Blind Image Deblurring with Unknown Kernel Size and Substantial Noise

Anonymous Author(s)
Affiliation
Address
email

Abstract

Blind image deblurring (BID) has been extensively studied in computer vision and adjacent fields. Modern methods for BID can be grouped into two categories: single-instance methods that deal with individual instances using statistical inference and numerical optimization, and data-driven methods that train deep-learning models to deblur future instances directly. Data-driven methods can be free from the difficulty in deriving accurate blur models, but are fundamentally limited by the diversity and quality of the training data—collecting sufficiently expressive and realistic training data is a standing challenge. In this paper, we focus on single-instance methods that remain competitive and indispensable, and address the challenging setting **unknown kernel size and substantial noise**, failing state-of-the-art (SOTA) methods. We propose a practical BID method that is stable against both, **the first of its kind**. Also, we show that our method, a non-data-driven method, can perform on par with SOTA data-driven methods on similar data the latter are trained on, and can perform consistently better on novel data. This is an extended abstract based on a recently published journal article; we will point to our full article with all the details in the camera version.

1 Introduction

Image blur is mostly caused by *optical blur* and *motion blur* [10–17]. It is often coupled with noticeable sensory noise, e.g. when one images fast-moving objects in low-light environments. Thus, in the simplest form, image blur is often modeled as $\boldsymbol{y} = \boldsymbol{k} * \boldsymbol{x} + \boldsymbol{n}$, where $\boldsymbol{y}$ is the observed blurry and noisy image, and $\boldsymbol{k}$, $\boldsymbol{x}$, $\boldsymbol{n}$ are the blur kernel, clean image, and additive sensory noise, respectively. The notation $*$ here is linear convolution, which encodes the assumption that the blur effect is uniform over the spatial domain. Given $\boldsymbol{y}$ and $\boldsymbol{k}$, estimating $\boldsymbol{x}$ is called (non-blind) *deconvolution*, a linear inverse problem that is relatively easy to solve. However, in practice, $\boldsymbol{k}$—including its size and numerical value—is often unavailable. This leads to *blind deconvolution* (BD), where $\boldsymbol{k}$ and $\boldsymbol{x}$ are estimated together from $\boldsymbol{y}$. Over the past decades, a rich set of ideas have been developed to tackle BID and BD, evolving from *single-instance methods* that rely on analytical processing or statistical inference and numerical optimization to solve one instance each time, to modern *data-driven methods* that aim to train deep learning (DL) models to solve all future instances. The sequence of landmark review articles [11, 13–16, 18] chronicle these developments.

In this paper, we focus on single-instance methods for BID. Although recent data-driven methods have shown great promise, as statistical learning methods, their generalizability is intrinsically limited by the availability and diversity of training data [16, 18]. Therefore, single-instance methods will likely be a mainstay alongside data-driven methods for practical BID.

Prior single-instance methods for BID seem vague on three critical issues toward practicality (see Fig. 1 also): (1) *unknown kernel ($\boldsymbol{k}$) size*: Except for methods that directly estimate $\boldsymbol{x}$ only (e.g.,

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

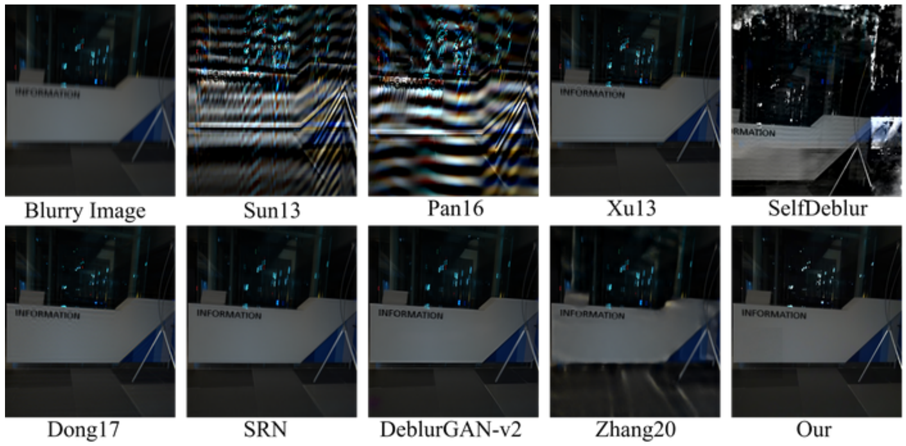

Figure 1: Deblurring results of several SOTA single-instance and data-driven BID methods on a real-world blurry image taken from [1]. The 6 single-instance methods are: Sun13 [2], Pan16 [3], Xu13 [4], Dong17 [5], SelfDeblur [6], and our method proposed in this paper; 3 data-driven methods are: SRN [7], DeblurGAN-v2 [8], Zhang20 [9], for which we directly take their pretrained models.

the inverse filtering approach to BD [19–21, 17]), a nearly-optimal estimate of the kernel size is needed [22]. But it is practically unclear how such an accurate estimate can be reliably obtained, and how sensitive the existing methods are to kernel-size misspecification; (2) *substantial noise* ($\boldsymbol{n}$): Sensory noise after convolution may still be substantial, while most previous methods assume noise-free or low-noise settings in their evaluations [23–25, 5, 26, 27]; and (3) *model stability*: The image may be blurry only, noisy only, or both. Whatever the case, in practice, an ideal BID method should work seamlessly across the different regimes. To quickly confirm these practicality issues, we pick 6 state-of-the-art (SOTA) single-instance BID methods (plus 3 representative data-driven methods by taking their pretrained models), and test them on a real-world image taken in a low-light setting, *with unknown kernel size and unknown noise type/level*. We specify a kernel size that is half of the image size in each dimension to provide a loose upper bound. Fig. 1 shows how miserably these single-instance methods can fail.

This paper aims to address these practicality issues. We follow the major modeling ideas in the statistical inference and optimization approach to BID, but parametrize both the kernel and the image using trainable structured deep neural networks (DNNs). This idea has recently been independently introduced to BID by [28], [6] (SelfDeblur), and [29], inspired by the remarkable success of deep image prior (DIP) [30] and its variants [31, 32] in solving a variety of inverse problems in computer vision and imaging [33, 34, 32, 35, 36] and beyond [37, 38]. Our key contributions include **(1) identifying three practicality issues of SOTA single-instance BID methods, including SelfDeblur**. As far as we are aware, this is the first time these three practicality issues have been discussed and addressed together in the BID literature. **BID with these three issues is a more difficult but practical version than what SelfDeblur and most classical BID methods target**. This is also the first time both classical and SOTA data-driven BID methods are systematically evaluated in the simultaneous presence of the three issues; (2) revamping SelfDeblur with six crucial modifications to address the three issues. In Section 2, we sketch our modifications, as well as the rationale and intuitions behind them. Figuring out these modifications and their right combination is a highly nontrivial task, making our algorithm pipeline sufficiently different from SelfDeblur. **(3) systematic evaluation of our method against SOTA single-instance BID methods on synthetic SOTA datasets, and against SOTA data-driven BID methods on real world datasets**, confirming the superior effectiveness and practicality of our method.

2 Our method to address the three practicality issues

SelfDeblur Deep image prior (DIP) hypothesizes that natural images, or, in general, natural visual objects, can be parameterized as the output of trainable DNNs [30]. Specifically, any visual object of interest, $\mathcal{O}$, is written as $\mathcal{O} = G_{\boldsymbol{\theta}}(\boldsymbol{z})$: $G_{\boldsymbol{\theta}}$ is a structured DNN (often convolutional DNN to have a bias toward natural visual structures) that can be thought of as a generator, and $\boldsymbol{z}$ is the seed (i.e., input) to $G_{\boldsymbol{\theta}}$. Often, $G_{\boldsymbol{\theta}}$ is trainable and $\boldsymbol{z}$ is randomly initialized and then fixed. So, when solving visual inverse problems using the typical regularized data-fitting formulation: $\min_{\mathcal{O}} \ \ell(\boldsymbol{y}, f(\mathcal{O})) + \lambda R(\mathcal{O})$, we can plug in the DIP parametrization in place of $\mathcal{O}$: $\min_{\boldsymbol{\theta}} \ \ell(\boldsymbol{y}, f \circ G_{\boldsymbol{\theta}}(\boldsymbol{z})) + \lambda R \circ G_{\boldsymbol{\theta}}(\boldsymbol{z})$, where the regularizer R that encodes other priors is sometimes omitted. This simple idea has fueled surprisingly competitive methods for solving numerous computational vision and imaging tasks [30, 31, 39, 28, 29, 34, 32, 35, 40, 41, 33, 42–46, 36]. When applying the DIP idea to BID, due to the asymmetric roles played by the kernel $\boldsymbol{k}$ and the image $\boldsymbol{x}$, it is natural to parameterize them separately following the Double-DIP idea [34] to obtain: $\min_{\boldsymbol{\theta_k}, \boldsymbol{\theta_x}} \ \ell(\boldsymbol{y}, G_{\boldsymbol{\theta_k}}(\boldsymbol{z_k}) * G_{\boldsymbol{\theta_x}}(\boldsymbol{z_x})) + \lambda_{\boldsymbol{k}} R_{\boldsymbol{k}} \circ G_{\boldsymbol{\theta_k}}(\boldsymbol{z_k}) + \lambda_{\boldsymbol{x}} R_{\boldsymbol{x}} \circ G_{\boldsymbol{\theta_x}}(\boldsymbol{z_x})$. This is the exact recipe followed by two previous works [28, 6]. SelfDeblur [6] takes the form

$$\min_{\boldsymbol{\theta_k}, \boldsymbol{\theta_x}} \ \|\boldsymbol{y} - G_{\boldsymbol{\theta_k}}(\boldsymbol{z_k}) * G_{\boldsymbol{\theta_x}}(\boldsymbol{z_x})\|_2^2 + \lambda_{\boldsymbol{x}} \|\nabla_{\boldsymbol{x}} G_{\boldsymbol{\theta_x}}(\boldsymbol{z_x})\|_1 \tag{1}$$

where $G_{\boldsymbol{\theta_k}}$ is a 2-layer MLP with a softmax final activation, and $G_{\boldsymbol{\theta_x}}$ is a convolutional. U-Net with sigmoid final activation. SelfDeblur works well only when $\boldsymbol{y}$ is blurry only and the kernel size is exactly specified. When there is considerable noise or the kernel-size is overspecified, SelfDeblur breaks down abruptly.

Algorithm 1 BID with unknown kernel size and substantial noise (uniform kernel)

Input: blurry and noisy image $\boldsymbol{y}$, kernel size $n_{\boldsymbol{k}} \times m_{\boldsymbol{k}}$ (default: $\lceil n_{\boldsymbol{y}}/2 \rceil \times \lceil m_{\boldsymbol{y}}/2 \rceil$), random seed $\boldsymbol{z_x}$ for $\boldsymbol{x}$, randomly initialized network weights $\boldsymbol{\theta}_{\boldsymbol{k}}^{(0)}$ and $\boldsymbol{\theta}_{\boldsymbol{x}}^{(0)}$, optimal image estimate $\boldsymbol{x}^* = G_{\boldsymbol{\theta}_{\boldsymbol{x}}^{(0)}}(\boldsymbol{z_x})$, regularization parameter $\lambda_{\boldsymbol{x}}$, iteration index $i = 1$, WMV-ES window size $W = 100$, WMV-ES patience number $P = 200$ (high noise) and $P = 500$ (low noise), WMV-ES empty queue $\mathcal{Q}$, WMV-ES $\mathrm{VAR}_{\min} = \infty$ (VAR: variance)

Output: estimated image $\widehat{\boldsymbol{x}}$

```
while not stopped do
    take an ADAM step to optimize Eq. (2) and obtain θ_k^(i), θ_x^(i), and x^(i) = G_{θ_x^(i)}(z_x)
    push x^(i) to Q, pop Q if |Q| > W
    if |Q| = W then
        compute VAR of elements inside Q
        if VAR < VAR_min then
            VAR_min ← VAR, x* ← x^(i)
        end if
    end if
    if VAR_min does not decrease over P iterations then
        exit and return x*
    end if
    i = i + 1
end while
extract x̂ of size n_y × m_y from x* using the sliding-window method
```

Six crucial modifications Given the blurry and noisy image $\boldsymbol{y} \in \mathbb{R}^{n_{\boldsymbol{y}} \times m_{\boldsymbol{y}}}$, we perform the following six steps: **(1)** we specify the kernel size as $n_{\boldsymbol{k}} \times m_{\boldsymbol{k}} = \lceil n_{\boldsymbol{y}}/2 \rceil \times \lceil m_{\boldsymbol{y}}/2 \rceil$ by default when the kernel size is unknown—surprisingly this aggressive over-specification does not hurt DIP and allows us to deal with the issue of unknown kernel size, and as given values when an estimate is available; **(2)** according to the property of linear convolution, we set the size of the image $\boldsymbol{x}$ as $(n_{\boldsymbol{y}} + n_{\boldsymbol{k}} - 1) \times (m_{\boldsymbol{y}} + m_{\boldsymbol{k}} - 1)$; **(3)** we choose ℓ as the Huber loss (with $\delta = 0.05$) and the scale-invariant ℓ_1/ℓ_2 regularizer to promote sparsity in the gradient domain. Both are shown to promote robustness to unknown noise type/level and reduce the sensitivity of $\lambda_{\boldsymbol{x}}$ to the noise level; **(4)** we choose the DIP model for the image, and the SIREN model [35] for the kernel—SIREN learns the typical high frequencies in the kernel better than DIP itself; **(5)** we employ the WMV-ES method [47] to resolve the overfitting issue due to the potential substantial noise—SelfDeblur does not consider this as they are evaluated only with very low noise; **(6)** we implement a sliding-window-based detection method to locate the estimated $\boldsymbol{x}$ over the over-specified image region—SelfDeblur does a simple central cropping. The size overspecification causes a shift symmetry between the kernel and the image, and the true image is not necessarily centered on the image region. Our complete BID pipeline is summarized in Algorithm 1; our double-DIP formulation reads

$$\min_{\boldsymbol{\theta_k}, \boldsymbol{\theta_x}} \ \ell_{\mathrm{Huber}}(\boldsymbol{y}, (\mathcal{D} \circ K_{\boldsymbol{\theta_k}}) * G_{\boldsymbol{\theta_x}}(\boldsymbol{z_x})) + \lambda_{\boldsymbol{x}} \|\nabla_{\boldsymbol{x}} G_{\boldsymbol{\theta_x}}(\boldsymbol{z_x})\|_1 / \|\nabla_{\boldsymbol{x}} G_{\boldsymbol{\theta_x}}(\boldsymbol{z_x})\|_2, \tag{2}$$

where $K_{\boldsymbol{\theta}_k}$ is a 2-layer MLP with 2 coordinate inputs and sigmoid output activation, $\mathcal{D}$ is a discretization operator that creates a discrete image out of a continuous one, and $G_{\boldsymbol{\theta}_x}$ is a convolutional U-Net with sigmoid final activation.

3 Experiments

We focus our comparison with SelfDeblur, and 3 SOTA data-driven methods, `SRN` [7], `DeblurGAN-v2` [8], and `ZHANG20` [9], on SOTA `NTIRE2020` and `RealBlur` BID datasets. Note that these data-driven methods directly predict sharp images from blurry images and hence bypass the problems caused by unknown kernel size and even inaccurate blur modeling [16]. Both `NTIRE2020` and

Table 1: Quantitative comparison of deblurring results on the 125 selected real-world images. For PSNR, SSIM, and VIF, higher the better. For LPIPS, lower the better. We report in the form of "mean (standard deviation)" (over the 125 images) for each method/metric combination. For each line, the first and second best numbers (according to the means) are marked in RED and GREEN, respectively.

		SRN	DeblurGAN-v2	ZHANG20	SelfDeblur	Ours
S1	PSNR	30.1 (1.159)	31.0 (1.149)	25.2 (1.188)	28.2 (1.198)	30.8 (1.168)
	SSIM	0.871 (0.0679)	0.883 (0.0609)	0.793 (0.0724)	0.832 (0.0734)	0.873(0.0618)
	VIF	0.784 (0.0686)	0.801 (0.0647)	0.705 (0.0705)	0.725 (0.0727)	0.796 (0.0651)
	LPIPS	0.972 (0.0966)	0.827 (0.08869)	1.025 (0.104)	0.987 (0.101)	0.821 (0.0879)
S2	PSNR	27.1 (1.256)	27.4 (1.352)	23.4 (1.449)	25.9 (1.471)	28.7 (1.236)
	SSIM	0.851 (0.0744)	0.859 (0.0695)	0.789 (0.0753)	0.821 (0.0758)	0.870 (0.0681)
	VIF	0.772 (0.0778)	0.783 (0.0758)	0.699 (0.0787)	0.713 (0.0777)	0.781 (0.0767)
	LPIPS	1.021 (0.116)	0.901 (0.0985)	1.076 (0.108)	1.001 (0.111)	0.811 (0.0947)
S3	PSNR	28.3 (1.197)	28.7 (1.139)	25.2 (1.236)	26.2 (1.227)	29.4 (1.144)
	SSIM	0.866 (0.0647)	0.867 (0.0608)	0.803 (0.0658)	0.827 (0.0637)	0.872 (0.0589)
	VIF	0.761 (0.0772)	0.787 (0.0727)	0.701 (0.0766)	0.731 (0.0776)	0.780 (0.0679)
	LPIPS	1.008 (0.0985)	0.869 (0.0936)	1.076 (0.107)	0.985 (0.110)	0.839 (0.0911)
S4	PSNR	26.7 (1.014)	27.1 (0.985)	23.3 (1.043)	25.8 (1.055)	28.5 (0.947)
	SSIM	0.849 (0.0542)	0.851 (0.0498)	0.780 (0.0567)	0.812 (0.0578)	0.861 (0.0481)
	VIF	0.756 (0.0621)	0.767 (0.0592)	0.687 (0.0663)	0.721 (0.0674)	0.776 (0.0574)
	LPIPS	1.015 (0.0941)	0.925 (0.0862)	1.050 (0.0927)	0.996 (0.0674)	0.893 (0.0848)
S5	PSNR	28.6 (1.352)	28.7 (1.314)	24.7 (1.410)	26.4 (1.400)	29.2 (1.284)
	SSIM	0.846 (0.0754)	0.855 (0.0694)	0.781 (0.0762)	0.818 (0.0771)	0.867 (0.0674)
	VIF	0.756 (0.0756)	0.771 (0.0754)	0.692 (0.0784)	0.710 (0.0793)	0.776 (0.0761)
	LPIPS	1.012 (0.1093)	0.874 (0.1085)	1.065 (0.1141)	0.992 (0.1149)	0.856 (0.0945)

`RealBlur` have their own strengths and limitations: images in `NTIRE2020` may contain multiple motions, but are captured in well-lit environments; `RealBlur` covers many dark scenes, but the scenes are static and relative motions are caused only by camera shakes. We select 125 representative, visually challenging images from the two datasets: for `NTIRE 2020`, we choose the most blurry frame from each folder that contains a sequence of consecutive frames; similarly, for `RealBlur`, we pick the most blurry from images about the same scene. The 125 images are grouped into 5 scenarios—25 images each: (S1) bright scene with high depth contrast; (S2) dark scene with high depth contrast; (S3) bright scene with low depth contrast; (S4) dark scene with low depth contrast; (S5) scene with high depth contrast and high brightness contrast.

Table 1 summarizes the quantitative results over the 125 selected images using the metrics: `PSNR`, `SSIM`, `VIF`, and `LPIPS`. Our method wins in most cases, followed by GAN-based `DeblurGAN-v2`. In fact, they are the top two in all cases. `DeblurGAN-v2` leads our method on `S1` by all metrics except for `LPIPS`, and on `S2` and `S3` only by `VIF`. This is likely because `S1` is sampled entirely from `NTIRE2020` that consists of bright scenes only, similar to the `GoPro` dataset that `DeblurGAN-v2` is trained on; only 10 out of 25 images from `S3` are from `NTIRE2020`. On `S2`, `S4`, and `S5` where each image consists of part of dark scenes, our method is a clear winner. This can be explained by the emphasis of the `RealBlur` dataset on dark scenes that have different distributions than `GoPro` that only includes bright scenes. **It is remarkable that our method, a non-data-driven method, can performs on par with SOTA data-driven methods on similar data the latter are trained on, and can perform consistently better on novel data**. The performance discrepancy of `DeblurGAN-v2` on different scenarios again underscores how data-driven methods can be limited by training data, although overall `DeblurGAN-v2` indeed shows reasonable generalizability to the novel dataset `RealDeblur`.

Numerous other details, comparisons, and analyses can be found in the full-length version of the paper, which we will link to in the camera-ready version.

References

[1] Jaesung Rim, Haeyun Lee, Jucheol Won, and Sunghyun Cho, "Real-world blur dataset for learning and benchmarking deblurring algorithms," in *European Conference on Computer Vision (ECCV)*, pp. 184–201. Springer International Publishing, 2020.

[2] Libin Sun, Sunghyun Cho, Jue Wang, and J. Hays, "Edge-based blur kernel estimation using patch priors," in *IEEE International Conference on Computational Photography (ICCP)*. apr 2013, IEEE.

[3] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang, "Blind image deblurring using dark channel prior," in *IEEE Conference on computer vision and pattern recognition (CVPR)*. jun 2016, IEEE.

[4] Li Xu, Shicheng Zheng, and Jiaya Jia, "Unnatural l0 sparse representation for natural image deblurring," in *IEEE Conference on computer vision and pattern recognition (CVPR)*. jun 2013, IEEE.

[5] Jiangxin Dong, Jinshan Pan, Zhixun Su, and Ming-Hsuan Yang, "Blind image deblurring with outlier handling," in *IEEE International conference on computer vision (ICCV)*. oct 2017, IEEE.

[6] Dongwei Ren, Kai Zhang, Qilong Wang, Qinghua Hu, and Wangmeng Zuo, "Neural blind deconvolution using deep priors," in *IEEE Conference on computer vision and pattern recognition (CVPR)*. jun 2020, IEEE.

[7] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia, "Scale-recurrent network for deep image deblurring," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8174–8182.

[8] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang, "Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8878–8887.

[9] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Bjorn Stenger, Wei Liu, and Hongdong Li, "Deblurring by realistic blurring," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2737–2746.

[10] Richard Szeliski, *Computer Vision: Algorithms and Applications*, Springer London, 2nd edition, 2021.

[11] D. Kundur and D. Hatzinakos, "Blind image deconvolution," *IEEE Signal Processing Magazine*, vol. 13, no. 3, pp. 43–64, may 1996.

[12] Neel Joshi, Richard Szeliski, and David J. Kriegman, "PSF estimation using sharp edge prediction," in *IEEE Conference on computer vision and pattern recognition (CVPR)*. jun 2008, IEEE.

[13] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding blind deconvolution algorithms," *IEEE Transactionson Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2354–2367, dec 2011.

[14] Rolf Köhler, Michael Hirsch, Betty Mohler, Bernhard Schölkopf, and Stefan Harmeling, "Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database," in *European Conference on Computer Vision (ECCV)*, pp. 27–40. Springer Berlin Heidelberg, 2012.

[15] Wei-Sheng Lai, Jia-Bin Huang, Zhe Hu, Narendra Ahuja, and Ming-Hsuan Yang, "A comparative study for single image blind deblurring," in *IEEE Conference on computer vision and pattern recognition (CVPR)*. jun 2016, IEEE.

[16] Jaihyun Koh, Jangho Lee, and Sungroh Yoon, "Single-image deblurring with neural networks: A comparative survey," *Computer Vision and Image Understanding*, vol. 203, pp. 103134, feb 2021.

[17] Qingyun Sun and David Donoho, "Convex sparse blind deconvolution," *arXiv:2106.07053*, June 2021.

[18] Kaihao Zhang, Wenqi Ren, Wenhan Luo, Wei-Sheng Lai, Björn Stenger, Ming-Hsuan Yang, and Hongdong Li, "Deep image deblurring: A survey," *International Journal of Computer Vision*, vol. 130, no. 9, pp. 2103–2130, jun 2022.

[19] Ralph A Wiggins, "Minimum entropy deconvolution," *Geoexploration*, vol. 16, no. 1-2, pp. 21–35, apr 1978.

[20] David Donoho, "ON minimum entropy deconvolution," in *Applied Time Series Analysis II*, pp. 565–608. Elsevier, 1981.

[21] Carlos A. Cabrelli, "Minimum entropy deconvolution and simplicity: A noniterative algorithm," *Geophysics*, vol. 50, no. 3, pp. 394–413, mar 1985.

[22] Li Si-Yao, Dongwei Ren, and Qian Yin, "Understanding kernel size in blind deconvolution," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*. jan 2019, IEEE.

[23] Yu-Wing Tai and S. Lin, "Motion-aware noise filtering for deblurring of noisy and blurry images," in *IEEE Conference on computer vision and pattern recognition (CVPR)*. jun 2012, IEEE.

[24] Lin Zhong, Sunghyun Cho, Dimitris Metaxas, Sylvain Paris, and Jue Wang, "Handling noise in single image deblurring using directional filters," in *IEEE Conference on computer vision and pattern recognition (CVPR)*. jun 2013, IEEE.

[25] Jinshan Pan, Zhouchen Lin, Zhixun Su, and Ming-Hsuan Yang, "Robust kernel estimation with outliers handling for image deblurring," in *IEEE Conference on computer vision and pattern recognition (CVPR)*. jun 2016, IEEE.

[26] Dong Gong, Mingkui Tan, Yanning Zhang, Anton van den Hengel, and Qinfeng Shi, "Self-paced kernel estimation for robust blind image deblurring," in *IEEE International conference on computer vision (ICCV)*. oct 2017, IEEE.

[27] Liang Chen, Faming Fang, Jiawei Zhang, Jun Liu, and Guixu Zhang, "OID: Outlier identifying and discarding in blind image deblurring," in *European Conference on Computer Vision (ECCV)*, pp. 598–613. Springer International Publishing, 2020.

[28] Zhunxuan Wang, Zipei Wang, Qiqi Li, and Hakan Bilen, "Image deconvolution with deep image and kernel priors," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. oct 2019, IEEE.

[29] Phong Tran, Anh Tran, Quynh Phung, and Minh Hoai, "Explore image deblurring via encoded blur kernel space," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[30] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky, "Deep image prior," *International Journal of Computer Vision*, vol. 128, no. 7, pp. 1867–1888, mar 2020.

[31] Reinhard Heckel and Paul Hand, "Deep decoder: Concise image representations from untrained non-convolutional networks," in *International Conference on Learning Representations*, 2019.

[32] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein, "Implicit neural representations with periodic activation functions," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[33] Mohammad Zalbagi Darestani and Reinhard Heckel, "Accelerated MRI with un-trained neural networks," *IEEE Transactions on Computational Imaging*, vol. 7, pp. 724–733, 2021.

[34] Yosef Gandelsman, Assaf Shocher, and Michal Irani, ""double-DIP": Unsupervised image decomposition via coupled deep-image-priors," in *IEEE Conference on computer vision and pattern recognition (CVPR)*. jun 2019, IEEE.

[35] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," in *Advances in Neural Information Processing Systems*, 2020.

[36] Adnan Qayyum, Inaam Ilahi, Fahad Shamshad, Farid Boussaid, Mohammed Bennamoun, and Junaid Qadir, "Untrained neural network priors for inverse imaging problems: A survey," *TechRxiv*, mar 2021.

[37] Sriram Ravula and Alexandros G. Dimakis, "One-dimensional deep image prior for time series inverse problems," *arXiv:1904.08594*, Apr. 2019.

[38] Michael Michelashvili and Lior Wolf, "Speech denoising by accumulating per-frequency modeling fluctuations," *arXiv:1904.07612*, Apr. 2019.

[39] Reinhard Heckel and Mahdi Soltanolkotabi, "Denoising and regularization via exploiting the structural bias of convolutional generators," *arXiv preprint arXiv:1910.14634*, 2019.

[40] Xudong Ma, Paul Hill, and Alin Achim, "Unsupervised image fusion using deep image priors," *arXiv:2110.09490*, Oct. 2021.

[41] Francis Williams, Teseo Schneider, Claudio Silva, Denis Zorin, Joan Bruna, and Daniele Panozzo, "Deep geometric prior for surface reconstruction," *arXiv:1811.10943*, 2019.

[42] Hannah Lawrence, David Bramherzig, Henry Li, Michael Eickenberg, and Marylou Gabrié, "Phase retrieval with holography and untrained priors: Tackling the challenges of low-photon nanoscale imaging," *arXiv preprint arXiv:2012.07386*, 2020.

[43] Emrah Bostan, Reinhard Heckel, Michael Chen, Michael Kellman, and Laura Waller, "Deep phase decoder: self-calibrating phase microscopy with an untrained deep neural network," *Optica*, vol. 7, no. 6, pp. 559–562, 2020.

[44] Kshitij Tayal, Raunak Manekar, Zhong Zhuang, David Yang, Vipin Kumar, Felix Hofmann, and Ju Sun, "Phase retrieval using single-instance deep generative prior," in *OSA Optical Sensors and Sensing Congress 2021 (AIS, FTS, HISE, SENSORS, ES)*. 2021, OSA.

[45] Kevin C. Zhou and Roarke Horstmeyer, "Diffraction tomography with a deep image prior," *Optics Express*, vol. 28, no. 9, pp. 12872, apr 2020.

[46] Zhong Zhuang, David Yang, Felix Hofmann, David Barmherzig, and Ju Sun, "Practical phase retrieval using double deep image priors," *arXiv preprint arXiv:2211.00799*, 2022.

[47] Hengkang Wang, Taihui Li, Zhong Zhuang, Tiancong Chen, Hengyue Liang, and Ju Sun, "Early stopping for deep image prior," *arXiv:2112.06074*, Dec. 2021.