Deep Unfolding Network for Image Super-Resolution

Kai Zhang   Luc Van Gool   Radu Timofte

Computer Vision Lab, ETH Zurich, Switzerland

{kai.zhang, vangool, timofter}@vision.ee.ethz.ch

https://github.com/cszn/USRNet

Abstract

Learning-based single image super-resolution (SISR) methods are continuously showing superior effectiveness and efficiency over traditional model-based methods, largely due to the end-to-end training. However, different from model-based methods that can handle the SISR problem with different scale factors, blur kernels and noise levels under a unified MAP (maximum a posteriori) framework, learning-based methods generally lack such flexibility. To address this issue, this paper proposes an end-to-end trainable unfolding network which leverages both learning-based methods and model-based methods. Specifically, by unfolding the MAP inference via a half-quadratic splitting algorithm, a fixed number of iterations consisting of alternately solving a data subproblem and a prior subproblem can be obtained. The two subproblems then can be solved with neural modules, resulting in an end-to-end trainable, iterative network. As a result, the proposed network inherits the flexibility of model-based methods to super-resolve blurry, noisy images for different scale factors via a single model, while maintaining the advantages of learning-based methods. Extensive experiments demonstrate the superiority of the proposed deep unfolding network in terms of flexibility, effectiveness and also generalizability.

1. Introduction

Single image super-resolution (SISR) refers to the process of recovering the natural and sharp detailed high-resolution (HR) counterpart from a low-resolution (LR) image. It is one of the classical ill-posed inverse problems in low-level computer vision and has a wide range of real-world applications, such as enhancing the image visual quality on high-definition displays [42, 53] and improving the performance of other high-level vision tasks [13].

Despite decades of studies, SISR still requires further study for academic and industrial purposes [35, 64]. The difficulty is mainly caused by the inconsistency between the
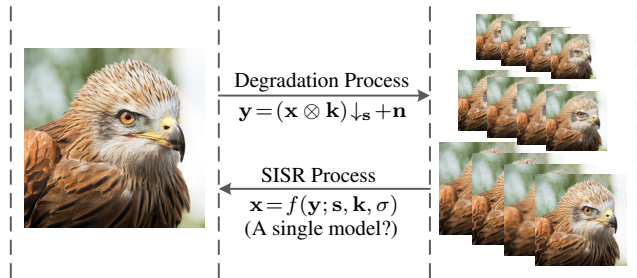

Figure 1. While a single degradation model (*i.e.*, Eq. (1)) can result in various LR images for an HR image, with different blur kernels, scale factors and noise, the study of learning a single deep model to invert all such LR images to HR image is still lacking.

simplistic degradation assumption of existing SISR methods and the complex degradations of real images [16]. Actually, for a scale factor of $\mathbf{s}$, the classical (traditional) degradation model of SISR [17, 18, 37] assumes the LR image $\mathbf{y}$ is a blurred, decimated, and noisy version of an HR image $\mathbf{x}$. Mathematically, it can be expressed by

$$\mathbf{y} = (\mathbf{x} \otimes \mathbf{k}) \downarrow_{\mathbf{s}} + \mathbf{n}, \tag{1}$$

where $\otimes$ represents two-dimensional convolution of $\mathbf{x}$ with blur kernel $\mathbf{k}$, $\downarrow_{\mathbf{s}}$ denotes the standard $\mathbf{s}$-fold downsampler, *i.e.*, keeping the upper-left pixel for each distinct $\mathbf{s} \times \mathbf{s}$ patch and discarding the others, and $\mathbf{n}$ is usually assumed to be additive, white Gaussian noise (AWGN) specified by standard deviation (or noise level) σ [71]. With a clear physical meaning, Eq. (1) can approximate a variety of LR images by setting proper blur kernels, scale factors and noises for an underlying HR images. In particular, Eq. (1) has been extensively studied in model-based methods which solve a combination of a data term and a prior term under the MAP framework.

Though model-based methods are usually algorithmically interpretable, they typically lack a standard criterion for their evaluation because, apart from the scale factor, Eq. (1) additionally involves a blur kernel and added noise. For convenience, researchers resort to bicubic degradation without consideration of blur kernel and noise level [14,

56, 60]. However, bicubic degradation is mathematically complicated [25], which in turn hinders the development of model-based methods. For this reason, recently proposed SISR solutions are dominated by learning-based methods that learn a mapping function from a bicubicly downsampled LR image to its HR estimation. Indeed, significant progress on improving PSNR [26, 70] and perceptual quality [31, 47, 58] for the bicubic degradation has been achieved by learning-based methods, among which convolutional neural network (CNN) based methods are the most popular, due to their powerful learning capacity and the speed of parallel computing. Nevertheless, little work has been done on applying CNNs to tackle Eq. (1) via a single model. Unlike model-based methods, CNNs usually lack flexibility to super-resolve blurry, noisy LR images for different scale factors via a single end-to-end trained model (see Fig. 1).

In this paper, we propose a deep unfolding super-resolution network (USRNet) to bridge the gap between learning-based methods and model-based methods. On one hand, similar to model-based methods, USRNet can effectively handle the classical degradation model (*i.e.*, Eq. (1)) with different blur kernels, scale factors and noise levels via a single model. On the other hand, similar to learning-based methods, USRNet can be trained in an end-to-end fashion to guarantee effectiveness and efficiency. To achieve this, we first unfold the model-based energy function via a half-quadratic splitting algorithm. Correspondingly, we can obtain an inference which iteratively alternates between solving two subproblems, one related to a data term and the other to a prior term. We then treat the inference as a deep network, by replacing the solutions to the two subproblems with neural modules. Since the two subproblems correspond respectively to enforcing degradation consistency knowledge and guaranteeing denoiser prior knowledge, USRNet is well-principled with explicit degradation and prior constraints, which is a distinctive advantage over existing learning-based SISR methods. It is worth noting that since USRNet involves a hyper-parameter for each subproblem, the network contains an additional module for hyper-parameter generation. Moreover, in order to reduce the number of parameters, all the prior modules share the same architecture and same parameters.

The main contributions of this work are as follows:

1) An end-to-end trainable unfolding super-resolution network (USRNet) is proposed. USRNet is the first attempt to handle the classical degradation model with different scale factors, blur kernels and noise levels via a single end-to-end trained model.

2) USRNet integrates the flexibility of model-based methods and the advantages of learning-based methods, providing an avenue to bridge the gap between model-based and learning-based methods.

3) USRNet intrinsically imposes a degradation constraint (*i.e.*, the estimated HR image should accord with the degradation process) and a prior constraint (*i.e.*, the estimated HR image should have natural characteristics) on the solution.

4) USRNet performs favorably on LR images with different degradation settings, showing great potential for practical applications.

2. Related work

2.1. Degradation models

Knowledge of the degradation model is crucial for the success of SISR [16, 59] because it defines how the LR image is degraded from an HR image. Apart from the classical degradation model and bicubic degradation model, several others have also been proposed in the SISR literature.

In some early works, the degradation model assumes the LR image is directly downsampled from the HR image without blurring, which corresponds to the problem of image interpolation [8]. In [34, 52], the bicubicly downsampled image is further assumed to be corrupted by Gaussian noise or JPEG compression noise. In [15, 42], the degradation model focuses on Gaussian blurring and a subsequent downsampling with scale factor 3. Note that, different from Eq. (1), their downsampling keeps the center rather than upper-left pixel for each distinct 3×3 patch. In [67], the degradation model assumes the LR image is the blurred, bicubicly downsampled HR image with some Gaussian noise. By assuming the bicubicly downsampled clean HR image is also clean, [68] treats the degradation model as a composition of deblurring on the LR image and SISR with bicubic degradation.

While many degradation models have been proposed, CNN-based SISR for the classical degradation model has received little attention and deserves further study.

2.2. Flexible SISR methods

Although CNN-based SISR methods have achieved impressive success to handle bicubic degradation, applying them to deal with other more practical degradation models is not straightforward. For the sake of practicability, it is preferable to design a flexible super-resolver that takes the three key factors, *i.e.*, scale factor, blur kernel and noise level, into consideration.

Several methods have been proposed to tackle bicubic degradation with different scale factors via a single model, such as LapSR [30] with progressive upsampling, MDSR [36] with scales-specific branches, Meta-SR [23] with meta-upscale module. To flexibly deal with a blurry LR image, the methods proposed in [44, 67] take the PCA dimension reduced blur kernel as input. However, these methods are limited to Gaussian blur kernels. Perhaps the

most flexible CNN-based works which can handle various blur kernels, scale factors and noise levels, are the deep plug-and-play methods [65, 68]. The main idea of such methods is to plug the learned CNN prior into the iterative solution under the MAP framework. Unfortunately, these are essentially model-based methods which suffer from a high computational burden and they involve manually selected hyper-parameters. How to design an end-to-end trainable model so that better results can be achieved with fewer iterations remains uninvestigated.

While learning-based blind image restoration has recently received considerable attention [12, 39, 43, 50, 62], we note that this work focuses on non-blind SISR which assumes the LR image, blur kernel and noise level are known beforehand. In fact, non-blind SISR is still an active research direction. First, the blur kernel and noise level can be estimated, or are known based on other information (*e.g.*, camera setting). Second, users can control the preference of sharpness and smoothness by tuning the blur kernel and noise level. Third, non-blind SISR can be an intermediate step towards solving blind SISR.

2.3. Deep unfolding image restoration

Apart from the deep plug-and-play methods (see, *e.g.*, [7, 10, 22, 57]), deep unfolding methods can also integrate model-based methods and learning-based methods. Their main difference is that the latter optimize the parameters in an end-to-end manner by minimizing the loss function over a large training set, and thus generally produce better results even with fewer iterations. The early deep unfolding methods can be traced back to [4, 48, 54] where a compact MAP inference based on gradient descent algorithm is proposed for image denoising. Since then, a flurry of deep unfolding methods based on certain optimization algorithms (*e.g.*, half-quadratic splitting [2], alternating direction method of multipliers [6] and primal-dual [1, 9]) have been proposed to solve different image restoration tasks, such as image denoising [11, 32], image deblurring [29, 49], image compressive sensing [61, 63], and image demosaicking [28].

Compared to plain learning-based methods, deep unfolding methods are interpretable and can fuse the degradation constraint into the learning model. However, most of them suffer from one or several of the following drawbacks. (i) The solution of the prior subproblem without using a deep CNN is not powerful enough for good performance. (ii) The data subproblem is not solved by a closed-form solution, which may hinder convergence. (iii) The whole inference is trained via a stage-wise and fine-tuning manner rather than a complete end-to-end manner. Furthermore, given that there exists no deep unfolding SISR method to handle the classical degradation model, it is of particular interest to propose such a method that overcomes the above mentioned drawbacks.

3. Method

3.1. Degradation model: classical vs. bicubic

Since bicubic degradation is well-studied, it is interesting to investigate its relationship to the classical degradation model. Actually, the bicubic degradation can be approximated by setting a proper blur kernel in Eq. (1). To achieve this, we adopt the data-driven method to solve the following kernel estimation problem by minimizing the reconstruction error over a large HR/bicubic-LR pairs $\{(\mathbf{x}, \mathbf{y})\}$

$$\mathbf{k}_{bicubic}^{\times \mathbf{s}} = \arg \min {}_{\mathbf{k}} \|(\mathbf{x} \otimes \mathbf{k})\downarrow_{\mathbf{s}} -\mathbf{y}\|. \qquad (2)$$

Fig. 2 shows the approximated bicubic kernels for scale factors 2, 3 and 4. It should be noted that since the downsamlping operation selects the upper-left pixel for each distinct $\mathbf{s} \times \mathbf{s}$ patch, the bicubic kernels for scale factors 2, 3 and 4 have a center shift of 0.5, 1 and 1.5 pixels to the upper-left direction, respectively.

(a) $\mathbf{k}_{bicubic}^{\times 2}$ (b) $\mathbf{k}_{bicubic}^{\times 3}$ (c) $\mathbf{k}_{bicubic}^{\times 4}$
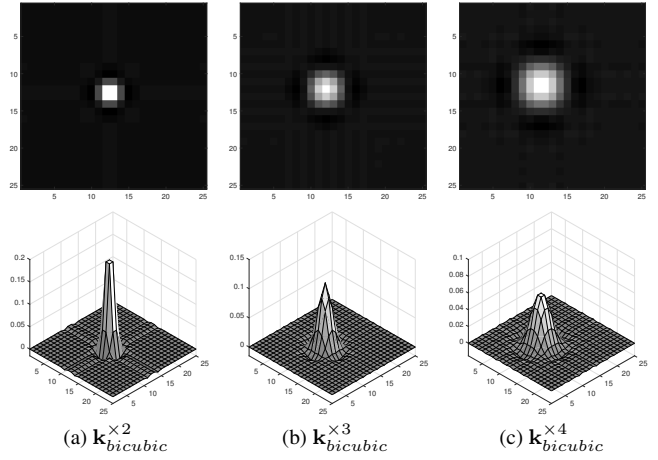
Figure 2. Approximated bicubic kernels for scale factors 2, 3 and 4 under the classical SISR degradation model assumption. Note that these kernels contain negative values.

3.2. Unfolding optimization

According to the MAP framework, the HR image could be estimated by minimizing the following energy function

$$E(\mathbf{x}) = \frac{1}{2\sigma^2}\|\mathbf{y} - (\mathbf{x} \otimes \mathbf{k})\downarrow_{\mathbf{s}}\|^2 + \lambda \Phi(\mathbf{x}), \qquad (3)$$

where $\frac{1}{2\sigma^2}\|\mathbf{y} - (\mathbf{x} \otimes \mathbf{k})\downarrow_{\mathbf{s}})\|^2$ is the data term, $\Phi(\mathbf{x})$ is the prior term, and λ is a trade-off parameter. In order to obtain an unfolding inference for Eq. (3), the half-quadratic splitting (HQS) algorithm is selected due to its simplicity and fast convergence in many applications. HQS tackles Eq. (3) by introducing an auxiliary variable $\mathbf{z}$, leading to the following approximate equivalence

$$E_\mu(\mathbf{x}, \mathbf{z}) = \frac{1}{2\sigma^2}\|\mathbf{y} - (\mathbf{z} \otimes \mathbf{k})\downarrow_{\mathbf{s}}\|^2 + \lambda \Phi(\mathbf{x}) + \frac{\mu}{2}\|\mathbf{z} - \mathbf{x}\|^2, \qquad (4)$$

where μ is the penalty parameter. Such problem can be addressed by iteratively solving subproblems for $\mathbf{x}$ and $\mathbf{z}$

$$\begin{cases} \mathbf{z}_k = \arg\min_{\mathbf{z}} \|\mathbf{y} - (\mathbf{z} \otimes \mathbf{k})\!\downarrow_{\mathbf{s}}\|^2 + \mu\sigma^2\|\mathbf{z} - \mathbf{x}_{k-1}\|^2, & (5) \\ \mathbf{x}_k = \arg\min_{\mathbf{x}} \frac{\mu}{2}\|\mathbf{z}_k - \mathbf{x}\|^2 + \lambda\Phi(\mathbf{x}). & (6) \end{cases}$$

According to Eq. (5), μ should be large enough so that $\mathbf{x}$ and $\mathbf{z}$ are approximately equal to the fixed point. However, this would also result in slow convergence. Therefore, a good rule of thumb is to iteratively increase μ. For convenience, the μ in the k-th iteration is denoted by μ_k.

It can be observed that the data term and the prior term are decoupled into Eq. (5) and Eq. (6), respectively. For the solution of Eq. (5), the fast Fourier transform (FFT) can be utilized by assuming the convolution is carried out with circular boundary conditions. Notably, it has a closed-form expression [71]

$$\mathbf{z}_k = \mathcal{F}^{-1}\left(\frac{1}{\alpha_k}\left(\mathbf{d} - \overline{\mathcal{F}(\mathbf{k})} \odot_{\mathbf{s}} \frac{(\mathcal{F}(\mathbf{k})\mathbf{d})\!\Downarrow_{\mathbf{s}}}{(\overline{\mathcal{F}(\mathbf{k})}\mathcal{F}(\mathbf{k}))\!\Downarrow_{\mathbf{s}} + \alpha_k}\right)\right)$$
$$(7)$$

where $\mathbf{d}$ is defined as

$$\mathbf{d} = \overline{\mathcal{F}(\mathbf{k})}\mathcal{F}(\mathbf{y}\uparrow_{\mathbf{s}}) + \alpha_k \mathcal{F}(\mathbf{x}_{k-1})$$

with $\alpha_k \triangleq \mu_k\sigma^2$ and where the $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ denote FFT and inverse FFT, $\overline{\mathcal{F}(\cdot)}$ denotes complex conjugate of $\mathcal{F}(\cdot)$, $\odot_{\mathbf{s}}$ denotes the distinct block processing operator with element-wise multiplication, *i.e.*, applying element-wise multiplication to the $\mathbf{s} \times \mathbf{s}$ distinct blocks of $\overline{\mathcal{F}(\mathbf{k})}$, $\Downarrow_{\mathbf{s}}$ denotes the distinct block downsampler, *i.e.*, averaging the $\mathbf{s} \times \mathbf{s}$ distinct blocks, $\uparrow_{\mathbf{s}}$ denotes the standard $\mathbf{s}$-fold upsampler, *i.e.*, upsampling the spatial size by filling the new entries with zeros. It is especially noteworthy that Eq. (7) also works for the special case of deblurring when $\mathbf{s} = 1$. For the solution of Eq. (6), it is known that, from a Bayesian perspective, it actually corresponds to a denoising problem with noise level $\beta_k \triangleq \sqrt{\lambda/\mu_k}$ [10].

3.3. Deep unfolding network

Once the unfolding optimization is determined, the next step is to design the unfolding super-resolution network (USRNet). Because the unfolding optimization mainly consists of iteratively solving a data subproblem (*i.e.*, Eq. (5)) and a prior subproblem (*i.e.*, Eq. (6)), USRNet should alternate between a data module $\mathcal{D}$ and a prior module $\mathcal{P}$. In addition, as the solutions of the subproblems also take the hyper-parameters α_k and β_k as input, respectively, a hyper-parameter module $\mathcal{H}$ is further introduced into US-RNet. Fig. 3 illustrates the overall architecture of USRNet with K iterations, where K is empirically set to 8 for the speed-accuracy trade-off. Next, more details on $\mathcal{D}$, $\mathcal{P}$ and $\mathcal{H}$ are provided.

Data module $\mathcal{D}$ The data module plays the role of Eq. (7) which is the closed-form solution of the data subproblem. Intuitively, it aims to find a clearer HR image which minimizes a weighted combination of the data term $\|\mathbf{y} - (\mathbf{z} \otimes \mathbf{k})\!\downarrow_{\mathbf{s}}\|^2$ and the quadratic regularization term $\|\mathbf{z} - \mathbf{x}_{k-1}\|^2$ with trade-off hyper-parameter α_k. Because the data term corresponds to the degradation model, the data module thus not only has the advantage of taking the scale factor $\mathbf{s}$ and blur kernel $\mathbf{k}$ as input but also imposes a degradation constraint on the solution. Actually, it is difficult to manually design such a simple but useful multiple-input module. For brevity, Eq. (7) is rewritten as

$$\mathbf{z}_k = \mathcal{D}(\mathbf{x}_{k-1}, \mathbf{s}, \mathbf{k}, \mathbf{y}, \alpha_k). \qquad (8)$$

Note that $\mathbf{x}_0$ is initialized by interpolating $\mathbf{y}$ with scale factor $\mathbf{s}$ via the simplest nearest neighbor interpolation. It should be noted that Eq. (8) contains no trainable parameters, which in turn results in better generalizability due to the complete decoupling between data term and prior term. For the implementation, we use PyTorch where the main FFT and inverse FFT operators can be implemented by `torch.rfft` and `torch.irfft`, respectively.

Prior module $\mathcal{P}$ The prior module aims to obtain a cleaner HR image $\mathbf{x}_k$ by passing $\mathbf{z}_k$ through a denoiser with noise level β_k. Inspired by [66], we propose a deep CNN denoiser that takes the noise level as input

$$\mathbf{x}_k = \mathcal{P}(\mathbf{z}_k, \beta_k). \qquad (9)$$

The proposed denoiser, namely ResUNet, integrates residual blocks [21] into U-Net [45]. U-Net is widely used for image-to-image mapping, while ResNet owes its popularity to fast training and its large capacity with many residual blocks. ResUNet takes the concatenated $\mathbf{z}_k$ and noise level map as input and outputs the denoised image $\mathbf{x}_k$. By doing so, ResUNet can handle various noise levels via a single model, which significantly reduces the total number of parameters. Following the common setting of U-Net, ResUNet involves four scales, each of which has an identity skip connection between downscaling and upscaling operations. Specifically, the number of channels in each layer from the first scale to the fourth scale are set to 64, 128, 256 and 512, respectively. For the downscaling and upscaling operations, 2×2 strided convolution (SConv) and 2×2 transposed convolution (TConv) are adopted, respectively. Note that no activation function is followed by SConv and TConv layers, as well as the first and the last convolutional layers. For the sake of inheriting the merits of ResNet, a group of 2 residual blocks are adopted in the downscaling and upscaling of each scale. As suggested in [36], each residual block is composed of two 3×3 convolution layers with ReLU activation in the middle and an identity skip connection summed to its output.
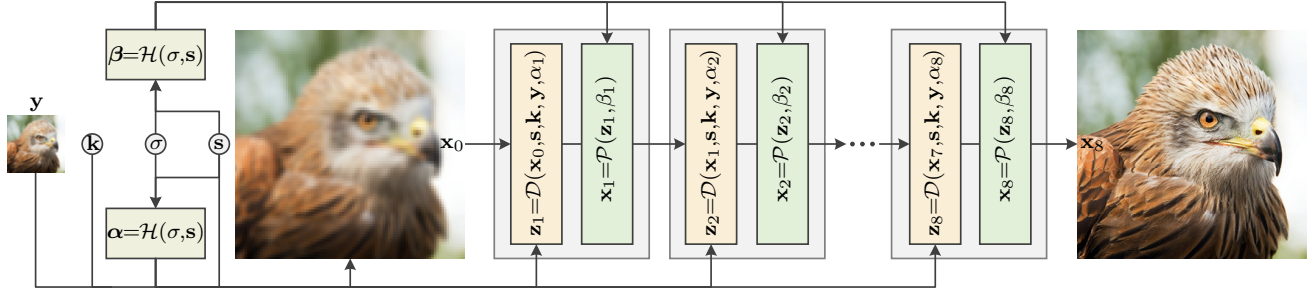
Figure 3. The overall architecture of the proposed USRNet with $K = 8$ iterations. USRNet can flexibly handle the classical degradation (*i.e.*, Eq. (1)) via a single model as it takes the LR image $\mathbf{y}$, scale factor $\mathbf{s}$, blur kernel $\mathbf{k}$ and noise level σ as input. Specifically, USRNet consists of three main modules, including the data module $\mathcal{D}$ that makes HR estimation clearer, the prior module $\mathcal{P}$ that makes HR estimation cleaner, and the hyper-parameter module $\mathcal{H}$ that controls the outputs of $\mathcal{D}$ and $\mathcal{P}$.

Hyper-parameter module $\mathcal{H}$ The hyper-parameter module acts as a 'slide bar' to control the outputs of the data module and prior module. For example, the solution $\mathbf{z}_k$ would gradually approach $\mathbf{x}_{k-1}$ as α_k increases. According to the definition of α_k and β_k, α_k is determined by σ and μ_k, while β_k depends on λ and μ_k. Although it is possible to learn a fixed λ and μ_k, we argue that a performance gain can be obtained if λ and μ_k vary with two key elements, *i.e.*, scale factor $\mathbf{s}$ and noise level σ, that influence the degree of ill-posedness. Let $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_K]$ and $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_K]$, we use a single module to predict $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$

$$[\boldsymbol{\alpha}, \boldsymbol{\beta}] = \mathcal{H}(\sigma, \mathbf{s}). \tag{10}$$

The hyper-parameter module consists of three fully connected layers with ReLU as the first two activation functions and Softplus [19] as the last. The number of hidden nodes in each layer is 64. Considering the fact that α_k and β_k should be positive, and Eq. (7) should avoid division by extremely small α_k, the output Softplus layer is followed by an extra addition of `1e-6`. We will show how the scale factor and noise level affect the hyper-parameters in Sec. 4.4.

3.4. End-to-end training

The end-to-end training aims to learn the trainable parameters of USRNet by minimizing a loss function over a large training data set. Thus, this section mainly describe the training data, loss function and training settings. Following [58], we use DIV2K [3] and Flickr2K [55] as the HR training dataset. The LR images are synthesized via Eq. (1). Although USRNet focuses on SISR, it is also applicable to the case of deblurring with $\mathbf{s} = 1$. Hence, the scale factors are chosen from $\{1, 2, 3, 4\}$. However, due to limited space, this paper does not consider the deblurring experiments. For the blur kernels, we use anisotropic Gaussian kernels as in [44, 51, 67] and motion kernels as in [5]. We fix the kernel size to 25×25. For the noise level, we set its range to $[0, 25]$.

With regard to the loss function, we adopt the L1 loss for PSNR performance. Following [58], once the model is obtained, we further adopt a weighted combination of L1 loss, VGG perceptual loss and relativistic adversarial loss [24] with weights 1, 1 and 0.005 for perceptual quality performance. We refer to such fine-tuned model as USRGAN. As usual, USRGAN only considers scale factor 4. We do not use additional losses to constrain the intermediate outputs since the above losses work well. One possible reason is that the prior module shares parameters across iterations.

To optimize the parameters of USRNet, we adopt the Adam solver [27] with mini-batch size 128. The learning rate starts from 1×10^{-4} and decays by a factor of 0.5 every 4×10^4 iterations and finally ends with 3×10^{-6}. It is worth pointing out that due to the infeasibility of parallel computing for different scale factors, each min-batch only involves one random scale factor. For USRGAN, its learning rate is fixed to 1×10^{-5}. The patch size of the HR image for both USRNet and USRGAN is set to 96×96. We train the models with PyTorch on 4 Nvidia Tesla V100 GPUs in Amazon AWS cloud. It takes about two days to obtain the USRNet model.

4. Experiments

We choose the widely-used color BSD68 dataset [40, 46] to quantitatively evaluate different methods. The dataset consists of 68 images with tiny structures and fine textures and thus is challenging to improve the quantitative metrics, such as PSNR. For the sake of synthesizing the corresponding testing LR images via Eq. (1), blur kernels and noise levels should be provided. Generally, it would be helpful to employ a large variety of blur kernels and noise levels for a thorough evaluation, however, it would also give rise to burdensome evaluation process. For this reason, as shown in Table 1, we only consider 12 representative and diverse blur kernels, including 4 isotropic Gaussian kernels with different widths (*i.e.*, 0.7, 1.2, 1.6 and 2.0), 4 anisotropic

Table 1. Average PSNR(dB) results of different methods for different combinations of scale factors, blur kernels and noise levels. The best two results are highlighted in red and blue colors, respectively.

Method	Scale Factor	Noise Level	Blur Kernel											
RCAN [70]	×2	0	29.48	26.76	25.31	24.37	24.38	24.10	24.25	23.63	20.31	20.45	20.57	22.04
	×3	0	24.93	27.30	25.79	24.61	24.57	24.38	24.55	23.74	20.15	20.25	20.39	21.68
	×4	0	22.68	25.31	25.59	24.63	24.37	24.23	24.43	23.74	20.06	20.05	20.33	21.47
ZSSR [51]	×2	0	29.44	29.48	28.57	27.42	27.15	26.81	27.09	26.25	14.22	14.22	16.02	19.39
	×3	0	25.13	25.80	25.94	25.77	25.61	25.23	25.68	25.41	16.37	15.95	17.35	20.45
	×4	0	23.50	24.33	24.56	24.65	24.52	24.20	24.56	24.55	16.94	16.43	18.01	20.68
IKC [20]	×4	0	22.69	25.26	25.63	25.21	24.71	24.20	24.39	24.77	20.05	20.03	20.35	21.58
IRCNN [65]	×2	0	29.60	30.16	29.50	28.37	28.07	27.95	28.21	27.19	28.58	26.79	29.02	28.96
	×3	0	25.97	26.89	27.07	27.01	26.83	26.76	26.88	26.67	26.22	25.59	26.14	26.05
	×3	2.55	25.70	26.13	25.72	25.33	25.28	25.18	25.34	24.97	25.00	24.64	24.90	24.73
	×3	7.65	24.58	24.68	24.59	24.39	24.24	24.20	24.27	24.02	23.94	23.77	23.75	23.69
	×4	0	23.99	25.01	25.32	25.45	25.36	25.26	25.34	25.47	24.69	24.39	24.44	24.57
USRNet	×2	0	30.55	30.96	30.56	29.49	29.13	29.12	29.28	28.28	30.90	30.65	30.60	30.75
	×3	0	27.16	27.76	27.90	27.88	27.71	27.68	27.74	27.57	27.69	27.50	27.50	27.41
	×3	2.55	26.99	27.40	27.23	26.78	26.55	26.60	26.72	26.14	26.90	26.80	26.69	26.49
	×3	7.65	26.45	26.52	26.10	25.57	25.46	25.40	25.49	25.00	25.39	25.47	25.20	25.01
	×4	0	25.30	25.96	26.18	26.29	26.20	26.15	26.17	26.30	25.91	25.57	25.76	25.70

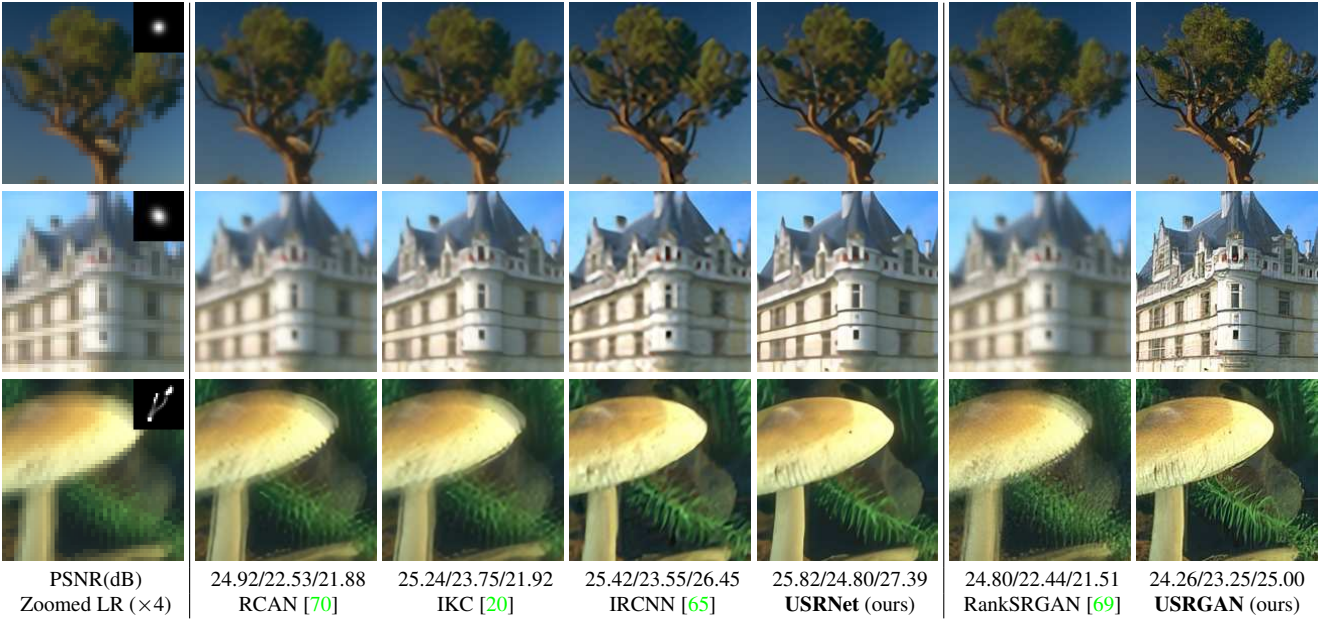| PSNR(dB) Zoomed LR (×4) | 24.92/22.53/21.88 RCAN [70] | 25.24/23.75/21.92 IKC [20] | 25.42/23.55/26.45 IRCNN [65] | 25.82/24.80/27.39 **USRNet** (ours) | 24.80/22.44/21.51 RankSRGAN [69] | 24.26/23.25/25.00 **USRGAN** (ours) |

Figure 4. Visual results of different methods on super-resolving noise-free LR image with scale factor 4. The blur kernel is shown on the upper-right corner of the LR image. Note that RankSRGAN and our USRGAN aim for perceptual quality rather than PSNR value.

Gaussian kernels from [67], and 4 motion blur kernels from [5, 33]. While it has been pointed out that anisotropic Gaussian kernels are enough for SISR task [44, 51], the SISR method that can handle more complex blur kernels would be a preferred choice in real applications. Therefore, it is necessary to further analyze the kernel robustness of different methods, we will thus separately report the PSNR results for each blur kernel rather than for each type of blur kernels. Although it has been pointed out that the proper blur kernel should vary with scale factor [64], we argue that the 12 blur kernels are diverse enough to cover a large kernel space. For the noise levels, we choose 2.55 (1%) and 7.65 (3%).

4.1. PSNR results

The average PSNR results of different methods for different degradation settings are reported in Table 1. The compared methods include RCAN [70], ZSSR [51], IKC [20] and IRCNN [65]. Specifically, RCAN is state-of-the-art PSNR oriented method for bicubic degradation; ZSSR is a non-blind zero-shot learning method with the ability to handle Eq. (1) for anisotropic Gaussian kernels; IKC is a blind iterative kernel correction method for isotropic Gaussian kernels; IRCNN a non-blind deep denoiser based plug-and-play method. For a fair comparison, we modified IRCNN to handle Eq. (1) by replacing its data
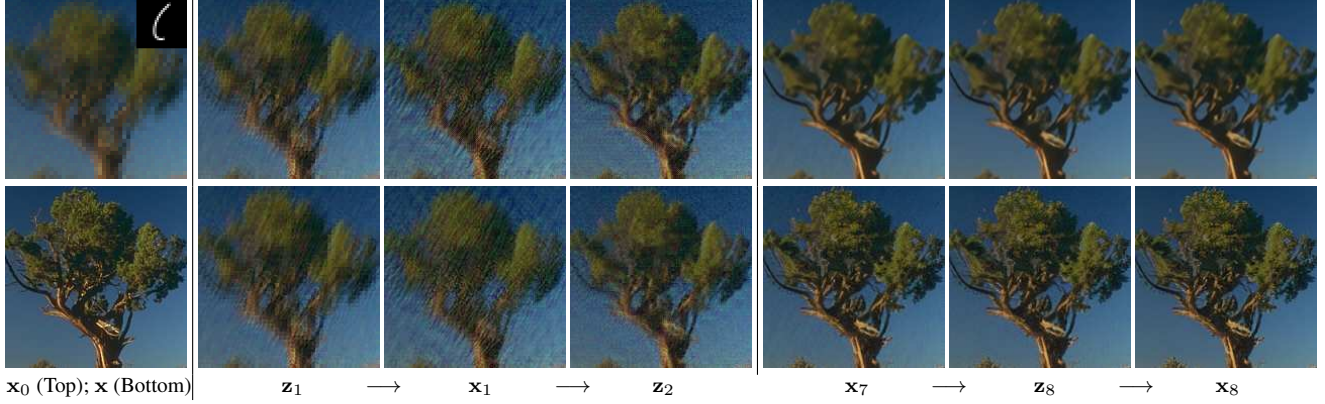
$\mathbf{x}_0$ (Top); $\mathbf{x}$ (Bottom) $\mathbf{z}_1 \longrightarrow \mathbf{x}_1 \longrightarrow \mathbf{z}_2$ $\mathbf{x}_7 \longrightarrow \mathbf{z}_8 \longrightarrow \mathbf{x}_8$

Figure 5. HR estimations in different iterations of USRNet (top row) and USRGAN (bottom row). The initial HR estimation $\mathbf{x}_0$ is the nearest neighbor interpolated version of LR image. The scale factor is 4, the noise level of LR image is 2.55 (1%), the blur kernel is shown on the upper-right corner of $\mathbf{x}_0$.

solution with Eq. (7). Note that following [37], we fix the pixel shift issue before calculating PSNR if necessary.

According to Table 1, we can have the following observations. First, our USRNet with a single model significantly outperforms the other competitive methods on different scale factors, blur kernels and noise levels. In particular, with much fewer iterations, USRNet has at least an average PSNR gain of 1dB over IRCNN with 30 iterations due to the end-to-end training. Second, RCAN can achieve good performance on the degradation setting similar to bicubic degradation but would deteriorate seriously when the degradation deviates from bicubic degradation. Such a phenomenon has been well studied in [16]. Third, ZSSR performs well on both isotropic and anisotropic Gaussian blur kernels for small scale factors but loses effectiveness on motion blur kernel and large scale factors. Actually, ZSSR has difficulty in capturing natural image characteristic on severely degraded image due to the single image learning strategy. Fourth, IKC does not generalize well to anisotropic Gaussian kernels and motion kernels.

Although USRNet is not designed for bicubic degradation, it is interesting to test its results by taking the approximated bicubic kernels in Fig. 2 as input. From Table 2, one can see that USRNet still performs favorably without training on the bicubic kernels.

Table 2. The average PSNR(dB) results of USRNet for bicubic degradation on commonly-used testing datasets.

Scale Factor	Set5	Set14	BSD100	Urban100
×2	37.72	33.49	32.10	31.79
×3	34.45	30.51	29.18	28.38
×4	32.42	28.83	27.69	26.44

4.2. Visual results

The visual results of different methods on super-resolving noise-free LR image with scale factor 4 are shown

in Fig. 4. Apart from RCAN, IKC and IRCNN, we also include RankSRGAN [69] for comparison with our USRGAN. Note that the visual results of ZSSR are omitted due to the inferior performance on scale factor 4. It can be observed from Fig. 4 that USRNet and IRCNN produce much better visual results than RCAN and IKC on the LR image with motion blur kernel. While USRNet can recover shaper edges than IRCNN, both of them fail to produce realistic textures. As expected, USRGAN can yield much better visually pleasant results than USRNet. On the other hand, RankSRGAN does not perform well if the degradation largely deviates from the bicubic degradation. In contrast, USRGAN is flexible to handle various LR images.

4.3. Analysis on $\mathcal{D}$ and $\mathcal{P}$

Because the proposed USRNet is an iterative method, it is interesting to investigate the HR estimations of data module $\mathcal{D}$ and prior module $\mathcal{P}$ in different iterations. Fig. 5 shows the results of USRNet and USRGAN in different iterations for an LR image with scale factor 4. As one can see, $\mathcal{D}$ and $\mathcal{P}$ can facilitate each other for iterative and alternating blur removal and detail recovery. Interestingly, $\mathcal{P}$ can also act as a detail enhancer for high-frequency recovery due to the task-specific training. In addition, it does not reduce blur kernel induced degradation which verifies the decoupling between $\mathcal{D}$ and $\mathcal{P}$. As a result, the end-to-end trained USRNet has a task-specific advantage over Gaussian denoiser based plug-and-play SISR. To quantitatively analyze the role of $\mathcal{D}$, we have trained an USRNet model with 5 iterations, it turns out that the average PSNR value will decreases about 0.1dB on Gaussian blur kernels and 0.3dB on motion blur kernels. This further indicates that $\mathcal{D}$ aims to eliminate blur kernel induced degradation. In addition, one can see that USRGAN has similar results with USRNet in the first few iterations, but will instead recover tiny structures and fine textures in last few iterations.

4.4. Analysis on $\mathcal{H}$

Fig. 6 shows outputs of the hyper-parameter module for different combinations of scale factor $\mathbf{s}$ and noise level σ. It can be observed from Fig. 6(a) that $\boldsymbol{\alpha}$ is positively correlated with σ and varies with $\mathbf{s}$. This actually accords with the definition of α_k in Sec. 3.2 and our analysis in Sec. 3.3. From Fig. 6(b), one can see that $\boldsymbol{\beta}$ has a decreasing tendency with the number of iterations and increases with scale factor and noise level. This implies that the noise level of HR estimation is gradually reduced across iterations and complex degradation requires a large β_k to tackle with the illposeness. It should be pointed out that the learned hyper-parameter setting is in accordance with that of IRCNN [65]. In summary, the learned $\mathcal{H}$ is meaningful as it plays the proper role.
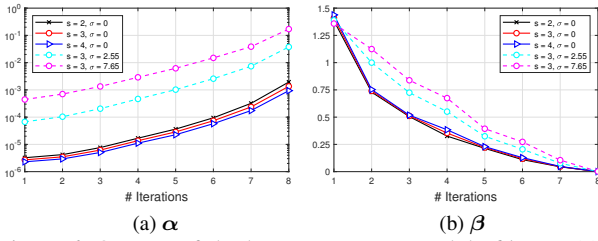

(a) $\boldsymbol{\alpha}$ (b) $\boldsymbol{\beta}$

Figure 6. Outputs of the hyper-parameter module $\mathcal{H}$, *i.e.*, (a) $\boldsymbol{\alpha}$ and (b) $\boldsymbol{\beta}$, with respect to different combinations of $\mathbf{s}$ and σ.

4.5. Generalizability

(a) Zoomed LR ($\times 3$)   (b) USRNet
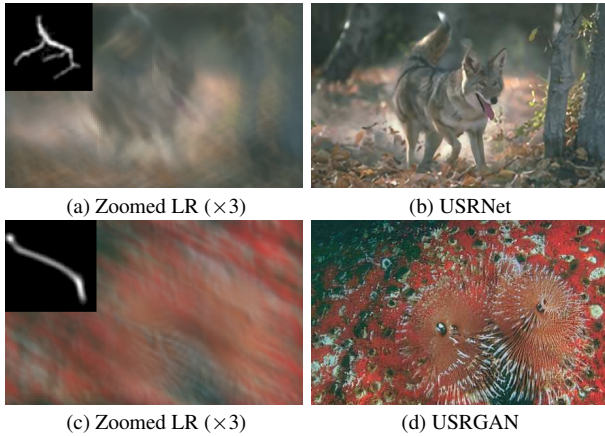
(c) Zoomed LR ($\times 3$)   (d) USRGAN

Figure 7. An illustration to show the generalizability of USRNet and USRGAN. The sizes of the kernels in (a) and (c) are 67×67 and 70×70, respectively. The two kernels are chosen from [41].

As mentioned earlier, the proposed method enjoys good generalizability due to the decoupling of data term and prior term. To demonstrate such an advantage, Fig. 7 shows the visual results of USRNet and USRGAN on LR image with a kernel of much larger size than training size of 25×25. It can be seen that both USRNet and USRGAN can produce visually pleasant results, which can be attributed to the trainable parameter-free data module. It is worth point-

ing out that USRGAN is trained on scale factor 4, while Fig. 7(b) shows its visual result on scale factor 3. This further indicates that the prior module of USRGAN can generalize to other scale factors. In summary, the proposed deep unfolding architecture has superiority in generalizability.

4.6. Real image super-resolution

Because Eq. (7) is based on the assumption of circular boundary condition, a proper boundary handling for the real LR image is generally required. We use the following three steps to do such pre-processing. First, the LR image is interpolated to the desired size. Second, the boundary handling method proposed in [38] is adopted on the interpolated image with the blur kernel. Last, the downsampled boundaries are padded to the original LR image. Fig. 8 shows the visual result of USRNet on real LR image with scale factor 4. The blur kernel is manually selected as isotropic Gaussian kernel with width 2.2 based on user preference. One can see from Fig. 8 that the proposed USRNet can reconstruct the HR image with improved visual quality.
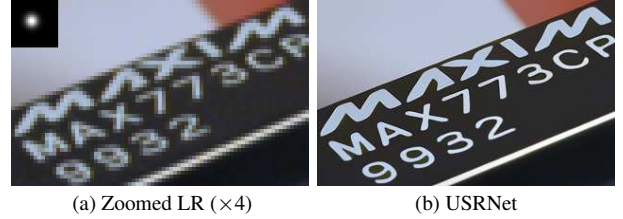

(a) Zoomed LR ($\times 4$)   (b) USRNet

Figure 8. Visual result of USRNet ($\times 4$) on a real LR image.

5. Conclusion

In this paper, we focus on the classical SISR degradation model and propose a deep unfolding super-resolution network. Inspired by the unfolding optimization of traditional model-based method, we design an end-to-end trainable deep network which integrates the flexibility of model-based methods and the advantages of learning-based methods. The main novelty of the proposed network is that it can handle the classical degradation model via a single model. Specifically, the proposed network consists of three interpretable modules, including the data module that makes HR estimation clearer, the prior module that makes HR estimation cleaner, and the hyper-parameter module that controls the outputs of the other two modules. As a result, the proposed method can impose both degradation constrain and prior constrain on the solution. Extensive experimental results demonstrated the flexibility, effectiveness and generalizability of the proposed method for super-resolving various degraded LR images. We believe that our work can benefit to image restoration research community.

References

[1] Jonas Adler and Ozan Öktem. Learned primal-dual reconstruction. *IEEE TMI*, 37(6):1322–1332, 2018. 3

[2] Manya V Afonso, José M Bioucas-Dias, and Mário AT Figueiredo. Fast image recovery using variable splitting and constrained optimization. *IEEE TIP*, 19(9):2345–2356, 2010. 3

[3] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *CVPRW*, volume 3, pages 126–135, July 2017. 5

[4] Adrian Barbu. Training an active random field for real-time image denoising. *IEEE TIP*, 18(11):2451–2462, 2009. 3

[5] Giacomo Boracchi and Alessandro Foi. Modeling the performance of image restoration from motion blur. *IEEE TIP*, 21(8):3502–3517, 2012. 5, 6

[6] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011. 3

[7] Alon Brifman, Yaniv Romano, and Michael Elad. Unified single-image and video super-resolution via denoising algorithms. *IEEE TIP*, 28(12):6063–6076, 2019. 3

[8] Vicent Caselles, J-M Morel, and Catalina Sbert. An axiomatic approach to image interpolation. *IEEE TIP*, 7(3):376–386, 1998. 2

[9] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011. 3

[10] Stanley H Chan, Xiran Wang, and Omar A Elgendy. Plug-and-Play ADMM for image restoration: Fixed-point convergence and applications. *IEEE Transactions on Computational Imaging*, 3(1):84–98, 2017. 3, 4

[11] Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE TPAMI*, 2016. 3

[12] Yu Chen, Ying Tai, Xiaoming Liu, Chunhua Shen, and Jian Yang. Fsrnet: End-to-end learning face super-resolution with facial priors. In *CVPR*, pages 2492–2501, 2018. 3

[13] Dengxin Dai, Yujian Wang, Yuhua Chen, and Luc Van Gool. Is image super-resolution helpful for other vision tasks? In *WACV*, pages 1–9, 2016. 1

[14] Chao Dong, C. C. Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE TPAMI*, 38(2):295–307, 2016. 1

[15] Weisheng Dong, Lei Zhang, Guangming Shi, and Xin Li. Nonlocally centralized sparse representation for image restoration. *IEEE TIP*, 22(4):1620–1630, 2013. 2

[16] Netalee Efrat, Daniel Glasner, Alexander Apartsin, Boaz Nadler, and Anat Levin. Accurate blur models vs. image priors in single image super-resolution. In *ICCV*, pages 2832–2839, 2013. 1, 2, 7

[17] Michael Elad and Arie Feuer. Restoration of a single super-resolution image from several blurred, noisy, and undersampled measured images. *IEEE TIP*, 6(12):1646–1658, 1997. 1

[18] Sina Farsiu, Dirk Robinson, Michael Elad, and Peyman Milanfar. Advances and challenges in super-resolution. *International Journal of Imaging Systems and Technology*, 14(2):47–57, 2004. 1

[19] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *ICAIS*, pages 315–323, 2011. 5

[20] Jinjin Gu, Hannan Lu, Wangmeng Zuo, and Chao Dong. Blind super-resolution with iterative kernel correction. In *CVPR*, pages 1604–1613, 2019. 6

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4

[22] Felix Heide, Steven Diamond, Matthias Niener, Jonathan Ragan-Kelley, Wolfgang Heidrich, and Gordon Wetzstein. Proximal: Efficient image optimization using proximal algorithms. *ACM TOG*, 35(4):84, 2016. 3

[23] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-SR: A magnification-arbitrary network for super-resolution. In *CVPR*, pages 1575–1584, 2019. 2

[24] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard GAN. *arXiv:1807.00734*, 2018. 5

[25] Robert Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6):1153–1160, 1981. 2

[26] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *CVPR*, pages 1646–1654, 2016. 2

[27] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[28] Filippos Kokkinos and Stamatios Lefkimmiatis. Deep image demosaicking using a cascade of convolutional residual denoising networks. In *ECCV*, pages 303–319, 2018. 3

[29] Jakob Kruse, Carsten Rother, and Uwe Schmidt. Learning to push the limits of efficient FFT-based image deconvolution. In *ICCV*, pages 4586–4594, 2017. 3

[30] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *CVPR*, pages 624–632, July 2017. 2

[31] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, pages 4681–4690, July 2017. 2

[32] Stamatios Lefkimmiatis. Non-local color image denoising with convolutional neural networks. In *CVPR*, pages 3587–3596, 2017. 3

[33] Anat Levin, Yair Weiss, Fredo Durand, and William T Freeman. Understanding and evaluating blind deconvolution algorithms. In *CVPR*, pages 1964–1971, 2009. 6

[34] Tao Li, Xiaohai He, Linbo Qing, Qizhi Teng, and Honggang Chen. An iterative framework of cascaded deblocking and superresolution for compressed images. *IEEE Transactions on Multimedia*, 20(6):1305–1320, 2017. 2

[35] Yawei Li, Shuhang Gu, Christoph Mayer, Luc Van Gool, and Radu Timofte. Group sparsity: The hinge between filter pruning and decomposition for network compression. In

CVPR, 2020. 1

[36] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *CVPRW*, pages 136–144, July 2017. 2, 4

[37] Ce Liu and Deqing Sun. On bayesian adaptive video super resolution. *IEEE TPAMI*, 36(2):346–360, 2013. 1, 7

[38] Renting Liu and Jiaya Jia. Reducing boundary artifacts in image deconvolution. In *ICIP*, pages 505–508, 2008. 8

[39] Andreas Lugmayr, Martin Danelljan, and Radu Timofte. Un-supervised learning for real-world super-resolution. In *IC-CVW*, pages 3408–3416, 2019. 3

[40] 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 *ICCV*, pages 416–423, 2001. 5

[41] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang. Blind image deblurring using dark channel prior. In *CVPR*, pages 1628–1636, 2016. 8

[42] Tomer Peleg and Michael Elad. A statistical prediction model based on sparse representations for single image super-resolution. *IEEE TIP*, 23(6):2569–2582, 2014. 1, 2

[43] Dongwei Ren, Kai Zhang, Qilong Wang, Qinghua Hu, and Wangmeng Zuo. Neural blind deconvolution using deep priors. In *CVPR*, pages 1628–1636, 2020. 3

[44] Gernot Riegler, Samuel Schulter, Matthias Ruther, and Horst Bischof. Conditioned regression models for non-blind single image super-resolution. In *ICCV*, pages 522–530, 2015. 2, 5, 6

[45] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 4

[46] Stefan Roth and Michael J Black. Fields of experts. *IJCV*, 82(2):205–229, 2009. 5

[47] Mehdi SM Sajjadi, Bernhard Schölkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *ICCV*, pages 4501–4510, 2017. 2

[48] Kegan GG Samuel and Marshall F Tappen. Learning optimized MAP estimates in continuously-valued MRF models. In *CVPR*, pages 477–484, 2009. 3

[49] Uwe Schmidt and Stefan Roth. Shrinkage fields for effective image restoration. In *CVPR*, pages 2774–2781, 2014. 3

[50] Ziyi Shen, Wei-Sheng Lai, Tingfa Xu, Jan Kautz, and Ming-Hsuan Yang. Deep semantic face deblurring. In *CVPR*, pages 8260–8269, 2018. 3

[51] Assaf Shocher, Nadav Cohen, and Michal Irani. "zero-shot" super-resolution using deep internal learning. In *ICCV*, pages 3118–3126, 2018. 5, 6

[52] Abhishek Singh, Fatih Porikli, and Narendra Ahuja. Super-resolving noisy images. In *CVPR*, pages 2846–2853, 2014. 2

[53] Wan-Chi Siu and Kwok-Wai Hung. Review of image inter-polation and super-resolution. In *The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, pages 1–10. IEEE, 2012. 1

[54] Jian Sun and Marshall F Tappen. Learning non-local range markov random field for image restoration. In *CVPR*, pages 2745–2752, 2011. 3

[55] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *CVPRW*, pages 114–125, 2017. 5

[56] Radu Timofte, Vincent De Smet, and Luc Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *ACCV*, pages 111–126, 2014. 2

[57] Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg. Plug-and-play priors for model based reconstruction. In *IEEE Global Conference on Signal and Information Processing*, pages 945–948, 2013. 3

[58] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. ESRGAN: Enhanced super-resolution generative adversarial networks. In *The ECCV Workshops*, 2018. 2, 5

[59] Chih-Yuan Yang, Chao Ma, and Ming-Hsuan Yang. Single-image super-resolution: A benchmark. In *ECCV*, pages 372–386, 2014. 2

[60] Jianchao Yang, John Wright, Thomas Huang, and Yi Ma. Image super-resolution as sparse representation of raw image patches. In *CVPR*, pages 1–8, 2008. 2

[61] Yan Yang, Jian Sun, Huibin Li, and Zongben Xu. Deep ADMM-Net for compressive sensing MRI. In *NeurIPS*, pages 10–18, 2016. 3

[62] Rajeev Yasarla, Federico Perazzi, and Vishal M Patel. De-blurring face images using uncertainty guided multi-stream semantic networks. *arXiv:1907.13106*, 2019. 3

[63] Jian Zhang and Bernard Ghanem. ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing. In *CVPR*, pages 1828–1837, 2018. 3

[64] Kai Zhang, Xiaoyu Zhou, Hongzhi Zhang, and Wangmeng Zuo. Revisiting single image super-resolution under internet environment: blur kernels and reconstruction algorithms. In *PCM*, pages 677–687, 2015. 1, 6

[65] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep CNN denoiser prior for image restoration. In *CVPR*, pages 3929–3938, July 2017. 3, 6, 8

[66] Kai Zhang, Wangmeng Zuo, and Lei Zhang. FFDNet: Toward a fast and flexible solution for CNN-based image denoising. *IEEE TIP*, 27(9):4608–4622, 2018. 4

[67] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *CVPR*, pages 3262–3271, 2018. 2, 5, 6

[68] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Deep plug-and-play super-resolution for arbitrary blur kernels. In *CVPR*, pages 1671–1681, 2019. 2, 3

[69] Wenlong Zhang, Yihao Liu, Chao Dong, and Yu Qiao. Ranksrgan: Generative adversarial networks with ranker for image super-resolution. In *ICCV*, pages 3096–3105, 2019. 6, 7

[70] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, pages 286–301, 2018. 2, 6

[71] Ningning Zhao, Qi Wei, Adrian Basarab, Nicolas Dobigeon, Denis Kouamé, and Jean-Yves Tourneret. Fast single image super-resolution using a new analytical solution for ℓ_2-ℓ_2 problems. *IEEE TIP*, 25(8):3683–3697, 2016. 1, 4