

Supplementary materials for "Variational-EM-based deep learning for noise-blind image deblurring"

Yuesong Nan¹, Yuhui Quan², and Hui Ji¹

¹Department of Mathematics, National University of Singapore, 119076, Singapore

²School 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}

1. Overview

The supplementary materials are organized as follows. In Section 2, we provide a detailed derivation of the VEM-based noise-blind deblurring procedure, which is outlined in the main paper. In Section 3, more details of experiments settings as well as runtime comparison is given. Section 4 is devoted to the visualization of some intermediate results from the proposed NN for interpretation and understanding. Section 5 presents visual results of deblurring images in the presence of Gaussian noise and Poisson noise, as well as deblurring real images using the kernel estimated from practical blind image deblurring methods.

2. Details on VEM-based noise-blind image deblurring

This section provides a detailed derivation of VEM-based noise-blind image deblurring framework outline in Section 3 of main manuscript.

2.1. E-Step

Provided an estimate $\theta := \theta^t \in \Theta$, the E-step estimates the $q(\mathbf{x}, \mathbf{z}) \in Q$ via minimization of KL divergence

$$q^{t+1}(\mathbf{x}, \mathbf{z}) = \underset{q \in Q}{\operatorname{argmin}} \text{KL}(q(\mathbf{x}, \mathbf{z}) || p(\mathbf{x}, \mathbf{z} | \mathbf{y}, \theta^t)). \quad (1)$$

Recall that

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

For computational feasibility, the feasible set Q is restricted to Gaussian distributions with fixed standard derivation $\gamma > 0$ and Dirac delta distributions

$$Q = \{q_x q_z | q_x(\mathbf{x}) = \mathcal{N}(\mathbf{x} | \boldsymbol{\nu}, \gamma^2 \mathbf{I}), \boldsymbol{\nu} \in R^N; q_z(\mathbf{z}) = \Pi_i^L \delta(\mathbf{z}_i - \boldsymbol{\mu}_i), \boldsymbol{\mu}_i \in R^N\}. \quad (3)$$

Then, by expanding (2) and then plugging it into (1), we have

$$\underset{\boldsymbol{\nu}, \{\boldsymbol{\mu}_i\}_i}{\operatorname{argmin}} \mathbb{E}_{q_x q_z} \log \frac{\mathcal{N}(\mathbf{x} | \boldsymbol{\nu}, \gamma^2 \mathbf{I}) \Pi_{i=1}^L \delta(\mathbf{z}_i - \boldsymbol{\mu}_i)}{\mathcal{N}(\mathbf{y} | \mathbf{k} \otimes \mathbf{x}, \sigma^2 \mathbf{I}) \Pi_{i=1}^L \mathcal{N}(\mathbf{f}_i \otimes \mathbf{x} | \mathbf{z}_i, \lambda_i^2 \mathbf{I}) p(\mathbf{z})}. \quad (4)$$

The optimization problem is separable for two variables $\boldsymbol{\nu}$ and $\boldsymbol{\mu}$. The problem (4) can be rewritten as:

$$\underset{\boldsymbol{\nu}}{\operatorname{argmin}} \mathbb{E}_{q_x} \log \frac{\mathcal{N}(\mathbf{x} | \boldsymbol{\nu}, \gamma^2 \mathbf{I})}{\mathcal{N}(\mathbf{y} | \mathbf{k} \otimes \mathbf{x}, \sigma^2 \mathbf{I}) \Pi_{i=1}^L \mathcal{N}(\mathbf{f}_i \otimes \mathbf{x} | \mathbf{z}_i, \lambda_i^2 \mathbf{I})}. \quad (5a)$$

$$\underset{\boldsymbol{\mu}}{\operatorname{argmin}} \mathbb{E}_{q_z} \log \frac{\Pi_{i=1}^L \delta(\mathbf{z}_i - \boldsymbol{\mu}_i)}{\Pi_{i=1}^L \mathcal{N}(\mathbf{f}_i \otimes \mathbf{x} | \mathbf{z}_i, \lambda_i^2 \mathbf{I}) p(\mathbf{z})}. \quad (5b)$$

The approximation of ν (5a) in q_x can be computed by expanding the PDF of two distributions $\mathcal{N}(\mathbf{y}|\mathbf{k} \otimes \mathbf{x}, \sigma^2 \mathbf{I})$ and $\mathcal{N}(\mathbf{y}|\mathbf{k} \otimes \mathbf{x}, \sigma^2 \mathbf{I})\Pi_{i=1}^L \mathcal{N}(\mathbf{f}_i \otimes \mathbf{x}|\mathbf{z}_i, \lambda_i^2 \mathbf{I})$ and then using the similar procedures when computing the KL divergence of two normal distributions. By a direct calculation, we can express (5a) as

$$\operatorname{argmin}_{\nu} \log \frac{\sigma \prod_{i=1}^L \lambda_i}{\gamma} + \frac{\gamma^2 \|\mathbf{k}\|^2}{2\sigma^2} + \sum_{i=1}^L \frac{\gamma^2 \|\mathbf{f}_i\|^2}{2\lambda_i^2} + \frac{\|\mathbf{y} - \mathbf{k} \otimes \nu\|^2}{2\sigma^2} + \sum_{i=1}^L \frac{\|\mathbf{f}_i \otimes \nu - \mathbf{z}_i\|^2}{2\lambda_i^2} - \frac{1}{2}. \quad (6)$$

With further simplification, we can conclude that

$$q_x = \mathcal{N}(\mathbf{x}|\nu^*, \gamma^2 \mathbf{I})$$

where ν^* is determined by

$$\nu^* = \operatorname{argmin}_{\nu} \frac{\|\mathbf{y} - \mathbf{k} \otimes \nu\|^2}{\sigma^2} + \sum_{i=1}^L \frac{\|\mathbf{f}_i \otimes \nu - \mathbf{z}_i\|^2}{\lambda_i^2}. \quad (7)$$

For (5b), we rewrite it into the integral form

$$\operatorname{argmin}_{\{\mu_i\}_i} \int \prod_{i=1}^L \delta(\mathbf{z}_i - \boldsymbol{\mu}_i) \log(\prod_{i=1}^L \delta(\mathbf{z}_i - \boldsymbol{\mu}_i)) d\mathbf{z} - \int \prod_{i=1}^L \delta(\mathbf{z}_i - \boldsymbol{\mu}_i) \log(\prod_{i=1}^L \mathcal{N}(\mathbf{f}_i \otimes \mathbf{x}|\mathbf{z}_i, \lambda_i^2 \mathbf{I}) p(\mathbf{z})) d\mathbf{z} \quad (8)$$

The first integration is zero. By expanding the second term, we have

$$\operatorname{argmin}_{\{\boldsymbol{\mu}_i\}_i} \sum_{i=1}^L -\log(\mathcal{N}(\mathbf{f}_i \otimes \mathbf{x}|\boldsymbol{\mu}_i, \lambda_i^2 \mathbf{I})) - \log(p(\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_L)). \quad (9)$$

The term $q_z(\mathbf{z}) = \prod_{i=1}^L \delta(\mathbf{z}_i - \boldsymbol{\mu}_i^*)$, where $\boldsymbol{\mu}^* = \{\boldsymbol{\mu}_i^*\}_i$ can be derived by

$$\boldsymbol{\mu}^* = \operatorname{argmin}_{\boldsymbol{\mu}} \sum_{i=1}^L \frac{\|\mathbf{f}_i \otimes \mathbf{x} - \boldsymbol{\mu}_i\|_2^2}{2\lambda_i^2} - \log(p(\boldsymbol{\mu})). \quad (10)$$

After obtaining q_x and q_z , the variational distribution $q(\mathbf{x}, \mathbf{z})$ can be written as

$$q(\mathbf{x}, \mathbf{z}) = q_x(\mathbf{x})q_z(\mathbf{z}) = \mathcal{N}(\mathbf{x}|\nu^*, \gamma^2 \mathbf{I})\prod_{i=1}^L \delta(\mathbf{z}_i - \boldsymbol{\mu}_i^*), \quad (11)$$

where ν^* and $\boldsymbol{\mu}^*$ are determined by (7) and (10).

By denoting

$$\mathbf{x}^{t+1} := \nu^* \quad \mathbf{z}^{t+1} := \boldsymbol{\mu}^*$$

for the $t+1$ step, the variational distribution can be rewritten as

$$q^{t+1}(\mathbf{x}, \mathbf{z}) = q_x(\mathbf{x})q_z(\mathbf{z}) = \mathcal{N}(\mathbf{x}|\mathbf{x}^{t+1}, \gamma^2 \mathbf{I})\prod_{i=1}^L \delta(\mathbf{z}_i - \mathbf{z}_i^{t+1}), \quad (12)$$

where the update procedure is defined by:

$$\begin{aligned} \mathbf{x}^{t+1} &= \operatorname{argmin}_{\mathbf{x}} \frac{\|\mathbf{y} - \mathbf{k} \otimes \mathbf{x}\|^2}{(\sigma^t)^2} + \sum_{i=1}^L \frac{\|\mathbf{f}_i \otimes \mathbf{x} - \mathbf{z}_i^t\|^2}{(\lambda_i^t)^2}, \\ \mathbf{z}^{t+1} &= \operatorname{argmin}_{\mathbf{z}} \sum_{i=1}^L \frac{\|\mathbf{f}_i \otimes \mathbf{x}^{t+1} - \mathbf{z}_i\|^2}{(\lambda_i^t)^2} - \log p(\mathbf{z}). \end{aligned} \quad (13)$$

2.2. M-Step

Provided the variational distribution $q^{t+1}(\mathbf{x}, \mathbf{z})$, the parameters set $\theta = \{\sigma, \boldsymbol{\lambda}\}$ is updated by

$$\theta^{t+1} = \operatorname{argmax}_{\theta} \mathbb{E}_{q^{t+1}(\mathbf{x}, \mathbf{z})} \log(p(\mathbf{x}, \mathbf{z}, \mathbf{y}, \theta)), \quad (14)$$

By writing

$$p(\mathbf{x}, \mathbf{z}, \mathbf{y}, \theta) \propto p(\mathbf{y}|\mathbf{x}, \mathbf{z}, \theta)p(\mathbf{x}|\mathbf{z}, \theta)p(\mathbf{z})p(\theta) \propto \mathcal{N}(\mathbf{y}|\mathbf{k} \otimes \mathbf{x}, \sigma^2 \mathbf{I}) \prod_{i=1}^L \mathcal{N}(\mathbf{f}_i \otimes \mathbf{x}|\mathbf{z}_i, \lambda_i^2 \mathbf{I}) p(\mathbf{z})p(\theta),$$

The $p(\theta) = p(\sigma)p(\boldsymbol{\lambda})$ where $p(\sigma)$ is a uniform distribution and $p(\boldsymbol{\lambda})$ is a joint distribution of $[\lambda_i]_i$. After expanding the distribution above and taking the logarithm, Eq (14) can be rewritten as:

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

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

$$\begin{aligned} \theta^{t+1} = \underset{\sigma, \boldsymbol{\lambda}}{\operatorname{argmin}} & \left\{ \frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{k} \otimes \mathbf{x}^{t+1}\|_2^2 + \frac{\gamma^2}{2\sigma^2} \|\mathbf{k}\|_2^2 + \sum_{i=1}^L \left[\frac{1}{2\lambda_i^2} \|\mathbf{f}_i \otimes \mathbf{x}^{t+1} - \mathbf{z}_i^{t+1}\|_2^2 + \frac{\gamma^2}{2\lambda_i^2} \|\mathbf{f}_i\|_2^2 + \frac{N}{2} \log(2\pi\lambda_i^2) \right] \right. \\ & \left. + \frac{N}{2} \log(2\pi\sigma^2) - \log(p(\boldsymbol{\lambda})) \right\}. \end{aligned} \quad (15)$$

The optimization problem above is separable for σ^{t+1} and $\boldsymbol{\lambda}^{t+1}$. Thus, we can solve the problem (15) separately:

$$\begin{aligned} \sigma^{t+1} &= \underset{\sigma}{\operatorname{argmin}} \frac{1}{2\sigma^2} \|\mathbf{y} - \mathbf{k} \otimes \mathbf{x}^{t+1}\|_2^2 + \frac{\gamma^2}{2\sigma^2} \|\mathbf{k}\|_2^2 + \frac{N}{2} \log(2\pi\sigma^2) \\ &= \left\{ \frac{1}{N} (\|\mathbf{y} - \mathbf{k} \otimes \mathbf{x}^{t+1}\|_2^2 + \gamma^2 \|\mathbf{k}\|_2^2) \right\}^{\frac{1}{2}} \end{aligned} \quad (16a)$$

$$\boldsymbol{\lambda}^{t+1} = \underset{\boldsymbol{\lambda}}{\operatorname{argmin}} \sum_{i=1}^L \left\{ \frac{1}{2\lambda_i^2} \|\mathbf{f}_i \otimes \mathbf{x}^{t+1} - \mathbf{z}_i^{t+1}\|_2^2 + \frac{\gamma^2}{2\lambda_i^2} \|\mathbf{f}_i\|_2^2 + N \log(\lambda_i) \right\} - \log p(\boldsymbol{\lambda}) \quad (16b)$$

In summary, the VEM iterations alternatively running E-step (13) for updating the variational distribution and running M-step (16) for updating distribution parameters.

3. Additional experimental settings

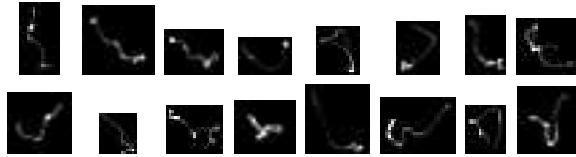


Figure 1: Examples of training kernels from a kernel set [8] with 192 motion kernels.



Figure 2: Eight test kernels from the kernel set [5].

Kernel Visualization See Fig. 1 for the samples of the kernels used in training from a 192 motion kernel set constructed by [8]. See Fig 2 for eight kernels from [5] used for test.

Metric Computation Follow the same procedures as [4][12], we discard the boundary of output images to compute the PSNR/SSIM. This is because the boundary is information-deficient, which will significantly degrade validity of metric.

Implementation details and runtime comparison. The experiments are conducted on a workstation with a 3.2GHz Intel Xeon E5-2620 v4 CPU, 64G RAM and a GeForce GTX 1080 Ti GPU. We used Pytorch to implement our proposed algorithm. As for runtime, the training took around 48 hours, while the comparison of the average testing time for images with size 256×256 is shown in the Table 1. We can find that our method is the second fastest method among all.

Table 1: The comparison of average testing time(s) for non-blind deconvolution methods when deconvoluting 256×256 images.

Method	CSF [9]	EPLL[15]	IDDBM3D [3]	FDN [4]	FCNN [12]	IRCNN [13]	DMSP [2]	Ours
Time(s)	0.08	105.18	115.83	0.15	0.02	0.31	182.40	0.07

4. Intermediate results of our proposed NN

Evaluation and visualization of intermediate deblurring results Some intermediate results of the proposed NN for deblurring are shown for understanding the network. Table 2 shows the average PSNR of intermediate results when deblurring images from Set12 with various noise levels. Fig 3 and Fig 4 visualize the intermediate examples from the same dataset with noise levels $\sigma = 7.65$ and $\sigma = 12.75$. It can be seen that the performance gain decreases gradually over the successive stages, which shows how each stage the result is refined and more artifacts are removed. This observation confirms the necessity of multiple iterations motivated from VEM frameworks in the proposed algorithm. As minor gain can be seen in the fourth stage, it is thus reasonable to use only $T = 4$ in our implementations.

Table 2: Average PSNR(dB) of the intermediate deblurring results for Set12 with various noise levels.

σ	$x^{(0)}$	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$	$x^{(4)}$
2.55	24.87	30.56	31.31	31.79	31.93
5.10	19.56	28.26	29.08	29.63	29.78
7.65	16.38	26.91	27.84	28.34	28.47
10.20	14.22	26.02	26.93	27.41	27.52
12.75	12.66	25.28	26.17	26.68	26.77

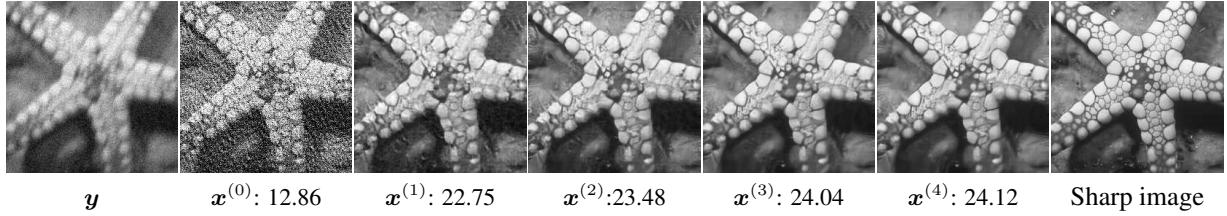


Figure 3: Intermediate results when deblurring one image from Set12. The noise level is set as $\sigma = 12.75$.

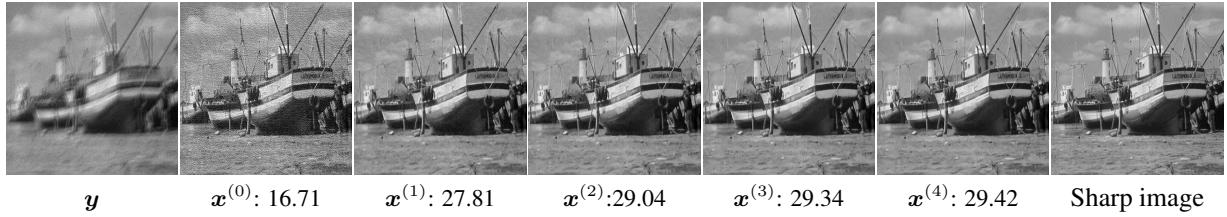


Figure 4: Intermediate results when deblurring one image from Set12. The noise level is set as $\sigma = 7.65$.

Visualization of the outputs from high-pass filter banks Our proposed methods apply linear-spine wavelet transform $\{f_i\}_{i=1}^8$ to E-step so as to extract high frequency information from denoised image z . Fig 5 visualizes some examples of the outputs from these high-pass information for better understanding.

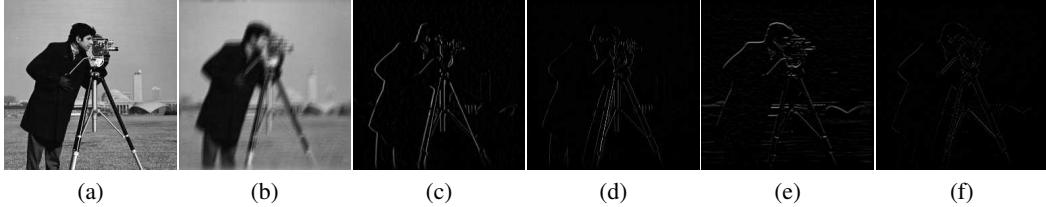


Figure 5: Visualization of outputs from high-pass filters. (a) Sharp image. (b) Blurry image. (c)-(f) Outputs from denoised image z filtering by high-pass filters $f_1 - f_4$.

5. Visualization and additional experiments

5.1. Visual comparison for non-blind deconvolution with Gaussian noise

In this section, we show extensive visual comparison among the state-of-the-art non-blind deconvolution methods including both noise-blind (unknown noise level) deconvolution algorithms IDD-BM3D[3], FDN [4], DMSP [2] and fixed-noise level algorithms including EPLL [15], CSF [9], IRCNN [13]. Fig 6 - 12 showed the visualization of the results from several examples from the dataset Set12 and Sun *et al.* [10] in the different noise levels $\sigma = 2.55, 7.65, 12.75$.

It can be seen that the results from our method are the best in terms of visual quality, which retained most fine details and contained least artifacts. The advantage on visual quality of the proposed method over existing is consistent with that on quantitative metric listed in main manuscript.

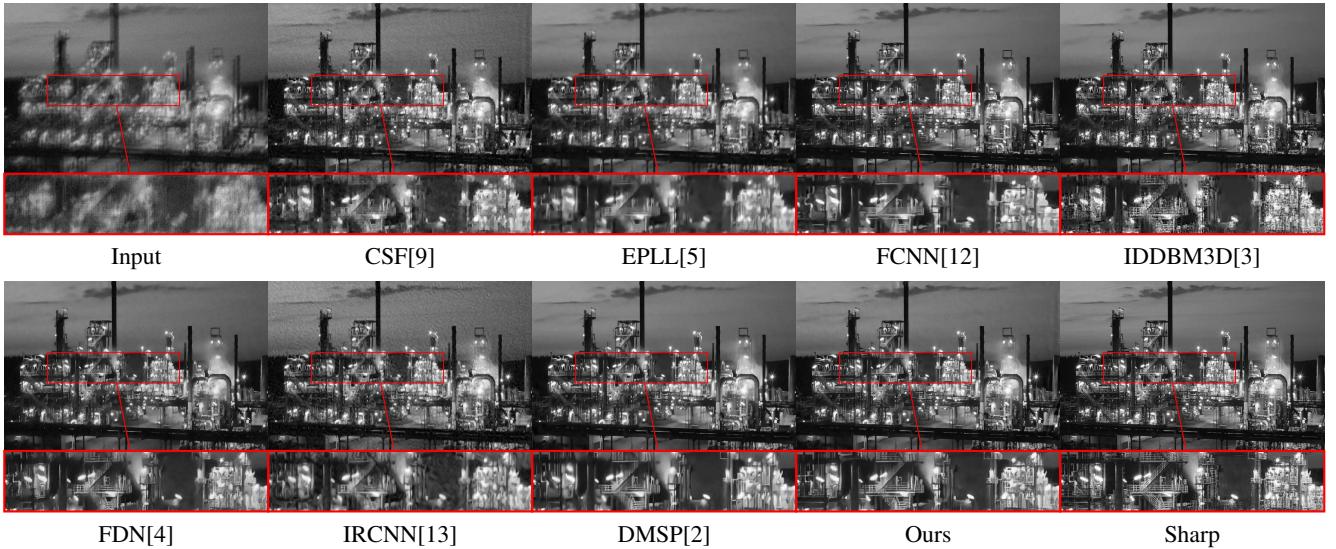


Figure 6: Visual comparison of the results of one example from Set12 dataset with noise level $\sigma = 7.65$ and kernel size 23×23 . Zoom-in for better inspection.

5.2. Visual comparison for deconvolution in the presence of Poisson noise

See Fig 13 - 14 for visual comparison of deconvolution results w.r.t Poisson noise. We compare our results with two methods specifically designed to address Poisson noise in deconvolution, including VST-BM3D [1], RWL2 [6] and one neural network method FCNN [12]. Note that both NNs, Ours and FCNN, are not re-trained with Poisson noise data in order to test the robustness to other noise types. From the experiments, our methods achieved best results in term of visual performance, the same as the results for Gaussian noise.

5.3. Visual comparison for Real images

In this section, we showed more examples when using the proposed method to deblur real images. For real images, the kernel is estimated by calling some representative motion blind deblurring method. In such a case, there is additional error in

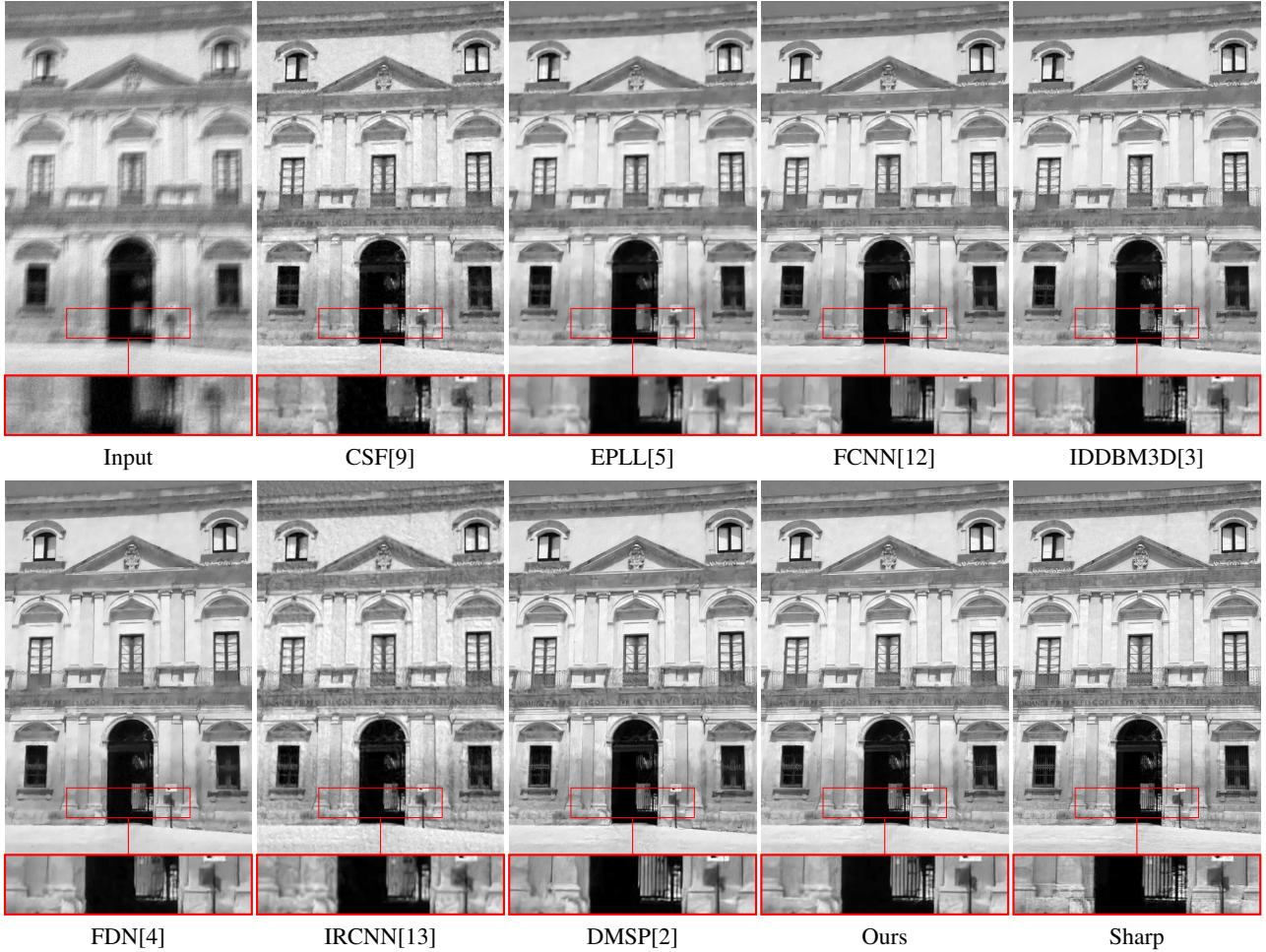


Figure 7: Visual comparison of the results of one example from Sun *et al.*'s dataset with noise level $\sigma = 7.65$ and kernel size 23×23 . Zoom-in for better inspection.

blur kernel. See Fig. 15 - 18 a visual inspection. As non-blind deblurring is sensitive to the noise/error in input blur kernel, the quality of the images deblurred by most methods is not as good as the case where blur kernel is error-free. In comparison to other methods, the proposed method still recovered more details.

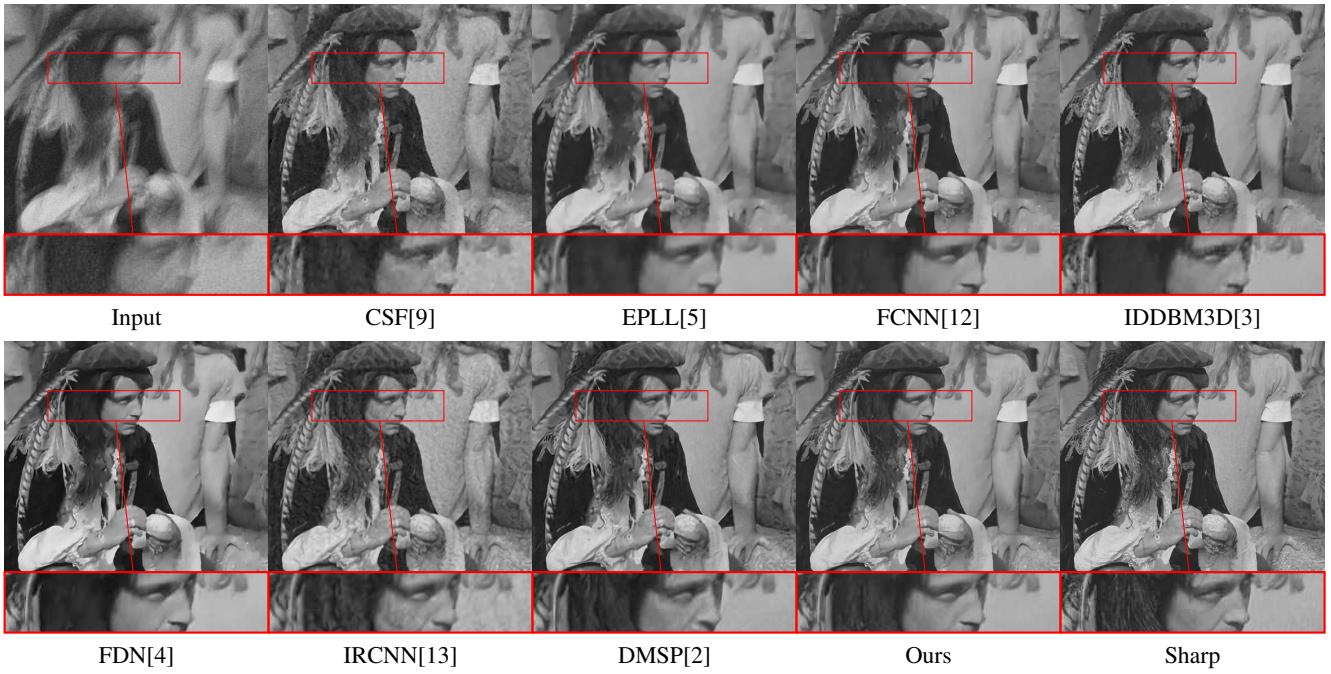


Figure 8: Visual comparison of the results of one example from Set12 dataset with noise level $\sigma = 7.65$ and kernel size 27×27 . Zoom-in for better inspection.

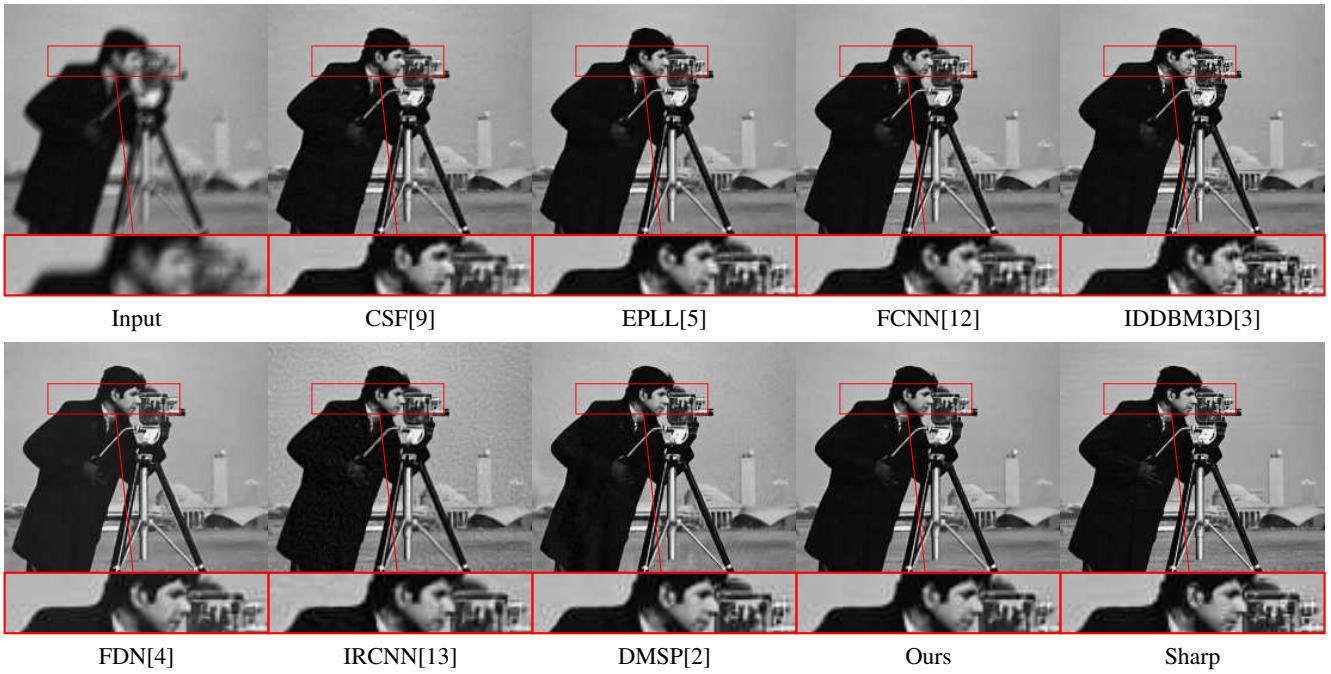


Figure 9: Visual comparison of the results of one example from Set12 dataset with noise level $\sigma = 2.55$ and kernel size 15×15 . Zoom-in for better inspection.

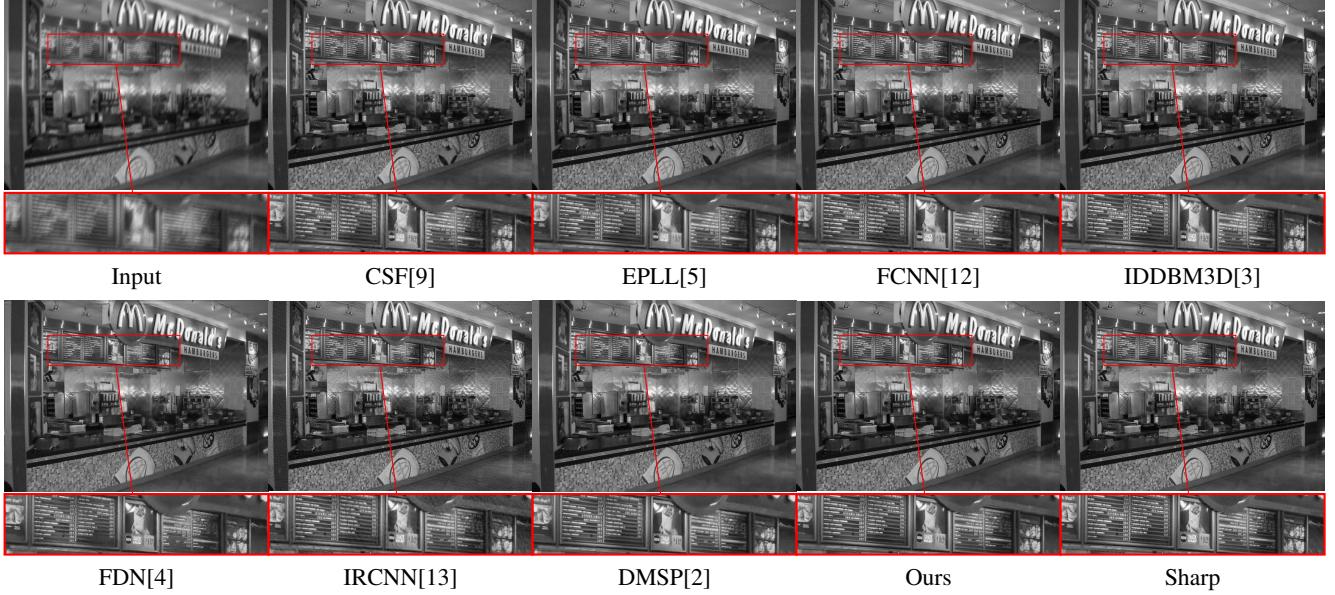


Figure 10: Visual comparison of the results of one example from Sun *et al.*'s dataset when training with noise level $\sigma = 2.55$ and kernel size 19×19 . Zoom-in for better inspection.

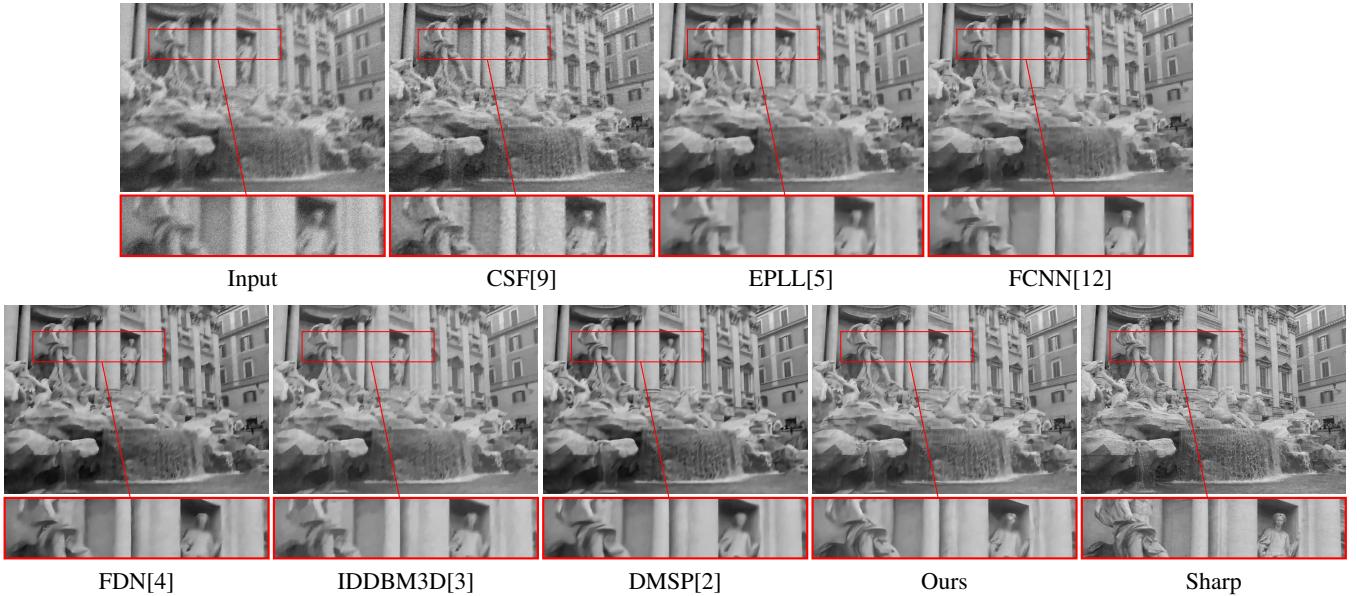


Figure 11: Visual comparison of the results of one example from Set12 dataset with noise level $\sigma = 12.55$ and kernel size 17×17 . Zoom-in for better inspection.

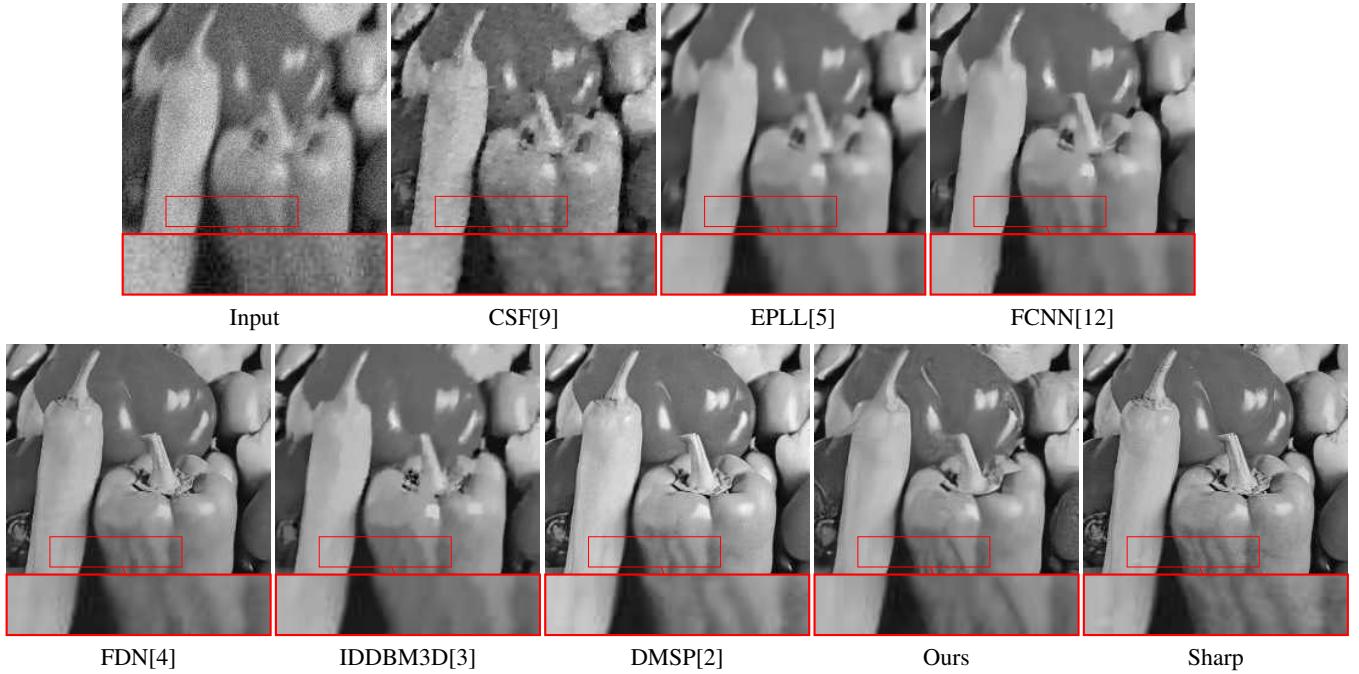


Figure 12: Visual comparison of the results of one example from Set12 dataset with noise level $\sigma = 12.55$ and kernel size 19×19 . Zoom-in for better inspection.



Figure 13: Visual comparison of deconvolution with Poisson noise for robustness test. The Peak value is set to be 255. The kernel size is 15×15 .



Figure 14: Visual comparison of deconvolution with Poisson noise for robustness test. The Peak value is set to be 512. The kernel size is 15×15 .



Figure 15: The visual comparison of recovery results with inaccurate kernel from Lai *et al.*'s real image named "cross_stitch". The kernel is estimated by Xu and Jia [11].

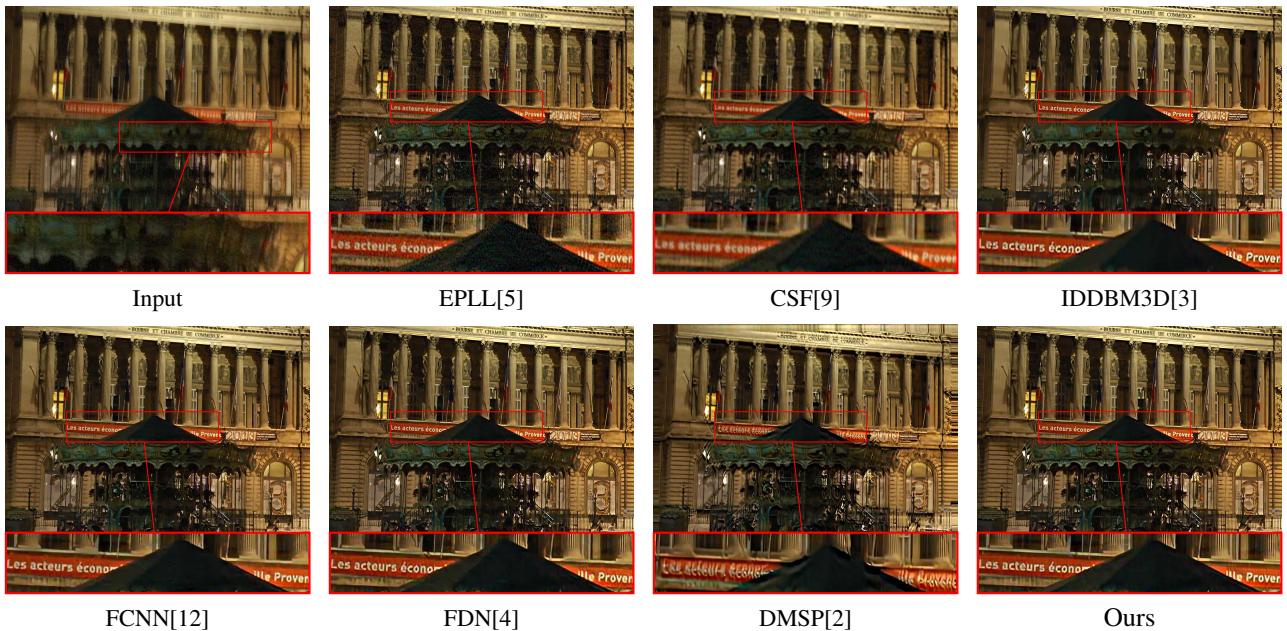


Figure 16: The visual comparison of recovery results with inaccurate kernel from Lai *et al.*'s real image named "building1". The kernel is estimated by Pan *et al.* [7].

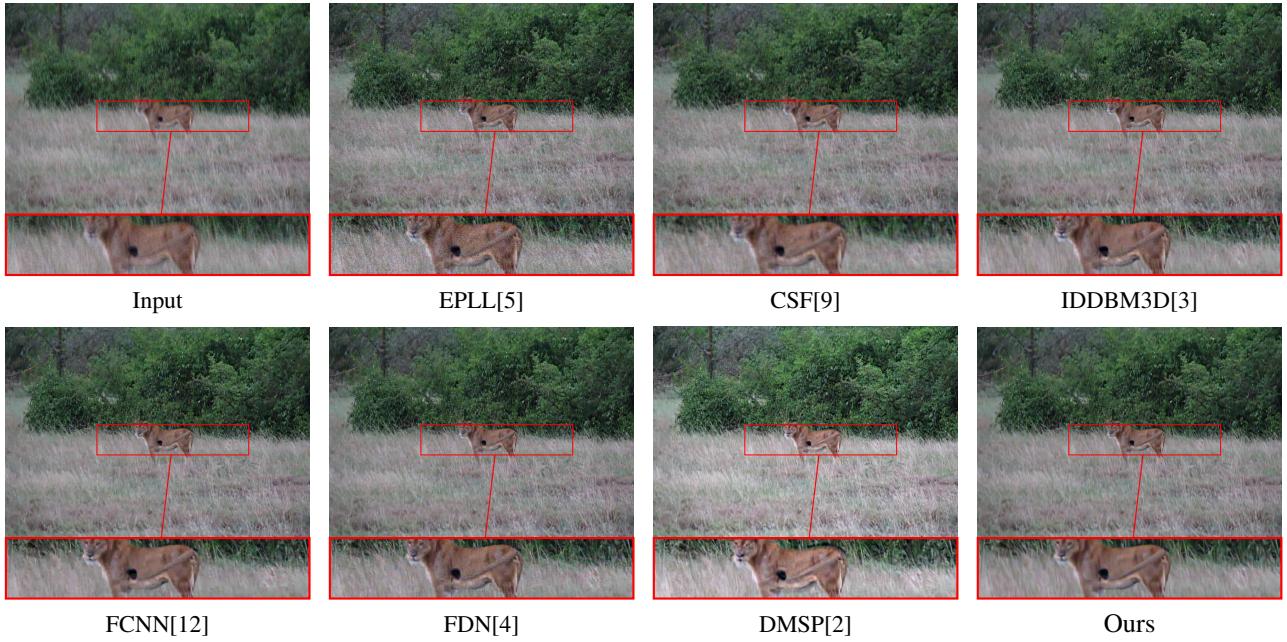


Figure 17: The visual comparsion of recovery results with inaccurate kernel from Lai *et al.*'s real image named "lion". The kernel is estimated by Levin *et al.* [5]

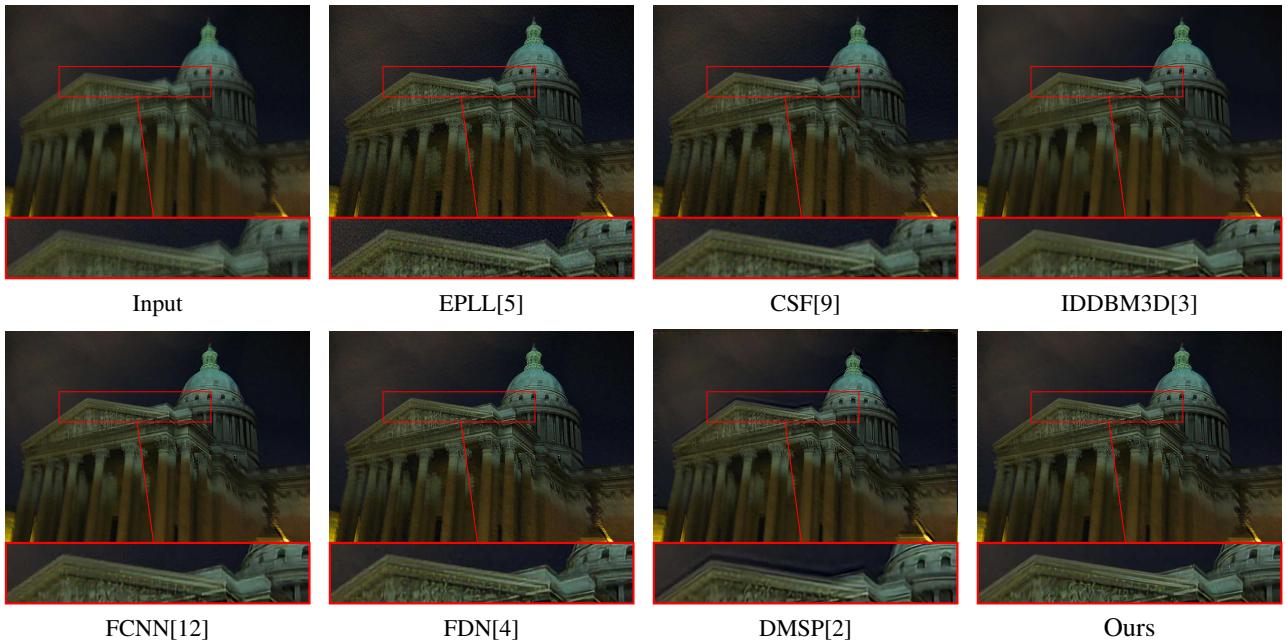


Figure 18: Visual comparsion of recovery results with inaccurate kernel from Lai *et al.*'s real image named "Pantheon". The kernel is estimated by Zhong *et al.* [14]

References

- [1] L. Azzari and A. Foi. Variance stabilization in poisson image deblurring. In *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pages 728–731. IEEE, 2017.
- [2] S. A. Bigdely, M. Zwicker, P. Favaro, and M. Jin. Deep mean-shift priors for image restoration. In *Proc. NIPS*, pages 763–772, 2017.
- [3] A. Danielyan, V. Katkovnik, and K. Egiazarian. BM3D frames and variational image deblurring. *IEEE Trans. Image Process.*, 21(4):1715–1728, 2012.
- [4] J. Kruse, C. Rother, and U. Schmidt. Learning to push the limits of efficient fft-based image deconvolution. In *Proc. CVPR*, pages 4586–4594, 2017.
- [5] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *Proc. CVPR*, pages 2657–2664. IEEE, 2011.
- [6] J. Li, Z. Shen, R. Yin, and X. Zhang. A reweighted l2 method for image restoration with poisson and mixed poisson-gaussian noise. *Inverse Probl. Imaging (Springfield)*, 9(3):875–894, 2015.
- [7] J. Pan, Z. Hu, Z. Su, and M.-H. Yang. Deblurring text images via l0-regularized intensity and gradient prior. In *Proc. CVPR*, pages 2901–2908, 2014.
- [8] U. Schmidt, J. Jancsary, S. Nowozin, S. Roth, and C. Rother. Cascades of regression tree fields for image restoration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(4):677–689, 2016.
- [9] U. Schmidt and S. Roth. Shrinkage fields for effective image restoration. In *Proc. CVPR*, pages 2774–2781, 2014.
- [10] L. Sun, S. Cho, J. Wang, and J. Hays. Edge-based blur kernel estimation using patch priors. In *Proc. ICCP*, pages 1–8. IEEE, 2013.
- [11] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *Proc. ECCV*, pages 157–170. Springer, 2010.
- [12] J. Zhang, J. Pan, W.-S. Lai, R. W. Lau, and M.-H. Yang. Learning fully convolutional networks for iterative non-blind deconvolution. In *Proc. CVPR*, pages 3817–3825, 2017.
- [13] K. Zhang, W. Zuo, S. Gu, and L. Zhang. Learning deep cnn denoiser prior for image restoration. In *Proc. CVPR*, volume 2. IEEE, 2017.
- [14] L. Zhong, S. Cho, D. Metaxas, S. Paris, and J. Wang. Handling noise in single image deblurring using directional filters. In *Proc. CVPR*, pages 612–619, 2013.
- [15] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *Proc. ICCV*, pages 479–486. IEEE, 2011.