Tampere University

Wenzhu Xing

JOINT IMAGE DEMOSAICING, DENOISING AND SUPER-RESOLUTION

Master of Science Thesis
Information Technology
Examiner: Prof. Karen Eguiazarian
April 2020

ABSTRACT

Wenzhu Xing: Joint image demosaicing, denoising and super-resolution
Master of Science Thesis
Tampere University
Master's Degree Programme
April 2020

Denoising, demosaicing and super-resolution (SR) are three important cores of image processing and these three ill-posed problems have been separately well studied in the passed decades. However, in practical applications, these three well defined problems appear simultaneously, which greatly increase the complexity of the problem. Recently, joint solution of multiple IR tasks has just begun to attract some attention. There are two types of strategies in existing joint solutions: sequential and combined. In sequential methods, a combination of existing or new denoising, demosaicing and SR methods in sequential order is used. In contrast, the combined methods mostly apply an end-to-end manner directly from the input noisy low-resolution mosaic to the final high-resolution color image.

In this thesis, first an overview of the research on image denoising, demosaicing and single image super-resolution (SISR), and their combination is given. Then, we mainly propose three joint solutions of multiple image restoration tasks, and compare these joint solutions with different metrics on two commonly used datasets. The comparison results show that the fully combined joint solution is the best selection to solve the mixture problem of multiple image restoration tasks.

Keywords: Denoising, demosaicing, single image super-resolution, convolutional neural networks (CNN), joint solutions

PREFACE

This thesis was started while I was working as a research assistant in the Computational Imaging Group of the Signal Processing Laboratory at Tampere University of Technology. I would like to take this opportunity to express my gratitude to all the people who helped me in this thesis.

My sincere and hearty thanks and appreciations go firstly to my supervisor, professor Karen Eguiazarian, whose suggestions and encouragement have given me much insight into these research works. It has been a great privilege and joy to study under his guidance and supervision. Furthermore, it is my honor to benefit from his personality and diligence, which I will treasure my whole life. My gratitude to him knows no bounds.

I am also grateful to my former colleague Ansse Saarimäki, who have kindly provided me assistance and companionship in the preparation works of this thesis.

In addition, many thanks go to my family for their unfailing love and unwavering support.

Finally, I am really grateful to all those who devote much time to reading this thesis and give me much advice, which will benefit me in my later study.

Tampere, 6th April 2020

Wenzhu Xing

CONTENTS

1 Introduction 1
2 Theoretical basics 4
2.1 Image restoration 4
2.2 Convolutional neural network 6
2.3 Image quality metrics 9
3 Related work 10
3.1 Denoising 10
3.2 Demosaicing 11
3.3 Super-Resolution 12
3.4 Mixture problems 14
4 Joint solutions of image restoration tasks 18
4.1 Joint Solutions of Two Tasks 19
4.1.1 Joint Solutions of Denoising and Demosaicing 19
4.1.2 Joint Solutions of Demosaicing and Super-Resolution 21
4.2 Joint Solutions of Denoising, Demosaicing and Super-Resolution 23
5 Experimental framework 27
5.1 Experimental setup 27
5.2 Experiments on different test datasets 28
6 Analysis of experimental results 31
6.1 Quantitative analysis 31
6.2 Qualitative analysis 36
7 Conclusion 41
References 44

LIST OF FIGURES

2.1 The butterfly image and its corresponding noisy images. The noise level is 5, 15, 25 and 50, separately. 5
2.2 **Left**: Single CCD sensor covered by a CFA. **Right**: Bayer CFA. 6
2.3 Example images produced by up-sampling with different interpolation methods. The ground truth image has been first down-sampled by a scaling factor of 1/4 to produce a low resolution input. Images b–c have been up-sampled from that LR image by a factor of 4 using the corresponding interpolation method. 6
2.4 An example of CNN illustration (ESCPN[16]) for SR. 7

3.1 The architecture of the proposed DnCNN network. 10
3.2 The network architectures. 11
3.3 VDSR Network Structure 13
3.4 The basic architecture of ESRGAN [46], where most computation is done in the LR feature space. The "basic blocks" (e.g., residual block [50], dense block [51], RRDB) can be selected or designed for better performance . . . 13
3.5 **Left**: The BN layers in residual block in SRGAN are removed. **Right**: RRDB block is used in ESRGAN deeper model and β is the residual scaling parameter. 14
3.6 Proposed network architecture. 15
3.7 Illustration of the architecture of residual blocks. 15
3.8 Illustration of the deep residual network architecture. 16
3.9 The proposed Trinity Enhancement Network. 16

4.1 The interactions between denoising and super-resolution. The denoising process leads to smooth the high frequency details. The selected denosing model is DnCNN [8] (color version) with $\sigma = 25$, and VDSR [9] for super-resolution with scale factor 2. 19
4.2 Illustration of our deep joint demosaicing and super-resolution network architecture. The network is a feed-forward fully-convolutional network that maps a low-resolution Bayer image to a high-resolution color image. Conceptually the network has three components: color extraction of Bayer image, non-linear mapping from Bayer image representation to color image representation with feature extraction, and high-resolution color image reconstruction. In this figure, the scale factor is 2. 22
4.3 The denosing strategy of DJDD [2] network. 25

4.4 Illustration of our deep joint denoising, demosaicing and super-resolution network JD_ND_MSR. 26

5.1 Data preprocessing of different mixture problems. D_n, D_m and SR denote denoising, demosaicing and super-resolution, respectively. 27

6.1 Comparison of the joint solutions of denoising and demosaicing. The first row is image13 of McMaster dataset, and the second row is image01 from Kodak dataset. The noise level of Gaussian noise is 10 and Bayer pattern is 'rggb'. 37

6.2 Comparison of the joint solutions of demosaicing and super-resolution. The first row is image1 of McMaster dataset, and the second row is image05 from Kodak dataset. The scale factor is 2 and Bayer pattern is 'rggb'. 37

6.3 Comparison of the joint solutions of denoising, demosaicing and super-resolution. Image08 from McMaster dataset. Noise level is 10, and scale factor is 2. Bayer pattern is 'rggb'. 38

6.4 Comparison of the joint solutions of denoising, demosaicing and super-resolution. Image05 from Kodak dataset. Noise level is 10, and scale factor is 2. Bayer pattern is 'rggb'. 40

LIST OF TABLES

4.1 The summary of our network architecture. The number of RRDBs is 6 and we set the number of filters $C = 256$ and $W = 64$. 22
4.2 The summary of our JD_NDMSR network architecture. The number of RRDBs is 6 and we set the number of filters $C = 256$ and $W = 64$. 26

5.1 Summary of the compared joint solutions of **Denoising** (Dn) and **Demosaicing** (Dm). JDnDm denotes that denoising and demosaicing are processed together. The networks marked by $*$ are re-implemented and trained. 28
5.2 Summary of the compared joint solutions of **Demosaicing** (Dm) and **Super-Resolution** (SR). JDmSR denotes that demosaicing and super-resolution are processed together. The networks marked by $*$ are re-implemented and trained. 29
5.3 Summary of the compared joint solutions of **Denoising** (Dn), **Demosaicing** (Dm) and **Super-Resolution** (SR). JDnDm denotes that denoising and demosaicing are processed together. JDmSR combines demosaicing and super-resolution. JDnDmSR is joint denoising, demosaicing and super-resolution. The networks marked by $*$ are re-implemented and trained. . . . 29

6.1 Quantitative comparison of different approaches on the mixture problem of joint denoising and demosaicing using dataset Kodak, McMaster [60]. The noise level is set to 10, 20 and 30. DJDD [2] do not provide the model for noise level more than 20. 32
6.2 Quantitative comparison of different approaches on the mixture problem of joint demosaicing and super-resolution using dataset Kodak, McMaster [60]. The scale factor is set to 2, 3 and 4. 33
6.3 Quantitative comparison of different approaches on the mixture problem of joint denoising, demosaicing and super-resolution using dataset Kodak, McMaster [60]. The noise level is 10 and the scale factor is set to 2. 34
6.4 Test a new DnCNN$\rightarrow$ JD_MSR on dataset Kodak, McMaster [60]. The noise level is 10 and the scale factor is set to 2. 34
6.5 The average cPSNR and SSIM results of $JD_ND_MSR^T$ on different datasets. The scale factor is 2 and the noise levels are 10, 20 and 30. 35
6.6 Average value of different image quality metrics on the McMaster testing dataset for $JD_ND_MSR^T$ trained with different cost functions. The noise level is 10 and the sale factor is 2. For cPSNR, SSIM, MS-SSIM, and cSSIM the value reported here has been obtained as an average of the three color channels. Best results are shown in bold. 35

6.7 Average value of different image quality metrics on the Kodak testing dataset for $JD_N D_M SR^T$ trained with different cost functions. The noise level is 10 and the sale factor is 2. For cPSNR, SSIM, MS-SSIM, and cSSIM the value reported here has been obtained as an average of the three color channels. Best results are shown in bold. 36

LIST OF SYMBOLS AND ABBREVIATIONS

CNN	convolutional neural network
HR	high resolution
IR	image restoration
LR	low resolution
PSNR	peak-signal-to-noise ratio
SR	super-resolution
SSIM	structural similarity index measure

1 INTRODUCTION

Because of the limitation of hardware conditions, most mobile devices, for example, digital cameras, generate degraded images rather than high quality high resolution images. The limitations comes from three design issues. Firstly, the sensor arrays contained in most digital cameras are covered by color filter arrays (CFAs, e.g. Bayer pattern), which capture only one color of red, green and blue instead of the full visible spectrum (RGB) at each pixel. Secondly, image noise is inevitable during imaging and the noise is directly proportional to pixel density of sensor. Last but not least, the size of the photon wells limits the resolution of images. Small photon wells have a low well capacity, which limits the dynamic range of the image capture. Large photon wells limit the number of pixels and thus resolution.

In order to break through the above hardware limitations, many image restoration methods are introduced to enhance the images in recent years. The goal of Image Restoration (IR) is to reconstruct a high resolution image from its degraded image. The IR tasks corresponding to above degradation forms are demosaicing, denoising, and super-resolution. As ill-posed inverse problems, above mentioned independent IR tasks have been thoroughly studied with some insurmountable problems. For denoising, most denoising algorithms smooth the high-frequency detail and texture while eliminating noise in the image. Generally, demosaicing algorithms are always unavoidable to generate some noticeable color artifacts in the high-frequency texture regions and strong edges. Since human eyes are more sensitive to luminance changes, most modern super-resolution methods only increase the resolution of the luminance channel in the YCbCr color space.

However, in practical application, the above well defined problems need to be solved simultaneously. A combination of two or more IR tasks has received much less attention in literature. Recently, joint solution of multiple IR tasks has just begun to attract some attention. These methods can be classified into two broad categories: model-based and learning-based. Learning-based approaches consider different combinations of IR tasks, such as joint demosaicing and super-resolution [1], joint demosaicing and denoising [2, 3, 4], and TENet [5]. On the other hand, ADMM is a popular method in the class of model-based methods. For example, [6] describes a unified object feature with hidden priors and the variant of ADMM to recover high-resolution color images from its noisy Bayer input.

Here we use another way to sort the research of joint inverse methods: sequential and combined. In sequential methods, a combination of existed or new denoising, demosaic-

ing and SR methods in sequential order is used. In this thesis, two kinds of sequential joint solutions are proposed, we joint existing SOTA methods in sequence, or specifically train these networks step by step. For the specific training, except for the first network, the training data are generated by the previous trained model, i.e. the output of one network is the input of its subsequent network. Therefore, the back networks not only perform its own image processing, but also fix the error of previous network. For the combined version, an end-to-end structure will be proposed for multiple tasks.

In this thesis, we mainly focus on the mixture problem of denoising, demosaicing and super-resolution. The techniques that perform these steps sequentially usually start with denoising [7], because noise will impact demosaicing and super-resolution and lead to noticeable artifacts. On the other hand, the super-resolution should be the last step. In reality, because of having bigger size, the super-resolved image will occupy more memory inside the device. Therefore, for the sequential version joint solutions, the execution order of three tasks is denoising, demosaicing and super-resolution.

We start our investigation from finding the joint solutions of two IR tasks. In order to support sufficient data to compare different joint solutions, we investigate two mixture problems: denoising and demosaicing, and joint demosaicing and SR. There are three joint solutions of the mixture problems of two tasks. The first joint solution is to apply suitable SOTA methods to combine two image processing in the sequential order. The trained models of DnCNN [8], DJDD [2], and VDSR [9] are adopted for denoising, demosaicing and SR, respectively. Unlike model-based methods, which have the flexibility to handle a variety of IR tasks with a particular degradation matrix, learning-based method need to use a specific training dataset with some degradation matrices to learn the model [10]. As a consequence, it is difficult to create one model for joint problems by a direct combination of two different learning models. To address this problem, the second joint solution is to train these networks specifically, and then exploit the specific trained models one by one. Although these two joint solutions can solve multiple tasks step by step, the defects generated by the previous image processing will affect the subsequent ones and lead to performance drop. Building on the success of deep learning based methods, we are able to solve complicated multi-task image processing problems in an end-to-end manner, which is regard as a combined network. When jointly performed, if one task produces the result that is difficult to process directly, the followed task will compensate for the middle state, and provide better final results. Thus, we suggest to perform denoising, demosaicing and SR in such a combined scheme for the mixture problem.

For the joint solutions of three tasks, in addition to the joint solutions mentioned above, we can also combine tasks partially by the combined network proposed for two tasks. In other words, we can combine denoising and demosaicing first, and sequential perform super-resolution, or execute denoising first, and process demosaicing and SR jointly.

The theoretical basics about CNN based image restoration methods are discussed in Chapter 2. To explain denoising, demosaicing and SR, general theories on Gaussian noise, Bayer mosaic, and down-sampling and up-sampling are outlined first. Then differ-

ent criterias of image quality are introduced. The last part of this chapter is the theoretical foundations of CNN.

Chapter 3 discusses related works of denoising, demosaicing and SR, and the joint solutions proposed in recent years. Chapter 4 introduces the different kinds of joint solutions of mixture problem of two or three image restoration tasks. The network for joint denoising, demosaicing and SR is proposed in this chapter, as well. The summary of experiments and the analysis of results are shown in Chapter 5 and Chapter 6, respectively. Findings are concluded in Chapter 7, which also discusses the potential applications and future research.

2 THEORETICAL BASICS

In this chapter, the theoretical basics used in this thesis are described. A brief introduction of white Gaussian noise, Bayer mosaic and down/up-sampling theory is given in 2.1. Section 2.2 explains the basics of convolutional neural networks. The image quality metrics are described in Section 2.3.

2.1 Image restoration

Image noise. Image noise is inevitable during imaging and it may heavily degrade the visual quality. It can be generated by the image sensor and the circuit of the scanner or digital camera. In this thesis, we only focus on image denoising beyond additive white Gaussian noise (AWGN). Gaussian noise is a statistical noise, whose probability density function (PDF) is equal to that of Gaussian distribution (normal distribution). In other words, the noise values of the image are Gaussian-distributed. The probability density function p of a Gaussian random variable z is given by:

$$p_G(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}}.$$

where z is the gray level, μ means the mean value and σ means the standard deviation. This normal distribution can also be regard as $\mathcal{N}(\mu, \sigma^2)$.

A special case of Gaussian noise is a white Gaussian noise, in which the values at any pair of times are identically distributed and uncorrelated. If a noise has a Gaussian distribution and its power spectral density is evenly distributed, it is called white Gaussian noise. For additive white Gaussian noise, the noise N is independent and identically distributed and drawn from a zero-mean normal distribution with variance σ^2. Then, noise N is computed as:

$$N \sim \mathcal{N}(0, \sigma^2).$$

Figure 2.1 shows the clean image and noisy images of image 'butterfly'. The far left image is the ground truth image, and the other four images are noisy images with noise levels 5, 15, 25 and 50, respectively. We can find that the noise level is higher, the image is corrupted more seriously. More information about denoising will be discussed in Section 3.1.

Bayer mosaic. In recent years, digital cameras become more and more popular, and

Ground-truth. *Noise level is 5.* *Noise level is 15.* *Noise level is 25.* *Noise level is 50.*

Figure 2.1. *The butterfly image and its corresponding noisy images. The noise level is 5, 15, 25 and 50, separately.*

have replaced traditional film-based cameras in most applications. In order to produce a color digital image, there should be at least three color components at each pixel location. This can be achieved by three CCD (Charge-Coupled Devices) or CMOS (Complementary Metal-Oxide Semiconductor) sensors, each of which receives a specific primary color. However, the limitations of space and cost cause that most digital cameras on the market use a single sensor. The sensor arrays are covered by color filter arrays (CFAs), which capture only one color of red, green and blue instead of the full visible spectrum (RGB) at each pixel. By this CFAs the associated cost and size are reduced greatly.

The left part of Figure 2.2 shows a single CCD sensor covered by a CFA. CFA consists of a set of spectral selection filters, which are arranged in a staggered pattern, so that each sensor pixel samples only one of the three primary color components. These sparsely sampled color values are called mosaic or CFA images. In order to recover a full-color image from a CFA sample, an image reconstruction process (commonly called CFA demosaicing) is needed to estimate the other two missing color values for each pixel. Among many possible CFA patterns, we focus on the widely used Bayer CFA pattern (the right part of Figure 2.2). The Bayer pattern samples the green band using a quincunx grid, while red and blue are obtained by a rectangular grid. Since green approximates the brightness perceived by the human eye, green pixels are sampled at a higher rate. More information about demosaicing is discussed in Section 3.2.

Down-sampling and up-sampling. Down-sampling and up-sampling are two fundamental and widely used image operations, which are used in image display, compression, and progressive transmission. Down-sampling is a reduction in spatial resolution while maintaining the same two-dimensional (2D) representation. It is typically used to reduce image storage and / or transmission requirements. Up-sampling is to increase the spatial resolution while maintaining a 2D representation of the image. It is usually used to zoom in on a small portion of an image, and to eliminate pixelation effects that occur when displaying low-resolution images on relatively large frames. More recently, down-sampling and up-sampling have been used in combination: in lossy compression [11], multiresolution lossless compression [12], and progressive transmission [11, 13].

Recently, some deep learning based super-resolution methods [8, 9, 14, 15] apply up-sampling before performing super-resolution. Among them, the methods [8, 9, 15] exploit residual learning, which only learn the differences between the up-sampled input and

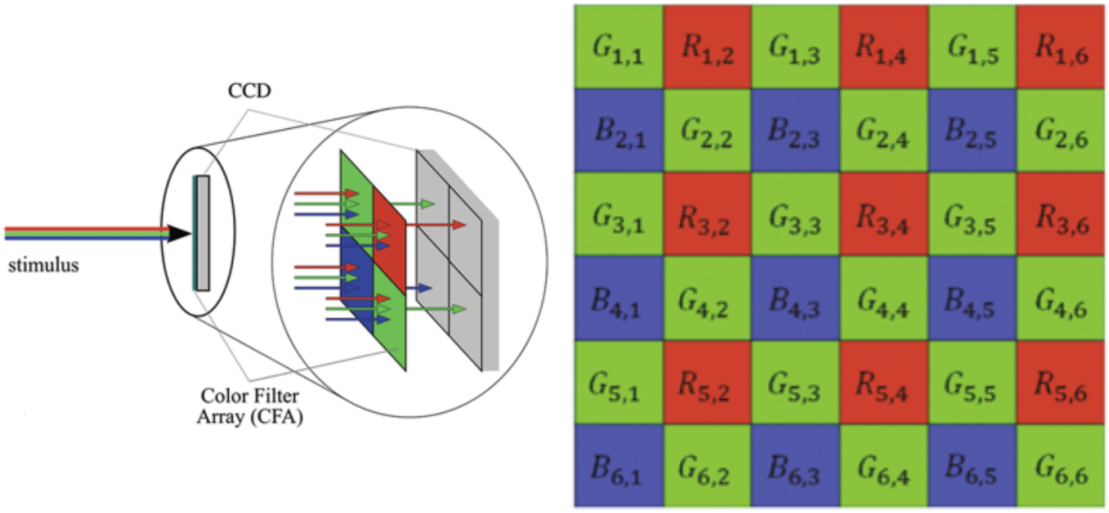


***Figure 2.2. Left**: Single CCD sensor covered by a CFA. **Right**: Bayer CFA.*

the ground truth image. The standard methods for down/up-sampling are BICUBIC and bilinear interpolation. The up-sampled images in Figures 2.3(b)–2.3(c) have all been produced by first down-sampling the original image (Figure 2.3(a)) by a factor of 4 and then up-sampled using the corresponding interpolation methods and scaling factor of 4. In this thesis, the BICUBIC interpolation is used for up-sampling processing. More information about super-resolution is discussed in Section 3.3.

Ground-truth. ***(a)** Bilinear.* ***(b)** BICUBIC.*

***Figure 2.3.** Example images produced by up-sampling with different interpolation methods. The ground truth image has been first down-sampled by a scaling factor of 1/4 to produce a low resolution input. Images b–c have been up-sampled from that LR image by a factor of 4 using the corresponding interpolation method.*

2.2 Convolutional neural network

Convolutional neural network (CNN) is a feed-forward neural network. Its artificial neurons can respond to a part of the surrounding cells in the coverage area, and it has excellent performance for large image processing. A simple CNN is a sequence of layers, which are introduced below. Figure 2.4 shows an example of CNN architecture, which is a super-resolution network ESCPN[16].

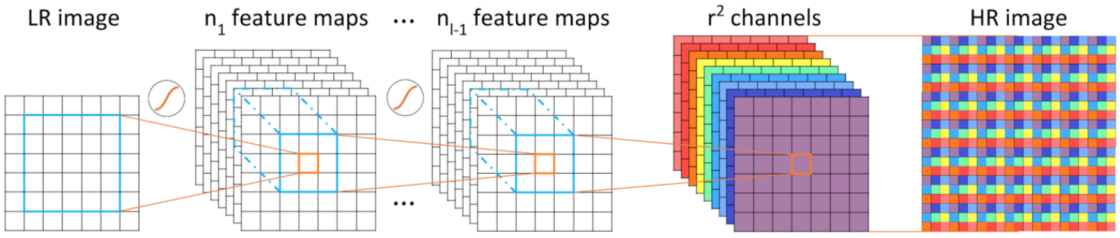


Figure 2.4. *An example of CNN illustration (ESCPN[16]) for SR.*

Convolutional layer. Each convolutional layer in a convolutional neural network consists of a number of convolutional units, and the parameters of each convolutional unit are optimized through a back-propagation algorithm. The purpose of the convolution operation is to extract different features of the input. The first convolutional layer may only extract some low-level features such as edges, lines, and corners. More layers of the network can iteratively extract more complex features from low-level features.

RELU layer. The Rectified Linear Units layer (ReLU layer) uses Rectified Linear Units (ReLU) $f(x) = max(0, x)$ as the activation function of this layer of neurons. It can enhance the non-linear characteristics of the decision function and the entire neural network without changing the convolutional layer itself. In fact, some other functions can also be used to enhance the non-linear characteristics of the network, such as hyperbolic tangent function $f(x) = thnh(x)$, $f(x) = |thnh(x)|$, and Sigmoid function $f(x) = (1 + e^{-x})^{-1}$. The ReLU function is more popular than other functions because it can speed up the neural network several times without significantly affecting the generalization accuracy of the model.

Loss layer. The loss function layer is used to decide how the training process penalizes the difference between the predicted result and the actual result of the network. It is usually the last layer of the network. Various loss functions are suitable for different types of tasks. For example, the Softmax cross-entropy loss function is often used to select one of K categories, and the Sigmoid cross-entropy loss function is often used for multiple independent binary classification problems. Euclidean loss function is often used in the problem that the value range of the label is arbitrary real number.

For an error function ε, the loss for a patch P can be written as :

$$L^{\varepsilon}(P) = \frac{1}{N} \sum_{p \in P} \varepsilon(p). \tag{2.1}$$

where N is the number of pixels in the patch.

The most common loss function used in image processing tasks like SR, is the l_2 loss, which is defined as the square of l_2 norm of the difference between the processed patch and the ground truth. Equation 2.1 for l_2 is simply:

$$L^{l_2}(P) = \frac{1}{N} \sum_{p \in P} |x(p) - y(p)|^2. \tag{2.2}$$

where p is the index of the pixel and P is the patch; $x(p)$ and $y(p)$ are the values of the pixels in the processed patch and the ground truth, respectively.

There are CNN based methods [17] applying l_1 error instead of l_2. Then, Equation 2.1 for l_1 is simply:

$$L^{l_1}(P) = \frac{1}{N}\sum_{p\in P}|x(p) - y(p)|. \tag{2.3}$$

SSIM (structural similarity index measure) is used to measure the structural similarity between two images. SSIM for pixel p is defined as:

$$SSIM(p) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \cdot \frac{2\sigma_{xy} + C_2}{\sigma_x^2 + \sigma_y^2 + C_1} = l(p) \cdot cs(p). \tag{2.4}$$

Means and standard deviations are computed with a Gaussian filter G_{σ_G}, whose standard deviation is σ_G. The loss function for SSIM can be then written as:

$$L^{SSIM}(P) = \frac{1}{N}\sum_{p\in P} 1 - SSIM(p). \tag{2.5}$$

Since the choice of σ_G influences the quality of the performance of a network that is trained with SSIM, an advanced form of SSIM is proposed, MS-SSIM. Given a dyadic pyramid of M levels, MS-SSIM is defined as:

$$MS\text{-}SSIM(p) = l_M^{\alpha}(p) \cdot \prod_{j=1}^{M} cs_j^{\beta_j}(p). \tag{2.6}$$

where l_M and cs_j are the terms defined in Equation 2.4, at scale M and j, respectively. The MS-SSIM loss for patch p IS computed as:

$$L^{MS\text{-}SSIM}(P) = \frac{1}{N}\sum_{p\in P} 1 - MS\text{-}SSIM(p). \tag{2.7}$$

Zhao et al. proposed a combination loss Mix1 [17], which captures both MS-SSIM error and l_1 error:

$$L^{Mix1}(P) = \alpha \cdot L^{MS\text{-}SSIM} + (1-\alpha) \cdot G_{\sigma_G^M} \cdot L^{l1}. \tag{2.8}$$

Similarly to Equation 2.8, another mix loss function which combines MS-SSIM error and l_2 error is:

$$L^{Mix2}(P) = \alpha \cdot L^{MS\text{-}SSIM} + (1-\alpha) \cdot G_{\sigma_G^M} \cdot L^{l2}. \tag{2.9}$$

where the parameter α need to be tuned by experiments.

In this thesis, all methods we applied or designed are convolutional neural networks. More examples of CNN based methods are showed in Chapter 3. We will train our final network with all above cost functions in Section.

2.3 Image quality metrics

There are various image quality metrics to evaluate the performance of networks. In this thesis, we use four metrics, PSNR (peak-signal-to-noise ratio), SSIM, MS-SSIM and CSSIM [18].

PSNR is based on MSE (mean square error), which is calculated as:

$$MSE(G,T) = \frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}(G_{h,w} - T_{h,w})^2. \tag{2.10}$$

where G is reference (i.e. ground truth image) and T is test image, whose size is $h \times w$. PSNR can be then written as:

$$PSNR(G,T) = 10\log_{10}(\frac{(Max_G)^2}{MSE(G,T)}). \tag{2.11}$$

where Max_G is the highest possible brightness value for G. For a 8-bit image, the value of Max_G is 255.

SSIM and MS-SSIM are defined in Equation 2.4 and Equation 2.6, respectively. CSSIM is also a modification of SSIM. However, compared with other metrics which work only on gray-scale images, CSSIM can work on color images by the different weights of color components (Cb and Cr channels) and intensity component (Y channel). More details of CSSIM can be found in [18]. For PSNR, SSIM and MS-SSIM, we average the values of three color channels to obtain the final values of color images.

The theoretical concepts described above will be used in later chapters. More application examples of CNN are given in Chapter 3. The brief introduction of white Gaussian noise, Bayer mosaic and down/up-sampling theory helps to understand the content of Chapter 4. Different loss functions are applied in Chapter 5 and the evaluation metrics are used in Chapter 6.

3 RELATED WORK

In this Chapter, firstly, related works of denoising, demosaicing and super-resolution are presented. Then, the literature of mixture problems of above three tasks is reviewed briefly.

3.1 Denoising

Image noise is inevitable during imaging and it may heavily degrade the visual quality. There have been several attempts to handle the denoising problem. Early methods [19, 20, 21] apply hand-craft features and algorithms to eliminate noise. Advance methods usually use effective image priors such as self-similarity [22, 23, 24] and sparse representation [25]. Among them, BM3D [23] is one of state-of-the-art methods, and is always used as a benchmark in the comparison of denosing methods. However, in recent years, more and more machine-learning based methods [8, 26, 27, 28, 29] successfully used in denoising, consisting of CNNs trained with noisy data and clean data. We select DnCNN [8] for further experiments and comparative analysis.

DnCNN. The denoising CNN model, DnCNN [8], is proposed in 2017 by Zhang et al. DnCNN uses the same model for three different tasks: multiscale SR, Gaussian denoising and JPEG deblocking. The network structure is almost identical to VDSR [9], with the exception of added batch normalization layers in the hidden layers. In SR task it performs almost identically with VDSR, while still performing well in the denoising and deblocking tasks.

For model learning, Zhang et al. adopt the residual learning formulation, and incorporate it with batch normalization for fast training and improved denoising performance. The input of our DnCNN is a noisy observation $\mathbf{y} = \mathbf{x} + \mathbf{v}$. The network target is to train a residual mapping $R(\mathbf{y}) \approx \mathbf{v}$, then attaining the clean image $\mathbf{x} = \mathbf{y} - R(\mathbf{y})$.

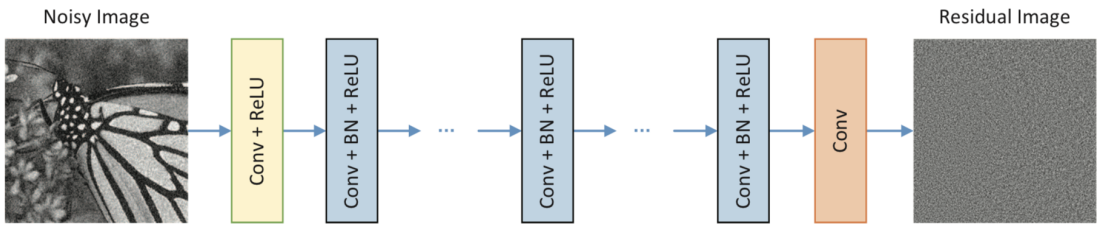


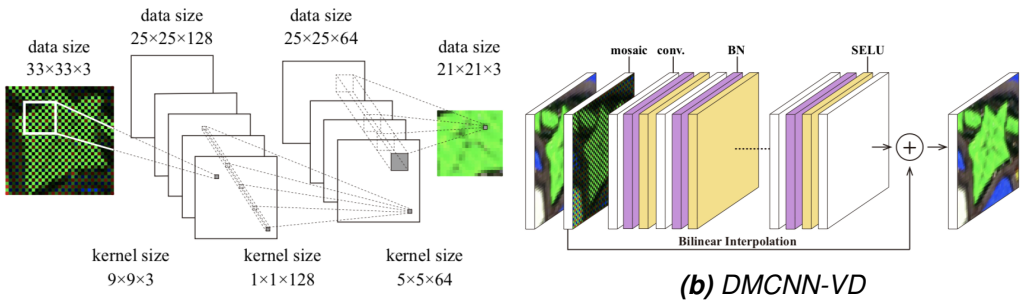
Figure 3.1. *The architecture of the proposed DnCNN network.*

Figure 3.1 illustrates the architecture of the proposed DnCNN for learning $R(\mathbf{y})$. The DnCNN model has two main features: the residual learning formulation is adopted to learn R(y), and batch normalization is incorporated to speed up training as well as to boost the denoising performance. By incorporating convolution with ReLU, DnCNN can gradually separate image structure from the noisy observation through the hidden layers.

3.2 Demosaicing

To reduce manufacturing costs, most digital camera sensors capture only one color (red, green and blue) at each pixel. The camera sensor is covered by color filter arrays (CFAs). The Bayer filter is the most popular CFA. Demosaicing is the process of interpolating full-resolution color image from incomplete color samples output by this kind of image sensor. Most demosaicing are designed specifically for the Bayer CFA. Existing algorithms can be mainly classified into two categories: model-based methods [30, 31, 32, 33, 34], which recovery images mathematical models and image priors in the spatial-spectral domain; and learning-based methods [34, 35], which are based on process mapping learned from abundant training data. Recently, deep learning is more and more popular in image restoration tasks. There are some deep learning methods [36, 37, 38] of demosaicing attaining competitive performance. Among them, DMCNN [38] is one of SOTA demosaicing algorithm.

DMCNN. Syu et al. [38] propose two networks for demosaicing: DMCNN which has a structure inspired by SRCNN [39], and DMCNN-VD, a deeper network inspired by VDSR [9]. The input Bayer mosaic is converted into 3-channel images before processing. For DMCNN they extend the Bayer CFA to 3 channels without any interpolation and replace the missing values with zeros. For DMCNN-VD they used the zero-filled image as network input but utilize a bilinear interpolated image for the residual connection. Figure 3.2 gives the network architectures of DMCNN and DMCNN-VD.



(a) *DMCNN* ***(b)*** *DMCNN-VD*

Figure 3.2. *The network architectures.*

DMCNN consists of three convolutional layers: feature extraction layer, non-linear mapping layer and reconstruction layer. However, DMCNN-VD is a very deep convolutional network, which includes 20 layers and applies the residual learning strategy. In both quantitative comparison and qualitative comparison, it is obvious that the deeper network

DMCNN-VD performs better than DMCNN. The resulting images of DMCNN-VD have less artifacts than the ones of DMCNN.

Although DMCNN-VD outperforms many existing demosaicing algorithms, we do not use it in further experiments. For demosaicing, a simpler deep network DJDD [2] is exploited, which performs good enough and can demosaic not only a noise-free image but also a noisy image.

3.3 Super-Resolution

Super-resolution aims to recover a high-resolution image from its corresponding low-resolution images. According to the number of low-resolution input images, the methods of super-resolution can be classified to multi-image SR (MISR) and single-image SR (SISR). In this thesis, we mainly concern about single-image super-resolution. Traditional methods utilize interpolation approaches based on sampling theory. Recently, learning methods [40, 41, 42] model a mapping from low-resolution to high-resolution by embedding prior knowledge from large datasets. Lately, CNN was applied to improve the accuracy and quality of resulting images. Inspired by the success of CNN in image classification tasks, there are more and more CNN methods [9, 14, 15, 43, 44, 45, 46] proposed for SISR. In our experiments, VDSR [9] is one selection of SOTA methods to solve the mixture problems. On the other hand, the network structure of ESRGAN [46] inspires the design of our comprehensive network for mixture problems.

SOTA method VDSR. Kim et al. [9] introduced the single image super-resolution method using very deep convolutional networks (VDSR) in 2016. As the first deep CNN based SISR, the VDSR made some improvements on SRCNN structure [14]. Inspired by the very deep convolutional network for image recognition in [47], the authors proved that the addition of the depth of network improved the accuracy of the model. Therefore, instead of only 3 convolutional layers in SRCNN, VDSR has 20 convolutional layers. Since the number of the network layers increased, the receptive field changed to 41×41 with the same filter size 3×3 in each layer. The bigger receptive field size means more image information is contained, which is helpful to image reconstruction. In order to keep the size of the feature map of each layer identical, VDSR applied zero-padding before convolutions. For SRCNN, the network kept all contents of the input image and through all values in the whole network, which required long-term memory. VDSR uses only the residual image, defined as the difference between desired high resolution image and the interpolated low resolution image. Figure 3.3 shows the VDSR network structure.

SOTA method ESRGAN. Various SISR networks attained great success, which are focus on improving Peak Signal-to-Noise Ratio (PSNR). However, these PSNR-oriented networks perform excellently at the expense of over-smoothed resulting images. Several perceptual-driven methods apply Generative Adversarial Network (GAN) to improve the visual quality of resulting images. In 2017, Ledig wt al. [45] propose SRGAN model that uses perceptual loss and adversarial loss to favor outputs residing on the manifold of

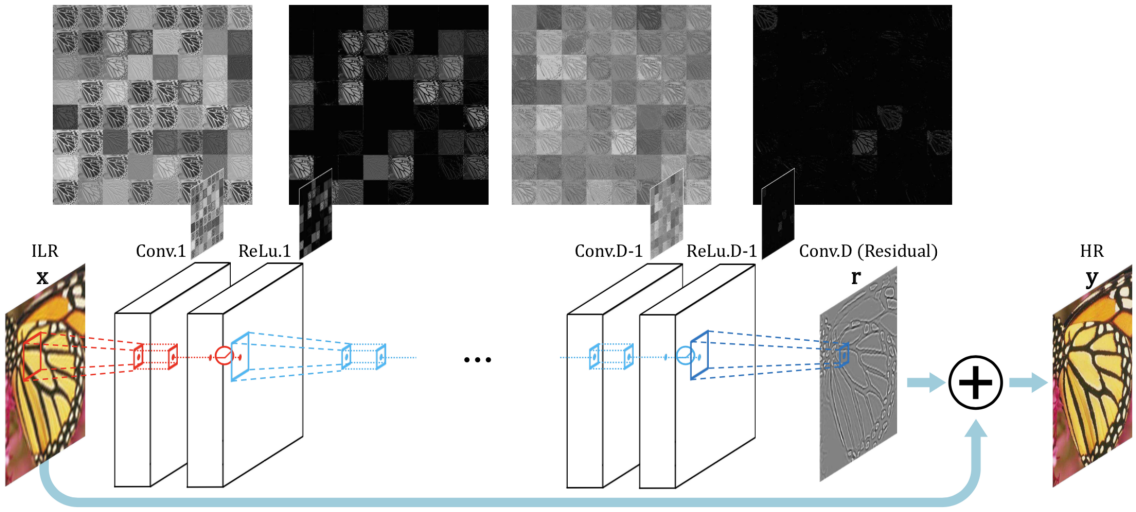


Figure 3.3. *VDSR Network Structure*

natural images.

Based on SRGAN, Wang et al. [46] enhanced the basic network architecture by replacing the original residual blocks with the Residual-in-Residual Dense Block (RRDB), which removes the batch normalization. Meanwhile, Wang et al. improve the discriminator using Relativistic average GAN (RaGAN) [48], which leads to sharper edges and more detailed textures. Compared with PSNR-oriented methods, perceptual-driven approaches improve the visual quality by minimizing the error in a feature space rather than pixel space. In order to be closer to perceptual similarity, perceptual loss [49] is proposed. Contrary to the convention, Wang et al. propose a more effective perceptual loss, which use the VGG features before the activation layers instead of after activation as in SRGAN. In addition, the network interpolation strategy is used to balance perceptual quality and PSNR.

ESRGAN keeps the high-level architecture design of SRGAN (see Figure 3.4), and use a novel basic block namely RRDB as depicted in Figure 3.5. The proposed Residual-in-Residual Dense Block (RRDB) combines multi-level residual network and dense connections, i.e. residual learning in different levels. The RRDB is exploited in our proposed network architecture (Section 4.1 and Section 4.2).

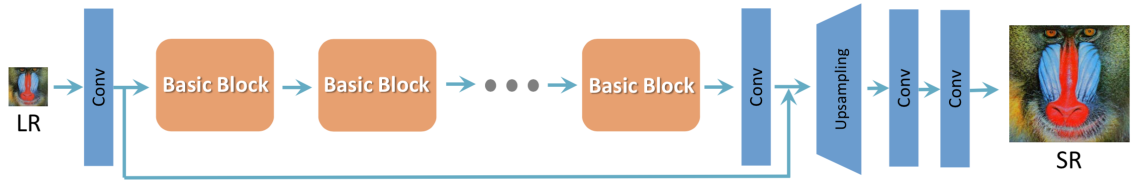


Figure 3.4. *The basic architecture of ESRGAN [46], where most computation is done in the LR feature space. The "basic blocks" (e.g., residual block [50], dense block [51], RRDB) can be selected or designed for better performance*

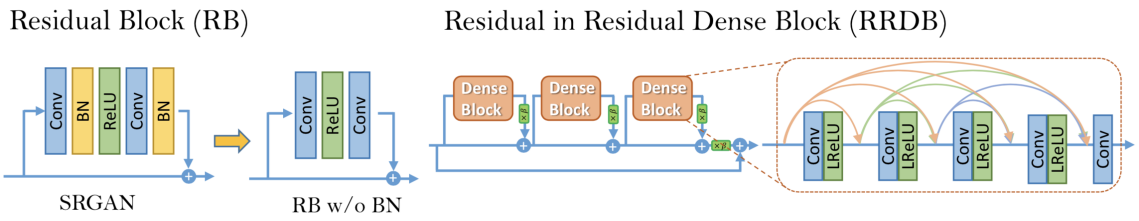


***Figure 3.5. Left**: The BN layers in residual block in SRGAN are removed. **Right**: RRDB block is used in ESRGAN deeper model and β is the residual scaling parameter.*

3.4 Mixture problems

In reality, the above well defined problems appear simultaneously, and it is the combined problems that we need to solve. In last few decades, denoising, demosaicing and super-resolution have been independently studied and applied in sequential steps. However, there are some limitations and problems for each task. For denoising, most denoising algorithms smooth the high-frequency detail and texture while eliminating noise in the image. Generally, demosaicing algorithms are always unavoidable to generate some noticeable color artifacts in the high-frequency texture regions and strong edges. Since human eyes are more sensitive to luminance changes, almost all modern super-resolution methods only increase the resolution of the luminance channel in the YCbCr color space.

Therefore, these errors are accumulated when these algorithms are separately applied. When we process super-resolution on the denoised images, the blur effect will be magnified and reduce the quality of resulting images (Figure 4.1). Sequential application of super-resolution algorithms after demosaicing algorithms leads to visually disturbing artifacts in the final output (the second row in Figure 6.2). This is because the super-resolution algorithms regards the artifacts such as color zippering caused by demosaicing algorithms as a valid signal of the input image. And the super-resolution algorithms only processes monochromatic image, and ignores the artifacts in chroma channels. In addition, SR algorithms not only enhance the image details and texture, but also enlarge the unexpected noise, blur and artifacts produced by a previous processing. For joint denoising and demosaicing, the complexity of demosaicing is increased by the presence of noise. The estimation of the edge direction in the noisy data is not reliable, which will cause obvious artifacts in the demosaic image. And some information is removed with the elimination of noise by denoising processing, the demosaicing algorithm sequential performs on denoised images may lead to color deviation and blur (Figure 6.1).

In recent years, the mixture problem of multiple image distortion is concerned, such as joint denoising and super-resolution [2, 4, 52, 53, 54], joint demosaicing and super-resolution [1, 55, 56], and joint denosing and SR [57]. But the research on the mixture problem of denosing, demosaicing and SR is lacking special attention. In 2019, Qian et al. [5] proposed a trinity network to solve this composite problem jointly. Among them, DJDD [2] is selected in our comparison experiments (see Section 5.2). The network structures of [1] and [5] inspire the design of our comprehensive network for mixture problems.

Joint demosaicing and denosing. In 2016 Gharbi et al. [2] proposed a standard feed-forward network architecture for demosaicing and denoising. Instead of using hand-crafted filters depended on hard-code heuristics, the authors trained a deep convolutional neural network (CNN) on both standard datasets and a new dataset composed of patches with artifacts. The proposed network is capable of handling a wide range of noise levels and removing the artifacts, such as luminance artifacts and color moiré, i.e. joint demosaicing and denoising. The quantitatively experimental results demonstrate that this network generates state-of-the-art performance both on noisy and noise-free data.

The proposed network architecture is showed in Figure 3.6, the mosaiced array M with the estimated noise level σ are the input of the network, and the output is a RGB image O with the same size of the input. This network is composed with 16 convolutional layers, with 64 feature maps and 3×3 kernel size for each layer. To correct the artifacts produced on some hard cases, the authors built a new dataset which contains more challenging patches, to fine-tune the parameters of the model in order to improve the performance on reducing artifacts.

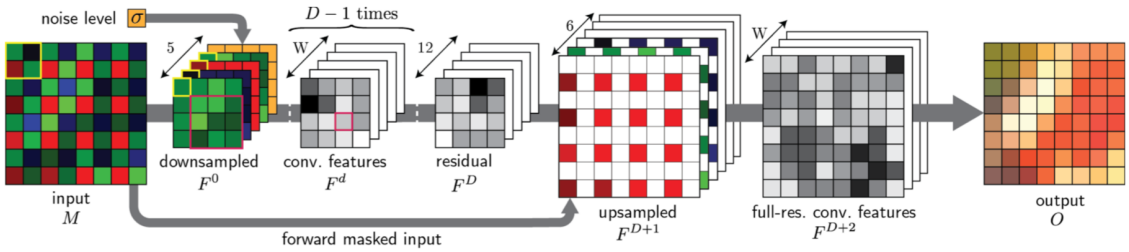


Figure 3.6. *Proposed network architecture.*

Joint demosaicing and super-Resolution. Zhou et al. [1] introduced the joint demosaicing and super-resolution method using a deep residual network in 2018. This network has a mono-channel Bayer image with low-resolution as input and outputs a high-resolution RGB image. In order to solve super-resolution, 24 residual blocks were used for feature extraction and non-linear mapping. The authors removed the batch normalization layers in the original residual blocks [50] and replaced ReLU with PReLU. The architecture of residual block is showed in Figure 3.7 and the network structure is given in Figure 3.8.

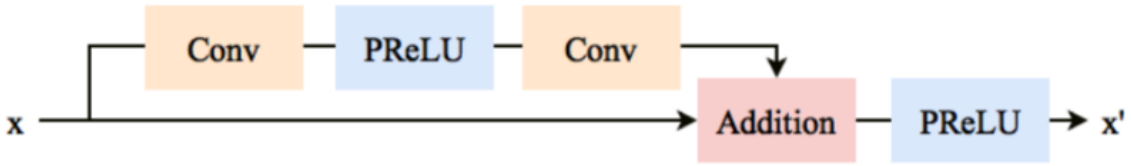


Figure 3.7. *Illustration of the architecture of residual blocks.*

These two networks JDD (Figure 3.6) and JDSR (Figure 3.8), are both based on deep CNN and with the mosaiced array as input and RGB image as output. The size of input and output of the network for joint demosaicing and denoising is identical. However, the later network has a bigger size output, i.e. super-resolution. In addition to this, first struc-

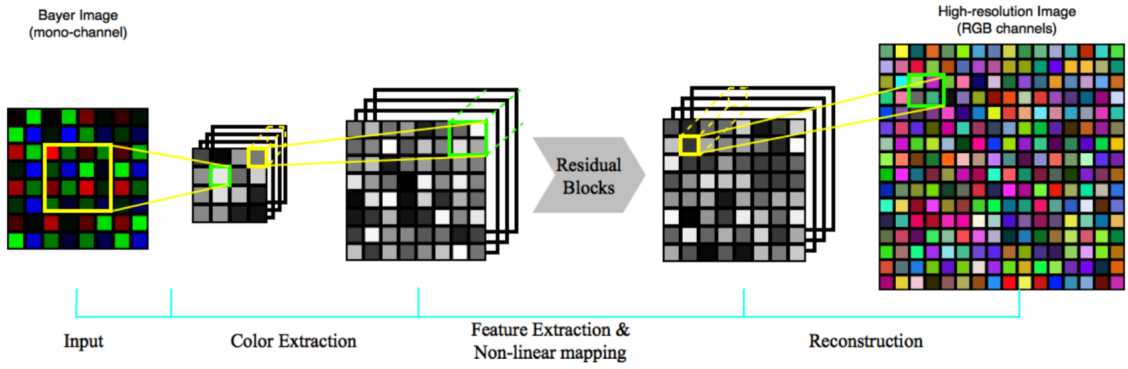


Figure 3.8. *Illustration of the deep residual network architecture.*

ture is composed of 15 convolutional layers. In contrast, the later one contains 24 residual blocks, which used more convolutional layers but just learned residual information only.

Joint demosaicing, denoising and Super-Resolution. TENet [5] is the newest joint solution for Demosaicking, Denoising and Super-Resolution, which is proposed by Qian et al. in 2019. They first decided the order of execution tasks, i.e. denoising, super-resolution and demosaicing. Then, the network is divided to two parts: the mapping of joint denoising and super-resolution, and the demosaic mapping. In order to optimize the performance of the network, the middle state of the network should be concerned. After the first mapping, the SR loss on raw image is calculated. The architecture of the network is inspired by the deep network ESRGAN [46]. Meanwhile, they contributed a novel dataset PixelFhift200, which applies advanced pixel shift technology to perform a full color sampling of the image. These artifacts-free images lead to better training results for demosaicing related tasks. The structure of Trinity Enhancement Network is shown in Figure 3.9.

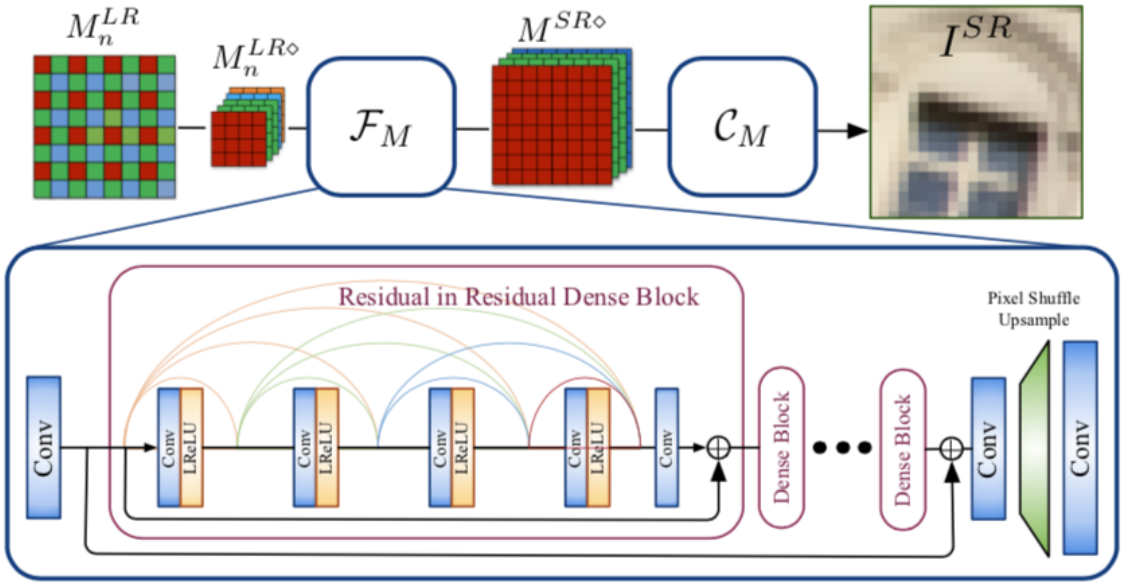


Figure 3.9. *The proposed Trinity Enhancement Network.*

Based on these existing methods, we tentatively proposed some joint solutions for the mixture problem in Chapter 4.

4 JOINT SOLUTIONS OF IMAGE RESTORATION TASKS

In this chapter, we generate joint solutions of multiple image restoration tasks by applying directly or modifying the existing SOTA methods introduced in Chapter 3. In total, there are three types of joint solutions to solve the mixture problem. The first joint solution is selecting one existed state-of-the-art (SOTA) CNN based method for each task, and then applying the trained models to solve the problems one by one. In this solution, for a given corrupted input image I_{Input}, its corresponding output image I_{Output} can be written as a composite function:

$$I_{Output} = M_C(M_B(M_A(I_{Input}))). \tag{4.1}$$

where M is the selected SOTA method, A, B and C are image restoration tasks, i.e. M_A means the SOTA method for task A.

The second joint solution solves the problems in sequential order as well. In contrast, this joint solution trains a specific CNN for each image restoration task, instead of using the trained model directly. Similarly, the integrated function for this solution is expressed as:

$$I_{Output} = \mathcal{F}_C(\mathcal{F}_B(\mathcal{F}_A(I_{Input}))). \tag{4.2}$$

where $\mathcal{F}$ denotes the mapping implemented by a deep convolutional neural network.

Although these two joint solutions can solve multiple tasks step by step, the defects generated by the previous image processing will affect the subsequent ones and lead to performance drop. For example, denoising algorithms not only eliminate noise, but also smooth details and texture in the image. When we process super-resolution on the denoised images, the blur effect will be magnified and reduce the quality of resulting images, as shown in Figure 4.1.

In order to minimize the effects caused by task interaction, the last joint solution exploits only one end-to-end network to solve multiple of problems at once. Therefore, this convolutional neural network is a combined version of all specific ones. Then, the synthesized function is:

$$I_{Output} = \mathcal{F}(I_{Input}). \tag{4.3}$$

In this thesis, we mainly focus on the mixture problem of denoising, demosaicing and super-resolution. We propose to process denoising before other two tasks like in the

(a) *Ground-truth image from Urban100.* ***(b)*** *SR result of denoised LR image.* ***(c)*** *SR result of clean LR image.*

Figure 4.1. *The interactions between denoising and super-resolution. The denoising process leads to smooth the high frequency details. The selected denosing model is DnCNN [8] (color version) with $\sigma = 25$, and VDSR [9] for super-resolution with scale factor 2.*

existing method [5], because a noise will impact demosaicing and super-resolution and lead to noticeable artifacts. On the other hand, the super-resolution should be the last step. In reality, because of having bigger size, the super-resolved image will occupy more memory inside the device. Therefore, for the sequential version joint solutions, the execution order of three tasks is denoising, demosaicing and super-resolution.

In this Chapter, the joint solutions of two tasks, joint denoising and demosaicing, and joint demosaicing and super-resolution, are first described in Section 4.1. Continuously, the joint solutions of denoising, demosaicing and super-resolution are presented in Section 4.2.

4.1 Joint Solutions of Two Tasks

We first start from finding the joint solutions of two image restoration (IR) tasks. As mentioned above, the order of processing of three IR tasks is denoising and demosaicing and super-resolution. Denoising and demosaicing can be processed on the mobile devices, then super-resolved images on the picture viewer. Or, digital cameras only generate denoised images and demosaicing and super-resolution is executed by a external image processor. Under this premise, the joint solutions of denoising and demosaicing (DnDm) and joint solutions of demosaicing and super-resolution (DmSR) are discussed in this part.

4.1.1 Joint Solutions of Denoising and Demosaicing

In the blend problem of denoising and demosaicing, for a given color image I, its noisy raw mosaic image I_M^N is the input, and the desired output shold be a clean color image $\tilde{I}$.

Apply existing SOTA methods. The SOTA methods selected for denosing and demosaicing are DnCNN [8] and DJDD [2], respectively. Both two SOTA methods publicly provide implementation code and trained models. We input noisy mosaic images into DnCNN first, and exploit DJDD to demosaic the noise removed images. As the compos-

ite function presented in equation (4.1), this formula can be materialized as:

$$\tilde{I} = DJDD(DnCNN(I_M^N)), \tag{4.4}$$

where DnCNN trained model is a gray scale version, because the raw input has only one color channel. And the DJDD model used in this joint solution is a noise-free version, which attains better performance than noisy version without noise input.

Train specific CNNs. Considering the published models of SOTA methods are only trained for single task, the specific networks are trained with specific data in this joint solution. In order to solve denosing and demosaicing in sequence, a denoise network D_N is trained first to generate denoised raw images, which are the input data of the following demosaic network D_M. Therefore, the compound equation (4.2) is transferred to:

$$\tilde{I} = \mathcal{F}_{D_M}(\mathcal{F}_{D_N}(I_M^N)). \tag{4.5}$$

For fair comparison, the structures of networks are same with the SOTA methods (DnCNN and DJDD). Since the output of previous network (D_N) is the input of back network (D_M), these two convolutional networks must be trained separately, and the previous one should be trained first. The denoising is first performed to produce noise-free mosaic image $\widetilde{I_M} = \mathcal{F}_{D_N}(I_M^N)$, i.e. the training target of D_N is corresponding raw mosaic image I_M. Then, the mapping of D_M network directly works on the denoised raw image $\widetilde{I_M}$, the output of D_N. So, the loss functions of networks D_N and D_M are computed as:

$$\begin{aligned} \mathcal{L}_{D_N} &= \mathcal{L}(\mathcal{F}_{D_N}(I_M^N) - I_M), \\ \mathcal{L}_{D_M} &= \mathcal{L}(\mathcal{F}_{D_M}(\mathcal{F}_{D_N}(I_M^N)) - I). \end{aligned} \tag{4.6}$$

where $\mathcal{L}$ is the loss function, such as the ℓ_1-norm loss and ℓ_2-norm loss.

Combined CNN. Even though training specific CNN for each task can solve the mixture problem of two tasks, it is complex in operation and redundant in network structure. Some layers in these two networks are doing same thing and can be combined together. It makes sense to merge two networks into an end-to-end CNN. According to equation (4.3), the noise-free color image $\tilde{I}$ can be obtained directly by the joint network for denoising and demosaicing:

$$\tilde{I} = \mathcal{F}_{JD_ND_M}(I_M^N)). \tag{4.7}$$

The structure of network JD_ND_M is identical to the structure of DJDD [2]. Although there is a public trained DJDD model, we have trained another specific model with our databases for comparison with other joint solutions fairly. The joint denoising and demosaicing mapping is performed directly on the noisy mosaic images to obtain noise-free color images. The loss function of network JD_ND_M is computed as:

$$\mathcal{L}_{JD_ND_M} = \mathcal{L}(\mathcal{F}_{JD_MD_N}(I_M^N) - I). \tag{4.8}$$

4.1.2 Joint Solutions of Demosaicing and Super-Resolution

For joint demosaicing and super-resolution, we have to convert the low-resolution (LR) mosaic image I_M^{LR} into high-resolution (HR) color image I^{HR}, i.e. $\tilde{I}$.

Applying existing SOTA methods. Similar to joint denoising and demosaicing, we choose two SOTA methods to process demosaicing and super-resolution separately. For demosaicing, the selection of trained model is still DJDD [2] noise-free version. For super-resolution, we apply VDSR [9], which only works on luminance channel (y channel) and adopts BICUBIC interpolation on chroma channels. Then, for a given low-resolution (LR) mosaic image I_M^{LR}, its corresponding HR color image I^{HR} is computed as:

$$\tilde{I} = I_{HR} = VDSR(BI(DJDD(I_M^{LR}))). \tag{4.9}$$

where BI means BICUBIC interpolation. VDSR only learns residual image between BICUBIC scaled input and desired HR output, i.e. residual learning. Therefore, before super-resolution, the demosaiced LR image generated by DJDD need to be up-sampled to the desired resolution first.

Training specific CNNs. In this joint solution, the demosaicing CNN network (D_M) and the SR CNN network (SR) are trained in a sequence and separately. The output image can be computed as:

$$\tilde{I} = I_{HR} = \mathcal{F}_{SR}(\mathcal{F}_{D_M}(I_M^{LR})). \tag{4.10}$$

In order to compare fairly, for demosaicing, the structure of CNN is similar to DJDD, and for SR, a modified VDSR model is trained. Since VDSR only works on luminance channel of YCbCr color space, we need to train a VDSR model with RGB input instead of single channel input. The mosaic low-resolution input I_M^{LR} first processed by D_M network to produce corresponding color LR image ($\widetilde{I_{LR}}$). Before performing the modified VDSR, the output of the demosaicing network should be up-scaled by BICUBIC Interpolation. The specific loss functions of two CNNs are:

$$\begin{aligned} \mathcal{L}_{D_M} &= \mathcal{L}(\widetilde{I_{LR}} - I_{LR}) = \mathcal{L}(\mathcal{F}_{D_M}(I_M^{LR}) - I_{LR}), \\ \mathcal{L}_{SR} &= \mathcal{L}(\mathcal{F}_{SR}(\widetilde{I_{LR}}) - I) = \mathcal{L}(\mathcal{F}_{SR}(\mathcal{F}_{D_M}(I_M^{LR})) - I). \end{aligned} \tag{4.11}$$

where I_{LR} is the low-resolution image of corresponding ground-truth I.

Combined CNN. Our combined CNN of joint demosaicing and super-resolution (JD_MSR) is illustrated in Figure 4.2. The network structure is inspired by another deep joint demosaicing and SR network [1], but we have replaced the 24 residual blocks (RB) [50] in original network with 6 residual-in-residual dense blocks (RRDB) [46]. The details of RB and RRDB are described in Section 3.3 and Figure 3.5. Similar to [1], the network architecture consists of three parts: color extraction, feature extraction and reconstruction. The Bayer input is first down-sampled by a convolutional layer with 2×2 kernel size and 2×2 stride size, resulting in a quarter-resolution multi-channel image. The color extraction step includes one convolutional layer with big filter (256), and one deconvolutional layer

with 2×2 stride size to upscale the feature maps to the prime resolution. Feature extraction stage applies 6 RRDBs and a big filter (256) convolution, which is prepared for the upsampling. In the reconstruction part, the deconvolutional layer is used again to convert the extracted features into full resolution features. The stride size of this deconvolutional layer should be equal to the scale factor. The following is the final convolutional layer to generate the desired resolution color image. The summary of the network architecture is shown in Table 4.1.

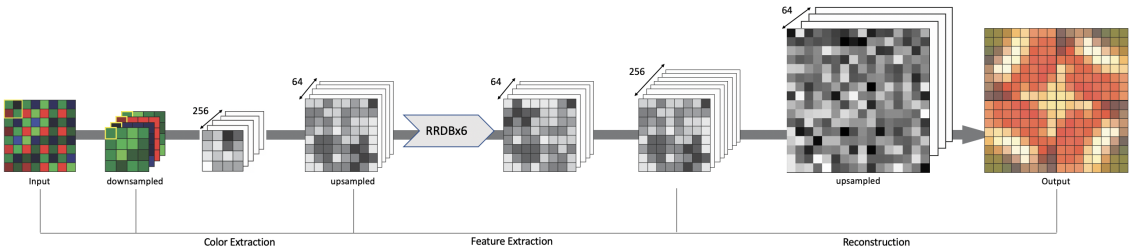


Figure 4.2. *Illustration of our deep joint demosaicing and super-resolution network architecture. The network is a feed-forward fully-convolutional network that maps a low-resolution Bayer image to a high-resolution color image. Conceptually the network has three components: color extraction of Bayer image, non-linear mapping from Bayer image representation to color image representation with feature extraction, and high-resolution color image reconstruction. In this figure, the scale factor is 2.*

Table 4.1. *The summary of our network architecture. The number of RRDBs is 6 and we set the number of filters* $C = 256$ *and* $W = 64$*.*

Stage	Layer	Output Shape
Input	Input (Bayer image)	$h \times w \times 1$
Color Extraction	Down-sampling	$\frac{h}{2} \times \frac{w}{2} \times 4$
	Conv	$\frac{h}{2} \times \frac{w}{2} \times C$
	Up-sampling	$h \times w \times \frac{C}{4}$
Feature Extraction	RRDB	$h \times w \times W$
	...	...
	RRDB	$h \times w \times W$
	Conv	$h \times w \times C$
Reconstruction	Up-sampling	$(sf \times h) \times (sf \times w) \times W$
	Conv	$(sf \times h) \times (sf \times w) \times 3$
Output	Output (color image)	$(sf \times h) \times (sf \times w) \times 3$

According to equation (4.3), the high-resolution color output I_{HR} is generated by the following function:

$$\tilde{I} = I_{HR} = \mathcal{F}_{JD_MSR}(I_M^{LR})), \tag{4.12}$$

and its loss function is:

$$\mathcal{L}_{JD_MSR} = \mathcal{L}(\mathcal{F}_{JD_MSR}(I_M^{LR}) - I). \tag{4.13}$$

4.2 Joint Solutions of Denoising, Demosaicing and Super-Resolution

Since the joint solutions of two tasks are investigated in Section 4.1, we cannot only apply the joint solutions proposed in equations (4.1-4.3), but also exploit new joint solutions to solve the mixture problems of three kinds of image processing. The new joint solution combines two tasks first, then sequentially process another task. And there are two versions of this new joint solution, which we call here 'existing methods' and 'specific training'. For the 'existing methods' version, the function is computed as:

$$\begin{aligned} I_{Output} &= M_C(M_{AB}(I_{Input})), \\ &or \\ I_{Output} &= M_{BC}(M_A(I_{Input})). \end{aligned} \tag{4.14}$$

where M is an existing method, A, B and C are image restoration tasks. M_{AB} means the method that processes A and B jointly. Similarly, the function of a 'specific training' is:

$$\begin{aligned} I_{Output} &= \mathcal{F}_C(\mathcal{F}_{AB}(I_{Input})), \\ &or \\ I_{Output} &= \mathcal{F}_{BC}(\mathcal{F}_A(I_{Input})). \end{aligned} \tag{4.15}$$

where $\mathcal{F}$ denotes the specific trained mapping.

This kind of joint strategy is meaningful and helpful in realistic applications. Considering the limitation of storage of digital cameras, it is sensible to process denoising and demosaicing on the digital cameras first and super-resolve the low resolution images on the external image viewer. Therefore, for the mixture problem of denoising, demosaicing and super-resolution, the SR processing may be performed separately (or demosaicing and SR).

Base on this, we can solve the joint problem of denoising, demosaicing and SR with more joint solutions. For a given noisy low resolution raw mosaic image LR_M^N, its corresponding HR color image I_{HR} can be computed by several ways.

Applying existing methods. First processing three tasks separately, the selections of SOTA methods are DnCNN [8] for denoising, DJDD [2] for demosaicing, and VDSR [9] for SR. The DJDD model used here is the noise-free version, and the VDSR only works on luminance channel (y channel in YCbCR color space). The execution order has been discussed above: denosing first, demosaicing next, and SR at last. Then, the methods in equation (4.1) are replaced by specific methods:

$$\tilde{I} = I_{HR} = VDSR(BI(DJDD(DnCNN(LR_M^N)))). \tag{4.16}$$

where $\tilde{I}$ is the desired output, and BI means BICUBIC interpolation (see also equation (4.9)).

Applying existing methods and combined CNN. We can also process denoising and demosaicing together, and then sequentially process super-resolution. The selection of joint network of denoising and demosaicing is DJDD [2], and the selection of SR methods is VDSR [9]. The composite function can be written as:

$$\tilde{I} = I_{HR} = VDSR(BI(DJDD(LR_M^N))). \tag{4.17}$$

Similarly, denoising can be performed separately, and then jointly demosaicing and SR. We have chosen SOTA method DnCNN [8] for denoising. Since there is no public trained model for joint demosaicing and SR, we adopt our own JD_MSR model, which is described in Section 4.1.2. An equation (4.14) is converted to:

$$\tilde{I} = I_{HR} = JD_MSR(DnCNN(LR_M^N)). \tag{4.18}$$

Training specific CNNs. In addition to applying existing methods, we can also specifically train these networks in a sequence to solve the compound problem of joint denoising, demosaicing and SR. Same with 'existing methods', we apply DnCNN [8] for denosing, DJDD [2] noise-free version for demosaicing, and VDSR [9] for super-resolution. These specific networks are referred to as D_N, D_M and SR. Then, the HR color image is computed as:

$$\tilde{I} = I_{HR} = \mathcal{F}_{SR}(\mathcal{F}_{D_M}(\mathcal{F}_{D_N}(LR_M^N))). \tag{4.19}$$

The loss functions of three networks are:

$$\begin{aligned} \mathcal{L}_{D_N} &= \mathcal{L}(\widetilde{I_M^{LR}} - I_M^{LR}) = \mathcal{L}(\mathcal{F}_{D_N}(LR_M^N) - I_M^{LR}), \\ \mathcal{L}_{D_M} &= \mathcal{L}(\mathcal{F}_{D_M}(\widetilde{I_M^{LR}}) - I_{LR}) = \mathcal{L}(\mathcal{F}_{D_M}(\mathcal{F}_{D_N}(LR_M^N)) - I_{LR}), \\ \mathcal{L}_{SR} &= \mathcal{L}(\mathcal{F}_{SR}(\widetilde{I_{LR}}) - I) = \mathcal{L}(\mathcal{F}_{SR}(\mathcal{F}_{D_M}(\mathcal{F}_{D_N}(LR_M^N))) - I). \end{aligned} \tag{4.20}$$

where I_M^{LR}, I_{LR} regards to mosaic LR image and LR image of ground-truth I.

Training a specific CNN and combined CNN. In addition, the partial combining strategy can be used in this joint solution. We have replaced denoising and demosaicing models with the specifically trained DJDD [2], which is a SOTA method of joint denoising and demosaicing. The composite function is:

$$\tilde{I} = I_{HR} = \mathcal{F}_{SR}(\mathcal{F}_{D_ND_M}(LR_M^N)). \tag{4.21}$$

where $\mathcal{F}_{D_ND_M}$ is the feature mapping of trained DJDD model. Its loss functions are:

$$\begin{aligned} \mathcal{L}_{D_ND_M} &= \mathcal{L}(\widetilde{I_{LR}} - I_{LR}) = \mathcal{L}(\mathcal{F}_{D_ND_M}(LR_M^N) - I_{LR}), \\ \mathcal{L}_{SR} &= \mathcal{L}(\mathcal{F}_{SR}(\widetilde{I_{LR}}) - I) = \mathcal{L}(\mathcal{F}_{SR}(\mathcal{F}_{D_ND_M}(LR_M^N)) - I). \end{aligned} \tag{4.22}$$

On the other hand, we sequentially train joint demosaicing and super-resolution model after a specific DnCNN model. The description of joint demosaicing and SR network is

given in Section 4.1.2. Equation (4.15) is then transferred to:

$$\tilde{I} = I_{HR} = \mathcal{F}_{JD_MSR}(\mathcal{F}_{D_N}(LR_M^N)). \tag{4.23}$$

where $\mathcal{F}_{JD_MSR}$ is the feature mapping of a trained JD_MSR model. The loss functions are computed by:

$$\begin{aligned} \mathcal{L}_{D_N} &= \mathcal{L}(\widetilde{I_M^{LR}} - I_M^{LR}) = \mathcal{L}(\mathcal{F}_{D_N}(LR_M^N) - I_M^{LR}), \\ \mathcal{L}_{JD_MSR} &= \mathcal{L}(\mathcal{F}_{JD_MSR}(\widetilde{I_M^{LR}}) - I) = \mathcal{L}(\mathcal{F}_{JD_MSR}(\mathcal{F}_{D_N}(LR_M^N)) - I). \end{aligned} \tag{4.24}$$

Combined CNN. We propose a deep CNN for joint denosing, demosaicing and super-resolution, regarded as JD_ND_MSR. The structure of JD_ND_MSR network combines the architecture of JD_MSR and the denoising strategy of DJDD [2]. Figure 4.3 shows the denoising strategy of DJDD. Because the noise level is known in advance, we can parametrize a network with this. The simplest way is to add a noise level input σ, and replicate it spatially and concatenating it with the packed mosaic vector. Through this noise vector, we will argument every pixel with the same noise value. Therefore, every layer downstream depends on it, which effectively parametrizes the learned filters.

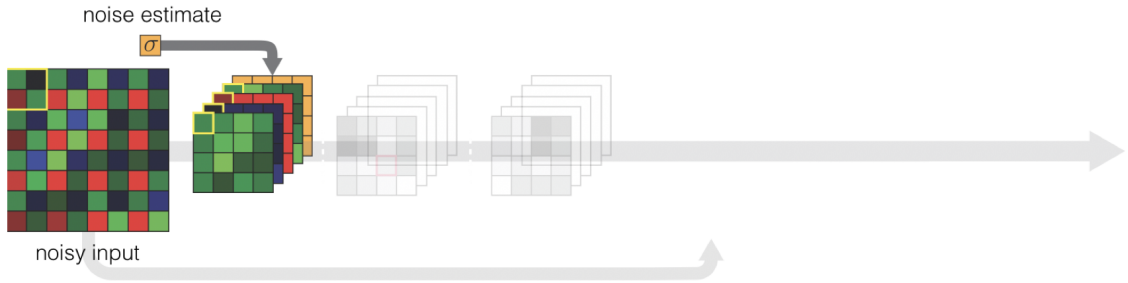


Figure 4.3. *The denosing strategy of DJDD [2] network.*

Inspired by this, the denoising processing of JD_ND_MSR is enable to implement easily by just adding a noise estimation input. The network architecture of JD_ND_MSR is illustrated in Figure 4.4. In addition to the noisy input, all layers are identical to the ones of JD_MSR. The mosaic input is first down-sampled into a 4D vector, and is concatenated with the noise input vector to form 5D vectors. After color extraction, a series of residual-in-residual-dense-blocks filter the image to interpolate the missing color values. We perform last group of convolutions to reconstruct the full resolution image. The summary of the network architecture is provided in Table 4.2.

According to equation (4.3), the high-resolution color output I_{HR} is generated by the following function:

$$\tilde{I} = I_{HR} = \mathcal{F}_{JD_ND_MSR}(LR_M^N)), \tag{4.25}$$

and its loss function by:

$$\mathcal{L}_{JD_ND_MSR} = \mathcal{L}(\mathcal{F}_{JD_ND_MSR}(LR_M^N) - I). \tag{4.26}$$

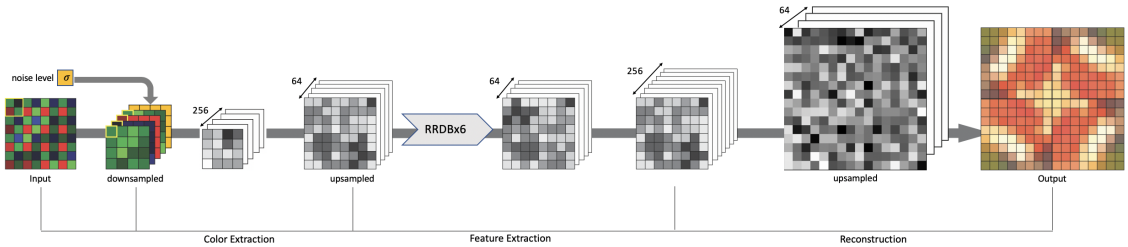


Figure 4.4. *Illustration of our deep joint denoising, demosaicing and super-resolution network* JD_ND_MSR.

Table 4.2. *The summary of our* JD_NDMSR *network architecture. The number of RRDBs is 6 and we set the number of filters* $C = 256$ *and* $W = 64$.

Stage	Layer	Output Shape
Input	Input (Bayer image)	$h \times w \times 1$
	Input (noise estimate)	$\frac{h}{2} \times \frac{w}{2} \times 1$
Color Extraction	Down-sampling Bayer input	$\frac{h}{2} \times \frac{w}{2} \times 4$
	Concatenate with noise input	$\frac{h}{2} \times \frac{w}{2} \times (4+1)$
	Conv	$\frac{h}{2} \times \frac{w}{2} \times C$
	Up-sampling	$h \times w \times \frac{C}{4}$
Feature Extraction	RRDB	$h \times w \times W$
	...	...
	RRDB	$h \times w \times W$
	Conv	$h \times w \times C$
Reconstruction	Up-sampling	$(sf \times h) \times (sf \times w) \times W$
	Conv	$(sf \times h) \times (sf \times w) \times 3$
Output	Output (color image)	$(sf \times h) \times (sf \times w) \times 3$

In the next Chapter 5, all above joint solutions will be arranged in the comparison experiments.

5 EXPERIMENTAL FRAMEWORK

5.1 Experimental setup

For the training, we have applied Nvidia Tesla P100 GPU with 16 GB memory from the TUT TCSC Narvi computing cluster. All testing experiments run on a Linux desktop computer, with 3.4 GHz Intel i7-3770 CPU, 32 GB of RAM, and Nvidia GTX 1050Ti GPU with 4GB of memory.

Training data. For training and validation of the network, we used publicly available dataset DIV2K [58] which contains 900 2K resolution images (800 for training, 100 for validation) for image restoration tasks.

Data preprocessing. To create simulated input images I_{Input}, we corrupt the GT images I by preprocessing. For different mixture problems, the input data is processed by different kinds of distortions in a certain sequence. Figure 5.1 shows how the input image I_{Input} is generated sequentially from the ground-truth image I_{GT}. For data preprocessing of denoising, noisy input image is generated by adding Gaussian noise (the orange blocks in Figure 5.1) with the noise levels (σ) 10, 20 and 30. For data preprocessing of demosaicing, we mosaic the color image to a single-channel image in the Bayer CFA pattern. In this thesis, all experiments adopt 'rggb' Bayer pattern order (the green blocks in Figure 5.1). For data preprocessing of super-resolution, the HR image is BICUBIC down scaled (the yellow blocks in Figure 5.1) with the scale factors (SF) 2, 3 and 4. The down-sampling is processed in MATLAB with imresize function.

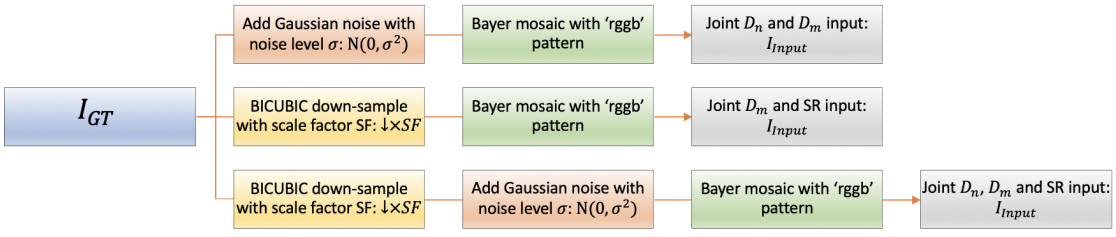


Figure 5.1. *Data preprocessing of different mixture problems. D_n, D_m and SR denote denoising, demosaicing and super-resolution, respectively.*

For each training epoch, the mini-batch size is 64, and the patch size is 64×64. However, for the training of models of super-resolution, we use smaller patch size to speed up the training when the scale factors are 3 and 4. In addition, we randomly augment the patches by flipping horizontally or vertically and rotating 90°.

Training details. All models are implemented in Python with Keras. For the optimization of network parameters, we use Adam [59] with $\beta_1 = 0.9, \beta_2 = 0.999$ and the learning rate is initialized to 0.001. All training continue 100 epochs. There are 2000 training steps and 200 validation steps in each epoch. For the first 40 epochs, the learning rate is constant, then linearly declines to 0 in the remaining 60 epochs. Except for DnCNN [8], which uses a sum of squares error as a loss function, the loss function applied during training is mean square error (MSE). Only the model with the smallest validation loss is saved.

5.2 Experiments on different test datasets

Test dataset. We compare different joint solutions on several public benchmark datasets, such as McMaster [60], Kodak, B100 [61] and Urban100 [62]. Among them, McMaster and Kodak are often used for benchmark in demosaicing work, and B100 and Urban100 are often applied in super-resolution methods. Dataset B100 contains 100 human segmented natural images, and dataset Urban100 is a dataset of 100 urban images with many similar structures. These four datasets are widely used in other image restoration works [2, 5, 8, 9, 63, 64].

Comparisons of joint solutions. In this thesis, there are three mixture problems discussed, denoising and demosaicing (DnDm), demosaicing and super-resolution (DmSR), and denoising, demosaicing and super-resolution (DnDmSR). For each synthesis problem, we propose several joint solutions, whose main features are described below.

Table 5.1 shows details of the joint solutions of denoising and demosaicing. The second and third columns are the execution order and the CNN networks used in each joint solution. DnCNN represents the deep denoising network proposed by Zhang et al. [8], and DJDD regards to the deep joint denoising and demosaicing network [2]. Since DJDD is an existing SOTA method that combines denoising and demosaicing, its joint solution type is existing combined CNN. The DJDD model used for demosaicing only is a noise-free version of the model. The last columns show what CNN library is applied and the number of convolutional layers in every joint solution.

Table 5.1. *Summary of the compared joint solutions of* ***Denoising*** *(Dn) and* ***Demosaicing*** *(Dm). JDnDm denotes that denoising and demosaicing are processed together. The networks marked by $*$ are re-implemented and trained.*

Joint Solution	Sequence	Network	Platform	Layers
Existing Method	Dn→Dm	DnCNN→DJDD	MatConvNet→CAFFE	17+15
Specific CNNs	Dn→Dm	DnCNN* →DJDD*	Keras→Keras	17+15
Combined CNN	JDnDm	DJDD*	Keras	15
Existing Combined CNN	JDnDm	DJDD	CAFFE	15

The summary table of joint solutions of demosaicing and super-resolution is showed in Table 5.2. For each solution, the solution type, processing sequence, exploited networks, platform and amount of layers are listed clearly. DJDD and VDSR are selected SOTA

methods for demosaicing and super-resolution, respectively. In addition to apply directly, these two networks are also specific trained, which marked with $*$. For the combined CNN, the network JD_MSR is used, which is proposed in Section 4.1.2.

Table 5.2. *Summary of the compared joint solutions of* ***Demosaicing*** *(Dm) and* ***Super-Resolution*** *(SR). JDmSR denotes that demosaicing and super-resolution are processed together. The networks marked by $*$ are re-implemented and trained.*

Joint Solution	Sequence	Network	Platform	Layers
Existing Method	Dm→SR	DJDD→VDSR	CAFFE→MatConvNet	15+20
Specific CNNs	Dm→SR	DJDD* →VDSR*	Keras→Keras	15+20
Combined CNN	JDmSR	JD_MSR	Keras	6 RRDBs

For the mixture problem of denoising, demosaicing and super-resolution, Table 5.3 presents the summary of comparison of different joint solutions. There are five types of joint solutions. For 'Existing Method' and 'Specific CNNs', tasks can be combined partial, for example, combine denoising and demosaicing first, or combine demosaicing and SR. So, DJDD* (Table 5.1) and JD_MSR (Table 5.2), the 'Combined CNN' joint solutions for two tasks, are exploited in this comparison. The second column is the sequence of processing. As it was mentioned above, the execution order of the mixture problem is denoising followed by demosaicing, and SR at the end. In the third column are the models we adopted, pre-trained models are not marked. The JD_MSR model in third row denotes the trained model in previous comparison, joint demosaicing and SR. The last two columns are the platform of implementation and the number of convolutional layers, respectively.

Table 5.3. *Summary of the compared joint solutions of* ***Denoising*** *(Dn),* ***Demosaicing*** *(Dm) and* ***Super-Resolution*** *(SR). JDnDm denotes that denoising and demosaicing are processed together. JDmSR combines demosaicing and super-resolution. JDnDmSR is joint denoising, demosaicing and super-resolution. The networks marked by $*$ are re-implemented and trained.*

Joint Solution	Sequence	Network	Platform	Layers
Existing Method	Dn→Dm →SR	DnCNN→DJDD →VDSR	MatConvNet→CAFFE →MatConvNet	17+15+20
Existing Method + Combined CNN	Dn→JDmSR	DnCNN→ JD_MSR	MatConvNet→Keras	17+6RRDBs
	JDnDm→SR	DJDD→VDSR	CAFFE→MatConvNet	15+20
Specific CNNs	Dn* → Dm* →SR*	DnCNN* → DJDD* →VDSR*	Keras→ Keras→Keras	17+15+20
Specific CNN + Combined CNN	Dn* →JDmSR*	DnCNN* → JD_MSR^*	Keras→Keras	17+6RRDBs
	JDnDm* →SR*	DJDD* →VDSR*	Keras→Keras	15+20
Combined CNN	JDnDmSR	JD_ND_MSR	Keras	6 RRDBs

This chapter shows the framework of experiments and the details of comparisons between joint solutions. Both numerical results and visualize results with the analysis will

be given in next Chapter 6.

6 ANALYSIS OF EXPERIMENTAL RESULTS

This chapter includes both the quantitative analysis and the qualitative analysis of experimental results, in Section 6.1 and Section 6.2, respectively.

6.1 Quantitative analysis

Evaluation metrics. Quantitative analysis was performed with cPSNR and SSIM metrics, by calculating them on full RGB image. The results are averaged over whole dataset. For super-resolved image, the borders of the image are shaved off, with the scaling factor as the width of the shaved border.

Quantitative analysis of joint denoising and demosaicing. Table 6.1 shows the cPSNR and SSIM results for noise levels 10, 20 and 30, respectively. Higher values indicate better performance, and in each column the best values for each noise level are highlighted with red color, the second best with blue and the third best with green. In this table, the joint solution 'Specific CNNs' obtains best performance for all noise levels. Because our combined network DJDD* is re-implemented from original DJDD [2], and the model is specifically trained with the fix noise level. However, the original DJDD is trained with a continuous range of noise levels, and the noise level is randomly sampled from 0 to 20. When the noise level is set to 10, the original DJDD trained model attains higher values than our specific trained DJDD* model. As the noise increases to 20, the cPSNR of DJDD* model is improved by at least 0.37dB than the original DJDD model. In contrast, applying SOTA methods get worst performance. It should be caused by the interactions between two processing. The denoising processing eliminates not only noise, but details in the image, which affects demosaicing processing. Although training specific models for each IR task can generate best result, the computational complexity and structural redundancy are more serious than the combined version, as well. This is a trade-off between high performance and complexity.

It should be noted that we connect the trained DnCNN* model with DJDD* network in the 'Specific CNNs' joint solution. Due to the limitation of the memory space, image patches are first creates from trained DnCNN*, then directly input into DJDD* without storage. But the parameters of DnCNN* part are not trainable, i.e. transferring the well trained denoising mapping. In addition to this transfer learning version, we also train a substantive DJDD* for demosaicing. The noise level is 10 and the training set and validation set are first added Gaussian noise, then denoised by trained denosing model DnCNN*. The

Table 6.1. *Quantitative comparison of different approaches on the mixture problem of joint denoising and demosaicing using dataset Kodak, McMaster [60]. The noise level is set to 10, 20 and 30. DJDD [2] do not provide the model for noise level more than 20.*

Method	Joint solution	Noise level	McMaster		Kodak	
			cPSNR	SSIM	cPSNR	SSIM
DnCNN→DJDD	Existing Method	10	28.65	0.8999	29.44	0.8691
DnCNN* →DJDD*	Specific CNNs		33.39	0.9658	33.60	0.9585
DJDD*	Combined CNN		32.74	0.9622	33.07	0.9535
DJDD	Existing Combined CNN		33.14	0.9629	33.22	0.9537
DnCNN→DJDD	Existing Method	20	24.48	0.7693	25.89	0.7218
DnCNN* →DJDD*	Specific CNNs		30.70	0.9399	30.65	0.9231
DJDD*	Combined CNN		30.52	0.9391	30.45	0.9220
DJDD	Existing Combined CNN		30.15	0.9313	30.00	0.9101
DnCNN→DJDD	Existing Method	30	21.58	0.6313	23.71	0.5897
DnCNN* →DJDD*	Specific CNNs		28.96	0.9184	28.91	0.8945
DJDD*	Combined CNN		28.82	0.9163	28.81	0.8927

input patches of DJDD* are generated from these images. The cPSNR values of this model (without transfer) are even higher, 33.49dB on McMaster and 33.71dB on Kodak, improved by 0.1dB than the transfer version.

Quantitative analysis of joint demosaicing and super-resolution. The cPSNR and SSIM values for scale factor 2, 3 and 4 are described in Table 6.2. Same as before, for each scale factor, in one column the best values are highlight with red, the second best and the third best ones are highlight with blue and green, respectively. From this table, we can find that the 'Combined CNNs' attains highest values for all scale factors. For cPSNR, the results of 'Combined CNNs' joint solution on McMaster dataset over the second place by at least 0.66 dB, and the values on Kodak dataset at least 0.27 dB beyond other joint solutions. For other joint solutions, directly applying the trained models perform better than specific trained models. We thought training a three-channel VDSR is more complex and harder than training a single-channel one. Although we would like to train both luminance channel and chroma channels together to improve the performance, the computational complexity increases in geometric progression. Therefore, the modification of VDSR needs deeper learning to surpass the original version.

Different to the experiments of joint denoising and demosaicing, we use transfer learning in 'Combined CNN' instead of 'Specific CNNs', and only for scale factor 4. For the 'Specific CNNs', the training image set of the later network is generated by the previous trained model. But for scale factor 4, in addition to a direct expansion, we train an upsampling twice model, i.e. add another 2×2 upscale after the model for scale factor 2. So the learned parameters for scale factor 2 can be transferred to the deeper model JD_MSR for scale factor 4. This kind of 'easy-hard' transfer learning has been pointed out in another low-level vision problem, compression artifacts reduction [65]. In this experiment,

Table 6.2. *Quantitative comparison of different approaches on the mixture problem of joint demosaicing and super-resolution using dataset Kodak, McMaster [60]. The scale factor is set to 2, 3 and 4.*

Method	Joint solution	Scale factor	McMaster cPSNR	McMaster SSIM	Kodak cPSNR	Kodak SSIM
DJDD→VDSR	Existing Method	2	31.67	0.9590	31.08	0.9404
DJDD* →VDSR*	Specific CNNs		31.37	0.9562	30.91	0.9395
JD_MSR	Combined CNN		32.46	0.9643	31.44	0.9445
DJDD→VDSR	Existing Method	3	28.52	0.9234	28.02	0.8870
DJDD* →VDSR*	Specific CNNs		28.53	0.9219	28.02	0.8870
JD_MSR	Combined CNN		29.19	0.9317	28.29	0.8929
DJDD→VDSR	Existing Method	4	26.64	0.8908	26.53	0.8477
DJDD* →VDSR*	Specific CNNs		26.59	0.8877	26.40	0.8435
JD_MSR	Combined CNN		27.24	0.8996	26.83	0.8556
JD_MSR^T	Combined CNN		27.32	0.9018	26.88	0.8566

the model used transfer learning regard as JD_MSR^T. Since the training bases on a well-trained one, the training only continue 40 epochs with a lower initial learning rate 0.0001, which linearly decreases to 0. In Table 5.2, the model with transfer learning (JD_MSR^T) gets better performance than the one without transfer learning (JD_MSR), because the features learned from relatively easier task support a good starting point, which is beneficial to converge. Therefore, this kind of 'easy-hard' transfer is very helpful, and we also utilize this strategy in the further experiments.

Quantitative analysis of joint denoising, demosaicing and super-resolution. For the comparison of joint solutions of denoising, demosaicing and SR, we fixed the noise level to 10 and the scale factor to 2. Table 6.3 compares 8 joint solutions. For our joint solutions, the highest three numbers in each column are highlighted with red, blue and red, respectively.

First, in Table 5.3, it is obvious that the performances of the 'Existing Methods' type of joint solutions are significantly worse than the performance of their specifically trained ones. For the fully sequential joint solutions, the specific trained models improves the cPSNR values at least by 2.3 dB. For partial combined joint solutions, both cPSNR and SSIM values are improved by the specific training.

Second, the combined version networks attain best results. Compared to JD_MSR, JD_ND_MSR only concatenates a noise input vector with the mosaic vectors and obtains the third best result. $JD_ND_MSR^T$ transfers the learned parameters of pre-trained JD_MSR model with scale factor 2 in Table 6.2. This 'easy-hard' transfer strategy raise the values a little.

Third, the sequential joint of DnCNN* and JD_MSR becomes the second best joint so-

Table 6.3. *Quantitative comparison of different approaches on the mixture problem of joint denoising, demosaicing and super-resolution using dataset Kodak, McMaster [60]. The noise level is 10 and the scale factor is set to 2.*

Joint solution	Method	McMaster		Kodak	
		cPSNR	SSIM	cPSNR	SSIM
Existing Method	DnCNN →DJDD→VDSR	25.99	0.8522	26.18	0.7868
Existing Method + Combined CNN	DnCNN→ JD_MSR	26.03	0.8546	26.19	0.7839
	DJDD→VDSR	28.40	0.9248	28.13	0.8812
Specific CNNs	DnCNN* → DJDD* →VDSR*	29.14	0.9248	28.53	0.8913
Specific CNN + Combined CNN	DnCNN* → JD_MSR^*	29.46	0.9288	28.73	0.8953
	DJDD* →VDSR*	28.88	0.9212	28.43	0.8887
Combined CNN	JD_ND_MSR	29.38	0.9268	28.72	0.8951
	$JD_ND_MSR^T$	29.48	0.9290	28.75	0.8959

lution in this comparison. And its values are very close to the best ones and higher than the combined CNN JD_ND_MSR. As mentioned in Section 4.1.2 and Section 4.2, the structures of JD_MSR and JD_ND_MSR are almost identical. Therefore, even though the quantitative performances are similar, JD_ND_MSR attains a comparable result with less complexity. In addition, the little better performance of $JD_ND_MSR^T$ proves that the DnCNN processing can be replaced by applying the denoising strategy (Figure 4.3).

In Table 5.3, there is an interesting phenomenon is that the performance of directly applying trained models of DnCNN and JD_ND_MSR (the second method) is lower than the specific trained ones (the fifth method) at least by 2.54 dB on cPSNR. In this experiment, we performed DnCNN trained model on raw image. Since DnCNN model is trained for gray-scale image, we reproduce the denoised raw image by processing denoising for each color channel (R, G, B). Then we apply JD_ND_MSR on the denoised images, and get the test results in Table 6.4. We can find that the cPSNR and SSIM values are worse than the ones in Table 5.3. Thus, we think the JD_ND_MSR network is very sensitive to the input, that is the reason why the specific trained models attain better performance significantly.

Table 6.4. *Test a new DnCNN→ JD_MSR on dataset Kodak, McMaster [60]. The noise level is 10 and the scale factor is set to 2.*

McM		Kodak	
cPSNR	SSIM	cPSNR	SSIM
25.51	0.8142	24.79	0.7216

More experiments on $JD_ND_MSR^T$. Since the combined network $JD_ND_MSR^T$ attains the best performance in above comparison of different joint solutions of denoising, demo-

saicing, and SR, we have tested it on more datasets with more noise levels. Table 6.5 shows the average cPSNR and SSIM results of $JD_ND_MSR^T$ on four different datasets, McMaster [60], Kodak, B100 [61] and Urban100 [62]. The scale factor is 2 and the noise levels are 10, 20 and 30. It is obvious that the noise affects the performance directly. As the noise level increasing, the qualities of $JD_ND_MSR^T$ on all datasets are dropped. In addition, the values of two challenging datasets B100 and Urban100 are conspicuously lower than the ones of McMaster and Kodak.

Table 6.5. *The average cPSNR and SSIM results of $JD_ND_MSR^T$ on different datasets. The scale factor is 2 and the noise levels are 10, 20 and 30.*

Noise	McMaster		Kodak		B100		Urban100	
level	cPSNR	SSIM	cPSNR	SSIM	cPSNR	SSIM	cPSNR	SSIM
10	29.48	0.9290	28.75	0.8959	27.31	0.8708	26.73	0.8885
20	27.49	0.8978	27.12	0.8578	25.67	0.8243	25.18	0.8504
30	26.17	0.8728	26.06	0.8303	24.61	0.7907	24.06	0.8177

Since $JD_ND_MSR^T$ surpasses other joint solutions in the quantitative comparison, we re-train it with some other cost functions besides MSE. Inspired by [17], we train the network with six different cost functions: MSE, MAE, SSIM, MS-SSIM, Mix1 and Mix2, which are introduced in 2.2. For the parameter α of Mix1 and Mix2, we test a few different values to balance the contribution of the two losses. We empirically set $\alpha = 0.84$, which is also recommended by the authors of [17]. Then, the resulting images of trained models are compared by four image quality metrics: cPSNR, SSIM, MS-SSIM and CSSIM [18]. Because of the limitation of GPU, the patch size is 32×32 during training. We still fix the scale factor to 2 and the noise level to 10. The comparisons results on McMaster and Kodak datasets are shown in Table 6.6 and Table 6.7, respectively. It is obvious that the model trained with MAE cost function obtains best performance for all image quality metrics and on all datasets. Compared with the model trained with MSE, the cPSNR values of MAE version is improved by at least 0.24dB.

Table 6.6. *Average value of different image quality metrics on the McMaster testing dataset for $JD_ND_MSR^T$ trained with different cost functions. The noise level is 10 and the sale factor is 2. For cPSNR, SSIM, MS-SSIM, and cSSIM the value reported here has been obtained as an average of the three color channels. Best results are shown in bold.*

McMaster	Training cost function					
Image quality metric	MSE	MAE	SSIM	MS-SSIM	Mix1	Mix2
cPSNR	29.22	**29.48**	28.70	28.54	28.81	28.80
SSIM	0.9245	**0.9288**	0.9213	0.9174	0.9223	0.9228
MS-SSIM	0.9545	**0.9575**	0.9535	0.9512	0.9533	0.9536
CSSIM	0.9782	**0.9799**	0.9770	0.9760	0.9769	0.9771

Summary. First, whether for mixture problem of two or three tasks, the combined CNN

Table 6.7. *Average value of different image quality metrics on the Kodak testing dataset for $JD_ND_MSR^T$ trained with different cost functions. The noise level is 10 and the sale factor is 2. For cPSNR, SSIM, MS-SSIM, and cSSIM the value reported here has been obtained as an average of the three color channels. Best results are shown in bold.*

Kodak	Training cost function					
Image quality metric	MSE	MAE	SSIM	MS-SSIM	Mix1	Mix2
cPSNR	28.50	**28.74**	28.09	28.00	28.13	28.11
SSIM	0.8890	**0.8953**	0.8870	0.8800	0.8861	0.8838
MS-SSIM	0.9401	**0.9442**	0.9406	0.9394	0.9406	0.9406
CSSIM	0.9711	**0.9732**	0.9696	0.9691	0.9699	0.9698

joint solution is the best selection, which attains best performance with minimal complexity and simplest network. Second, specific training the models with specific data can get better results than using trained models directly. Third, the 'easy-hard' transfer strategy [65] helps the high-level network to start from a good point by the features from well-trained low-level network. Another helpful strategy is the denoising strategy [2], which only inputs a noise estimation vector. In summary, the $JD_ND_MSR^T$ model trained with MAE loss gets best performance of the mixture problem of denoising, demosaicing and SR.

6.2 Qualitative analysis

Qualitative analysis of joint denoising and demosaicing. Figure 6.1 compares three joint solutions of denoising and demosaicing, DnCNN$\rightarrow$DJDD, DnCNN* $\rightarrow$DJDD* and DJDD*. We selected one image from each of the two test datasets, McMaster and Kodak. It is obvious that the resulting images (the third column) of exploiting existing methods contain residual noise (the top image) and color deviation (the bottom image). For the resulting images of other joint solutions, the noise is eliminated enough. However, for the resulting images of DJDD*, there are some false color artifacts in McMaster image13. And compared to the image01 of DnCNN* $\rightarrow$DJDD*, more textures on wooden doors and windows are removed (we recommend zoom in the images to observe).

Qualitative analysis of joint demosaicing and super-resolution. The resulting images of joint demosaicing and SR are showed in Figure 6.2. The scale factor is 2 and the Bayer pattern is 'rggb'. From left to right, there are ground-truth images and the resulting images of three joint solutions: 'Existing Method', 'Specific CNNs' and 'Combined CNN'. The test images still are selected from datasets McMaster and Kodak. In this comparison, it is obvious that the images produced by the combined CNN JD_MSR keep more details (image1 of McMaster) and eliminate more color artifacts caused by demosaicing (image05 of Kodak). However, although DJDD$\rightarrow$VDSR performs better than the specific trained version DJDD* $\rightarrow$VDSR* on quantitative comparison, none of the resulting images of these two joint solutions are more outstanding. The serious problems caused by

Ground Truth *Noisy mosaic* *DnCNN→DJDD* *DnCNN* →DJDD** *DJDD**

Figure 6.1. *Comparison of the joint solutions of denoising and demosaicing. The first row is image13 of McMaster dataset, and the second row is image01 from Kodak dataset. The noise level of Gaussian noise is 10 and Bayer pattern is 'rggb'.*

task interaction, such as blurring and error color artifacts, are not solved by the specific training. In contrast, the combination of two networks not only recover more details for super-resolution, but correct error color artifacts for demosaicing.

Ground Truth *DJDD→VDSR* *DJDD* →VDSR** JD_MSR

Figure 6.2. *Comparison of the joint solutions of demosaicing and super-resolution. The first row is image1 of McMaster dataset, and the second row is image05 from Kodak dataset. The scale factor is 2 and Bayer pattern is 'rggb'.*

Qualitative analysis of joint denoising, demosaicing and super-resolution. For comparison, we selected image08 from McMaster dataset, which is a sewing machine with a yellow trademark 'SINGER'. Figure 6.3 compares all joint solutions of denoising, demosaicing and SR tested on image08. For 'Existing Method' type joint solutions (three images on first row), the edges of the letters are not clear, such as, the bottom of letter 'S' and letter 'G' seems like 'C'. The 'Specific CNNs' joint solutions (second row) recover the letters edges but there are errors in the letters. There is discontinuity in letter 'S' of DnCNN* →DJDD* →VDSR and DnCNN* → JD_MSR^*, and the same problem happens in letter 'N' of DJDD* →VDSR*. And these three joint solutions generates color artifacts (in letter 'G'). Compared with the ground-truth image, our two combined networks

($JD_N D_M SR$ and $JD_N D_M SR^T$) successfully restored the trademark with clear and clean edges and without artifacts.

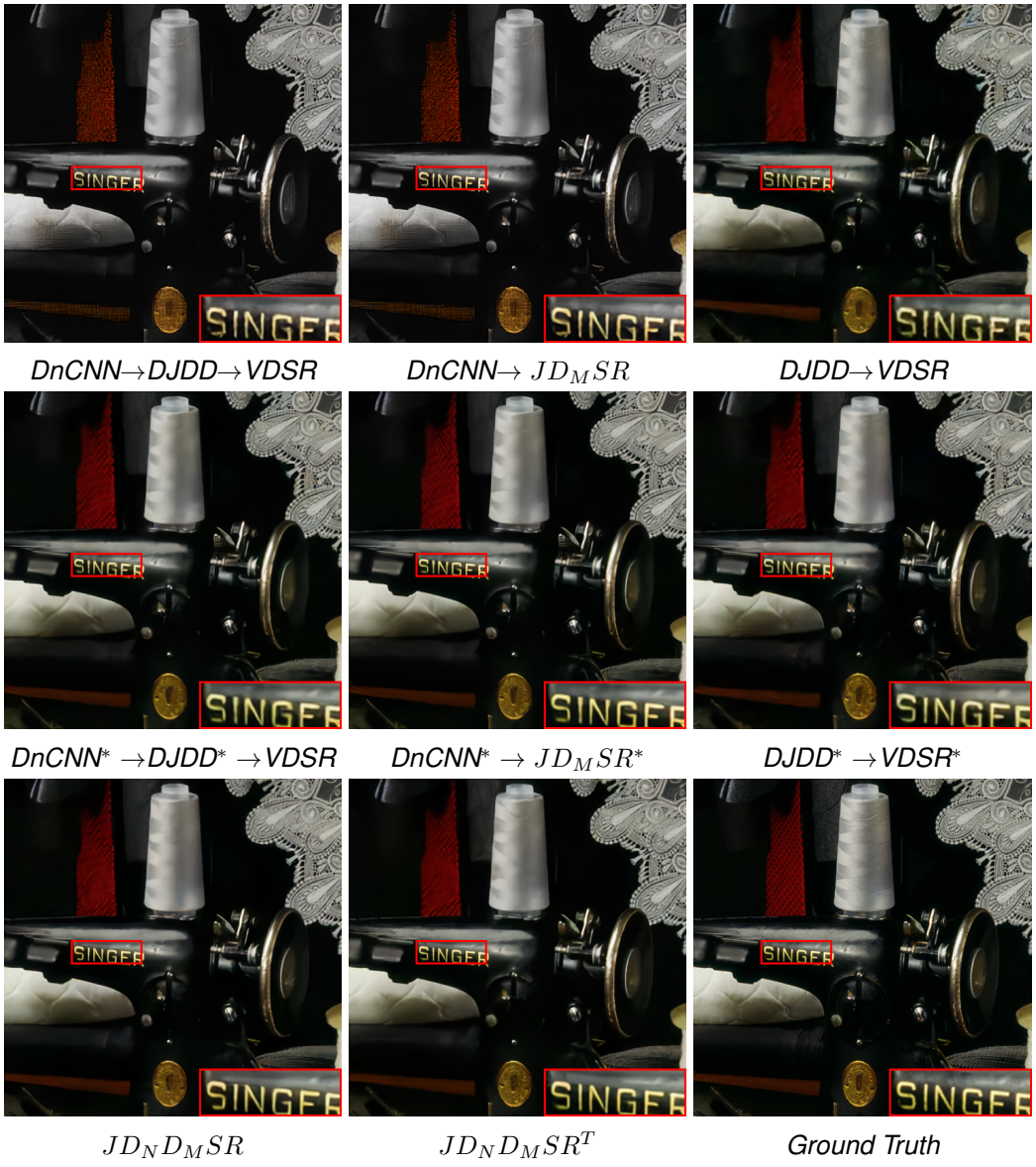


Figure 6.3. *Comparison of the joint solutions of denoising, demosaicing and super-resolution. Image08 from McMaster dataset. Noise level is 10, and scale factor is 2. Bayer pattern is 'rggb'.*

Another example image is image05 from Kodak dataset, used as an example in the comparison of joint demosaicing and SR. The high-frequency details of image05 is a good tester for demosaicing. Because demosaicing algorithms are always unavoidable to generate some noticeable color artifacts in the high-frequency texture regions and strong edges. Figure 6.4 shows the test results of 8 joint solutions on image05. The hard region is marked with rectangle and zoomed at the right bottom corner of the image. For first three rows, left is the 'Existing Method' type joint solutions and right three images are results of their specific version joint solutions. We can find that these six images include serious and noticeable color artifacts. There are checkerboard artifacts in the resulting

images of DnCNN→DJDD→VDSR and DnCNN→ JD_MSR two joint solutions. On the other hand, the specific trained CNNs cause error color artifacts. There are less artifacts in the test image of DJDD→VDSR because the original DJDD [2] model is trained with abundant challenging patches to improve the performance of demosaicing. However, too much details and textures are removed with noise and error color artifacts, i.e. it introduces a blur. In contrast, there are less color artifacts in the resulting images of our two combined networks JD_ND_MSR and $JD_ND_MSR^T$. $JD_ND_MSR^T$ even tried to restore some texture, although the texture is not correct.

DnCNN→DJDD→VDSR *DnCNN* →DJDD* →VDSR*

DnCNN→ JD_MSR *DnCNN** $\rightarrow JD_MSR^*$

DJDD→VDSR *DJDD* →VDSR**

JD_ND_MSR $JD_ND_MSR^T$

Ground Truth

Figure 6.4. *Comparison of the joint solutions of denoising, demosaicing and super-resolution. Image05 from Kodak dataset. Noise level is 10, and scale factor is 2. Bayer pattern is 'rggb'.*

7 CONCLUSION

In this thesis, a comparative analysis on joint solutions of the mixture problems of multiple image restoration (IR) tasks is performed. These joint solutions are mainly focus on deep convolutional neural network (CNN) methods, whose theoretical basics are discussed in Chapter 2. The IR tasks we concerned in this thesis are denoising, demosaicing and super-resolution (SR). The general theory about these ill-posed problems are also introduced in Chapter 2, and their recently related works are presented in Chapter 3.

The joint solutions of the mixture problems can be categorised to five groups, which are discussed in Chapter 4. The first kind of joint solution is applying suitable existing SOTA methods to solve multiple tasks in sequence. However, there is a challenge is that deep learning often requires to train a new network or to fine-tune an existing one for even slightly different instances of a problem. Thus, these well-trained networks need to be specifically trained for the specific input and output. This is the second joint solution. The third one is exploiting the existed or creating a combined network, which are able to solve complicated multi-task image processing problems in an end-to-end manner. The last two kinds of joint solutions include both sequential processing and combined processing. It means that the multiple tasks can be solved by pair of tasks with a single task, or a single task with the combination of two tasks. The existing methods version and the specific trained version are the fourth and fifth joint solution, respectively.

We suggested the execution order of the mixture problem, to perform a denoising first, demosaicing next, and SR last. This is done because noise will impact demosaicing and super-resolution and lead to noticeable artifacts, and the processing of super-resolution will magnify the errors in the image and increase task's difficulty. Our investigation starts from finding the joint solutions of two IR tasks. In order to support sufficient values to compare different joint solutions, we investigated the mixture problem of denoising and demosaicing, and joint of demosaicing and SR. After the joint two tasks, we compare the joint solutions of denoising, demosaicing and SR, based on the results of joint two tasks.

In this thesis, we proposed a combined network for joint demosaicing and SR, JD_MSR. For a given low resolution mosaic raw image, JD_MSR can generated the high resolution color RGB image directly. The structure of JD_MSR is inspired by an existing network [1]. We have replaced 24 residual blocks (RB) [50] in original network by 6 residual-in-residual dense blocks (RRDB) [46]. The network architecture consists of three parts: color extraction, feature extraction and reconstruction. The network framework and architecture in details is separately described in Figure 4.2 and 4.1. Built on JD_MSR, we developed the

JD_NDMSR network, which is a combined CNN network for joint of denoising, demosaicing and SR. We have only added a noise estimation input for the denoising part, which is shown in Figure 4.4.

For training and validation of the network, we used the publicly available dataset DIV2K [58] which contains 900 2K resolution images (800 for training, 100 for validation) for image restoration tasks. We generated the simulated corrupted data by adding white Gaussian noise, Bayer mosaic and BICUBIC down-sampling. The details of data preprocessing are given in Chapter 5. The training details are included in this chapter, as well. We tested all joint solutions on McMaster [60] and Kodak public benchmark datasets. We compared the joint solutions of three mixture problems based on three experiments, joint denoising and demosaicing, joint demosaicing and SR, and joint denoising, demosaicing and SR. The summary table of the comparison for each mixture problem is shown in Table 5.1-5.3.

Quantitative analysis was performed with cPSNR and SSIM metrics, by calculating them on full RGB image. Results are averaged over the whole dataset. For super-resolved image, the borders of the image are shaved off, with the scaling factor as the width of the shaved border.

There is an important strategy used in our experiments, 'easy-hard' transfer learning. We transfer the features learned from well trained models, to a deeper model. This kind of 'easy-hard' transfer learning has been pointed out in another low-level vision problem, compression artifacts reduction [65]. And it is beneficial for convergence, because the features learned from relatively easier task support a good starting point. For each experiment, the details of the transfer learning are described in Chapter 6.

The numerical results are recorded in Table 6.1-6.3. There are three main points that can be summarized. First, whether for mixture problem of two or three tasks, the combined CNN joint solution is the best selection, which attains best performance with minimal complexity and simplest network. Second, specific training the models with specific data can get better results than using trained models directly. Third, the 'easy-hard' transfer strategy [65] helps the high-level network to start at a good point by the features from well-trained low-level network. Another helpful strategy is the denoising strategy [2], which only inputs a noise estimation vector.

Table 6.4 proves that our JD_MSR network is very sensitive to the input. To process the noise corrupted images, the model of JD_MSR should be trained specifically, otherwise, the performance will drop significantly.

We also tested the JD_NDMSR network with a wider range of noise and more datasets. In addition to McMaster and Kodak, B100 [61] and Urban100 [62] are adopted. Datasets B100 and Urban100 are often applied in super-resolution methods, where they are challenging datasets. The results are shown in Table 6.5. It is obvious that the noise affects the performance directly.

The comparisons are also been made by qualitative analysis. For each mixture problem,

we selected an image from McMaster dataset and an image from Kodak dataset. Figure 6.1 compares the joint solutions of denoising and demosaicing. The joint solution exploiting existing methods fails on eliminating noise and causes color distortions. The other joint solutions can not only remove the noise, but save more textures. The comparison of joint demosaicing and SR is shown in Figure 6.2. Compared with others, the combined version joint solution not only recovers more details for super-resolution, but also corrects color artifacts for demosaicing. For the qualitative comparison of joint solutions of joint denoising, demosaicing and SR, the transferred combined version network $JD_ND_MSR^T$ outperforms all joint solutions (Figure 6.3 and Figure 6.4). $JD_ND_MSR^T$ can reconstruct the image with clearer texture and less color artifacts. For the high-frequency region, we also find that $JD_ND_MSR^T$ generates some seemingly reasonable but incorrect texture to recover the image.

In the future works, we will apply more abundant image datasets including real-world images. Moreover, we will concern about adding the reduction of compression artifacts task in our mixture problem in future, as well.

REFERENCES

[1] Zhou, R., Achanta, R. and Süsstrunk, S. Deep Residual Network for Joint Demosaicing and Super-Resolution. *arXiv preprint arXiv:1802.06573* (2018).

[2] Gharbi, M., Chaurasia, G., Paris, S. and Durand, F. Deep joint demosaicking and denoising. *ACM Transactions on Graphics (TOG)* 35.6 (2016), 191.

[3] Dong, W., Yuan, M., Li, X. and Shi, G. Joint Demosaicing and Denoising with Perceptual Optimization on a Generative Adversarial Network. *arXiv preprint arXiv:1802.04723* (2018).

[4] Klatzer, T., Hammernik, K., Knobelreiter, P. and Pock, T. Learning joint demosaicing and denoising based on sequential energy minimization. *Computational Photography (ICCP), 2016 IEEE International Conference on*. IEEE. 2016, 1–11.

[5] Qian, G., Gu, J., Ren, J. S., Dong, C., Zhao, F. and Lin, J. Trinity of Pixel Enhancement: a Joint Solution for Demosaicking, Denoising and Super-Resolution. *arXiv preprint arXiv:1905.02538* (2019).

[6] Tan, H., Zeng, X., Lai, S., Liu, Y. and Zhang, M. Joint demosaicing and denoising of noisy bayer images with ADMM. *Image Processing (ICIP), 2017 IEEE International Conference on*. IEEE. 2017, 2951–2955.

[7] Park, S. H., Kim, H. S., Lansel, S., Parmar, M. and Wandell, B. A. A case for denoising before demosaicking color filter array data. *2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*. IEEE. 2009, 860–864.

[8] Zhang, K., Zuo, W., Chen, Y., Meng, D. and Zhang, L. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing* 26.7 (2017), 3142–3155.

[9] Kim, J., Kwon Lee, J. and Mu Lee, K. Accurate image super-resolution using very deep convolutional networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, 1646–1654.

[10] Zhang, K., Zuo, W., Gu, S. and Zhang, L. Learning deep CNN denoiser prior for image restoration. *IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 2. 2017.

[11] Chiariglione-Convenor, L. *MPEG-2: Generic coding of moving pictures and associated audio information. ISO*. Tech. rep. IEC JTC1/SC29/WG11, 1996.

[12] Burgett, S. and Das, M. Multiresolution multiplicative autoregressive coding of images. *Proceedings of the IEEE International Conference on System Engineering*. 1991, 276–279.

[13] Burt, P. and Adelson, E. The Laplacian pyramid as a compact image code. *IEEE Transactions on communications* 31.4 (1983), 532–540.

[14] Dong, C., Loy, C. C., He, K. and Tang, X. Learning a deep convolutional network for image super-resolution. *European conference on computer vision*. Springer. 2014, 184–199.

[15] Kim, J., Kwon Lee, J. and Mu Lee, K. Deeply-recursive convolutional network for image super-resolution. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, 1637–1645.

[16] Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D. and Wang, Z. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, 1874–1883.

[17] Zhao, H., Gallo, O., Frosio, I. and Kautz, J. Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging* 3.1 (2016), 47–57.

[18] Ponomarenko, M., Egiazarian, K., Lukin, V. and Abramova, V. Structural similarity index with predictability of image blocks. *2018 IEEE 17th International Conference on Mathematical Methods in Electromagnetic Theory (MMET)*. IEEE. 2018, 115–118.

[19] Perona, P. and Malik, J. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence* 12.7 (1990), 629–639.

[20] Rudin, L. I., Osher, S. and Fatemi, E. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena* 60.1-4 (1992), 259–268.

[21] Simoncelli, E. P. and Adelson, E. H. Noise removal via Bayesian wavelet coring. *Proceedings of 3rd IEEE International Conference on Image Processing*. Vol. 1. IEEE. 1996, 379–382.

[22] Buades, A., Coll, B. and Morel, J.-M. A non-local algorithm for image denoising. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 2. IEEE. 2005, 60–65.

[23] Dabov, K., Foi, A., Katkovnik, V. and Egiazarian, K. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on image processing* 16.8 (2007), 2080–2095.

[24] Gu, S., Zhang, L., Zuo, W. and Feng, X. Weighted nuclear norm minimization with application to image denoising. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, 2862–2869.

[25] Aharon, M., Elad, M. and Bruckstein, A. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing* 54.11 (2006), 4311–4322.

[26] Jain, V. and Seung, S. Natural image denoising with convolutional networks. *Advances in neural information processing systems*. 2009, 769–776.

[27] Burger, H. C., Schuler, C. J. and Harmeling, S. Image denoising: Can plain neural networks compete with BM3D?: *2012 IEEE conference on computer vision and pattern recognition*. IEEE. 2012, 2392–2399.

[28] Zhang, K., Zuo, W. and Zhang, L. FFDNet: Toward a fast and flexible solution for CNN-based image denoising. *IEEE Transactions on Image Processing* 27.9 (2018), 4608–4622.

[29] Xu, J., Zhang, L. and Zhang, D. A trilateral weighted sparse coding scheme for real-world image denoising. *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, 20–36.

[30] Malvar, H. S., He, L.-w. and Cutler, R. High-quality linear interpolation for demosaicing of Bayer-patterned color images. *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*. Vol. 3. IEEE. 2004, iii–485.

[31] Zhang, L. and Wu, X. Color demosaicking via directional linear minimum mean square-error estimation. *IEEE Transactions on Image Processing* 14.12 (2005), 2167–2178.

[32] Hirakawa, K. and Parks, T. W. Adaptive homogeneity-directed demosaicing algorithm. *IEEE Transactions on Image Processing* 14.3 (2005), 360–369.

[33] Su, C.-Y. Highly effective iterative demosaicing using weighted-edge and color-difference interpolations. *IEEE Transactions on Consumer Electronics* 52.2 (2006), 639–645.

[34] He, F.-L., Wang, Y.-C. F. and Hua, K.-L. Self-learning approach to color demosaicking via support vector regression. *2012 19th IEEE International Conference on Image Processing*. IEEE. 2012, 2765–2768.

[35] Sun, J. and Tappen, M. F. Separable Markov random field model and its applications in low level vision. *IEEE transactions on image processing* 22.1 (2012), 402–407.

[36] Go, J., Sohn, K. and Lee, C. Interpolation using neural networks for digital still cameras. *IEEE Transactions on Consumer Electronics* 46.3 (2000), 610–616.

[37] Kapah, O. and Hel-Or, H. Z. Demosaicking using artificial neural networks. *Applications of Artificial Neural Networks in Image Processing V*. Vol. 3962. International Society for Optics and Photonics. 2000, 112–120.

[38] Syu, N.-S., Chen, Y.-S. and Chuang, Y.-Y. Learning deep convolutional networks for demosaicing. *arXiv preprint arXiv:1802.03769* (2018).

[39] Dong, C., Loy, C. C., He, K. and Tang, X. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence* 38.2 (2015), 295–307.

[40] Chang, H., Yeung, D.-Y. and Xiong, Y. Super-resolution through neighbor embedding. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* Vol. 1. IEEE. 2004, I–I.

[41] Elad, M. *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer Science & Business Media, 2010.

[42] Egiazarian, K. and Katkovnik, V. Single image super-resolution via BM3D sparse coding. *2015 23rd European Signal Processing Conference (EUSIPCO)*. IEEE. 2015, 2849–2853.

[43] Dong, C., Loy, C. C. and Tang, X. Accelerating the super-resolution convolutional neural network. *European conference on computer vision*. Springer. 2016, 391–407.

[44] Mao, X., Shen, C. and Yang, Y.-B. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. *Advances in neural information processing systems*. 2016, 2802–2810.

[45] 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. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, 4681–4690.

[46] Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y. and Change Loy, C. Esrgan: Enhanced super-resolution generative adversarial networks. *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, 0–0.

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

[48] Jolicoeur-Martineau, A. The relativistic discriminator: a key element missing from standard GAN. *arXiv preprint arXiv:1807.00734* (2018).

[49] Johnson, J., Alahi, A. and Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. *European conference on computer vision*. Springer. 2016, 694–711.

[50] He, K., Zhang, X., Ren, S. and Sun, J. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, 770–778.

[51] Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K. Q. Densely connected convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, 4700–4708.

[52] Condat, L. and Mosaddegh, S. Joint demosaicking and denoising by total variation minimization. *2012 19th IEEE International Conference on Image Processing*. IEEE. 2012, 2781–2784.

[53] Khashabi, D., Nowozin, S., Jancsary, J. and Fitzgibbon, A. W. Joint demosaicing and denoising via learned nonparametric random fields. *IEEE Transactions on Image Processing* 23.12 (2014), 4968–4981.

[54] Ehret, T., Davy, A., Arias, P. and Facciolo, G. Joint demosaicing and denoising by overfitting of bursts of raw images. *arXiv preprint arXiv:1905.05092* (2019).

[55] Farsiu, S., Elad, M. and Milanfar, P. Multiframe demosaicing and super-resolution from undersampled color images. *Computational Imaging II*. Vol. 5299. International Society for Optics and Photonics. 2004, 222–233.

[56] Vandewalle, P., Krichane, K., Alleysson, D. and Süsstrunk, S. Joint demosaicing and super-resolution imaging from a set of unregistered aliased images. *Digital Photography III*. Vol. 6502. International Society for Optics and Photonics. 2007, 65020A.

[57] Zhang, K., Zuo, W. and Zhang, L. Learning a single convolutional super-resolution network for multiple degradations. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, 3262–3271.

[58] Agustsson, E. and Timofte, R. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. July 2017.

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

[60] Zhang, L., Wu, X., Buades, A. and Li, X. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *Journal of Electronic imaging* 20.2 (2011), 023016.

[61] Martin, D., Fowlkes, C., Tal, D., Malik, J. et al. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. Iccv Vancouver: 2001.

[62] Huang, J.-B., Singh, A. and Ahuja, N. Single image super-resolution from transformed self-exemplars. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, 5197–5206.

[63] Timofte, R., De Smet, V. and Van Gool, L. A+: Adjusted anchored neighborhood regression for fast super-resolution. *Asian conference on computer vision*. Springer. 2014, 111–126.

[64] Yang, C.-Y. and Yang, M.-H. Fast direct super-resolution by simple functions. *Proceedings of the IEEE international conference on computer vision*. 2013, 561–568.

[65] Yu, K., Dong, C., Loy, C. C. and Tang, X. Deep convolution networks for compression artifacts reduction. *arXiv preprint arXiv:1608.02778* (2016).