Neural Blind Deconvolution Using Deep Priors
Supplementary Material

Dongwei Ren, Kai Zhang, Qilong Wang, Qinghua Hu, Wangmeng Zuo

This supplementary file includes details of network architecture, ablation studies, comparison on benchmark datasets and more results on real blurry images.

1. Network Architecture

1.1. Architecture of $\mathcal{G}_k$

Due to the sparsity of blur kernel $\mathbf{k}$, we only use simple fully connected network to implement $\mathcal{G}_k$. The *SoftMax* is used to meet the normalization constraint of blur kernel. The 1D output of $\mathcal{G}_k$ is finally reshaped to 2D blur kernel.

Table s1: The architecture of $\mathcal{G}_k$. Fully connected layer is with the form Linear(input channel, output channel).

Input: kernel size $m_k \times n_k$, $\mathbf{z}_k$ (200) from the uniform distribution with seed 0. **Output:** blur kernel $\mathbf{k}$ with size $m_k \times n_k$.	
Hidden layer	Linear(200, 1000); *ReLU*
Output layer	Linear(1000, $m_k \times n_k$); *SoftMax*
	Reshape 1D output to obtain 2D blur kenrel with size $m_k \times n_k$

1.2. Architecture of $\mathcal{G}_x$

As for $\mathcal{G}_x$, we employ the encoder-decoder network with skip connections [9]. As shown in Fig. s1, the i-th unit encoder-decoder architecture is first demonstrated. Taking e_i as an example, we use the form $e_i(n_f, k, p)$ to represent that the convolutions in e_i have n_f filters with size $k \times k$ and $p \times p$ padding. We note that the filter size in the last convolution of d_i is fixed as 1×1. The slope of *LeakyReLU* is 0.2, downsampling is implemented using $stride = 2$, and upsampling is implemented using 2x *bilinear* interpolation. The generative network $\mathcal{G}_x$ have five units in Fig. s1, and its parameter settings are detailed presented in Table s2. In our current implementation, $\mathcal{G}_x$ is used to only process the y channel given a color image in ycbcr space.

Table s2: The architecture of $\mathcal{G}_x$, which consists of five e_i, d_i and s_i in Fig. s1. Convolution is with the form Conv.(input channel, output channel, kernel size, padding size)

Input: $\mathbf{z}_x$ ($8 \times m_x \times n_x$) from the uniform distribution with seed 0. **Output:** latent image $\mathbf{x}$ ($1 \times m_x \times n_x$).	
Encoder unit 1	$e_1(128, 3, 1)$, $s_1(16, 3, 1)$
Encoder unit 2	$e_2(128, 3, 1)$, $s_2(16, 3, 1)$
Encoder unit 3	$e_3(128, 3, 1)$, $s_3(16, 3, 1)$
Encoder unit 4	$e_4(128, 3, 1)$, $s_4(16, 3, 1)$
Encoder unit 5	$e_5(128, 3, 1)$, $s_5(16, 3, 1)$
Decoder unit 5	$d_5(128, 3, 1)$
Decoder unit 4	$d_4(128, 3, 1)$
Decoder unit 3	$d_3(128, 3, 1)$
Decoder unit 2	$d_2(128, 3, 1)$
Decoder unit 1	$d_1(128, 3, 1)$
Output layer	Conv.(128,1,1,0); *Sigmoid*

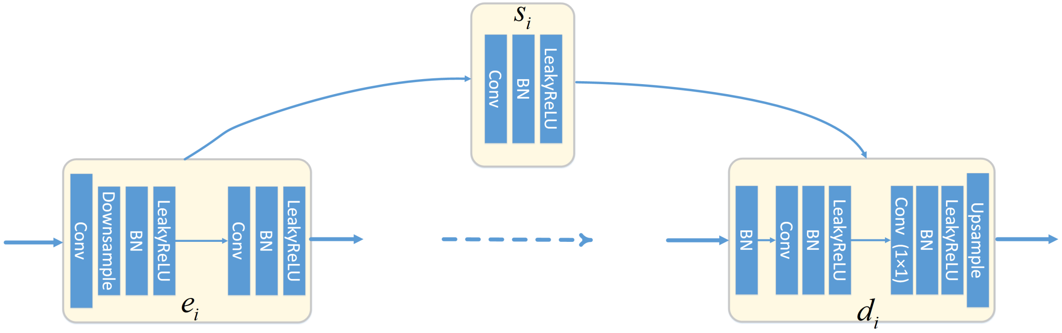
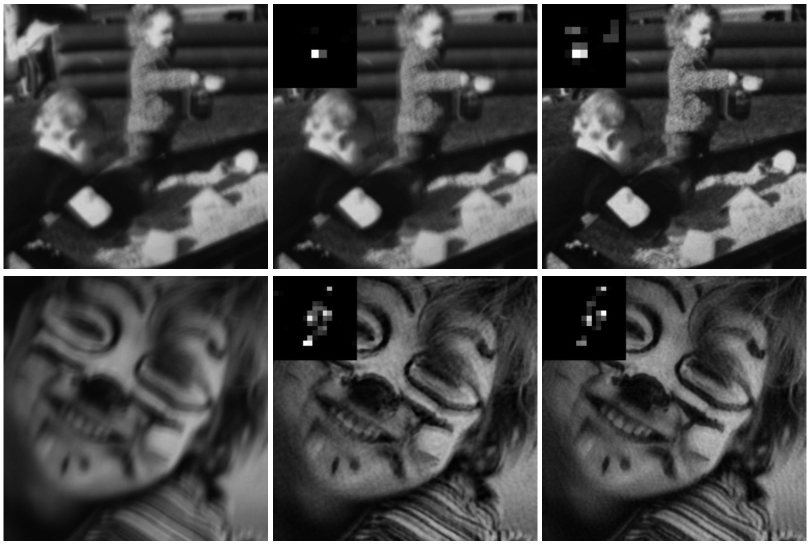


Figure s1: One unit in encoder-decoder with skip connection, where e_i is the i-th unit in encoder, d_i is the i-th unit in decoder and s_i is the skip connection between e_i and d_i. Taking e_i as an example, we use the form $e_i(n_f, k, p)$ to represent that the convolutions in e_i have n_f filters with size $k \times k$ and $p \times p$ padding. We note that the filter size in the last convolution of d_i is fixed as 1×1. The slope of *LeakyReLU* is 0.2, downsampling is implemented using $stride = 2$, and upsampling operation is implemented as 2x *bilinear* interpolation.

2. Ablation Study

2.1. Alternating Optimization *vs*. Joint Optimization

Fig. s2 provides several failure cases of SelfDeblur-A. It can be seen that SelfDeblur-A may converge to delta kernel and worse solution, while SelfDeblur-J performs favorably on these cases.



Blurry images SelfDeblur-A SelfDeblur-J

Figure s2: Failure cases of SelfDeblur-A, while SelfDeblur-J performs well.

2.2. Discussion of selecting λ

To determine the choice of λ, different Gaussian noise levels $\sigma = 1\times10^{-3}$ and 1×10^{-2} were added to Levin dataset, where original noise levels of blurry images approach 0. The results are reported in Talbe s3. (1) We tested SelfDeblur with various λ values, and found that $\lambda = 0.1\times\sigma$ can generally leads to state-of-the-art performance. (2) On original Levin dataset, SelfDeblur without TV regularizer (*i.e.*, $\lambda = 0$) achieved better quantitative metrics than SelfDeblur ($\lambda = 1\times10^{-6}$) in Table 3. Combining (1)&(2), $\mathcal{G}_k$ and $\mathcal{G}_x$ are sufficient regularizations for blind deconvolution, and TV regularizer improves robustness for various noise levels.

Table s3: Comparison of SelfDeblur with various λ values on Levin dataset with different noise levels. The noise levels in original dataset approach 0. Different Gaussian noise levels $\sigma = 1\times10^{-3}$ and 1×10^{-2} were added to Levin dataset, respectively.

Noise σ	Original					1×10^{-3}					1×10^{-2}				
$\lambda = 1\times$	0	10^{-6}	10^{-5}	10^{-4}	10^{-3}	0	10^{-5}	10^{-4}	10^{-3}	10^{-2}	0	10^{-4}	10^{-3}	10^{-2}	10^{-1}
PSNR	33.54	33.07	33.23	31.87	27.14	24.28	29.83	30.59	29.44	29.85	23.09	23.99	29.62	29.64	28.75
SSIM	0.9349	0.9313	0.9307	0.9185	0.8322	0.7430	0.8551	0.8788	0.8406	0.8533	0.6899	0.7261	0.8504	0.8338	0.8000

3. Comparison on dataset of Levin *et al*. [5]

Besides the comparison of deblurring images, we further evaluate the estimated kernels using two metrics, *i.e.*, MSE best aligned to ground-truth blur kernel $\mathbf{k}^{gt}$ and maximum of normalized convolution (MNC) [2],

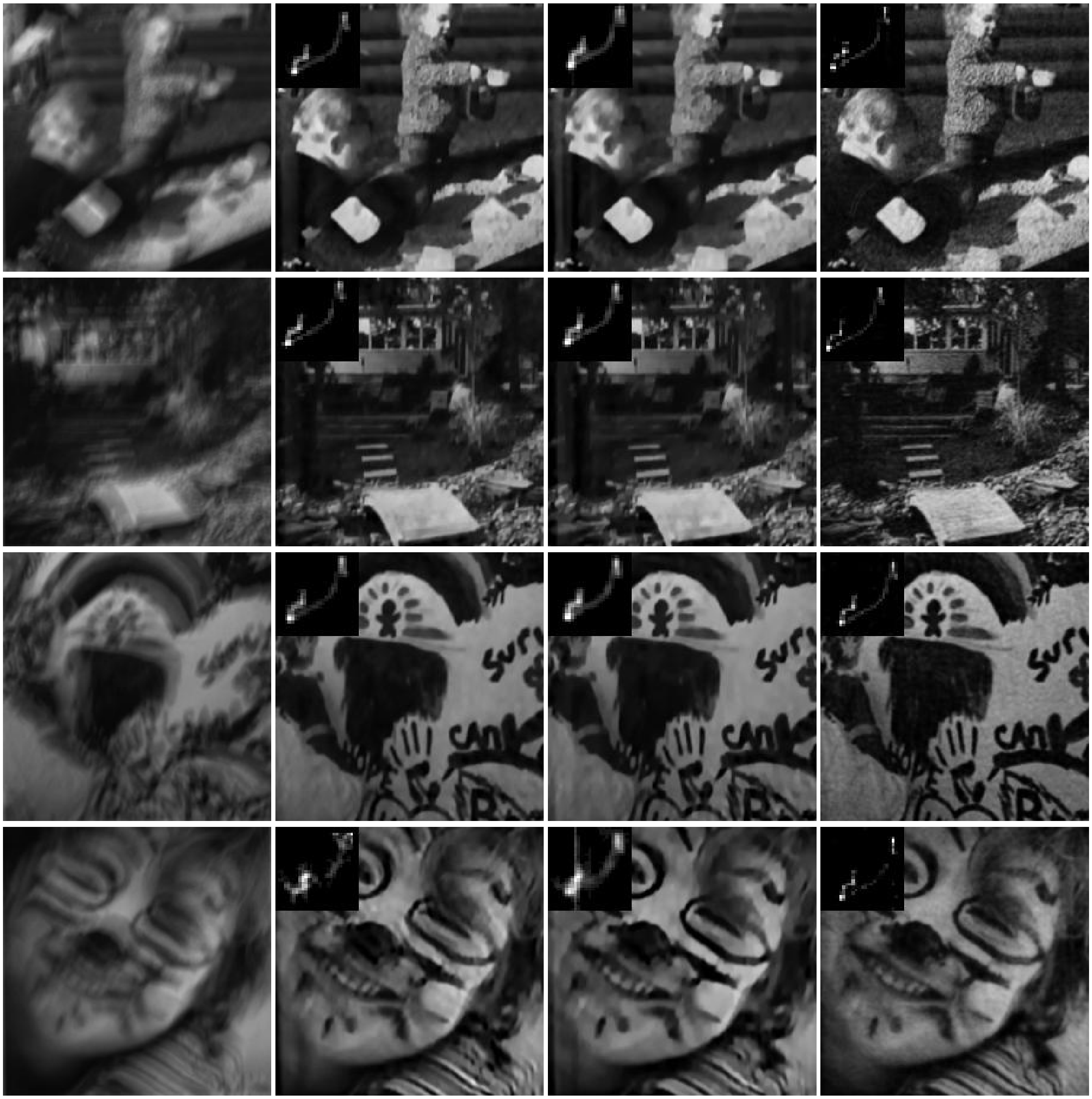
$$\text{MNC} = \max\left(\frac{\mathbf{k}\otimes\mathbf{k}^{gt}}{\|\mathbf{k}\|_2\|\mathbf{k}^{gt}\|_2}\right).$$

From Table s4, our SelfDeblur is much superior to the competing methods in terms of both the metrics.

Table s4: Comparison of the blur kernel estimation performance on the dataset of Levin *et al.*

	Krishnan *et al.* [3]	Levin *et al.* [6]	Cho&Lee [1]	Xu&Jia [10]	Sun *et al.* [8]	Zuo *et al.* [11]	Pan-DCP [7]	SelfDeblur
MSE	675.4	307.0	340.8	402.1	254.7	351.9	347.2	149.5
MNC [2]	0.8261	0.8693	0.8635	0.8811	0.9305	0.9042	0.8936	0.9408

In Fig. s3, we compare the deblurring results on #4 kernel, which is the most difficult to handle. The deblurring results by our SelfDeblur are with finer textures, while the results by the other methods are often over-smoothing.

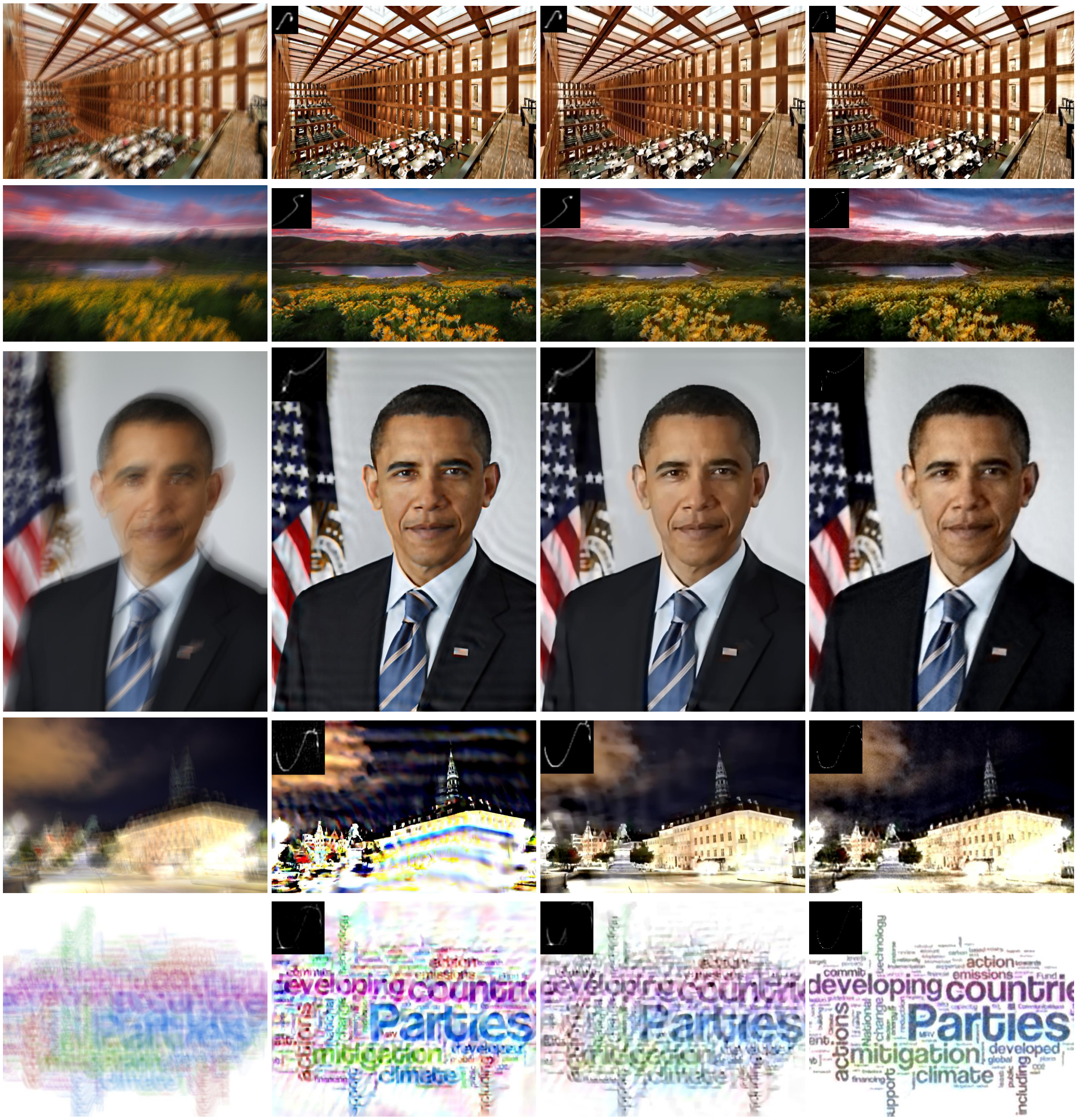


Blurry images Sun *et al*. [8] Pan-DCP [7] SelfDeblur

Figure s3: Comparison on Levin *et al*.'s dataset.

4. Comparison on dataset of Lai *et al*. [4]

In Fig. s4, we demonstrate comparison results from 5 categories, *i.e*., *Manmade*, *Natural*, *People*, *Saturated* and *Text*. The images from this dataset are usually with high resolution, so please zoom in the deblurring results to compare texture details.

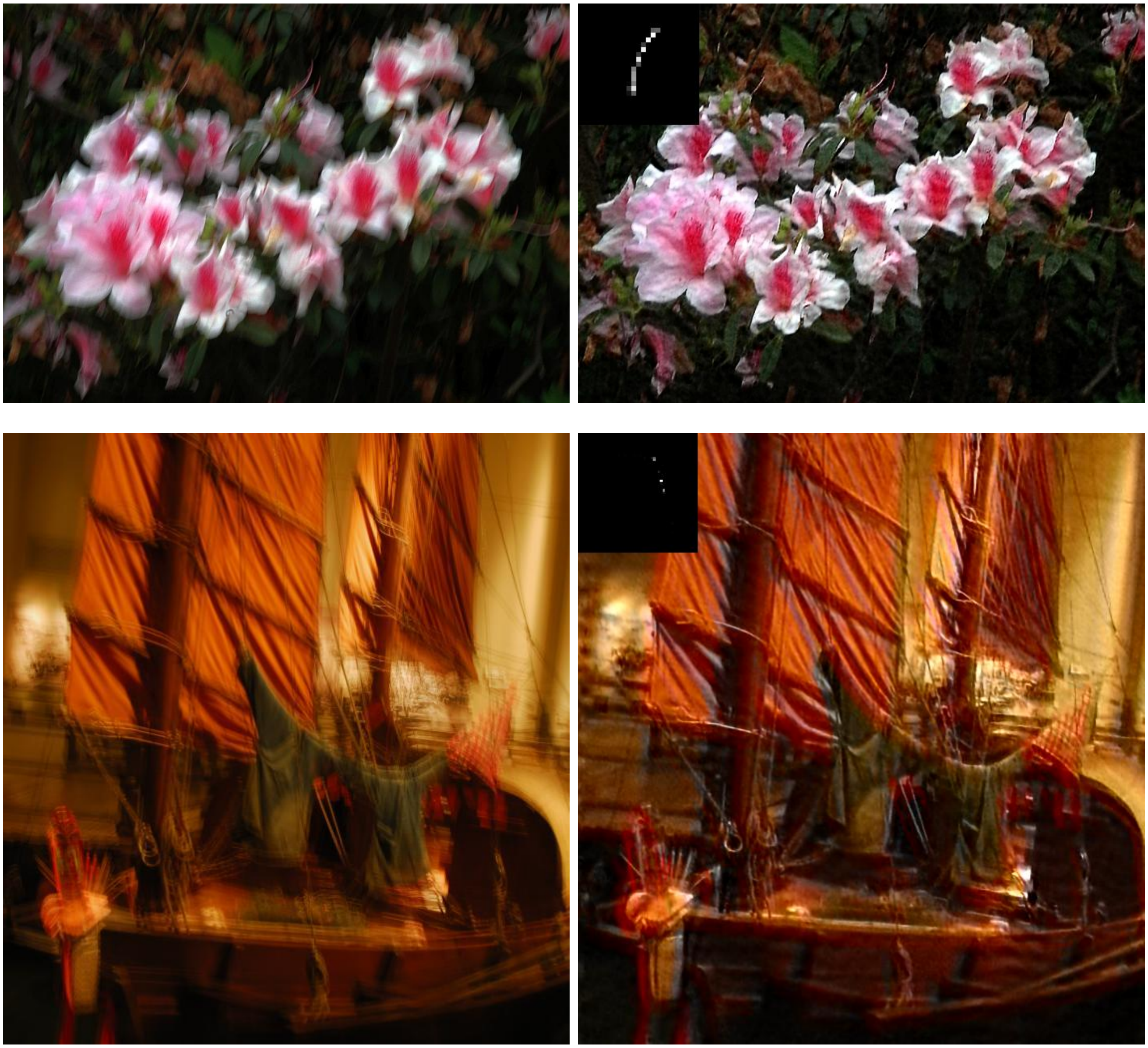


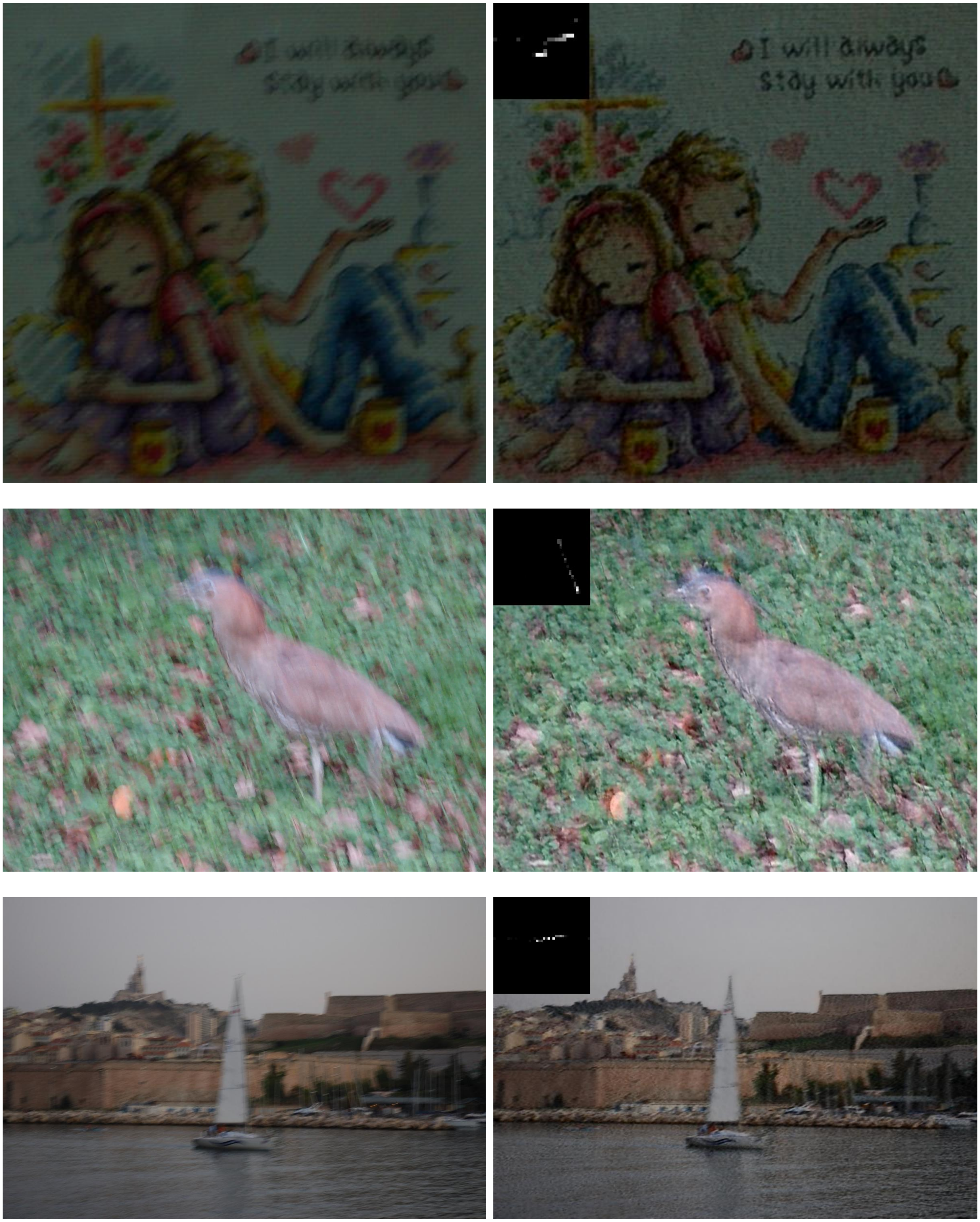
Blurry images Xu&Jia [10] Pan-DCP [7] SelfDeblur

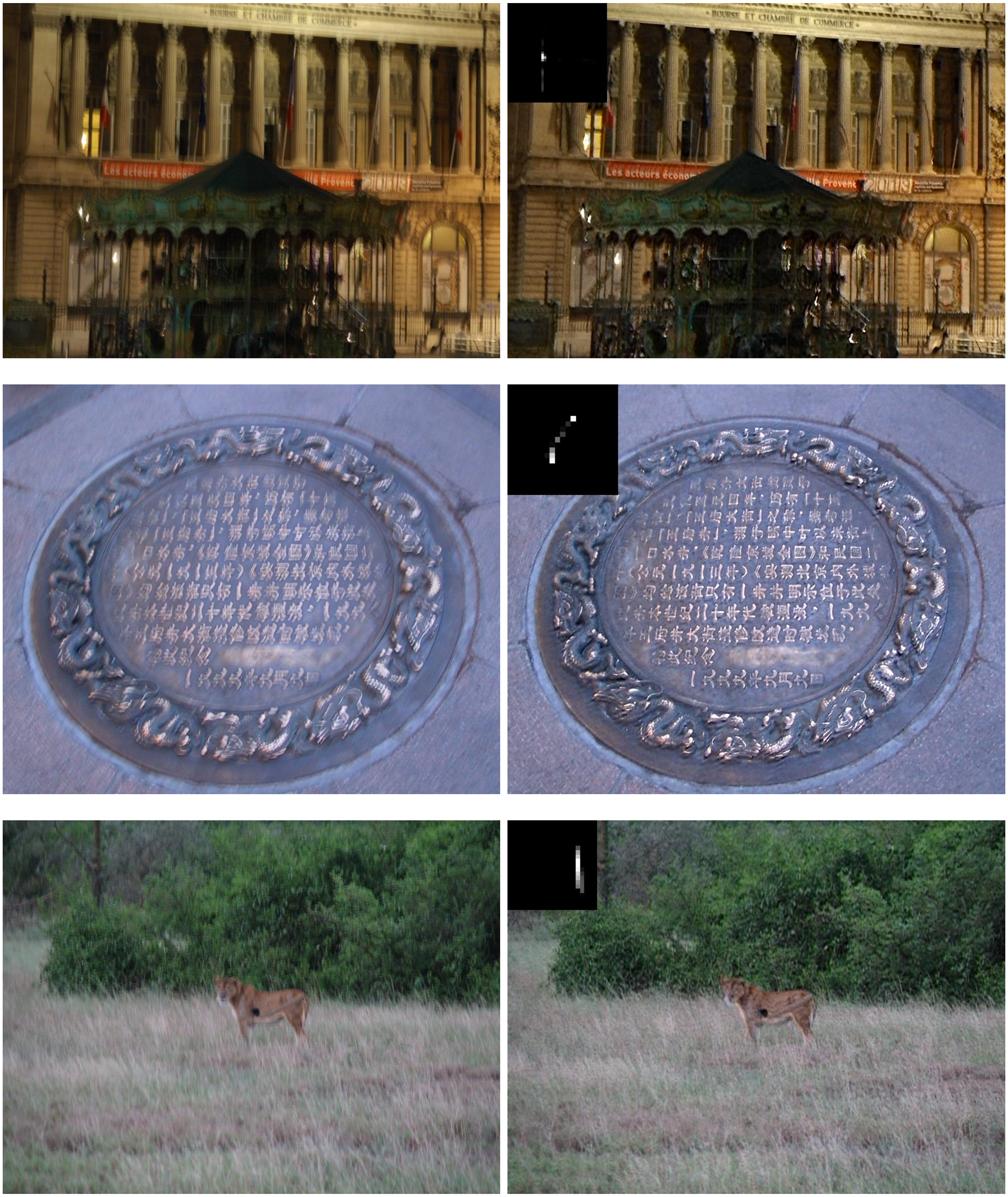
Figure s4: Comparison on Lai *et al*.'s dataset. Please zoom in to see texture details.

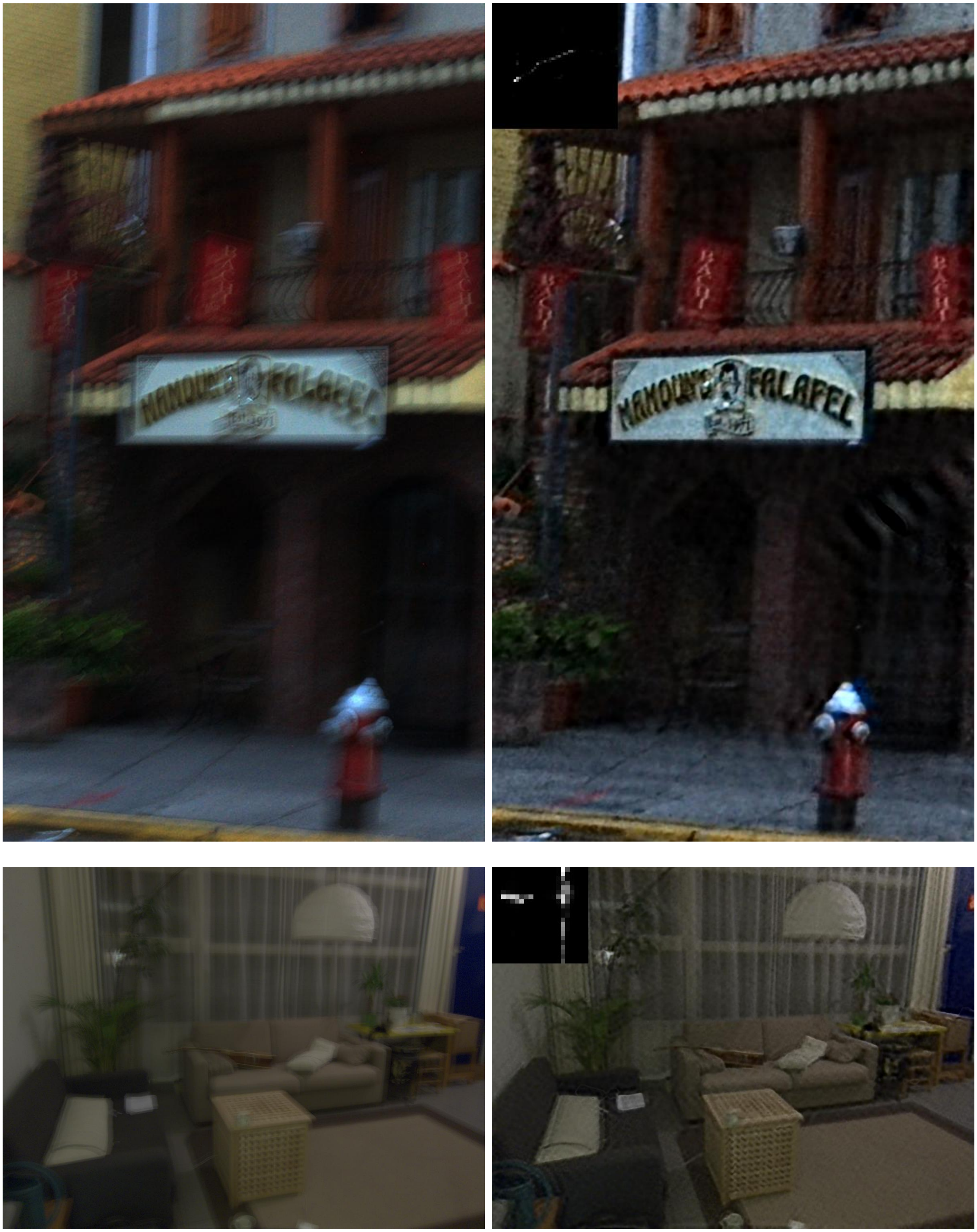
5. More Results on Real Blurry Images

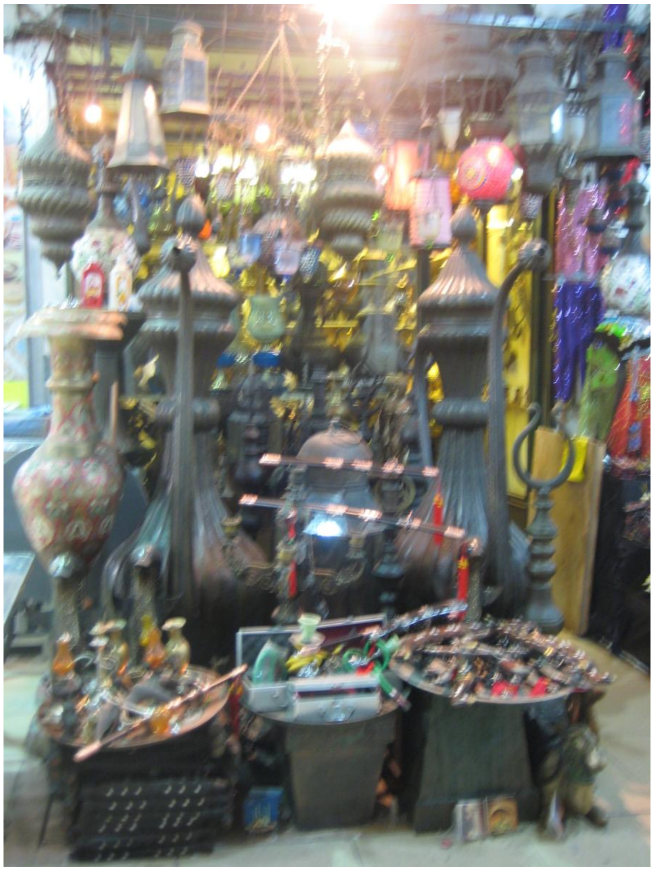
Here, we present more results on real blurry images. The left is blurry images, and the right presents the estimated blur kernels and deblurring results by our SelfDeblur.

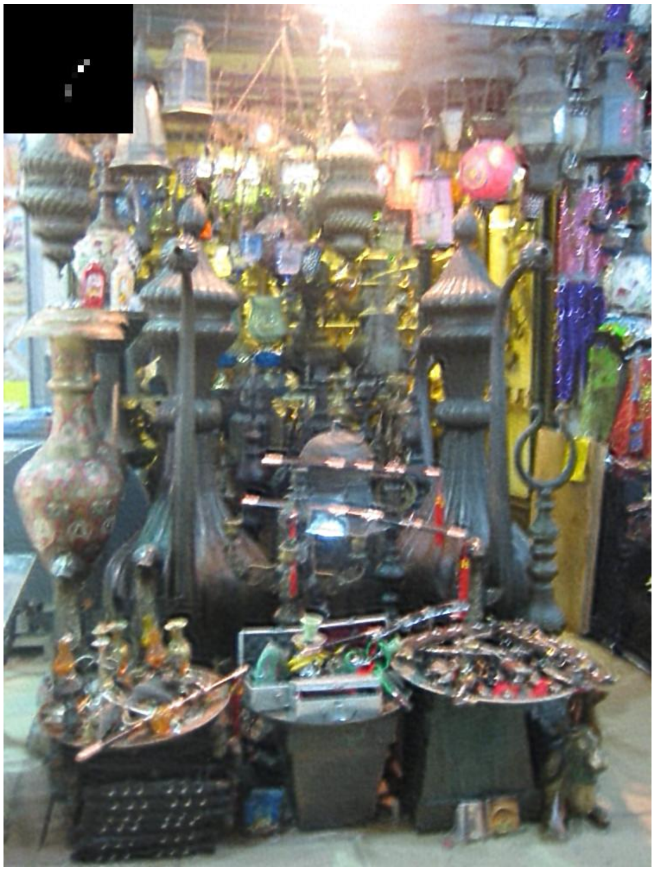


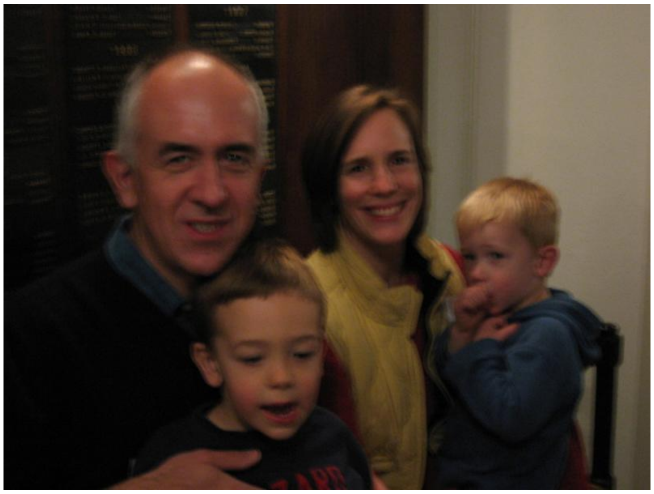

I will always
stay with you
I will always
stay with you

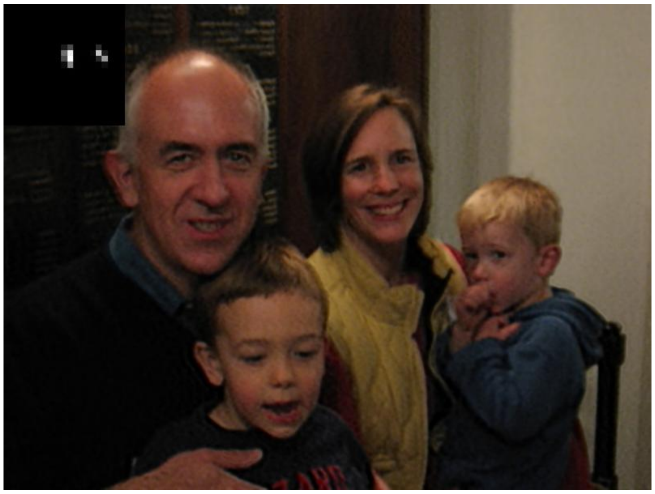

BOURSE ET CHAMBRE DE COMMERCE
Les acteurs économ
BOURSE ET CHAMBRE DE COMMERCE
Les acteurs économ

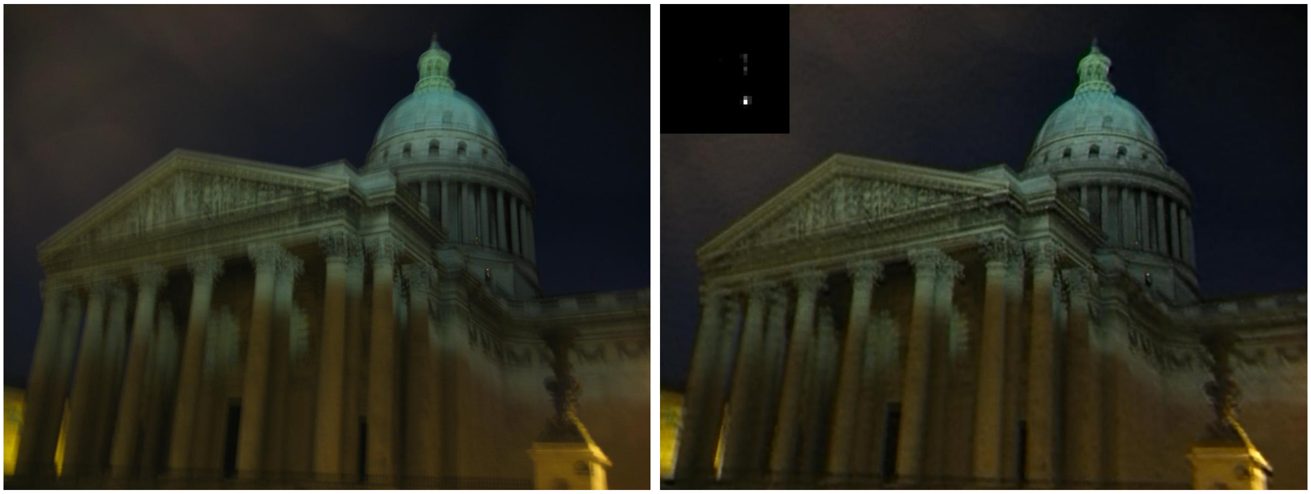

MAMOUN'S FALAFEL
Est. 1971











References

[1] S. Cho and S. Lee. Fast motion deblurring. *ACM Transactions on Graphics*, 28(5):145, 2009. 3

[2] Z. Hu and M.-H. Yang. Good regions to deblur. In *European conference on computer vision*, pages 59–72, 2012. 3

[3] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 233–240, 2011. 3

[4] W.-S. Lai, J.-B. Huang, Z. Hu, N. Ahuja, and M.-H. Yang. A comparative study for single image blind deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1709, 2016. 5

[5] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1964–1971, 2009. 3

[6] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2657–2664, 2011. 3

[7] J. Pan, D. Sun, H. Pfister, and M.-H. Yang. Deblurring images via dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(10):2315–2328, 2018. 3, 4, 5

[8] L. Sun, S. Cho, J. Wang, and J. Hays. Edge-based blur kernel estimation using patch priors. In *IEEE International Conference on Computational Photography*, pages 1–8, 2013. 3, 4

[9] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018. 1

[10] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *European Conference on Computer Vision*, pages 157–170, 2010. 3, 5

[11] W. Zuo, D. Ren, D. Zhang, S. Gu, and L. Zhang. Learning iteration-wise generalized shrinkage–thresholding operators for blind deconvolution. *IEEE Transactions on Image Processing*, 25(4):1751–1764, 2016. 3