

# Plug-and-Play Image Restoration with Deep Denoiser Prior

Kai Zhang, Yawei Li, Wangmeng Zuo, *Senior Member, IEEE*, Lei Zhang, *Fellow, IEEE*,  
Luc Van Gool and Radu Timofte, *Member, IEEE*

**Abstract**—Recent works on plug-and-play image restoration have shown that a denoiser can implicitly serve as the image prior for model-based methods to solve many inverse problems. Such a property induces considerable advantages for plug-and-play image restoration (e.g., integrating the flexibility of model-based method and effectiveness of learning-based methods) when the denoiser is discriminatively learned via deep convolutional neural network (CNN) with large modeling capacity. However, while deeper and larger CNN models are rapidly gaining popularity, existing plug-and-play image restoration hinders its performance due to the lack of suitable denoiser prior. In order to push the limits of plug-and-play image restoration, we set up a benchmark deep denoiser prior by training a highly flexible and effective CNN denoiser. We then plug the deep denoiser prior as a modular part into a half quadratic splitting based iterative algorithm to solve various image restoration problems. We, meanwhile, provide a thorough analysis of parameter setting, intermediate results and empirical convergence to better understand the working mechanism. Experimental results on three representative image restoration tasks, including deblurring, super-resolution and demosaicing, demonstrate that the proposed plug-and-play image restoration with deep denoiser prior not only significantly outperforms other state-of-the-art model-based methods but also achieves competitive or even superior performance against state-of-the-art learning-based methods. The source code is available at <https://github.com/cszn/DPIR>.

**Index Terms**—Denoiser Prior, Image Restoration, Convolutional Neural Network, Half Quadratic Splitting, Plug-and-Play

## 1 INTRODUCTION

IMAGE RESTORATION (IR) has been a long-standing problem for its highly practical value in various low-level vision applications [1], [2]. In general, the purpose of image restoration is to recover the latent clean image  $\mathbf{x}$  from its degraded observation  $\mathbf{y} = \mathcal{T}(\mathbf{x}) + \mathbf{n}$ , where  $\mathcal{T}$  is the noise-irrelevant degradation operation,  $\mathbf{n}$  is assumed to be additive white Gaussian noise (AWGN) of standard deviation  $\sigma$ . By specifying different degradation operations, one can correspondingly get different IR tasks. Typical IR tasks would be image denoising when  $\mathcal{T}$  is an identity operation, image deblurring when  $\mathcal{T}$  is a two-dimensional convolution operation, image super-resolution when  $\mathcal{T}$  is a composite operation of convolution and down-sampling, color image demosaicing when  $\mathcal{T}$  is a color filter array (CFA) masking operation.

Since IR is an ill-posed inverse problem, the prior which is also called regularization needs to be adopted to constrain the solution space [3], [4]. From a Bayesian perspective, the solution  $\hat{\mathbf{x}}$  can be obtained by solving a Maximum A Posteriori (MAP) estimation problem,

$$\hat{\mathbf{x}} = \arg \max_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x}) + \log p(\mathbf{x}), \quad (1)$$

K. Zhang, Y. Li and R. Timofte are with the Computer Vision Lab, ETH Zürich, Zürich, Switzerland (e-mail: kai.zhang@vision.ee.ethz.ch; yawei.li@vision.ee.ethz.ch; radu.timofte@vision.ee.ethz.ch).

L. Van Gool is with the Computer Vision Lab, ETH Zürich, Zürich, Switzerland, and also with KU Leuven, Leuven, Belgium (e-mail: van-gool@vision.ee.ethz.ch).

W. Zuo is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China (e-mail: cswmzuo@gmail.com).

L. Zhang is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China (e-mail: cslzhang@comp.polyu.edu.hk).

where  $\log p(\mathbf{y}|\mathbf{x})$  represents the log-likelihood of observation  $\mathbf{y}$ ,  $\log p(\mathbf{x})$  delivers the prior of clean image  $\mathbf{x}$  and is independent of degraded image  $\mathbf{y}$ . More formally, (1) can be reformulated as

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \frac{1}{2\sigma^2} \|\mathbf{y} - \mathcal{T}(\mathbf{x})\|^2 + \lambda \mathcal{R}(\mathbf{x}), \quad (2)$$

where the solution minimizes an energy function composed of a data term  $\frac{1}{2\sigma^2} \|\mathbf{y} - \mathcal{T}(\mathbf{x})\|^2$  and a regularization term  $\lambda \mathcal{R}(\mathbf{x})$  with regularization parameter  $\lambda$ . Specifically, the data term guarantees the solution accords with the degradation process, while the prior term alleviates the ill-posedness by enforcing desired property on the solution.

Generally, the methods to solve (2) can be divided into two main categories, i.e., model-based methods and learning-based methods. The former aim to directly solve (2) with some optimization algorithms, while the latter mostly train a truncated unfolding inference through an optimization of a loss function on a training set containing  $N$  degraded-clean image pairs  $\{(\mathbf{y}_i, \mathbf{x}_i)\}_{i=1}^N$  [5], [6], [7], [8], [9]. In particular, the learning-based methods are usually modeled as the following bi-level optimization problem

$$\left\{ \begin{array}{l} \min_{\Theta} \sum_{i=1}^N \mathcal{L}(\hat{\mathbf{x}}_i, \mathbf{x}_i) \\ \text{s.t. } \hat{\mathbf{x}}_i = \arg \min_{\mathbf{x}} \frac{1}{2\sigma^2} \|\mathbf{y}_i - \mathcal{T}(\mathbf{x})\|^2 + \lambda \mathcal{R}(\mathbf{x}), \end{array} \right. \quad (3a)$$

$$(3b)$$

where  $\Theta$  denotes the trainable parameters,  $\mathcal{L}(\hat{\mathbf{x}}_i, \mathbf{x}_i)$  measures the loss of estimated clean image  $\hat{\mathbf{x}}_i$  with respect to ground truth image  $\mathbf{x}_i$ . By replacing the unfolding inference (3b) with a predefined function  $\hat{\mathbf{x}} = f(\mathbf{y}, \Theta)$ , one can

treat the plain learning-based methods as general case of (3).

It is easy to note that one main difference between model-based methods and learning-based methods is that, the former are flexible to handle various IR tasks by simply specifying  $\mathcal{T}$  and can directly optimize on the degraded image  $\mathbf{y}$ , whereas the later require cumbersome training to learn the model before testing and are usually restricted by specialized tasks. Nevertheless, learning-based methods can not only enjoy a fast testing speed but also tend to deliver better performance due to the end-to-end training. In contrast, model-based methods are usually time-consuming with sophisticated priors for the purpose of good performance [10]. As a result, these two categories of methods have their respective merits and drawbacks, and thus it would be attractive to investigate their integration which leverages their respective merits. Such an integration has resulted in the deep plug-and-play IR method which replaces the denoising subproblem of model-based optimization with learning-based CNN denoiser prior.

The main idea of deep plug-and-play IR is that, with the aid of variable splitting algorithms, such as alternating direction method of multipliers (ADMM) [11] and half-quadratic splitting (HQS) [12], it is possible to deal with the data term and prior term separately [13], and particularly, the prior term only corresponds to a denoising subproblem [14], [15], [16] which can be solved via deep CNN denoiser. Although several deep plug-and-play IR works have been proposed, they typically suffer from the following drawbacks. First, they either adopt different denoisers to cover a wide range of noise levels or use a single denoiser trained on a certain noise level, which are not suitable to solve the denoising subproblem. For example, the IR-CNN [17] denoisers involve 25 separate 7-layer denoisers, each of which is trained on an interval noise level of 2. Second, their deep denoisers are not powerful enough, and thus, the performance limit of deep plug-and-play IR is unclear. Third, a deep empirical understanding of their working mechanism is lacking.

This paper is an extension of our previous work [17] with a more flexible and powerful deep CNN denoiser which aims to push the limits of deep plug-and-play IR by conducting extensive experiments on different IR tasks. Specifically, inspired by FFDNet [18], the proposed deep denoiser can handle a wide range of noise levels via a single model by taking the noise level map as input. Moreover, its effectiveness is enhanced by taking advantages of both ResNet [19] and U-Net [20]. The deep denoiser is further incorporated into HQS-based plug-and-play IR to show the merits of using powerful deep denoiser. Meanwhile, a novel periodical geometric self-ensemble is proposed to potentially improve the performance without introducing extra computational burden, and a thorough analysis of parameter setting, intermediate results and empirical convergence are provided to better understand the working mechanism of the proposed deep plug-and-play IR.

The contribution of this work is summarized as follows:

- A flexible and powerful deep CNN denoiser is trained. It not only outperforms the state-of-the-art deep Gaussian denoising models but also is suitable to solve plug-and-play IR.

- The HQS-based plug-and-play IR is thoroughly analyzed with respect to parameter setting, intermediate results and empirical convergence, providing a better understanding of the working mechanism.
- Extensive experimental results on deblurring, super-resolution and demosaicing have demonstrated the superiority of the proposed plug-and-play IR with deep denoiser prior.

## 2 RELATED WORKS

Plug-and-play IR generally involves two steps. The first step is to decouple the data term and prior term of the objective function via a certain variable splitting algorithm, resulting in an iterative scheme consisting of alternately solving a data subproblem and a prior subproblem. The second step is to solve the prior subproblem with any off-the-shelf denoisers, such as K-SVD [21], non-local means [22], BM3D [23]. As a result, unlike traditional model-based methods which needs to specify the explicit and hand-crafted image priors, plug-and-play IR can implicitly define the prior via the denoiser. Such an advantage offers the possibility of leveraging very deep CNN denoiser to improve effectiveness.

### 2.1 Plug-and-Play IR with Non-CNN Denoiser

The plug-and-play IR can be traced back to [4], [14], [16]. In [24], Danielyan et al. used Nash equilibrium to derive an iterative decoupled deblurring BM3D (IDDBM3D) method for image deblurring. In [25], a similar method equipped with CBM3D denoiser prior was proposed for single image super-resolution (SISR). By iteratively updating a back-projection step and a CBM3D denoising step, the method has an encouraging performance for its PSNR improvement over SRCNN [26]. In [14], the augmented Lagrangian method was adopted to fuse the BM3D denoiser to solve image deblurring task. With a similar iterative scheme as in [24], the first work that treats the denoiser as “plug-and-play prior” was proposed in [16]. Prior to that, a similar plug-and-play idea is mentioned in [4] where HQS algorithm is adopted for image denoising, deblurring and inpainting. In [15], Heide et al. used an alternative to ADMM and HQS, i.e., the primal-dual algorithm [27], to decouple the data term and prior term. In [28], Teodoro et al. plugged class-specific Gaussian mixture model (GMM) denoiser [4] into ADMM to solve image deblurring and compressive imaging. In [29], Metzler et al. developed a denoising-based approximate message passing (AMP) method to integrate denoisers, such as BLS-GSM [30] and BM3D, for compressed sensing reconstruction. In [31], Chan et al. proposed plug-and-play ADMM algorithm with BM3D denoiser for single image super-resolution and quantized Poisson image recovery for single-photon imaging. In [32], Kamilov et al. proposed fast iterative shrinkage thresholding algorithm (FISTA) with BM3D and WNNM [10] denoisers for non-linear inverse scattering. In [33], Sun et al. proposed FISTA by plugging TV and BM3D denoiser prior for Fourier ptychographic microscopy. In [34], Yair and Michaeli proposed to use WNNM denoiser as the plug-and-play prior for inpainting and deblurring. In [35], Gavaskar and Chaudhury investigated the convergence of ISTA-based plug-and-play IR with non-local means denoiser.

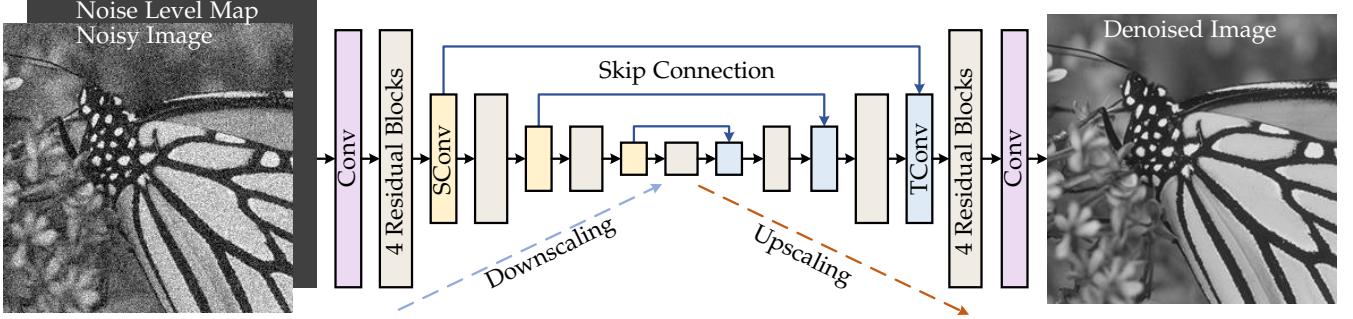


Fig. 1. The architecture of the proposed DRUNet denoiser prior. DRUNet takes an additional noise level map as input and combines U-Net [20] and ResNet [36]. “SConv” and “TConv” represent strided convolution and transposed convolution, respectively.

## 2.2 Plug-and-Play IR with Deep CNN Denoiser

With the development of deep learning techniques such as network design and gradient-based optimization algorithm, CNN-based denoiser has shown promising performance in terms of effectiveness and efficiency. Following its success, a flurry of CNN denoiser based plug-and-play IR works have been proposed. In [37], Romano et al. proposed explicit regularization by TNRD denoiser for image deblurring and SISR. In our previous work [17], different CNN denoisers are trained to plug into HQS algorithm to solve deblurring and SISR. In [38], Tirer and Giryes proposed iterative denoising and backward projections with IRCNN denoisers for image inpainting and deblurring. In [39], Gu et al. proposed to adopt WNNM and IRCNN denoisers for plug-and-play deblurring and SISR. In [40], Tirer and Giryes proposed to use the IRCNN denoisers for plug-and-play SISR. In [41], Li and Wu plugged the IRCNN denoisers into the split Bregman iteration algorithm to solve depth image inpainting. In [42], Ryu et al. provided the theoretical convergence analysis of plug-and-play IR based on forward-backward splitting algorithm and ADMM algorithm, and proposed spectral normalization to train a DnCNN denoiser. In [43], Sun et al. proposed a block coordinate regularization-by-denoising (RED) algorithm by leveraging DnCNN [44] denoiser as the explicit regularizer.

Although plug-and-play IR can leverage the powerful expressiveness of CNN denoiser, existing methods generally exploit DnCNN or IRCNN which do not make full use of CNN. Typically, the denoiser for plug-and-play IR should be non-blind and requires to handle a wide range of noise levels. However, DnCNN needs to separately learn a model for each noise level. To reduce the number of denoisers, some works adopt one denoiser fixed to a small noise level. However, according to [37] and as will be shown in Sec. 5.1.3, such a strategy tends to require a large number of iterations for a satisfying performance which would increase the computational burden. While IRCNN denoisers can handle a wide range of noise levels, it consists of 25 separate 7-layer denoisers, among which each denoiser is trained on an interval noise level of 2. Such a denoiser suffers from two drawbacks. First, it does not have the flexibility to hand a specific noise level. Second, it is not effective enough due to the shallow layers. Given the above considerations, it is necessary to devise a flexible and powerful denoiser to boost the performance of plug-and-play IR.

## 2.3 Why not Use a Blind Gaussian Denoiser for Plug-and-Play IR?

It is worth to emphasize that the denoiser for plug-and-play IR should be designed for non-blind Gaussian denoising. The reason is two-fold. First, as will be shown in (6b) of the iterative solution for plug-and-play IR, the sub-problem actually corresponds to a non-blind Gaussian denoising problem with a Gaussian noise level. Second, although the non-blind Gaussian denoising problem could be solved via the blind Gaussian denoiser, the role of the Gaussian denoiser for plug-and-play IR is to smooth out the unknown noise (e.g., structural noise introduced during iterations) rather than remove the Gaussian noise. As will be shown in Sec. 6, the noise distribution during iterations is usually non-Gaussian and varies across different IR tasks and even different iterations. Moreover, as we will see in Sec. 5.1.4, while the non-blind Gaussian denoiser can smooth out such non-Gaussian noise by setting a proper noise level, the blind Gaussian denoiser does not have such ability as it can only remove the Gaussian-like noise [18].

## 2.4 Difference to Deep Unfolding IR

It should be noted that, apart from plug-and-play IR, deep unfolding IR [45], [46], [47], [48] can also incorporate the advantages of both model-based methods and learning-based methods. The main difference between them is that the latter interprets a truncated unfolding optimization as an end-to-end trainable deep network and thus usually produce better results with fewer iterations. However, deep unfolding IR needs separate training for each task. On the contrary, plug-and-play IR is easy to deploy without such additional training.

## 3 LEARNING DEEP CNN DENOISER PRIOR

Although various CNN-based denoising methods have been recently proposed, most of them are not designed for plug-and-play IR. In [50], [51], [52], a novel training strategy without ground-truth is proposed. In [53], [54], [55], [56], real noise synthesis technique is proposed to handle real digital photographs. However, from a Bayesian perspective, the denoiser for plug-and-play IR should be a Gaussian denoiser. Hence, one can add synthetic Gaussian noise to clean image for supervised training. In [57], [58], [59], [60], the non-local module was incorporated into the network





10 represents lower quality and higher compression, and vice versa. Specifically, the quality factor  $q$  are normalized with  $(100 - q)/100$  for 0-1 normalization. We use the same training data as in denoising for training. Table 3 reports the average PSNR(dB) results of different methods for JPEG image deblocking with quality factors 10, 20, 30 and 40 on Classic5 and LIVE1 datasets. The compared methods include ARCNN [73], TNRD [9], DnCNN3 [44] and RNAN [58], QGAC [74]. Since DnCNN3 is trained also for denoising and SISR, we re-trained a non-blind DnCNN3 model with our training data. Compared to the original DnCNN3 model, the new model has average PSNR gains of 0.21dB and 0.19dB on the two testing datasets. To quantify the performance contribution of training data, we also trained a DRUNet model with less training data as in [18]. The results show that the PSNR decreases by 0.04dB on average, which demonstrates that a large training data can slightly improve the performance for JPEG image deblocking. From Table 3, we can see that DRUNet outperforms ARCNN, TNRD, DnCNN3 and QGAC by a large margin and has an average PSNR gain of 0.15dB over RNAN, which further demonstrates the flexibility and effectiveness of the proposed DRUNet.

### 3.3.4 Generalizability to Unseen Noise Level

In order to show the advantage of the bias-free DRUNet, we also train a DRUNet+B model whose biases were randomly initialized from a uniform distribution in  $[-1, 1]$ . Fig. 4 provides the visual results comparison between different models on a noisy image with an extremely large unseen noise level of 200. Note that since DnCNN and IRCNN do not have the flexibility to change the noise level, we first multiply the noisy image by a factor of 0.25 so that the noise level changes from 200 to 50. We then apply the DnCNN and IRCNN models for denoising and finally obtain the denoising results with a multiplication of 4. From Fig. 4, we can see that, even trained on noise level range of  $[0, 50]$ , the bias-free DRUNet can still perform well, whereas DRUNet+B (with biases) introduces noticeable visual artifacts while having a much lower PSNR. As we will see in Sec. 3.3.5, DRUNet+B has a comparable performance with bias-free DRUNet for noise levels in  $[0, 50]$ . Thus, we can conclude that bias-free DRUNet can enhance the generalizability to unseen noise level. Note that whether bias-free network can benefit other tasks or not remains further study.

### 3.3.5 Ablation Study

In order to further analyze the proposed DRUNet, we have performed an ablation study to quantify the performance contribution of different factors such as residual blocks, training data, biases, and noise level map. Table 4 reports the comparisons between DRUNet with four different cases, including case 1: DRUNet without skip connections of the residual blocks, case 2: DRUNet with less training data as in [18], case 3: DRUNet without removing the biases (i.e., DRUNet+B), and case 4: DRUNet without taking noise level map as input. By comparison, we can have the following conclusions. First, the residual blocks can ease the training to get a better performance. Second, the performance on Set12 dataset tends to be saturated with enough training data as the DRUNet model with more training data only improves the PSNR by an average of 0.01dB. Third, DRUNet with

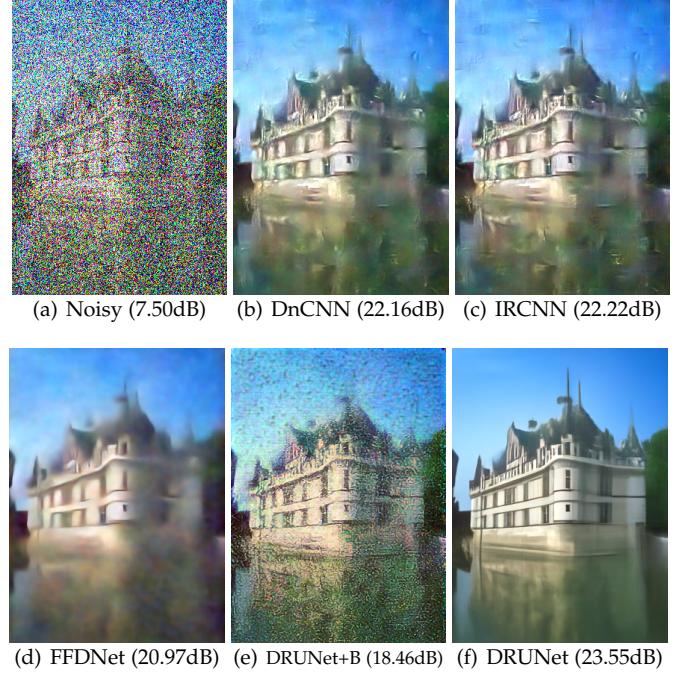


Fig. 4. An example to show the generalizability advantage of the proposed bias-free DRUNet. The noise level of the noisy image is 200.

biases has a similar performance to bias-free DRUNet for the trained noise levels, however, the bias-free DRUNet can enhance the generalizability to unseen noise level. Fourth, the noise level map can improve the performance as it introduces extra information of the noise. Such a phenomenon has also been reported in [18].

TABLE 4

The PSNR results comparison with different cases (i.e, case 1: DRUNet without residual blocks, case 2: DRUNet with less training data as in [18], case 3: DRUNet without removing the biases, and case 4: DRUNet without taking noise level map as input) with noise levels 15, 25 and 50 on Set12 dataset.

Datasets	Noise Level	Case 1	Case 2	Case 3	Case 4	DRUNet
Set12	15	33.12	33.23	33.24	33.16	33.25
	25	30.82	30.92	30.93	30.86	30.94
	50	27.78	27.87	27.89	27.83	27.90

### 3.3.6 Runtime, FLOPs and Maximum GPU Memory Consumption

Table 5 reports the runtime, FLOPs and maximum GPU memory consumption comparison with four representative methods (i.e., DnCNN, IRCNN, FFDNet and RNAN) on images of size  $256 \times 256$  and  $512 \times 512$  with noise level 50. We do not report other methods such as FOCNet for comparison since they are not implemented in PyTorch for a fair and easy comparison. Note that, in order to reduce the memory caused by the non-local module, RNAN splits the input image into overlapped patches with predefined maximum spatial size and then aggravates the results to obtain the final denoised image. The default maximum spatial size is 10,000 which is equivalent to a size of  $100 \times 100$ . We also compare RNAN\* which sets maximum spatial size to 70,000. As a simple example, RNAN and RNAN\* splits an



noise level range of  $[0, 50]$  is supposed to be enough for  $\sigma_k$ . Inspired by such domain knowledge, we can instead set  $\sigma_k$  and  $\lambda$  to implicitly determine  $\mu_k$ . Based on the fact that  $\mu_k$  should be monotonically increasing, we uniformly sample  $\sigma_k$  from a large noise level  $\sigma_1$  to a small one  $\sigma_K$  in log space. This means that  $\mu_k$  can be easily determined via  $\mu_k = \lambda/\sigma_k^2$ . Following [17],  $\sigma_1$  is fixed to 49 while  $\sigma_K$  is determined by the image noise level  $\sigma$ . Since  $K$  is user-specified and  $\sigma_K$  has clear physical meanings, they are practically easy to set. As for the theoretical convergence of plug-and-play IR, please refer to [31].

By far, the remaining parameter for setting is  $\lambda$ . Due to the fact that  $\lambda$  comes from the prior term and thus should be fixed, we can choose the optimal  $\lambda$  by a grid search on a validation dataset. Empirically,  $\lambda$  can yield favorable performance from the range of  $[0.19, 0.55]$ . In this paper, we fix it to 0.23 unless otherwise specified. It should be noted that since  $\lambda$  can be absorbed into  $\sigma$  and plays the role of controlling the trade-off between data term and prior term, one can implicitly tune  $\lambda$  by multiplying  $\sigma$  by a scalar. To have a clear understanding of the parameter setting, by denoting  $\alpha_k \triangleq \mu_k \sigma^2 = \lambda \sigma^2 / \sigma_k^2$  and assuming  $\sigma_K = \sigma = 1$ , we plot the values of  $\alpha_k$  and  $\sigma_k$  with respect to different number of iterations  $K = 8, 24$ , and 40 in Fig. 5.

### 4.3 Periodical Geometric Self-Ensemble

Geometric self-ensemble based on flipping and rotation is a commonly-used strategy to boost IR performance [75]. It first transforms the input via flipping and rotation to generate 8 images, then gets the corresponding restored images after feeding the model with the 8 images, and finally produces the averaged result after the inverse transformation. While a performance gain can be obtained via geometric self-ensemble, it comes at the cost of increased inference time.

Different from the above method, we instead periodically apply the geometric self-ensemble for every successive 8 iterations. In each iteration, there involves one transformation before denoising and the counterpart inverse transformation after denoising. Note that the averaging step is abandoned because the input of the denoiser prior model varies across iterations. We refer to this method as periodical geometric self-ensemble. Its distinct advantage is that the total inference time would not increase. We empirically found that geometric self-ensemble can generally improve the PSNR by 0.02dB~0.2dB.

Based on the above discussion, we summarized the detailed algorithm of HQS-based plug-and-play IR with deep denoiser prior, namely DPIR, in Algorithm 1.

## 5 EXPERIMENTS

To validate the effectiveness of the proposed DPIR, we consider three classical IR tasks, including image deblurring, single image super-resolution (SISR), and color image demosaicing. For each task, we will provide the specific degradation model, fast solution of (6a) in Algorithm 1, parameter setting for  $K$  and  $\sigma_K$ , initialization of  $\mathbf{z}_0$ , and the performance comparison with other state-of-the-art methods. To further analyze DPIR, we also provide the visual

**Algorithm 1:** Plug-and-play image restoration with deep denoiser prior (DPIR).

---

**Input :** Deep denoiser prior model, degraded image  $\mathbf{y}$ , degradation operation  $\mathcal{T}$ , image noise level  $\sigma$ ,  $\sigma_k$  of denoiser prior model at  $k$ -th iteration for a total of  $K$  iterations, trade-off parameter  $\lambda$ .

**Output:** Restored image  $\mathbf{z}_K$ .

```

1 Initialize  $\mathbf{z}_0$  from  $\mathbf{y}$ , pre-calculate  $\alpha_k \triangleq \lambda \sigma^2 / \sigma_k^2$ .
2 for  $k = 1, 2, \dots, K$  do
3    $\mathbf{x}_k = \arg \min_{\mathbf{x}} \|\mathbf{y} - \mathcal{T}(\mathbf{x})\|^2 + \alpha_k \|\mathbf{x} - \mathbf{z}_{k-1}\|^2$  ;// Solving data subproblem
4    $\mathbf{z}_k = \text{Denoiser}(\mathbf{x}_k, \sigma_k)$  ;// Denoising with deep DRUNet denoiser and periodical geometric self-ensemble
5 end
```

---

results of  $\mathbf{x}_k$  and  $\mathbf{z}_k$  at intermediate iterations as well as the convergence curves. Note that in order to show the advantage of the powerful DRUNet denoiser prior over IRCNN denoiser prior, we refer to DPIR with IRCNN denoiser prior as IRCNN+.

### 5.1 Image Deblurring

The degradation model for deblurring a blurry image with uniform blur (or image deconvolution) is generally expressed as

$$\mathbf{y} = \mathbf{x} \otimes \mathbf{k} + \mathbf{n} \quad (8)$$

where  $\mathbf{x} \otimes \mathbf{k}$  denotes two-dimensional convolution between the latent clean image  $\mathbf{x}$  and the blur kernel  $\mathbf{k}$ . By assuming the convolution is carried out with circular boundary conditions, the fast solution of (6a) is given by

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

where the  $\mathcal{F}(\cdot)$  and  $\mathcal{F}^{-1}(\cdot)$  denote Fast Fourier Transform (FFT) and inverse FFT,  $\overline{\mathcal{F}(\cdot)}$  denotes complex conjugate of  $\mathcal{F}(\cdot)$ . It can be noted that the blur kernel  $\mathbf{k}$  is only involved in (9). In other words, (9) explicitly handles the distortion of blur.

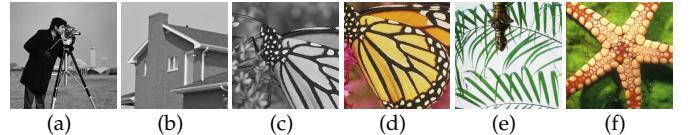


Fig. 6. Six classical testing images. (a) Cameraman; (b) House; (c) Monarch; (d) Butterfly; (e) Leaves; (f) Starfish.

#### 5.1.1 Quantitative and Qualitative Comparison

For the sake of making a quantitative analysis on the proposed DPIR, we consider six classical testing images as shown in Fig. 6 and two of the eight real blur kernels from [76]. Specifically, the testing images which we refer to as Set6 consist of 3 grayscale images and 3 color images. Among them, *House* and *Leaves* are full of repetitive





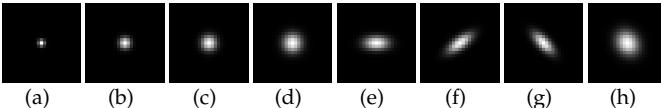


Fig. 10. The eight testing Gaussian kernels for SISR. (a)-(d) are isotropic Gaussian kernels; (e)-(f) are anisotropic Gaussian kernels.

### 5.2.1 Quantitative and Qualitative Comparison

To evaluate the flexibility of DPIR, we consider bicubic degradation model, and classical degradation model with 8 diverse Gaussian blur kernels as shown in Fig. 10. Following [83], the 8 kernels consist of 4 isotropic kernels with different standard deviations (i.e., 0.7, 1.2, 1.6 and 2.0) and 4 anisotropic kernels. We do not consider motion blur kernels since it has been pointed out that Gaussian kernels are enough for SISR task. To further analyze the performance, three different combinations of scale factor and noise level, including  $(s = 2, \sigma = 0)$ ,  $(s = 3, \sigma = 0)$  and  $(s = 3, \sigma = 7.65)$ , are considered.

For the compared methods, we consider the bicubic interpolation method, RCAN [84], SRFBN [85], MZSR [86], IRCNN and IRCNN+. Specifically, RCAN is the state-of-the-art bicubic degradation based deep model consisting of about 400 layers. SRFBN is a recurrent neural network with feed-back mechanism. Note that we do not retrain the RCAN and SRFBN models to handle the testing degradation cases as they lack flexibility. Moreover, it is unfair because our DPIR can handle a much wider range of degradations. MZSR is a zero-shot method based on meta-transfer learning which learns an initial network and then fine-tunes the model on a pair of given LR image and its re-degraded LR image with a few gradient updates. Similar to IRCNN and DPIR, MZSR is a non-blind method that assumes the blur kernel is known beforehand. Since MZSR needs to downsample the LR image for fine-tuning, the scale factor should be not too large in order to capture enough information. As a result, MZSR is mainly designed for scale factor 2.

Table 8 reports the average PSNR(dB) results of different methods for bicubic degradation and classical degradation on color BSD68 dataset. From Table 8, we can have several observations. First, as expected, RCAN and SRFBN achieve promising results on bicubic degradation with  $\sigma = 0$  but lose effectiveness when the true degradation deviates from the assumed one. We note that SAN [87] has a similar performance to RCAN and SRFBN since they are trained for bicubic degradation. Second, with the accurate classical degradation model, MZSR outperforms RCAN on most of the blur kernels. Third, IRCNN has a clear PSNR gain over MZSR on smoothed blur kernel. The reason is that MZSR relies heavily on the internal learning of LR image. Fourth, IRCNN performs better on bicubic kernel and the first isotropic Gaussian kernel with noise level  $\sigma = 0$  than others. This indicates that the IBP solution has very limited generalizability. On the other hand, IRCNN+ has a much higher PSNR than IRCNN, which demonstrates the advantage of closed-form solution over the IBP solution. Last, DPIR can further improves over IRCNN+ by using a more powerful denoiser.

Fig. 11 shows the visual comparison of different SISR

methods on an image corrupted by classical degradation model. It can be observed that MZSR and IRCNN produce better visual results than bicubic interpolation method. With an inaccurate data term solution, IRCNN fails to recover sharp edges. In comparison, by using a closed-form data term solution, IRCNN+ can produce much better results with sharp edges. Nevertheless, it lacks the ability to recover clean HR image. In contrast, with a strong denoiser prior, DPIR produces the best visual result with both sharpness and naturalness.

### 5.2.2 Intermediate Results and Convergence

Fig. 12(a)-(e) provides the visual results and PSNR results of  $\mathbf{x}_k$  and  $\mathbf{z}_k$  at different iterations of DPIR on the testing image from Fig. 11. One can observe that, although the LR image contains no noise, the the closed-form solution  $\mathbf{x}_1$  would introduce severe structured noise. However, it has a better PSNR than that of RCAN. After passing  $\mathbf{x}_1$  through the DRUNet denoiser, such structured noise is removed as can be seen from  $\mathbf{z}_1$ . Meanwhile, the tiny textures and structures are smoothed out and the edges become blurry. Nevertheless, the PSNR is significantly improved and is comparable to that of MZSR. As the number of iterations increases,  $\mathbf{x}_6$  contains less structured noise than  $\mathbf{x}_1$ , while  $\mathbf{z}_6$  recovers more details and sharper edges than  $\mathbf{z}_1$ . The corresponding PSNR convergence curves are plotted in Fig. 12(f), from which we can see that  $\mathbf{x}_k$  and  $\mathbf{z}_k$  converge quickly to the fixed point.

## 5.3 Color Image Demosaicing

Current consumer digital cameras mostly use a single sensor with a color filter array (CFA) to record one of three R, G, and B values at each pixel location. As an essential process in camera pipeline, demosaicing aims to estimate the missing pixel values from a one-channel mosaiced image and the corresponding CFA pattern to recover a three-channel image [88], [89], [90]. The degradation model of mosaiced image can be expressed as

$$\mathbf{y} = \mathbf{M} \odot \mathbf{x} \quad (15)$$

where  $\mathbf{M}$  is determined by CFA pattern and is a matrix with binary elements indicating the missing pixels of  $\mathbf{y}$ , and  $\odot$  denotes element-wise multiplication. The closed-form solution of (6a) is given by

$$\mathbf{x}_{k+1} = \frac{\mathbf{M} \odot \mathbf{y} + \alpha_k \mathbf{z}_k}{\mathbf{M} + \alpha_k}. \quad (16)$$

In this paper, we consider the commonly-used Bayer CFA pattern with RGGB arrangement. For the parameters  $K$  and  $\sigma_K$ , they are set to 40 and 0.6, respectively. For  $\mathbf{z}_0$ , it is initialized by matlab's `demosaic` function.

### 5.3.1 Quantitative and Qualitative Comparison

To evaluate the performance of DPIR for color image demosaicing, the widely-used Kodak dataset (consisting of 24 color images of size  $768 \times 512$ ) and McMaster dataset (consisting of 18 color images of size  $500 \times 500$ ) are used. The corresponding mosaiced images are obtained by filtering the color images with the Bayer CFA pattern. The compared methods include matlab's `demosaic` function [88],



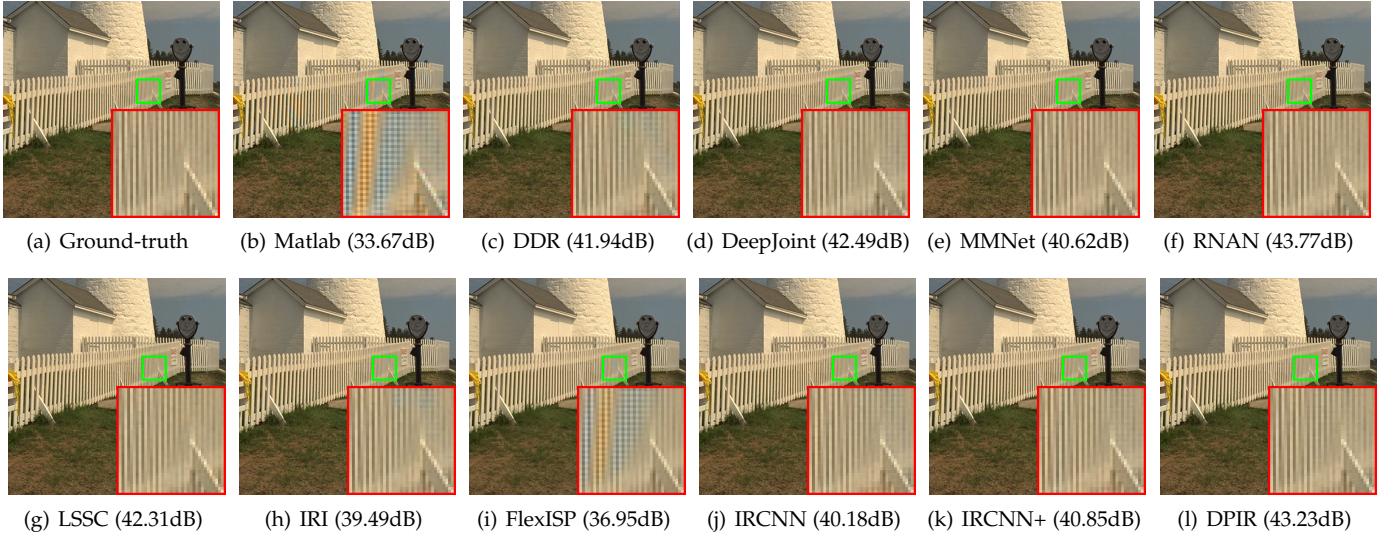


Fig. 13. Visual results comparison of different demosaicing methods on image *kodim19* from Kodak dataset.

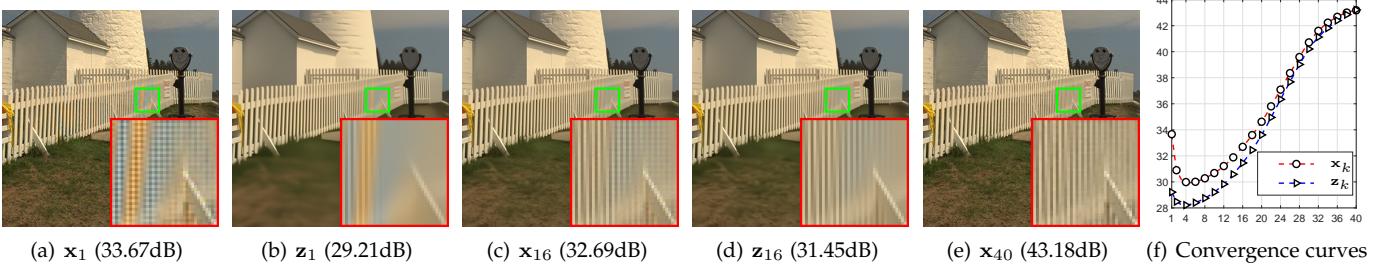


Fig. 14. (a)-(e) Visual results and PSNR results of  $x_k$  and  $z_k$  at different iterations; (f) Convergence curves of PSNR results (y-axis) for  $x_k$  and  $z_k$  with respect to number of iterations (x-axis).

directional difference regression (DDR) [91], deep unfolding majorization-minimization network (MMNet) [92], residual learning-based joint demosaicing-denoising (RLDD) [93], deep joint demosaicing and denoising (DeepJoint) [89], very deep residual non-local attention network (RNAN) [58] learned simultaneous sparse coding (LSSC) [94], iterative residual interpolation (IRI) [95], minimized-Laplacian residual interpolation (MLRI) [96], primal-dual algorithm with CBM3D denoiser prior (FlexISP) [15], IRCNN and IRCNN+. Note that DDR, MMNet, RLDD, DeepJoint, and RNAN are learning-based methods, while LSSC, IRI, MLRI, FlexISP, IRCNN, IRCNN+, and DPIR are model-based methods.

Table 9 reports the average PSNR(dB) results of different methods on Kodak dataset and McMaster dataset. It can be seen that while RNAN and MMNet achieve the best results, DPIR can have a very similar result and significantly outperforms the other model-based methods. With a stronger denoiser, DPIR has an average PSNR improvement up to 1.8dB over IRCNN+.

Fig. 13 shows the visual results comparison of different methods on a testing image from Kodak dataset. As one can see, the Matlab's simple demosaicing method introduces some zipper effects and false color artifacts. Such artifacts are highly reduced by learning-based methods such as DeepJoint, MMNet and RNAN. For the model-based methods, DPIR produces the best visual results whereas the

others give rise to noticeable artifacts.

### 5.3.2 Intermediate Results and Convergence

Figs. 14(a)-(e) show the visual results and PSNR results of  $x_k$  and  $z_k$  at different iterations. One can see that the DRUNet denoiser prior plays the role of smoothing out current estimation  $x$ . By passing  $z$  through (16), the new output  $x$  obtained by a weighted average of  $y$  and  $z$  becomes unsmooth. In this sense, the denoiser also aims to diffuse  $y$  for a better estimation of missing values. Fig. 14(f) shows the PSNR convergence curves of  $x_k$  and  $z_k$ . One can see that the two PSNR sequences are not monotonic but they eventually converge to the fixed point. Specifically, a decrease of the PSNR value for the first four iterations can be observed as the denoiser with a large noise level removes much more useful information than the unwanted artifacts.

## 6 DISCUSSION

While the denoiser prior for plug-and-play IR is trained for Gaussian denoising, it does not necessarily mean the noise of its input (or more precisely, the difference to the ground-truth) has a Gaussian distribution. In fact, the noise distribution varies across different IR tasks and even different iterations. Fig. 15 shows the noise histogram of  $x_1$  and  $x_8$  in Fig. 8 for deblurring,  $x_1$  and  $x_{24}$  in Fig. 12 for super-resolution, and  $x_1$  and  $x_{40}$  in Fig. 14 for demosaicing. It

can be observed that the three IR tasks has very different noise distributions. This is intuitively reasonable because the noise also correlates with the degradation operation which is different for the three IR tasks. Another interesting observation is that the two noise distributions of  $x_1$  and  $x_8$  in Fig. 15(a) are different and the latter tends to be Gaussian-like. The underlying reason is that the blurriness caused by blur kernel is gradually alleviated after several iterations. In other words,  $x_8$  suffers much less from the blurriness and thus is dominated by Gaussian-like noise.

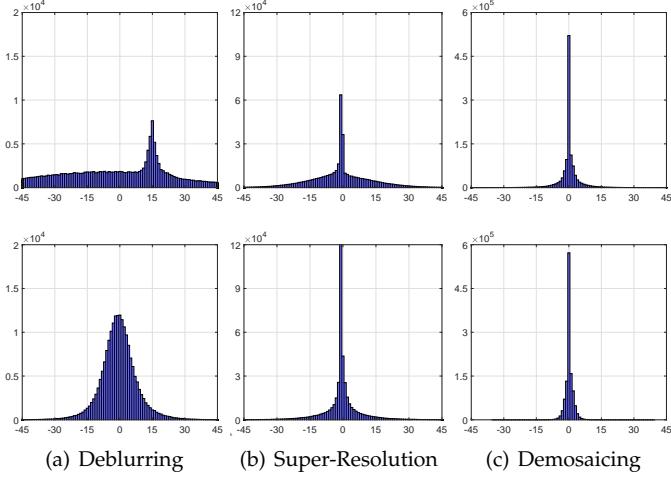


Fig. 15. Histogram of the noise (difference) between the ground-truth and input of the denoiser in the first iteration (first row) and last iteration (second row) for (a) deblurring, (b) super-resolution, and (c) demosaicing. The histograms are based on  $x_1$  and  $x_8$  in Fig. 8,  $x_1$  and  $x_{24}$  in Fig. 12 and  $x_1$  and  $x_{40}$  in Fig. 14.

According to the experiments and analysis, it can be concluded that the denoiser prior mostly removes the noise along with some fine details, while the subsequent data subproblem plays the role of alleviating the noise-irrelevant degradation and adding the lost details back. Such mechanisms actually enable the plug-and-play IR to be a generic method. However, it is worth noting that this comes at the cost of losing efficiency and specialization because of such general-purpose Gaussian denoiser prior and the manual selection of hyper-parameters. In comparison, deep unfolding IR can train a compact inference with better performance by jointly learning task-specific denoiser prior and hyper-parameters. Taking SISR as an example, rather than smoothing out the fine details by deep plug-and-play denoiser, the deep unfolding denoiser can recover the high-frequency details.

## 7 CONCLUSION

In this paper, we have trained flexible and effective deep denoisers for plug-and-play image restoration. Specifically, by taking advantage of half-quadratic splitting algorithm, the iterative optimization of three different image restoration tasks, including deblurring, super-resolution and color image demosaicing, consists of alternately solving a data subproblem which has a closed-form solution and a prior subproblem which can be replaced by a deep denoiser. Extensive experiments and analysis on parameter setting,

intermediate results, empirical convergence were provided. The results have demonstrated that plug-and-play image restoration with powerful deep denoiser prior have several advantages. On the one hand, it boosts the effectiveness of model-based methods due to the implicit but powerful prior modeling of deep denoiser. On the other hand, without task-specific training, it is more flexible than learning-based methods while having comparable performance. In summary, this work has highlighted the advantages of deep denoiser based plug-and-play image restoration. It is worth noting that there also remains room for further study. For example, one direction would be how to integrate other types of deep image prior, such as deep generative prior [97], for effective image restoration.

## 8 ACKNOWLEDGEMENTS

This work was partly supported in part by the ETH Zürich Fund (OK), in part by Huawei, in part by the National Natural Science Foundation of China under grant No. U19A2073, in part by Amazon AWS and in part by Nvidia grants.

## REFERENCES

- [1] W. H. Richardson, "Bayesian-based iterative method of image restoration," *JOSA*, vol. 62, no. 1, pp. 55–59, 1972.
- [2] H. C. Andrews and B. R. Hunt, "Digital image restoration," *Prentice-Hall Signal Processing Series, Englewood Cliffs: Prentice-Hall*, 1977, vol. 1, 1977.
- [3] S. Roth and M. J. Black, "Fields of experts," *International Journal of Computer Vision*, vol. 82, no. 2, pp. 205–229, 2009.
- [4] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *IEEE International Conference on Computer Vision*, 2011, pp. 479–486.
- [5] M. F. Tappen, "Utilizing variational optimization to learn markov random fields," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.
- [6] A. Barbu, "Training an active random field for real-time image denoising," *IEEE Transactions on Image Processing*, vol. 18, no. 11, pp. 2451–2462, 2009.
- [7] J. Sun and M. F. Tappen, "Separable markov random field model and its applications in low level vision," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 402–407, 2013.
- [8] U. Schmidt and S. Roth, "Shrinkage fields for effective image restoration," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2774–2781.
- [9] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1256–1272, 2016.
- [10] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2862–2869.
- [11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [12] D. Geman and C. Yang, "Nonlinear image recovery with half-quadratic regularization," *IEEE Transactions on Image Processing*, vol. 4, no. 7, pp. 932–946, 1995.
- [13] N. Parikh, S. P. Boyd *et al.*, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.
- [14] A. Danielyan, V. Katkovnik, and K. Egiazarian, "Image deblurring by augmented lagrangian with BM3D frame prior," in *Workshop on Information Theoretic Methods in Science and Engineering*, 2010, pp. 16–18.
- [15] F. Heide, M. Steinberger, Y.-T. Tsai, M. Rouf, D. Pajak, D. Reddy, O. Gallo, J. Liu, W. Heidrich, K. Egiazarian *et al.*, "FlexISP: A flexible camera image processing framework," *ACM Transactions on Graphics*, vol. 33, no. 6, p. 231, 2014.



- Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis.* Springer, 2018, pp. 277–284.
- [63] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, “Enhanced deep residual networks for single image super-resolution,” in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 136–144.
- [64] S. Mohan, Z. Kadkhodaie, E. P. Simoncelli, and C. Fernandez-Granda, “Robust and interpretable blind image denoising via bias-free convolutional neural networks,” in *International Conference on Learning Representations*, 2019.
- [65] K. Ma, Z. Duanmu, Q. Wu, Z. Wang, H. Yong, H. Li, and L. Zhang, “Waterloo exploration database: New challenges for image quality assessment models,” *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 1004–1016, 2017.
- [66] E. Agustsson and R. Timofte, “NTIRE 2017 challenge on single image super-resolution: Dataset and study,” in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, vol. 3, July 2017, pp. 126–135.
- [67] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *International Conference for Learning Representations*, 2015.
- [68] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.
- [69] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, “Residual non-local attention networks for image restoration,” in *International Conference on Learning Representations*, 2019.
- [70] X. Jia, S. Liu, X. Feng, and L. Zhang, “Focnet: A fractional optimal control network for image denoising,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6054–6063.
- [71] R. Franzen, “Kodak lossless true color image suite,” *source: http://r0k.us/graphics/kodak*, vol. 4, no. 2, 1999.
- [72] L. Zhang, X. Wu, A. Buades, and X. Li, “Color demosaicking by local directional interpolation and nonlocal adaptive thresholding,” *Journal of Electronic Imaging*, vol. 20, no. 2, p. 023016, 2011.
- [73] C. Dong, Y. Deng, C. C. Loy, and X. Tang, “Compression artifacts reduction by a deep convolutional network,” in *IEEE International Conference on Computer Vision*, 2015, pp. 576–584.
- [74] M. Ehrlich, L. Davis, S.-N. Lim, and A. Shrivastava, “Quantization guided jpeg artifact correction,” in *European Conference on Computer Vision*, 2020.
- [75] R. Timofte, R. Rothe, and L. Van Gool, “Seven ways to improve example-based single image super resolution,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1865–1873.
- [76] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, “Understanding and evaluating blind deconvolution algorithms,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1964–1971.
- [77] J. Kruse, C. Rother, and U. Schmidt, “Learning to push the limits of efficient fft-based image deconvolution,” in *IEEE International Conference on Computer Vision*, 2017, pp. 4586–4594.
- [78] H. Zhang, Y. Dai, H. Li, and P. Koniusz, “Deep stacked hierarchical multi-patch network for image deblurring,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5978–5986.
- [79] N. Efrat, D. Glasner, A. Apartsin, B. Nadler, and A. Levin, “Accurate blur models vs. image priors in single image super-resolution,” in *IEEE International Conference on Computer Vision*, 2013, pp. 2832–2839.
- [80] K. Zhang, X. Zhou, H. Zhang, and W. Zuo, “Revisiting single image super-resolution under internet environment: blur kernels and reconstruction algorithms,” in *Pacific Rim Conference on Multimedia*, 2015, pp. 677–687.
- [81] M. Irani and S. Peleg, “Motion analysis for image enhancement: Resolution, occlusion, and transparency,” *Journal of Visual Communication and Image Representation*, vol. 4, no. 4, pp. 324–335, 1993.
- [82] N. Zhao, Q. Wei, A. Basarab, N. Dobigeon, D. Kouamé, and J.-Y. Tourneret, “Fast single image super-resolution using a new analytical solution for  $\ell_2$ - $\ell_2$  problems,” *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3683–3697, 2016.
- [83] K. Zhang, L. Van Gool, and R. Timofte, “Deep unfolding network for image super-resolution,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3217–3226.
- [84] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, “Image super-resolution using very deep residual channel attention networks,” in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 286–301.
- [85] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, and W. Wu, “Feedback network for image super-resolution,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3867–3876.
- [86] J. W. Soh, S. Cho, and N. I. Cho, “Meta-transfer learning for zero-shot super-resolution,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3516–3525.
- [87] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, “Second-order attention network for single image super-resolution,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11065–11074.
- [88] H. S. Malvar, L.-w. He, and R. Cutler, “High-quality linear interpolation for demosaicing of bayer-patterned color images,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, 2004, pp. iii–485.
- [89] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand, “Deep joint demosaicking and denoising,” *ACM Transactions on Graphics*, vol. 35, no. 6, pp. 1–12, 2016.
- [90] L. Liu, X. Jia, J. Liu, and Q. Tian, “Joint demosaicing and denoising with self guidance,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2240–2249.
- [91] J. Wu, R. Timofte, and L. Van Gool, “Demosaicing based on directional difference regression and efficient regression priors,” *IEEE Transactions on Image Processing*, vol. 25, no. 8, pp. 3862–3874, 2016.
- [92] F. Kokkinos and S. Lefkimiatis, “Iterative joint image demosaicking and denoising using a residual denoising network,” *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 4177–4188, 2019.
- [93] Y. Guo, Q. Jin, G. Facciolo, T. Zeng, and J.-M. Morel, “Residual learning for effective joint demosaicing-denoising,” *arXiv preprint arXiv:2009.06205*, 2020.
- [94] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, “Non-local sparse models for image restoration,” in *IEEE International Conference on Computer Vision*, 2009, pp. 2272–2279.
- [95] W. Ye and K.-K. Ma, “Color image demosaicing using iterative residual interpolation,” *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5879–5891, 2015.
- [96] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi, “Beyond color difference: Residual interpolation for color image demosaicking,” *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1288–1300, 2016.
- [97] A. Brock, J. Donahue, and K. Simonyan, “Large scale gan training for high fidelity natural image synthesis,” in *International Conference on Learning Representations*, 2018.