Towards Universal Deep Learning Models for Image Restoration

TOWARDS UNIVERSAL DEEP LEARNING MODELS FOR IMAGE RESTORATION

By FANGZHOU LUO, M.A.Sc.

A Thesis Submitted to the School of Graduate Studies in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy

McMaster University

DOCTOR OF PHILOSOPHY (2023)

Hamilton, Ontario, Canada (Electrical & Computer Engineering)

TITLE:	Towards Universal Deep Learning Models for Image Restoration
AUTHOR:	Fangzhou Luo M.A.Sc. (Electrical & Computer Engineering), McMaster University, Hamilton, Canada
SUPERVISOR:	Dr. Xiaolin Wu
NUMBER OF PAGES:	xi, 120

Abstract

Recent years have witnessed the remarkable successes of deep learning methods in the field of image restoration. However, despite the similarities across different image restoration tasks, researchers often adopt a problem-specific approach. Most deep learning based image restoration algorithms are tailored to a specific type of degradation, performing poorly when being applied to degradations that are deviated from those of the training dataset. This lack of universality limits the adaptability and robustness of these algorithms in real-world scenarios. The approach of training and storing multiple models for various degradation types wastes resources and reduces efficiency, and it still tends to struggle with unseen and complex degradation sources. In this thesis, we depart from current problem-specific methodologies for image restoration and strive to improve the universality and robustness of the existing methods. We propose three novel methods to achieve the above goal; they are a new inference method, a new network model, and a new training method, respectively.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr. Xiaolin Wu, whose unwavering support, insightful guidance, and scholarly expertise have been invaluable throughout my doctoral journey. His encouragement and mentorship have not only shaped the direction of my research but have also inspired me to reach new heights in my academic pursuits.

I extend my sincere appreciation to the members of my thesis committee, Dr. Zixiang Xiong, Dr. Shahram Shirani, Dr. Dongmei Zhao, and Dr. Sorina Dumitrescu. Your constructive feedback, thoughtful critiques, and expertise in your respective fields have significantly enriched the quality of this work. I am truly grateful for the time and effort you dedicated to shaping this research.

To my classmates and colleagues, who have been a source of inspiration, camaraderie, and intellectual exchange, I am grateful for the collaborative environment that enhanced my research experience. Your friendship and shared academic endeavors have made this challenging pursuit more enjoyable and rewarding.

Lastly, my deepest appreciation goes to my parents for their unwavering support, love, and sacrifices. Their encouragement and belief in my abilities have been my driving force. This achievement is as much theirs as it is mine.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the rapidly advancing landscape of digital image processing, the pivotal role of digital images spans diverse domains, ranging from video surveillance [32] and autonomous driving [53] to medical imaging [20] and remote sensing [22]. The inherent quality of images captured by cameras directly influences the efficacy of systems operating in these domains. However, the challenge lies in the fact that images obtained are not consistently clear, and they can suffer from a spectrum of degradations arising from various sources, including capture processes, device defects, and environmental conditions [43]. Surveillance and medical imaging outputs often exhibit low resolution, images from moving cameras may manifest motion blur, and those captured in adverse weather conditions may display color distortions, blurs, and noise. These degradations significantly impede the performance of visual systems in critical tasks such as segmentation, detection, and target tracking [36].

Moreover, beyond contemporary images, there is a burgeoning demand for the digitization of historical and cultural artifacts [124]. However, digitized images may carry inherent degradation, or the digitization process itself may introduce noise from

the environment. Therefore, the development of efficient image restoration algorithms becomes imperative for the recovery of degraded images [146]. This not only impacts the functionality of modern technological systems but also plays a crucial role in the preservation of cultural and historical facets encapsulated in digitized imagery.

The research landscape on image restoration has been a focal point for several decades, consistently captivating researchers due to its evolving challenges and wide-ranging applications. The spectrum of image restoration tasks includes image super-resolution [39], deblurring [40], denoising [181], inpainting [33], and the removal of compression artifacts [38]. At its core, image restoration aims to reconstruct a high-quality image with optimal visibility and unblemished content from a degraded counterpart. The complexity of this task is compounded by the multitude of potential mappings between degraded observations and their corresponding restored images, presenting a formidable challenge in determining the inverse function.

Historically, conventional image restoration methods treated the restoration process as signal processing, employing hand-crafted algorithms to mitigate artifacts from both spatial and frequency perspectives [8]. However, the advent of deep learning has ushered in a transformative shift in the landscape of image restoration [37]. Contemporary endeavors in image restoration have curated extensive datasets tailored to specific tasks, fostering the training of deep learning models. These models leverage well-designed backbones, often rooted in Convolutional Neural Networks or Transformer architectures, to learn intricate mappings and patterns from the data. This paradigm shift towards deep learning not only signifies a departure from traditional methods but also underscores the potential for more adaptive and sophisticated approaches in addressing the complexities of image restoration.

1.1 Problem Formulation of Image Restoration

In the realm of image restoration, traditional methodologies have conventionally harnessed sophisticated mathematical techniques and probabilistic models to address inverse problems. These methodologies predominantly rely on either maximum likelihood or Bayesian approaches, employing iterative processes to rectify the estimated degradations [48, 133]. Within the conventional image restoration framework, the degraded image y is elegantly conceptualized through the following expression:

$$y = (x \otimes k) \downarrow_s + n \tag{1.1.1}$$

In this equation, the convolution between the blurry kernel k and the unknown high-quality image x is denoted by $x \otimes k$, where the downsampling operator with a scale factor of s is represented by $\downarrow_s$, and n encapsulates the independent noise term. This formulation succinctly captures the intricate interplay between the blurred input, the unknown high-quality image, and the additive noise, forming a foundational basis for addressing challenges in image restoration.

Applying maximum a posteriori estimation yields the formulation for the latent image $\hat{x}$:

$$\hat{x} = \underset{x}{\operatorname{argmax}} \log(p(y|x)) + \log(p(x)) \tag{1.1.2}$$

where $p(y|x)$ denotes the likelihood of the degraded observation y given the clean image x, while $p(x)$ represents the prior distribution of the clean image x.

Furthermore, the problem can be cast as a constrained maximum likelihood

estimation:

$$\hat{x} = \underset{x}{\operatorname{argmin}} \|y - (x \otimes k) \downarrow_s\|^2 + \lambda\phi(x) \tag{1.1.3}$$

In this formulation, the fidelity term $\|y - (x \otimes k) \downarrow_s\|^2$ approximates the likelihood $p(y|x)$, while the regularization term $\phi(x)$ represents priors of the latent image x or constraints on the solution. The choice of priors can be adapted depending on the specific requirements of various image restoration tasks.

1.2 Deep Learning for Image Restoration

Deep learning [93], a subset of machine learning [13], is distinguished by its intrinsic ability to autonomously acquire diverse representations of data, marking a departure from traditional task-specific algorithms reliant on manually crafted features. This capacity for holistic learning is underpinned by the high approximating capacity and hierarchical nature of artificial neural networks, constituting the foundation for contemporary deep learning models. While the roots of deep learning can be traced back to the perceptron algorithms of the 1960s, a pivotal moment occurred in the 1980s with the introduction of the multilayer perceptron and the backpropagation algorithm [135]. Concurrently, the emergence of the convolutional neural network [91] and recurrent neural network [42] played crucial roles, leaving enduring impacts in computer vision and speech recognition, respectively.

Despite early successes, inherent deficiencies impeded the progress of neural networks. A resurgence in modern artificial neural networks commenced with the advent of pretraining techniques, notably leveraging the restricted Boltzmann machine in

2006 [138]. Harnessing the surge in computing power and advancements in algorithms, deep neural network models showcased exceptional performance across various supervised tasks [87]. Simultaneously, the rise of unsupervised algorithms, including the deep Boltzmann machine [137], variational autoencoder [83], and generative adversarial nets [55], gained prominence for their efficacy in addressing challenging unlabeled data.

Image restoration, an essential aspect of computer vision, has undergone transformative advancements through the incorporation of deep learning methodologies. The advent of convolutional neural networks has marked a paradigm shift in this domain, offering robust solutions to address a myriad of challenges associated with restoring degraded or corrupted images. Neural networks excel at automatically learning hierarchical features from input data, which proves particularly advantageous in tasks such as denoising, deblurring, and super-resolution.

One prominent architecture in the realm of image restoration is the U-Net [134], renowned for its success in various medical imaging tasks. The U-Net's unique architecture, featuring contracting and expansive pathways, enables the network to capture both local and global contextual information effectively. Additionally, deep residual networks have demonstrated exceptional performance by facilitating the training of remarkably deep networks, mitigating the vanishing gradient problem, and enhancing the overall representational capacity for image restoration tasks [103]. Generative adversarial networks have also emerged as a pivotal tool in image restoration, leveraging a generative model's ability to synthesize realistic-looking images and a discriminative model's capacity to provide feedback for refinement [94].

The success of deep learning methods in image restoration can be attributed to their

ability to learn intricate and nonlinear mappings between degraded input and ground truth images. The models' capacity for automatic feature extraction and representation learning allows them to adapt to diverse data distributions, resulting in improved generalization performance. The availability of large-scale datasets has played a pivotal role in training these deep models effectively. Moreover, the continuous evolution of deep learning techniques, including the development of attention mechanisms [190], self-supervised learning [90], and transfer learning [192], further enhances the efficacy of image restoration algorithms. These advancements contribute to the field's ability to handle real-world challenges, such as varying lighting conditions, diverse image modalities, and complex noise patterns.

1.2.1 Deep Learning for Image Super-Resolution.

The pursuit of generating high-resolution images with enhanced edge structures and intricate texture details from low-resolution counterparts has been a focal point in computer vision. In a groundbreaking contribution, Dong et al.[37] introduced the Super-Resolution Convolutional Neural Network (SRCNN), marking a significant advancement in the application of deep learning to single-image super-resolution (SR). Building on the success of SRCNN, subsequent developments have witnessed the emergence of more efficient and intricate architectures. Shi et al.[142] proposed the Efficient Sub-Pixel Convolutional Network (ESPCN), incorporating a sub-pixel convolution layer to facilitate real-time SR. Lim et al.[103] elevated SR outcomes by strategically modifying ResNet, selectively omitting batch normalization layers to enhance model performance. Extending the capabilities of SR frameworks, Zhang et al.[190] introduced residual channel attention, augmenting the overall SR quality. Departing

from conventional Mean Squared Error (MSE)-minimizing approaches, contemporary methodologies integrate perceptual constraints to achieve superior visual quality. The Super-Resolution Generative Adversarial Network (SRGAN) [94] exemplifies this paradigm shift, leveraging generative adversarial networks and employing a multi-task loss function comprising MSE, perceptual, and adversarial components to predict high-resolution outputs. This dynamic landscape of deep learning methodologies underscores the continual pursuit of more sophisticated and effective solutions in the domain of image super-resolution. The evolution from SRCNN to advanced architectures reflects the ongoing commitment to pushing the boundaries of computational image enhancement.

1.2.2 Deep Learning for Image Denoising.

Image denoising, a pivotal task in the field of image processing, plays a crucial role in restoring the true representation of corrupted images by eliminating unwanted noise. In recent years, substantial strides have been made in this domain through the application of deep learning techniques. A notable contribution comes from Zhang et al. [181], who introduced the Denoising Convolutional Neural Network (DnCNN), a simple yet remarkably effective method that has established a new benchmark for denoising performance. Their research underscores the potency of integrating residual learning and batch normalization to enhance the overall denoising outcomes. In the pursuit of achieving clearer images, Mao et al. [117] developed the Residual Encoder-Decoder Network with 30 layers (RED30), a deep architecture characterized by multiple convolutions and subsequent transposed convolutions. This approach highlights the significance of employing intricate network architectures to capture

and restore intricate details in images. These advancements collectively emphasize the diverse approaches and innovations within the realm of deep learning for image denoising. Addressing the challenge of training complexity, Liu et al. [105] proposed the Multi-level Wavelet Convolutional Neural Network (MWCNN), which integrates a U-Net architecture with wavelet transformations to capture frequency features for image restoration tasks. This innovative combination showcases a thoughtful integration of traditional signal processing techniques with modern deep learning architectures. Tai et al. [150] introduced the Persistent Memory Network (MemNet), a deep architecture that employs recursive and gate units to recover high-quality images by delving into more accurate information. This approach underlines the importance of memory and recursive structures in learning long-range dependencies, thereby contributing to improved image denoising outcomes. In the quest for flexibility and speed, Zhang et al. [183] presented FFDNet, a fast and flexible denoising Convolutional Neural Network. FFDNet incorporates a tunable noise level map as input, catering to diverse denoising requirements. This adaptability is particularly valuable in real-world scenarios where noise characteristics can vary significantly.

1.2.3 Deep Learning for Image Deblurring.

In the realm of image processing, the challenge of image deblurring arises when confronted with a blurred image corrupted by an unknown blur kernel or a spatially variant kernel. The objective of image deblurring is to restore the sharp rendition of the original image by mitigating or eliminating the undesirable blur present in the degraded image. Noteworthy contributions to this field include the work of Sun et al. [147], who introduced a CNN-based model designed to estimate blur kernels and effectively

eliminate non-uniform motion blur. Chakrabarti [26], on the other hand, employs a network-centric approach to compute estimations of sharp images that have been blurred by elusive motion kernels. In the pursuit of enhancing deblurring efficacy, Nah et al. [123] proposed a multi-scale loss function, implementing a coarse-to-fine strategy and introducing an adversarial loss into the framework. Kupyn et al. [88] presented DeblurGAN, a model leveraging adversarial learning to eliminate blur kernels. Beyond CNN-centric approaches, RNN-based methodologies have emerged in the literature. Zhang et al. [180] proposed a spatially variant neural network, incorporating three CNNs and one RNN for comprehensive image deblurring. Additionally, Tao et al. [151] introduced SRN-DeblurNet, a model integrating one LSTM unit alongside CNNs to address multi-scale image deblurring challenges. Shen et al. [141] contributed a human-aware deblurring method aimed at selectively removing blur from foreground humans and background elements. Gao et al. [52] introduced a nested skip connection structure, showcasing state-of-the-art performance in image deblurring tasks. These diverse methodologies collectively contribute to the evolving landscape of image deblurring techniques, offering valuable insights and advancements in the pursuit of high-fidelity image restoration.

1.2.4 Deep Learning for Image Compression Artifacts Removal.

To address the challenge of mitigating artifacts induced by image compression, a multitude of methodologies have been advanced within the realm of image processing. Pioneering this endeavor, Dong et al. [38] leveraged the efficacy of deep learning, drawing inspiration from the remarkable achievements of super-resolution networks,

in order to effectively eliminate JPEG artifacts. In a similar vein, Guo et al. [61] devised a highly precise methodology for artifact removal in JPEG-compressed images. Their approach involved the joint learning of an intricate convolutional network operating seamlessly in both the Discrete Cosine Transform (DCT) and pixel domains. Building on this foundation, Zhang et al. [189] integrated batch normalization and residual learning strategies, strategically enhancing the training process and overall performance, particularly in the domain of general blind image restoration tasks. Fu et al. [49] introduced a novel paradigm by proposing a deep convolutional sparse coding network that amalgamates traditional model-based techniques with the power of deep learning. Ehrlich et al. [41] further expanded the frontier by training their networks with the incorporation of quantization tables as prior information. This innovative approach empowers a singular model to rectify artifacts across a spectrum of quality factors, achieving state-of-the-art results in the process.

1.2.5 Dataset

In the realm of deep learning-based image restoration methodologies, the significance of datasets, whether employed for model training or testing, cannot be overstated. A pivotal requirement for the efficacy of such methods lies in the inclusion of not only pristine images but also their corresponding degraded counterparts. Recent strides in research have witnessed a substantial augmentation in both the volume and diversity of images incorporated into datasets. Moreover, there has been a concerted effort to ensure that the degradation introduced in these datasets is more reflective of the complexities encountered in real-world scenarios. This evolving landscape of dataset composition and quality plays a pivotal role in advancing the robustness and

generalization capabilities of deep learning models in the domain of image restoration.

For the training and evaluation of super-resolution models, pivotal datasets such as Set5 [10], Set14 [178], Urban100 [68], Manga109 [50], and DIV2K [3] are extensively employed. It is noteworthy that these datasets exclusively comprise high-resolution images. Consequently, the corresponding low-resolution counterparts must be synthetically generated using assumed degradation models. The widely embraced bicubic downsampling model emerges as the predominant choice for both training and testing. The DIV2KRK [9] dataset stands out by introducing LR images through a distinctive process involving random kernel-blurring and downsampling of HR counterparts. RealSR [21] and DRealSR [166] datasets offer a unique perspective by providing HR and LR image pairs derived from identical scenes, achieved by adjusting the focal length of digital cameras. Meanwhile, the SupER [86] dataset introduces HR and LR image pairs through camera hardware binning, a technique that aggregates adjacent pixels on the sensor array. Furthermore, the ImagePairs [76] dataset contributes aligned HR and LR image pairs captured by separate HR and LR cameras mounted on a rig with a beam splitter.

Within the domain of image denoising, similar to challenges encountered in super-resolution studies, the evaluation of previous denoising algorithms has predominantly relied on synthetic data. This involves the introduction of additive white Gaussian noise at varying levels into a clean image to simulate the corresponding noisy input. Commonly utilized datasets for such synthetic evaluations include CBSD68 [118], Kodak24 [46], and McMaster [186], recognized for their widespread use in generating noisy images. Acknowledging the necessity to broaden assessments beyond synthetic contexts, datasets such as DND [130], SIDD [1], and PolyU [169] have been established

to address real-world image denoising challenges. These datasets comprise noisy and ground truth images captured under diverse lighting conditions, featuring distinct ISO values and exposure times.

In the pursuit of constructing a image deblurring dataset, Levin et al. [96] laid the groundwork by securing the camera on a tripod, deliberately inducing blur through actual camera shake. Building upon this foundation, Sun et al. [148] and Köhler et al. [85] expanded the dataset's richness by introducing a diverse set of blur kernels and incorporating a heightened degree of freedom in simulating camera shake. The dataset's evolution continued with Nah et al. [123], who innovatively devised the GoPro dataset. This expansive dataset seeks to emulate real-world blur scenarios through frame averaging. High-speed captures from a GoPro camera provide sharp images, which are subsequently averaged over time windows of varying durations to synthesize blurred counterparts, and the sharp image at the temporal midpoint of each time window serves as the ground truth reference.

The dataset employed for compression artifacts removal tasks is notably straightforward to construct, owing to the controlled nature of the degradation process induced by artificial compression algorithms. The construction of such a dataset involves the manipulation of pristine images through image compression algorithms, thereby introducing compression artifacts at various compression levels. This controlled approach ensures a systematic and reproducible generation of degradation. Widely adopted datasets for conducting comprehensive evaluations in this domain encompass Classic5 [45], LIVE1 [140], and BSD500 [5].

1.2.6 Image Quality Assessment

To comprehensively evaluate the efficacy of diverse image restoration algorithms, a crucial facet lies in image quality assessment. This pivotal process aims to precisely forecast the perceived quality of images as perceived by human observers. The landscape of image quality assessment can be broadly classified into two distinct categories: subjective assessment reliant on human perception and objective assessment grounded in quality metrics. Subjective assessment, often quantified through the Mean Opinion Score (MOS), entails the derivation of average scores through manual assignment by a multitude of human evaluators. This approach, while providing valuable insights, is burdened by notable drawbacks, including elevated costs, time-intensive nature, and vulnerability to individual preferences, thus rendering assessment results susceptible to personal biases.

In light of these limitations, the utilization of objective assessment methods has become prevalent in the evaluation of various image restoration algorithms. Notably, full-reference objective assessment metrics prove most accurate when ground truth images are available for comparison, while no-reference objective assessment metrics find utility when ground truth images are absent from the dataset. Among the most widely adopted full-reference metrics are Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) [164], and Learned Perceptual Image Patch Similarity (LPIPS) [188]. PSNR and SSIM primarily gauge disparities in signal characteristics, while LPIPS is employed to evaluate distinctions in human-perceived differences. Conversely, the realm of no-reference objective assessment metrics encompasses widely-used measures such as Naturalness Image Quality Evaluator (NIQE) [121] and No-Reference Quality Metric (NRQM) [112]. These metrics facilitate the quantitative evaluation of

image quality in the absence of direct comparisons with original images.

1.3 Universality Issue in Image Restoration Methods

As discussed in section 1.1, different image restoration tasks, such as super-resolution, deblurring and denoising, are the same of type of inverse problems. The only differences are in the type and severity of the degradations. The same mathematical formulation and similar architectures of restoration DNNs seem to suggest that a unified algorithms should be able to perform diverse image restoration tasks. Unfortunately the current methods fall far short of the universality expectation. Despite great efforts have been devoted to deep learning based image restoration with a large number of papers published in this fields, the issue of algorithm universality largely remains open.

The current state of the art is still at the level of one solution per degradation type or even per degradation severity. Most image restoration DNNs are designed for a rather narrow domain of degradation data. For example, networks specifically trained for super-resolution are not suited for denoising, while those trained for denoising will fail on supere-resolution tasks. Moreover, even within the same type of degradations, the DNN restoration methods cannot adapt well across different levels of degradations. For example, a model trained for the $2\times$ super-resolution task may falter when applied to $3\times$ super-resolution, and a denoising model trained for noise variance 15 may struggle if the noise variance increases to 25 at inference time.

Finally, real-world image degradations are caused by multiple compounded sources. In the image acquisition process, which is carried out by the image signal processing

pipeline (ISP) of the camera, each ISP step introduces some noises or artifacts, including sensor noises, insufficient sampling rate, color demosaicing errors, compression artifacts, camera jitters, etc. To aggravate the problem, these degradation sources are compounded to each other, making their effects very difficult to model precisely. Sequentially applying restoration algorithms designed to neutralize each of the degradation sources alone the ISP pipeline is too simplistic and cumbersome to remove the complex degradation effects. Furthermore, for the deep learning based image restoration approach, it is difficult and expensive to acquire or synthesize large training datasets of clean and degraded image pairs, due to the complex compound mechanism of the cascaded ISP degradation sources. In the current state of the art, any deviations in data domain between the training to inference stages can significantly reduce the performance of the learnt restoration DNN models. Therefore, making trained models more universal and robust against the complex reality of diverse degradation scenarios in practice is a much desired and worthy goal, which is the central topic of this thesis.

The lack of universality in restoration DNNs, which are specifically trained for particular image degradations, has two main drawbacks. Firstly, there is a decline in their restoration performance due to domain shift during inference. Secondly, there is a waste of resources caused by the need to train and store multiple models to address different degradations. Here, we will discuss these two points in detail.

The technical challenge for the DNN domain generalization is the well known fragility of deep learning models with respect to slight domain shifts; the good model performance heavily depends on the strict assumption of working on independent and identically distributed (i.i.d.) data in training and inference. However, this assumption, while foundational, is too idealistic and often unrealistic in practice.

This problem is particularly acute with deep neural networks, which are the most complicated but also the most fragile systems in modern machine learning. These networks become highly vulnerable when dealing with out-of-distribution (OOD) situations [102]. Despite many state-of-the-art image restoration networks performing well in controlled settings or on certain datasets, their usefulness is limited when faced with diverse and unpredictable conditions. These conditions include changes in lighting, shifts in image content, and the presence of complex artifacts. In such cases, the performance of deep neural networks can even decline to the levels comparable to the simplest baseline methods. This sensitivity to mismatches between assumed and real data distributions is a major hurdle in creating robust and widely applicable machine learning systems.

An important side benefit of unifying image restoration models for different degradation sources is improved computational and storage efficiencies of the deployed system. Nowadays, as DNN models are getting larger and larger, and accordingly their training costs are steadily increasing, training and storing multiple DNN models, one for each type of image degradation or even distinct levels of degradation severity, becomes inefficient and computationally expensive. This is especially true when restoration networks are deployed on edge devices with limited performance, such as smartphones. Collectively, multiple degradation-specific restoration networks put heavy resource burdens on end devices. These burdens can be greatly alleviated if these designated networks can be merged into one without material performance loss across different restoration tasks and degradation sources.

In this thesis, we depart from the current problem-specific methodology for image restoration and strive to improve the domain adaptability and efficiency of the existing

methods. We propose three novel methods to achieve the above goal; they are a new inference method, a new network model, and a new training method, respectively. In Chapter 2, we abstract any image degradation process as a many-to-one function and propose a general method with only one trained model for various image restoration problems. The general image restoration is formulated as a constrained optimization problem. Its objective is to maximize a posteriori probability of latent variables, and its constraint is that the image generated by these latent variables must be the same as the degraded image. In Chapter 3, we propose a novel system called the functional neural network (FuncNet) to solve a parametric image restoration problem with a single model. Unlike a plain neural network, the smallest conceptual element of our FuncNet is no longer a floating-point variable, but a function of the degradation intensity parameter of the problem. In Chapter 4, we propose a novel adversarial neural degradation (AND) model to solve the task of real-world super-resolution, which is the most common complex degradation combination. Instead of attempting to exhaust all degradation variants in simulation, which is unwieldy and impractical, the AND model, when trained in conjunction with a deep restoration neural network under a minmax criterion, can generate a wide range of highly nonlinear complex degradation effects without any explicit supervision.

Chapter 2

Maximum a Posteriori on a Submanifold: a General Image Restoration Method with GAN

We propose a general method for various image restoration problems, such as denoising, deblurring, super-resolution and inpainting. The method can use only one model, after only one-time training procedure, to handle any type of image degradation, as long as the degradation can be modeled and is differentiable. The problem is formulated as a constrained optimization problem. Its objective is to maximize a posteriori probability of latent variables, and its constraint is that the image generated by these latent variables must be the same as the degraded image. We use a Generative Adversarial Network (GAN) as our density estimation model. Convincing results are obtained on MNIST dataset.

2.1 Introduction

Image restoration has been researched for many years, but in a case-by-case way [127, 116, 59, 126, 181]. Almost all image restoration algorithms are only designed for certain type of images or degradation. This research paradigm has some obvious disadvantages. It is exhausting to invent new algorithms or train new models for slightly different situations. Even if we can, those specialized solutions are not so elegant, because they are very unlike one another even though the problems they focus on are fundamentally so similar.

It is worth noting that any image degradation process can be abstracted as a many-to-one function. More specifically, for any given degradation process, one degraded image could be degraded from any of many possible original images. From that point of view, we propose a general method for various image restoration problems, such as denoising, deblurring, super-resolution and inpainting. Our algorithm chooses the most probable original image from all those possible original images, and uses it as the restoration of the given degraded image. To be more precise, the general image restoration is formulated as a constrained optimization problem. Its objective is to maximize a posteriori probability of latent variables, and its constraint is that the image generated by these latent variables must be the same as the degraded image.

Recent progress of density estimation techniques makes our algorithm possible. In the field of image generation, Generative Adversarial Networks (GANs) make a huge success in recent years [55, 132, 16]. As research continues, images generated by GANs become more and more realistic and clear, and training procedure of GANs become more and more stable [139, 6]. Besides being an image generation technique, GANs can also be used for density estimation. The generator part of a GAN is an

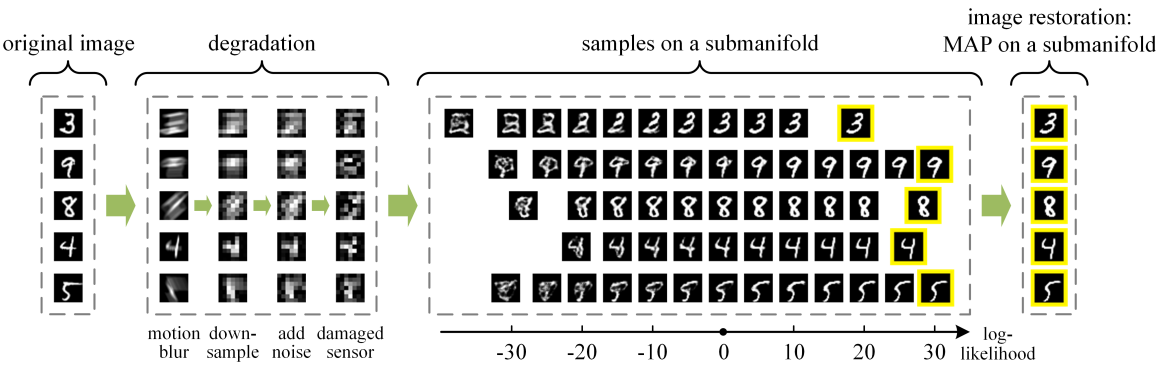


Figure 2.1: An illustration of our image restoration method.

implicit probability distribution model, and it will converge to a good estimator of the data distribution after training. In this work, we solve the inference problem with the probability density estimated by a GAN.

Figure 2.1 provides an illustration of how our image restoration method works. There are four dashed boxes from left to right in Figure 2.1, corresponding to four different phases of image capture and restoration process. Images in the first dashed box are original images, which are clear and undegraded. These images undergo a series of degradation in the second dashed box, and then are captured by our camera. In the image restoration process, we hope to estimate the original images with the degraded images we captured. As we pointed out before, every degraded image could be degraded from any of many possible original images. To be more precise, there is a particular subset of the original image manifold for any degraded image, and all image samples on the submanifold could be degraded to the given degraded image. Images in the third dashed box are those samples on the submanifold, and they are arranged in ascending order of log-likelihood from left to right. Images marked by yellow boxes are samples with the highest log-likelihood in their group, and they are placed in the last dashed box as restoration outputs. Overall, the contributions of

this work are mainly in two aspects:

1. We propose a general method for various image restoration problems. In the method, we explicitly use density information estimated by a GAN, an implicit model; and we directly solve the image restoration problem, an inference problem, with a GAN, a generative model. To the best of our knowledge, our work is the first to do those two things.

2. We propose a new algorithm to solve the optimization problem in our method. It is a first-order iterative algorithm for constrained problems, and it works well even for problems with highly nonlinear objective and constraints. These features make it especially suited to neural network related constrained optimization problems.

2.2 Related Work

The most similar works to ours are found in [171, 172]. [171] propose a image inpainting method, which can generate missing content with a trained GAN. They search in the latent space of the GAN for the image which is close to the corrupted image, and use the discriminator loss as an indicator of how realistic their restoration is. [172] then improve their theory and apply it to various image restoration problems. Unfortunately, there is a major theoretical flaw in their method. [55] prove that the discriminator is unable to identify how realistic an input is after several steps of training, if the GAN has enough capacity. During the training, the information of the data distribution gradually transfer from the discriminator to the generator. Ideally, the generator will have all the information of the data distribution while the discriminator will have

none. That is why we use the generator instead of discriminator to measure how realistic the restoration is. Even worse, [172] ignore a term $|\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}}|$ intentionally in their Eq. (5), because they think it is intractable. We will demonstrate in Section 2.3.1 that $|\frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}}|$ directly determines the density of data space. Another difference between their work and ours is that we use a more radical strategy of optimization. They simply add their image prior term to their distortion. This will lead to a compromise between plausibility and visual quality of restoration. However, we choose the most probable image only from images which could degrade to the input. This makes our restorations more plausible while still keeps them similar to the real.

The maximum a posteriori (MAP) has existed for a long time as a classic estimation method [154, 23]. But before GANs, people do not have a probability density model which is good enough to describe the distribution of images. After GANs make a huge success in image generation, researchers start to use them in image restoration tasks to get more realistic results [72, 15]. [94] and [144] try to use the MAP estimation on GANs to solve image super-resolution problem. However, they only use the MAP estimation implicitly and indirectly, while our method use it explicitly and directly. We suspect that all methods do implicit MAP estimation on GANs would require redesigning or retraining when the image restoration task changes, and this makes implicit methods not as general as our explicit method.

[157] is another work which is seemingly similar to ours, but they are actually quite different. They use a randomly-initialized neural network as a prior to solve image restoration problems. The prior in their method is elaborate, neural network related but still handcrafted, while in our method the prior is learned from data. So our data-driven prior has better adaptability to specific image distribution.

2.3 Maximum a Posteriori on a Submanifold

2.3.1 Formulation

Consider a general image degradation model $\tilde{\boldsymbol{x}} = F(\boldsymbol{x}, \boldsymbol{\Omega})$, where $\boldsymbol{x}$, $\tilde{\boldsymbol{x}}$, and $\boldsymbol{\Omega}$ represent the original image, the degraded image, and the parameters of the degradation model, respectively. The image degradation function F is a deterministic function. That means, given an original image $\boldsymbol{x}$ and a particular set of parameters $\boldsymbol{\Omega}$, the image degradation model will always produce the same degraded image $\tilde{\boldsymbol{x}}$.

Our goal is to get a reasonable estimate of $\boldsymbol{x}$ with given $\tilde{\boldsymbol{x}}$ and F. In this paper, we use the maximum a posteriori probability (MAP) estimate of $\boldsymbol{x}$ as the restoration of $\tilde{\boldsymbol{x}}$. Compared to MSE-based method, MAP estimate of $\boldsymbol{x}$ is perceptually more convincing [94, 14]. We can perform inference by maximizing the posterior $p(\boldsymbol{x}, \boldsymbol{\Omega}|\tilde{\boldsymbol{x}})$:

$$\begin{aligned}\{\hat{\boldsymbol{x}}, \hat{\boldsymbol{\Omega}}\} &= \underset{\boldsymbol{x},\boldsymbol{\Omega}}{\operatorname{argmax}}\, p(\boldsymbol{x}, \boldsymbol{\Omega}|\tilde{\boldsymbol{x}}) \\ &= \underset{\boldsymbol{x},\boldsymbol{\Omega}}{\operatorname{argmax}}\, \frac{p(\tilde{\boldsymbol{x}}|\boldsymbol{x}, \boldsymbol{\Omega})p(\boldsymbol{x}|\boldsymbol{\Omega})p(\boldsymbol{\Omega})}{p(\tilde{\boldsymbol{x}})}\end{aligned} \tag{2.3.1}$$

where $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{\Omega}}$ represent MAP estimate of $\boldsymbol{x}$ and $\boldsymbol{\Omega}$. Note that $p(\tilde{\boldsymbol{x}})$ is always positive and does not depend on $\boldsymbol{x}$ and $\boldsymbol{\Omega}$, and typically we assume that $\boldsymbol{x}$ and $\boldsymbol{\Omega}$ are independent. Therefore,

$$\{\hat{\boldsymbol{x}}, \hat{\boldsymbol{\Omega}}\} = \underset{\boldsymbol{x},\boldsymbol{\Omega}}{\operatorname{argmax}}\, p(\tilde{\boldsymbol{x}}|\boldsymbol{x}, \boldsymbol{\Omega})p(\boldsymbol{x})p(\boldsymbol{\Omega}) \tag{2.3.2}$$

Note that $\tilde{\boldsymbol{x}} = F(\boldsymbol{x}, \boldsymbol{\Omega})$ is a deterministic function, i.e., $p(\tilde{\boldsymbol{x}}|\boldsymbol{x}, \boldsymbol{\Omega}) = \delta(\tilde{\boldsymbol{x}} - F(\boldsymbol{x}, \boldsymbol{\Omega}))$.

Therefore, the estimation is equivalent to

$$\begin{aligned} \{\hat{\boldsymbol{x}}, \hat{\boldsymbol{\Omega}}\} = \underset{\boldsymbol{x},\boldsymbol{\Omega}}{\operatorname{argmax}} \quad & p(\boldsymbol{x})p(\boldsymbol{\Omega}) \\ \text{s.t.} \quad & \|\tilde{\boldsymbol{x}} - F(\boldsymbol{x}, \boldsymbol{\Omega})\| = 0 \end{aligned} \tag{2.3.3}$$

Here we write $p(\boldsymbol{x})$ more specifically as $p_r(\boldsymbol{x})$, which stand for the probability density of real data distribution. We can estimate $p_r(\boldsymbol{x})$ with the generator part of a trained GAN, which is an implicit probability distribution model with distribution $p_G(\boldsymbol{x})$. The trained generator G represents a mapping from latent space of $\boldsymbol{z}$ to data distribution of original image $\boldsymbol{x}$, i.e., $p_r(\boldsymbol{x}) = p_G(\boldsymbol{x})$, and $p_G(\boldsymbol{x})$ is a probability density function implicitly defined by $\boldsymbol{x} = G(\boldsymbol{z})$, where $\boldsymbol{z}$ is typically sampled from some simple distribution, such as the uniform distribution or the normal distribution. Assuming $G : \mathbf{R}^n \to \mathbf{R}^m$ is an injective function, the estimation is equivalent to

$$\begin{aligned} \{\hat{\boldsymbol{z}}, \hat{\boldsymbol{\Omega}}\} = \underset{\boldsymbol{z},\boldsymbol{\Omega}}{\operatorname{argmax}} \quad & p_G(G(\boldsymbol{z}))p(\boldsymbol{\Omega}) \\ \text{s.t.} \quad & \|\tilde{\boldsymbol{x}} - F(G(\boldsymbol{z}), \boldsymbol{\Omega})\| = 0 \end{aligned} \tag{2.3.4}$$

$$\text{and} \quad \hat{\boldsymbol{x}} = G(\hat{\boldsymbol{z}}) \tag{2.3.5}$$

Generally the dimension of vector space of $\boldsymbol{z}$ is far lower than the dimension of vector space of $\boldsymbol{x}$. Note that $p_G(\boldsymbol{x})$ is nonnegative if and only if $\boldsymbol{x}$ is on the low dimensional manifold $\mathcal{M}$ defined by $\boldsymbol{x} = G(\boldsymbol{z})$, we can replace the probability density on the original space $p_G(G(\boldsymbol{z}))$ in Eq. (2.3.4) by the probability density on the manifold $p_{\mathcal{M}}(\boldsymbol{z})$, and end up with the same estimation result $\hat{\boldsymbol{z}}$. According to [129], the probability density

on the manifold can be calculated by

$$p_{\mathcal{M}}(\boldsymbol{z}) = \frac{p(\boldsymbol{z})}{\sqrt{\det Gram(\frac{\partial G}{\partial \boldsymbol{z}_1}, \ldots, \frac{\partial G}{\partial \boldsymbol{z}_n})}} \tag{2.3.6}$$

where $Gram$ represents the Gram matrix, and $\sqrt{\det Gram(\frac{\partial G}{\partial \boldsymbol{z}_1}, \ldots, \frac{\partial G}{\partial \boldsymbol{z}_n})}$ is the volume of the parallelotope spanned by the vectors $(\frac{\partial G}{\partial \boldsymbol{z}_1}, \ldots, \frac{\partial G}{\partial \boldsymbol{z}_n})$, so the square root of the Gram determinant can serve as a local scale factor. It has an effect similar to the Jacobian determinant, but we can only use the Gram determinant here because G is a function from $\mathbf{R}^n$ to $\mathbf{R}^m$, and generally n is much less than m.

The Gram matrix can be simply calculated by $Gram(\frac{\partial G}{\partial \boldsymbol{z}_1}, \ldots, \frac{\partial G}{\partial \boldsymbol{z}_n}) = \boldsymbol{V}^T\boldsymbol{V}$, where $\boldsymbol{V}$ is an $m \times n$ matrix, whose entries are given by $\boldsymbol{V}_{ij} = \frac{\partial \boldsymbol{x}_i}{\partial \boldsymbol{z}_j}$. Therefore, Eq. (2.3.4) is equivalent to

$$\begin{aligned} \{\hat{\boldsymbol{z}}, \hat{\boldsymbol{\Omega}}\} = \underset{\boldsymbol{z}, \boldsymbol{\Omega}}{\operatorname{argmax}} \quad & \frac{p(\boldsymbol{z})p(\boldsymbol{\Omega})}{\sqrt{\det \boldsymbol{V}^T\boldsymbol{V}}} \\ \text{s.t.} \quad & \|\tilde{\boldsymbol{x}} - F(G(\boldsymbol{z}), \boldsymbol{\Omega})\| = 0 \end{aligned} \tag{2.3.7}$$

To solve the estimation problem efficiently, we represent probabilities in Eq. (2.3.7) in logarithmic space, i.e.,

$$\log \frac{p(\boldsymbol{z})p(\boldsymbol{\Omega})}{\sqrt{\det \boldsymbol{V}^T\boldsymbol{V}}} = -\frac{1}{2}\log\det \boldsymbol{V}^T\boldsymbol{V} + \log p(\boldsymbol{z}) + \log p(\boldsymbol{\Omega}) \tag{2.3.8}$$

Matrix $\boldsymbol{V}^T\boldsymbol{V}$ is a positive-definite matrix, so we can use Cholesky decomposition to calculate $\log\det \boldsymbol{V}^T\boldsymbol{V}$ efficiently, i.e.,

$$\log\det \boldsymbol{V}^T\boldsymbol{V} = 2\,\mathrm{tr}(\log(\mathrm{chol}(\boldsymbol{V}^T\boldsymbol{V}))) \tag{2.3.9}$$

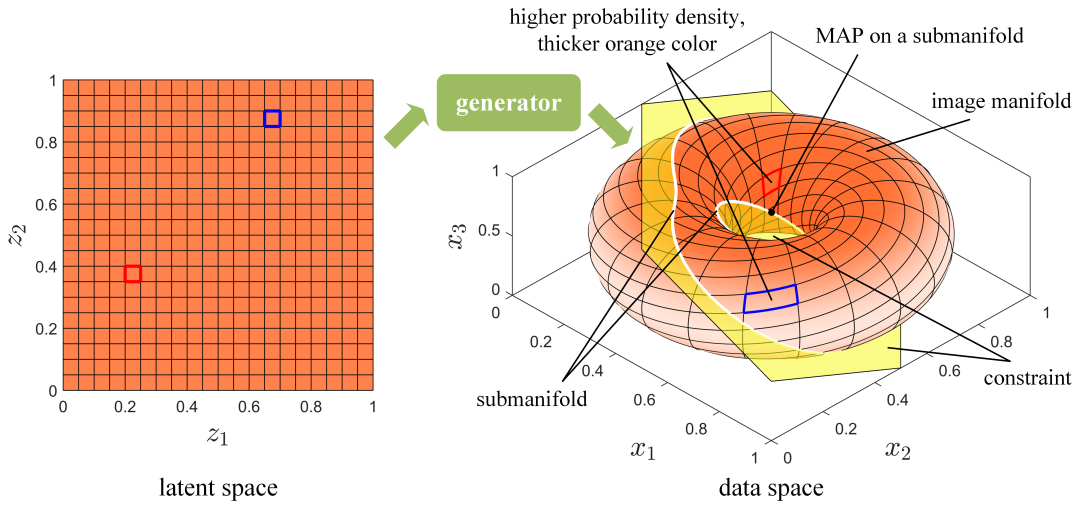


Figure 2.2: A toy example to show the basic idea of our formulation.

Finally we deduce a set of expressions which can be calculated directly, and their final outcome $\hat{\boldsymbol{x}}$ is the restored image we want, i.e.,

$$\begin{aligned} \{\hat{\boldsymbol{z}}, \hat{\boldsymbol{\Omega}}\} = \underset{\boldsymbol{z},\boldsymbol{\Omega}}{\operatorname{argmax}} \quad & -\operatorname{tr}(\log(\operatorname{chol}(\boldsymbol{V}^T\boldsymbol{V}))) + \log p(\boldsymbol{z}) + \log p(\boldsymbol{\Omega}) \\ \text{s.t.} \quad & \|\tilde{\boldsymbol{x}} - F(G(\boldsymbol{z}), \boldsymbol{\Omega})\| = 0 \end{aligned} \tag{2.3.10}$$

$$\text{and} \quad \hat{\boldsymbol{x}} = G(\hat{\boldsymbol{z}}) \tag{2.3.11}$$

Note that $(G(\boldsymbol{z}), \boldsymbol{\Omega})$ form a low dimensional manifold which is embedded in the space of $(\boldsymbol{x}, \boldsymbol{\Omega})$, and the feasible solutions of Eq. (2.3.10) is on a subset of the manifold, which is defined by $\|\tilde{\boldsymbol{x}} - F(G(\boldsymbol{z}), \boldsymbol{\Omega})\| = 0$. So our method basically makes a MAP estimate on a submanifold.

Figure 2.2 is a toy example to show the basic idea of our formulation in a very visible way. Suppose there is a grayscale original image $\boldsymbol{x}$, which has only three pixels. Then it is downsampled to only one pixel during the image capture process, and our task is to estimate $\boldsymbol{x}$ with the one pixel image we captured. Suppose we have trained

a GAN as an implicit model of data distribution of $\boldsymbol{x}$. More specifically, the generator of the trained GAN represents a mapping from its input noise $\boldsymbol{z}$ to data distribution of $\boldsymbol{x}$. The left part of Figure 2.2 describes the two dimensional latent space of $\boldsymbol{z}$. We use the saturation of orange color to represent probability density level, i.e., a thicker orange color means a higher probability density. So the uniform orange color in the latent space means that the input noise $\boldsymbol{z}$ is sampled from a uniform distribution.

Then the two dimensional vector $\boldsymbol{z}$ is mapped to three dimensional space of image $\boldsymbol{x}$ by the generator of the trained GAN, and the big orange square in the latent space of $\boldsymbol{z}$ is transformed into a twisted torus in the three dimensional data space of $\boldsymbol{x}$, which is described in the right part of Figure 2.2. Some areas in space of $\boldsymbol{z}$ expand during the transformation, while other areas shrink. We can find this out by comparing the red and blue quadrilateral between the latent and data space. Therefore, the probability density on the torus is no longer uniform. The orange colors of the expanded areas become lighter, and the colors of the shrunken areas become thicker. Quantitatively speaking, the square root of the Gram determinant in Eq. (2.3.6) is the local area scale factor of the mapping, and its inverse, of course, is the local density scale factor.

The pale yellow plane in the data space represents the constraint in the toy example. All points on the plane would exactly be downsampled to the one pixel image we captured. So the intersection curve of the plane and the torus is the submanifold we are looking for, and that white curve is the feasible set of the toy problem. In this problem, $p(\boldsymbol{z})$ is a constant in the domain, and degradation parameters $\boldsymbol{\Omega}$ does not exist at all. According to Eq. (2.3.7), what we need to do is to maximize the inverse of the square root of the Gram determinant on the submanifold. In other words, the point with the thickest orange color on the intersection curve is the restored image

$\hat{\boldsymbol{x}}$, the MAP estimate on the submanifold. We can find out that the method is both intuitive and rational for this toy example.

2.3.2 Optimization Algorithm

We propose a new optimization algorithm to solve Eq. (2.3.10). Note that the objective function and the equality constraint in Eq. (2.3.10) are both highly nonlinear, so gradient-based method seems a natural choice for the problem. Our algorithm is inspired by Projected Gradient Descent Method.

To solve an unconstrained problem with ordinary Gradient Descent Method, we take small steps in the direction of the negative gradient. To solve a constrained problem, we can try to use Projected Gradient Descent Method, take small step as usual and then project variables back onto the feasible set. But unfortunately, Projected Gradient Descent Method is only valid for problems with very simple feasible set, such as solution set of linear equations, some simple polyhedra and simple cone, etc. If constraints of a problem is too complex, like the constraint in Eq. (2.3.10), it is very hard to project variables back onto the feasible set.

To overcome this shortage, we propose a new optimization algorithm called Quasi Projected Gradient Descent Method. In our algorithm, the gradient information is not only used to improve the objective function, but helps to satisfy the constraints as well. Consider the standard form of continuous optimization problem,

$$\begin{aligned} \underset{\boldsymbol{u}}{\text{minimize}} \quad & f(\boldsymbol{u}) \\ \text{s.t.} \quad & h_i(\boldsymbol{u}) = 0, \quad i = 1, \ldots, m \\ & h_j(\boldsymbol{u}) \leq 0, \quad j = m+1, \ldots, m+p \end{aligned} \tag{2.3.12}$$

where $f, h_i, h_j : \mathbf{R}^n \to \mathbf{R}$, and they are all highly nonlinear. Algorithm 1 is the proposed algorithm for the problem.

Algorithm 1 Quasi Projected Gradient Descent Method

Input: objective function $f(\boldsymbol{u})$, equality constraints $h_i(\boldsymbol{u})$ and inequality constraints $h_j(\boldsymbol{u})$

Parameter: step size $\eta_{\|}$ and $\eta_{\perp}$, positive factors c_i and c_j, number of iterations n, small positive constant ϵ for numerical stability, initial guess $\boldsymbol{u}_0$

Output: local optimum $\boldsymbol{u}_n$

Let $h(\boldsymbol{u}) = \sum_{i=1}^{m} c_i \cdot \|h_i(\boldsymbol{u})\|^2 + \sum_{j=m+1}^{m+p} c_j \cdot H(h_j(\boldsymbol{u})) \cdot \|h_j(\boldsymbol{u})\|^2$, where H represents the Heaviside step function

for $i = 1$ **to** n **do**

$\boldsymbol{g}_f = \nabla f(\boldsymbol{u}_{i-1})$

$\boldsymbol{g}_h = \nabla h(\boldsymbol{u}_{i-1})$

$\boldsymbol{g}_{\|} = \boldsymbol{g}_f - \frac{\boldsymbol{g}_f \cdot \boldsymbol{g}_h}{\boldsymbol{g}_h \cdot \boldsymbol{g}_h + \epsilon} \cdot \boldsymbol{g}_h$

$\boldsymbol{g}_{\perp} = \boldsymbol{g}_h$

$\boldsymbol{u}_i = \boldsymbol{u}_{i-1} - \eta_{\|} \cdot \boldsymbol{g}_{\|}$ (or use more advanced optimizer)

$\boldsymbol{u}_i = \boldsymbol{u}_i - \eta_{\perp} \cdot \boldsymbol{g}_{\perp}$ (or use more advanced optimizer)

end for

return $\boldsymbol{u}_n$

To solve Eq. (2.3.10) with Algorithm 1, we only need to set $\boldsymbol{u} = \{\hat{\boldsymbol{z}}, \hat{\boldsymbol{\Omega}}\}$, objective function $f(\boldsymbol{u}) = -(-\operatorname{tr}(\log(\operatorname{chol}(\boldsymbol{V}^T\boldsymbol{V}))) + \log p(\boldsymbol{z}) + \log p(\boldsymbol{\Omega}))$, and the only equality constraint function $h_1(\boldsymbol{u}) = \|\tilde{\boldsymbol{x}} - F(G(\boldsymbol{z}), \boldsymbol{\Omega})\|$.

In the proposed algorithm, we first define an overall constraint function $h(\boldsymbol{u}) : \mathbf{R}^n \to \mathbf{R}_{\geq 0}$, and the feasible set of the optimization problem is the region where $h(\boldsymbol{u}) = 0$. In each iteration of the algorithm, we calculate the gradients of $f(\boldsymbol{u})$ and $h(\boldsymbol{u})$ at $\boldsymbol{u}_{i-1}$. If we take a small step in the direction of the negative $\boldsymbol{g}_f$, the value

of $f(\boldsymbol{u})$ will decrease a little bit, but it may have a unwanted impact on the value of $h(\boldsymbol{u})$. In order to avoid this problem, we calculate $\boldsymbol{g}_{\parallel}$, the tangential component of $\boldsymbol{g}_f$ on the isocontour of $h(\boldsymbol{u}_{i-1})$, which can be calculated by vector rejection of $\boldsymbol{g}_f$ on $\boldsymbol{g}_h$. In each iteration, we actually take a small step in the direction of the negative $\boldsymbol{g}_{\parallel}$, the value of $f(\boldsymbol{u})$ will still decrease, while it has almost no impact on the value of $h(\boldsymbol{u})$. We also take a small step in the direction of the negative $\boldsymbol{g}_{\perp}$, i.e., $\boldsymbol{g}_h$ itself, which is perpendicular to the isocontour of $h(\boldsymbol{u}_{i-1})$. Repeat these steps, and the sequence $\boldsymbol{u}$ will converge to the desired optimal solution.

Behaviors of our Quasi Projected Gradient Descent Method is similar to behaviors of the original Projected Gradient Descent Method. Consider a point $\boldsymbol{u}$ which is very close to the feasible region. The summation of two moves against $\boldsymbol{g}_{\parallel}$ and $\boldsymbol{g}_{\perp}$ is actually an inaccurate Projected Gradient Descent. That is why we name our method as Quasi Projected Gradient Descent Method.

Here we use the same toy example we used in Section 2.3.1, to show how our Quasi Projected Gradient Descent Method works. In Figure 2.3, solid curves in black and white are isocontour of constraint function h. The whiter the curve, the lower value of h it corresponds; Dashed lines in color are isocontour of objective function f. The redder the line, the lower value of f it corresponds. Note that the white solid curve is the feasible set of the toy problem, so intersection points of the white solid curve and the red dashed line in the latent space is $\hat{\boldsymbol{z}}$ in Eq. (2.3.10), while the intersection points in the data space is $\hat{\boldsymbol{x}}$ in Eq. (2.3.11).

Our iterative optimization algorithm starts from the bottom left corner of the latent space. The red vector is a gradient step of h. It is pointing towards the direction of the negative $\boldsymbol{g}_h$, and is perpendicular to the black solid curve, an isocontour of

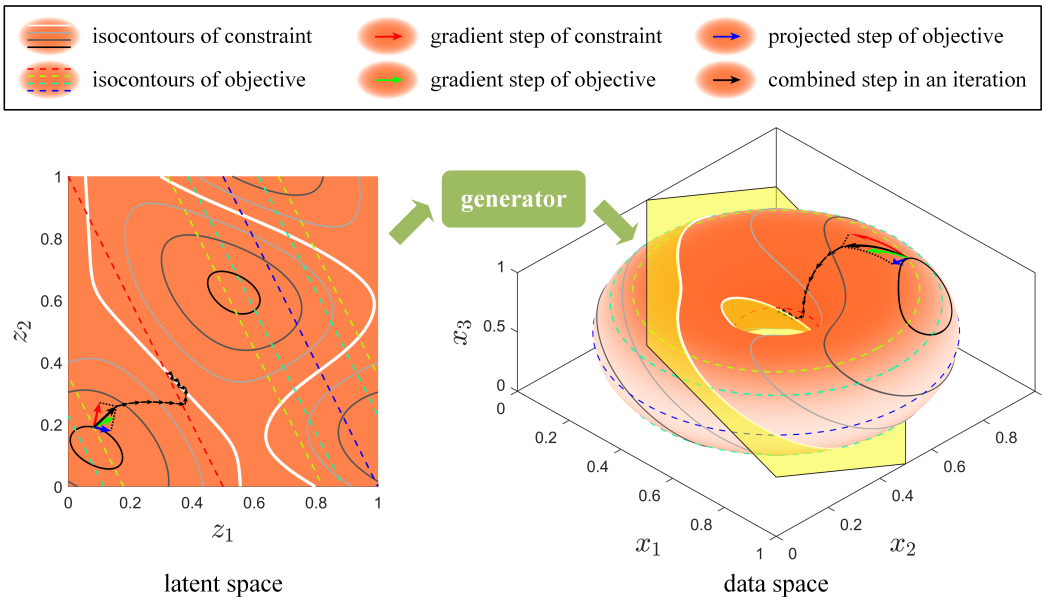


Figure 2.3: A toy example to show how our Quasi Projected Gradient Descent Method works.

h. The green vector is a gradient step of f. It is pointing towards the direction of the negative $\boldsymbol{g}_f$, and is perpendicular to the yellow dashed line, an isocontour of f. The blue vector is a projected gradient step. It is pointing towards the direction of the negative $\boldsymbol{g}_{\|}$, and is the tangential component of the green vector on the black solid curve, which can be calculated by vector rejection of the green vector on the red vector. We only plot green, red and blue vector for the first iteration to keep Figure 2.3 clean and easy to understand. Black vectors are combined gradient steps, which are vector sums of red and blue vectors. We move along these black vectors and we can find out that our optimization algorithm reaches a desired solution quickly.

2.4 Experiments

In this Section, we use MNIST dataset [92] to test our image restoration method. The dataset is divided in 50k for the training set, 10k for each of the validation and test set. We use a WGAN-GP [60] trained on the training set as the density estimation model. The architecture of the WGAN-GP we used is shown in Table 2.1 and Table 2.2, and we add a L2 weight decay term with decay parameter of 0.001 on the generator loss to prevent over-fitting. The network we used is very simple, but it is enough to prove the effectiveness of our method.

Table 2.1: Architecture of the generator

	Kernel size	Output shape
$\boldsymbol{z}$	-	16
Linear, tanh	-	$64 \times 4 \times 4$
Deconv, tanh	5×5	$32 \times 7 \times 7$
Deconv, tanh	5×5	$16 \times 14 \times 14$
Deconv, sigmoid	5×5	$1 \times 28 \times 28$

Table 2.2: Architecture of the discriminator

	Kernel size	Output shape
$G(\boldsymbol{z})$	-	$1 \times 28 \times 28$
Conv, LeakyReLU	5×5	$16 \times 14 \times 14$
Conv, LeakyReLU	5×5	$32 \times 7 \times 7$
Conv, LeakyReLU	5×5	$64 \times 4 \times 4$
Linear	-	1

We use four different kinds of degradation to test the generality of our method. The first three kinds of degradation are relatively simple. They are 7× downsampling, making a 14×14 square hole in the center of the image, and adding Gaussian white noise with a standard deviation of 1.0, respectively. The last kind of degradation is a

Table 2.3: Quantitative comparison with other general image restoration methods.

	Downsample		Hole		Noise		Composition	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Total Variation	12.88	0.4556	11.06	0.5928	11.94	0.1736	12.64	0.2559
[172]	13.23	0.6612	12.13	0.6205	11.99	0.5910	12.79	0.6434
Ours	**17.02**	**0.8287**	**14.63**	**0.7815**	**14.47**	**0.7238**	**14.73**	**0.7403**

composition of a series of degradation in order, which are (a) adding linear motion blur by at most 14 pixels in any direction, (b) 4× downsampling, (c) adding uniform noise between -0.05 and 0.05, (d) randomly removing 10% of the pixels.

We use two independent ADAM optimizer [82] with $\boldsymbol{g}_{\parallel}$ and $\boldsymbol{g}_{\perp}$ respectively in the Quasi Projected Gradient Descent Method. For all four kinds of degradation, we run the algorithm with the same settings. Settings for both ADAM optimizer are learning rate $\alpha = 0.01$ (decayed linearly to 0), $\beta_1 = 0.9$, $\beta_2 = 0.99$, and number of iterations $n = 500$.

In the experiments, we compare our method with two other general image restoration methods. The first is Total Variation (TV) [2], a traditional method; and the second is [172], a GAN based state-of-the-art approach. We use the SSIM index [164] and PSNR value as quantitative metrics for these restoration methods, and the results are shown in Table 2.3. We also present some restoration images in Table 2.4 without cherry-picking. We can find out that our general image restoration method is better than two baseline methods by large margins. This is due to the more accurate prior information of images and the more radical strategy of optimization in our method.

Table 2.4: Visual comparison with other general image restoration methods.

	Downsample	Hole	Noise	Composition
Original image				
Degraded image				
Total Variation				
[172]				
Ours				

2.5 Conclusions

We propose a general image restoration method in this work. Compared with traditional image restoration algorithms, our method is much more powerful. Image restoration is an inherently ill-posed problem, so additional prior knowledge is needed. In our method, we use all prior knowledge of original images, i.e., the probability distribution of original images; and we use all prior knowledge of degradation, i.e., the degradation model itself. Traditional image restoration like Total Variation, by contrast, just uses a small part of the prior, typically some statistical properties. Besides, unlike our method, there is usually no guarantee that an output restoration from a traditional method can be degraded back accurately to its input. This makes restorations from a traditional method less plausible than restorations from our method.

To solve the constrained optimization problem in our method, we propose a new first-order iterative algorithm. It works well even for problems with highly nonlinear objective and constraints. These features make it especially suited to neural network related constrained optimization problems. To our best knowledge, it is the first gradient-based method which can handle highly nonlinear constraints directly, rather than convert the problem into an equivalent unconstrained one.

For future work, we think our method can be straightforwardly extended to other domains which GANs are gifted in, such as video, audio and language. We will try to solve restoration problems and other inference problems in these domains with our paradigm. The convergence and other properties of the Quasi Projected Gradient Descent Method would be interesting as well.

Chapter 3

Functional Neural Networks for Parametric Image Restoration Problems

Almost every single image restoration problem has a closely related parameter, such as the scale factor in super-resolution, the noise level in image denoising, and the quality factor in JPEG deblocking. Although recent studies on image restoration problems have achieved great success due to the development of deep neural networks, they handle the parameter involved in an unsophisticated way. Most previous researchers either treat problems with different parameter levels as independent tasks, and train a specific model for each parameter level; or simply ignore the parameter, and train a single model for all parameter levels. The two popular approaches have their own shortcomings. The former is inefficient in computing and the latter is ineffective in performance. In this work, we propose a novel system called functional neural network (FuncNet) to solve a parametric image restoration problem with a single model. Unlike

a plain neural network, the smallest conceptual element of our FuncNet is no longer a floating-point variable, but a function of the parameter of the problem. This feature makes it both efficient and effective for a parametric problem. We apply FuncNet to super-resolution, image denoising, and JPEG deblocking. The experimental results show the superiority of our FuncNet on all three parametric image restoration tasks over the state of the arts.

3.1 Introduction

Image restoration [120] is a classical yet still active topic in low-level computer vision, which estimates the original image from a degraded measurement. For example, single image super-resolution [47] estimates the high-resolution image from a downsampled one, image denoising [17] estimates the clean image from a noisy one, and JPEG deblocking [27] estimates the original image from a compressed one. It is a challenging ill-posed inverse problem which aims to recover the information lost to the image degradation process [12], and it is also important since it is an essential step in various image processing and computer vision applications [195, 173, 11, 97, 145, 89, 63].

Almost every single image restoration problem has a closely related parameter, such as the scale factor in super-resolution, the noise level in image denoising, and the quality factor in JPEG deblocking. The parameter in an image restoration problem tends to have a strong connection with the image degradation process. In the super-resolution problem, the blur kernel of the downsampling process is determined by the scale factor [184]. In the image denoising problem, the standard deviation of the additive white Gaussian noise is determined by the noise level [34]. In the JPEG deblocking problem, the quantization table for DCT coefficients is determined by the

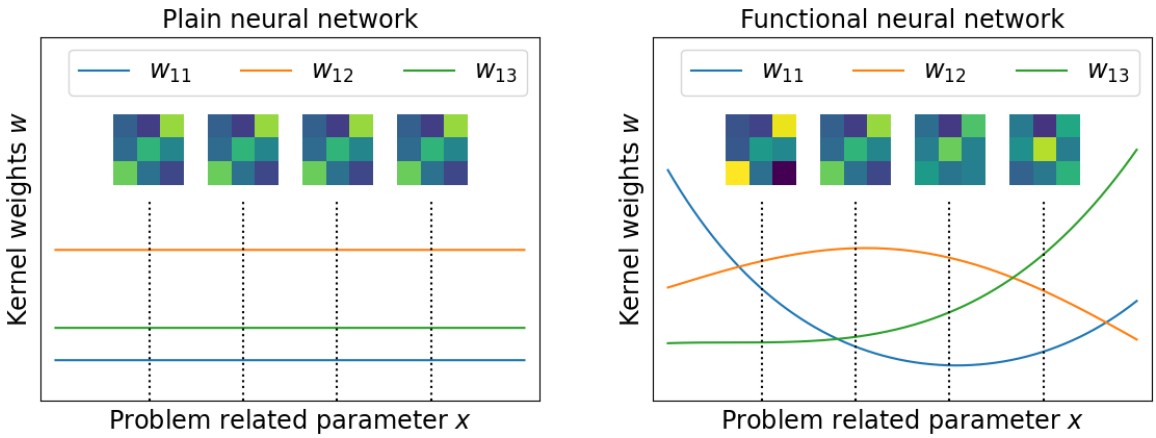


Figure 3.1: The difference between a plain neural network and our functional neural network (FuncNet). The left and right figure visualize a 3×3 convolution kernel in a plain neural network and its counterpart in a FuncNet respectively. For the kernel in a plain network, its weights remain unchanged for different problem related parameter levels, so the network only has a limited adaptability to parametric image restoration problems. Unlike a plain network, the smallest conceptual element of our FuncNet is no longer a floating-point variable, but a function of the problem related parameter. In other words, the kernel weights of our FuncNet can change for different situations and make our FuncNet perform better for parametric image restoration problems.

quality factor [128]. When we try to restore the clean image from a corrupt one, we might know the value of the corresponding parameter for various reasons. In the super-resolution problem, the scale factor is specified by users [79]. In the image denoising problem, the noise level could be measured by other devices [131, 106]. In the JPEG deblocking problem, the quality factor could be derived from the header of the JPEG file [31]. Therefore, it is very important to use the known parameter well in such a parametric image restoration problem.

Recently, deep convolutional neural network based methods are widely used to tackle the image restoration tasks, including super-resolution [37, 79, 142, 103, 149, 155, 191, 190, 35, 125], image denoising [73, 19, 182, 95, 150, 181, 183, 175, 4], and JPEG deblocking [107, 38, 165, 61, 25, 51, 105, 189, 41]. They have achieved significant improvements over conventional image restoration methods due to their powerful learning ability. However, they have not been paying attention to the parameter involved in an image restoration problem, and handled it in an unsophisticated way. Most previous researchers either treat problems with different parameter levels as independent tasks, and train a specific model for each parameter level [37, 142, 190, 181, 189]; or simply ignore the parameter, and train a single model for all parameter levels [79, 175]. The two popular approaches have their own shortcomings. The former is inefficient in computing, because they may have to train and store dozens of models for different parameter levels. The latter is ineffective in performance, since they ignore some important information that could have helped the restoration process.

To overcome these weaknesses, we propose a novel system called functional neural network (FuncNet) to solve a parametric image restoration problem with a single model. The difference between a plain neural network and our FuncNet is shown in

Figure 4.1. Unlike a plain neural network, the smallest conceptual element of our FuncNet is no longer a floating-point weight, but a function of the parameter of the problem. When we train a FuncNet, we gradually change these functions to reduce the value of the loss function. When we use the trained FuncNet to restore a degraded image with a given parameter, we first evaluate those functions with the parameter, use the evaluation values as weights of the network, and then do inference as normal. This feature makes it both efficient and effective for a parametric problem. By this way, we neatly blend the parameter information into a neural network model, use it to help the restoration process, and only increase a negligible amount of computation as we will demonstrate later. We apply FuncNet to super-resolution, image denoising, and JPEG deblocking. The experimental results show the superiority of our FuncNet on all three parametric image restoration tasks over the state of the arts.

The remainder of the chapter is organized as follows. Section 3.2 provides a brief survey of related work. Section 3.3 presents the proposed FuncNet model, discusses the details of implementation, and analyses its storage and computational efficiency. In Section 3.4, extensive experiments are conducted to evaluate FuncNets on three parametric image restoration tasks. Section 3.5 concludes the paper.

3.2 Related Work

Neural networks for parametric problems. To the best of our knowledge, there are seven ways to solve a parametric problem with neural network based methods. We list them below roughly in the order of popularity, and discuss their advantages and disadvantages.

The first method treats problems with different parameter levels as independent

tasks, and trains a specific model for each parameter level [37, 142, 190, 181, 189]. The overwhelming majority of previous papers use this approach. This method is easy to understand, and generally has good performance since the parameter level is implied in a model. But it is inefficient in computing, because we may have to train and store dozens of models for different parameter levels.

The second method simply ignores the parameter, and trains a single model for all parameter levels [79, 175]. This method is also easy to understand, and is very efficient since we only need to train and store a single model. But its performance is typically lower than the first method, since we ignore some important information that could have helped the restoration process.

The third method trains a model with a shared backbone for all parameter levels and multiple specific blocks for each parameter level [103]. It is a compromise between the first and the second method, and has acceptable performance and efficiency. But we may still have to store dozens of specific blocks for different parameter levels. And if the capacity of a specific block is not large enough, the block cannot take full advantage of the parameter information.

The fourth method converts the parameter scalar into a parameter map, and treats the map as an additional channel of the input degraded image [183]. It is another way to blend the parameter information into a neural network model. However, the performance of this method is only marginally higher than the second method, and it is still not as good as the first method. Due to the huge semantic difference between the corrupt image and the parameter map, it is hard to make much use of the parameter information for the model.

The fifth method conditions a network by modulating all its intermediate features

by scalar parameters [30, 64] or maps [160]. It is another way to make a network adapt to different situations. But unlike our FuncNet, the method changes only features rather than parameters.

The sixth method trains a model with a relatively shallow backbone network, and each filter of the backbone network is generated by a different filter generating network [84, 74, 77, 41]. The filter generating networks are usually multilayer perceptrons, and they take the parameter as input. Since the total size of a model is limited, assigning each filter a unique complex network severely limits the size of the backbone network. Such a shallow backbone network only leads to a mediocre performance. Considering the universal approximation ability of the multilayer perceptron, this method is not that different from training a unique shallow model for each parameter level.

The seventh method searches in the latent space of a generative model, and returns the most probable result which is not contradicting the degradation model with the parameter [171, 109]. It is a general image restoration method which can solve various image restoration problems with a single model, as long as the degradation model is continuously differentiable. However, this method is slower than a feedforward neural network based method, since it requires multiple forward and backward passes in a search. And due to the limited representation ability of the generative model, the performance of this method is also worse than a discriminative model based method.

Neural network interpolation. In order to attain a continuous transition between different imagery effects, the neural network interpolation method [161, 162, 159] applies linear interpolation in the parameter space of two trained networks. Although at first glance it is similar to our method, there is a big difference between

network interpolation and our FuncNet. The former is a simple interpolation technique while the latter is a regression technique. In the network interpolation method, two CNNs are trained separately for two extreme cases, and then blended in an ad hoc way. This may suffice for tasks [161] and [162], because users will accept roughly characterized visual results, such as "half GAN half MSE" or "half photo half painting". However, this is not good enough for an image restoration task whose goal is to restore the signal as accurately as possible. FuncNet is optimized for the entire value range of the task parameter (e.g., the noise level, SR scale factor), so its accuracy stays high over the entire parameter range, rather than just for the two extreme points like in the network interpolation method.

Deep CNN for Single Image Super-Resolution. The first convolutional neural network for single image super-resolution is proposed by Dong et al. [37] called SRCNN, and it achieved superior performance against previous works. Shi et al. [142] firstly proposed a real-time super-resolution algorithm ESPCN by proposing the sub-pixel convolution layer. Lim et al. [103] removed batch normalization layers in the residual blocks, and greatly improved the SR effect. Zhang et al. [190] introduced the residual channel attention to the SR framework. Hu et al. [67] proposed the Meta-Upscale Module to replace the traditional upscale module.

Deep CNN for Image Denoising. Zhang et al. [181] proposed DnCNN, a plain denoising CNN method which achieves state-of-the-art denoising performance. They showed that residual learning and batch normalization are particularly useful for the success of denoising. Tai et al. [150] proposed MemNet, a very deep persistent memory network by introducing a memory block to mine persistent memory through an adaptive learning process. Zhang et al. [183] proposed FFDNet, a fast and flexible

denoising convolutional neural network, with a tunable noise level map as the input.

Deep CNN for JPEG deblocking. Dong et al. [38] proposed ARCNN, a compact and efficient network for seamless attenuation of different compression artifacts. Guo and Chao [61] proposed a highly accurate approach to remove artifacts of JPEG-compressed images, which jointly learned a very deep convolutional network in both DCT and pixel domains. Zhang et al. [189] proposed DMCNN, a Dual-domain Multi-scale CNN to take full advantage of redundancies on both the pixel and DCT domains. Ehrlich et al. [41] proposed a novel architecture which is parameterized by the JPEG files quantization matrix.

3.3 Functional Neural Network (FuncNet)

In this section, we describe the proposed FuncNet model. To transform a plain neural network into a FuncNet, we replace every trainable variable in a plain neural network by a specific function, such as weights and biases in convolution layers or fully connected layers, affine parameters in Batch Normalization layers [71], and slopes in PReLU activation layers [65]; and keep other layers without trainable variables unchanged, such as pooling layers, identity layers, and pixel shuffle layers [142]. We first describe the method to specify the functions in our FuncNet models, then we describe the initialization, training and inference method for these functions, next we describe the network architectures to contain those functions, and finally we analyse the storage and computational efficiency of our FuncNet models.

3.3.1 Specification of Functions

The functions used in our FuncNet model should be simple enough. Let us consider the following failure case as a negative example. Suppose the number of the parameter levels is large but still finite, and we use polynomial functions in our FuncNet. If the polynomial function is too complex, and its degree is greater than or equal to the number of the parameter levels minus one, then our FuncNet is no different from training a specific model for each parameter level. In this case, the failing FuncNet takes as much or even more storage space than multiple independent models, and its inference speed is also slightly slower than a same size plain neural network. This negative example demonstrates the necessity of choosing simple functions for our FuncNet.

We choose the simplest and the most basic kind of function, the linear function, as the functions used in our FuncNet model. In this case, the FuncNet model takes exactly double storage space than a same size plain neural network, and it is still more efficient than storing dozens of models for different parameter levels. And it only increases a negligible amount of computations than a same size plain neural network, as we will demonstrate later. Choosing such a simple function does not lead to a poor performance of the final FuncNet model. With multiple activation layers, the final FuncNet model retains the power of nonlinear fitting. The linear function used in our FuncNet model can be defined as:

$$G(x;\theta_a,\theta_b) = \frac{x - x_a}{x_b - x_a}(\theta_b - \theta_a) + \theta_a \tag{3.3.1}$$

where x is the parameter of the problem, x_a and x_b are lower and upper bound of the

support of the parameter distribution respectively, θ_a and θ_b are trainable variables, and $G(x; \theta_a, \theta_b)$ is the function used in our FuncNet model to generate variables for different parameter levels.

The parameters have different properties for different problems, and we can make the linear function 3.3.1 to suit different problems better by replacing x with $H(x)$, where $H(x)$ is a problem-related function. Then the linear function 3.3.1 will become:

$$G(H(x); \theta_a, \theta_b) = \frac{H(x) - H(x_a)}{H(x_b) - H(x_a)}(\theta_b - \theta_a) + \theta_a \tag{3.3.2}$$

The chosen $H(x)$ should have a physical interpretation related to the problem, and of course should make the final FuncNet model perform well. In the super-resolution problem, we use $H(x) = 1/x$ because the reciprocal of the scale factor is the rescaled length on a low-resolution image from a unit length on a high-resolution image. In the image denoising problem, we use $H(x) = x$ because the noise level is equal to the standard deviation of the additive white Gaussian noise. In the JPEG deblocking problem, we use $H(x) = 5000/x$ for $x \leq 50$ and $H(x) = 200 - 2x$ for $x > 50$. This is the formula used in JPEG standard [128], and it transforms the quality factor into a scale factor of the quantization table for DCT coefficients. The choices of $H(x)$ for different problems are still empirical, just like when people determine the depth, the width or other configuration for a neural network. But we hope that we can determine $H(x)$ automatically in the future, just like what people do in Neural Architecture Search right now [194].

3.3.2 Initialization, Training and Inference

Proper initialization is crucial for training a neural network. Even with modern structures and normalization layers, a bad initialization can still hamper the learning of the highly nonlinear system. The goal of initialization for a plain neural network is to set the value of every trainable variable in a proper range, and to avoid reducing or magnifying the magnitude of input signals exponentially. To properly initialize a function in FuncNet, we have to guarantee that all possible output values of the function are in a proper range. Suppose $H(x)$ in function 3.3.2 is a monotonic function, then what we need to do is to use initialization algorithm for a plain neural network [65] to initialize θ_a and θ_b independently. In this way, all possible output values of the function 3.3.2 lie somewhere between θ_a and θ_b, and must also be in a proper range if θ_a and θ_b are well set. If θ_a and θ_b are both sampled from a zero-mean distribution whose standard deviation is σ, then for all possible output values of the function 3.3.2, their expected values are still zero, and their standard deviations are between $\sigma/\sqrt{2}$ and σ. Experiments have shown that such a small deviation is acceptable for training.

Training a FuncNet is not very different from training a plain neural network. In every iteration, we first sample a fixed number of parameter levels uniformly, use them to construct a minibatch, and then perform stochastic gradient based optimization as normal. As suggested in [193], we train our FuncNet using L1 loss. During the training, we gradually change the values of trainable variables θ_a and θ_b to reduce the value of the loss function. The problem can be formulated as

$$\min_{\theta_a,\theta_b} \mathbb{E}_{I^O,I^D,x} \|F(I^D; G(H(x); \theta_a, \theta_b)) - I^O\|_1 \tag{3.3.3}$$

where I^D is the degraded version of its original counterpart I^O, and F is our FuncNet model.

When we use the trained FuncNet to restore a degraded image with a given parameter x, we first evaluate function 3.3.2 with x, trained θ_a and θ_b, use the evaluation values as variables of the corresponding plain neural network, and then do inference with the generated plain neural network as normal.

3.3.3 Network Architectures

The requirements of the networks for the three image restoration problems are very different. For the super-resolution problem, the degradation is deterministic and relatively mild, and the network can concentrate on a relatively small area. For the image denoising problem, the degradation is random and relatively severe, and the network needs to pay attention to a larger area. For the JPEG deblocking problem, the degradation occurs in the DCT domain, and the network should have the ability to utilize the information in the DCT domain. So we use individually designed architectures for the three image restoration problems to meet their own requirements.

We directly use architectures of the state-of-the-art plain neural networks for the three tasks as the architectures of our FuncNet models, and we only make essential modifications to them. For the super-resolution problem, we use the architecture of RCAN [190], and replace the upscale module for integer scale factors [142] with the meta upscale module for non-integer scale factors [67]. For the image denoising problem, we apply a modified U-net [134] structure as the backbone and use RCAB [190] as residual blocks. For the JPEG deblocking problem, we use the architecture of DMCNN [189] for reference. For the DCT domain branch of our JPEG deblocking model, we use

frequency component rearrangement to get a more meaningful DCT representation as suggested in [41]; and for the pixel domain branch, we use the same architecture of our image denoising model.

For the backbone of our super-resolution network, we use the architecture of RCAN [190] directly. It has 10 residual groups with 20 residual channel attention blocks (RCAB) each. For our image denoising network and the pixel domain branch of our JPEG deblocking network, we apply a modified U-net [134] structure with identity shortcuts as the backbone. The network has a similar size as RCAN [190]. It has 4 scale levels. Each scale level has 4 residual groups, and each residual group has 10 RCABs. We set 3×3 as the size of all convolution layers in RCABs, and set 1×1 as the size of all channel attention layers in RCABs. The number of features of RCAB is set as 64, and the reduction ratio of RCAB is set as 16.

3.3.4 Storage and Computational Efficiency Analysis

Our FuncNet models have high storage efficiency. As we described in Section 3.3.1, we use two-degree-of-freedom functions in FuncNet models. This takes only twice as much space as what a plain neural network with the same architecture will take. So storing a FuncNet model is much cheaper than storing dozens of plain networks for different parameter levels. Take the super-resolution task as an example. Suppose the scale factor varies from 1.1 to 4 with stride 0.1 as suggested in [67], we can save 93.3% on storage space by using FuncNet rather than plain neural networks.

Our FuncNet models have high computational efficiency as well. For the training phase, we only need to train one FuncNet model rather than to train dozens of plain networks individually. The computational efficiency analysis for this phase is similar

to the preceding storage efficiency analysis. For the inference phase, as we described in Section 3.3.2, we first evaluate functions with the problem related parameter, use the evaluation values as variables of the corresponding plain neural network, and then do inference with the generated plain neural network as normal. So compared to a plain neural network with the same architecture, our FuncNet model only needs a little extra effort to evaluate functions. This part of computation is directly proportional to the number of parameters in the corresponding plain network, and it is several orders of magnitude smaller than the number of multi-adds for a plain image restoration network. Still take the super-resolution task as an example. Suppose we need to double the size of a 360p image, our FuncNet model only needs extra 0.0001% computation than a plain neural network with the same architecture.

3.4 Experiments

3.4.1 Training Settings

Training datasets. Following [190, 67, 175], we use the DIV2K dataset [153] for training. There are 1000 high-quality images in the DIV2K dataset, 800 images for training, 100 images for validation and 100 images for testing. All our three FuncNet models for the three parametric image restoration tasks are trained with the DIV2K training images set.

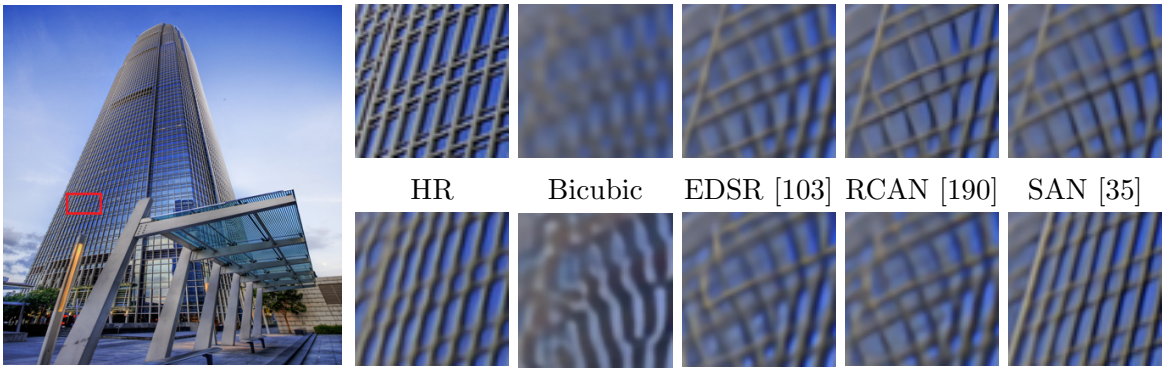
Parametric settings. In all three parametric problems, the problem related parameters are sampled uniformly. In the super-resolution problem, the training scale factors vary from 1.1 to 4 with stride 0.1. In the image denoising problem, the training noise levels are sampled from the uniform distribution on the interval (0, 75]. In the

JPEG deblocking problem, the quality factors vary from 10 to 80 with stride 2.

Degradation models. In the super-resolution problem, we use the bicubic interpolation by adopting the Matlab function imresize to simulate the LR images. In the image denoising problem, we generate the additive white Gaussian noise dynamically by using the Numpy function. In the JPEG deblocking problem, we use the Matlab JPEG encoder to generate the JPEG images.

Data augmentations. In all three parametric problems, we use the same data augmentation method. We randomly augment the image patches by flipping horizontally, flipping vertically and rotating 90°.

Optimization settings. In the super-resolution problem, we randomly extract 32 LR RGB patches with the size of 40×40 as a batch input. In the image denoising problem, we randomly extract 32 RGB patches with the size of 96×96 as a batch input. In the JPEG deblocking problem, we randomly extract 32 gray patches with the size of 96×96 as a batch input, and we make sure that the image patches are aligned with boundaries of Minimum Coded Unit blocks. All our three FuncNet models are trained by ADAM optimizor with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The initial learning rate is set to 10^{-4} and then decreases by half for every 2×10^5 iterations of back-propagation. All experiments run in parallel on 4 GPUs.



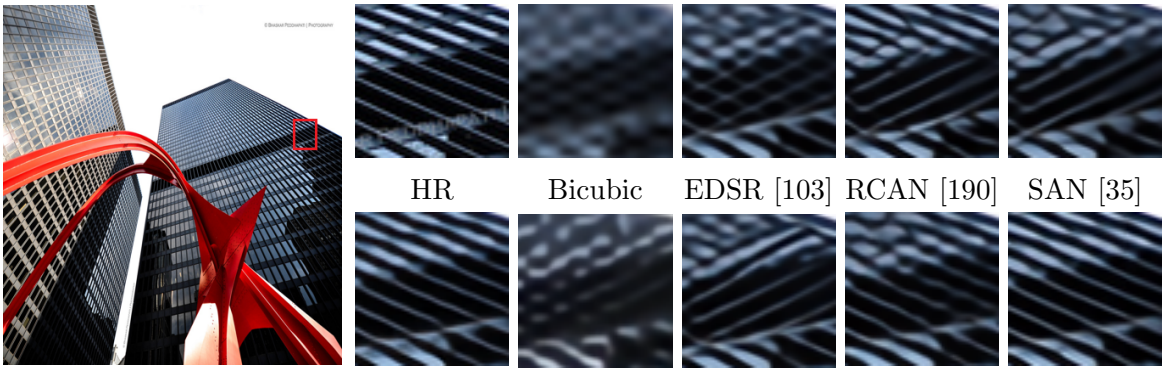


Figure 3.2: Visual comparison between different super-resolution methods

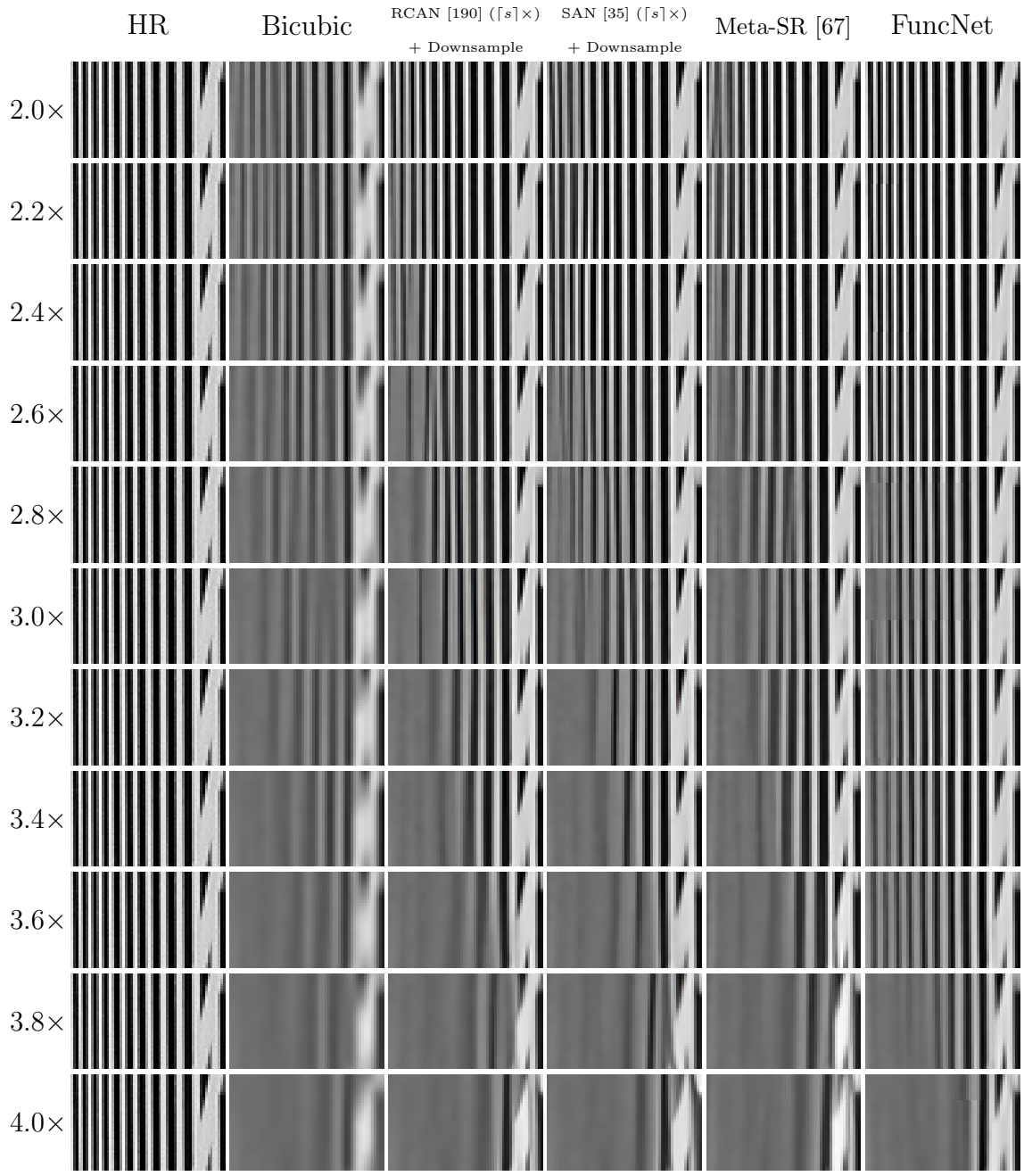
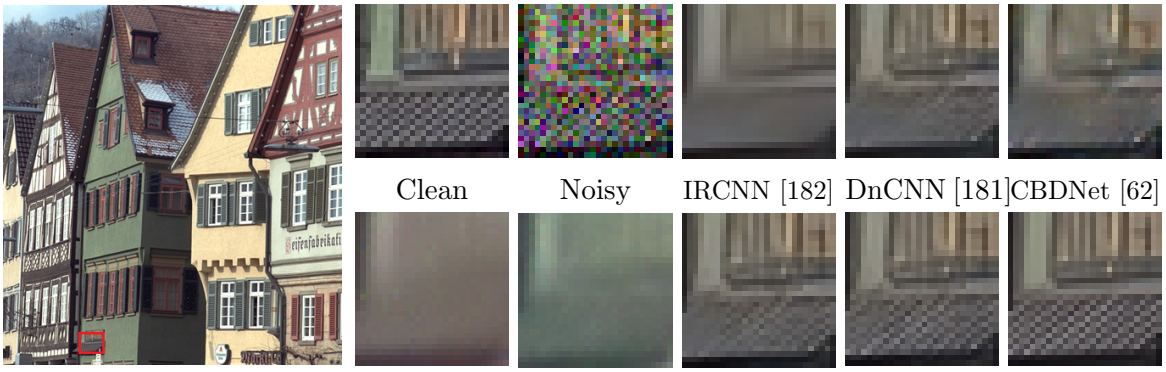


Figure 3.3: Visual comparison between different decimal upscale super-resolution methods



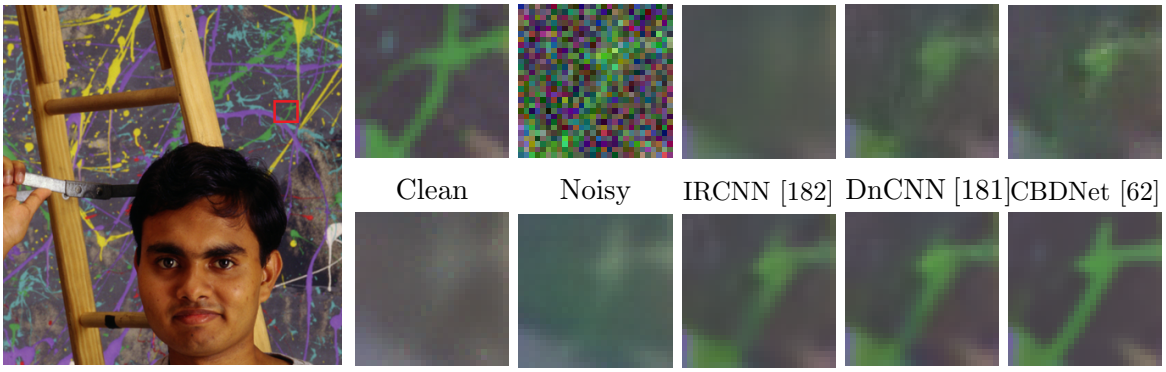
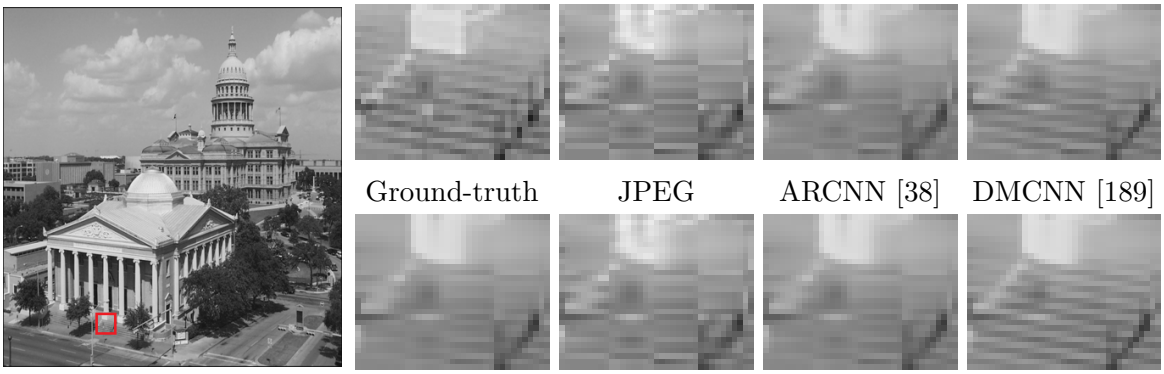


Figure 3.4: Visual comparison between different image denoising methods



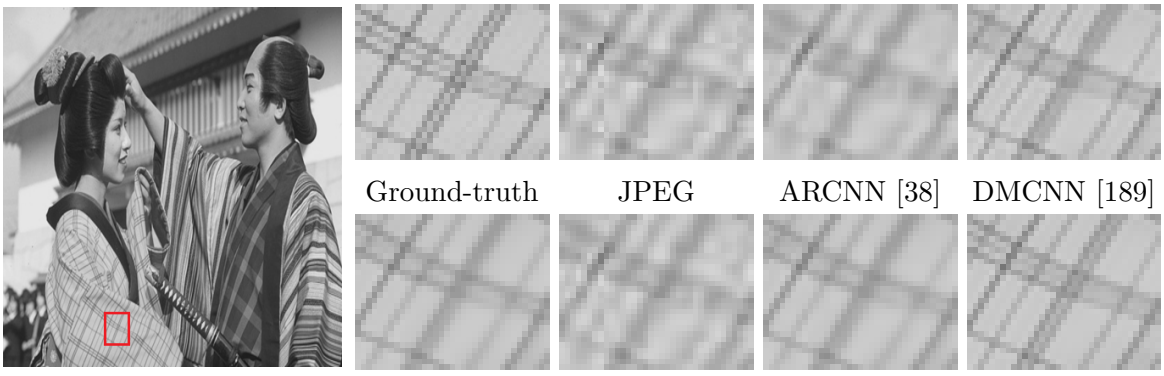


Figure 3.5: Visual comparison between different JPEG deblocking methods

Table 3.1: Results of decimal upscale SR on B100. Best and second best results are **highlighted** and underlined

Method \ Scale	×1.1	×1.2	×1.3	×1.4	×1.5	×1.6	×1.7	×1.8	×1.9
Meta-SR [67]	42.82	40.04	38.28	36.95	35.86	34.90	34.13	33.45	32.86
FuncNet	**43.43**	40.41	38.55	37.16	36.02	35.08	34.26	33.60	32.98
FuncNet+	43.36	**40.46**	**38.59**	**37.21**	**36.06**	**35.12**	**34.30**	**33.64**	**33.02**
Method \ Scale	×2.1	×2.2	×2.3	×2.4	×2.5	×2.6	×2.7	×2.8	×2.9
Meta-SR [67]	31.82	31.41	31.06	30.62	30.45	30.13	29.82	29.67	29.40
FuncNet	31.99	31.59	31.23	30.87	30.58	30.30	30.05	29.77	29.59
FuncNet+	**32.02**	**31.62**	**31.26**	**30.90**	**30.62**	**30.34**	**30.09**	**29.81**	**29.63**
Method \ Scale	×3.1	×3.2	×3.3	×3.4	×3.5	×3.6	×3.7	×3.8	×3.9
Meta-SR [67]	28.87	28.79	28.68	28.54	28.32	28.27	28.04	27.92	27.82
FuncNet	29.17	29.02	28.81	28.62	28.46	28.34	28.21	28.06	27.93
FuncNet+	**29.20**	**29.06**	**28.85**	**28.67**	**28.51**	**28.38**	**28.26**	**28.11**	**27.97**

Table 3.2: Results of integer upscale SR. Best and second best results are **highlighted** and underlined. B100, Urban and Manga represent datasets B100, Urban100, and Manga109 respectively.

Method	Scale = 2			Scale = 3			Scale = 4		
	B100	Urban	Manga	B100	Urban	Manga	B100	Urban	Manga
EDSR [103]	32.32	32.93	39.10	29.25	28.80	34.17	27.71	26.64	31.02
RCAN [190]	32.41	33.34	39.44	29.32	29.09	34.44	27.77	26.82	31.22
SAN [35]	32.42	33.10	39.32	29.33	28.93	34.30	27.78	26.79	31.18
CSNLN [119]	32.40	33.25	39.37	29.33	29.13	34.45	27.80	27.22	31.43
MIRNet [177]	-	-	-	27.04	24.53	26.99	25.96	23.24	25.50
Meta-SR [67]	32.35	-	39.18	29.30	-	34.14	27.75	-	31.03
LIIF [29]	32.32	32.87	-	29.26	28.82	-	27.74	26.68	-
FuncNet	32.48	33.61	39.73	29.39	29.42	34.90	27.87	27.15	31.71
FuncNet+	**32.51**	**33.78**	**39.87**	**29.43**	**29.57**	**35.10**	**27.90**	**27.29**	**31.97**

Table 3.3: Results of image denoising. Best and second best results are **highlighted** and underlined. CBSD, Kodak and Mac represent datasets CBSD68, Kodak24 and McMaster respectively.

Method	$\sigma = 15$			$\sigma = 35$			$\sigma = 75$		
	CBSD	Kodak	Mac	CBSD	Kodak	Mac	CBSD	Kodak	Mac
CBM3D [34]	33.52	34.28	34.06	28.89	29.90	29.92	25.74	26.82	26.79
DnCNN [181]	33.89	34.48	33.44	29.58	30.46	30.14	24.47	25.04	25.10
CBDNet [62]	32.67	33.32	32.87	28.11	28.87	28.77	24.05	24.64	24.38
MIRNet [177]	27.44	28.30	27.92	22.39	23.19	22.47	18.77	18.88	18.76
FFDNet [183]	33.87	34.63	34.66	29.58	30.57	30.81	26.24	27.27	27.33
CResMD [64]	33.97	34.80	34.80	29.70	30.75	31.00	26.26	27.36	27.39
FuncNet	34.26	35.21	35.39	30.02	31.24	31.61	26.72	27.98	28.18
FuncNet+	**34.28**	**35.25**	**35.44**	**30.05**	**31.29**	**31.67**	**26.76**	**28.05**	**28.26**

Table 3.4: Results of JPEG deblocking. Best and second best results are **highlighted** and underlined. LIVE and BSDS represent datasets LIVE1 and BSDS500 respectively.

Method	Quality = 10		Quality = 20		Quality = 30		Quality = 40	
	LIVE	BSDS	LIVE	BSDS	LIVE	BSDS	LIVE	BSDS
ARCNN [38]	29.13	29.10	31.40	31.28	32.69	32.64	33.63	33.55
DMCNN [189]	29.73	29.67	32.09	31.98	-	-	-	-
CResMD [64]	27.89	27.92	30.58	30.55	32.46	32.37	33.87	33.73
QGAC [41]	29.53	29.54	31.86	31.79	33.23	33.12	-	-
FuncNet	29.77	29.68	32.20	32.05	33.63	33.44	34.63	34.41
FuncNet+	**29.81**	**29.71**	**32.23**	**32.07**	**33.66**	**33.47**	**34.66**	**34.44**

3.4.2 Evaluation on Standard Benchmark Datasets

In the super-resolution problem, we use the B100 dataset for non-integer scale factor testing, and we use five standard benchmark datasets for integer scale factor testing: Set5, Set14, B100, Urban100, and Manga109. The results are evaluated with PSNR and SSIM [164] on Y channel of transformed YCbCr space. In the image denoising problem, we use three standard benchmark datasets: CBSD68, Kodak24, and McMaster. The results are evaluated with PSNR and SSIM [164] on RGB channel as suggested in [181]. In the JPEG deblocking problem, we use two standard benchmark datasets: LIVE1 and BSDS500. The results are evaluated with PSNR, SSIM [164], and PSNR-B [174] on Y channel.

We compare our results with those of state-of-the-art methods for all three parametric problems. Similar to [103], we also apply a self-ensemble strategy to further improve our FuncNet model and denote the self-ensembled one as FuncNet+. The quantitative results are shown in Table 4.1, 4.2, 3.3, and 3.4. The visual comparisons are shown in Figure 4.2, 3.3, 4.4, and 3.5.

3.4.3 Kernel Visualization and Interpretation

We visualize kernels of our FuncNet model and try to understand and interpret them. The key point of the analysis is to find out how kernels change with the problem related parameter. Here we show samples of kernels from the first and the last layer of the denoising FuncNet, since the denoising problem has the most definite physical meaning among the three image restoration problems. The first and the last layer are also easier to understand. The results are shown in Figure 3.6. We can find out that the FuncNet uses more radical kernels for features when the noise level is low.

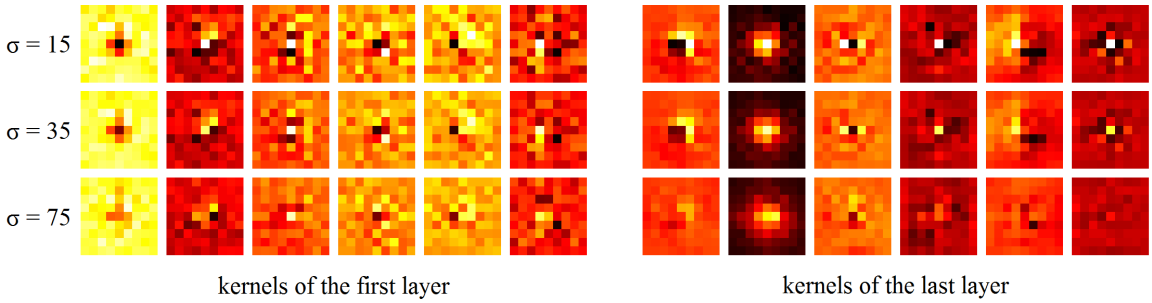


Figure 3.6: Kernel visualization of the denoising FuncNet. The left part is sampled from the first layer and the right part is sampled from the last layer. We can find out that the FuncNet uses more radical kernels when the noise level is low, and uses more moderate kernels when the noise level is high.

By doing so, the FuncNet can get more information. And the FuncNet uses more moderate kernels when the noise level is high, so the FuncNet can get less error.

3.4.4 Ablation Study

As we discussed earlier, using functional kernels instead of numerical kernels is the key to making networks perform better for parametric image restoration problems. To verify the effectiveness of our FuncNet models, we train plain counterparts of our FuncNet models, and compare their evaluation results with FuncNets. And to measure the impact of choice on the problem-related function $H(x)$, we train another two versions of FuncNet. The first one always uses the simplest non-trivial mapping $H(x) = x$, and the second one uses a small multilayer perceptron (MLP) with a hidden layer as a universal function approximator for any possible $H(x)$. We then also compare their evaluation results with FuncNet which uses $H(x)$ with a physical interpretation related to the problem. All the networks for ablation study share the same architectures with their corresponding FuncNet models, and all training and

Table 3.5: Results of the ablation study. Super-resolution, denoising and deblocking are tested on Urban100, Kodak24 and LIVE1 respectively.

Method	Super-resolution			Denoising		Deblocking	
	$s = 2$	$s = 3$	$s = 4$	$\sigma = 15$	$\sigma = 35$	$q = 10$	$q = 20$
FuncNet	33.61	29.42	27.15	35.21	31.24	29.77	32.20
Plain net	33.07	28.93	26.70	34.83	30.89	29.58	31.94
FuncNet ($H(x) = x$)	33.48	29.33	27.05	35.21	31.24	29.64	32.16
FuncNet (H is a MLP)	33.60	29.38	27.02	35.19	31.20	29.69	32.17

evaluation settings remain unchanged. The evaluation results are shown in Table 3.5.

This ablation study shows that the adaptability of our FuncNet model is important for parametric image restoration problems. Once our FuncNet degenerates into a plain network, its adaptability to different parameter levels disappears, and its performance drops remarkably. The results also prove that both identity function and MLP are acceptable choices for $H(x)$. We can simply use those functions for a problem which is hard to design a $H(x)$ with a physical interpretation.

3.5 Conclusions

We propose a novel neural network called FuncNet to solve parametric image restoration problems with a single model. To transform a plain neural network into a FuncNet, all trainable variables in the plain network are replaced by functions of the parameter of the problem. Our FuncNet has both high storage efficiency and high computational efficiency, and the experimental results show the superiority of our FuncNet on three common parametric image restoration tasks over the state of the arts.

Chapter 4

AND: Adversarial Neural Degradation for Learning Blind Image Super-Resolution

Learnt deep neural networks for image super-resolution fail easily if the assumed degradation model in training mismatches that of the real degradation source at the inference stage. Instead of attempting to exhaust all degradation variants in simulation, which is unwieldy and impractical, we propose a novel adversarial neural degradation (AND) model that can, when trained in conjunction with a deep restoration neural network under a minmax criterion, generate a wide range of highly nonlinear complex degradation effects without any explicit supervision. The AND model has a unique advantage over the current state of the art in that it can generalize much better to unseen degradation variants and hence deliver significantly improved restoration performance on real-world images.

4.1 Introduction

Deep learning has made great strides in the applications of image restoration. It has demonstrated superior performances over traditional methods on almost all common image restoration tasks, including super-resolution [39], denoising [181], compression artifacts removal [38], deblurring [147], etc. But the margin of performance gains made by deep learning methods of image restoration decreases sharply if the degradation processes assumed in training mismatch those of the real world images at inference stage [21]. It is well known that, for any real-world problems, the efficacy of a machine learning technique relies not only on the design of the technique itself but also, sometimes even more critically, on the statistical agreement between the training and test data [158].

In reality, it is either intractable or highly expensive to obtain both degraded images and the corresponding latent images (ground truth). The most common practice in literature is to use a degradation model to generate paired degraded and ground truth images for training the restoration networks [185, 163, 100, 99]. This synthesis approach cannot accurately simulate the realistic digital imaging pipeline that is affected by multiple complex and compounded degradation sources; for instances, insufficient sampling rate, color demosaicing errors, sensor noises, camera jitters, compression distortions and etc. In this paper, we focus on the task of super-resolution, namely assuming that the dominant degradation cause is insufficient sampling rate, which is compounded by other degradation sources in the imaging pipeline. The said complex nonlinear phenomena often defy explicit analytical modeling. A brute force approach may be to build multiple simpler parametric degradation models, one for each type of degradation (e.g., downsampling, noises, compression, motion, etc.)

and apply them in different combinations, orders and parameter setting to generate training data, in hope to simulate as wide a range of degradations encountered in practice as possible. This amounts, however, to fighting a losing battle because it is impossible to exhaust all degradation variants, many of which are not even known or understood.

This work represents a fundamental departure from the current ways of coping with mismatches between the simulated training and real-world image data. Instead of attempting to exhaust all degradation types in simulation, we propose a novel adversarial neural degradation (AND) model that can, when trained in conjunction with a deep restoration neural network under a minmax criterion, generate a wide range of highly nonlinear complex degradation effects without any explicit supervision. Adversarial attack and defense (training) [57, 115] is a proven learning strategy to vaccinate neural network models of signal classification against being misled by imperceptible disturbances in input signals. But regrettably, adversarial learning has not been applied to neural network models of signal restoration. The AND model, the main contribution of this paper, demonstrates for the first time how adversarial learning can effectively boost the robustness of deep networks for signal restoration. In particular, we adopt the minmax optimization criterion when training the AND model, aiming to withstand the attacks by the most difficult but nuance degradations that otherwise defy modeling. As a result, the AND model enjoys a unique advantage over the current state of the art in being generic in terms of degradation types. It can generalize much better to unseen degradation types and variants and hence deliver significantly improved restoration performance on real-world images.

Our insight of the AND model comes from the following observations. We observe

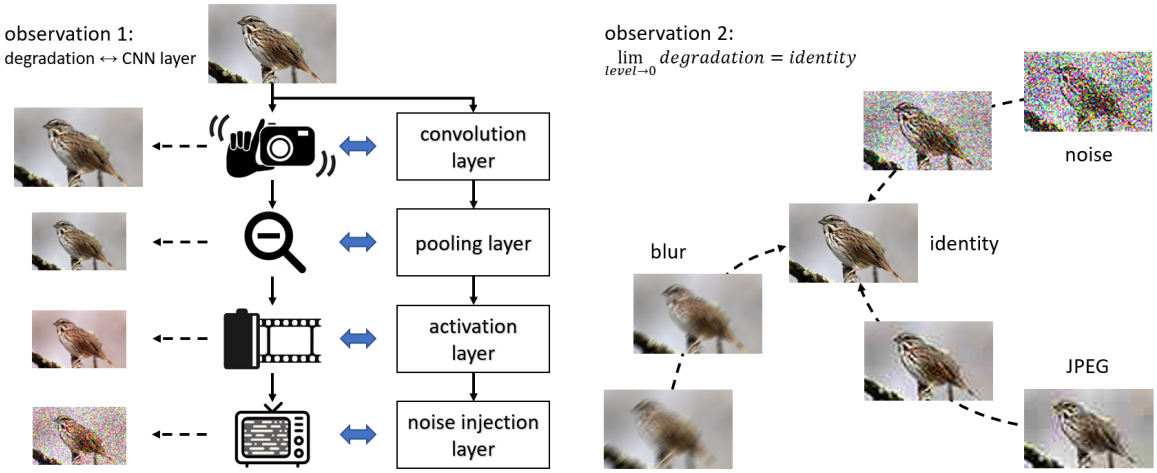


Figure 4.1: We observe two properties in most image degradations. Firstly, almost all types of image degradation could find a corresponding operation in a standard convolutional neural network. Secondly, almost all moderate image degradations could be considered as small deviations from the identity transformation. The neural degradation prior proposed for our real-world super-resolution method is inspired by these observations.

two properties in most image degradations, as shown in Fig. 4.1. Firstly, almost all types of image degradation could find a corresponding operation in a standard convolutional neural network. For example, blur and ringing could correspond to a convolution layer, downsampling could correspond to a pooling layer, color fading and posterization [136] could correspond to a non-linear activation layer, sensor noise and film grain could correspond to a noise injection layer [78]. Secondly, most moderate image degradations could be considered as small deviations from the identity transformation. For example, blur, noise and lossy image compression all obviously tend to the identity transformation pointwise as the degradation level approaches the slightest level. As the degradation level get higher, those degradations gradually deviate from the identity transformation. Inspired by the two observations of image degradations, we initialize untrained convolutional neural networks to the identity transformation, make parameters of these networks slightly deviated from the start, and take them as prior for various real-world image degradations. When we train a SR model with HR and LR image pairs constructed by the proposed degradation prior, we adversarially search small deviations which could make the SR model perform the worst, and optimize the SR model based on the worst degradation case to achieve a good lower performance bound for various real-world image degradations.

4.2 Related Work

Single Image Super-Resolution. The first convolutional neural network for single image super-resolution is proposed by Dong et al. [37] called SRCNN, and it achieved superior performance against previous works. Since that the field has witnessed a variety of developments. Shi et al. [142] firstly proposed a real-time super-resolution

algorithm ESPCN by proposing the sub-pixel convolution layer. Lim et al. [103] removed batch normalization layers in the residual blocks, and greatly improved the SR effect. Zhang et al. [190] introduced the residual channel attention to the SR framework. To achieve photo-realistic results with detailed textures, Ledig et al. [94] introduced the generative adversarial network [56] into the SR framework, and employed it as loss supervisions to push the SR solutions closer to the natural manifold. Wang et al. [161] later improved the GAN based SR method, and achieved better SR visual quality with more realistic and natural textures.

Blind Image Super-Resolution. The field is also named as real-world image super-resolution. Different from the classical SR field which assumes that the image degradation model is an ideal bicubic downsampling, the blind SR field aims to solve SR problems with unknown degradation. Researchers tried to solve the problem by implicitly or explicitly estimating the degradation model. Gu et al. [58] proposed a method to iteratively estimate the blur kernel. Kligler et al. [9] introduced KernelGAN, which trains solely on the LR test image at test time, and learns its internal distribution of patches. Researchers also built complex models for image degradation, to augment the robustness of the SR model. Zhang et al. [185] designed a complex degradation model that consists of randomly shuffled blur, downsampling and noise degradations. Wang et al. [163] used a high-order degradation model to better simulate complex real-world degradations.

Degradation Learning. In order to enhance the performance of super-resolution models on real-world images, researchers have explored a two-stage learning approach [18, 98]. They employ a trained network to emulate degradation effects using a provided LR image dataset. Subsequently, the acquired knowledge of degradation

is utilized to train the super-resolution model. It is important to note that this methodology differs significantly from our own, and this dissimilarity in approach can result in the former's limited generalization capabilities compared to the latter. Their approach is vulnerable to failure if the degradation inherent in the chosen LR training images does not align with the actual degradation encountered during the inference stage. In contrast, our method demonstrates superior generalization to unforeseen degradation variations by leveraging an untrained neural degradation prior.

Adversarial Training. Adversarial training improves the model robustness by training on adversarial examples generated by gradient-based method [57]. Madry et al. [115] studied the adversarial robustness of neural networks through the lens of robust optimization. Tramer et al. [156] proposed an ensemble adversarial training on adversarial examples generated from a number of pretrained models. Kolter and Wong [168] developed a provable robust model that minimizes worst-case loss over a convex outer region. Athalye et al. [7] demonstrated that adversarial training on PGD adversarial examples was to be the state-of-of-art defense model.

Researchers have also attempted to utilize adversarial examples during training to enhance the capacity of SR models in processing noisy inputs [24, 176]. They borrowed the earlier research in adversarial training for image classification tasks, but did not account for the differences between the classification and restoration tasks. In the previous research on adversarial training for image classification tasks, Out-of-Distribution (OOD) perturbations are introduced through the deliberate efforts of malicious attackers. This approach utilizes pixelwise additive high-frequency noise as a concealed and effective perturbation attack. Note that the adversarial attack is very different from the type of signal degradations in restoration tasks.

Specifically, the perturbations in super-resolution tasks encompass a mixture of blur, noise, and nonlinear transformations. As a result, for restoration tasks noise no longer predominantly influences the OOD perturbation as in classification tasks. Therefore, the proposed neural degradation prior is a more suitable perturbation model for real-world image restoration.

Domain Generalization. Domain generalization aims to achieve out-of-distribution generalization by using only source data for model learning. Most existing approaches belong to the category of domain alignment [122], where the central idea is to minimize the difference among source domains for learning domain-invariant representations. Meta-learning [44] are also used to solve domain generalization by exposing a model to domain shift during training with a hope that the model can better deal with domain shift in unseen domains. In the context of domain generalization, the work most related to ours is RandConv [170]. It is based on the idea of using randomly initialized, single-layer convolutional neural network to transform the input images to novel domains. Since the weights are randomly sampled from a Gaussian distribution at each iteration and no learning is performed, the transformed images mainly contain random color distortions, which do not contain meaningful variations and are best to be mixed with the original images before passing to the task network.

Identity Mapping in Deep Learning. Identity mappings are widely used in deep learning methods, typically as network layers rather than degradation priors. Sun et al. [66] used identity mappings as the skip connections and after-addition activation, to make the training easier and improves model generalization. Zhang et al. [179] used an identity mapping task to study memorization and generalization of overparameterized networks in the extreme cases.

4.3 Adversarial Neural Degradation for Blind Super-Resolution

Before discussing our new robust real-world SR method, we would like to emphasize once again that the degradation prior employed in our approach is inspired by the following two observations of image degradations.

1. Almost all types of image degradation could find a corresponding operation in a standard convolutional neural network.

2. Almost all moderate image degradations could be considered as small deviations from the identity transformation.

Once we identify the commonalities among various image degradations, we can naturally propose a simple and elegant degradation prior that encompasses all of these degradations. That is, we take slightly deviated identity convolutional neural networks as prior for various real-world image degradations.

We illustrate the entire training procedure of our SR method with the proposed degradation prior in Fig. 4.2. The entire neural network can be divided into three parts: a degradation network, a restoration network, and an optional discriminator network. A single optimization step of the entire network can be further divided into the following four sub-steps. First, we initialize the degradation network to the identity transformation. Next, we adversarially search for a small deviation of the initialized degradation network that would cause the restoration network to perform the worst. Then, we optimize the restoration network based on the identified degradation case. Finally, we upgrade the discriminator network to distinguish restoration outputs from real images. We repeat the optimization steps of the entire network multiple

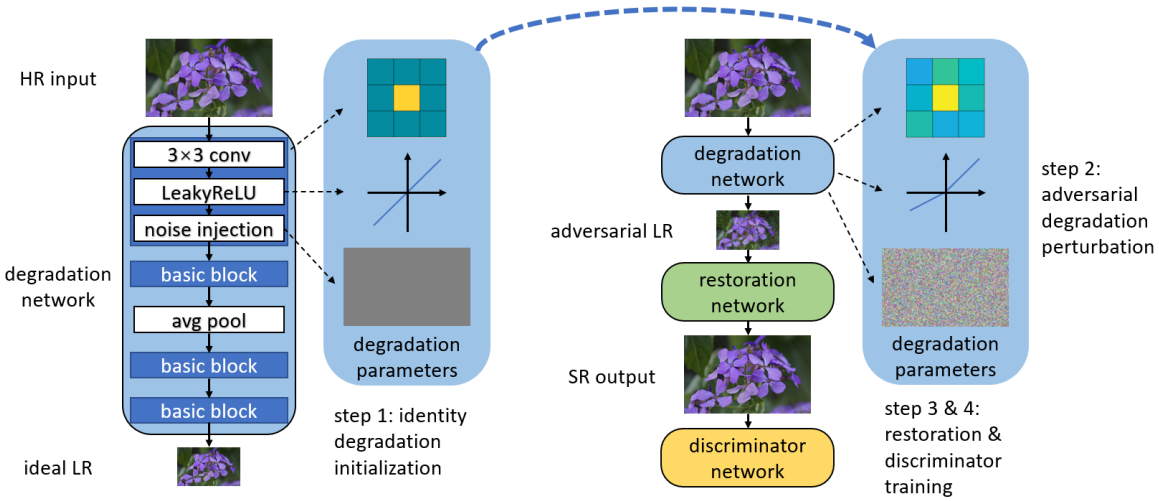


Figure 4.2: Illustration of the training procedure of our real-world super-resolution method with the proposed adversarial neural degradation model. Every single optimization step of the whole network can be divided into four sub-steps, and we highlight the internal state of the degradation network in the first two sub-steps.

times during training. Once the training is complete, we can discard the degradation network and the discriminator network, and only use the trained restoration network for inference.

In the following subsections, we will first describe the degradation network architecture and the reason why it can function as a prior to incorporate various image degradations. Then, we will explain the method used to initialize the degradation network to the identity transformation. Next, we will discuss the approach to perturb the degradation network, enabling it to represent complex image degradations. Subsequently, we will outline the adversarial training procedure of the SR model. Finally, we will analyze the training and inference efficiency of the method.

4.3.1 Degradation Network Architecture

Since we argue that almost all types of image degradation could find a corresponding operation in a standard convolutional neural network, the degradation network does not need much design for its architecture to work as a prior to include various image degradations. We can simply concatenate common convolutional neural network layers which could represent these image degradations.

Convolution layer, which is the most common layer type, is used in the degradation network to represent filter related degradations, like blur and ringing. These degradation types are also very common in the real world. Blur can be caused by camera movement or out of focus, and ringing can be caused by image compression or image sharpening technique.

Non-linear activation layer is used in the degradation network to represent global non-linear color changes, like color fading and posterization [136]. Color fading can be caused by inaccurate color response of old films, and posterization can be caused by color quantization in image compression.

Both convolution layer and activation layer can only represent spatially homogeneous image degradations, and their abilities are limited by the space invariant property of normal convolutional neural network. To represent spatially heterogeneous degradations like block artifacts in compressed images or dust spots in old images, we need to use a relatively less common layer called noise injection layer [78], which adds noise to its input in a pixel-wise manner. Combined with other layers, noise injection layer makes the degradation network able to represent complex spatially heterogeneous degradations.

Pooling layer is the last layer type we would like to discuss, and it can directly

represent a downsampling process. We use anti-aliased average pooling layer [187] rather than a normal average pooling layer to avoid aliasing in the downsampling process.

For convolution, activation and noise layer, we would like to use multiple layers of the same type in the degradation network, to make the network able to represent complicated and higher order degradations. But only one pooling layer is used in the degradation network, since it is hard to break one pooling layer with integer downsampling factor down into multiple ones with non-integer factor. We combine a 3×3 convolution layer, a LeakyReLU activation layer [114] and a noise layer to form our basic block, put 5 basic blocks before and after an average pooling layer, and put a 3×3 convolution layer at the end. Number of channels in the degradation network are all 64, except for the input channels of the first convolution layer and the output channels of the last convolution layer, which are both 3 to take RGB images as input and output of the degradation network.

4.3.2 Identity Degradation Network Initialization

Due to overparameterization of neural networks, there are infinitely many parameter solutions to make a network represent the identity transformation, even if the network architecture is fixed [179]. However, two different parameter solutions, which could identically represent the same function, may have totally different behaviors of functions in their own parameter neighbourhood. We take slightly deviated identity neural networks as prior for various real-world image degradations. If all these networks are deviated from one or a few parameter solutions, their behaviors would be severely biased and cannot cover a variety of image degradations. Therefore, we need a fast

initialization method to generate a lot of identity neural networks with different parameters.

The most straightforward initialization method, which trains networks on the identity mapping task by minimizing error using gradient descent, is way too slow for our SR model training. We propose a fast method that can initialize the degradation network to the identity transformation. Our method only takes a few small matrix multiplications and one singular value decomposition, while the most straightforward initialization method takes millions of training steps.

Before discussing our method to initialize the degradation network to the identity transformation, we would like to first clarify the meaning of the identity transformation in this work. For convolution, activation and noise layer in our degradation network, the definition of the identity transformation is strictly applicable, since the size of their input is the same as the size of their output. But for the pooling layer, the input feature is downsampled by a scale factor, so the strictly defined identity transformation no longer exists. In this work, we treat the ideal downsampling operation as a visually identity transformation[1], and use the anti-aliased average pooling layer [187] as an approximation of the ideal downsampling. This interpretation is reasonable, because an image and its ideally downsampled counterpart are very similar from the perspective of the human visual system.

The process of identity degradation initialization is illustrated in Fig. 4.3. To make the degradation network represent the identity transformation, it must first be linear. So first we remove all nonlinearities in the network, by setting the negative slopes of all LeakyReLU activation layers [114] to one. And we set all additive noises in all

[1]More formally, a function f is a visually identity transformation if for every image X, there exists a scale factor s such that $f(X) = D(X; s)$, where D is the ideal downsampling function.

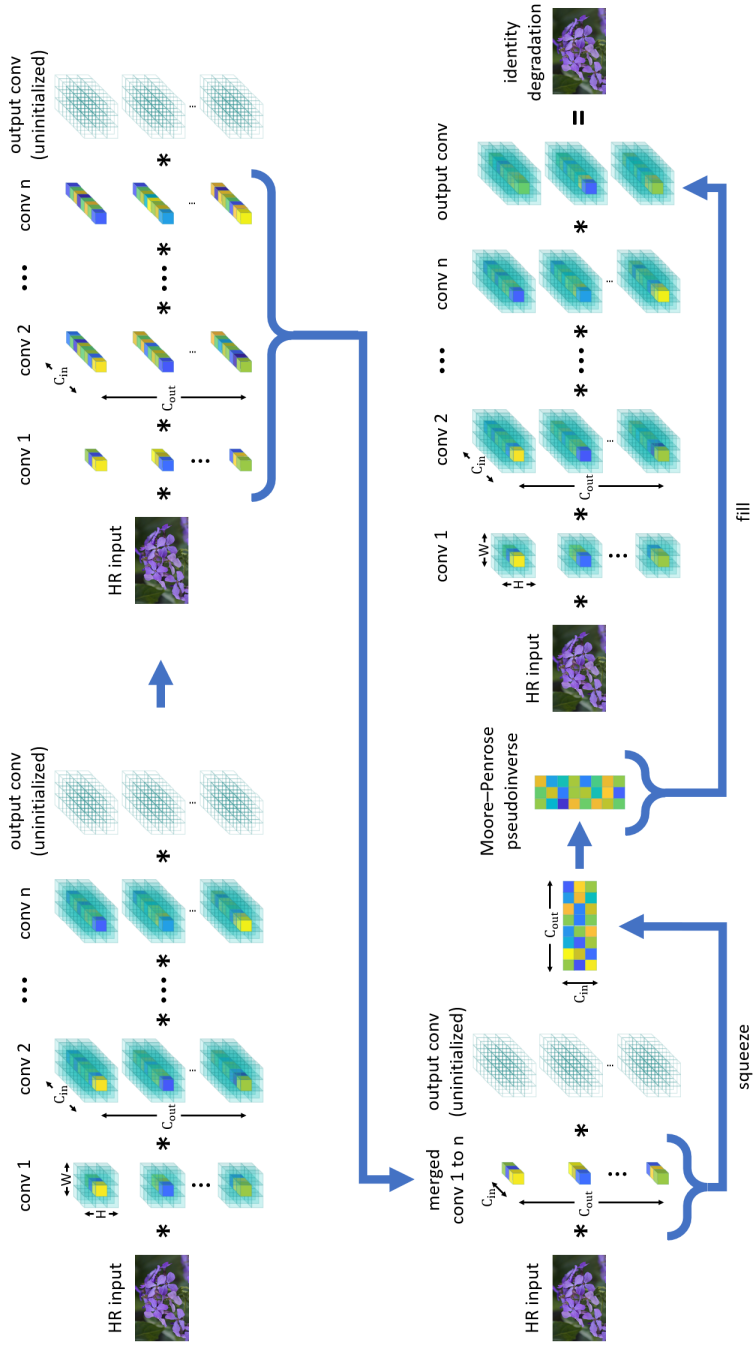


Figure 4.3: Illustration of the identity degradation initialization method in our training procedure. Only the convolution layers in the degradation network are shown in the figure.

noise injection layers to zero. Thus, these two types of layers can be removed from the degradation network and are not shown in Fig.4.3. Moreover, an output pixel of an identity network can only be affected by the counterpart pixel of the input. So then we initialize the center slices of all but the last convolution kernels (conv 1 to n) with Xavier Initialization [54] of 1×1 convolutional fan mode. We set all other values of these kernels to zero and leave the last convolution kernel (output conv) uninitialized. We can simplify these initialized 3×3 convolutions (conv 1 to n) to 1×1 convolutions and merge them into one 1×1 convolution, taking advantage of the associativity of convolution. Finally, we squeeze the merged 1×1 convolution filter (4-D tensor) into a 2-D matrix of the same size, compute the Moore-Penrose pseudoinverse of the merged matrix, fill the result into the center slice of the last 3×3 convolution kernel (output conv), and set all other values of the kernel to zero. This way, each output pixel of the degradation network is only affected by the counterpart pixel of the HR input. Computing the output pixel is equivalent to performing a matrix multiplication of the input pixel, a randomly initialized matrix, and its pseudoinverse. Thus, the property of the pseudoinverse guarantees that the entire degradation network is initialized to the identity transformation.

Please note that the anti-aliased average pooling layer [187] is not shown in Fig. 4.3. Since it is a strictly defined linear operator and we claim it as a visually identity transformation, as long as the remaining part of the network is a strictly defined identity transformation, the entire network would be a visually identity transformation or, in other words, the ideal downsampling.

Due to overparameterization of neural networks, there are infinitely many parameter solutions to make the degradation network represent the identity transformation. Our

initialization method only takes $n-1$ matrix multiplications, and one Moore-Penrose pseudoinverse by using the singular value decomposition (SVD). Our method strikes a nice balance between randomness in the network neighborhood and initialization speed.

4.3.3 Adversarial Degradation Perturbation

Once we initialize the degradation network to the identity transformation, we are at the starting point to various image degradations. The next thing we need to do is to perturb the identity degradation network, and the slightly deviated identity degradation network would represent a real-world image degradation case. We can use the network to quickly generate abundant perfectly aligned HR and LR image pairs, by taking HR images as input of the degradation network and collecting the outputs. And finally, we can train a SR model with the collected HR and LR image pairs. Since the degradation prior includes various real-world image degradations, the SR model trained by this way could reconstruct various real-world degraded images well.

So how do we perturb the identity degradation network? The most straightforward way is to add small random numbers to all parameters of the degradation network, and that means to take a small random step from the original identity transformation on the degradation space. A lot of degradation networks which are independently perturbed by this way would cover a neighbourhood of the identity transformation on the degradation space. If we train a SR model with HR and LR image pairs constructed by many of those networks, the trained model would have a good average performance on the covered degradation set.

However, instead of having a good average performance for regular degradations, we want our real-world SR model to be as robust as possible. In other word, we want our SR model to have a good worst-case performance. That is because the real-world situation is always more complicated than laboratory situations. We want our real-world SR model to keep having a satisfactory performance, even for images might have been suffered from rare or unpredictable real-world degradations.

To achieve such a goal, we perturb the identity degradation network adversarially instead of randomly. That means, we adversarially search small perturbations on all parameters of the degradation network, which could make the SR model perform the worst on HR and LR image pairs constructed by the degradation network. During SR model training, we keep searching those worst cases dynamically, and keep optimizing the SR model based on the worst degradation case for the moment. By this way, the worst-case performance of the SR model would be gradually improved, and will finally converge to the robust model with the highest lower bound.

4.3.4 Super-Resolution Model Training

To better show the advantage of the proposed neural degradation prior and the adversarial degradation training, we adopt the ESRGAN [161] as our SR model. The SR model training procedure solves the following optimization problem:

$$\begin{aligned}\min_{\theta_G}\{\ &\mathbb{E}_{I^{HR}}[\max_{\theta_F \in S} L_{cont}(I^{HR};\theta_G,\theta_F)] \\ &+ \lambda \max_{\theta_D} \mathbb{E}_{I^{HR}}[\max_{\theta_F \in S} L_{GAN}(I^{HR};\theta_G,\theta_D,\theta_F)]\}\end{aligned} \tag{4.3.1}$$

where $S = \{\theta | \|\theta - \theta^{id}\|_2 < \varepsilon$, and $F_{\theta^{id}}$ is the identity transformation$\}$. G and D are generator (restoration network) and discriminator of the SR model respectively. F

is the degradation network. θ stands for parameter of network. I^{HR} represents the high-resolution images. L_{cont} and L_{GAN} are the content loss and the GAN loss [94] respectively. λ is the coefficient to balance the two loss terms. The content loss L_{cont} is the sum of the 1-norm loss and the VGG loss [75]:

$$\begin{aligned} L_{cont}(I^{HR};\theta_G,\theta_F) =& \|I^{HR} - G_{\theta_G}(F_{\theta_F}(I^{HR}))\|_1 \\ &+ \sum_j c_j \|\phi_j(I^{HR}) - \phi_j(G_{\theta_G}(F_{\theta_F}(I^{HR})))\|_2^2 \end{aligned} \tag{4.3.2}$$

where ϕ_j is the feature map of jth convolution layer of the VGG network [143], and c_j is the coefficient for term of the jth layer. The GAN loss L_{GAN} is:

$$\begin{aligned} L_{GAN}(I^{HR};\theta_G,\theta_D,\theta_F) =& \log D_{\theta_D}(I^{HR}) \\ &- \log D_{\theta_D}(G_{\theta_G}(F_{\theta_F}(I^{HR}))) \end{aligned} \tag{4.3.3}$$

The purpose of the entire training procedure is to solve the optimization problem shown in Equation 4.3.1. We present the general algorithm for AND training in Algorithm 2. The algorithm is more complicated than the training algorithms of previous GAN-based SR methods [94, 161, 163], because there are not two, but three players in the minimax problem, i.e., the degradation network F, the generator G, and the discriminator D. For each single optimization step of the entire network, we first initialize the degradation network to the identity transformation. Next, we adversarially perturb the degradation network within a small neighborhood. The degradation network takes HR images as input and generates LR images with moderate yet complex image degradations. The restoration network, also known as the generator,

Algorithm 2 The general algorithm for AND training

Require: epoch number N, batch size m, step size α, perturbation bound ε, perturbation steps K, learning rate η

Require: initial generator parameters θ_G, initial discriminator parameters θ_D.

for *epoch* $= 1$ to N **do**

Initialize θ_F which makes the degradation network F represent the identity transformation.

Initialize perturbation on parameters of the degradation network $\delta \leftarrow 0$

Sample a minibatch $\{x^i\}_{i=1}^m$ from the high-resolution images I^{HR}.

for $k = 1$ to K **do**

$g_F \leftarrow \nabla_{\theta_F}[\frac{1}{m}\sum_{i=1}^m(L_{cont}(x^i;\theta_G,\theta_F+\delta)+\lambda L_{GAN}(x^i;\theta_G,\theta_D,\theta_F+\delta))]$

$\delta \leftarrow \delta + \alpha\frac{g_F}{\|g_F\|_2}$

if $\|\delta\|_2 > \varepsilon$ **then**

$\delta \leftarrow \varepsilon\frac{\delta}{\|\delta\|_2}$

end if

end for

$\theta_F \leftarrow \theta_F + \delta$

$g_G \leftarrow \nabla_{\theta_G}[\frac{1}{m}\sum_{i=1}^m(L_{cont}(x^i;\theta_G,\theta_F)+\lambda L_{GAN}(x^i;\theta_G,\theta_D,\theta_F))]$

$\theta_G \leftarrow \text{Adam}(-g_G,\theta_G,\eta)$

$g_D \leftarrow \nabla_{\theta_D}[\frac{1}{m}\sum_{i=1}^m \lambda L_{GAN}(x^i;\theta_G,\theta_D,\theta_F)]$

$\theta_D \leftarrow \text{Adam}(g_D,\theta_D,\eta)$

end for

aims to restore SR images from the degraded LR images. The adversarial degradation perturbation is designed to cause the restoration network to produce unsatisfactory results, characterized by low PSNR and easy distinguishability as fake images by the discriminator network. This adversarial degradation perturbation is accomplished through $K = 5$ projected perturbation steps [115]. Finally, we optimize the restoration network and the discriminator network using the adversarial LR images. We repeat the optimization steps of the entire network multiple times during training until the restoration network becomes robust enough to generate perceptually satisfying SR results, even when the LR input images are affected by complex degradations.

4.3.5 Local Worst-Case Degradation

In our proposed method, it is essential to explicitly clarify that when we mention the worst-case degradation, we specifically focus on the local worst-case scenario rather than the global worst-case scenario. This distinction is subtly implied by the proposed optimization algorithm. Our focus is solely on local worst-case degradations, not global ones, and this decision is based on the following reasons.

Firstly, searching for the global worst-case degradation poses a formidable challenge due to the inherent complexity of the image restoration network in our approach. This network, being highly nonlinear, lacks an efficient algorithm capable of quickly identifying its corresponding global worst degradation network. In the training phase of the image restoration network, every optimization step requires a search for the hard degradation scenario corresponding to the current state of the restoration network, and the entire training process demands numerous searches for hard degradations. Given these complexities, the pursuit of the global worst-case scenario proves to be an

impractical endeavor.

Secondly, it's crucial to note that even if we pinpoint the global worst-case scenario, its practical impact remains limited. The model training process involves searching for hard degradation within a small neighborhood centered around a randomly selected initial point that corresponds to the identity transformation. As a result, the global worst-case scenario is very likely to fall outside of this neighborhood. This implies that the global worst-case degradation is exceptionally severe, but such extreme circumstances are not commonly encountered in typical image restoration tasks.

Thirdly, we observed that the image restoration model, when trained using local worst-case scenarios, exhibits commendable performance in real-world image restoration tasks. This approach not only guarantees a robust lower performance limit but, crucially, by taking into account a diverse array of local worst-case degradations, it yields a strong average performance. This dual benefit is particularly significant in addressing practical tasks, emphasizing the model's ability to deliver consistent and reliable results across a spectrum of challenging scenarios.

4.3.6 Training and Inference Efficiency

Compared with previous SR methods, our SR model needs to cost more time during training phase. When we train our GAN-based SR method with adversarial neural degradation, we need to perturb the identity degradation network, optimize the restoration network and the discriminator network, in an alternating manner. The adversarial perturbation of the identity degradation network is done by taking gradient steps. In this work, before each operation step of the SR model, we use 5 gradient steps for adversarial degradation perturbation. That requires additionally 5 forward

and backward passes through the whole network, including the degradation network and the SR model. Thankfully, the degradation network is much smaller than the SR model. That is reasonable because degradation is much easier than restoration, which is a general property for all inverse problems. So the degradation network itself does not cost much, most of the additional training cost is due to the adversarial training procedure. As we mentioned before, we adopt the ESRGAN as our SR model. It would cost 8.92 TFLOPs for one training step of ESRGAN on a training batch, while would cost 57.85 TFLOPs for our SR model training with the same training setting. So our method need an increase in training time of a factor of 6.49.

However, what is more important for our real-world SR method is inference efficiency. That is because once we finish the training, the robust SR model we get would not need retraining or finetune for unpredicted image degradations. Once the training is done, we can discard the degradation network and the discriminator network, and only use the trained restoration network for inference. That means, compared with previous SR methods, our SR method does not need additional computational and storage cost during inference phase.

4.3.7 Limitations

There is no single SR model that can handle every possible image degradation. This is a simple deduction drawn from the "no free lunch" theorem [167], and our method is certainly not an exception. Our SR method relies on the proposed neural degradation prior, which is inspired by the commonalities observed in various image degradations. Therefore, naturally, our SR model cannot effectively deal with a specific image degradation that deviates significantly from the two summarized commonalities.

When an image degradation introduces artifacts containing strong structures that are not accounted for in the neural degradation prior, our SR model struggles to handle the degradation. An illustrative degradation example is extreme JPEG compression [189], which produces severe block artifacts characterized by strong spatial structures in 8×8 blocks. While networks trained specifically for this task can utilize such structures, they are not explicitly captured in our proposed neural degradation prior. As a result, our SR model would not outperform an expert network in this case. Another similar degradation example is halftoning [80]. Halftone printing is a technique that uses ink dots of different sizes to simulate different grayscale levels, and it is used in old publications such as newspapers or books. Since the positions of the ink dots of a halftone image always have a very strong spatial pattern, which is not included in the neural degradation prior, our method cannot restore the halftone image well compared to an expert network.

The prior assumes that image degradations can be viewed as small deviations from the identity transformation. If this assumption fails for a specific image degradation, our SR model cannot handle the degradation well. A notable degradation example is the conversion of truecolor images to grayscale images [70]. This particular degradation is not a small deviation from the identity transformation, and thus, our SR model would not be able to colorize the input grayscale images in such cases. For the same reason, our method is also not suitable for directly enhancing low-light images [108]. Since the image degradations used in our SR training are slightly deviated from identity degradation networks, our trained model would not automatically adjust the image brightness, contrast, and color.

For certain image degradations, both assumptions in our neural degradation prior

can fail simultaneously, representing the most challenging cases for our method. An example of such a case is image degradation in the image inpainting task [104]. In this task, the missing pixels often exhibit strong spatial structures, such as forming holes and stripes on the image. Moreover, the signal of these pixels is entirely absent, rather than being a small deviation from the identity transformation. Consequently, our image restoration method is unable to address this type of degradation.

4.4 Experiments

4.4.1 Datasets

Widely used datasets for SR evaluation, like Urban100 [68] and DIV2K [153], are not suitable for the study of real-world SR, because they only contain HR images and their LR counterpart need to be generated by bicubic downsampling. We perform experiments on four datasets constructed for real-world SR evaluation: RealSR [21], DRealSR [166], SupER [86], and ImagePairs [76]. RealSR and DRealSR are datasets containing HR and LR image pairs captured on the same scene by adjusting the focal length of digital cameras. SupER includes HR and LR image pairs constructed by camera hardware binning, which aggregates adjacent pixels on the sensor array. ImagePairs includes HR and LR image pairs captured by a HR camera and a LR camera, which are aligned and mounted on a rig with a beam splitter. These datesets are constructed in different ways, so they could provide a comprehensive evaluation for real-world SR methods.

4.4.2 Quantitative Metrics

We use four quantitative metrics for quality assessment of SR images: PSNR, SSIM [164], LPIPS [188], and NIQE [121]. PSNR and SSIM are calculated on Y channel of transformed YCbCr space for fair comparison [152]. They are more focused on low-level pixel-wise image differences, and they are metrics suitable for PSNR-oriented SR models. LPIPS is a learned metric for full-reference image quality assessment. We could use the preceding three full-reference metrics, since all datasets we used in the experiments have pixel-wise aligned HR and LR image pairs. Considering that GAN-based SR methods may generate detailed textures, which is although realistic but different from the ground truth, we also use NIQE, a no-reference metric for image quality evaluation. Both LPIPS and NIQE agree better with human visual perception, and they are metrics suitable for perceptual quality-oriented SR models.

4.4.3 Training Details

We use DIV2K [153], Flickr2K [103] and WED [113] as HR image datasets for training. The training HR patch size is set to 256 and the batch size is set to 48. Following BSRGAN [185] and Real-ESRGAN [163], we train two SR models with our method: a PSNR-oriented model noted as ANDNet, and a perceptual quality-oriented model noted as ANDGAN. First, we train ANDNet with the L1 loss only, for 1×10^6 iterations with 1×10^{-4} learning rate. Then we use the trained ANDNet as an initialization for the generator of the ANDGAN, and train the whole ANDGAN model with both the content loss and the GAN loss in Equation 4.3.1, which are balanced by $\lambda = 0.1$, for 5×10^5 iterations with 1×10^{-4} learning rate. We use Adam optimizer [82] for both generator and discriminator training.

Table 4.1: Quantitative comparison with state-of-the-art methods on real-world blind image super-resolution benchmarks. Best and second best results are **highlighted** and underlined

Method	RealSR(×4) [21]		DRealSR(×4) [166]		SupER(×4) [86]		ImgPairs(×2) [76]	
	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓	PSNR↑	LPIPS↓
KernelGAN [9]	25.13	0.3349	28.56	0.3978	25.65	0.3445	26.74	0.3340
DAN [69]	27.80	0.4114	30.59	0.4111	32.19	0.2064	28.56	0.2802
BSRNet [185]	27.35	0.3084	29.49	0.3411	32.11	0.2532	28.59	0.3915
BSRGAN [185]	26.51	0.2685	28.35	0.2929	29.18	0.2181	28.13	0.3346
Real-ESRNet [163]	26.79	0.2939	28.50	0.3257	30.89	0.2496	28.34	0.3858
Real-ESRGAN [163]	25.85	0.2728	27.92	0.2818	27.55	0.2046	28.12	0.3679
SwinIR-Real [100]	26.43	0.2515	28.29	0.2739	28.27	0.1889	28.11	0.3464
DCLS [110]	27.83	0.4080	28.32	0.4760	32.71	0.1985	28.64	0.2844
PDM-SRGAN [111]	21.96	0.3717	24.32	0.3668	25.31	0.2710	26.11	0.3788
FeMaSR [28]	25.42	0.2937	26.59	0.3374	25.45	0.2419	27.03	0.3400
DASR [101]	27.18	0.3113	29.72	0.2962	29.73	0.1476	28.34	0.3412
ReDegNet [98]	24.77	0.2800	26.24	0.2995	26.60	0.1785	27.06	0.3930
ANDNet (ours)	**28.47**	0.2599	**30.97**	0.3381	**32.96**	0.2125	**28.75**	0.2786
ANDGAN (ours)	26.34	**0.2326**	28.95	**0.2610**	29.85	**0.1372**	27.78	**0.2598**

Method	RealSR(×4) [21]		DRealSR(×4) [166]		SupER(×4) [86]		ImgPairs(×2) [76]	
	SSIM↑	NIQE↓	SSIM↑	NIQE↓	SSIM↑	NIQE↓	SSIM↑	NIQE↓
KernelGAN [9]	0.7407	6.946	0.8314	8.550	0.7831	6.844	0.7467	6.291
DAN [69]	0.7882	8.099	0.8608	9.137	0.8880	5.886	0.7917	5.692
BSRNet [185]	0.8076	7.271	0.8587	8.060	0.8800	6.322	0.8311	6.391
BSRGAN [185]	0.7750	4.650	0.8205	4.681	0.8292	4.549	0.8152	5.450
Real-ESRNet [163]	0.8067	7.142	0.8484	7.829	0.8563	6.408	0.8264	6.026
Real-ESRGAN [163]	0.7735	4.676	0.8247	4.716	0.8082	3.944	0.8192	4.812
SwinIR-Real [100]	0.7865	4.678	0.8272	4.665	0.8360	3.776	0.8133	4.315
DCLS [110]	0.7892	8.023	0.8188	9.273	0.8927	5.892	0.7962	5.773
PDM-SRGAN [111]	0.6815	6.798	0.7728	7.518	0.8048	5.015	0.7788	5.316
FeMaSR [28]	0.7531	4.737	0.7683	4.218	0.7429	4.873	0.7477	4.719
DASR [101]	0.7867	5.969	0.8543	6.347	0.8508	3.881	0.8204	4.830
ReDegNet [98]	0.7754	5.049	0.8124	4.685	0.8305	3.993	0.8130	5.368
ANDNet (ours)	**0.8232**	7.541	**0.8773**	8.360	**0.9004**	6.302	**0.8415**	5.514
ANDGAN (ours)	0.7854	**4.018**	0.8244	**4.045**	0.8579	**3.493**	0.7858	**3.748**

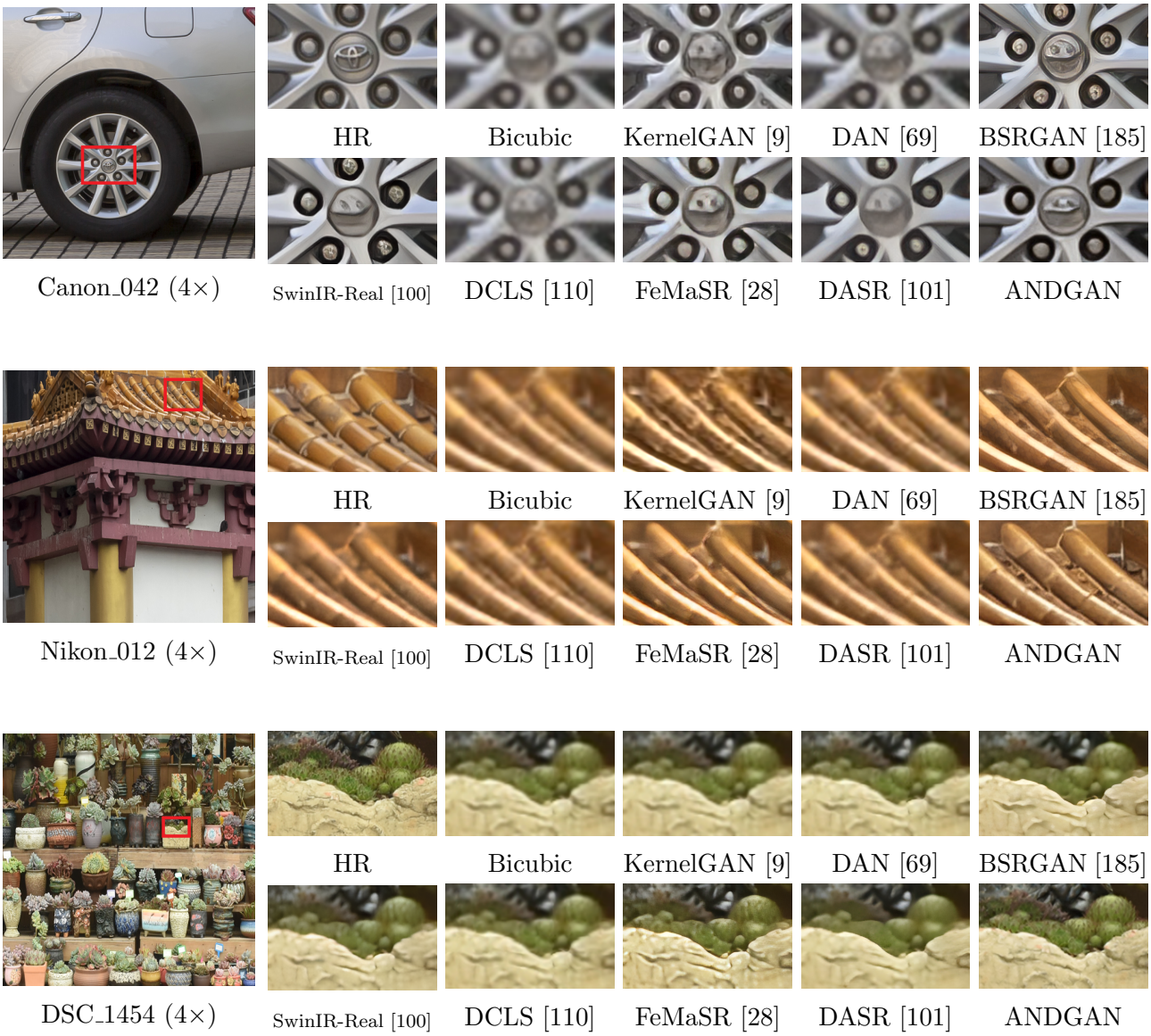


Figure 4.4: Qualitative comparisons on real-world images from RealSR [21] and DRealSR [166] dataset with scale factor 4.

We use projected gradient descent method [115] to adversarially search a small deviation of the identity degradation network. Before every training step for the restoration network, we initialize the degradation network to the identity transformation, and run 5 iterations of projected gradient descent with step size of 6 and perturbation size $\varepsilon = 20$. We also need to balance the weights for convolutional degradation, noise and nonlinearity, to make their respective intensity close to real-world degradation. When we calculate the L2 norm $\|\theta - \theta^{id}\|_2$ to determine the perturbation set S in Equation 4.3.1, we use scale factors of 1, 10, 50 for term of convolutional degradation, noise and nonlinearity respectively. Note that a larger scale factor means a stronger suppression for the degradation type.

4.4.4 Comparisons with Prior Works

We compare both our PSNR-oriented model and the perceptual quality-oriented model with several state-of-the-art methods. Quantitative results are shown in Table 4.1, and visual comparison between different methods are shown in Fig. 4.4. The experimental results show the superiority of our method on all four real-world SR datasets over the state of the arts. As shown in Fig. 4.4, our ANDGAN model is the only SR method which could handle the difficult degradations and recover the image details of the wheel hub, roofing tiles, and cactus spines.

4.4.5 Ablation Study

The ablation studies are intended to investigate the effects of the three major components of our method: adversarial perturbation, neural degradation, and identity initialization. In each ablation setting, we retain only specific components and assess

the method's performance. For cases where neural degradation is removed, we either use an additive noise model for adversarial training or employ bicubic downsampling for non-adversarial training. In situations where identity initialization is eliminated, we randomly initialize all 3×3 convolution kernels in the degradation network using Xavier Initialization. The meanings of each specific ablation setting are further explained below.

If we do not use adversarial perturbation and neural degradation at all, our method would become a classical SR training method, which assumes that the image degradation model is an ideal bicubic downsampling. Most SR researches are with this setting, such as SRCNN [39], EDSR [103] and RCAN [190].

If we retain solely the adversarial perturbation without inducing neural degradation, it implies the utilization of an adversarial noise training method similar to [24, 176]. The perturbation under this setting is additive noise, following most adversarial training researches on image classification tasks.

If we employ neural degradation with identity initialization but without adversarial perturbation, our method becomes similar to SR training with synthetic data augmentation, functioning akin to BSRGAN [185] and Real-ESRGAN [163]. During model training, neural degradation involves random sampling near the identity initialization rather than adversarial sampling, and the generated LR patches function as synthetic data augmentations.

If we only remove the identity initialization from our method, the generated LR patches would no longer be visually similar to the HR patches. While the skeleton of the LR patches would remain unaffected, their color and texture would undergo a dramatic change, as the mapping would no longer be an identity mapping [170]. This

Table 4.2: Comparisons showing the effects of each component in the AND model, tested on the RealSR [21] dataset with a scale factor of 4.

Configuration	Adversarial Perturbation	Neural Degradation	Identity Initialization	PSNR↑ of ANDNet	LPIPS↓ of ANDGAN
Classical SR training	✗	✗	-	26.53	0.4245
Adversarial noise training	✓	✗	-	26.60	0.4194
Synthetic data augmentation	✗	✓	✓	27.31	0.3089
Severe random style shift	✓	✓	✗	11.84	-
Complete AND model	✓	✓	✓	28.47	0.2326

phenomenon is referred to as "severe random style shift" in our experiments.

The comparisons are shown in Table 4.2. We can observe that all three major components, namely adversarial perturbation, neural degradation, and identity initialization, are necessary.

4.5 Conclusions

We propose a neural degradation prior that encompasses various image degradations in the real world. Specifically, an untrained convolutional neural network, which deviates slightly from the identity transformation, can serve as a prior for various real-world image degradations. We employ adversarial searches to find small deviations in the degradation network during the training of the SR model. This approach allows the restoration model to continuously optimize itself on the worst degradation case, thus achieving robustness.

Chapter 5

Conclusion

This thesis departs from the current practice of designing an image restoration method for a given known narrow class of degradations, and it strives for a degree of universality of restoration methods. We improve the universality of the current DNN methods for image restoration in three aspects.

1. **Network design.** We propose a novel system called the functional neural network (FuncNet) to solve a parametric image restoration problem with a single model. Unlike a plain neural network, the smallest conceptual element of our FuncNet is no longer a floating-point variable, but a function of the degradation severity parameter of the problem.

2. **Training strategy.** We propose a novel adversarial neural degradation (AND) model to solve the problem of real-world super-resolution, which is the most common type of image restoration against complex compounded degradations. Instead of attempting to exhaust all degradation variants in simulation, which is unwieldy and impractical, the AND model, when trained in conjunction with

a deep restoration neural network under a minmax criterion, can generate a wide range of highly nonlinear complex degradation effects without any explicit supervision.

3. **Inference process.** We abstract any image degradation process as a many-to-one function and propose a general restoration method with only one trained model for various image restoration problems. The general image restoration is formulated as a constrained optimization problem. Its objective is to maximize a posteriori probability of latent variables, and its constraint is that the image generated by these latent variables must be sufficiently close to the degraded image.

The above three contributions endow deep learning based image restoration methods with a much desired degree of universality. They lead to appreciable improvements of both subjective and objective quality of restored images.

Bibliography

[1] A. Abdelhamed, S. Lin, and M. S. Brown. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1692–1700, 2018.

[2] M. V. Afonso, J. M. Bioucas-Dias, and M. A. Figueiredo. An augmented lagrangian approach to the constrained optimization formulation of imaging inverse problems. *IEEE Transactions on Image Processing*, 20(3):681–695, 2011.

[3] E. Agustsson and R. Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 126–135, 2017.

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

[5] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2010.

[6] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[7] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018.

[8] M. R. Banham and A. K. Katsaggelos. Digital image restoration. *IEEE signal processing magazine*, 14(2):24–41, 1997.

[9] S. Bell-Kligler, A. Shocher, and M. Irani. Blind super-resolution kernel estimation using an internal-gan. *Advances in Neural Information Processing Systems*, 32, 2019.

[10] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012.

[11] K. K. Bhatia, A. N. Price, W. Shi, J. V. Hajnal, and D. Rueckert. Super-resolution reconstruction of cardiac mri using coupled dictionary learning. In *2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI)*, pages 947–950. IEEE, 2014.

[12] J. M. Bioucas-Dias and M. A. Figueiredo. A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Transactions on Image processing*, 16(12):2992–3004, 2007.

[13] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[14] Y. Blau and T. Michaeli. The perception-distortion tradeoff. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6228–6237, 2018.

[15] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 7, 2017.

[16] A. Brock, J. Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

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

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

[19] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *2012 IEEE conference on computer vision and pattern recognition*, pages 2392–2399. IEEE, 2012.

[20] J. T. Bushberg and J. M. Boone. *The essential physics of medical imaging*. Lippincott Williams & Wilkins, 2011.

[21] J. Cai, H. Zeng, H. Yong, Z. Cao, and L. Zhang. Toward real-world single image super-resolution: A new benchmark and a new model. In *Proceedings of the*

IEEE/CVF International Conference on Computer Vision, pages 3086–3095, 2019.

[22] J. B. Campbell and R. H. Wynne. *Introduction to remote sensing*. Guilford press, 2011.

[23] P. Campisi and K. Egiazarian. *Blind image deconvolution: theory and applications*. CRC press, 2016.

[24] A. Castillo, M. Escobar, J. C. Pérez, A. Romero, R. Timofte, L. Van Gool, and P. Arbelaez. Generalized real-world super-resolution through adversarial robustness. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1855–1865, 2021.

[25] L. Cavigelli, P. Hager, and L. Benini. Cas-cnn: A deep convolutional neural network for image compression artifact suppression. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 752–759. IEEE, 2017.

[26] A. Chakrabarti. A neural approach to blind motion deblurring. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 221–235. Springer, 2016.

[27] H. Chang, M. K. Ng, and T. Zeng. Reducing artifacts in jpeg decompression via a learned dictionary. *IEEE transactions on signal processing*, 62(3):718–728, 2013.

[28] C. Chen, X. Shi, Y. Qin, X. Li, X. Han, T. Yang, and S. Guo. Real-world blind super-resolution via feature matching with implicit high-resolution priors. In

Proceedings of the 30th ACM International Conference on Multimedia, pages 1329–1338, 2022.

[29] Y. Chen, S. Liu, and X. Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8628–8638, 2021.

[30] Y. Choi, M. El-Khamy, and J. Lee. Variable rate deep image compression with a conditional autoencoder. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3146–3154, 2019.

[31] R. Cogranne. Determining jpeg image standard quality factor from the quantization tables. *arXiv preprint arXiv:1802.00992*, 2018.

[32] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, et al. A system for video surveillance and monitoring. *VSAM final report*, 2000(1-68):1, 2000.

[33] A. Criminisi, P. Pérez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing*, 13 (9):1200–1212, 2004.

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

[35] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang. Second-order attention network for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11065–11074, 2019.

[36] S. Dodge and L. Karam. Understanding how image quality affects deep neural networks. In *2016 eighth international conference on quality of multimedia experience (QoMEX)*, pages 1–6. IEEE, 2016.

[37] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13*, pages 184–199. Springer, 2014.

[38] C. Dong, Y. Deng, C. C. Loy, and X. Tang. Compression artifacts reduction by a deep convolutional network. In *Proceedings of the IEEE international conference on computer vision*, pages 576–584, 2015.

[39] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[40] W. Dong, L. Zhang, G. Shi, and X. Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Transactions on image processing*, 20(7):1838–1857, 2011.

[41] M. Ehrlich, S.-N. Lim, L. Davis, and A. Shrivastava. Quantization guided jpeg artifact correction. *arXiv*, pages arXiv–2004, 2020.

[42] J. L. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[43] A. M. Eskicioglu and P. S. Fisher. Image quality measures and their performance. *IEEE Transactions on communications*, 43(12):2959–2965, 1995.

[44] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[45] A. Foi, V. Katkovnik, and K. Egiazarian. Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images. *IEEE transactions on image processing*, 16(5):1395–1411, 2007.

[46] R. Franzen. Kodak lossless true color image suite. *source: http://r0k. us/graphics/kodak*, 4(2):9, 1999.

[47] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *International journal of computer vision*, 40(1):25–47, 2000.

[48] B. R. Frieden. Restoring with maximum likelihood and maximum entropy. *JOSA*, 62(4):511–518, 1972.

[49] X. Fu, Z.-J. Zha, F. Wu, X. Ding, and J. Paisley. Jpeg artifacts reduction via deep convolutional sparse coding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2501–2510, 2019.

[50] A. Fujimoto, T. Ogawa, K. Yamamoto, Y. Matsui, T. Yamasaki, and K. Aizawa. Manga109 dataset and creation of metadata. In *Proceedings of the 1st international workshop on comics analysis, processing and understanding*, pages 1–5, 2016.

[51] L. Galteri, L. Seidenari, M. Bertini, and A. Del Bimbo. Deep generative adversarial compression artifact removal. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4826–4835, 2017.

[52] H. Gao, X. Tao, X. Shen, and J. Jia. Dynamic scene deblurring with parameter selective sharing and nested skip connections. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3848–3856, 2019.

[53] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[54] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[55] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[56] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[57] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

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

[59] C. Guillemot and O. Le Meur. Image inpainting: Overview and recent advances. *IEEE signal processing magazine*, 31(1):127–144, 2014.

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

[61] J. Guo and H. Chao. Building dual-domain representations for compression artifacts reduction. In *European Conference on Computer Vision*, pages 628–644. Springer, 2016.

[62] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang. Toward convolutional blind denoising of real photographs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1712–1722, 2019.

[63] Y. Guo, X. Zhang, and X. Wu. Deep multi-modality soft-decoding of very low bit-rate face videos. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 3947–3955, 2020.

[64] J. He, C. Dong, and Y. Qiao. Interactive multi-dimension modulation with dynamic controllable residual learning for image restoration. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 53–68. Springer, 2020.

[65] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[66] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 630–645. Springer, 2016.

[67] X. Hu, H. Mu, X. Zhang, Z. Wang, T. Tan, and J. Sun. Meta-sr: A magnification-arbitrary network for super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1575–1584, 2019.

[68] J.-B. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5197–5206, 2015.

[69] Y. Huang, S. Li, L. Wang, T. Tan, et al. Unfolding the alternating optimization for blind super resolution. *Advances in Neural Information Processing Systems*, 33:5632–5643, 2020.

[70] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Let there be color! joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics (ToG)*, 35(4):1–11, 2016.

[71] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[72] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

[73] V. Jain and S. Seung. Natural image denoising with convolutional networks. *Advances in neural information processing systems*, 21:769–776, 2008.

[74] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool. Dynamic filter networks. In *Advances in neural information processing systems*, pages 667–675, 2016.

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

[76] H. R. V. Joze, I. Zharkov, K. Powell, C. Ringler, L. Liang, A. Roulston, M. Lutz, and V. Pradeep. Imagepairs: Realistic super resolution dataset via beam splitter camera rig. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 518–519, 2020.

[77] D. Kang, D. Dhar, and A. B. Chan. Crowd counting by adapting convolutional neural networks with side information. *arXiv preprint arXiv:1611.06748*, 2016.

[78] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

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

[80] T.-H. Kim and S. I. Park. Deep context-aware descreening and rescreening of halftone images. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018.

[81] Y. Kim, J. W. Soh, and N. I. Cho. Agarnet: adaptively gated jpeg compression artifacts removal network for a wide range quality factor. *IEEE Access*, 8: 20160–20170, 2020.

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

[83] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[84] B. Klein, L. Wolf, and Y. Afek. A dynamic convolutional layer for short range weather prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4840–4848, 2015.

[85] R. Köhler, M. Hirsch, B. Mohler, B. Schölkopf, and S. Harmeling. Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VII 12*, pages 27–40. Springer, 2012.

[86] T. Köhler, M. Bätz, F. Naderi, A. Kaup, A. Maier, and C. Riess. Toward bridging the simulated-to-real gap: Benchmarking super-resolution on real data. *IEEE transactions on pattern analysis and machine intelligence*, 42(11):2944–2959, 2019.

[87] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[88] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8183–8192, 2018.

[89] Y. Kwon, K. I. Kim, J. Tompkin, J. H. Kim, and C. Theobalt. Efficient learning of image super-resolution and compression artifact removal with semi-local gaussian processes. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1792–1805, 2015.

[90] S. Laine, T. Karras, J. Lehtinen, and T. Aila. High-quality self-supervised deep image denoising. *Advances in Neural Information Processing Systems*, 32, 2019.

[91] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[92] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[93] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[94] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image

super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[95] S. Lefkimmiatis. Non-local color image denoising with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3587–3596, 2017.

[96] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *2009 IEEE conference on computer vision and pattern recognition*, pages 1964–1971. IEEE, 2009.

[97] S. Li, H. Yin, and L. Fang. Group-sparse representation with dictionary learning for medical image denoising and fusion. *IEEE Transactions on biomedical engineering*, 59(12):3450–3459, 2012.

[98] X. Li, C. Chen, X. Lin, W. Zuo, and L. Zhang. From face to natural image: Learning real degradation for blind image super-resolution. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVIII*, pages 376–392. Springer, 2022.

[99] Y. Li, Y. Fan, X. Xiang, D. Demandolx, R. Ranjan, R. Timofte, and L. Van Gool. Efficient and explicit modelling of image hierarchies for image restoration. *arXiv preprint arXiv:2303.00748*, 2023.

[100] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1833–1844, 2021.

[101] J. Liang, H. Zeng, and L. Zhang. Efficient and degradation-adaptive network for real-world image super-resolution. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVIII*, pages 574–591. Springer, 2022.

[102] S. Liang, Y. Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

[103] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017.

[104] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European conference on computer vision (ECCV)*, pages 85–100, 2018.

[105] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo. Multi-level wavelet-cnn for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 773–782, 2018.

[106] X. Liu, M. Tanaka, and M. Okutomi. Single-image noise level estimation for blind denoising. *IEEE transactions on image processing*, 22(12):5226–5237, 2013.

[107] X. Liu, X. Wu, J. Zhou, and D. Zhao. Data-driven sparsity-based restoration of jpeg-compressed images in dual transform-pixel domain. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5171–5178, 2015.

[108] K. G. Lore, A. Akintayo, and S. Sarkar. Llnet: A deep autoencoder approach to natural low-light image enhancement. *Pattern Recognition*, 61:650–662, 2017.

[109] F. Luo and X. Wu. Maximum a posteriori on a submanifold: a general image restoration method with gan. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2020.

[110] Z. Luo, H. Huang, L. Yu, Y. Li, H. Fan, and S. Liu. Deep constrained least squares for blind image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17642–17652, 2022.

[111] Z. Luo, Y. Huang, S. Li, L. Wang, and T. Tan. Learning the degradation distribution for blind image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6063–6072, 2022.

[112] C. Ma, C.-Y. Yang, X. Yang, and M.-H. Yang. Learning a no-reference quality metric for single-image super-resolution. *Computer Vision and Image Understanding*, 158:1–16, 2017.

[113] K. Ma, Z. Duanmu, Q. Wu, Z. Wang, H. Yong, H. Li, and L. Zhang. Waterloo exploration database: New challenges for image quality assessment models. *IEEE Transactions on Image Processing*, 26(2):1004–1016, 2016.

[114] A. L. Maas, A. Y. Hannun, A. Y. Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Atlanta, Georgia, USA, 2013.

[115] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[116] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Transactions on image processing*, 17(1):53–69, 2008.

[117] X. Mao, C. Shen, and Y.-B. Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. *Advances in neural information processing systems*, 29, 2016.

[118] D. Martin, C. Fowlkes, D. Tal, and J. Malik. 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 2001*, volume 2, pages 416–423. IEEE, 2001.

[119] Y. Mei, Y. Fan, Y. Zhou, L. Huang, T. S. Huang, and H. Shi. Image super-resolution with cross-scale non-local attention and exhaustive self-exemplars mining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5690–5699, 2020.

[120] P. Milanfar. A tour of modern image filtering: New insights and methods, both practical and theoretical. *IEEE signal processing magazine*, 30(1):106–128, 2012.

[121] A. Mittal, R. Soundararajan, and A. C. Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal processing letters*, 20(3):209–212, 2012.

[122] K. Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. In *International conference on machine learning*, pages 10–18. PMLR, 2013.

[123] S. Nah, T. Hyun Kim, and K. Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3883–3891, 2017.

[124] J. Newell. Old objects, new media: Historical collections, digitization and affect. *Journal of Material Culture*, 17(3):287–306, 2012.

[125] B. Niu, W. Wen, W. Ren, X. Zhang, L. Yang, S. Wang, K. Zhang, X. Cao, and H. Shen. Single image super-resolution via a holistic attention network. In *European Conference on Computer Vision*, pages 191–207. Springer, 2020.

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

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

[128] W. B. Pennebaker and J. L. Mitchell. *JPEG: Still image data compression standard*. Springer Science & Business Media, 1992.

[129] X. Pennec. *Probabilities and statistics on riemannian manifolds: A geometric approach*. PhD thesis, INRIA, 2004.

[130] T. Plotz and S. Roth. Benchmarking denoising algorithms with real photographs.

In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1586–1595, 2017.

[131] S. Pyatykh, J. Hesser, and L. Zheng. Image noise level estimation by principal component analysis. *IEEE transactions on image processing*, 22(2):687–699, 2012.

[132] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[133] W. H. Richardson. Bayesian-based iterative method of image restoration. *JoSA*, 62(1):55–59, 1972.

[134] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[135] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[136] E. Rusak, L. Schott, R. S. Zimmermann, J. Bitterwolf, O. Bringmann, M. Bethge, and W. Brendel. A simple way to make neural networks robust against diverse image corruptions. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 53–69. Springer, 2020.

[137] R. Salakhutdinov and G. Hinton. Deep boltzmann machines. In *Artificial intelligence and statistics*, pages 448–455. PMLR, 2009.

[138] R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, pages 791–798, 2007.

[139] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

[140] H. Sheikh. Live image quality assessment database release 2. *http://live. ece. utexas. edu/research/quality*, 2005.

[141] Z. Shen, W. Wang, X. Lu, J. Shen, H. Ling, T. Xu, and L. Shao. Human-aware motion deblurring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5572–5581, 2019.

[142] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.

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

[144] C. K. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár. Amortised map inference for image super-resolution. *arXiv preprint arXiv:1610.04490*, 2016.

[145] V. Soni, A. K. Bhandari, A. Kumar, and G. K. Singh. Improved sub-band adaptive thresholding function for denoising of satellite image based on evolutionary algorithms. *IET Signal Processing*, 7(8):720–730, 2013.

[146] F. Stanco, G. Ramponi, and A. De Polo. Towards the automated restoration of old photographic prints: a survey. In *The IEEE Region 8 EUROCON 2003. Computer as a Tool.*, volume 2, pages 370–374. IEEE, 2003.

[147] J. Sun, W. Cao, Z. Xu, and J. Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 769–777, 2015.

[148] L. Sun, S. Cho, J. Wang, and J. Hays. Edge-based blur kernel estimation using patch priors. In *IEEE international conference on computational photography (ICCP)*, pages 1–8. IEEE, 2013.

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

[150] Y. Tai, J. Yang, X. Liu, and C. Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE international conference on computer vision*, pages 4539–4547, 2017.

[151] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia. Scale-recurrent network for deep image deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8174–8182, 2018.

[152] R. Timofte, V. De Smet, and L. Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *Computer Vision–ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part IV 12*, pages 111–126. Springer, 2015.

[153] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, and L. Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 114–125, 2017.

[154] M. E. Tipping and C. M. Bishop. Bayesian image super-resolution. In *Advances in neural information processing systems*, pages 1303–1310, 2003.

[155] T. Tong, G. Li, X. Liu, and Q. Gao. Image super-resolution using dense skip connections. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4799–4807, 2017.

[156] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[157] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. *arXiv preprint arXiv:1711.10925*, 2017.

[158] M. Wang and W. Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[159] W. Wang, R. Guo, Y. Tian, and W. Yang. Cfsnet: Toward a controllable feature

space for image restoration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4140–4149, 2019.

[160] X. Wang, K. Yu, C. Dong, and C. C. Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 606–615, 2018.

[161] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018.

[162] X. Wang, K. Yu, C. Dong, X. Tang, and C. C. Loy. Deep network interpolation for continuous imagery effect transition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1692–1701, 2019.

[163] X. Wang, L. Xie, C. Dong, and Y. Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1905–1914, 2021.

[164] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[165] Z. Wang, D. Liu, S. Chang, Q. Ling, Y. Yang, and T. S. Huang. D3: Deep dual-domain based fast restoration of jpeg-compressed images. In *Proceedings*

of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2764–2772, 2016.

[166] P. Wei, Z. Xie, H. Lu, Z. Zhan, Q. Ye, W. Zuo, and L. Lin. Component divide-and-conquer for real-world image super-resolution. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pages 101–117. Springer, 2020.

[167] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.

[168] E. Wong and Z. Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International conference on machine learning*, pages 5286–5295. PMLR, 2018.

[169] J. Xu, H. Li, Z. Liang, D. Zhang, and L. Zhang. Real-world noisy image denoising: A new benchmark. *arXiv preprint arXiv:1804.02603*, 2018.

[170] Z. Xu, D. Liu, J. Yang, C. Raffel, and M. Niethammer. Robust and generalizable visual representation learning via random convolutions. *arXiv preprint arXiv:2007.13003*, 2020.

[171] R. A. Yeh, C. Chen, T.-Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do. Semantic image inpainting with deep generative models. In *CVPR*, volume 2, page 4, 2017.

[172] R. A. Yeh, T. Y. Lim, C. Chen, A. G. Schwing, M. Hasegawa-Johnson, and M. Do. Image restoration with deep generative models. In *2018 IEEE International*

Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6772–6776. IEEE, 2018.

[173] D. Yıldırım and O. Güngör. A novel image fusion method using ikonos satellite images. *Journal of Geodesy and Geoinformation*, 1(1):75–83, 2012.

[174] C. Yim and A. C. Bovik. Quality assessment of deblocked images. *IEEE Transactions on Image Processing*, 20(1):88–98, 2010.

[175] S. Yu, B. Park, and J. Jeong. Deep iterative down-up cnn for image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[176] J. Yue, H. Li, P. Wei, G. Li, and L. Lin. Robust real-world image super-resolution against adversarial attacks. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 5148–5157, 2021.

[177] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao. Learning enriched features for real image restoration and enhancement. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pages 492–511. Springer, 2020.

[178] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. In *Curves and Surfaces: 7th International Conference, Avignon, France, June 24-30, 2010, Revised Selected Papers 7*, pages 711–730. Springer, 2012.

[179] C. Zhang, S. Bengio, M. Hardt, M. C. Mozer, and Y. Singer. Identity crisis:

Memorization and generalization under extreme overparameterization. *arXiv preprint arXiv:1902.04698*, 2019.

[180] J. Zhang, J. Pan, J. Ren, Y. Song, L. Bao, R. W. Lau, and M.-H. Yang. Dynamic scene deblurring using spatially variant recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2521–2529, 2018.

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

[182] K. Zhang, W. Zuo, S. Gu, and L. Zhang. Learning deep cnn denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3929–3938, 2017.

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

[184] K. Zhang, L. V. Gool, and R. Timofte. Deep unfolding network for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3217–3226, 2020.

[185] K. Zhang, J. Liang, L. Van Gool, and R. Timofte. Designing a practical degradation model for deep blind image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4791–4800, 2021.

[186] L. Zhang, X. Wu, A. Buades, and X. Li. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. *Journal of Electronic imaging*, 20(2):023016–023016, 2011.

[187] R. Zhang. Making convolutional networks shift-invariant again. In *International conference on machine learning*, pages 7324–7334. PMLR, 2019.

[188] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[189] X. Zhang, W. Yang, Y. Hu, and J. Liu. Dmcnn: Dual-domain multi-scale convolutional neural network for compression artifacts removal. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 390–394. IEEE, 2018.

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

[191] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2472–2481, 2018.

[192] Z. Zhang, Z. Wang, Z. Lin, and H. Qi. Image super-resolution by neural texture transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7982–7991, 2019.

[193] H. Zhao, O. Gallo, I. Frosio, and J. Kautz. Loss functions for neural networks for image processing. *arXiv preprint arXiv:1511.08861*, 2015.

[194] B. Zoph and Q. V. Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

[195] W. W. Zou and P. C. Yuen. Very low resolution face recognition problem. *IEEE Transactions on image processing*, 21(1):327–340, 2011.