EPFL

Thèse n° 7317

Towards Real-World Super-Resolution using Deep Neural Networks

Présentée le 6 août 2020

à la Faculté informatique et communications
Laboratoire d'images et représentation visuelle
Programme doctoral en informatique et communications

pour l'obtention du grade de Docteur ès Sciences

par

Ruofan ZHOU

Acceptée sur proposition du jury

Prof. W. A. Jakob, président du jury
Prof. S. Süsstrunk, directrice de thèse
Prof. C. C. Loy, rapporteur
Dr R. Timofte, rapporteur
Prof. P. Frossard, rapporteur

École polytechnique fédérale de Lausanne

2020

Acknowledgements

The five years of my PhD research have been, by far, the best five years of my life until now. This experience has significantly changed my life and helped me evolve and grow in many positive ways. I would like to express my most sincere gratitude to the people who helped me and shaped me during my PhD work.

First, I would like to express my deepest gratitude to my thesis advisor Prof. Sabine Süsstrunk for the continuous support of my PhD research. Her patience, motivation, enthusiasm, and immense knowledge guided me to be professional and helped me out when the road got rough. She was very supportive and gave me the freedom to pursue various projects. Her technical and editorial advice was essential to the completion of this thesis. I am also very grateful for the innumerable lessons and insights she taught me about academic research and social skills. I could not have imagined having a better advisor and mentor for my PhD experience.

I would like to thank the rest of my thesis committee, Prof. Wenzel Jakob, Prof. Pascal Frossard, Prof. Chen Change Loy, and Dr. Radu Timofte for their time in reviewing my thesis, their insightful comments and their encouragement.

I also thank my fellow labmates in IVRL. Thanks to my great office-mate Majed El Helou, our friendship has led to many interesting and good-spirited discussions on research and life. Besides Majed, I am also grateful for the great opportunities I had to work with Dr. Radhakrishna Achanta, Fayez Lahoud and Marjan Shahpaski, they helped me a lot on my projects with various aspects. I thank our great secretary, Françoise Behn for the administrative assistance. I thank Damian Firmenich, Dr. Bin Jin, Chen Liu, Dr. Siavash Arjomand Bigdeli, Deblina Bhattacharjee, Bahar Aydemir, Leonardo Laurence Impett, Dr. Sami Arpa, Dr. Edo Collins, Baran Ozaydin and Huck Yang. I was very fortunate to meet these great colleagues, they made life in IVRL very interesting and colorful.

I further thank people who were not part of IVRL but who helped me out in research in general, including Xiaoyu Xiang, Yapeng Tian, Kaicheng Yu, Frederike Dümbgen, Vidit Vidit, Hanjie Pan, Yoann Moulin, and many more. I thank them for all their technical assistance, as well as for the helpful discussions I had with them. I also thank Holly Cogliati for her editorial assistance.

I would like to thank my friends in PolyProg. I really enjoyed joining and organizing coding competitions with Wajeb, Sam, Weiyu, Tasho, Solal, Tatiana, Betty, Jakub, Slobodan, Andrii, Joseph and Michalina. I also appreciate the wonderful time with my Chinese friends, Cheng, Min, Xiaokang, Jingjing, Ziyi, Ruojun, Nan, Donghe, Jingyan, Runwei, Su, Shina, Xin, and many others. Their company helped me to overcome the homesickness.

Acknowledgements

My special thanks go to my boyfriend, Laurent, for his valuable advice on my thesis, as well as his love, patience and understanding. I thank him for the priceless support that he and his family gave to me.
Finally, my wholehearted appreciation to my parents for their endless love, support and encouragement. Without them, I would never have enjoyed so many opportunities, and I dedicate this thesis to them.

Lausanne, July 29, 2020

Abstract

Image super-resolution reconstructs a higher-resolution image from the observed low-resolution image. In recent years, machine learning models have been widely employed and deep learning networks have achieved state-of-the-art super-resolution performance. Most of these methods, however, are trained and evaluated on simulated datasets that assume a simple and uniform degradation model. The deep learning models trained on such simulated datasets fail to generalize to practical applications because the actual or real degradation in real-world low-resolution images is much more complex.

In this thesis, we propose several approaches to improve the robustness and generalization of deep super-resolution models. The first technique reduces the accumulation of errors in the camera pipeline. We build a deep residual network for learning an end-to-end mapping between raw images and high-resolution images. Our proposed network, trained on high-quality samples, is able to reconstruct in a single step high-quality super-resolved images from low-resolution Bayer mosaics . Extensive experiments show that the proposed method achieves better results than the state-of-the-art techniques, both qualitatively and quantitatively.

To resolve the problem of the mismatch between the applied blur kernel in the synthetic dataset and the real-world camera blur, we propose to incorporate blur kernel modeling in the training. We generate the super-resolution training dataset by employing a set of realistic blur kernels estimated from real low-resolution photographs. We build a pool of realistic blur kernels with a generative adversarial network; then, we train a super-resolution network using the low-resolution images constructed with the generated kernels. Our method reconstructs more visually plausible high-resolution images compared to other state-of-the-art methods that rely on a simple degradation model.

In order to study the effect of noise on super-resolution, we collect a dataset that contains pairs of noisy low-resolution images and the corresponding high-resolution images by using microscopy. We then benchmark the combinations of denoising methods and super-resolution networks on the collected dataset. Our experimental results show that the super-resolution networks are sensitive to noise, and that the consecutive application of two applications suffers from the accumulation of errors. The benchmark results also suggest that the networks can benefit from joint optimization, hence we use a single network for joint denoising and super-resolution. Our network, trained with a novel texture loss, outperforms any combination of state-of-the-art deep denoising and super-resolution networks.

Finally, to take advantage of multi-modal data available in certain applications, we propose a super-resolution system based on the fusion of information from multiple sources. For

the application of spectral image super-resolution, we use two downsampled versions of the same image to infer a better high-resolution image for training. We refer to these inputs as a multi-scale modality. As color images are usually taken at a resolution higher than spectral images, we make use of color images as another modality to improve the super-resolution network. We build a pipeline that learns to super-resolve by using multi-scale spectral inputs guided by a color image by combining both modalities. Our proposed method is economical in time and memory consumption, yet achieves competitive results.

Keywords: super-resolution, signal processing, image restoration, dataset, neural network, deep learning, generative adversarial network, multi-modal, signal processing

Résumé

La super-résolution (SR) vise à créer, à partir d'une image de définition donnée, une image de plus haute définition. Les techniques récentes d'apprentissage profond ont permis d'atteindre de nouveau sommets en SR. Toutefois la plupart des méthodes sont entrainées et testées avec des bases de données simulées, qui utilisent des modèles de dégradation uniformes simples. Ces méthodes échouent lorsqu'elles sont appliquées à de vraies images de faible définition, qui ne coïncident pas à ces modèles de dégradation simples.

Nous proposons plusieurs approches afin d'améliorer la robustesse et la polyvalence des modèles SR. La première vise à réduire l'accumulation d'erreur dans la chaine de traitement de la caméra. Nous avons construit un réseau profond résiduel pour la mise en rapport direct entre les images brutes du capteur et leurs équivalents haute définition. Le réseau, entrainé sur des échantillons de haute qualité, est capable de créer en une seule étape une image SR de haute qualité à partir d'une mosaïque de Bayer. Des testes montrent que cette approche donne de meilleurs résultats que les autres méthodes de pointes, aussi bien au niveau quantitatif qu'à celui de la qualité perçue.

Pour résoudre le problème de la non-concordance entre le masque de flou utilisé pour la base de donnée simulée et celui des vrais appareils photos, nous proposons d'ajouter un modèle de ce masque dans la phase d'entrainement. Nous avons généré la base de données d'entrainement grâce à un ensemble de masque de flou réaliste, obtenu à partir de vraies photographies en basse définition. Nous avons construit une collection de masques réaliste grâce à un réseau adverse génératif. Nous avons entrainé un réseau à la SR en utilisant les images de faible définition obtenues avec ces masques. La méthode permet de reconstruire des images de haute définition plus réalistes par rapport aux méthodes conventionnelles.

Afin d'étudier l'impact du bruit, inévitable en SR, nous avons collecté un ensemble d'images faites par microscopie. Cet ensemble comprend des photos bruitées de faible définition, ainsi que leur équivalant en haute définition. Nous avons ensuite évalué la combinaison de dé-bruitage et de réseau de SR sur cet ensemble de données. Cela nous a permis de montrer que les réseaux de SR sont sensibles au bruit, et l'application consécutive de ces deux approches souffrait de l'accumulation d'erreur de chaque étape. Par contre les résultats suggèrent que les réseaux peuvent bénéficier d'une optimisation conjointe. Nous avons donc utilisé un seul réseau effectuant simultanément le dé-bruitage et la SR. Notre réseau, entrainé avec une nouvelle fonction objectif sur la perte de texture, a surpassé toutes les combinaisons des meilleurs réseaux de dé-bruitage et de SR.

Pour les applications disposant de données multimodales, nous proposons une SR basée

sur la fusion de sources multiples. Pour l'application à l'imagerie spectrale, nous utilisons deux versions sous-échantillonnées de la même image afin de mieux entrainer notre réseau. Puisque l'imagerie spectrale donne souvent des photos de plus basse définition que les photos conventionnelles, nous utilisons ces dernières comme moyen pour améliorer notre réseau. Nous construisons une chaine de traitement qui apprend la SR en utilisant des images spectrales à différentes échelles et guidé par une image conventionnelle. Cette méthode est peu gourmande en temps et en mémoire et donne des résultats compétitifs.

List of Figures

List of Tables

Contents

Abbreviations and Notation

List of Abbreviations

Abbreviation	Description
SR	Super-resolution
HR	High-resolution
LR	Low-resolution
ISP	Image Signal Processor
CMOS	Complementary Metal–Oxide–Semiconductor
DL	Deep learning
NN	Neural network
CNN	Convolutional neural network
GAN	Generative adversarial network
ReLU	Rectified-linear unit
MSE	Mean Square Error
PSNR	Peak Signal to Noise Ratio
SSIM	Structural SIMilarity Index
MRAE	Mean Relative Absolute Error
SID	Spectral Information Divergence

List of Symbols

Symbol	Description
I	An image
I^{HR},I^{LR},I^{SR}	The high-resolution, low-resolution, and super-resolved version of the image
I^{raw}	The raw version of the image
I^{clean}, I^{noisy}	An image of its noise-free and noisy version
$I_{spectral}$, I_{color}	A multi-spectral image and its 3-channel (RGB) color version
$\mathbb{I}$	A set of images
P	A patch extracted from an image
h, w	Hight and width of an image

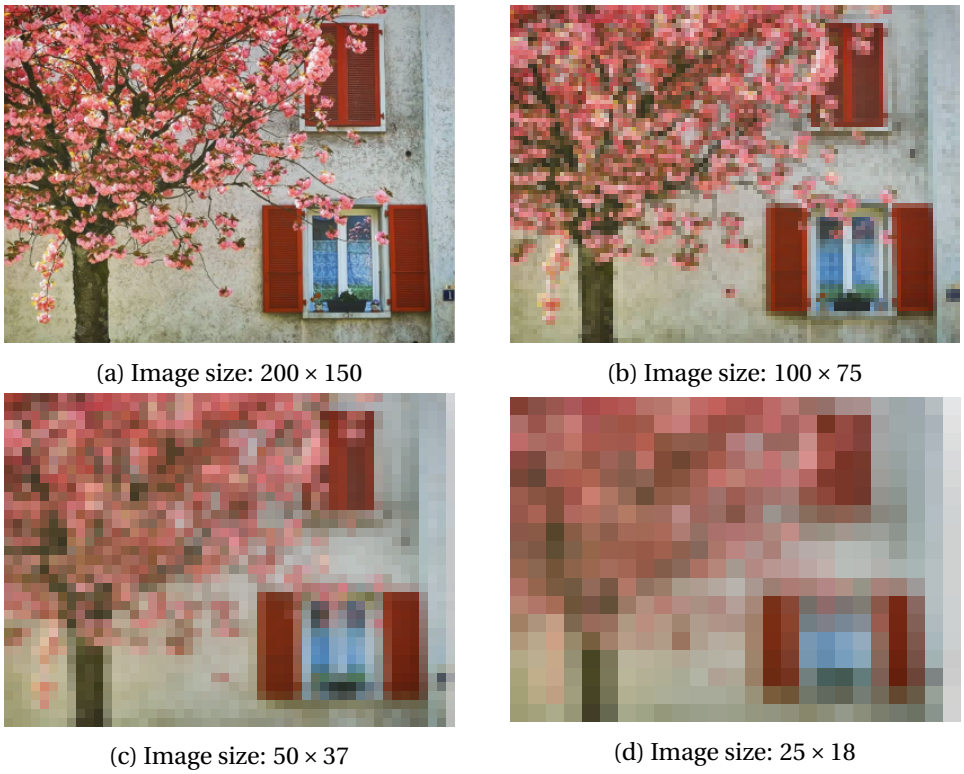
Symbol	Description
k	A blur kernel
$\mathbb{K}$	A set of blur kernels
σ	Variance for noise modeling or kernel modeling
g_σ	A gaussian kernel with variance of σ
$\otimes$	Convolution operator
$\odot$	Element-wise multiplication
s	An upscaling or downscaling factor
$\times s$	Super-resolution by a factor of s
$\downarrow_s$	A downsampling operator with a factor of s
$\uparrow_{bic}$	A bicubic upsacling operator
n	noise
$\mathscr{L}$	A loss function defining an optimization objective
$\mathbb{E}$	An expectation
x	An observed variable
y	The output of the network
ϵ	A random variable of noise
p	A possibility
$\mathbb{P}$	A probability distritubion
$N(\mu,\sigma)$	A normal distribution with mean μ and variance
z	A randomer vector following some distribution
D, G	Discriminator and Generator of a GAN system
C_i	A convolutional layer
W_i, b_i	Parameters of a convolutional layer
$\Phi(x)$	The extracted VGG features of an input x
β_1,β_2	Parameters for ADAM optimization
α, θ, μ	Parameters for algorithm
δ	Gradient of a variable
Mean$(\cdot)$	Mean value
Var$(\cdot)$	Variance value
Cov$(\cdot)$	Covariance value

1 Introduction

Resolution is one of the most important attributes that affect the visual quality of an image. An image at a higher resolution contains more detailed information, which is beneficial for further image processing tasks, such as image recognition, segmentation, and analysis by either humans or machines. With the development of high-definition display devices, such as 4K and 8K TVs, it is more desirable to acquire high-resolution (HR) images than low-resolution (LR) ones because the images are more pleasant to view on high-definition monitors. Figure 1.1 illustrates an image of the same scene displayed at varying resolutions. The resolution decreases from (a) to (d), successively by a factor of two. Figure 1.1a is the original image that provides details such as the texture of the flowers and the curtain behind the window. These details gradually disappear with the reduction of the resolution. When the image resolution is decreased to 25×18 in (d), it becomes difficult to identify the content of the image. In daily life, the LR images are more often collected due to deficiencies in imaging equipment or a limitation in storage. Therefore, methods that enhance the resolution of images are in great demand.

There are two common approaches to acquiring HR images. The first approach is based on hardware improvement. The more pixels a complementary metal-oxide-semiconductor (CMOS) camera has, the higher the resolution will be. Therefore, one method to increase the spatial resolution is to enlarge the chip size so that more CMOS sensors can be included on the chip [100]. An alternative is to reduce the pixel size per unit area so that more pixels are implemented on a fixed size chip [95]. However, smaller pixel sizes might not result in higher resolution as the maximum sampling frequency is determined by the diffraction limit (Airy disc) of the optics. Both solutions suffer from the limitation of higher cost and development of sophisticated manufacturing technology.

The second approach is based on software improvement; it requires designing more accurate and faster algorithms to enhance the resolution from low to high. This solution is feasible, as highly developed computing units, such as graphic processing unit (GPU) and image signal processor (ISP), can handle computationally intensive tasks. Algorithms that reconstruct HR images from the LR ones are named super-resolution (SR) techniques. These techniques



(a) Image size: 200 × 150

(b) Image size: 100 × 75

(c) Image size: 50 × 37

(d) Image size: 25 × 18

Figure 1.1 – The original image (a) with its downsampled LR versions (c)-(d) displayed with same width and height.

enhance images by providing a visual quality better than its LR counterpart.

Deep learning (DL) is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Deep learning has shown prominent superiority over other machine learning algorithms in various fields such as computer vision, natural language processing, audio recognition, machine translation, *etc.* Benefiting from the high capacity of extracting effective high-level abstractions that bridge the LR and HR space, recent DL based SR methods have achieved significant improvements over the conventional signal processing based methods.

Despite their success, these methods are severely limited by their reliance on the assumed degradation model between the HR image and the LR image. It is known that they do not generalize to natural images, as the authentic degradation in real-world LR images are much more complicated. Figure 1.2 shows such an example, where the state-of-the-art SR network fails on the noisy photograph. Hence, it becomes increasingly important to develop SR techniques that are more practical under real scenarios.

In this thesis, we develop several such approaches. These algorithms can be summarized into two categories:

(a) a noisy LR image (b) SR result from RCAN [141]

Figure 1.2 – Example of a failure case of the state-of-the-art SR network [141] where the SR network amplify the noise and introduces unpleasant artifacts.

- **To build robust single image SR approaches that take natural image characteristics into account.** SR on natural images is challenging as many natural images suffer from various corruptions such as sensor noise, artifacts, aperture blur, *etc.* In this thesis, we address these problems by modeling different parameters in SR methods. Our SR methods are able to produce visually pleasing results and generalize on unknown corruptions.

- **To develop a technique that can take advantage of multi-modal data for better reconstruction results.** In some applications of SR, the signal is available in multiple modalities. We show an example in spectral image SR where we are able to fuse the information from different modalities to obtain better reconstruction results.

1.1 Super-Resolution on Natural Images

SR in practical applications, *e.g.* SR on natural images, is usually a problem mix of noise, blur, and resolution limitations such as color mosaic and insufficient resolution. Although the problems can be partially resolved by applying different image restoration tasks (denoising, deblur, demosaicing, *etc.*) sequentially, this might bring on new problems such as unexpected artifacts and blur. Applying SR on top of demosaiced images will amplify the error and lead to unpleasant results. Therefore, it is important to consider these aspects in the SR process.

Among all possible aspects, we concentrate on three applications of SR: (1) SR with raw images, (2) blind SR where the blur kernel is unknown, and (3) SR on noisy images.

1.1.1 Super-Resolution on Raw Images

Most SR algorithms take only the color image as input, whereas modern cameras provide both the raw data and the pre-processed color image produced by the image signal processing system (ISP). Hence, these SR systems do not make full use of the radiance information existing in raw data. A typical ISP pipeline is shown in Figure 1.3: it includes several nonlinear operations such as demoisaicing, tone adjustment, and compression. The linear degradations in the imaging process, including blur and noise, are nonlinear in the processed RGB space, which makes image restoration more difficult. Furthermore, the demosaicing step in the ISP is highly related to SR, as these two problems refer to the resolution limitations of cameras. Therefore, solving the SR problem with pre-processed images is sub-optimal and could be inferior to a single unified model that simultaneously solves both problems.

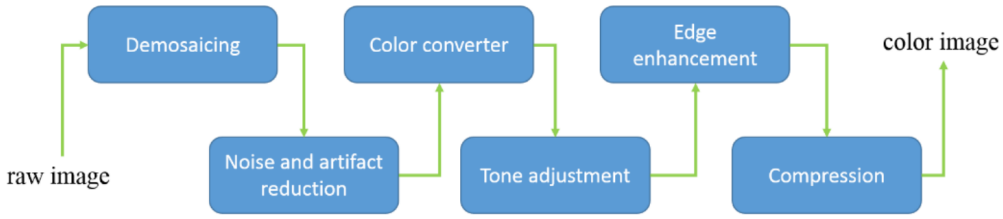


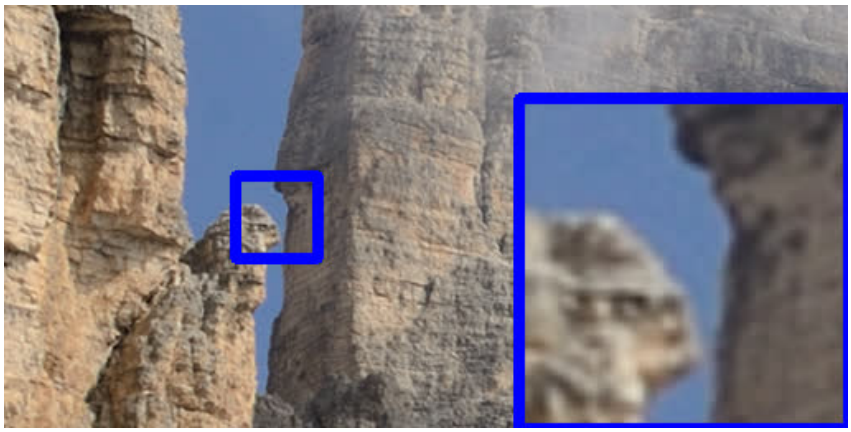
Figure 1.3 – A typical ISP pipeline [10].

We propose a deep residual network for learning an end-to-end mapping between raw images and HR images. By training on high-quality samples, our deep residual demosaicing and SR network is able to recover high-quality super-resolved images from LR Bayer mosaics in a single step without producing the artifacts commonly produced when the two operations are done separately. We perform extensive experiments to show that our deep residual network achieves demosaiced and super-resolved images that are superior, both qualitatively and quantitatively, to the state-of-the-art. Figure 1.4 shows example results of our joint demoisaicing and SR network.

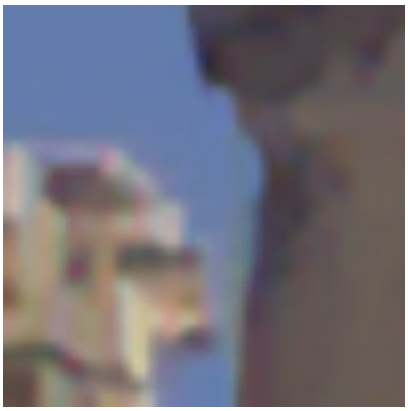
1.1.2 Blind Super-Resolution

Most of the existing advanced SR methods assume that the downsampling blur kernel is known and pre-defined. However, the blur kernels involved in real applications are typically unknown and can vary. As previous works [28, 94, 45] show, learning-based SR methods suffer severe performance drop when the pre-defined blur kernel is different from the real one. This phenomenon of kernel mismatch will cause the network to produce undesired artifacts. Hence, the problem with unknown blur kernels, also known as *blind SR*, fails in most of the deep learning based SR methods and largely limits their usage in real-world applications.

To improve generalization and robustness of deep SR CNNs in real applications, we propose to incorporate blur-kernel modeling in the training of the SR networks. The proposed method

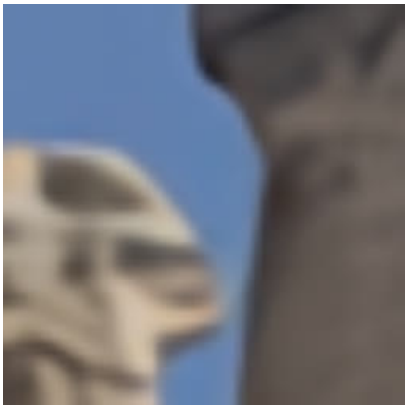


(a) Reference image

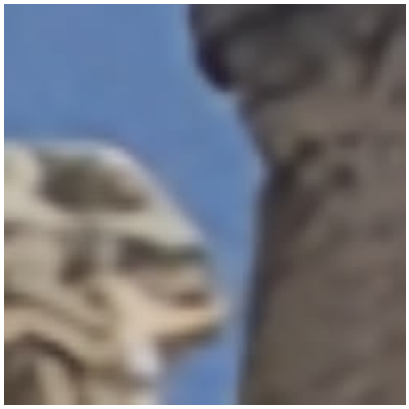


(b) ADMM[74]+SRCNN[24]

(c) FlexISP[60]+SRCNN[24]



(d) DemosaicNet[42]+SRCNN[24]



(e) Our output

Figure 1.4 – Comparison of our joint demosaicing and SR output to the results from state-of-the-art methods. Our method exhibits none of the unpleasant artifacts and is able to faithfully reconstruct the original.

consists of two stages: (1) We first build a pool of realistic blur-kernels with a generative adversarial network (GAN) and (2) then train a SR network with HR and corresponding LR images constructed with the generated kernels. Our extensive experimental validations demonstrate

(a) LR image (b) result from RCAN [141] (c) result from our method

Figure 1.5 – Super-resolving (×4) natural image (left) with RCAN [141], and our approach. Our approach learns to handle the unknown blur kernel in natural images, while RCAN fails to generalize.

the effectiveness of our single-image SR approach on photographs with unknown blur-kernels, as shown in Figure 1.5.

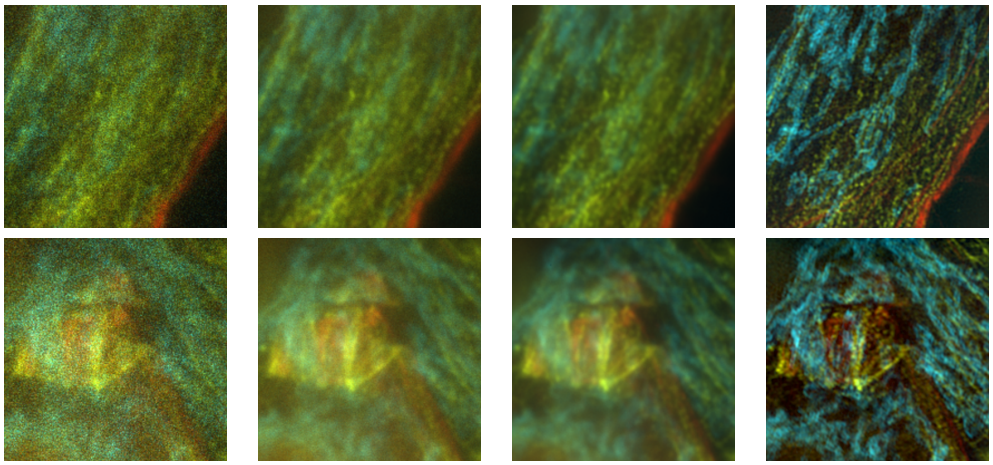
1.1.3 Super-Resolution on Noisy Images

Due to the sensing technologies, image processing, and transmission, images are inevitably contaminated with noise during acquisition, transmission, and compression, thus leading to distortion and loss of image information [36]. With the presence of noise, most SR algorithms fail as they magnify the noise as much as the image details and texture, causing unexpected artifacts in the reconstruction results. Such an example is shown in Figure 1.2.

In order to study the effect of noise on SR algorithms, we collect a dataset by using microscopy equipment and techniques [49, 127]. Our dataset is comprised of noisy LR images with various noise levels, a noise-free LR image, and a corresponding high-quality HR image. Visual examples of the images in the dataset are shown in Figure 1.6. This dataset enables us to benchmark denoising and SR methods. Our experimental results show that state-of-the-art SR networks perform very poorly on noisy inputs. However, applying the best denoiser, in terms of reconstruction error, followed by the best SR method does not yield the best result. Therefore, we propose a joint optimization for denoising and SR through a single network. To enable the network to reproduce a faithful texture, we use a texture loss that exploits the second-order statistics of feature maps. Our SR network, trained with the proposed texture loss, outperforms any combination of state-of-the-art deep denoising and SR networks, even though it contains far fewer parameters than the other methods,

1.2 Multi-modal Super-Resolution

In many practical application scenarios, different sensors can yield different image modalities of a certain scene. For example, it is typical in remote sensing to have various image modalities



high noise level LR | low noise level LR | Noise-free LR | Ground-truth HR

Figure 1.6 – Example of image sets in the proposed joint denoising and SR dataset.

of each observation, such as a multi-spectral band image, an infrared (IR) band image, and a panchromatic image. In order to balance bandwith, complexity, and cost, these multi-modal images are usually acquired with different resolutions [117]. In these applications, we need algorithms that are able to super-resolve the LR images with the help of the HR images of a different modality .

In this thesis, we address the problem of multi-modal spectral image SR yet constrain ourselves to a small dataset. We propose the use of different modalities for improving the performance of neural networks on the spectral SR problem. First, we use multiple downsampled versions of the same image to infer a better HR image for training, we refer to these inputs as a multi-scale modality. As color images are usually taken at a resolution higher than spectral images, we make use of color images as another modality for improving the SR network. By combining both modalities, we build a pipeline that learns to super-resolve using multi-scale spectral inputs guided by a color image. Finally, we validate our method and show that it is economic in terms of parameters and computation time yet still produces state-of-the-art results.

1.3 Thesis Outline

This thesis consists of seven chapters, the current chapter being the introduction. The general outline and structure of the remainder of the thesis are summarized below.

- Chapter 2: Related Work
  We discuss the work relevant to this thesis, summarized into two sub-fields: (1) neural networks, and (2) image restoration

- Chapter 3: Super-Resolution with Raw Images
  In this chapter, we introduce a neural network for joint demoisaicing and SR on raw

images. We compare this design with other methods and show the advantage of our approach.

- Chapter 4: Blind Image Super-Resolution
  In this chapter, we introduce the kernel-modeling SR for unknown kernel in blind SR. We discuss the importance of modeling kernel in SR and validate performance through qualitative and quantitative experiments.

- Chapter 5: Joint Denoising and Super-Resolution
  We present a dataset on joint denoising and SR. We benchmark the state-of-the-art methods on the dataset and discuss the importance of joint optimization.

- Chapter 6: Multi-modal Super-Resolution
  We present a fusion system that combines the information from multiple modalities for spectral image SR. The experimental results show that our method is economic in terms of parameters and computation time.

- Chapter 7: Conclusion
  In this chapter, we present a summary of the thesis and highlights the major contributions. We also suggest several future research directions.

1.4 Publications

During my research undertaken for this thesis and my PhD studies, I wrote and contributed to the following publications:

1. Zhou, R., Achanta, R., and Süsstrunk, S. (2018). Deep residual network for joint demosaicing and super-resolution. *Color and Imaging Conference (CIC)*.

2. Zhou, R. and Süsstrunk, S. (2019). Kernel modeling super-resolution on real lo-resolution images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

3. Zhou, R., El Helou, M. and Süsstrunk, S. (2020). W2S: A Joint Denoising and Super-Resolution Dataset. Submitted to *European Conference on Computer Vision Workshops (ECCVW)*

4. Zhou, R., Lahoud, F. and Süsstrunk, S. (2018). Multi-modal spectral image super-resolution. In *Proceedings of the European Conference on Computer Vision Workshops (ECCVW)*.

2 Related Work

The super-resolution algorithms developed in this thesis are related to several research fields. As all the algorithms involve neural networks (NNs) to some extent, and specifically convolutional neural networks, we start with an introduction of NNs in Section 2.1. A great number of techniques have been proposed in this field. In this chapter, we review mainly three types of networks that are most related to this thesis: (1) Convolutional Neural Networks (CNNs), (2) Fully-Convolutional Neural Networks (FCNs), and (3) Generative Adversarial Networks (GANs).

In Section 2.2, we introduce the inverse problem of image restoration. Then we review the state-of-the-art super-resolution (SR) algorithms in Section 2.2.2. From Section 2.2.3, we briefly explain the other image restoration tasks that are involved in the real-world SR applications and discuss the recent algorithms.

2.1 Neural Networks

The history of neural networks began in the 1940s, when the functionality of neurons was discovered by neurophysiologists [92]. Since then, researchers developed various models with multiple layers of neurons. For many years, the performance of neural networks lagged far behind the requirements of real-world applications, mainly because of the limitation of datasets and the computing resources, However, due to recent progress in computing hardware and dataset collection, training very deep neural networks is now feasible. In 2012, Krizhevsky *et al.* [76] proposed a deep neural network for an image classification task that achieved breakthrough performance. Thereafter, deep neural networks have been widely adopted for many fields, such as computer vision, natural language processing, and robotics. Various network architectures with increasing complexity and performance have been published.

In Section 2.1.1, we inspect the CNNs developed for classification and regression tasks. We then discuss, in Section 2.1.2, the FCNs that are designed for image segmentation and that are widely used for image restoration. We present the details about GANs in Section 2.1.3.

2.1.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a subtype of discriminative deep architecture and have shown satisfactory performance in processing two-dimensional data with grid-like topology, such as images and videos. The concept of CNNs is inspired by time-delay neural networks (TDNNs). In a TDNN, the weights are shared in a temporal dimension, which leads to reduction in computation. In CNNs, the convolution has replaced the general matrix multiplication in standard NNs. In this way, the number of weights is decreased, thereby reducing the complexity of the network. Furthermore, the images, as raw inputs, can be directly imported to the network, thus avoiding the feature extraction procedure in the standard learning algorithms. It should be noted that CNNs are the first truly successful deep learning architecture due to the successful training of the hierarchical layers. The CNN topology uses spatial relationships to reduce the number of parameters in the network, and the performance is improved using the standard backpropagation algorithms. Another advantage of the CNN model is that it requires minimal pre-processing. With the rapid development of computation techniques, the GPU-accelerated computing techniques have been exploited to train CNNs more efficiently. Today, CNNs are successfully applied to hand writing recognition, face detection, behavior recognition, speech recognition, recommender systems, image classification, and natural language processing.

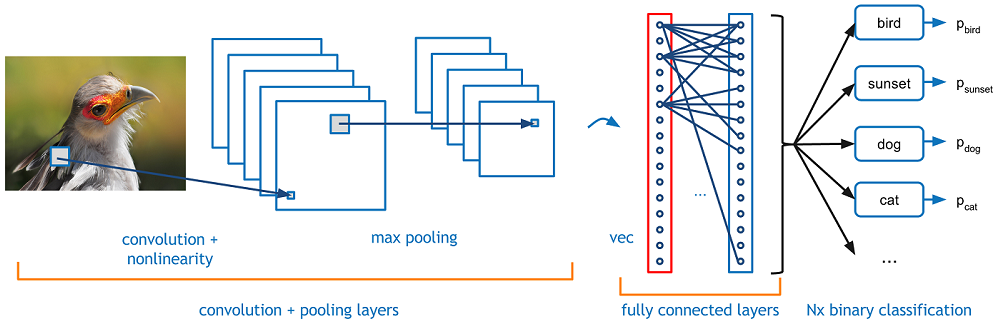


Figure 2.1 – Schematic structure of CNNs[1].

A CNN is a multi-layer neural network that consists of several different types of layers, *i.e.*, convolution layers, activation functions, pooling layers, normalization layers, and fully connected layers. The schematic structure of CNN for a classification task is shown in Figure 2.1. In this example, the input image is convolved with trainable filters in order to produce feature maps in the first convolution layer. A layer of connection weights are included in each filter. Passed through a nonlinear activation function, these pixels produce additional feature maps in the first pooling layer. This procedure carries on, and we can thus obtain the feature maps in the following convolution layers and pooling layers. Finally, the values of these pixels are rasterized and displayed in a single vector as the output of the network. In this section, we briefly introduce two network architectures that we use in our work: VGG [115] and ResNet [59].

[1] Illustration taken from: https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/

The successful use of CNNs in image recognition tasks has accelerated the research in architectural design. In this regard, Simonyan [115] proposed a simple and effective design principle for CNN architectures. Their architecture, named as VGG, is shown in Figure 2.2. The architecture of VGG is inspired by experimental results that suggest that small size filters can improve the performance of the CNNs. Based on these findings, VGG replaced the 11×11 and 5×5 filters that were commonly used in the previous CNNs with a stack of 3×3 filter layers and experimentally demonstrated that concurrent placement of small size filters could induce the effect of large size filters. The use of small size filters provides an additional benefit of low computational complexity by reducing the number of parameters. These findings set a new trend in research to work with smaller size filters in CNN; and these settings are commonly used in the later network architectures. VGG regulates the complexity of a network by placing 1×1 convolutions in between the convolutional layers that learn a linear combination of the resultant feature maps. For the tuning of the network, max-pooling is placed after the convolutional layer, and padding is performed to maintain the spatial resolution. VGG showed good results both for image classification and localization problems. The features extracted from different layers of the VGG network are also used in many image restoration and image generation works for evaluating perceptual difference [67].

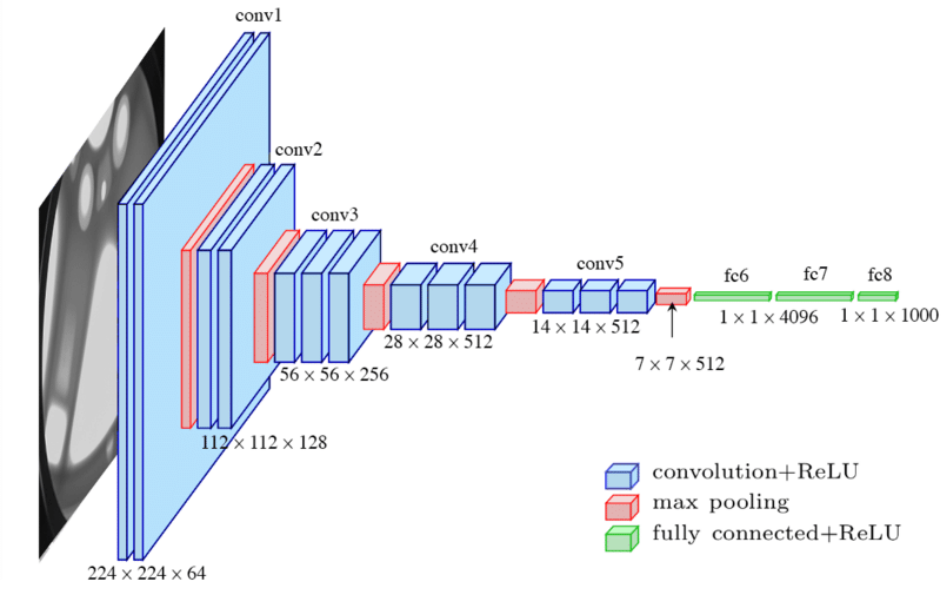


Figure 2.2 – The structure of VGG16. Figure taken from [38]

ResNet is proposed by He [59]; it is considered as a continuation of deep networks. ResNet revolutionized the CNN architecture by introducing the concept of residual learning and by devising an efficient methodology for the training of deep networks. ResNet is a network comprised of 35 residual blocks. The architecture of the residual block of ResNet is shown in Figure 2.3. Although ResNet contains 152 layers, which is eight times deeper than VGG, it has a computational complexity lower than previously proposed networks. The good performance

of ResNet on image recognition and localization tasks shows that representational depth is of central importance for many visual recognition tasks. The residual block architecture and the concept of residual learning is also successfully employed in other vision tasks, such as image restoration.

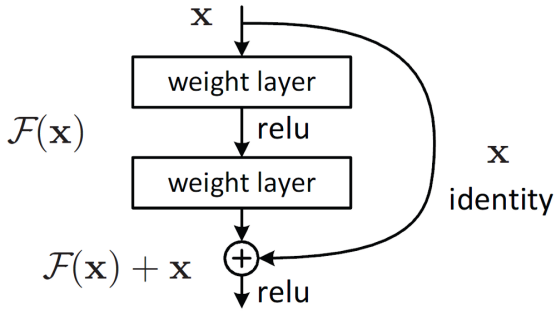


Figure 2.3 – Residual block in ResNet [59].

2.1.2 Fully Convolutional Networks

Many current CNNs are designed for image classification tasks. They transform the input image into a 1D feature vector, thus rendering it difficult to maintain the spatial information (the local information at each position) from the input image in the feature vector. For some applications, however, spatial information is critical. For instance, image restoration assigns a new pixel value to each pixel in the input image. The new pixel value is determined mainly by the local information at that specific position. In such a scenario, traditional CNNs are therefore not suitable.

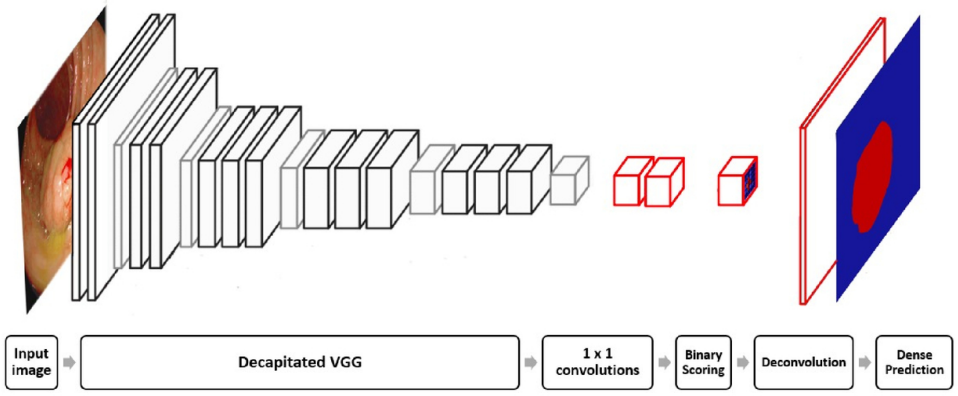


Figure 2.4 – Architecture of FCN-VGG [87].

Long [87] modified the VGG network into a fully convolutional architecture, which can handle inputs of various sizes and generate outputs of the same size as the inputs. Each fully connected layer is transformed into a convolution layer with a kernel that covers the entire input region. Transforming all the fully connected layers in a CNN leads to a fully convolutional

network (FCN), shown in Figure 2.4. FCN effectively performs image classification on every patch of the input image by using a sliding window, thus generating a probability value for each patch. The output from FCN is a 3D heatmap containing probabilities of each class. Bilinear interpolation is then applied to the heat map to generate a segmentation mask that is of the same size as the input image.

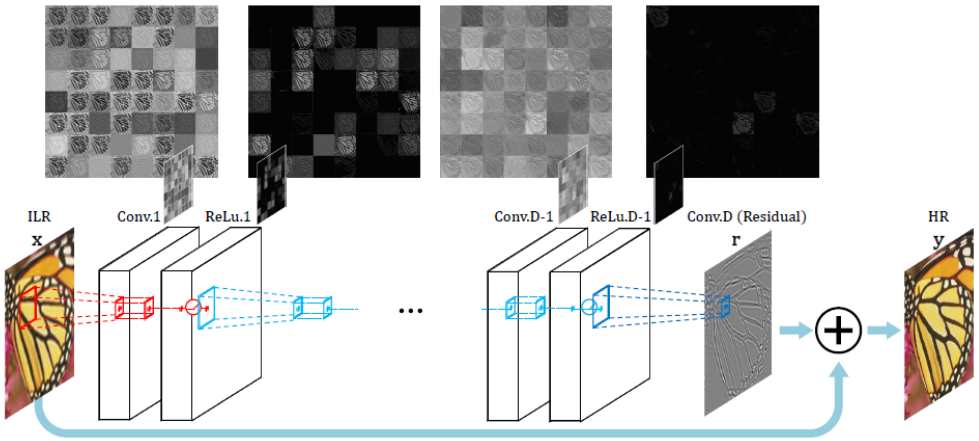


Figure 2.5 – Architecture of VDSR [71] for SR.

FCN-based networks are utilized for image restoration tasks. However, due to the pooling layers in CNNs, spatial information is gradually removed as the deeper one goes in the network architecture. Hence, for the image restoration tasks, pooling layers are usually abandoned. An example of a SR network that contains only convolution layers and activation layers is shown in Figure 2.5. More details are given in Section 2.2.

2.1.3 Generative Adversarial Networks

Generative models are a class of models that describes how a dataset is generated in terms of a probabilistic model. Among different generative models, Generative Adversarial Networks (GANs) [43] introduce a very clever internal adversarial training mechanism and has achieved success in various situations.

GANs were inspired by game theory, where the generator and discriminator compete with each other to achieve the Nash equilibrium in the training process. The architecture of GANs is illustrated in Figure 2.6. The purpose of generator G is to fit as much as possible the potential distribution of real data, whereas the purpose of discriminator D is to correctly distinguish real data from fake data. The input of the generator is a random noise vector z (usually a uniform or normal distribution). The noise is mapped to a new data space via generator G to obtain a fake sample, $G(z)$, a multi-dimensional vector. The discriminator D is a binary classifier, it takes both the real sample from the dataset and the fake sample generated by generator G as the input. The output of discriminator D represents the probability that the sample is a real rather than a fake. When the discriminator D cannot determine whether the data comes

from the real dataset or the generator, the optimal state is reached. At this point, we obtain a generator model G that has learned the distribution of the real data.

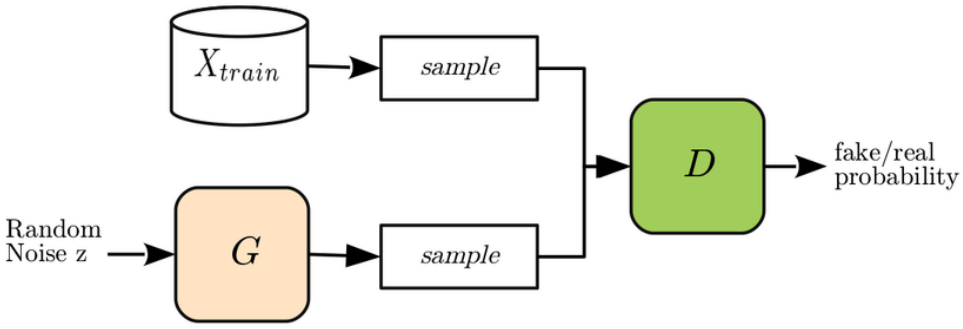


Figure 2.6 – The structure of GANs consisting of a generator and a discriminator. Figure taken from [55].

As two players in game theory, both the generator and the discriminator have their own loss functions. In this case, we call them $\mathscr{L}^G$ and $\mathscr{L}^D$, respectively. In [43], the discriminator D is defined as a binary classifier, and the loss function is represented by the cross entropy,

$$\mathscr{L}^D = -\frac{1}{2}\mathbb{E}_{x\sim p_{data}(x)}\log D(x) - \frac{1}{2}\mathbb{E}_z \log(1 - D(G(z))) \tag{2.1}$$

where x represents the real sample, z represents the random noise vector, $G(z)$ is the data generated by the generator, and $\mathbb{E}$ represents the expectation. $D(x)$ indicates the probability that D discriminates x as real data, and $D(G(z))$ indicates the probability that D determines the data generated by G to be real. The purpose of D is to correctly determine the source of the data, hence it wants $D(G(z))$ to approach 0, whereas the purpose of G is to bring it closer to 1. As a result, there exists a conflict between these two models (*i.e.*, zero-sum game). Therefore, the loss of the generator can be derived by the discriminator:

$$\mathscr{L}^G = -\mathscr{L}^D \tag{2.2}$$

Consequently, the optimization problem of GANs is transformed into the minmax problem as shown below,

$$\min_G \max_D \mathbb{E}_{e\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - D(G(z)))] \tag{2.3}$$

In the training process, the parameters in G are updated, along with the parameters' updating process in D. When $D(G(z)) = 0.5$, the discriminator cannot determine the differences between these two distributions and, in this state, the model will achieve the global optimal solution.

To enhance the GANs stability, it has been proposed to optimize the objective function. Che *et*

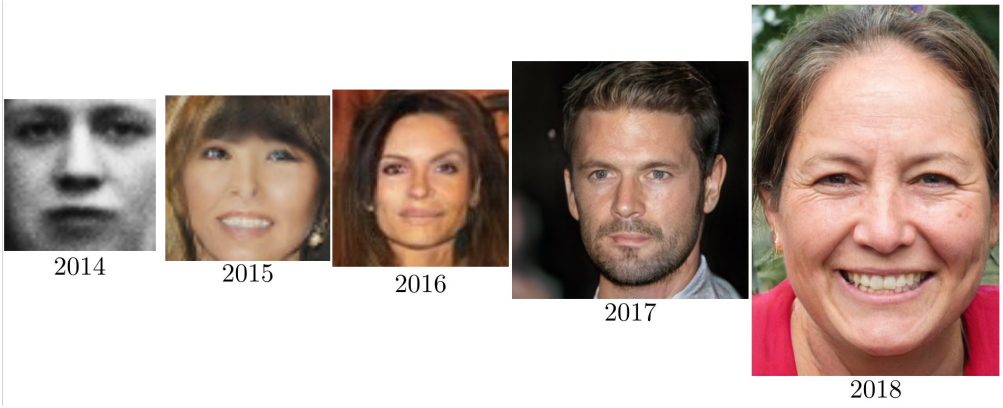


Figure 2.7 – Example of the progression in the capabilities of GANs from 2014 to 2018. From left to right: GAN [43], DCGAN [105], CoupledGAN [85], PGGAN [68], StyleGAN [69].

al. [15] propose two regularizers to make the learning more stable. If there is no overlap between the distribution of generated data and real data, or if the overlap is negligible, the divergence will be set to a constant. At this time, the gradient is zero, which will cause the vanishing gradient problem. In order to address this problem, Arjovsky *et al.* [5] propose Wasserstein Generative Adversarial Networks (WGAN). They theoretically show that the Earth mover's distance produces better gradient behaviors in distribution learning, compared to other distance metrics. This approach provides a weight-clipping method to enforce the Lipschitz constraint and finds a novel loss metric to address the problem of unstable training processes. Due to the use of weight clipping in the discriminator, Gulrajani *et al.* [46] find that the WGAN might still have unsatisfactory results or could not converge. Hence, they propose a gradient penalty named WGAN-GP to enforce the Lipschitz constraint. Their method also has a better performance than the original WGAN, and it enables training of various GAN architectures, more stably than before with almost no hyper-parameter tuning. Finally, Petzka *et al.* [101] propose a new penalty term, known as WGAN-LP, to enforce the Lipschitz constraint. This method further improves the stability of network training.

Generating new plausible samples was the application described in the original GAN paper [43], where GANs were used to generate new plausible examples for the MNIST handwritten digit dataset, the CIFAR-10 small object photograph dataset, and the Toronto Face Database. Although the plain GANs show some difficulties in generating high-resolution images, Radford *et al.* [105] first decompose an image into a Laplacian pyramid, then they use multiple GANs to generate the details at different levels of the pyramid. The algorithm, known as DCGAN, uses the concatenation of multiple convolutions to generate, in one shot, high-resolution images. DCGAN also demonstrates the ability to perform vector arithmetic with the input to the GANs (in the latent space)with both generated bedrooms and generated faces. Later, different algorithms were proposed to improve the quality of the generated images. Figure 2.7 shows an example of the rapid progress of GANs.

As GANs are able to produce realistic textures [66, 79, 106, 137, 152], they are also used in

image restoration tasks to push the network to reconstruct high-quality details [79, 109, 128]. We will discuss in more details in Section 2.2.

For deep learning applications, the massive data development (*e.g.*, collecting, labeling), which is an essential process in building practical applications, still incurs seriously high costs. One of the most common techniques for alleviating the costs of labeled data is data augmentation. Based on the fact that GANs are able to generate various and realistic data-samples by learning data distributions and that they can generate unseen samples from the learned distributions, several methods have been presented to augment data by applying GANs. These methods [7, 18] employ the ability of GANs and use the generated samples for additional input for the target task. Calimeri *et al.* [12] propose to simply apply generated samples as additional data in medical imaging tasks. Zhu *et al.* [153] have shown an application by using conditional GANs to augment plant images. For re-identification tasks in computer vision, the study of [147] presents a training method with unconditional generated samples. Tran *et al.* [124] proposes a way to train classification models with GANs in semi-supervised fashion. These works show the power of GANs in generating high-quality samples; and they show that the networks trained on the extended dataset obtain higher accuracy. We also use this technique,in Chapter 4, to resolve the problem of a limited training dataset.

2.2 Image Restoration

Image restoration techniques are used to improve the appearance of an image by applying a restoration process that uses an underlying mathematical model describing image degradation. The types of degradation include blur, noise, and low-resolution(LR) Generally, it is assumed that the degradation model is known or can be estimated. Techniques used for image restoration are oriented toward modeling the degradations, and toward applying inverse procedures to obtain an approximation of the original scene. A general block diagram of image restoration is shown in Figure 2.8. The information about the acquisition process is fed as inputs for the improvement of the degradation model. To obtain the restored image, which presents an estimate of the original image, the inverse degradation process is then performed to the degraded image. The information obtained by analyzing the difference between the restored image and the original image is used to further refine the degradation model.

The typical image restoration tasks in this thesis include SR, denoising, deblurring, and demoisaicing, with examples shown in Figure 2.9. In this section, we first introduce the inverse problem that underlies image restoration. We then present the SR methods and the other image restoration tasks that are involved in real-world SR. Lastly, we review the datasets that are used to evaluate image restoration tasks.

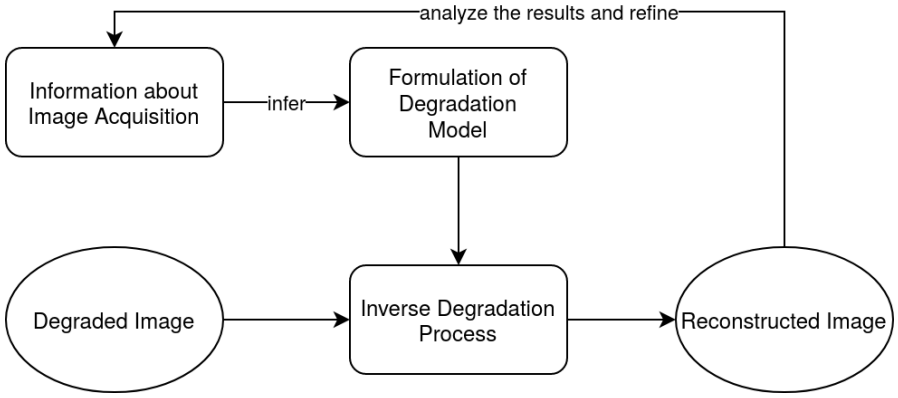


Figure 2.8 – Block diagram for image restoration.

2.2.1 Inverse Problems

Solving an inverse problem is a process that reconstructs unobserved variables from a set of observed data. Inverse problem methods are widely used in signal processing, medical imaging, computer vision, and especially in image restoration. For example, HR images provide rich textures that are pleasant to view on high-definition monitors, but HR images are hard to acquire due to the hardware limitations. Image SR is an inverse problem that generates a HR image from an observed LR image. Another example of an inverse problem is image denoising that generates a clean image from a noisy image.

Mathematically, in an inverse problem, an unknown variable $y \in \mathbf{Y} \rightarrow \mathbb{R}^{dim_1}$ is reconstructed from an observed variable $x \in \mathbf{X} \rightarrow \mathbb{R}^{dim_2}$, where dim_1 and dim_2 are dimensions of space $\mathbf{Y}$ and $\mathbf{X}$, respectively. It is usually modeled as a linear system:

$$x = Ay + \epsilon \tag{2.4}$$

where $A : \mathbf{Y} \rightarrow \mathbf{X}$ is a linear projection model and ϵ is a random noise variable. A is a human-designed projection matrix, or a learned function from a training data set.

Inverse problems have two difficulty categories: well-posed and ill-posed. A well-posed inverse problem has two features: (1) the number of unknowns y is smaller than the number of observations x; (2) the condition number of A is small. A well-posed problem has a unique and stable solution. A simple method, such as the ordinary least squares (OLS), can generate a good result. Whereas, in an ill-posed inverse problem, the number of unknowns is larger than that of observations, or the condition number of A is large, which causes numerical instabilities or overfitting problems. In practice, many important inverse problems are ill-posed. For example, single image SR, addressed in this thesis, is an ill-posed inverse problem.

Regarding model complexity, there are different kinds of inverse problem methods. In this thesis, we focus on the deep learning methods. In particular, to use deep learning methods to resolve the inverse problem of estimating $y \in \mathbf{Y}$ from an observation $x \in \mathbf{X}$, where the pair

Figure 2.9 – Image restoration tasks included in this thesis.

$s = (x, y)$ is drawn from the sample space $\mathbf{D} = \mathbf{X} \times \mathbf{Y}$ according to some unknown distribution μ. With the access to a set of training samples $\mathbf{S} = (x_i, y_i)$, a regressor can be learned

$$R(\cdot) : \mathbf{Y} \rightarrow \mathbf{X} \tag{2.5}$$

that can be used to deliver an estimate of $x \in \mathbf{X}$ given $y \in \mathbf{Y}$. Here, the regressor is a CNN that will be learned from the given dataset. Different architectures have been designed for different tasks, and they are reviewed in the following sections.

2.2.2 Image Super-Resolution

Single image super-resolution (SISR) refers to the task of restoring HR images from the LR observation of the same scene (an example is shown in Figure 2.9). It is a notoriously challenging ill-posed problem because a specific LR input can correspond to a number of possible HR images, and because the HR space (in most instances, it refers to the natural image space) that we intend to map the LR input to is usually intractable. Previous non-deep learning SR methods have two main drawbacks: One is the unclear definition of the mapping that develop between the LR space and the HR space, and the other is the inefficiency of establishing a complex high-dimensional mapping, given the massive data. Benefiting from the strong

capacity of extracting effective high-level abstractions that bridge the LR and HR space, recent CNN-based SR methods have achieved significant improvements, both quantitatively and qualitatively.

In this section, we give an overview of recent CNN-based SR algorithms. We focus on the efficient neural network architectures designed for SR and effective optimization objectives for CNN-based SR learning. From the perspective of deep learning, although many other techniques such as data preprocessing and model training techniques are also quite important, the combination of deep learning and domain knowledge in SR is usually the key to success and is often reflected in the innovations of neural network architectures and optimization objectives for SR.

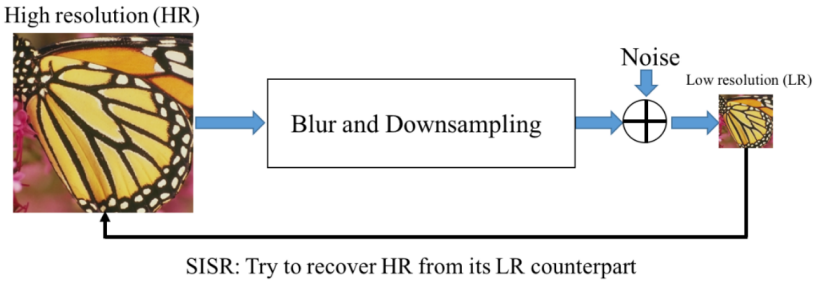


Figure 2.10 – Sketch of the overall framework of SR. Figure taken from [134].

In the typical SR framework, as depicted in Figure 2.10, the LR image I^{LR} is modeled as follows:

$$\mathrm{I}^{LR} = (\mathrm{I}^{HR} \otimes k) \downarrow_s + n \tag{2.6}$$

where $\mathrm{I}^{HR} \otimes k$ is the convolution between the blur kernel k and the unknown HR image I^{HR}, $\downarrow_s$ is the downsampling operator with scale factor s, and n is the independent noise term. In image SR, the goal is to minimize the data fidelity term associated with the model presented in Equation 2.6.

Network Architectures for Super-Resolution

The early SR networks proposed an early upscaling design, where the LR inputs are first upscaled to match the desired HR output size before feeding them into the network. A super-resolution convolutional neural network, abbreviated as SRCNN [24] is the first successful attempt towards using only convolutional layers for SR. This effort can rightfully be considered as the pioneering work in deep learning based SR that inspired several later attempts in this direction. The SRCNN structure is straightforward, it consists of only convolutional layers where each layer (except the last one) is followed by a rectified linear unit (ReLU) [97] non-linearity. There are a total of three convolutional and two ReLU layers, all stacked together linearly. Although the layers are the same (*i.e.*, convolution layers), the authors named the layers according to their functionality. The first convolutional layer is termed patch extraction

or feature extraction: it creates the feature maps from the input images. The second convolutional layer is called non-linear mapping: it converts the feature maps onto high-dimensional feature vectors. The last convolutional layer aggregates the feature maps to output the final high-resolution image. The structure of SRCNN is shown in Figure 2.11.

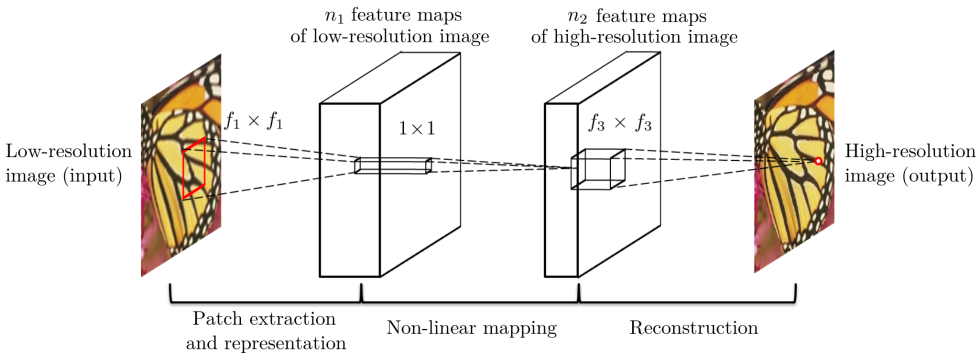


Figure 2.11 – Architecture for the first SR network: SRCNN [24].

Following SRCNN, many SR networks were proposed to improve SR. VDSR [71] is a 20-layer VGG-based [115] SR network. In addition to the architecture, VDSR is the first SR network that uses it for multiple scales. VDSR also proposes a residual learning strategy; it uses deep CNN to learn the mapping from the input to the residual between the input LR and HR. The authors argue that residual learning can improve performance and accelerate convergence. EDSR [84], proposed by Lim *et al.*, utilize a modified residual block from ResNet [59], and achieve the first place in NTIRE2017 SR challenge [122]. Except for the increase of regular depth, to achieve better performance EDSR also increases the number of output features of each layer on a large scale. Some SR networks also incorporate with frequency domain [150, 34] to improve the performance. Recently, Zhang *et al.* [142] proposed RDN, it uses a structure similar to DenseNet [62] and achieves the state-of-the-art performance. In an RDN block, basic convolution units are densely connected similar to DenseNet. At the end of an RDN block, a bottleneck layer is used, followed by residual learning across the whole block. Before entering the reconstruction part, features from all previous blocks are fused by the dense connection and residual learning.

The input to early SR networks is the bicubic-upscaled LR that serves as the approximation of HR. However, these interpolated inputs have several drawbacks: (1) detail-smoothing effects introduced by these inputs could lead to further wrong estimations of the image structure, (2) employing interpolated versions as input is very memory-consuming, and (3) when the downsampling kernel is unknown, one specific interpolated input as a raw estimation is unreasonable. To solve these problems, Shi *et al.* [112] proposed an efficient subpixel convolution layer, namely Pixelshuffle layer, in ESPCN [112]. The structure of ESPCN is shown in Figure 2.12. Rather than increasing the resolution by explicitly enlarging feature maps as the deconvolution layer does, ESPCN expands the channels of the output features for storing the extra points to increase resolution then rearranges these points to obtain the HR

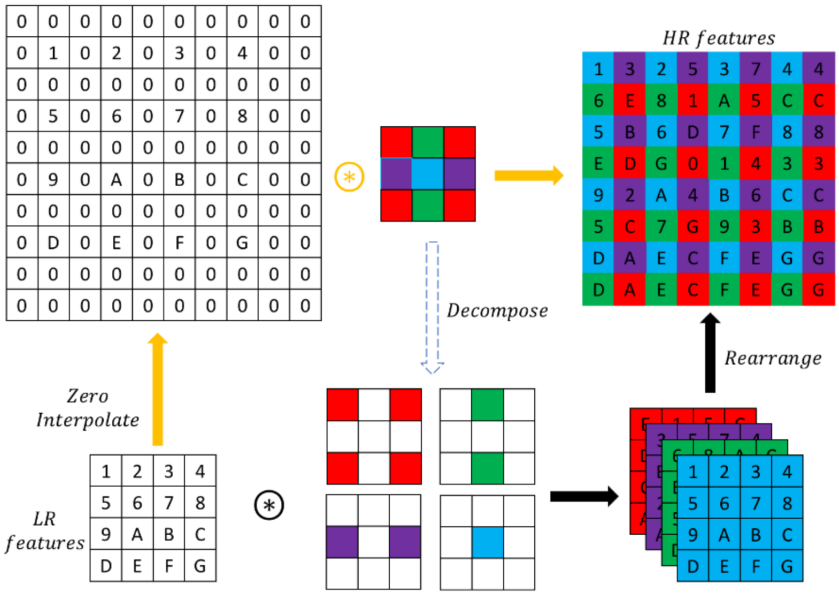


Figure 2.12 – Detailed sketch of ESPCN [112]. Figure taken from [134].

output through a specific mapping criterion. As the expansion is carried out in the channel dimension, a smaller kernel size is sufficient.

Loss Functions for Super-Resolution Networks

As peak signal-to-noise ratio (PSNR) is a widely used metric for quantitatively evaluating image restoration quality, existing SR approaches predominantly use pixel-level error measures, *e.g.*, l_1 and l_2 distances or a combinations of both. As these measures encapsulate only local pixel-level information, the resulting images do not always provide perceptually sound results. For example, it has been shown that images with high PSNR and SSIM values give overly smooth images with low perceptual quality. To counter this issue, several perceptual loss measures are proposed in the literature. Johnson *et al.* [67] use the feature maps extracted from VGG net [115] to enhance the visual quality by minimizing the error in a feature space instead of pixel space. Contextual loss [93] is developed to generate images with natural image statistics by using an objective that focuses on the feature distribution rather than merely comparing the appearance. Ledig *et al.* [79] propose the SRGAN model that uses perceptual loss and adversarial loss to favor outputs that reside on the manifold of natural images. Sajjadi *et al.* [109] develop a similar approach and further explore the local texture matching loss. Using these works, Wang *et al.* [128] propose ESRGAN; it won first place in PIRM2018 SR competition.

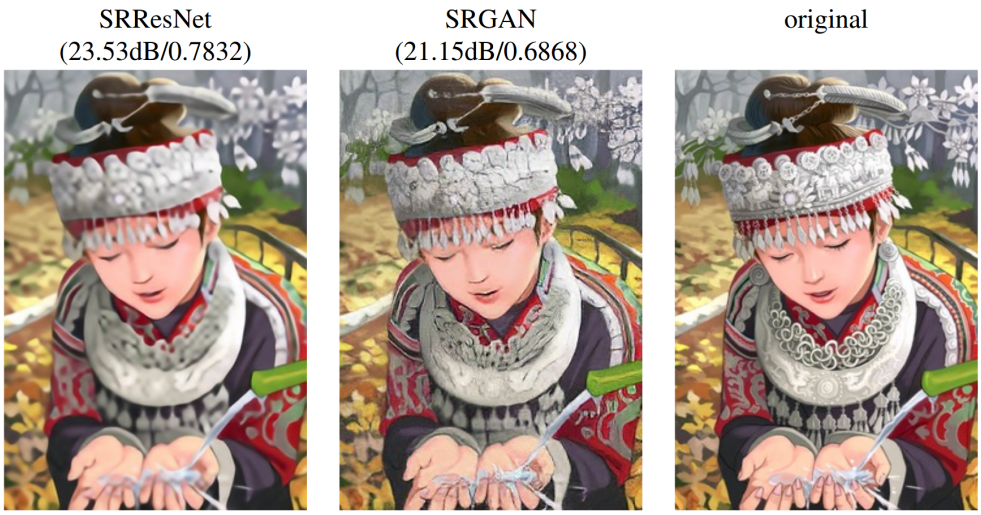
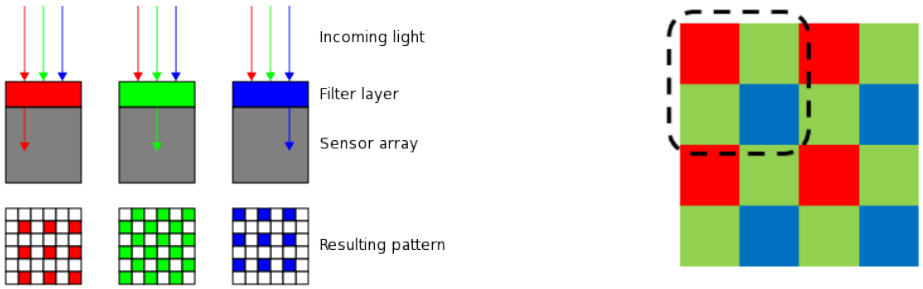


Figure 2.13 – Alghouth the SR result from SRGAN [79] obtains lower PSNR and SSIM than the same network trained without adversarial loss, the image quality is much higher. Figure taken from [79].

2.2.3 Raw Image Processing

To apply SR on real photographs, it is important to understand how images are produced from the camera sensor. Most camera sensors capture only red, green, or blue information at each pixel. This is achieved by placing a color-filter array (CFA) in front of the CMOS, which is illustrated in Figure 2.14a. The Bayer pattern, as shown in Figure 2.14b is a very common example of such a CFA. The resulting image, without any post-processing, is called a raw image. The process for recovering the full-color image from the incomplete raw image is called demosaicing and is the crucial first step of most digital camera pipelines. An example of a Bayer pattern raw image and demoisaicing is shown in Figure 2.9.



(a) Illustration of CFA. (b) Bayer pattern

Figure 2.14 – Illustration of CFA and Bayer pattern.

Many demosaicing algorithms have been proposed. To fill the missing pixels, early approaches use different interpolations for luminance and chrominance in spatial domain or the frequency domain. Advanced works tend to build on the underlying image statistics. They rely on

techniques ranging from SVM regression to shallow neural network architectures. They outperform the traditional methods and give state-of-the-art results. Heide *et al.* [60] formulate demosaicing as an image reconstruction problem and, to achieve natural results, they embed a non-local natural image prior to an optimization approach, called FlexISP . To efficiently learn a suitable regularization term from training data, Klatzer *et al.* [74] build a variational energy minimization framework SEM, thus yielding high-quality results in the presence of noise. More recently, to improve the quality of demosaicing by training on a large dataset, Gharbi *et al.* [42] proposed a deep learning-based demosaicing method.

Although demoisaicing is usually applied before other image restoration tasks, prior works have shown that using raw sensor data is able to enhance other image processing tasks. Farsiua *et al.* [37] propose a maximum a posteriori technique for joint multi-frame demosaicing and SR estimation with raw sensor data. Gharbi *et al.* [42] train a deep neural network for joint demosaicing and denoising. These methods use synthetic Bayer mosaics. Similarly, Mildenhall *et al.* [96] synthesize raw burst sequences for denoising. Chen [16] present a learning-based image processing pipeline for extreme low-light photography by using raw sensor data. DeepISP is an end-to-end deep learning model that enhances the traditional camera image signal processing pipeline [111]. Similarly, we operate on raw sensor data and propose a method to super-resolve raw images in Chapter 3.

2.2.4 Image Deblurring

As presented in the imaging model in Equation 2.6, the blur kernel k is an important parameter in SR. In recent years, we have witnessed significant advances in single-image deblurring, as well as blur kernel estimation. Efficient methods based on Maximum A Posteriori (MAP) formulations were developed with different likelihood functions and image priors [11]. In particular, heuristic edge-selection methods for kernel estimation [19] were proposed for the MAP estimation framework. Multi-spectral information can also be used to improve the deblurring results [32, 31]. To better recover the blur kernel and better reconstruct sharp edges for image deblurring, some exemplar-based methods [52] exploit the information contained in both the blurred input and example images from an external dataset. More recently, the dark-channel prior [57] was used by Pan *et al.* [99] to simply and efficiently estimate the blur kernel of natural images. As they achieve significant performance on deblurring tasks [78], in Chapter 4, we adopt their kernel estimation algorithm for collecting blur kernels of real images.

2.2.5 Image Denoisnig

For various reasons, such as photon absorption statistics, sensor architecture, and light levels, images are inevitably contaminated with noise during acquisition. This leads to distortions and loss of image information. SR algorithms are adversely affected by the presence of noise. Therefore, image denoising plays an important role in image restoration.

Image denoising removes noise from a noisy image, in order to restore the clean image. An example is in Figure 2.9. Mathematically, the problem of image denoising can be modeled as follows:

$$\mathrm{I}^{noisy} = \mathrm{I}^{clean} + n \tag{2.7}$$

where I^{noisy} is the observed noisy image, I^{clean} is the unknown clean image, and n represents the noise, which is usually an additive white Gaussian noise (AWGN) with standard deviation σ_n.

Image denoising is also an ill-posed problem and has been studied extensively in the past decades. In recent years, CNN-based denoising methods have been proposed [138, 120, 4]. Compared to conventional methods, such as BM3D [21], the denoising performance of these methods has greatly improved. Zhang *et al.* [138] propose DnCNN that uses batch normalization and ResNet [59] architecture to perform image denoising. This network not only deals with blind image denoising, but also addresses image SR and JPEG image deblocking. Its architecture is shown in Figure 2.15. Specifically, it obtains the residual image from the model instead of the denoised image. FFDNet [139] uses a noise level map and a noisy image as input to deal with different noise levels. This method exploits a single model to deal with multiple noise levels. Anwar *et al.* [4] proposes a single-stage, blind real image denoising network (RIDNet) by employing a modular architecture. A residual in the residual structure is used to ease the flow of low-frequency information and, to exploit the channel dependencies, it applies feature attention . The architecture of RIDNet is shown in Figure 2.16. To address the unseen noise levels, El Helou *et al.* [33] propose an optimal denoising modal that learns the SNR function for an optimal fusion of the noisy image with the learned prior to improving the generalization of the denoising modals.

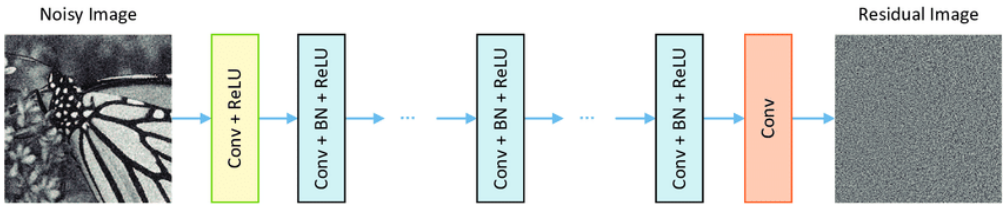


Figure 2.15 – Architecture of DnCNN [138].

2.2.6 Multi-modal Image Processing

Multi-modal image processing has been attracting increasing interest from the computer vision community and is used in a variety of of intriguing applications, *e.g.*, image denoising, image fusion, image style transfer, and guided depth SR, as shown in Figure 2.17. Given an image I^x and another image I^y, multi-modal image processing recovers a better version of I^x, with the guidance of I^y, or fuses I^x and I^y to a new image I^z, which has the advantage of both I^x and I^y. Different from uni-modal image processing tasks, *e.g.*, single image SR, multi-modal image processing usually requires proper modeling of the dependencies, across

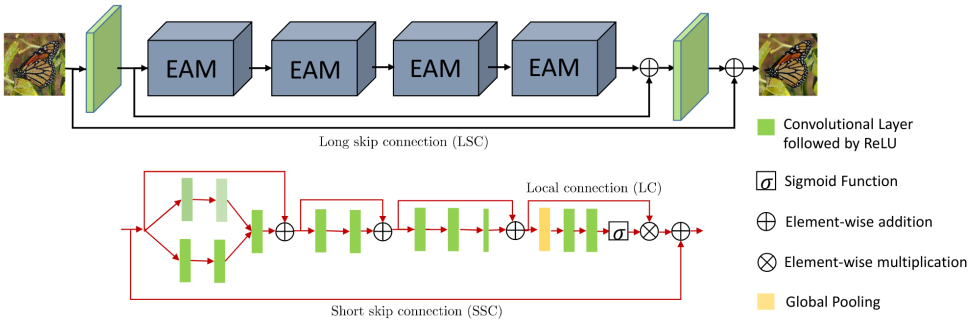


Figure 2.16 – Architecture of RIDNet [4].

different modalities.

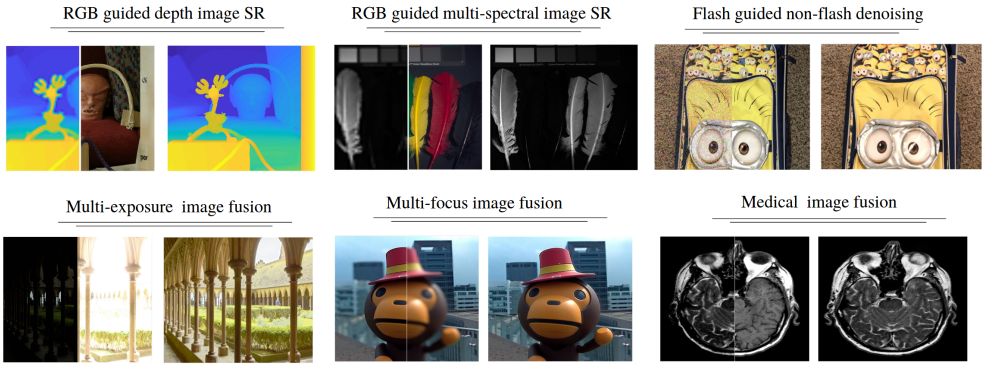


Figure 2.17 – Examples of different multi-modal image processing tasks.

An example of multi-modal image processing is spectral image SR. Due to hardware limitations, high spectral resolution images come at the cost of lower spatial resolution. To mitigate this problem, they are often combined with higher spatial-resolution yet lower spectral-resolution images. Previous works [129, 130, 135] used statistical methods to mix spatial information from the high spatial-resolution low spectral-resolution image with the color information from the multi-spectral bands. However, it is expensive and time-consuming to generate a large set of registered spectral and color images. To cope with the limited training data, a model can be trained on a large but related dataset then adapted to perform on the smaller given dataset. Prior work on domain adaptation [22, 41, 48] show the merit of these techniques in handling small or difficult-to-label datasets. Similarly, in Chapter 6, we use our original framework for super-resolving the multi-spectral images, then we use a small residual network to refine the result through a color image guide.

2.2.7 Image Quality

Full Reference Image Quality Assessment

There is a large interest in the automatic assessment of the image quality, and numerous measures have been proposed. When a ground-truth image I^G with N pixels is available, the quality of a corresponding (degraded or restored) image I can be defined as the pixel-level fidelity to the ground-truth. Representatives are

- **Mean Square Error (MSE)** defined by $\mathrm{MSE} = \frac{1}{\mathrm{N}} \sum_{i=1}^{N} (\mathrm{I}_i^G - \mathrm{I}_i)^2$, where I_i^G (or I_i) is the i-th pixel of I^G (or I).

- **Peak Signal-to-Noise Ratio (PSNR)** defined by $\mathrm{PSNR} = 10\log 10 \frac{\mathrm{MAX}_{\mathrm{I}^G}^2}{\mathrm{MSE}}$ where $\mathrm{MAX}_{\mathrm{I}^G}^2$ is the maximum possible pixel value of the image, which is usually 255.

Small shifts in the content of I, however, leads to poor MSE and PSNR scores – even when the contents are identical. Therefore, another group of measures counts for such structural similarity. The **Structural Similarity Index (SSIM)** is a perception-based model that considers image degradation as a perceived change in structural information. PSNR and SSIM are the common full-reference image quality metrics that are used for evaluating the quality of image reconstruction results; thea are also used to evaluate different methods in this thesis. Note, however, that their usability in evaluating perceived quality is modest.

2.3 Image Datasets

Large datasets are the basis for the deep learning methods. We introduce the datasets that we use in the thesis in the following.

RAISE Dataset

Dang-Nguyen *et al.* [23] build up the RAISE dataset that contains 8,156 HR images with their raw version that is uncompressed and guaranteed to be camera-native (*i.e.*, never touched or processed). The images are collected, from 2011 to 2014, by four photographers who used three different camera models. The dataset contains different scenes and moments in over 80 places in Europe. All images are provided with tags: "outdoor", "indoor", "landscape", "nature", "people", "objects" and "buildings". Metadata is also provided, together with the embedded annotations. Figure 2.18 shows some sample images from the RAISE dataset. As the RAISE dataset provides untouched raw images, we use it in Chapter 4 for joint demoisaicing and SR.

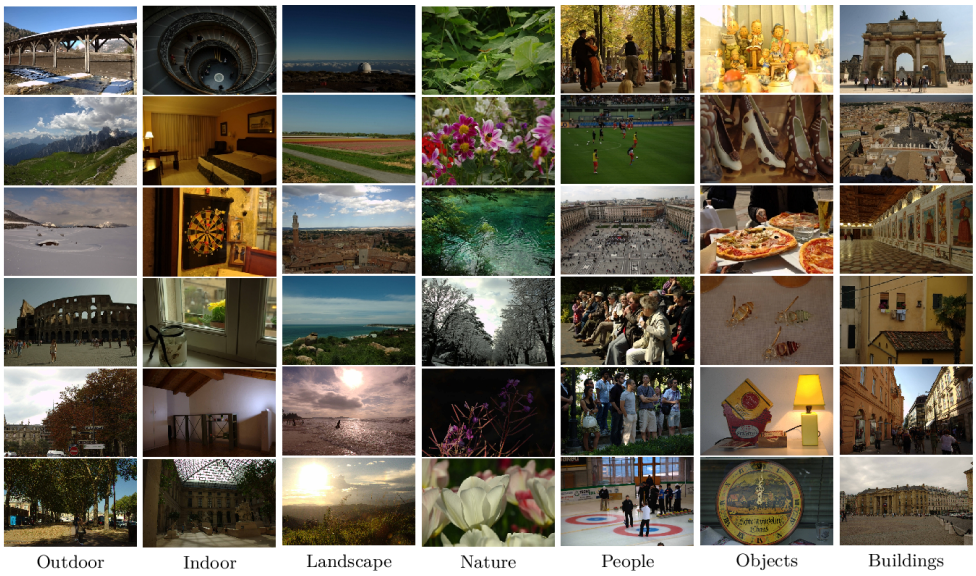


Figure 2.18 – Examples of images in RAISE dataset [23].

DIV2K Dataset

To provide a large dataset for example-based single image SR and to study the state-of-the-art, as emerged from the NTIRE 2017 challenge [122], Timofte *et al.* [122] collected the DIVerse 2K-resolution image dataset, namely DIV2K. The dataset contains 1,000 color RGB images from the Internet; they have a large diversity of sources (sites and cameras). All 1,000 images are 2K resolution, *i.e.*, they have 2K pixels on at least one of the axes (vertical or horizontal). The images are of high quality, both aesthetically and in terms of only containing small amounts of noise and other corruptions such as blur and color shifts. DIV2K has a large diversity of content, ranging from people, handmade objects, and environments (cities, villages), to flora and fauna, and natural sceneries, including underwater and dim lighting conditions. Examples of images in DIV2K are shown in Figure 2.19. As DIV2K serves as the state-of-the-art benchmark for SR, in Chapter 4, we use DIV2K to evaluate our SR network.

DPED Dataset

The DPED dataset (DSLR Photo Enhancement Dataset) is collected by Ignatov *et al.* [64] to solve the problem of image translation from poor quality images captured by smartphone cameras to superior quality images achieved via a professional DSLR camera. DPED consists of photos taken in the wild, synchronously by three smartphones (iPhone 3GS, Blackberry Passport, Sony XperiaZ) and one DSLR camera (Canon 70D DSLR). Example quadruplets can be seen in Figure 2.20. In total, DPED consists of 4549 photos from a Sony smartphone, 5727 from an iPhone, and 6015 photos in total from a Canon and a BlackBerry camera. The photos were taken during the daytime in a wide variety of places and in various illumination and weather conditions. To address the problem of misalignment, an overlapping region is

Figure 2.19 – Examples of images in DIV2K dataset [122].

determined by SIFT descriptor matching, followed by a non-linear transform and a crop, thus resulting in two images of the same resolution representing the same scene. As DPED dataset provides real photographs taken from low-end cameras, in Chapter 4, we use it for references of real-world LR images.

StereoMSI Dataset

The StereoMSI dataset [114] was first published and promoted during the PIRM2018 Spectral Image SR Challenge [113]. This dataset contains 350 registered color-spectral image pairs and benchmarks example-based spectral image SR. The color images are captured with an RGB XiQ camera model MQ022CG-CM and the spectral images are captured with a XiQ multi-spectral camera model MQ022HG-IM-SM4x4 covering wavelength from $470nm$ to $620nm$. The RGB camera contains a 2×2 Bayer RGGB pattern ; this is half of the size of the IMEC spectral sensor's pattern that has a 4×4 pattern that delivers 16 wavelength bands. As a result, the resolution of the RGB images in each axis is twice that of the spectral images. We show an example image set in Figure 2.21. In Chapter 6, we use StereoMSI dataset for evaluating our spectral image SR.

Figure 2.20 – Example quadruplets of images taken synchronously with DPED's [64] four cameras.

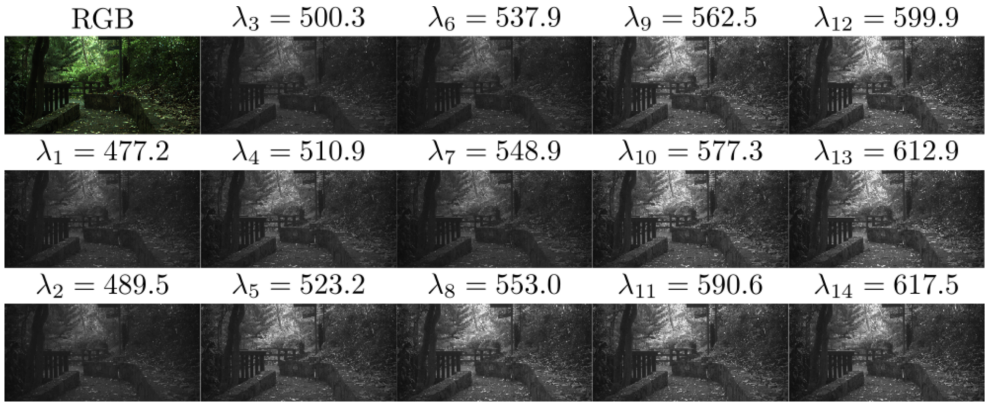


Figure 2.21 – A sample image from the StereoMSI dataset [114]. The dataset contains RGB images with 14 wavelengths channels multi-spectral images.

3 Super-Resolution with Raw Images

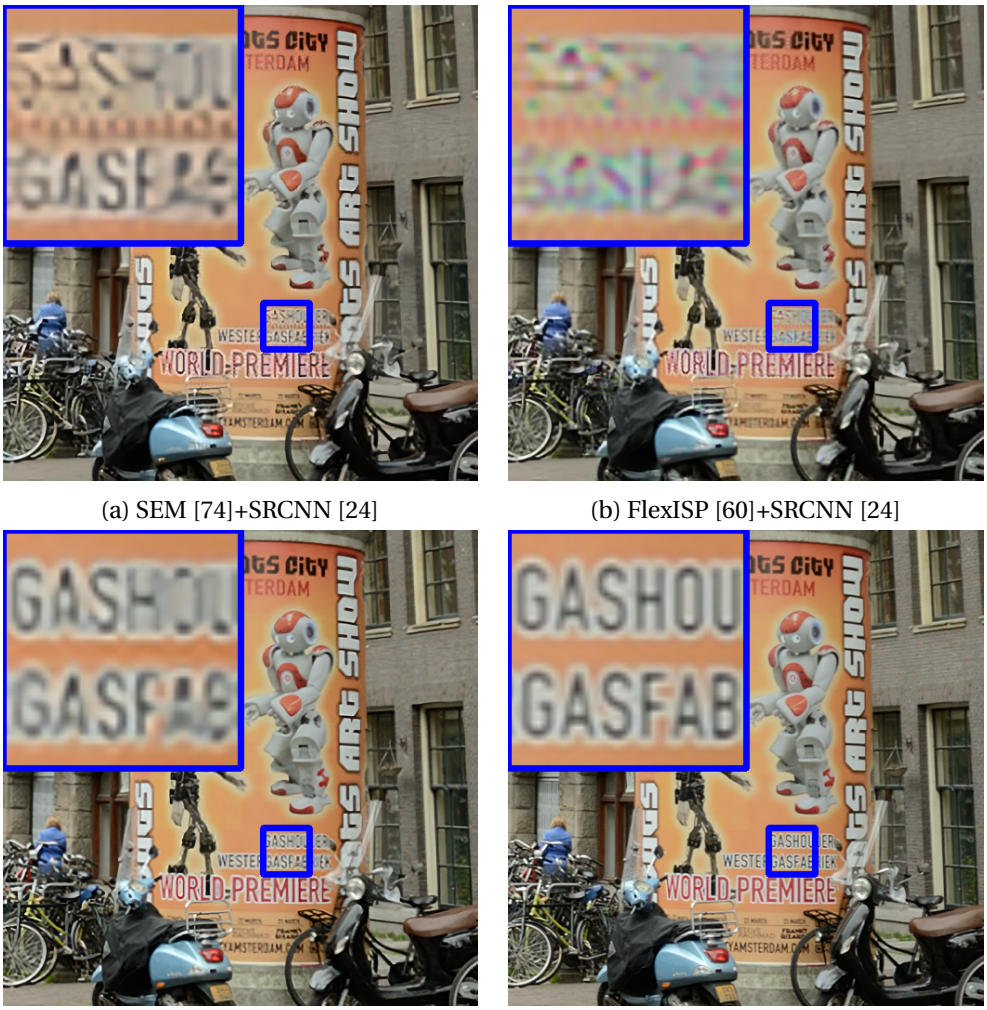
3.1 Introduction

There is an ever growing interest in capturing HR images that is in step with the increasing quality of camera sensors and display devices. Ironically, the most prevalent image capture devices are mobile phones, which are equipped with small lenses and compact sensors. Despite the large advancements made in improving the dynamic range and resolution of images captured by mobile devices, the inherent design choices still limit the ability to capture very high-quality images.

The limitations come from two design issues. Firstly, the single CMOS sensor in most cameras, including mobile cameras, measures at each spatial location only a limited range of wavelengths (red, green or blue) of the electromagnetic radiation instead of the full visible spectrum. This is achieved by placing a color filter array (CFA) in front of the sensor, as introduced in Section 2.2.3. The most common type of CFA is the Bayer pattern, which captures an image mosaic with twice green for each red and blue waveband. Raw images are direct readings from image sensors with these CFAs. These digital signals are further post-processed to obtain RGB images through demosaicing that contain color information at each spatial location.

Secondly, as the sensor needs to be compact to fit into the device, resolution is limited by the size of the photon wells. 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. To reconstruct full color from the CFA mosaiced image, demosaicing algorithms are applied, while LR can only be dealt with using SR algorithms in a post-processing step.

In the last few decades, demosaicing and SR have been independently studied and applied in sequential steps. However, the separate application of demosaicing and SR is sub-optimal and usually leads to error accumulation, as shown in Figure 3.1. This is because artifacts such as color zippering introduced by demosaicing algorithms is treated as a valid signal of the input image by the SR algorithms. As a result, the application of SR algorithms after demosaicing algorithms may lead to visually disturbing artifacts in the final output.



(a) SEM [74]+SRCNN [24]

(b) FlexISP [60]+SRCNN [24]

(c) Our SR on raw image

(d) Reference image

Figure 3.1 – Comparison of our SR result on raw image to the combinitions of state-of-the-art demosaicing and SR methods. Note how the sequential application of demoisacing and SR carries forward color artifacts or blurring (first row). Our method exhibits none of these artifacts and is able to faithfully reconstruct the original.

The algorithms for demosaicing and SR are meant to overcome the sampling limitations of digital cameras. It is reasonable to address them in a unified context. With the advent of deep learning, there are several methods for SR [24, 25, 71, 72, 84, 112] that successfully outperform traditional SR methods [29, 35, 40, 116, 118, 123, 133]. Only recently, deep learning has also been used successfully for image demosaicing [42]. When using deep learning, it is possible to address demosaicing and SR simultaneously, as we show in this chapter.

We train a CNN for SR on raw images. Our model is trained to learn an end-to-end mapping between raw images and HR color images to perform joint demoisaicing and SR. The model is trained on the RAISE dataset [23] that contains more than 8,000 HR raw images with

their post-processed version, as introduced in Chapter 2. The proposed model has several appealing properties. First, its structure is intentionally designed with simplicity in mind, and yet provides superior accuracy compared to the sequential application of state-of-the-art demoisaicing and SR methods. Figure 3.1 shows a comparison on an example. Second, with moderate numbers of filters and layers, our method achieves fast speed for practical online usage. Our method is faster than a series of optimization-based methods, because it is fully feed-forward and does not need to solve any iterative optimization. Third, the restoration quality of the network argues well for the use of our approach in camera image processing pipelines. For mobile devices this can encourage the use of sensors with large pixels that capture a better dynamic range, rather than sacrificing dynamic range for higher resolution as is done at the moment.

To summarize, our contributions in this chapter are[1]:

- We propose to use a deep residual network for end-to-end joint demosaicing and SR. More specifically, our network can learn an end-to-end mapping between raw images and HR color images.
- We present that this network generalizes to other color filter arrays (CFA) with a simple modification of two layers of the network.
- We demonstrate both quantitatively and qualitatively that our approach generates higher quality results than state-of-the-art methods. In addition, our method is computationally more efficient because of the joint operation. Our approach can be extended to videos, and can potentially be integrated into the imaging pipeline.
- We demonstrate that deep learning is useful in the inverse problem of real scene SR, and can achieve good quality and speed and be applied in camera image processing pipelines.

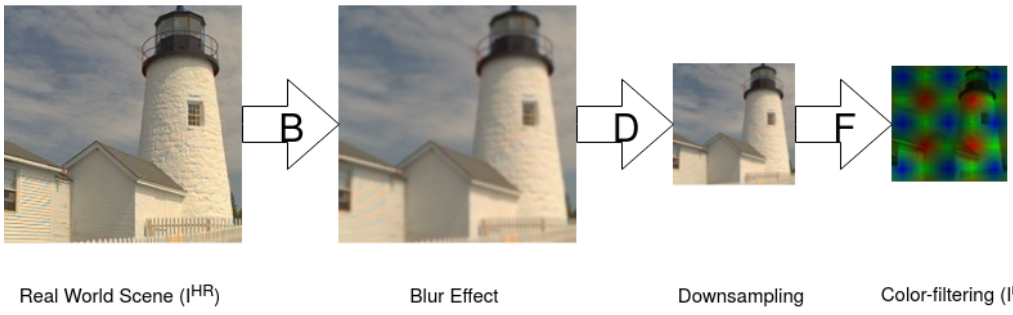


Figure 3.2 – Block diagram presenting the assumed image formation in our model. Where I^{HR} is the intensity distribution of the real scene, **B**, **D**, **F** present the blurring, downsampling and mosaicing process, I^{raw} is the observed Bayer image.

[1] This work was published in [148].

3.2 Method

3.2.1 Imaging Model for Joint Demoisaicing and Super-Resolution

Figure 3.2 shows a typical observation model relating the HR real scene with LR raw image as introduced in the literature. In this formation model, the real world scene I^{HR} is smoothed by a blur kernel which represents the point spread function of the camera, then it is downsampled by a factor of s and mosaiced by the CFA of the CMOS to get the observed raw image I^{raw}. These downsampled images might be further affected by sensor noise and color filtering noise, however, we ignore the noise factor in this chapter and will discuss it in Chapter 5. Thus, the raw images captured by the LR imaging system is the blurred, decimated, and mosaiced version of the underlying true scene.

Let I^{HR} denote the HR color image desired, and I^{raw} be its LR raw observation from the camera. The LR raw observation is related with the HR scene I^{HR} by

$$\mathrm{I}^{raw} = \mathbf{F}(\mathbf{D}(\mathbf{B}(\mathrm{I}^{HR}))), \tag{3.1}$$

where **B** models the blurring effects, **D** is the downsampling operator, and **F** is the mosaicing procedure. The objective of joint demoisaicing and SR is to provide an approximate inverse operation estimating an HR image $\mathrm{I}^{SR} \approx \mathrm{I}^{HR}$ given an LR raw image I^{raw}. In general, I^{raw} is a real-valued tensor of size $h \times w \times 1$, while I^{HR} is a tensor of $s \cdot h \times s \cdot w \times 3$. This problem is ill-posed as the downsampling and mosaicing are non-invertible.

3.2.2 Deep Residual Network for Joint Demoisaicing and Super-Resolution

To solve the problem described in the previous section, traditional methods usually design nonlinear filters that incorporate prior heuristics about inter- and intra-channel correlation. A deep CNN is a better substitute for such methods, as convolutional layers can automatically learn to exploit inter- and intra-channel correlation through a large dataset of training images. Moreover, the exclusive use of a set of convolutional layers enables joint optimization of all the parameters to minimize a single objective as is the case in joint demosaicing and SR.

We thus build our framework in a data-driven fashion: we create the training set from a large set of high-quality images I^{HR}, and produce the input measurements I^{raw} using the same process as the image formation model illustrated in Figure 3.2, then we train our deep convolutional network on this dataset.

The objective of the network is to learn a mapping from I^{raw} to I^{HR} which conceptually consists of three operations:

- **Color extraction**: this operation separates the color pixels into different channels from the mono-channel raw image. With this operation, no hand-crafted rearrangement of the raw input is needed unlike in other demosaicing algorithms. This operation gives a

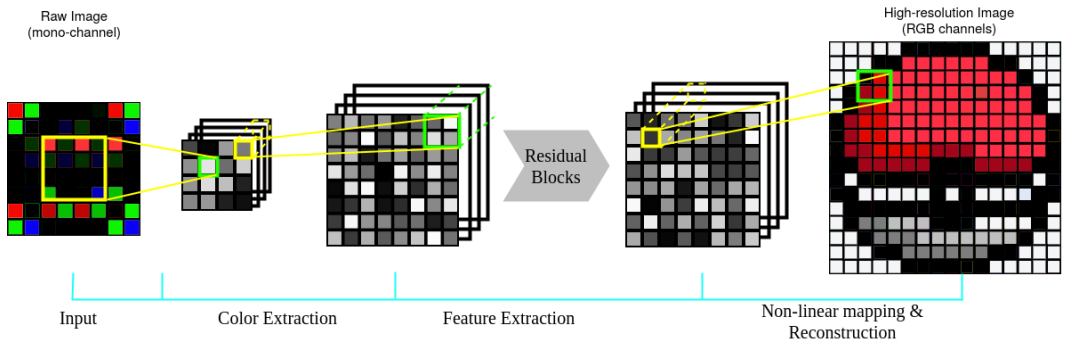


Figure 3.3 – Illustration of our proposed network architecture. The network is a feed-forward fully-convolutional network that maps an LR raw image to an HR color image. Conceptually the network has three components: color extraction of raw image, non-linear mapping from raw image representation to color image representation with feature extraction, and HR color image reconstruction.

set of color features from the raw input.

- **Feature extraction and non-linear mapping:** following the intuition of building the first deep neural network for SR [24], this operation extracts overlapping patches from the color features to use high-dimensional vectors to represent the raw image in an LR manifold, which is then mapped to the HR manifold.

- **Reconstruction:** this operation aggregates HR representations to generate the final high-resolution color image I^{SR}.

Color Extraction

The raw image is a matrix with the three color channel samples arranged in a regular pattern in a single channel as introduced in Chapter 2. To make the spatial pattern translation-invariant and reduce the computational cost in latter steps, it is essential to separate the colors in the raw image into different channels at the beginning. The Bayer pattern is regular and has a spatial size of 2×2 with RGGB (red-green-green-blue) pattern. Since the neighboring colors may also affect the result, we build our first convolutional layer C_1 with a spatial size of $2 \cdot s$ and a stride of s:

$$\mathrm{I}^1 = C_1(\mathrm{I}^{raw})_{(i,j)} = (W_1 \otimes \mathrm{I}^{raw} + b_1)_{(2 \cdot i, 2 \cdot j)}, \tag{3.2}$$

where I^1 represents the output from the first layer, W_1 and b_1 represent the filters and biases of the first convolutional layer, and $\otimes$ denotes the convolution operation. Here, W_1 corresponds to 256 filters of support $2 \cdot s \times 2 \cdot s$. (i, j) present the pixel location.

We build an efficient Pixelshuffle layer [112] C_2 which has been introduced in Chapter 2 to

upsample the color features back to the original resolution:

$$C_2(\mathrm{I}^1)_{(i,j,k)} = \mathrm{I}^1_{(\lfloor \frac{i}{s} \rfloor, \lfloor \frac{j}{s} \rfloor, \frac{c \cdot mod(j,s)}{s} + \frac{c \cdot mod(i,s)}{s^2} + k)}, \tag{3.3}$$

here, the Pixelshuffle layer is equivalent to a shuffling operation which reshapes a tensor of size $h \times w \times c$ into a tensor of size $s \cdot h \times s \cdot h \times \frac{c}{s^2}$. c is the number of channels of the input tensor. We find that applying this sub-pixel convolutional layer helps reduce checkerboard artifacts in the output.

Due to the linearity of the separation operation, no activation function is utilized in either layer. Note that the color extraction operation can be generalized to other CFAs by modifying s with respect to the spatial size and arrangement of the specific CFA. Thus we have $s = 2$ for all kinds of Bayer CFAs, CYGM CFAs or RGBE CFAs, and $s = 6$ for the X-trans pattern [3].

Feature Extraction and Non-linear Mapping

Inspired by Dong *et al.* [24], to explore relationships within each color channel and between channels, as well as to represent the raw image in a high-resolution manifold, we exploit a group of convolutional layers in this step.

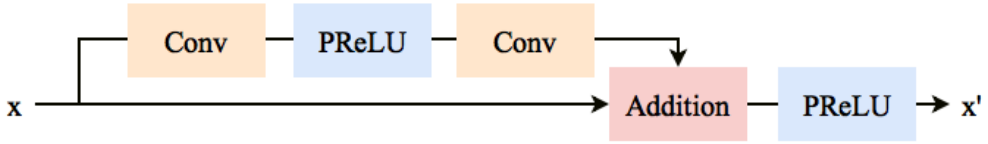


Figure 3.4 – Illustration of the architecture of our residual blocks. We remove the batch normalization layer in the original residual blocks and replace the ReLU with Parametric ReLU. This structure enables faster convergence and better performance.

Previous works [59] have demonstrated that residual networks exhibit excellent performance both in accuracy and training speed in computer vision problems ranging from low-level to high-level tasks. We build a set of 24 residual blocks, each having a similar architecture to Lim *et al.* [84], which is demonstrated in Figure 3.4. We remove the batch normalization layers in the original residual blocks [59] since these layers get rid of range flexibility from networks by normalizing the features [84], because the scale of the features may be useful for image restoration. We also replace the activation functions ReLU with Parametric ReLU [58](PReLU) for preventing dead neurons and vanishing gradients caused by ReLU. These modifications help stabilize the training and reduce color shift artifacts in the output. For convenience, we set all residual network blocks to have the same number of filters of 256.

Reconstruction

In the reconstruction stage, we apply another Pixelshuffle layer to upsample the extracted features to the desired resolution. This is followed by a final convolutional layer to reconstruct

the HR color image.

The summary of our network architecture is shown in Table 3.1.

Stage	Layer	Output Shape
Input (raw image)		$h \times w \times 1$
1	Conv with a stride of 2	$\frac{h}{2} \times \frac{w}{2} \times 256$
	Pixelshuffle	$h \times w \times \frac{256}{4}$
	Conv, PReLU	$h \times w \times 256$
2	Residual Block	$h \times w \times 256$
	...	$h \times w \times 256$
	Residual Block	$h \times w \times 256$
3	Pixelshuffle	$2 \cdot h \times 2 \cdot w \times \frac{256}{4}$
	Conv, PReLU	$2 \cdot h \times 2 \cdot w \times 256$
	Conv	$2 \cdot h \times 2 \cdot w \times 3$
Output (color image)		$2 \cdot h \times 2 \cdot w \times 3$

Table 3.1 – The summary of our network architecture. The stages $1,2,3$ of the first column correspond to the three stages of color extraction, feature extraction and non-linear mapping, and reconstruction, respectively illustrated in Figure 3.3. We set the number of filters as 256 and use 24 residual blocks in stage 2.

3.3 Experiments

For training and evaluation of the network, we use the publicly available dataset RAISE [23], which provides 8,162 uncompressed raw images as well as their demosaiced counterparts in TIFF format. RAISE contains a variety of content including landscape, people, building, animal, plant, *etc.* Some example images from RAISE are shown in Figure 3.5.

It is to be noted that if we use images that are already demosaiced by a given algorithm to our network, the network will learn to generate any artifacts introduced by the demosaicing algorithm. Moreover, we only deal with demosaicing and SR. We make the assumption that other image restoration tasks, such as denoising, would be resolved in other steps in the image processing pipeline, thus noise should not be modeled in our image sampling pipeline. To circumvent this problem, we use the demosaiced images of RAISE that are larger than 16 megapixels in size. We then perform a progressive downsizing of the image in steps by a factor of 1.25 each time until we obtain one-fourth of the original image size (*i.e.*, down to about 4 megapixels). This is done to eliminate artifacts that have potentially been introduced by the demosaicing algorithm as well as by other factors in the camera processing pipeline. This way we obtain high-quality ground-truth I^{HR}, to serve as the super-resolved images.

To create input raw images I^{raw} from these ground-truth images, we further downsample the previously downsampled images to one-fourth of the size (to about 1 megapixels) also using the progressive downsizing. We follow the assumed image formation demonstrated in Figure

Figure 3.5 – Example images in RAISE [23].

3.2. As required for the Bayer pattern, we set the downsample factor $s = 2$, and sample pixels from the three channels in the Bayer CFA pattern to obtain a single-channel mosaiced image as low-resolution input image for training. Thus, for a $h \times w \times 1$ raw image input, the desired color image output is of size $2 \cdot h \times 2 \cdot w \times 3$. These steps are illustrated in Figure 3.6.

To train our network, we use a subset of RAISE of 6,000 images. In particular, we randomly selected 4,000 photos from the landscape category and randomly selected 2,000 photos from other categories, as the landscape images usually contain more challenging details than other images. We also randomly select 50 images from the rest of the RAISE dataset to build the testing set. We made sure that there were no duplicate images in the training and testing sets.f

3.3.1 Training Details

For training, we use $64 \times 64 \times 1$ sized patches from the Bayer mosaic images as input. As output images we use color image patches of size $128 \times 128 \times 3$ from the high-resolution (4 megapixel) images. We train our network with ADAM optimizer [73] by setting the learning rate = 0.0001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. We set the batch size to 16. For better convergence of the network, we halve the learning rate after every 10000 mini-batch updates.

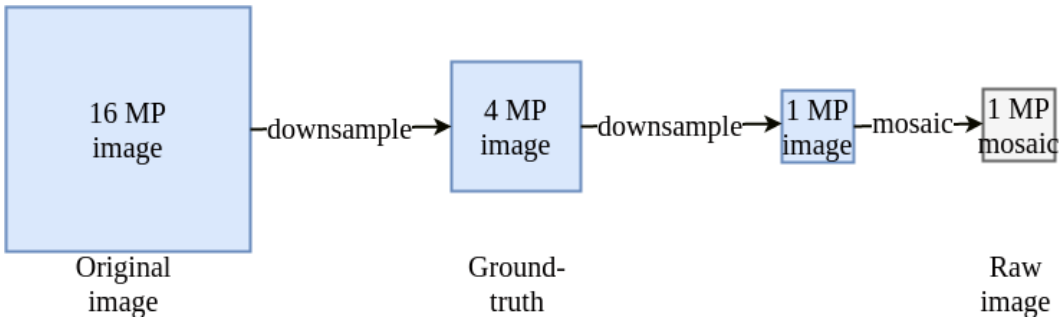


Figure 3.6 – Illustration of the steps we take to create the input and output images of our training and testing dataset. The original 16 megapixel images are downsampled to 4 megapixel eliminate demosaicing errors. The 4 megapixel images serve as reference SR images, whose downsampled 1 megapixel version provide the the single-channel raw images used as input to our network. Note that all the downsampling operation in the procedure is using a progressive downsizing to avoid the aliasing artifacts.

3.3.2 Results

Since we are not aware of any other joint demosaicing and SR algorithms for single raw image in the existing literature, there is no existing method for us to compare with. To illustrate the performance of our proposed end-to-end network, we designed experiments to simulate the conventional image processing pipeline for comparison. We compare our method with the sequential application of different state-of-the-art demosaicing algorithms (ADMM [121], FlexISP [60], SEM [74] and DemosaicNet [42]) and the state-of-the-art SR algorithm (SRCNN [24] and MDSR$^+$ [84]). We use the published code from these works. As in the conventional image processing pipeline, demosaicing and SR are two different components which are supposed to be resolved independently, we do not fine-tune the SR algorithms on the demosaicing algorithms.

Note that ADMM [121], SEM [74], and DemosaicNet [42] perform joint demosaicing and denoising, for fair comparison, we set the noise-level $\sigma = 0$ for these methods. As SRCNN only provides upsampling in the luminance channel, we upsample the chroma channels using bicubic interpolation. The process is shown in Figure 3.7. We use the best model with most features of SRCNN (the 9-5-5 model).

Quantitative Results

In Table 3.2 we report the PSNR values of our approach in comparison to other methods on the testing dataset. In terms of PSNR, Our approach outperforms the next best combination of state-of-the-art techniques of demosaicing and SR by a significant PSNR difference of $1.3dB$ on average computed over the 50 images of the test-set. Note that the deep models for demoisaicing algorithms outperform the conventional methods on our test set (when comparing the last three rows with the other rows in the table), which demonstrate the power

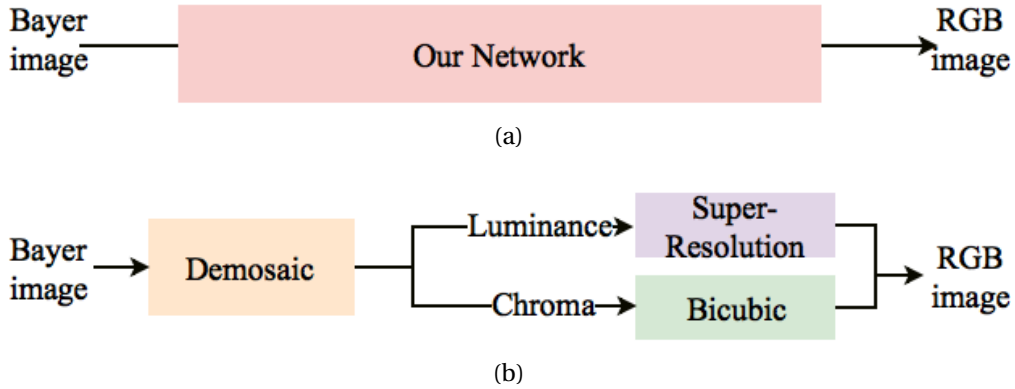


Figure 3.7 – (a) is our framework for joint demosaicing and SR, our network can perform the whole process in an end-to-end manner. (b) shows a typical pipeline to combine the demosaic algorithms and SR algorithms, which we use for comparing with other algorithms. Unlike most SR algorithms that output only the luminance channel, we directly generate full color output.

of CNNs. We also notice that although MDSR$^+$ is a better SR network than SRCNN according to some SR benchmark [122], the SR results of MDSR$^+$ on the demoisaiced images achieve worse PSNR and SSIM results than SRCNN. It is due to the fact that the SR network followed by the demoisaicing algorithm will also accumulate the artifacts created by the demoisaicing algorithm and the better SR modal amplify more the noise. Our proposed network, benefiting from the joint optimization, is able to reduce these kinds of artifacts.

Method	PSNR	SSIM
ADMM* [121]+SRCNN [24]	28.18	0.895
ADMM* [121]+MDSR$^+$ [84]	26.93	0.883
FlexISP [60]+SRCNN [24]	29.61	0.918
FlexISP* [60]+MDSR$^+$ [84]	29.12	0.919
SEM* [74]+SRCNN [24]	29.50	0.935
SEM* [74]+MDSR$^+$ [84]	29.37	0.938
DemosaicNet* [42]+SRCNN [24]	30.13	0.937
DemosaicNet* [42]+MDSR$^+$[84]	30.12	0.929
Ours	**31.41**	**0.948**

Table 3.2 – The mean PSNR and SSIM of different methods evaluated on our testing dataset. For the methods that perform joint demosaicing and denoising, we set their noise-level to 0 for fair comparison. There is a significant difference between the PSNRs and SSIMs of our proposed network and existing state-of-the-art methods.

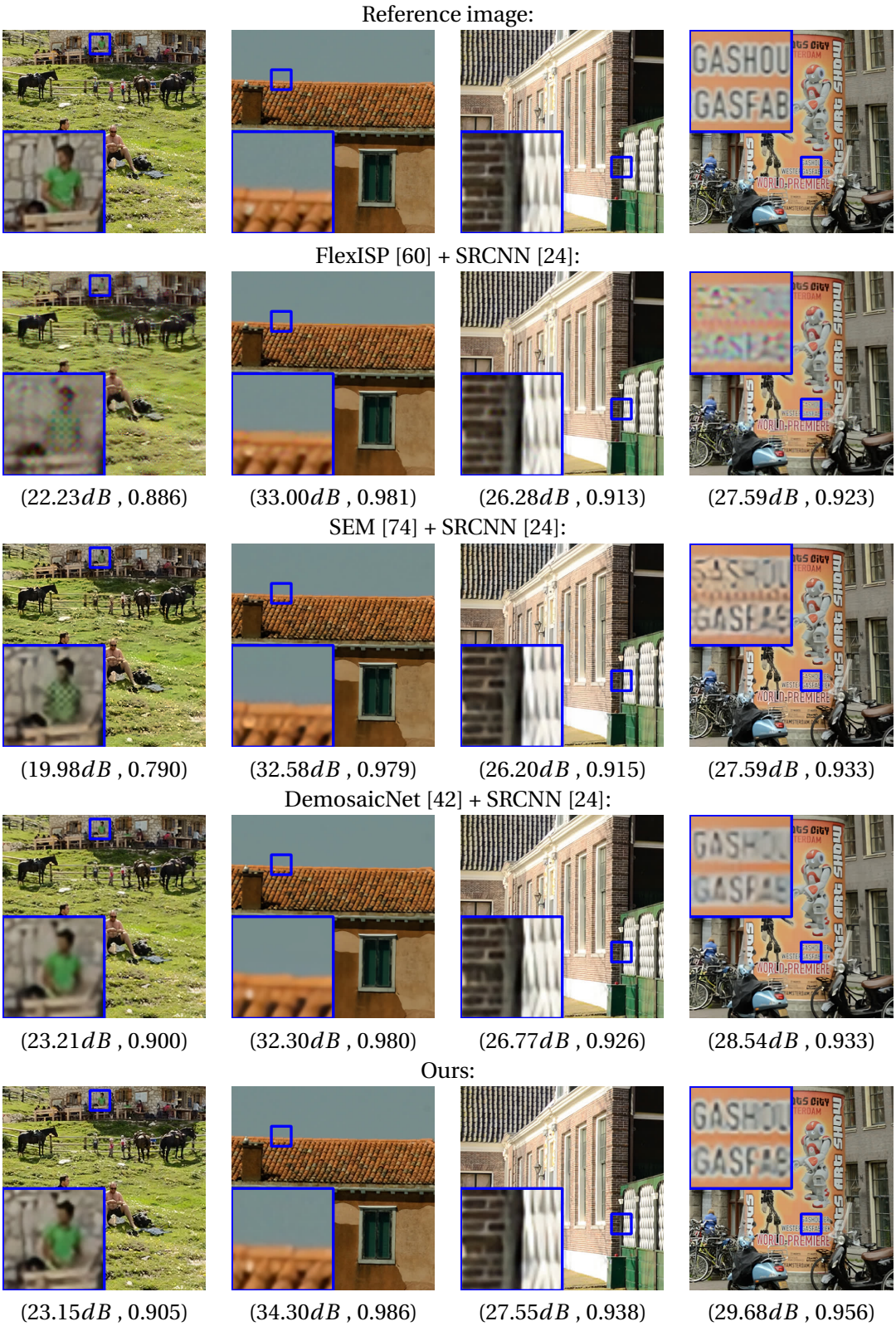


Figure 3.8 – Joint demosaicing and SR results on images from the RAISE [23] dataset. The two numbers in the brackets are the PSNR and SSIM scores, respectively.

Figure 3.9 – Joint demosacing and SR results on images from the RAISE [23] dataset. The two numbers in the brackets are the PSNR and SSIM scores, respectively.

Qualitative Results

To further validate the quality of our results, we show qualitative comparisons in Figure 3.8. We have omitted the comparisons with ADMM [121] and MDSR$^+$ [84] because the combination with these two algorithms performs worse than the presented results of the other combinations. Note that although MDSR$^+$ [84] is a superior SR algorithm to SRCNN [24], sometimes it performs worse than SRCNN as it accumulates more the artifacts produced by the demosaicing algorithms.

The combination of FlexISP [60] and SEM [74] produces some disturbing artifacts such as zippering around the edge and false color artifacts. These are particularly visible in the man's clothes (in the first column of Figure 3.8) and the text (in the last column of Figure 3.8).

Both DemosaicNet [42] and our network can produce demosaiced images without these artifacts, but our network is able to recover more realistic details. This is demonstrated in the second and the fourth column of Figure 3.8. Our network is able to produce higher quality color images without the visually disturbing artifacts introduced by the other methods.

Runtime

We test the runtime of our method and the algorithms we compared to on 100 input images of size 256×256 using a Nvidia TITAN X. The results are shown in Table 3.3. As FlexISP and SEM rely on iterative optimization, they take more than 100,000 ms on average. While DemosaicNet takes on average 650 ms for demosaicing alone, our method has an average of 619 ms for the joint operation of demosaicing and SR.

	Method	CPU (ms)	GPU (ms)
SR	SRCNN		297
	MDSR$^+$		5,283
Demoisaicing	ADMM	235,785	
	FlexISP	185,240	
	SEM	711,039	
	DemosaicNet		650
	Ours		619

Table 3.3 – Runtime of the tested demosaicing and SR algorithms. Our network is faster than other methods.

3.4 Conclusion and Discussion

The ill-posed problems of demosaicing and SR have always been dealt with as separate problems and then applied sequentially to obtain HR images. This has continued to remain the trend even after the advent of CNN's. In this chapter, we propose the first CNN-based

joint demosaicing and SR framework for single image, which is capable of directly recovering high-quality color HR images from raw images. Our proposed method outperforms all the tested combinations of the state-of-the-art demosaicing algorithms and the state-of-the-art SR algorithms in quantitative measurements of PSNR and SSIM as well as visually.

As our network performs demosaicing and SR in one shot. Our approach does not produce disturbing color artifacts akin to algorithms in the literature for the input images produced by our imaging pipeline. Although these demosaicing artifacts may not appear in the real-world cases as these noise and aberrations are eliminated by the lens blur, our approach provides the sharpest and the most realistic result compared with other methods even when ignoring the artifacts.

Note that as demosaicing is a spectral SR that is executed using different scales on different color channels to get a full color image, it can be viewed as a multi-scale SR on the Bayer CFA patterns. Thus it is reasonable to address demosaicing and SR in a unified context. This argues well for the use of our approach of jointly performing both tasks in camera image processing pipelines. Recent work published by Xu *et al.* [132] also have shown the superiority of learning SR with raw data. For mobile devices this can encourage the use of sensors with large pixels that capture a better dynamic range, rather than sacrificing dynamic range for higher resolution as is done at the moment.

4 Blind Image Super-Resolution

4.1 Introduction

As shown in the previous chapter, CNN-based SR models [24, 71, 128] for learning the mapping from LR images to HR images require large sets of paired LR and HR images for training. However, it is non-trivial to obtain such paired LR and HR ground-truth images of real scenes. Current SR networks, including our method in the previous chapter, rely on synthetically generated LR images [122]. The most common technique to generate an LR image is to apply bicubic interpolation [70] to the HR image. However, the bicubic convolution kernel is different from real camera blur [94]. The loss of high-frequency details in camera-captured images is due to several factors, such as optical blur, atmospheric blur, camera shake, and lens aberrations [98]. As a result, even though these CNN-based SR networks perform well on bicubic-downsampled LR images, their performance is limited on real photographs as they operate under a wrong kernel assumption [28, 94]. SR networks are sensitive to kernel mismatch [28] and SR results will contain obvious artifacts if an inaccurate kernel is used for SR training. As shown in Figure 4.1, when the kernel used for training SR networks are "sharper" than the real one, the SR results are overly smoothed, while when the kernel used for training SR networks are smoother than the real one, the SR results contain unnatural ringing artifacts [45]. Here, a sharper kernel means the value of the kernel is more concentrated in the middle and the pixels that are far away from the center has less coefficient on the filtering effect. Some GAN-based methods [8, 79, 109, 128] can be extended to train SR networks on unpaired datasets, they still rely on unrealistic blur kernels. SR on real LR photographs with unknown camera blur thus remains a challenging problem.

To resolve the SR problem of unknown blur kernels, namely *blind image SR*, we need to generate synthetic LR images with real camera blur for training the SR network. We can use kernel estimation algorithms [75, 78, 99] to extract realistic blur kernels from real LR photographs. However, as each camera, lens, aperture, and atmospheric condition combination may result in a different blur kernel, it is challenging to generate a sufficiently large and diverse dataset [75, 78] needed to train an SR network.

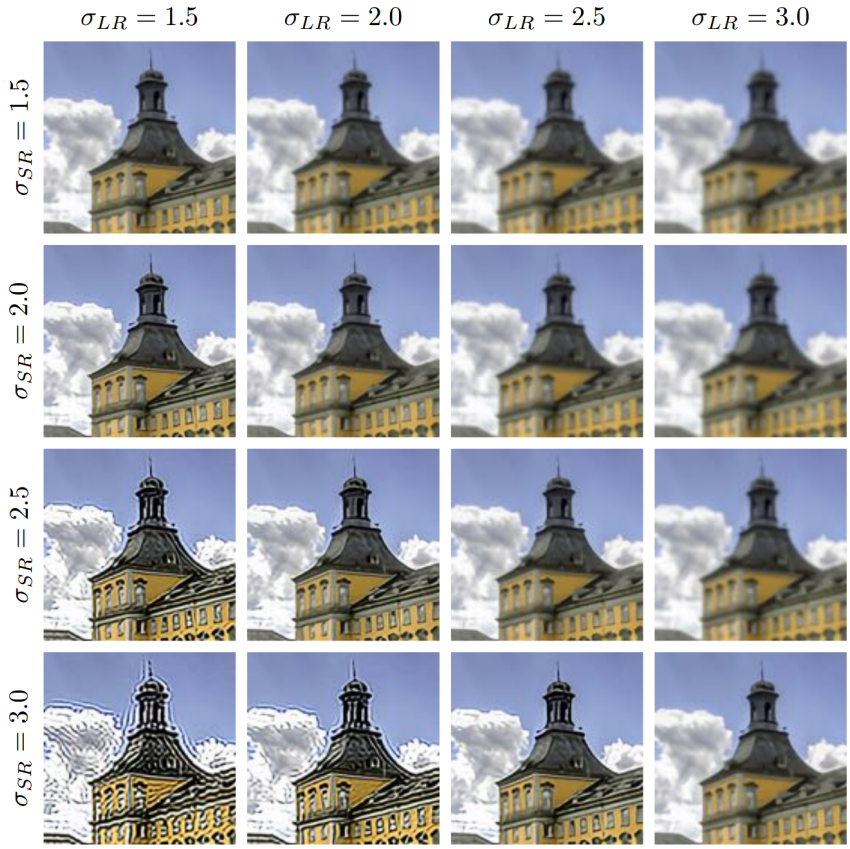


Figure 4.1 – SR sensitivity to kernel mismatch. σ_{LR} denotes the kernel used for generating the testing LR image and σ_{SR} denotes the kernel used for training SR network. Image taken from [45].

One approach is to generate synthetic LR images using many blur kernels [102], which will improve the generalization ability of the SR network. Using a kernel estimator, we first extract blur kernels from real photographs and use them for training a GAN. First proposed in [43], GANs are a class of neural networks that learn to generate synthetic samples with the same distribution as the given training data [7]. We thus augment the limited kernel set we obtained using kernel estimation by leveraging the GAN's ability to approximate complex distributions [66, 79, 106, 137] to learn and generate additional blur kernels.

Our Kernel Modeling SR (KMSR) thus consists of two stages, as shown in Figure 4.2. We first generate a GAN augmented realistic blur kernel pool by extracting real blur kernels from photographs with a kernel estimation algorithm and by training a GAN to augment the kernel pool. We then construct a paired LR-HR training dataset with kernels sampled from the kernel pool, and train a deep CNN for SR.

Our major contributions in this chapter are as follows[1]:

[1] This work was published in [151]

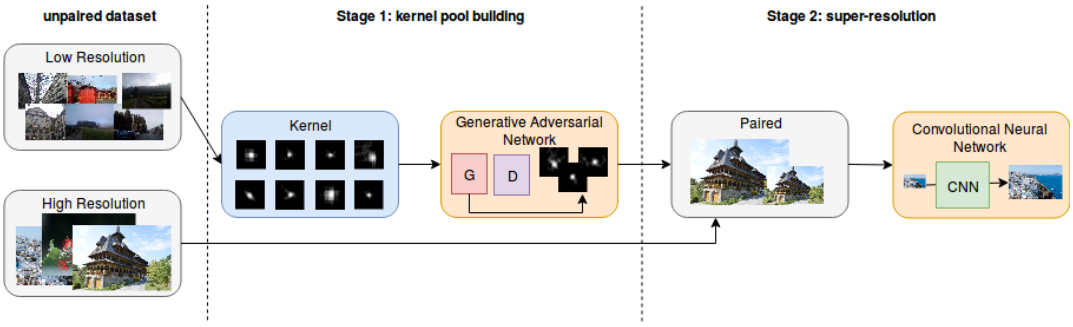


Figure 4.2 – Illustration of our proposed kernel modeling SR (KMSR) framework. The first stage consists of blur kernel estimation from real photographs, which are used in training a GAN to generate a large pool of realistic blur kernels. These generated blur kernels are then utilized to create a paired dataset of corresponding HR and LR images for the training of a deep CNN.

- We introduce Kernel Modeling SR, namely KMSR, to improve blind SR on real photographs by incorporating realistic blur kernels in the framework, which improves the generalization capability of the network to unseen blur kernels.

- We show that a GAN can reliably generate realistic blur kernels.

- We test the blind SR performance on both selected blur kernels and real images. Extensive experiments show that the proposed KMSR achieves state-of-the-art results in terms of both visual quality and objective metrics.

4.2 Blind Image Super-Resolution

4.2.1 Kernel Modeling Blind Super-Resolution

Let I^{HR} be an HR image of size $h \times w$ pixels, and let I^{LR} be an LR observation of I^{HR} of size $\lfloor h/s \rfloor \times \lfloor w/s \rfloor$, where $s > 1$ is the downsampling factor. The relation between I^{HR} and I^{LR} is expressed as in Chapter 2:

$$\mathrm{I}^{LR} = (\mathrm{I}^{HR} \otimes k) \downarrow_s + n, \tag{4.1}$$

where k denotes an unknown blur kernel, $\downarrow_s$ denotes a decimation operator by a factor s, $\otimes$ is the convolution operation and n is the noise. We assume here that there is no noise in the LR image acquisition model or the denoising is performed perfectly in the other process, *i.e.*, $n = 0$.

We upscale the LR image to a coarse HR image $\mathrm{I}^{LR'}$ with the desired size $h \times w$ with traditional bicubic interpolation by the same factor s:

$$\mathrm{I}^{LR'} = (\mathrm{I}^{LR} \otimes b_s), \tag{4.2}$$

where b_s is the bicubic upscaling kernel with scale s. Thus we have

$$\mathrm{I}^{LR'} = ((\mathrm{I}^{HR} \otimes k) \downarrow_s) \otimes b_s, \tag{4.3}$$

Simplified,

$$\mathrm{I}^{LR'} = \mathrm{I}^{HR} \otimes k' \tag{4.4}$$

where $k' = (k \otimes b_s) \downarrow^s$.

To train a blind CNN SR network, we need paired training data I^{HR} and $\mathrm{I}^{LR'}$, obtained according to Equation 4.4 with different kernels k'. We adopt a GAN to help solve this problem. It is difficult to train a generative network to consistently recover HR images without artifacts. Thus, alternatively, our GAN is trained to produce blur kernels rather than images.

4.2.2 Blur Kernel Pool

As mentioned in the previous section, SR networks are sensitive to kernel mismatch, thus it is important to collect realistic blur kernels for building the synthetic LR-HR training pairs. In this thesis, we estimate realistic blur kernels from real photographs. However, each camera has multiple Point Spread Functions (PSFs) resulting from optical blur (aberrations, aperture, focal length, *etc.*) and motion blur. It is not realistic to collect all possible PSFs. Thus, to make the SR network more generic to unseen kernels, we extend the collected kernels by using GAN for kernel modeling and kernel generation. The combination of the estimated kernels and the GAN generated kernels forms the large kernel-pool used in building paired LR-HR training data. We explain the blur kernel estimation and extension with GAN in this section.

Blur Kernel Estimation

To generate a set of realistic blur kernels $\mathbb{K}' = \{k'_1, k'_2, \ldots, k'_e\}$, we first randomly extract a patch P of size $d \times d$ from the bicubic-upscaled LR image (or coarse HR image) $\mathrm{I}^{LR'}$. We then estimate the blur kernel k' of size 25×25 from P using the blur kernel estimation algorithm of [99]. Their standard formulation for image deblurring, based on the dark channel prior [57], is as follows:

$$\min_{\hat{\mathrm{P}},k'} \left\| \hat{\mathrm{P}} \otimes k' - \mathrm{P} \right\| + \theta \left\| k' \right\|_2^2 + \mu \left\| \nabla \hat{\mathrm{P}} \right\|_0 + \left\| \nabla \hat{\mathrm{P}}^{dark} \right\|_0. \tag{4.5}$$

P is the extracted patch from $\mathrm{I}^{LR'}$, $\hat{\mathrm{P}}$ is the estimated deblurred patch, and $\hat{\mathrm{P}}^{dark}$ is the dark channel [57] of the patch. Coordinate descent is used to alternatively solve [99] for the latent patch P:

$$\min_{\hat{\mathrm{P}}} \left\| \hat{\mathrm{P}} \otimes k' - \mathrm{P} \right\| + \mu \left\| \nabla \hat{\mathrm{P}} \right\|_0 + \left\| \nabla \hat{\mathrm{P}}^{dark} \right\|_0, \tag{4.6}$$

and the blur kernel k'

$$\min_{k'} \left\| \hat{\mathrm{P}} \otimes k' - \mathrm{P} \right\| + \theta \left\| k' \right\|_2^2. \tag{4.7}$$

To minimize Equation 4.6, half-quadratic splitting is utilized. The kernel estimation in Equation 4.7 is a least squares problem and can be solved using its closed form solution. Note that the optimization algorithm for solving Equation 4.5 is proposed by Pan *et al.*and is not the contribution of this thesis. More details about the algorithms can be found in [99].

To eliminate patches that are lacking high frequency details (such as patches extracted from the sky, ground, and walls without much texture, *etc.*) in which the blur kernel estimation algorithm might fail, we define constraints for P as follows:

$$|\text{Mean(P)} - \text{Var(P)}| \geq \alpha \cdot \text{Mean(P)} \tag{4.8}$$

where Mean(P) and Var(P) calculate the mean intensity and the variance, respectively, and $\alpha \in (0,1)$. If the constraint is satisfied, P will be regarded as a valid patch and the estimated blur kernel k' from P is added to the set K'.

We extract 5 patches from each bicubic-upscaled LR image $\mathrm{I}^{LR'}$. We empirically set the patch size $d = 512$ and $\alpha = 0.03$.

Kernel Modeling with GAN

In practice, input LR images may be hard to obtain and limited to few camera models. In addition, the kernel-estimation algorithm [99] is computationally expensive. As such, the quantity and diversity of kernels collected in the last subsection is limited, and the results of training a deep CNN only with these kernels will not suffice. We thus propose to model the blur kernel distribution over the estimated kernel set $\mathbb{K}'$, and to generate a larger blur kernel pool $\mathbb{K}^+$ that contains more examples of realistic blur kernels with more diversity. We use a GAN to generate these realistic blur kernels.

We use WGAN-GP [46], which is an improved version of WGAN [5], for the objective function of our GAN:

$$\mathscr{L}_{GAN} = \mathop{\mathbb{E}}_{\tilde{f}\sim\mathbb{P}_g}[D(\tilde{f})] - \mathop{\mathbb{E}}_{f\sim\mathbb{P}_r}[D(f)] + \lambda \mathop{\mathbb{E}}_{\hat{f}\sim\mathbb{P}_{\hat{f}}}[(\left\|\nabla D(\hat{f})\right\|_2 - 1)^2], \tag{4.9}$$

where D is the discriminative network, $\mathbb{P}_r$ is the distribution over $\mathbb{K}'$, and $\mathbb{P}_g$ is the generator distribution. $\mathbb{P}_{\hat{f}}$ is defined as a distribution sampling uniformly along straight lines between pairs of points sampled from $\mathbb{P}_r$ and $\mathbb{P}_g$. f, $\tilde{f}$, $\hat{f}$ are the random samples following the distribution $\mathbb{P}_r$, $\mathbb{P}_g$ and $\mathbb{P}_{\hat{f}}$, respectively.

We adopt a similar network architecture to DCGAN [46]. The generative network G takes $z \sim N(0,1)$, a vector of length 100 and generates a blur kernel sample. It contains 4 fractionally-strided convolutions [27] of filter size 4×4, with batch normalization [65], ReLU [97], and a final convolution layer of filter size 8×8. The filter number of G from the second to the last unit is 1025, 512, 256, 1, respectively. The discriminative network D takes a kernel sample as input and identifies if it is fake, it contains 3 convolution layers with instance batch-normalization [125]

and leaky ReLU [131]. The filter number of D from the first to the third unit is 256, 512, 1024, respectively. The overall architecture of GAN is shown in Figure 4.3.

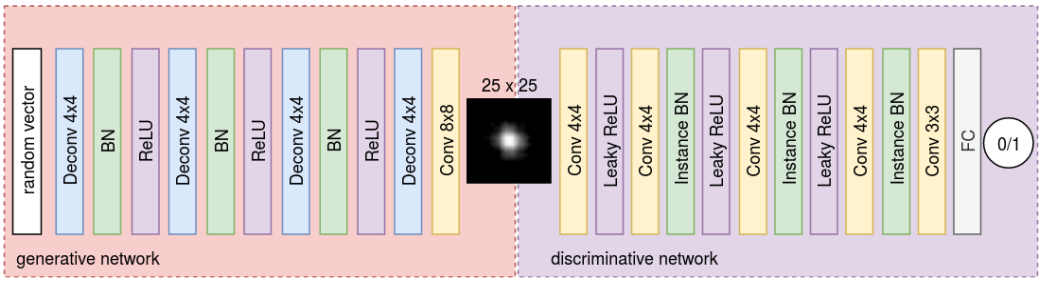


Figure 4.3 – The network architecture of the Generative Adversarial Network for estimating the distribution of the blur kernels. The generative network takes a $z \sim N(0,1)$, a vector of length of 100 and generates a kernel sample, the discriminative network takes a kernel sample as input and identifies if it is fake. The filter number of the generative network from the second to the last unit is 256, 128, 64, and 1, respectively. The filter number of the discriminative network from the first to the fourth unit is 64, 128, 256, and 512, respectively.

The trained GAN model G is used to generate blur kernel samples for augmenting $\mathbb{K}'$ until the final kernel pool $\mathbb{K}^+ = \mathbb{K}' \cup \{G(z_1), G(z_2), G(z_3), \ldots\}$ is obtained. Like the normalization of kernels in [99], we apply sum-to-one and non-negative constraints on the generated kernels.

4.2.3 Super-Resolution with CNN

Deep neural networks implicitly learn the latent model from the paired training dataset, and thus do not require explicit knowledge of image priors. Hence, we utilize a CNN in our SR framework.

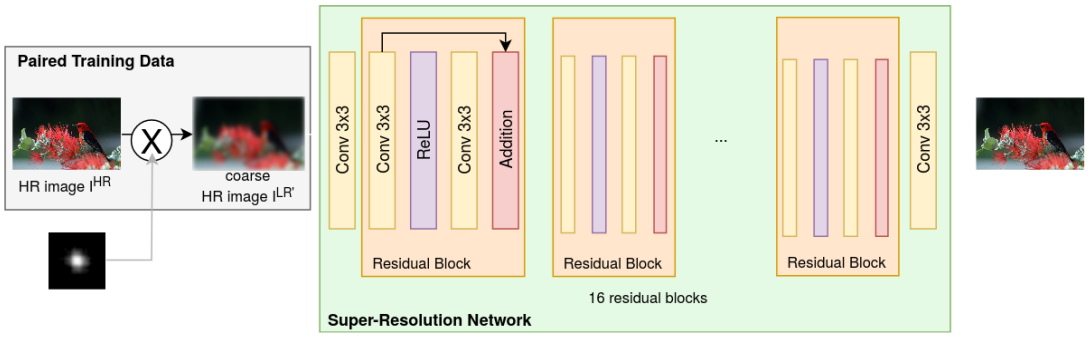


Figure 4.4 – The Convolutional Neural Network architecture of KMSR. We convolve the HR image I^{HR} with a blur kernel k' randomly chosen from the blur kernel pool $\mathbb{K}^+$ to generate the coarse HR image $I^{LR'}$. The other units each have 64 filters except for the last unit, where the filter number is equal to the number of output channels.

We create the training dataset in the following manner: the HR images are divided into small patches of size $m \times m$, which form the set $\mathbb{I}_{HR} = \{I_1^{HR}, I_2^{HR}, \ldots\}$. Blur kernels in K^+ obtained in Section 4.2.2 are randomly chosen to convolve with patches in $\mathbb{I}_{HR}$ to obtain

$\mathbb{I}_{LR'} = \{I_1^{LR'}, I_2^{LR'}, \ldots\}$, where $I_i^{LR'} = I_j^{HR} \otimes k'_l$. The sets $\mathbb{I}_{LR'}$ and $\mathbb{I}_{HR}$ form a paired training dataset $\{\mathbb{I}_{LR'}, \mathbb{I}_{HR}\}$.

The network structure of the CNN, which consists of 16 residual blocks [59], is illustrated in Figure 4.4. The batch normalization is removed similary to in the model in Chapter 3 to improve the performance of the network. Zero padding is adopted to ensure consistent input and output dimensions. The objective function of our network is

$$\mathscr{L}_1 = \sum |I^{HR} - I^{SR}| \tag{4.10}$$

and enables the network to obtain better performance than using a Euclidean norm [145].

4.3 Experiments

4.3.1 Implementation Details

We utilize the DPED [64] images to build the realistic blur kernel set $\mathbb{K}'$. DPED [64], as introduced in Chapter 2, is a large-scale dataset that consists of over 22,000 real photos captured with 3 different low-end phone models. Different phone models have different camera characteristics and result in different camera blur effect in the photos taken of the same scene, which is shown in Figure 4.5 and Figure 4.6.

(a) Blackberry Passport (b) Sony Xperia Z (c) iPhone 3GS

Figure 4.5 – Patches from photos of the same scene using different cameras in the DEPD [64] dataset. We can clearly see that different cameras have result in different blur in the photos.

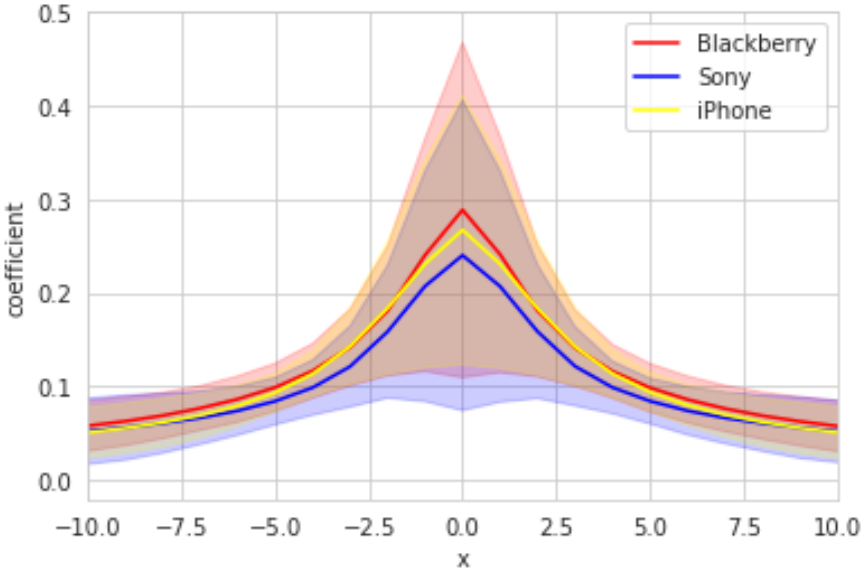


Figure 4.6 – Plot of blur kernels estimated from DPED dataset. The shadow area illustrates the variance. The figure shows that different phone models have slightly different camera characteristics.

We separate the dataset into two parts, *DPED-training* and *DPED-testing*, according to the camera models. *DPED-training* consists of photos taken with the Blackberry Passport and Sony Xperia Z, and serves as the reference real-photography LR set for extracting the realistic blur kernels k'_e described in Section 4.2.2. *DPED-testing* consists of photos captured with the iPhone 3GS, and is used as a validation dataset. We collect 1000 realistic blur kernels $\mathbb{K}' = \{k'_1, k'_2, \ldots, k'_{1000}\}$ from *DPED-training* by using the kernel estimation codes from [99]. We use these kernels in the training of the kernel modeling GAN G. We set the batch size as 32 and $\lambda = 10$ for the loss function (see Equation 4.9). G is trained for 20,000 epochs. The extended blur kernel pool $\mathbb{K}^+$ is obtained by generating 1,000 kernels using the trained G and adding them to $\mathbb{K}'$.

We use the training set of DIV2K [122] as HR images, from which we extract patches of size 128×128. We build the paired dataset $\{\mathbb{I}_{LR'}, \mathbb{I}_{HR}\}$ during training of the SR network: in each epoch, each HR patch is convolved with a kernel k' randomly chosen from $\mathbb{K}^+$ to obtain a coarse HR patch. We train our SR network with ADAM optimizer [73]. We set the batch size to 32. The learning rate is initialized as 10^{-4} and is halved at every 10 epochs.

4.3.2 Estimated Kernels

We study the distributions of blur kernels. We show examples of kernels k'_e generated with KMSR in Figure 4.7 and Figure 4.8. Since all images in DPED were captured using a tripod, there is no motion blur exists in the dataset, thus none of the kernels shows any motion blur.

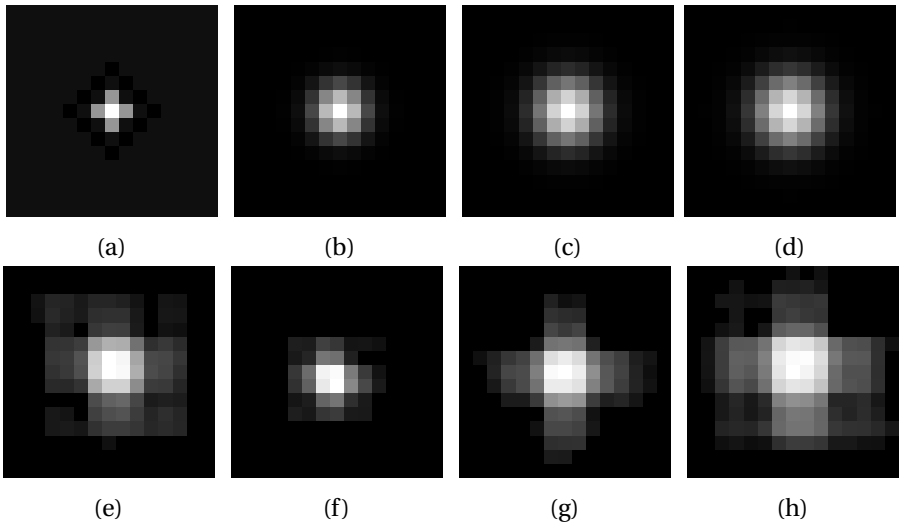


Figure 4.7 – Visualization of different blur kernels for scale $s = 2$ (×2 SR). To better visualize the kernels, we only show a 15 × 15 patch cropped from the center. (a) the bicubic kernel [70] with anti-aliasing implemented in Matlab [44]; (b), (c) and (d) three isotropic Gaussian kernels $g_{1.25}$, $g_{1.6}$ and $g_{1.7}$, respectively, which are widely used in ×2 SR. (e), (f) two kernel samples k'_e estimated from real photos, (g) and (h) two blur kernels $G(z_i)$ generated with the GAN.

We also visualize the Matlab bicubic kernel and three isotropic Gaussian kernels g_σ with σ 1.25, 1.6 and 1.7 that are commonly used to synthesize LR images in ×2 SR [26, 56, 146]. Note that the bicubic kernel is band-pass compared to the low-pass shape of the other kernels. The bicubic kernel is designed to keep the sharpness of the image and to avoid aliasing during the downsampling operation [70]. As stated in [28], the bicubic kernel is *not* a proper approximation of the real blur kernel in image acquisition, as camera blur is low-pass and often attenuates the high-frequency information of the scene more. In Figure 4.8, the kernels generated by by KMSR encompass a wide range of distributions, including the Gaussian kernels that are a better approximation of the real camera blur [98] than the bicubic kernel. KMSR is thus able to generate very diverse coarse HR images.

4.3.3 Experiments on Bicubic and Gaussian Blur Kernels

In this section, we evaluate KMSR and other CNN-based SR networks on synthetic LR images by applying different blur kernels to the validation set of the DIV2K [122] dataset.

We test on two upscaling factors, $s = 2$ (×2 SR) and $s = 4$ (×4 SR) and on four synthetic LR datasets that are generated using four different kernels on the DIV2K [122] validation set. We include the anti-aliasing bicubic kernel, as it is used by many algorithms even though it is not a physically feasible camera blur for real images [28]. We also test on 3 isotropic Gaussian kernels, $g_{1.25}$ [146], $g_{1.6}$ [26] and $g_{1.7}$ [56]; they are commonly used as blur kernels in the generation of synthetic LR images [98]. The four kernels are visualized in the first row of

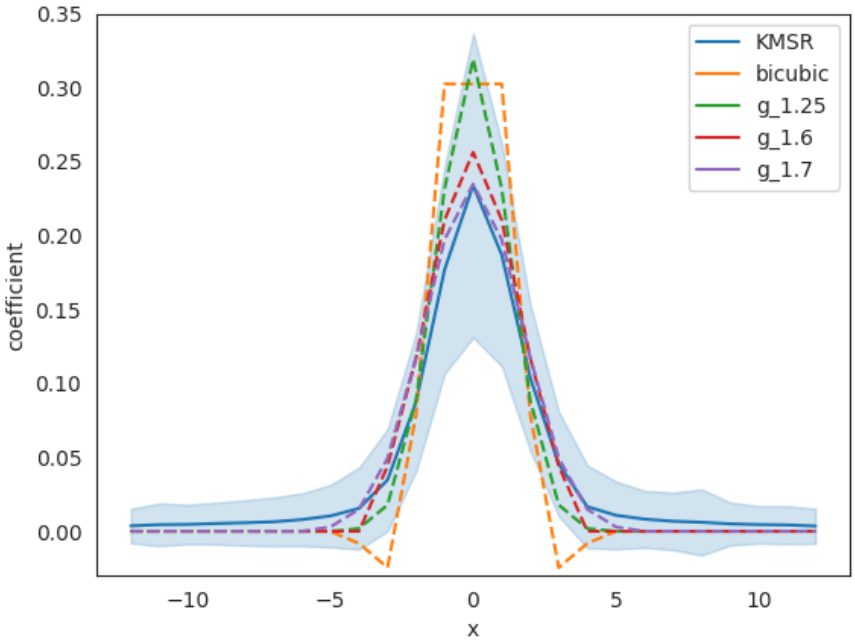


Figure 4.8 – Plot of different blur kernels. The solid line shows the mean kernel shape from the blur kernel pool $\mathbb{K}^+$ generated with KMSR. The shadow area illustrates the variance. The dashed lines show the shape of the bicubic kernel [70] and three Gaussian kernels that are commonly used in synthesizing LR images [26, 56, 146].

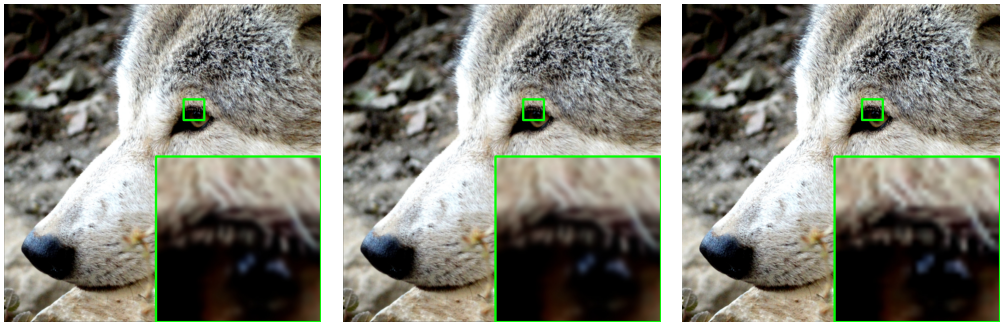
Figure 4.7.

We compare our proposed KMSR to the state-of-the-art CNN-based SR methods: SRCNN [24] (we use the 9-5-5 model), VDSR [71], EDSR [84] and DBPN [54]. We use the published codes and models from the respective authors. Note that these four networks are trained using only the bicubic kernel in the generation of corresponding LR images from HR images.

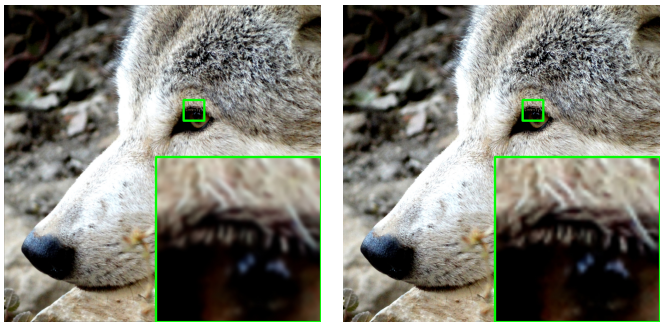
The quantitative results of the different SR networks on the different LR datasets are provided in Table 4.1. Although KMSR produces worse results on LR images generated with the bicubic kernel, it outperforms all other networks on all other experimental settings on both upscaling factors $s = 2$ and $s = 4$. We can also observe that the performance of SR networks that are trained using only bicubic LR images is limited when the bicubic kernel deviates from the true blur kernel. These networks gain less than $0.4dB$ improvement in PSNR compared to simple bicubic interpolation (column 3 in Table 4.1). Even with deeper layers, EDSR [84] and DBPN [54] do not outperform shallow networks SRCNN [24] and VDSR [71]. By modeling realistic kernels, our KMSR outperforms them all by up to $1.91dB$. A visual comparison using $g_{1.6}$ as blur kernel and $s = 2$ as upscaling factor is given in Figure 4.9. Note that KMSR produces results that visually appear sharper than other methods, as it is trained using more realistic blur kernels.

Blur Kernel	Scale	Bicubic	SRCNN [24]	VDSR [71]	EDSR [84]	DBPN [54]	**KMSR**
bicubic	×2	29.94	31.89	32.63	33.58	33.84	33.52
$g_{1.25}$		26.14	26.56	26.54	26.58	26.60	27.94
$g_{1.6}$		25.49	25.72	25.72	25.69	25.70	27.63
$g_{1.7}$		25.11	25.30	25.34	25.28	25.28	27.15
bicubic	×4	26.28	27.89	28.04	28.95	29.03	27.99
$g_{2.3}$		24.71	24.83	24.91	25.10	25.18	26.14
$g_{2.5}$		24.34	24.30	24.34	24.39	24.42	25.64
$g_{2.7}$		24.11	24.14	24.05	24.27	24.23	25.33

Table 4.1 – [122] in terms of PSNR in the evaluation of bicubic and Gaussian blur kernels. We highlight the best results in red color and the second best in blue color. Note that our proposed KMSR outperforms other state-of-the-art SR networks by up to 1.91dB on Gaussian kernels.



(a) SRCNN [24] (b) VDSR [71] (c) EDSR [84]



(d) DBPN [54] (e) **KMSR**

Figure 4.9 – Qualitative comparison of ×2 SR on image "0805" from DIV2K [122], using a Gaussian blur kernel $g_{1.6}$ as the blur kernel and $s = 2$ as upscaling factor.

4.3.4 Experiments on Realistic Kernels

To validate the capability of the proposed KMSR on images with real unknown kernels, we conduct experiments on synthesizing LR images with unseen realistic blur kernels on ×2 and ×4 SR. We collect 100 blur kernels from the LR images in the *DEPD-testing* dataset (*i.e.*, the iPhone3GS images), which are not seen in the training of KMSR. We then apply these blur kernels to generate coarse HR images using the DIV2K [122] validation set. Table 4.2 shows

(a) SRCNN [24] (b) VDSR [71] (c) EDSR [84]

(d) DBPN [54] (e) **KMSR**

Figure 4.10 – Qualitative comparison on ×4 SR on image "0816" from DIV2K [122], using a Gaussian blur kernel $g_{2.5}$ as the blur kernel and $s = 4$ as upscaling factor.

the resulting PSNR and SSIM of the different SR networks. As before, the performance of the SR networks trained using only the bicubic kernel is limited on these images.

This highlights the sensitivity of CNN-based SR networks to wrong kernels in the creation of the training dataset. Blur kernel modeling is a promising venue for improving SR networks if the algorithm is to be applied to real camera data.

We present qualitative results in Figure 4.12. KMSR successfully reconstructs the detailed textures and edges in the HR images and produces visually more pleasing images.

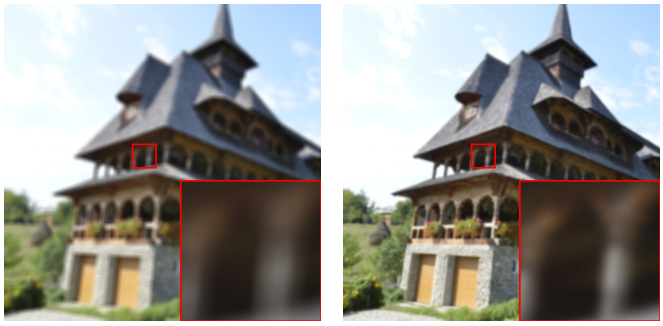
4.3.5 Experiments on Real Photographs

We also conduct ×2 SR experiments on real photographs. Figure 4.13 illustrates the KMSR output on one photograph captured by the iPhone3GS in the *DEPD-testing* dataset. Perceptual-driven SR methods usually recover more detailed textures and achieve better visual quality than previous SR networks. In addition to the four SR methods we compare to, we thus also show the output from the perceptually-optimized SR network ESRGAN [128]. It is noticeable that the networks trained using only the bicubic-downsampled LR images tend to produce overly smooth images, whereas KMSR can recover a sharp image with better details.

Method	Scale	PSNR	SSIM
bicubic interpolation	×2	25.06	0.72
SRCNN [24]		25.30	0.74
VDSR [71]		25.29	0.74
EDSR [84]		25.28	0.74
DBPN [54]		25.30	0.75
KMSR		27.52	0.79
bicubic interpolation	×4	23.32	0.69
SRCNN [24]		23.42	0.69
VDSR [71]		23.39	0.69
EDSR [84]		23.49	0.69
DBPN [54]		23.51	0.70
KMSR		25.13	0.74

Table 4.2 – Comparison on DIV2K [122] in the evaluation of realistic blur kernels estimated from *DEPD-testing*. We highlight the best results in red color and the second best in blue color.

(a) SRCNN [24] (b) VDSR [71] (c) EDSR [84]



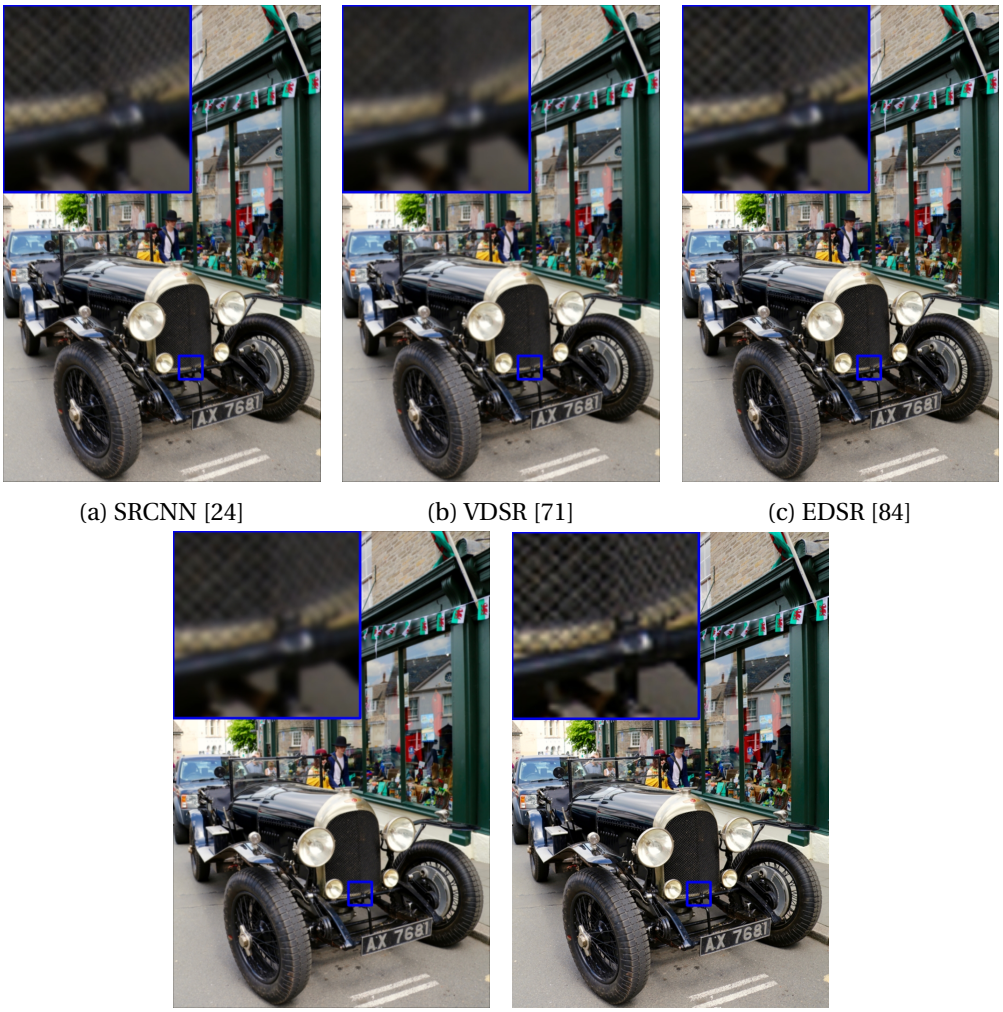
(d) DBPN [54] (e) **KMSR**

Figure 4.11 – Qualitative comparison on ×4 SR on image "0834" from DIV2K [122], using a realistic blur kernel estimated from *DPED-testing*.

Psychovisual Experiment

As no ground-truth or reference image is available for these super-resolved images, we conduct a psychovisual experiment on 50 images on a crowd-sourcing website[2] to quantitatively

[2] www.clickworker.com



(a) SRCNN [24] (b) VDSR [71] (c) EDSR [84]

(d) DBPN [54] (e) **KMSR**

Figure 4.12 – Qualitative comparison on ×2 SR on image “0847” from DIV2K [122], using a realistic blur kernel estimated from *DPED-testing*.

prove the effectiveness of the proposed approach. We only compare KMSR to EDSR [84] and DBPN [54] as they are the state-of-the-art CNN-based SR networks. Note that because of the resolution limitations of display devices, we could not show full-resolution images. We randomly select 50 images from *DEPD-testing* and crop patches of size 500 × 500 from each image. We perform the experiment as a three Alternative-Forced-Choice test (3-AFC). For each image, the observer was shown three SR results from our proposed KMSR, EDSR [84], and DBPN [54]. The screenshot of the interface is shown in Figure 4.14. A zoom window allowed the observers to enlarge details of the images. The methods’ names were hidden and the order is randomly shuffled. The observers were asked to select "the clearest and sharpest" image. To make sure the observers check the details of each SR result, they can only proceed to the next images once they have clicked the all three buttons that will show the three SR results.

(a) VDSR [71] (b) EDSR [84] (c) DBPN [54]

(d) ESRGAN [128] (e) **KMSR**

Figure 4.13 – ×2 SR qualitative comparison of different SR networks on image “83” from *DPED-testing*.

We collected 35 votes for each of the 50 image triplets. Table 4.3 shows the raw number of votes for each method. We also summarize the number of preferences in the table. Because guessing will lead to a percentage of preference equal to the reciprocal of the number of alternatives, the level at which the threshold is defined is adjusted for chance. Thus, for a 3-AFC experiment, the threshold is defined as 67%. For 44 out of 50 images, the SR results from our proposed KMSR are preferred, which shows that KMSR also qualitatively outperform the other SR methods.

	EDSR [84]	DBPN [54]	KMSR
Number of preferences	2/50	0/50	44/50
Raw Votes	119/1750	26/1750	1650/1750

Table 4.3 – Results of the psychovisual experiment. Number of preferences show the number of SR results from the specific method that are chosen as "the clearest and the sharpest image" by more than 67% of the participants. For 44 out of 50 images, results from our KMSR are favored over the other two methods.

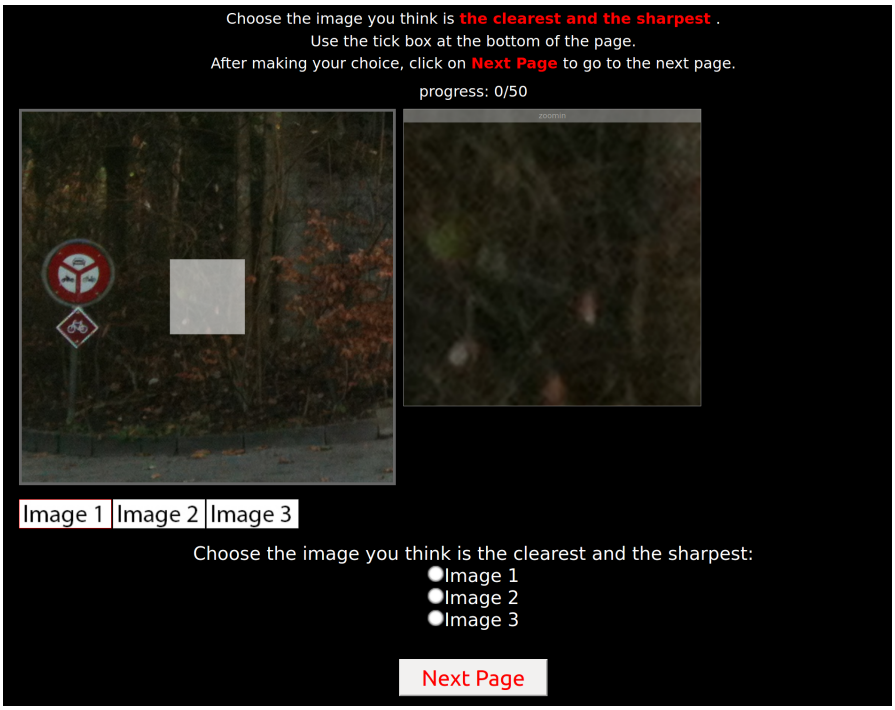
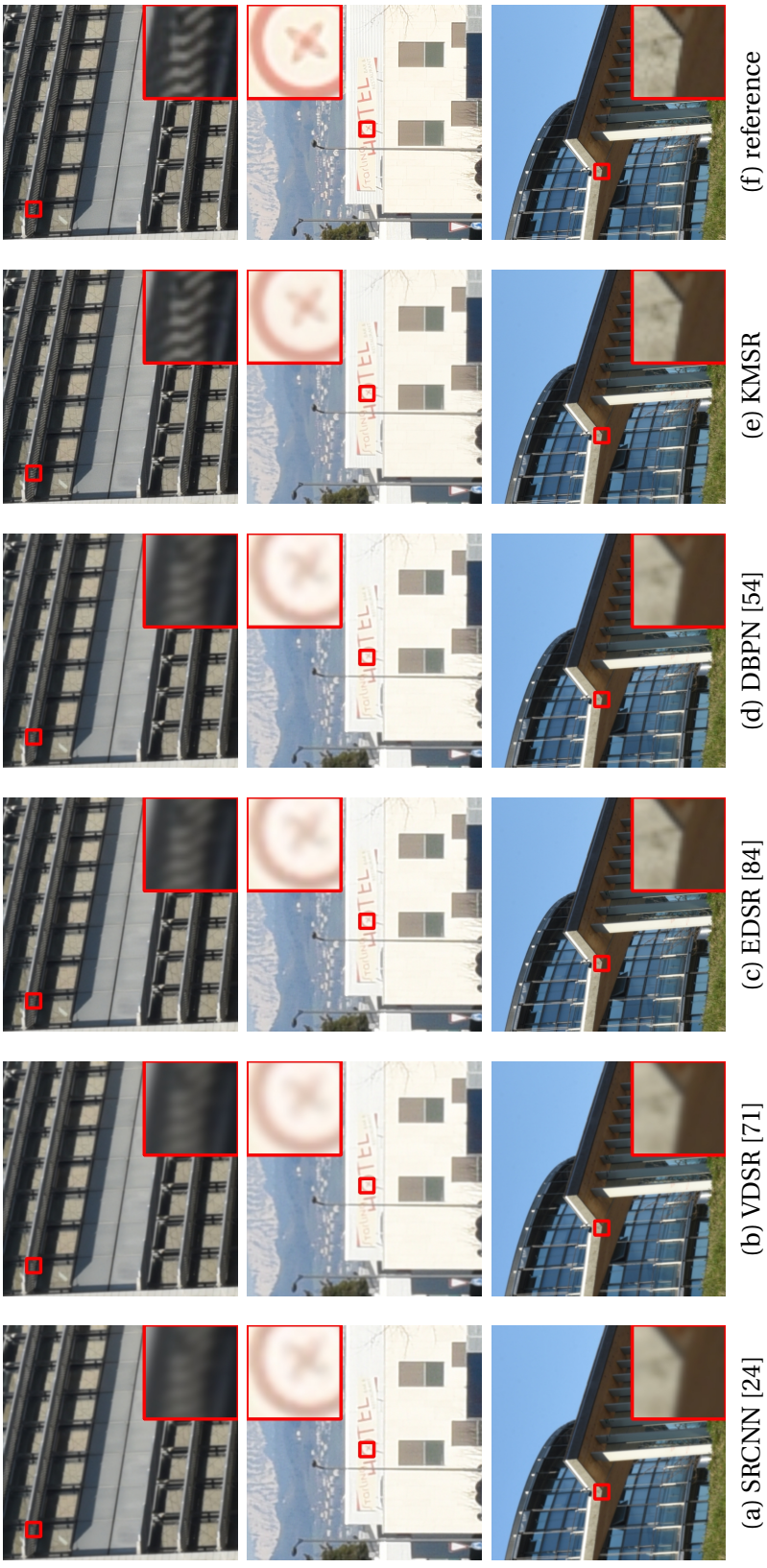


Figure 4.14 – Interface for the psychophysical experiment to validate the proposed KMSR.

4.3.6 Experiments on Zoom-in Super-Resolution

To further verify the performance of the proposed KMSR, we conduct experiments on images captured with the same camera, but at different focal lengths. We use a Nikon AF-S 24-70mm zoom lens to collect three pairs of images. The 35mm focal length photo serves as LR image, and the photo taken at the same position with the 70mm focal length serves as the reference HR image for a $\times 2$ SR of the LR image. We capture all the photos with a small aperture (f/22) to minimize the depth-of-field differences. We crop patches of size 250×250 from the LR image



(a) SRCNN [24] (b) VDSR [71] (c) EDSR [84] (d) DBPN [54] (e) KMSR (f) reference

Figure 4.15 – Qualitative comparison of different SR networks on ×2 zoom-in. (a)-(e) The SR results on the LR image taken with a 35mm focal length. (f) the reference HR image taken with a 70mm focal length.

and patches of size 500 × 500 from the reference HR image.

A slight misalignment is unavoidable because of focal length variations in the center of projection when the lens zooms in and out. We align the bicubic-upscaled 35mm "LR" image with the "zoomed-in" ground-truth 70mm image by applying a grid search in horizontal and vertical shifts (within 100 pixels) as well as a stretching (range between 0.9 to 1.1).

Table 4.4 shows the results of different SR networks on this zoom-in task. KMSR outperforms all other SR networks by a large margin both in PSNR and SSIM. A visual result is shown in Figure 4.15. KMSR is capable of generating a sharper image than the other SR networks.

Method	PSNR	SSIM
bicubic interpolation	26.93	0.79
SRCNN [24]	27.07	0.80
VDSR [71]	27.11	0.80
EDSR [84]	27.45	0.81
DBPN [54]	27.42	0.81
KMSR	29.13	0.84

Table 4.4 – Average PSNR and SSIM of different SR networks on the ×2 zoom-in dataset. The evaluation is performed only on the luminance channel to alleviate the effect of bias caused by the color variations of the two images. We highlight the best results in red color and the second best in blue color.

4.3.7 Ablation studies

To demonstrate the effectiveness of using realistic kernels and also to show the precision of the kernel estimation algorithm [99] that we use, we train and test another version of the proposed network, KMSR$_{A1}$, without collecting the realistic kernels. In building the kernel pool $\mathbb{K}'_{A1}$ for KMSR$_{A1}$, we use the bicubic-downsampled HR images as LR images, *i.e.* we estimate the blur kernels k'_{A1} on the bicubic-downsampled, and bicubic-upscaled coarse HR images $I_{A1}^{LR'}$. We then follow the same procedure as KMSR. We train a GAN on $\mathbb{K}'_{A1}$ and generate the larger kernel pool $\mathbb{K}^{+}_{A1}$ used to train KMSR$_{A1}$. We test KMSR$_{A1}$ on different experimental settings, the quantitative results are shown in Table 4.5. For the Gaussian and realistic kernels, KMSR$_{A1}$ achieves comparative results with the state-of-the-art SR networks (see Table 4.1), which implies that KMSR$_{A1}$ is capable of learning the mapping from bicubic-downsampled LR images to HR images. The results also shows that we achieve significant performance gains with KMSR that is trained with the realistic kernels of $\mathbb{K}^{+}$ (last column in Table 4.1).

To test the contribution of the GAN in improving generalization, we trained KMSR$_{A2}$, which is KMSR trained only on the 1,000 estimated kernels, KMSR$_{A2*}$ trained on 2,000 estimated kernels and KMSR$_{A3}$, which is KMSR without using the GAN but with simple data augmentation to expand the kernel pool. In this case, KMSR$_{A2}$ and KMSR$_{A2*}$ are only trained on $\mathbb{K}'_{A2}$ and $\mathbb{K}'_{A2*}$ which contains only the original estimated kernels, and KMSR$_{A3}$ is trained

on $\mathbb{K}'_{A3}$ which contains the k' in $\mathbb{K}'_{A2}$ and their rotated and flipped versions. We also trained KMSR$_{A4}$, where we use 2,000 generated kernels. Results are shown in Table 4.5. Marginal gains are obtained by using more kernels or applying simple augmentation. For example, on the realistic kernel testing set, KMSR$_{A3}$ that is trained with simple augmentation results in a PSNR value of 27.10dB instead of 26.98dB, and KMSR trained with GAN augmentation results in a PSNR value of 27.52dB. On average, KMSR obtains 0.5dB improvements on KMSR$_{A3}$. The results on realistic kernels other than bicubic kernel are further improved by using more generated kernels in the training of KMSR$_{A4}$, leading us to believe that using a GAN to augment the kernel pool results in a more diverse representation than simple data augmentation. This further validates the effectiveness of incorporating a GAN in order to augment the realistic kernel-pool.

Methods	KMSR$_{A1}$	KMSR$_{A2}$	KMSR$_{A2*}$	KMSR$_{A3}$	KMSR	KMSR$_{A4}$
#kernels used	1000+1000	1000	2000	1000×8	1000+1000	1000+2000
Blur kernel						
bicubic	33.66	33.26	33.30	33.28	33.52	33.46
$g_{1.25}$	26.47	27.31	27.48	27.42	27.94	27.98
$g_{1.6}$	25.62	26.87	26.99	27.02	27.63	27.75
$g_{1.7}$	25.28	26.79	26.94	26.90	27.15	27.14
realistic	25.29	26.98	27.15	27.10	27.52	27.69

Table 4.5 – Evaluation of KMSR in terms of PSNR scores on ×2 SR in different training setting. We highlight the best results in red color and the second best in blue color.

4.4 Conclusion and Discussion

In this chapter, to handle to problem of mismatch kernels in generating training pairs for SR, we propose to improve the performance of CNN-based SR networks on real LR images by modeling realistic blur kernels. In contrast to existing methods that use a bicubic kernel in the imaging model to obtain LR training images, we generate the SR training dataset by employing a set of realistic blur kernels estimated from real photographs. We further augment the blur kernel pool by training a GAN to output additional realistic kernels. Our KMSR is able to produce visually plausible HR images, demonstrated by both quantitative metrics, qualitative comparisons, and a psychovisual experiment. Our KMSR offers a feasible solution toward practical CNN-based SR on real photographs.

In our SR system, we always assume that the images are noise-free or a denoising step is preformed beforehand. However it is not the case in some real world applications. In our experiments, we notice that applying SR on noisy LR images can cause some unpleasant ringing artifacts as the SR algorithms also magnify the noise. Thus it is important to address noise in the SR system, which will be discussed in the next chapter.

5 Joint Denoising and Super-Resolution

5.1 Introduction

For modularity and simplicity, in the literature and the previous chapters of this thesis, images are always assumed to be noise-free, or a denoising step is applied before SR. However, noise is unvoidable in the image acquisition pipeline, and most of the denoising algorithms cannot reconstruct a perfect noise-free image. This unfortunately might lead to error accumulation. Most denoising algorithms not only eliminate noise, but also smooth out the high-frequency detail and texture in the image, causing downstream tasks such as SR to magnify over-smoothing, eventually affecting the image quality [104]. Alternatively, the SR networks trained with only noise-free images are proven to be highly vulnerable to small noise [20]. Both noise and limited resolution are inherent in real-world photographs, and the results of applying SR on top of the denoised image will be affected by the error from the denoising algorithm. Thus it is valid to jointly study both problems.

Both denoising and SR are very important in fluorescence microscopy. The microscopy images are often taken under extremely low-light conditions and thus suffer from very high noise. Due to the diffraction on the capture device, the images also have limited resolution. Although there are microscopy techniques, such as structured-illumination microscopy (SIM) [49] that can improve the resolution and the quality of the images, they require multiple captures with several tuned parameters to achieve good-quality images. Multiple or high-light capture of these techniques may cause photo-bleach and damage the samples, the cells will be affected and possibly killed. Moreover, for live cells, it is impossible to increase the exposure time or take multiple capture as the sample is not static. Therefore, joint denoising and SR is of great importance in microscopy.

In this chapter, we address these problems by collecting a joint denoising and SR dataset for benchmarking the denoising and SR algorithms by means of real microscopy images. This dataset, to the best of our knowledge, is the first JDSR dataset. We leverage microscopy equipment and techniques to acquire data satisfying the described requirements above. Our noisy LR images are captured using widefield imaging of human cells. We capture a total of 400

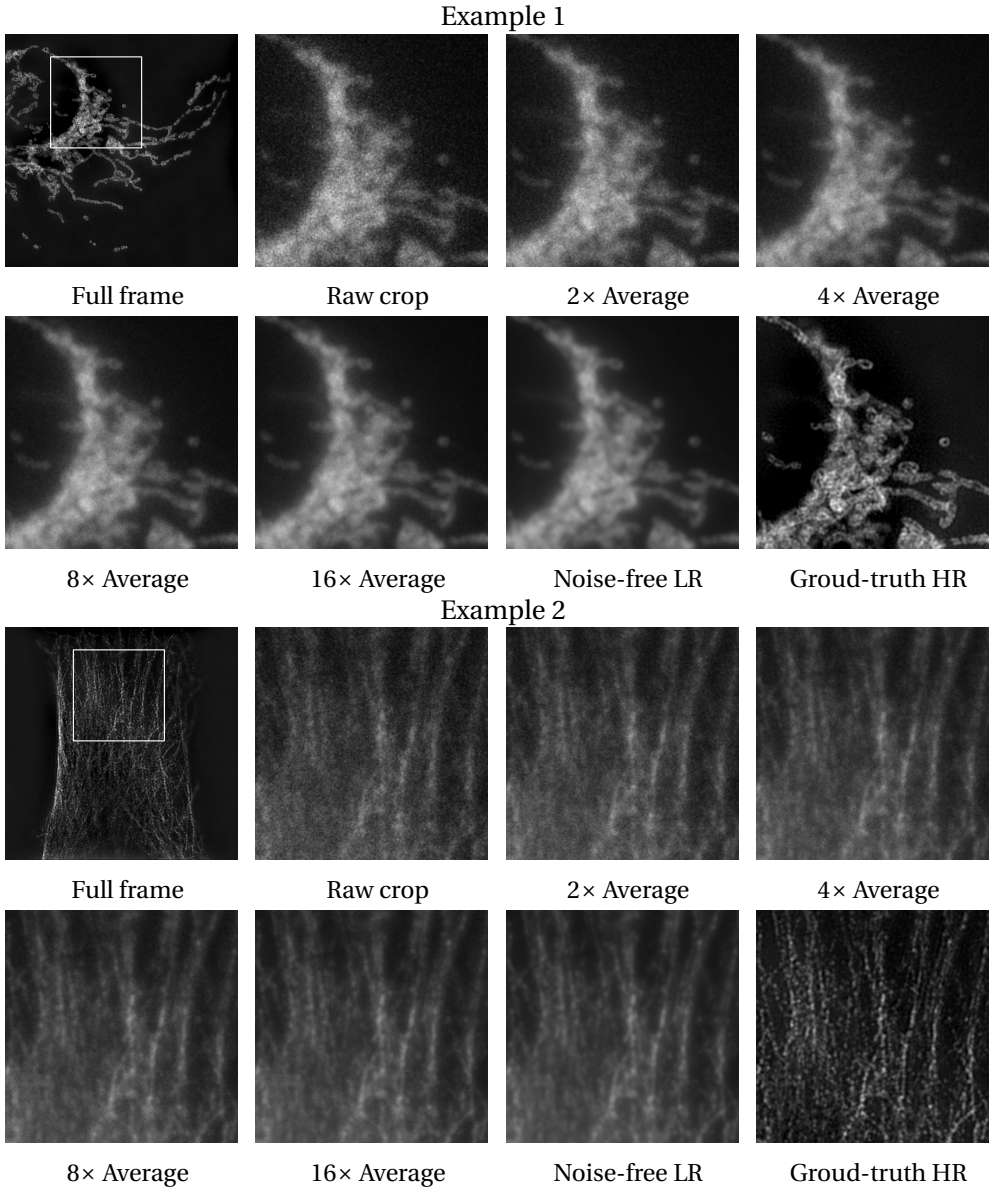


Figure 5.1 – Example of image sets in the proposed W2S. We obtain 5 LR images with different noise levels by either taking a single raw image or averaging different numbers of raw images. The more images we average, the lower the noise level as shown. The noise-free LR images are the average of 400 raw images, and the HR images are obtained using SIM [49]. Gamma correction is applied for better visualization.

replica raw images per field of view, 8 times more than a recent denoising-only dataset using similar technology [143]. We average several of the LR images to obtain images with different noise levels, and all of the 400 replicas to obtain the noise-free LR image. We leverage the SIM technique [49] to obtain high-quality HR images. Using this acquisition setup, we create **W**idefield**2S**IM (W2S), which consists of 360 sets of LR and HR image pairs, with different fields of view, and acquisition wavelengths. Visual examples of the images in W2S are shown

in Figure 5.1.

Using W2S, we benchmark different approaches for solving denoising and SR restoration. We compare the sequential use of different denoisers and SR methods, the direct use of an SR method on a noisy LR image, and the use of SR methods on the noise-free LR images of our dataset for reference. We additionally evaluate the performance of retraining SR networks on our JDSR dataset. Results show a significant drop of $8dB$ to $14dB$ in the performance of SR networks when the low-resolution (LR) input is noisy compared to it being noise-free. We also find that the costly consecutive application of denoising and SR is better. It is, however, not as performing in terms of PSNR and perceptual texture reconstruction as training a single model on the joint denoising and SR task, due to the accumulation of error. The best results are obtained with a light-weight model, trained with a texture loss that exploits the second-order statistics of feature maps.

In summary, the contribution of this chapter includes[1]:

- We create the first real joint denoising and SR dataset, W2S, containing noisy images with 5 noise levels, noise-free LR images, and the corresponding high-quality HR images.
- We analyze our dataset by comparing the noise magnitude, the blur kernel, and the power spectral density (PSD) of our images, to those of existing denoising and SR datasets.
- We benchmark state-of-the-art denoising and SR algorithms on W2S, by evaluating different settings and on different noise levels.
- We train a single model for joint denoising and SR with a texture loss and show it achieves the best reconstruction error and perceptual results.

5.2 Joint Denoising and Super-Resolution Dataset

Several datasets have been commonly used for benchmarking SR and denoising, including Set5 [6], Set14 [136], BSD300 [90], Urban100 [63], Manga109 [91], DIV2K [122], *etc.* Although these datasets contain various content ranging from nature photographs to comics images, their evaluation is based on a synthetic setup. *i.e.*, the noisy inputs are generated by adding Gaussian noise for the denoising algorithms, and the LR images are generated by downsampling the blurred HR images for SR methods. These assumed degradation models deviate from the degradation in real-world scenario [17]. To better capture the real-world characteristics and to evaluate denoising and SR methods under real scenarios, recent attempts have been made on capturing real-world denoising and SR datasets, such as DND [103], SSID [1], SR-RAW [140], and RealSR [10]. However, these datasets are designed only for denoising or SR. In addition, some of them also suffer from misalignment and color mismatch due to the inevitable perspective changes or lens distortion.

[1] This work was published in [149]

To build a joint denoising and SR dataset and to overcome the acquisition difficulties, we turn to microscopy as it allows us to capture LR-HR pairs without alignment or color mismatch problems. In this section, we describe the experimental setup that we use to acquire the sets of LR and HR images and present an analysis of our dataset covering noise levels, blur kernels, and power spectral density comparisons.

5.2.1 Structured-Illumination Microscopy

Structured-illumination microscopy (SIM) is a technique used in microscopy imaging that allows samples to be captured with a higher resolution than the one imposed by the physical limits of the imaging system [49]. Its operation is based on the interference principle of the Moiré effect. We use SIM to extend the resolution of standard widefield microscopy images. This allows us to obtain aligned LR and HR image pairs to create our dataset. Here, we provide an introduction to the principle of structured interference acquisition with 1D signals. The extension to higher dimensions follows the same principle [49].

We define $sig(t)$ to be the signal we want to acquire, where t represents a certain spatial dimension. In the Fourier domain, the corresponding signal $S(\omega)$ is not necessarily band-limited and can be non-zero for arbitrary frequencies ω. The impulse response of the capturing system is called its point spread function, and its Fourier transform is its optical transfer function (OTF) that we call $O(\omega)$. The resulting visible signal through that imaging system is given by $V(\omega) = O(\omega) \odot S(\omega)$, where $\odot$ is the element-wise multiplication. The OTF limits the captured content to a certain range of frequency components as $O(\omega) = 0 \ \forall \omega > \omega_c$, where ω_c is the cut-off frequency of the OTF. Therefore, only frequency components $\omega < \omega_c$ can be captured. By using a structured-illumination pattern, the frequency content can be manipulated. For instance, if we multiply the signal $s(t)$ with a cosine function of frequency ω_0, the captured signal becomes $s(t) \odot cos(\omega_0 t)$ and the corresponding frequency-domain equivalent is given by $\frac{1}{2}[S(\omega - \omega_0) + S(\omega + \omega_0)]$. Using the same imaging system, the visible signal becomes

$$V(\omega) = O(\omega) \odot \frac{1}{2}[S(\omega - \omega_0) + S(\omega + \omega_0)], \tag{5.1}$$

with $O(\omega)$ still equal to zero above its cut-off frequency ω_c. However, frequency components such that $\omega - \omega_0 < \omega_c$ can now be acquired, effectively pushing the cut-off to $\omega_c + \omega_0$, where ω_0 can be controlled by modifying the periodicity of the illumination pattern. In other words, higher frequencies that could not be visible to the imaging system can be shifted down to lower ones that lie within the observable range of that system. The shifted components can overlap in the frequency domain, and multiple shifted captures are needed to resolve the ambiguity and recover the true signals. SIM acquires 9 different structured-illumination images to perform an upscaling by a factor of two. In practice, the OTF is also not necessarily an ideal low-pass filter and a deconvolution post-processing step can be required.

Note that theoretically, applying nonlinear SIM can produce images of arbitrary resolution [50], illustrating SIM's potential of producing a higher upscaling-factor SR dataset.

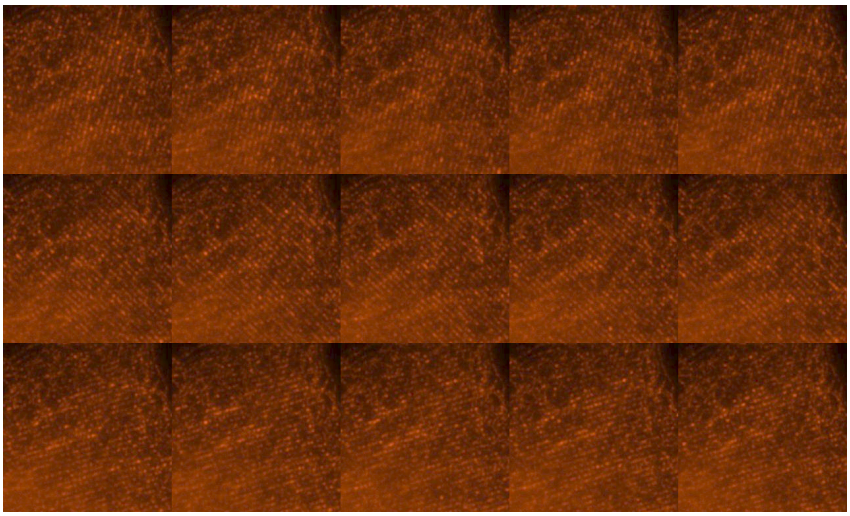
5.2.2 Data Acquisition

We capture the W2S dataset using widefield microscopy [127]. Images were acquired with a high-quality commercial fluorescence microscope and with real biological samples.

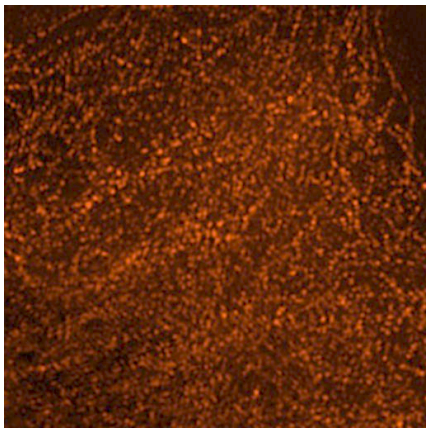
Widefield Images A time-lapse widefield of 400 images is acquired using a Nikon SIM setup (Eclipse T1) microscope. The microscope is fitted with a Plan Apochromat TIRF 100X, 1.49NA objective and an electron-multiplying charge-coupled device camera (IXON3; Andor Technology). The acquisition is taken with a 5ms exposure time using a 488nm Coherent sapphire laser at 0.37mW, a 5ms exposure time using a 561nm Cobolt Laser at 0.28mW, and a 5ms exposure using a Coherent 640nm Cobolt Laser at 0.26mW. All images have a resolution of 512×512. In total, we capture 120 different fields-of-view (FOVs), each FOV with 400 captures in 3 different wavelengths. All images are *raw*, *i.e.*, are linear with respect to focal plane illuminance. We generate different noise-level images by averaging 2, 4, 8, and 16 raw images of the same FOV. The larger the number of averaged raw images is, the lower the noise level. The noise-free LR image is estimated as the average of all 400 captures of a single FOV. Examples of images with different noise levels and the noise-free LR image are presented in Figure 5.1.

SIM Imaging The HR images are captured using SIM. The SIM images are captured using the same device (a microscope fitted with a Plan Apochromat TIRF 100X). We use the 3D SIM acquisition mode [51] (15 images per plane; five phases of three rotations) with a $70ms$ exposure time, using a $488nm$ Coherent sapphire laser at $0.20mW$; $30ms$ exposure time using a $561nm$ Cobolt Laser at $0.27mW$, and a $100ms$ exposure using a Coherent $640nm$ Cobolt Laser at $0.14mW$. Image reconstruction and processing are performed using the NIS-Elements software. The example of the acquired images and the reconstructed HR results are shown in Figure 5.2. The HR images' resolution is higher by a factor of 2, resulting in a resolution of 1024×1024.

Alignment As the acquisition process of LR and HR images is controlled through software that runs the microscope, and the device is well-stabilised throughout the capturing process, all images captured for a single FOV are very well aligned. However, we in addition verified the alignment of the images in post-processing. We use brute force matching based on the Hamming distance, with ORB [108] (FAST keypoint detector [107] and BRIEF descriptor [13]). The matching is carried out between our LR and HR pairs, after a bicubic downsampling of the HR images. We show two keypoint alignment examples in Figure 5.3. After discarding incorrectly-matched keypoints, we find that the estimated keypoint translation between image pairs is zero and the homography is equal to the identity matrix. We use this registration approach to validate that all image pairs are indeed well-aligned, and no human error occurred during the acquisition steps.



(a) SIM input captures

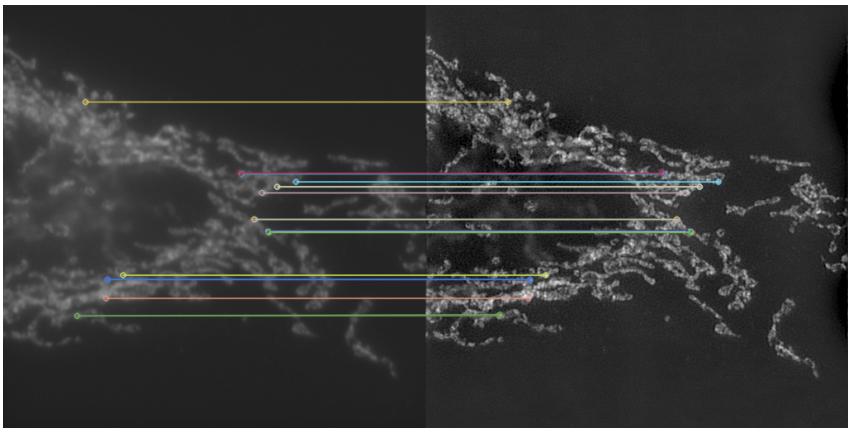


(b) SIM reconstruction

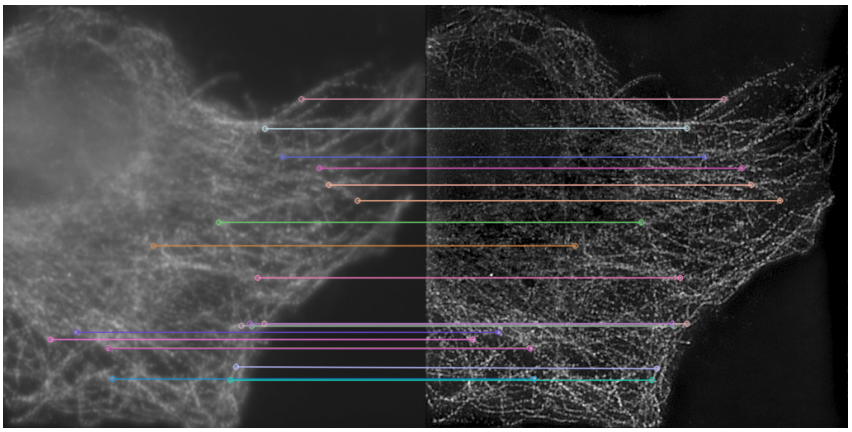
Figure 5.2 – Example FOV showing the different captured images in (a) that are given as input to the SIM method, and the reconstructed result of SIM in (b). Gamma correction is applied for better visualization.

5.2.3 Data Analysis

W2S includes 120 different FOVs, each FOV is captured in 3 channels (corresponding to the wavelengths of 488nm, 561nm and 640nm). As the texture of the cells are different and independent in different channels [110], the different channels can be counted as different images, thus resulting in 360 views. For each view, 1 HR image and 400 LR images are captured. We obtain LR images with different noise levels by averaging different numbers of images of the same FOV and the same channel. In summary, W2S provides 360 different sets of images, each image set includes LR images with 5 different noise levels (corresponding to 1, 2, 4, 8, and 16 averaged LR images), the corresponding noise-free LR image (averaged over 400 LR images) and the corresponding HR image acquired with SIM.



(a) Image 001, wavelength 640nm



(b) Image 007, wavelength 561nm

Figure 5.3 – Keypoint matching with a brute-force approach using Hamming distance, on the ORB detector and descriptor. In both figures, the left half is our ground-truth noise-free widefield image, and the right half is our SIM capture with a bicubic downsampling.

To quantitatively evaluate the difficulty of recovering the HR image from the noisy LR observation in W2S, we analyse the degradation model of how the LR observations are obtained from the HR image. We use the same degradation model as in Chapter 4, with an additional noise component that is assumed to be larger than 0,

$$\mathrm{I}^{noisyLR} = (\mathrm{I}^{HR} \otimes k) \downarrow_s + n, \tag{5.2}$$

where $\mathrm{I}^{noisyLR}$ and I^{HR} correspond, respectively, to the noisy LR observation and the HR image, $\circledast$ is the convolution operation, k is a blur kernel, $\downarrow_s$ is a downsampling operation with a factor of s, and n is the additive noise. Note that n is usually assumed to be 0 or follow a Gaussian distribution in most of the SR networks' imaging models, while it is not the case for our dataset. As the downsampling factor s is equal to the targeted SR factor, it is well defined for each dataset. We thus analyse the two unknown variables of the degradation model for W2S; the noise n and the blur kernel k.

Noise Estimation We use the noise modeling method in [39] to estimate the noise magnitude in raw images from W2S, from the denoising dataset FMD [143], and from the SR datasets RealSR [10] and City100 [17]. The approach of [39] models the noise as Poisson-Gaussian, where the measured noisy pixel intensity is given by $y = x + n_{Poisson}(x) + n_{Gaussian}$. x is the noise-free pixel intensity, n_G is zero-mean Gaussian noise, and $x + n_{Poission}(x)$ follows a Poisson distribution. This approach yields an estimate for the parameter of the Poisson distribution.

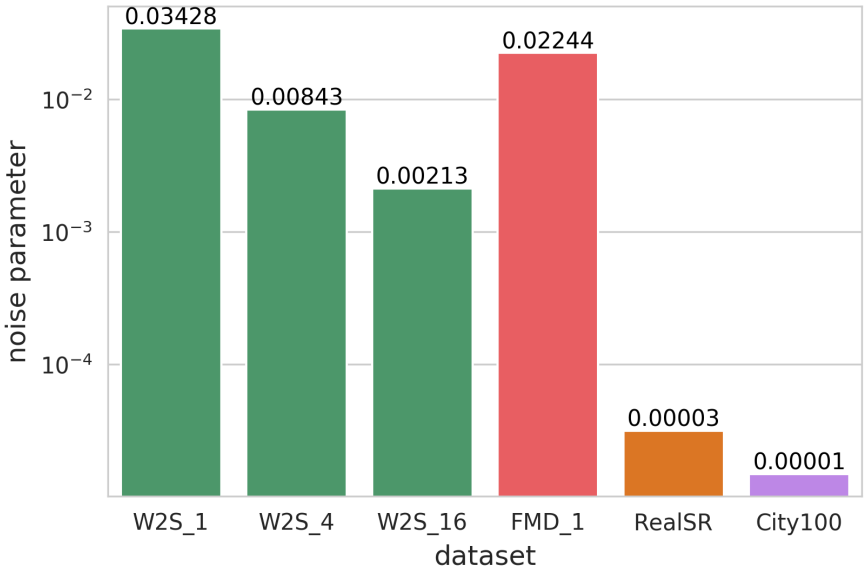


Figure 5.4 – Noise and kernel of different datasets. A higher noise indicate the the HR images of W2S are challenging to recover from the noisy LR.

We evaluate the Poisson parameter of the noisy images from the three noise levels (obtained by averaging 1, 4 and 16 images) of W2S, the raw noisy images of a microscopy denoising dataset FMD [143], and the LR images of the SR datasets for comparison. We show the mean of the estimated noise magnitude for the different datasets in Figure 5.4. We see that the raw noisy images of W2S have a high noise level, comparable to that of FMD. On the other hand, the estimated noise parameters of the SR datasets are almost zero, up to small imprecision, and are thus significantly lower than even the estimated noise magnitude of the LR images from the lowest noise level in W2S. Our evaluations demonstrates that additive noise, such as Poisson noise, is not taken into consideration in current state-of-the-art SR datasets. The learning-based SR methods using these datasets are consequently not tailored to deal with noisy inputs that are common in many practical applications, leading to potentially poor performance. In contrast, W2S contains images with high (and low) noise magnitude comparable to the noise magnitude of a state-of-the-art denoising dataset [143].

We compare W2S with other denoising datasets, DND [20], SSID [1], FMD [143] and other SR datasets, SR-RAW [140], City100 [17], RealSR [10]. The summary of the characteristics of the datasets is in Table 5.1. W2S contains 400 noisy images for each view, DND contains only 1,

SSID contains 150, and FMD, which also uses widefield imaging contains 50. W2S can thus provide a wide range of noise levels by averaging a varying number of images out of the 400. In addition, W2S provides LR and HR images that do not suffer from alignment problems and color mismatch.

	Dataset	RAW LR	RAW HR	Alignment	Color matching	# Noisy LR images	Main type of noise	# Ground-truth images
Denoising	DND [103]	✓	✗	Yes	Yes	1	Gaussian	50
	SSID [1]	✓	✗	Yes	Yes	150	Gaussian	200
	FMD [143]	✓	✗	Yes	Yes	50	Poisson	240
SR	SR-RAW [140]	✓	✗	No	No	0	None	500
	City100 [17]	✗	✗	Yes	Yes	0	None	100
	RealSR [10]	✗	✗	Yes	No	0	None	598
	W2S	✓	✓	Yes	Yes	400	Poisson	360

Table 5.1 – Characteristics of different state-of-the-art denoising and SR datasets. Our W2S contains raw noisy LR images, a noise-free raw LR image, and the corresponding HR image, thus enabling joint denoising and super-resolution evaluations.

Blur Kernel Estimation We estimate the blur kernel k shown in Eq. (5.2) as

$$k = \underset{k}{\operatorname{argmin}} ||\mathrm{I}^{noise-freeLR} \uparrow^{bic} - k \otimes \mathrm{I}^{HR}||_2^2, \tag{5.3}$$

where $I_{LR}^{noise-free} \uparrow^{bic}$ is the noise-free LR image upscaled using bicubic interpolation. We solve for k directly in the frequency domain using the Fast Fourier Transform [30].

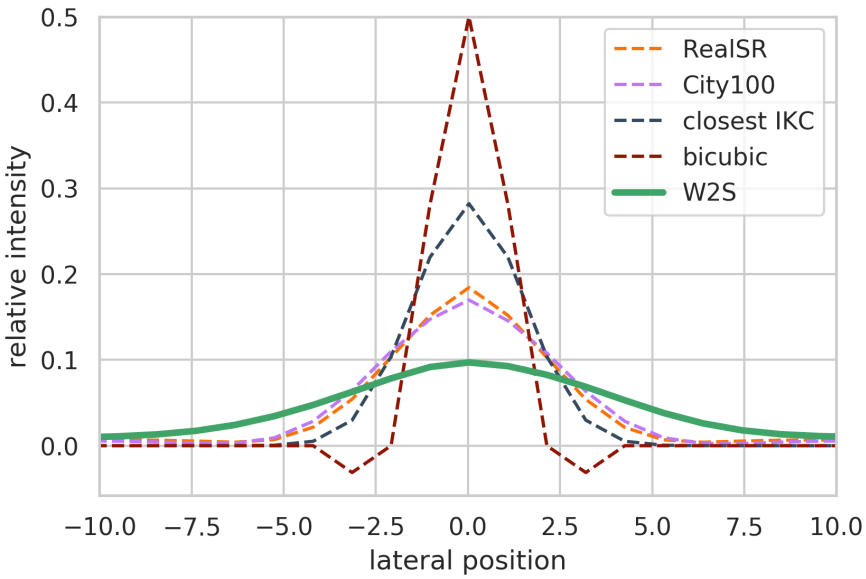


Figure 5.5 – Kernel estimation of different datasets. A wide kernel indicate that the HR images of W2S are challenging to recover from the noisy LR.

We use the aforementioned method to estimate the blur kernel k of W2S on the HR image

and the noise-free LR image. The estimated blur kernel is visualized in Figure 5.5. For the purpose of comparison, we show the estimated blur kernel from two SR datasets: RealSR [10] and City100 [17]. We also visualize the two other blur kernels: the MATLAB bicubic kernel that is commonly used in the synthetic SR datasets, and the Gaussian blur kernel with a sigma of 2.0 which is the largest kernel used by the state-of-the-art blind SR network [45] for the upscaling factor of 2. From the visualization we clearly see the bicubic kernel and Gaussian blur kernel that are commonly used in synthetic datasets are much different from the blur kernels of real captures. The blur kernel of W2S has a long tail compared to the blur kernels estimated from the other SR datasets, illustrating that more high-frequency information is removed for the LR images in W2S, and thus making the recovery of HR images from these LR images much more challenging.

Compared to the SR datasets, the LR and HR pairs in W2S are well-aligned during the capture process, and no further registration is needed. On the other hand, to obtain high-quality images, the SR datasets are captured under low ISO and contain almost zero noise. W2S contains LR images with different noise levels, which makes it a more comprehensive benchmark for testing under different imaging conditions. Moreover, as shown in Section 5.2.3, the estimated blur kernel of W2S is wider than that of other datasets, and hence averages pixels over a larger window, making W2S a more challenging dataset for SR.

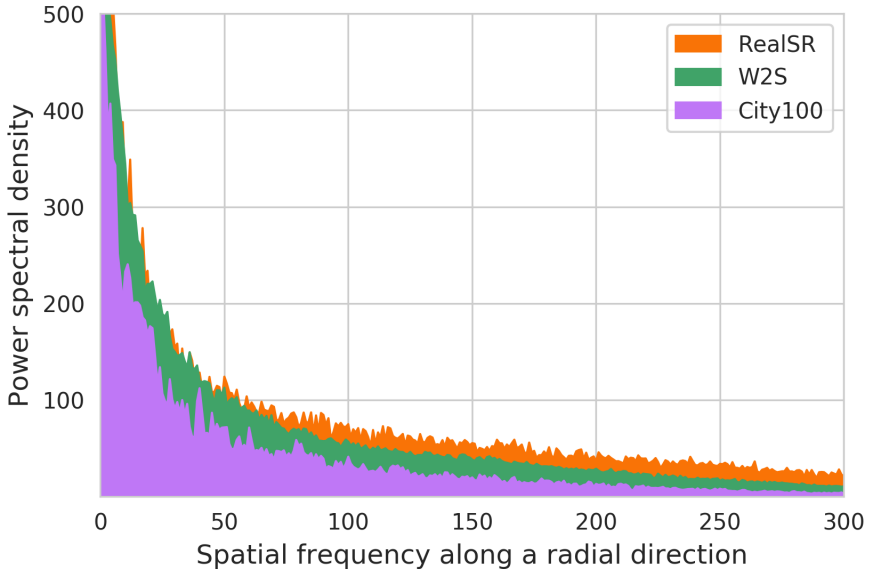


Figure 5.6 – Average PSDs

Figure 5.7 – Average PSDs of different datasets. The PSD plots show that although W2S is comprised of microscopy images, these images have a similar spatial frequency distribution as the natural image datasets RealSR [10] and City100[17].

Power Spectral Density To illustrate that the images of W2S share similar spatial frequency characteristics as other datasets, we plot the average power spectral density (PSD) for W2S and

two other SR datasets City100 [17] and RealSR [10], which contain natural images, in Figure 5.6. The PSD curve of W2S lies between the PSD curves of City100 [17] and RealSR [10], and shares a similar distribution that follows the well-known $1/f$ natural image power-law as a function of spatial frequency [9].

5.3 Benchmark

We benchmark on the sequential application of state-of-the-art denoising and SR algorithms on W2S. Note that we do not consider the inverse order, *i.e.*, applying SR on noisy images, as this will amplify the noise and cause a large decrease in PSNR as shown in the last row of Table 5.3. With current methods, it would be impossible for a subsequent denoiser to recover the real signals.

5.3.1 Setup

We split W2S into two disjoint training and test sets. The training set consists of 240 sets of LR and HR images, and the test set consists of 120 sets of images. There is no overlap between the training set and the test set. We retrain the learning-based methods on the training set. The evaluation of all methods is on the test set.

For denoising, we evaluate different approaches from both classical methods and deep-learning methods. We use a method tailored to address Poisson denoising, PURE-LET [88], and the classical Gaussian denoising methods EPLL [154] and BM3D [21]. The Gaussian denoisers are combined with the Anscombe variance-stabilization transform (VST) [89] to first modify the distribution of the image noise into a Gaussian distribution, denoise, and then invert the result back with the inverse VST. We estimate the noise magnitude using the method in [39], to be used as input for both the denoiser and for the VST when the latter is needed. We also use the state-of-the-art deep-learning methods DnCNN [138], MemNet [120], and RIDNet [4]. For a fair comparison with the traditional non-blind methods that are given a noise estimate, we train each of these denoising methods for every noise level separately, and test with the appropriate model per noise level. DnCNN and MemNet use a batch size of 128 and a starting learning rate of 10^{-3}, while RIDNet uses batches of 64 patches and a starting learning rate of 5×10^{-4}, all trained with the Adam optimizer [73] for 50 epochs, and with a ten-fold decrease in the learning rate after the milestone of 30 epochs. The same settings are used when training for the noise levels corresponding to an average of 1, 2, 4, 8, and 16 raw images.

We use five state-of-the-art SR networks for the benchmark: three pixel-wise distortion-based SR networks, RCAN [141], RDN [142], SRFBN [83], and two perceptually-optimized SR networks, EPSR [126] and ESRGAN [128]. The networks are trained for SR and the inputs are assumed to be noise-free, *i.e.*, they are trained to map from the noise-free LR images to the high-quality HR images. All these networks are trained using the same settings. The initial learning rate and loss function are set as the default value as presented in their papers. For

fair comparison, we use the same training setup for all models. For each training batch, 16 LR patches of size 64×64 are extracted. All models are trained using the Adam optimizer [73] for 50 epochs. The learning rate decreases by half every 10 epochs. Data augmentation is performed on the training images with a probability of 0.5, which are randomly rotated by 90 degrees, flipped horizontally, flipped vertically.

5.3.2 Results and Discussion

We apply the denoising algorithms on the noisy LR images, and calculate the PSNR and SSIM values between the denoised image and the corresponding noise-free LR image in the test set of W2S. The results of the 6 benchmarked denoising algorithms are shown in Table 5.2. Comparing to the previous results on DND and SSID image datasets, the algorithms achieve lower PSNR values on W2S [1, 4, 103], illustrating that W2S contains more challenging noisy images. Denoisers based on deep learning outperform the classical denoising methods for the highest noise level (*e.g.*, 1), however, BM3D achieves higher PSNR and SSIM when the noise level decreases (*e.g.*, 2, 4, 8, 16). The molecular structure of the cells contain sufficient texture repetition for the block matching of BM3D to perform well, and when the noise level is not very high, the generative power of deep-learning methods is not as solicited. However, one very interesting observation is that a higher PSNR results, in some cases, in unwanted smoothing in the form of a local filtering or averaging that incurs a loss of detail. Although the PSNR results of RIDNet are not the best (Table 5.2), when they are used downstream by the SR networks in Table 5.3, the RIDNet images achieve the best performance.

		Number of raw images averaged before denoising				
	Method	1	2	4	8	16
1-7	PURE-LET [88]	37.29/0.939	38.93/0.952	40.83/0.964	42.44/0.972	44.05/0.977
Denoisers	VST+EPLL [154]	37.51/0.951	39.56/0.965	41.59/0.973	43.36/0.980	45.16/0.984
	VST+BM3D [21]	37.80/0.955	39.78/0.967	41.78/0.975	43.52/0.980	45.30/0.985
	DnCNN† [138]	38.20/0.952	39.40/0.963	41.54/0.972	43.16/0.977	44.13/0.978
	MemNet† [120]	38.25/0.952	39.69/0.962	41.04/0.970	42.84/0.976	44.45/0.980
	RIDNet† [4]	37.82/0.949	38.99/0.958	41.47/0.970	42.07/0.975	44.95/0.982

Table 5.2 – PSNR (dB)/SSIM results on denoising the W2S test images. We benchmark a variety of standard methods, three classical ones (of which PURE-LET is designed for Poisson noise removal), and three deep learning based methods. The larger the number of averaged raw images is, the lower the noise level. †These learning-based methods are trained for each noise level separately, on our training set. A very interesting observation is that the best PSNR results (in red) do not necessarily give the best result after the downstream SR method, as we see in Table 5.3. We highlight the results under highest noise level with gray background for easier comparison with Table 5.3.

The SR networks are applied on the denoised results of the denoising algorithms mentioned above. We show the PSNR and SSIM results in Table 5.3 for the highest noise level (grey background column in 5.2). We also include the results of applying the SR networks on the

		Super-resolution networks				
		RCAN	RDN	SRFBN	EPSR	ESRGAN
Denoisers	PURE-LET	22.01/0.65	22.85/0.66	22.64/0.67	22.57/0.60	23.12/0.67
	VST+EPLL	24.16/0.71	24.71/0.71	24.21/0.71	23.76/0.64	24.13/0.70
	VST+BM3D	24.44/0.71	24.72/0.72	24.35/0.71	23.81/0.63	24.28/0.71
	DnCNN†	24.34/0.71	24.66/0.71	24.29/0.70	23.82/0.62	24.66/0.71
	MemNet†	24.45/0.71	24.71/0.71	24.51/0.70	23.83/0.63	24.66/0.70
	RIDNet†	24.52/0.71	24.76/0.71	24.62/0.71	23.86/0.62	24.41/0.70
	Noise-free LR	26.83/0.78	26.81/0.78	26.81/0.78	24.63/0.65	26.02/0.76
	Noisy LR	12.56/0.27	17.39/0.41	19.38/0.43	16.37/0.36	16.64/0.38

Table 5.3 – PSNR (dB)/SSIM results on the sequential application of denoising and SR methods on the W2S test images with the highest noise level, which correspond to the first column of Table 5.2. For each SR method, we highlight the best PSNR value in red. †The learning-based denoising methods are retrained for each noise level; the SR networks are trained to map the noise-free LR images to the high-quality HR images.

noise-free LR images. We see that although the three distortion-based SR networks (RCAN, RDN, SRFBN) have very similar performance when applied on the noise-free LR images (the largest difference in PSNR is $0.02dB$, between RCAN and SRFBN), we see larger difference when these networks are applied to the denoised images of the previous step. We notice that there is a significant drop in PSNR and SSIM when the SR networks receive the denoised LR images instead of the noise-free LR images, even though the difference between the denoised LR images and the noise-free LR images are small. For example, applying RDN on noise-free LR images results in a PSNR value of $26.81dB$, while the PSNR value of the same network applied to the denoised results of RIDNet on the lowest noise level is $25.85dB$ (shown in the first row, last column in Table 5.5). This illustrates the SR networks are strongly affected by noise in the inputs. Among all the distortion-based SR networks, RDN shows the most robustness as it outperforms all other networks in terms of PSNR when applied on denoised LR images. As mentioned above, another interesting observation is that although RIDNet results in lower PSNR than other networks for denoising at the highest noise level, RIDNet still provides a better input for the SR networks.

We present additional results of the sequential application of state-of-the-art denoisers and SR methods on low-resolution (LR) images with different noise levels on W2S. The results are shown in Table 5.4 The different noise levels correspond to a different number of averaged raw images. We note that there is no consistent and significantly-better denoiser across all SR methods and noise levels. Between SR models, the best is RDN but not with a large margin.

Qualitative results are given in Figure 5.8, where for each SR network we show the results for the denoising algorithm that achieves the highest PSNR value for the joint task. Notice that none of networks is able to produce results with detailed texture. As the denoising algorithms have removed some high-frequency signals as noise, the SR results from the distortion-based networks are blurry and many texture details are missing. Although the perception-based

			Super-resolution networks				
			RCAN	RDN	SRFBN	EPSR	ESRGAN
Denoisers on	2× average	PURE-LET	22.04/0.66	22.97/0.67	22.70/0.66	22.54/0.60	23.01/0.67
		VST+EPLL	24.56/0.72	25.04/0.73	24.57/0.71	23.92/0.64	24.46/0.71
		VST+BM3D	24.82/0.72	25.09/0.73	24.78/0.72	24.01/0.63	24.58/0.71
		DnCNN†	24.77/0.72	25.04/0.72	24.86/0.71	24.13/0.64	24.95/0.72
		MemNet†	24.78/0.72	25.03/0.72	24.87/0.71	24.11/0.64	24.96/0.71
		RIDNet†	25.01/0.73	25.08/0.73	25.06/0.73	24.11/0.64	24.72/0.71
Denoisers on	4× average	PURE-LET	23.21/0.70	24.07/0.71	23.62/0.70	23.28/0.63	23.60/0.69
		VST+EPLL	25.01/0.73	25.41/0.74	25.00/0.73	24.15/0.65	24.81/0.72
		VST+BM3D	25.21/0.74	25.46/0.74	25.19/0.73	24.23/0.64	24.91/0.73
		DnCNN†	25.18/0.73	25.39/0.74	25.29/0.73	24.31/0.64	25.23/0.73
		MemNet†	25.19/0.73	25.38/0.74	25.25/0.73	24.36/0.64	25.22/0.72
		RIDNet†	25.06/0.73	25.29/0.73	25.20/0.72	24.24/0.63	25.13/0.72
Denoisers on	8× average	PURE-LET	24.06/0.72	24.88/0.73	24.75/0.73	23.91/0.65	23.99/0.72
		VST+EPLL	25.40/0.74	25.74/0.75	25.40/0.74	24.35/0.65	25.14/0.73
		VST+BM3D	25.56/0.75	25.77/0.75	25.54/0.74	24.42/0.65	25.22/0.73
		DnCNN†	25.29/0.74	25.56/0.74	25.42/0.73	24.47/0.64	25.46/0.73
		MemNet†	25.46/0.74	25.65/0.74	25.57/0.73	24.52/0.65	25.42/0.73
		RIDNet†	25.45/0.74	25.63/0.74	25.58/0.74	24.59/0.65	25.34/0.73
Denoisers on	16× average	PURE-LET	24.90/0.74	25.52/0.75	25.49/0.75	24.34/0.67	24.63/0.74
		VST+EPLL	25.64/0.75	25.96/0.75	25.76/0.75	24.50/0.66	25.32/0.74
		VST+BM3D	25.72/0.75	25.96/0.75	25.83/0.75	24.56/0.65	25.36/0.74
		DnCNN†	25.64/0.75	25.87/0.75	25.85/0.75	24.72/0.66	25.60/0.74
		MemNet†	25.77/0.75	25.90/0.75	25.87/0.75	24.75/0.66	25.60/0.74
		RIDNet†	25.67/0.75	25.85/0.75	25.84/0.74	24.63/0.65	25.54/0.74

Table 5.4 – PSNR (dB)/SSIM results on the sequential application of denoising and SR methods on the W2S test images for different noise levels. †The learning-based denoising methods are retrained for each noise level, and the SR networks are trained to map the noise-free LR images to the high-quality HR images. For each SR method, we highlight the best PSNR and SSIM value in red. For each denoising method, we underline the best PSNR and SSIM value.

methods (EPSR and ESRGAN) are able to produce sharp results, they fail to reproduce faithful texture.

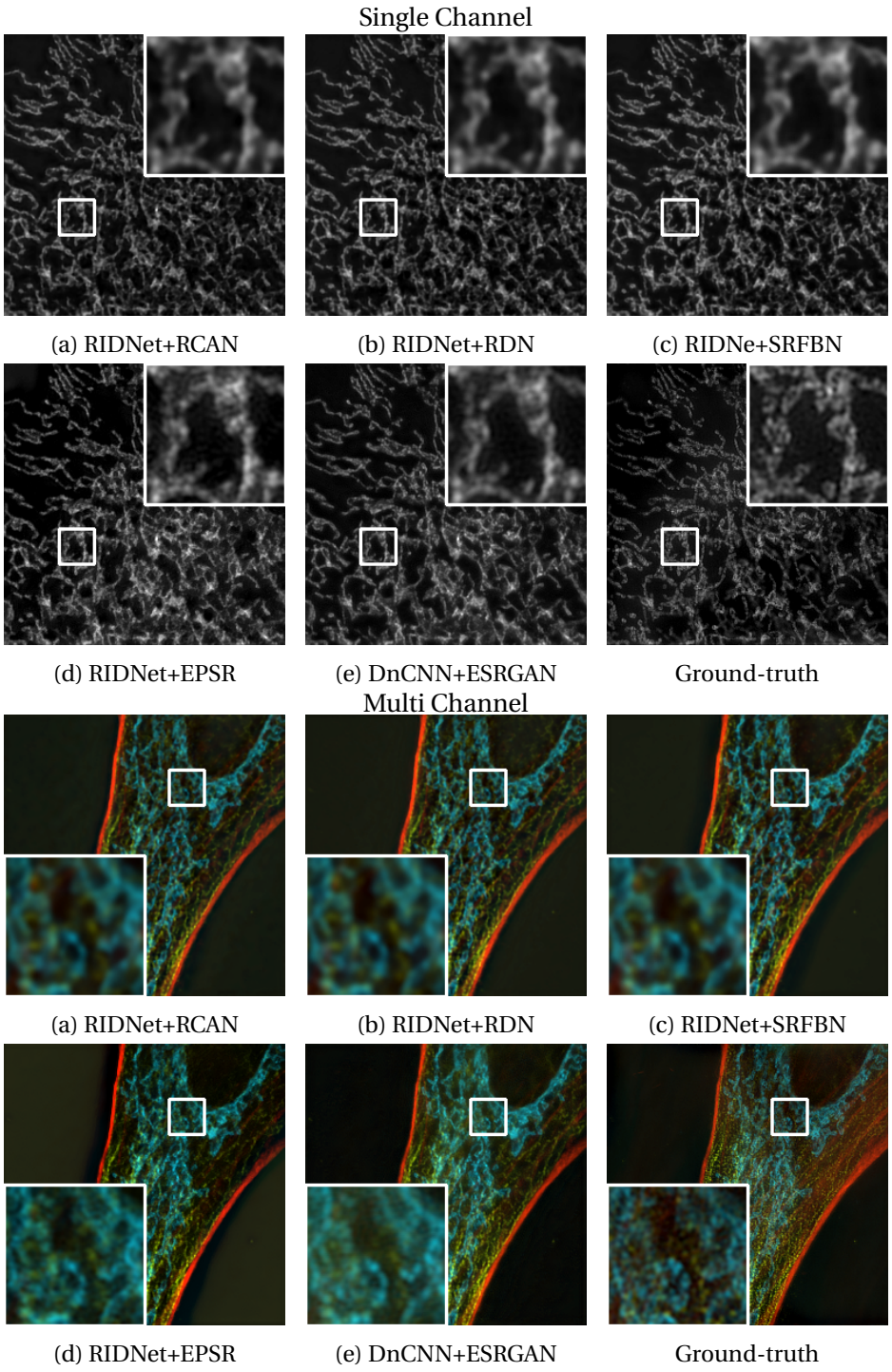


Figure 5.8 – Qualitative results on the sequential application of denoising and SR algorithms on the W2S test images with the highest noise level. The multi-channel images are formed by mapping the three single-channel images of different wavelengths to RGB. Gamma correction is applied for better visualization.

5.4 Joint Denoising and Super-Resolution

Our benchmark results in Section 5.3 show that the successive application of denoising and SR algorithms does not produce the highest-quality HR outputs. In this section, we demonstrate that it is more effective to train a joint denoising and SR model that directly transforms the noisy LR image into an HR image.

5.4.1 Texture Loss

To enable the network to better recover texture, we replace the GAN loss in the training with a texture loss that is similar to what Sajjadi *et al.*has proposed [109]. GAN loss usually results in SR networks producing realistic but fake textures that are different from the ground-truth and may result in a significant drop in PSNR [128]. Instead, we utilize a texture loss that exploits the features' second-order statistics to help the network to produce high-quality and real textures. This choice is motivated by the fact that traditional second-order descriptors have proven particularly effective for tasks such as texture recognition [53]. We leverage the second-order statistics of VGG features of the images to match the similarity of the texture. The texture loss is defined as

$$\mathscr{L}_{texture} = ||\text{Cov}(\phi(\text{I}^{est})) - \text{Cov}(\phi(\text{I}^{HR}))||_2^2, \tag{5.4}$$

where I^{est} is the estimated result from the network for joint denoising and SR, I^{HR} is the ground-truth HR image, $\phi(\cdot)$ is a neural feature space, and $\text{Cov}(\cdot)$ computes the covariance. We follow the implementation of MPN-CONV [81] for the forward and backward feature covariance calculation.

5.4.2 Training Setup

For joint denoising and SR, we adopt a 16-layer Residual in Residual Dense Block (RRDB) architecture. RRDB architecture is proposed by Wang *et al.* [128], it combines multi-level residual network and dense connections and has shown its efficiency in SR training. We use one Pixelshuffle up-convolution layer [112] in the network for mapping the features to the HR image.

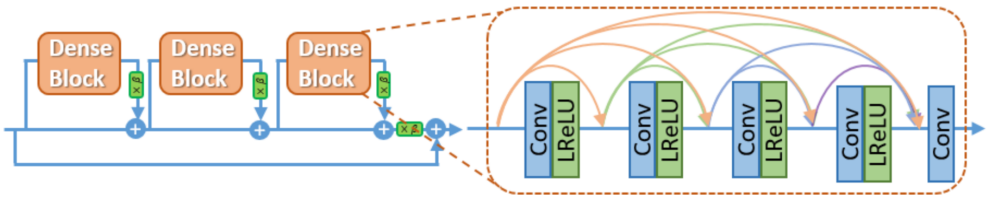


Figure 5.9 – Architecrue of RRDB [128].

In order to improve the visual quality of the resulting image, we also incorporate perceptual

loss into the training objective

$$\mathscr{L}_{perceptual} = ||\phi(\mathrm{I}^{est}) - \phi(\mathrm{I}^{HR})||_2^2. \tag{5.5}$$

Our final loss function is

$$\mathscr{L} = \mathscr{L}_1 + \alpha \cdot \mathscr{L}_{perceptual} + \beta \cdot \mathscr{L}_{texture}, \tag{5.6}$$

where $\mathscr{L}_1$ represents the $\ell 1$ loss between the estimated image and the ground-truth. We empirically set $\alpha = 0.05$ and $\beta = 0.05$. For the neural feature space, we use a pre-trained 19-layer VGG [115]. For the perceptual loss, we take the features from 'conv1_2', 'conv2_2' ,'conv3_4', 'conv4_4' and 'conv5_4'. For the texture loss, we take the features from 'conv4_4'.

We follow the same training setup as the experiments in Section 5.3. For comparison, we also train RDN [142] and ESRGAN [128] on joint denoising and SR.

5.4.3 Results and Discussion

Method	Number of raw images averaged				#Parameters
	1	2	4	8	
RIDNet†+RDN‡	24.76/0.712	24.98/0.714	25.29/0.736	25.63/0.741	1.5M+22M
DnCNN†+ESRGAN‡	24.66/0.705	24.95/0.716	25.23/0.726	25.46/0.733	0.5M+16M
JDSR-ESRGAN*	24.69/0.707	24.96/0.715	25.27/0.724	25.51/0.735	16M
JDSR-RDN*	25.07/0.707	25.16/0.717	25.57/0.732	25.95/0.752	22M
Ours*	25.17/0.713	25.28/0.721	25.61/0.737	25.89/0.759	11M

Table 5.5 – Joint denoising and SR PSNR (dB)/SSIM results on the W2S test set. †The denoising networks are separately retrained per noise level. ‡The SR networks are trained to map noise-free LR images to HR images. *The networks trained for joint denoising and SR are also retrained per noise level. For each noise level, we show the best PSNR and SSIM value in red.

The quantitative results of different methods on different noise levels are reported in Table 5.5. The results indicate that compared to the sequential application of denoising and SR, a single network trained on joint denoising and SR is more effective even though it contains less parameters. GAN-based methods generate fake textures and lead to low PSNR and SSIM scores. Our model, trained with texture loss, is able to outperform RDN and effectively recover high-fidelity texture information even when high noise levels are present in the LR inputs. We show the qualitative results of joint denoising and SR on the highest noise level (which corresponds to the first column of Table 5.2) in Figure 5.10. We see that other networks have difficulties to recover the shape of the cells in the presence of noise, and that our method trained with texture loss is able to generate a higher-quality HR image with faithful texture.

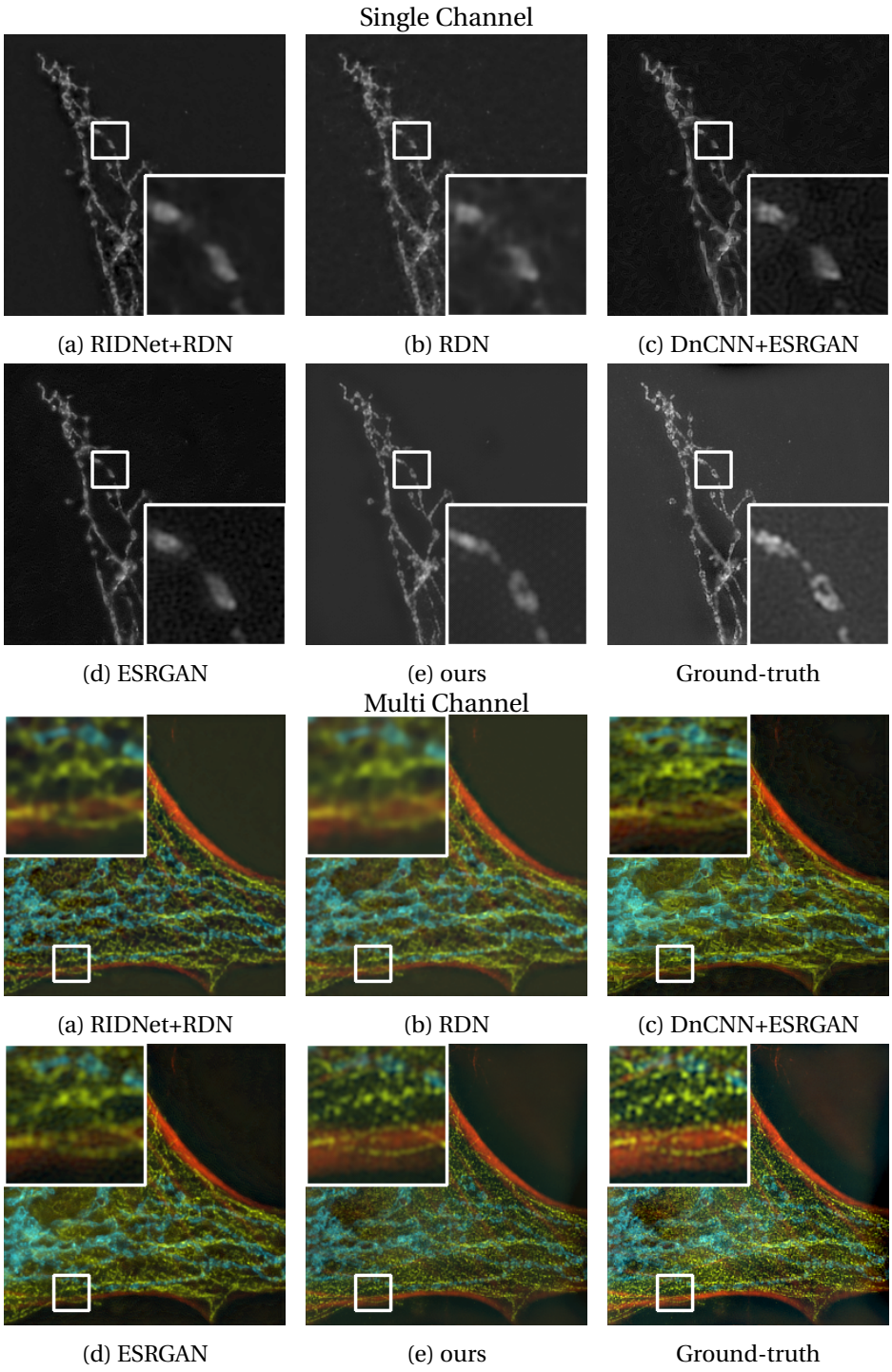


Figure 5.10 – Qualitative results of denoising and SR on the W2S test images with the highest noise level. The multi-channel images are formed by mapping the three single-channel images of different wavelengths to RGB. Gamma correction is applied for better visualization.

5.5 Conclusion

We propose the first joint denoising and SR dataset: **W**idefield**2S**IM. We use image averaging to obtain LR images with different noise levels and the noise-free LR. The HR images are obtained through the SIM technique. With W2S, we benchmark the combination of various denoising and SR methods. Our results show that SR networks are very sensitive to noise, and that the consecutive application of two approaches is sub-optimal and suffers from the accumulation of errors from both stages. Our results also show the networks benefit from joint optimization for denoising and SR. We train a single network for joint denoising and SR, with a texture loss to enable the network to reproduce faithful texture.

Although W2S contains microscopy images, we have shown that it shares similar spatial frequency characteristics as datasets capture in natural environments. We hypothesize that the observations and conclusions we gain from the experiments on W2S also apply to natural images.

6 Spectral Image Super-Resolution

6.1 Introduction

The methods presented in the previous chapters focus on SR for images captured by conventional RGB color cameras. Spectral cameras that acquire images with a CMOS array have more spatial resolution constraints. This is due to the fact that spectral image sensors need to cover a larger number of spectral bands than the three bands that the RGB image sensors capture. Spectral image processing has become an important field in many computer vision tasks, such as medical diagnosis, remote sensing, material detection, food inspection, *etc*. However, as mentioned beforehand, capturing spectral data is difficult due to the limitations of the imaging technology. One way to obtain spectral data is to apply scanning in the spectral domain to acquire the full spectrum. This acquisition process is time consuming and the equipment is expensive. An alternative way is to fill in the missing band information with the guidance of existing information. This problem is referred to as spectral reconstruction or spectral SR.

In this thesis, we address two real-world situations in spectral image SR: (a) *multi-scale* spectral image SR: several LR spectral images with different scale factors are available for spectral image SR; (b) *multi-modal* spectral image SR: a guided HR color image is provided in addition to the LR spectral data. Both situations are challenging as only a small number of images are provided due to the difficulties in capturing the dataset.

For the first situation, we use residual learning to reconstruct the residuals between the LR and HR images, rather than learning how to rebuild the HR image from LR. Our assumption is that learning the residual mapping is much easier than learning the original HR image. Furthermore, several image restoration methods such as VDSR [71], DnCNN [138], and DWSR [47] use residual connections from the input to the output and reduce their training time through faster convergence. By combining the image completion upscaling method with residual learning, we build a model suited for multi-scale image SR.

One often can obtain a high spatial resolution panchromatic image accompanying the multi-

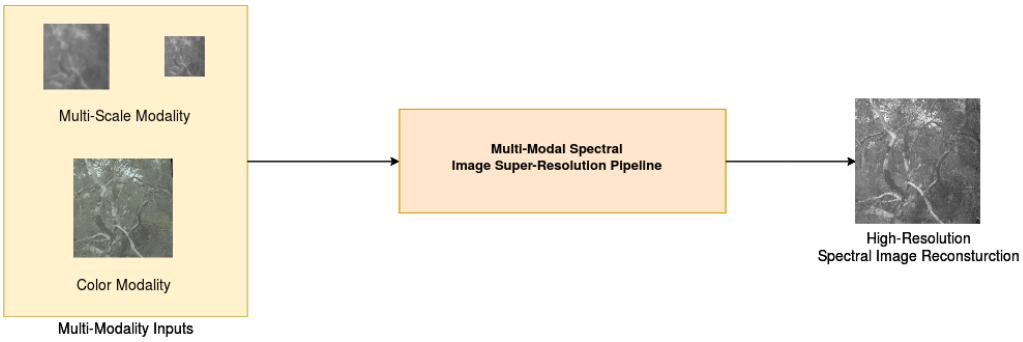


Figure 6.1 – Our proposed framework for spectral image SR, which is able to reconstrcut high-quality HR spectral images by taking advantage of multi-modal data consisting of multi-scale spectral images and color images.

spectral low resolution image. The fusion of both images allows obtaining both high spatial and spectral resolution images. This is helpful for many remote-sensing applications like agriculture, earth exploration, and astronomy. We make use of a 3-color RGB high spatial resolution image to guide the SR of the 14-band LR spectral images in the second part of our experiments. Thus, we design our pipeline to incorporate the guiding images to achieve higher performance on top of our previous residual network results.

In summary, we propose an efficient framework for multi-modal spectral image SR shown in Figure 6.1. Our contributions in this chapter are[1]:

- We build a residual learning network that is suitable for spectral SR due to the sparse nature of the problem.
- We design a data pre-processing approach that can fuse multi-scale images in order to create an upscaled input image to the network. This approach combines the information from multi-scale modalities with an image completion algorithm to provide a candidate image to the network that performs better than the typical bicubic interpolation.
- We build a two-stage pipeline for guided SR under consideration that very few data samples containing guiding information are available. The framework resembles transfer learning, as it allows to transfer information learned using one modality to another to compensate the lack of data.

6.2 Method

6.2.1 Imaging Model for Spectral Image Super-Resolution

The goal of spectral image SR is to recover an HR spectral image $I^{HR}_{spectral}$ of resolution $h \times w$ from one or a set of LR spectral image $I^{LRs}_{spectral}$ of resolution $\frac{h}{s} \times \frac{w}{s}$. For some real-world

[1] This work was published in [77]

applications, an HR color image I^{HR}_{color} that has the same resolution as the HR spectral image is available. The HR spectral image and its LR version share a similar relationship with HR and LR of RGB images:

$$\mathrm{I}^{LRs}_{spectral} = (\mathrm{I}^{HR}_{spectral} \otimes k) \downarrow_s, \tag{6.1}$$

where $\mathrm{I}^{LRs}_{spectral}$ denotes the LR spectral image with a scale factor of s. $\mathrm{I}^{HR}_{spectral}$ is the HR spectral image, k is the blur kernel, and $\downarrow_s$ is the downsampling operation with a factor of s.

The imaging model for the HR color image I^{HR}_{color} is:

$$\mathrm{I}^{HR}_{color}(c) = \sum_{\lambda \in V} S_c(\lambda) I^{HR}_{spectral}(\lambda), \tag{6.2}$$

where $S_c(\lambda)$ defines the spectral sensitivity of the color channel $c \in R, G, B$, and $\mathrm{I}^{HR}_{spectral}(\lambda)$ are the channels of wavelength λ of the HR spectral image.

Following the PIRM 2018 Spectral Image SR Challenge [114], we assume two settings for spectral image SR:

Task-I: Spectral Image Super-Resolution focuses on the problem of super-resolving the spatial resolution of spectral images. The goal is to recover an HR spectral image $\mathrm{I}^{HR}_{spectral}$ given a twice-downsampled version ($\mathrm{I}^{LR2}_{spectral}$) and a thrice-downsampled version ($\mathrm{I}^{LR3}_{spectral}$). Here, the blur kernel k in Equation 6.1 is omitted for simplicity.

Task-II: Color-Guided Spectral Image Super-Resolution aims at leveraging the link between spectral and color images of the scene to facilitate the use of on-sensor filter arrays. Thus, the computational objective of this task is to obtain spatially super-resolved spectral images from the twice- and thrice-downsampled LR spectral image with the guidance of a perfectly-aligned color image I^{HR}_{color} that has the same resolution as the targeted HR spectral image.

6.2.2 Residual Learning Framework

We propose a residual learning framework for spectral image SR as shown in Figure 6.2.

Similar to bicubic interpolation adopted in many SR algorithms [71, 24], we first upscale the LR spectral inputs $\mathrm{I}^{LR2}_{spectral}$ and $\mathrm{I}^{LR3}_{spectral}$, which are subsampled from the full resolution spectral image by a factor of 2 and 3. We use an image completion algorithm [2] on the multi-scale inputs to generate an HR spectral image candidate with the desired size. Then we train residual learning networks for spectral image SR.

For Task-I, Stage-I uses one 12-layer residual learning network to reconstruct HR results from the image candidate. These reconstructions are used to generate the solution for Task-I. In Task-II, we have less training data. So we design our solution to take advantage of Stage-I. Stage-II takes the concatenation of Stage-I's proposed output and the higher-resolution color image as inputs. It is trained on the small dataset of image pairs and refines Stage-I results

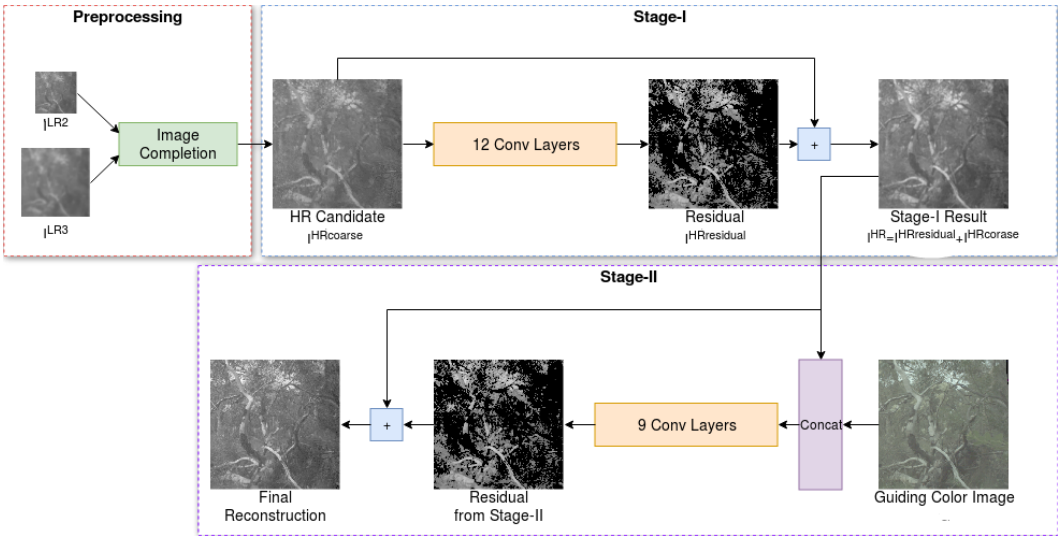


Figure 6.2 – Illustration of our proposed stacked residual learning framework for spectral image SR. It contains three steps: pre-processing, Stage-I, and Stage-II. Image completion is done in pre-processing to generate an HR candidate. Then Stage-I reconstruct the HR using a 12-layer residual learning network. Stage-II refines Stage-I results using guiding color image G through a 9-layer residual learning network.

through guiding color images.

6.2.3 Image Completion

$I^{LR2}_{spectral}$ and $I^{LR3}_{spectral}$ were both obtained by downsampling the original HR version using nearest-neighbor downsampling. Therefore, a large amount of pixel information is preserved, which means we can already recover part of the ground-truth immediately from the LR samples. In fact, we can recover $\frac{1}{4}$ of the data from $I^{LR2}_{spectral}$ and $\frac{1}{9}$ from $I^{LR3}_{spectral}$ by simply upscaling the image and setting the new pixels to black (unfilled). Together, $I^{LR2}_{spectral}$ and $I^{LR3}_{spectral}$ give us $\frac{1}{3}$ of the original image pixels. Figure. 6.3 shows how we recover the partial HR image, named $I^{HRpartial}_{spectral}$, from both LR examples.

Image completion is the task of completing an image with a percentage of pixels missing. This has a wide range of applications such as noise-removal, demosaicing, inpainting, artifact removal as well as image editing. One particular usage is image-scaling and SR. There have been multiple approaches to fill the missing parts of an image. One main category of methods relies on matrix completion [61, 82, 86]. While these methods are well suited for large number of retained pixels, they do not work when the input matrix has fully missing columns and rows such as ours. We also do not have many connected pixels to form patches, so patch-based methods [80, 119] are not suited.

The extreme image completion [2] method FAN is able to complete a 1% pixel image with

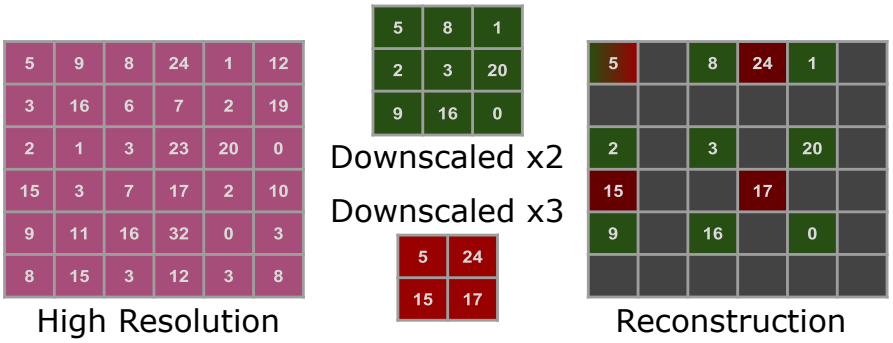
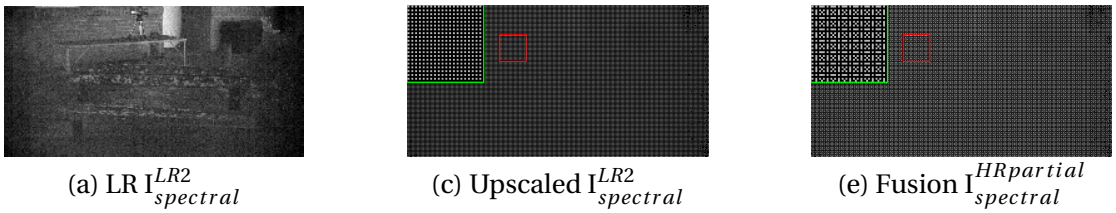
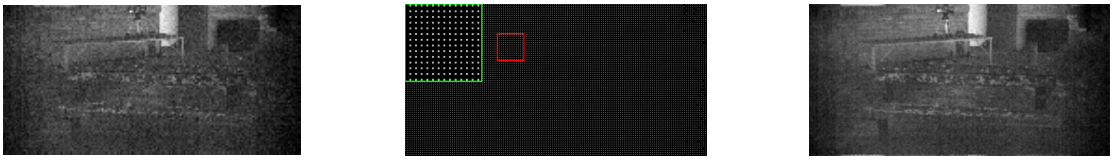


Figure 6.3 – Illustration of downscaling and upscaling.



(a) LR $I^{LR2}_{spectral}$ (c) Upscaled $I^{LR2}_{spectral}$ (e) Fusion $I^{HRpartial}_{spectral}$



(b) LR $I^{LR3}_{spectral}$ (d) Upscaled $I^{LR3}_{spectral}$ (f) FAN restul $I^{HRcoarse}_{spectral}$

Figure 6.4 – Illustration of Image Completion on channel 1 of one example from the validation set: (a-b) are the LR images, (c-d) their upscaled version, (e) the fusion of both upscaled versions and (f) the image completion result.

low computation time, and returns visually interpretable images. FAN relies on an efficient implementation of a modified truncated Gaussian filter. The sparse image is filtered with a Gaussian to interpolate missing entries with Gaussian weights assigned to available pixels in a window surrounding the missing entry, on which the Gaussian filter is centered. The modification is that the Gaussian weights are adjusted to account for the number of locally available pixels.

We use FAN to obtain our input $I^{HRcoarse}_{spectral}$. Note, that we keep the ground truth pixels in $I^{HRcoarse}_{spectral}$ even though FAN outputs different values for them. Figure 6.4 shows the steps to obtain the completed image from both inputs.

6.2.4 Stage-I: Residual Learning

The input of our Stage-I network $I^{HRcoarse}_{spectral}$ is a low-frequency estimation with partially correct high-frequencies $I^{HRresidual}_{spectral}$. Thus we can formulate it as $I^{HRcoarse}_{spectral} = I^{HR}_{spectral} - I^{HRresidual}_{spectral}$, where $I^{HRresidual}_{spectral}$ contains information of the HR spectral image, such as textures and edges.

We adopt a residual learning formulation to train a residual mapping $f(\mathrm{I}_{spectral}^{HRcoarse}) = \mathrm{I}_{spectral}^{HRresidual}$. The architecture of the residual learning network is shown in the Stage-I part of Figure 6.2. By adopting residual learning, the network only learns to predict the high-frequency details without preserving all low-frequency details. This allows us to use a smaller model and train faster than conventional CNN methods. In our residual learning network (Stage-I) for spectral image SR, we use 12 convolutional layers of the same setting except for the last layer: 64 filters of size 3×3 and followed by a ReLU activation. The last layer for generating residual images, consists of 14 filters of size 3×3.

As discussed previously, loss functions in image restoration task is very important when the resulting image is going to be shown to a human observer, typical losses include the $\mathscr{L}_1$ and $\mathscr{L}_2$ distance measures. However, these methods are not well suited to deal with multi-spectral data. The spectral information divergence [14] (SID) compares the similarity of two pixels by measuring the discrepancy between their spectral signatures. This measure has been widely used in hyper-spectral data processing. By defining the relative entropy of the prediction $\mathbf{P}$ with respect to the ground-truth $\mathbf{G}$ containing M pixels as:

$$\mathscr{D}(\mathbf{P}||\mathbf{G}) = \sum_{i=0}^{M} \mathbf{P}_i \log(\frac{\mathbf{P}_i}{\mathbf{G}_i}) \tag{6.3}$$

The SID can then be defined as the symmetric sum of both relative entropy measures:

$$\mathrm{SID} = \mathscr{D}(\mathbf{P}||\mathbf{G}) + \mathscr{D}(\mathbf{G}||\mathbf{P}) \tag{6.4}$$

Additionally, the pixel values are in the range $[0, 65536)$, so a relative error measure is well suited to reduce the large error that an absolute measure could have at the higher end of the range. The mean relative absolute error (MRAE) does exactly that by punishing errors relative to the value of the ground-truth. The MRAE is calculated as:

$$\mathrm{MRAE} = \left| \frac{\mathbf{P} - \mathbf{G}}{\mathbf{G}} \right| \tag{6.5}$$

To better optimize along both metrics, we use a loss function of a sum of MRAE and SID to train our network:

$$\mathscr{L} = \mathrm{SID} + \mathrm{MRAE} \tag{6.6}$$

6.2.5 Stage-II: Color Guided Super-Resolution

We propose a further improvement by using registered pairs of spectral and color images. In fact, mixing information from both modalities allows obtaining both high spatial and spectral

resolution images. However, due to the difficulty of obtaining a large set of registered image pairs, we introduced a transfer learning method built on top of the previous residual network. We build a new residual learning network that takes as input the previous super-resolved image (obtained from Stage-I) concatenated with a 3-channel color image. The new network acts as a fine-tuner for the SR based on the new color data accompanying its input. The network architecture is shown in Stage-II part of Figure 6.2. Here we use 8 convolutional layers with 64 filters of size 3×3 each followed by a ReLU activation, and we use a final convolutional layer with 14 filters of size 3×3 to produce the residual image. We use the same loss function to train this network as discussed above.

6.3 Experiments

6.3.1 Dataset

Both Task-I and Task-II are based on the StereoMSI (Stereo Multi-spectral Image) dataset, which has been introduced in Chapter 2. The dataset consists of hundreds of stereo RGB-spectral image pairs. The images in the dataset depict a wide variety of scenes under natural and artificial illuminants. The nature of the images ranges from natural settings to industrial and office environments. For Task-I, 240 spectral images have been split into 200 for training, 20 for validation, and 20 for testing. For Task-II, 120 stereo image pairs are used, with 100 of these employed for training, 10 for validation and 10 for testing. As only the training and validation data is available to the public, we train the networks on the training data and report the evaluation results on the validation sets, namely Validation-I for Task-I and Validation-II for Task-II.

6.3.2 Comparative Results

We separately train the two stages. For Stage-I, we use spectral patches of size 96×96 with a stride of 24 cropped from the fused $I^{LR2}_{spectral}$ and $I^{LR3}_{spectral}$ images following the described image completion scheme. We use spectral images from both tracks to obtain a larger training set for Stage-I. We use Adam [73] for optimizing the network with weight decay = 10^{-5} and a learning rate of 0.001. We decay the learning rate by 10 every 30 epochs. We set the batch size to 64. After Stage-I converges, we use the Task-II dataset for training Stage-II. We crop 48×48 overlapping patches with a stride of 16 from the dataset. We use the same training strategy as in Stage-I for Stage-II. We use a sum of SID and MRAE for the loss function in the training of both stages.

Table 6.1 shows the results on Validation-I, we compare our image completion method by training the same architecture on inputs from bicubic upscaled images taken from $I^{LR2}_{spectral}$. Our image completion input outperforms this commonly used upscaling method on all metrics. This also applies to the Validation-II dataset.

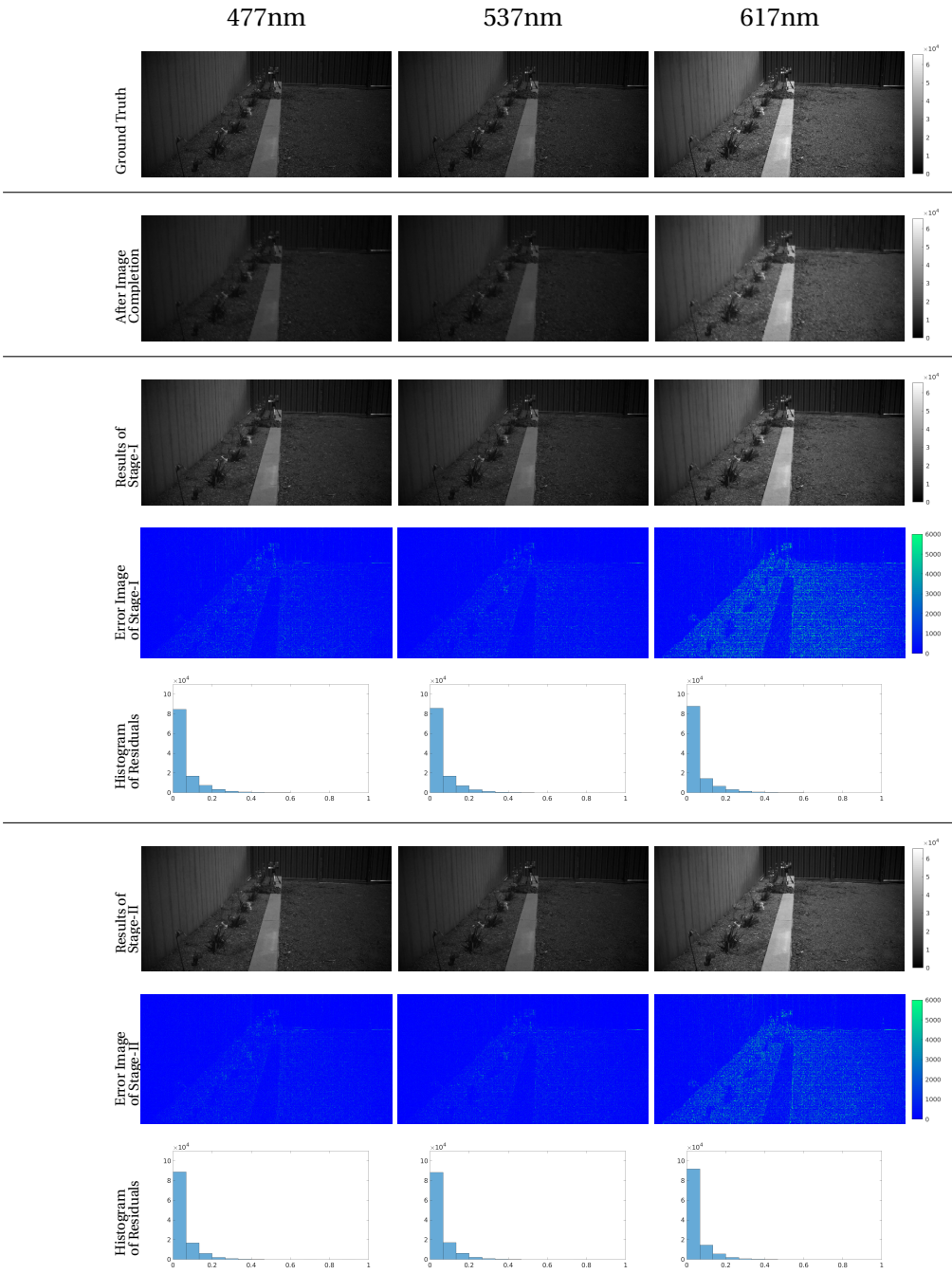


Figure 6.5 – Example of results from different stages. Error images show the absolute difference from our reconstruction to the ground truth spectral image. The histograms of residuals show the histogram of relative absolute errors on the error images.

Metric	Bicubic	Stage-I Results	EDSR
MRAE	0.11	**0.08**	0.10
SID	57.39	**43.48**	43.57
PSNR	36.07	**37.44**	37.27

Table 6.1 – Test results on Validation-I. The bold values indicate the best performance.

Metric	Bicubic	Stage-I Results	Stage-II Results	Residual Net	EDSR
MRAE	0.13	0.10	**0.09**	0.23	0.16
SID	43.32	38.04	**24.51**	36.29	30.67
PSNR	36.48	37.02	**39.17**	36.62	37.13

Table 6.2 – Test results on Validation-II. The bold values indicate the best performance.

We show an example of results from different stages of our pipeline on Validation-II in Figure 6.5. The error images in Figure 6.5 clearly show that with the help of guiding color image, Stage-II is able to improve the results from Stage-I.

We display the comparison with other methods on Validation-II in Table 6.2. To show the merit of our transfer learning model, we train a residual learning network [71] and the state-of-the-art SR network EDSR [84] using both spectral images (after applying image completion on $I^{LR2}_{spectral}$ and $I^{LR3}_{spectral}$) and guiding color images as inputs. For the residual network, we use 21 convolutional layers to obtain the equivalent size of our stacked stages. We set all convolutional layers of the residual network with a configuration of 64 filters of size 3×3 and ReLU activation except the last layer which has 14 filters of size 3×3 with no activation function.

For EDSR, we use the same configuration as the original paper except we ignore the Pixel Shuffle (since we already use an upscaled input) layer [112] and modify the last layers to have 14 filters to reconstruct the 14-band spectral image. EDSR has 32 residual blocks with 256 filters for each convolutional layers. We train both networks using only the Stage-II dataset, and we also do image completion before feeding $I^{LR2}_{spectral}$ and $I^{LR3}_{spectral}$ inputs to the networks. All networks are trained for 300 epochs. Although trained without guiding color images, our Stage-I gives slightly better results than the residual network and EDSR trained on pairs. With guiding color images, Stage-II gains significant improvements on all three metrics.

We also show in Fig 6.6 the visual comparison of EDSR [84] and our method trained on bicubic interpolated input and the completed HR candidates. The error images show that our method outperforms the other two methods.

In addition to performance, we also evaluate the memory and time consumption of the proposed model. For a 240×480 spectral image (with $I^{LR2}_{spectral}$ size of 120×240), our method only takes 0.5 seconds (0.3 seconds on Stage-I and 0.2 seconds on Stage-II) and 800MB memory on Titan X GPU. For EDSR, it takes 1.1 seconds and 8000MB memory on the same device.

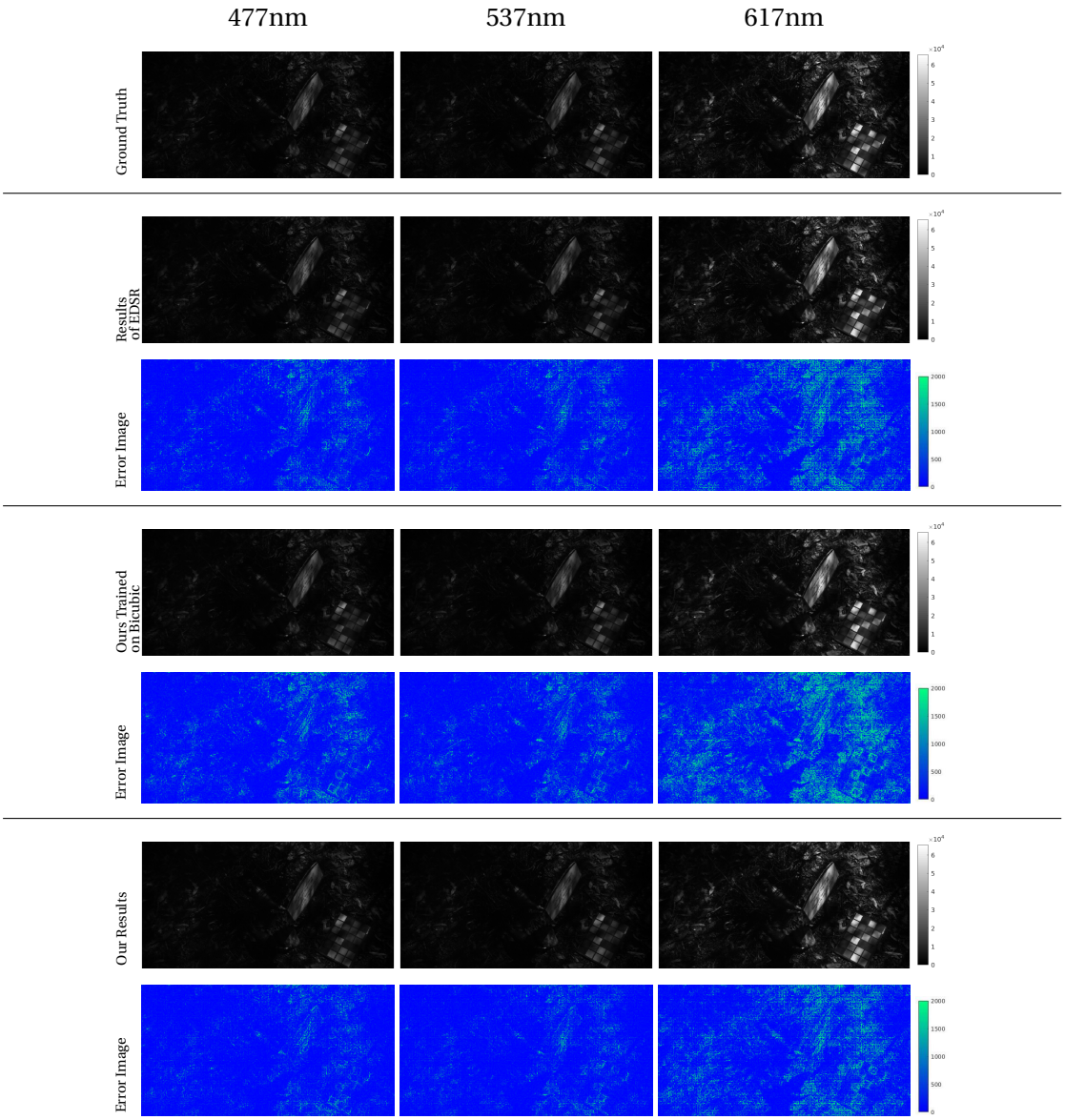


Figure 6.6 – Visual comparsion of results from different methods: EDSR [84] and our method trained on bicubic interpolated inputs and the completed HR candidates. Error images show the absolute difference from our reconstruction to the ground truth spectral image.

6.3.3 Ablation Studies

We run ablation studies on our Stage-I network to study how different factors affect the architecture's performance. First, we study the effect of using different upscaling factors together and alone. Second, we study the effect of the depth on the network on its ability to generalize. Finally, we experiment with changing the loss metrics between MRAE, SID and their sum.

In all the experiments, we train the same residual network with the previously stated configurations, while varying only the one factor in question. We use Adam [73] for optimizing the network with weight decay = $1e-5$ and a learning rate of 0.001. We decay the learning rate by 10 every 20 epochs, and we train all the networks for 100 epochs. We report our results on the Task-I validation set.

Upscaling Factors

In this section, we change the input of the network to understand how different scales affect its performance. We separate the $I^{LR2}_{spectral}$ and $I^{LR3}_{spectral}$ images, and create image completions from each one of these and train two networks separately using those inputs. Both networks are using the sum of MRAE and SID as loss function. We compare both of them against the original network trained on the completed $I^{LR2}_{spectral}$ and $I^{LR3}_{spectral}$ images together. Table 6.3 shows the performance of each network given different inputs. All networks achieve the best performance on the type of input they were trained on, we use those values to compare across models. The completed $I^{LR2}_{spectral}$ includes more original pixels than the completed $I^{LR3}_{spectral}$, the network trained on $I^{LR2}_{spectral}$ outperforms the network trained on $I^{LR3}_{spectral}$. Naturally, the network trained on image completion on both $I^{LR2}_{spectral}$ and $I^{LR3}_{spectral}$ obtains better results than the network trained on $I^{LR2}_{spectral}$ only. This also demonstrates that although $I^{LR3}_{spectral}$ has a lower resolution than $I^{LR2}_{spectral}$, it contains extra original pixels that help to reconstruct a higher-quality HR spectral image.

	$I^{LR2}_{spectral}$		$I^{LR3}_{spectral}$		$I^{LR2}_{spectral} + I^{LR3}_{spectral}$	
	MRAE	SID	MRAE	SID	MRAE	SID
$I^{LR2}_{spectral}$	0.12	55.50	0.25	192.0	0.12	62.97
$I^{LR3}_{spectral}$	0.16	106.24	0.18	117.17	0.14	123.63
$I^{LR2}_{spectral}$+$I^{LR3}_{spectral}$	0.13	58.10	0.24	178.04	**0.10**	**47.20**

Table 6.3 – Test results on Validation-I. The rows represent the type of input the networks were trained on, the columns show the results on inputs taken with different downsampling factors. The bold values indicate the best performance.

Depth Effect

We study the effect of the depth on the network accuracy and generalization. We empirically determine the best depth for the residual network architecture on the Stage-I problem. We vary the depth between 8 and 16 by steps of 2 and report the progress of this networks during training, as well as their best performances on the validation set. Table 6.4 shows the metrics for these 5 networks. We can see that at depth 12, we obtain the best performance in terms of MRAE and PSNR.

Metric	Depth				
	8	10	12	14	16
MRAE	0.11	0.11	**0.10**	0.11	0.12
SID	47.27	**46.94**	47.20	47.26	50.44
PSNR	35.07	35.13	**35.15**	35.05	31.21

Table 6.4 – Test results on Validation-I based on network depth. Numbers in the header row indicate the number of convolutional layers.

Loss Metrics

In this section, we train multiple residual networks with the same parameters using different loss functions. We train with only MRAE, only SID, and a combination of both. We show that using both provides better super-resolved spectral images than using a single metric. Table 6.5 shows the results from these three models. While the network trained on MRAE only outperforms the others on the MRAE metric, its results have a high SID loss. Combining both MRAE and SID losses during training gives the best of both metric results while also scoring high on PSNR.

Metric	MRAE	SID	MRAE+SID
MRAE	**0.09**	0.11	0.10
SID	87.75	**47.20**	**47.20**
PSNR	31.14	34.94	**35.15**

Table 6.5 – Test results on Validation-I based on loss metric. Metrics in the header row indicate the loss used during the training of the network. All networks have a similar structure.

6.4 Conclusion

Our work presents a spectral SR technique based on the fusion of information from multiple sources. First, we introduce an upscaling scheme to combine multi-scale downsampled images based on image completion, and demonstrate it performs better than the commonly used bicubic method. We feed our upscaled images into a two-stage residual network pipeline. In the first stage, we infer original high-resolution images from the upscaled input. In the second stage, we further fine-tune the prediction by appending color images and input it into a smaller residual network. Both networks are economical in time and memory consumption while achieving competitive results.

In conclusion, we demonstrated different schemes combining multi-modal inputs for spectral SR. While this work limited itself to the data provided by the challenge, it can be expanded into other modalities, namely different scales, near-infrared, or even depth inputs.

7 Conclusion

7.1 Thesis Summary

In this thesis, we developed several useful techniques for SR in real-world scenarios. All the proposed algorithms use CNNs. Therefore, in Chapter 2, we first reviewed the network architecture and deep-learning techniques that we used in our methods, including FCNs and GANs. Afterwards, we introduced previous works on SR and other related image restoration tasks. Finally, we have presented the datasets used in this thesis.

As demoisaicing and SR are both related to the resolution limitation of cameras,in Chapter 3, we proposed to jointly perform demoisaicing and SR. We showed that with joint optimization, the accumulation of errors is avoided, and better results are achieved compared to the sequential application of demoisaicing and SR.

To address the problem of unknown blur kernels in real photographs, in Chapter 4, we introduced kernel modeling in SR systems. Instead of using hand-crafted blur kernels, we estimated realistic blur kernels from real LR images. As the estimated blur kernels are limited in quantity, we used a GAN to expand and augment the kernel pool that we collected from the LR images. We validated the performance on synthetic data, realistic data, a collected zoom-in dataset, and through a psychovisual experiment. The trained SR network on the kernels that we collected and expanded achieves better results on real photos, both quantitatively and qualitatively.

As noise is inevitable in image acquisition, in Chapter 5, we studied the problem of joint denoising and SR. To be able to quantitatively evaluate the joint problem, we built a microscopy dataset, namely Widefield2SIM (W2S); it consists of hundreds of image sets that contain,at different noise levels, HR images and their LR versions. We then benchmarked the combinations of state-of-the-art denoising and SR algorithms on our dataset. The results indicate that the denoising algorithm that achieves the best result on denoising does not necessarily provide the best input for the SR algorithms. This motivated us to train a single network for the joint problem. Experimental results show that the single network benefits from joint optimization

and achieves much better results than the sequential application of the two algorithms. We further proposed a novel texture loss that enables the network to produce better results with more faithful details.

In Chapter 6, we developed a fusion system for spectral image SR that contains multi-modal data. We proposed to use an image completion algorithm for the multi-scale data, and we designed a style transfer learning scheme for the multi-modal data. Experimental results show that our approaches are economical in time and memory consumption yet achieve the best results in the PIRM 2018 challenge.

7.2 Future Work

In this thesis, we developed several approaches for single image super-resolution for natural images and other domains. More explorations could be conducted in both directions.

In Chapters 3, 4 and 5, We separately addressed the important factors (*i.e.*, CFA, blur kernel, and noise) of the imaging model. A direct extension would be to combine all these factors in one SR system. Attempts have been made in this direction [104], where a network has been trained to learn the mapping from a LR noisy raw image to a HR clean color image. However, collecting a realistic dataset for this problem is non-trivial. The network also trained on synthetic data that is built using the bicubic blur kernel, and additive Gaussian noise might not perform well on real, natural images. One way to handle this is to develop similar approaches, as in Chapter 4, for collecting or generating realistic CFA and noise. An alternative way is to collect a dataset, as W2S proposed in Chapter 5, but in the domain of natural images.

In all our SR approaches, we solve the SR problem of $\times 2$ or $\times 4$. Currently, however, we want extreme SR that does $\times 8$, $\times 16$ or even $\times 32$. The recently released high-end phone (Samsung Galaxy S20) features a $\times 100$ zoom. Such extreme SR is much more challenging if we want to preserve high perceptual-quality local details. Furthermore, the upscaling factors are unknown in many practical scenarios. Therefore, it is important to develop an SR systems that can handle any arbitrary SR factors and that can characterize the level of degradation.

Along with the promising performance that CNNs have achieved in SR, there remain several important challenges in deep learning. One of these challenges is the need to develop lighter architectures for efficient SR. As our methods and the state-of-the-art SR systems [128, 141, 144] usually contain very deep architectures to achieve high performance, it is very hard to deploy such models in any real-world scenarios. Another challenge is to better understand the deep SR models, especially the representation and features extracted by the networks. Without this knowledge, it is hard to further explore novel network architectures. Lastly, in many applications, a desired loss function is designed for training or evaluating the SR networks. However, many works simply employ MSE or MSE related metrics, such as PSNR and SSIM, as assessment criterion; this has been shown to be a poor criterion for perceptual quality. Consequently, it is of great importance to study and design more perceptually relevant

assessment criteria for SR in real applications.

For the SR in domains other than natural images, such as spectral imaging and medical imaging, it is usually harder to collect large paired datasets for SR training. In practice, it is usually difficult to obtain ground-truth images for all the samples. This has lead to the availability of large databases of images without ground truth or with very limited ground-truth. To address this issue, advances in unsupervised and semi-supervised learning might help.

Bibliography

[1] Abdelhamed, A., Lin, S., and Brown, M. S. (2018). A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1692–1700.

[2] Achanta, R., Arvanitopoulos, N., and Süsstrunk, S. (2017). Extreme image completion. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1333–1337.

[3] Alleysson, D., De Lavarene, B. C., and Herault, J. (2013). Digital image sensor, image capture and reconstruction method and system for implementing same. US Patent 8,564,699.

[4] Anwar, S. and Barnes, N. (2019). Real image denoising with feature attention. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3155–3164.

[5] Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 214–223.

[6] Bevilacqua, M., Roumy, A., Guillemot, C., and Alberi-Morel, M. L. (2012). Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *British Machine Vision Conference (BMVC)*.

[7] Bowles, C., Chen, L., Guerrero, R., Bentley, P., Gunn, R., Hammers, A., Dickie, D. A., Hernández, M. V., Wardlaw, J., and Rueckert, D. (2018). GAN augmentation: augmenting training data using generative adversarial networks. In *arXiv preprint arXiv:1810.10863*.

[8] Bulat, A., Yang, J., and Tzimiropoulos, G. (2018). To learn image super-resolution, use a gan to learn how to do image degradation first. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 185–200.

[9] Burton, G. J. and Moorhead, I. R. (1987). Color and spatial structure in natural scenes. *Applied optics*, 26(1):157–170.

[10] Cai, J., Zeng, H., Yong, H., Cao, Z., and Zhang, L. (2019). Toward real-world single image super-resolution: A new benchmark and a new model. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3086–3095.

[11] Cai, J.-F., Ji, H., Liu, C., and Shen, Z. (2012). Framelet-based blind motion deblurring from a single image. *IEEE Transactions on Image Processing (TIP)*, 21(2):562–572.

[12] Calimeri, F., Marzullo, A., Stamile, C., and Terracina, G. (2017). Biomedical data augmentation using generative adversarial neural networks. In *International Conference on Artificial Neural Networks*, pages 626–634.

[13] Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). Brief: Binary robust independent elementary features. In *European conference on computer vision*, pages 778–792. Springer.

[14] Chang, C.-I. (2000). An information-theoretic approach to spectral variability, similarity, and discrimination for hyperspectral image analysis. *IEEE Transactions on Information Theory*, 46(5):1927–1932.

[15] Che, T., Li, Y., Jacob, A. P., Bengio, Y., and Li, W. (2016). Mode regularized generative adversarial networks. *arXiv preprint arXiv:1612.02136*.

[16] Chen, C., Chen, Q., Xu, J., and Koltun, V. (2018a). Learning to see in the dark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3291–3300.

[17] Chen, C., Xiong, Z., Tian, X., Zha, Z.-J., and Wu, F. (2019). Camera lens super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1652–1660.

[18] Chen, J., Chen, J., Chao, H., and Yang, M. (2018b). Image blind denoising with generative adversarial network based noise modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3155–3164.

[19] Cho, S. and Lee, S. (2009). Fast motion deblurring. *ACM Transactions on Graphics (TOG)*, 28(5):145.

[20] Choi, J.-H., Zhang, H., Kim, J.-H., Hsieh, C.-J., and Lee, J.-S. (2019). Evaluating robustness of deep image super-resolution against adversarial attacks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 303–311.

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

[22] Damodaran, B. B., Kellenberger, B., Flamary, R., Tuia, D., and Courty, N. (2018). Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *arXiv preprint arXiv:1803.10081*.

[23] Dang-Nguyen, D.-T., Pasquini, C., Conotter, V., and Boato, G. (2015). RAISE: a raw images dataset for digital image forensics. In *Proceedings of the 6th ACM Multimedia Systems Conference*, pages 219–224.

[24] Dong, C., Loy, C. C., He, K., and Tang, X. (2016a). Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(2):295–307.

[25] Dong, C., Loy, C. C., and Tang, X. (2016b). Accelerating the super-resolution convolutional neural network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 391–407.

[26] Dong, W., Zhang, L., Shi, G., and Li, X. (2013). Nonlocally centralized sparse representation for image restoration. *IEEE Transactions on Image Processing (TIP)*, 22(4):1620–1630.

[27] Dumoulin, V. and Visin, F. (2016). A guide to convolution arithmetic for deep learning. In *arXiv preprint arXiv:1603.07285*.

[28] Efrat, N., Glasner, D., Apartsin, A., Nadler, B., and Levin, A. (2013). Accurate blur models vs. image priors in single image super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2832–2839.

[29] Egiazarian, K. and Katkovnik, V. (2015). Single image super-resolution via bm3d sparse coding. In *Proceedings of the 23rd European Signal Processing Conference (EUSIPCO)*, pages 2849–2853.

[30] El Helou, M., Dümbgen, F., Achanta, R., and Süsstrunk, S. (2018a). Fourier-domain optimization for image processing. In *arXiv preprint arXiv:1809.04187*.

[31] El Helou, M., Dümbgen, F., and Siisstrunk, S. (2018b). Aam: An assessment metric of axial chromatic aberration. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2486–2490. IEEE.

[32] El Helou, M., Sadeghipoor, Z., and Süsstrunk, S. (2017). Correlation-based deblurring leveraging multispectral chromatic aberration in color and near-infrared joint acquisition. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 1402–1406. IEEE.

[33] El Helou, M. and Süsstrunk, S. (2020). Blind universal bayesian image denoising with gaussian noise level learning. *IEEE Transactions on Image Processing*, 29:4885–4897.

[34] El Helou, M., Zhou, R., and Süsstrunk, S. (2020). Stochastic frequency masking to improve super-resolution and denoising networks. *arXiv preprint arXiv:2003.07119*.

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

[36] Fan, L., Zhang, F., Fan, H., and Zhang, C. (2019). Brief review of image denoising techniques. *Visual Computing for Industry, Biomedicine, and Art*, 2(1):7.

[37] Farsiu, S., Elad, M., and Milanfar, P. (2005). Multiframe demosaicing and super-resolution of color images. *IEEE Transactions on Image Processing (TIP)*, 15(1):141–159.

[38] Ferguson, M., Ak, R., Lee, Y.-T. T., and Law, K. H. (2017). Automatic localization of casting defects with convolutional neural networks. In *2017 IEEE International Conference on Big Data*, pages 1726–1735. IEEE.

[39] Foi, A., Trimeche, M., Katkovnik, V., and Egiazarian, K. (2008). Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing (TIP)*, 17(10):1737–1754.

[40] Freedman, G. and Fattal, R. (2011). Image and video upscaling from local self-examples. *ACM Transactions on Graphics (TOG)*, 30(2):1–11.

[41] Ganin, Y. and Lempitsky, V. (2014). Unsupervised domain adaptation by backpropagation. In *arXiv preprint arXiv:1409.7495*.

[42] Gharbi, M., Chaurasia, G., Paris, S., and Durand, F. (2016). Deep joint demosaicking and denoising. *ACM Transactions on Graphics (TOG)*, 35(6):1–12.

[43] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2672–2680.

[44] Grant, M. and Boyd, S. (2008). Graph implementations for nonsmooth convex programs. In Blondel, V., Boyd, S., and Kimura, H., editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited. http://stanford.edu/~boyd/graph_dcp.html.

[45] Gu, J., Lu, H., Zuo, W., and Dong, C. (2019). Blind super-resolution with iterative kernel correction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1604–1613.

[46] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017). Improved training of wasserstein GANs. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5767–5777.

[47] Guo, T., Mousavi, H. S., Vu, T. H., and Monga, V. (2017). Deep wavelet prediction for image super-resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 104–113.

[48] Gupta, S., Hoffman, J., and Malik, J. (2016). Cross modal distillation for supervision transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2827–2836.

[49] Gustafsson, M. G. (2000). Surpassing the lateral resolution limit by a factor of two using structured illumination microscopy. *booktitle of microscopy*.

[50] Gustafsson, M. G. (2005). Nonlinear structured-illumination microscopy: wide-field fluorescence imaging with theoretically unlimited resolution. *Proceedings of the National Academy of Sciences*.

[51] Gustafsson, M. G., Shao, L., Carlton, P. M., Wang, C. R., Golubovskaya, I. N., Cande, W. Z., Agard, D. A., and Sedat, J. W. (2008). Three-dimensional resolution doubling in wide-field fluorescence microscopy by structured illumination. *Biophysical journal*.

[52] Hacohen, Y., Shechtman, E., and Lischinski, D. (2013). Deblurring by example using dense correspondence. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2384–2391.

[53] Harandi, M., Salzmann, M., and Porikli, F. (2014). Bregman divergences for infinite dimensional covariance matrices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1003–1010.

[54] Haris, M., Shakhnarovich, G., and Ukita, N. (2018). Deep back-projection networks for super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1664–1673.

[55] Hayes, J., Melis, L., Danezis, G., and De Cristofaro, E. (2017). LOGAN: evaluating privacy leakage of generative models using generative adversarial networks. *arXiv preprint arXiv:1705.07663*.

[56] He, H. and Kondi, L. P. (2005). A regularization framework for joint blur estimation and super-resolution of video sequences. In *IEEE International Conference on Image Processing (ICIP)*, volume 3, pages III–329.

[57] He, K., Sun, J., and Tang, X. (2011). Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(12):2341–2353.

[58] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034.

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

[60] Heide, F., Steinberger, M., Tsai, Y.-T., Rouf, M., Pająk, D., Reddy, D., Gallo, O., Liu, J., Heidrich, W., Egiazarian, K., Kautz, J., and Pulli, K. (2014). Flexisp: A flexible camera image processing framework. *ACM Transactions on Graphics (TOG)*, 33(6):1–13.

[61] Hu, Y., Zhang, D., Ye, J., Li, X., and He, X. (2012). Fast and accurate matrix completion via truncated nuclear norm regularization. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, page 1.

[62] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR*, pages 4700–4708.

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

[64] Ignatov, A., Kobyshev, N., Timofte, R., Vanhoey, K., and Van Gool, L. (2017). Dslr-quality photos on mobile devices with deep convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3277–3285.

[65] Ioffe, S. and Szegedy, C. (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*.

[66] Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 1125–1134.

[67] Johnson, J., Alahi, A., and Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European conference on computer vision (ECCV)*, pages 694–711.

[68] Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2017). Progressive growing of GANs for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

[69] Karras, T., Laine, S., and Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410.

[70] Keys, R. (1981). Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6):1153–1160.

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

[72] Kim, J., Kwon Lee, J., and Mu Lee, K. (2016b). Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1637–1645.

[73] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the 33rd International Conference on Learning Representations (ICLR)*.

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

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

[76] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1097–1105.

[77] Lahoud, F., Zhou, R., and Süsstrunk, S. (2018). Multi-modal spectral image super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0.

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

[79] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., and Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4681–4690.

[80] Levin, A., Zomet, A., and Weiss, Y. (2003). Learning how to inpaint from global image statistics. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, page 305.

[81] Li, P., Xie, J., Wang, Q., and Zuo, W. (2017). Is second-order information helpful for large-scale visual recognition? In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2070–2078.

[82] Li, W., Zhao, L., Lin, Z., Xu, D., and Lu, D. (2015). Non-local image inpainting using low-rank matrix completion. In *Computer Graphics Forum*, volume 34, pages 111–122.

[83] Li, Z., Yang, J., Liu, Z., Yang, X., Jeon, G., and Wu, W. (2019). Feedback network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3867–3876.

[84] Lim, B., Son, S., Kim, H., Nah, S., and Mu Lee, K. (2017). Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 136–144.

[85] Liu, M.-Y. and Tuzel, O. (2016). Coupled generative adversarial networks. In *Advances in Neural Information Processing Systems*, pages 469–477.

[86] Liu, Q., Lai, Z., Zhou, Z., Kuang, F., and Jin, Z. (2016). A truncated nuclear norm regularization method based on weighted residual error for matrix completion. *IEEE Transactions on Image Processing (TIP)*, 25(1):316–330.

[87] Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440.

[88] Luisier, F., Blu, T., and Unser, M. (2010). Image denoising in mixed Poisson–Gaussian noise. *IEEE Transactions on Image Processing (TIP)*, 20(3):696–708.

[89] Makitalo, M. and Foi, A. (2012). Optimal inversion of the generalized Anscombe transformation for Poisson-Gaussian noise. *IEEE Transactions on Image Processing (TIP)*, 22(1):91–103.

[90] Martin, D., Fowlkes, C., Tal, D., and Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 416–423.

[91] Matsui, Y., Ito, K., Aramaki, Y., Fujimoto, A., Ogawa, T., Yamasaki, T., and Aizawa, K. (2017). Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, 76(20):21811–21838.

[92] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

[93] Mechrez, R., Talmi, I., Shama, F., and Zelnik-Manor, L. (2018). Maintaining natural image statistics with the contextual loss. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 427–443.

[94] Michaeli, T. and Irani, M. (2013). Nonparametric blind super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 945–952.

[95] Milanfar, P. (2017). *Super-resolution imaging*. CRC press.

[96] Mildenhall, B., Barron, J. T., Chen, J., Sharlet, D., Ng, R., and Carroll, R. (2018). Burst denoising with kernel prediction networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2502–2510.

[97] Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML)*.

[98] Nasrollahi, K. and Moeslund, T. B. (2014). Super-resolution: a comprehensive survey. *Machine Vision and Applications*, 25(6):1423–1468.

[99] Pan, J., Sun, D., Pfister, H., and Yang, M.-H. (2016). Blind image deblurring using dark channel prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1628–1636.

[100] Park, S. C., Park, M. K., and Kang, M. G. (2003). Super-resolution image reconstruction: a technical overview. *IEEE signal processing magazine*, 20(3):21–36.

[101] Petzka, H., Fischer, A., and Lukovnicov, D. (2017). On the regularization of Wasserstein gans. *arXiv preprint arXiv:1709.08894*.

[102] Peyrard, C., Baccouche, M., and Garcia, C. (2016). Blind super-resolution with deep convolutional neural networks. In *International Conference on Artificial Neural Networks*, pages 161–169.

[103] Plotz, T. and Roth, S. (2017). Benchmarking denoising algorithms with real photographs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1586–1595.

[104] Qian, G., Gu, J., Ren, J. S., Dong, C., Zhao, F., and Lin, J. (2019). Trinity of pixel enhancement: a joint solution for demosaicking, denoising and super-resolution. In *arXiv preprint arXiv:1905.02538*.

[105] Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. In *Proceedings of the 34th International Conference on Learning Representations (ICLR)*.

[106] Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. (2016). Generative adversarial text-to-image synthesis. In *Proceedings of The 33rd International Conference on Machine Learning (ICML)*.

[107] Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 430–443.

[108] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2564–2571.

[109] Sajjadi, M. S., Scholkopf, B., and Hirsch, M. (2017). EnhanceNet: Single image super-resolution through automated texture synthesis. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4491–4500.

[110] Schneider, C. A., Rasband, W. S., and Eliceiri, K. W. (2012). NIH Image to ImageJ: 25 years of image analysis. *Nature methods*, 9(7):671–675.

[111] Schwartz, E., Giryes, R., and Bronstein, A. M. (2018). DeepISP: Toward learning an end-to-end image processing pipeline. *IEEE Transactions on Image Processing (TIP)*, 28(2):912–923.

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

[113] Shoeiby, M., Robles-Kelly, A., Timofte, R., Zhou, R., Lahoud, F., Süsstrunk, S., Xiong, Z., Shi, Z., Chen, C., Liu, D., Zha, Z.-J., Wu, F., Wei, K., Zhang, T., Wang, L., Fu, Y., Zhong, Z., Nagasubramanian, K., Singh, A. K., Singh, A., Sarkar, S., and Baskar, G. (2018a). PIRM2018 challenge on spectral image super-resolution: methods and results. In *Proceedings of the European Conference on Computer Vision Workshops (ECCVW)*, pages 0–0.

[114] Shoeiby, M., Robles-Kelly, A., Wei, R., and Timofte, R. (2018b). PIRM2018 challenge on spectral image super-resolution: Dataset and study. In *Proceedings of the European Conference on Computer Vision Workshops (ECCVW)*, pages 0–0.

[115] Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *Proceedings of the 33rd International Conference on Learning Representations (ICLR)*.

[116] Smith, D. C. (2011). Super-resolution of text images through neighbor embedding. In *Proceedings of the 13th IASTED International Conference on Signal and Image Processing*, pages 19–26.

[117] Song, P., Deng, X., Mota, J. F., Deligiannis, N., Dragotti, P.-L., and Rodrigues, M. (2019). Multimodal image super-resolution via joint sparse representations induced by coupled dictionaries. *IEEE Transactions on Computational Imaging*.

[118] Sun, J., Xu, Z., and Shum, H.-Y. (2008). Image super-resolution using gradient profile prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8.

[119] Sun, J., Yuan, L., Jia, J., and Shum, H.-Y. (2005). Image completion with structure propagation. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 861–868.

[120] Tai, Y., Yang, J., Liu, X., and Xu, C. (2017). MemNet: A persistent memory network for image restoration. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4539–4547.

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

[122] Timofte, R., Agustsson, E., Van Gool, L., Yang, M.-H., and Zhang, L. (2017). NTIRE 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 114–125.

[123] Timofte, R., De Smet, V., and Van Gool, L. (2014). A+: adjusted anchored neighborhood regression for fast super-resolution. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 111–126.

[124] Tran, T., Pham, T., Carneiro, G., Palmer, L., and Reid, I. (2017). A bayesian data augmentation approach for learning deep models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2797–2806.

[125] Ulyanov, D., Vedaldi, A., and Lempitsky, V. (2016). Instance normalization: the missing ingredient for fast stylization. In *arXiv preprint arXiv:1607.08022*.

[126] Vasu, S., Thekke Madam, N., and Rajagopalan, A. (2018). Analyzing perception-distortion tradeoff using enhanced perceptual super-resolution network. In *Proceedings of the European Conference on Computer Vision Workshops (ECCVW)*, pages 0–0.

[127] Verveer, P. J., Gemkow, M. J., and Jovin, T. M. (1999). A comparison of image restoration approaches applied to three-dimensional confocal and wide-field fluorescence microscopy. *booktitle of microscopy*.

[128] Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., and Loy, C. C. (2018). ESRGAN: enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision Workshops (ECCVW)*, pages 63–79.

[129] Wei, Q., Dobigeon, N., and Tourneret, J.-Y. (2015). Fast fusion of multi-band images based on solving a sylvester equation. *IEEE Transactions on Image Processing (TIP)*, 24(11):4109–4121.

[130] Wycoff, E., Chan, T.-H., Jia, K., Ma, W.-K., and Ma, Y. (2013). A non-negative sparse promoting algorithm for high resolution hyperspectral imaging. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1409–1413.

[131] Xu, B., Wang, N., Chen, T., and Li, M. (2015). Empirical evaluation of rectified activations in convolutional network. In *arXiv preprint arXiv:1505.00853*.

[132] Xu, X., Ma, Y., and Sun, W. (2019). Towards real scene super-resolution with raw images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1723–1731.

[133] Yang, J., Wang, Z., Lin, Z., Cohen, S., and Huang, T. (2012). Coupled dictionary training for image super-resolution. *IEEE Transactions on Image Processing (TIP)*, 21(8):3467–3478.

[134] Yang, W., Zhang, X., Tian, Y., Wang, W., Xue, J.-H., and Liao, Q. (2019). Deep learning for single image super-resolution: A brief review. *IEEE Transactions on Multimedia*, 21(12):3106–3121.

[135] Yokoya, N., Yairi, T., and Iwasaki, A. (2012). Coupled nonnegative matrix factorization unmixing for hyperspectral and multispectral data fusion. *IEEE Transactions on Geoscience and Remote Sensing*, 50(2):528–537.

[136] Zeyde, R., Elad, M., and Protter, M. (2010). On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730.

[137] Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., and Metaxas, D. N. (2017a). StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5907–5915.

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

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

[140] Zhang, X., Chen, Q., Ng, R., and Koltun, V. (2019a). Zoom to learn, learn to zoom. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3762–3770.

[141] Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., and Fu, Y. (2018b). Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 286–301.

[142] Zhang, Y., Tian, Y., Kong, Y., Zhong, B., and Fu, Y. (2018c). Residual dense network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2472–2481.

[143] Zhang, Y., Zhu, Y., Nichols, E., Wang, Q., Zhang, S., Smith, C., and Howard, S. (2019b). A Poisson-Gaussian denoising dataset with real fluorescence microscopy images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11710–11718.

[144] Zhang, Z., Wang, Z., Lin, Z., and Qi, H. (2019c). Image super-resolution by neural texture transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7982–7991.

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

[146] Zhao, X., Wu, Y., Tian, J., and Zhang, H. (2016). Single image super-resolution via blind blurring estimation and anchored space mapping. *Computational Visual Media*, 2(1):71–85.

[147] Zheng, Z., Zheng, L., and Yang, Y. (2017). Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *Proceedings of the IEEE International Conference on Computer Vision (CCV)*, pages 3754–3762.

[148] Zhou, R., Achanta, R., and Süsstrunk, S. (2018). Deep residual network for joint demosaicing and super-resolution. *Color and Imaging Conference (CIC)*, 2018(1):75–80.

[149] Zhou, R., Helou, M. E., Sage, D., Laroche, T., Seitz, A., and Süsstrunk, S. (2020). W2s: A joint denoising and super-resolution dataset. *arXiv preprint arXiv:2003.05961*.

[150] Zhou, R., Lahoud, F., El Helou, M., and Süsstrunk, S. (2019). A comparative study on wavelets and residuals in deep super resolution. *Electronic Imaging*, 2019(13):135–1.

[151] Zhou, R. and Süsstrunk, S. (2019). Kernel modeling super-resolution on real low-resolution images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2433–2443.

[152] Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2223–2232.

[153] Zhu, Y., Aoun, M., Krijn, M., Vanschoren, J., and Campus, H. T. (2018). Data augmentation using conditional generative adversarial networks for leaf counting in arabidopsis plants. In *proceedings of British Machine Vision Conference(BMVC)*, page 324.

[154] Zoran, D. and Weiss, Y. (2011). From learning models of natural image patches to whole image restoration. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 479–486.

Ruofan ZHOU

✉: BC318, Station 14, EPFL, Lausanne, 1015, Switzerland
✆: +41 78 829 77 42
@: ruofan.zhou@epfl.ch

PROFILE

Highly dependable Computer Science professional with experience and education in Technology, proficient in programming, supervisory, presentations, and identifying, modeling and solving complex problems.

EDUCATION

École Polytechnique Fédérale de Lausanne Lausanne, Switzerland
PhD in Computer Science, Advisor: Sabine Süsstrunk Expected: Jul 2020
Research Area: deep learning, image processing, computational photography, computer vision

Tsinghua University Beijing, China
BE in Computer Science and Technology, GPA: 90/100, graduated with honors Jun 2015
Second Major: *BA in Digital Entertainment Design,* GPA: 88/100

PROFESSIONAL EXPERIENCE

École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
Doctoral Assistant 09/2015 – present

- Design of deep learning model for image restoration, results published in ICCV, ECCV, CIC and EI
- Development of phychovisual experiment for perceptual evaluation of image restoration results
- Present research results multiple times at conferences and seminars
- Supervisor for 8 Bachelor semester projects and 7 Master semester projects, including topics of dataset collection, website development, image processing, video processing, software development, recommendation system, and audio processing

Research Assistant 07/2014 – 08/2014

- Development of easy-use image and text crawler on different social platforms
- Design of model to analysis emotion from texts to identify people's brand engagement

YITU-Inc, Shanghai, China
Research Intern 02/2015 – 08/2015

- Design and implement semi-automatic annotation tools for object tracking and face grouping
- Literature review, presentation and implementation of tracking, detection, face grouping algorithms
- Design and build a powerful SVM training tool for visual features

Tsinghua University, Beijing, China
Research Assistant (part-time) 10/2013 – 02/2014

- Design of model and algorithm to analysis user's personality through his/her social media use
- Build an app on renren.com that can predict user's personality
- Awarded Third Prize at Tsinghua Challenge Cup

Google, Beijing, China
Software Engineer Intern (part-time) 07/2013 – 12/2013

- Design, implement and test of algorithms to detect non-informative part on webpages

SELECTED PUBLICATIONS

- **Ruofan Zhou**, Sabine Süsstrunk, “Kernel Modeling Super-Resolution on Real Low-Resolution Images.”, International Conference in Computer Vision (ICCV), 2019
- **Ruofan Zhou*,** Fayez Lahoud*, Majed El Helou, Sabine Süsstrunk, “A Comparative Study on Wavelets and Residuals in Deep Super Resolution”, Electronic Imaging (EI), 2019 (*joint 1st authors)
- **Ruofan Zhou**, Radhakrishna Achanta, Sabine Süsstrunk, “Deep Residual Network for Joint Demosaicing and Super-Resolution”, Color and Imaging Conference (CIC), 2018
- **Ruofan Zhou***, Fayez Lahoud*, Sabine Süsstrunk, “Multi-Modal Spectral Image Super-Resolution”, European Conference in Computer Vision (ECCV) Workshops, 2018 (*joint 1st authors)
- Frank Schmutz, Fabrice Guibert, **Ruofan Zhou**, Majed El Helou, Sabine Süsstrunk, “Extreme Video Completion”, International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020

HONORS & AWARDS

- Winner Award in PIRM Challenge on Spectral Image Super-Resolution, in ECCV 2018
- Honorable Mention Award in NTIRE 2018 Challenge on Spectral Reconstruction, in CVPR 2018
- Google Anita Borg Memorial Scholarship, 2014

PROGRAMMING CONTESTS & HACKATHONS

- Rank 76/2815 in Google Hash Code, 2017
- Coach for EPFL teams at Southwestern Europe Regional Contest (SWERC), 2018
- Bronze Medal (12th place) at Southwestern Europe Regional Contest (SWERC), 2015
- Problem setter and tester for Helvetic Coding Contest (HC2), 2016 - 2019
- Participate in Global Game Jam 2014 & 2015, develop game demos in 48 hours
- Bronze medal in Asia-Pacific Informatics Olympiad (China Regional), 2009
- First prize in National Olympiad in Informatics in Provinces (Shanghai Regional), 2008 & 2010

EXTRACURRICULAR EXPERIENCE

Treasurer | Polyprog: competitive programming association at EPFL 09/2017 – present
Recording financial status of the association, coaching on algorithms and competition strategies

Deputy Secretary-General | Tsinghua Alumni in Switzerland 10/2016 – present
Organizing outing events, maintaining public account, photographer for events including Davos Forum

President | Chinese Students & Scholars Association (CSSA) Lausanne 12/2015 – 03/2017
Representing the CSSA Lausanne (over 500 members), interacting with the Chinese embassy, promoting Sino-Swiss relations and enhancing internal interactions, maintaining official website and public account, building partnerships with local business, organized over 20 events

Leader & Guitar Player | SPLAY: a campus band in Tsinghua University 06/2013 – 07/2015
Performed several live shows on campus, won ”Best Original Music Award” in 24th Campus Singer Competition, recorded 2 orignial songs

Percussionist | Tsinghua University Military Band 09/2011 – 01/2014
Performed in school opening ceremonys, the New Year Party, Music Salon, concerts, etc.

ADDITIONAL INFORMATION

Born 23.04.1993, Chinese Passport, B permit in Switzerland
Civil status: single
Languages: native in Chinese, proficient in English, basic in French (A2)
Hobbies: interacting between cultures, traveling, music, photography, gourmet, sports, video games