Image Restorations Using Deep Learning Techniques

IMAGE RESTORATIONS USING DEEP LEARNING TECHNIQUES

BY

ZHIXIANG CHI, B.Eng.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

Master of Applied Science (2018) McMaster University

(Electrical & Computer Engineering) Hamilton, Ontario, Canada

TITLE:	Image Restorations Using Deep Learning Techniques
AUTHOR:	Zhixiang Chi B.Eng., (Electrical and Computer Engineering) McMaster University, Hamilton,Ontario,Canada
SUPERVISOR:	Dr. Xiaolin Wu
NUMBER OF PAGES:	xi, 63

To my family and friends

Abstract

Conventional methods for solving image restoration problems are typically built on an image degradation model and on some priors of the latent image. The model of the degraded image and the prior knowledge of the latent image are necessary because the restoration is an ill posted inverse problem. However, for some applications, such as those addressed in this thesis, the image degradation process is too complex to model precisely; in addition, mathematical priors, such as low rank and sparsity of the image signal, are often too idealistic for real world images. These difficulties limit the performance of existing image restoration algorithms, but they can be, to certain extent, overcome by the techniques of machine learning, particularly deep convolutional neural networks. Machine learning allows large sample statistics far beyond what is available in a single input image to be exploited. More importantly, the big data can be used to train deep neural networks to learn the complex non-linear mapping between the degraded and original images. This circumvents the difficulty of building an explicit realistic mathematical model when the degradation causes are complex and compounded.

In this thesis, we design and implement deep convolutional neural networks (DCNN) for two challenging image restoration problems: reflection removal and joint demosaicking-deblurring. The first problem is one of blind source separation; its DCNN solution requires a large set of paired clean and mixed images for training. As these paired

training images are very difficult, if not impossible, to acquire in the real world, we develop a novel technique to synthesize the required training images that satisfactorily approximate the real ones. For the joint demosaicking-deblurring problem, we propose a new multiscale DCNN architecture consisting of a cascade of subnetworks so that the underlying blind deconvolution task can be broken into smaller subproblems and solved more effectively and robustly. In both cases extensive experiments are carried out. Experimental results demonstrate clear advantages of the proposed DCNN methods over existing ones.

Acknowledgements

I would like to take the special opportunity to thank all the people who supported me for the last two years. Foremost, I would like to express my sincerest gratitude to my supervisor Dr. Xiaolin Wu for his consistent guidance and support of my M.A.Sc research. His wide knowledge and kind encouragement inspired me in my research and writing of this thesis. His kindness is highly appreciated and cherished in the future.

Besides, I would like to thank Dr. Xiao Shu for his valuable suggestions that keep me stay at the right path. I could not imagine the difficulties to break through the dilemmas without his help. For this thesis, I also would like to thank Dr. Jiankang Zhang and Dr. Jun Chen for being the committee members. Their time and effort on reviewing my thesis are appreciated.

Last but not the least, I would like to thank my parents for their selfless love and support, and also to my friends who have been supporting me throughout the last two years. To them, I dedicate this thesis.

Notation and abbreviations

DCNN: deep convolutional neural network

GAN: generative adversarial network

ResNet: residual network

MSE: mean square error

ReLU: rectified linear unit

Contents

List of Figures

Chapter 1

Introduction

Image restoration is a process to recover high quality clean images from their degraded counterparts. The degradation can have a variety of causes, such as sensor noises, downsampling, motion blur, reflection and some combinations of the above. Restoring an input image to its best form is a necessary and important step in many applications of image processing and low-level computer vision. There is a large body of literature of image restoration, covering techniques of denoising, superresolution, deblurring, demosaicking, reflection removal, etc.

Main stream image restoration methods are based on image degradation models and on some priors of the latent image. However, due to the complexity of real-world image degradation, it is often extremely difficult to model the process precisely. In addition, image restoration is in general a severely underdetermined inverse problem, as a large number of different latent images can degrade into the same image. Solving the inverse problem requires additional priors, such as low rank and sparsity of the image signal, to restrict the solution space. However, the priors are necessarily mathematical constructs that are often too idealistic for real-world images. These difficulties greatly limit the performance of the conventional image restoration algorithms.

The newly emerged data-driven machine learning method, particularly deep convolutional neural networks, open up new possibilities for more powerful and versatile solutions of image restoration. In machine learning one can use very large sample data set to make statistical inferences that is much more accurate and robust than relying only on the observations of a single input image. Also, big data facilitates the training of deep neural networks to learn highly complex non-linear mapping between the degraded and original images. This makes it possible to solve the inverse problem of image restoration without building an explicit realistic mathematical model. Indeed, past few years have witnessed significant improvements of image restoration results made by deep learning over traditional image processing methods.

Artificial neural network (ANN) is a widely used method of machine learning that mimics how the biological neural network processes information in human brain. It can solve difficult problems in diverged areas of applications, such as computer vision, natural language processing, text processing, etc. ANNs consist of layers of neurons, each neuron has a weight associated with it. A neuron is a basic computation unit; it multiplies the weight with its input, and computes the output with a non-linear activation function. The networks are optimized by back propagation algorithms to minimize a loss function, which is, in the case of image restoration, the difference between the restored and ground truth images.

During training, the network performs forward and backward propagation alternatively to evaluate the loss function and update the parameters. The loss function is the key of training and is carefully designed so that the training process can be seen as an optimization problem. Besides the loss function, training data also affects the performance of the network. Insufficient training data and statistically mismatched training data will cause over-fitting and non-convergence of the network. By increasing the number of hidden layers, the neural network can fit more complex functions

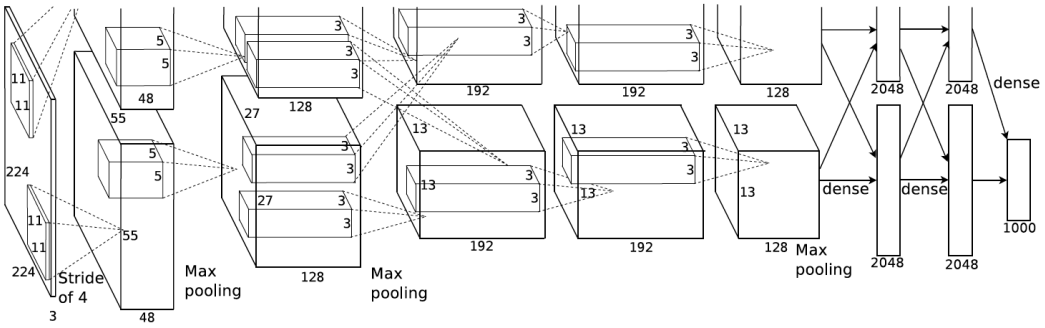


Figure 1.1: Architecture of Alex net.

and it also requires more training data.

With the invention of convolutional neural network (CNN), many attempts have been made to address different computer vision problems. CNNs are multi-layer structure neural network and they are aimed to extract complicated information from the input images. They are trained to learn high-level discriminative features in an incremental way for classifying structured data such as images and videos. CNN architecture consists of convolutional, nonlinear, pooling and fully connected layers. Each convolutional layer applies convolution on its input tensor with its filters, each filter generates feature maps with an non-linear activation function. Higher level convolutional layers extract more abstract features that contains discriminative information for making the final decision.

The first successful deep neural network is the well known AlexNet published in 2012 [9]. It made a breakthrough in visual object recognition and achieved unprecedented accuracy. The power of AlexNet largely comes from its adoption of convolutional kernels in first few layers of the network, thus this type of network is called convolutional neural network (CNN). Figure 1.1 depicts the AlexNet architecture.

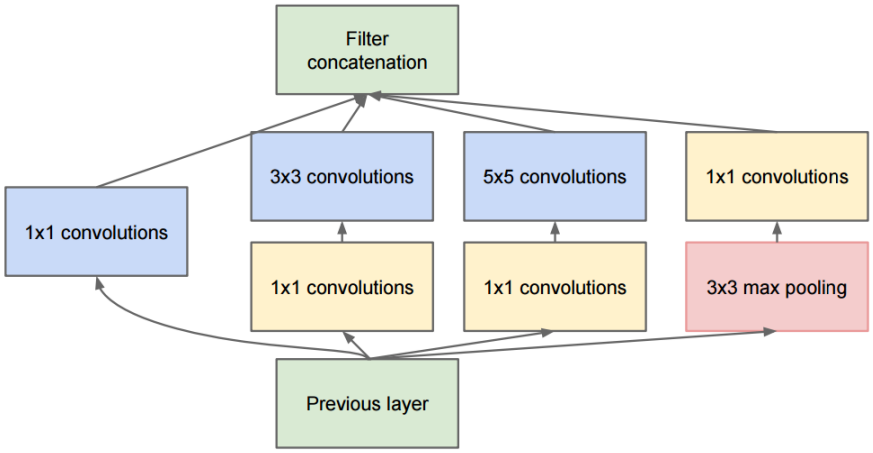


Figure 1.2: Building block of GoogLeNet [1].

It consists of five stacked convolutional layers followed by two fully connected layers. Convolutional layers extract features, while fully connected layers estimate the probabilities of the output classes based on the features.

Shortly after the AlexNet, deep learning has become an extremely active research field, drawing attentions from almost all academic disciplines. Thanks to the rapid advances of high performance computation hardware, especially GPUs, larger and larger network models can be built. Many researchers have been trying to make CNNs more powerful by increasing the network depth and width. Based on AlexNet, VGG net is proposed [10] that increases the number of convolutional layers while using smaller 3×3 kernels. The increase in network depth boosts the performance significantly, reducing the classification error by a large margin.

In GoogLeNet [1] the authors tried to increase the depth and width of the network. In order to prevent data overfitting and prevent excessive uses of memory and power due to the great depth, the authors use 1×1 filters to reduce the problem dimensionality. The GoogLeNet is stacked by the building blocks as shown in Figure 1.2. Even through GoogLeNet has a deeper and wilder architecture, but it contains far fewer

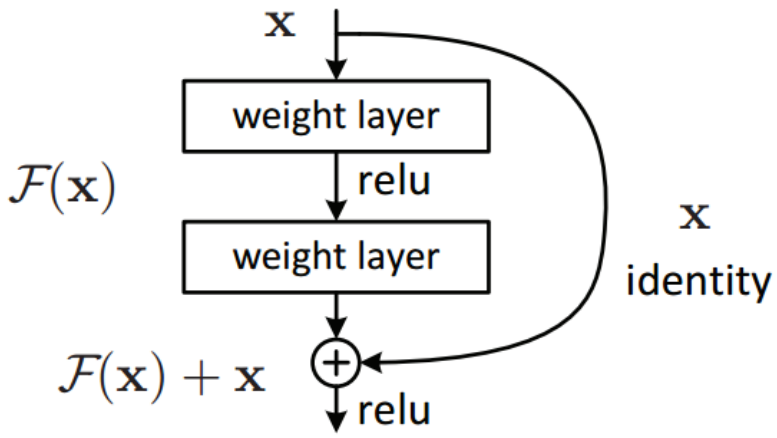


Figure 1.3: Residual block [2].

parameters.

Obviously, the increase in network depth gives better performance, but new issues have emerged. As the network depth increases, it becomes harder to train because of the notorious gradient vanishing problem. He et al. [2] claimed that as the network is going deeper, accuracy gets saturated and then suddenly degrades. To solve the problem, they proposed a residual block as shown in Figure 1.3. The input of the residual block x goes through convolution-rectified linear unit-convolution sequentially to learn the residual $F(x)$. The output is obtained by the addition of the residual and its input as $F(x) + x$. The authors stacked the residual blocks to build a 152-layer network that reduced the error rate significantly. The reason for why the network can go deeper is that the gradient can easily flow through the identity mapping during the back propagation process.

In addition to the successes of deep learning techniques in image recognition, image restoration can also be performed by CNN-based deep learning with the quality of restored images far exceeding that of traditional methods. A straightforward way of deep learning for image restoration is to remove the fully connected layers of a CNN

classification network. Early methods simply stack convolution layers to perform mapping between degraded and latent images with respect to a specific restoration problem [11]. A more effective approach is to use deep convolutional encoder-decoder networks, first proposed by Mao et al. [12]. In many applications, the degraded image and its clean version are very close, it will be more efficient to only learn the difference between them. Hence, stacking residual blocks becomes a popular technique and it generates superior results [13, 5].

For supervised image restoration algorithms, CNNs are commonly optimized to minimize the mean squared error (MSE) between restored image and its ground truth. However, algorithms optimized by MSE tend to average possible solutions hence limit the ability to capture the significant high-frequency information. With the invention of Generative Adversarial Network (GAN) [14], it has been applied to generate photo-realistic images [13]. The generator network is aimed to learn the true data distribution and is encouraged to produce images towards to natural images. It will be penalized by a discriminator network if the restored image is not a sample from the ground truth distribution. Even though the generated images sacrifice the pixel-wise measurements such as PSNR and SSIM, they are more perceptually satisfying.

In this thesis, we design and implement deep convolutional neural networks (DCNN) for two challenging image restoration problems: single reflection removal and joint demosaicking-deblurring. The first problem is one of blind source separation. Since the DCNN solution requires a large-scale data for training and gathering paired degradated and clean images is not easy in real world, we develop a novel technique to synthesize the required training images that satisfactorily approximate the real ones. For the joint demosaicking-deblurring problem, we propose a new multiscale DCNN architecture consisting of a cascade of subnetworks so that the underlying blind deconvolution task can be broken into smaller subproblems and solved more effectively

and robustly. In both cases extensive experiments are carried out. Experimental results demonstrate although the neural networks learn only from synthetic data, the proposed techniques generalize well on real-world images, outperforming the other tested state-of-the-art techniques.

Chapter 2

Single image reflection removal

2.1 Introduction

Photographing a scene behind a transparent medium, most commonly glasses, tends to be interfered by the reflections of the objects on the side of the camera. The intended reflection-free image, which we call the transmission image T, becomes intertwined with the reflection image R, and is consequently recorded as a mixture image I. The reflections cause annoying image degradations of arguably the worst kind and make many computer vision tasks, such as segmentation, classification, recognition, etc., very difficult if not impossible. For a range of important applications, the separation and removal of reflection image R from the acquired mixture image I is a challenging image restoration task out of necessity.

A widely adopted and satisfactory model for the formation of the mixture image is

$$I = \alpha T + \beta R + n. \tag{2.1}$$

where n is the noise term, α and β are the transmittance and reflection rate of the

glass, respectively; they determine the mixing weights of the two component images. Compared with other image restoration tasks, such as denoising, superresolution, deblurring, etc., reflection removal is far more difficult. The underlying inverse problem is one of blind source separation and it is more severely underdetermined as there are not one but two unknown images T and R that need to be estimated from the observed image I. Adding to the level of difficulty is that both component signals T and R are natural images of similar statistics.

Many researchers have taken on the technical challenge of reflection removal and proposed a number of solutions for the problem. But the current state of the art is still quite limited in terms of the performance, robustness and generality. One approach is the use of specially designed optical devices, such as polarizing filters, to obtain a series of perfectly aligned images with different levels of reflections for layer separation [15]. Although such optical devices make the reflection removal problem easier to tackle, they incur additional hardware costs, reduce light influx, and have limited scope of applications. Thus, many techniques use multiple images of the targeted scene taken from slightly different viewing positions instead to get varied reflections [16, 17]. However, as these techniques require accurate image registration, they are only applicable when the imaged objects are relatively flat and not in motion.

Ideally, a reflection removal algorithm should work with a single mixture image, albeit a daunting task. Some attempts have been reported in the literature on single-image reflection removal [18, 19]. In these papers, the authors adopted the image formation model of Eq. (3.1) and formulated the problem as the decomposition of the observed mixture image I into two components T and R of different characteristics. In order to separate the reflection from the transmission, they all made some explicit assumptions about the reflection image R so it can be distinguished from the transmission image T. For instance, Shih et al. proposed to use the double image caused by

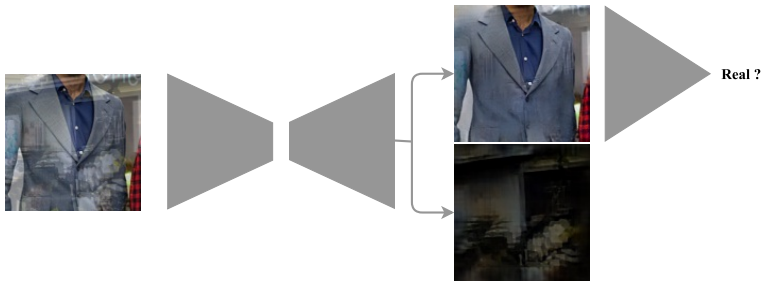


Figure 2.1: The proposed reflection removal technique separates reflections from a given reflection-interfered image (left) and outputs a clean image (upper right) without the reflection artifacts. The technique also employs adversarial loss to preserve the naturalness of the output.

the two surfaces of the glass to identify reflection [20]. But unless the camera is close to the glass, the double image effect is too insignificant to be useful. Other techniques also try to exploit the smoothness and sparsity of the reflection layer [21]. However, transmission image can have smooth and sparse regions as well, making these priors non-discriminative.

Apparently, the task of single image reflection removal has been greatly hindered by the inability of conventional statistical models to separate reflection and transmission images. We humans can, on the other hand, mentally separate the two images although being visually disturbed. The difference lies in that humans can perform the separation task largely relying on the coherence of high-level semantics of the two images, while existing models cannot. This suggests that machine learning is a sensible and promising strategy for overcoming the persistent difficulties in removing reflections based on a single mixture image. Furthermore, the machine learning approach, with properly chosen training data, can also circumvent the obstacle of grossly ill conditioning in solving the problem directly using the image formation model of Eq. (3.1), in which the number of unknowns greatly exceeds the number of equations.

In this paper, instead of using an explicit model like most existing techniques, we propose a data-driven approach based on deep convolutional neural network (CNN) for removing reflections in a single image as shown in Figure 2.1. Similar to many CNN based image restoration techniques for problems like inpainting, denoising and superresolution, the proposed approach recovers a reflection-free image from a given image by learning an end-to-end mapping of image pairs with and without reflections. To fully exploit the fact that the reflection image, the addictive "noise" to be removed, is also a natural image as discussed previously, we design a novel three-stage deep encoder-decoder network that first estimates the reflection layer and then reconstructs a high quality transmission layer based on the estimated reflection and perceptual merits. Due to the difficulty of obtaining a sufficiently large set of real image pairs for training the neural network, we carefully model the physical formation of reflection and synthesize a large number of photo-realistic reflection-tainted images from reflection-free images collected online. Extensive experimental results show that the neural network can generalize well using only synthesized training data and significantly outperform other tested techniques for real-world images.

2.2 Related Work

Many existing reflection removal methods rely on two or more input images of the same scene with different reflections to estimate the transmission layer. To obtain such images of varied reflections, several photography techniques can be used. Some reflection removal methods employ polarizing filter to manipulate the level of the reflections [22, 23, 24]. The physically-based method proposed by Schechner et al. shows the advantages of using orthogonal polarized input images [25]. Kong et al. further improve this idea by exploiting the spatial properties of polarization [15].

Similar to polarizing filtering, flash lighting [26, 27] and defocus blurring [28, 29] can help generate images with varied reflections without moving the camera as well. Keeping the camera relatively stationary is crucial to the efficacy of these device-based reflection removal techniques, as if the objects behind the glass are also not in motion, the only changing components among the input images are the reflections while the transmission layer is invariant.

There are also many multi-image techniques that exploit the motion of the camera as a cue for reflection removal. These techniques first align the objects in a series of images taken from slightly different viewing positions and then separate the invariant layer as the reflection-free image [30, 16]. To align images interfered by reflections, Tsin et at. [31] and Sinha et al. [32] use efficient stereo matching algorithms. Guo et al. exploit the sparsity and independence of the transmission and reflection layers to improve the robustness of image alignment [33]. Off-the-shelf optical flow algorithms are also employed for aligning images [17, 34]. With motion smoothness constraints, optical flow techniques can be more accurate and robust for the layer separation task [35, 36]. For the cases where the camera is stationary while the objects are moving, the reflection layer is relatively static and must be handled differently [37, 38].

If only one image of the scene is given, which is the case tackled by this thesis, the task of reflection removal becomes much more challenging. Only a few single image reflection removal techniques have been reported in the literature. Many of these techniques still rely on some extra information provided by light field camera [39, 40] or the user [41, 42]. One of the first attempts to solve the problem without any user assistance is [18], which minimizes the total amount of edges and corners in the two decomposed layers of the input image. Akashi et al. [43] employ sparse non-negative matrix factorization to separate the reflection layer without a explicit smoothness prior. The work of Li and Brown [21] assumes that the reflection layer is smoother

than the transmission layer due to defocus blur and hence has a short tail gradient distribution. With a similar smoothness assumption for the reflection layer, Fan et al. [19] use two cascaded CNN networks to reconstruct a reflection reduced image from the edges of the input image. Arvanitopoulos et al. [44] formulate the reflection suppression problem as an optimization problem with a Laplacian data fidelity term and a total variation term. Wan et al. [45] combine the sparsity prior and nonlocal prior of image patches in both the transmission and reflection layers together. They further increase the effectiveness of the nonlocal prior using image patches retrieved from an external dataset. The work of Shih et al. [20] takes advantage of ghosting, the phenomenon of multiple reflections caused by thicker glass, and decomposes the input image based on Gaussian mixture model (GMM). To deal with reflections from eyeglasses in frontal face image, Sandhan and Choi [46] exploit the bilateral symmetry of human face and use a mirrored input image as another input of varied reflections.

For more detailed review on the existing techniques for reflection removal, we refer readers to two excellent surveys [47] and [48].

2.3 Preparation of Training Data

The proposed technique recognizes and separates reflections from the input image using an end-to-end mapping trained by image pairs with and without reflections. The effectiveness of our technique, or any machine learning approaches, greatly relies on the availability of a representative and sufficiently large set of training data. In this section, we discuss the methods for collecting and preparing the training images for our technique.

To help the proposed technique identify the patterns of reflections in real-world scenarios, ideally, the training algorithm should only use real photographs as the

training data. Obtaining an image with real reflections is not difficult; we can capture such a mixture image I, as in Eq. (3.1), by placing a piece of reflective glass of transmittance α in front of the camera. The corresponding clean image T of the same scene is also attainable using the same camera setup but without the glass. However, training images collected using this scheme have several non-negligible drawbacks and limitations. First, it is almost impossible to get a pair of images that are perfectly aligned. Even with a tripod that stabilizes the camera, the motions of objects within the scene can still cause misalignment between two images captured consecutively.

Furthermore, due to the effects of refraction, the glass shielding the scene shifts the path of light transmitting through the glass and can also lead to the alignment problem. By the reflection formation model in Eq. (3.1), these differences introduced by the misalignment between the mixture image I and its reflection-free counterpart T can be seen as a part of the noise term n, where

$$\beta R + n = I - \alpha T. \tag{2.2}$$

Since the noise introduced by misalignment has similar characteristics as a natural image, it is difficult to accurately distinguish the noise n from the reflection image βR. As a result, a training algorithm could erroneously attribute part of the noise n as the effects of the reflection βR, interfering the learning of the true reflections. Similarly, regional illumination changes between a pair of images can lead to the increase of structural noise in training data as well. Although it is possible to reduce these adverse effects by carefully shooting only static scene from a stationary camera or using thinner reflective glass with small refraction, these methods greatly limit the flexibility and practicality of collecting real images as training data.

Due to the unavoidable limitations discussed above and the prohibitive cost of

building a large enough training set of real images, we use synthetic images constructed from images collected online for training instead. The main idea of the synthesis process follows the physical reflection formation model in Eq. (3.1), which interprets a reflection-interfered image I as the linear combination of two reflection-free natural images T, R. The formation model, however, cannot be applied directly to most of the JPEG-compressed images available online. The reason is that, to take advantage of human's non-linear light sensitivity, JPEG images have to be gamma corrected before being stored on camera, hence their pixel values are not linear to the light intensities captured by image sensor. Consequently, the direct summation of two gamma-corrected images does not conform the physics of light superposition as required by Eq. (3.1), resulting unrealistic reflection-interfered image. To correct this problem, we can either only use raw image or apply inverse gamma correction on the collected JPEG images, as follows

$$X = (X')^{1/\gamma}, \tag{2.3}$$

where X' is a gamma corrected image and X is the corresponding light intensity image. The gamma correction coefficient γ for each color channel is often available in exchangeable image file format (EXIF) segment attached in each JPEG image. In the following discussion, we still use the linear formula as in Eq. (3.1) and assume that all the pixel values are restored to the raw light intensity readings from image sensor.

To accurately simulate the formation of reflection-interfered image, we also consider the blur effect in the reflections. In many real images, the focal planes of the camera are on the objects behind the glass rather than the reflected objects, since what behind the reflective glass are normally the objects of interest. As a result, reflections are often blurry due to the defocus effect [21, 19]. To reflect this common

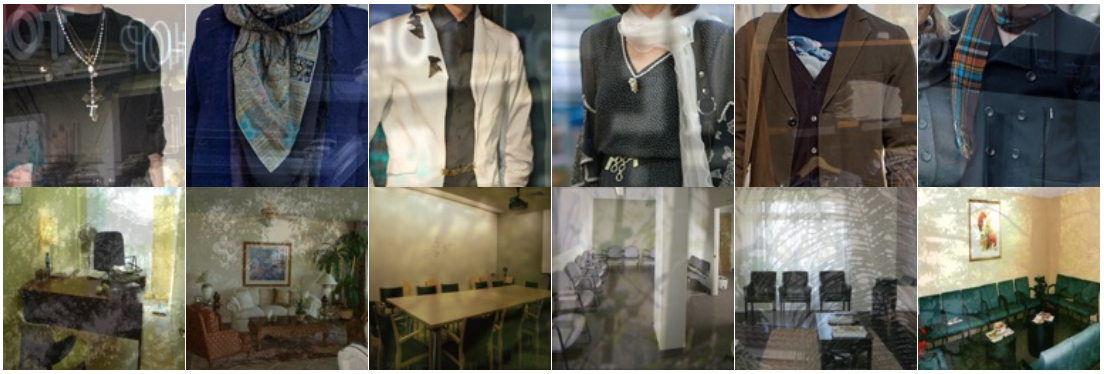


Figure 2.2: Samples of synthesized images. The synthetic training set contains images with blurred, sharp and double reflections.

effect in the training data, we blur some reflection image R with a Gaussian kernel G of random variance before superimposing it into a synthetic image I_B, as follows,

$$I_B = \alpha T + \beta R * G \tag{2.4}$$

Since the reflected objects can be in focus as well in practice, synthetic images with sharp reflections are also included in the training set.

In addition to the reflection blurring, we also consider the double reflection effect in synthesized image. Double reflection effect is formed due to the reflections from the two surfaces of the reflective glass [49]. This reflection effect can be simulated by convolving the reflection image R with a random kernel K with two pulses of amplitude 1 and α^2, where α is the transmittance [20]. Combining the blurring and double reflection effects together, we arrive at a generic formula for synthesizing reflection-interfered image I.

$$I = \alpha T + \beta R * G * K. \tag{2.5}$$

Shown in Figure 2.2 are some samples of the synthesized images.

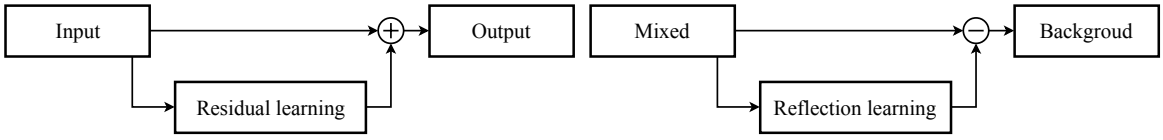


Figure 2.3: The left is the residual learning from [2] and the right is the proposed method

2.4 Proposed Method

2.4.1 Network Architecture

Similar to many deep learning based image restoration techniques, the proposed reflection removal method adopts the basic architecture of a convolutional encoder-decoder network [12]. The ultimate goal of our method is to find an optimal end-to-end mapping $T' = F(I)$ from a reflection-interfered image $I = T' + R'$ to its transmission layer T', where R' is the reflection layer of I. The transmission layer $T' = \alpha T$ is a glass-free image T of the targeted scene attenuated by α, the transmittance of the reflective glass in image I, as in the training data synthesis formula Eq. (2.5). Since the reflection layer R' is likely weak and smooth in comparison with T' [21, 44], T' should be similar to I in pixel values. Therefore, it is easier to optimize mapping $T' = F(I)$ than to optimize the mapping from I to T directly, even if transmittance α is known to the network. Once the solution to the transmission layer T' is given, it is trivial to restore a realistic reflection free image T from T'.

Another option is to train a residual mapping from image I to its reflection layer $R' = I - F(I)$. Since R' is relatively flat, the optimization of the residual mapping should be very effective. Residual learning has set the state of the art for many different image restoration problems [13, 50, 2]. However, none of the existing residual

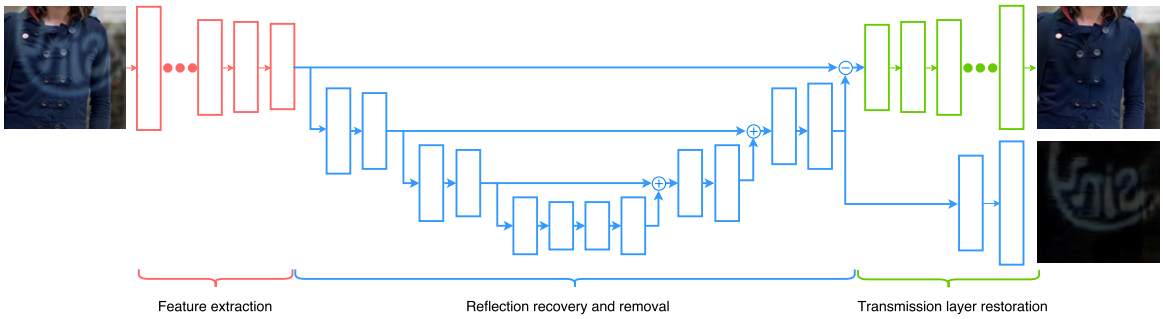


Figure 2.4: Architecture of the used convolutional auto-encoder with symmetric shortcut connection.

learning networks suits the characteristics of the reflection removal problem. In most residual learning techniques, the residual is the missing detail to be recovered and added back to the input image, while in our case, the residual is unwanted reflection that should be subtracted from the input. Furthermore, residual learning tends to emphasize on the fidelity of the recovered residual as the loss function is normally applied on the residual. But for the reflection removal problem, as long as the recovered transmission layer has reduced interference and looks natural, the restoration quality of the reflection layer is irrelevant. Due to these limitations, it is difficult to get satisfactory results for reflection removal by using residual learning directly. Thus, we place a residual learning based reflection recovery sub-network at the middle of our end-to-end mapping $T' = F(I)$ network, in order to exploit the efficiency of residual learning without affecting the output quality. The analogy of our method to the residual learning is shown in Figure 2.3

The proposed network consists of 12 convolutional layers and 12 deconvolutional with one rectified linear unit (ReLU) following each of the layers. The left graph of Figure **??** shows how ReLU works.The convolutional layers are designed to extract and condense features from the input, while deconvolutional layers rebuild the details of reflection-free image from feature abstractions. Overall, the architecture of proposed

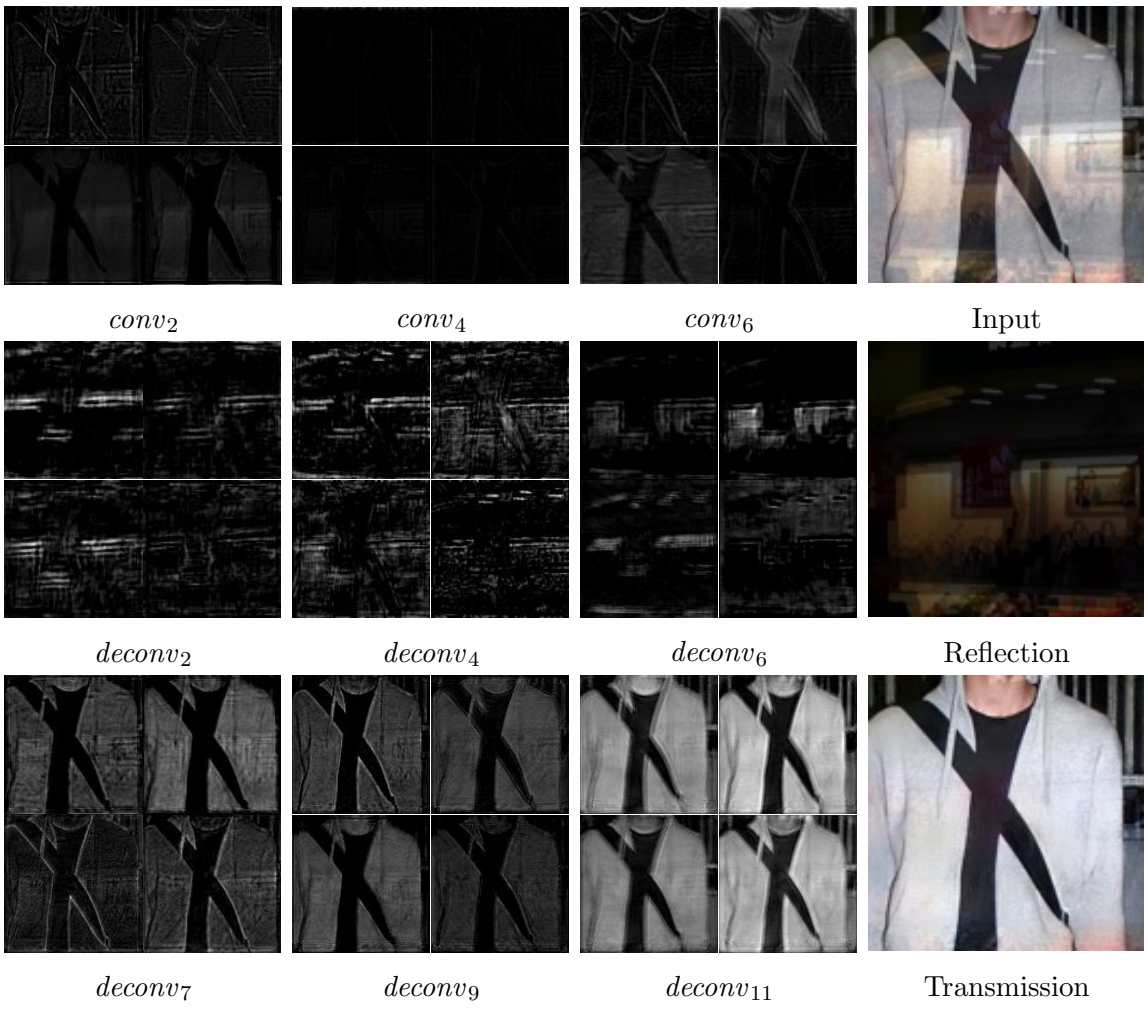


Figure 2.5: Sample feature maps at different stages of the network. These feature maps show that the network is working as intended in each stage.

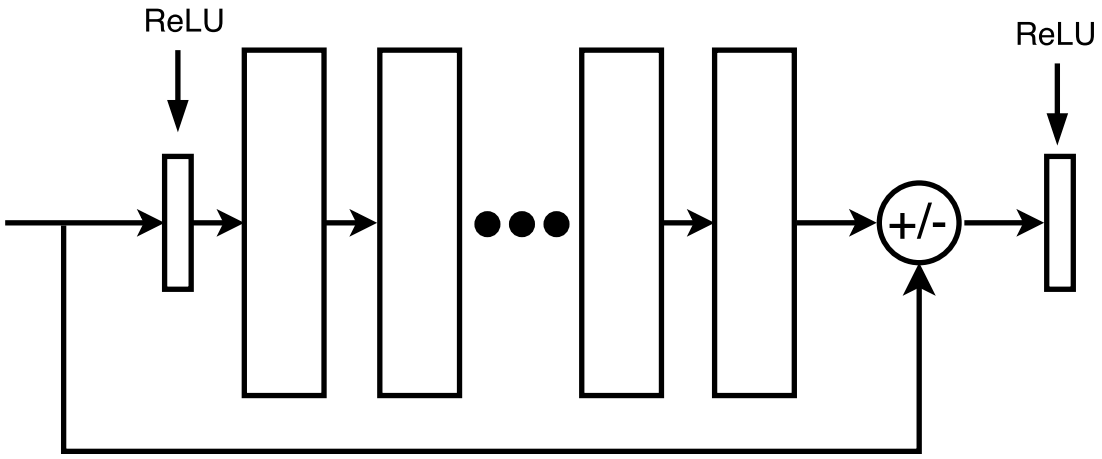


Figure 2.6: The topology of a single shortcut connection.

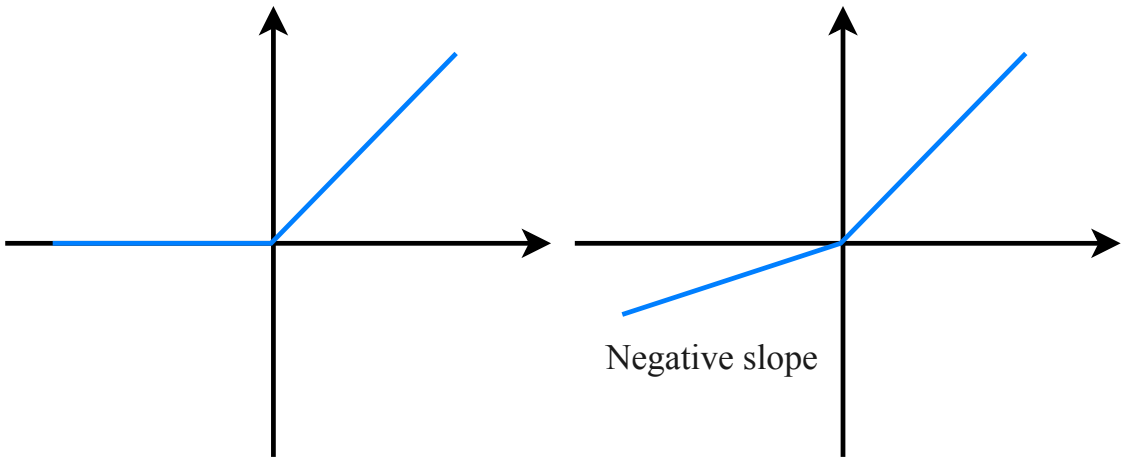


Figure 2.7: Rectified linear unit (ReLU) and leaky rectified linear unit (leaky ReLU).

encoder-decoder network can be divided into three stages:

1. **Feature extraction**. The 6 convolutional layers in this stage extract features for both transmission and reflection layers as illustrated in Figure 3.2. The outputs of the convolutional layers are shown in the first row of Figure 2.5.

2. **Reflection recovery and removal**. In this stage, the first 6 convolutional layers and following 6 deconvolutional layers are set to learn and recover reflection. Additionally, to preserve the details of the reflection layer better, two skip connections are added in the second stage to inherit the features learned from previous convolutional layers [12]. The topology of skip connection is illustrated in Figure 2.6. As shown in the second row of Figure 2.5, by minimizing the error of the output against the ground-truth reflection, this stage preserves only the features of the reflections and removes the transmission layer gradually from the input. At the end of this stage, the extracted reflection features are removed before $deconv_7$, the seventh deconvolutional layer, by using an element-wise subtraction [51] followed by a ReLU activation $\max(0, conv_6 - deconv_6)$.

3. **Transmission layer restoration**. The reflections might not be removed completely after previous stage by simply subtracting the estimated reflection, as shown in the last row of Figure 2.5. Thus, this stage tries to restore a visually pleasing transmission image from the reflection subtracted image. To achieve this goal, 6 deconvolutional layers are used to recover transmission layer from the features of the targeted scene.

For image classification tasks, pooling layers are necessary as it extracts main abstract features that are crucial for final decision [2]. However, as the redundant information increases the difficulty for deconvolutional layers to recover the image

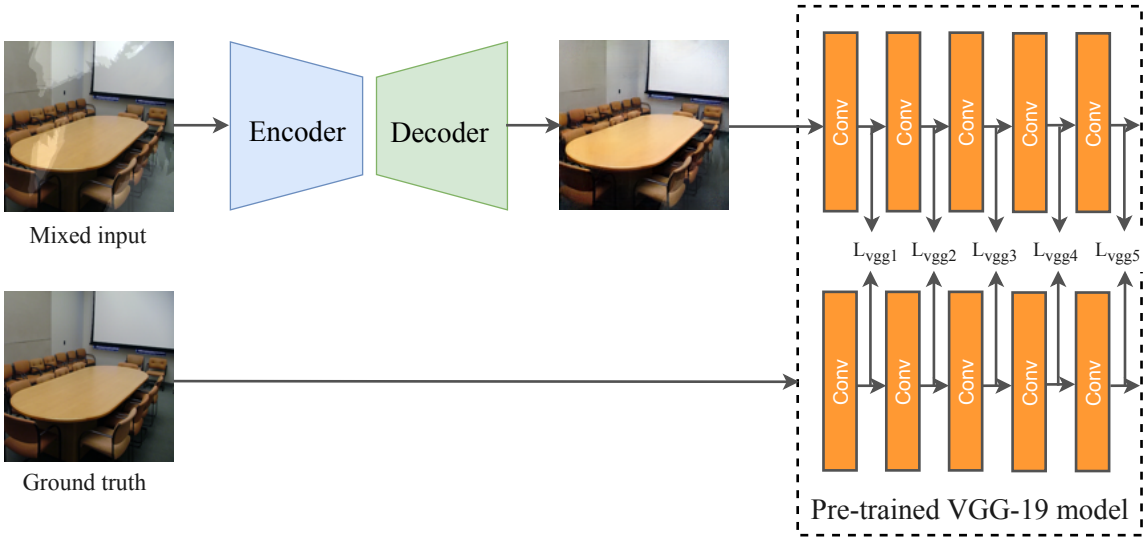


Figure 2.8: Illustration of VGG 19 perceptual loss

[12], pooling layers are omitted in our reflection removal network. Another important factor that affects the performance of our encoder-decoder network is the size of the convolutional kernel. To make the network learn the semantic context of an image, we employ relatively large kernels (5×5). But if the input image contains double reflections of large disparity, we find a larger kernel (9×9) for the first convolutional layers ($conv_1$ and $conv_2$) and the last deconvolutional layers ($deconv_{11}$ and $deconv_{12}$) is necessary to achieve the best performance.

2.4.2 Loss functions

Content and perceptual loss

In many neural network based image restoration techniques, such as denoising [52], debluring [53] and super resolution [50], the networks are commonly optimized using mean square error (MSE) between the output and the ground truth as the loss

function as follows,

$$L_{\ell_2} = \|F(I) - \alpha T\|_2^2 + \|F_R(I) - (1-\alpha)R\|_2^2, \tag{2.6}$$

where, in our case, $F(I)$ and $F_R(I)$ correspond to restored transmission and reflection layers. However, a model optimized using only ℓ_2-norm loss function often fails to preserve high-frequency contents. In the case of reflection removal, both the reflection and transmission layers are natural images with different characteristics. To get the best restoration result, the network should learn the perceptual properties of the transmission layer. Inspired by [54, 55], we employ a loss function that is closer to high-level feature abstractions. Based on Ledig et al. [13], the VGG loss is calculated as the ℓ_2-norm of the difference between the layer representations of the restored transmission $T' = F(I)$ and the real transmission image αT on the pre-trained 19 layers VGG network proposed by Simonyan and Zisserman [10]:

$$L_{\text{VGG}} = \sum_{i=1}^{M} \frac{1}{W_i H_i} |\phi_i(\alpha T) - \phi_i(F(I))\|_2^2 \tag{2.7}$$

where ϕ_i is the feature maps obtained by the i-th convolution layer (after activation) within the VGG19 network; M is the number of convolution layers used; and W_i and H_i are the dimension of i-th feature map. In our model, the feature maps of the first 5 convolution layers are used ($M = 5$) to build the perceptual loss. The illustration of VGG 19 loss is shown in Figure 2.8

Figure 2.9 presents some of the results from two reflection removal models trained with loss functions in Eqs. (2.6) and (2.10) respectively. As shown in the figure, the output images (3rd column) from the model trained with the mixed loss function as in Eq. (2.10) is much cleaner and closer to the ground-truths (4th column) than those

Figure 2.9: From left to right: input reflection-interfered images, network optimized for ℓ_2-norm loss, network optimized for both ℓ_2-norm and VGG losses, the ground-truth transmission images.

output images (2nd column) from the model trained solely with ℓ_2-norm loss as in Eq. (2.6).

Adversarial loss

To reduce any objectionable restoration artifacts and bring the restored image closer to natural images, we follow the idea of the two-player minimax game in [14] and employ a discriminator network D_{ϕ_D} which optimizes the following problem:

$$\min_{\phi_F} \max_{\phi_D} \mathbb{E}_{\alpha T \sim p_{data}(\alpha T)}[\log D_{\phi_D}(\alpha T)] + \mathbb{E}_{I \sim p_{data}(I)}[\log(1 - D_{\phi_D}(F_{\phi_F}(I))]. \tag{2.8}$$

The discriminator network D_{ϕ_D} is a trained classifier that estimates the likelihood of an input being a real reflection-free image or a restored image. If a well-trained D_{ϕ_D} cannot easily differentiate the output of the proposed reflection removal network from real reflection-free images, then the output should look natural and free of artifacts. Thus, in addition to the interlayer feature loss of VGG, we also include the naturalism

loss based on the probability of $D_{\phi_D}(F(I))$ as:

$$L_n = -\sum_i \log D_{\phi_D}(F_{\phi_F}(I)) \tag{2.9}$$

The discriminator network D_{ϕ_D} consists of five convolution layers followed by a fully connected layer and a softmax activation. Following each convolution layer are a batch normalization layer and a LeakyReLU layer with negative slope coefficient 0.2. The filter size is set to 5×5 with stride 2. The number of feature maps doubles after each convolution layer from 64 to 512.

The final loss of our model is calculated as weighted sum of VGG interlayer feature loss and naturalism loss:

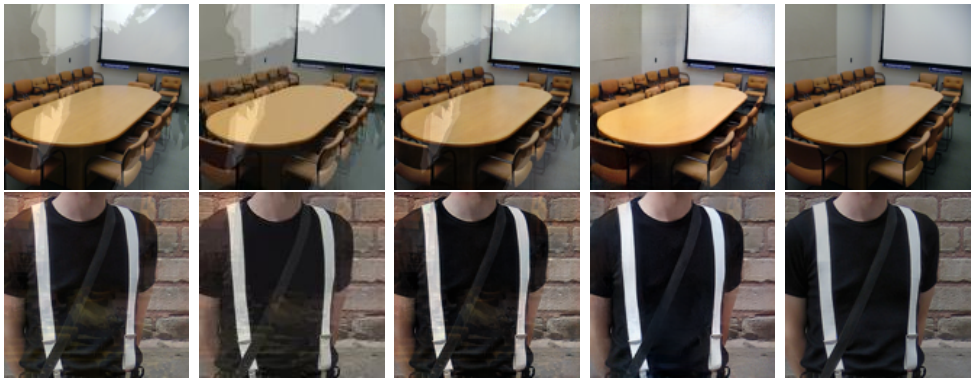
$$L = L_{\ell_2} + \lambda_1 L_{\text{VGG}} + \lambda_2 L_n, \tag{2.10}$$

where λ_1 and λ_2 balance the contribution from each portion.

2.5 Experiments

2.5.1 Data Preparation

To simulate the scenarios where reflections interfere the formation of images, 2303 images from the indoor scene recognition dataset [56] and 2622 street snap images [57] are collected online. We choose the images of natural landscapes and images taken inside a mall as the reflections. Leaves could create sparse shadow on the window interfering the transmission image, and the lights from a shop sign create strong and sharp reflection, which are also extremely common in real life reflection-interfered images. To ensure the size of training dataset and avoid over-fitting, each



Input [44] [19] Ours Ground truth

Figure 2.10: Comparison of reflection removal algorithms using synthetic images.

transmission image is synthesized with 18 randomly chosen reflection images using Eq. (2.5). To simulate the different blurriness of the reflections, the variance of the Gaussian blur kernel is selected randomly from 1 to 5. The transmittance α is also a random number between 0.75 to 0.8 for each synthetic image. Before generating a synthetic image, the transmission layer is resized to 128×128, whereas the reflection layer is randomly cropped from a large reflection image and then resized to 128×128. The reason for this step is that the reflected objects are normally far away from the glass, as a result, larger objects in the reflection scene appear relatively smaller in the reflection-interfered image. Finally, all the synthesized images are split into a training set of 66540 images and a testing set of 22110 images.

2.5.2 Network Training

The training of the end-to-end mapping network $T' = F(I)$ can be seen as solving the following optimization problem,

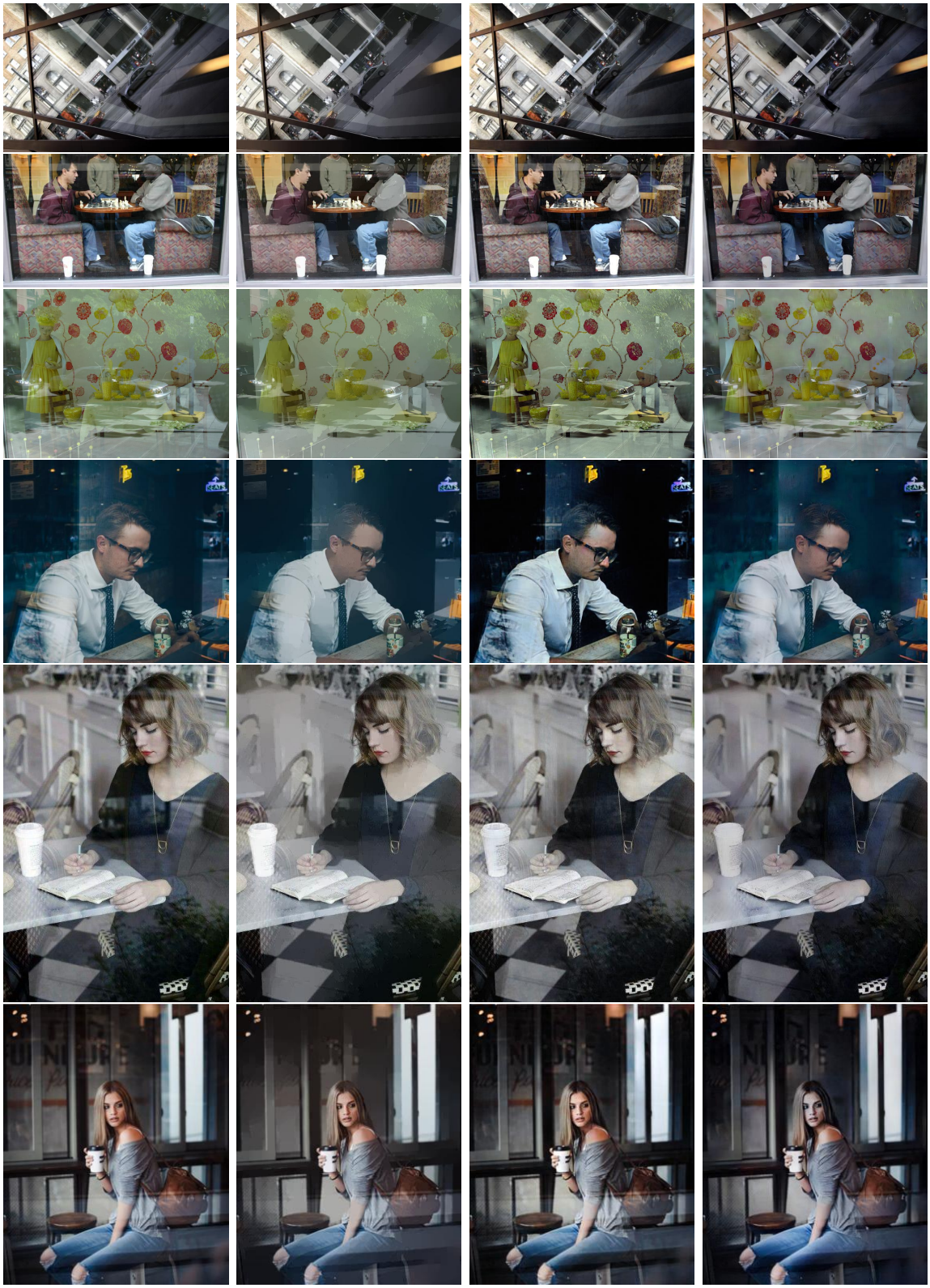
$$\min_{\theta_F} \frac{1}{N} \sum_{n=1}^{N} L(F(I_n), \alpha T_n) \tag{2.11}$$

where θ_F denotes the weights and biases of the network, N is the size of training set and L is the combined loss functions defined in Eq. (2.10). We set 64 filters for all the convolutional and deconvolutional layers and λ_1 and λ_2 are set to 10^{-3} and 10^{-4} respectively. The network is optimized by Adam optimizer [58] with a learning rate of 10^{-4} and $\beta_1 = 0.9$, batch size is set to 64 to accelerate the training process. All the experiments, including the following performance evaluation, are conducted on a computer with a Intel i7-6700K CPU, 16GB RAM and a NVidia Titan X GPU.

2.5.3 Evaluation

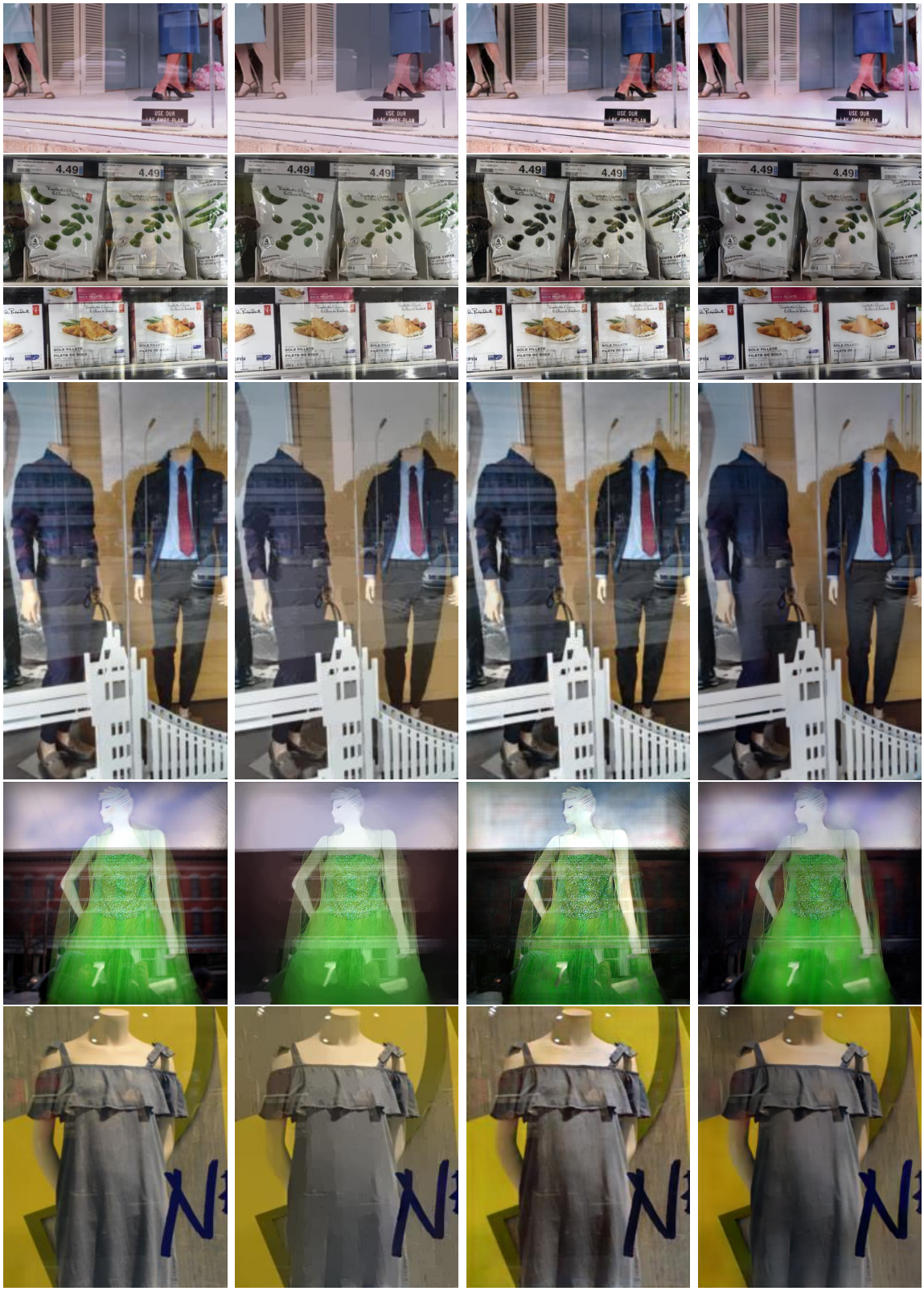
Figure 2.10 shows some results of the proposed method in comparison with the results of two state-of-the-art reflection removal techniques [44] and [19] using synthetic data. It is expected that the proposed method outperforms the compared techniques in this case, as our network is trained with images generated by the same data synthesizer. For real-life images collected by us or provided by the authors of [44], the results of proposed method are still the best among the tested techniques, as shown in Figure 2.11. The technique in [19] relies on the smoothness prior of the reflection image. However, in cases where the assumption is not true, as exemplified in the sample images, [19] could even enhance the reflections, making the results worse than the inputs. Our method, on the other hand, does not require the reflection image to be smooth; it works well even if the reflection is in focus and sharp. The problem of [44] is the severe loss of details in its output, resulting unnatural looking image. In the cases where the reflections are much stronger than the transmission layer, none of the tested algorithms can yield satisfactory results.

Reported in Table 2.1 are the PSNR and SSIM results of the tested techniques. In addition to the synthetic images, we tested benchmark sets provided by the authors



Input [44] [19] Ours

Figure 2.11: Comparison of reflection removal algorithms using real images.



Input [44] [19] Ours

Figure 2.12: Comparison of reflection removal algorithms using real images.

	[44]		[19]		Ours	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Synthetic Images	22.15	0.8455	22.14	0.8396	**29.85**	**0.9303**
Postcard Set [48]	21.05	0.8602	21.29	0.8692	**22.53**	**0.8743**
Solid Object Set [48]	22.36	0.8327	24.00	0.8669	**24.93**	**0.8808**

Table 2.1: PSNR and SSIM results of tested techniques using synthetic images and a benchmark dataset.

of [48]. Our method achieves the highest average PSNR and SSIM in these tests. The running times of the proposed method are comparable to other deep learning based image restoration techniques. The proposed method takes around 0.6 s to process a 128×128 image and 2 s to process a 512×512 image.

2.6 Conclusion

The task of removing reflection interference from a single image is a highly ill-posed problem. We propose a new reflection formation model taking into the consideration of physics of digital camera imaging, and apply the model in a deep convolutional encoder-decoder network based data-driven technique. Extensive experimental results show that, although the neural network learns only from synthetic data, the proposed method is effective on real-world images.

Chapter 3

Joint demosaicking and deblurring

3.1 Introduction

In consumer photography applications, objectionable blur artifacts caused by the shake of handheld camera is often unavoidable. The movement of the imaged subjects further complicates the blur problem. As shown in [59], the blur artifacts can significantly degrade the performance of an image processing technique that is tuned only for clear images. Therefore, it is important to clear up a blurry image not only for viewing pleasure but also for the accuracy of many computer vision applications. Deblurring algorithms can be categorized into two streams, non-blind and blind deblurring. The former is to recover sharp images with known blur kernels, whereas blind deblur methods simultaneously estimate the blur kernel and latent sharp image. Early deblurring methods focused on idealistic spatial-invariant blur or blur caused by simple translational/rotational motion. However, in practice, blur comes from various sources, such as depth variation, object motion and camera shake, and it is highly non-uniform. Due to the large number of unknown variables to the deblurring problem, it is extremely challenging to get a good estimate of the latent sharp image,

especially in non-uniform blur cases. Nevertheless, most of the deblurring techniques are built on the blur image degradation model as follows [60, 61]:

$$Y = X \circledast K + N, \tag{3.1}$$

where $\circledast$ denotes convolution operation, Y is the blurred image, X is the sharp and clean image, and K, N are blur kernel and unknown noise respectively. Additionally, these techniques often rely on some explicit assumptions on blur kernels or image statistical priors in order to make the under-determined inverse problem tractable. These assumptions, however, might not hold in real-world scenarios, causing new artifacts in the deblurred image.

Recently, convolutional neural network (CNNs) based techniques have made great progress for various computer vision problems including blind deblurring. For instance, [5, 62] restore sharp images directly from the blur images without learning explicit motion blur estimation; [63, 64, 65] learn the complex non-linear mapping between blur and sharp images through extensive training pairs; [63, 64] train networks on synthetic images and generalize well on real-world situations; [5] creates a training set by taking average on consecutive frames from videos captured in dynamic scene and does not need any degradation model.

The majority of cameras are designed to retain only one of primary colours per spatial pixel location by placing a colour filter array (CFA) in front of the sensor array. The most wildly used CFA is Bayer pattern which keeps twice green as red and blue as shown in Figure 3.1. The process of inferring two missing colours for each pixel in the mosaicked image to get the full RGB colour image is commonly referred as demosaicking. Demosaicing is one of the first and key components in the pipeline of camera image processing. Common demosaicking techniques, especially the

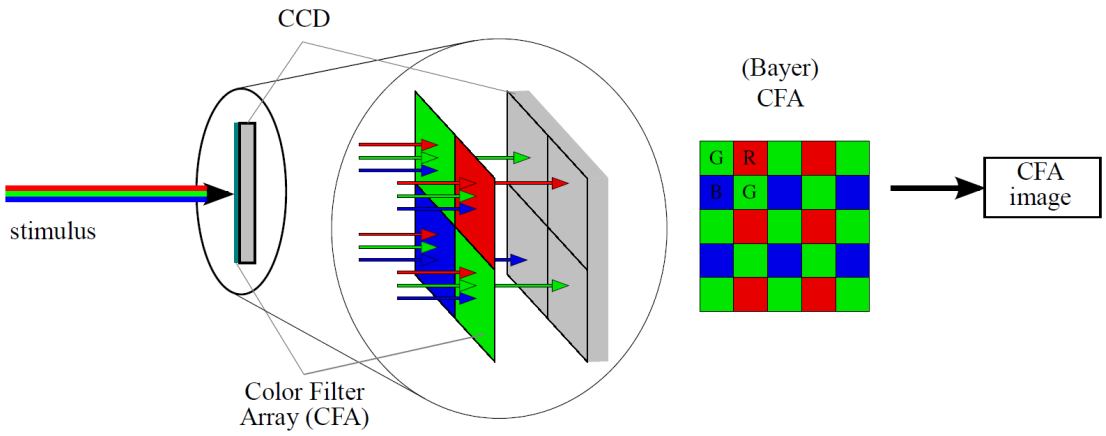


Figure 3.1: Mono-CCD technology outline, using the Bayer Colour Filter Array [3].

ones implemented in cameras, use some simple interpolation methods to estimate the missing colours from the surrounding pixels. These techniques often cause moire and zigzagging artifacts near edges, adversely affecting the performance of other computer vision operators, such as image restoration, classification, segmentation etc.

Demosaicking and deblurring have been studied in depth independently over the last a few decades. However, as we show in this chapter, demosaicking artifacts can greatly deteriorate the performance of a deblurring technique, thus, a good strategy is to tackle the two problems jointly. In this chapter, we propose a novel network architecture for joint demosaicking and deblurring.

3.2 Related Work

3.2.1 Demosaicking

To estimate the true colour image, early works use different interpolations to fill in the missing colours [66]. These methods are easy to implement, but are also prone to severe artifacts and false colour near the high frequency areas. Since there are more green pixels in a Bayer CFA, many techniques try to improve the green channel

estimate accuracy near the high frequency areas. Edge-adaptive methods can yield better solution by demosaicking along edges instead of crossing them [67, 68]. Heide et al. [69] proposed a flexible end-to-end camera image processing system (FlexISP) to solve image reconstruction problems using natural priors via joint optimization. Recently, Gharbi et al. [70] proposed a deep neural network based method trained with a large dataset. This method yields better results than traditional methods.

3.2.2 Deblur

Conventional blind deblur algorithms require additional assumptions or priors on image statistics to restrict the solution space. [60, 61] modeled non-uniform blur caused by camera shake and parametrized blur kernels to recover sharp images iteratively. Chakrabarti [63] built a CNN to find Fourier coefficients of a deconvolution filter to restore image patches and them estimate a global blur kernel to restore the observed blurry image. Nah et al. [5] stacked modified residual blocks to build a multi-scale CNN to mimic coarse-to-fine approaches. They claimed that latent sharp images in coarser scale provide crucial features that can help deblurring in finer scale. Ramakrishnan et al. [62] proposed a generative model based on generative adversarial network (GAN) and densely connected network to tackle the blind non-uniform blur removal problem. The network was designed to make independent decision for every convolutional unit based on the entire lower level activations.

3.2.3 Joint application multi-tasking

It has been shown that jointly solving multiple tasks in image processing pipeline can achieve superior results. Gharbi et al. [7] proposed a neural network architecture to solve demosaicking and denoising. They first extracted four colours in Bayer

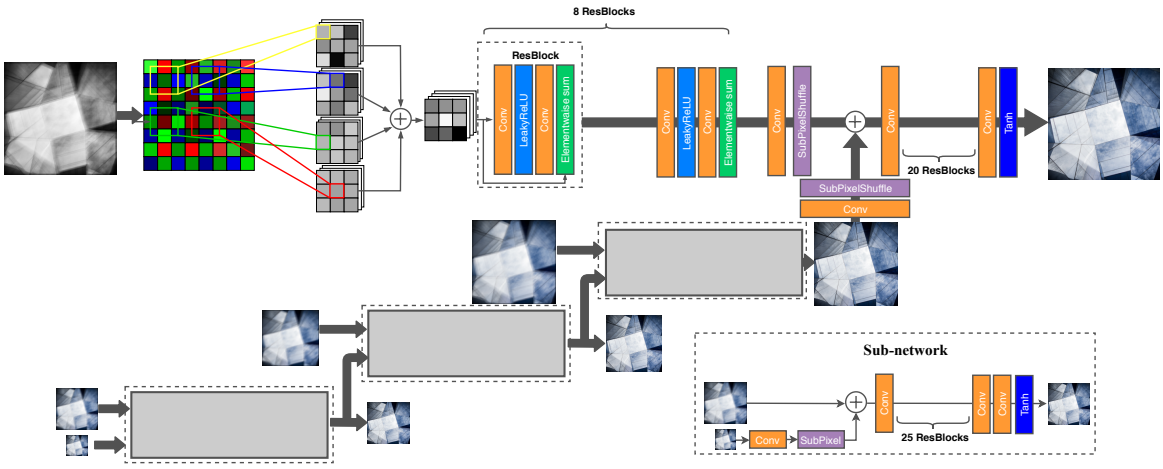


Figure 3.2: Architecture of proposed joint demosaicking and deblur neural network.

pattern to form a 5D tensor with an additional Gaussian noise term. The network is specialized to exploit the structure of Bayer pattern but also generalizes well on non-Bayer patterns. Zhou et al. [71] jointly solved demosaicking and super resolution in a single step using deep residual network. To match the Bayer structure and also consider the neighboring information, they set the first convolutional layer with kernel size 4×4 and stride 2. The proposed method successfully generated artifacts free colour high-resolution images from Bayer low-resolution images. Yoo et al. [72] tried to solve demosaicking and deblurring simultaneously. Their technique first estimates the edge direction and edge strength from Bayer domain and then performs edge adaptive deblurring and edge-orientated interpolation from the estimated edge information.

In this chapter, we introduce a deep leaning architecture to perform end-to-end mapping between Bayer blurry images to colour sharp images. The network consists of a subnetwork to perform deblurring recursively in a coarse-to-fine manner. To the best of our knowledge, we are the first to address joint demosaicking and blind deblur using deep learning techniques.

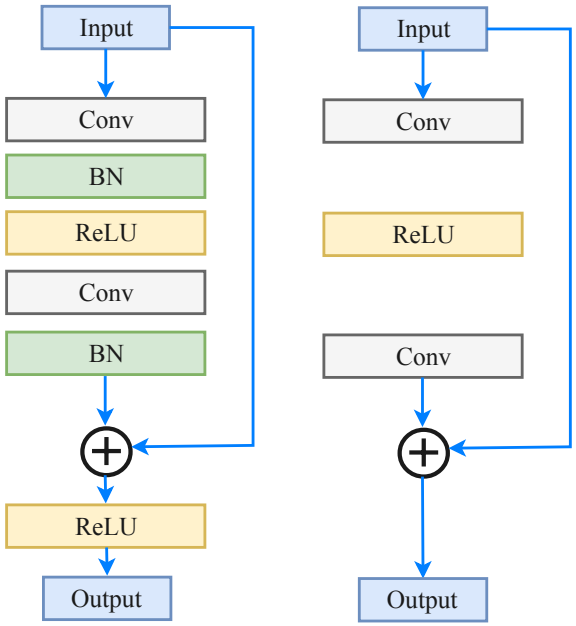


Figure 3.3: Residual blocks. The left one is the original one proposed by [2] and the right one is the modified residual block.

3.3 Proposed method

The proposed technique is a single end-to-end neural network that recovers colour and sharp images from blurry images recorded in Bayer pattern. The technique employs a novel demosaicking module which contains four separate convolutional layers applied on different structures of Bayer pattern respectively. With large amount of training data, the network is able to learn the colour of each pixel accurately without using explicit model of Bayer patterns. The network can then generate correct colour sharp images with a recursive multi-scale debur module in a coarse-to-fine fashion.

[13, 2, 5] successfully applied the idea of residual networks to build deeper and more powerful architectures for various image restoration problems. Since the blurry image and its ground truth are similar in pixel values, it is essential to learn only the residual between each two layers. The significant features can be inherited by identity mapping. In the proposed technique, we employ stacked residual blocks to

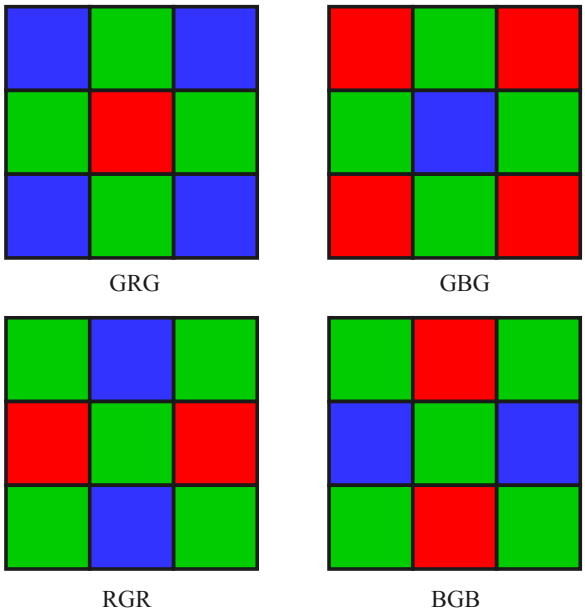


Figure 3.4: 3x3 structures in Bayer pattern

build the network. Like [5], we do not use the activation function in residual block after shortcut and we also omit the two batch normalization layers as shown in Figure 3.3. Batch normalization layer is not necessary in our method as we are using small mini-batch size during training. As each normalization layer consumes a large amount of GPU memory, as much as a convolutional layer, it is more cost-effective to remove normalization layer and invest the resource into building a deeper and larger network instead.

3.3.1 Demosaicking Module

In Bayer images, only one colour, red (R), green (G) or blue (B), is sampled for each spatial pixel location. The Bayer pattern is repetitive in the unit of a 2×2 window, where each window contains 2 green, 1 blue and 1 red pixels. As we can see, there are four different 3×3 structures in Bayer pattern; each has one of four colours at the center as shown in Figure 3.4. [7, 70] rearranged colours in the mono-channel

Bayer image to form a 4D quarter-resolution tensor at the beginning of the network. [71] claimed the neighboring colours also may also effect the results. They performed colour extraction in the first convolutional layer by setting the filter size to 4×4 with spatial stride 2.

We apply four branches of convolutional layers on the independent Bayer structures separately. Each branch contains 32 (3×3) kernels and is only responsible for convolving with the same Bayer structure centered at one of the $'GRBG'$ pixel as illustrated in Figure 3.2. The center of the kernels is the spatial location under the consideration for demosaicking. The neighbors of the centered pixel are crucial as they provide correlated information to estimate the missing values. In the first layer, each branch takes one exact colour and estimates the other two with its eight local neighbors. After that, the feature maps are concatenated to form a 128-channel tensor. The next 8 residual blocks as described above extract the features non-linearly. Convolution on the concatenated feature maps achieved information sharing among all the Bayer structures. Thus, the network has wilder receptive field and better colour estimation accuracy than the existing techniques.

To implement this, the Bayer input of dimension $n \times n$ is pre-paded with one extra pixel on the border while maintaining the Bayer pattern. Before first convolution, each branch extracts $n \times n$ slice starting from one of the coordinates: $[0, 0], [0, 1], [1, 0], [1, 1]$. The purpose is to align the center of the kernels with one specific colour for each branch. During the convolutions, padding is not used and spatial stride is set to 2 to ensure every kernel is convolved with the same structure. After the first layers, the spatial dimension is decreased to $\frac{n}{2} \times \frac{n}{2}$.

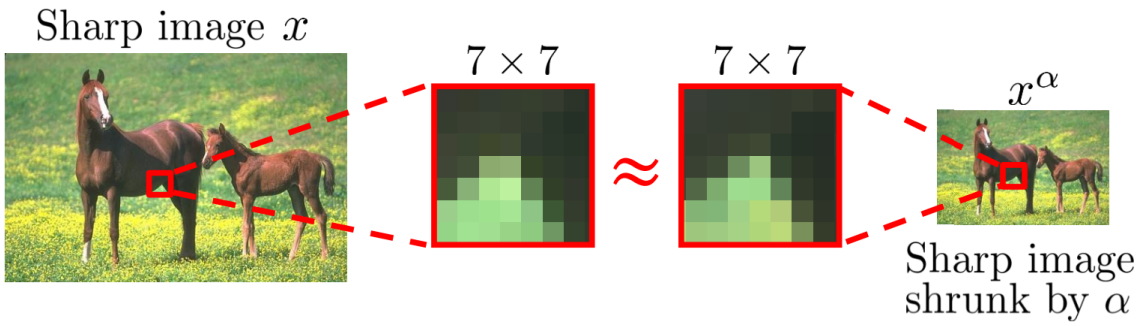


Figure 3.5: Patch recurrence across scales in sharp images [4].

3.3.2 Deblur Module

Cross-scale recurrence property indicates that redundant patches recur across scale in sharp images as shown in Figure 3.5. This property has been successfully applied in blind super-resolution [73]. [4] borrowed that idea and the fact that blurriness diminishes at coarser scales of the image. They claimed that sharper image patches maintain the similar structures in coarser scale blur image. They have shown that the patches in coarser image scales can be served as a stable patch prior for deblurring. Thus, it is relatively easier to perform deblurring in coarser scale.

[5] adopted the coarse-to-fine approach, and designed a three-scale network to recover sharp images in different resolutions. Finer scale stage inherits the property of the recovered latent sharp images in coarser scale. However, two flaws are presented in the network: *i*) The network for each scale level is doing the same work which brings redundancy to the whole network. *ii*) Degraded images with various degree of blur require networks trained on different number of scale levels. The networks have to be re-trained from the beginning. The training process is time consuming and computationally expensive as the number of convolutional layers increases linearly with the number of scale levels.

To tackle the aforementioned problems, we propose a subnetwork to perform the

coarse-to-fine approach recursively. As indicated in Figure 3.2, the subnetwork takes two inputs: a blurry image and its latent sharp image in half of its resolution. The output of the subnetwork is a sharp image of the same size as the blurry input image. Generally speaking, the low-resolution latent sharp input has the full low-frequency structures of the image, and it also preserves some of the high-frequency information, useful for the reconstruction of the full-resolution image. In the subnetwork, the latent sharp image is first fed through a convolution layer then followed by a upconvolution layer [74]. The image is then concatenated with the blurry image followed by a convolution layer to convert them to feature maps. After that, the subnetwork employs 25 stacked ResBlocks as described above to generate the output sharp image. Spatial resolution is preserved with zero-padding throughout the subnetwork. Our intention is to use the same subnetwork to perform deblurring at different scale levels. Compared with [5] where 40 convolution layers are used for each scale level, our recursive subnetwork uses 54 convolution layers, regardless of the number of scale levels. For instance, if there are 3 scale levels, the proposed network only needs less than half of the convolution layers used by [5]. Additionally, to add more scale levels for more severe blur, we can simply cascade the subnetworks without training the whole network.

Shown in Figure 3.2 is the architecture of the proposed network for joint demosaicking-deblurring. The outputs of mosaicking module and subnetwork are concatenated. The concatenated tensor contains inferred chrominance as well as high-frequency information in intensity. It is then fed to a convolutional layer followed by 20 ResBlocks. The last layer outputs sharp colour images. We set 12 filters for convolution layers before the sub-pixel upsampling layer [74]. The upsampling layer efficiently reshapes the tensor of shape $H \times W \times C$ into a tensor of shape $r \cdot H \times r \cdot W \times \frac{C}{r^2}$, where r represents the up-sampling factor. The up-sampling factor r is set to 2 in our method. We set

128 filters for convolution layers in ResBlocks in demosaicking module and 64 filters for those that are not specified. Small kernels (3×3) are employed throughout the whole network. Although in each subnetwork, the size of the convolution kernels are small, the combined network still has a large effective receptive field that is excellent for dealing with very blurry images, thanks to the coarse-to-fine multi-scale structure.

3.3.3 Loss Functions

Many supervised learning based image restoration techniques employ mean square error (MSE) between the restored image and ground truth as the loss function[12]. In the proposed multi-scale network, each subnetwork also uses MSE loss, as the network is aimed to produce images as close to the ground truth as possible at every scale level. We define the content loss function as follows:

$$L_{\ell_2} = \sum_{i=1}^{I} \frac{1}{h_i w_i c_i} \left\| F_i(B_i, \theta) - S_i \right\|_2^2 \tag{3.2}$$

where B_i and S_i are training pairs of blurry and sharp images. Function $F(B, \theta)$ is the CNN parametrized by θ. I denotes the total level of scales used in the network. The loss is weighted by the total number of pixels $h_i w_i c_i$. However, using MSE alone tends to result overly smoothed images with significant loss of high-frequency information [75]. Moreover, the MSE of a slightly blurry image is already low in general, making further optimization by MSE difficult. To restore sharper edges and get perceptual satisfying results, we need additional comparison in structure. Inspired by [55], we employ a loss function that compares features in high-level abstraction to compensate the drawbacks from MSE optimization. As defined as follows, the loss function takes the Euclidean distance between feature maps in a pre-trained CNN of the ground

truth and restored image,

$$L_{vgg} = \frac{1}{h_{i,j} w_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(F(B))_{x,y} - \phi_{i,j}(s)_{x,y}) \tag{3.3}$$

where $\phi_{i,j}$ are the feature maps generated from a pre-trained VGG-19 network [10] at ReLU 4_4 layer. Scalars $H_{i,j}$ and $W_{i,j}$ are the corresponding height and width. Overall, the loss function used for training the network is a weighted average between MSE and VGG interlayer perceptual loss:

$$L = L_{\ell_2} + \lambda L_{vgg}, \tag{3.4}$$

where λ is empirically set to 2×10^{-6} in our experiments to balance the two portions of losses.

3.4 Experiments

3.5 Experiments

3.5.1 Training Data Preparation

Half of the image pairs for training the proposed network come from the GoPro dataset [5]. In this dataset, each blur image is synthesized by averaging several successive frames from 240 fps video clips captured by a GoPro camera, and the corresponding sharp group truth image is the middle frame in the sequence. Using this method, we can obtain a large amount of paired training data without modelling blur kernels. The drawback is that, coming from a video clip, the ground truth images are

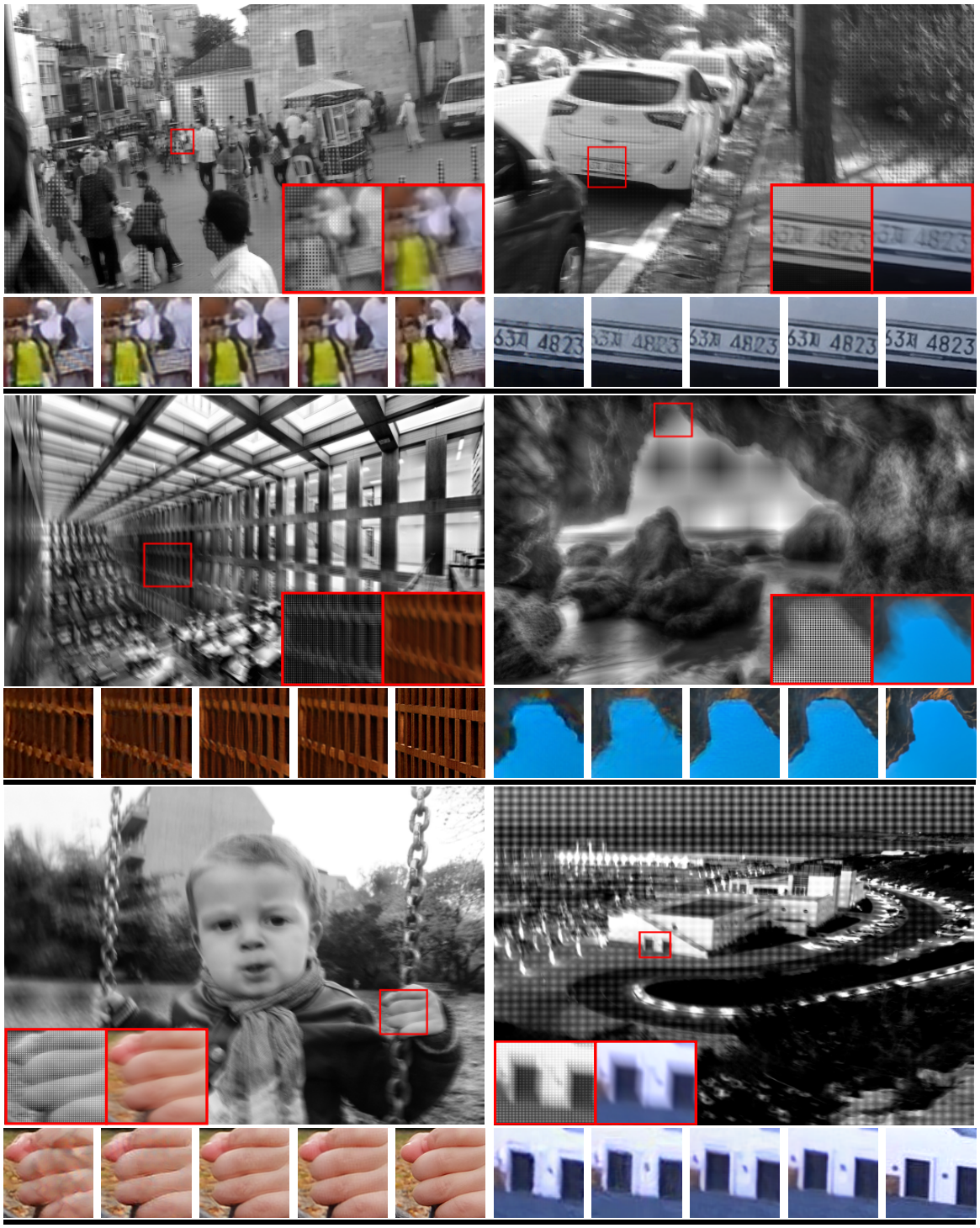


Figure 3.6: Demosaicking and deblurring results on different datasets. The first row are results on two images taken from GoPro dataset [5]. The rest are from Lai et al.'s dataset [6]. From left to right, the shown results are Matlab + [5], [7] + [5], proposed multi-scale method, proposed recursive multi-scale method and ground truth.

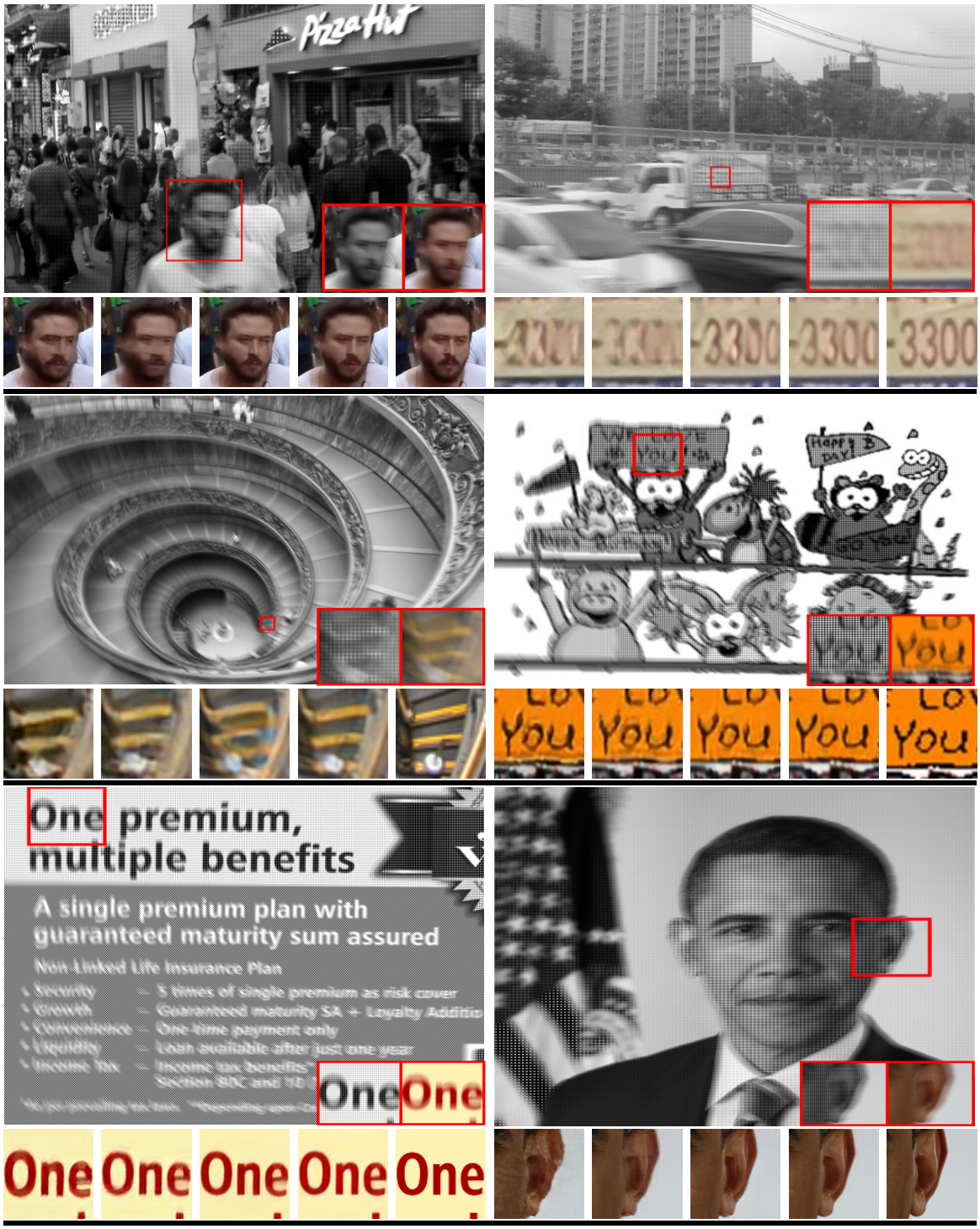


Figure 3.7: Demosaicking and deblurring results on different datasets. The first row are results on two images taken from GoPro dataset [5]. The rest are from Lai et al.'s dataset [6]. From left to right, the shown results are Matlab + [5], [7] + [5], proposed multi-scale method, proposed recursive multi-scale method and ground truth.

often not sharp enough, as a result, the network cannot properly learn the high frequency features of sharp images from such a non-ideal training dataset. In addition, this datasse contained limited number of scenes and that may lead to network overfitting. To alleviate this problem, we also synthesize training data from the DIV2K dataset [76] using non-uniform blur kernels with a method similar to [77].

The DIV2K dataset contains 900 high-quality images with different types of scenes. From the dataset, we extract colour patches of size 256×256, and resample them to Bayer pattern images. These Bayer pattern images are then pre-padded to size 258×258 and used as the inputs to the finest level. For the coarser levels, the colour patches are used directly, as the demosaicking problem has been solved in the first level and all the images are now in full colour. To get more data, we also perform data argumentation by flipping and rotating the blur kernels. In total, 0.22 million training pairs of Bayer pattern blurry and sharp images are produced.

3.5.2 Training Details and Evaluation

Loss functions	MSE			
Scale levels	I = 1	I = 2	I = 3	I = 4
PSNR	27.0591	27.8193	28.2261	28.3373
SSIM	0.8756	0.8899	0.8959	0.9004
Loss functions	MSE+VGG			
PSNR	26.9725	27.8335	28.2907	**28.4663**
SSIM	0.8719	0.887	0.894	**0.9033**

Table 3.1: Progressive results of multi-scale recursive subnetwork on GoPro dataset [5].

In our experiment, we train the subnetwork recursively using 3 scale levels. If more scale levels are necessary in the testing phase, we can simply stack more of the trained subnetwork together without retraining the whole network. In the testing phase, the PSNR and SSIM performances of the network generally increase with the

GoPro dataset [5]						
Measure	Colour + [5]	Matlab + [5]	[7] + [5]	Ours (multi-scale)		
				I = 1	I = 2	I = 3
PSNR	29.0570	26.1923	26.8237	26.4943	29.3	29.3812
SSIM	0.9106	0.8618	0.8663	0.8672	0.9133	0.9135
Köhler dataset [77]						
PSNR	24.5376	22.3496	24.2043	23.831	24.234	24.5675
SSIM	0.8517	0.7937	0.8399	0.8424	0.85	0.8558

Table 3.2: PSNR and SSIM results of previous techniques on GoPro dataset [5] and Köhler dataset [77].

GoPro dataset [5]					
Measure	Ours (multi-scale + recursive)				
	I = 1	I = 2	I = 3	I = 4	I = 5
PSNR	26.3177	27.4621	29.0395	**29.4134**	29.3061
SSIM	0.8677	0.8828	0.9121	**0.9176**	0.9156
Köhler dataset [77]					
PSNR	23.7651	23.8505	23.9787	24.8294	**24.9083**
SSIM	0.8447	0.8457	0.8464	0.8613	**0.8625**

Table 3.3: PSNR and SSIM results of the proposed method on GoPro dataset [5] and Köhler dataset [77].

number of scale levels. Even though the network is trained recursively using 3 scale levels, it performs better when we use 4 scale levels in the testing phase. As 4 scale levels have already offered $2^4 = 16$ scale factor, the blur in the downsampled image is negligible, hence the benefit of adding more levels in the testing phase is marginal. Shown in Table 3.1 are the results of the network evaluated using the GoPro dataset. As demonstrated in the table, the VGG perceptual loss can also slightly improve the results.

As there is no existing joint deep learning based joint demosaicking and deblurring technique in the literature, we cascade state-of-the-art deblurring algorithm [5] and demosaicing algorithm [7] together and solve the problem in two steps. The deblurring algorithm [5] uses a 3-scale-level network, which is retrained using the same training

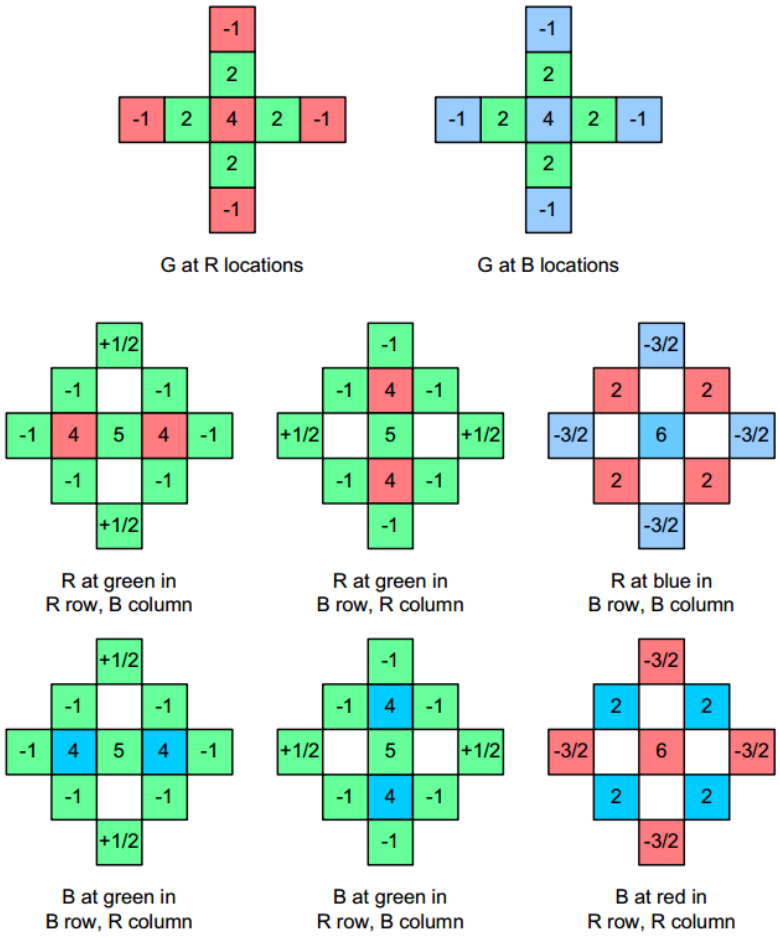


Figure 3.8: Filter coefficients for linear interpolation [8].

data as our technique. To further investigate effects of demosaicking to the second step deblurring, we also test the Matlab build-in function *demosaic* as a demosaicking operator, due to its similarity to the algorithms used by common cameras. The Matlab function uses simple linear interpolation, which can be implemented efficiently with low-power hardware. The filter coefficients for this function is shown in Figure 3.8. The deep neural network based demosaicking approach [7] can also deal with noise. To reduce the interference from the denoising part, we turned this function off in our experiments. We also trained a non-recursive version of our method as in [5]. We set 14 residual blocks for each coarser scale level so that the total number of parameters is similar to the recursive method. The recursive network is trained using 4 levels: demosaicking module plus 3 stacked subnetworks. We adopt the trained subnetwork from experiments of Table 3.1, it is then fine tuned during the training of the whole network. The pixel value of input images are scaled between range $[-1, 1]$. All convolutional layers are activated by LeakyReLU with negative slope equals to 0.2 except the output layers use *tanh* activation. T he right graph of Figure 2.7 shows how LeakyReLU works. The network is optimized by Adam optimizer [58] with mini-batch size 4. We set the initial learning rate to 10^{-4} and reduce to 10^{-5} when the loss plateaus. All the experiments are implemented with Tensorflow library on a computer with a Intel i7-6700K CPU and a NVIDIA Titan X GPU.

We report three separate networks for non-recursive method, each uses different number of scale levels as [5]. As for our recursive method, we report 5 number of scales levels using the same subnetwork recursively. All methods are evaluated and compared on the following benchmarks:

GoPro Dataset [5]: This dataset contains blurry images generated by averaging consecutive sharp frames in videos captured by GoPro. 634 images are used for testing and the rest are used to generate training data. The PSNR and SSIM results

are reported in Table 3.2 and Table 3.3. When demosaicking and deblurring are solved in sequence, the performance of the deblur model deteriorates because of the induced artifacts by demosaicking. Deep learning based demosaicking is ineffective in improving either PSNR or SSIM. For our multi-scale methods, more stages results in higher PSNR and SSIM as expected. The recursive method yields the best results when the number of scale levels equals to 4.

Köhler dataset [77]: The authors recorded 6D camera trajectories to simulate the blur over 4 latent sharp images, each image is blurred by 12 different kernels. The quantitative results are reported in Table 3.2 and Table 3.3. Similar results are observed as in GoPro dataset: the joint method performs better than the sequential methods. It is notable that, unlike the results on GoPro dataset, the proposed recursive method has the highest PSNR and SSIM values when the number of scale levels is 5.

To investigate the degradation from demosaicking process, deblurring algorithm [5] is performed directly on colour images. The results can be found in the first column of Table 3.2. As shown in the table, given Bayer pattern input, our technique still outperforms deblurring [5] on full colour images. This demonstrates the great effectiveness of our technique.

Some sample images are provided in Figure 3.6 and Figure 3.7. Since the blurry and ground truth images are not aligned in Lai et al.'s dataset [6], we only report qualitative results. The first row contains the testing cases from GoPro dataset and the rest of the rows are from Lai et al.'s dataset. As shown in the results, the tested two demosaicking methods bring different impacts to the same deblur model. Matlab recovers colour images using simple interpolation and generates severe artifacts such as zigzagging and false colour. Those artifacts will be accumulated as they are not typically observed as dominated noise in blurred images. A better demosaicking

method alleviates those problems by passing reduced noises to the next stage, as referred to the second columns in Figure 3.6. But, the performance highly depends on how the demosaicking model is trained. In some cases, it interferes the deblurring operator more than the Matlab method. The proposed recursive method produces more visually pleasing results than the non-recursive version.

3.6 Conclusion

Two non-invertible tasks, demosaicking and deblurring, have been studied extensively and independently in the literature. However, treating the two problems separately often results severe artifacts accumulation. To combat this problem, the proposed technique works on blurry Bayer pattern images directly. Extensive experiments show that the joint demosaicking and deblur method gives better results compared with alternative solutions.

Chapter 4

Conclusion

Various common types of image degradations, such as sensor noise, downsampling, motion blur, etc., can greatly affect the visual quality of an image. For many image processing and computer vision applications, it is imperative to digitally inverse the degradation processes and restore high fidelity images from the degraded versions. However, most image restoration problems are ill-posed; there are often countless possible solutions that yield the same degraded image. In order to find the best one of the solutions, conventional approaches generally rely on some image degradation models and image statistical priors to limit the solution space. Unfortunately, these explicit assumptions pertaining to the degradation processes and image statistics are often too simplistic and idealistic out of engineering necessity. As a result, conventional approaches often perform poorly in real-world scenarios with complex or compound degradations or both.

In comparison, the newly emerged data driven approach, DCNN, is potentially more robust for real-world image restoration problems. Without employing any explicit degradation models, DCNN based techniques learn the mapping from a degraded image to its corresponding clean image through a large number of such sample pairs.

Given sufficient training data, a well designed DCNN technique can generalize well on images that are statistically similar to the training data.

Armed with the deep learning techniques, in this thesis, we tackle two image restoration problems: single image reflection removal and joint demosaicking and deblurring. For the former problem, we propose a new reflection formation model for generating synthetic photo-realistic images as training data, and we design a three-stage convolutional encoder-decoder network to remove reflections. The feature maps in each convolutional layer for the trained encoder-decoder network show that each stage is working as intended. For the second problem, we proposed recursive multi-scale network to solve demosaicking and deblurring jointly in a coarse-to-fine manner. The network explores the information on four independent Bayer structures. Combining with a subnetwork that performs deblurring process recursively, the network is able to produce high fidelity images from blurry Bayer images. Extensive experimental results show that the proposed methods outperform other state of arts for both applications.

Bibliography

[1] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 1–9

[2] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016) 770–778

[3] Losson, O., Macaire, L., Yang, Y.: Comparison of color demosaicing methods. In: Advances in Imaging and Electron Physics. Volume 162. Elsevier (2010) 173–265

[4] Michaeli, T., Irani, M.: Blind deblurring using internal patch recurrence. In: European Conference on Computer Vision, Springer (2014) 783–798

[5] Nah, S., Kim, T.H., Lee, K.M.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: IEEE CVPR. Volume 2017. (2017)

[6] Lai, W.S., Huang, J.B., Hu, Z., Ahuja, N., Yang, M.H.: A comparative study for single image blind deblurring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 1701–1709

[7] Gharbi, M., Chaurasia, G., Paris, S., Durand, F.: Deep joint demosaicking and denoising. ACM Transactions on Graphics (TOG) **35**(6) (2016) 191

[8] Malvar, H.S., He, L.w., Cutler, R.: High-quality linear interpolation for demosaicing of bayer-patterned color images. In: Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on. Volume 3., IEEE (2004) iii–485

[9] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105

[10] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

[11] Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: European conference on computer vision, Springer (2014) 184–199

[12] Mao, X.J., Shen, C., Yang, Y.B.: Image restoration using convolutional auto-encoders with symmetric skip connections. arXiv preprint arXiv:1606.08921 (2016)

[13] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. arXiv preprint arXiv:1609.04802 (2016)

[14] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. (2014) 2672–2680

[15] Kong, N., Tai, Y.W., Shin, J.S.: A physically-based approach to reflection separation: from physical modeling to constrained optimization. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **36**(2) (2014) 209–221

[16] Gai, K., Shi, Z., Zhang, C.: Blind separation of superimposed moving images using image statistics. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **34**(1) (2012) 19–32

[17] Li, Y., Brown, M.S.: Exploiting reflection change for automatic reflection removal. In: IEEE International Conference on Computer Vision (ICCV). (2013) 2432–2439

[18] Levin, A., Zomet, A., Weiss, Y.: Separating reflections from a single image using local features. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Volume 1. (2004) 306–313

[19] Fan, Q., Yang, J., Hua, G., Chen, B., Wipf, D.: A generic deep architecture for single image reflection removal and image smoothing. In: IEEE International Conference on Computer Vision (ICCV). (2017) 3238–3247

[20] Shih, Y., Krishnan, D., Durand, F., Freeman, W.T.: Reflection removal using ghosting cues. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2015) 3193–3201

[21] Li, Y., Brown, M.S.: Single image layer separation using relative smoothness. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2014) 2752–2759

[22] Ohnishi, N., Kumaki, K., Yamamura, T., Tanaka, T.: Separating real and virtual objects from their overlapping images. In: European Conference on Computer Vision (ECCV). (1996) 636–646

[23] Farid, H., Adelson, E.H.: Separating reflections and lighting using independent components analysis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Volume 1. (1999) 262–267

[24] Sarel, B., Irani, M.: Separating transparent layers through layer information exchange. In: European Conference on Computer Vision (ECCV). (2004) 328–341

[25] Schechner, Y.Y., Shamir, J., Kiryati, N.: Polarization and statistical analysis of scenes containing a semireflector. Journal of the Optical Society of America A (JOSA A) **17**(2) (2000) 276–284

[26] Feris, R., Raskar, R., Tan, K.H., Turk, M.: Specular reflection reduction with multi-flash imaging. In: Brazilian Symposium on Computer Graphics and Image Processing. (2004) 316–321

[27] Agrawal, A., Raskar, R., Nayar, S.K., Li, Y.: Removing photography artifacts using gradient projection and flash-exposure sampling. ACM Transactions on Graphics (TOG) **24**(3) (2005) 828–835

[28] Schechner, Y.Y., Kiryati, N., Shamir, J.: Blind recovery of transparent and semireflected scenes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Volume 1. (2000) 38–43

[29] Schechner, Y.Y., Kiryati, N., Basri, R.: Separation of transparent layers using focus. International Journal of Computer Vision (IJCV) **39**(1) (2000) 25–39

[30] Szeliski, R., Avidan, S., Anandan, P.: Layer extraction from multiple images containing reflections and transparency. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Volume 1. (2000) 246–253

[31] Tsin, Y., Kang, S.B., Szeliski, R.: Stereo matching with linear superposition of layers. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **28**(2) (2006) 290–301

[32] Sinha, S.N., Kopf, J., Goesele, M., Scharstein, D., Szeliski, R.: Image-based rendering for scenes with reflections. ACM Transactions on Graphics (TOG) **31**(4) (2012) 100:1–100:10

[33] Guo, X., Cao, X., Ma, Y.: Robust separation of reflection from multiple images. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2014) 2187–2194

[34] Han, B.J., Sim, J.Y.: Reflection removal using low-rank matrix completion. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017) 5438–5446

[35] Xue, T., Rubinstein, M., Liu, C., Freeman, W.T.: A computational approach for obstruction-free photography. ACM Transactions on Graphics (TOG) **34**(4) (2015) 79:1–79:11

[36] Yang, J., Li, H., Dai, Y., Tan, R.T.: Robust optical flow estimation of double-layer images under transparency or reflection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016) 1410–1419

[37] Sarel, B., Irani, M.: Separating transparent layers of repetitive dynamic behaviors. In: IEEE International Conference on Computer Vision (ICCV). Volume 1. (2005) 26–32

[38] Simon, C., Kyu Park, I.: Reflection removal for in-vehicle black box videos. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2015) 4231–4239

[39] Chandramouli, P., Noroozi, M., Favaro, P.: Convnet-based depth estimation, reflection separation and deblurring of plenoptic images. In: Asian Conference on Computer Vision (ACCV). (2016) 129–144

[40] Ni, Y., Chen, J., Chau, L.P.: Reflection removal based on single light field capture. In: IEEE International Symposium on Circuits and Systems (ISCAS). (2017) 1–4

[41] Levin, A., Weiss, Y.: User assisted separation of reflections from a single image using a sparsity prior. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) **29**(9) (2007) 1647–1654

[42] Yeung, S.K., Wu, T.P., Tang, C.K.: Extracting smooth and transparent layers from a single image. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2008) 1–7

[43] Akashi, Y., Okatani, T.: Separation of reflection components by sparse non-negative matrix factorization. In: Asian Conference on Computer Vision (ACCV). (2014) 611–625

[44] Arvanitopoulos Darginis, N., Achanta, R., Süsstrunk, S.: Single image reflection suppression. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017)

[45] Wan, R., Shi, B., Tan, A.H., Kot, A.C.: Sparsity based reflection removal using external patch search. In: IEEE International Conference on Multimedia and Expo (ICME). (2017) 1500–1505

[46] Sandhan, T., Choi, J.Y.: Anti-glare: Tightly constrained optimization for eyeglass reflection removal. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2017) 1241–1250

[47] Artusi, A., Banterle, F., Chetverikov, D.: A survey of specularity removal methods. Computer Graphics Forum **30**(8) (2011) 2208–2230

[48] Wan, R., Shi, B., Duan, L.Y., Tan, A.H., Kot, A.C.: Benchmarking single-image reflection removal algorithms. In: IEEE International Conference on Computer Vision (ICCV). (2017) 3922–3930

[49] Diamant, Y., Schechner, Y.Y.: Overcoming visual reverberations. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2008) 1–8

[50] Kim, J., Kwon Lee, J., Mu Lee, K.: Accurate image super-resolution using very deep convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016) 1646–1654

[51] Yang, W., Tan, R.T., Feng, J., Liu, J., Guo, Z., Yan, S.: Joint rain detection and removal from a single image. arXiv preprint arXiv:1609.07769 (2016)

[52] Xie, J., Xu, L., Chen, E.: Image denoising and inpainting with deep neural networks. In: Advances in Neural Information Processing Systems (NIPS). (2012) 341–349

[53] Xu, L., Ren, J.S., Liu, C., Jia, J.: Deep convolutional neural network for image deconvolution. In: Advances in Neural Information Processing Systems (NIPS). (2014) 1790–1798

[54] Gatys, L., Ecker, A.S., Bethge, M.: Texture synthesis using convolutional neural networks. In: Advances in Neural Information Processing Systems (NIPS). (2015) 262–270

[55] Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision (ECCV). (2016) 694–711

[56] Quattoni, A., Torralba, A.: Recognizing indoor scenes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2009) 413–420

[57] Jokinen, L., Sampo, K.: Hel looks. https://www.hel-looks.com/ [Online; accessed 19-Oct-2017].

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

[59] Vasiljevic, I., Chakrabarti, A., Shakhnarovich, G.: Examining the impact of blur on recognition by convolutional networks. arXiv preprint arXiv:1611.05760 (2016)

[60] Hirsch, M., Schuler, C.J., Harmeling, S., Schölkopf, B.: Fast removal of non-uniform camera shake. In: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE (2011) 463–470

[61] Harmeling, S., Michael, H., Schölkopf, B.: Space-variant single-image blind deconvolution for removing camera shake. In: Advances in Neural Information Processing Systems. (2010) 829–837

[62] Ramakrishnan, S., Pachori, S., Gangopadhyay, A., Raman, S.: Deep generative filter for motion deblurring. arXiv preprint arXiv:1709.03481 (2017)

[63] Chakrabarti, A.: A neural approach to blind motion deblurring. In: European Conference on Computer Vision, Springer (2016) 221–235

[64] Schuler, C.J., Hirsch, M., Harmeling, S., Schölkopf, B.: Learning to deblur. IEEE transactions on pattern analysis and machine intelligence **38**(7) (2016) 1439–1451

[65] Sun, J., Cao, W., Xu, Z., Ponce, J.: Learning a convolutional neural network for non-uniform motion blur removal. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 769–777

[66] Li, X., Gunturk, B., Zhang, L.: Image demosaicing: A systematic survey. In: Visual Communications and Image Processing 2008. Volume 6822., International Society for Optics and Photonics (2008) 68221J

[67] Wu, X., Zhang, N.: Primary-consistent soft-decision color demosaic for digital cameras. In: Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on. Volume 1., IEEE (2003) I–477

[68] Hamilton Jr, J.F., Adams Jr, J.E.: Adaptive color plan interpolation in single sensor color electronic camera (May 13 1997) US Patent 5,629,734.

[69] Heide, F., Steinberger, M., Tsai, Y.T., Rouf, M., Pajak, D., Reddy, D., Gallo, O., Liu, J., Heidrich, W., Egiazarian, K., et al.: Flexisp: A flexible camera image processing framework. ACM Transactions on Graphics (TOG) **33**(6) (2014) 231

[70] Dong, W., Yuan, M., Li, X., Shi, G.: Joint demosaicing and denoising with perceptual optimization on a generative adversarial network. arXiv preprint arXiv:1802.04723 (2018)

[71] Zhou, R., Achanta, R., Süsstrunk, S.: Deep residual network for joint demosaicing and super-resolution. arXiv preprint arXiv:1802.06573 (2018)

[72] Yoo, D.S., Park, M.K., Kang, M.G.: Joint deblurring and demosaicing using edge information from bayer images. IEICE TRANSACTIONS on Information and Systems **97**(7) (2014) 1872–1884

[73] Glasner, D., Bagon, S., Irani, M.: Super-resolution from a single image. In: Computer Vision, 2009 IEEE 12th International Conference on, IEEE (2009) 349–356

[74] Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: 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. (2016) 1874–1883

[75] Bruna, J., Sprechmann, P., LeCun, Y.: Super-resolution with deep convolutional sufficient statistics. arXiv preprint arXiv:1511.05666 (2015)

[76] Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. (July 2017)

[77] Köhler, R., Hirsch, M., Mohler, B., Schölkopf, B., Harmeling, S.: Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In: European Conference on Computer Vision, Springer (2012) 27–40