

Deep Plug-and-Play Super-Resolution for Arbitrary Blur Kernels

Kai Zhang^{1,2}, Wangmeng Zuo^{1,3,*}, Lei Zhang^{2,4}

¹School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China

²Dept. of Computing, The Hong Kong Polytechnic University, Hong Kong, China

³Peng Cheng Laboratory, Shenzhen, China

⁴DAMO Academy, Alibaba Group

cskaizhang@gmail.com, wzmzuo@hit.edu.cn, cslzhang@comp.polyu.edu.hk

<https://github.com/csxn/DPSR>

Abstract

While deep neural networks (DNN) based single image super-resolution (SISR) methods are rapidly gaining popularity, they are mainly designed for the widely-used bicubic degradation, and there still remains the fundamental challenge for them to super-resolve low-resolution (LR) image with arbitrary blur kernels. In the meanwhile, plug-and-play image restoration has been recognized with high flexibility due to its modular structure for easy plug-in of denoiser priors. In this paper, we propose a principled formulation and framework by extending bicubic degradation based deep SISR with the help of plug-and-play framework to handle LR images with arbitrary blur kernels. Specifically, we design a new SISR degradation model so as to take advantage of existing blind deblurring methods for blur kernel estimation. To optimize the new degradation induced energy function, we then derive a plug-and-play algorithm via variable splitting technique, which allows us to plug any super-resolver prior rather than the denoiser prior as a modular part. Quantitative and qualitative evaluations on synthetic and real LR images demonstrate that the proposed deep plug-and-play super-resolution framework is flexible and effective to deal with blurry LR images.

1. Introduction

Single image super-resolution (SISR), with the goal of estimating the clean high-resolution (HR) counterpart \mathbf{x} of a given low-resolution (LR) image \mathbf{y} , is a classical problem with highly academic and practical values [3]. Basically, the relationship between LR and HR images is characterized by the degradation model which defines how the LR image is degraded from an HR image. Empirical and theoretical studies have proven that an accurate degradation model is crucial for the success of SISR [20, 60]. Thus, it is important to first review the degradation model for SISR.

In fact, most existing SISR methods are designed under the assumption of certain degradation model. There are two widely-used degradation models. The first one, which is acknowledged as a general degradation model for SISR, is given by

$$\mathbf{y} = (\mathbf{x} \otimes \mathbf{k}) \downarrow_s + \mathbf{n}, \quad (1)$$

where $\mathbf{x} \otimes \mathbf{k}$ represents the convolution between blur kernel \mathbf{k} and HR image \mathbf{x} , \downarrow_s is a subsequent downsampling operation with scale factor s , and \mathbf{n} is additive white Gaussian noise (AWGN) with noise level σ . Such degradation model has been extensively studied on developing model-based optimization methods [13, 19]. However, these methods mostly assume a priori known blur kernel which in practice is difficult to estimate. Although several works such as [41, 53] focus on estimating blur kernel, their source codes are unfortunately not publicly available.

The second and perhaps the most widely-used one which we refer to as bicubic degradation is formalized as

$$\mathbf{y} = \mathbf{x} \downarrow_s. \quad (2)$$

Here, unless otherwise specified, \downarrow_s instead represents the bicubic downsampler (Matlab default function `imresize`) with scale factor s . Due to its simplicity, bicubic degradation model has become the benchmark setting to evaluate SISR methods [29, 36]. In particular, it greatly facilitates the development of powerful deep neural networks (DNN) for SISR [2]. However, such a simple degradation inevitably gives rise to poor results in many practical scenarios [20, 65]. Nevertheless, little work has been done on extending to more realistic degradation models.

Given the above considerations, it is very necessary to address the following two issues: 1) designing an alternative degradation model, and 2) extending existing DNN-based methods for bicubic degradation to the new degradation model so as to exploit the power of DNN. To this end, we first propose a simple yet effective degradation model which assumes the LR image is a bicubically downsampled, blurred and noisy version of an HR image. Compared to the

*Corresponding author.

general degradation model given by Eqn. (1), the proposed one offers two favorable merits. First, it generalizes the well-studied bicubic degradation model given by Eqn. (2). Second, it enables us to adopt the available blind deblurring methods to estimate the blur kernel from the given LR image. In order to extend DNN-based SISR methods to the new degradation model, we propose a well-principled deep plug-and-play super-resolution (DPSR) framework which integrates DNN-based super-resolver into a variable splitting based iterative optimization scheme. It turns out that the distortion of blur can be effectively handled in Fourier domain. For this reason, it is possible to deal with arbitrary blur kernels, which is one of the main goals in this paper. Moreover, different from existing plug-and-play framework [13] which generally plugs the off-the-shelf Gaussian denoiser as a modular part, the proposed instead achieves the plug-in step by applying any existing DNN-based super-resolvers with only a small modification.

By far, it is worth emphasizing that we mainly focus on non-blind SISR for arbitrary uniform blur kernels rather than blind SISR for arbitrary non-uniform blur kernels. On the one hand, non-blind SISR is important to blind SISR which generally involves alternatively updating the blur kernel and applying non-blind SISR to update the super-resolved image. While some recent works attempt to train DNN to directly estimate the clean image for blind deblurring, their practicability needs further evaluations. On the other hand, although the non-uniform blur kernel tends to be a more realistic assumption, it is too complex and still remains to be a difficult problem for image deblurring [31]. In fact, the arbitrary uniform blur kernel assumption is already a much better choice than the simple bicubic kernel for practical applications. Briefly speaking, our work makes a valuable intermediate step from existing bicubic degradation based SISR to ultimate blind SISR.

The contribution of this work is summarized as follows:

- A more realistic degradation model than bicubic degradation model for SISR is proposed. It considers arbitrary blur kernels and enables to use existing deblurring methods for blur kernel estimation.
- A deep plug-and-play super-resolution framework is proposed to solve SISR with the new degradation model. DPSR is applicable beyond bicubic degradation and can handle LR images with arbitrary blur kernels.
- The proposed DPSR is well-principled as the iterative scheme aims to solve the new degradation induced energy function.
- The proposed DPSR extends existing plug-and-play framework, showing that the plug-and-play prior for SISR is not limited to Gaussian denoiser.
- The proposed DPSR performs favorably on LR images with arbitrary blur kernels, which makes a useful step towards practical SISR applications.

2. Related work

2.1. DNN-based SISR

1) Bicubic degradation. The first DNN-based SISR method, termed as SRCNN [17], employs a relatively shallow network and follows the previous SISR methods such as A+ [55] and ScSR [61] to synthesize LR images with bicubic interpolation. Since then, by fixing the degradation model to bicubic degradation, some researchers began to improve SISR performance via DNN from different aspects, including PSNR and SSIM values, efficiency, and perceptual visual quality at a large scale factor.

To improve the SISR performance in terms of PSNR and SSIM, the very deep super-resolution (VDSR) network proposed by Kim *et al.* [29] shows that the most direct way is to increase the network depth. However, VDSR operates on the bicubically interpolated LR images which hinders the efficiency. To that end, the FSRCNN [18] and ESPCN [50] are proposed to directly manipulate the LR input and adopt an upscaling operation at the end of the network. Considering the fact that the visual results tend to be oversmoothed at a large scale factor (*e.g.*, 4), VGG [52] loss and generative adversarial network (GAN) [24] loss are utilized to improve the perceptual visual quality in [34, 49, 58].

While achieving great success on bicubic degradation [36, 44, 68], these methods perform poorly on most of real images due to the mismatch of degradation models.

2) Beyond bicubic degradation. In [20], the authors pointed out that an accurate estimate of the blur kernel is more important than sophisticated image prior. Since then, several attempts have been made to tackle with LR images beyond bicubic degradation. Zhang *et al.* [63] proposed a plug-and-play framework (IRCNN) to solve the energy function induced by Eqn. (1). Although in theory IRCNN can handle arbitrary blur kernel (please refer to [13]), the blur kernel of such degradation model in practice is difficult to estimate. Zhang *et al.* [65] proposed a general DNN-based solution (SRMD) which takes two key degradation parameters as input. However, SRMD only considers the Gaussian blur kernels. In [51], the authors proposed a zero-shot super-resolution (ZSSR) method which trains image-specific DNN on the testing LR image and can also take the degradation parameters, such as the estimated blur kernel, to improve the performance. While showing impressive results for LR image with internal repetitive structures, ZSSR is less effective for severely blurred LR image.

As discussed, the above methods have two main drawbacks. First, they have difficulty in blur kernel estimation. Second, they are generally designed for Gaussian-like blur kernel and thus cannot effectively handle severely blurred LR image. It should be noted that a deep blind SISR method for motion blur is proposed in [66]. However, it has limited ability to handle the distortion of arbitrary blur kernels.

2.2. Plug-and-play image restoration

The plug-and-play image restoration which was first introduced in [15, 57, 69] has attracted significant attention-s due to its flexibility and effectiveness in handling various inverse problems. Its main idea is to unroll the energy function by variable splitting technique and replace the prior associated subproblem by any off-the-shelf Gaussian denoiser. Different from traditional image restoration methods which employ hand-crafted image priors, it can implicitly define the plug-and-play prior by the denoiser. Remarkably, the denoiser can be learned by DNN with large capability which would give rise to promising performance.

During the past few years, a flurry of plug-and-play works have been developed from the following aspects: 1) different variable splitting algorithms, such as half-quadratic splitting (HQS) algorithm [1], alternating direction method of multipliers (ADMM) algorithm [8], FISTA [4], and primal-dual algorithm [11, 42]; 2) different applications, such as Poisson denoising [47], demosaicking [26], deblurring [56], super-resolution [9, 13, 28, 63], and inpainting [40]; 3) different types of denoiser priors, such as BM3D [14, 21], DNN-based denoisers [6, 62] and their combinations [25]; and 4) theoretical analysis on the convergence from the aspect of fixed point [13, 37, 38] and Nash equilibrium [10, 16, 45].

To the best of our knowledge, existing plug-and-play image restoration methods mostly treat the Gaussian denoiser as the prior. We will show that, for the application of plug-and-play SISR, the prior is not limited to Gaussian denoiser. Instead, a simple super-resolver prior can be employed to solve a much more complex SISR problem.

3. Method

3.1. New degradation model

In order to ease the blur kernel estimation, we propose the following degradation model

$$\mathbf{y} = (\mathbf{x} \downarrow_s) \otimes \mathbf{k} + \mathbf{n}, \quad (3)$$

where \downarrow_s is the bicubic downsampler with scale factor s . Simply speaking, Eqn. (3) conveys that the LR image \mathbf{y} is a bicubically downsampled, blurred and noisy version of a clean HR image \mathbf{x} .

Since existing methods widely use bicubic downsampler to synthesize or augment LR image, it is a reasonable assumption that bicubically downsampled HR image (*i.e.*, $\mathbf{x} \downarrow_s$) is also a clean image. Following this assumption, Eqn. (3) actually corresponds to a deblurring problem followed by a SISR problem with bicubic degradation. Thus, we can fully employ existing well-studied deblurring methods to estimate \mathbf{k} . Clearly, this is a distinctive advantage over the degradation model given by Eqn. (1).

Once the degradation model is defined, the next step is to formulate the energy function. According to Maximum A Posteriori (MAP) probability, the energy function is formally given by

$$\min_{\mathbf{x}} \frac{1}{2\sigma^2} \|\mathbf{y} - (\mathbf{x} \downarrow_s) \otimes \mathbf{k}\|^2 + \lambda \Phi(\mathbf{x}), \quad (4)$$

where $\frac{1}{2\sigma^2} \|\mathbf{y} - (\mathbf{x} \downarrow_s) \otimes \mathbf{k}\|^2$ is the data fidelity (likelihood) term¹ determined by the degradation model of Eqn. (3), $\Phi(\mathbf{x})$ is the regularization (prior) term, and λ is the regularization parameter. For discriminative learning methods, their inference models actually correspond to an energy function where the degradation model is implicitly defined by the training LR and HR pairs. This explains why existing DNN-based SISR methods trained on bicubic degradation perform poorly for real images.

3.2. Deep plug-and-play SISR

To solve Eqn. (4), we first adopt the variable splitting technique to introduce an auxiliary variable \mathbf{z} , leading to the following equivalent constrained optimization formulation:

$$\begin{aligned} \hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{z} \otimes \mathbf{k}\|^2 + \lambda \Phi(\mathbf{x}), \\ \text{subject to } \mathbf{z} = \mathbf{x} \downarrow_s. \end{aligned} \quad (5)$$

We then address Eqn. (5) with half quadratic splitting (HQS) algorithm. Note that other algorithms such as ADMM can also be exploited. We use HQS for its simplicity.

Typically, HQS tackles with Eqn. (5) by minimizing the following problem which involves an additional quadratic penalty term

$$\mathcal{L}_\mu(\mathbf{x}, \mathbf{z}) = \frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{z} \otimes \mathbf{k}\|^2 + \lambda \Phi(\mathbf{x}) + \frac{\mu}{2} \|\mathbf{z} - \mathbf{x} \downarrow_s\|^2, \quad (6)$$

where μ is the penalty parameter, and a very large μ will enforce \mathbf{z} approximately equals to $\mathbf{x} \downarrow_s$. Usually, μ varies in a non-descending order during the following iterative solution to Eqn. (6)

$$\begin{cases} \mathbf{z}_{k+1} = \arg \min_{\mathbf{z}} \|\mathbf{y} - \mathbf{z} \otimes \mathbf{k}\|^2 + \mu \sigma^2 \|\mathbf{z} - \mathbf{x}_k \downarrow_s\|^2, & (7) \\ \mathbf{x}_{k+1} = \arg \min_{\mathbf{x}} \frac{\mu}{2} \|\mathbf{z}_{k+1} - \mathbf{x} \downarrow_s\|^2 + \lambda \Phi(\mathbf{x}). & (8) \end{cases}$$

It can be seen that Eqn. (7) and Eqn. (8) are alternating minimization problems with respect to \mathbf{z} and \mathbf{x} , respectively. In particular, by assuming the convolution is carried out with circular boundary conditions, Eqn. (7) has a fast closed-form solution

$$\mathbf{z}_{k+1} = \mathcal{F}^{-1} \left(\frac{\overline{\mathcal{F}(\mathbf{k})} \mathcal{F}(\mathbf{y}) + \mu \sigma^2 \mathcal{F}(\mathbf{x}_k \downarrow_s)}{\overline{\mathcal{F}(\mathbf{k})} \mathcal{F}(\mathbf{k}) + \mu \sigma^2} \right), \quad (9)$$

¹In order to facilitate and clarify the parameter setting, we emphasize that, from the Bayesian viewpoint, the data fidelity term should be $\frac{1}{2\sigma^2} \|\mathbf{y} - (\mathbf{x} \downarrow_s) \otimes \mathbf{k}\|^2$ rather than $\frac{1}{2} \|\mathbf{y} - (\mathbf{x} \downarrow_s) \otimes \mathbf{k}\|^2$.

where $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ denote the Fast Fourier Transform (FFT) and inverse FFT, $\overline{\mathcal{F}(\cdot)}$ denotes complex conjugate of $\mathcal{F}(\cdot)$.

To analyze Eqn. (8) from a Bayesian perspective, we rewrite it as follows

$$\mathbf{x}_{k+1} = \arg \min_{\mathbf{x}} \frac{1}{2(\sqrt{1/\mu})^2} \|\mathbf{z}_{k+1} - \mathbf{x}\downarrow_s\|^2 + \lambda\Phi(\mathbf{x}). \quad (10)$$

Clearly, Eqn. (10) corresponds to super-resolving \mathbf{z}_{k+1} with a scale factor s by assuming \mathbf{z}_{k+1} is bicubically downsampled from an HR image \mathbf{x} , and then corrupted by AWGN with noise level $\sqrt{1/\mu}$. From another viewpoint, Eqn. (10) solves a super-resolution problem with the following simple bicubic degradation model

$$\mathbf{y} = \mathbf{x}\downarrow_s + \mathbf{n}. \quad (11)$$

As a result, one can plug DNN-based super-resolver trained on the widely-used bicubic degradation with certain noise levels to replace Eqn. (10). For brevity, Eqn. (8) and Eqn. (10) can be further rewritten as

$$\mathbf{x}_{k+1} = \mathcal{SR}(\mathbf{z}_{k+1}, s, \sqrt{1/\mu}). \quad (12)$$

Since the prior term $\Phi(\mathbf{x})$ is implicitly defined in $\mathcal{SR}(\cdot)$, we refer to it as super-resolver prior.

So far, we have seen that the two sub-problems given by Eqn. (7) and Eqn. (8) are relatively easy to solve. In fact, they also have clear interpretation. On the one hand, since the blur kernel \mathbf{k} is only involved in the closed-form solution, Eqn. (7) addresses the distortion of blur. In other words, it pulls the current estimation to a less blurry one. On the other hand, Eqn. (8) maps the less blurry image to a more clean HR image. After several alternating iterations, it is expected that the final reconstructed HR image contains no blur and noise.

3.3. Deep super-resolver prior

In order to take advantage of the merits of DNN, we need to specify the super-resolver network which should take the noise level as input according to Eqn. (12). Inspired by [23, 64], we only need to modify most of the existing DNN-based super-resolvers by taking an additional noise level map as input. Alternatively, one can directly adopt SRMD as the super-resolver prior because its input already contains the noise level map.

Since SRResNet [34] is a well-known DNN-based super-resolver, in this paper we propose a modified SRResNet, namely SRResNet+, to plug in the proposed DPSR framework. SRResNet+ differs from SRResNet in several aspects. First, SRResNet+ additionally takes a noise level map \mathbf{M} as input. Second, SRResNet+ increases the number of feature maps from 64 to 96. Third, SRResNet+ removes the batch normalization layer [27] as suggested in [58].

Before training a separate SRResNet+ model for each scale factor, we need to synthesize the LR image and its noise level map from a given HR image. According to the degradation model given by Eqn. (11), the LR image is bicubically downsampled from an HR image, and then corrupted by AWGN with a noise level σ from predefined noise level range. For the corresponding noise level map, it has the same spatial size of LR image and all the elements are σ . Following [65], we set the noise level range to $[0, 50]$. For the HR images, we choose the 800 training images from DIV2K dataset [2].

We adopt Adam algorithm [30] to optimize SRResNet+ by minimizing the ℓ_1 loss function. The learning rate starts from 10^{-4} , then decreases by half every 5×10^5 iterations and finally ends once it is smaller than 10^{-7} . The mini-batch size is set to 16. The patch size of LR input is set to 48×48 . The rotation and flip based data augmentation is performed during training. We train the models with PyTorch on a single GTX 1080 Ti GPU.

Since this work mainly focuses on SISR with arbitrary blur kernels. We omit the comparison between SRResNet+ and other methods on bicubic degradation. As a simple comparison, SRResNet+ can outperform SRResNet [34] by an average PSNR gain of 0.15dB on Set5 [5].

3.4. Comparison with related methods

In this section, we emphasize the fundamental differences between the proposed DPSR and several closely related DNN-based methods.

1) Cascaded deblurring and SISR. To super-resolve LR image with arbitrary blur kernels, a heuristic method is to perform deblurring first and then super-resolve the deblurred LR image. However, such a cascaded two-step method suffers from the drawback that the perturbation error of the first step would be amplified at the second step. On the contrary, DPSR optimizes the energy function given by Eqn. (4) in an iterative manner. Thus, DPSR tends to deliver better performance.

2) Fine-tuned SISR model with more training data. Perhaps the most straightforward way is to fine-tune existing bicubic degradation based SISR models with more training data generated by the new degradation model (*i.e.*, Eqn. (3)), resulting in the so-called blind SISR. However, the performance of such methods deteriorates seriously especially when large complex blur kernels are considered, possibly because the distortion of blur would further aggravate the pixel-wise average problem [34]. As for DPSR, it takes the blur kernel as input and can effectively handle the distortion of blur via Eqn. (9).

3) Extended SRMD or DPSR with end-to-end training. Inspired by SRMD [65], one may attempt to extend it by considering arbitrary blur kernels. However, it is difficult

to sample enough blur kernels to cover the large kernel space. In addition, it would require a large amount of time to train a reliable model. By contrast, DPSR only needs to train the models on the bicubic degradation, thus it involves much less training time. Furthermore, while SRMD can effectively handle the simple Gaussian kernels of size 15×15 with many successive convolutional layers, it loses effectiveness to deal with large complex blur kernels. Instead, DPSR adopts a more concise and specialized modular by FFT via Eqn. (9) to eliminate the distortion of blur. Alternatively, one may take advantage of the structure benefits of DPSR and resort to jointly training DPSR in an end-to-end manner. However, we leave this to our future work.

From the above discussions, we can conclude that our DPSR is well-principled, structurally simple, highly interpretable and involves less training.

4. Experiments

4.1. Synthetic LR images

Following the common setting in most of image restoration literature, we use synthetic data with ground-truth to quantitatively analyze the proposed DPSR, as well as making a relatively fair comparison with other competing methods.

Blur kernel. For the sake of thoroughly evaluating the effectiveness of the proposed DPSR for arbitrary blur kernels, we consider three types of widely-used blur kernels, including Gaussian blur kernels, motion blur kernels, and disk (out-of-focus) blur kernels [12, 59]. The specifications of the blur kernels are given in Table 1. Some kernel examples are shown in Fig. 1. Note that the kernel sizes range from 5×5 to 35×35 . As shown in Table 2, we further consider Gaussian noise with two different noise levels, *i.e.*, 2.55 (1%) and 7.65 (3%), for scale factor 3.

Table 1. Three different types of blur kernels.

Type	#	Specification
Gaussian	16	8 isotropic Gaussian kernels with standard deviations uniformly sampled from the interval [0.6, 2], and 8 selected anisotropic Gaussian blur kernels from [65].
Motion	32	8 blur kernels from [35] and their augmented 8 kernels by random rotation and flip; and 16 realistic-looking motion blur kernels generated by the released code of [7].
Disk	8	8 disk kernels with radius uniformly sampled from the interval [1.8, 6]. They are generated by matlab function <code>fspecial('disk', r)</code> , where <code>r</code> is the radius.

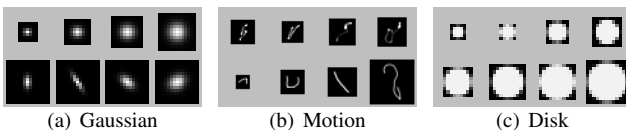


Fig. 1. Examples of (a) Gaussian blur kernels, (b) motion blur kernels and (c) disk blur kernels.

Parameter setting. In the alternating iterations between Eqn. (7) and Eqn. (8), we need to set λ and tune μ to obtain a satisfying performance. Setting such parameters has been considered as a non-trivial task [46]. However, the parameter setting of DPSR is generally easy with the following two principles. First, since λ is fixed and can be absorbed into σ , we can instead multiply σ by a scalar $\sqrt{\lambda}$ and therefore ignore the λ in Eqn. (8). Second, since μ has a non-descending order during iterations, we can instead set the $\sqrt{1/\mu}$ from Eqn. (12) with a non-ascending order to indirectly determine μ in each iteration. Empirically, a good rule of thumb is to set λ to 1/3 and exponentially decrease $\sqrt{1/\mu}$ from 49 to a small σ -dependent value (*e.g.*, $\max(2.55, \sigma)$) for a total of 15 iterations.

Compared methods. We compare the proposed DPSR with six methods, including two representative DNN-based methods for bicubic degradation (*i.e.*, VDSR [29] and RCAN [67]), two cascaded deblurring and SISR methods (*i.e.*, IRCNN+RCAN and DeblurGAN+RCAN), and two specially designed methods for blurry LR images (*i.e.*, GFN [66] and ZSSR [51]). To be specific, VDSR is the first very deep network for SISR; RCAN consists of more than 400 layers and achieves state-of-the-art performance for bicubic degradation; IRCNN is a plug-and-play method with deep denoiser prior that can handle non-blind image deblurring; DeblurGAN [32] is a deep blind deblurring method based on generative adversarial network (GAN) [24]; GFN is a DNN-based method for joint blind motion deblurring and super-resolution; ZSSR is an unsupervised DNN-based method that can super-resolve blurry and noisy LR images. Note that IRCNN, ZSSR and DPSR can take the blur kernel and noise level as input. For a fair comparison, we modify ZSSR to our new degradation model.

Quantitative results. The PSNR and SSIM results of different methods for different degradation settings on the color BSD68 dataset [39, 48, 62] are shown in Table 2, from which we have several observations. First, while RCAN outperforms VDSR by a large margin for bicubic degradation (see [67]), it has comparable performance to VDSR and even bicubic interpolation for the complex degradations settings. Such phenomenon has also been reported in [51, 65]. Second, after a deblurring step by IRCNN, IRCNN+RCAN can significantly improve the PSNR and SSIM values. Third, DeblurGAN+RCAN and GFN give rise to poor performance which may be attributed to the limited capability of successive convolutional layers in handling distortion of large complex blur. Fourth, ZSSR is less effective for large complex blur kernels due to the lack of recurrence property of the blurry LR image. Last, our DPSR achieves the best performance since it directly optimizes the energy function for the given degradation and can effectively handle the distortion of blur via Eqn. (9).

Table 2. Average PSNR and SSIM results of different methods for different degradation settings on the color BSD68 dataset [39, 48, 62]. The best two results are highlighted in red and blue colors, respectively.

Degradation Setting			Methods							
Scale Factor	Kernel Type	Noise Level	Bicubic	VDSR	RCAN	IRCNN +RCAN	DeblurGAN +RCAN	GFN	ZSSR	DPSR
×2	Gaussian	0	23.47/0.596	23.36/0.589	23.59/0.603	24.79/0.629	19.36/0.400	–	21.44/0.542	27.28/0.763
	Motion		19.84/0.449	19.86/0.451	19.82/0.448	28.54/0.806	17.46/0.268	–	17.99/0.367	30.05/0.869
	Disk		21.85/0.507	21.86/0.508	21.85/0.507	25.48/0.671	19.33/0.370	–	21.25/0.490	28.61/0.816
	Gaussian	0	22.11/0.526	22.03/0.520	22.20/0.532	23.52/0.566	18.18/0.347	–	18.97/0.442	25.22/0.665
	Motion		18.83/0.424	18.84/0.424	18.81/0.422	25.88/0.699	16.25/0.228	–	16.80/0.348	27.22/0.769
	Disk		20.70/0.464	20.70/0.465	20.69/0.464	23.82/0.594	18.28/0.336	–	19.05/0.430	26.19/0.716
×3	Gaussian	2.55 (1%)	22.05/0.513	21.95/0.507	22.05/0.495	22.95/0.540	17.64/0.195	–	20.52/0.477	23.92/0.595
	Motion		18.80/0.412	18.81/0.413	18.74/0.390	24.18/0.604	15.55/0.122	–	18.19/0.402	24.94/0.648
	Disk		20.65/0.452	20.66/0.453	20.59/0.429	22.33/0.517	16.74/0.142	–	19.88/0.441	23.44/0.567
	Gaussian	7.65 (3%)	21.60/0.436	21.54/0.433	21.01/0.342	22.07/0.463	14.93/0.094	–	20.22/0.407	23.17/0.557
	Motion		18.58/0.346	18.60/0.350	18.21/0.259	22.47/0.514	13.91/0.065	–	18.07/0.340	23.28/0.564
	Disk		20.32/0.380	20.34/0.384	19.81/0.288	21.19/0.483	14.43/0.071	–	19.64/0.373	22.25/0.515
×4	Gaussian	0	21.18/0.486	21.11/0.482	21.27/0.491	22.61/0.526	17.52/0.329	21.66/0.507	16.83/0.380	24.04/0.604
	Motion		18.10/0.407	18.11/0.408	18.08/0.406	24.42/0.625	15.68/0.225	17.79/0.398	15.58/0.329	25.69/0.695
	Disk		19.86/0.439	19.86/0.440	19.85/0.439	22.72/0.540	17.59/0.321	20.38/0.460	16.91/0.392	24.84/0.649

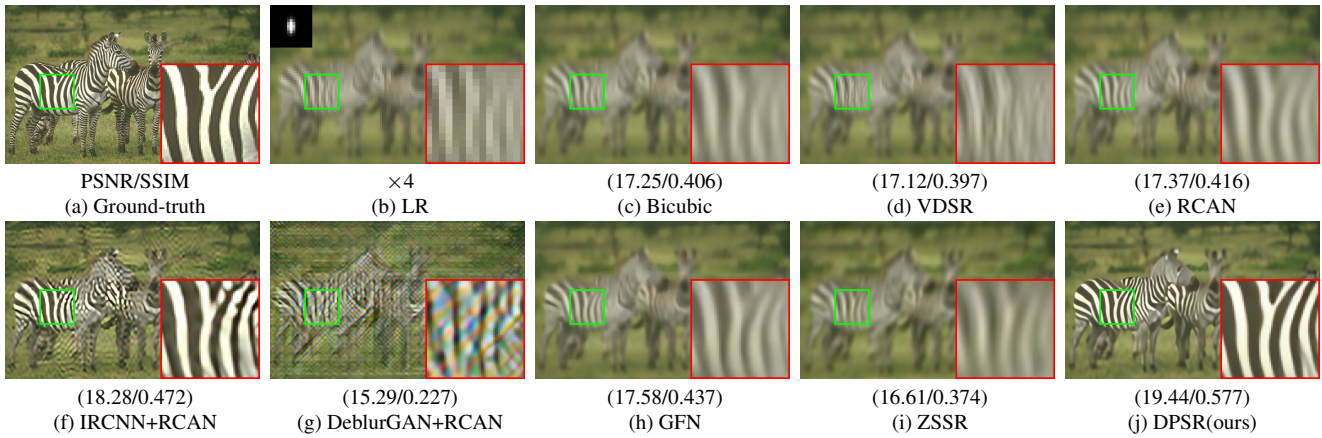


Fig. 2. The performance comparison of different methods for scale factor 4 on image “253027” with Gaussian kernel. The blur kernel is shown on the upper-left corner of the LR image.

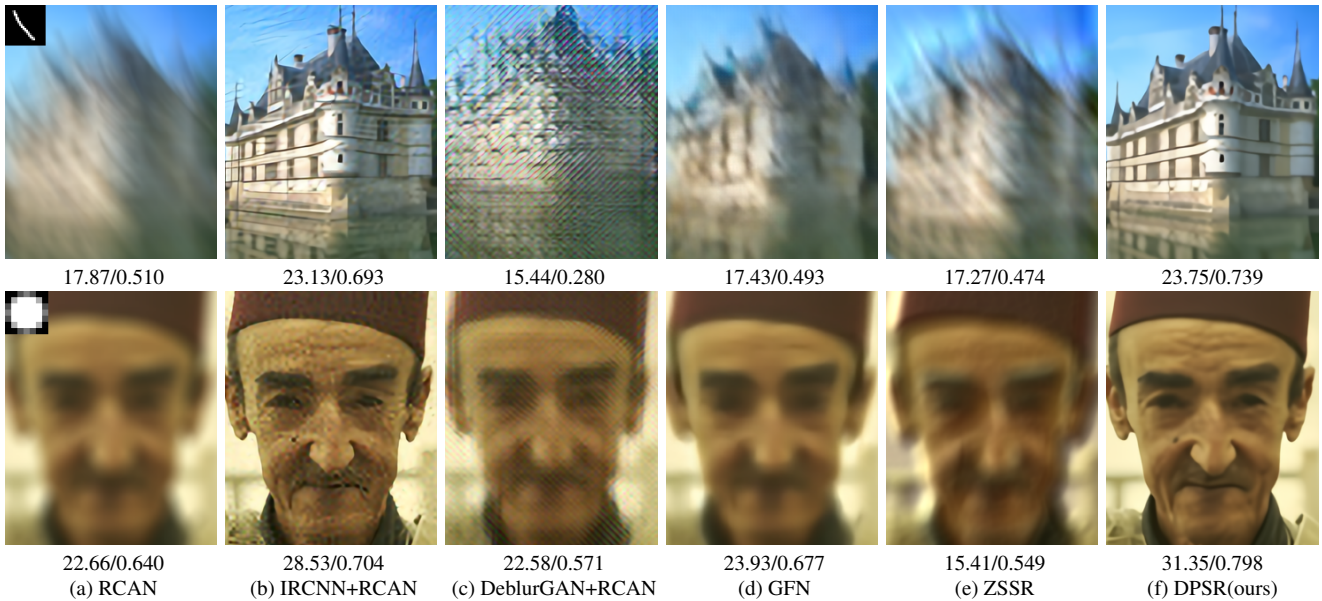


Fig. 3. The performance comparison of different methods for scale factor 4 on image “102061” with motion kernel (first row) and image “189080” with disk kernel (second row). The blur kernel is shown on the upper-left corner of the super-resolved image by RCAN.



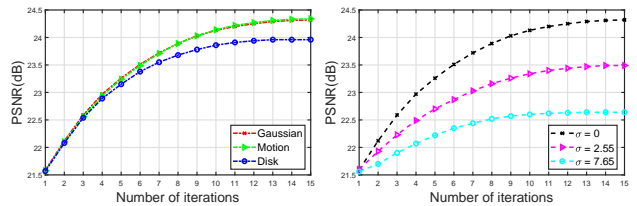
(a) LR ($\times 4$) (b) IRCNN+RCAN (c) ESRGAN (d) IRCNN+ESRGAN (e) RCAN+DeblurGAN (f) DPSRGAN(ours)

Fig. 4. The performance comparison of different methods for scale factor 4 on image “101087” (first row) and image “102061” (second row). The blur kernel is shown on the upper-left corner of the LR image.

Visual results. Fig. 2 shows the visual comparison of different methods for super-resolving LR image with Gaussian blur by a scale factor of 4. As one can see, VDSR and RCAN undoubtedly produce unpleasant results due to the degradation mismatch. DeblurGAN+RCAN generates very unpleasant artifacts and does not alleviate the blurriness compared to the LR image. GFN does not perform well and ZSSR is less effective for super-resolving the very blurry LR image. While IRCNN+RCAN produces better results than DeblurGAN+RCAN, GFN and ZSSR, it generates some noise-like artifacts which is possibly introduced by IRCNN and then amplified by RCAN. In comparison, our DPSR produces the most visually pleasant results.

Fig. 3 further shows the visual comparison on other two types of blur kernels. It can be observed that DPSR consistently gives rise to the best visual results. Particularly, although GFN can deal with the motion blur to some degree, its visual result is significantly inferior to those of IRCNN+RCAN and DPSR. The underlying reason is that it has limited ability to blindly handle the distortion of blur via successive convolutional layers. We note that other learning based methods (*e.g.*, [54]) also suffer from such problem.

Convergence. Since our DPSR is a special case of plug-and-play framework for SISR, one may refer to [10, 45] to analyze the theoretical convergence. In this paper, we only provide an empirical evidence to show the convergence of DPSR. Fig. 5 shows the convergence curves of the proposed DPSR on image “102061” with respect to different types of blur kernels and different noise levels. In Fig. 5(a), we fix the noise level to 0 and choose the third kernel for each kernel type. In Fig. 5(b), we fix the blur kernel to the third Gaussian kernel and choose three different noise levels, including 0, 2.55 and 7.65. It can be observed that DPSR converges very fast.



(a) Different types of kernels (b) Different noise levels
Fig. 5. Convergence curves of DPSR on image “102061” with respect to different types of blur kernels and different noise levels.

It is worth pointing out, in order to reduce the computational cost, our DPSR does not necessarily need iterative solution for bicubic degradation since our adopted super-resolver prior is already trained end-to-end on such degradation. Obviously, this is also an advantage over existing plug-and-play SISR. For the case of other degradations, the run time of our DPSR mainly depends on the total number of iterations. On a single GPU, it takes about 1.8 seconds for DPSR to super-resolve an LR image of size 256×256 by different scale factors. As a comparison, ZSSR spends 12, 14 and 18 seconds for scale factor 2, 3 and 4, respectively. In practice, one can tune the total number of iterations to balance the performance and speed.

Super-resolver prior with GAN. In the above experiments, SRResNet+ is trained by ℓ_1 loss. It is natural to rise the question: does the proposed method perform well if the super-resolver prior is trained with GAN loss? Following the training strategy of [58], we trained a SRGAN+ model for scale factor 4 by a weighted combination of L1 loss, VGG loss and GAN loss. For convenience, we refer to the proposed method with SRGAN+ as DPSRGAN. In this special case, we compare DPSRGAN with IRCNN+RCAN and four GAN-related methods, including ESRGAN [58], IRCNN+ESRGAN and RCAN+DeblurGAN.

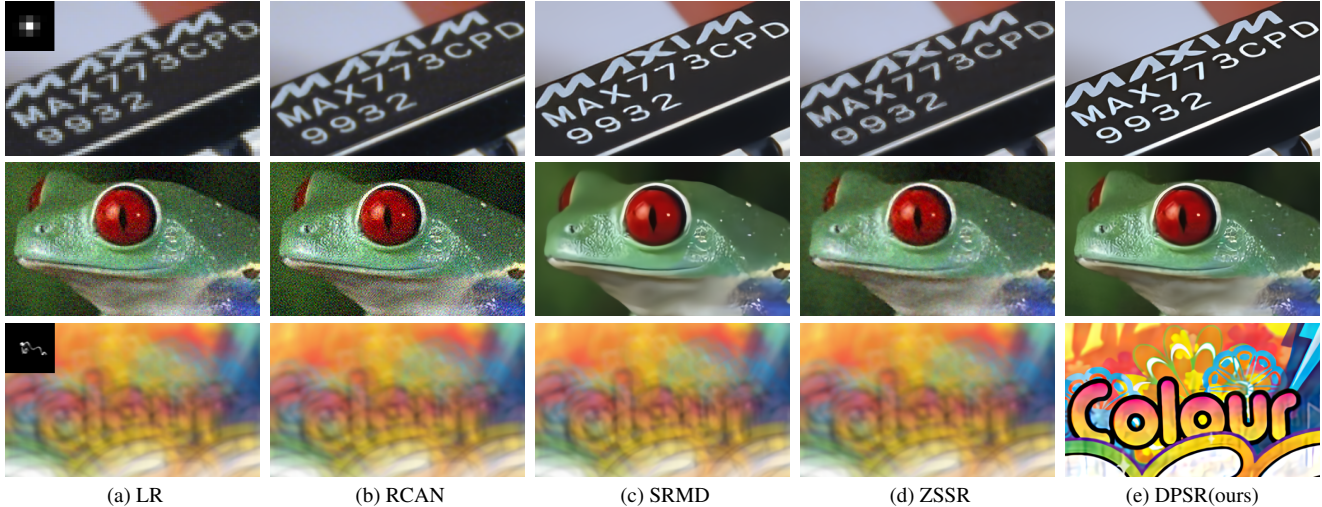


Fig. 6. The visual comparison of different methods on “chip” with scale factor 4 (first row), “frog” with scale factor 3 (second row) and “colour” with scale factor 2 (third row). The estimated blur kernel is shown on the upper-left corner of the LR image. Note that we assume the noisy image “frog” has no blur.

Fig. 4 shows the visual comparison of different methods. It can be seen that directly super-resolving the blurry LR image by ESRGAN does not improve the visual quality. By contrast, IRCNN+ESRGAN can deliver much better visual results as the distortion of blur is handled by IRCNN. Meanwhile, it amplifies the perturbation error of IRCNN, resulting in unpleasant visual artifacts. Although DeblurGAN is designed to handle motion blur, RCAN+DeblurGAN does not perform as well as expected. In comparison, our DPSR-GAN produces the most visually pleasant HR images with sharpness and naturalness.

4.2. LR images with estimated kernel

In this section, we focus on experiments on blurry LR images with estimated blur kernel. Such experiments can help to evaluate the feasibility of the new degradation model, the practicability and kernel sensitivity of the proposed DPSR. It is especially noteworthy that we do not know the HR ground-truth of the LR images.

Fig. 6 shows the visual comparison with state-of-the-art SISR methods (*i.e.*, RCAN [67], SRMD [63] and ZSSR [51]) on classical image “chip” [22], noisy image “frog” [33] and blurry image “colour” [43]. For the “chip” and “colour”, the blur kernel is estimated by [43]. For the noisy “frog”, we assume it has no blur and directly adopt our super-resolver prior to obtain the HR image. Note that once the blur kernel is estimated, our DPSR can reconstruct HR images with different scale factors, whereas SRMD and ZSSR with Eqn. (1) need to estimate a separate blur kernel for each scale factor.

From Fig. 6, we can observe that RCAN has very limited ability to deal with blur and noise because of its oversimplified bicubic degradation model. With a more general degradation model, SRMD and ZSSR yield better results than

RCAN on “chip” and “frog”. However, they cannot recover the latent HR image for the blurry image “colour” which is blurred by a large complex kernel. In comparison, our DPSR gives rise to the most visually pleasant results. As a result, our new degradation model is a feasible assumption and DPSR is an attractive SISR method as it can handle a wide variety of degradations.

5. Conclusion

In this paper, we proposed a well-principled deep plug-and-play super-resolution method for LR image with arbitrary blur kernels. We first design an alternative degradation model which can benefit existing blind deblurring methods for kernel estimation. We then solve the corresponding energy function via half quadratic splitting algorithm so as to exploit the merits of plug-and-play framework. It turns out we can explicitly handled the distortion of blur by a specialized modular. Such a distinctive merit actually enables the proposed method to deal with arbitrary blur kernels. It also turns out that we can plug super-resolver prior rather than denoiser prior into the plug-and-play framework. As such, we can fully exploit the advances of existing DNN-based SISR methods to design and train the super-resolver prior. Extensive experimental results demonstrated the feasibility of the new degradation model and effectiveness of the proposed method for super-resolving LR images with arbitrary blur kernels.

6. Acknowledgements

This work is supported by HK RGC General Research Fund (PolyU 152216/18E) and National Natural Science Foundation of China (grant no. 61671182, 61872118, 61672446).

References

- [1] Manyá V Afonso, José M Bioucas-Dias, and Mário AT Figueiredo. Fast image recovery using variable splitting and constrained optimization. *IEEE transactions on image processing*, 19(9):2345–2356, 2010. [3](#)
- [2] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, volume 3, pages 126–135, July 2017. [1](#), [4](#)
- [3] Simon Baker and Takeo Kanade. Limits on super-resolution and how to break them. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1167–1183, 2002. [1](#)
- [4] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009. [3](#)
- [5] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie-Line Alberi Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *British Machine Vision Conference*, 2012. [4](#)
- [6] Siavash Arjomand Bigdeli, Meiguang Jin, Paolo Favaro, and Matthias Zwicker. Deep mean-shift priors for image restoration. In *Advances in Neural Information Processing Systems*, 2017. [3](#)
- [7] G. Boracchi and A. Foi. Modeling the performance of image restoration from motion blur. *IEEE Transactions on Image Processing*, 21(8):3502–3517, 2012. [5](#)
- [8] 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](#)
- [9] Alon Brifman, Yaniv Romano, and Michael Elad. Turning a denoiser into a super-resolver using plug and play priors. In *IEEE International Conference on Image Processing*, pages 1404–1408, 2016. [3](#)
- [10] Gregory T Buzzard, Stanley H Chan, Suhas Sreehari, and Charles A Bouman. Plug-and-play unplugged: Optimization-free reconstruction using consensus equilibrium. *SIAM Journal on Imaging Sciences*, 11(3):2001–2020, 2018. [3](#), [7](#)
- [11] 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](#)
- [12] Stanley H Chan and Truong Q Nguyen. Single image spatially variant out-of-focus blur removal. In *IEEE International Conference on Image Processing*, pages 677–680, 2011. [5](#)
- [13] 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. [1](#), [2](#), [3](#)
- [14] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007. [3](#)
- [15] Aram Danielyan, Vladimir Katkovnik, and Karen Egiazarian. Image deblurring by augmented lagrangian with BM3D frame prior. In *Workshop on Information Theoretic Methods in Science and Engineering*, pages 16–18, 2010. [3](#)
- [16] Aram Danielyan, Vladimir Katkovnik, and Karen Egiazarian. BM3D frames and variational image deblurring. *IEEE Transactions on Image Processing*, 21(4):1715–1728, 2012. [3](#)
- [17] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European Conference on Computer Vision*, pages 184–199, 2014. [2](#)
- [18] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *European Conference on Computer Vision*, pages 391–407, 2016. [2](#)
- [19] Weisheng Dong, Lei Zhang, Guangming Shi, and Xin Li. Nonlocally centralized sparse representation for image restoration. *IEEE Transactions on Image Processing*, 22(4):1620–1630, 2013. [1](#)
- [20] Netalee Efrat, Daniel Glasner, Alexander Apartsin, Boaz Nadler, and Anat Levin. Accurate blur models vs. image priors in single image super-resolution. In *IEEE International Conference on Computer Vision*, pages 2832–2839, 2013. [1](#), [2](#)
- [21] Karen Egiazarian and Vladimir Katkovnik. Single image super-resolution via BM3D sparse coding. In *European Signal Processing Conference*, pages 2849–2853, 2015. [3](#)
- [22] Raanan Fattal. Image upsampling via imposed edge statistics. In *ACM transactions on graphics*, volume 26, page 95, 2007. [8](#)
- [23] Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédéric Durand. Deep joint demosaicking and denoising. *ACM Transactions on Graphics*, 35(6):191, 2016. [4](#)
- [24] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. [2](#), [5](#)
- [25] Shuhang Gu, Radu Timofte, and Luc Van Gool. Integrating local and non-local denoiser priors for image restoration. In *International Conference on Pattern Recognition*, 2018. [3](#)
- [26] Felix Heide, Markus Steinberger, Yun-Ta Tsai, Mushfiqur Rouf, Dawid Pajak, Dikpal Reddy, Orazio Gallo, Jing Liu, Wolfgang Heidrich, Karen Egiazarian, et al. Flexisp: A flexible camera image processing framework. *ACM Transactions on Graphics*, 33(6):231, 2014. [3](#)
- [27] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015. [4](#)
- [28] Junjun Jiang, Yi Yu, Jinhui Hu, Suhua Tang, and Jiayi Ma. Deep CNN denoiser and multi-layer neighbor component embedding for face hallucination. In *International Joint Conference on Artificial Intelligence*, pages 771–778, 2018. [3](#)
- [29] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1646–1654, 2016. [1](#), [2](#), [5](#)
- [30] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference for*

- Learning Representations*, 2015. 4
- [31] Jakob Kruse, Carsten Rother, and Uwe Schmidt. Learning to push the limits of efficient fft-based image deconvolution. In *IEEE International Conference on Computer Vision*, pages 4586–4594, 2017. 2
- [32] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiri Matas. DeblurGAN: Blind motion deblurring using conditional adversarial networks. In *IEEE International Conference on Computer Vision*, pages 945–952, 2013. 5
- [33] Marc Lebrun, Miguel Colom, and Jean-Michel Morel. The noise clinic: a blind image denoising algorithm. *Image Processing On Line*, 5:1–54, 2015. 8
- [34] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4681–4690, July 2017. 2, 4
- [35] Anat Levin, Yair Weiss, Fredo Durand, and William T Freeman. Understanding and evaluating blind deconvolution algorithms. pages 1964–1971, 2009. 5
- [36] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 136–144, July 2017. 1, 2
- [37] Risheng Liu, Xin Fan, Shichao Cheng, Xiangyu Wang, and Zhongxuan Luo. Proximal alternating direction network: A globally converged deep unrolling framework. In *AAAI Conference on Artificial Intelligence*, pages 1371–1378, 2018. 3
- [38] Risheng Liu, Long Ma, Yiyang Wang, and Lei Zhang. Learning converged propagations with deep prior ensemble for image enhancement. *IEEE Transactions on Image Processing*, 28(3):1528–1543, 2019. 3
- [39] 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 *IEEE International Conference on Computer Vision*, volume 2, pages 416–423, July 2001. 5, 6
- [40] Tim Meinhardt, Michael Möller, Caner Hazirbas, and Daniel Cremers. Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. In *IEEE International Conference on Computer Vision*, pages 1781–1790, 2017. 3
- [41] Tomer Michaeli and Michal Irani. Nonparametric blind super-resolution. In *IEEE International Conference on Computer Vision*, pages 945–952, 2013. 1
- [42] Shunsuke Ono. Primal-dual plug-and-play image restoration. *IEEE Signal Processing Letters*, 24(8):1108–1112, 2017. 3
- [43] Jinshan Pan, Zhe Hu, Zhixun Su, and Ming-Hsuan Yang. Deblurring text images via L0-regularized intensity and gradient prior. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2908, 2014. 8
- [44] Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. In *Advances in Neural Information Processing Systems*, pages 1095–1106, 2018. 2
- [45] Edward T Reehorst and Philip Schniter. Regularization by denoising: Clarifications and new interpretations. *arXiv preprint arXiv:1806.02296*, 2018. 3, 7
- [46] Yaniv Romano, Michael Elad, and Peyman Milanfar. The little engine that could: Regularization by denoising (red). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844, 2017. 5
- [47] Arie Rond, Raja Giryes, and Michael Elad. Poisson inverse problems by the plug-and-play scheme. *Journal of Visual Communication and Image Representation*, 41:96–108, 2016. 3
- [48] Stefan Roth and Michael J Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205–229, 2009. 5, 6
- [49] Mehdi SM Sajjadi, Bernhard Schölkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *IEEE International Conference on Computer Vision*, pages 4501–4510, 2017. 2
- [50] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1874–1883, 2016. 2
- [51] Assaf Shocher, Nadav Cohen, and Michal Irani. “zero-shot” super-resolution using deep internal learning. In *IEEE International Conference on Computer Vision*, pages 3118–3126, 2018. 2, 5, 8
- [52] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2
- [53] F Sroubek, Gabriel Cristóbal, and J Flusser. Simultaneous super-resolution and blind deconvolution. In *Journal of Physics: Conference Series*, volume 124, page 012048, 2008. 1
- [54] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Ji-aya Jia. Scale-recurrent network for deep image deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8174–8182, 2018. 7
- [55] Radu Timofte, Vincent De Smet, and Luc Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *Asian Conference on Computer Vision*, pages 111–126, 2014. 2
- [56] Tom Tirer and Raja Giryes. Image restoration by iterative denoising and backward projections. *IEEE Transactions on Image Processing*, 28(3):1220–1234, 2019. 3
- [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 European Conference on Computer Vision Workshops*, 2018. 2, 4, 7
- [59] Li Xu, Jimmy SJ Ren, Ce Liu, and Jiaya Jia. Deep convolutional neural network for image deconvolution. In *NIPS*, pages 1790–1798, 2014. 5
- [60] Chih-Yuan Yang, Chao Ma, and Ming-Hsuan Yang. Single-image super-resolution: A benchmark. In *European Confer-*

- ence on Computer Vision*, pages 372–386, 2014. [1](#)
- [61] Jianchao Yang, John Wright, Thomas Huang, and Yi Ma. Image super-resolution as sparse representation of raw image patches. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. [2](#)
- [62] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, pages 3142–3155, 2017. [3](#), [5](#), [6](#)
- [63] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep CNN denoiser prior for image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3929–3938, July 2017. [2](#), [3](#), [8](#)
- [64] Kai Zhang, Wangmeng Zuo, and Lei Zhang. FFDNet: Toward a fast and flexible solution for CNN based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018. [4](#)
- [65] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3262–3271, 2018. [1](#), [2](#), [4](#), [5](#)
- [66] Xinyi Zhang, Hang Dong, Zhe Hu, Wei-Sheng Lai, Fei Wang, and Ming-Hsuan Yang. Gated fusion network for joint image deblurring and super-resolution. In *British Machine Vision Conference*, 2018. [2](#), [5](#)
- [67] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European Conference on Computer Vision*, pages 286–301, 2018. [5](#), [8](#)
- [68] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. [2](#)
- [69] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *IEEE International Conference on Computer Vision*, pages 479–486, 2011. [3](#)