Blind Image Deconvolution by Generative-based Kernel Prior and Initializer via Latent Encoding

Jiangtao Zhang[1], Zongsheng Yue[2], Hui Wang[1], Qian Zhao[1]✉, and Deyu Meng[1,3]

[1] Xi'an Jiaotong University, Shaanxi, China
[2] S-Lab, Nanyang Technological University, Singapore
[3] Pazhou Laboratory (Huangpu), Guangdong, China
zhangjt2021@stu.xjtu.edu.cn, zsyzam@gmail.com, huiwang2019@stu.xjtu.edu.cn, {timmy.zhaoqian, dymeng}@xjtu.edu.cn

Abstract. Blind image deconvolution (BID) is a classic yet challenging problem in the field of image processing. Recent advances in deep image prior (DIP) have motivated a series of DIP-based approaches, demonstrating remarkable success in BID. However, due to the high non-convexity of the inherent optimization process, these methods are notorious for their sensitivity to the initialized kernel. To alleviate this issue and further improve their performance, we propose a new framework for BID that better considers the prior modeling and the initialization for blur kernels, leveraging a deep generative model. The proposed approach pre-trains a generative adversarial network-based kernel generator that aptly characterizes the kernel priors and a kernel initializer that facilitates a well-informed initialization for the blur kernel through latent space encoding. With the pre-trained kernel generator and initializer, one can obtain a high-quality initialization of the blur kernel, and enable optimization within a compact latent kernel manifold. Such a framework results in an evident performance improvement over existing DIP-based BID methods. Extensive experiments on different datasets demonstrate the effectiveness of the proposed method.

Keywords: Generative kernel prior · Kernel initializer · DIP

1 Introduction

Blind image deconvolution (BID), also known as deblurring, is a classical problem in image processing [8–10, 14, 29, 34], which aims at recovering the latent clean image from the observed blurry counterpart. This blurry counterpart can be approximately modeled in mathematics as (assuming the blur process is uniform and spatially invariant)

$$\boldsymbol{y} = \boldsymbol{k} \otimes \boldsymbol{x} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{y}$ is the blurry image, $\boldsymbol{x}$ the underlying clean image, $\boldsymbol{k}$ the blur kernel, $\boldsymbol{n}$ usually the additive white Gaussian noise (AWGN), and $\otimes$ denotes the 2D convolution operator. Then the goal of BID is to estimate $\boldsymbol{x}$ under this degradation

Code is available at https://github.com/jtaoz/GKPILE-Deconvolution.

model without knowing the ground truth $\boldsymbol{k}$. As is well-known, this problem is highly ill-posed, since different pairs of $\boldsymbol{x}$ and $\boldsymbol{k}$ may result in the same $\boldsymbol{y}$.

The BID task was conventionally formulated as an optimization problem, corresponding to a Bayesian posterior estimation problem from the probabilistic perspective. Thus, the maximum a postriori (MAP) [28,29,41,45,46,55,62] and variational inference (VI) [2,14,70] approaches were developed for point-wise and distribution-wise estimation, respectively. The core challenge of these approaches is to properly design the priors for both the clean image [28, 41, 45, 46, 62] and blur kernel [42, 70]. However, these handcrafted priors are relatively subjective and may not precisely characterize the intrinsic distributions of natural images and blur kernels in the real world, which limits the performance of the deblurring algorithm. Besides, the resulting optimization is typically endowed with highly non-convex characteristics, leading to potential convergence issues such as reaching a trivial solution of delta kernel or unsatisfactory local minimum [34].

Owing to their wide and successful applications, deep learning technologies have been introduced to solve the BID problem. Early explorations [6, 16, 56] embedded deep neural networks (DNNs) into the traditional methods to facilitate the estimation of blur kernels. Recent works have pursued a more direct approach, wherein the mapping from a blurry image to its clean counterpart is directly learned [30,31,43,58,67,68]. The latter strategies have achieved promising results on several benchmarks, mainly attributed to the well-designed network architectures. However, these methods tend to overfit the training datasets, and thus may not generalize well to unseen scenarios, particularly for images with large and complex motion blur kernels.

Recently, deep image prior (DIP) [59] has exhibited great potential in addressing the task of BID [22,52], particularly under the scenario of motion blur. The core idea is to parameterize both the blur kernel and clean image as DNNs with random inputs, and then optimize these DNNs through an energy minimization problem conducted by Eq. (1). Compared with the traditional BID methods, these approaches employed more effective DNN-oriented priors and thus achieved superior performance. However, these methods did not fully explore the statistical characteristics of blur kernels, resulting in inaccurate kernel estimations, particularly in cases involving large kernel sizes. Additionally, since the optimization is highly non-convex, the random initialization of the blur kernel often leads to algorithmic instability and compromises the overall performance.

To alleviate the aforementioned issues of DIP-based methods, we proposed a new approach to better characterize and initialize the blur kernel with deep generative prior (DGP) [7, 48, 60, 64]. Specifically, focusing on the BID of motion blurring, we train a kernel generator using generative adversarial network (GAN) [18] as kernel prior. Owing to GAN's powerful fitting capability on complex distributions, the learned generator is expected to well characterize the blur kernels in a low-dimensional latent space. Then a mapping from the blurry image to the latent code of the kernel generator is trained, acting as a kernel initializer for the subsequent BID task. Intuitively, such a kernel initializer is easier to learn as the latent kernel space is more compact than that of the original kernel. To

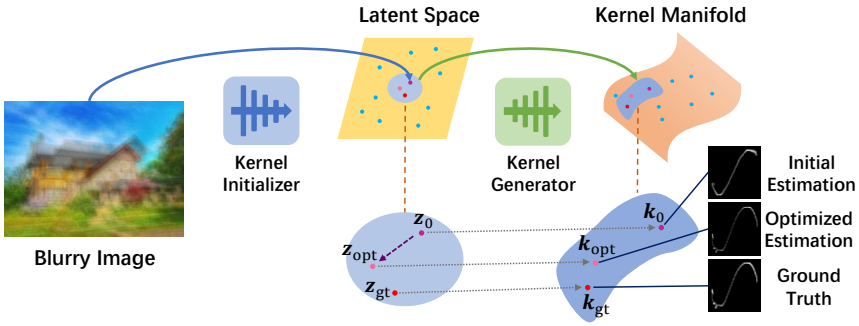

Fig. 1: Illustration of the latent encoding-based kernel initialization and fine-tuning mechanism in our BID framework. The blur kernel is first initialized from the blurry image by a pre-trained kernel initializer via the latent code $\boldsymbol{z}_0$, which is expected to be close to the code of the ground truth kernel, $\boldsymbol{z}_{\text{gt}}$. The corresponding kernels $\boldsymbol{k}_0$ and $\boldsymbol{k}_{\text{gt}}$ in the kernel manifold is also expected to be close. Then the optimization is performed within the latent space, such that $\boldsymbol{z}_0$ is fine-tuned to $\boldsymbol{z}_{\text{opt}}$, and the final estimated kernel $\boldsymbol{k}_{\text{opt}}$ is closer to the ground truth $\boldsymbol{k}_{\text{gt}}$. See Sec. 3 for more details.

solve the problem of BID, we first obtain a coarse initialization for the blur kernel by the initializer from the blurry image, and then jointly fine-tune the DIP for the image and the DGP for the blur kernel. The mechanism of the latent encoding-based kernel initialization and fine-tuning is illustrated in Fig. 1. Since the kernel generator sufficiently fits the kernel distribution and the kernel initializer provides a better initial estimation, the overall BID performance can be significantly improved with faster convergence, especially for large blur kernels.

Our main contributions are summarized as follows:

- We use GAN to learn the DGP for blur kernels, which can better depict the kernel structures and thus offer a more effective and compact kernel prior.
- We propose to learn a kernel initializer that maps from the blurry image to the latent code of the corresponding kernel. Attributed to the compactness of the latent kernel space, the proposed initializer can provide a more accurate kernel initialization for the subsequent BID process.
- By harnessing the designed kernel prior and initializer, our BID method achieves state-of-the-art (SotA) performance on the challenging benchmark dataset provided by Lai et al. [33], especially with large blur kernels.

2 Related work

2.1 Blind image deconvolution

BID methods can be roughly divided into two main categories, namely optimization-based methods and deep learning-based methods.

Optimization-based BID methods. Due to its high ill-posedness, the solution spaces of both the clean image and blur kernel should be properly constrained to make the BID problem solvable. To this aim, traditional optimization-based methods mainly focus on designing appropriate priors for clean images

and blur kernels, and plenty of studies have been conducted. For example, total variation [8], hyper-Laplacian prior [28], l_1/l_2-norm [29], and transform-specific l_1-norm [4, 62] were considered to model the image prior in the gradient domain, and patch-based priors [41, 57], low-rank prior [53] and dark/bright channel prior [46, 63] were directly applied in the image domain. As for blur kernels, the non-negative and normalization constraints result in a basic regularization. In addition, focusing on the properties of kernel, sparsity prior [47] and spectral prior [38] have been proposed. To get an ideal solution, tricks such as delayed normalization [50] and multi-scale implementation [71] for blur kernels are also suggested. However, these methods highly rely on handcrafted priors, which may not accurately characterize the intrinsic properties of the clean images and blur kernels, and thus are less competitive in this deep learning era.

Deep learning-based BID methods. Having witnessed great success in a wide range of applications, deep learning techniques have also been adopted for the BID task. Early methods attempted to replace certain components of the traditional optimization-based methods with DNNs, by virtue of their flexibility. For example, some researchers introduced DNNs as kernel predictors [6,16,56,65]. As the computing resources increase, it is then more popular to directly learn a mapping from the blurred image to its clean counterpart in a fully supervised way, with paired training data and well-designed DNNs [5,21,30,31,43,44,58,67, 68]. Though achieved SotA performance on several benchmark datasets, such as GoPro [43] and RealBlur [54], these methods may encounter the limitations in generalization to images with large complex blur kernels that are not simulated in the pre-defined training sets.

Recently, focusing on motion blurring, a new type of deep learning methods for the BID problem that leverages the deep priors (will be reviewed in the next subsection), has attracted increasing attention. Along this line, Ren et al. [52] proposed the SelfDeblur method, which adopted DIP to parameterize both the clean image and blur kernel for the first time. After that, multiple variants and extensions, mainly paying attention to the image prior, have been proposed for further improvement of performance [22,35]. These methods achieved promising results in certain cases, especially that they outperform fully supervised deep learning methods on the challenging benchmark by Lai et al. [33]. However, the properties of the blur kernel were not explicitly explored, and the performance tends to be unstable for large kernels due to the non-convex optimization. In contrast to these studies that used DIP with untrained DNNs, Asim et al. [1] proposed to pre-train generators for both the clean images and blur kernels as DGPs, and then fine-tune the generators when doing BID.

2.2 Deep prior for image processing

Due to their flexibility, DNNs have been used to characterize the image priors in recent years. There are mainly two kinds of such deep priors, namely DIP and DGP, which are briefly reviewed in the following.

Deep image prior. DIP was originally proposed by Ulyanov et al. [59] who trained a DNN to approximate the single target (maybe up to a transformation)

with random noise as its input. Since there are complex structures and operators involved, the DNN is expected to well depict the manifold of images, playing a similar role as the prior or regularizer in traditional image processing methods. After being raised, DIP has attracted remarkable attention and been applied to various image processing tasks (other than BID mentioned before), including natural image denoising [59], super-resolution [36, 66], inpainting [59], image decomposition [15], low-light enhancement [69], PET image reconstruction [17] and hyperspectral image denoising [39].

Deep generative prior. Attributed to their powerful generation capabilities, the trained deep generators, such as GANs [18], can be seen as approximators to the distributions of images, hoping to provide reasonable priors for images. Such priors can be referred to as DGPs [48]. After a pre-training stage for the generator, DGP can be used as an estimator to approximate the target image via fine-tuning like DIP. There are two common ways to fine-tune DGP. The first is to fix the generator and optimize the input. For example, Menon et al. [40] used this idea for photo upsampling. This strategy is also closely related to GAN-inversion [61]. The second is to jointly fine-tune the latent code and the parameters of the pre-trained generator like [48], which may better locate the optimal estimation within the image manifold.

In addition to directly modeling the images, DGP was also used to characterize the degradation operators, such as blur kernels. For example, Asim et al. [1] proposed to pre-train a variational auto-encoder (VAE) [26] as DGP for motion blur kernels for the BID task; and Liang et al. [36] used the normalizing flow (NF) [11, 12, 27] as the DGP for Gaussian blur kernels, applying to blind super-resolution. These studies highlight the effectiveness of DGP for blur kernels, and partially inspire our work.

3 Proposed method

3.1 DIP-based BID revisit

Before presenting our BID method, we first revisit DIP-based BID approaches, or more specifically SelfDeblur [52], and discuss the issues we try to address.

In SelfDeblur [52], the clean image $\boldsymbol{x}$ in Eq. (1) was approximated by a convolutional network equipped with encoder-decoder architecture, and the blur kernel $\boldsymbol{k}$ was estimated by a fully-connected network with one hidden layer. The resulting optimization can then be formulated as

$$\min_{\theta_k, \theta_x} \left\| G_k(\boldsymbol{z}_k; \theta_k) \otimes G_x(\boldsymbol{z}_x; \theta_x) - \boldsymbol{y} \right\|^2, \tag{2}$$

where $G_k(\cdot; \theta_k)$ and $G_x(\cdot; \theta_x)$ are networks for the kernel and image, respectively, and $\boldsymbol{z}_k$ and $\boldsymbol{z}_x$ are input random noise. Note that we ignore the additional constraints imposed on the blur kernel and clean image, positing that these constraints can be readily satisfied through thoughtful network design. Thus, the problem of estimating the blur kernel $\boldsymbol{k}$ and the clean image $\boldsymbol{x}$ is reduced to the optimization of parameters θ_k^* and θ_x^*, respectively.

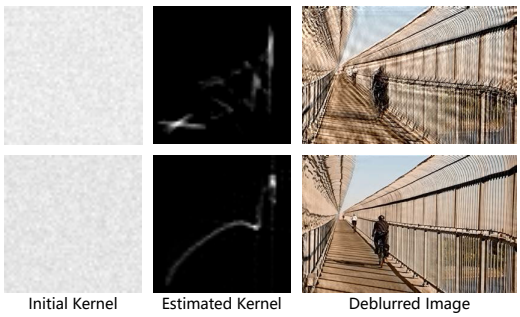

Fig. 2: Illustration of the initialization effect of the DIP-based BID. The two rows correspond to two independent runs of SelfDeblur [52]. From left to right: the randomly initialized kernel, the finally estimated kernel, and the deblurred image.

Owing to the powerful capabilities of DNNs, SelfDeblur obtained promising results on the BID task with motion blurring. Notably, it established a new SotA performance on the challenging benchmark introduced by Lai et al. [33]. Huo et al. [22] further improved SelfDeblur by incorporating a more sophisticated DIP within the variational Bayesian framework. However, these DIP-based BID methods did not fully explore the statistical structures or priors for blur kernels, thereby limiting their performance. A more important concern arises from the non-convexity of the optimization, where the randomness of $\boldsymbol{z}_k$ causes unstable solutions, particularly when the blur kernel is large. Fig. 2 shows a typical example with two different random runs. Despite the initializations following the same distribution, the estimated blur kernels can be diverse, consequently leading to significantly different deblurring results.

The aforementioned issues redirect our focus toward the blur kernels. Therefore, this study explores more precise statistical structures for blur kernels with DGP, followed by the development of a kernel initializer in the latent space.

3.2 Overview of the proposed method

Our proposed method consists of two stages. The first is the pre-training stage, aiming to learn a *kernel generator* and a *kernel initializer*. The second stage is to solve the deconvolution problem following the paradigm of the DIP-based BID method, while with the aid of the learned generator and initializer.

In the pre-training stage, we train a kernel generator $G_k(\cdot;\theta_k^*)$ within the GAN [18] framework, where θ_k^* denotes the generator parameters after training. Once learned, it can generate a blur kernel from the randomly sampled noise $\boldsymbol{z}_k$. This means the generator indeed models the distribution of the blur kernels and thus characterizes their statistical structures. Such a learned kernel generator can be referred to as the DGP of blur kernels, with $\boldsymbol{z}_k$ serving as the kernel latent code. Next, with the generator $G_k(\cdot;\theta_k^*)$ fixed, we train an encoder $E(\cdot;\theta_E^*)$ that maps a blurry image $\boldsymbol{y}$ to the corresponding kernel latent code through the technique of GAN-inversion [61]. The learned encoder allows for a relatively accurate prediction for the blur kernel based on its latent code, and thus acts as an effective initializer for the kernel DGP.

In the second stage, we first predict the kernel latent code corresponding to the input blurry image $\boldsymbol{y}$ using the pre-trained kernel initializer, which provides an initial value for the kernel DGP in the latent space. Then we jointly optimize

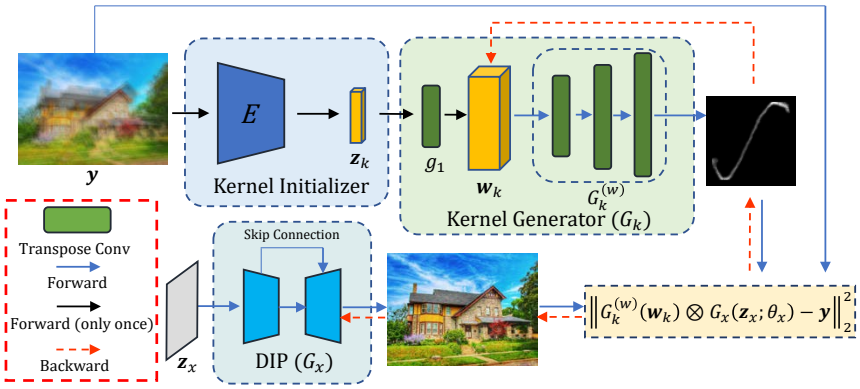

Fig. 3: Overview of the proposed BID process, with the pre-trained kernel generator and initializer, of our proposed method.

the parameters of DIP for the clean image and the latent code (or more precisely its mapped features as discussed in Sec. 3.5) of the DGP for the blur kernel. The overview of the proposed BID process is shown in Fig. 3.

In Sec. 3.3 and Sec. 3.4, we provide details for learning the kernel generator and kernel initializer, respectively. Then in Sec. 3.5, we discuss how to apply the pre-trained generator and initializer to the BID process.

3.3 Kernel generator learning

To learn the kernel generator, a large amount of motion blur kernels are synthesized to simulate their distribution based on the physical generation mechanism proposed in [30] or [33]. Then we train the kernel generator $G_k(\cdot;\theta_k^*)$ with these synthesized blur kernels. Note that our kernel generator is made up of several convolutional layers inspired by DCGAN [51]. Though can be seen as a straightforward application of GAN, this step is crucial for our method. On the one hand, the learned generator depicts the statistical structures of blur kernels, serving as a kernel DGP in the BID process; on the other hand, the learned kernel generator also forms the milestone for learning the kernel initializer in a low-dimensional latent space, which will be discussed in Sec. 3.4. Fig. 4 shows several blur kernels generated by $G_k(\cdot;\theta_k^*)$ and the physical model according to [33]. As can be observed, the blur kernels generated by our learned generator are morphologically very similar to that synthesized by the physical model, indicating its capability to characterize the underlying manifold of the blur kernels.

Remark. It should be mentioned there are some studies also considering learning DGP for blur kernels while with various generative models. For example, Asim et al. [1] used the VAE for the BID task, and Liang et al. [36] adopted the NF for the blind image super-resolution. In this work, we prefer GAN due to the following reasons. First, compared with the vanilla VAE that tends to generate blurry samples [3], GAN can provide sharper ones. Second, in contrast to NF, GAN imposes no constraints on the invertibility between the latent code and the generated sample, thereby facilitating easier learning of the kernel manifold in

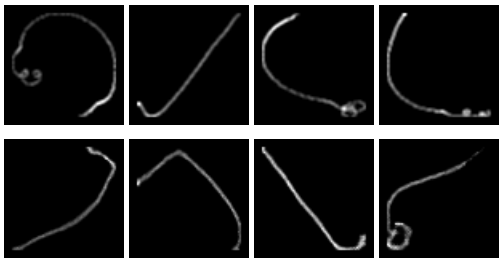

Fig. 4: Comparison of blur kernels synthesized according to physical model and generated by our pre-trained generator. Top row: blur kernels synthesized according to [33]. Bottom row: blur kernels generated by the pre-trained kernel generator.

a lower-dimensional latent space. This aspect is particularly advantageous when considering motion blur kernels, whose sizes can be relatively large (up to 75×75 in the Lai dataset [33]).

3.4 Kernel initializer learning

As discussed before, kernel initialization is important for the overall performance in the BID task since the optimization is highly non-convex. Therefore, we propose to initialize the blur kernel in the latent space using an encoder $E(\cdot;\theta_E^*)$ that maps the blurry image $\boldsymbol{y}$ to the latent code $\boldsymbol{z}$ of the learned kernel DGP.

A direct way to train the encoder is to solve the following optimization:

$$\theta_E^* = \arg\min_{\theta_E} \sum\nolimits_n \ell(G_k(E(\boldsymbol{y}_n;\theta_E);\theta_k^*), \boldsymbol{k}_n), \tag{3}$$

where $\boldsymbol{y}_n$ denotes the n-th blurry image, $\boldsymbol{k}_n$ the corresponding blur kernel, θ_k^* is the parameters of the pre-trained kernel generator, and $\ell(\cdot,\cdot)$ is some loss function. This optimization can be seen as training a predictor that directly estimates the blur kernel from the blurry image, which is not easy due to the relatively complex structures of blur kernels. On the other hand, if we can find the latent code $\boldsymbol{z}_n$ of $\boldsymbol{k}_n$, and learn to predict $\boldsymbol{z}_n$, the task could be simpler, since the latent space is expected to be much more compact than the original kernel space. Therefore, we propose to use a collaborative learning strategy motivated by [19], to guide the encoder training within the latent space. Specifically, we first formulate the following optimization to train the encoder:

$$\begin{aligned} &\min_{\theta_E} \sum_n \Big\{ \|G_k(E(\boldsymbol{y}_n;\theta_E);\theta_k^*) - G_k(\boldsymbol{z}_n;\theta_k^*)\|_1 + \lambda \|E(\boldsymbol{y}_n;\theta_E) - \boldsymbol{z}_n\|_2^2 \Big\}, \\ &s.t. \;\; \boldsymbol{z}_n = \arg\min_{\boldsymbol{z}} \; \|G_k(\boldsymbol{z};\theta_k^*) - \boldsymbol{k}_n\|_1, \end{aligned} \tag{4}$$

where λ is a tuning parameter set to 0.1 in this work. For convenience, we denote the objective functions of the outer and inner optimizations as $\mathcal{L}_E(\theta_E)$ and $\mathcal{L}_z^n(\boldsymbol{z})$, respectively, in the following.

The optimization of Eq. (4) involves a sub-task of solving $\boldsymbol{z}_n$ that corresponds to a GAN-inversion problem [61]. Following [19], we alternatively optimize $\boldsymbol{z}_n$ and θ_E in each iteration. Specifically, $\boldsymbol{z}_n$ is firstly initialized as $E(\boldsymbol{y}_n;\theta_E)$, and then updated by several gradient descent steps with respect to the lower-level objective $\mathcal{L}_z^n(\boldsymbol{z})$. Based on the updated $\boldsymbol{z}_n$, θ_E is subsequently optimized according to the upper-level objective $\mathcal{L}_E(\theta_E)$, also by gradient descent steps. The whole procedure is summarized in Algorithm 1.

Algorithm 1 Kernel initializer learning

Input: Pre-trained kernel generator $G_k(\cdot;\theta_k^*)$, blurry image-kernel pairs $\{\boldsymbol{y}_n, \boldsymbol{k}_n\}$, step size ϵ

Initialize $\theta_E^{(0)}$
for $t = 1, \cdots, T$ **do**
  For each n:
    $\boldsymbol{z}^{(0)} = E(\boldsymbol{y}_n; \theta_E^{(t-1)})$
    for $s = 1, \cdots, S$ **do** $\boldsymbol{z}^{(s)} = \boldsymbol{z}^{(s-1)} - \epsilon \nabla_{\boldsymbol{z}} \mathcal{L}_z^n(\boldsymbol{z})|_{\boldsymbol{z}=\boldsymbol{z}^{(s-1)}}$ **end for**
    $\boldsymbol{z}_n^{(t)} = \boldsymbol{z}^{(S)}$
  $\theta^{(0)} = \theta_E^{(t-1)}$
  for $l = 1, \cdots, L$ **do** $\theta^{(l)} = \theta^{(l-1)} - \epsilon \nabla_\theta \mathcal{L}_E(\theta)|_{\theta=\theta^{(l-1)}}$ **end for**
  $\theta_E^{(t)} = \theta^{(L)}$
end for

Output: Kernel initializer $E(\cdot;\theta_E^*)$, where $\theta_E^* = \theta_E^{(T)}$

Fig. 5: Illustration of the estimating ability of the pre-trained kernel initializer. Left: the blurry image. Top-right: the estimated blur kernel by the kernel initializer. Bottom-right: the ground-truth blur kernel.

Once trained, encoder $E(\cdot;\theta_E^*)$ can be regarded as an effective kernel initializer, providing a promising prediction of the blur kernel from the blurry image. Fig. 5 shows an example of the predicted kernel by the learned kernel initializer, in comparison with the ground truth. It can be observed that, though not perfect, the prediction is indeed very close to the ground truth, and thus is expected to be a good starting point for further processing.

3.5 BID Process

With the pre-trained kernel generator $G_k(\cdot;\theta_k^*)$ and initializer $E(\cdot;\theta_E^*)$, we can formulate the BID problem as follows:

$$(\boldsymbol{z}_k^*, \theta_x^*) = \arg\min_{\boldsymbol{z}_k, \theta_x} \|G_k(\boldsymbol{z}_k;\theta_k^*) \otimes G_x(\boldsymbol{z}_x;\theta_x) - \boldsymbol{y}\|^2, \tag{5}$$

where $\boldsymbol{z}_x$ is a random vector, $\boldsymbol{z}_k$ is initialized by $\boldsymbol{z}_k^{(0)} = E(\boldsymbol{y};\theta_E^*)$. After optimization, the desired clean image and blur kernel can be accessed via $\hat{\boldsymbol{x}} = G_x(\boldsymbol{z}_x;\theta_x^*)$ and $\hat{\boldsymbol{k}} = G_k(\boldsymbol{z}_k^*;\theta_k^*)$, respectively. In fact, such a naive implementation still cannot achieve satisfactory results as empirically illustrated in Sec. 4.3, in particular when the blur kernel is large. This can be attributed to the relatively low dimensionality of $\boldsymbol{z}_k$, compared with the blur kernel, which enforces too strong constraints and makes the optimization more difficult.

To address this issue, let's first recall one small yet effective trick in StyleGAN [23]. It introduced a network to map the latent code $\boldsymbol{z}$ to a style vector $\boldsymbol{w}$, enabling more precise control of the generated style. Inspired by this trick, we propose to optimize the feature map of the first layer in the kernel generator $G_k(\cdot;\theta_k^*)$, whose dimension is relatively higher, instead of the latent code $\boldsymbol{z}$. By

Algorithm 2 BID process with pre-trained kernel generator and initializer

Input: Pre-trained kernel generator $G_k(\cdot;\theta_k^*) = G_k^{(w)}(g_1(\cdot))$, pre-trained kernel initializer $E(\cdot;\theta_E^*)$, blurry image $\boldsymbol{y}$, random vector $\boldsymbol{z}_x$, step size ϵ

Initialize $\boldsymbol{w}_k^{(0)} = g_1(E(\boldsymbol{y};\theta_E^*))$, $\theta_x^{(0)}$, denote $\mathcal{L}(\boldsymbol{w},\theta) = \left\| G_k^{(w)}(\boldsymbol{w}) \otimes G_x(\boldsymbol{z}_x;\theta) - \boldsymbol{y} \right\|^2$

for $t = 1, \cdots, T$ **do**

  $\boldsymbol{w}_k^{(t)} = \boldsymbol{w}_k^{(t-1)} - \epsilon \nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w},\theta)|_{\boldsymbol{w}=\boldsymbol{w}_k^{(t-1)},\theta=\theta_x^{(t-1)}}$

  $\theta_x^{(t)} = \theta_x^{(t-1)} - \epsilon \nabla_{\theta} \mathcal{L}(\boldsymbol{w},\theta)|_{\boldsymbol{w}=\boldsymbol{w}_k^{(t-1)},\theta=\theta_x^{(t-1)}}$

end for

$\boldsymbol{w}_k^* = \boldsymbol{w}_k^{(T)}$, $\theta_x^* = \theta_x^{(T)}$

Output: Estimated clean image $\hat{\boldsymbol{x}} = G_x(\boldsymbol{z}_x;\theta_x^*)$, estimated blur kernel $\hat{\boldsymbol{k}} = G_k^{(w)}(\boldsymbol{w}_k^*)$

denoting the first layer of $G_k(\cdot;\theta_k^*)$ as $g_1(\cdot)$ and the reduced generator without $g_1(\cdot)$ as $G_k^{(w)}(\cdot)$, the BID problem of Eq. (5) is reformulated as

$$(\boldsymbol{w}_k^*, \theta_x^*) = \arg\min_{\boldsymbol{w}_k,\theta_x} \left\| G_k^{(w)}(\boldsymbol{w}_k) \otimes G_x(\boldsymbol{z}_x;\theta_x) - \boldsymbol{y} \right\|^2, \tag{6}$$

where $\boldsymbol{w}_k$ is initialized by $\boldsymbol{w}_k^{(0)} = g_1\left(E(\boldsymbol{y};\theta_E^*)\right)$. The whole BID process for solving Eq. (6) is summarized in Algorithm 2.

It should be noted that the parameters of the generator network can also be fine-tuned in previous studies [48]. However, due to the highly non-convex nature of the BID optimization, too many to-be-optimized parameters may hinder its performance. We thus only update $\boldsymbol{w}_k$ for the blur kernel in this work. Empirical results in Sec. 4.3 substantiate the superiority of this strategy over updating the whole kernel generator.

4 Experiments

In this section, we present experimental results to demonstrate the effectiveness of the proposed method and ablation studies to analyze some key components in our framework. More results can be found in the supplementary material, together with discussions on the limitations of this work.

4.1 Datasets and implementation details

Datasets. We evaluate our proposed framework mainly on two datasets: one is synthesized by ourselves, and the other is the well-known BID benchmark constructed by Lai et al. [33]. For the synthetic dataset, we randomly select 80 source images from MSCOCO [37] and apply the motion blur synthesized by [30] on them. The second Lai dataset consists of 25 clean images and 4 large-size blur kernels, resulting in 100 testing blur images in total. Beyond the two synthetic datasets, we also test our method real blurry images collected by Lai et al. [33].

Implementation details. As discussed in Sec. 3, our whole framework consists of a pre-training stage and a BID process. In the pre-training stage, we first train the kernel generator $G_k(\cdot;\theta_k^*)$ with DCGAN [51], using kernels synthesized following the way in [30] or [33]. Then we freeze the weights of $G_k(\cdot;\theta_k^*)$ and

Table 1: Quantitative comparisons of various methods on our synthetic dataset.

Method	PSNR	SSIM	Method	PSNR	SSIM
Pan et al. [46]	24.09	0.808	Zamir et al. [68]	22.27	0.652
Dong et al. [13]	23.85	0.799	Ren et al. [52]	25.48	0.765
Tao et al. [58]	23.21	0.699	Huo et al. [22]	28.13	0.898
Kupyn et al. [31]	23.46	0.716	Li et al. [35]	24.49	0.744
Kaufman and Fattal [24]	25.42	0.814	Ours	**29.67**	**0.928**

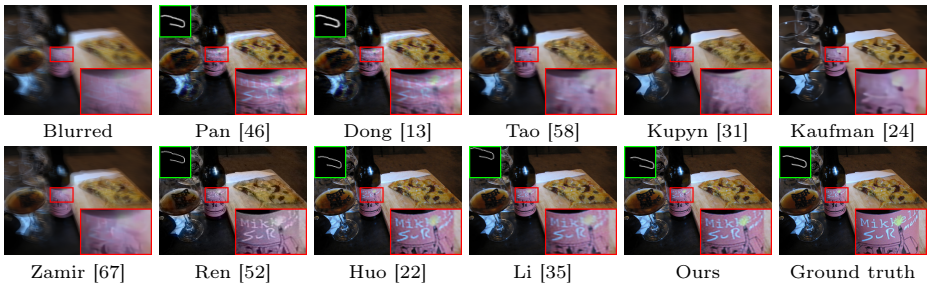

Fig. 6: Visual results on our synthetic dataset. The estimated blur kernel is placed on the top-left corner for each method if available.

train the kernel initializer $E(\cdot;\theta_E^*)$ that is a ResNet-18 [20]. To train the kernel initializer, we randomly crop 256×256 patches from the sources in OpenImages [32] and convolve them with blur kernels to get blurry images. The Adam optimizer [25] with its default configuration in PyTorch [49] is adopted. The initial learning rate is 1e-4 and reduced to 1e-5 as the loss becomes stable and finally to 1e-6. In the BID process, we jointly estimate the blur kernel $\boldsymbol{k}$ and the clean image $\boldsymbol{x}$ by optimizing $\boldsymbol{w}_k$ and θ_x according to Eq. (6). The initial learning rate of $\boldsymbol{w}_k$ and θ_x are set as 5e-4 and 1e-2, respectively, and decayed following [52]. For the architecture of the DIP network $G_x(\cdot;\theta_x)$, we follow the settings in SelfDeblur [52].

4.2 Experimental results

Results on our synthetic dataset. We first verify the effectiveness of the proposed method on the synthetic dataset by ourselves. Nine comparison methods are considered, including two traditional model-based ones (Pan et al. [46], Dong et al. [13]), four supervised deep learning ones (Tao et al. [58], Kupyn et al. [31], Kaufman and Fattal [24], Zamir et al. [68]), and three DIP-based ones (Ren et al. [52], Huo et al. [22], Li et al. [35]). We follow the default settings in their papers for these methods or tune them by ourselves for the best performance.

The quantitative results of all competing methods are summarized in Table 1. As can be seen from this table, the proposed method achieves the best performance in terms of both PSNR and SSIM, demonstrating its effectiveness. The superiority of our method can be visually observed from Fig. 6. Specifically, the traditional model-based methods are able to obtain accurate estimations for the blur kernel, but the BID results are not very satisfactory due to the insufficiency of the handcrafted image priors. Most of the supervised deep learning methods generate over-smoothed outcomes, mainly attributed to the lack of exploitation of the physical blur model. Even though achieving relatively better results, the

Table 2: Quantitative results of various methods on the dataset by Lai et al. [33].

Method	Manmade	Natural	People	Saturated	Text	Average
Cho and Lee [10]	16.35/0.389	20.14/0.520	19.90/0.556	14.05/0.493	14.87/0.443	17.06/0.480
Krishnan et al. [29]	15.73/0.408	19.44/0.526	21.50/0.647	14.09/0.507	15.40/0.485	17.23/0.514
Xu et al. [62]	17.99/0.599	21.58/0.679	24,40/0.813	14.53/0.538	17.64/0.668	19.23/0.659
Perrone and Favaro [50]	17.41/0.551	21.04/0.676	22.77/0.735	14.24/0.511	16.94/0.593	18.48/0.613
Pan et al. [46]	18.25/0.561	22.15/0.672	22.72/0.710	16.01/0.591	16.69/0.530	19.16/0.613
Dong et al. [13]	17.30/0.432	21.18/0.586	21.89/0.652	15.85/0.554	16.18/0.480	18.48/0.541
Tao et al. [58]	16.77/0.342	19.90/0.459	21.41/0.616	15.12/0.506	15.41/0.437	17.72/0.472
Kupyn et al. [31]	17.05/0.367	20.37/0.489	21.70/0.642	15.20/0.522	15.86/0.476	18.04/0.499
Kaufman and Fattal [24]	19.36/0.650	23.14/0.756	26.78/0.873	16.40/0.654	17.92/0.683	20.72/0.723
Zamir et al. [68]	16.73/0.341	19.80/0.447	20.88/0.587	15.14/0.505	14.49/0.383	17.41/0.453
Ren et al. [52]	19.00/0.617	22.94/0.715	26.43/0.781	18.38/0.688	22.82/0.776	21.91/0.715
Huo et al. [22]	22.51/0.820	26.20/0.901	31.38/0.953	18.14/0.691	27.72/0.933	25.19/0.860
Li et al. [35]	20.81/0.721	24.07/0.781	28.45/0.860	14.81/0.527	23.10/0.834	22.25/0.744
Ours	**24.67/0.942**	**28.12/0.969**	**32.86/0.973**	**18.98/0.737**	**28.91/0.951**	**26.71/0.914**

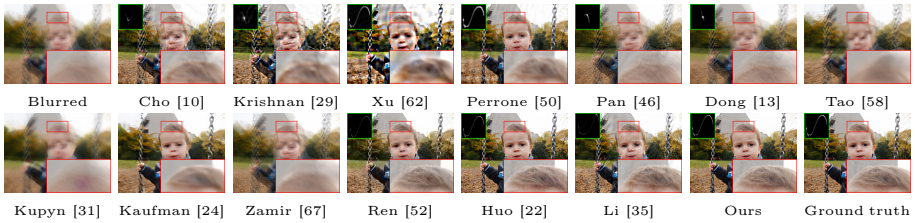

Fig. 7: Visual results on the dataset by Lai et al. [33]. The estimated blur kernel is placed on the top-left corner of each method if available.

existing DIP-based methods still have some limitations. For example, Ren et al.'s method performs unsatisfactorily, and the structural details in the images by Huo et al. and Li et al.'s methods are not sharp enough. In contrast to these methods, our method can not only accurately estimate the blur kernel, but also recover more image details with fewer distortions.

Results on the dataset by Lai et al. [33]. We evaluate our proposed method on the challenging BID benchmark by Lai et al. [33]. This dataset consists of 100 blurry images synthesized by applying 4 blur kernels, with sizes ranging from 31×31 to 75×75, to 25 clean images. These 25 clean images are categorized into 5 groups, namely *Manmade*, *Natural*, *People*, *Saturated* and *Text*. We compare our method with 13 existing methods, including six traditional model-based methods (Cho and Li [10], Krishnan et al. [29], Xu et al. [62], Perrone and Favaro [50], Pan et al [46], Dong et al. [13]), four supervised deep learning methods (Tao et al. [58], Kupyn et al. [31], Kaufman and Fattal [24], Zamir et al. [68]), and three DIP-based methods (Ren et al. [52], Huo et al. [22], Li et al. [35]).

Table 2 reports the quantitative results of all competing methods, in terms of PSNR and SSIM. We can see that our method significantly outperforms existing methods in all of the five categories. Notably, our method is the only one that achieves an SSIM value higher than 0.9 on average. The visual results on one typical example are shown in Fig. 7. It can be observed that most of the traditional model-based methods fail to estimate the blur kernel due to the relatively larger kernel size in this dataset. The supervised deep learning methods still produce distorted results as before, because they do not carefully consider the physical blur model. In comparison, DIP-based methods, including ours, can accurately estimate the blur kernel and further achieve promising deblurring results. Among them, our method recovers more details, such as hairs, with reference to the ground truth image, showing its superiority.

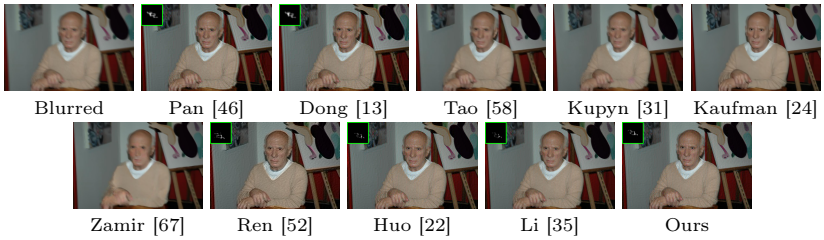

Fig. 8: Example results on the real blurry image provided by Lai et al. [33]. The estimated blur kernel is placed on the top-left corner of each method if available.

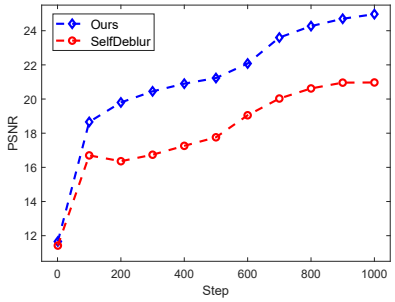

Fig. 9: Performance comparison (PSNR) of the DIP-based BID method (SelfDeblur [52]) and our method in different iteration steps on the dataset by Lai et al. [33].

Example results on real blurry images. We further apply comparison BID methods to real blurry images collected by Lai et al. [33], and show one example in Fig. 8. As can be seen, on this very challenging image, none of the methods generate perfect results, while our method provides a relatively promising one.

4.3 Ablation study

As shown in the previous experiments, the proposed method has demonstrated its effectiveness on various blurry images. In this section, we provide further analyses for better understanding its advantageous properties.

Convergence speed. One major methodological difference between our method and existing DIP-based methods is that we can initialize the blur kernel with pre-trained models instead of random sampling. This property can not only result in more stable and better performance but also improve the convergence speed. To verify this point, we compare the PSNR values of our methods with that of Ren et al.'s [52] across various iterations on the Lai dataset in Fig. 9. It can be seen that the performance of our method in the 400-th iteration is comparable with that of Ren et al.'s method in the 1000-th iteration. This faster convergence can be mostly attributed to the better initialization and characterization of the blur kernel, since both of them adopt the same DIP network for the image.

Optimization strategy for blur kernel. As discussed in Sec. 3.5, during the BID process, we optimize the higher-dimensional feature map $\boldsymbol{w}_k$ instead of the original latent code $\boldsymbol{z}_k$ for the blur kernel. Now we empirically show the different behaviors of the two optimization strategies. As a reference, the results by optimizing the entire kernel generator are also reported, which has a larger search space for optimization. The detailed comparison results on the Lai dataset are

Table 3: Performance comparison (PSNR/SSIM) of different kernel optimization strategy on the dataset by Lai et al. [33].

Kernel size	31	51	55	75	Average
$\boldsymbol{w}_k$	27.89/0.932	26.31/0.901	25.91/0.916	26.73/0.908	26.71/0.914
$\boldsymbol{z}_k$	27.11/0.927	23.46/0.830	22.75/0.825	22.43/0.763	23.94/0.836
$\boldsymbol{z}_k$ and θ_k	24.38/0.830	22.33/0.716	24.45/0.819	24.17/0.814	23.83/0.795

Table 4: Performance comparison (PSNR/SSIM) of different kernel initialization strategies on the dataset by Lai et al. [33].

Baseline	Random	$G(\boldsymbol{z}_0;\theta_k^*)$	$G(E(\boldsymbol{y};\theta_E^*);\theta_k^*)$
Metrics	19.51/0.563	23.00/0.743	26.71/0.914

summarized in Table 3. We can observe that optimizing the whole generator is not competitive due to the too-large search space. When the blur kernel is relatively small, the other two strategies have similar performance, while optimizing $\boldsymbol{w}_k$ is slightly better. As the blur kernel becomes larger, the performance by optimizing $\boldsymbol{z}_k$ significantly drops, while optimizing $\boldsymbol{w}_k$ still produces promising results. These observations demonstrate the reasonability of the proposed optimization strategy and also emphasize the importance of a proper search space.

Effect of kernel initialization. In our BID framework, a pre-trained encoder is employed to initialize the blur kernel. To show the necessity of such an initializer, we compare the proposed initialization strategy to two baselines. The first is the random initialization that randomly samples a latent code $\boldsymbol{z}$ for the blur kernel. The second is to initialize the kernel with all elements being the same, which can be seen as an "average" kernel, and then find the corresponding $\boldsymbol{z}_0$ by GAN-inversion. It should be noted that both of these two initialization strategies are independent of the blurry image. Table 4 lists the comparison results on the Lai dataset. We can see that the random initialization strategy produces the worst result since the randomly initialized kernel might largely deviate from the target without any constraint. The fixed "average" kernel performs better, and indeed already outperforms many other methods in Table 2. That's because this strategy, in some sense, can be regarded as an "average" approximation to any blur kernel. Not surprisingly, the proposed initialization with a pre-trained encoder obtains the best performance as the initialized kernel has been very close to the ground-truth one, as shown in Fig. 5. These results substantiate the significant benefits brought by the pre-trained kernel initializer for the BID task.

5 Conclusion

In this paper, we have proposed a new framework for the BID task, by virtue of deep generative model. Within this framework, we first pre-train a kernel generator as a DGP for blur kernels and a kernel initializer that can offer a well-initialized kernel. Then during the BID process, the blur kernel is initialized in the latent feature space and jointly optimized with the DIP network for the final result. Comprehensive experiments have been conducted and demonstrated the effectiveness of the proposed method. The ablation study verifies the necessity of each component in our framework. In the future, we will try to generalize our method to non-uniform blur modeling, overcoming its current limitations.

Acknowledgments

This work was supported in part by the NSFC Projects (No. 12226004, 62076196, 62331028, 62272375). The authors thank Prof. Jiangxin Dong for providing the code for generating the blur kernels with the type of Lai dataset.

References

1. Asim, M., Shamshad, F., Ahmed, A.: Blind image deconvolution using deep generative priors. IEEE Transactions on Computational Imaging (TCI) **6**, 1493–1506 (2020)
2. Bishop, T.E., Babacan, S.D., Amizic, B., Katsaggelos, A.K., Chan, T., Molina, R.: Blind image deconvolution: problem formulation and existing approaches. In: Blind Image Deconvolution, pp. 21–62. CRC press (2017)
3. Bredell, G., Flouris, K., Chaitanya, K., Erdil, E., Konukoglu, E.: Explicitly minimizing the blur error of variational autoencoders. In: Proceedings of International Conference on Learning Representations (ICLR) (2023)
4. Cai, J.F., Ji, H., Liu, C., Shen, Z.: Blind motion deblurring from a single image using sparse approximation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 104–111 (2009)
5. Cai, J., Zuo, W., Zhang, L.: Dark and bright channel prior embedded network for dynamic scene deblurring. IEEE Transactions on Image Processing (TIP) **29**, 6885–6897 (2020)
6. Chakrabarti, A.: A neural approach to blind motion deblurring. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 221–235 (2016)
7. Chan, K.C., Wang, X., Xu, X., Gu, J., Loy, C.C.: Glean: Generative latent bank for large-factor image super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14245–14254 (2021)
8. Chan, T.F., Wong, C.K.: Total variation blind deconvolution. IEEE Transactions on Image Processing (TIP) **7**(3), 370–375 (1998)
9. Chen, L., Fang, F., Wang, T., Zhang, G.: Blind image deblurring with local maximum gradient prior. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1742–1750 (2019)
10. Cho, S., Lee, S.: Fast motion deblurring. ACM Transactions on Graphics (TOG) **28**(5), 1–8 (2009)
11. Dinh, L., Krueger, D., Bengio, Y.: NICE: Non-linear Independent Components Estimation (Apr 2015)
12. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using Real NVP. In: Proceedings of International Conference on Learning Representations (ICLR) (2017)
13. Dong, J., Pan, J., Su, Z., Yang, M.H.: Blind image deblurring with outlier handling. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2478–2486 (2017)
14. Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T.: Removing camera shake from a single photograph. ACM Transactions on Graphics (TOG) **25**(3), 787–794 (2006)
15. Gandelsman, Y., Shocher, A., Irani, M.: "Double-DIP": Unsupervised image decomposition via coupled deep-image-priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11018–11027 (2019)

16. Gong, D., Yang, J., Liu, L., Zhang, Y., Reid, I., Shen, C., Van Den Hengel, A., Shi, Q.: From motion blur to motion flow: A deep learning solution for removing heterogeneous motion blur. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2319–2328 (2017)
17. Gong, K., Catana, C., Qi, J., Li, Q.: Pet image reconstruction using deep image prior. IEEE Transactions on Medical Imaging (TMI) **38**(7), 1655–1665 (2019)
18. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems (NeurIPS) (2014)
19. Guan, S., Tai, Y., Ni, B., Zhu, F., Huang, F., Yang, X.: Collaborative learning for faster stylegan embedding. arXiv preprint arXiv:2007.01758 (2020)
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)
21. Hradiš, M., Kotera, J., Zemcık, P., Šroubek, F.: Convolutional neural networks for direct text deblurring. In: Proceedings of British Machine Vision Conference (BMVC) (2015)
22. Huo, D., Masoumzadeh, A., Kushol, R., Yang, Y.H.: Blind image deconvolution using variational deep image prior. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **45**(10), 11472–11483 (2023)
23. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4401–4410 (2019)
24. Kaufman, A., Fattal, R.: Deblurring using analysis-synthesis networks pair. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5811–5820 (2020)
25. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of the International Conference for Learning Representations (ICLR) (2015)
26. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: Proceedings of International Conference on Learning Representations (ICLR) (2014)
27. Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. In: Advances in Neural Information Processing Systems (NeurIPS) (2018)
28. Krishnan, D., Fergus, R.: Fast image deconvolution using hyper-laplacian priors. In: Advances in Neural Information Processing Systems (NeurIPS) (2009)
29. Krishnan, D., Tay, T., Fergus, R.: Blind deconvolution using a normalized sparsity measure. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 233–240 (2011)
30. Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., Matas, J.: Deblurgan: Blind motion deblurring using conditional adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8183–8192 (2018)
31. Kupyn, O., Martyniuk, T., Wu, J., Wang, Z.: Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 8878–8887 (2019)
32. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., et al.: The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. International Journal of Computer Vision (IJCV) **128**(7), 1956–1981 (2020)
33. Lai, W.S., Huang, J.B., Hu, Z., Ahuja, N., Yang, M.H.: A comparative study for single image blind deblurring. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1701–1709 (2016)

34. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Understanding and evaluating blind deconvolution algorithms. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1964–1971 (2009)
35. Li, J., Wang, W., Nan, Y., Ji, H.: Self-supervised blind motion deblurring with deep expectation maximization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13986–13996 (2023)
36. Liang, J., Zhang, K., Gu, S., Gool, L.V., Timofte, R.: Flow-based kernel prior with application to blind super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10596–10605 (2021)
37. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 740–755 (2014)
38. Liu, G., Chang, S., Ma, Y.: Blind image deblurring using spectral properties of convolution operators. IEEE Transactions on Image Processing (TIP) **23**(12), 5047–5056 (2014)
39. Luo, Y.S., Zhao, X.L., Jiang, T.X., Zheng, Y.B., Chang, Y.: Hyperspectral mixed noise removal via spatial-spectral constrained unsupervised deep image prior. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (JSTARS) **14**, 9435–9449 (2021)
40. Menon, S., Damian, A., Hu, S., Ravi, N., Rudin, C.: Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2437–2445 (2020)
41. Michaeli, T., Irani, M.: Blind deblurring using internal patch recurrence. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 783–798 (2014)
42. Molina, R., Katsaggelos, A.K., Abad, J., Mateos, J.: A bayesian approach to blind deconvolution based on dirichlet distributions. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). vol. 4, pp. 2809–2812 (1997)
43. Nah, S., Hyun Kim, T., Mu Lee, K.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3883–3891 (2017)
44. Pan, J., Dong, J., Liu, Y., Zhang, J., Ren, J., Tang, J., Tai, Y.W., Yang, M.H.: Physics-based generative adversarial models for image restoration and beyond. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **43**(7), 2449–2462 (2021)
45. Pan, J., Hu, Z., Su, Z., Yang, M.H.: Deblurring text images via l0-regularized intensity and gradient prior. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2901–2908 (2014)
46. Pan, J., Sun, D., Pfister, H., Yang, M.H.: Blind image deblurring using dark channel prior. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1628–1636 (2016)
47. Pan, J., Sun, D., Pfister, H., Yang, M.H.: Deblurring images via dark channel prior. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **40**(10), 2315–2328 (2017)
48. Pan, X., Zhan, X., Dai, B., Lin, D., Loy, C.C., Luo, P.: Exploiting deep generative prior for versatile image restoration and manipulation. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **44**(11), 7474–7489 (2021)

49. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems (NeurIPS) (2019)
50. Perrone, D., Favaro, P.: Total variation blind deconvolution: The devil is in the details. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2909–2916 (2014)
51. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: Proceedings of International Conference on Learning Representations (ICLR) (2016)
52. Ren, D., Zhang, K., Wang, Q., Hu, Q., Zuo, W.: Neural blind deconvolution using deep priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3341–3350 (2020)
53. Ren, W., Cao, X., Pan, J., Guo, X., Zuo, W., Yang, M.H.: Image deblurring via enhanced low-rank prior. IEEE Transactions on Image Processing (TIP) **25**(7), 3426–3437 (2016)
54. Rim, J., Lee, H., Won, J., Cho, S.: Real-world blur dataset for learning and benchmarking deblurring algorithms. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 184–201 (2020)
55. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. ACM Transactions on Graphics (TOG) **27**(3), 1–10 (2008)
56. Sun, J., Cao, W., Xu, Z., Ponce, J.: Learning a convolutional neural network for non-uniform motion blur removal. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 769–777 (2015)
57. Sun, L., Cho, S., Wang, J., Hays, J.: Edge-based blur kernel estimation using patch priors. In: Proceedings of the IEEE International Conference on Computational Photography (ICCP). pp. 1–8 (2013)
58. Tao, X., Gao, H., Shen, X., Wang, J., Jia, J.: Scale-recurrent network for deep image deblurring. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8174–8182 (2018)
59. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Deep image prior. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9446–9454 (2018)
60. Wang, X., Li, Y., Zhang, H., Shan, Y.: Towards real-world blind face restoration with generative facial prior. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9168–9178 (2021)
61. Xia, W., Zhang, Y., Yang, Y., Xue, J.H., Zhou, B., Yang, M.H.: Gan inversion: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **45**(3), 3121–3138 (2022)
62. Xu, L., Zheng, S., Jia, J.: Unnatural l0 sparse representation for natural image deblurring. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1107–1114 (2013)
63. Yan, Y., Ren, W., Guo, Y., Wang, R., Cao, X.: Image deblurring via extreme channels prior. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4003–4011 (2017)
64. Yang, T., Ren, P., Xie, X., Zhang, L.: Gan prior embedded network for blind face restoration in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 672–681 (2021)
65. Yue, Z., Yong, H., Zhao, Q., Zhang, L., Meng, D., Wong, K.Y.K.: Deep variational network toward blind image restoration. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2024)

66. Yue, Z., Zhao, Q., Xie, J., Zhang, L., Meng, D., Wong, K.Y.K.: Blind image super-resolution with elaborate degradation modeling on noise and kernel. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2118–2128 (2022)
67. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H.: Restormer: Efficient transformer for high-resolution image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5728–5739 (2022)
68. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Multi-stage progressive image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14821–14831 (2021)
69. Zhao, Z., Xiong, B., Wang, L., Ou, Q., Yu, L., Kuang, F.: Retinexdip: A unified deep framework for low-light image enhancement. IEEE Transactions on Circuits and Systems for Video Technology (TCSVT) **32**(3), 1076–1088 (2022)
70. Zhou, X., Mateos, J., Zhou, F., Molina, R., Katsaggelos, A.K.: Variational dirichlet blur kernel estimation. IEEE Transactions on Image Processing (TIP) **24**(12), 5127–5139 (2015)
71. Zuo, W., Ren, D., Zhang, D., Gu, S., Zhang, L.: Learning iteration-wise generalized shrinkage–thresholding operators for blind deconvolution. IEEE Transactions on Image Processing (TIP) **25**(4), 1751–1764 (2016)