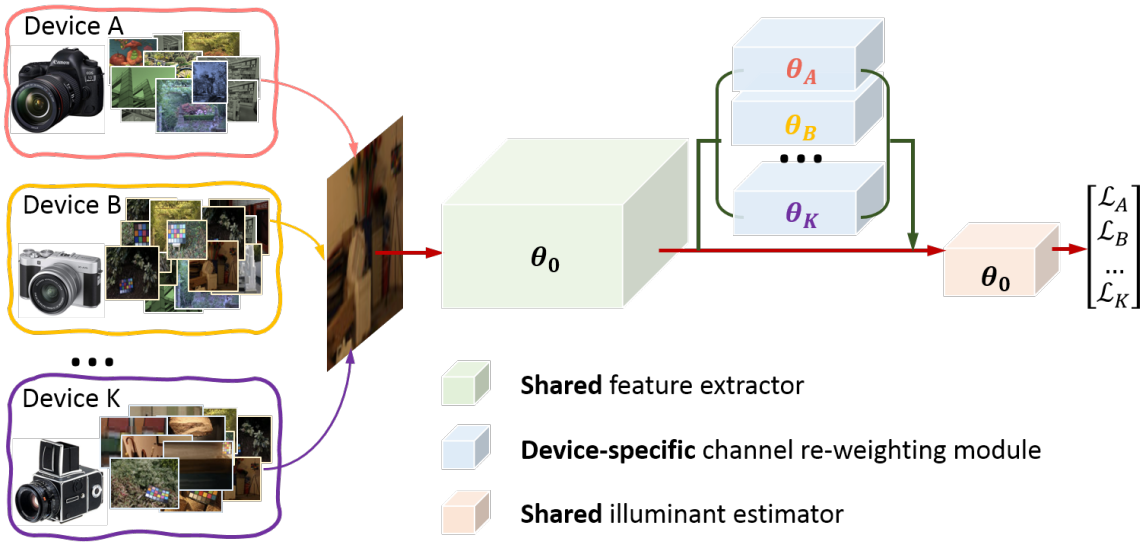


Figure 2.1: Overview of our proposed multi-domain learning color constancy method. We train color constancy networks for different devices simultaneously. Different networks share the same feature extractor and illuminant estimator with shared parameter θ_0, and only have their individual channel re-weighting module with parameters θ_A, θ_B and θ_K, respectively.

pable of delivering accurate color constancy by merely learning the camera-specific parameters from the few-shot dataset.

2.1 Introduction

2.1.1 Color Constancy

Human vision system naturally has the ability to compensate for different illuminants to a scene, named color constancy. The color of images captured by cameras, however are easily affected by different illuminants, and might appear "blueish" under sunlight and "yellowish" under indoor incandescent light. Aiming at estimating the scene illuminant from the captured image, color constancy is an important unit in camera pipeline to correct the color of captured images.

Classical color constancy methods utilize image statistics or physical properties

to estimate illuminant of the scene. The performance of these approaches is highly dependent on the assumptions and these methods falter in cases where assumptions fail to hold [89]. In the last decade, another category of methods, i.e., the learning-based methods, have become more popular. Early learning-based methods [44, 28] adopt hand-crafted features and only learn the estimating function from the training data. Inspired by the success of deep neural networks (DNN) in other low-level vision tasks [55, 53, 32, 111], recently proposed DNN based approaches [15, 106, 62] learn image representation as well as the estimating function jointly, and have achieved state-of-the-art estimation accuracy.

DNN-based methods directly learn a mapping function between the input image and ground truth illuminant label. Given enough training data, they are able to use highly complex nonlinear function to capture the relationship between input images and the corresponding illuminants. However, the acquisition of data for training color constancy network is often costly: firstly, images, each contains the physical calibration objects, in a large variety of scenes under various illuminants must be collected; and then, ground-truth illuminant in each image needs to be estimated through the corresponding calibration object. In addition, as raw data from different cameras exhibit distinct distributions, existing DNN-based color constancy approaches assume each camera has an independent network, and therefore require a large amount of labelled images for each camera. Due to the above reasons, the capacity of existing DNN-based color constancy methods are largely limited by the scale of training dataset. Great attempts have been made to improve the performance of color constancy models under insufficient training data.

2.1.2 Motivation

In this chapter, we proposed multi-domain learning color constancy (MDLCC) method to leverage labelled color constancy data from different datasets and devices. In-

spired by conventional imaging pipelines, which employ camera-specific estimation functions to estimate the illuminant from common low-level features, MDLCC adopts the same feature extractor to extract low-level features from input raw data, and use a camera-specific channel re-weighting module to transform device-specific features to a common feature space for adapting to different cameras. The common feature extractor is trained using data from different devices, and we train device-specific channel re-weighting module with data from different domains for domain adaptation. Such a strategy enables us to address the CSS difference among different cameras while leveraging different datasets to train a more powerful deep feature extractor. The proposed MDLCC framework learns most of the network parameters in each network with a much larger dataset, which significantly improves the color constancy accuracy of each camera.

Besides improving the color constancy performance of well established devices which already have a considerable amount of labelled data, our multi-domain network architecture also enables us to adapt our network to new cameras easily. Given insufficient number of labelled samples from a new camera device, MDLCC only needs to learn the device-specific parameters, and most of the network parameters are inherited from the meta-model which was trained on large scale dataset. Such a few-shot color constancy problem has been investigated in a recent chapter [89]. McDonagh *et al.* [89] utilized the meta-learning technique [42] to learn a color constancy network which is easier to adapt to new cameras. However, as [89] still needs to fine-tune all the network parameters on the few-shot dataset, it has only achieved limited illuminant estimation performance in the few-shot setting. In contrast, the proposed MDLCC approach only needs to learn a small number of parameters from the few-show dataset, and is able to achieve higher few-shot estimation accuracy.

2.2 Literature Review

In this section, we firstly provide an overview of color constancy methods and then introduce previous work in handling insufficient training data. Lastly, we present a brief introduction to the multi-domain methods, which is closely related to our contributions.

2.2.1 Color Constancy: An Overview

Existing color constancy methods can be divided into two categories: the statistics-based methods [18, 17, 41, 115] and the learning-based methods [28, 44, 14, 106, 62, 10, 11]. Based on different priors of the 'true' white-balanced image, statistics-based methods use statistics of the observed image to estimate the illuminant. Despite its fast estimating speed, the simple assumptions adopted in these approaches may not fit well the complex scenes, and thus limited the estimation performance of the statistics-based methods. The learning-based methods learn color constancy models from training data. Early works along this branch used handcraft features, followed by decision tree [28] or support vector regression approach [44] to regress the scene illuminants. To take full advantage of training data, recent works have started to learn features from data for color constancy. In [14], Bianco *et al*. used a 3 layer convolutional network to estimate local illuminants for image patches. Shi *et al*. [106] designed two sub-networks to adapt to the ambiguity of local estimates. In [62], Hu *et al*. proposed the FC^4 approach which introduced a confidence-weighted pooling layer in a fully convolutional network to estimate illuminants from images with arbitrary sizes. Besides extracting features from the raw image, [10, 11] constructed histograms in log-chromatic space, and then apply a learned conv filter to the histograms to estimate illuminant. In spite of the strong performances, learning-based color constancy methods often require a large amount of training data and

have limited generalization capacity to new devices.

2.2.2 Color Constancy with Insufficient Training Data

Since the construction of large scale datasets with enough variety and manual annotations is often laborious and costly, a large number of approaches have been proposed to remedy the insufficiency of training data.

Data augmentation Data augmentation is a commonly used strategy for training models with insufficient data. Currently, most of the learning-based color constancy works have utilized the data augmentation strategy for improving the estimation accuracy. Specifically, random cropping [62] and image relighting [62, 15] are the most commonly used data augmentation schemes. However, as such simple augmentation schemes can not increase the diversity of scenes, they can only bring marginal improvement to the learned color constancy model. Recently, Banić *et al*. [7] designed a image generator to simulate images under various illuminants which however, is faced with the gap between synthetic and real data.

Pre-training Besides data augmentation, another strategy for improving color constancy performance is pre-training. FC^4 [62] started with the AlexNet, which is pre-trained on ImageNet dataset as feature extractor. A smaller learning rate is then used to fine-tune these parameters.

Weakly supervised learning Several works also resorted to unsupervised learning methods. In [112], Tieu *et al*. proposed to learn a linear statistical model on a single device from video frame observations. Banić *et al*. [8] utilize statistical approach to approximate the unknown ground-truth illumination of the training images, and learn color constancy model from approximated illumination values. Currently, the

unsupervised learning approach has achieved better performance than conventional statistical-based methods, but is still not on par with supervised state-of-the-arts.

Inter-camera transformation Due to the distinction among raw images by different devices, large scale dataset needs to be collected for each device. Several work also focused on reducing the workload of constructing camera-specific dataset. Gao *et al*. [45] attempt to discount the variation among different devices by learning a transformation matrix based on camera spectral sensitivity. Banić *et al*. [8] proposed to learn transformation matrix among ground truth distributions of two cameras, before inter-camera experiments. The existing inter-camera approaches only study pairs of sensors and there has not been any works which could leverage data from a large number of devices.

Few-shot learning Recently, McDonagh *et al*. [89] have formulated the color constancy of different cameras and color temperature as a few-shot learning problem. The model-agnostic meta-learning method [42] has been adopted to learn a meta model which is capable of adapting to new cameras using only a small number of training samples. However, as McDonagh *et al*. did not exploit domain knowledge of color constancy and only rely on the adaptation capacity of MAML algorithm [89], only achieved limited performance in the few-shot setting.

2.3 Multi-Domain Learning Color Constancy Network

In this section, we introduce our proposed multi-domain learning color constancy (MDLCC) method. We start with the formulation of color constancy problem and the target of our MDLCC model. Then, we introduce the network architecture of MDLCC as well as how MDLCC could be utilized to solve the few-shot color

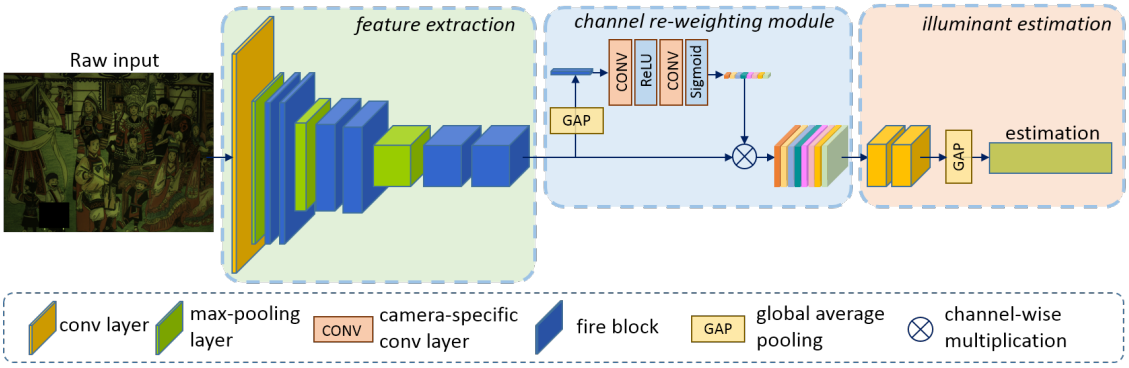


Figure 2.2: The proposed multi-domain color constancy network architecture. We used shared layers among multiple devices for feature extraction. A camera-specific channel re-weighting module was then used to adapt to each device. The illuminant estimation stage finally predicted the scene illuminant.

constancy problem.

2.3.1 Problem Formulation

We focus on the single illuminant color constancy problem which assumes the scene illuminant is global and uniform. Under the Lambertian assumption, the image formation can be simplified as:

$$\mathbf{Y}_c = \Sigma_{n=1}^{N} \mathbf{C}_c(\lambda_n)\mathbf{I}(\lambda_n)\mathbf{R}(\lambda_n), \;\; c \in \{r, g, b\} \tag{2.1}$$

where $\mathbf{Y}$ is the observed raw image. λ_n for $n = 1, 2, ...N$ represents the discrete sample of wavelength λ. $\mathbf{C}_c(\lambda_n)$ represents the camera spectral sensitivity (CSS) of color channel c. $\mathbf{I}(\lambda_n)$ is the spectral power distribution of illuminant, and $\mathbf{R}(\lambda_n)$ denotes the surface reflectance of the scene. Color constancy aims to estimate the illuminant $\mathbf{L} = [L_r, L_g, L_b]$ given the observed image $\mathbf{Y}$. The latent 'white-balanced' image $\mathbf{W}$ can then be derived according to the von Kries model [116] by

$$\mathbf{W}_c = \mathbf{Y}_c / L_c, \;\; c \in \{r, g, b\}. \tag{2.2}$$

Since different cameras use distinct CSS, raw image $\mathbf{Y}$ by different camera occupies different color subspaces. Existing learning based methods generally train

independent model for each device. In this work we combine raw images by different devices to jointly learn a color constancy model. Denote the training data from device k as $D_k = \{\mathbf{Y}_{k,i}, \mathbf{L}_{k,i}\}_{i=1}^{N_k}$, where the superscript k, i denote the device index and sample index, respectively, and N_k is the number of samples for D_k. The proposed multi-domain learning color constancy aims to learn a branch of networks which take raw images from different domains as inputs to estimate the illuminant of the scene:

$$\{\theta_0^*, \theta_k^*\} = \arg\min_{\theta_0, \theta_k} \sum_{k=1}^{K} \sum_{i=1}^{N_k} \mathcal{L}\left(\mathbf{L}_{k,i}, f(\mathbf{Y}_{k,i}; \theta_0, \theta_k)\right), \tag{2.3}$$

where the same network architecture $f(\cdot)$ is adopted for all the devices, and θ_0 and θ_k are the shared and device-specific parameters in the networks, respectively. $\mathcal{L}$ is the loss function which measures the difference between ground truth and estimated illuminants.

2.3.2 Architecture

As introduced in the previous section, we proposed to utilize the same network architecture and only use partial device-specific parameters to adapt to different devices. In order to validate our idea of using multi-domain learning to improve color constancy performance for different devices, we do not investigate new network architecture and utilize FC^4 (SqueezeNet model) as our backbone. Specifically, we assume FC^4 can be divided into two stages: 1) the first 10 layers of network, which gradually reduce the spatial resolution of feature maps, constitute a low-level feature extractor; 2) the last 2 layers of network constitute an estimator which summarizes the extracted feature to estimate the illuminant. Inspired by previous inter-camera approaches [45] which proposed to learn a transformation matrix to correlate different cameras, we propose a device-specific channel re-weighting module and apply different transforms, in the high dimensional feature space, for features

extracted from different devices.

An illustration of our network architecture is presented in Fig. 5.1. For different devices, we employ the same feature extraction module to extract features from input images; and then use the device-specific channel re-weighting module to transform the features; finally, the same estimator is utilized to generate the final illuminant estimation. The details of the feature extraction, channel re-weighting and illuminant estimation modules are introduced as follows.

Feature extraction. We use the first 10 layers in FC^4 as our feature extractor. For the first layer, stride 2 convolution with 64 filters of size 3×3 is used to generate 64 feature maps. Then, 3 blocks, each consists of a max pooling layer and two fire blocks [63] are followed to increase receptive field and further reduce the spatial resolution of feature map by factor 8. The channel dimension of feature maps after each block is 128, 256 and 384 respectively. The ReLU [93] is used as activation function following each conv layer.

Channel re-weighting module. In order to adapt the low-level features from different domains to a common space, we propose a device-specific channel re-weighting module to transform features. Concretely, we derive the scaling factors from statistic of extracted features and device-specific parameters. Denote the output of feature extractor for image $\mathbf{Y}_{k,i}$ as $\mathbf{F}_{k,i}$, we use a global average pooling layer to calculate the mean values for each channel of $\mathbf{F}_{k,i}$. Then, the channel-wise scaling vector $\boldsymbol{\omega}_{k,i}$ can be obtained by:

$$\boldsymbol{\omega}_{k,i} = g_{sigmoid}(\mathbf{W}_{k,b} * g_{ReLU}(\mathbf{W}_{k,a} * \mathbf{z}_{k,i})), \tag{2.4}$$

where $\mathbf{z}_{k,i}$ is the mean values of $\mathbf{F}_{k,i}$, $\{\mathbf{W}_{k,a}, \mathbf{W}_{k,b}\}$ are device-specific parameters, $*$ is the convolution operator, g_{ReLU} and $g_{sigmoid}$ are the ReLU and sigmoid functions, respectively. Eq. (2.4) utilizes two device-specific fully connected layers to generate

the channel scaling factors from the statistics of input feature map. Having $\boldsymbol{\omega}_{k,i}$, the transformed feature $\mathbf{G}_{k,i}$ can be obtained by:

$$\mathbf{G}_{k,i} = \boldsymbol{\omega}_{k,i} \otimes \mathbf{F}_{k,i}, \tag{2.5}$$

where $\otimes$ represents the channel-wise multiplication.

Illuminant estimation. With the transformed feature $\mathbf{G}_{k,i}$, we utilize two convolution layers to estimate local illuminants and the final global illuminant value $\hat{\mathbf{L}}_{k,i}$ is achieved by a subsequent global average pooling layer.

During the training phase, all the training samples contribute to the training of feature extraction and illuminant estimation modules, while only the samples from device k affect the device-specific parameters $\{\mathbf{W}_{k,a}, \mathbf{W}_{k,b}\}$ in the channel re-weighting module.

2.3.3 Few-Shot Color Constancy

MDLCC learns shared and device-specific parameters to leverage the labelled data from different devices. Most of the parameters are shared by different devices and only a small portion (6.7%) of parameters are device-specific. Such a property of MDLCC makes it an ideal architecture for few-shot color constancy. Specifically, given limited number of training samples from a new unseen device, we only need to learn the device-specific parameters from these samples and the shared parameters can be inherited from existing MDLCC models. More details of our few-shot color constancy settings will be introduced in section 2.4.2.

2.4 Experimental Results

In this section, we provide experimental results to show the advantage of our proposed MDLCC. We first present the experimental settings, including training and

testing datasets, as well as implementing details. We then conduct the ablation study to verify the effectiveness of the multi-domain learning strategy and the proposed camera-specific re-weighting module. Furthermore, we compare our MDLCC with state-of-the-art color constancy methods on benchmark datasets. Finally, we evaluate our MDLCC under few-shot setting to validate the capacity of our method for few-shot color constancy problem.

2.4.1 Datasets

We evaluate our proposed method using three widely-used color constancy datasets: the reprocessed [104] Gehler-Shi dataset [46], the NUS 8-camera dataset [27] and the Cube+ dataset [8]. The Gehler-Shi dataset was collected using two cameras, i.e., Canon 1D and Canon 5D. It contains both indoor and outdoor scenes, and comprises 568 scenes in total. The NUS dataset contains 1,736 images which were collected using 8 cameras in about 260 scenes. While the Cube+ dataset is a recently released large scale color constancy dataset. It contains 1,365 outdoor scenes and 342 indoor scenes. And all the images were captured by a Canon 550D camera. For each dataset, we follow previous work [10, 11, 62] to use the linear RGB images for experiments. The linear RGB images were obtained by applying a simple down-sample de-mosaicking operation to the raw images, followed by black-level subtraction and saturation pixel removal.

We follow previous works [11, 62, 27] to use 3-fold cross validation for each dataset. Specifically, for the Gehler-Shi dataset, we used the cross validation splits provided in the author's homepage. The subsets for each camera in NUS dataset contain images from the same scene. To ensure that the same scene would not be in both training and testing sets when combining multiple subsets in the NUS dataset, we split the training and testing set for NUS dataset according to scene content. As for the cube+, we randomly split the testing set into 3 folds for cross validation. We

use the angular error in degree as quantitative measure, which has been utilized in previous methods [10, 11, 62, 27]. In all of our experiments, we report 5 metrics of the angular errors, i.e., the mean, median, tri-mean of all errors, mean of the lowest 25% of errors, and mean of the highest 25% of errors.

2.4.2 Implementing Details

We train our networks with the angular loss:

$$\mathcal{L}(\mathbf{L}, \hat{\mathbf{L}}) = cos^{-1}(\frac{\hat{\mathbf{L}} \odot \mathbf{L}}{||\hat{\mathbf{L}}|| \times ||\mathbf{L}||}), \tag{2.6}$$

where $\odot$ represents the inner product, and $cos^{-1}(\cdot)$ is the inverse of cosine function.

Our framework is implemented based on TensorFlow [3] with CUDA support. For both the multi-domain setting and few-shot setting, we train our networks with inputs of size $384 \times 384 \times 3$. Image random cropping and relighting [62] are used as data augmentations. We employ the Adam solver [72] as optimizer and set the learning rate as 1×10^{-4}. The weight decay value is set as 0.0001 and momentum is set as 0.9. For the experiments with all the training samples, we train our model for 750,000 iterations with batch size 8. While for few-shot experiments, we train our model for 15,000 iterations with batch size 8.

For the multi-domain setting, we train all the parameters from scratch and initialize them with normal distribution. For the few-shot setting, the shareable weights are directly inherited from the meta-model (more details of meta model will be introduced in section 2.4.5) and we only train camera-specific parameters. The camera-specific parameters are initialized with normal distribution.

2.4.3 Ablation Study and Analysis

In this section, we carry out ablation study to evaluate the effectiveness of multi-domain learning as well as our proposed camera-specific channel re-weighting module.

Table 2.1: Ablation study by comparing Single Device model, Multi-device Combination model and our proposed MDLCC model, under different combinations of cameras. The best is shown in red.

Method / Dataset	Single Device Color Constancy					Multi-device Combination					MDLCC				
	Mean	Med.	Tri.	Best 25%	Worst 25%	Mean	Med.	Tri.	Best 25%	Worst 25%	Mean	Med.	Tri.	Best 25%	Worst 25%
Gehler-Shi	1.66	1.14	1.24	0.38	3.86	1.91	1.34	1.41	0.42	4.47	1.62	1.10	1.17	0.36	3.79
NUS-C600D	1.97	1.39	1.54	0.47	4.37	1.92	1.34	1.47	0.44	4.26	1.82	1.26	1.39	0.44	4.15
Gehler-Shi	1.66	1.14	1.24	0.38	3.86	1.89	1.35	1.46	0.41	4.45	1.61	0.99	1.11	0.37	3.79
NUS-C1	2.04	1.45	1.60	0.50	4.55	1.98	1.42	1.54	0.48	4.35	1.87	1.33	1.48	0.46	4.19
Cube+	1.35	0.95	1.02	0.32	3.04	1.35	0.93	1.00	0.31	3.10	1.24	0.83	0.96	0.26	2.97
NUS-Fuj.	2.08	1.59	1.73	0.50	4.45	2.04	1.54	1.66	0.49	4.32	1.97	1.39	1.51	0.45	4.43
NUS-N52	2.33	1.65	1.82	0.50	5.34	2.21	1.53	1.73	0.45	4.89	2.00	1.47	1.53	0.45	4.59
Cube+	1.35	0.95	1.02	0.32	3.04	1.35	0.92	1.01	0.31	3.08	1.26	0.84	0.94	0.25	2.97
Gehler-Shi	1.66	1.14	1.24	0.38	3.86	1.87	1.33	1.46	0.43	4.40	1.59	0.95	1.11	0.37	3.77
NUS-C1	2.04	1.45	1.60	0.50	4.55	2.00	1.43	1.55	0.45	4.39	1.86	1.35	1.49	0.46	4.11
NUS-C600D	1.97	1.39	1.54	0.47	4.37	1.93	1.35	1.45	0.44	4.33	1.65	1.16	1.29	0.35	3.73
NUS-Fuj.	2.08	1.59	1.73	0.50	4.45	2.03	1.55	1.67	0.47	4.36	1.87	1.37	1.48	0.45	4.18
NUS-N52	2.33	1.65	1.82	0.50	5.34	2.25	1.66	1.79	0.44	5.01	1.96	1.38	1.52	0.44	4.54
NUS-Oly.	1.86	1.37	1.51	0.47	4.08	1.80	1.34	1.48	0.46	3.97	1.68	1.15	1.30	0.34	3.85
NUS-Pan.	1.98	1.41	1.48	0.41	4.52	1.90	1.38	1.46	0.42	4.37	1.69	1.20	1.33	0.45	3.73
NUS-Sam.	2.18	1.66	1.75	0.54	4.79	2.13	1.52	1.69	0.52	4.62	1.78	1.33	1.42	0.41	3.95
NUS-Son.	1.91	1.51	1.56	0.55	4.05	1.86	1.47	1.54	0.53	3.89	1.74	1.36	1.44	0.46	3.70
Cube+	1.35	0.95	1.02	0.32	3.04	1.36	0.92	1.05	0.33	3.15	1.24	0.84	0.95	0.27	2.95

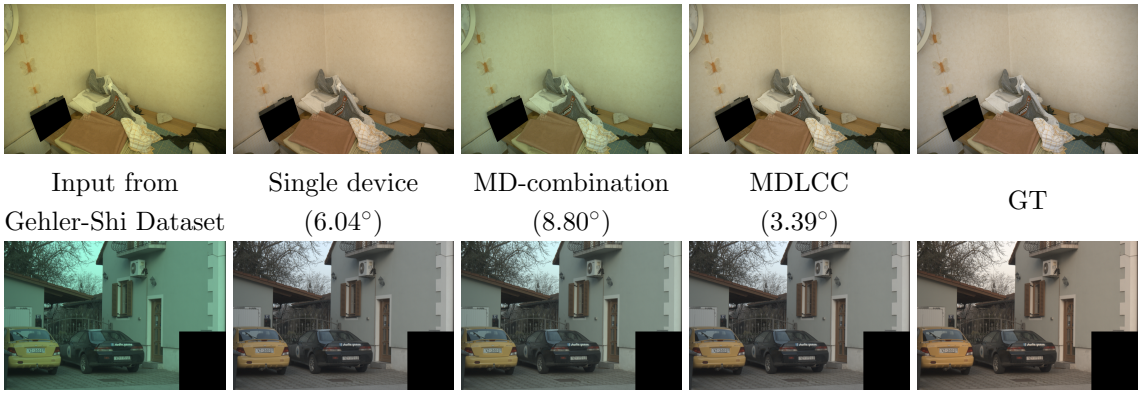


Figure 2.3: Visualization of color constancy results by single device color constancy model, multiple device combination model, and our proposed MDLCC model. Images are converted to sRGB for visualization.

To validate the effectiveness of multi-domain color constancy, we implement two variants: 1) single device color constancy and 2) multi-device combination model. Concretely, the single device color constancy model utilizes our network architecture and trains independently network for each device; the multi-device combination method collects training data from all devices and trains a unique network to process images from different devices. For fair comparison, all the hyper parameters are kept the same as in our MDLCC approach. Furthermore, in order to analyze the effect of device number for our multi-domain learning model, we present 4 groups of experiments which utilize images from different numbers of cameras for training. The details of the combined cameras are listed in Table 2.1. In the last group, we combine all the cameras from Gehler-Shi, NUS and Cube+ dataset, which contain 11 different cameras in total. The quantitative performances are listed in Table 2.1.

Multi-domain learning Compared with single device approach which learns distinct network on each dataset, our method achieves better performance on all the sub-datasets. Even for the large scale Cube+ dataset which contains 1707 training samples, data from related domains is beneficial. This clearly demonstrates the

Table 2.2: Color constancy results by different methods on reprocessed Gehler-Shi [104], NUS [27] and Cube+ dataset [8]. The best and second metric is shown in red and blue respectively.

Method \ Dataset	Gehler-Shi					NUS					Cube+				
	Mean	Med.	Tri.	Best 25%	Worst 25%	Mean	Med.	Tri.	Best 25%	Worst 25%	Mean	Med.	Tri.	Best 25%	Worst 25%
White-Patch [17]	7.55	5.68	6.35	1.45	16.12	9.91	7.44	8.78	1.44	21.27	6.80	3.85	5.21	0.68	16.93
Grey-world [18]	6.36	6.28	6.28	2.33	10.58	4.59	3.46	3.81	1.16	9.85	3.52	2.55	2.82	0.60	7.98
Edge-based Gamut [9]	6.52	5.04	5.43	1.90	13.58	4.40	3.30	3.45	0.99	9.83	-	-	-	-	-
1st-order Gray-Edge [115]	5.33	4.52	4.73	1.86	10.03	3.35	2.58	2.76	0.79	7.18	3.06	2.05	2.32	0.55	7.22
2nd-order Gray-Edge [115]	5.13	4.44	4.62	2.11	9.26	3.36	2.70	2.80	0.89	7.14	3.28	2.34	2.58	0.66	7.44
Shades-of-Gray [41]	4.93	4.01	4.23	1.14	10.20	3.67	2.94	3.03	0.98	7.75	3.22	2.12	2.44	0.43	7.77
Bayesian [46]	4.82	3.46	3.88	1.26	10.49	3.50	2.36	2.57	0.78	8.02	-	-	-	-	-
Natural Image Statistics [47]	4.19	3.13	3.45	1.00	9.22	3.45	2.88	2.95	0.83	7.18	-	-	-	-	-
Spatio-spectral Statistics [22]	3.59	2.96	3.10	0.95	7.61	3.06	2.58	2.74	0.87	6.17	-	-	-	-	-
Intersection-based Gamut [9]	4.20	2.39	2.93	0.51	10.70	-	-	-	-	-	-	-	-	-	-
Pixels-based Gamut [9]	4.20	2.33	2.91	0.50	10.72	5.27	4.26	4.45	1.28	11.16	-	-	-	-	-
Cheng 2014 [27]	3.52	2.14	2.47	0.50	8.74	2.18	1.48	1.64	0.46	5.03	-	-	-	-	-
Exemplar-based [67]	2.89	2.27	2.42	0.82	5.97	-	-	-	-	-	-	-	-	-	-
Corrected-Moment [40]	2.86	2.04	2.22	0.70	6.34	2.95	2.05	2.16	0.59	6.89	-	-	-	-	-
Regression Tree [28]	2.42	1.65	1.75	0.38	5.87	-	-	-	-	-	-	-	-	-	-
CCC [10]	1.95	1.22	1.38	0.35	4.76	2.38	1.48	1.69	0.45	5.85	-	-	-	-	-
DS-Net (HypNet+SelNet) [106]	1.90	1.12	1.33	0.31	4.84	2.24	1.46	1.68	0.48	6.08	-	-	-	-	-
FC^4 (SqueezeNet-FC^4) [62]	1.65	1.18	1.27	0.38	3.78	2.23	1.57	1.72	0.47	5.15	1.35	0.93	1.01	0.30	3.24
FFCC (model J) [11]	1.80	0.95	1.18	0.27	4.65	1.99	1.31	1.43	0.35	4.75	1.38	0.74	0.89	0.19	3.67
FFCC+metadata+semantics [11]	1.61	0.86	1.02	0.23	4.27	-	-	-	-	-	-	-	-	-	-
MDLCC	1.58	0.95	1.11	0.37	3.77	1.78	1.29	1.40	0.42	3.97	1.24	0.83	0.92	0.26	2.91

effectiveness of multi-domain learning in the color constancy area.

Camera-specific channel re-weighting module By comparing single device results and multi-device combination results, we found that directly combining several datasets without the camera-specific module can not constantly improve the color constancy performance. It might lead to improved performance for one camera but degrades severely for the others. For example, when combining Gehler-Shi with NUS-C600D, the performance on Gehler-Shi dataset degrades dramatically from 1.66 to 1.91 in mean error. This reveals that directly combining multiple dataset without device-specific module can not take full advantage of the cross-device training data. While, by adopting the camera-specific channel re-weighting module, our MDLCC approach significantly outperforms the multi-device combination baseline.

Number of devices From Table 2.1 we also observed that by increasing the number of devices in MDLCC, the performance can be further improved. This is because more training samples comprise more scenes and illuminants, and are beneficial for learning more generalized representations. For example, the mean error of MDLCC on NUS-600D is 1.82 when combining with Gehler-Shi, which can be further decreased to 1.65 when combining with all the other cameras. This also demonstrates the effectiveness of our proposed camera-specific channel re-weighting module. Our model is still effective in handling 11 devices.

2.4.4 Comparison with State-of-the-art

In this section, we compare our proposed multi-domain color constancy approach with other color constancy algorithms. We compare our approach with competing methods on the Gehler-Shi [104], NUS [27] and Cube+ [8] datasets. For the NUS dataset, we follow previous work [11, 62] and take the geometric mean of each metric over 8 cameras. We train our model by combining all the devices in the three datasets. The results of comparison methods on the Gehler-Shi dataset and NUS dataset are

Table 2.3: Comparison of few-shot color constancy models.

Test set \ Method		Single device	FMLCC [89]		MDLCC			
			K=10	K=20	K=1	K=5	K=10	K=20
NUS-C1	Mean	2.04	–	–	2.93	2.36	2.27	2.18
	Median	1.45	–	–	2.27	1.72	1.61	1.59
	Tri-mean	1.60	–	–	2.40	1.87	1.81	1.75
	Best 25%	0.50	–	–	0.95	0.60	0.57	0.51
	Worst 25%	4.55	–	–	6.05	5.08	4.97	4.80
Cube	Mean	1.21	1.63	1.59	2.02	1.63	1.56	1.47
	Median	0.85	1.08	1.02	1.75	1.20	1.14	1.06
	Tri-mean	0.90	1.20	1.15	1.83	1.30	1.24	1.14
	Best 25%	0.23	0.31	0.30	0.85	0.50	0.43	0.39
	Worst 25%	2.85	3.89	3.85	3.67	3.46	3.33	3.27
Gehler-Shi	Mean	1.66	2.66	2.57	3.00	2.43	2.32	2.26
	Median	1.14	1.91	1.84	2.32	1.76	1.68	1.60
	Tri-mean	1.24	1.99	1.94	2.49	1.94	1.83	1.75
	Best 25%	0.38	0.49	0.47	0.88	0.59	0.57	0.56
	Worst 25%	3.86	6.20	6.11	6.24	5.33	5.17	5.08

Input from Cube Dataset | $K = 1$ (1.68°) | $K = 5$ (0.64°) | $K = 10$ (0.48°) | $K = 20$ (0.31°) | GT

Input from Gehler-Shi | $K = 1$ (3.83°) | $K = 5$ (0.93°) | $K = 10$ (0.54°) | $K = 20$) (0.39°) | GT

Figure 2.4: Visualization of few-shot color constancy results. Images are converted to sRGB for visualization. The two input images are taken from Cube and Gehler-Shi dataset respectively. We present the few-shot color constancy results with different training sample K. The angular error in degree is also given.

collected from [11, 62]. While, for the Cube+ dataset, we present the results using open source codes from the authors' webpages. We retrain the FFCC [11] and FC^4 [62] models on the Cube+ dataset, and the hyper-parameters have been carefully tuned to achieve the best performance.

The experimental results are listed in Table 2.2. Except the state-of-the-art FFCC approach, the proposed MDLCC outperforms all competing approaches in all metrics. Specifically, our model constantly outperforms our backbone architecture, i.e., the FC^4 approach, this clearly validates the effectiveness of multi-domain learning for color constancy. Compared to the FFCC approach, our model generally outperforms the base FFCC model which only exploits image content for color constancy, and is comparable to the full FFCC model which additionally takes the camera metadata (exposure setting and camera info) and semantic information as inputs. Concretely, our model shows better performance in terms of mean error and the worst 25% of mean errors, while inferior performances in the other three metrics. A possible reason is that our loss function has the tendency to reduce the average error over all training samples, which better fits the mean error and worst 25% metrics.

2.4.5 Few-shot Evaluations

In this section, we conduct experiments to validate the capacity of the proposed model for few-shot color constancy problem. We used the Gehler-Shi, Cube dataset and one subset from NUS (NUS-C1) as the few-shot testing datasets. Note that Cube dataset is a subset from Cube+ which contains only the outdoor scenes. We choose Cube instead of Cube+ for the purpose of directly comparing our method with the recently proposed Few-shot Meta-Learning Color Constancy method (FMLCC) [89]. For training our model, we use the remaining 7 datasets, i.e., 7 subsets from NUS dataset, as the training set and only finetune those device-specific parameters on the few-shot dataset. Specifically, we vary the number of few-shot sample K as 1, 5, 10

and 20 respectively, for thoroughly validating our method. We split each test dataset into three folds. For each fold, we randomly chose K samples from the remaining folds to construct the training samples, which were used to learn the camera-specific parameters. To avoid the randomness and disturbance by the selection of K training samples, we repeated the few-shot experiments for 10 times, each with different random choices of K images. We then present the average of each metric over 10 runs. The few-shot performances are listed in Table 2.3. We choose FMLCC [89] for comparison and the results of FMLCC are copied from the original chapter [89]. The performance of the single device color constancy, which used the whole dataset for training, is also provided for reference.

Compared with previous few-shot color constancy approach FMLCC [89], our model achieved much better results in most of metrics. In addition, as FMLCC needs to fine-tune all the network weights, they might not be able to deliver good results for extreme few-shot cases, for example $K = 1$. While, as our model only requires retraining the camera-specific weights, we can still obtain good color constancy performance. From Table 2.3 and Table 2.2, one can see that with only single shot ($K = 1$), our model outperforms most of statistical-based approaches. Moreover, when using $K = 20$ training samples, our model achieves comparable performance with single device model, which used the whole dataset for training. Some visual examples of our few shot color constancy results are provided in Fig. 2.4.

2.5 Conclusion

Deep networks can largely improve the color constancy accuracy with large scale annotated dataset. However, the acquisition of such dataset is laborious and costly, especially for color constancy problem which requires independent dataset for each camera due to the distinction in devices. In this chapter, we start a pioneer work

to leverage the multi-domain learning method for color constancy problem. Specifically, we utilized training data by different devices to train a single model, to learn complementary representations and improve generalization capability. Experimental results show that with the proposed shareable modules and camera-specific module, our model achieves much better results than training independent model for each device, and also achieves state-of-the-art performance on three benchmark datasets. We also tested the color constancy performances under few-shot setting. Experimental results show that the proposed model can effectively adapt to a new device with only a few, e.g., 20, training samples.

Chapter 3

Learning to Remove Diffraction Blur in Real-World Photography

In this chapter, we investigate another factor leading to image quality deterioration: image diffraction blurring. As mentioned in Chapter 1.1.1, diffraction is the nature of light when passing through small holes, e.g., the small apertures in camera lens. The diffraction blur deteriorates the image details and hinders the further improvement of image resolution under restricted sensor size. The existing physical or optical solutions for overcoming diffraction blur are infeasible for consumer camera devices, and the general image deblurring methods cannot effectively remove the diffraction blur. To the best of our knowledge, little attention has been paid to investigating this practical problem using a learning based method. In order to facilitate the research on this important problem, in this chapter we make the first attempt to remove the diffraction blur in real-world photography using a learning based approach. Specifically, we first discuss in detail the characteristics of diffraction blur as well as its differences from the general image blurs caused by motion or defocus. We then construct a real-world diffraction blur (ReDB) dataset, which consists of 333 image pairs with and without diffraction. We further design a progressive learning strategy and a robust loss function to train a neural network for diffraction blur removal. Experimental results show that our model achieves much better diffraction blur removal

results than existing image deblurring algorithms.

3.1 Introduction

3.1.1 Diffraction Blur Removal

The aperture size plays a key role in taking photos [122]. Small aperture is necessary to endow large depth-of-field (DoF [97]) when framing a front-to-back clear spectacle landscape or exhibiting details in macro or product photography. Because of the wave nature of light [123], however, small aperture inevitably triggers the diffraction of light, leading to unpleasant diffraction blur of captured images. An example image with diffraction blur captured by a Nikon D810 is shown in Fig. 3.1(b). As will be discussed later in Sec. 3.3, the diffraction blur becomes severer with the decrease of aperture size and pixel size in camera sensors. Given the fact that the sensor size is usually limited in consumer camera devices, diffraction becomes one of the most important constraints to further improve the image quality of consumer camera devices.

Physical and optical solutions for overcoming diffraction blur usually needs new material, e.g., super-lenses, immersion technique, to enlarge numerical aperture. Unfortunately, these solutions are too expensive and inconvenient for consumer camera devices. Existing image deblurring algorithms are mainly focused on removing blurs caused by motion and defocus [128, 35, 127, 94, 125, 79, 144, 110, 107], whose patterns are significantly different from diffraction blur. As shown in Fig. 3.1(c), directly applying existing deblurring algorithms cannot effectively remove the diffraction blur, and employing the general sharpening algorithms can hardly recover the lost details. It is highly desirable to develop new diffraction blur removal method to address this issue, which however has attracted little attention in the computer vision community.

(a) Image captured by Nikon D810

f/22 f/18 f/14 f/7.1

(b) Details of images taken at different aperture sizes

SRN [110] ECP [130] PS USM Ours

(c) Diffraction removal results by different methods

Figure 3.1: Illustration of (a) an image captured by Nikon D810, (b) the details of the same scene captured using different aperture sizes and (c) diffraction blur removal results (on image taken using f/22 aperture size) obtained by general image deblurring models SRN [110], ECP [130], the PhotoShop sharpening algorithm and our model.

3.1.2 Motivation

In order to facilitate the research towards diffraction blur removal, in this chapter we make the first attempt to solve this problem using a learning approach. The non-uniform property of diffraction blur makes it difficult to simulate the realistic degradation of diffraction blur. We thus construct the first real-world diffraction blur (ReDB) dataset to enable training and evaluation of diffraction blur removal models. Specifically, we use different aperture sizes to capture a set of images of the same scenes. The clear in-focus region of images taken at an appropriate aperture are used as the groundtruth while images taken at smaller aperture sizes suffer from different degrees of diffraction blur. We conduct image registration to obtain globally aligned image pairs to enable pairwise training. Our ReDB dataset consists of 333 different scenes captured by two digital single-lens reflex (DSLR) cameras.

The constructed ReDB dataset enables us to train a convolutional neural network (CNN) to remove the diffraction blur. Training a model that directly maps from the severely diffraction blurred images to the clear counterparts, is a typical ill-posed problem due to the heavy loss of details. To promote the recovery of textures, we propose a progressive learning strategy to progressively recover the details by leveraging the guidance of images captured at increasingly wider apertures in our ReDB. Meanwhile, the widely-used Mean Square Error (MSE) loss function gives equal emphasis on both image smooth and texture regions. We thus design a new loss function aiming at recovering edges and textures. Combining these two strategies, our model can recover better image quality from the diffraction blurred image, as shown in Fig. 3.1(c).

3.2 Literature Review

Diffraction is a principle limit to the resolution of any optical system. Considerable efforts have been made to break the diffraction limits and improve the resolution of optical systems in microscope or telescope. The immersion technique [87, 114], superlenses [138, 69, 39] and the differential interference contrast microscopy [6, 5] have been invented to enlarge the numerical aperture and improve the resolving ability of microscope. The near-field technique [134, 99] is developed to capture extra information contained in the evanescent wave which is unlimited by diffraction and can propagate to the sensor. However, all these physical techniques require complex fabrication or enormous cost, which are infeasible for consumer camera devices.

Several work also proposed to use digital image processing techniques to recover information from diffraction limited imagery. HARRIS *et al.* [60] analyzed the diffraction-limited optical system and further proposed to use digital shift of diffraction images to extract information. Wang *et al.* [117] proposed a support vector machine based classifier to discriminate cancel cells from diffraction images. However, the restoration of diffraction blurred images remains an open problem. In this chapter, we attempt to remove diffraction blur and improve image quality using a cost-effective learning based method.

Traditional blind image deblurring methods focus on incorporating priors of the degradation kernel and natural image into an optimization framework. The sparse or gradient sparse prior [127, 143, 76] and spectral prior [49] have been proposed to regularize the estimation of blur kernels. The sparse prior [128, 144, 35, 96], nonlocal self-similarity prior [71], dark and bright channel prior [94, 130] have been exploited as natural image priors to maximize a posterior framework for image deblurring. These prior based methods generally have limited performance when handling complex scenes, and their optimization process is time-consuming especially on high resolution

images.

Recently, training CNN models for image deblurring has obtained promising performance. Earlier work followed the traditional scheme to iteratively update the estimated blur kernel and clear image [107, 103]. Specifically, Sun *et al.* [107] trained a CNN to explicitly estimate the blur kernel of each patch, which were then used to deconvolve the blurred image. Schuler *et al.* [103] further combined kernel estimation and image estimation in an end-to-end framework. Li *et al.* [39] trained a CNN to distinguish between clear and blury images. The network was then embedded as image priors into the maximization of a posterior framework. Tao *et al.* [110] proposed a scale recurrent network to directly output clear images via a multi-scale strategy, which can effectively solve large motion blur. Most of these models were trained and evaluated on images with motion blur. However, the characteristics of diffraction blur are significantly different from motion blur, making these deblurring models ineffective on removing diffraction blur. To study the problem of diffraction blur removal, we construct a real-world dataset with paired images. A deep learning based method was then proposed for diffraction blur removal.

It is worth mentioning that there are also many other sources of blur caused by different elements including lens aberration, lens imperfection, light integration and anti-aliasing filter in an imaging system. These blurs may also correlate with the diffraction blur. In this chapter, we mainly focus on the diffraction blur which is most significant when using small aperture.

3.3 Diffraction Blur in Optical Imaging Systems

Diffraction comes from the wave nature of light, defined as the bending of waves when passing through an aperture. In a camera system, since the lens has the ability to focus parallel rays to a point, the equivalent distance between aperture

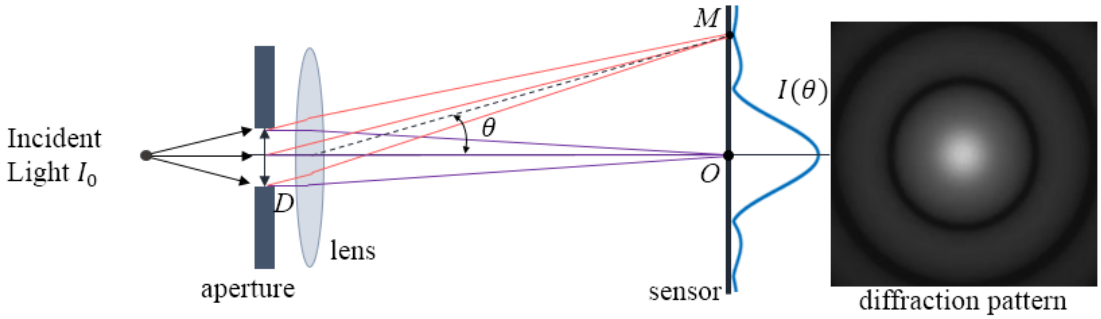


Figure 3.2: Illustration of the diffraction effect in a camera system.

and observation plane is infinite. The diffraction of an ideal point source light in a camera system can be modelled by the Fraunhofer equation [16].

As illustrated in Fig. 3.2, a point light source with intensity I_0 travels into the camera lens and diffracts by a small aperture, forming an alternating light and dark rings, namely the diffraction pattern. According to the Fraunhofer equation [16], the angular distribution of intensity $I(\theta)$ of light diffracted by a circular aperture is:

$$I(\theta) = I_0(\frac{2J_1(\rho)}{\rho})^2 \quad with \quad \rho = \frac{\pi D \sin\theta}{\lambda}, \tag{3.1}$$

where λ denotes the wavelength, D is the diameter of the circular aperture, θ is the angular between direction of diffracted light and incident light, and $J_1(\rho)$ is the 1^{st} kind and 1^{st} order Bessel function. Usually, the outer rings are not apparent and can be ignored. The bright central region is the so-called Airy disk [124]. The diffraction blur occurs when the diameter of the Airy disk is larger than the sensor pixel size, and becomes severer with the decrease of aperture and pixel size. For a certain camera sensor with pixel size d, the diffraction limit aperture (DLA), at which diffraction becomes distinct, can be derived according to the Rayleigh criterion [31] as:

$$DLA = d/1.22\lambda. \tag{3.2}$$

Two characteristics of diffraction blur can be observed from the above analyses. First, diffraction blur is non-uniform since the diameter of Airy disk is characterized

Table 3.1: Information of the ReDB dataset and the employed cameras devices.

	Canon 5D3	**Nikon D810**
Lens	35mm prime	24-105mm zoom
Sensor type	full frame	full frame
Resolution	5760×3840	7360×4912
Pixel size	6.3um	4.9um
DLA	$f/11$	$f/9.0$
Apertures	$f/22, f/18, f/14$	$f/22, f/18, f/14$
Total scenes	167	166
Image fotmat	TIFF	TIFF

by the wavelength of incident light. Second, the degree of diffraction blur is correlated to both the aperture size and pixel size in camera sensor. These two properties distinguish diffraction blur from those blurs caused by motion and defocus.

3.4 Real-world Diffraction Blur Dataset

In order to leverage the learning approach for diffraction blur removal, we propose to construct a diffraction blur dataset. The diffraction process of a practical scene is much more complex than an ideal point source based model described in Eqs. 3.1 and 3.2, and thus it is very difficult to simulate realistic diffraction blurred images. We instead construct a real-world diffraction blur dataset to learn diffraction removal models.

3.4.1 Image collection

Eq. 3.2 reveals that using aperture larger than the DLA can avoid diffraction blur. This motivates us to capture a set of images of the same scenes using different aperture sizes to produce diffraction blurred images and their clear counterparts. Two DSLR cameras with different pixel sizes are employed for data collection: Canon 5D3 and Nikon D810. Each scene is captured using 4 typical apertures: $f/22$, $f/18$, $f/14$ and $f/7.1$ Images captured at the former three small apertures suffer from different

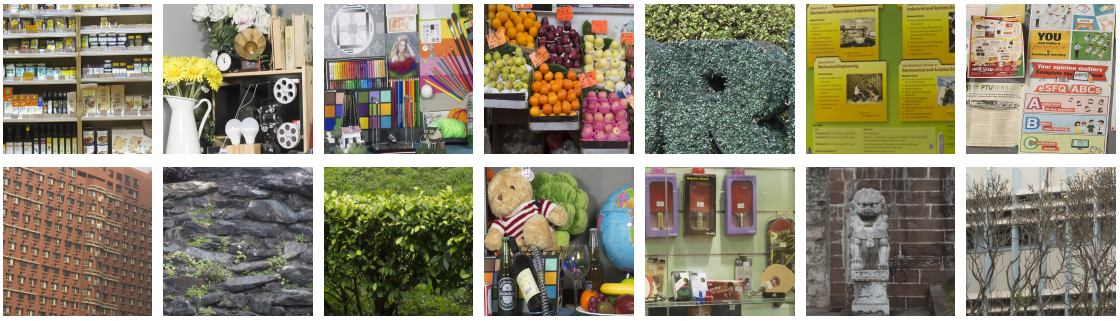


Figure 3.3: Example images in our ReDB dataset.

degrees of diffraction blur while images captured at $f/7.1$ are used to generate the groundtruth clear images. We choose $f/7.1$ since it presents the optimal resolution for the given lens and simultaneously provides large enough DoF.

We set the camera to aperture priority mode. The focal length is fixed as 35mm for both cameras. Small ISO (e.g., 100) is used to alleviate noise. The focus, white balance and exposure are set to automatic mode. To minimize the camera movement, we use a tripod to stabilize camera and use a remote shutter release to avoid camera shake when pressing the shutter button. We also use the mirror up (MUP) mode in DSLR to reduce tiny camera movement when the mirror is raised.

Scenes having abundant textures are preferred for our dataset. To avoid object motion in sequential image capturing, we capture static indoor and outdoor scenes. For each scene, we first shoot using $f/7.1$ aperture, then keep the camera unmoved and manually change the aperture to $f/14$, $f/18$, and $f/22$ to capture the diffraction blurred images. After collection, images containing local motion or illuminant changes are abandoned. In total, 333 scenes are captured, including 167 from the Nikon D810 and 166 from the Canon 5D3. The images are taken in several cities, covering various indoor and outdoor scenes and illuminations. The information about our dataset and employed camera devices are summarized in Table 3.1, and several examples from the dataset are shown in Fig. 3.3.

3.4.2 Image Pair Registration

The image pairs collected are not pixel-wise aligned since images taken at different apertures have different DoF. Besides, color and spatial misalignment also exist due to the changing of aperture. Image pair registration is necessary to obtain pixel-wise aligned image pairs for modeling training. Our three-step registration procedure is illustrated in Fig. 3.4.

In-focus region cropping. A wider aperture has shallower DoF compared with smaller apertures, leading to depth blur in out-of-focus region. Benefiting from the fact that the current commonly-used DSLR cameras have very high-resolution, we simply crop the in-focus region of images taken at $f/7.1$ aperture as the clear ground truth. The same area of images taken at other smaller apertures are cropped to construct the image pairs. After cropping, most images still have a resolution higher than 1000×2000.

It is worth mentioning that cropping the in-focus regions to construct image pairs does not affect the generalization capability of diffraction blur removal models trained on our dataset, since diffraction blur only appears when using small aperture which has sufficiently large DoF to avoid out-of-focus blur.

Color matching. The different apertures used in image pairs capturing result in different amount of light reaching the sensor. Though the auto exposure mode in DSLR would automatically adjust shutter speed to ensure normal exposure, the exposure and color of an image pair are not strictly matched. Considering that the color change among image pairs is global, we apply linear transform to the diffraction blurred images to transfer the channel-wise mean vector and covariance matrix to coincide with those of clear images. The linear transformation matrix is derived according to the 3D color matching algorithm in [61].

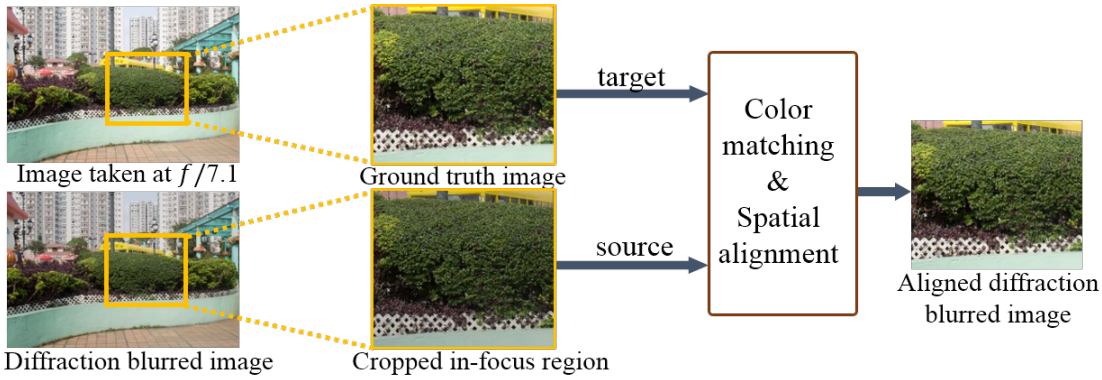


Figure 3.4: The image registration procedure to obtain aligned image pairs.

Spatial alignment. Despite the usage of tripod, MUP mode and remote shutter, image pairs may not be precisely aligned due to the camera movement when adjusting the aperture or slight shocking of soft ground. Fortunately, the spatial shift among image pairs is generally global and not severe in our dataset. We simply use the SURF features [12] based image registration algorithm, which is robust to rotation and illuminant change, to obtain spatially aligned image pairs.

3.5 Learn to Remove Diffraction Blur

The constructed ReDB dataset enables us to train a deep CNN model for diffraction blur removal. However, directly learning the transform from severely blurred images to clear ones is a difficult ill-posed task. We propose a progressive learning architecture to progressively recover more details with the guidance of images taken at increasingly larger apertures. We also design a new loss function to enhance textures and details in real-world photography.

3.5.1 Progressive Learning Loss

The overview of our progressive learning architecture is shown in Fig. 5.1. Our diffraction removal model contains several stages of convolutions to progressively recover image details. Each stage outputs an intermediate result, which is supervised

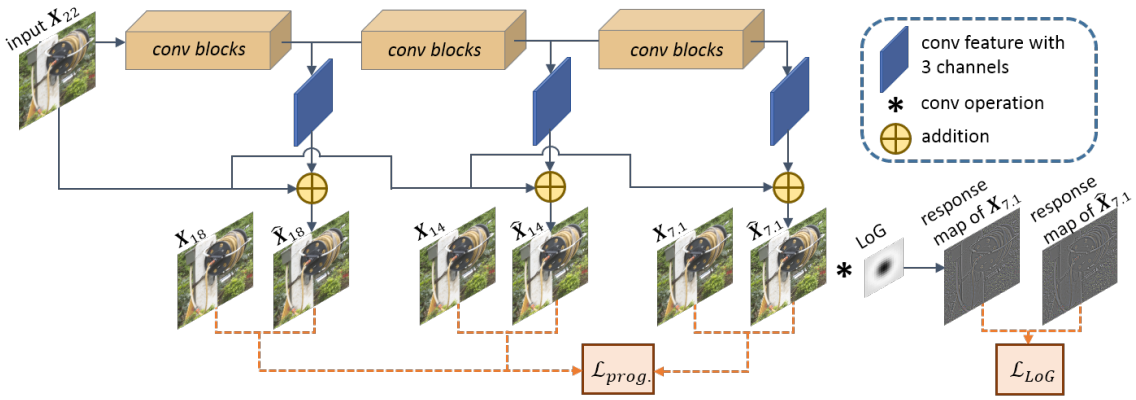


Figure 3.5: Overview of the proposed progressive learning architecture for diffraction blur removal. Our model outputs several intermediate results as well as a LoG response map, which are used as supervisions to progressively recover image details.

by the aligned image captured at a larger aperture. We consider two types of designs for the basic convolutional blocks in our model: a lightweight version following the VDSR [70] and a deeper version following the RCAN [141]. More details will be discussed in the experimental section.

Let f_i represent the feature extractor at the i-th stage. Denote the diffraction blurred images captured at aperture $f/22$, $f/18$, $f/14$, $f/7.1$ of the same scene by $\mathbf{X}_{22}$, $\mathbf{X}_{18}$, $\mathbf{X}_{14}$ and $\mathbf{X}_{7.1}$, respectively, where $\mathbf{X}_{7.1}$ is taken as the ground truth. Given the blurred input image $\mathbf{X}_{22}$, the progressive diffraction removal process is as follows:

$$\begin{aligned} \mathbf{F}_1 &= f_1(\mathbf{X}_{22}), \quad \hat{\mathbf{X}}_{18} = \mathbf{X}_{22} + \mathbf{w}_1 * \mathbf{F}_1; \\ \mathbf{F}_2 &= f_2(\mathbf{F}_1), \quad \hat{\mathbf{X}}_{14} = \mathbf{X}_{22} + \mathbf{w}_2 * \mathbf{F}_2; \\ \mathbf{F}_3 &= f_3(\mathbf{F}_2), \quad \hat{\mathbf{X}}_{7.1} = \mathbf{X}_{22} + \mathbf{w}_3 * \mathbf{F}_3; \end{aligned} \tag{3.3}$$

where $\mathbf{F}_i$ is the extracted feature at the i-th stage, $\hat{\mathbf{X}}_{18}$, $\hat{\mathbf{X}}_{14}$ and $\hat{\mathbf{X}}_{7.1}$ are the predicted images at aperture $f/18$, $f/14$ and $f/7.1$, respectively, $\mathbf{w}_i$ denotes the additional convolution layer at the i-th stage to transform the high-dimensional feature map $\mathbf{F}_i$ into an image, and $*$ represents the convolution operation.

The loss function of our progressive learning is:

$$\mathcal{L}_{prog} = \lambda_{s1}||\hat{\mathbf{X}}_{18} - \mathbf{X}_{18}||_2^2 + \lambda_{s2}||\hat{\mathbf{X}}_{14} - \mathbf{X}_{14}||_2^2 + ||\hat{\mathbf{X}}_{7.1} - \mathbf{X}_{7.1}||_2^2, \tag{3.4}$$

where λ_{s1} and λ_{s2} are two constant parameters, which are fixed as 0.01 and 0.1, respectively. We choose smaller weights for the losses at earlier stages since the corresponding gradients have larger influence to the earlier convolutional blocks.

Now we have introduced the three-stage progressive learning process for images captured at aperture $f/22$. Regarding images captured at aperture $f/18$, the process naturally degrades to two stages, with only the $\mathbf{X}_{14}$ as the intermediate guidance. The loss function consequently consists of the last two terms in Eq. 4.4, and the associated constant parameter is fixed at 0.1. Regarding aperture $f/14$, since the diffraction blur is not as severe as that at aperture $f/22$ and $f/18$, we thus train a model directly mapping from input to ground truth. The loss function is defined as the L_2-norm between network outputs and ground truth labels.

3.5.2 LoG based Loss

Diffraction blur removal aims at recovering fine-grained edges and textures whereas MSE loss gives equal emphasis on both image smooth content and edges. The Laplacian of Gaussian (LoG) [88] filter is a simple and computationally efficient edge detector. It can effectively detect edges with reduced sensitivities to noise, since in practical image capturing the camera sensor receives different photons at different time and there exists certain inevitable random noise. Moreover, early psychophysical experiments [37, 92] have shown that LoG filter matches well with the discriminability in human vision system. We thus employ the LoG filter as a robust edge detector and design a LoG based loss function to further enhance the details.

The LoG filter is designed as the Laplacian operator following a Gaussian smooth filter. The embedded Gaussian kernel can effectively suppress the small noise and

the Laplacian kernel detects the edges. As shown in Fig. 5.1, we apply the LoG filter to the output image and put supervision on the response maps to recover more textures. The LoG filter has explicit numerical form when the kernel size and standard deviation are given. In this work, we employ a 7×7 LoG filter with standard deviation $\sigma = 0.6$. Denote by $\mathbf{k}$ the LoG filter. The LoG loss is defined as:

$$\mathcal{L}_{LoG} = ||\mathbf{k} * \hat{\mathbf{X}}_{7.1} - \mathbf{k} * \mathbf{X}_{7.1}||_2^2, \tag{3.5}$$

where $\hat{\mathbf{X}}_{7.1}$ and $\mathbf{X}_{7.1}$ are the output image and ground truth image, respectively.

The final loss function in our method is:

$$\mathcal{L} = \mathcal{L}_{prog} + \lambda_{LoG} * \mathcal{L}_{LoG}, \tag{3.6}$$

where λ_{LoG} is a constant weight and we set $\lambda_{LoG} = 0.5$ in all experiments.

3.6 Experiments

3.6.1 Implementing Details

As mentioned in Sec. 3.5.1, we train two versions of our progressive diffraction removal (PDR) model. The lightweight version (denoted by PDR_L) uses the convolutional blocks in VDSR [70] which is a representative architecture in image restoration area. The deep model (denoted by PDR_D) employs the convolutional blocks of RCAN [141] which is proved to be effective for training very deep networks. When processing images captured at $f/22$, the three-stage PDR_L and PDR_D contain $\{5, 5, 10\}$ and $\{30, 30, 40\}$ convolutional layers for each stage, respectively. Regarding images captured at $f/18$, PDR_L and PDR_D have two equally partitioned stages with 10 and 50 convolutional layers for each stage, respectively. As for images captured at $f/14$, PDR_L and PDR_D have only one stage with 20 and 100 convolutional layers, respectively.

Since the diffraction degree is related to both the aperture and sensor pixel size, we trained an independent model for each aperture of each camera. Both the Nikon and Canon datasets were randomly split into 140 scenes for training and the remaining scenes were used for testing. In the training stage, images were cropped into $128 \times 128 \times 3$ patches. Left-right and up-down flips were used for data augmentation. The network parameters were initialized using the Xavier initializer [48]. The Adam optimizer [72] with the default parameter setting ($\beta_1 = 0.9$, $\beta_2 = 0.999$) was used to optimize our models for 100 epochs. The learning rate was fixed as $1e^{-4}$ for all models. The batch size was set as 16 and 8 for training PDR_L and PDR_D, respectively. PSNR and SSIM [120] are employed to evaluate performance.

3.6.2 Ablation Study

In this section, we conduct ablation study on the two major designs of our diffraction removal model: the progressive learning strategy and the LoG based loss function for detail enhancement. The lightweight model PDR_L was employed to evaluate the effectiveness of these two designs. Four variants of models are compared: a baseline model using the VDSR architecture and the MSE loss, our progressive model using the MSE loss, the baseline model plus our LoG based loss and our progressive model plus the LoG based loss. All the hyper parameters were kept the same for fair comparison. The PSNR and SSIM metrics obtained by all variant methods are listed in Table 5.1 and the visual quality comparison is shown in Fig. 5.3.

From Table 5.1, one can observe that employing the progressive architecture achieves consistently better PSNR (more than 0.12 dB in all cases) compared with the baseline model. Engaging the LoG based loss improves the PSNR by 0.13 dB on average among all cases over the baseline. Combining both the progressive method and LoG based loss achieves the best performance in most cases, with an average improvement of 0.25 dB among all cases. The visual comparison can be observed

Table 3.2: PSNR and SSIM results of variants of our proposed network. The best in each column is shown in red. For aperture $f/14$ we train a model directly mapping from input to ground truth, thus the progressive learning is not applicable (N.A.).

Method	Nikon D810			Canon 5D3		
	$f/22$	$f/18$	$f/14$	$f/22$	$f/18$	$f/14$
	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
MSE	34.40/0.9740	35.78/0.9802	37.25/0.9854	35.37/0.9787	36.78/0.9838	37.93/0.9871
MSE+Prog.	34.52/0.9749	35.92/0.9815	N.A./N.A.	35.50/0.9798	36.91/0.9837	N.A./N.A.
MSE+LoG	34.53/0.9750	35.93/0.9818	37.40/0.9861	35.51/0.9800	36.93/0.9843	38.06/0.9875
MSE+Prog.+LoG	34.65/0.9755	36.04/0.9817	N.A./N.A.	35.62/0.9801	37.05/0.9846	N.A./N.A.

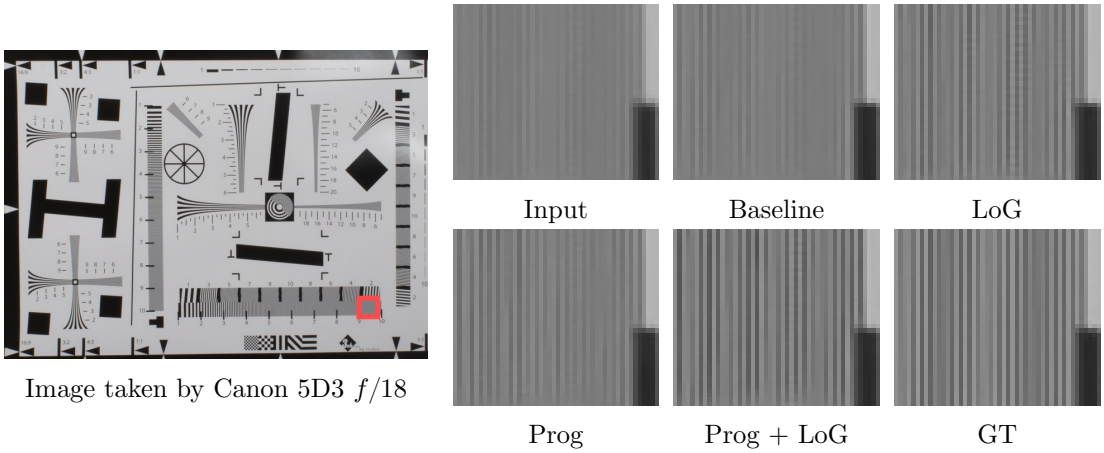


Image taken by Canon 5D3 $f/18$

Figure 3.6: Visual result comparison of variants of our model. The input image is taken by Canon 5D3 at $f/18$. One can see that the progressive learning method and the LoG based loss help recover more details.

from Fig. 5.3. The result obtained by the baseline model is still blurry. Employing either the progressive learning or the LoG loss can help recover more textures, and utilizing both of them achieves the best visual quality. Both the quantitative metrics and the visual quality comparison validate that our progressive learning method and LoG based loss function can help recover more details from diffraction blur.

3.6.3 Comparison with Other Methods

In this section we compare the proposed diffraction blur removal models with other image deblurring methods. The existing image deblurring methods can be generally divided into two categories: the traditional prior based methods and the learning based methods. Regarding the prior based methods, we choose Xu *et al.* [128], Xu *et al.* [127], GST [144], DCP [94], ECP [130] for comparison. The sharpening results obtained by the Photoshop USM (denoted as PS USM) are also included for comparison. As for the learning based methods, we choose two representatives: Sun *et al.* [107] and SRN [110]. In addition, we compare with two representative super resolution models VDSR [70] and RCAN [141] which can also enhance image details. For all learning based methods, we compare with both their original version provided by the authors (if applicable) and the retrained version on our ReDB dataset (denoted using superscript $*$). We did not retrain Sun *et al.* [107] since the training code is unavailable. The PSNR and SSIM results of all competing methods are listed in Table 3.3.

As can be seen, most traditional prior based deblurring methods perform poorly on the diffraction blur removal task, with a gap of about 3 dB compared with our PDR_L model. This is because most of these methods assume the blur kernel to be uniform while the blur kernel is non-uniform in our ReDB dataset. The Photoshop sharpening algorithm is not qualified on removing diffraction blur, either. As for the learning based methods, we can see that directly applying their pre-trained models on our ReDB dataset obtain unsatisfied results. The PSNR results are at least 2.5 dB inferior to our PDR_L model. These results again validate the difference between diffraction blur removal and general image deblurring or super resolution. The re-trained version of SRN [110] and VDSR [70] obtain much better performance than their original version, which validates the effectiveness and value of our constructed

Table 3.3: Comparison of various methods on ReDB dataset. The best, second and third are shown in red, blue and green respectively. The superscript $*$ denotes the re-trained model on our ReDB dataset.

Dataset	Nikon D810						Canon 5D3					
Aperture Size	f/22		f/18		f/14		f/22		f/18		f/14	
Metrics	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Xu *et al.* [127]	30.56	0.8921	31.92	0.9164	32.69	0.9256	30.78	0.8709	30.71	0.8615	31.72	0.8839
GST [144]	30.97	0.9032	32.67	0.9333	33.67	0.9471	33.05	0.9329	33.92	0.9458	33.94	0.9513
DCP [94]	31.44	0.9228	32.82	0.9453	32.37	0.9452	32.10	0.9241	32.00	0.9269	31.17	0.9238
Xu *et al.* [128]	31.87	0.9248	32.78	0.9369	32.41	0.9398	31.55	0.8923	31.42	0.8992	30.46	0.8869
ECP [130]	31.93	0.9224	33.37	0.9470	33.49	0.9542	33.51	0.9402	33.88	0.9480	33.04	0.9478
PS USM	31.27	0.9080	32.76	0.9384	33.36	0.9550	33.10	0.9312	34.80	0.9515	35.38	0.9640
Sun *et al.* [107]	27.58	0.8171	28.03	0.8340	28.49	0.8503	28.76	0.8420	29.35	0.8582	29.86	0.8721
SRN [110]	30.04	0.8920	31.49	0.9209	32.99	0.9404	31.15	0.9103	32.20	0.9263	33.06	0.9369
VDSR [70]	32.07	0.9247	33.47	0.9470	33.40	0.9503	33.30	0.9401	33.75	0.9475	33.15	0.9464
SRN*	34.23	0.9729	35.50	0.9799	36.76	0.9837	34.99	0.9757	36.41	0.9832	37.62	0.9864
VDSR*	34.45	0.9745	35.90	0.9808	37.31	0.9857	35.44	0.9790	36.86	0.9841	37.93	0.9871
RCAN* [141]	34.85	0.9760	36.21	0.9821	37.56	0.9865	35.70	0.9795	37.23	0.9842	38.21	0.9873
PDR_L	34.65	0.9755	36.04	0.9817	37.40	0.9861	35.62	0.9801	37.05	0.9846	38.06	0.9875
PDR_D	34.97	0.9765	36.35	0.9831	37.67	0.9868	35.83	0.9802	37.32	0.9855	38.32	0.9881

ReDB dataset. However, they are still inferior to our PDR_L, which demonstrates the effectiveness of our designs for removing diffraction blur. Compared with PDR_L, PDR_D further improves the PSNR by at least 0.2 dB in all cases, which indicates that employing deeper architecture can further improve the performance. In addition, PDR_D also outperforms its baseline variant, i.e., the retrained version of RCAN in all cases. This again proves the effectiveness of our model designs for diffraction removal even under much deeper architecture.

The visual results of all competing algorithms on several examples are shown in Fig. 3.5. As can be seen, many prior based methods, e.g., DCP [94], ECP [130] and GST [144], can only mildly recover the details but result in obvious artifacts. Xu *et al.*'s method [128] over-sharpens edges with obvious ringing artifacts. Some

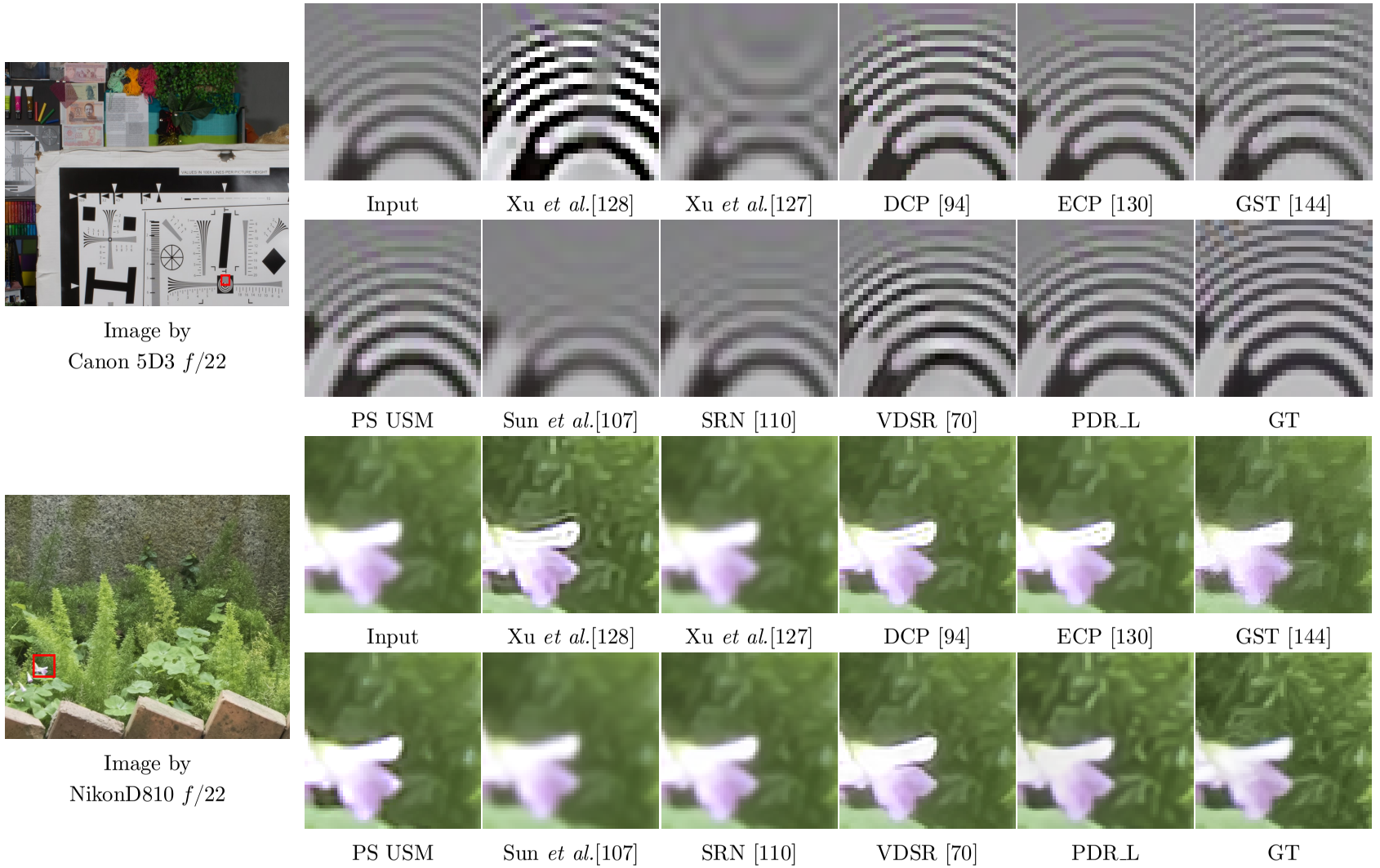


Table 3.4: Qualitative comparison of visual results obtained by different methods on ReDB dataset. Our method can effectively recover more details with little artifact.

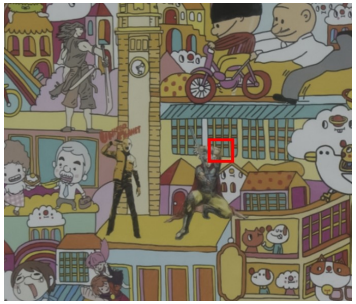
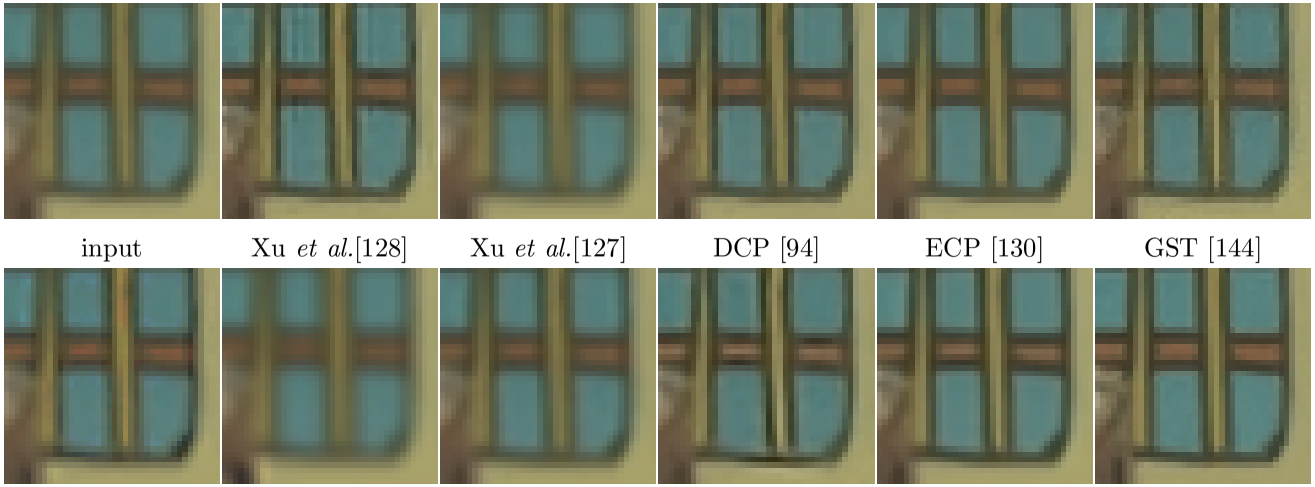


Image by
NikonD810 $f/18$



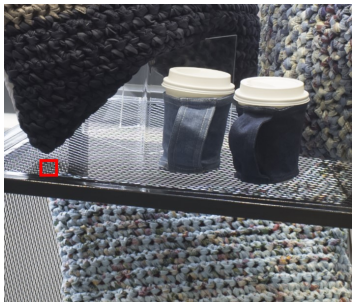


Image by
Canon5D3 $f/18$

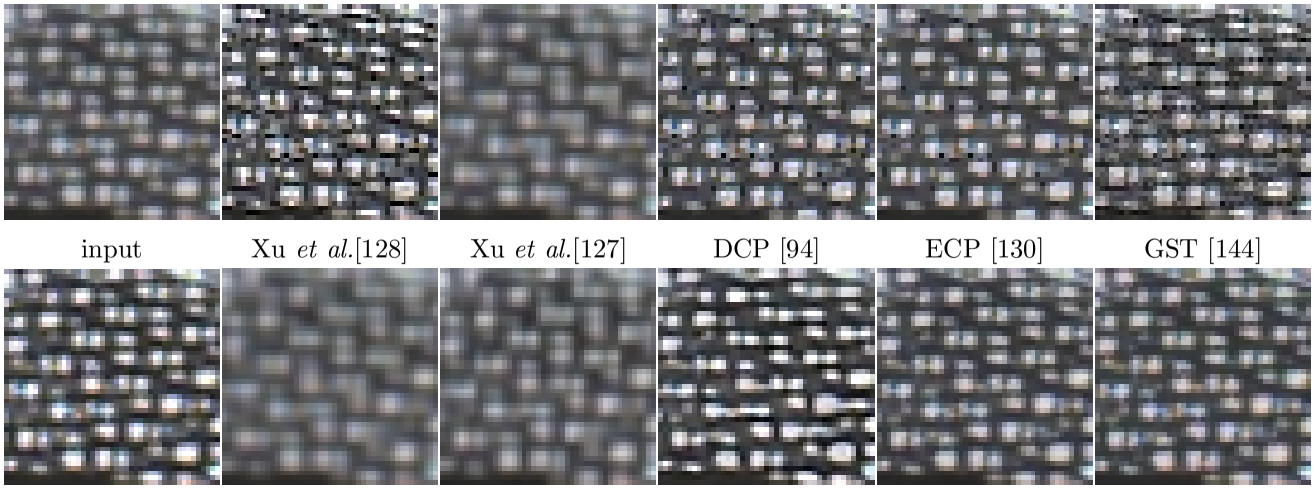


Table 3.5: More qualitative comparison of visual results obtained by different methods on ReDB dataset.

methods including the two learning based deblurring methods [127, 107, 110] even makes the images more blurry. The photoshop USM operator can only moderately enhance the details. The results of VDSR are not stable. In contrast, our PDR_L model can effectively recover the textures without generating artifacts, presenting the best visual quality.

3.6.4 Cross Camera Evaluations

In this section we conduct cross camera evaluation to study the generalization ability of our model. As mentioned before, the degree of diffraction blur is not only related to the aperture size but also affected by the sensor pixel size. As shown in Table 3.1, the two cameras we employed have different pixel size. To provide a more explicit index indicating the degrees of diffraction blur, we define a new index $\delta = N/d$ as the ratio of aperture f-number N ($N = 22$ for aperture $f/22$) to sensor pixel size d. Larger δ implies severer diffraction blur.

Canon 5D3 vs. Nikon D810. We then conduct cross camera evaluations between the Canon 5D3 and Nikon D810 cameras using the PDR_L model. Specifically, on each aperture of one camera, we evaluate the three models trained on the other camera and list the cross camera results in Table 3.6. The in-camera results on each aperture and the diffraction blur degree index δ are also provided for reference. We can observe that when diffraction blur degree δ of two datasets are close, the model trained on one camera obtains better results when applying to the other. For example, when testing on $f/22$ on Canon 5D3 ($\delta = 3.49$), the PDR_L model trained for aperture $f/18$ of Nikon ($\delta = 3.67$), exhibis better metrics than PDR_L trained for aperture $f/22$ of Nikon ($\delta = 4.49$). We also observed that models trained on Canon dataset exhibit better cross camera performances compared with models trained on Nikon. This probably because the image pairs in Canon dataset are of relatively larger size than those in Nikon dataset. The more training patches leads to better

Table 3.6: Quantitative results of our PDR_L model on cross camera experiments. The in-camera results are shown in red for reference and the best cross camera results are shown in blue. The number in bracket is the diffraction blur index δ.

Train set \ Test set		Canon 5D3			Nikon D810		
		$f/22$ (3.49)	$f/18$ (2.86)	$f/14$ (2.22)	$f/22$ (4.49)	$f/18$ (3.67)	$f/14$ (2.86)
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Canon 5D3	$f/22$ (3.49)	35.62/0.9801	–/–	–/–	33.69/0.9700	35.86/0.9816	36.76/0.9851
	$f/18$ (2.86)	–/–	37.05/0.9846	–/–	32.78/0.9628	35.75/0.9808	37.31/0.9859
	$f/14$ (2.22)	–/–	–/–	38.06/0.9875	32.06/0.9564	34.95/0.9768	37.28/0.9859
Nikon D810	$f/22$ (4.49)	34.79/0.9749	35.51/0.9752	33.36/0.9682	34.65/0.9755	–/–	–/–
	$f/18$ (3.67)	35.03/0.9776	36.25/0.9816	37.28/0.9853	–/–	36.04/0.9817	–/–
	$f/14$ (2.86)	34.20/0.9710	36.48/0.9824	37.53/0.9857	–/–	–/–	37.40/0.9861

generalized representations of PDR_L model trained on Canon dataset.

Other cameras. To further validate the cross camera tendency, we also evaluate the trained models on images taken by other cameras. The visual results on five images out of our ReDB dataset are shown in Fig. 3.7. The five images were taken by Sony A77 $f/16$, Sony A77 $f/14$, Nikon D5200 $f/14$, Olympus E-520 $f/14$ and Fujifilm X-T20 $f/18$ respectively. The sensor pixel size of Sony A77, Nikon D5200 and Fijufilm X-T20 is $3.9um$, and is $4.7um$ for Olympus E-520. And the diffraction blur degree δ of the five images are 4.10, 3.89, 3.89, 2.98 and 4.61 respectively. We apply the PDR_L model trained on Nikon D810 with $\delta = 4.49, 3.67, 3.67, 2.86, 4.49$ respectively to the five testing images. One can see that our model can effectively recover details from the blurry input and present pleasant visual quality, which demonstrates that our model is still effective when applied to another camera.

3.7 Conclusions

In this chapter, we studied the practical diffraction blur removal problem, for the first time, using a learning based method. We analyzed the special properties of diffraction

Figure 3.7: Visual results obtained by applying PDR_L model trained on our Nikon D810 dataset, to images out of our ReDB dataset. (a) and (c) are the zoomed input patches. (b) and (d) are the results obtained by diffraction removal model.

blur and clarified its difference from other types of image blurring problems. A real-world diffraction blur dataset with aligned image pairs was constructed for training and evaluating diffraction blur removal models. A progressive learning method and a robust loss function were designed to train a diffraction blur removal model, which achieved significantly better performance than existing image deblurring methods on both quantitative metrics and visual quality. We also studied the generalization capability of trained model and defined a diffraction blur degree index for use in practical applications. In the future, we will extend the database and study training one single model to handle various degrees of diffraction blur to further improve the generalization performance.

Chapter 4

Degradation Model Learning for Real-World Single Image Super-resolution

In this chapter, we focus on the task of real-world single image super-resolution (SISR). Despite the fast growth of deep learning based SISR methods, the real-world SISR task remains challenging. It is well-known that the SISR models trained on those synthetic datasets, where a low-resolution (LR) image is generated by applying a simple degradation operator (e.g., bicubic downsampling) to its high-resolution (HR) counterpart, have limited generalization capability on real-world LR images, whose degradation process is much more complex. Several real-world SISR datasets have been constructed to reduce this gap; however, their scale is relatively small due to laborious and costly data collection process. To remedy this issue, we propose to learn a realistic degradation model from the existing real-world datasets, and use the learned degradation model to synthesize realistic HR-LR image pairs. Specifically, we learn a group of basis degradation kernels, and simultaneously learn a weight prediction network to predict the pixel-wise spatially variant degradation kernel as the weighted combination of the basis kernels. With the learned degradation model, a large number of realistic HR-LR pairs can be easily generated to train a more robust

SISR model. Extensive experiments are performed to quantitatively and qualitatively validate the proposed degradation learning method and its effectiveness in improving the generalization performance of SISR models in practical scenarios.

4.1 Introduction

4.1.1 Single Image Super-Resolution

Single image super-resolution (SISR) aims to recover a high-resolution (HR) image from its low-resolution (LR) observation, which is a highly valuable technique for improving the resolution and quality of digital photography. As a typical ill-posed inverse problem, SISR has been widely studied during the past decades [131, 95, 133, 113, 20, 83], yet it is still a challenging and active research topic. The traditional methods generally utilize powerful image priors [85, 34, 119, 57, 132, 36] for SISR, and have made remarkable progresses. However these handcrafted image priors are limited in representing the complex image textures.

Benefitting from the rapid development and great success of deep convolutional neural networks (CNNs) [50], recently SISR has witnessed significant progresses by employing deep CNNs [33, 70, 136, 105, 109, 81, 30, 142, 78, 141, 80]. Most of the existing CNN based SISR models are trained on synthetic HR-LR image pairs, which are generated by applying a simple degradation model (e.g., bicubic downsampling) to the HR images [33, 70, 109, 81, 142, 141, 80]. However, the authentic HR to LR image degradation process is much more complicated than these simple uniform downsample operators. As a result, the SISR networks trained on such synthetic datasets have low generalization capability to real-world LR images, largely limiting their value in practical applications.

Efforts have been made to address the generalization problem of SISR models [136, 139, 25, 21]. Zhang *et al.* [136] proposed to use multiple Gaussian kernels to-

gether with additive white Gaussian noise to increase the diversity of HR-LR pairs, yet the selection and combination of these kernels are very sensitive. Very recently, researchers have started to construct real-world datasets by using digital cameras to capture images of the same scene under different focal lengths [139, 25, 21]. Particularly, Cai *et al.* [21] carefully designed a registration algorithm to obtain pixel-wise aligned HR-LR image pairs. The so-called RealSR dataset enables supervised learning of SISR models, and the learned models demonstrate better performance than previous ones on real-world scenarios. However, constructing such datasets of real-world HR-LR pairs is laborious and costly, and the existing datasets of this kind [139, 25, 21] are all limited in number of image pairs, diversity of scenes and illuminating conditions. For example, the RealSR dataset contains only 559 scenes in total, limiting the generalization capability of trained SISR models to a wider range of scenarios.

While constructing real-world datasets of HR-LR image pairs, researchers have also proposed to learn the image degradation process from unpaired HR and LR images, and use the learned degradation model to generate HR-LR image pairs for SISR model learning [43, 19, 59, 82]. All these methods employ the Generative Adversarial Network (GAN) [51] to learn the degradation process by differentiating the distribution between generated LR and real LR images. Unfortunately, training such a GAN with unpaired data is very difficult and may not converge to the desired result. Moreover, using a network to model the degradation from HR to LR images makes it hard to interpret the degradation process, ignoring some prior knowledge on the image formation.

4.1.2 Motivation

In this chapter, we model the image degradation process by using spatially variant degradation kernels instead of a network, and propose to learn this model from

the HR-LR image pairs in the RealSR dataset instead of the unpaired HR and LR images. It is widely agreed in literature [95, 131, 102, 136, 133] that the LR image formation process can be formulated as first blurring the HR image with a degradation kernel, followed by downsampling and noise addition, while in real scenarios the degradation kernel is spatially variant, relating to the depth and local content in the scene. Clearly, the pixel-wise degradation kernels are the key to model the degradation process. One may propose to learn a network to directly map the HR image to LR image, or propose to learn a network to directly predict the pixel-wise degradation kernel. However, the learning space of those two proposals can be too big for modeling the degradation process, while they ignore the common knowledge of image degradation. Considering the fact that blurring kernels in an optical imaging system can be generally described as bell-shaped smooth functions [23], we argue that the plausible degradation kernels distribute in a small subspace, which can be approximated as a linear combination of a group of basis kernels. Therefore, we propose to learn a group of basis kernels as well as a weight prediction network to predict the combination coefficients at each pixel.

An end-to-end learning scheme is designed to learn the basis kernels and the weight prediction network from the RealSR dataset [21]. Once learned, our degradation model takes an HR image as input, predicts the spatially variant kernels at each location, and outputs the degraded LR image. In this way, we can easily generate a large amount of realistic HR-LR image pairs using the HR images on hand. Finally, we can train SISR models by using these synthetic yet realistic HR-LR pairs. Experimental results show that the trained SISR models achieve better generalization performance than the models trained only on the RealSR dataset, owing to the enlarged training data of realistic HR-LR image pairs.

4.2 Literature Review

4.2.1 Single Image Super-resolution

Single image super-resolution (SISR) is an active topic in low-level vision, and a plenty of works have been proposed in the past decades, including interpolation-based [137], model-based [36, 57] and learning-based methods [33, 70, 105, 109, 81, 30, 142, 141, 80]. Traditional methods are usually limited in representing the complex image local structures, while the recently developed deep CNN have shown great advantages in image structure representation and consequently improved much the SISR performance [33, 70, 77, 81, 80, 141]. For example, Kim *et al.* [70] employed the residual learning strategy to design the VDSR model with 20 convolutional layers. Liu *et al.* [81] proposed to utilize contextual information by exploiting the image non-locally correlation. Zhang *et al.* [141] proposed a very deep CNN with over 400 layers, and improved much the SISR performance. Despite the great success, most of the CNN based SISR models are trained on synthetic datasets, where the LR images are generated by applying simple operators such as bicubic downsampling to the HR images [33, 70, 105, 109, 81, 30, 142, 141, 80]. Unfortunately, the real-world image degradation process is far more complex than bicubic downsampling. Such a gap between synthetic data and real data makes the trained deep SISR models hardly be generalized to real-world LR images.

4.2.2 Real-world SISR

To solve the problem of real-world SISR, one intuitive way is to use a more complex degradation process to simulate LR images. Zhang *et al.* [136] proposed to use multiple Gaussian kernels with additive white Gaussian noise to simulate LR images, whereas the selection of suitable kernels is difficult and ad hoc for practical applications. Another recently popular solution is to employ the generative adversarial

network (GAN) [51] with unsupervised learning. E.g., SRGAN [78] is proposed to utilize adversarial loss to improve the perceptual quality of images. While the GAN-based methods show some interesting results on SISR, their results are not stable and often exhibit some unnatural visual artifacts.

Instead of simulating HR-LR image pairs, recently efforts have been devoted to construct real-world SISR datasets. Qu *et al.* [100] proposed to use a beam splitter to acquire paired HR-LR images. Kóhler *et al.* [74] used hardware binning on camera sensor to generate LR images. However, these two datasets contain very limited scenes, 31 in [100] and 14 in [74]. Very recently, DSLR cameras have been used to construct real-world SISR datasets by capturing the same scene under different focal lengths. Chen *et al.* [25] collected 100 image pairs of printed postcards. Zhang *et al.* [139] constructed the SR-RGB dataset with 500 scenes, whereas the image pairs are not strictly aligned. To enable pairwise learning, an image registration algorithm is proposed in [21] to carefully handle the misalignment between HR and LR images caused in the data collection process. The so-called RealSR dataset contains a set of aligned real-world HR-LR image pairs, which allow direct pairwise training of SISR models. However, the collection and processing of such a dataset is laborious and costly, and the scale and diversity of RealSR dataset is relatively limited (559 scenes in total).

4.3 Degradation Model Learning for SISR

In this section, we first formulate the LR image degradation model based on the real-world LR image formation process. We then present how to learn the pixel-wise degradation models. Finally, we present how to use the learned degradation models to generate realistic HR-LR datasets for training real-world SISR models.

4.3.1 Formulation of Image Degradation Model

Denote by $\mathbf{I}^H$ an HR image and by $\mathbf{I}^L$ its LR counterpart. In literature [95, 131, 102, 136, 133], the image degradation from an HR image to an LR image can be generally represented as

$$\mathbf{I}^L = (\mathbf{I}^H * \mathbf{k})\downarrow_d + \mathbf{v}, \tag{4.1}$$

where "$*$" is the convolution operator, $\mathbf{k}$ is the degradation kernel, $\downarrow_d$ is the down-sampling operator, and $\mathbf{v}$ is the random observation noise. The goal of SISR is to recover the underlying HR image $\mathbf{I}^H$ given its LR observation $\mathbf{I}^L$.

Most of existing SISR works [33, 70, 109, 81, 142, 141, 80] assumes that the degradation kernel $\mathbf{k}$ is uniform, i.e., spatially invariant, over the whole image. Particularly, they apply the bicubic kernel to HR images to simulate the HR-LR image pairs, and then use those pairs to train SISR models. Whereas in real-world SISR problems, the degradation kernel is much more complex, correlating with the depth and local content of the scene [21]. Therefore, the degradation kernel is typically non-uniform and spatially variant. At each location (i, j), the kernel may vary, and we use $\mathbf{k}_{i,j}$ to denote the per-pixel degradation kernel. The spatially variant image degradation from HR to LR can be formulated as:

$$\mathbf{I}^L(i, j) = \mathbf{I}^H_{i,j} \odot \mathbf{k}_{i,j} + \mathbf{v}(i, j), \tag{4.2}$$

where $\mathbf{I}^H_{i,j}$ denotes a local image window centered at (i, j) with the same size as kernel $\mathbf{k}_{i,j}$, and "$\odot$" is the inner product operator.

From Eq. (4.2), one can see that the key to model the real-world image degradation process is how to predict the pixel-wise degradation kernel $\mathbf{k}_{i,j}$. One intuitive idea is to learn a CNN from the available HR-LR pairs (e.g., the RealSR dataset [21]) to predict the kernel $\mathbf{k}_{i,j}$; however, the learning space of a CNN can be too big for the kernels and the network can be over-fitted by the limited training data. On

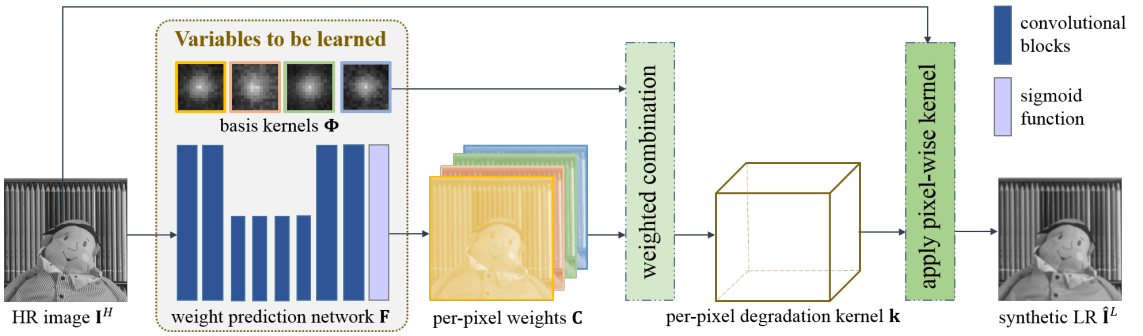


Figure 4.1: Overview of the proposed approach for degradation model learning. A group of basis kernels $\mathbf{\Phi}$ are learned together with a weight prediction network $\mathbf{F}$, which are used to generate the pixel-wise degradation kernels. The LR image is obtained by applying the pixel-wise degradation kernels to the HR image.

the other hand, the predicted kernel may have poor interpretability since they may not accord with our prior knowledge on the image degradation process (please refer to our ablation study in Sec. 4.4.3 for more discussions). It is commonly agreed that the degradation kernels in an optical imaging system can be generally described as bell-shaped smooth functions [23]. This means that the plausible degradation kernels are not arbitrary but actually fall into a small subspace, which can be spanned by a group of basis kernels. Denote by $\mathbf{\Phi} = \{\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_M\}$ the set of M basis kernels. We propose to approximate the pixel-wise degradation kernel $\mathbf{k}_{i,j}$ as a weighted combination of $\mathbf{\Phi}$ as follows:

$$\mathbf{k}_{i,j} \approx \sum_{m=1}^{M} \mathbf{C}_{i,j}(m)\boldsymbol{\phi}_m, \tag{4.3}$$

where $\boldsymbol{\phi}_m$ is the m^{th} basis kernel and $\mathbf{C}_{i,j}$ represents the combination weight vector at location (i, j). The above formulation constrains the kernels in a subspace which can be more easily learned, especially when the available training dataset (e.g., RealSR) is not very big.

4.3.2 Degradation Model Learning

From Eq. (4.3), one can see that the learning of pixel-wise kernels $\mathbf{k}_{i,j}$ is turned into the learning of basis kernels $\mathbf{\Phi}$ and the weight vectors $\mathbf{C}_{i,j}$. The basis kernels are global to all image regions, while the weights depend on the image local contents. We propose to use a network to predict the weights and learn it simultaneously with the basis kernels from some real-world HR-LR dataset.

Our degradation model learning (DML) approach is illustrated in Fig. 5.1. With the HR image $\mathbf{I}^H$ as input, a CNN $\mathbf{F}$ with parameters $\mathbf{\Theta}$ is learned to predict the weights, i.e., $\mathbf{C} = \mathbf{F}(\mathbf{I}^H|\mathbf{\Theta})$, where $\mathbf{C}$ is the set of weight vectors $\mathbf{C}_{i,j}$. The basis kernels $\boldsymbol{\phi}_m$ are also learned so that the kernels $\mathbf{k}_{i,j}$ can be predicted according to Eq. (4.3). The predicted degradation kernels are applied to the HR image $\mathbf{I}^H$ to output the predicted LR image, denoted by $\hat{\mathbf{I}}^L$. Suppose there are N pairs of HR-LR training images, the learning objective can be formulated as

$$\min_{\mathbf{\Phi},\mathbf{\Theta}} \sum_{n=1}^{N} ||\hat{\mathbf{I}}_n^L - \mathbf{I}_n^L||_2^2. \tag{4.4}$$

We learn the basis kernels $\mathbf{\Phi}$ and weight prediction network $\mathbf{F}$ in an end-to-end manner by using the RealSR dataset [21].

We design the weight prediction network $\mathbf{F}$ following an encoder-decoder structure. It takes an HR image as input and outputs a weight vector at each location. To embrace large receptive field, we use a max pooling layer for feature down-sampling, and employ the bilinear upsampling layer to increase the feature resolution and ensure pixel-wise outputs. Convolutional layer with filters of size 3×3 is used, and ReLU is used as the activation function. To output the per-pixel weights, we use sigmoid function after the last convolutional layer for normalization. The whole network can be easily optimized by the SGD or ADAM optimizer. Examples of the learned kernels, the visualization of the predicted weight maps and more discussions

will be provided in the ablation study (see Section 4.4.3).

4.3.3 SISR Model Learning

Once the basis kernels $\mathbf{\Phi}$ and the weight prediction network $\mathbf{F}$ are learned by using the DML approach presented in Section 4.3.2, we can use them to synthesize HR-LR image pairs by using a set of collected HR images as inputs. However, directly using the synthesized LR images to train SISR models is problematic. As described in Eqs. (4.1) and (4.2), the real-world LR images are usually corrupted by a certain amount of noise. However, the training objective in Eq. (4.4) encourages to generate a noise-free LR image since the random noise is hard to predict. If we use the synthesized clean LR images to train the SISR model and then apply the model to real-world noisy LR data, the noise will be exaggerated and lead to unpleasant visual artifacts.

To address this issue and further diminish the gap between synthetic and real LR images, we add random noise to the synthesized LR image $\hat{\mathbf{I}}_n^L$ according to the LR image formulation process described in Eq. (4.1). Without additional information on the imaging system (e.g., sensors, lens), we simply assume additive white Gaussian noise (AWGN) and empirically set the noise level as $\sigma = 5$.

Finally, we collect a set of high quality images as the HR set, and use the learned degradation model together with AWGN to generate synthetic yet realistic HR-LR image pairs. These image pairs are used to train the SISR model. In this chapter, we adopt two representative SISR network architectures, a lightweight network VDSR [70] and a deeper network RCAN [141], to validate the proposed DML method.

4.4 Experimental Results

4.4.1 Experiment setup

We carry out both quantitative and qualitative experiments to demonstrate the effectiveness of our proposed DML method for SISR model training. Considering that there are a few issues to be validated and explained, here we summarize how we set up the experiments for a better understanding of our work.

- In Section 4.4.2, we introduce the training dataset and the testing dataset in our experiments, as well as some implementation details of our algorithm.

- Section 4.4.3 conducts some ablation studies. First, we discuss the selection of the number of basis kernels in DML. Then we compare our DML with another two potential solutions to synthesize HR-LR pairs. One is to learn a CNN to directly map an HR image to an LR one, and another is to learn a CNN to predict the pixel-wise degradation kernel.

- In Section 4.4.4 we demonstrate that our DML can result in more robust real-world SISR performance. We first use the RealSR dataset [21], where aligned real-world HR-LR pairs are available so that PSNR/SSIM/LPIPS indices can be computed, to perform quantitative experiments. We then use other real-world data out of the training dataset to perform qualitative experiments, which are to demonstrate that our DML can improve the robustness and generalization performance of real-world SISR models.

4.4.2 Datasets and Implementation Details

Datasets. There are three types of datasets required to validate the performance of DML in degradation process learning and SISR model training.

- The first one is the RealSR [21] dataset (version 2), which contains aligned HR-LR image pairs of 559 scenes collected by two cameras with 3 zooming factors: $\times 2$, $\times 3$ and $\times 4$. We follow [21] to split the RealSR dataset into 459 scenes for training and the remaining 100 for testing. We use the training part of this dataset to train our degradation model by the method described in Section 4.3.2, and use the testing part to quantitatively evaluate the performance of DML and its application to real-world SISR.

- Once the degradation model is learned, we can apply it to an HR image dataset to generate synthetic HR-LR pairs. We construct an HR dataset by combining the Flickr2K dataset [80] and Internet images, containing 3150 images in total. The Flickr2k dataset has 2650 high quality images of various scenes, whose resolution is mostly 1500×2000. To diminish the effect of compression artifacts, we downsample those Flickr2k images by a factor of 2 after Gaussian smoothing (with scale $\sigma = 1$). We also download 500 raw images of 4K resolution from [121], and then apply the PhotoShop CameraRaw tool to them so that uncompressed high quality RGB images of 4K resolution are obtained.

- The third dataset is to validate the effectiveness of DML for real-world SISR. We use the SR-RGB dataset [139] which consists of real-world LR images and their unaligned HR counterparts obtained by optical zoom of DSLR. Since the HR and LR images are not aligned, the PSNR/SSIM/LPIPS measures can not be calculated but the HR images can be used as references for visual comparison.

Implemention Details. We set the size of basis kernels to be learned as 15×15 for all zooming scales $\times 2$, $\times 3$, and $\times 4$. The basis kernels are randomly initialized, and then normalized to have summation 1 for further updating. The weight prediction network is initialized using the Xavier initializer [48]. In the training of both DML

Table 4.1: Evaluation of the quality of generated LR images and super-resolved HR images by using the RealSR [21] dataset. The best and second results are highlighted in red and blue, respectively.

Method	Generated LR						Super-resolved HR					
	×2		×3		×4		×2		×3		×4	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
DML (N=4)	37.82	0.9862	36.46	0.9848	35.61	0.9840	33.23	0.9544	30.09	0.9150	28.50	0.8856
DML (N=8)	37.93	0.9864	36.54	0.9850	35.75	0.9842	33.32	0.9552	30.18	0.9157	28.60	0.8864
DML (N=16)	37.90	0.9863	36.51	0.9849	35.73	0.9841	33.28	0.9548	30.16	0.9153	28.58	0.8859
DirectNet	37.70	0.9864	36.33	0.9843	35.50	0.9838	33.13	0.9539	30.01	0.9144	28.42	0.8853
DirectKPN	37.77	0.9863	36.35	0.9844	35.56	0.9836	33.16	0.9545	30.06	0.9147	28.48	0.8860

and SISR networks, we convert the RGB images to YCbCr color space, and train or test on the Y channel. Images are cropped into 192 × 192 patches for training of all models. Left-right and up-down flips are used for data augmentation. The Adam optimizer [72] with the default parameter setting ($\beta_1 = 0.9$, $\beta_2 = 0.999$) is used as the optimizer. We train DML and SISR models using fixed learning rate of $1e^{-4}$ for 100K and 300K iterations, respectively. The batch size is set as 16 in DML training. As for SISR models, we adopt two representative network architectures: VDSR [70] and RCAN [141]. We implement RCAN with 100 convolutional layers. The batch size is set as 16 and 2, respectively, for training VDSR and RCAN models.

4.4.3 Ablation study

We conduct ablation studies to investigate the following two issues of DML: (1) selection of the number of basis kernels in DML; and (2) comparison of DML with the other two potential HR-LR pair synthesis approaches. We train our DML and its variants on the training set (459 image pairs) of RealSR [21], and use the testing set of RealSR to evaluate the quality of generated LR images and the quality of super-resolved HR images. PSNR and SSIM are used as the quantitative metrics.

Number of basis kernels. We first study the suitable number of basis kernels in

our DML. By using the training part of the RealSR dataset, we learn N=4, 8, 16 basis kernels and their associated weight predict networks. We then apply the learned models to the HR images in the testing part of the RealSR dataset to generate LR images. By comparing the synthesized and real LR images, we compute and list the PSNR/SSIM results in Table 4.1. One can see that by increasing the number from N=4 to N=8, better LR generation performances can be achieved, whereas the performance of using N=16 basis kernels is slightly worse than N=8. This means that the underlying degradation process can be well approximated by using N=8 basis kernels.

We visualize the learned 8 basis kernels for different zooming factors in Fig. 4.2. One can see that with the increase of zooming factor from 2 to 4, the kernels becomes more dispersed and complex, which are in accordance with our common knowledge of image degradation process. We also visualize the basis coefficients predicted by our weight prediction network in Fig. 4.3. One can see that the learned network can adaptively assign different weights to the kernels according to the scene content and image local structure to generate realistic LR images.

Since our final goal is to improve the SISR performance via DML, it is also necessary to test the effect of N on the final SISR results. We apply the learned DML models to our collected HR image dataset (see Sec. 4.2) to synthesize 3150 HR-LR images pairs, which are then used to train a VDSR super-resolution model. By applying the trained VDSR model to the LR images in the RealSR testing set, we compute the PSNR/SSIM indices of the super-resolved HR images. Table 4.1 lists the results. One can see that N=8 again achieves the best results for real-world SISR. Therefore, we set N=8 for DML in our experiments.

Comparison with other HR-LR pair synthesis strategies. Besides the proposed DML, there are two other intuitive strategies to synthesize HR-LR image pairs. One is to learn a CNN that directly maps an HR image to an LR one, denoted as

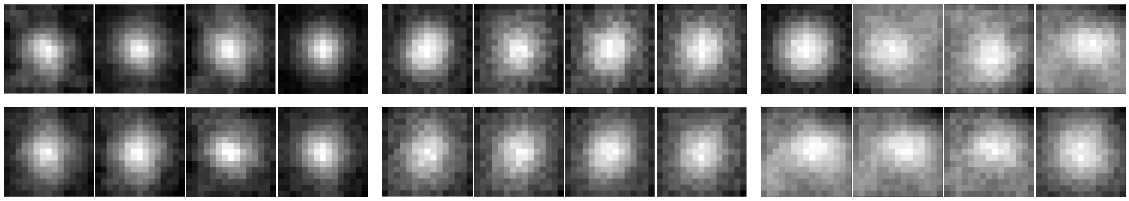


Figure 4.2: Visualization of the learned degradation basis kernels by our DML (N=8) model. The left, middle and right 4 columns represent the basis kernels for SR zooming factors $\times 2$, $\times 3$ and $\times 4$, respectively.

Figure 4.3: Visualization of the predicted combination weights of the basis kernels by our DML method for zooming factor $\times 2$. The leftmost image is the input HR image, and the right 8 images visualize the predicted weights corresponding to each basis kernels (refer to Fig. 4.2 for the 8 kernels). The brighter intensity denotes larger weight. One can see that our weight prediction network can adaptively assign different weights according to the scene content and local structures.

DirectNet, and the other is to learn a kernel prediction network [91] to predict the degradation kernel, denoted as DirectKPN. To validate the advantages of our proposed DML method, we implement these two strategies by using the same backbone (with the same hyper-parameters) of the weight prediction network in our DML for fair comparison. For DirectNet, we implement it using the residual learning strategy [70] for better convergence. All the three competitors are trained on the training set of RealSR [21], and tested on the RealSR testing set. PSNR and SSIM are used as quantitative measures.

We first evaluate the performance of the three strategies on LR image generation. The results are listed in Table 4.1. One can see that DML performs constantly better than DirectNet or DirectKPN on all the three zooming factors, with an improvement of $0.23dB$ and $0.20dB$ in PSNR, respectively. This shows that DML can generate more realistic LR images, owing to our proposed strategy of learning basis kernels and

predicting pixel-wise combination weights. Besides, it is observed that DirectKPN performs slightly better than DirectNet. This shows that by taking into account the image degradation process, better LR generation performance can be achieved by learning to predict pixel-wise kernels than directly predicting LR image pixels.

We then evaluate their effectiveness on improving SISR. We apply the three LR image generation models to the collected HR image dataset, synthesizing 3150 HR-LR images pairs by each model. We add small AWGN to those HR-LR pairs (refer to Section 4.3.3 for details), and train three VDSR models. Finally, we apply these three VDSR models to the LR images in the testing part of the RealSR dataset, and obtain the super-resolved HR images. The PSNR/SSIM results are listed in Table 4.1. One can see that the VDSR network trained on synthetic HR-LR pairs generated by our DML method, performs constantly better (around $0.15dB$ in PSNR) than those trained on pairs generated by DirectNet or DirectKPN. This validates the superiority of DML to DirectNet and DirectKPN on improving SISR performance. Our DML method can generate realistic LR images with a smaller gap to real-world LR images, therefore leading to better SISR results than DirectNet and DirectKPN.

We visualize the pixel-wise degradation kernels predicted by our DML and DirectKPN in Fig. 4.4 (note that DirectNet does not predict kernels). One can see that predicted degradation kernels by DML vary with the image local content, whereas the degradation kernels predicted by DirectKPN are simple and rather uniform across the whole image. This is probably because when we directly learn the pixel-wise degradation kernel, the solution space is too large so that DirectKPN can only converge to a simple solution, resulting in uniform kernels for an input image. In contrast, our DML strategy can effectively reduce the kernel space and thus result in a more robust adaptive degradation kernel prediction model. We also visualize the SISR results by the three degradation models in Fig. 4.4. It can be seen that our DML based SISR method exhibits better visual quality with more details and less artifacts.

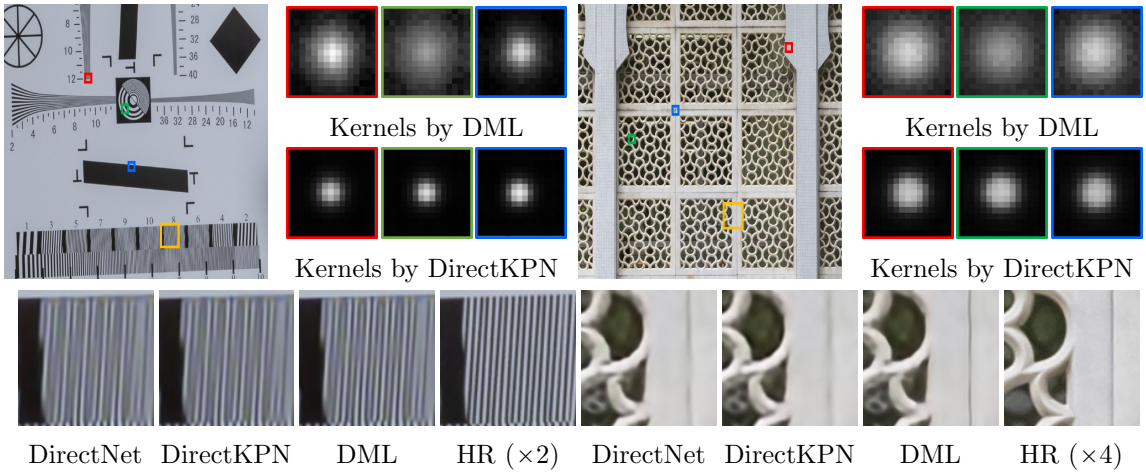


Figure 4.4: Visualization of predicted degradation kernels by DML and DirectKPN. One can see that the degradation kernels predicted by DML vary with the image local content, whereas the kernels predicted by DirectKPN are simple and rather uniform across the whole image. We also show the SISR results of the VDSR models trained on the synthetic HR-LR pairs by DML, DirectNet and DirectKPN. One can see that the model based on DML can recover more details with less artifacts.

4.4.4 Experiments on Real-World SISR

As discussed in the Introduction section in this chapter, the goal of our DML is to synthesize realistic HR-LR image pairs to supplement the limited number of real-world HR-LR pairs so that more robust SISR models can be trained. To validate whether this goal is achieved by our DML method, in this section we use VDSR [70] (20 layers) and RCAN [141] (100 layers) as two representative SISR models to perform extensive SISR experiments. By using the HR image dataset we collected, we synthesized 3150 HR-LR pairs via the learned DML model, and denote this dataset by Syn-DML. Note that recently a GAN based HR-LR pair synthesis method called DSGAN [43] was developed. We finetuned this model on the RealSR dataset, and applied it to our HR image dataset to synthesize another dataset of HR-LR pairs, denoted by Syn-DSGAN. Therefore, we can train variants of VDSR/RCAN models by using: only RealSR, only Syn-DSGAN, only Syn-DML, the combination of RealSR and DSGAN, and the combination of RealSR and Syn-DML dataset, resulting in a

Table 4.2: Evaluation of SISR performances on the RealSR [21] dataset by models trained using different training data. The best, second and third results for each SISR network architecture are highlighted in red, blue and green, respectively.

SISR model	Training dataset	LPIPS ↓			PNSR ↑			SSIM ↑		
		×2	×3	×4	×2	×3	×4	×2	×3	×4
VDSR	RealSR	0.141	0.224	0.291	33.60	30.53	28.92	0.957	0.919	0.887
	Syn-DSGAN	0.145	0.240	0.309	32.47	29.57	27.20	0.949	0.908	0.851
	Syn-DML	0.137	0.218	0.284	33.32	30.18	28.60	0.955	0.916	0.886
	RealSR+Syn-DSGAN	0.151	0.234	0.289	33.35	30.13	28.56	0.954	0.915	0.885
	RealSR+Syn-DML	0.124	0.198	0.267	33.50	30.37	28.86	0.957	0.918	0.889
RCAN	RealSR	0.141	0.227	0.283	33.91	30.86	29.26	0.960	0.924	0.896
	Syn-DSGAN	0.148	0.239	0.319	32.45	29.78	27.95	0.948	0.916	0.877
	Syn-DML	0.131	0.210	0.265	33.38	30.29	28.66	0.956	0.918	0.887
	RealSR+Syn-DSGAN	0.143	0.230	0.288	33.50	30.56	28.80	0.956	0.920	0.888
	RealSR+Syn-DML	0.123	0.195	0.242	33.73	30.61	28.99	0.958	0.921	0.891

total of 10 SISR models.

We evaluate the 10 VDSR/RCAN models on two real-world datasets. One is the testing set of RealSR [21]. Since the aligned HR-LR pair are available, we can compute the PSNR/SSIM/LPIPS indices to perform quantitative evaluation. Another is the SR-RGB dataset [139], which consists of many LR images and their unaligned HR counterparts. Qualitative visual comparisons can be made on it for the different SISR models. We'd like to stress that the testing on the second dataset is more important (though qualitative) because it is independent of the RealSR dataset, part of whose samples are used to train the DML and VDSR/RCAN models. The testing results on the SR-RGB [139] dataset can more faithfully reflect the generalization capability of competing SISR models than those on the RealSR dataset.

Results on the RealSR dataset [21]. We apply the competing VDSR/RCAN models to the testing set of RealSR, and the PSNR/SSIM/LPIPS indices are shown in Table 4.2. Note that LPIPS is a perceptual index that measures the perceptual quality of images (lower the better). We can have the following findings. First, the VDSR/RCAN models trained on Syn_DML achieve better LPIPS score in all cases

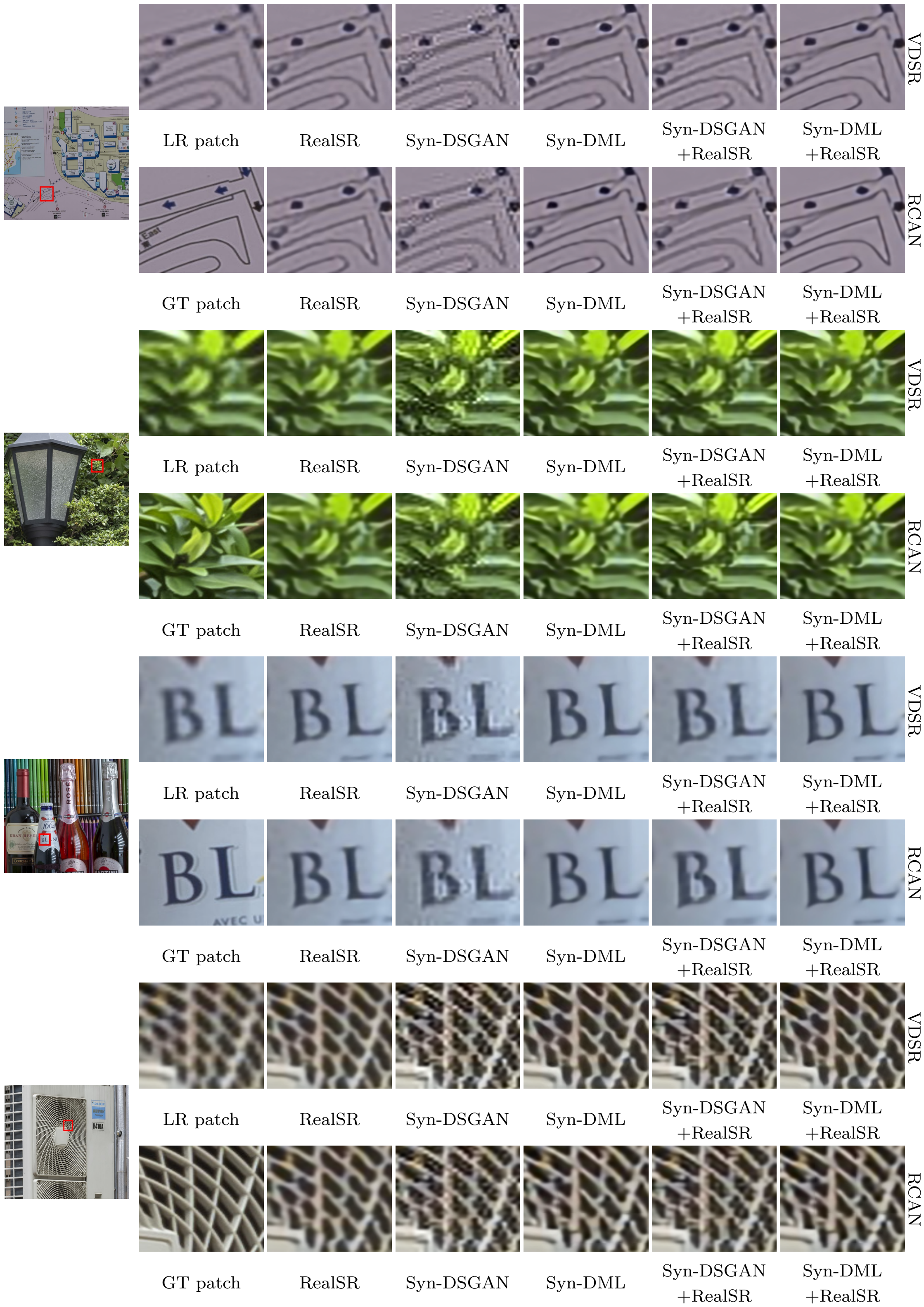


Figure 4.5: Visual comparison of the competing SISR models on RealSR [21] dataset with SR scale ×4.

than the models trained on RealSR. This validates the effectiveness of our model in improving perceptual quality by synthesizing realistic HR-LR image pairs. Second, the VDSR/RCAN models trained on the Syn-DML dataset achieve comparable but slightly inferior PSNR/SSIM indices to the models trained on RealSR. This is not a surprise because the training and testing data for the latter model are from the same source. Third, SISR models trained on Syn-DML perform significantly better (about 1dB) than those trained on Syn-DSGAN, which demonstrates the superiority of our DML method to the GAN based DSGAN [43]. Last, by combining RealSR with the synthetic dataset for training, better quantitative results can be achieved than training using only synthetic dataset. Particularly, the VDSR model ($\times 4$) trained on RealSR+Syn-DML achieves even high SSIM scores than the model trained on RealSR.

In Fig. 4.5, we compare the visual quality of super-resolved HR images by the ten SISR models. One can see that models trained on Syn-DML and RealSR+Syn-DML can effectively recover more image details with more pleasant perceptual quality than the trained using only the RealSR dataset. In particular, the models trained on RealSR+Syn-DML achieve the best visual quality. This validates that our DML method can largely improve the generalization performance of real-world SISR models by synthesizing realistic HR-LR pairs for training.

Results on the SR-RGB dataset [139]. The SR-RGB dataset contains real-world HR and LR images of the same scene, which are however not well aligned. Though it is hard to compute PSNR/SSIM metrics, the HR images in this dataset can be well used a reference for visual comparison of SISR methods. Since the SR-RGB dataset was constructed independently of the RealSR dataset by using different cameras and lens, the results can more fairly demonstrate the generalization capability of an SISR model to real-world scenarios.

In Fig. 4.6, we visualized the super-resolved HR images on SR-RGB dataset

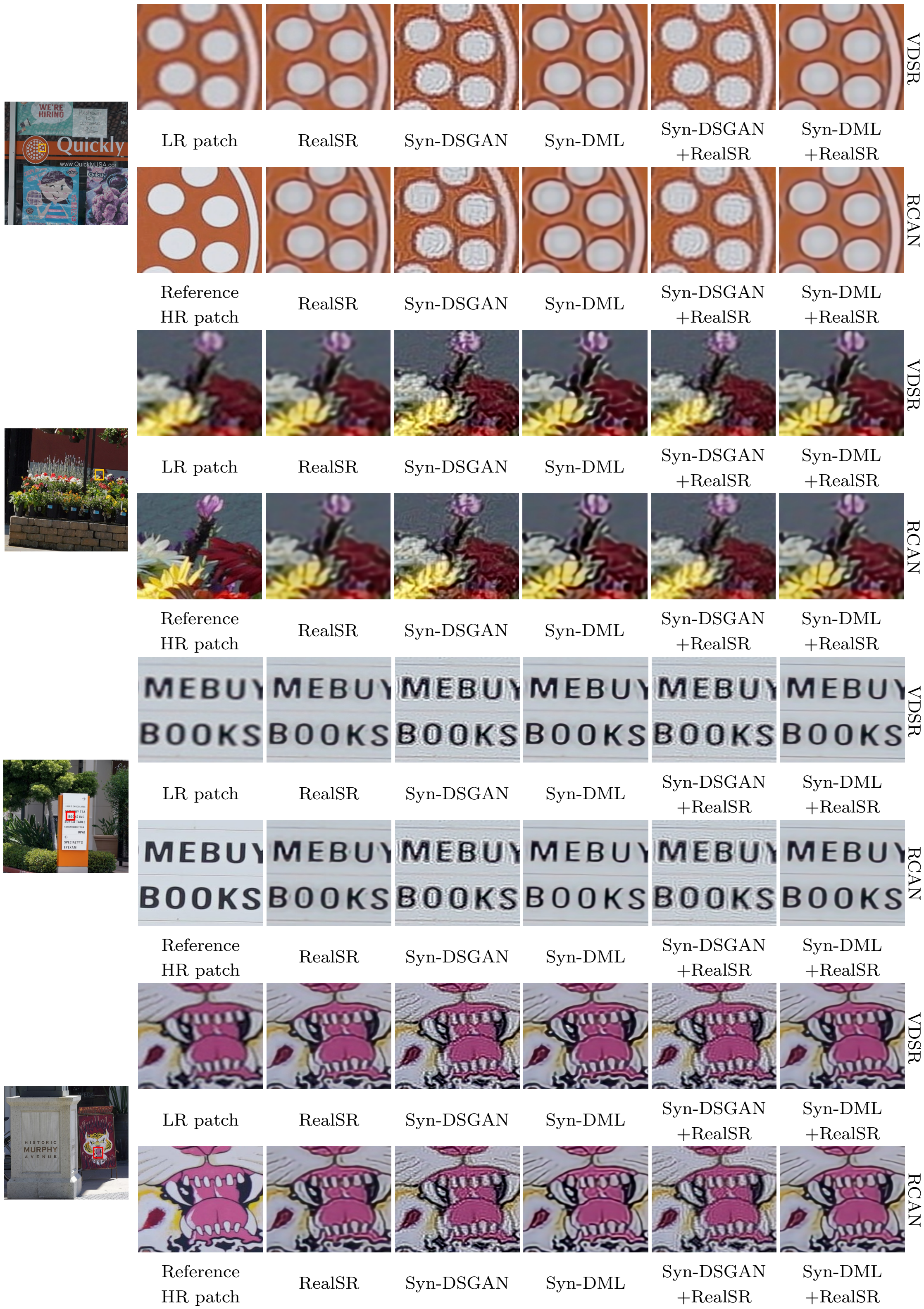


Figure 4.6: Qualitative comparison of competing SISR methods on the SR-RGB [139] dataset with SR scale ×4.

[139] by the ten VDSR/RCAN models trained on different training datasets. One can see that models trained on RealSR dataset can only moderately recover some details. Models trained on Syn-DSGAN produce severe artifacts. Benefitting from the enlarged realistic training data, SISR models trained on Syn-DML can produce visually pleasing results with more fine-grained details. Particularly, the models trained on combined RealSR+Syn-DML deliver the best perceptual quality of super-resolved HR images. The experiments on SR-RGB dataset demonstrate that the SISR models trained by our DML method can be effectively generalized to real-world applications.

4.5 Conclusions

In this chapter we proposed to tackle the generalization problem of real-world SISR models by synthesizing realistic HR-LR pairs. To achieve this goal, we first learned an image degradation model from real-world HR-LR image pairs. Specifically, we learned a set of basis degradation kernels together with a weight prediction network. The degradation kernel at any location was estimated as the linear combination of the basis kernels using the weights predicted by the weight prediction network. The learned degradation model was then used to synthesize 3150 HR-LR image pairs covering various scenes for SISR model training. Our extensive analyses and experiments showed that the proposed degradation model learning method can effectively improve the generalization performance of SISR models to real-world applications.

Chapter 5

Blind Super-Resolution for Real-World Images

In this chapter, we keep on working on the task of real-world single image super-resolution (SISR). As illustrated in chapter 4, despite the prominent results brought by the recent deep learning based SISR methods, most of them assume a certain type of degradation, and therefore struggle to generalize to real-world scenarios where the authentic degradation is complicated and unknown. In this chapter, different from the degradation model learning based framework as used in chapter 4, we address the task of real-world SISR by proposing a blind super-resolution (BSR) method for spatially variant and complex degradations.

Different from the time-consuming iterative updating scheme used in previous BSR methods, we propose to estimate the pixel-wise degradations in a one-step manner, and perform BSR using a deep CNN, whose local filters are adaptive to the estimated degradations. Specifically, we leverage the image edge map to guide the degradation estimation, and design a pyramid U-shaped sub-network to constrain the smoothness of estimated degradation map, with which a hyper-parameter network is trained to generate the adaptive filters to perform BSR. Extensive experiments are carried out on synthetic benchmark datasets and real-world images, and the results show that our BSR method achieves leading performance both quantitatively

and qualitatively. It also has better runtime efficiency compared to existing BSR methods.

5.1 Introduction

In section 4.1 of the previous chapter, we have thoroughly introduced the background of single image super-resolution (SISR). Here, we briefly retrospect the background and challenge of SISR, and focus on the introduction of the sub-field of SISR, i.e., blind image super-resolution. Finally, we give the motivation of the proposed method in this section. For the detailed background of SISR, please refer to section 4.1 in chapter 4.

SISR aims to recover a high-resolution (HR) image from its low-resolution (LR) observation, which is a highly valuable technique for improving the resolution of digital images. Despite the great success brought by deep convolutional neural networks (CNNs) based SISR methods, the real-world SISR task is still challenging. This is mainly due to that most CNN based SISR models are trained for a fixed type of degradation, e.g., bicubic downsampling, whereas the degradation in real-world scenarios is much more complicated. Therefore these SISR models have poor generalization ability to real-world LR images.

Several works proposed to bridge the domain gap between realistic degradation type and the synthetic ones by constructing real-world SISR datasets [21, 140, 24, 108, 75], to characterize the authentic degradations in real-world scenarios. However, datasets of this kind are often limited in the number of images and degradation types. To overcome this, in chapter 4 we proposed to learn the degradation model from these datasets, and then generate large scale synthetic but realistic training image pairs. In this chapter, instead of relying on these RealSR dataset, we work on another line, i.e., blind image super-resolution, to tackle the task of real-world SISR.

5.1.1 Blind Image Super-Resolution

Apart from collecting realistic HR/LR image pairs, efforts have also been made to design blind super-resolution (BSR) models [13, 29, 52, 90, 118] for images with unknown degradation. They generally first estimate the degradation model of the given image, and then utilize a non-blind SISR model to adapt to the estimated degradation.

Existing BSR methods can be grouped into two categories. The first category of methods iteratively and alternately estimate the degradation kernel and the HR image. For example, Wang *et al.* [118] employed the maximum a-posterior framework and modeled the prior of HR image patches using Markov random field. They then alternatively updated the HR patches and the degradation kernel. Based on the observation that an incorrect degradation would cause either over-smoothing or over-sharpening of the HR image, a corrector was trained in IKC [52] to gradually correct the estimated degradation from the estimated HR image. To handle spatially variant degradation, a discriminator was trained in BSRSVD [29] to estimate the image artifacts caused by mismatched degradation kernel. This category of methods have achieved nice SISR performance for simple uniform degradations, whereas the iterative updating scheme is very costly, impeding their usage in real applications.

Another category of BSR methods however directly estimate the degradation from the given LR image only by exploiting natural image priors. Inspired by the similarity of image patches across scales, Michaeli & Irani [90] proposed to extract the nearest-neighbor patches to estimate the degradation kernel. They further proposed KernelGAN [13] using deep CNN by assuming that the correct degradation kernel could generate a downsampled version of the LR image that follows similar distribution to the LR image itself. These methods consider only the degradation type of blurring, and can easily fail on images with complex degradations, e.g., noise

and compression.

However, existing BSR methods have several limitations. First, most of them can only estimate the global degradation of an image, whereas the degradation of a real-world image is generally non-uniform and spatially variant. Second, these methods adopt a simple blur+downsampling degradation model, whereas real-world images often suffer from complex degradations involving noise and compression artifacts. Third, most existing BSR methods employ an iterative scheme to estimate the degradation hyper-parameters, which is time-consuming and hard to be used in practical systems. These limitations motivate us to propose a BSR method for real-world SISR, which is capable to process spatially variant complex degradations with fast speed.

5.1.2 Motivation

In this chapter, we address the above issues by proposing a fast BSR method for real-world images, namely BSR-RW. The proposed BSR-RW method learns to handle real-world LR images with spatially variant and complex degradations of various types of blurring, noise, compression, etc., and the combination of them. Specifically, our model consists of two branches of networks: one branch estimates the pixel-wise degradations from the given LR image, while another branch performs super-resolution, whose local filters are adaptive to the estimated pixel-wise degradations. In order to achieve high processing speed, we adopt a one-step degradation estimation strategy which is free of the time-consuming iterative updating process in previous BSR methods. However, directly estimating the non-uniform degradation from an LR image is a non-trivial job. Considering the fact that the edge region provides stronger hint of the underlying degradation than the flat region [129, 26], we utilize the image edge map to guide the degradation estimation. Furthermore, a pyramid U-shaped sub-network is designed to regularize the smoothness of the estimated

pixel-wise degradation map. With the estimated spatially variant degradation map, we employ a hyper-parameter network to generate degradation-adaptive filters, which are used to extract image features and perform image super-resolution.

We evaluate the effectiveness of our BSR-RW method on both synthetic datasets with uniform and non-uniform degradations and real-world images. Experimental results show that our method achieves state-of-art performance, and greatly improves the runtime efficiency over existing BSR methods. We also used the RealSR dataset [21] for evaluation, where the HR/LR image pairs were collected by using lens zooming of digital cameras. By fine-tuning our model on the RealSR datasets, it can easily adapt to the degradation type of lens zooming, and achieves highly competitive results on RealSR dataset. Furthermore, we collect a set of real-world images, whose underlying HR counterparts are unavailable, from various sources for evaluation. Our proposed BSR-RW method exhibits best qualitative results among the competing methods, owing to its capability of degradation estimation and the degradation-adaptive super-resolution filters.

5.2 Blind Super-Resolution for Real-World Images

Most existing SISR methods [13, 52, 29] consider only blurring and downsampling in the degradation formulation from an HR image to the LR image. Such degradation modeling ignores the effects of noise and JPEG compression. To better describe the degradation process in real-world images, we adopt the following model:

$$\mathbf{I}^L = f_{Jpeg}((\mathbf{I}^H \odot \mathbf{k}) \downarrow_d + \mathbf{N}), \tag{5.1}$$

where $\mathbf{I}^H$ and $\mathbf{I}^L$ denote the HR and LR images, respectively, $\mathbf{k}$ is the blur kernel, $\downarrow_d$ is the bicubic downsampling operator, f_{Jpeg} is the function of JPEG compression, and $\mathbf{N}$ is the random noise. Please note that the JPEG compression artifacts are image content dependent and spatially variant.

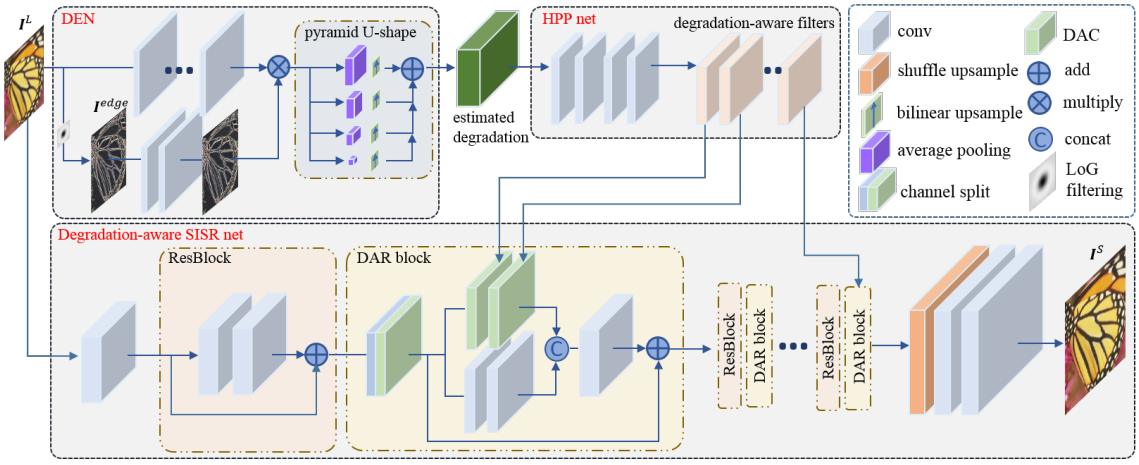


Figure 5.1: Overview of our proposed network architecture. Given an LR image, our method first estimates the pixel-wise degradation with the guidance of edge map and the designed pyramid U-shaped module. The estimated degradation then goes through the HPP-net to generate degradation-aware filters, which are finally used in the SISR network for super-resolution.

To achieve fast inference time, we propose a one-stage degradation estimation module, followed by an SISR network for adaptive super-resolution. The overview of our network architecture is illustrated in Figure 5.1. Our network, namely BSR-RW, consists of three major parts: a one-stage degradation estimation module to predict the pixel-wise degradation, a hyper-parameter network which takes the estimated degradation as input, and a degradation-aware SISR network whose convolutional filters are based on the hyper-parameter network outputs. These three parts are trained jointly. Note that the SISR network can be implemented with different structures and depths to balance between performance and efficiency. In the following, we present the design of these three parts in detail.

5.2.1 Degradation estimation network

The degradation estimation network (DEN) takes the LR image $\mathbf{I}^L$ as input to estimate the pixel-wise degradation. To meet the demand of practical applications, a fast and accurate one-stage degradation estimation is desired. Considering that the

large bulk of image flat region contributes little to the degradation estimation since degradations occur more distinctly in image edge region, we leverage the edge map to generate a spatial attention mask to guide the feature extraction for degradation estimation.

Guidance of edge map. We stack 6 conv layers for feature extraction from the given LR image. The extracted feature map $\mathbf{F}$ is then multiplied with the spatial mask $\mathbf{M}_{edge}$ to draw attention on the edge region. Considering that Laplacian of Gaussian (LoG) filter is an efficient edge detector with robustness to noise, we explicitly compute the edge map $\mathbf{I}^{edge}$ by applying the LoG filter on the LR image. The edge map $\mathbf{I}^{edge}$ is further encoded via two conv layers and the sigmoid function is used to generate the mask $\mathbf{M}_{edge}$. The final feature map, denoted as $\mathbf{F}_s$, can be formulated as:

$$\mathbf{F}_s = \mathbf{F} * S(f_e(\mathbf{I}^{edge})), \tag{5.2}$$

where f_e denotes the encoding function for the edge map, and S is the sigmoid function.

Pyramid U-shaped module. With the extracted feature map $\mathbf{F}_s$, one may use a conv layer to directly map it to the degradation map. Unfortunately, such a direct mapping would lead to an unstable estimation of the degradation map. Actually, for most of the natural images, the degradations vary locally but smoothly. Therefore, the degradation map should be locally smooth. Taking this prior knowledge into consideration, we design a U-shaped structure to first downsample the feature $\mathbf{F}_s$ using an average pooing layer, then resize it to the original resolution by bilinear upsampling. The average pooling layer perceives contextual information from a larger receptive field, and the upsampling ensures the smoothness of the estimated degradation. Furthermore, to exploit the information across multiple image scales, we use a pyramid of four U-shaped units and sum up the feature maps, as shown

in Figure 5.1. The pooling strides of the first three units are empirically set as 7, 15 and 31, respectively. Global average pooling is used in the last U-shaped unit to leverage the global degradation.

Loss function. With the features output by the pyramid U-shaped module, we use a single conv layer to predict the pixel-wise degradation, denoted as $\widetilde{\mathbf{d}} \in R^{h*w*l}$, where h, w are the height and width of the input LR image, and l is the dimension of the estimated degradation vector. Unlike methods in [136, 52], which predict the low-dimensional representation of degradation using pre-computed PCA basis, we learn the latent representation to improve the generalization ability of our model to complex degradations. This is accomplished by adopting a reconstruction loss on $\widetilde{\mathbf{d}}$. By feeding $\widetilde{\mathbf{d}}$ into a shallow CNN, denoted as F_{rec}, the groundtruth degradation $\mathbf{h}$ can be estimated. The loss function is given as:

$$l_{DEN} = ||F_{rec}(\widetilde{\mathbf{d}}) - \mathbf{h}||_1. \tag{5.3}$$

We use two conv layers to constitute the CNN F_{rec}. We concatenate the vectorized blur kernel $\mathbf{k}$, noise variance σ and JPEG compression quality to constitute the groundtruth degradation vector $\mathbf{h}$.

5.2.2 Degradation-aware SISR network

With the estimated pixel-wise degradation $\widetilde{\mathbf{d}}$, we then design a degradation-aware network for super-resolution. To improve the robustness of our SISR network to inaccurate $\widetilde{\mathbf{d}}$, we add slight Gaussian noise to $\widetilde{\mathbf{d}}$ during training. Different from [136, 52] which concatenate the estimated degradation with either the input image or the extracted features, we design a degradation-aware residual (DAR) block to adapt to different degradations.

The proposed SISR network is illustrated in Figure 5.1. We use the architecture of EDSR [80] as our backbone, and place the DAR block after each ResBlock. The DAR

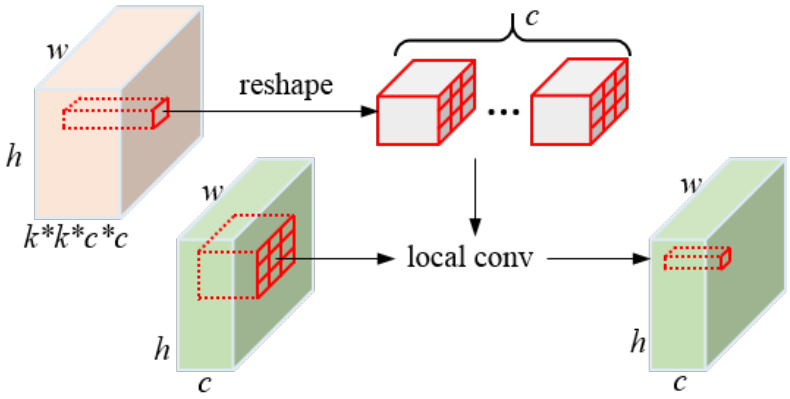


Figure 5.2: Illustration of our DAC layer.

block is composed of several degradation-aware convolutional (DAC) layers, which are similar to the dynamic local convolutional layer [66], but their filters depend on the estimated degradation map. The local filters in DAC layer are generated via the hyper-parameter network (HPP-net), which will be discussed in Sec. 5.2.3. Given the input feature $\mathbf{F} \in R^{h*w*c}$, the HPP-net first generates the pixel-wise filters $\mathbf{w}_{i,j} \in R^{k*k*c*c}$, where i, j denote the spatial coordinates, and k, c are the filter size and channel dimension, respectively. As illustrated in Figure 5.2, $\mathbf{w}_{i,j}$ is then applied on the local region centered at i, j of feature map $\mathbf{F}$ to obtain the filtered results. To balance between performance and efficiency, we split the input features along the channel dimension into two parts. One part is the common conv layer, and the other goes through the DAC layer for transformation with regard to the degradation. We then concatenate the two parts, and use the shortcut connection to obtain the final output of the DAR block.

The input LR image first goes through a single conv layer to increase the channel dimension to 128. We then stack several groups of Resblocks and DAR blocks to build our adaptive feature extraction module. The extracted deep features are finally passed through the shuffle upsampling layer and the subsequent two conv layers to generate the super-resolved image $\mathbf{I}^S$. All the conv filters are of size 3×3. We use

ReLU as the activation function. The loss function is defined as the L_2 norm between the recovered image $\mathbf{I}^S$ and groundtruth HR image $\mathbf{I}^H$:

$$l_{SR} = ||\mathbf{I}^S - \mathbf{I}^H||_2^2. \tag{5.4}$$

5.2.3 Hyper-parameter network

The HPP-net takes the estimated degradation $\widetilde{\mathbf{d}}$ as input and produces a bunch of filters for the DAC layers. We first stack 4 conv layers to encode the latent representation of estimated degradation $\widetilde{\mathbf{d}}$. The channel of the encoded degradation is set as 32. We then use one conv layer for each DAC layer to generate the pixel-wise degradation-aware filters.

We jointly train the three branches, i.e., DEN, HPP-net and degradation-aware SISR network, from scratch. The final loss function is the combination of l_{DEN} and l_{SR}:

$$l = l_{DEN} + l_{SR}. \tag{5.5}$$

By using different backbones, in this chapter we implement two models: *BSR-RW light* and *BSR-RW*, which contain 4 and 8 groups of Resblocks and DAR blocks, respectively. Details will be introduced in next section.

5.3 Experiments

5.3.1 Experimental setting

We conduct experiments on synthetic and real-world super-resolution datasets. We use the training set of [136], which comprises 800 images from the DIV2K [4] dataset and 4,744 images from the WED [84] dataset, to generate our training data. Except in Sec. 5.3.3, where we follow the setting in existing BSR methods [13, 29, 52] to fairly compare with them, in all the other experiments we generate our training data

by Eq. (5.1) to better model the degradation process of real-world data. We follow [52] to model the blurring kernel as anisotropic Gaussian kernel of size 15×15 with the kernel width ranging in [0.2, 2.0], [0.2, 3.0] and [0.2, 4.0] for zooming factors 2, 3 and 4, respectively. As for the noise and JPEG compression artifacts, we use the additive white Gaussian noise (AWGN) with variance from 0 to 15, and sample the JPEG compression factor within the range of [70, 100]. More details about the experiment settings will be introduced in the corresponding sections.

We crop HR patches of size 192×192 from the HR images, and generate the corresponding LR patches of sizes 96×96, 64×64, and 48×48 for zooming factors $\times 2$, $\times 3$ and $\times 4$, respectively. As in [52, 29, 136], we train our model in the RGB space, and evaluate on the Y channel in the YCbCr space. We train our model by using the Adam optimizer [73] with the default setting ($\beta_1 = 0.9$, $\beta_2 = 0.999$). The batch size is set as 16. We train our models for 500K iterations in total. The learning rate is set as $1e^{-4}$ for the first 300K iterations and the learning rate for the remaining 200K iterations is $1e^{-5}$.

5.3.2 Ablation study

We first perform ablation experiments to demonstrate the role of different components of our BSR-RW network. Specifically, we implement four variants of our BSR-RW model: *BSR-RW w/o edge* and *BSR-RW w/o U-shape* by removing the guidance of edge map and the pyramid U-shaped module from our full *BSR-RW* model; *BSR-RW uniform* by considering only spatially invariant degradation and replacing the pyramid U-shaped module in *BSR-RW* with a global average pooling layer; *BSR-RW w/o DEN* is the non-blind baseline of *BSR-RW* without estimating the degradation model, which simply utilizes a unified network to deal with different kinds of degraded images. As *BSR-RW w/o edge*, *BSR-RW w/o U-shape* and *BSR-RW uniform* have a degradation estimation component, for fair comparison we add

Table 5.1: The PSNR/SSIM results of BSR-RW and its variants on synthetic non-uniform degradation. The zooming factor is $\times 4$.

Method	Set5		Set14		BSD100	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
BSR-RW w/o DEN	29.06	0.9025	26.61	0.8232	26.08	0.7847
BSR-RW uniform	29.06	0.9064	26.71	0.8254	26.20	0.7914
BSR-RW w/o edge	29.25	0.9101	26.82	0.8279	26.20	0.7919
BSR-RW w/o U-shape	29.28	0.9098	26.84	0.8267	26.22	0.7921
BSR-RW	29.35	0.9112	26.93	0.8301	26.32	0.7947

five more ResBlocks to the *BSR-RW w/o DEN* to make the four variants to have similar number of parameters and model complexity.

We first use Set5, Set14 and BSD100 for quantitative evaluation, and synthesize non-uniformly degraded LR images by applying Gaussian blurring on the HR images with gradually increased kernel width (increase from 0.2 to 2 from left to right), followed by $\times 4$ downsampling, AWGN with $\sigma = 5$ and JPEG compression with factor 90. The PSNR/SSIM results of our BSR-RW and its variants are shown in Table 5.1. We can see that *BSR-RW uniform* achieves slightly better PSNR indexes over *BSR-RW w/o DEN*. By considering spatially variant degradation maps, *BSR-RW w/o edge* and *BSR-RW w/o U-shape* achieve better results than *BSR-RW w/o DEN* and *BSR-RW uniform*. This clearly demonstrates the importance of estimating spatially variant degradation maps. Moreover, *BSR-RW* further improves *BSR-RW w/o U-shape* and *BSR-RW w/o edge*, which validates the effectiveness of the proposed edge map guidance strategy and the pyramid U-shaped module.

In addition to the quantitative study by using synthetic LR images, we also evaluate qualitatively *BSR-RW* and its variants on real-world images. We take the original images in City100 [24] and BSD100 datasets as input and use the five models to enlarge the input images with zooming factor $\times 4$. The SR results by different methods are shown in Figure 5.3. One can see that both *BSR-RW w/o DEN* and

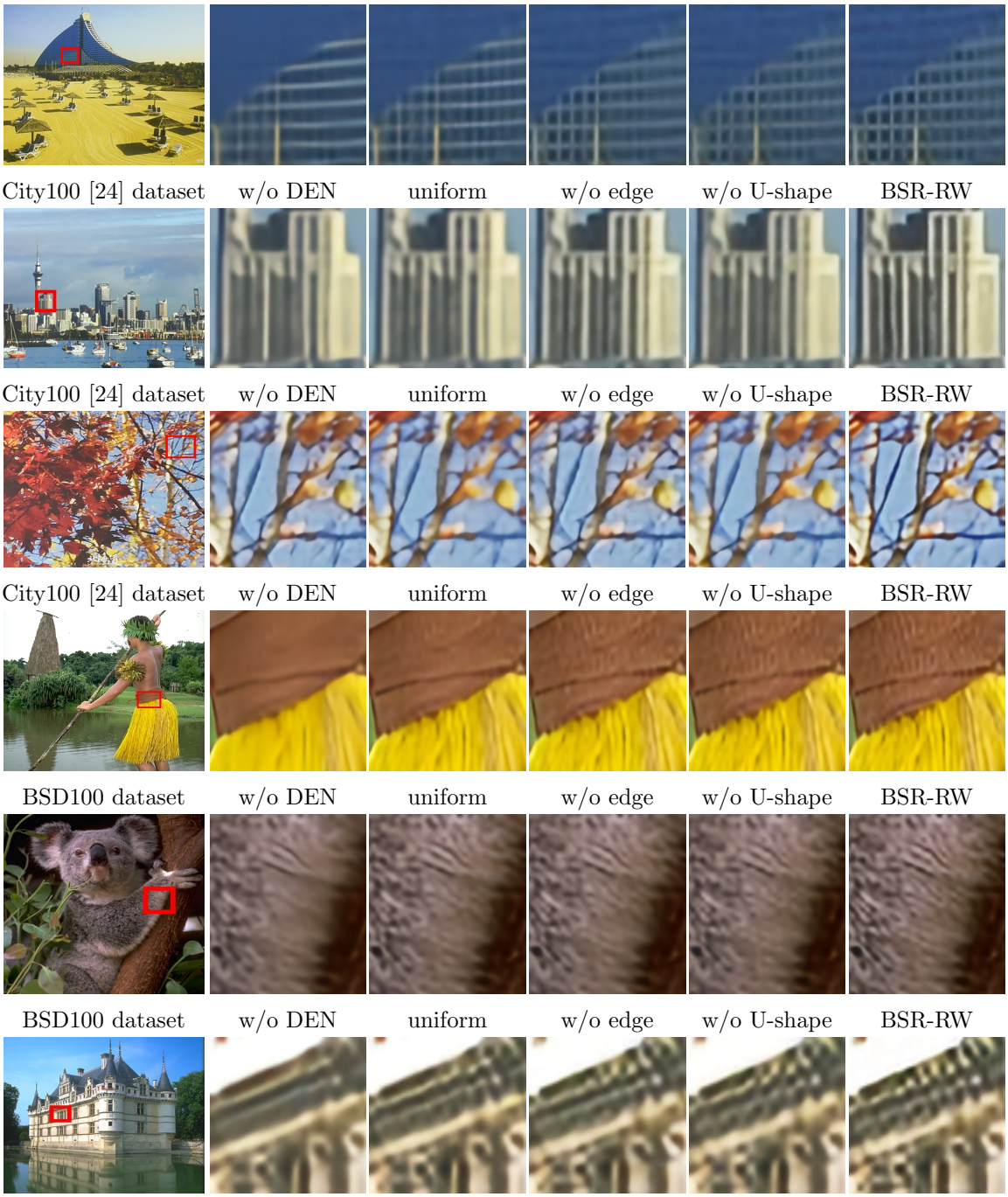


Figure 5.3: Visual comparison of blind SISR results by BSR-RW and its variants on real-world images. The input images are from City100 [24] and BSD100 datasets, respectively.

BSR-RW uniform fail to recover the edges and detailed textures, resulting in blurry SR images. *BSR-RW w/o edge* and *BSR-RW w/o U-shape* deliver better quality SR images by estimating the spatially variant degradation map. By leveraging the edge map and the pyramid U-shaped module, *BSR-RW* presents more fine-grained details than *BSR-RW w/o edge* and *BSR-RW w/o U-shape*, exhibiting the best visual quality.

5.3.3 Comparison with the state-of-arts

In this section, we compare the proposed BSR-RW with state-of-the-art BSR methods. The recently developed KernelGAN [13], IKC [52] and BSRSVD [29] are employed for the comparison. Since the source codes of IKC and BSRSVD are not released by the original authors, we use the pre-trained models from the third-party implementation in [1] and [2] for evaluation. For KernelGAN, we use the source codes from authors' homepage to estimate the blur kernel, and then use the non-blind SISR method SRMD [136] to obtain the final SR images. We also report the results of representative non-blind SISR approaches, EDSR [80] and RCAN [141], for reference. The models of EDSR and RCAN are obtained from the authors' homepages, which are trained using bicubically downsampled HR/LR pairs. In addition to our full BSR-RW model, we also provide the results of its non-blind baseline *BSR-RW w/o DEN* and a light-weight version *BSR-RW light*.

We conduct experiments on two different settings. In our first experiment, we follow the experimental setting of existing blind SR works [52, 13] which only take uniform degradation into consideration. While in our second experiment, we evaluate different algorithms on the more challenging non-uniform degradation.

Uniform degradation. We follow the experimental setting of existing blind SISR methods to evaluate different methods on uniformly degraded images. Specifically, we

Table 5.2: The PSNR results of competing methods on benchmark datasets with synthetic uniform degradation. The best, second and third results are highlighted in red, blue and green, respectively. "-" means the result is not available.

Method	Set5			Set14			BSD100		
	×2	×3	×4	×2	×3	×4	×2	×3	×4
bicubic downsampling									
EDSR [80]	38.11	34.65	32.46	33.92	30.52	28.80	32.32	29.25	27.71
RCAN [141]	38.27	34.74	32.63	34.12	30.65	28.87	32.41	29.32	27.77
KernelGAN [13]	17.78	21.97	26.07	16.82	20.13	23.83	16.55	20.74	22.97
BSRSVD [29]	28.24	-	-	26.27	-	-	26.40	-	-
IKC [52]	-	-	32.00	-	-	28.41	-	-	27.51
w/o DEN	37.55	33.99	31.84	32.86	30.04	28.36	31.88	28.91	27.46
uniform light	37.67	34.16	31.98	33.12	30.17	28.46	31.97	28.99	27.55
BSR-RW uniform	37.84	34.35	32.26	33.43	30.21	28.66	32.09	29.08	27.65
Gaussian kernel with width 1.3									
EDSR [80]	30.63	28.64	29.97	27.84	25.14	25.04	26.71	26.66	25.46
RCAN [141]	30.62	28.65	29.88	27.83	25.14	25.08	26.72	26.65	25.48
KernelGAN [13]	28.51	32.21	29.07	27.17	27.68	26.32	26.11	27.08	25.60
BSRSVD [29]	28.34	-	-	26.33	-	-	26.46	-	-
IKC [52]	-	-	31.63	-	-	28.27	-	-	27.36
w/o DEN	37.24	33.99	31.81	32.99	30.01	28.33	31.65	28.86	27.40
uniform light	37.45	34.13	32.05	33.01	30.02	28.47	31.77	28.98	27.56
BSR-RW uniform	37.53	34.22	32.21	33.13	30.16	28.63	31.87	29.01	27.61
Gaussian kernel with width 2.6									
EDSR [80]	26.37	25.84	26.32	24.65	24.16	24.33	24.76	24.84	24.69
RCAN [141]	26.37	25.84	26.32	24.66	24.16	24.33	24.77	24.86	24.70
KernelGAN [13]	27.63	25.43	27.63	25.53	24.29	24.95	25.65	25.22	24.74
BSRSVD [29]	26.40	-	-	24.29	-	-	24.57	-	-
IKC [52]	-	-	30.48	-	-	27.96	-	-	27.17
w/o DEN	36.12	33.13	31.67	32.26	28.95	28.19	31.19	28.35	27.41
uniform light	36.26	33.42	32.05	32.27	29.11	28.21	31.17	28.45	27.55
BSR-RW uniform	36.48	33.51	32.26	32.48	29.17	28.33	31.40	28.60	27.59

synthesize LR images as in [136] by applying isotropic Gaussian blurring kernel with width {0.2, 1.3, 2.6} on the HR images, followed by bicubic downsampling. Please note that the kernel with width 0.2 is actually an impulse signal, and this degradation is equivalent to bicubic downsampling. For fair comparison, we use the version of *BSR-RW uniform* in this experiment, which adopts the global average pooling layer to estimate the global degradation. The testing sets include the commonly used benchmark datasets: Set5, Set14 and BSD100.

The PSNR results of the competing methods are listed in Table 5.2. EDSR and RCAN achieve better results for degradation type of bicubic downsampling because their models are specifically trained for this simple degradation; however, their performances deteriorate severely on the other degradation types. Compared with other blind SISR methods, the proposed *BSR-RW uniform* achieves consistently better results on all the three zooming factors. Moreover, our lightweight version, *BSR-RW uniform light*, also shows superior results with better efficiency (please refer to Table 5.4 for the FLOPs and runtime). This verifies that BSR-RW is a robust framework which is able to deliver stable BSR results with different backbone networks.

Non-uniform degradation. We then evaluate different methods on the more challenging non-uniform degradation setting. We first apply Gaussian blurring (from left to right) to HR images with gradually increased kernel width in [0.2, 2], and then utilize bicubic downsampling to generate LR images. The same testing sets, i.e., Set5, Set14 and BSD100, are adopted to compare different methods.

The SISR results by different methods are shown in Table 5.3. As KernelGAN and IKC are designed for uniformly degraded images, they fail to generate satisfactory SISR results. The non-blind SISR models EDSR and RCAN cannot achieve good results either. Comparing with BSRSVD, which is designed for non-uniform degradations, the proposed BSR-RW achieves significantly better results.

Table 5.3: The PSNR results of competing methods on benchmark datasets with synthetic non-uniform degradation. The best results are highlighted in **bold**. "–" means the result is not available.

Method	Set5			Set14			BSD100		
	×2	×3	×4	×2	×3	×4	×2	×3	×4
EDSR [80]	29.45	28.89	28.13	26.92	26.23	25.54	26.98	26.18	25.45
RCAN [141]	29.47	28.93	28.30	26.96	26.31	25.66	27.02	26.24	25.56
KernelGAN [13]	22.72	27.65	26.91	20.91	24.43	25.43	20.46	23.57	24.42
IKC [52]	-	-	28.73	-	-	25.94	-	-	25.74
BSRSVD [29]	27.44	-	-	25.50	-	-	25.61	-	-
w/o DEN	34.81	32.83	31.17	31.24	29.44	27.83	30.44	28.46	27.11
light	35.01	32.74	31.20	31.35	29.25	27.99	30.36	28.37	27.22
BSR-RW	**35.08**	**32.97**	**31.27**	**31.55**	**29.55**	**28.05**	**30.52**	**28.60**	**27.27**

Table 5.4: The FLOPs and runtime of competing methods. The FLOPs and runtime are tested on 128 × 128 color image for ×4 SISR with an Nvidia 2080Ti GPU.

Method	FLOPs (G)	params (M)	runtime (s)
EDSR [80]	1277	43	0.0765
RCAN [141]	543	15.56	0.0594
KernelGAN [13]	N/A	N/A	53.2528
IKC [52]	876	5.21	0.3553
BSRSVD [29]	N/A	**1.10**	13.0611
BSR-RW w/o DEN	485	10.32	0.0453
BSR-RW light	**233**	7.06	**0.0298**
BSR-RW	470	9.87	0.0514

We also provide the FLOPs, no. of parameters and runtime of the competing BSR methods in Table 5.4. Our method achieves largely improved runtime efficiency over KernelGAN, IKC and BSRSVD. The superior PSNR performance and the smaller computational burden make our BSR-RW framework with one-stage degradation estimation a very attractive choice for BSR in real applications.

Table 5.5: The PSNR/SSIM results of competing methods on the RealSR [21] dataset. The superscripts † and * denote fine-tuning and training from scratch using the RealSR training set, respectively. The best results are shown in **bold**. "–" means the result is not available.

Method	×2		×3		×4	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
KernelGAN [13]	26.23	0.7579	26.97	0.7669	25.92	0.7284
IKC [52]	-	-	-	-	27.18	0.7839
BSRSVD [29]	27.27	0.8028	–	–	-	-
BSR-RW	31.33	0.9412	28.72	0.9054	27.38	0.8738
LP-KPN [21]	33.35	0.9562	33.30	0.9190	28.65	0.8858
BSR-RW w/o DEN*	33.38	0.9567	30.22	0.9175	28.53	0.8849
BSR-RW uniform†	33.45	0.9570	30.28	0.9188	28.69	0.8861
BSR-RW†	**33.59**	**0.9587**	**30.45**	**0.9217**	**28.83**	**0.8907**

5.3.4 Evaluation on RealSR dataset [21]

In this section, we evaluate our method on real-world LR images by using the RealSR [21] testing set (version 3), where aligned HR/LR pairs are provided to enable quantitative comparison. Considering that the degradation model in Eq. (5.1) is general but not optimal to specific type of degradations, we fine-tune a little our models using the RealSR training set to adapt to the degradation of lens zooming in RealSR. The learning rate is fixed as $1e^{-5}$ for 100K iterations. We use "†" to denote the fine-tuned models. The results of LP-KPN in [21] are also provided for comparison. As LP-KPN was trained on the Y channel of YCbCr space, for fair comparison we retrained LP-KPN in RGB space. We also trained an SISR model from scratch by using the network of *BSR-RW w/o DEN* on the RealSR training set, namely *BSR-RW w/o DEN**.

The PSNR/SSIM results of competing methods on RealSR dataset are listed in Table 5.5. We can see that *BSR-RW* without fine-tuning still achieves acceptable results, much higher than KernelGAN [13], IKC [52] and BSRSVD [29]. With fine-

Figure 5.4: Visual comparison of competing SISR methods on RealSR [21] dataset with SR factor ×2 and ×4.

tuning, our *BSR-RW*† method can quickly adapt to the degradation type of lens zooming, and achieves better performance than *BSR-RW w/o DEN** and LP-KPN, which are fully trained using the RealSR training set.

Fig. 5.4 shows the super-resolved images on RealSR [21] dataset with zooming factors $\times 2$, $\times 3$ and $\times 4$, respectively. One can observe that both KernelGAN [13], IKC [52] and BSRSVD [29] produce either over-smoothed results, or shaper edges with severe artifacts. LP-KPN and *BSR-RW w/o DEN**, which are trained using RealSR [21] training set, can effectively reduce artifacts, however still can not recover fine-grained textures, partially because of the limited training data in RealSR training set. Our *BSR-RW*, trained using synthetic image pairs with the degradation model in Eq. (1), delivers sharper edges and finer textures with less artifacts. By fine-tuning our *BSR-RW* on the RealSR training set, our *BSR-RW* † delivers the best visual quality.

5.3.5 Visual comparison on real-world images

Finally, we collect real-world images from various sources to validate the effectiveness of our method. Specifically, we use the images from SRRGB [140], City100 [24] and Zurich [64] datasets for evaluation, which were captured using different devices under various scenes. We compare our *BSR-RW* with the BSR methods KernelGAN and IKC. We also provide the results of *BSR-RW w/o DEN**, RCAN [141] and EDSR [80] for reference. Since there are no ground-truth HR counterparts for the testing images, we compare the visual quality of the SISR results by different methods.

The visual results of competing SISR methods are shown in Figure 5.5 $\sim$ 5.7. One can see that EDSR/RCAN produce rather smoothed results with blurry edges. KernelGAN generates sharper edges but introduces much artifacts at the same time. IKC generates smoothed images with some ringing artifacts. *BSR-RW w/o DEN** generates sharper edge as well as noticeable artifacts, partially because it is over-

fitted to the lens zooming degradation type in the RealSR dataset. The proposed BSR-RW can effectively and stably recover sharp image edges with little artifacts.

5.4 Conclusions

In this chapter we proposed a novel blind super-resolution (BSR) method for real-world images, namely BSR-RW, which is capable of handling unknown and spatially variant image degradations. Different from the costly iterative scheme in previous BSR methods, we designed a one-stage degradation estimation branch and a degradation-aware SISR branch for adaptive super-resolution. We also leveraged the guidance of edge map and used a pyramid U-shaped sub-network for fast and stable degradation estimation. Extensive experiments on both synthetic and real-world datasets showed that our BSR-RW achieved leading performance quantitatively and qualitatively, recovering sharp edges and details without introducing much artifacts.

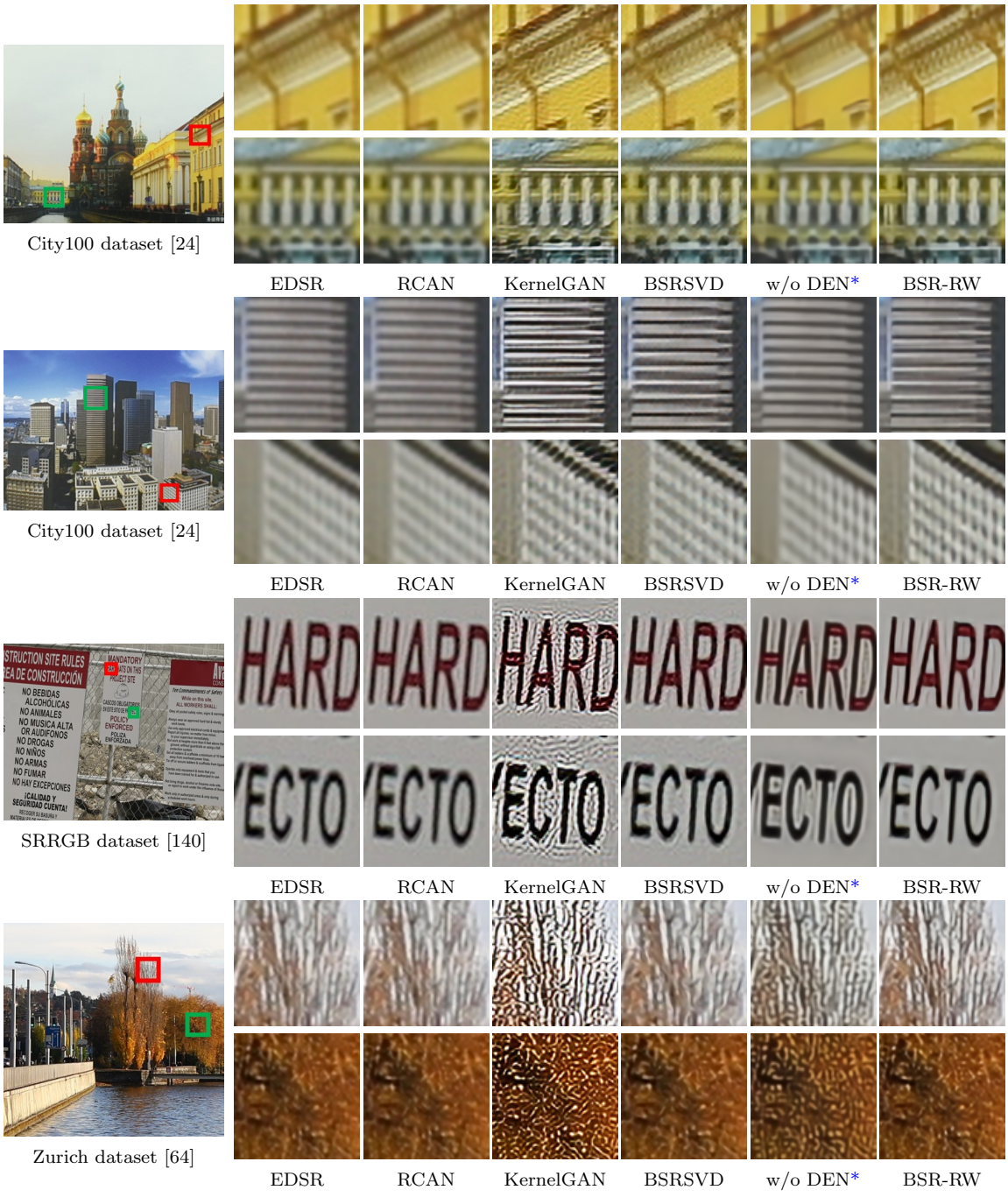


Figure 5.5: Visual comparison of competing methods for $\times 2$ SR on real-world images.

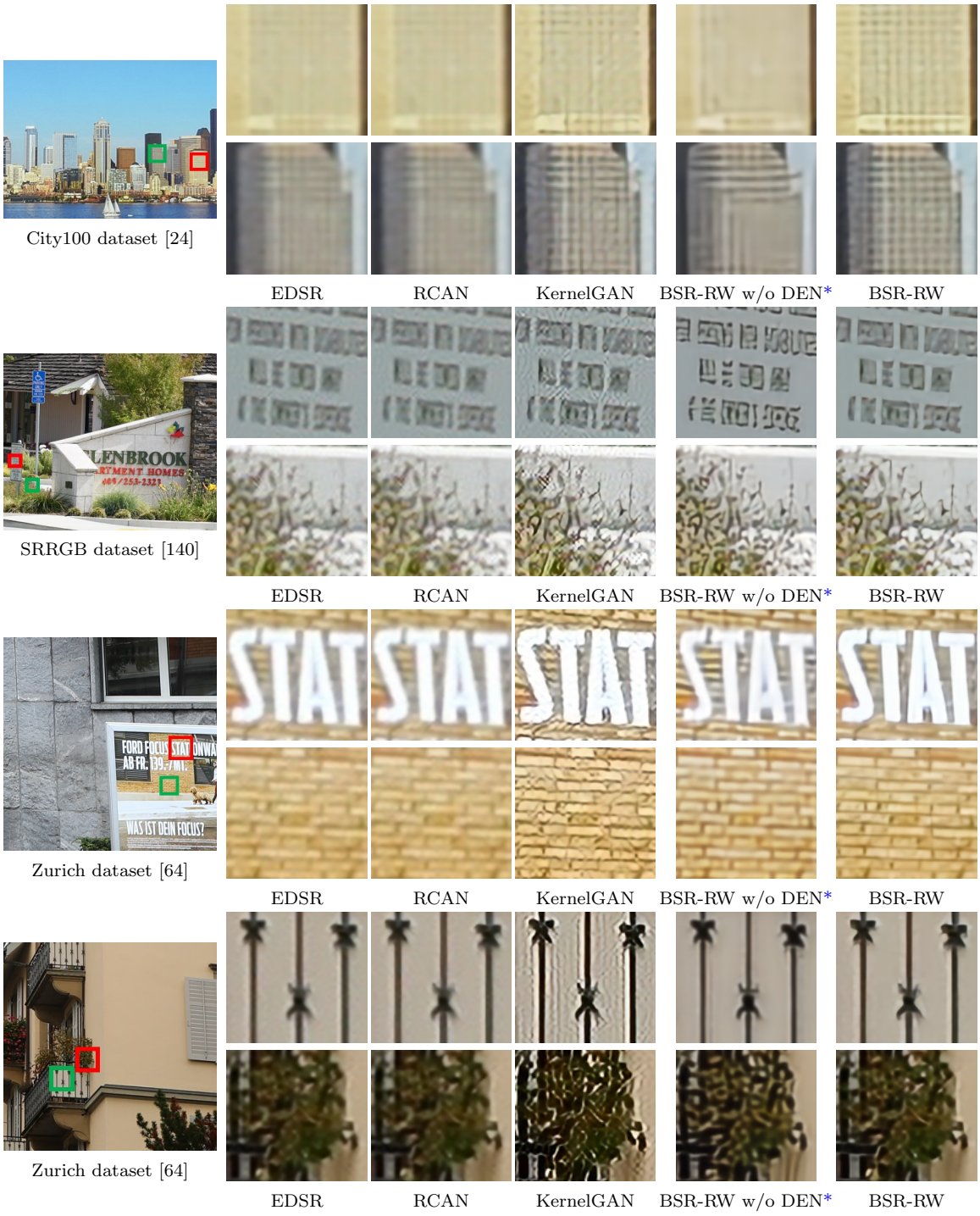


Figure 5.6: Visual comparison of competing methods for $\times 3$ SR on real-world images.

Figure 5.7: Visual comparison of competing methods for $\times 4$ SR on real-world images.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Color constancy and image enhancement play a significant role in image processing area, to reconstruct the original scene and further deliver high quality images. Recently both these two fields have been largely improved by leveraging deep convolutional neural network (CNN), owing to its powerful representation ability in exploiting the latent priors from large scale external datasets. Despite the great success achieved, challenges still exist. In this thesis, we investigate the task of color constancy, diffraction blur removal and single image super-resolution, and design efficient and robust algorithms by leveraging deep CNNs to improve their performances.

As the foremost unit in camera signal processing pipeline, color constancy aims to estimate the scene illumination and correct the color bias of the captured images. In the past several years, deep networks largely improve the color constancy accuracy by leveraging its powerful representation ability and annotated dataset. However, the acquisition of large scale annotated dataset is laborious and costly. This is especially true for color constancy which operates on the camera color space and as a result requires independent dataset for each camera due to the distinction in devices. In chapter 2, we start a pioneer work to leverage the multi-domain learning method for color constancy problem. Specifically, we utilized training data by different devices to

train a single model, to learn complementary representations and improve generalization capability. Experimental results show that with the proposed shareable modules and camera-specific module, our model achieves much better results than training independent model for each device, and also achieves state-of-the-art performance on three benchmark datasets. We also evaluate the color constancy performances under few-shot setting. Experimental results show that the proposed model can effectively adapt to a new device with only a few, e.g., 20, training samples.

General image deblurring (motion- and focal-blur) is a long-standing task in image enhancement area and has been widely studied for several decades. On the contrary, image diffraction blur removal receives relatively less study which however is a practical problem, largely degrading the image perceptual quality. In chapter 3, we studied the diffraction blur removal problem, for the first time, using a learning based method. We analyzed the characteristic of diffraction blur and clarified its difference from other types of image blurring problems. A real-world diffraction blur dataset with aligned image pairs was constructed for training and evaluating diffraction blur removal models. As far as we know, this is the first dataset of this kind. We also designed a progressive learning method and a robust loss function to train a diffraction blur removal model, which achieved significantly better performance than the general image deblurring methods in removing f diffraction blur. Lastly, we studied the generalization capability of trained model to other cameras and aperture sizes for use in practical applications.

Finally, we work on one of the fundamental task in image enhancement: single image super-resolution (SISR) task. By using the deep learning techniques, the SISR performance has been significantly improved. However, their generalization ability to real-world scenario is still limited, due to its high complexity. To tackle the real-world SISR problem, we develop two methods for in chapter 4 and chapter 5 respectively, from two different perspectives.

In chapter 4, we tackle the generalization problem of real-world SISR models by synthesizing realistic training image pairs, to diminish the domain gap between synthetic and authentic degradation models in SISR. To achieve this goal, we first learned an image degradation model from real-world SISR image pairs. Specifically, we learned a set of basis degradation kernels together with a weight prediction network. The degradation kernel at any location was estimated as the linear combination of the basis kernels using the weights predicted by the weight prediction network. The learned degradation model was then used to synthesize a large number of realistic image pairs covering various scenes for SISR model training. Our extensive analyses and experiments showed that the proposed degradation model learning method can effectively improve the generalization performance of SISR models to real-world applications.

In chapter 5 we handle the complex real-world SISR from a different angle. Different from the strategy of synthesizing realistic training dataset used in chapter 4, we design a novel blind super-resolution (BSR) method which is capable of handling unknown and spatially variant image degradations, although trained on merely synthetic dataset. This is achieved by the idea of first estimating local degradation of the given image, and then adaptively performing SISR, and the complex degradation model used, including blurring, noise and compression. Specifically, different from the costly iterative scheme in previous BSR methods, we designed a one-stage degradation estimation branch and a degradation-aware SISR branch for adaptive super-resolution. The fast and robust degradation estimation is achieved by leveraging the guidance of edge map and a pyramid U-shaped branch. Extensive experiments on both synthetic and real-world datasets showed that our method achieved leading performance quantitatively and qualitatively, recovering sharp edges and details without introducing much artifacts.

6.2 Future Work

The proposed methods in this thesis advance much the performance of color constancy and image enhancement. In future work, we will expand our study from the following perspectives:

- Due to the distinctions in spectral sensitivities of different sensors, the captured raw images are in different camera color spaces. To convert them to the common color space for display, current camera processing pipeline uses two successive units, i.e., first applying white balance and then color space transformation (CST). Such a divide-and-conquer strategy neglects the correlations between these two tasks and accumulates errors. In the future, we will study to tackle these two tasks in a unified framework, to take benefits from end-to-end training and reduce the introduction of cumulative errors.

- Our collected real-world diffraction blur dataset consists only 333 scenes and are captured by merely two digital cameras. In the future, we will extend the database by collecting more image pairs covering more scene varieties and using more types of cameras. And currently we use different models for different aperture sizes. We will study training one single model to handle various degrees of diffraction blur to further improve the generalization performance.

- Video super-resolution (VSR) is another important application, which also suffers from the deviation in degradation model between training and real-world scenarios. Unfortunately, unlike SISR, the aligned real-world VSR dataset is difficult to collect due to the motion among multiple frames. In the future, we plan to apply our degradation model learned from SISR on video sequence to generate realistic SISR videos. We will then train VSR-CNN using realistic dataset to improve the generalization ability on real-world VSR task.

- Currently, our methods are trained using the L_2 norm as loss function to decrease the distance between super-resolved images and the corresponding ground truth ones. The L_2 loss achieves a prominent PSNR index however can hardly generate hallucinated high-frequency details. In the future, we will study the using of perceptual loss or Generative Adversarial network (GAN) to generate more vivid realistic textures to improve the visual quality.

We will investigate these research directions in our future work.

Bibliography

[1] `https://github.com/yuanjunchai/IKC`.

[2] `https://github.com/sunreef/BlindSR`.

[3] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

[4] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 126–135, 2017.

[5] John M Aletta and Lloyd A Greene. Growth cone configuration and advance: a time-lapse study using video-enhanced differential interference contrast microscopy. *Journal of Neuroscience*, 8(4):1425–1435, 1988.

[6] Matthew R Arnison, Kieran G Larkin, Colin JR Sheppard, Nicholas I Smith, and Carol J Cogswell. Linear phase imaging using differential interference contrast microscopy. *Journal of microscopy*, 214(1):7–12, 2004.

[7] Nikola Banić, Karlo Koščević, Marko Subašić, and Sven Lončarić. Crop: Color constancy benchmark dataset generator. *arXiv preprint arXiv:1903.12581*, 2019.

[8] Nikola Banić and Sven Lončarić. Unsupervised learning for color constancy. *arXiv preprint arXiv:1712.00436*, 2017.

[9] Kobus Barnard. Improvements to gamut mapping colour constancy algorithms. In *European conference on computer vision*, pages 390–403. Springer, 2000.

[10] Jonathan T Barron. Convolutional color constancy. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 379–387, 2015.

[11] Jonathan T Barron and Yun-Ta Tsai. Fast fourier color constancy. In *IEEE Conf. Comput. Vis. Pattern Recognit*, 2017.

[12] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[13] Sefi Bell-Kligler, Assaf Shocher, and Michal Irani. Blind super-resolution kernel estimation using an internal-gan. In *Advances in Neural Information Processing Systems*, pages 284–293, 2019.

[14] Simone Bianco, Claudio Cusano, and Raimondo Schettini. Color constancy using cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 81–89, 2015.

[15] Simone Bianco, Claudio Cusano, and Raimondo Schettini. Single and multiple illuminant estimation using convolutional neural networks. *IEEE Transactions on Image Processing*, 26(9):4347–4362, 2017.

[16] Max Born and Emil Wolf. *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. Elsevier, 2013.

[17] David H Brainard and Brian A Wandell. Analysis of the retinex theory of color vision. *JOSA A*, 3(10):1651–1661, 1986.

[18] Gershon Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin institute*, 310(1):1–26, 1980.

[19] Adrian Bulat, Jing Yang, and Georgios Tzimiropoulos. To learn image super-resolution, use a gan to learn how to do image degradation first. In *Proceedings of the European conference on computer vision (ECCV)*, pages 185–200, 2018.

[20] Jianrui Cai, Shuhang Gu, Radu Timofte, and Lei Zhang. Ntire 2019 challenge on real image super-resolution: Methods and results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[21] Jianrui Cai, Hui Zeng, Hongwei Yong, Zisheng Cao, and Lei Zhang. Toward real-world single image super-resolution: A new benchmark and a new model. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3086–3095, 2019.

[22] Ayan Chakrabarti, Keigo Hirakawa, and Todd Zickler. Color constancy with spatio-spectral statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8):1509–1519, 2012.

[23] Subhasis Chaudhuri. *Super-resolution imaging*, volume 632. Springer Science & Business Media, 2001.

[24] Chang Chen, Zhiwei Xiong, Xinmei Tian, Zheng-Jun Zha, and Feng Wu. Camera lens super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1652–1660, 2019.

[25] Chang Chen, Zhiwei Xiong, Xinmei Tian, Zheng-Jun Zha, and Feng Wu. Camera lens super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1652–1660, 2019.

[26] Liang Chen, Faming Fang, Tingting Wang, and Guixu Zhang. Blind image deblurring with local maximum gradient prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1742–1750, 2019.

[27] Dongliang Cheng, Dilip K Prasad, and Michael S Brown. Illuminant estimation for color constancy: why spatial-domain methods work and the role of the color distribution. *JOSA A*, 31(5):1049–1058, 2014.

[28] Dongliang Cheng, Brian Price, Scott Cohen, and Michael S Brown. Effective learning-based illuminant estimation using simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1000–1008, 2015.

[29] Victor Cornillere, Abdelaziz Djelouah, Wang Yifan, Olga Sorkine-Hornung, and Christopher Schroers. Blind image super-resolution with spatially variant degradations. *ACM Transactions on Graphics (TOG)*, 38(6):1–13, 2019.

[30] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11065–11074, 2019.

[31] Arnold Jan Den Dekker and A Van den Bos. Resolution: a survey. *JOSA A*, 14(3):547–557, 1997.

[32] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[33] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2016.

[34] Weisheng Dong, Lei Zhang, Guangming Shi, and Xin Li. Nonlocally centralized sparse representation for image restoration. *IEEE transactions on Image Processing*, 22(4):1620–1630, 2012.

[35] Weisheng Dong, Lei Zhang, Guangming Shi, and Xin Li. Nonlocally centralized sparse representation for image restoration. *IEEE Transactions on Image Processing*, 22(4):1620–1630, 2013.

[36] Weisheng Dong, Lei Zhang, Guangming Shi, and Xiaolin Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Transactions on image processing*, 20(7):1838–1857, 2011.

[37] Somia Mostafa El-Hefnawy and Abed Nasr. Mathematical modeling of human eye retina for solving edge-detection problem. In *Parallel and Distributed Methods for Image Processing III*, volume 3817, pages 146–157. International Society for Optics and Photonics, 1999.

[38] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006.

[39] Nicholas Fang, Hyesog Lee, Cheng Sun, and Xiang Zhang. Sub–diffraction-limited optical imaging with a silver superlens. *Science*, 308(5721):534–537, 2005.

[40] Graham D Finlayson. Corrected-moment illuminant estimation. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1904–1911. IEEE, 2013.

[41] Graham D Finlayson and Elisabetta Trezzi. Shades of gray and colour constancy. In *Color and Imaging Conference*, volume 2004, pages 37–41. Society for Imaging Science and Technology, 2004.

[42] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

[43] Manuel Fritsche, Shuhang Gu, and Radu Timofte. Frequency separation for real-world super-resolution. *arXiv preprint arXiv:1911.07850*, 2019.

[44] Brian Funt and Weihua Xiong. Estimating illumination chromaticity via support vector regression. In *Color and Imaging Conference*, volume 2004, pages 47–52. Society for Imaging Science and Technology, 2004.

[45] Shao-Bing Gao, Ming Zhang, Chao-Yi Li, and Yong-Jie Li. Improving color constancy by discounting the variation of camera spectral sensitivity. *JOSA A*, 34(8):1448–1462, 2017.

[46] Peter Vincent Gehler, Carsten Rother, Andrew Blake, Tom Minka, and Toby Sharp. Bayesian color constancy revisited. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[47] Arjan Gijsenij and Theo Gevers. Color constancy using natural image statistics and scene semantics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):687–698, 2011.

[48] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[49] Amit Goldstein and Raanan Fattal. Blur-kernel estimation from spectral irregularities. In *European Conference on Computer Vision*, pages 622–635. Springer, 2012.

[50] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[51] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[52] Jinjin Gu, Hannan Lu, Wangmeng Zuo, and Chao Dong. Blind super-resolution with iterative kernel correction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1604–1613, 2019.

[53] Shuhang Gu, Shi Guo, Wangmeng Zuo, Yunjin Chen, Radu Timofte, Luc Van Gool, and Lei Zhang. Learned dynamic guidance for depth image reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[54] Shuhang Gu, Yawei Li, Luc Van Gool, and Radu Timofte. Self-guided network for fast image denoising. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2511–2520, 2019.

[55] Shuhang Gu, Yawei Li, Luc Van Gool, and Radu Timofte. Self-guided network for fast image denoising. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2511–2520, 2019.

[56] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2862–2869, 2014.

[57] Shuhang Gu, Wangmeng Zuo, Qi Xie, Deyu Meng, Xiangchu Feng, and Lei Zhang. Convolutional sparse coding for image super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1823–1831, 2015.

[58] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1712–1722, 2019.

[59] Zhen Han, Enyan Dai, Xu Jia, Shuaijun Chen, Chunjing Xu, Jianzhuang Liu, and Qi Tian. Unsupervised image super-resolution with an indirect supervised path. *arXiv preprint arXiv:1910.02593*, 2019.

[60] James L Harris Sr. Information extraction from diffraction limited imagery. *Pattern Recognition*, 2(2):69–77, 1970.

[61] Aaron Philip Hertzmann. *Algorithms for rendering in artistic styles*. PhD thesis, New York University, Graduate School of Arts and Science, 2001.

[62] Yuanming Hu, Baoyuan Wang, and Stephen Lin. Fc 4: Fully convolutional color constancy with confidence-weighted pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4085–4094, 2017.

[63] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[64] Andrey Ignatov, Luc Van Gool, and Radu Timofte. Replacing mobile camera isp with a single deep learning model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 536–537, 2020.

[65] Xixi Jia, Sanyang Liu, Xiangchu Feng, and Lei Zhang. Focnet: A fractional optimal control network for image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6054–6063, 2019.

[66] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *Advances in neural information processing systems*, pages 667–675, 2016.

[67] Hamid Reza Vaezi Joze and Mark S Drew. Exemplar-based color constancy and multiple illumination. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):860–873, 2014.

[68] Hakki Can Karaimer and Michael S Brown. A software platform for manipulating the camera imaging pipeline. In *European Conference on Computer Vision*, pages 429–444. Springer, 2016.

[69] Satoshi Kawata, Yasushi Inouye, and Prabhat Verma. Plasmonics for near-field nano-imaging and superlensing. *Nature photonics*, 3(7):388, 2009.

[70] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016.

[71] Stefan Kindermann, Stanley Osher, and Peter W Jones. Deblurring and denoising of images by nonlocal functionals. *Multiscale Modeling & Simulation*, 4(4):1091–1115, 2005.

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

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

[74] Thomas Köhler, M Batz, Farzad Naderi, et al. Bridging the simulated-to-real gap: benchmarking super-resolution on real data. *Arxiv: 180906420 [Cs]*, 2018.

[75] Thomas Köhler, Michel Bätz, Farzad Naderi, André Kaup, Andreas Maier, and Christian Riess. Toward bridging the simulated-to-real gap: Benchmarking super-resolution on real data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[76] Jan Kotera, Filip Šroubek, and Peyman Milanfar. Blind deconvolution using alternating maximum a posteriori estimation with heavy-tailed priors. In *International Conference on Computer Analysis of Images and Patterns*, pages 59–66. Springer, 2013.

[77] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 624–632, 2017.

[78] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[79] Taihao Li, Huai Chen, Min Zhang, Shupeng Liu, Shunren Xia, Xinhua Cao, Geoffrey S Young, and Xiaoyin Xu. A new design in iterative image deblurring for improved robustness and performance. *Pattern recognition*, 90:134–146, 2019.

[80] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 136–144, 2017.

[81] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. In *Advances in Neural Information Processing Systems*, pages 1680–1689, 2018.

[82] Andreas Lugmayr, Martin Danelljan, and Radu Timofte. Unsupervised learning for real-world super-resolution. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3408–3416. IEEE, 2019.

[83] Andreas Lugmayr, Martin Danelljan, Radu Timofte, Manuel Fritsche, Shuhang Gu, Kuldeep Purohit, Praveen Kandula, Maitreya Suin, AN Rajagoapalan, Nam Hyung Joon, et al. Aim 2019 challenge on real-world image super-resolution: Methods and results. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3575–3583. IEEE, 2019.

[84] Kede Ma, Zhengfang Duanmu, Qingbo Wu, Zhou Wang, Hongwei Yong, Hongliang Li, and Lei Zhang. Waterloo exploration database: New challenges for image quality assessment models. *IEEE Transactions on Image Processing*, 26(2):1004–1016, 2016.

[85] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Non-local sparse models for image restoration. In *2009 IEEE 12th international conference on computer vision*, pages 2272–2279. IEEE, 2009.

[86] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Non-local sparse models for image restoration. In *2009 IEEE 12th international conference on computer vision*, pages 2272–2279. IEEE, 2009.

[87] Scott Marshall Mansfield and GS Kino. Solid immersion microscope. *Applied physics letters*, 57(24):2615–2616, 1990.

[88] David Marr and Ellen Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 207(1167):187–217, 1980.

[89] Steven McDonagh, Sarah Parisot, Zhenguo Li, and Gregory Slabaugh. Meta-learning for few-shot camera-adaptive color constancy. *arXiv preprint arXiv:1811.11788*, 2018.

[90] Tomer Michaeli and Michal Irani. Nonparametric blind super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 945–952, 2013.

[91] Ben Mildenhall, Jonathan T Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. Burst denoising with kernel prediction networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2502–2510, 2018.

[92] María S Millán and Edison Valencia. Color image sharpening inspired by human vision models. *Applied Optics*, 45(29):7684–7697, 2006.

[93] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[94] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang. Blind image deblurring using dark channel prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1628–1636, 2016.

[95] Sung Cheol Park, Min Kyu Park, and Moon Gi Kang. Super-resolution image reconstruction: a technical overview. *IEEE signal processing magazine*, 20(3):21–36, 2003.

[96] Juncai Peng, Yuanjie Shao, Nong Sang, and Changxin Gao. Joint image deblurring and matching with feature-based sparse representation prior. *Pattern Recognition*, page 107300, 2020.

[97] Alex Paul Pentland. A new sense for depth of field. *IEEE transactions on pattern analysis and machine intelligence*, (4):523–531, 1987.

[98] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1586–1595, 2017.

[99] Dieter W Pohl and Daniel Courjon. *Near field optics*, volume 242. Springer Science & Business Media, 2012.

[100] Chengchao Qu, Ding Luo, Eduardo Monari, Tobias Schuchert, and Jürgen Beyerer. Capturing ground truth super-resolution data. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 2812–2816. IEEE, 2016.

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

[102] Yaniv Romano, John Isidoro, and Peyman Milanfar. Raisr: Rapid and accurate image super resolution. *IEEE Transactions on Computational Imaging*, 3(1):110–125, 2016.

[103] Christian J Schuler, Michael Hirsch, Stefan Harmeling, and Bernhard Schölkopf. Learning to deblur. *IEEE transactions on pattern analysis and machine intelligence*, 38(7):1439–1451, 2016.

[104] Lilong Shi. Re-processed version of the gehler color constancy dataset of 568 images. *http://www. cs. sfu. ca/˜ color/data/*, 2000.

[105] Wenzhe Shi, Jose Caballero, Ferenc Huszár, 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.

[106] Wu Shi, Chen Change Loy, and Xiaoou Tang. Deep specialized network for illuminant estimation. In *European Conference on Computer Vision*, pages 371–387. Springer, 2016.

[107] Jian Sun, Wenfei Cao, Zongben Xu, and Jean Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 769–777, 2015.

[108] Camera Lens Super-Resolution. Component divide-and-conquer for real-world image super-resolution.

[109] Ying Tai, Jian Yang, and Xiaoming Liu. Image super-resolution via deep recursive residual network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3147–3155, 2017.

[110] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8174–8182, 2018.

[111] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8174–8182, 2018.

[112] Kinh Tieu and Erik G Miller. Unsupervised color constancy. In *Advances in neural information processing systems*, pages 1327–1334, 2003.

[113] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and

results. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 114–125, 2017.

[114] Kei Tomimatsu. Fluid immersion microscope objective lens, September 8 1998. US Patent 5,805,346.

[115] Joost Van De Weijer, Theo Gevers, and Arjan Gijsenij. Edge-based color constancy. *IEEE Transactions on image processing*, 16(9):2207–2214, 2007.

[116] J von Kries. Chromatic adaptation, festschrift der albercht-ludwig-universität, 1902.

[117] He Wang, Yuanming Feng, Yu Sa, Jun Q Lu, Junhua Ding, Jun Zhang, and Xin-Hua Hu. Pattern recognition and classification of two cancer cell lines by diffraction imaging at multiple pixel distances. *Pattern Recognition*, 61:234–244, 2017.

[118] Qiang Wang, Xiaoou Tang, and Harry Shum. Patch based blind image super resolution. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 709–716. IEEE, 2005.

[119] Shenlong Wang, Lei Zhang, and Yan Liang. Nonlocal spectral prior model for low-level vision. In *Asian Conference on Computer Vision*, pages 231–244. Springer, 2012.

[120] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[121] wesaturate. Photo sharing. `http://www.wesaturate.com`, 2016.

[122] Wikipedia contributors. Aperture — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Aperture&oldid=887068083`. [Online; accessed 16-March-2019].

[123] Wikipedia contributors. Diffraction — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/w/index.php?title=Diffraction&oldid=888294170`. [Online; accessed 16-March-2019].

[124] Wikipedia contributors. Airy disk — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/wiki/Airy_disk`, 2019. [Online; accessed 16-March-2019].

[125] Shiming Xiang, Gaofeng Meng, Ying Wang, Chunhong Pan, and Changshui Zhang. Image deblurring with matrix regression and gradient evolution. *Pattern Recognition*, 45(6):2164–2179, 2012.

[126] Jun Xu, Lei Zhang, Wangmeng Zuo, David Zhang, and Xiangchu Feng. Patch group based nonlocal self-similarity prior learning for image denoising. In *Proceedings of the IEEE international conference on computer vision*, pages 244–252, 2015.

[127] Li Xu and Jiaya Jia. Two-phase kernel estimation for robust motion deblurring. In *European conference on computer vision*, pages 157–170. Springer, 2010.

[128] Li Xu, Shicheng Zheng, and Jiaya Jia. Unnatural l0 sparse representation for natural image deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1107–1114, 2013.

[129] Li Xu, Shicheng Zheng, and Jiaya Jia. Unnatural l0 sparse representation for natural image deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1107–1114, 2013.

[130] Yanyang Yan, Wenqi Ren, Yuanfang Guo, Rui Wang, and Xiaochun Cao. Image deblurring via extreme channels prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4003–4011, 2017.

[131] Chih-Yuan Yang, Chao Ma, and Ming-Hsuan Yang. Single-image super-resolution: A benchmark. In *European Conference on Computer Vision*, pages 372–386. Springer, 2014.

[132] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010.

[133] Wenming Yang, Xuechen Zhang, Yapeng Tian, Wei Wang, Jing-Hao Xue, and Qingmin Liao. Deep learning for single image super-resolution: A brief review. *IEEE Transactions on Multimedia*, 21(12):3106–3121, 2019.

[134] Frederic Zenhausern, MP O'boyle, and HK Wickramasinghe. Apertureless near-field optical microscope. *Applied Physics Letters*, 65(13):1623–1625, 1994.

[135] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

[136] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3262–3271, 2018.

[137] Lei Zhang and Xiaolin Wu. An edge-guided image interpolation algorithm via directional filtering and data fusion. *IEEE transactions on Image Processing*, 15(8):2226–2238, 2006.

[138] Xiang Zhang and Zhaowei Liu. Superlenses to overcome the diffraction limit. *Nature materials*, 7(6):435, 2008.

[139] Xuaner Zhang, Qifeng Chen, Ren Ng, and Vladlen Koltun. Zoom to learn, learn to zoom. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3762–3770, 2019.

[140] Xuaner Zhang, Qifeng Chen, Ren Ng, and Vladlen Koltun. Zoom to learn, learn to zoom. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3762–3770, 2019.

[141] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 286–301, 2018.

[142] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2472–2481, 2018.

[143] Wangmeng Zuo, Dongwei Ren, Shuhang Gu, Liang Lin, and Lei Zhang. Discriminative learning of iteration-wise priors for blind deconvolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3232–3240, 2015.

[144] Wangmeng Zuo, Dongwei Ren, David Zhang, Shuhang Gu, and Lei Zhang. Learning iteration-wise generalized shrinkage–thresholding operators for blind deconvolution. *IEEE Transactions on Image Processing*, 25(4):1751–1764, 2016.