Variational-EM-based Deep Learning for Noise-blind Image Deblurring

Yuesong Nan[1], Yuhui Quan[2], and Hui Ji[1]

[1]Department of Mathematics, National University of Singapore, 119076, Singapore
[2]School of Computer Science & Engineering, South China University of Technology, Guangzhou 510006, China
nanyuesong@u.nus.edu, csyhquan@scut.edu.cn, and matjh@nus.edu.sg

Abstract

Non-blind deblurring is an important problem encountered in many image restoration tasks. The focus of non-blind deblurring is on how to suppress noise magnification during deblurring. In practice, it often happens that the noise level of input image is unknown and varies among different images. This paper aims at developing a deep learning framework for deblurring images with unknown noise level. Based on the framework of variational expectation maximization (EM), an iterative noise-blind deblurring scheme is proposed which integrates the estimation of noise level and the quantification of image prior uncertainty. Then, the proposed scheme is unrolled to a neural network (NN) where image prior is modeled by NN with uncertainty quantification. Extensive experiments showed that the proposed method not only outperformed existing noise-blind deblurring methods by a large margin, but also outperformed those state-of-the-art image deblurring methods designed/trained with known noise level.

1. Introduction

Image deblurring, recovering a clear image with sharp details from a blurred one, is an important problem with many applications. A blurring process is often modeled as

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

where $\boldsymbol{y}$ denotes the blurred image, $\boldsymbol{x}$ denotes the latent image, $\boldsymbol{k}$ denotes the blur kernel, and $\boldsymbol{n}$ denotes the measurement noise. The operator $\otimes$ represents the discrete convolution. When the kernel $\boldsymbol{k}$ is unknown, estimating $\boldsymbol{x}$ from $\boldsymbol{y}$ is called *blind deblurring*. Otherwise, it is called *non-blind deblurring*. In other words, blind deblurring needs to estimate both the kernel and latent image, while non-blind deblurring only needs to estimate the latent image.

With blur kernel in hand, how to deblur a noisy blurry image is a challenging ill-conditioned linear inverse problem. The blurring process will significantly attenuate or erase high-frequency information of an image, *i.e*. image details of small scale. Recovering such attenuated/missing high-frequency information is very sensitive to measurement noise. Thus, the study of non-blind image deblurring focuses on suppressing noise magnification while recovering all image details.

1.1. Motivation

The treatment on noise in image deblurring is usually done by imposing certain prior on the clear image $\boldsymbol{x}$ during recovery. The noise level plays an important role, as one needs to balance the strength of the regularization on latent image and the approximation error to the blurred image, *i.e*., the so called bias-variance trade-off in regression. For the images with different noise levels, most non-learning-based regularization methods adjust regularization parameters accordingly to fit the noise level. Thus, when using such a regularization method, one needs to call some separate noise level estimators in advance.

In recent years, learning-based image deblurring approaches, especially those deep-learning-based ones, are more preferred solutions with impressive performance. However, as the regularization/prior is trained using many pairs of latent images and their degraded counterparts, one has to train the model for a specific noise level to achieve optimal performance. In other words, one needs to train many NNs with respect to different noise levels, and store multiple such trained NNs for usage. When calling such a deep learning method to deblur a noisy blurred image, one needs to first determine its noise level and then call the corresponding NN. Such an proccedure is certainly not appealing in practice.

There is practical demand on developing deep-learning-based deblurring methods that are blind to noise level. In other words, the trained model should be a universal one that can deblur images with varying unknown noise level. So far, the works along this line are scant in existing liter-

ature. Kruse *et al*. [17] trained a universal CNN using the noisy samples with varying noise levels. The GradNet proposed in Jin *et al*. [14] combines a Bayesian MAP estimator of noise level and a CNN-based gradient descent for deblurring. Bigdeli *et al*. [4] proposed a denoising autoencoder that learns the mean-shift based gradient on image prior. Nevertheless, there is still much room for performance improvement on noise-blind deblurring, in comparison to the ones designed and trained with known noise level.

1.2. Main contribution

This paper aims at developing a deep learning method that can handle the images with unknown noise level, and provides the state-of-the-art performance on image deblurring. In this paper, built on the framework of variational expected maximization (VEM), we formulated a VEM approach to noise-blind image deblurring. Different from existing Bayesian approaches, the proposed VEM approach not only treats noise level as a distribution parameter integrated into the data fidelity term of the cost function, but also introduces a set of parameters to quantify the uncertainty of learned image prior w.r.t. each individual image. To the best of our knowledge, our approach is the first maximization likelihood (ML) estimator that quantify image prior uncertainty in image restoration. In other words, our formulation of image deblurring is a noise-adaptive framework that integrates both the estimation of noise-level and the quantification of learned image prior.

Based on the proposed VEM algorithm for noise-blind image deblurring, we developed a deep-learning-based computational scheme that uses a convolutional neural network (CNN) to learn image prior and uses a multi-layer perception network (MLP) to quantify prior uncertainty. Together with the integrated formula for estimating noise level, the proposed image deblurring NN provides a powerful solution to the problem of noise-blind image deblurring with the following contributions:

- A VEM-based iterative scheme is proposed for noise-blind deblurring, which integrates the estimations of both noise level and image prior uncertainty.
- By unrolling the VEM-based iteration, a deep learning method is developed which uses one NN to learn image prior and another NN to quantify prior uncertainty.
- One only needs to train one universal model to deblur the images with unknown noise level.
- Extensive experiments showed that, not only did the proposed method outperforms existing state-of-the-art noise-blind deblurring methods by a large margin, but also it noticeably outperforms those deep learning methods designed and trained for a specific known noise level.

2. Related Work

2.1. Deblurring images with known noise level

The main difference among different methods is the image prior used for regularization. There have been extensive studies on the priors of images for regularizing the deblurring. Wiener filter or Tikhonov methods assume the images or image gradients are smooth. Based on sparsity prior of image gradient, the ℓ_1-norm-relating regularization methods were developed for image deblurring, *e.g*. total variation (TV) methods (*e.g*. [23]), wavelet methods (*e.g*. [5, 13, 12]) and Hyper-Laplacian methods [16]. Another approach is the non-local ones, which assume the recurrence of similar local patches (*e.g*. [6, 9]). The most representative nonlocal method, IDD-BM3D [6], formulates the non-local prior as a regularization problem. These methods need to know the noise level in advance, and then they can adjust regularization parameters accordingly to achieve good performance.

Instead of using pre-defined priors, learning image prior is a more promising approach. Roth and Black [25] proposed to learn fields of experts to describe image gradients. Zoran and Weiss [35] developed the so-called Expected Patch Log-Likelihood (EPLL) algorithm, which uses a Gaussian mixture model (GMM) prior learned on clean image patches. Schmidt and Roth [27] used a shrinkage field to model image filters and shrinkage functions. The prevalence of deep learning also is observed in the recent advance of image deblurring. Early works along this line used the NN as a post-process to denoise the estimate from existing methods, *e.g*. Schuler *et al*. [28], Xu *et al*. [30], and Ren *et al*. [24]. A better approach to use deep learning is to unroll the iterative scheme of some existing method and replace the pre-defined priors by the NN-based learnable ones. Meinhardt *et al*. [22] unrolled the primal-dual hybrid gradient method. Zhang *et al*. [31] unrolled a half-quadratic splitting method with a CNN-based denoiser and learnable regularization parameters. These deblurring NNs need to be trained on a fixed noise level for good performance.

2.2. Deblurring methods with unknown noise level

A standard approach to deblurring blurred images with unknown noise levels is to take a two-stage approach: first estimate the noise level and then call some non-noise-blind method discussed in Sec. 2.1. There are several noise level estimators available, *e.g*. wavelet-based estimator [10], smoothness-based estimator [20] and PCA-based estimator [21]. These noise level estimators are designed for noisy but un-blurred images, not for the blurred ones. The kurtosis-value-based estimator proposed in [34] is applicable to noisy and blurred images, which is called in [35] to extend the EPLL method to noise-blind deblurring.

In recent years, there have been a few deep-learning-based methods proposed to tackle noise-blind image deblur-

ring. Kruse *et al*. [17] proposed the FDN method which unrolls an iterative scheme whose each stage contains an FFT-based deconvolution process and a CNN-based denoiser. For noise-blind deblurring, the FDN method trains the NN using the samples with noise level varying within a range and then deblurs images. Dong *et al*. [8] followed the same training procedures but with a learnable fidelity term of the cost function. Aiming at noise-blind deblurring, Jin *et al*. [14] proposed a Bayes MAP estimator that simultaneously estimates the noise level and deblurs image. The gradient descent method is then called and modeled by a network architecture called GradNet. Bigdeli *et al*. [4] proposed a Bayes estimator for image restoration with the so-called deep mean-shift priors, which uses a denoiser autoencoder to learn the density of natural images.

Despite existing efforts on exploiting the potential of deep learning in noise-blind image deblurring, there is still much room for performance improvement, especially when noise level is not high. Considering the discriminative power of deep learning demonstrated in many applications, there are certainly better ways to materialize the power of deep learning in noise-blind image deblurring.

3. VEM-based noise-blind image deblurring

Recall that non-blind image deblurring is about solving the following linear system (1), which can be interpreted as estimating the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{k})$ given the observed image $\boldsymbol{y}$ and the kernel $\boldsymbol{k}$. By the Bayes's rule, this posterior distribution can be decomposed into the likelihood term and the prior term, *i.e*.,

$$p(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{k}) \propto p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{k})p(\boldsymbol{x}). \tag{2}$$

Consider Gaussian white noise $\boldsymbol{n}$ with standard deviation σ. The likelihood term in (2) is then

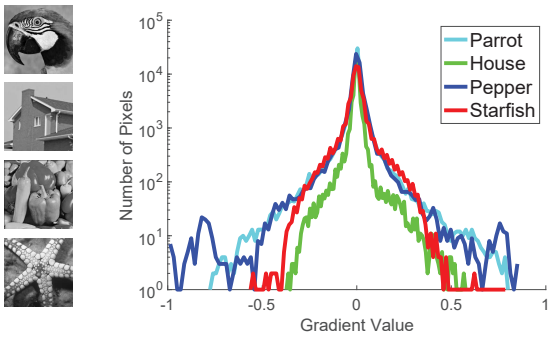
$$p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{k}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{x},\sigma^2\boldsymbol{I}).$$

The term $p(\boldsymbol{x})$ in (2) is critical which refers to the prior distribution of clear image $\boldsymbol{x}$. Instead of directly defining $p(\boldsymbol{x})$, most existing methods consider a prior distribution on image gradients $p(\nabla\boldsymbol{x}) = p(\frac{\partial\boldsymbol{x}}{\partial x}, \frac{\partial\boldsymbol{x}}{\partial y})$. For instance, based on the statistics on natural image gradients, the ℓ_p-norm relating regularization method [16] assumes $\nabla\boldsymbol{x}$ follows a Hyper-Laplacian distribution:

$$p(\nabla\boldsymbol{x}) \propto \prod_k e^{-\frac{(\nabla\boldsymbol{x}[k])^p}{\lambda}}, \quad 0 < p \leq 1. \tag{3}$$

Instead, deep-learning-based image deblurring methods use an NN to learn the prior distribution $p(\nabla\boldsymbol{x})$.

Bayes estimator. A Bayes estimator for noise-blind image deblurring needs to estimate both the latent image $\boldsymbol{x}$ and the noise distribution parameter σ. Recall that the prior $p(\boldsymbol{x})$ is for fitting image gradients of all natural images, not for one specific image. Vast variation of image content leads to noticeable variation among the gradient distributions of different images. See Fig. 1 for an illustration. Clearly, there is the need to take into consideration of the uncertainty of the prior $p(\nabla\boldsymbol{x})$ for each individual image. It can be done by introducing a latent variable $\boldsymbol{z}$ such that



(a) (b)

Figure 1: Variations of image gradient distributions among different images. (a) Four images with varying content; (b) the histogram of log magnitude of horizontal gradients of four images.

$$p(\nabla\boldsymbol{x}) = p(\nabla\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z}), \tag{4}$$

where $p(\boldsymbol{z})$ is a universal distribution function that fits most natural image gradients. The term $p(\nabla\boldsymbol{x}|\boldsymbol{z})$ measures the prior uncertainty of $\boldsymbol{x}$ w.r.t. $p(\boldsymbol{z})$. Suppose it follows a normal distribution, we have

$$p(\nabla\boldsymbol{x}|\boldsymbol{z}) \propto \mathcal{N}(\boldsymbol{z}, \Sigma).$$

To conclude, the proposed estimator that estimates two variables: image $\boldsymbol{x}$ and latent variable $\boldsymbol{z}$, and two parameters: noise level σ and uncertainty variance matrix Σ.

VEM-based formulation. The proposed Bayes estimator for deblurring is formulated via the VEM algorithm [3]. Consider a model involving the observed variable y and the latent variable z, parameterized by $\theta \in \Theta$. Instead of maximizing the marginal log-likelihood of parameters:

$$\max_{\theta\in\Theta} \log p(y;\theta) = \max_{\theta\in\Theta} \log(\int p(y,z;\theta)dz),$$

the EM algorithm maximizes a lower bound of $\log p(y;\theta)$:

$$\max_{q,\theta} F(q;\theta) = \max_{q,\theta} \int q(z) \log \frac{p(y,z;\theta)}{q(z)} dz.$$

The VEM method solves the optimization above by constraining probability distribution q inside some family of distributions Q. This optimization problem is solved by alternatively maximizing the function $F(q,\theta)$ between $q(z) \in Q$ and $\theta \in \Theta$:

1. *E-step*. Update $q(z)$ using θ^t:

$$\begin{aligned} q^{t+1} &= \text{argmax}_{q\in Q}\ E_{q(z)}[\log \tfrac{p(z,y;\theta^t)}{q(z)}] \\ &= \text{argmin}_{q\in Q}\ \text{KL}(q(z)||p(z|y;\theta^t)). \end{aligned} \tag{5}$$

2. *M-step*. Update θ using q^{t+1}:

$$\theta^{t+1} = \text{argmax}_{\theta\in\Theta} E_{q^{t+1}(z)}[\log p(z, y; \theta)], \quad (6)$$

where Θ denotes the feasible set of parameters, and $\text{KL}(q||p)$ denotes the KL-divergence between q and p. In the context of VEM, the noise-blind image deblurring problem is formulated as follows.

- The observed variable $\boldsymbol{y} \in R^N$.
- The latent variables $\boldsymbol{x} \in R^N, \boldsymbol{z} = \{\boldsymbol{z}_i \in R^N | i = 1, \ldots, L\}$. The latent variables $\boldsymbol{z} = \{\boldsymbol{z}_i\}_i$ follow a joint prior distribution $p(\boldsymbol{z})$. The latent variable $\boldsymbol{x}$ follows the conditional distribution based on $\boldsymbol{z}$

$$p(\boldsymbol{x}|\boldsymbol{z}) \propto \Pi_{i=1}^{L} \mathcal{N}(\boldsymbol{f}_i \otimes \boldsymbol{x}|\boldsymbol{z}_i, \lambda_i^2 \boldsymbol{I}), \quad (7)$$

  where $\{\boldsymbol{f}_i\}_{i=1}^{L}$ is a filter bank. The likelihood is $p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{k} \otimes \boldsymbol{x}, \sigma^2 \boldsymbol{I})$.
- The parameter set

$$\Theta = \{(\sigma, \lambda_i)|\sigma > 0, \lambda_i > 0, i = 1, \cdots, L\},$$

  where σ denotes the noise level, and the vector $\boldsymbol{\lambda} = [\lambda_i]_{i=1}^{L}$ denotes the standard deviation of the prior which follows a joint prior distribution $p(\boldsymbol{\lambda})$.

Remark 1 (Filter bank $\{\boldsymbol{f}_i\}_i$)**.** The filter bank of the linear-spine wavelet transform [7] is used in our approach, which contains totally 8 high-pass filters:

$$\{\boldsymbol{f}_i\}_{i=1}^{8} := \{h_{k_1} h_{k_2}^\top\}_{0\le k_1,k_2\le 2} \setminus \{h_0 h_0^\top\},$$

generated by the three 1D filters listed as follows,

$$h_0 = [\frac{1}{4}, \frac{1}{2}, \frac{1}{4}], h_1 = [-\frac{1}{4}, \frac{1}{2}, -\frac{1}{4}], h_2 = [\frac{\sqrt{2}}{4}, 0, -\frac{\sqrt{2}}{4}].$$

The motivation comes from the more-refined frequency sub-bands and its better performance in image recovery than the commonly-used $\{[1, -1], [1, -1]^\top\}$; see *e.g.* [7, 13, 12].

In the next, we only give a sketch of E-step and M-step. More details can be found in supplementary materials.

E-step. Provided an estimate $\theta := \theta^t \in \Theta$, the E-step estimates $q(\boldsymbol{x}, \boldsymbol{z}) \in Q$ via (5). Using Bayes rule, we have

$$\begin{aligned} p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{y}, \theta) &\propto p(\boldsymbol{y}|\boldsymbol{x}, \theta) p(\boldsymbol{x}|\boldsymbol{z}, \theta) p(\boldsymbol{z}) \\ &\propto \mathcal{N}(\boldsymbol{y}|\boldsymbol{k} \otimes \boldsymbol{x}, \sigma^2 \boldsymbol{I}) \Pi_{i=1}^{L} \mathcal{N}(\boldsymbol{f}_i \otimes \boldsymbol{x}|\boldsymbol{z}_i, \lambda_i^2 \boldsymbol{I}) p(\boldsymbol{z}). \end{aligned}$$

For computational feasibility, the set Q is restricted to normal distributions and Dirac delta distributions

$$\begin{aligned} Q = \{q_x q_z | q_x(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\nu}, \gamma^2 \boldsymbol{I}), \boldsymbol{\nu} \in R^N; \\ q_z(\boldsymbol{z}) = \Pi_i^L \delta(\boldsymbol{z}_i - \boldsymbol{\mu}_i), \boldsymbol{\mu}_i \in R^N\}. \end{aligned} \quad (8)$$

The variational distribution

$$q^{t+1}(\boldsymbol{x}, \boldsymbol{z}) = q_x^{t+1}(\boldsymbol{x}) q_z^{t+1}(\boldsymbol{z})$$

can be obtained with

$$q_x^{t+1}(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{x}^{t+1}, \gamma^2 \boldsymbol{I}), \quad q_z^{t+1}(\boldsymbol{z}) = \Pi_{i=1}^{L} \delta(\boldsymbol{z}_i - \boldsymbol{z}_i^{t+1}),$$

where $(\boldsymbol{x}^{t+1}, \{\boldsymbol{z}_i^{t+1}\}_i^L)$ is the expectation of the posterior $p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{y}, \theta)$ defined by

$$\begin{aligned} \boldsymbol{x}^{t+1} &= \underset{\boldsymbol{x}}{\text{argmin}} \frac{1}{\sigma^2} \|\boldsymbol{y} - \boldsymbol{k} \otimes \boldsymbol{x}\|_2^2 + \sum_{i=1}^{L} \frac{1}{\lambda_i^2} \|\boldsymbol{f}_i \otimes \boldsymbol{x} - \boldsymbol{z}_i\|_2^2, \\ \boldsymbol{z}^{t+1} &= \underset{\boldsymbol{z}}{\text{argmin}} \sum_{i=1}^{L} \frac{1}{\lambda_i^2} \|\boldsymbol{f}_i \otimes \boldsymbol{x} - \boldsymbol{z}_i\|_2^2 + \log p(\boldsymbol{z}). \end{aligned} \quad (9)$$

M-step. Provided the variational distribution $q^{t+1}(\boldsymbol{x}, \boldsymbol{z})$, the parameters set $\theta = \{\sigma, \boldsymbol{\lambda}\}$ is updated via (6). We can rewrite the distribution as

$$p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}, \theta) = p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z}, \theta) p(\boldsymbol{x}, \boldsymbol{z}) p(\theta).$$

Suppose that $\boldsymbol{\lambda}$ follows the prior distribution $p(\boldsymbol{\lambda})$ and σ follows an uniform distribution. By direct expanding the distribution above and taking the negative logarithm, we have

$$\begin{aligned} \theta^{t+1} = \underset{\sigma,\boldsymbol{\lambda}}{\text{argmin}} \mathbb{E}_{q^{t+1}(\boldsymbol{x},\boldsymbol{z})} \{ \frac{1}{2\sigma^2} \|\boldsymbol{y} - \boldsymbol{k} \otimes \boldsymbol{x}\|_2^2 + \frac{N}{2} \log(2\pi\sigma^2) \} \\ + \sum_{i=1}^{L} [\frac{1}{2\lambda_i^2} \|\boldsymbol{f}_i \otimes \boldsymbol{x} - \boldsymbol{z}_i\|_2^2 + \log(2\pi\lambda_i^2)] + \log(p(\boldsymbol{\lambda})). \end{aligned}$$

Taking the expectation over $q^{t+1}(\boldsymbol{x}, \boldsymbol{z})$, we have

$$\begin{aligned} \theta^{t+1} = \underset{\sigma,\boldsymbol{\lambda}}{\text{argmin}} \frac{1}{2\sigma^2} \|\boldsymbol{y} - \boldsymbol{k} \otimes \boldsymbol{x}\|_2^2 + \frac{\gamma^2}{2\sigma^2} \|\boldsymbol{k}\|^2 + \frac{N}{2} \log(2\pi\sigma^2) \\ + \sum_{i=1}^{L} [\frac{1}{2\lambda_i^2} \|\boldsymbol{f}_i \otimes \boldsymbol{x} - \boldsymbol{z}_i\|_2^2 + \frac{\gamma^2}{2\lambda_i^2} \|\boldsymbol{f}_i\|^2 + \frac{N}{2} \log(2\pi\lambda_i^2)] + \log(p(\boldsymbol{\lambda})). \end{aligned}$$

Finally, we have

$$\sigma^{t+1} = \{ \frac{1}{N} (\|\boldsymbol{y} - \boldsymbol{k} \otimes \boldsymbol{x}\|^2 + \gamma^2 \|\boldsymbol{k}\|^2) \}^{\frac{1}{2}} \quad (10a)$$

$$\begin{aligned} \boldsymbol{\lambda}^{t+1} = \underset{\boldsymbol{\lambda}}{\text{argmin}} \sum_{i=1}^{L} \{ \frac{1}{2\lambda_i^2} \|\boldsymbol{f}_i \otimes \boldsymbol{x} - \boldsymbol{z}_i\|_2^2 \\ + \frac{\gamma^2}{2\lambda_i^2} \|\boldsymbol{f}_i\|^2 + N \log(\lambda_i) \} + \log p(\boldsymbol{\lambda}) \end{aligned} \quad (10b)$$

In summary, the VEM algorithm is done by the alternating iteration between (9) and (10). See the below for the outline of the complete VEM-based iterative scheme:

$$\begin{aligned} \boldsymbol{x}^{t+1} &= \underset{\boldsymbol{x}}{\text{argmin}} \|\boldsymbol{y} - \boldsymbol{k} \otimes \boldsymbol{x}\|_2^2 + \sum_{i=1}^{L} (\frac{\sigma^t}{\lambda_i^t})^2 \|\boldsymbol{f}_i \otimes \boldsymbol{x} - \boldsymbol{z}_i^t\|_2^2, \\ \boldsymbol{z}^{t+1} &= \underset{\boldsymbol{z}}{\text{argmin}} \sum_{i=1}^{L} \frac{1}{(\lambda_i^t)^2} \|\boldsymbol{f}_i \otimes \boldsymbol{x}^{t+1} - \boldsymbol{z}_i\|_2^2 + \log p(\boldsymbol{z}), \\ \sigma^{t+1} &= \{ \frac{1}{N} (\|\boldsymbol{y} - \boldsymbol{k} \otimes \boldsymbol{x}^{t+1}\|^2 + \gamma^2 \|\boldsymbol{k}\|^2) \}^{\frac{1}{2}}, \\ \boldsymbol{\lambda}^{t+1} &= \underset{\boldsymbol{\lambda}}{\text{argmin}} \sum_{i=1}^{L} \{ \frac{1}{2\lambda_i^2} \|\boldsymbol{f}_i \otimes \boldsymbol{x}^{t+1} - \boldsymbol{z}_i^{t+1}\|_2^2 + \frac{\gamma^2}{2\lambda_i^2} \|\boldsymbol{f}_i\|^2 \\ &\quad + N \log(\lambda_i) \} + \log p(\boldsymbol{\lambda}). \end{aligned}$$

4. VEM-induced noise-blind deblurring NN

It can be seen from the VEM-based iterative scheme presented Sec. 3 that, the key part lies in the second and forth step in the scheme, which involves two prior distributions: $p(\boldsymbol{z})$ for the universal prior on natural image gradients, and $p(\boldsymbol{\lambda})$ for the prior on predicting the uncertainty of universal image prior w.r.t. individual image. We propose to learn these two priors via deep learning, and use NNs to model these two steps.

4.1. Updating $\{z_i^{t+1}\}_i$ with learnable prior in E-step

There are two steps in the E-step. The first is to update $\boldsymbol{x}$ by a least squares estimation, which can be efficiently solved via FFT. The second step on updating $\{\boldsymbol{z}_i^{t+1}\}_i$ can be viewed as a denoising process, *i.e.*, remove noises from current estimate of truth x^{t+1} using the prior $p(\boldsymbol{z})$. We proposed a denoising CNN [31, 32] based network, with a few modifications to fit our need.

Firstly, we train a CNN to remove noise in $\boldsymbol{x}^{t+1}$ and then output it to L high-pass channels: $\{\boldsymbol{f}_i \otimes\}_i$. In other words, the module takes the current estimate of $\boldsymbol{x}$ as input, and outputs a de-artifacted estimate, denoted by $\widetilde{\boldsymbol{x}}$. Then the variable $\boldsymbol{z}^{t+1}$ is defined as

$$\boldsymbol{z}^{t+1} := \{\boldsymbol{z}_i^{t+1}\}_i := \{\boldsymbol{f}_i \otimes \widetilde{\boldsymbol{x}}\}_i.$$

The motivation is that the L channels of an image from the filter bank $\{\boldsymbol{f}_i\}$ are not independent. Our approach keeps such dependence in the output of the module.

Secondly, instead of only taking $\boldsymbol{x}^{t+1}$ as input, the CNN concatenates all previous estimates of truth image $\{\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^{t+1}\}$ as input. The reason is twofold. One is for alleviating possible vanishing gradients, the same as dense connection [11]. The other is that these previous estimates contain different aspects of the truth and different artifacts. Thus, fusing them is likely to to provide more information for refinement.

In short, the mapping for updating $\{\boldsymbol{z}^{t+1}\}$ is defined as:

$$\mathcal{D}^{t+1}(\cdot|\phi^{t+1}) : [\boldsymbol{x}^1, \cdots, \boldsymbol{x}^{t+1}] \to \widetilde{\boldsymbol{x}} \to \{\boldsymbol{z}_i^{t+1} := \boldsymbol{f}_i \otimes \widetilde{\boldsymbol{x}}\},$$

where ϕ^{t+1} denotes the parameters of $\mathcal{D}^{t+1}$. We use 17-block residual-CNN [32] with the structure Conv $\to$ BN $\to$ 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 set to 3×3. The channel size is set to 64.

4.2. Updating λ^{t+1} with learnable prior in M-step

In the M-step, one needs to predict both noise level σ and prior uncertainty parameter vector $\boldsymbol{\lambda}$. The noise level σ is updated by an explicit definition, determined by the residual and magnitude of the kernel. For $\boldsymbol{\lambda}$, it can be seen from (10b) that the vector $\boldsymbol{\lambda}$ is determined from the vector $\{\|\boldsymbol{f}_i \otimes \boldsymbol{x}^{t+1} - \boldsymbol{z}_i^{t+1}\| + \gamma^2 \|\boldsymbol{f}_i\|^2\}_{i=1}^L$. Thus we propose to use an MLP to predict $\boldsymbol{\lambda}$ from such a vector. The mapping for predicting $\{\boldsymbol{\lambda}^{t+1}\}$ is defined as

$$\mathcal{P}^{t+1}(\cdot|\psi^{t+1}) : \{\|\boldsymbol{f}_i \otimes \boldsymbol{x}^{t+1} - \boldsymbol{z}_i^{t+1}\| + \gamma^2 \|\boldsymbol{f}_i\|\}_{i=1}^L \to \boldsymbol{\lambda}^{t+1},$$

where ψ^{t+1} denotes the parameters of $\mathcal{P}^{t+1}$. Our MLP implements three Linear and LeakyReLU layers with 64 intermediate channels. As only the ratio $\sigma^{t+1}/\lambda_i^{t+1}$ is used when updating $\boldsymbol{x}$ in E-step. our map then directly outputs such a ratio, denoted by $\boldsymbol{\beta}^{t+1} := \{\beta_i\} = \{\frac{\sigma^{t+1}}{\lambda_i^{t+1}}\}$. See Fig. 3 for the diagram.

4.3. Overall architecture and other details

The proposed NN has totally $T+1$ stages, denoted by $\{\mathcal{S}_t\}_{t=1}^{T+1}$, corresponding to $T+1$ steps in VEM algorithm. See Fig. 2 for the outline of the proposed NN. The proposed NN generates a sequence of deconvolved images $\{\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^{T+1}\}$: for $1 \le t \le T$,

$$\begin{aligned} &S_i : (\boldsymbol{y}, \boldsymbol{k}, [\boldsymbol{x}^1, \cdots, \boldsymbol{x}^{i-1}], \boldsymbol{z}^{i-1}, \boldsymbol{\beta}^{i-1}) \to \boldsymbol{x}^i, \boldsymbol{z}^i, \boldsymbol{\beta}^i; \\ &S_{T+1} : (\boldsymbol{y}, \boldsymbol{k}, \boldsymbol{z}^T, \boldsymbol{\beta}^T) \to \boldsymbol{x}^{T+1}. \end{aligned} \tag{11}$$

In stage $\mathcal{S}_1$, the parameters $\boldsymbol{z}^0, \boldsymbol{\beta}^0$ are initialized by $\boldsymbol{z}^0 = \boldsymbol{0}, \beta_i^0 = 0.005/\|\boldsymbol{f}_i\|_1$ for $1 \le i \le L$. We set $T = 4$ in our implementation as performance gain is little after the 4-th stage. Such a phenomena is consistent with ℓ_1-norm-relating regularizations. For instance, 6 iterations in the ADMM method are often sufficient to have a satisfactory result. The loss function is defined as

$$\mathcal{L} := \|\boldsymbol{x}^{T+1} - \boldsymbol{x}\|_2^2 + \sum_{i=2}^{T} \mu_i \|\boldsymbol{x}^i - \boldsymbol{x}\|_2^2, \tag{12}$$

where $(\boldsymbol{x}, \boldsymbol{y})$ denotes the pair of truth images and their noisy blurred counterparts, and $\boldsymbol{x}^i$ denotes the output of S_i. The weight set $\{\mu_i\}_{i=2}^T$ are set to $4/5$ throughout all experiments. The cost function encourages both the final result and intermediate results close to the truth. This is for facilitating the training of NN. The parameter γ presets as 10^{-2}.

5. Experiments

5.1. Experimental settings

Training data. Following the same procedure for preparing the training data as [33, 17], we generate a set of 1500 latent images by randomly cropping the images in the BSDS500 dataset [1] into the ones of size 256×256. A set of 192 motion-blur kernels is synthesized using the procedure in [26] and then resized to different sizes ranging from 16×16 to 30×30. Then, blurred images are generated by convolving latent images with motion blur kernels randomly selected from 192 kernels under circular boundary extension. Gaussian white noise with different noise levels is then added to the blurred images

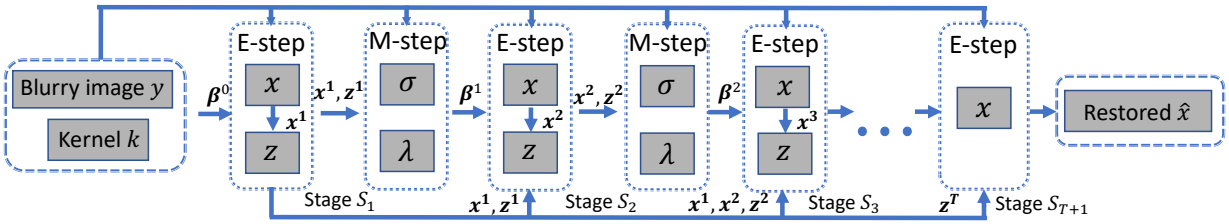


Figure 2: Diagram of the proposed NN for noise-blind image deblurring, which alternates between the E-step and the M-step in totally $T+1$ stages. Blocks named "$\boldsymbol{x}$", "$\boldsymbol{z}$" ,"σ", "$\boldsymbol{\lambda}$" represent the update procedures detailed in (3). for the corresponding variables.

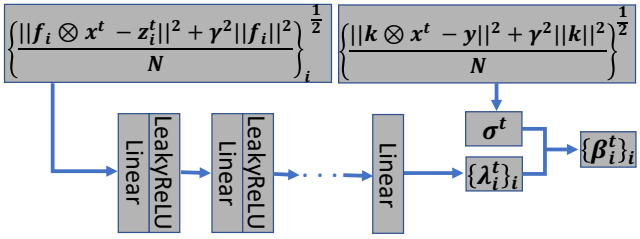


Figure 3: Diagram of the M-step for updating $\boldsymbol{\lambda}$ and σ for S_{t+1}.

Test data. We test on three datasets that are often used for image deblurring: Levin *et al.*'s dataset [18], Sun *et al.*' dataset [29] and Set12 [32]. Levin *et al.*'s dataset has 4 images with size 255×255. Sun *et al.*'s dataset has 80 clear images with size roughly 900×700. Set12 dataset has 7 256×256 images and 5 512×512 images widely used in the literature. Eight real motion-blur kernels from [18] are used in test data. Clear images are first blurred by these 8 motion-blur kernels, followed by adding Gaussian white noise with different σs. Following [31], we adopt the popular "edgetapper" method provided in MATLAB Image Processing Toolbox for simulating boundary condition of practical blurring. The number of noisy blurred images is: 32 from Levin *et al.*, 640 from Sun *et al.*, and 96 from Set12. The images and blur kernels in training data do not have any overlap with the images and blur kernels in testing data.

Training details. For initialization, the convolution weights of network is initialized by orthogonal matrices [**?**], and the biases are set to zeros. The model is trained using Adam method [15]. The model is trained for 500 epochs. The learning rate initializes by 1×10^{-3} and drops with rate 0.2 after 350 epochs. The NN is jointly trained from the scratch. See supplementary materials for more details.

Methods for comparison. Our method is compared to recent state-of-the-art deconvolution methods, as long as they have pre-trained models or codes available online. The methods for comparison include EPLL [35], CSF [27], IDD-BM3D[6], FCNN [31], FDN [17], IRCNN [33], EPLL+NA [14], GradNet [14], DMSP [4]. Regarding experimental results of these methods, whenever possible, we directly quote the results reported in the literature. Otherwise, we use the pre-trained models from the authors to generate the results. If only code is available, we made the effort on adjusting the parameters for optimal performance on test data. If none is available, we leave it blank.

5.2. Ablation study and Discussion

Most image deblurring networks have the CNN-based denoising module for removing artifacts. Thus, the ablation study focuses on studying performance gain brought by the introduction of M-step and the wavelet filter bank as the replacement of first-order difference operators. The NN is trained using the images whose noise level σ uniformly sampled from range $[1, 14]$. Then, we test the performance of such an NN on all images from Set12 with different noise levels. Results are shown in Table 1.

With vs. without parameter prediction in M-step. The results from the proposed VEM-Net are compared to that from the same one but replacing the M-step in VEM by a fixed vector $\boldsymbol{\beta}$. It can be seen from Table 1 that for low noise level, the perform gain by prediction NN of $\boldsymbol{\lambda}$ is around 0.4-0.7 dB and is around 0.2 dB for high noise level. Such a perform gain justified the necessity of the introduction of M-step for parameter prediction.

With vs without CNN-based prediction of $\boldsymbol{\lambda}$. The results from the proposed NN are compared to the same one but using a fixed vector tuned-up for optimal performance. It can be seen that from Table 1 that for low noise level, the perform gain of by prediction NN of $\boldsymbol{\lambda}$ is around 0.3-0.5 dB and is around 0.2 dB for high noise level. Such a perform gain justified the necessity of addressing prior uncertainty of individual images w.r.t. the distribution learned for fitting many training images.

Wavelet filter bank $\{\boldsymbol{f}_i\}_i$ vs difference operator ∇. Our NN replaced difference operator ∇ used in most denoising CNNs by a wavelet filter bank. The results from the NN with wavelet filter bank are slightly better to that from the same one with ∇. Table 1 shows that the overall performance gain using wavelet filter bank is around 0.2 dB.

Visualization of intermediate results. See Fig. 4 for the visualization of intermediate results at different stages of the proposed deblurring NN, when applying the trained model to deblur an image blurred by a motion-blur kernel of size 19×19 with noise level $\sigma = 7.65$. It can be seen that each stage noticeably improves the result until the third stage and

$\boldsymbol{y}$: 14.54 $\boldsymbol{x}^{(0)}$: 16.02 $\boldsymbol{x}^{(1)}$: 26.29 $\boldsymbol{x}^{(2)}$:27.14 $\boldsymbol{x}^{(3)}$: 27.71 $\boldsymbol{x}^{(4)}$: 27.84 Truth image

Figure 4: Visualization of intermediate results from different stages of the proposed NN, with PSNR values.

Table 1: Ablation study on the proposed NN.

σ	2.55	5.10	7.65	10.20	12.75
using fixed $\boldsymbol{\beta}$	31.14	29.38	28.15	27.29	26.52
using fixed $\boldsymbol{\lambda}$	31.36	29.47	28.24	27.33	26.57
using ∇	31.75	29.59	28.29	27.31	26.55
Ours	31.93	29.78	28.47	27.52	26.77

Table 2: The average prediction of the noise level $\hat{\sigma}$ from the M-step of the last stage of the proposed method.

True σ	1%	2%	3%	4%	5%
Predicted $\hat{\sigma}$	1.3%	2.0%	2.8%	3.7%	4.6%

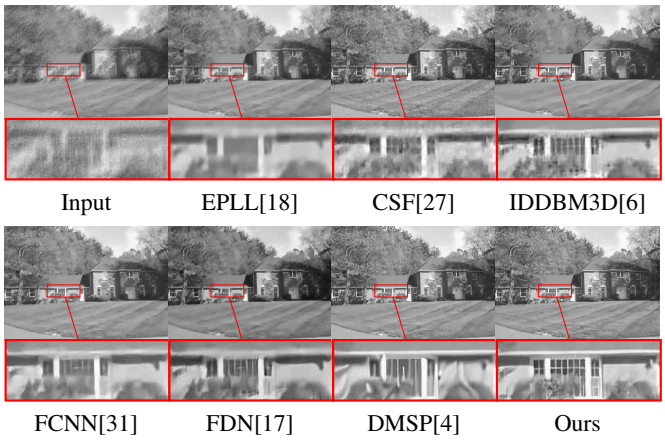
only minor gain is obtained at the forth stage, which is due to the usage of very deep denoising CNN for updating $\boldsymbol{z}_i$.

Prediction of noise level. See Table 2 for the noise level predicted by the M-step. The prediction from our method is quite accurate with around 10% bias.

5.3. Evaluation on noise-blind image deblurring

The proposed NN is designed for noise-blind image deblurring, *i.e.*, deblurring images with varying unknown noise levels. In the experiment, the proposed NN is trained using the images from training set whose noise level σ uniformly sampled from $[1, 14]$. Such a trained universal model is then tested by deblurring test images with 5 different noise levels. The method is compared to IDD-BM3D, FDN, EPLL-NA, GradNet7S and DMSP. Recall that EPLL-NA, GradNet7S and DMSP are all specially designed for noise-blind image deconvolution. The available model of FDN is trained using data with noise level σ in $[0.1, 12.75]$.

See Table 3 for the comparison of the results from the proposed methods and other methods. It can be seen that the results of our method are significantly better than that of all other methods over all noise levels. The performance gain by our method over the second best is: from 0.3 to 0.7 dB on Set12, from 0.4 to 0.5 dB on Sun *et al.*'s dataset, from 0.5 to 0.8 dB on Levin *et al.*'s dataset. The same conclusion holds true for performance gain in terms of SSIM by our method. It showed that the proposed VEM-based image deblurring NN has its merits. Together with adaptive model-uncertainty modeling and the adoption of wavelet filer bank, the proposed method showed its advantages over existing solutions in noise-blind deblurring.



Input EPLL[18] CSF[27] IDDBM3D[6]

FCNN[31] FDN[17] DMSP[4] Ours

Figure 5: Visual comparison of the results using different methods to deblur one example from Sun *et al.*'s dataset, whose noise level is $\sigma = 12.75$ and kernel size is 27×27.

The proposed method also showed its advantage over other method in terms of visual quality. See Fig 5 for the visual illustration of some results in the experiments. More examples can be found in supplementary materials.

5.4. Evaluation on other aspects

Comparison to the methods designed with known noise level. Intuitively, the methods designed for deblurring images with known noise levels should outperform noise-blind methods, as they can be optimized for specific noise level. It is interesting to see how good our noise-blind method is compared to those non-noise-blind ones. In this experiment, the compared methods are either trained using the data with the same noise level as test data or the related parameters are adjusted according to the noise level. In contrast, the results from our methods are directly quoted from Section 5.3. See Table 4 for the comparison. It shows that, even though our method trained an universal model to deblurring images with unknown varying noise levels, our universal model still noticeably outperformed those non-noise-blind methods by a large margin. Such a comparison demonstrated the merits of our VEM-based deblurring NN.

Image deblurring in the presence of Poisson noise. The statistical property of practical measurement noise can be complex. For instance, the images captured under low-lighting conditions are corrupted mostly by Poisson noise.

Table 3: Average PSNR(dB)/SSIM of the deblurring results with unknown varying noise levels using different methods.

Dataset	σ	IDD-BM3D[6]	FDN[17]	EPLL-NA[14]	GradNet7S[14]	DMSP[4]	Ours
Set12	2.55	31.43/0.88	31.43/0.89	- / -	- / -	31.06/0.87	**31.93/0.90**
	5.10	28.97/0.83	29.18/0.84	- / -	- / -	28.97/0.81	**29.78/0.86**
	7.65	27.56/0.80	27.89/0.81	- / -	- / -	27.87/0.79	**28.47/0.83**
	10.20	26.66/0.77	26.98/0.78	- / -	- / -	27.07/0.78	**27.52/0.80**
	12.75	25.95/0.75	26.28/0.76	- / -	- / -	26.39/0.76	**26.77/0.78**
Sun *et al.*'s	2.55	32.24/0.88	32.30/0.88	32.18/0.88	31.75/0.87	31.76/0.86	**32.73/0.90**
	5.10	29.95/0.82	30.12/0.82	30.08/0.83	29.31/0.80	29.62/0.80	**30.57/0.84**
	7.65	28.74/0.78	28.97/0.78	28.77/0.76	28.04/0.75	28.68/0.77	**29.41/0.81**
	10.20	27.93/0.75	28.21/0.75	27.81/0.74	27.81/0.73	28.06/0.75	**28.65/0.78**
	12.75	27.30/0.73	27.62/0.73	- / -	- / -	27.53/0.74	**28.04/0.76**
Levin *et al.*'s	2.55	33.75/0.92	33.65/0.93	32.16/0.92	31.43/0.91	32.61/0.91	**34.31/0.94**
	5.10	30.96/0.88	31.18/0.89	30.25/0.89	28.88/0.84	30.40/0.86	**32.02/0.91**
	7.65	29.26/0.85	29.79/0.86	28.96/0.86	27.55/0.80	29.31/0.84	**30.50/0.88**
	10.20	28.17/0.81	28.84/0.84	27.85/0.82	26.96/0.78	28.52/0.83	**29.42/0.86**
	12.75	27.33/0.79	28.02/0.82	- / -	- / -	27.79/0.82	**28.52/0.83**

Table 4: Average PSNR(dB)/SSIM of deblurring results using the methods w/ known noise levels and our noise-blind method.

Dataset	σ	EPLL[18]	CSF[27]	IDD-BM3D[6]	FCNN[31]	IRCNN[33]	Ours
Set12	2.55	27.61/0.85	29.37/0.85	31.43/0.89	30.68/0.88	30.53/0.82	**31.93/0.90**
	7.65	25.24/0.77	26.41/0.74	27.56/0.80	27.40/0.80	27.09/0.75	**28.47/0.83**
	12.75	23.87/0.72	25.10/0.68	25.95/0.75	25.84/0.75	- / -	**26.77/0.78**
Sun *et al.*'s	2.55	30.53/0.87	31.04/0.86	32.24/0.88	32.19/0.88	30.91/0.82	**32.73/0.90**
	7.65	27.46/0.75	27.84/0.73	28.74/0.78	28.93/0.78	27.93/0.74	**29.41/0.81**
	12.75	26.08/0.69	26.53/0.66	27.30/0.73	27.55/0.73	- / -	**28.04/0.78**
Levin *et al.*'s	2.55	32.03/0.92	29.85/0.88	33.75/0.92	33.10/0.93	32.66/0.87	**34.31/0.94**
	7.65	28.31/0.83	27.28/0.78	29.26/0.85	29.50/0.86	29.15/0.82	**30.50/0.88**
	12.75	27.15/0.75	26.25/0.72	27.33/0.79	27.81/0.82	- / -	**28.52/0.83**

Although our approach is not explicitly optimized for Poisson noise, the built-in treatment of model uncertainty via NN learning can be helpful to the robustness to different noise types. Thus, instead of training our model using images corrupted by Poisson noise, we directly apply our model trained using Gaussian white noise to deblur the images from Set12 corrupted by Poisson noise. See Table 5 for the comparison to the methods specifically designed for deblurring images with Poisson noise: VST-BM3D [2], RWL2 [19], and a deep learning method FCNN [31]. Our model still outperformed these methods. This clearly indicates the robustness to noise type brought by the treatment of model uncertainty in our method.

Table 5: Average PSNR(dB)/SSIM of the deblurring results w.r.t. Poisson noise with different peak values.

Peaks	VST-BM3D[2]	RWL2[19]	FCNN[31]	Ours
128	24.39/0.69	24.72/0.70	25.09/**0.72**	**25.69/0.72**
256	24.98/0.72	25.52/0.74	26.09/**0.76**	**26.69**/0.75
512	25.50/0.74	25.81/0.75	27.27/**0.79**	**27.93/0.79**
1024	- / -	26.30/0.76	27.95/0.81	**29.15/0.83**

6. Conclusion

This paper proposed a deep learning based method for noise-blind image deblurring. Based on a VEM-based method that integrates both noise level estimation and prior uncertainty quantification, a deblurring network is constructed by rolling the iterative VEM algorithm with CNN-based learnable image prior and MLP-based prediction of image prior uncertainty. The experiments showed the proposed method significantly outperformed existing methods.

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). Yuhui Quan would like to acknowledge the support from National Natural Science Foundation of China (61872151, U1611461).

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] Lucio Azzari and Alessandro Foi. Variance stabilization for noisy+ estimate combination in iterative poisson denoising. *IEEE signal processing letters*, 23(8):1086–1090, 2016.

[3] JM Bernardo, MJ Bayarri, JO Berger, AP Dawid, D Heckerman, AFM Smith, M West, et al. The variational bayesian em algorithm for incomplete data: with application to scoring graphical model structures. *Bayesian statistics*, 7:453–464, 2003.

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

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

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

[7] Bin Dong and Zuowei Shen. *MRA based wavelet frames and applications*, volume 19 of *IAS Lecture Notes Series*. Park City Mathematics Institute, 2010.

[8] Jiangxin Dong, Jinshan Pan, Deqing Sun, Zhixun Su, and Ming-Hsuan Yang. Learning data terms for non-blind deblurring. In *Proc. ECCV*, pages 748–763. Springer, 2018.

[9] 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.

[10] D. L. Donoho and J. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *biometrika*, 81(3):425–455, 1994.

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

[12] Hui Ji, Jia Li, Zuowei Shen, and Kang Wang. Image deconvolution using a characterization of sharp images in wavelet domain. *Appl. Comput. Harmonic Anal.*, 32(2):295–304, 2012.

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

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

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

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

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

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

[19] Jia Li, Zuowei Shen, Rujie Yin, and Xiaoqun Zhang. A reweighted l2 method for image restoration with poisson and mixed poisson-gaussian noise. *Inverse Probl. Imaging (Springfield)*, 9(3):875–894, 2015.

[20] C. Liu, W. T. Freeman, R. Szeliski, and S. Kang. Noise estimation from a single image. In *Proc. CVPR*, volume 1, pages 901–908. IEEE, 2006.

[21] X. Liu, M. Tanaka, and M. Okutomi. Single-image noise level estimation for blind denoising. *IEEE Trans. Image Process.*, 22(12):5226–5237, 2013.

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

[23] 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.

[24] 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 *Proc. NIPS*, pages 295–305, 2018.

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

[26] 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.

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

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

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

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

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

[32] 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.

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

[34] D. Zoran and Y. Weiss. Scale invariance and noise in natural images. In *Proc. ICCV*, pages 2209–2216. IEEE, 2009.

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