Deep Learning for Handling Kernel/model Uncertainty in Image Deconvolution

Yuesong Nan and Hui Ji
Department of Mathematics, National University of Singapore, 119076, Singapore
nanyuesong@u.nus.edu and matjh@nus.edu.sg

Abstract

Most existing non-blind image deconvolution methods assume that the given blurring kernel is error-free. In practice, blurring kernel often is estimated via some blind deblurring algorithm which is not exactly the truth. Also, the convolution model is only an approximation to practical blurring effect. It is known that non-blind deconvolution is susceptible to such a kernel/model error. Based on an error-in-variable (EIV) model of image blurring that takes kernel error into consideration, this paper presents a deep learning method for deconvolution, which unrolls a total-least-squares (TLS) estimator whose relating priors are learned by neural networks (NNs). The experiments showed that the proposed method is robust to kernel/model error. It noticeably outperformed existing solutions when deblurring images using noisy kernels, e.g. *the ones estimated from existing blind motion deblurring methods.*

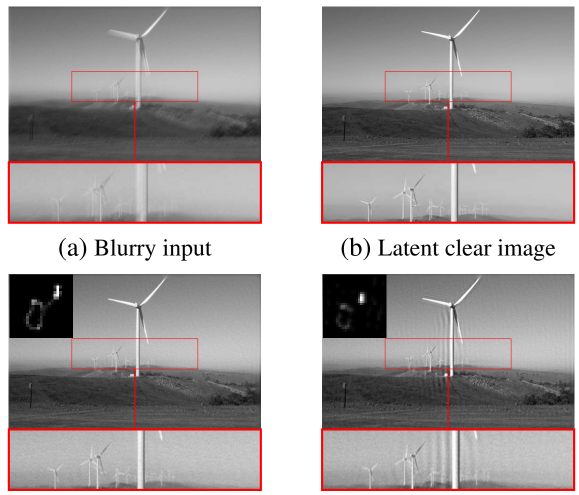
1. Introduction

Image blurring is one prime loss of image quality in practice, which often is modeled by a convolution process:

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

where $\boldsymbol{y}$ denotes the blurred image, $\boldsymbol{x}$ denotes the latent image, $\boldsymbol{k}$ represents the kernel and $\boldsymbol{n}$ represents noise. The operator '$\otimes$' stands for the discrete convolution. Image deconvolution is about recovering $\boldsymbol{x}$ from $\boldsymbol{y}$ by solving (1). Depending on the availability of $\boldsymbol{k}$, image deconvolution can be classified into (1) *blind image deblurring* which needs to estimate both $\boldsymbol{k}$ and $\boldsymbol{x}$, and (2) *non-blind image deconvolution* which takes $\boldsymbol{k}$ as input and only estimates $\boldsymbol{x}$. For example, removing motion blur from images is a typical blind image deblurring problem whose blur kernel needs to be estimated for individual image.

In the past, many methods have been proposed for tackling blind motion deblurring; see *e.g.* [12, 3, 23, 7, 45, 20, 40, 47, 31, 30, **?**]. Most of them take an iterative scheme to alternatively estimate the kernel $\boldsymbol{k}$ and the latent image $\boldsymbol{x}$. Although non-blind image deconvolution is called inside the iteration, the iteration focuses on the estimation of the kernel $\boldsymbol{k}$. For such a purpose, it is actually a good practice to only recover partial salient structures in those intermediate estimates of image $\boldsymbol{x}$; *e.g.* [7, 45, 47, 30]. At last, once the kernel is determined, an non-blind deconvolution method is called to recover the image $\boldsymbol{x}$ with all details.



(a) Blurry input (b) Latent clear image

(c) Result from truth kernel (d) Result from estimated kernel

Figure 1: Sensitivity of image deconvolution to kernel error. (a)-(b): Noisy blurred input ($\sigma = 1\%$) and the truth sharp image; (c)-(d): The results when using the non-blind deconvolution method Krishnan and Fergus [19] to deblur the input shown in (a) using truth kernel and the kernel estimated by Cho and Lee [7].

1.1. Deconvolution is sensitive to kernel/model error

Blind motion deblurring remains a challenge problem, and motion-blur kernel estimated by existing methods is hardly error-free. Furthermore, convolution-based model of motion-blurring holds true only if scene depths are roughly constant and camera movement is the translation on image plane. Same as image noise, without specific treatment, the kernel/model error will cause severe artifacts in the result.

Re-writing (1) in the form of matrix-vector, we have

$$y = Kx + n.$$

where y, x, n denotes $\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{n}$ in the form of column-wise vector, and K denotes the toepliz matrix representing convolution. Suppose that $\widehat{\boldsymbol{k}}$ is an inexact estimate of the truth kernel from some existing method with $\widehat{\boldsymbol{k}} = \boldsymbol{k} + \boldsymbol{\Delta_k}$. Then,

$$y = Kx + n = (\widehat{K} - \Delta_K)x + n. \quad (2)$$

Note that Δ_K can be viewed either as the convolution matrix w.r.t. kernel error $\boldsymbol{\Delta_k}$, or viewed as model error when motion blur is not exactly an uniform convolution. A direct inversion then leads to

$$\widehat{x} = (\widehat{K})^{-1} y = (K - \Delta_K)^{-1}(Kx + n)$$

By Taylor expansion, we have then

$$\begin{aligned} \hat{x} &= (K - \Delta_K)^{-1}(Kx + n) \\ &= (I + K^{-1}\Delta_K)(x + K^{-1}n) + O(\|\Delta_K\|_F^2) \\ &= x + \underbrace{K^{-1}(\Delta_K x)}_{\text{Kernel error}} + \underbrace{(I + K^{-1}\Delta_K)(K^{-1}n)}_{\text{measurement noise}} + O(\|\Delta_K\|_F^2) \end{aligned} \quad (3)$$

As K is an ill-conditioned matrix, same as image noise, the term caused by the kernel error Δ_K will also be significantly magnified.

Most non-blind deblurring methods impose additional regularization on image to suppressing noise amplification. These regularizations do not effectively suppress the artifacts caused by kernel/model error, *i.e.* $K^{-1}\Delta_K x$. See Fig 1 for an illustration of deblurring an image using the ℓ_1-norm relating regularization method [19], where the kernel is estimated by the blind deblurring method [7]. It can be seen that there are strong ringing artifacts in the results. In other words, there is such a need to study non-blind deconvolution method that is robust to kernel/model error, which can see its practical usage in blind image deblurring.

1.2. Main idea

This paper aims at developing a powerful image deconvolution method that is robust to likely kernel/model errors. Such robustness is important when solving many image restoration problems in practice, especially blind motion deblurring. Image blurring model considered in this paper is as follows,

$$y = (\widehat{K} - \Delta_K)x + n = \widehat{K}x - \Delta_K x + n, \quad (4)$$

where y is an input blurred image, x is the latent clear image to be recovered, $\widehat{K}$ is the matrix form of the 2D convolution operator w.r.t. the estimated blur kernel $\widehat{\boldsymbol{k}}$. There are two noise sources:

1. Measurement noise n, which is assumed to be additive Gaussian white noise as most do;
2. Model error Δ_K, introduced by either kernel error from blind deblurring algorithm or by modeling error when blurring is not exactly uniform.

The problem (4) is the so-called Error-in-Variable (EIV) model [5] in statistical regression. In the case that the matrix K is well-posed, the total least squares (TLS) estimator [14] estimates the solution to (4) by solving a constrained optimization problem:

$$\min_{\Delta_K, n, x} \|\Delta_K\|_F^2 + \|n\|_2^2, \quad \text{s.t.} \quad (\widehat{K}x - \Delta_K x = y - n).$$

In the case of image deconvolution, as the matrix K is ill-conditioned, certain prior needs to be imposed on x to suppress noise amplification in the standard TLS estimator. Consider a variable u that represents the error term $\Delta_K x$. Then, we propose to formulate the problem (4) as an optimization problem:

$$\min_{x,u} \|y - Kx - u\|_2^2 + \phi(x) + \psi(u|x), \quad (5)$$

where

$$\psi(u|x) = \min_{\Delta_k \in \Omega} \|\Delta_k\|_F^2 + \lambda \|u - \Delta_K x\|_2^2.$$

The feasible set Ω for Δ_k denote structure prior for the matrix Δ_K. For instance, the set of doubly Toeplitz matrices w.r.t. kernel error. It can be seen that there are two regularization terms in (5). The term $\phi(\cdot)$ denotes the regularization term defined by the image prior imposed on x. Another term $\psi(\cdot|x)$ denotes the regularization term on u that is related to x.

In the presence of kernel/model error, how close the solution of (5) to the truth x largely depends on the design of two regularization terms ϕ and ψ. The first is based on the image prior of natural images, and the second is based on the prediction of the correction term u. Recent development on deep-NN-based image restoration methods showed the effectiveness of deep NN as the tool to learn complex prior on image data, see *e.g.* [39, 46, 34, 48, 21, 27, 17, 2]. Thus, by learning ϕ, ψ, this paper proposed a deep learning method for non-blind deconvolution with specific treatment on kernel/model error.

In brief, the proposed method unrolls an iterative optimization algorithm for solving (5), and replaces the processes relating to two regularization terms ϕ, ψ by learnable NN-based mappings. At each stage of the proposed method, there are two NNs involved. One is a deep convolutional neural network (CNN) for removing undesirable artifacts caused by measurement noise and model error. The other is a deep U-Net for predicting correction term $\boldsymbol{u}$. Such a deep learning method not only effectively addresses measurement noise, but also handles kernel/noise error well, when recovering a noisy blurred image using some estimated blur kernel.

1.3. Main contribution

In practical blind image deblurring, especially blind motion deblurring, kernel/model error is the main factor account for poor recovery quality. Despite its practical importance, the robustness to kernel/model error did not receive sufficient attention in the development of non-blind deblurring method, see the short list in existing literature [16, 43, 32, 33].

This paper presented a deep learning based approach for image deconvolution with the focus on handling kernel/model error. See the below for the summary of the main contributions of the paper.

- The paper analyzed the impact of kernel/model error to image deconvolution in the context of EIV model, and proposed a TLS-based optimization model for addressing model error
- Built on an iterative scheme for solving the model, a deep learning method is presented that unrolls the scheme with deep-NN-based priors on both images and correction terms
- Training samples play an important role in the generalization capacity of deep-NN-based methods. This paper presents a new method for synthesizing motion-blur kernels which provides high-quality training samples.

Extensive experiments are conducted in this paper, which shows that the proposed method can effectively handle model/kernel errors when being used for deblurring images using the kernel from existing blind deblurring methods. The proposed NN can be trained using the training samples synthesized by the proposed procedure to outperform existing related methods by a noticeable margin. In summary, the proposed image deconvolution method provides a better solution than existing ones on handling kernel/model error. The work certainly can see its value to many image restoration tasks, including blind image deblurring.

2. Related Work

Owing to space limitation, we give a detailed discussion on the methods focusing on handling kernel/model error, while having a very brief review on the methods focusing on noise robustness.

2.1. Image deblurring focusing on noise robustness

The robustness to noise in deblurring comes from the regularization on latent image, derived from certain image prior assumed by the method. Earlier linear methods, *e.g.* Wiener filtering and Tikhonov regularization, assume smoothness prior on latent image. Non-linear methods assume that image gradients follow certain heavy-tailed distributions. For example, the ℓ_1-norm relating regularization, including total-variation (TV) method [29]) and wavelet methods [3, **?**], assumes sparsity prior of image in gradient/wavelet domain. The ℓ_p-norm based method [19] assumes hyper-Laplacian prior on image gradients. Non-local methods, *e.g.* [9, 11, **?**], assume recurrence prior of image patches.

In recent years, learning-based methods emerges as a promising approach which learns image prior from data. See *e.g.* [35, 51, 38, 39, 46, 34, 21, 27, 50, 48, 17, 2]. Roth and Black [35], Zoran and Weiss [51], and Schmidt and Roth [38] proposed to learn the parameters of some statistical model on images or image patches for characterizing images. For deep learning based non-blind deblurring methods, one approach is directly applied NN to map blurred image to latent image, including Schuler *et al*. [39], Xu *et al*. [46], and Ren *et al*. [34]. A more prevalent class of NN-based methods is based the so-called optimization unrolling, which follows the iterative scheme by solving some regularization methods and uses NN to replace certain modules, *e.g.* Kruse *et al*. [21], Meinhardt *et al*. [27], and Zhang *et al*. [48]. The NNs listed above are trained with known noise level. the deblurring networks proposed in Jin *et al*. [17] and Bigdeli *et al*. [2] are adaptive to different noise levels.

While most methods assume Gausian white noise, there are also studies on the methods that are robust to non-Gaussian noise. Carlavan and Laure [4] proposed a deblurring method in the presence of Poisson measurement noise. Dong *et al*. [10] proposed to learn fidelity term to address complex real noise. When there are saturated regions in blurred images, these saturated pixels can be viewed as outliers. The robustness to such outliers are addressed in Whyte *et al*. [44] and Cho *et al*. [8].

2.2. Image deblurring focusing on handling kernel/model error

As it does not assume the blurring process is known, blind image deblurring needs to estimate blurring kernel before deblurring images. As blind deblurring is a challenging ill-posed non-linear inverse problem, the estimation of blurring kernel is hardly free of error. In addition, the convolution-based blurring model itself is only an approximation to practical blurring process.

There is limited literature on non-blind deconvolution that focuses on handling kernel/model error. Ji and Wang [16] proposed an ℓ_1-norm relating regularization method with two auxiliary variables that address kernel error and result artifacts. The drawback of such method is that the sparsity prior imposed on these two variables does not always hold true in practice. Ren *et al*. [32] proposed a partial convolution model with the estimation of a confidential map for modeling kernel estimation error in Fourier domain, and deblurring the image using such a confidential map. Vasu *et al*. [43] proposed a deep learning based approach, whose main idea is to produce multiple estimations of the latent image w.r.t. different regularization hyper-parameters and then fuse them together using DNN to have the final deblurring result. The success of the method [43] depends on appropriate setting of regularization hyper-parameters which can be tricky in practice. Ren *et al*. [33] discussed a more general image restoration formulation which also covers non-blind deblurring with an erroneous kernel. The main idea of [33] is to simultaneously estimate fidelity term and image prior in the NN, in which fidelity term is composed by different norms of the residue under a learnable filter bank. The method [33] is not specific designed for image deconvolution, and its performance is not better than that of [43].

3. Main body

The proposed method is based on the unrolling of the iterative scheme for solving the following optimization problem:

$$\min_{\boldsymbol{x},\boldsymbol{u}} \|\boldsymbol{y} - \widehat{\boldsymbol{k}} \otimes \boldsymbol{x} - \boldsymbol{u}\|_2^2 + \phi(\boldsymbol{x}) + \psi(\boldsymbol{u}|\boldsymbol{x}), \tag{6}$$

where $\phi(\cdot)$ and $\psi(\cdot|\cdot)$ are two regularization terms related to the priors imposed on the latent image and the correction term caused by kernel/model error. As image prior usually is imposed on the high-frequency components of $\boldsymbol{x}$, one often introduces an auxiliary variable $\boldsymbol{z}$ to facilitate the design of efficient numerical solver. In this paper, we apply the half-quadratic splitting [13] to reformulate the problem (6) as:

$$\min_{\boldsymbol{x},\boldsymbol{z},\boldsymbol{u}} \|\boldsymbol{y} - \widehat{\boldsymbol{k}} \otimes \boldsymbol{x} - \boldsymbol{u}\|_2^2 + \|\mathrm{diag}(\lambda)(\Gamma\boldsymbol{x} - \boldsymbol{z})\|_2^2 + \rho(\boldsymbol{z}) + \psi(\boldsymbol{u}|\boldsymbol{x}), \tag{7}$$

where Γ denote the set of high-pass filters such that $\Gamma\boldsymbol{x}$ covers high-frequency components of the image $\boldsymbol{x}$. For instance, gradient operator ∇ or wavelet filter bank $\{f_i \otimes\}$

3.1. Iterative scheme and optimization unrolling

The optimization problem (7) can be solved via an alternating iterative scheme:

$$\begin{aligned} \boldsymbol{x}^{(t)} = \operatorname*{argmin}_{\boldsymbol{x}} & \|\boldsymbol{y} - \widehat{\boldsymbol{k}} \otimes \boldsymbol{x} - \boldsymbol{u}^{(t-1)}\|_2^2 \\ & + \lambda \|\mathrm{diag}(\lambda)(\Gamma\boldsymbol{x} - \boldsymbol{z}^{(t-1)})\|_2^2; \end{aligned} \tag{8}$$

$$\boldsymbol{z}^{(t)} = \operatorname*{argmin}_{\boldsymbol{z}} \mu\|\Gamma\boldsymbol{x}^{(t)} - \boldsymbol{z}\|_2^2 + \rho(\boldsymbol{z}); \tag{9}$$

$$\boldsymbol{u}^{(t)} = \operatorname*{argmin}_{\boldsymbol{u}} \|\boldsymbol{y} - \widehat{\boldsymbol{k}} \otimes \boldsymbol{x}^{(t)} - \boldsymbol{u}\|_2^2 + \psi(\boldsymbol{u}|\boldsymbol{x}^{(t)}). \tag{10}$$

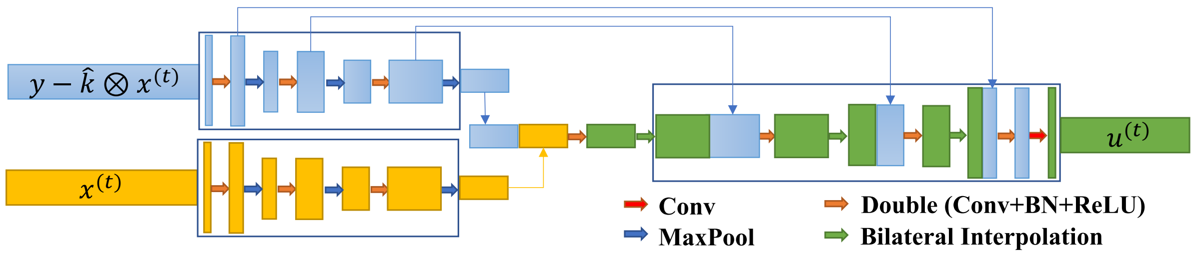


Figure 2: The structure of Dual-Path U-net based correction process

There are three steps involved in each iteration. The first step (8) is an *inversion process* that gives a least squares solution to the latent image, provided the given input corrected by $\boldsymbol{u}^{(t-1)}$ and the estimate of high-pass image channels $\boldsymbol{z}^{(t-1)}$. Such a least squares solution has an analytic solution, which can be efficiently solved using discrete Fourier transform when Γ is composed by multiple convolutions with high-pass filters. The second step (9) is a *denoising process* or *de-artifacting process*, which removes possible artifacts from high-pass image channels using certain prior encoded in ρ. The third step (10) is a *correction process* for correcting the term relating to model error, which relies on a functional $\psi(\cdot|\boldsymbol{x})$ driven by $\boldsymbol{x}$.

It can be seen that the challenging parts in the iterative scheme above are both the second step and the third step, which involve complex regularizations on both image and correction terms. The design of these two regularization terms is also critical to the robustness of the method to kernel/model error. In the next, we will give a detailed discussion on how to use deep NN as a tool to learn the second and the third step.

3.2. CNN-based denoising process

CNN-based learnable image prior has been extensively exploited in image denoising and restoration (*e.g.* [49, 50, 27]), and showed its superior performance over pre-defined image priors in many experiments. In our implementation, we also use a CNN to model the denoising process, i.e., the second step (9). As the channels of $\boldsymbol{z}$ are correlated in the sense that they are high-pass components of the same image, we first train a CNN to remove noise in $\boldsymbol{x}^{(t)}$ and then pass the result high-pass channels to $\boldsymbol{z}$. Such a modification keeps the correlation among different high-pass channels of the same image. Furthermore, similar to [15], we use all possible estimates in all previous stages as the input $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(t)}$, which help avoiding the issue of vanishing gradients. In short, the function of the denoising process at the stage t for denoising takes the forms as

$$\mathcal{D}^{(t)}(\cdot|\theta_D^{(t)}) : [\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(t)}] \rightarrow \tilde{\boldsymbol{x}} \rightarrow \nabla\tilde{\boldsymbol{x}} \rightarrow \boldsymbol{z}^{(t)},$$

where $\theta_D^{(t)}$ denotes the parameters of $\mathcal{D}^{(t)}$, and the CNN is used for modeling the mapping from $[\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(t)}]$ to $\tilde{\boldsymbol{x}}$.

The CNN-based denoising process is called *Dn-CNN*, whose implementation details are given as follows. At each stage, We use 17-block standard CNN with the structure

$$\text{Conv} \rightarrow \text{BN} \rightarrow \text{ReLU}.$$

except the first block and the last block. The first block omits the BN layer, and the last block only contains one Conv layer. For all Conv layers in the CNN, The kernel size is 3×3 and the channel size is 64.

3.3. Dual-path U-net based correction process

The third step (10) is about estimating the correction term $\boldsymbol{u}$ from the residual $\boldsymbol{r}^{(t)} = \boldsymbol{y} - \widehat{\boldsymbol{k}} \otimes \boldsymbol{x}^{(t)}$, regularized by the term $\psi(\cdot|\boldsymbol{x}^{(t)})$. The term $\psi(\cdot|\boldsymbol{x}^{(t)})$ is dependent on the latent variable $\boldsymbol{x}$. In other words, the variable $\boldsymbol{u}$ is determined by both the residual $\boldsymbol{r}^{(t)}$ and the estimate $\boldsymbol{x}^{(t)}$. We propose to learn a deep NN to approximate the mapping from $(\boldsymbol{r}^{(t)}, \boldsymbol{x}^{(t)})$ to $\boldsymbol{u}^{(t)}$, which can be expressed as

$$\mathcal{P}^{(t)}(\cdot|\theta_P^{(t)}) : \begin{pmatrix} \boldsymbol{y} - \widehat{\boldsymbol{k}} \otimes \boldsymbol{x}^{(t)} \\ \boldsymbol{x}^{(t)} \end{pmatrix} \rightarrow \boldsymbol{u}^{(t)}, \tag{11}$$

where $\theta_P^{(t)}$ denotes NN parameters of $\mathcal{P}^{(t)}$. Our proposed approximation module is called *DP-Unet*. The DP-Unet implements the U-net [26] with the combinations of the downsampled codes from the dual inputs. See Fig 2 for the diagram of DP-Unet.

3.4. Overall network structure and loss function

The proposed CNN has totally $T + 1$ stages, denoted by $\{\mathcal{S}_t\}_{t=0}^T$, corresponding to $T + 1$ iterations in the optimization algorithm. The proposed NN generates a sequence of deconvoluted images $\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(T+1)}\}$:

$$\begin{aligned} &S_0 : (\boldsymbol{y}, \widehat{\boldsymbol{k}}) \rightarrow \boldsymbol{x}^{(1)}, \\ &S_t : (\boldsymbol{y}, \widehat{\boldsymbol{k}}, [\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(t)}]) \rightarrow \boldsymbol{x}^{(t+1)}, 1 \leq t \leq T. \end{aligned}$$

In stage $\mathcal{S}_0$, the $\boldsymbol{u}^{(0)}$, $\boldsymbol{z}^{(0)}$ are set to be $\mathbf{0}$. All other stages contain three components: Db-INV for the inversion process; DP-Unet for estimating fidelity correction term, Dn-CNN for removing artifacts from the estimate of image gradients passed from Db-INV. It is observed that after the stage S_4, little performance gain has been seen in later stages. Thus we set $T = 4$. See Fig 3 for the outline of the proposed NN.

Given a set of training data $\{\boldsymbol{x}_j, \boldsymbol{y}_j\}_{j=1}^J$ where $(\boldsymbol{x}_j, \boldsymbol{y}_j)$ denotes the pair of latent image and its noisy blurred counterpart. Let $\boldsymbol{x}_j^{(i)}$ represents the output of the i-th stage in our NN w.r.t. the input $\boldsymbol{y}_j$. The loss function is defined as

$$\mathcal{L} := \frac{1}{J}\sum_{j=0}^{J}\Big(\|\boldsymbol{x}_j^{(T+1)} - \boldsymbol{x}_j\|_2^2 + \sum_{i=2}^{T}\mu_i\|\boldsymbol{x}_j^{(i)} - \boldsymbol{x}_j\|_2^2\Big), \tag{12}$$

where the weights $\{\mu_i\}_{i=1}^{T-1}$ are set to 0.8 throughout all experiments. The first term in (12) is for encouraging the output of the

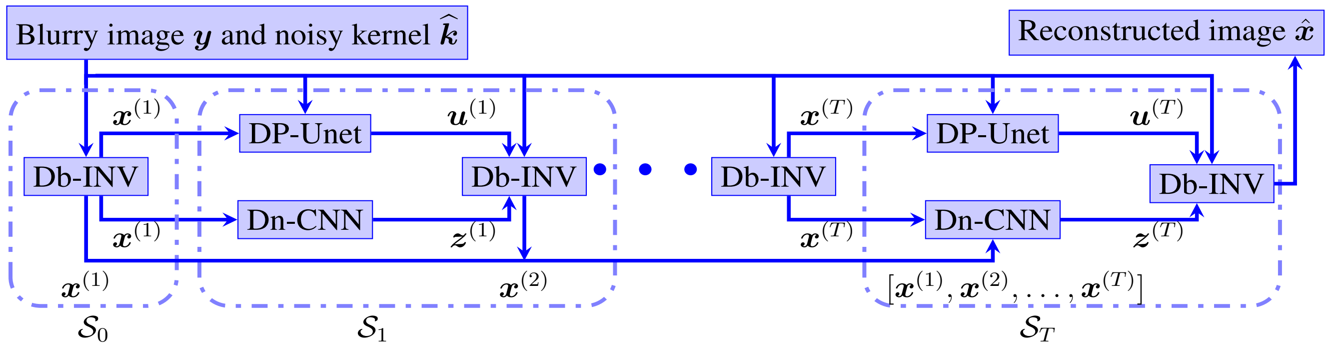


Figure 3: Diagram of the proposed NN for image deblurring in the presence of kernel/model error.

NN close to the truth. The second term is for avoiding the intermediate results too far from the truth.

3.5. Synthesis of erroneous kernel

Besides NN architecture, sufficient high-quality training samples are also vital for the success of deep-learning based model. However, constructing an inaccurate kernel set for training is a great challenge. It not only needs to run time-consuming blind deblurring algorithms to obtain kernels, but the generated kernels are algorithm-dependent and not representative enough to reflect various types of kernel errors. Thus, we propose a procedure to synthesize erroneous kernels that can cover a wide range of error patterns shown in different blind deblurring methods.

There are several patterns observed in kernel error from existing blind deblurring algorithms, including

- kernel diffusion (overly smoothed),
- missing pieces of kernels,
- random spike-like noise.

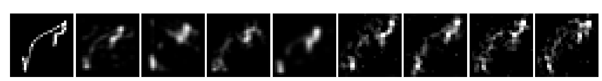
See Fig 4 (b-e) for an illustration. Each of these error patterns has a profound effect of the quality of the results. We propose an algorithm 1 to generate noisy kernels that can reflect these error patterns. There are 4 steps in the synthesis of noisy kernels: (1) Adding Gaussian noise around the contour of the ground true kernel with s.t.d σ_1; (2) randomly sampling the kernel using Bernoulli sampling model with probability β; (3) blurring the kernel with a Gaussian function with kernel size $m \times n$ and s.t.d μ; and (4) adding Gaussian noise out of the kernel contour with s.t.d σ_2. Note that the noises added around and away the contour are treated differently. The noise around the contour tends to be smoothed out while the noises off the contour are randomly distributed. See Fig 4 (f-i) for the visualization of noisy kernels synthesized by the proposed procedure. The procedure is outlined in Algorithm 1. The error degree is controlled by the noise level. In our implementation, noise levels are set as $\sigma_1 = 0.01, \sigma_2 = 0.002$. The scale and standard deviation of blurry function are sampled from some uniform distributions: $m, n \sim \mathcal{U}(1, 5), \mu \sim \mathcal{U}(0.05, 8)$ to generate kernels with different error degrees. The sampling rate in Bernoulli sampling model is set as 0.95.

Algorithm 1 Kernel error synthesis

Input: Ground truth kernel $\boldsymbol{k}$. Noise level σ_1, σ_2. Size and standard deviation of kernel blurry function $([m, n], \mu)$. Bernoulli sampling probability β.

Output: Generated noisy kernel $\widehat{\boldsymbol{k}}$.

%% Generate noises and kernel blurry function
$\boldsymbol{n}_1 = \mathcal{N}(0, \sigma_1^2 \mathbf{I}); \boldsymbol{n}_2 = \mathcal{N}(0, \sigma_2^2 \mathbf{I}); \boldsymbol{f} = \mathcal{N}([m, n], \mu^2 \mathbf{I})$
%% Generate mask indicated the region around kernel
$\boldsymbol{M} = where(\boldsymbol{k} > 0)$ % Find the contour of kernel
$\boldsymbol{M} = dilate(\boldsymbol{M})$ % Dilate kernel contour
%% noisy kernel generation process
$\widehat{\boldsymbol{k}} = \boldsymbol{n}_1 \odot \boldsymbol{M} + \boldsymbol{k}$ % Add noise around the kernel
$\widehat{\boldsymbol{k}} = \mathcal{B}(\widehat{\boldsymbol{k}}, \beta)$ % Bernoulli random sampling
$\widehat{\boldsymbol{k}} = \widehat{\boldsymbol{k}} \otimes \boldsymbol{f}$ % Blurring the kernel
$\widehat{\boldsymbol{k}} = \widehat{\boldsymbol{k}} + \boldsymbol{n}_2 \odot (\mathbf{I} - \boldsymbol{M})$ % Add noise off the contour



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 4: Illustration of real and synthesized noisy kernels. (a) True kernel. (b-e) The kernels estimated from existing blind deblurring methods including Cho and Lee [7], Levin *et al.* [24], Sun *et al.* [41], Michaeli and Irani[28]; (f-i) the noisy kernels generated from the proposed synthesis procedure.

4. Experiments

4.1. Experimental settings

Training data. We use the BSDS500 dataset [1] to prepare training data. A set of 500 latent images is generated by randomly cropping the images in the BSDS500 into the ones of size 256×256. As for kernel preparation, we use both the synthetic and real blurry kernels for training. As for synthetic kernels, we use the proposed noisy kernel synthetic procedure to deal with a 192 motion-blur kernel set from [37]. For true kernels, we adopt the same procedure as [43] to utilize the kernels return by blind deblurring methods [20, 25, 41, 6]. Synthetic kernels and true kernels both take the half of the noisy kernel set. Totally, about 140k noisy kernels are used for training. We also apply the affine registration [42] in the kernel set to address possible misalignment. It is noted that there is no any overlap between training and testing sets.

Test data. We use three standard benchmark datasets in image restoration as test datasets. Levin *et al*.'s dataset [25] contains 32 gray-scale images produced by 4 sharp images convolved with 8 ground truth kernels from [25]. The estimated kernels are obtained by applying 4 blind deblurring algorithms on them: Cho and Lee [7], Levin *et al*. [25], Pan *et al*. [30], and Sun *et al*. [41]. Sun *et al*.'s dataset [41] has totally 640 images, generated by 80 clear images and the same truth kernels from [25]. The estimated kernels are obtained by applying 3 blind motion deblurring algorithms on them: Cho and Lee [7], Xu and Jia [45], Michaeli and Irani [28]. Lai *et al*.'s dataset [22] contains about 100 color images and 4 ground truth kernels. In order to have the same configuration as other deep learning method, we only use a subset of the dataset[1], the same one used in Vasu's [43]. The estimated kernels are obtained by applying 4 blind deblurring algorithms on them: Xu and Jia [45], Xu *et al*. [47], Sun *et al*. [41], Perrone and Favaro [31].

Other important details. For initialization, all weights in NN are initialized by orthogonal matrices [36], and the biases are set to zeros. As for $\{\lambda_i^{(t)}\}_i$, we set $\lambda_i^{(0)} = 0.005$ for stage $\mathcal{S}_0$. For later stage, $\lambda_i^{(t)}$ is set as 0.1 for no noise case and 0.5 for 1% noise case. The NN is trained using the Adam method [18]. The model is trained with 500 epochs. The learning rate is initially set be 1×10^{-3} and drops with rate 0.2 after epoch 350. As for metric calculation, we follow the same procedure as [25, 43], *i.e*. first aligning output images with the sharp images with sub-pixel shift and then cutting off the boundary pixels.

4.2. Ablation study

Our ablation studies focus on the performance gain brought by two components: (1) the introduction of DP-Unet, and (2) the proposed procedures for simulating kernels estimated in practice. We train our NN with the same settings. See Table 1 for the results on the Levin *et al*.'s dataset.

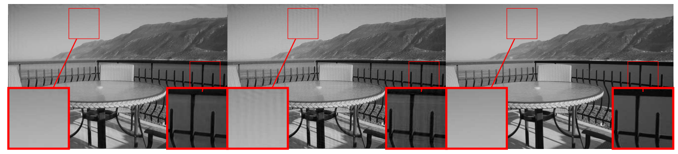
Table 1: Ablation study on the proposed NN

Levin *et al.*	[7]	[25]	[30]	[41]
w/o DP-Unet	30.61	30.85	34.30	32.90
w/o Synthesis Kernels	30.06	30.35	33.86	32.39
Ours	30.92	31.14	34.66	33.36

With vs. without DP-Unet. Table 1 shows that the DP-Unet module provides around $0.3 - 0.4$ dB performance gain which is quite noticeable. See Fig 5 for an illustration how DP-Unet helps to reduce artifacts in the deblurred result. The ringing artifacts in (b) is mostly attenuated by using DP-Unet shown in (a). Such an improvement justified the need of the explicit treatment of kernel/model error and the effectiveness of the DP-Unet for predicting correction term.

With vs. without synthetic kernels. In this study, while keeping all other settings the same, the NN is trained twice. One only uses the kernels returned from existing blind deblurring methods. The other uses both the the kernels returned from existing blind deblurring methods and the kernels synthesized from Algorithm 1. It can be seen from Table 1 that, for the network trained without using the kernels synthesized by Algorithm 1, there is a significant performance hit in terms of PSNR value, about 0.7 dB. Such a performance loss clearly indicates the effectiveness of Algorithm 1 on generating noisy kernels whose error patterns are close to that from practical blind deblurring methods.

[1] Main reason using such a subset is for fair comparison to the NN presented in Vasu's [43] whose code or model is not available online.



(a) With DP-Unet (b) Without DP-Unet (c) Ground truth

Figure 5: Visual inspection of recovered image with and without DP-Unet part in the NN. Zoom in for better visualization

4.3. Performance evaluation and comparison

The performance evaluation is split into two parts. The first part focuses on the comparison of the proposed method to existing representative non-blind deblurring algorithms without specific treatment on kernel/model error. The second part focuses on the comparison of the proposed method to the ones that focus on the robustness to kernel/model error.

In the first part, the proposed method is compared to representative non-blind image deblurring algorithms without special treatment on kernel error, including Krishnan and Fergus [19], Zoran and Weiss [51], Kruse *et al*. [21], Zhang *et al*. [50], Zhang *et al*. [48]. Note that the last three are deep learning based methods. For this experiment, we use "edge-taper" [38] in MATLAB Image Processing Toolbox for simulating the boundary condition of practical blurring, and the Gaussian noise level is set to be 1%.

See Table 2 for the comparison of the proposed method to these 5 methods in different configurations. The results from Kruse *et al*. [21] are marked as N/A as it fails to generate meaningful images in certain color channel for some instances in Lai *et al*.'s dataset. It can be seen that the proposed algorithm is better than other methods about <u>from 0.4 to 1.0 dB</u> in Levin *et al*.'s dataset and <u>from 0.3 to 0.6 dB</u> in Sun *et al*.'s dataset. For color images blurred by large kernels in Lai *et al*.'s dataset, the kernel error is much more severe than that in the other two datasets. Thus, the results from those existing methods without specific treatment on kernel error did quite poorly in comparison to ours. Indeed, the performance gain from our method is about <u>2 dB</u>. The quantitative performance gain of the proposed method is also consistent with the improvement of visual quality. See Fig 6 for the illustration of some results. It shows the importance of handling kernel/model error in practical deblurring.

In the second part, two experiments are conducted. The first experiment is comparing the proposed method to the other two non-learning based regularization methods that have specific treatment on possible kernel error. One is Ji and Wang [16] and the other is Whyte *et al*. [44]. Again, the boundary extension is the same as the first part and the noise level is set to 1%. See Table 3 for the

Table 2: Average PSNR(dB)/SSIM of the results, in comparison to image deblurring methods without specific treatment on kernel error

NBD	Method for Kernel estimation										
	Levin *et al.*				Sun *et al.*			Lai *et al.*			
	[7]	[24]	[30]	[41]	[7]	[45]	[28]	[45]	[47]	[41]	[31]
[19]	28.03/0.79	28.03/0.80	30.08/0.84	29.26/0.82	28.66/0.79	29.20/0.80	28.85/0.79	19.89/0.72	19.17/0.65	19.53/0.63	18.60/0.65
[51]	27.75/0.81	27.82/0.82	29.10/0.85	28.57/0.83	28.78/0.81	29.08/0.81	28.66/0.80	20.33/**0.74**	19.88/0.72	20.02/0.71	19.58/**0.70**
[21]	28.54/0.86	29.00/0.88	31.19/**0.92**	30.83/0.90	29.69/**0.87**	30.51/**0.88**	29.82/**0.86**	N/A	N/A	N/A	N/A
[50]	27.99/0.82	28.09/0.81	30.42/0.86	29.56/0.83	28.84/0.81	29.54/0.83	29.23/0.82	19.99/0.70	19.36/0.67	19.46/0.67	18.68/0.68
[48]	28.67/0.85	28.74/0.86	31.12/0.90	30.27/0.88	29.79/0.86	30.45/0.86	29.84/0.84	20.27/**0.74**	19.52/0.70	19.80/0.70	19.12/**0.70**
Ours	**29.76/0.88**	**29.78/0.89**	**31.97/0.92**	**31.24/0.91**	**30.44/0.87**	**30.84**/0.87	**30.27/0.86**	**22.53/0.74**	**22.27/0.73**	**22.31/0.72**	**21.61/0.70**

Table 3: Average PSNR(dB)/SSIM of the results on three benchmark datasets, in comparison to robust regularization methods with specific treatment on kernel error/outliers

Levin *et al.*	[7]	[24]	[30]	[41]
[16]	27.81/0.83	27.94/0.84	29.19/0.86	28.86/0.85
[44]	27.94/0.80	28.02/0.81	29.55/0.84	29.10/0.82
Ours	**29.76/0.88**	**29.78/0.89**	**31.97/0.92**	**31.24/0.91**
Sun *et al.*	[7]	[45]	[28]	
[16]	27.65/0.76	27.85/0.76	27.66/0.75	
[44]	28.57/0.78	29.06/0.79	28.74/0.78	
Ours	**30.44/0.87**	**30.84/0.87**	**30.27/0.86**	
Lai *et al.*	[45]	[47]	[41]	[31]
[16]	19.70/0.72	19.55/0.66	19.53/0.69	19.15/**0.70**
[44]	20.21/0.72	19.87/0.70	19.91/0.69	19.35/**0.70**
Ours	**22.52/0.74**	**22.25/0.73**	**22.29/0.72**	**21.55/0.70**

comparison of the results from different methods. It can be seen that the proposed deep-learning-based method noticeably outperformed these two regularization methods. It shows the advantage of the proposed deep learning method over traditional regularization methods on handling kernel/model error.

In the second experiment, the proposed method is compared to another deep-learning method Vasu *et al.* [43]. It is noted that Ren *et al.*'s method [33] is not specifically designed for handling kernel error, and there is no training code or trained model available online for deblurring motion-blurred images. Thus it is not included in the experiment. For fairness, we following the same setting as Vasu *et al.* [43], which keeps the boundary information and blurred images are noise-free.

See Table 4 for the comparison. It can be seen that the performance of our deep-learning-based approach was modestly better than Vasu *et al.* [43] on the datasets with small kernel errors: from 0.2 to 0.4 dB gain on Levin *et al.*'s dataset. For other two datasets with large kernel error, our method outperformed Vasu *et al.* [43] by a large margin: from 1 to 2 dB gain in most cases. This shows the effectiveness of our method on handling kernel/model error over existing deep learning method.

4.4. Illustration on real images

As there is no ground truth for quantitative evaluation, only some examples are shown for visual comparison. Real images are deblurred using the same trained model in the previous experiments. See Fig 7 for visual comparison of some examples from

Table 4: Average PSNR(dB)/SSIM of the results, in comparison to existing deep learning method Vasu *et al.* [43] that focuses on handling kernel error

Levin *et al.*	[7]	[24]	[30]	[41]
[43]	30.62/0.89	30.87/0.91	34.35/0.95	33.22/**0.94**
Ours	**30.92/0.90**	**31.14/0.92**	**34.66/0.96**	**33.36/0.94**
Sun *et al.*	[7]	[45]	[28]	
[43]	29.54/0.88	32.60/0.91	30.69/0.85	
Ours	**32.12/0.92**	**32.76/0.93**	**31.60/0.90**	
Lai *et al.*	[45]	[47]	[41]	[31]
[43]	22.90/0.77	22.38/0.76	22.60/0.75	21.86/0.75
Ours	**24.83/0.81**	**24.52/0.80**	**24.79/0.80**	**23.35/0.76**

Lai *et al.*'s dataset [22]. It can be seen that our results are noticeably better than those from other methods in terms of visual quality. More examples can be found in the supplementary file. It clearly indicated the benefit of the proposed method to practical image deblurring, *i.e.* by simply calling the proposed deblurring method in the last stage of blind motion deblurring, one can have the results with better visual quality.

5. Conclusion

This paper aimed at developing a deep learning method for non-blind image deconvolution that can handle kernel/model error well. Based on the EIV model of image blurring in the presence of kernel/model error, a TLS-based iterative optimization scheme was first proposed for deblurring the image. Then, we presented a deep learning method that unrolls the iterative scheme with deep-NN-based priors on both images and correction terms. In addition, an algorithm is proposed for simulating erroneous kernels for NN training. The experiments showed that our proposed method significantly outperformed existing methods when being used in blind motion deblurring, which justifies the benefit of special treatment on kernel/model uncertainty in our method and our algorithm of simulating practical erroneous kernels,

Acknowledgment

Yuesong Nan and Hui Ji would like to acknowledge the support from the Singapore MOE Academic Research Fund (AcRF) Tier 2 research project (MOE2017-T2-2-156).

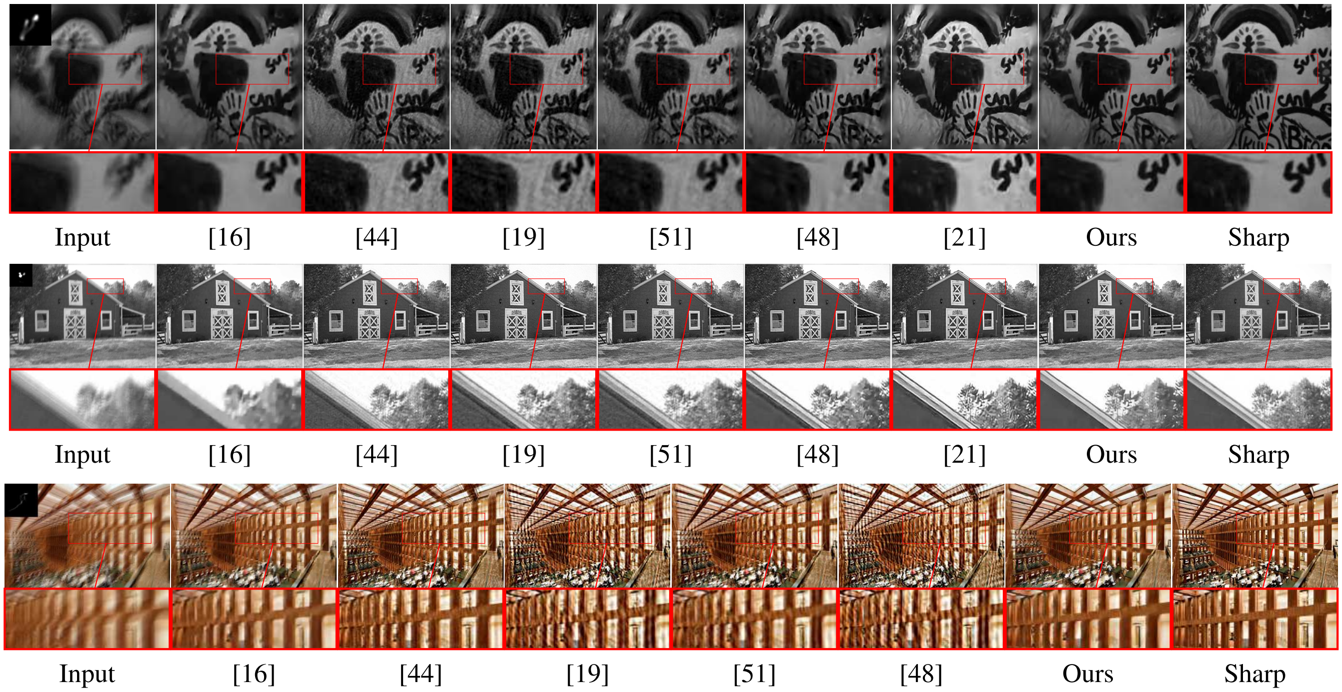


Figure 6: Deblurred results from Levin *et al.*'s dataset with the kernels returned by Pan *et al.* [30], Sun *et al.*'s dataset with the kernels returned by Michal and Irani [28], and Lai *et al.*'s dataset with the kernel returned by Xu and Jia [45]. The noise level is set as 1%.

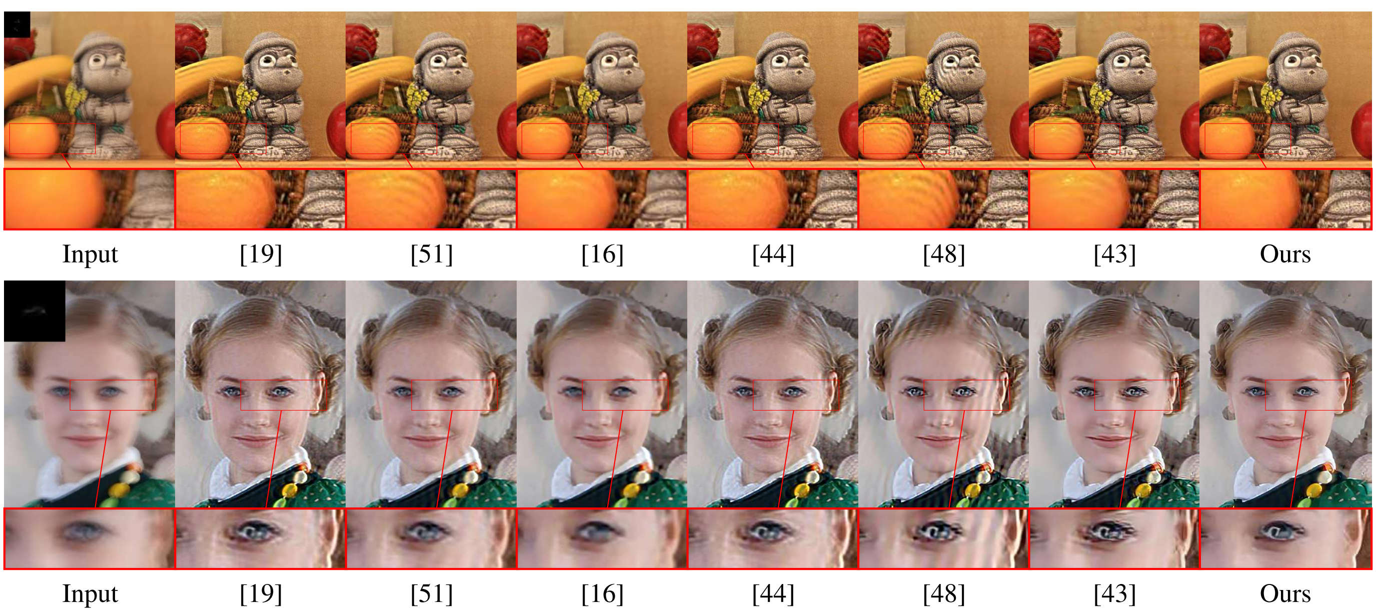


Figure 7: Deblurred results of image "harubang" and "face2" from the real dataset in Lai *et al.* [22]. The kernels are estimated by Cho and Lee [7] for image "harubang" and by Pan *et al.* [30] for image "face2". See more examples in the supplementary file.

References

[1] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, May 2011.

[2] Siavash Arjomand Bigdeli, Matthias Zwicker, Paolo Favaro, and Meiguang Jin. Deep mean-shift priors for image restoration. In *NIPS*, pages 763–772, 2017.

[3] Jian-Feng Cai, Hui Ji, Chaoqiang Liu, and Zuowei Shen. Blind motion deblurring from a single image using sparse approximation. In *CVPR*, pages 104–111. IEEE, 2009.

[4] Mikael Carlavan and Laure Blanc-Féraud. Sparse poisson noisy image deblurring. *IEEE Transactions on Image Processing*, 21(4):1834–1846, 2011.

[5] Raymond J Carroll, David Ruppert, Leonard A Stefanski, and Ciprian M Crainiceanu. *Measurement error in nonlinear models: a modern perspective*. Chapman and Hall/CRC, 2006.

[6] Ayan Chakrabarti. A neural approach to blind motion deblurring. In *European conference on computer vision*, pages 221–235. Springer, 2016.

[7] Sunghyun Cho and Seungyong Lee. Fast motion deblurring. *ACM Transactions on graphics (TOG)*, 28(5):145, 2009.

[8] Sunghyun Cho, Jue Wang, and Seungyong Lee. Handling outliers in non-blind image deconvolution. In *2011 International Conference on Computer Vision*, pages 495–502. IEEE, 2011.

[9] Aram Danielyan, Vladimir Katkovnik, and Karen Egiazarian. BM3D frames and variational image deblurring. *IEEE Trans. Image Process.*, 21(4):1715–1728, 2012.

[10] Jiangxin Dong, Jinshan Pan, Deqing Sun, Zhixun Su, and Ming-Hsuan Yang. Learning data terms for non-blind deblurring. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 748–763, 2018.

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

[12] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM TOG (Proc. SIGGRAPH)*, 25(3):787–794, 2006.

[13] Donald Geman and Chengda Yang. Nonlinear image recovery with half-quadratic regularization. *IEEE Trans. Image Process.*, 4(7):932–946, 1995.

[14] Gene H Golub and Charles F Van Loan. An analysis of the total least squares problem. *SIAM journal on numerical analysis*, 17(6):883–893, 1980.

[15] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017.

[16] Hui Ji and Kang Wang. Robust image deblurring with an inaccurate blur kernel. *IEEE Transactions on Image processing*, 21(4):1624–1634, 2011.

[17] Meiguang Jin, Stefan Roth, and Paolo Favaro. Noise-blind image deblurring. In *CVPR*, pages 3510–3518, 2017.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Dilip Krishnan and Rob Fergus. Fast image deconvolution using hyper-laplacian priors. In *NIPS*, pages 1033–1041, 2009.

[20] Dilip Krishnan, Terence Tay, and Rob Fergus. Blind deconvolution using a normalized sparsity measure. In *CVPR 2011*, pages 233–240. IEEE, 2011.

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

[22] Wei-Sheng Lai, Jia-Bin Huang, Zhe Hu, Narendra Ahuja, and Ming-Hsuan Yang. A comparative study for single image blind deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1709, 2016.

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

[24] Anat Levin, Yair Weiss, Fredo Durand, and William T Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *CVPR*, pages 2657–2664. IEEE, 2011.

[25] Anat Levin, Yair Weiss, Fredo Durand, and William T Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *CVPR*, pages 2657–2664. IEEE, 2011.

[26] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[27] Tim Meinhardt, Michael Moller, Caner Hazirbas, and Daniel Cremers. Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. In *ICCV*, pages 1781–1790, 2017.

[28] Tomer Michaeli and Michal Irani. Blind deblurring using internal patch recurrence. In *European Conference on Computer Vision*, pages 783–798. Springer, 2014.

[29] Stanley Osher, Martin Burger, Donald Goldfarb, Jinjun Xu, and Wotao Yin. An iterative regularization method for total variation-based image restoration. *SIAM Multiscale Modeling Simul.*, 4(2):460–489, 2005.

[30] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang. Blind image deblurring using dark channel prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1628–1636, 2016.

[31] Daniele Perrone and Paolo Favaro. Total variation blind deconvolution: The devil is in the details. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2909–2916, 2014.

[32] Dongwei Ren, Wangmeng Zuo, David Zhang, Jun Xu, and Lei Zhang. Partial deconvolution with inaccurate blur kernel. *IEEE Transactions on Image Processing*, 27(1):511–524, 2017.

[33] D. Ren, W. Zuo, D. Zhang, L. Zhang, and M. Yang. Simultaneous fidelity and regularization learning for image restoration. *arXiv preprint arXiv:1804.04522*, 2018.

[34] Wenqi Ren, Jiawei Zhang, Lin Ma, Jinshan Pan, Xiaochun Cao, Wangmeng Zuo, Wei Liu, and Ming-Hsuan Yang. Deep non-blind deconvolution via generalized low-rank approximation. In *NIPS*, pages 295–305, 2018.

[35] Stefan Roth and Michael J Black. Fields of experts: A framework for learning image priors. In *CVPR*, pages 860–867. IEEE, 2005.

[36] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[37] Uwe Schmidt, Jeremy Jancsary, Sebastian Nowozin, Stefan Roth, and Carsten Rother. Cascades of regression tree fields for image restoration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(4):677–689, 2016.

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

[39] Christian J Schuler, Harold Christopher Burger, Stefan Harmeling, and Bernhard Scholkopf. A machine learning approach for non-blind image deconvolution. In *CVPR*, pages 1067–1074, 2013.

[40] Libin Sun, Sunghyun Cho, Jue Wang, and James Hays. Edge-based blur kernel estimation using patch priors. In *ICCP*, pages 1–8. IEEE, 2013.

[41] Libin Sun, Sunghyun Cho, Jue Wang, and James Hays. Edge-based blur kernel estimation using patch priors. In *IEEE International Conference on Computational Photography (ICCP)*, pages 1–8. IEEE, 2013.

[42] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[43] Subeesh Vasu, Venkatesh Reddy Maligireddy, and AN Rajagopalan. Non-blind deblurring: Handling kernel uncertainty with cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3272–3281, 2018.

[44] Oliver Whyte, Josef Sivic, and Andrew Zisserman. Deblurring shaken and partially saturated images. *International journal of computer vision*, 110(2):185–201, 2014.

[45] Li Xu and Jiaya Jia. Two-phase kernel estimation for robust motion deblurring. In *European conference on computer vision*, pages 157–170. Springer, 2010.

[46] Li Xu, Jimmy SJ Ren, Ce Liu, and Jiaya Jia. Deep convolutional neural network for image deconvolution. In *NIPS*, pages 1790–1798, 2014.

[47] Li Xu, Shicheng Zheng, and Jiaya Jia. Unnatural l0 sparse representation for natural image deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1107–1114, 2013.

[48] Jiawei Zhang, Jinshan Pan, Wei-Sheng Lai, Rynson WH Lau, and Ming-Hsuan Yang. Learning fully convolutional networks for iterative non-blind deconvolution. In *CVPR*, pages 3817–3825, 2017.

[49] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Trans. Image Process.*, 26(7):3142–3155, 2017.

[50] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *CVPR*, volume 2, 2017.

[51] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, pages 479–486. IEEE, 2011.