Learning Data Terms for Non-blind Deblurring

Jiangxin Dong[1], Jinshan Pan[2], Deqing Sun[3], Zhixun Su[1,4], and Ming-Hsuan Yang[5]

[1] Dalian University of Technology dongjxjx@gmail.com, zxsu@dlut.edu.com
[2] Nanjing University of Science and Technology sdluran@gmail.com
[3] NVIDIA deqings@nvidia.com
[4] Guilin University of Electronic Technology
[5] University of California at Merced mhyang@ucmerced.edu

Abstract. Existing deblurring methods mainly focus on developing effective image priors and assume that blurred images contain insignificant amounts of noise. However, state-of-the-art deblurring methods do not perform well on real-world images degraded with significant noise or outliers. To address these issues, we show that it is critical to learn data fitting terms beyond the commonly used ℓ_1 or ℓ_2 norm. We propose a simple and effective discriminative framework to learn data terms that can adaptively handle blurred images in the presence of severe noise and outliers. Instead of learning the distribution of the data fitting errors, we directly learn the associated shrinkage function for the data term using a cascaded architecture, which is more flexible and efficient. Our analysis shows that the shrinkage functions learned at the intermediate stages can effectively suppress noise and preserve image structures. Extensive experimental results show that the proposed algorithm performs favorably against state-of-the-art methods.

Keywords: Image deblurring, learning data terms, shrinkage function, noise and outliers

1 Introduction

The recent years have witnessed significant advances in non-blind deconvolution [3, 7, 18, 26], mainly due to the development of effective image priors [7, 18, 9, 29]. When combined with a data term based on ℓ_1 or ℓ_2 norm, these methods perform well where the image blur is the main or only source of degradation. However, these methods tend to fail when the real-world images contain significant amounts of noise or outliers.

When the image blur is spatially invariant, this process is modeled by

$$b = l * k + n, \tag{1}$$

where b, l, k, and n denote the blurred image, latent image, blur kernel, and noise; and $*$ is the convolution operator. While the ℓ_1 or ℓ_2 norm is often used in this model, it implicitly assumes that the noise can be modeled by the Laplacian

or Gaussian distribution. Thus, methods based on the model with ℓ_1 or ℓ_2 norm cannot well handle significant noise or outliers in real-world images.

To address these issues, some recent methods design models for specific distortions, e.g., Gaussian noise [6, 28], impulse noise [3], and saturated pixels [3, 22]. However, domain knowledge is required to design an appropriate model specifically for a particular distortion.

Directly learning the noise model from the training data is an appealing solution. Xu et al. [26] learn the deconvolution operation from the training data and this method can handle significant noise and saturation. However, their method requires the blur kernel to be separable. Furthermore, the network needs to be fine-tuned for each kernel because this method needs to use the singular value decomposition of the pseudo inverse kernel.

In this work, we propose a discriminative framework to learn the data term in a cascaded manner. To understand the role of the data term for image deblurring, we use the standard hyper-Laplacian prior for the latent image [7, 8]. Our framework learns both the data term and regularization parameters directly from the image data. As learning the distribution of the noise results in an iterative solution, we propose to learn the associated shrinkage functions for the data term using a cascaded architecture. We find that the learned shrinkage functions are more flexible at the intermediate stages, which can effectively detect and handle outliers. Our algorithm usually reaches a good solution in fewer than 20 stages. The learned data term and regularization parameters can be directly applied to other synthetic and real images. Extensive experimental results show that the proposed method performs favorably against the state-of-the-art methods on a variety of noise and outliers.

2 Related Work

Non-blind deconvolution has received considerable attention and numerous methods have been developed in recent years. Early methods include the Wiener filter [23] and the Richardson-Lucy algorithm [10, 13]. To effectively restore image structures and suppress noise for this ill-posed problem, recent methods have focused on the regularization terms, e.g., total variation [21], hyper-Laplacian prior on image gradients [7, 9], Gaussian mixture model [29], and multi-layer perceptron [18].

To learn good priors for image restoration, Roth et al. [14] use a field of experts (FoE) model to fit the heavy-tailed distribution of gradients of natural images. Schmidt et al. [17] extend the FoE framework based on the regression tree field model [5]. The cascade of shrinkage fields model [16] has been proposed to learn effective image filters and reaction (shrinkage) functions for image restoration. Xiao et al. [24] extend the shrinkage fields model to blind image deblurring. Chen et al. [2] extend conventional nonlinear reaction-diffusion models by discriminatively learning filters and parameterized influence functions. In particular, they allow the filters and influence functions to be different at each stage of the reaction-diffusion process.

Based on the maximum a posteriori (MAP) framework, existing image priors are usually combined with one specific type of data term for deblurring. The commonly used data term is based on the ℓ_2 norm which models image noise with a Gaussian distribution [7]. However, the Gaussian assumption often fails when the input image contains significant noise or outliers. Bar et al. [1] assume a Laplacian noise model and derive a data term based on the ℓ_1 norm to handle impulsive noise. These methods can suffer from artifacts for other types of noise or outliers, since these data terms are designed for specific noise models.

Recently, Cho et al. [3] show that a few common types of outliers cause severe ringing artifacts in the deblurred images. They explicitly assume a clipped blur model to deal with outliers and develop a non-blind deconvolution method based on the Expectation-Maximization (EM) framework. This method relies on the preset values of the tradeoff parameters, which requires hand-crafted tuning for different levels of noise and outliers. Taking into account the non-linear property of saturated pixels, Whyte et al. [22] propose a modified version of the Richardson-Lucy algorithm for image deblurring. Jin et al. [6] present a Bayesian framework-based approach to non-blind deblurring with the unknown Gaussian noise level.

To deal with image noise and saturation, Xu et al. [26] develop a deblurring method based on the convolutional neural network (CNN). However, their network needs to be fine-tuned for every kernel as it is based on the singular value decomposition of the pseudo inverse kernel. CNNs have also been used to learn image priors for image restoration [28, 27]. However, these methods [28, 27] mainly focus on Gaussian noise and cannot apply to other types of outliers. In addition, Schuler et al. [19] propose to learn the data term by a feature extraction module, which uses a fixed ℓ_2 norm penalty function and focuses on blind deconvolution. In contrast, our method can handle different types and levels of noise and does not require fine-tuning for a particular blur kernel.

For low-level vision tasks, numerous methods have been developed to model the data fitting errors [20]. However, the learned distributions are often highly complex and require an iterative solution. In addition, the probabilistically trained generative models usually require ad-hoc modifications to obtain good performance [15]. Compared with learning the distribution (penalty function), learning the associated solution (shrinkage function) is more flexible, because non-monotonic shrinkage functions can be learned [16]. Furthermore, shrinkage functions make learning efficient, since the model prediction and the gradient update have closed forms. We are inspired by shrinkage fields [16] to discriminatively learn shrinkage functions for the data term, which is more flexible than generatively learning the penalty function. Thus our framework can handle various types of noise and outliers and is also computationally efficient.

3 Proposed Algorithm

This paper proposes to present a discriminative deconvolution framework that performs robustly in the presence of significant amounts of noise or outliers. We

estimate the latent image from the blurred image using the MAP criterion

$$l = \underset{l}{\operatorname{argmax}}\, p(l|k,b) = \underset{l}{\operatorname{argmax}}\, p(b|k,l)p(l), \tag{2}$$

where the first (data) term models the distribution of the residue image $\log p(b|k,l) = -\frac{1}{\lambda}\mathcal{R}(b - l * k)$, and $\mathcal{R}(\cdot)$ measures the data fitting error. In addition, $p(l)$ is the latent image prior $\log p(l) = -\mathcal{P}(l)$. The objective function in (2) can be equivalently solved by minimizing the following energy function

$$\min_l E(l|k,b) = \min_l \mathcal{R}(b - l * k) + \lambda\mathcal{P}(l). \tag{3}$$

As our focus is on the data term, we use a standard hyper-Laplacian prior and set $\mathcal{P}(l) = \|\nabla l\|_p = \sum_i |(\nabla_h l)_{\mathrm{x}}|^p + |(\nabla_v l)_{\mathrm{x}}|^p$, where ∇_h and ∇_v denote the horizontal and vertical differential operators respectively and x is the pixel index. In this paper, we fix $p = 1$ for the image prior in (3) and show that even with a simple total variation prior, the proposed algorithm is able to deblur images with significant amounts of noise and saturation. We show that this algorithm requires low computational load and can be easily integrated with more expressive image priors for better performance. Different from most existing methods that use ℓ_1 or ℓ_2 norm for $\mathcal{R}$, we assume a flexible form for $\mathcal{R}$ that can be learned along with the tradeoff parameter λ from images.

3.1 Data Term

To enforce the modeling capacity, the data term $\mathcal{R}$ is parameterized to characterize the spatial information and the complex distribution of the residue image $b - l * k$,

$$\mathcal{R}(b - l * k) = \sum_{i=0}^{N_f} \mathcal{R}_i(f_i * (b - l * k)), \tag{4}$$

where f_i is the i-th linear filter (particularly, f_0 is set as the delta filter to exploit the information of the residue image in the raw data space), $\mathcal{R}_i$ is the i-th corresponding non-linear penalty function that models the i-th filter response, and N_f is the number of non-trivial linear filters for the data term.

3.2 Inference

Formulation. We first describe the scheme to minimize the energy function (3) and then explain how to learn the data term. We use the half-quadratic optimization method and introduce auxiliary variables $\mathbf{z}$, $\mathbf{v}_i$, and $\mathbf{u}$, corresponding to $\mathbf{b} - \mathbf{Kl}$, $\mathbf{F}_i(\mathbf{b} - \mathbf{Kl})$, and $\nabla\mathbf{l} = [\nabla_h\mathbf{l}, \nabla_v\mathbf{l}]$. Here, $\mathbf{K}$, $\mathbf{F}_i$, $\mathbf{l}$, and $\mathbf{b}$ denote the matrix/vector forms of k, f_i, l, and b. Thus, the energy function (3) becomes

$$\min_{\mathbf{z},\mathbf{v},\mathbf{u},\mathbf{l}} \frac{\tau}{2}\|\mathbf{b}-\mathbf{Kl}-\mathbf{z}\|_2^2 + \mathcal{R}_0(\mathbf{z}) + \sum_{i=1}^{N_f}\left(\frac{\beta}{2}\|\mathbf{F}_i(\mathbf{b}-\mathbf{Kl})-\mathbf{v}_i\|_2^2 + \mathcal{R}_i(\mathbf{v}_i)\right) + \lambda\left(\frac{\gamma}{2}\|\nabla\mathbf{l}-\mathbf{u}\|_2^2 + \|\mathbf{u}\|_1\right), \tag{5}$$

where τ, β, and γ are penalty parameters. For brevity, we denote $\mathbf{F}_h\mathbf{l} = \nabla_h\mathbf{l}$ and $\mathbf{F}_v\mathbf{l} = \nabla_v\mathbf{l}$. We use the coordinate descent method to minimize the relaxed energy function (5),

$$\mathbf{z} = \underset{\mathbf{z}}{\operatorname{argmin}} \frac{\tau}{2}\|\mathbf{b} - \mathbf{K}\mathbf{l} - \mathbf{z}\|_2^2 + \mathcal{R}_0(\mathbf{z}), \tag{6}$$

$$\mathbf{v}_i = \underset{\mathbf{v}_i}{\operatorname{argmin}} \frac{\beta}{2}\|\mathbf{F}_i(\mathbf{b} - \mathbf{K}\mathbf{l}) - \mathbf{v}_i\|_2^2 + \mathcal{R}_i(\mathbf{v}_i), \tag{7}$$

$$\mathbf{u} = \underset{\mathbf{u}}{\operatorname{argmin}} \frac{\gamma}{2}\|\nabla\mathbf{l} - \mathbf{u}\|_2^2 + \|\mathbf{u}\|_1, \tag{8}$$

$$\mathbf{l} = \underset{\mathbf{l}}{\operatorname{argmin}} \frac{\tau}{2}\|\mathbf{b} - \mathbf{K}\mathbf{l} - \mathbf{z}\|_2^2 + \sum_{i=1}^{N_f} \frac{\beta}{2}\|\mathbf{F}_i(\mathbf{b} - \mathbf{K}\mathbf{l}) - \mathbf{v}_i\|_2^2 + \frac{\lambda\gamma}{2}\|\nabla\mathbf{l} - \mathbf{u}\|_2^2. \tag{9}$$

Given the current estimates of $\mathbf{z}$, $\mathbf{v}$, and $\mathbf{l}$, finding the optimal $\mathbf{u}$ is reduced to a shrinkage operation [21]. Given the current estimates of $\mathbf{z}$, $\mathbf{v}$, and $\mathbf{u}$, the energy function (9) is quadratic w.r.t. $\mathbf{l}$ and has a closed-form solution

$$\mathbf{l} = \zeta^{-1}\xi, \tag{10}$$

where $\zeta = \mathbf{K}^\top\mathbf{K} + \frac{\beta}{\tau}\sum_{i=1}^{N_f}\mathbf{K}^\top\mathbf{F}_i^\top\mathbf{F}_i\mathbf{K} + \frac{\lambda\gamma}{\tau}(\mathbf{F}_h^\top\mathbf{F}_h + \mathbf{F}_v^\top\mathbf{F}_v)$ and $\xi = \mathbf{K}^\top(\mathbf{b} - \mathbf{z}) + \frac{\beta}{\tau}\sum_{i=1}^{N_f}\mathbf{K}^\top\mathbf{F}_i^\top(\mathbf{F}_i\mathbf{b} - \mathbf{v}_i) + \frac{\lambda\gamma}{\tau}(\mathbf{F}_h^\top\mathbf{u}_h + \mathbf{F}_v^\top\mathbf{u}_v)$.

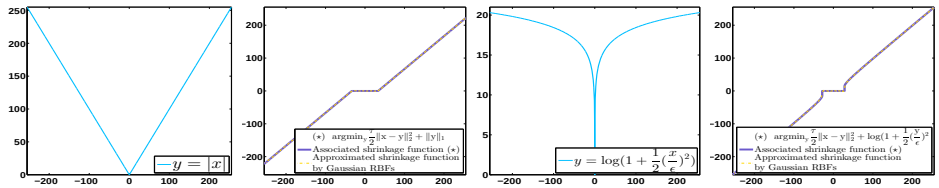
Learning the data term. Instead of learning the penalty functions $\mathcal{R}_i$ in (6) and (7), we directly learn the solutions to the optimization problems (6) and (7), i.e., the shrinkage functions associated with the penalty functions $\mathcal{R}_i$. We model each shrinkage function for (6) and (7) as a linear combination of Gaussian RBF components [16]

$$\mathbf{z} = \phi_0(\mathbf{b} - \mathbf{K}\mathbf{l}, \boldsymbol{\pi_0}) = \sum_{j=1}^{M} \pi_{0j}\exp(-\frac{\alpha}{2}(\mathbf{b} - \mathbf{K}\mathbf{l} - \mu_j)^2), \tag{11}$$

$$\mathbf{v}_i = \phi_i(\mathbf{F}_i(\mathbf{b} - \mathbf{K}\mathbf{l}), \boldsymbol{\pi_i}) = \sum_{j=1}^{M} \pi_{ij}\exp(-\frac{\alpha}{2}(\mathbf{F}_i(\mathbf{b} - \mathbf{K}\mathbf{l}) - \mu_j)^2), \tag{12}$$

where $\boldsymbol{\pi_i} = \{\pi_{ij}|j = 1, \ldots, M\}$ are the weights for each component. Similar to [16], we assume that the mean μ_j and variance α are fixed, which allows fast prediction and learning. The learned shrinkage functions are cheap to compute and can be pre-computed and stored in look-up tables.

Learning the shrinkage function is more expressive than learning the penalty function. Given a penalty function, a linear combination of Gaussian RBFs can well approximate its shrinkage function. As shown in Figure 1, the approximated functions by Gaussian RBFs match well the shrinkage functions of the optimization problem with the ℓ_1 and Lorentzian penalty functions. Furthermore, we can learn non-monotonic shrinkage functions, which cannot be obtained by learning the penalty functions [16]. More details will be discussed in Section 5.



(a) (b) (c) (d)

Fig. 1. The flexibility of the Gaussian RBFs to approximate the shrinkage function of the optimization problem (6) with different penalty functions. (a) and (c) plot the shapes of the ℓ_1 and Lorentzian penalty functions, respectively. The purple lines in (b) and (d) draw the corresponding shrinkage functions of (6) with the ℓ_1 and Lorentzian penalty functions, which are stated in the legend, respectively. The yellow dashdot lines in (b) and (d) plot the approximated shrinkage functions (using Gaussian RBFs), which match well the associated shrinkage functions

3.3 Cascaded Training

By directly learning the shrinkage functions for (6) and (7), we can compute the gradients of the recovered latent image w.r.t. the model parameters in closed forms. This allows efficient parameter learning. The half-quadratic optimization involves several iterations of (8), (10), (11), and (12) and we refer to one iteration as one stage. As noted by [16, 2], the model parameters should be adaptively tuned at each different stage. In the first few stages, the data term should be learned to detect useful information from the blurred image and avoid the effect of significant outliers. In the later stages, the data term should mainly focus on recovering clearer images with finer details. To learn the stage-dependent model parameters $\Omega_t = \{\lambda_t, \beta_t, \boldsymbol{\pi}_{ti}, f_{ti}\}$[6] for stage t from a set of S training samples $\{\mathbf{l}_{gt}^{\{s\}}, \mathbf{b}^{\{s\}}, k^{\{s\}}\}_{s=1}^{S}$, we use the negative peak signal-to-noise ratio (PSNR) as the loss function

$$J(\Omega_t) = \sum_{s=1}^{S} \mathcal{L}(\mathbf{l}_t^{\{s\}}, \mathbf{l}_{gt}^{\{s\}}) = \sum_{s=1}^{S} -10 \log_{10} \left(\frac{I_{\max}^2}{\mathrm{MSE}\left(\mathbf{l}_t^{\{s\}}, \mathbf{l}_{gt}^{\{s\}}\right)} \right), \tag{13}$$

where $I_{\max}$ denotes the maximum pixel value of the ground truth image, and $\mathrm{MSE}(\mathbf{l}_t^{\{s\}}, \mathbf{l}_{gt}^{\{s\}})$ is the mean squared error between $\mathbf{l}_t^{\{s\}}$ and $\mathbf{l}_{gt}^{\{s\}}$. We use the gradient-based L-BFGS method to minimize (13). To simplify notations, we omit the superscripts below. At stage t, the gradient of the loss function w.r.t. the parameters $\Omega_t = \{\lambda_t, \beta_t, \boldsymbol{\pi}_{ti}, f_{ti}\}$ is

$$\frac{\partial \mathcal{L}(\mathbf{l}_t, \mathbf{l}_{gt})}{\partial \Omega_t} = \frac{\partial \mathcal{L}(\mathbf{l}_t, \mathbf{l}_{gt})}{\partial \mathbf{l}_t} \cdot \frac{\partial \mathbf{l}_t}{\partial \Omega_t} = \frac{\partial \mathcal{L}(\mathbf{l}_t, \mathbf{l}_{gt})}{\partial \mathbf{l}_t} \zeta_t^{-1} \left[\frac{\partial \xi_t}{\partial \Omega_t} - \frac{\partial \zeta_t}{\partial \Omega_t} \mathbf{l}_t \right]. \tag{14}$$

The derivatives for specific model parameters are provided in the supplementary material. We optimize the loss function in a cascaded way and discriminatively

[6] The parameters τ and γ are included into the weights $\boldsymbol{\pi}_i$ of (11) and (12) and fused with β and λ in (10).

Algorithm 1 Discriminative learning algorithm

Input: Blurred images $\{\mathbf{b}^{\{s\}}\}$ and ground truth blur kernels $\{k^{\{s\}}\}$.
Initialization: $\mathbf{l}_0 = \mathbf{b}$, $\lambda_0 = 0.3, \beta_0 = 0.001$.
for $t = 1$ to T **do**
  Update $\mathbf{z}_t$ using (11).
  Update $\mathbf{v}_{ti}$ using (12).
  Update $\mathbf{u}_t$ using (8).
  Update $\mathbf{l}_t$ using (10).
  Update $\{\lambda_t, \beta_t, \boldsymbol{\pi}_{ti}, f_{ti}\}$ using the gradients (14).
end for
Output: Learned model parameters $\{\lambda_t, \beta_t, \boldsymbol{\pi}_{ti}, f_{ti}\}_{t=1}^{T}$.

learn the model parameters stage by stage. The main steps of the discriminative learning procedure are summarized in Algorithm 1.

4 Experimental Results

Experimental setup. To generate the training data for our experiments, we generate 20 blur kernels according to [17]. We use 200 images from the BSDS dataset [11] to construct the datasets for the experiments with significant noise (e.g., Gaussian noise, impulse noise, etc.) and crop a 280×280 patch from each of the images. These blurred images are then corrupted with various noise levels (i.e., the proportion of pixels affected by noise in an image). The test dataset is generated similarly, but with 20 blur kernels from [16] and 200 images from [11], which has no overlap with the training data. In addition, we create a dataset containing 100 ground-truth low-light images with saturated pixels, one half for the training and one half for the test. These images are resized to the size of 600×800 pixels. For each noise type, we train one discriminative model using blurred images corrupted by the noise of different levels (1%-5%), such that this range can cover the possible noise levels in practice. We use real captured examples to illustrate the robustness of the proposed method to real unknown noise and estimated kernels. We set T, N_f, and filter size to be 20, 8, and 3×3, respectively. More results are included in the supplemental material. The MATLAB code is publicly available on the authors' websites.

Gaussian noise. We first evaluate the proposed method and the state-of-the-art non-blind deblurring methods using blurred images corrupted by Gaussian noise. Table 1 summarizes the PSNR and SSIM results. Since most existing deblurring approaches have been developed for small Gaussian noise, all methods have the reasonable performance at low noise levels (1% and 2%). Under such conditions, performance mainly depends on the prior for the latent image. It is reasonable that EPLL [29] performs best because it uses an expressive, high-order prior for the latent image. CSF [16] also focuses on learning effective image priors. We train the model for CSF using the same data as ours, which contains blurred images with different noise levels. CSF [16] achieves balanced results for all

Table 1. PSNR/SSIM results on blurred images corrupted by Gaussian noise at different levels

Noise level	TVL2 [7]	EPLL [29]	MLP [18]	CSF [16]	TVL1 [25]	Whyte [22]	Cho [3]	FCN [27]	Ours
1%	27.00/0.7844	**27.64/0.7975**	26.49/0.6000	25.55/0.7017	26.52/0.7561	25.48/0.6829	27.24/0.7833	26.46/0.7416	26.47/0.7457
2%	25.49/0.6944	**25.90/0.7068**	24.92/0.6002	25.36/0.6991	25.55/0.7008	22.95/0.5213	25.74/0.6780	25.18/0.6837	25.59/0.7021
3%	24.54/0.6337	24.86/0.6456	23.90/0.5951	**24.99/0.6637**	24.73/0.6493	20.78/0.4096	24.92/0.6562	24.47/0.6510	24.86/0.6551
4%	23.86/0.5901	24.12/0.6011	23.21/0.5889	**24.33/0.6317**	24.04/0.6038	19.10/0.3330	23.95/0.5932	23.69/0.5811	24.26/0.6186
5%	23.35/0.5576	23.56/0.5667	22.71/0.5787	23.45/0.5855	23.42/0.5633	17.77/0.2800	23.35/0.5537	23.61/0.5665	**23.73/0.5901**

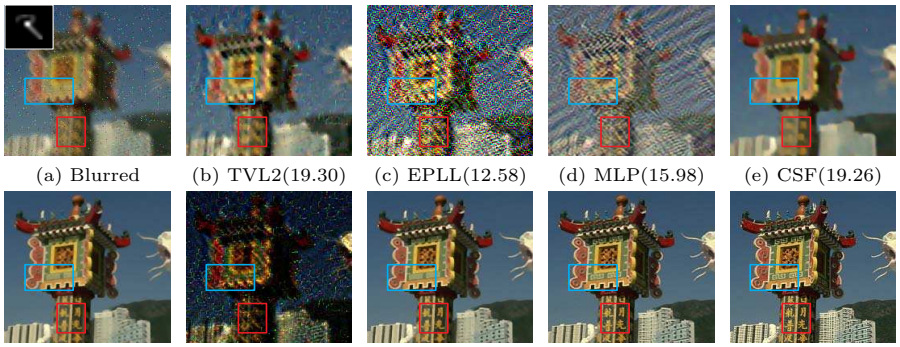
Table 2. PSNR/SSIM results on blurred images corrupted by impulse noise at different levels

Noise level	TVL2 [7]	EPLL [29]	CSF [16]	TVL1 [25]	Whyte [22]	Cho [3]	Ours
1%	22.34/0.5400	13.59/0.2514	23.48/0.5634	27.10/0.7873	18.57/0.4262	27.73/0.8033	**29.75/0.9143**
2%	20.64/0.4556	15.48/0.5397	23.04/0.5397	26.99/0.7849	16.39/0.2992	27.65/0.8034	**29.39/0.9075**
3%	19.23/0.3867	17.79/0.5179	22.62/0.5179	26.89/0.7823	14.88/0.2358	27.57/0.8039	**29.09/0.9005**
4%	17.94/0.3251	19.66/0.4906	22.11/0.4906	26.74/0.7797	13.59/0.1907	27.41/0.8020	**28.76/0.8920**
5%	16.84/0.2809	20.87/0.4658	21.62/0.4658	26.61/0.7772	12.49/0.1622	27.24/0.8012	**28.47/0.8828**

levels and distinctive performance at noise levels 3% and 4%. As the noise level becomes higher, the data term begins to play an important role. Whyte [22] performs poorly because it has been designed for saturated pixels. In addition, TVL2 [7], TVL1 [25], and Cho [3] need specifically hand-crafted tuning for the model parameters, which will significantly affect the deblurred results. Some deep learning-based methods [18, 27] also aim to learn effective image priors for the non-blind deblurring problem. For MLP [18], we use the model provided by the authors, which is trained for motion blur and Gaussian noise as stated in [18]. As demonstrated by Zhang et al. [27], their algorithm (FCN) is able to handle Gaussian noise. However, this method does not generate better results compared to the proposed algorithm when the noise level is high. In contrast, with a simple total variation prior and no manual designing, the proposed method performs comparably with other methods at various noise levels, suggesting the benefit of discriminatively learning the data term.

Impulse noise. Next we test on blurred images corrupted by impulse noise, as shown in Table 2 and Figure 2. As expected, methods developed for Gaussian noise do not perform well, including TVL2 [7], EPLL [29], and Whyte [22]. With the ℓ_2 norm-based data term, cascaded training the image prior by CSF [16] smooths the image details significantly. TVL1 [25] performs better by combining the standard TV prior with a robust data term based on ℓ_1 norm. Among existing methods, Cho [3] performs the best because it uses a hand-crafted model to handle impulse noise. Still, the proposed method has a concrete improvement over Cho [3], both numerically and visually, suggesting the benefits of correctly modeling the data term.

Other types of noise and saturation. We further evaluate these methods on blurred images corrupted by mixed noise (of Gaussian and impulse noise), Poisson noise, and saturated pixels, as shown in Tables 3 and 4. The proposed method consistently outperforms the state-of-the-art methods.



(a) Blurred (b) TVL2(19.30) (c) EPLL(12.58) (d) MLP(15.98) (e) CSF(19.26)

(f) TVL1(21.64) (g) Whyte(17.46) (h) Cho(22.35) (i) Ours(24.28) (j) GroundTruth

Fig. 2. Deblurred results on images corrupted by impulse noise. The numbers in parenthesis are PSNR values. Methods designed for Gaussian noise lead to strong artifacts (b-e, g). TVL1 (f) obtains some robustness to noise by using a ℓ_1 norm. Cho et al. [3] use a hand-crafted model to deal with impulse noise but the proposed method can better recover the details

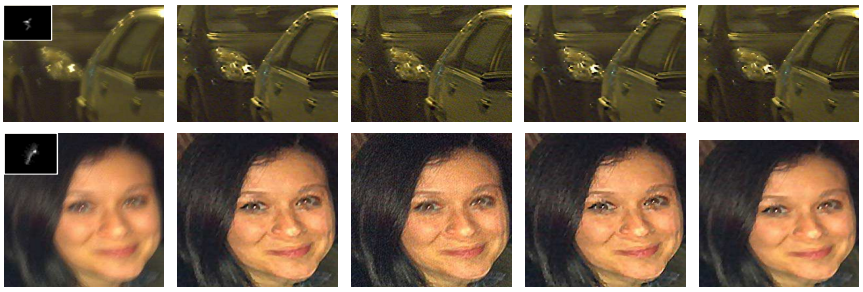
Table 3. PSNR/SSIM results on blurred images corrupted by mixed noise at different levels

Noise level	TVL2 [7]	EPLL [29]	MLP [18]	CSF [16]	TVL1 [25]	Whyte [22]	Cho [3]	Ours
1%	22.32/0.5364	13.42/0.2377	16.98/0.3628	23.43/0.5580	26.43/0.7537	18.42/0.3990	**27.15/0.7886**	26.23/0.7358
2%	20.55/0.4489	15.17/0.2622	18.58/0.4470	23.05/0.5422	25.40/0.6919	16.12/0.2693	24.87/0.6813	**25.50/0.6928**
3%	18.95/0.3716	17.37/0.3158	18.85/0.4270	22.58/0.5242	24.50/0.6409	14.45/0.2054	23.23/0.4884	**24.77/0.6462**
4%	17.47/0.3036	19.16/0.3678	18.84/0.4002	22.04/0.4968	23.73/0.5903	12.96/0.1627	22.89/0.3383	**24.14/0.6030**
5%	16.16/0.2518	20.37/0.4157	18.80/0.3826	21.42/0.4680	23.03/0.5459	11.65/0.1338	22.38/0.2488	**23.50/0.5611**

Table 4. PSNR/SSIM results on blurred images corrupted by Poisson noise and saturated pixels

Method	TVL2 [7]	EPLL [29]	CSF [16]	TVL1 [25]	Whyte [22]	Cho [3]	Ours
Poisson	23.95/0.5981	14.31/0.2083	23.90/0.5946	23.96/0.6043	19.22/0.3461	17.51/0.3526	**24.31/0.6214**
Saturation	23.07/0.7321	27.77/0.8804	26.72/0.8921	26.40/0.8610	22.85/0.7984	28.17/0.8918	**29.55/0.9351**

Evaluation on real-world images. The proposed method has been trained using images corrupted by known noise and outliers. One may wonder whether it works on real-world images, because their noise statistics are unknown and different from the training data. In this experiment, we apply the model trained for Gaussian noise to real-world images from the literature, since Gaussian noise is the predominant noise type in practice, as shown in Figure 3. The blur kernels are estimated by the kernel estimation methods [4] and [12]. The deblurred images by TVL1 and Cho et al. have ringing artifacts around the headlights because of saturated pixels. The method by Whyte et al. [22] can handle saturated pixels but significantly boosts the image noise. The recovered image by the proposed method is clearer and almost artifact-free, demonstrating the effectiveness and robustness of the proposed method to real unknown noise and inaccurate kernels.



(a) Blurred (c) TVL1 [25] (d) Whyte [22] (e) Cho [3] (f) Ours

Fig. 3. Real captured examples with unknown noise

5 Analysis and Discussions

As discussed above, most non-blind deconvolution methods are sensitive to significant noise and outlier pixels because of improper assumptions on the data term. Previous work [16] has shown the benefit of discriminatively learning the image prior for deblurring when the noise level is small. Our work has shown that discriminatively learning the data term can effectively deal with significant noise and outliers. In this section, we analyze the behavior of the proposed method to understand why it is effective.

Model properties and effectiveness. To understand what our algorithm learns from the training data, we plot in Figure 4 the learned shrinkage functions for the residue image (i.e., $\phi_0(\cdot)$, the approximated solution to the subproblem (6)), trade-off parameters, and the deblurred images at different stages of the proposed method on a test image with significant impulse noise. To facilitate the analysis, we assume that $N_f = 0$ in (4) in this section. We will discuss the effect of learning filters in the following sections.

The shrinkage function is initialized with a line-shaped function[7], as shown in the first column of Figure 4(b). Similar to most state-of-the-art methods, the latent image is initialized to be the blurred image, i.e., $l_0 = b$. To solve the optimization problems (6) - (9), we first apply the blur kernel k to the initial latent image. The convolution smoothes out impulse noise in the blurred image and the output, $k * l_0 = k * b$, is a smoothed version of the blurred image. Then the residual image $b - k * l_0 = b - k * b$ is only large at pixels that correspond to either salient structures or are corrupted by impulse noise in the input blurred image, as shown in the first column in Figure 4(a). With the initial line-shaped shrinkage function ϕ_0 in (11), the auxiliary variable at stage 1 becomes $z_1 = \phi_0(b - k * b) \approx b - k * b$ (see Figure 4(c)). Consequently, the first term of the non-blind deblurring problem (9) is close to $\|k * b - k * l\|_2^2$. The "effective" blurred image, $k * b$, is almost noise-free because the blurring operation suppresses the impulse noise.

[7] Note that we assume $\mathcal{R}_0(\cdot) = \|\cdot\|_2^2$ as the initialization in (6) and its closed-form solution is a line-shaped function.

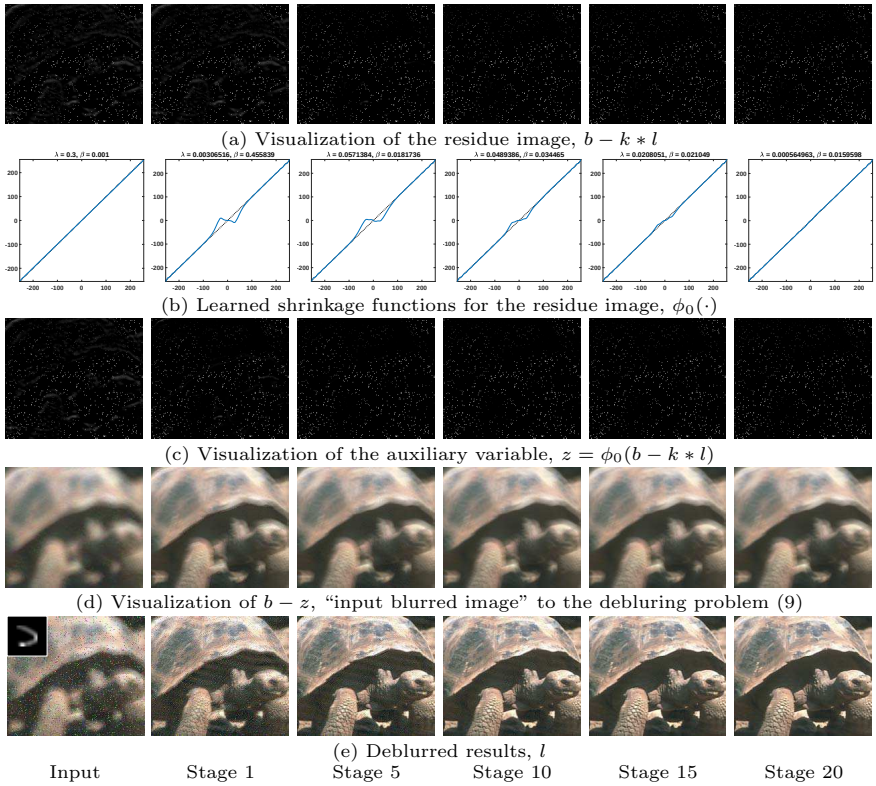
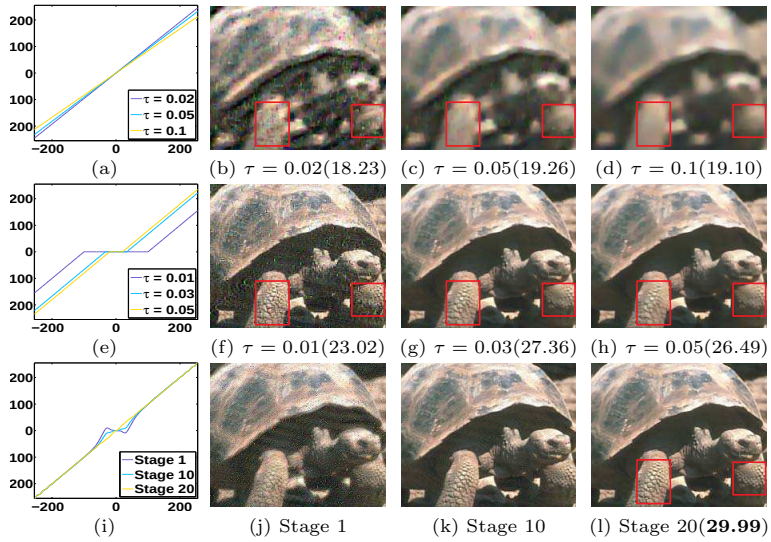


Fig. 4. Illustration of the proposed algorithm. The columns from left to right indicate different stages. Flexible shrinkage functions (b) are discriminatively learned for the residue image (a) at different stages, which can detect outliers (c) and suppress its effect on the "input" (d) of the deblurring problem (9). The proposed method can generate clear results with fine details (e)

At the next few stages, the learned shrinkage functions resemble soft-thresholding operators. At stage t, the auxiliary variable $z_t = \phi_0(b - k * l_{t-1})$ is close to $b - k * l_{t-1}$ if the latter is large and 0 otherwise (Figure 4(b)). The first (data) term in the deblurring problem (9) is close to $\|b - k * l\|_2^2$ when $z_t = 0$ and $\|k * l_{t-1} - k * l\|_2^2$ otherwise. That is, the proposed algorithm is learning to identify outlier pixels via the auxiliary variable z. As a result, inlier pixels are used but outliers are replaced with those of the smoothed image, $k * l_{t-1}$, which is almost noise-free.

At the final stages, the learned shrinkage functions become line-shaped functions again, since the deblurred image l_{t-1} becomes much clearer and the residue image $b - k * l_{t-1}$ mainly contains outliers (Figure 4(a)). This makes the first term of deblurring subproblem (9) close to $\|k * l_{t-1} - k * l\|_2^2$, where $k * l_{t-1}$ almost becomes a denoised version of the input blurred image, as shown in Figure 4(d). At these stages, the recovered images approach the ground truth l_{gt}, as shown in Figure 4(e).



(a) (b) $\tau = 0.02(18.23)$ (c) $\tau = 0.05(19.26)$ (d) $\tau = 0.1(19.10)$

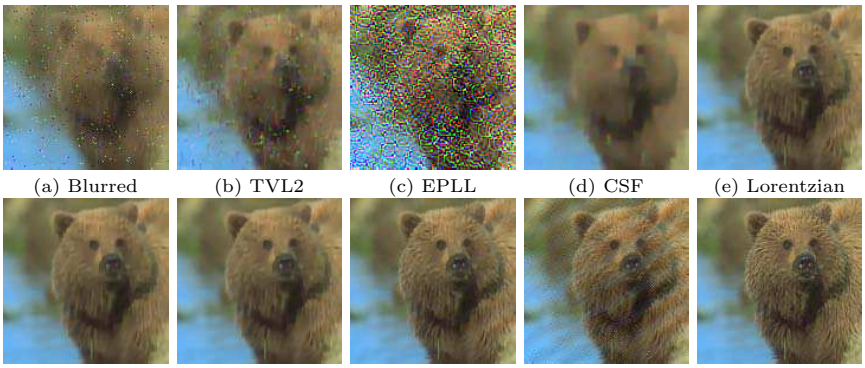
(e) (f) $\tau = 0.01(23.02)$ (g) $\tau = 0.03(27.36)$ (h) $\tau = 0.05(26.49)$

(i) (j) Stage 1 (k) Stage 10 (l) Stage 20(**29.99**)

Fig. 5. Effects of the learned shrinkage functions for (6). (a) and (e) are the solution functions of the subproblem (6) when $\mathcal{R}_0(\cdot)$ is ℓ_2 and ℓ_1 norm with different τ, respectively. (b)-(d) and (f)-(h) are the deblurred results with the corresponding τ using the ℓ_2 and ℓ_1 norm-based data term, respectively. (i) plots the learned shrinkage functions for (6) at different stages. (j)-(l) are the results restored at stage 1, 10, and 20. The method with the learned flexible shrinkage function generates clearer images with finer details (The numbers in the parenthesis denote the corresponding PSNR values.)

Figure 5 shows the effectiveness of the learned shrinkage functions. To ensure fair comparisons, we set $N_f = 0$ and let $\mathcal{R}_0(\cdot)$ be ℓ_2 norm, ℓ_1 norm, and the unknown penalty function associated with the learned shrinkage function. The shrinkage functions of the ℓ_2 norm-based data terms in Figure 5(a) cannot distinguish the impulse noise and smooth both image structure and noise. The method with ℓ_1 norm has some improvement. However, it heavily relies on the manual selection of the weight τ. Furthermore, some details of the latent images are smoothed out (Figure 5(h)). In contrast, the proposed method can discriminatively learn flexible shrinkage functions of (6) for different stages and generate clearer images with finer details (Figure 5(i)-(l)).

Effects of data terms. To further understand the role played by the data term when the blurred images contain outliers, Figure 6 shows the deblurred results by different approaches, including ℓ_2 norm-based methods with different image priors (TVL2 [7], EPLL [29], and CSF [16]), methods with robust data terms (Lorentzian function-based term and TVL1 [25]), Cho et al. [3], and the proposed method. When the blurred images contain significant impulse noise, methods using a data term based on the ℓ_2 norm all fail, despite the image priors, e.g., TVL2 [7], EPLL [29], and CSF [16]. Approaches with robust data term, e.g., Lorentzian data term, TVL1 [25], and Cho et al. [3], can handle outlier pixels



(a) Blurred (b) TVL2 (c) EPLL (d) CSF (e) Lorentzian

(f) TVL1 (g) Cho (h) Ours (i) Ours-s (j) Groundtruth

Fig. 6. Effectiveness of the proposed method compared with approaches with various spatial and data terms. The data term plays a more important role for images with significant impulse noise. Our method generates the images with fine details

(a) Blurred (b) EPLL (c) EPLL+Ours (d) Ours (j) GroundTruth

Fig. 7. Effectiveness of the proposed framework to improve EPLL for images with outliers

but may not recover fine details. In contrast, the proposed method generates clearer images with finer details.

Effects of stage-wise discriminative learning. Since our method uses stage-dependent parameters, one may wonder whether this feature is important. To answer this question, we learn a stage-independent data term, referred to as Ours-s in Figure 6(i). Although a stage-independent data term can recover the fur of the bear better than some existing methods, it causes severe artifacts. By comparison, the proposed algorithm using a stage-dependent data term recovers fine details without the artifacts. In addition, we note that our method performs better than EPLL when the level of Gaussian noise is high. This can also be attributed to the stage-wise discriminative learning. As the deblurring processes, the noise level becomes lower. The noise distribution changes to a certain extent. More importantly, as the deblurred result becomes clearer and cleaner, the roles of the data term and regularizer have also changed. Thus, it is necessary to discriminatively learn the trade-off parameter for each stage. Figure 6(h)-(i) shows that stage-dependent data terms are more effective than stage-independent ones.

Extension of the proposed algorithm. To understand the role of the data term, we have used a standard total variation prior for the latent image. It is

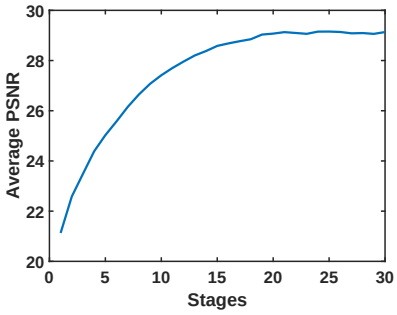


Fig. 8. Fast convergence property of the proposed algorithm

Table 5. Runtime (second)/PSNR comparisons for image deconvolution (of size 280× 280 with impulse noise and 600×800 with saturated pixels, respectively)

Method	EPLL [29]	CSF [16]	TVL1 [25]	Cho [3]	Ours
280 × 280	58.97/12.58	**1.98**/19.26	2.78/21.64	4.28/22.35	3.71/**24.28**
600 × 800	329.70/24.00	**13.79**/27.89	15.07/29.64	26.30/30.13	23.23/**31.63**

straightforward to combine the proposed data term with different image priors. As an example, we use EPLL [29] as the image prior and Figure 7 shows the deblurred results. While EPLL is sensitive to significant noise and outliers, combining its image prior with the proposed framework generates clearer results in Figure 7(c). Furthermore, using the more expressive EPLL prior better recovers fine details than the TV prior, such as the koala's fur.

Convergence and runtime. We empirically examine the convergence of the proposed method on the proposed dataset corrupted by impulse noise. The proposed algorithm converges in fewer than 20 steps, as shown in Figure 8. In addition, Table 5 summarizes the running time, which is based on an Intel Core i7-6700 CPU@3.40GHz and 16 GB RAM. The proposed method runs faster than Cho et al. [3]. Our method is not the fastest, but performs much better than all the other methods.

6 Conclusion

We have presented a discriminative learning framework for non-blind deconvolution with significant noise and outliers. In contrast to existing methods, we allow the data term and the tradeoff parameter to be discriminatively learned and stage-dependent. The proposed framework can also be applied to improve existing deconvolution methods with various image priors. Experimental results show that the proposed algorithm performs favorably against the state-of-the-art methods for different types of noise and outliers.

Acknowledgements. This work has been supported in part by the NSFC (No. 61572099) and NSF CAREER (No. 1149783).

References

1. Bar, L., Kiryati, N., Sochen, N.: Image deblurring in the presence of impulsive noise. IJCV **70**(3), 279–298 (2006)
2. Chen, Y., Yu, W., Pock, T.: On learning optimized reaction diffusion processes for effective image restoration. In: CVPR. pp. 87–90 (2015)
3. Cho, S., Wang, J., Lee, S.: Handling outliers in non-blind image deconvolution. In: ICCV. pp. 495–502 (2011)
4. Dong, J., Pan, J., Su, Z., Yang, M.H.: Blind image deblurring with outlier handling. In: ICCV. pp. 2478–2486 (2017)
5. Jancsary, J., Nowozin, S., Rother, C.: Loss-Specific Training of Non-Parametric Image Restoration Models: A New State of the Art. Springer Berlin Heidelberg (2012)
6. Jin, M., Roth, S., Favaro, P.: Noise-blind image deblurring. In: CVPR. pp. 3510–3518 (2017)
7. Krishnan, D., Fergus, R.: Fast image deconvolution using hyper-laplacian priors. In: NIPS. pp. 1033–1041 (2009)
8. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Understanding and evaluating blind deconvolution algorithms. In: CVPR. pp. 1964–1971 (2009)
9. Levin, A., Fergus, R., Durand, F., Freeman, W.T.: Image and depth from a conventional camera with a coded aperture. ACM TOG **26**(3), 70 (2007)
10. Lucy, L.B.: An iterative technique for the rectification of observed distributions. Astronomical Journal **79**(6), 745 (1974)
11. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV. pp. 416–423 (2001)
12. Pan, J., Lin, Z., Su, Z., Yang, M.H.: Robust kernel estimation with outliers handling for image deblurring. In: CVPR. pp. 2800–2808 (2016)
13. Richardson, W.H.: Bayesian-based iterative method of image restoration*. Journal of the Optical Society of America **62**(1), 55–59 (1972)
14. Roth, S., Black, M.J.: Fields of experts: A framework for learning image priors. In: CVPR. pp. 860–867 (2005)
15. Schmidt, U., Gao, Q., Roth, S.: A generative perspective on mrfs in low-level vision. In: CVPR. pp. 1751–1758 (2010)
16. Schmidt, U., Roth, S.: Shrinkage fields for effective image restoration. In: CVPR. pp. 2774–2781 (2014)
17. Schmidt, U., Rother, C., Nowozin, S., Jancsary, J., Roth, S.: Discriminative non-blind deblurring. In: CVPR. pp. 604–611 (2013)
18. Schuler, C.J., Burger, H.C., Harmeling, S., Scholkopf, B.: A machine learning approach for non-blind image deconvolution. In: CVPR. pp. 1067–1074 (2013)
19. Schuler, C.J., Hirsch, M., Harmeling, S., Schölkopf, B.: Learning to deblur. TPAMI **38**(7), 1439–1451 (2016)
20. Sun, D., Roth, S., Lewis, J.P., Black, M.J.: Learning optical flow. In: ECCV. pp. 83–97 (2008)
21. Wang, Y., Yang, J., Yin, W., Zhang, Y.: A new alternating minimization algorithm for total variation image reconstruction. Siam Journal on Imaging Sciences **1**(3), 248–272 (2008)
22. Whyte, O., Sivic, J., Zisserman, A.: Deblurring shaken and partially saturated images. IJCV **110**(2), 185–201 (2014)

23. Wiener, N.: Extrapolation, interpolation, and smoothing of stationary time series: With engineering applications. Mit Press **113**(21), 1043–54 (1949)
24. Xiao, L., Wang, J., Heidrich, W., Hirsch, M.: Learning high-order filters for efficient blind deconvolution of document photographs. In: ECCV. pp. 734–749 (2016)
25. Xu, L., Jia, J.: Two-phase kernel estimation for robust motion deblurring. In: ECCV. pp. 157–170 (2010)
26. Xu, L., Ren, J.S., Liu, C., Jia, J.: Deep convolutional neural network for image deconvolution. In: NIPS. pp. 1790–1798 (2014)
27. Zhang, J., Pan, J., Lai, W.S., Lau, R., Yang, M.H.: Learning fully convolutional networks for iterative non-blind deconvolution. In: CVPR. pp. 3817–3825 (2017)
28. Zhang, K., Zuo, W., Gu, S., Zhang, L.: Learning deep cnn denoiser prior for image restoration. In: CVPR. pp. 3929–3938 (2017)
29. Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: ICCV. pp. 479–486 (2012)