Supplementary Materials

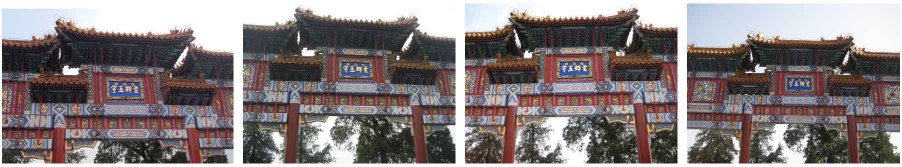
Good Image Priors for Non-blind Deconvolution: Generic *vs* Specific

Libin Sun[1], Sunghyun Cho[2⋆], Jue Wang[2], and James Hays[1]

[1]Brown University, Providence RI 02912, USA
[2]Adobe Research, Seattle WA 98103, USA
{lbsun,hays}@cs.brown.edu, sodomau@postech.ac.kr,
juewang@adobe.com

1 Synthetic Test Set

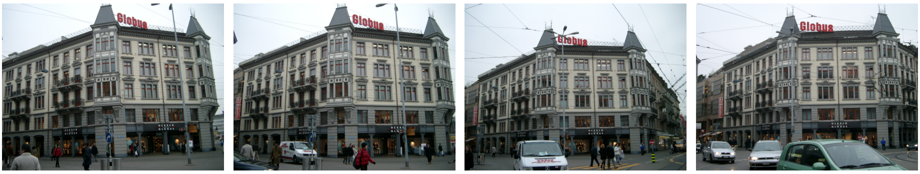
In Fig. 1 through Fig. 5 we show the 5 groundtruth images along with their example images. These 5 images are then blurred with kernels $2, 4, 6, 8$ from [1] to generate the synthetic testset of our experiments. 1% i.i.d Gaussian noise is added to the luminance channel. These images come from the dataset provided by [2]. As can be seen, the sharp example images are instance-level scene matches, with slight variations in perspectives and lighting. Foreground objects might be different as well. We use this test set to compare to state-of-the-art universal priors for deblurring [3–6], as well as the recently presented by-example work of HaCohen *et al.* [7].



(a) groundtruth (b) example 1 (c) example 2 (d) example 3

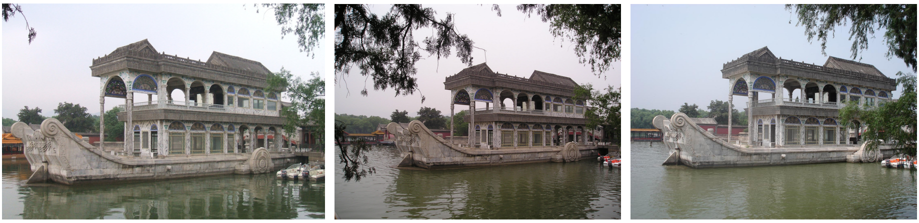
Fig. 1: Test image 1. Leftmost: groundtruth. Right: sharp examples.

⋆ Sunghyun Cho is now with Samsung Electronics.



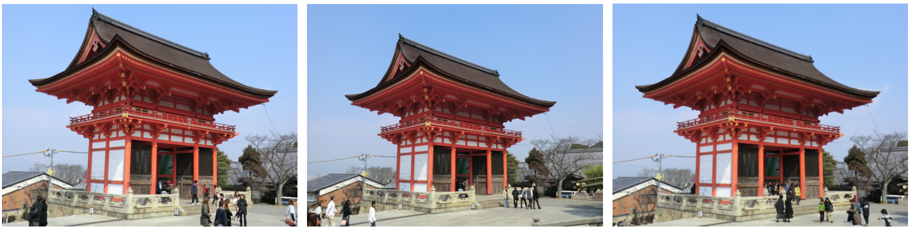
(a) groundtruth (b) example 1 (c) example 2 (d) example 3

Fig. 2: Test image 2. Leftmost: groundtruth. Right: sharp examples.



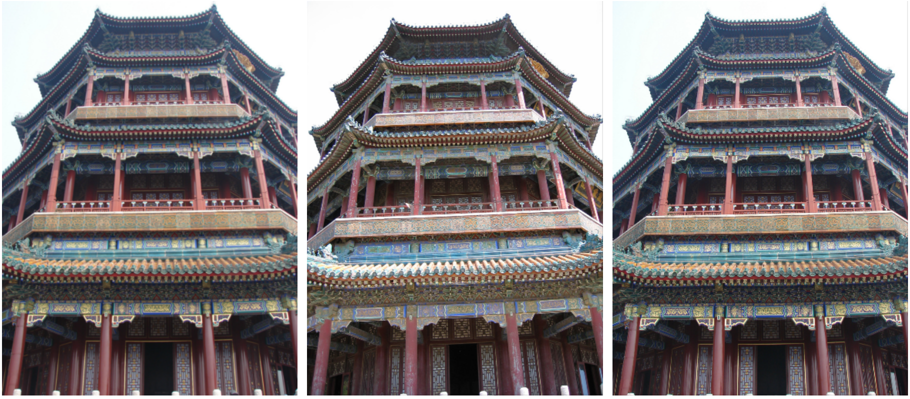
(a) groundtruth (b) example 1 (c) example 2

Fig. 3: Test image 3. Leftmost: groundtruth. Right: sharp examples.



(a) groundtruth (b) example 1 (c) example 2

Fig. 4: Test image 4. Leftmost: groundtruth. Right: sharp examples.



(a) groundtruth (b) example 1 (c) example 2

Fig. 5: Test image 5. Leftmost: groundtruth. Right: sharp examples.

2 Comparing Against Generic Non-blind Deconvolution Methods

For each of the $5 \times 4 = 20$ test images, assuming the groundtruth PSF's are known, we ran the online code packages with default parameters provided by the authors of [3–5] to generate deblurred results. The results from [6] were kindly provided by the authors. We used their best results (RTF-6) for comparison. Note that these represent the leading methods in non-blind deconvolution trained on generic image information, hence not context-specific (like ours or the method of [7]).

2.1 Quantitative Comparisons

First, we show quantitative improvement relative to each competing leading method, in terms of PSNR and SSIM. A positive number indicates how much improvement our method introduces over the competing method. We present these evaluations in Table 1 through Table 4. Please note that, according to these two scores, our method strictly outperforms all competing methods for all test images. For all evaluations in this section, we consider the best possible integer alignment between the output image and the groundtruth. Since the groundtruth PSF is used, subpixel shifts are not possible. However, this is not the case for Sec. 3.

PSNR gain (mean= $+1.8898$)

	kernel 2	kernel 4	kernel 6	kernel 8
image 1	1.5312	1.5171	1.2207	1.3474
image 2	1.8298	2.0841	1.3625	1.8841
image 3	0.9655	0.9668	0.8094	0.9878
image 4	3.3443	3.2498	3.0828	3.1117
image 5	2.3962	2.0999	1.6952	2.3106

SSIM gain (mean= $+0.0361$)

	kernel 2	kernel 4	kernel 6	kernel 8
image 1	0.0455	0.0512	0.0248	0.0434
image 2	0.0311	0.0410	0.0156	0.0361
image 3	0.0190	0.0231	0.0101	0.0220
image 4	0.0519	0.0575	0.0367	0.0505
image 5	0.0460	0.0452	0.0223	0.0501

Table 1: Performance gain over Schmidt *et al.* [6].

PSNR gain (mean= $+1.9366$)

	kernel 2	kernel 4	kernel 6	kernel 8
image 1	1.7049	1.6874	1.2668	1.6530
image 2	1.8646	1.9400	1.2994	1.9730
image 3	0.9729	0.9901	0.7099	1.1077
image 4	3.3896	3.2859	3.0363	3.3131
image 5	2.4213	2.0977	1.6362	2.3817

SSIM gain (mean= $+0.0461$)

	kernel 2	kernel 4	kernel 6	kernel 8
image 1	0.0647	0.0711	0.0360	0.0643
image 2	0.0366	0.0434	0.0190	0.0416
image 3	0.0296	0.0335	0.0178	0.0321
image 4	0.0603	0.0654	0.0437	0.0607
image 5	0.0562	0.0553	0.0306	0.0599

Table 2: Performance gain over Zoran and Weiss[5].

PSNR gain (mean= +2.8501)

	kernel 2	kernel 4	kernel 6	kernel 8
image 1	2.5485	2.4983	2.0070	2.3341
image 2	2.8706	3.0392	2.0334	2.7770
image 3	1.9631	1.9835	1.5481	1.9932
image 4	4.6140	4.4987	3.8523	4.3748
image 5	3.3945	3.1603	2.2122	3.2990

SSIM gain (mean= +0.0634)

	kernel 2	kernel 4	kernel 6	kernel 8
image 1	0.0803	0.0867	0.0469	0.0787
image 2	0.0553	0.0670	0.0290	0.0572
image 3	0.0448	0.0519	0.0294	0.0476
image 4	0.0850	0.0935	0.0585	0.0822
image 5	0.0745	0.0812	0.0378	0.0799

Table 3: Performance gain over Levin *et al.* [3].

PSNR gain (mean= +3.3583)

	kernel 2	kernel 4	kernel 6	kernel 8
image 1	2.9097	2.8851	2.2914	2.6368
image 2	3.5174	3.8071	2.5802	3.3974
image 3	2.2855	2.4338	1.8530	2.2939
image 4	5.2160	5.1261	4.5192	4.9035
image 5	4.0035	3.8009	2.9000	3.8061

SSIM gain (mean= +0.0839)

	kernel 2	kernel 4	kernel 6	kernel 8
image 1	0.1062	0.1164	0.0644	0.1047
image 2	0.0757	0.0935	0.0416	0.0782
image 3	0.0617	0.0742	0.0423	0.0666
image 4	0.1020	0.1126	0.0703	0.1005
image 5	0.0985	0.1090	0.0559	0.1040

Table 4: Performance gain over Krishnan *et al.* [4].

2.2 Qualitative Comparisons

We show detailed qualitative comparisons in Fig. 6 through Fig. 25. In each figure, the first column shows the blurred input image, input PSF, the groundtruth, followed by latent images from our method, [6],[5], [3], and [4]. Three detailed crops are shown in each row in column 2 through 4 to provide better comparison of image details. Because modern deconvolution algorithms can reliably recover low frequencies, it is important to examine these results closely at high resolution so that the differences in mid and high frequency content are apparent.

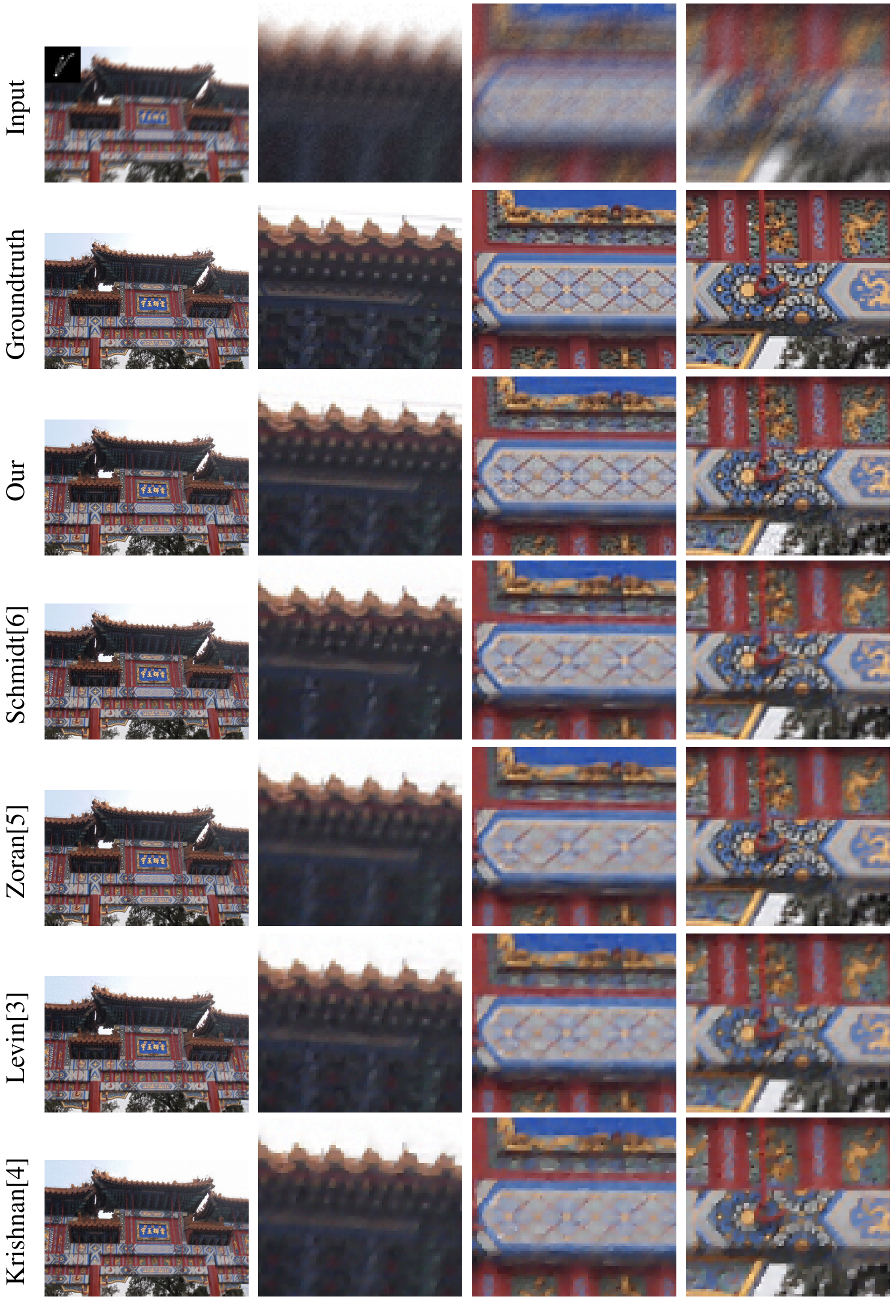


Fig. 6: results for image 1, kernel 2

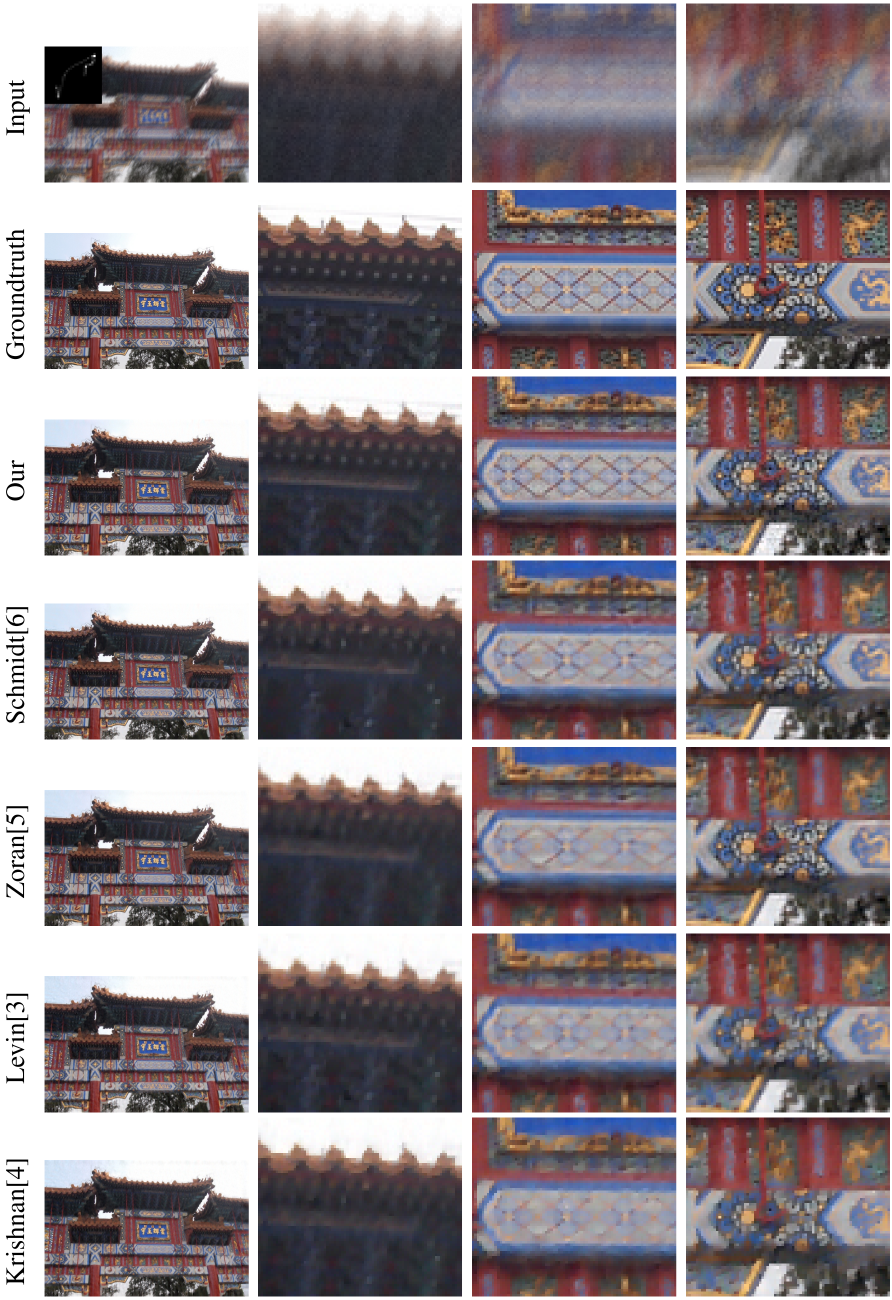


Fig. 7: results for image 1, kernel 4

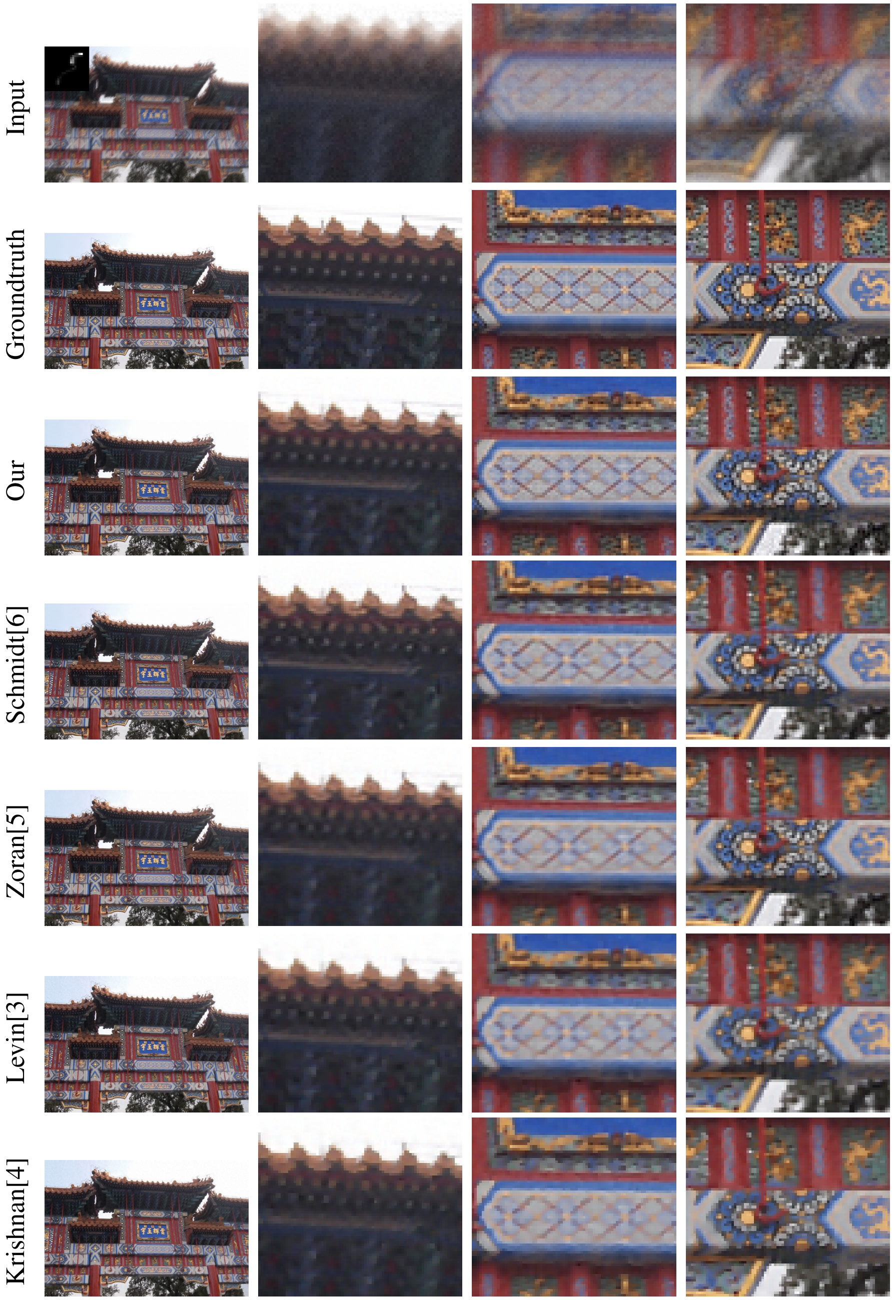


Fig. 8: results for image 1, kernel 6

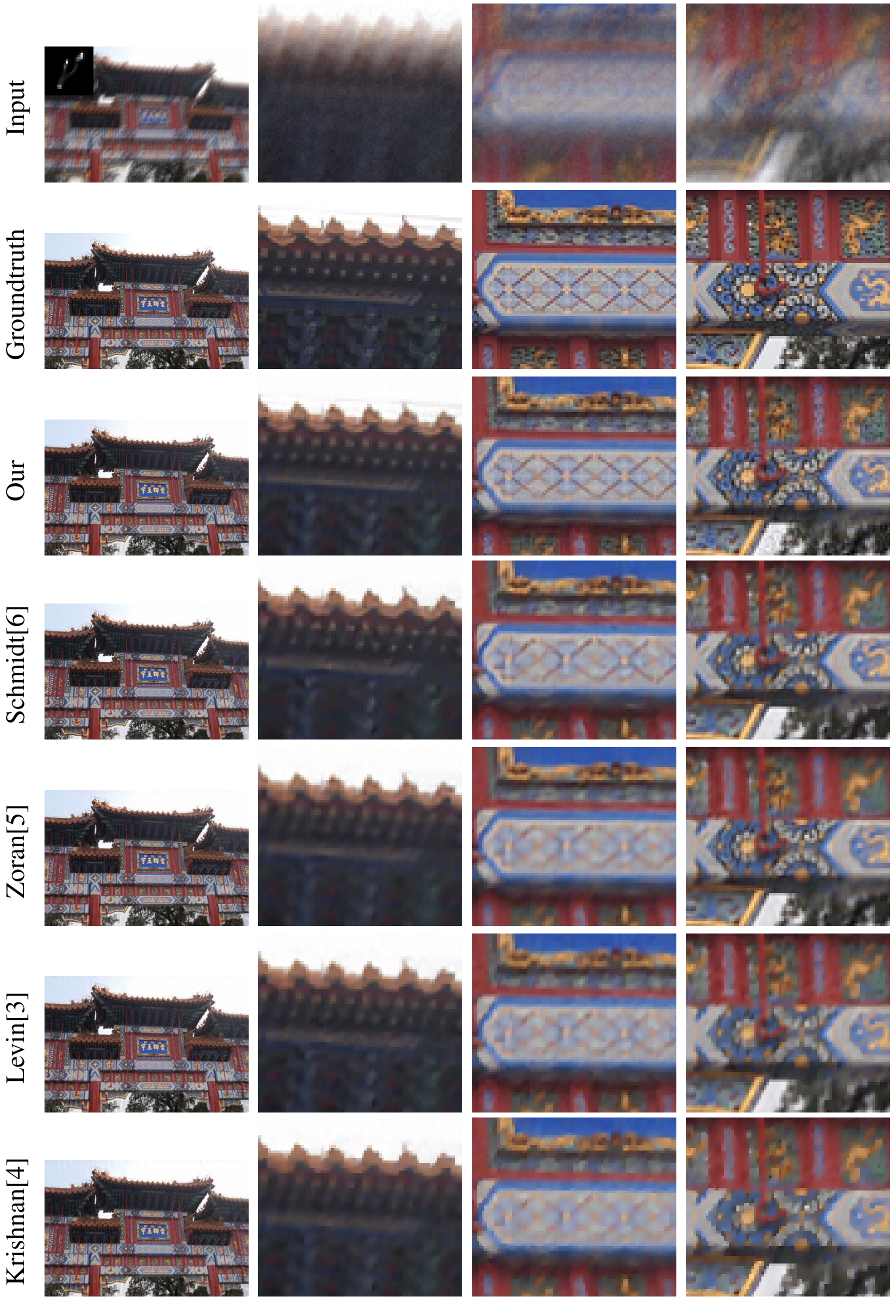


Fig. 9: results for image 1, kernel 8

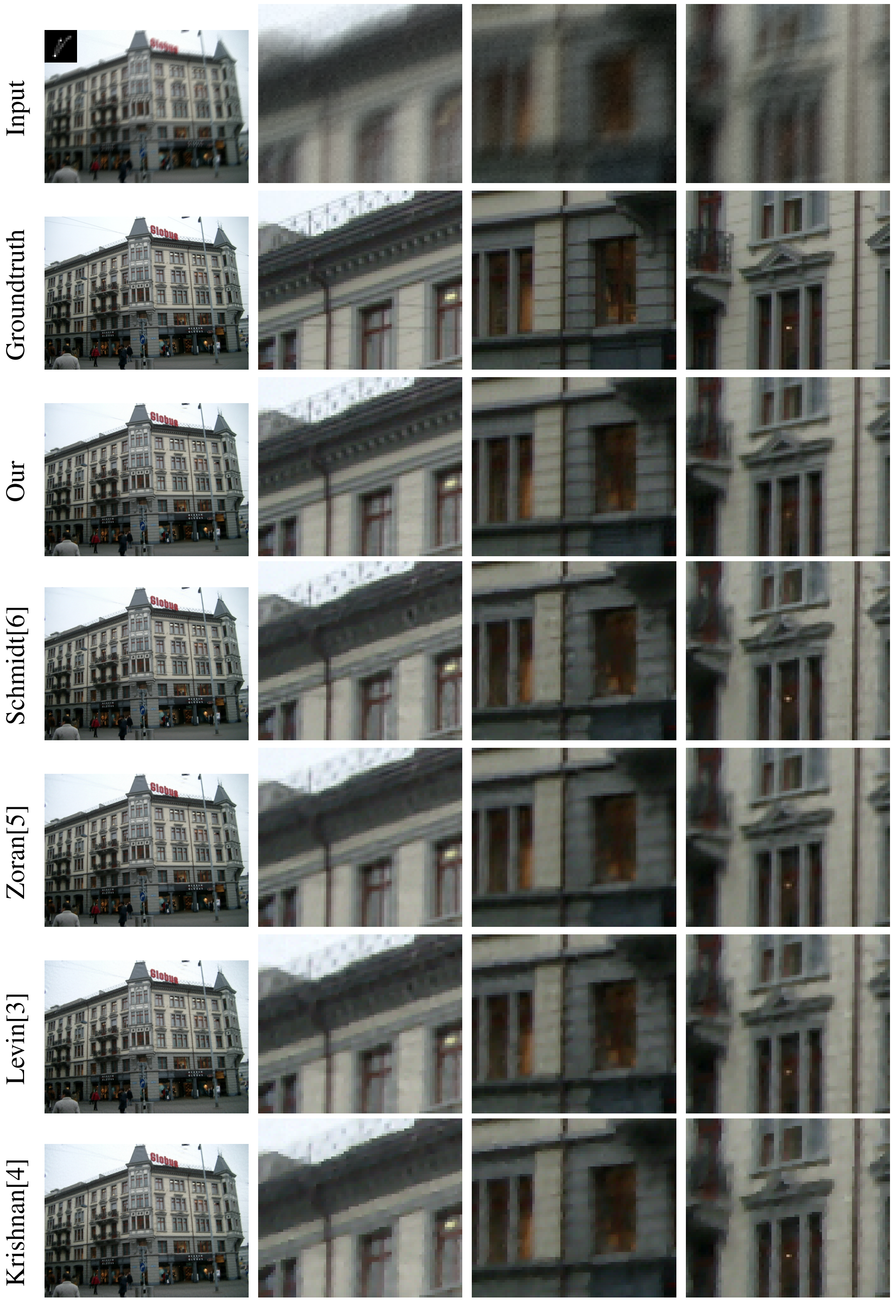


Fig. 10: results for image 2, kernel 2

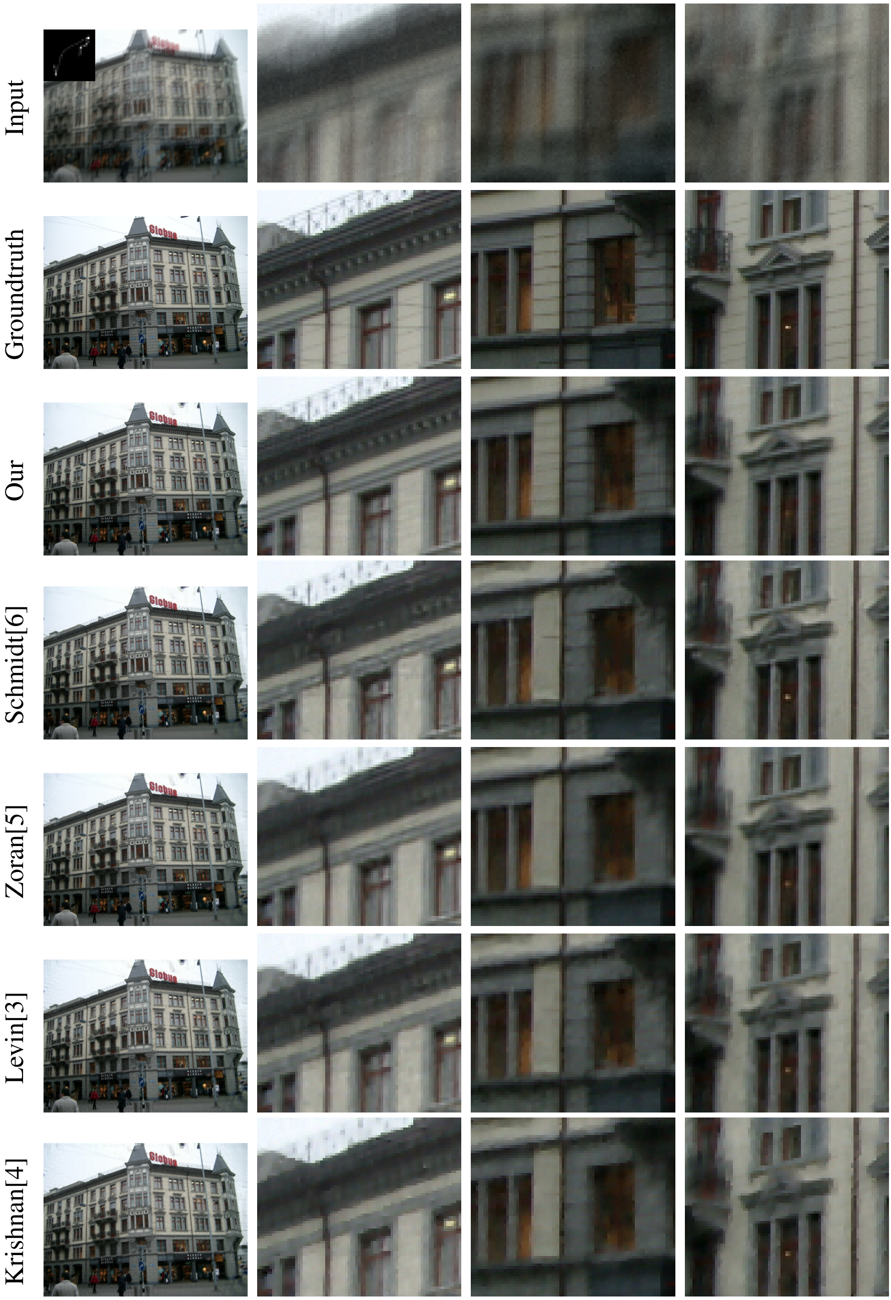


Fig. 11: results for image 2, kernel 4

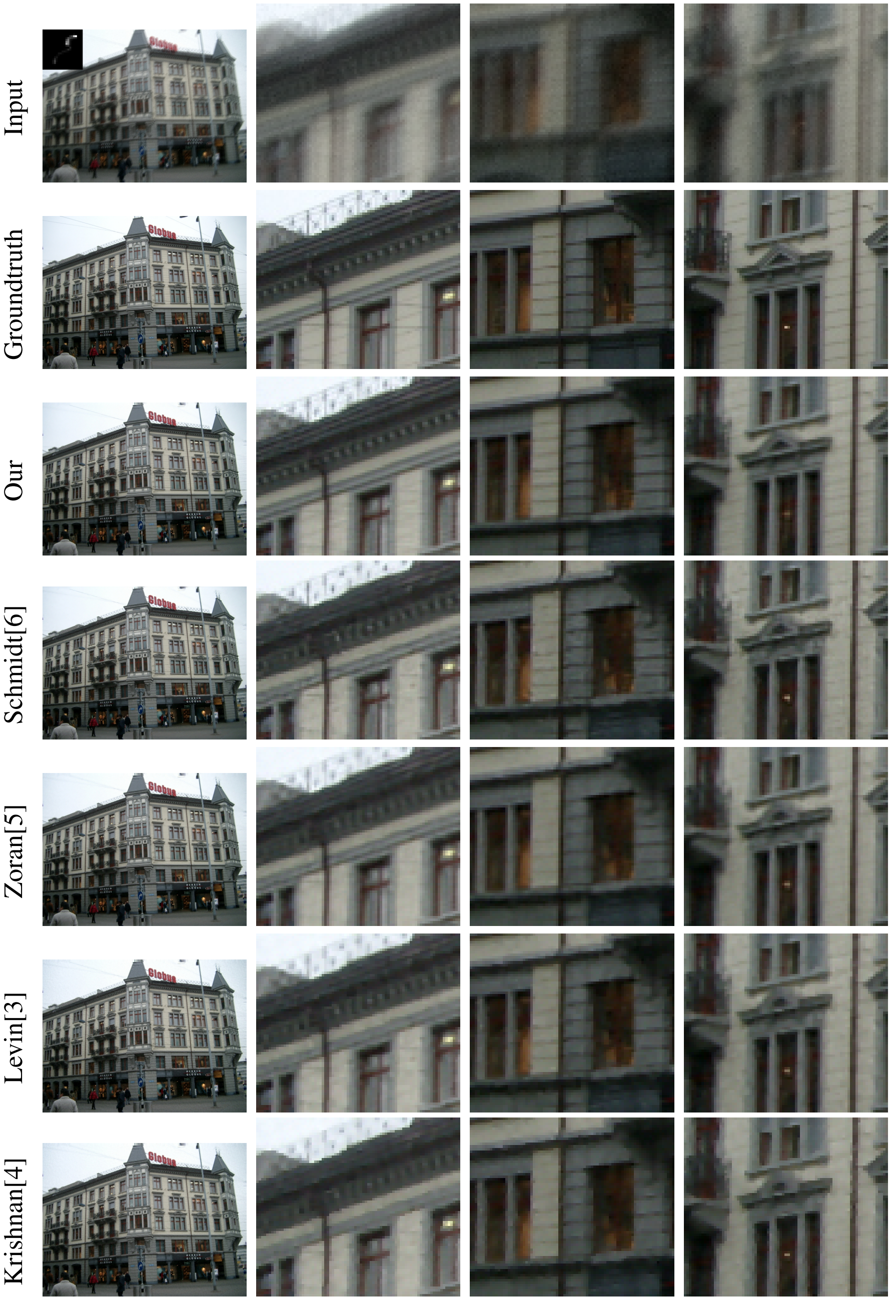


Fig. 12: results for image 2, kernel 6

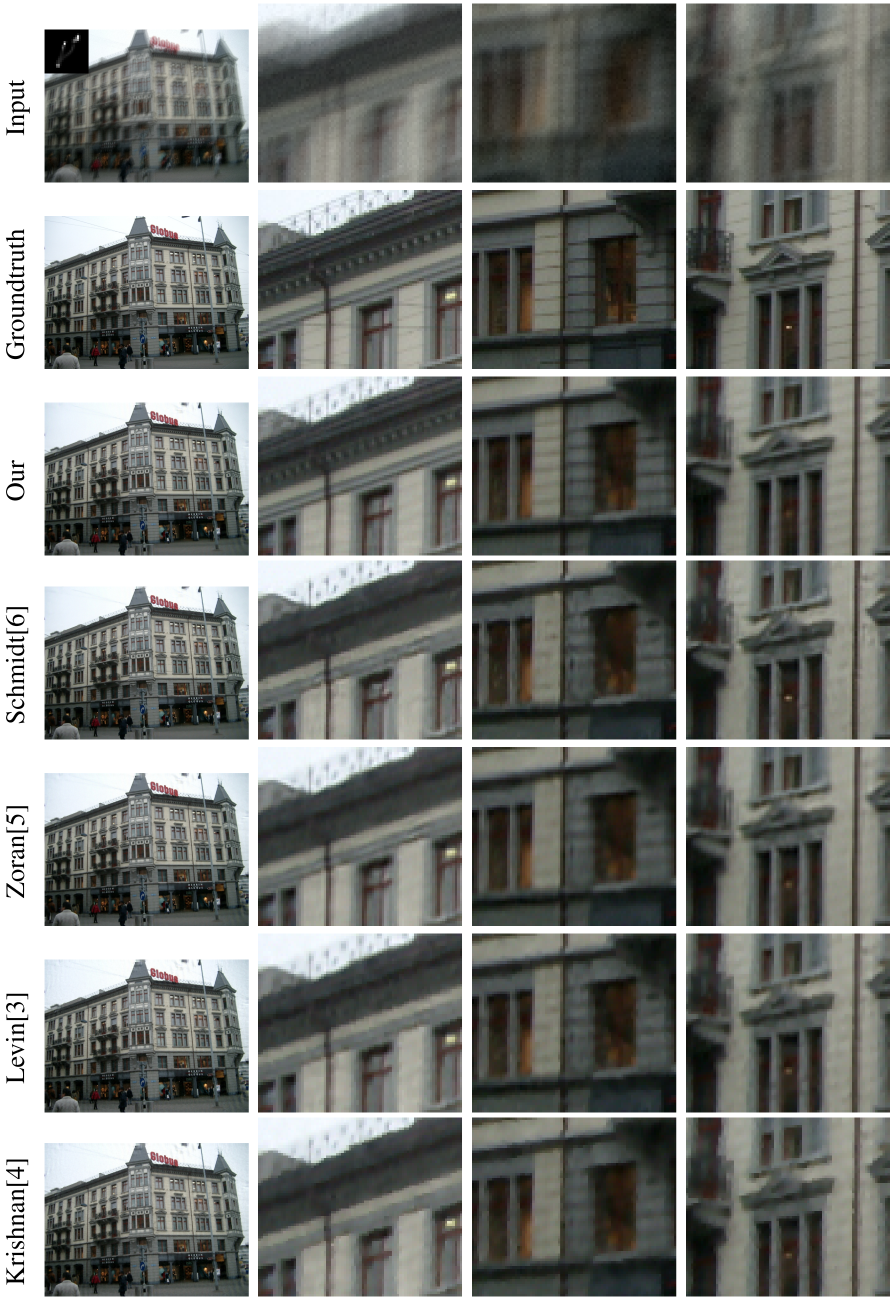


Fig. 13: results for image 2, kernel 8

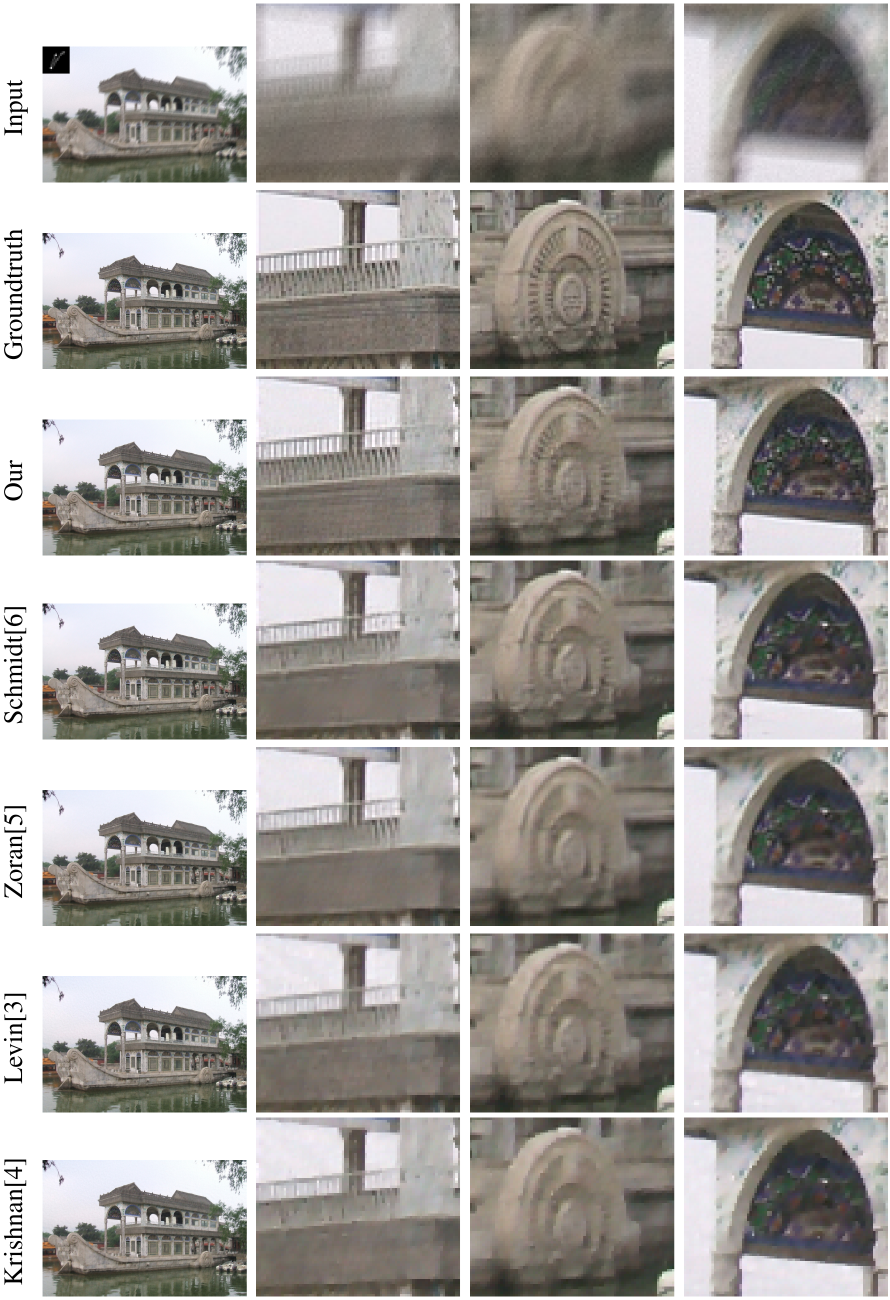


Fig. 14: results for image 3, kernel 2

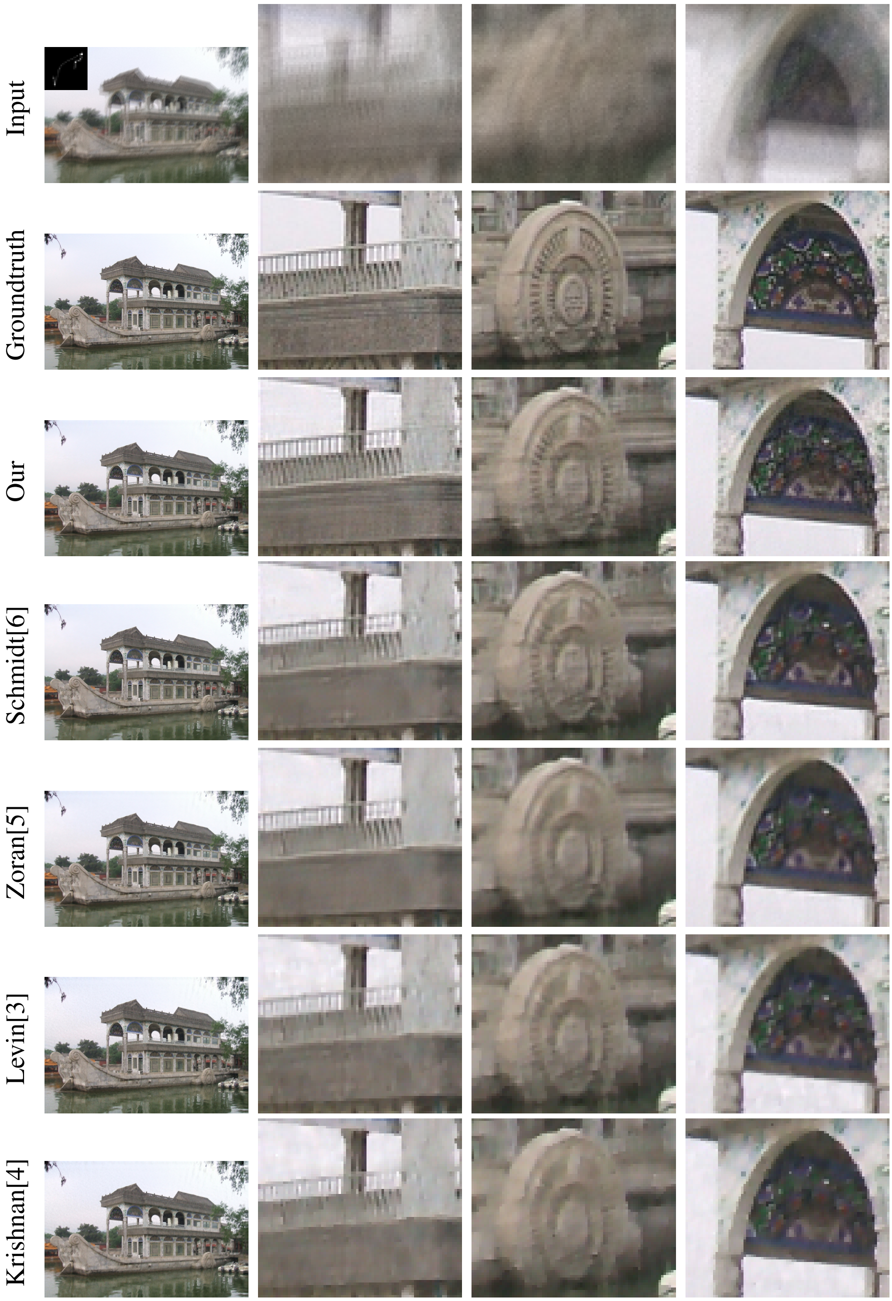


Fig. 15: results for image 3, kernel 4

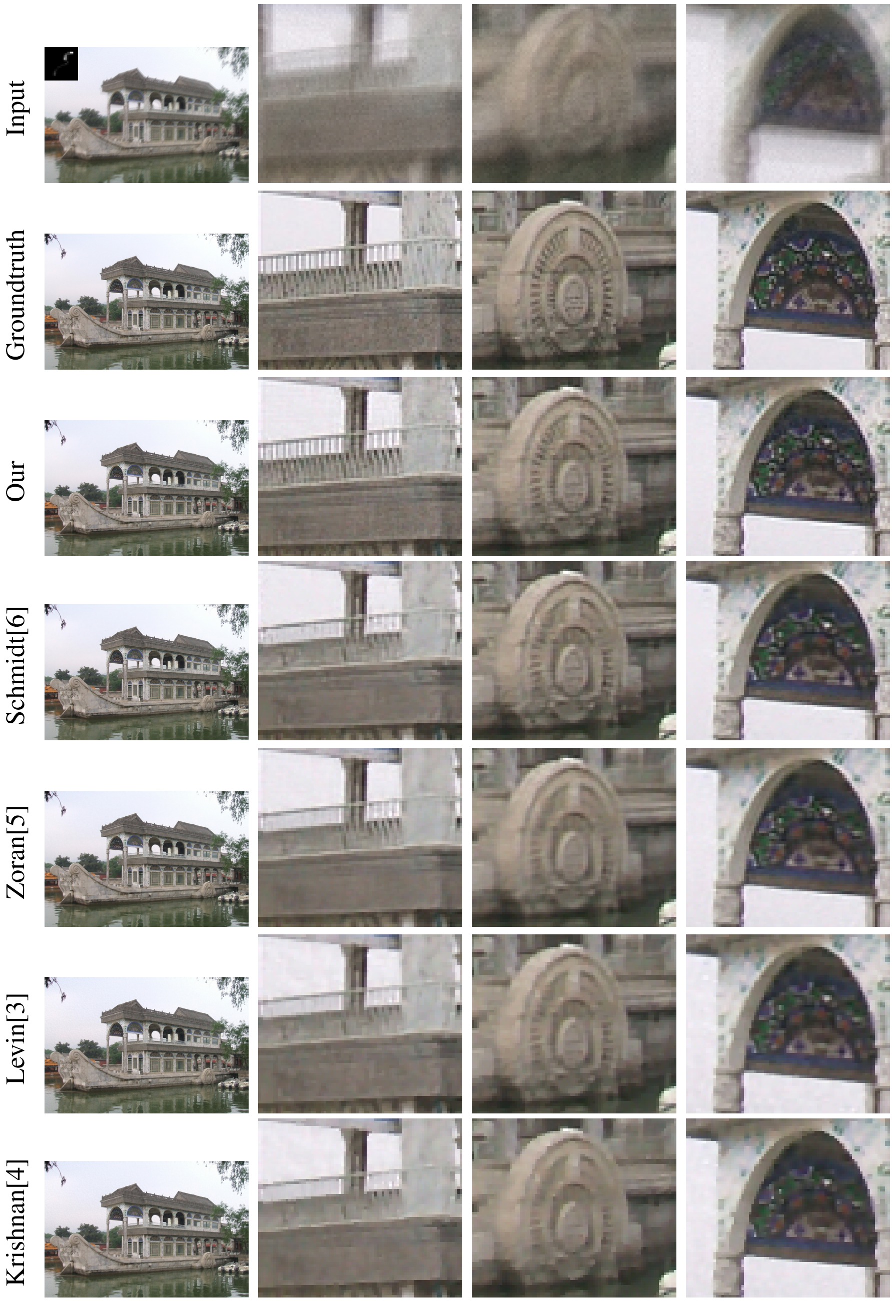


Fig. 16: results for image 3, kernel 6

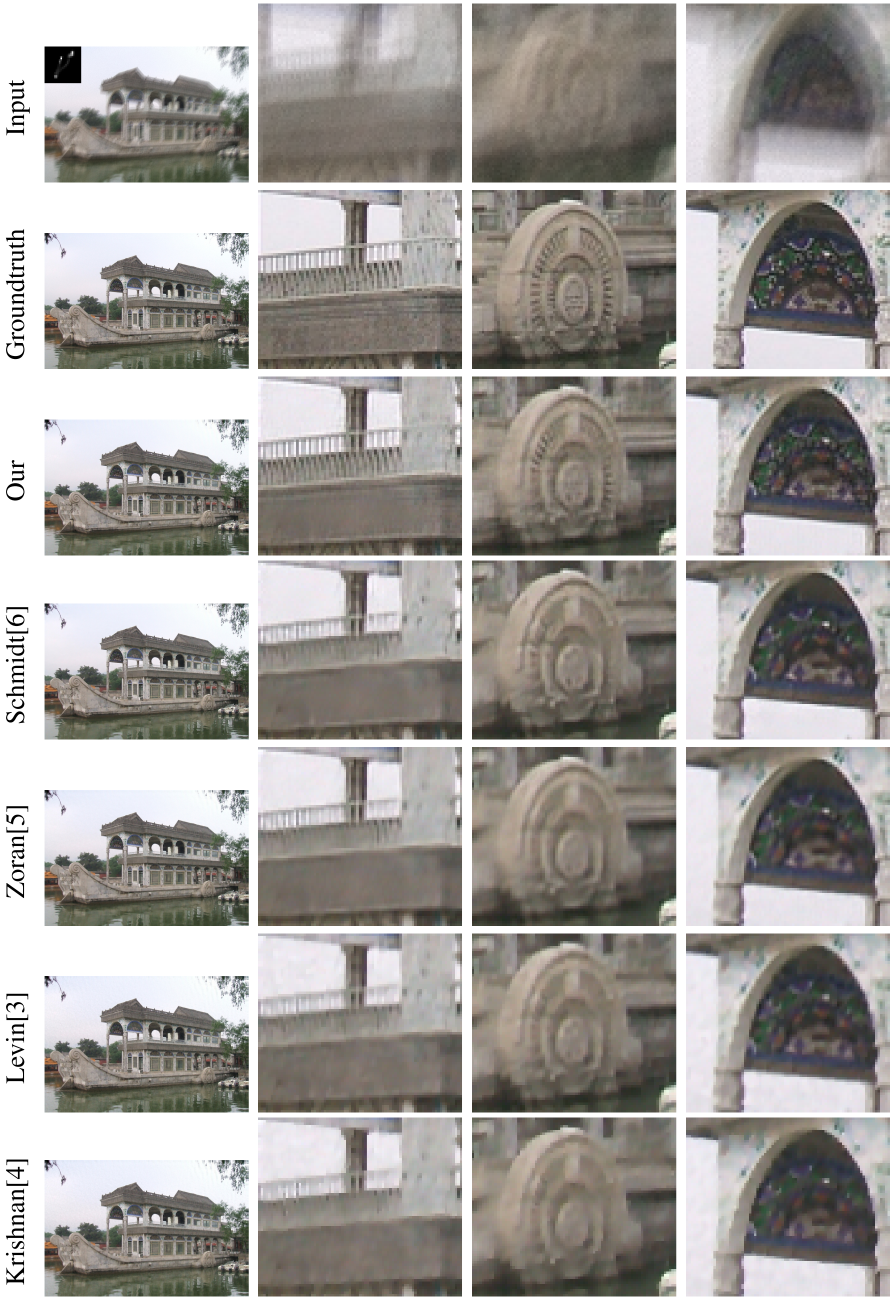


Fig. 17: results for image 3, kernel 8

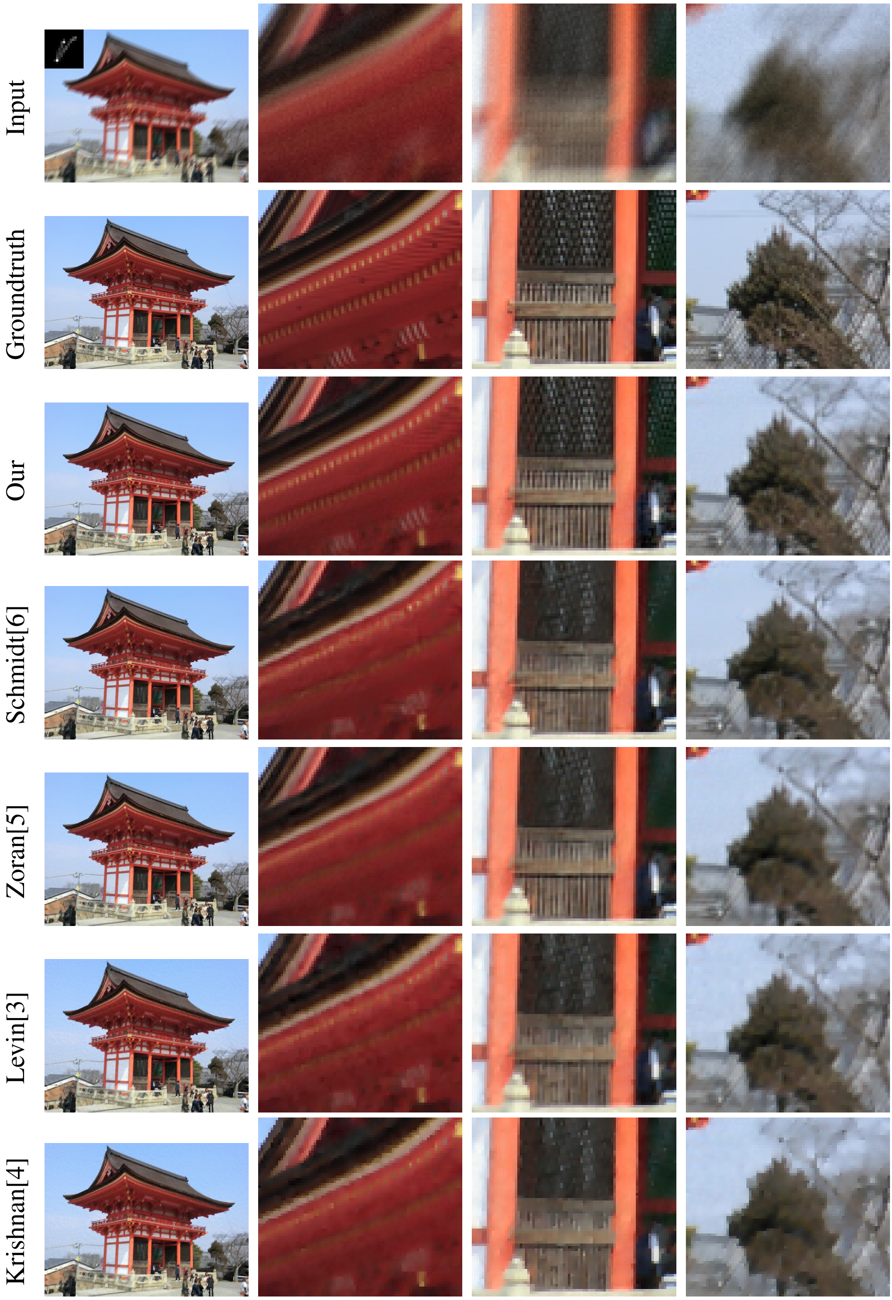


Fig. 18: results for image 4, kernel 2

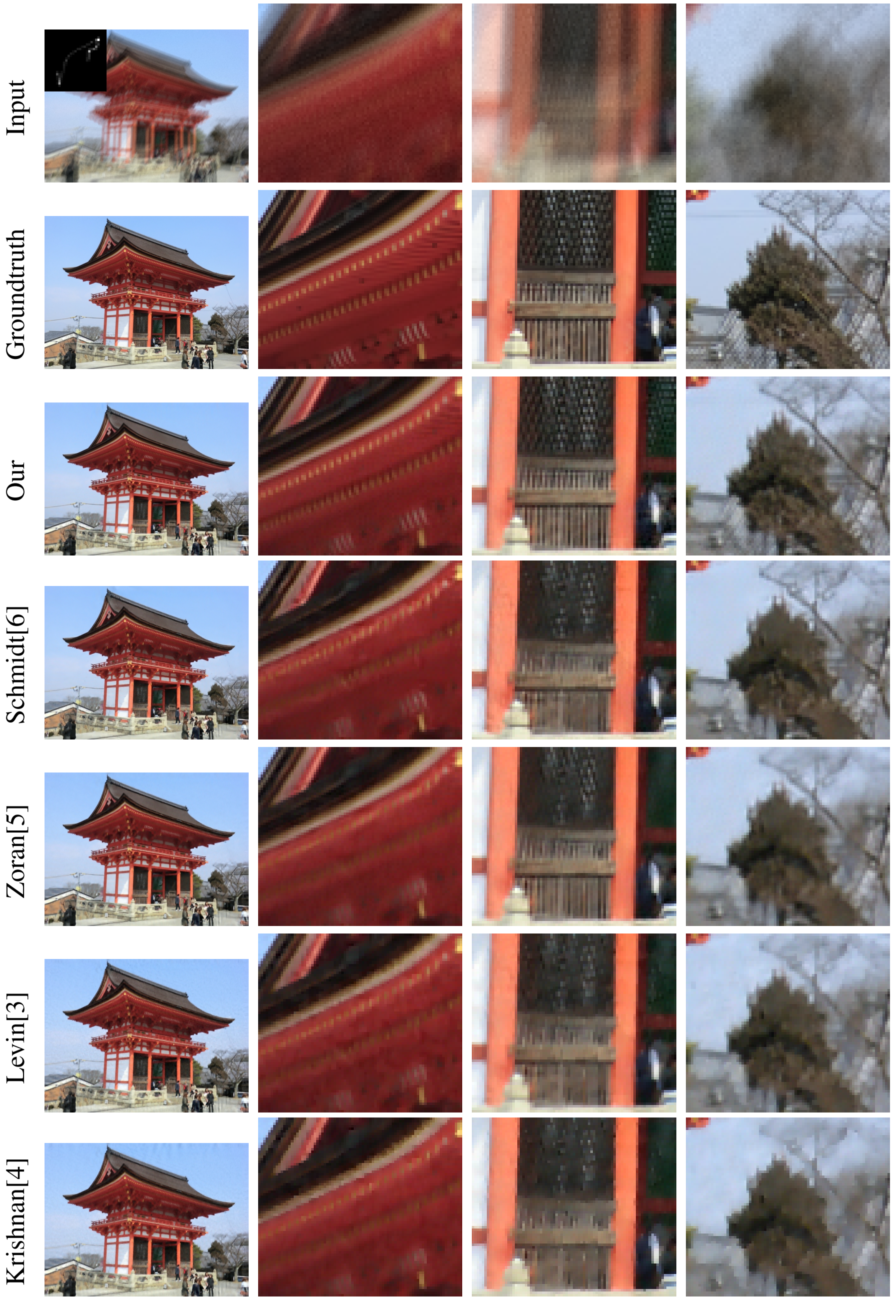


Fig. 19: results for image 4, kernel 4

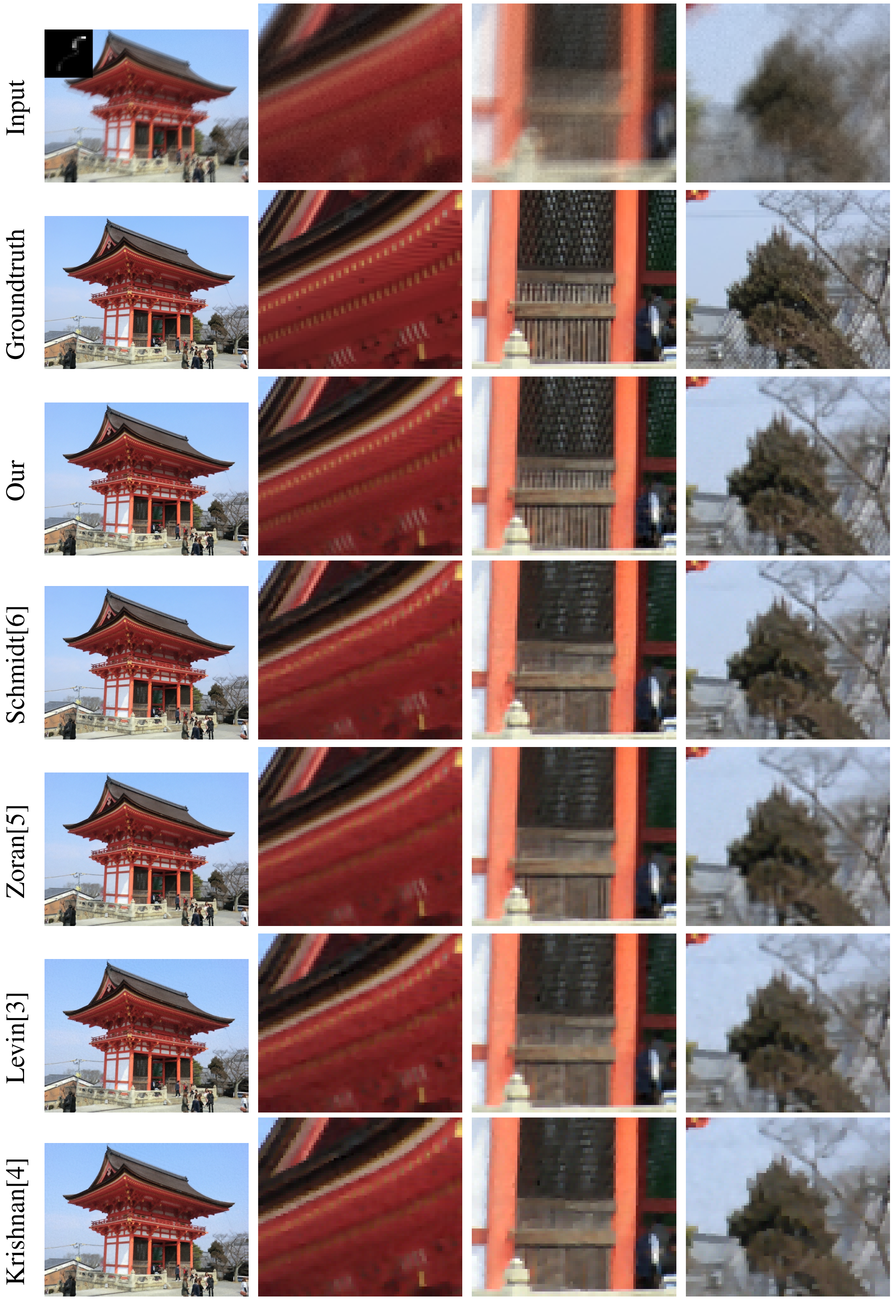


Fig. 20: results for image 4, kernel 6

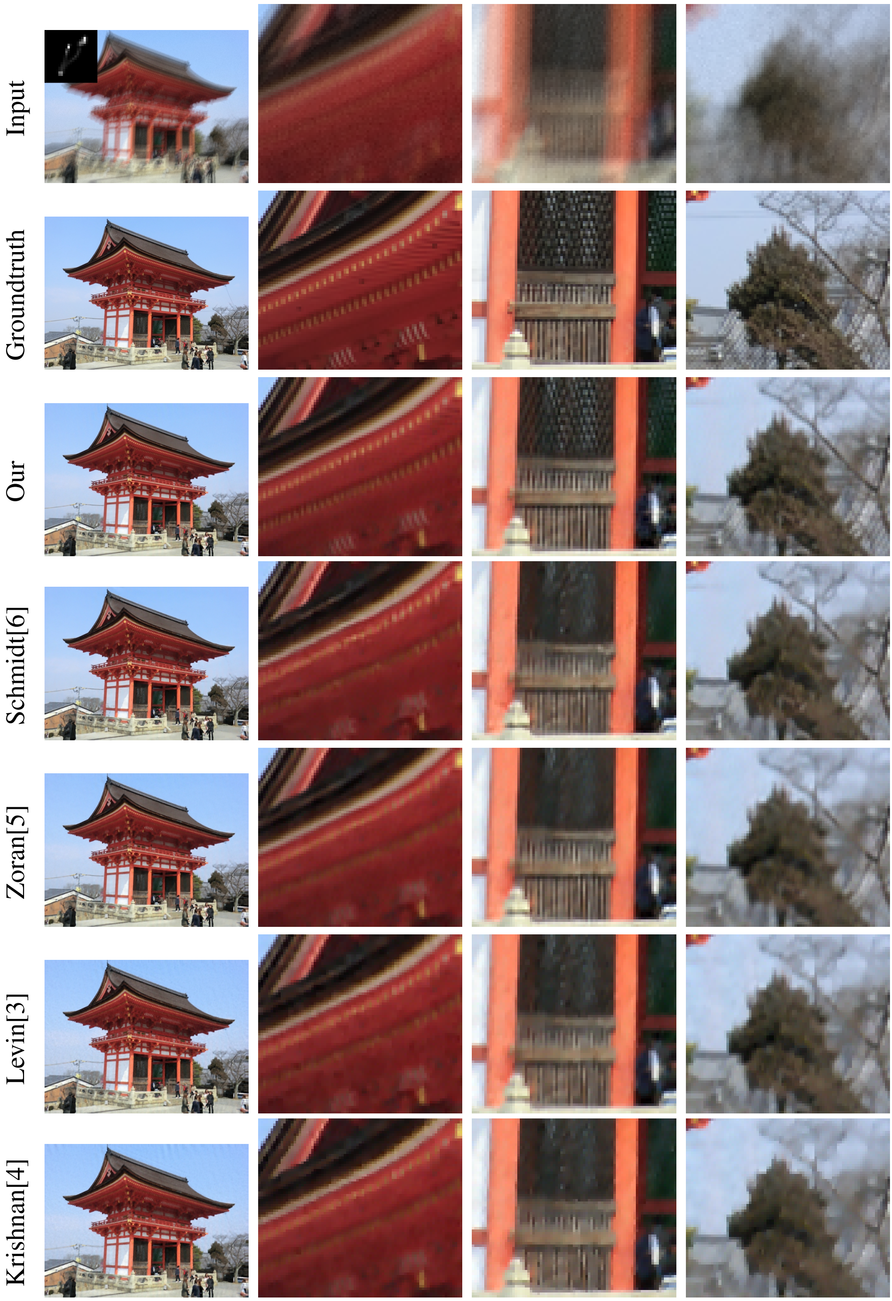


Fig. 21: results for image 4, kernel 8

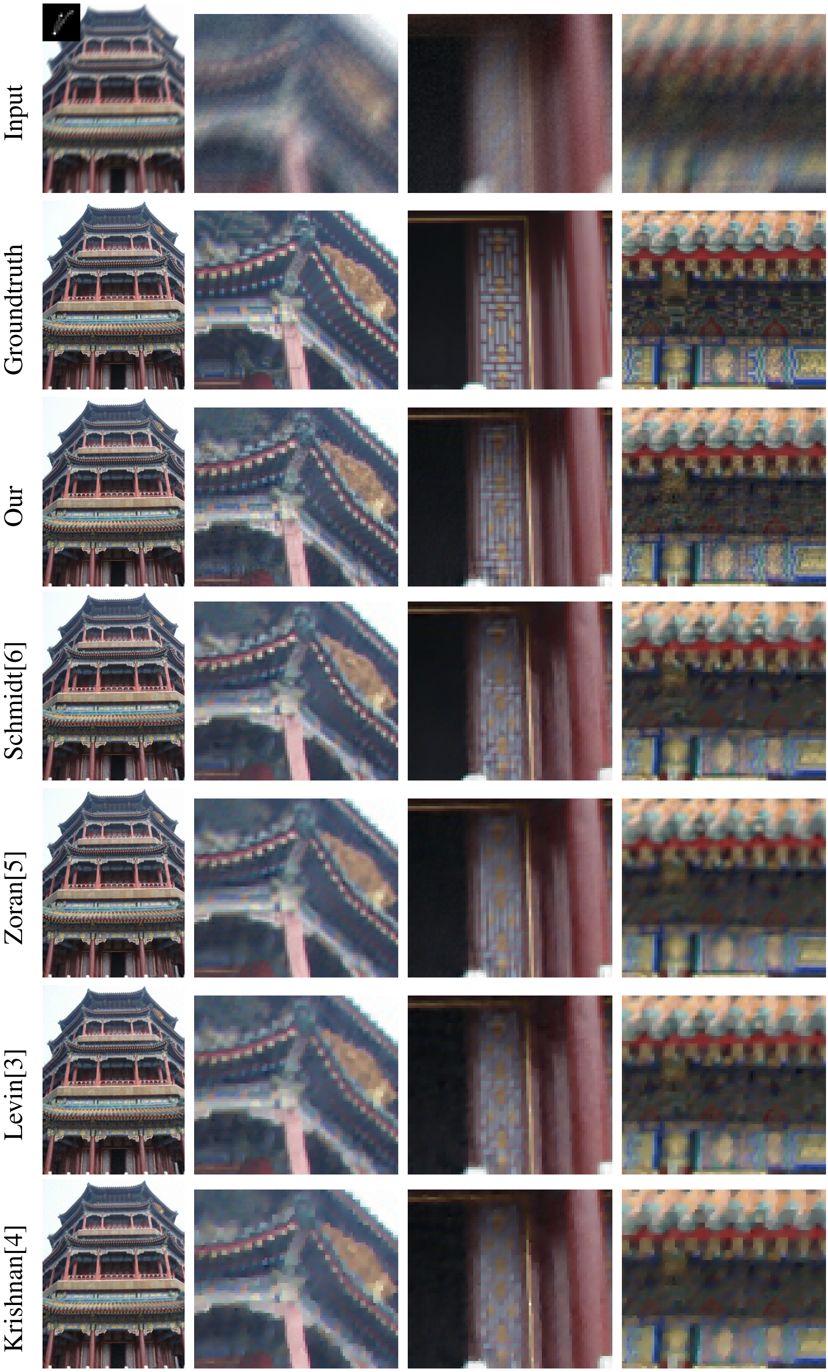


Fig. 22: results for image 5, kernel 2

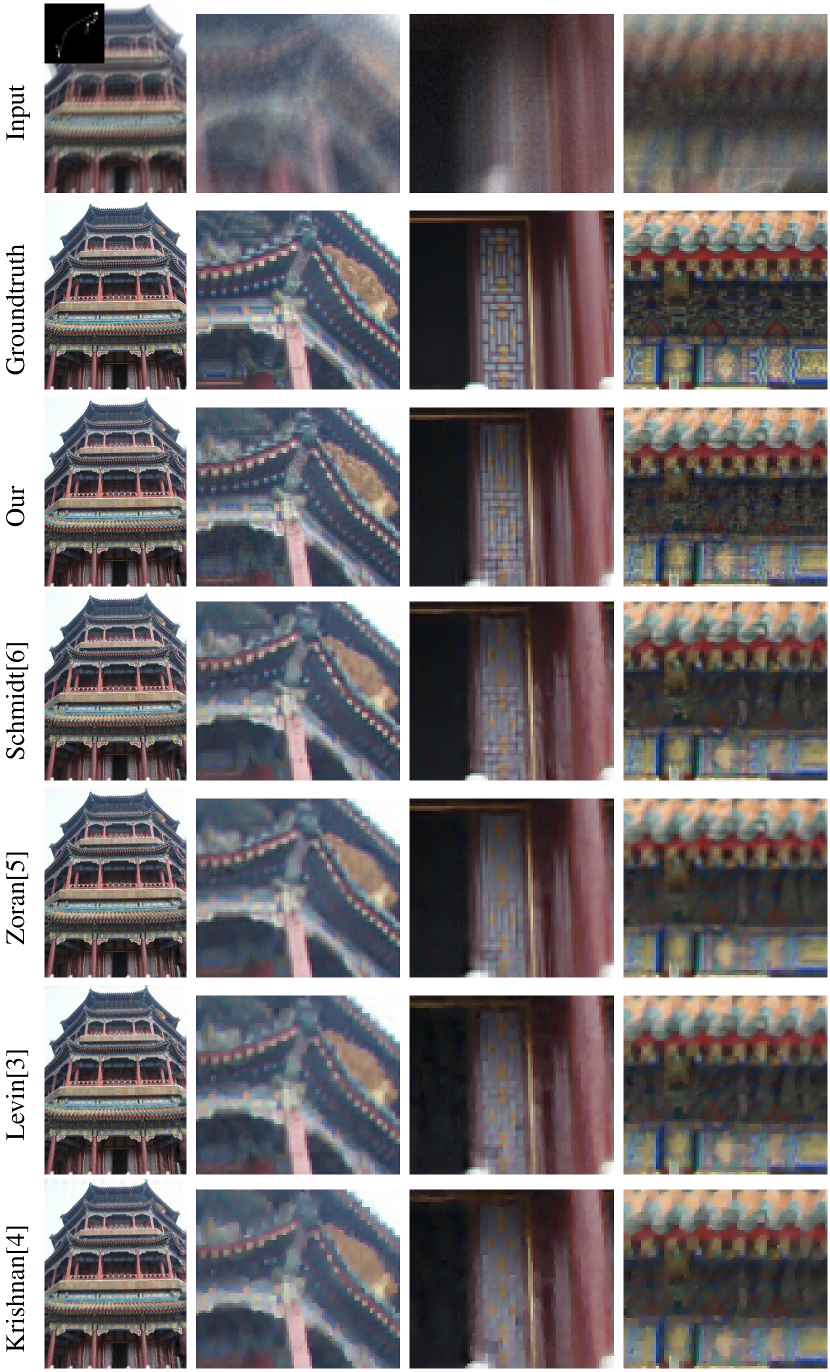


Fig. 23: results for image 5, kernel 4

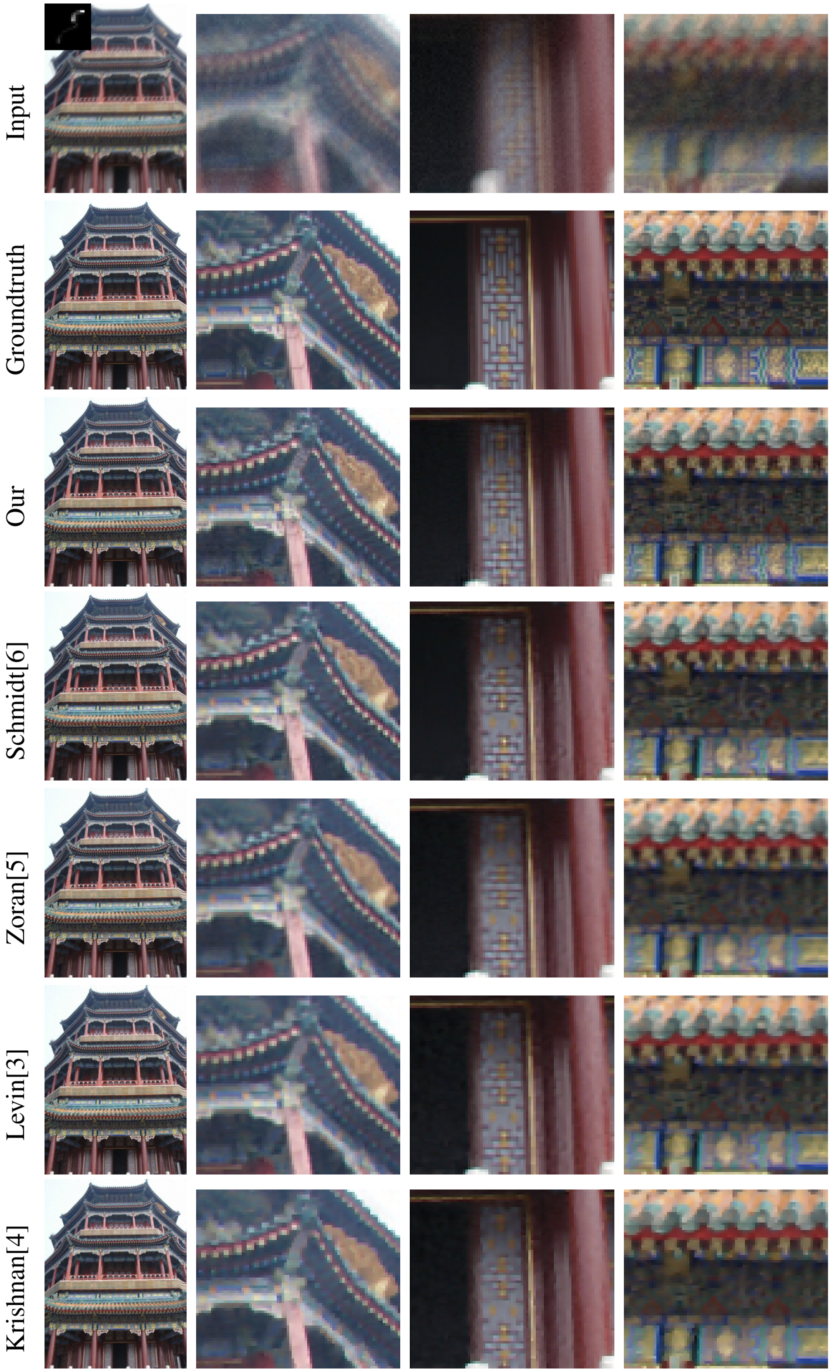


Fig. 24: results for image 5, kernel 6

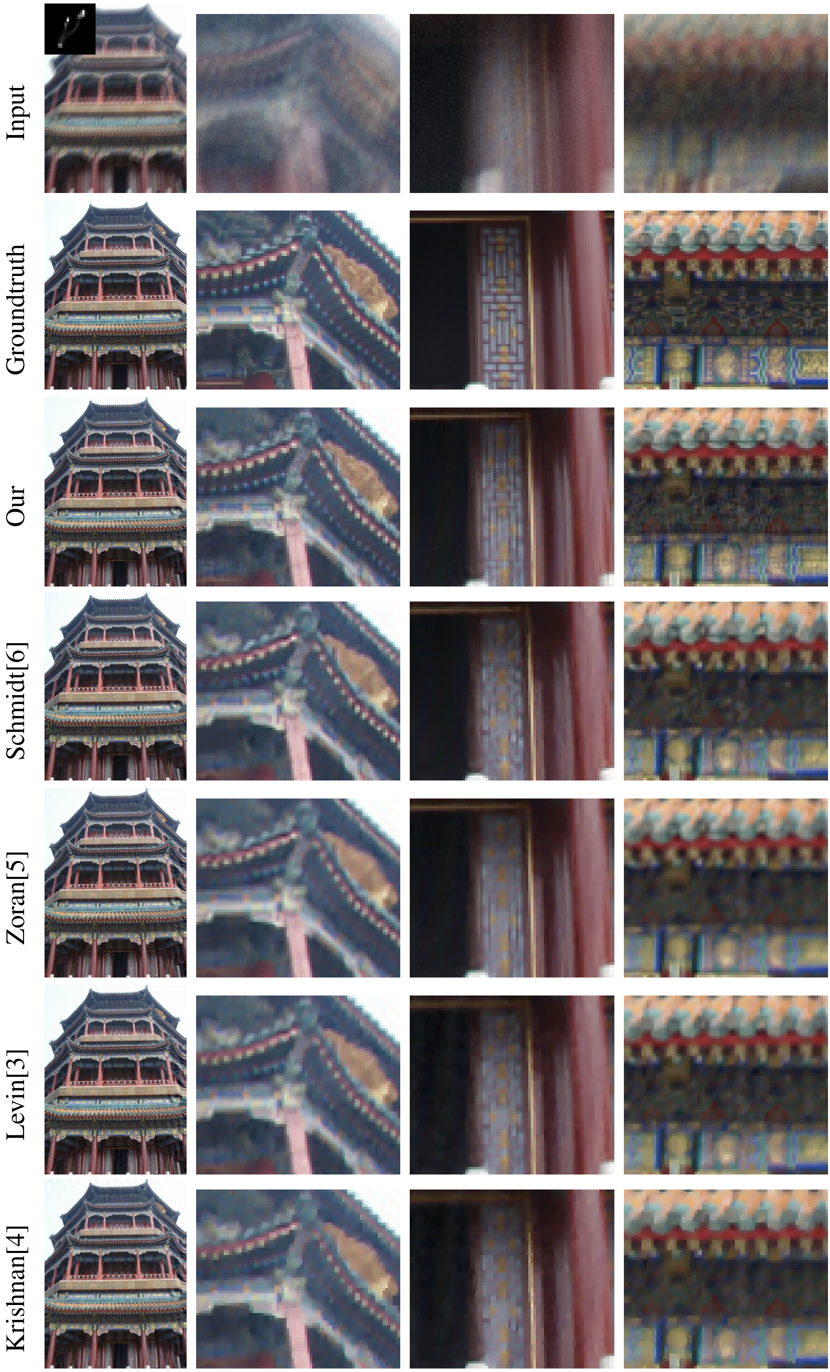


Fig. 25: results for image 5, kernel 8

3 Comparing Against By-example Methods

We are only able to identify a single work–[7]–tackling the problem of by-example deblurring. The authors of [7] kindly ran their system on our 20 test images using the supplied sharp example images. However, only one example image (per test case) is chosen by the authors of [7] to assist their blind deblurring method, since their system does not support multiple example images at the moment.

Using the PSF estimates from their results, we compute our latent image estimate based on our non-blind deconvolution method so that both methods use the same input information, hence the final outputs are directly comparable. The groundtruth PSF's are assumed unknown and not used in the deconvolution process. We show detailed comparisons against the by-example deblurring method of [7] in Fig. 26 through Fig. 45. In each figure, we show the results from [7] in the top row, and our results in the bottom row. The first column shows the full latent image, followed by two crops for close-up view of image details.

Because modern deconvolution algorithms can reliably recover low frequencies, it is important to examine these results closely at high resolution so that the differences in mid and high frequency content are apparent.

Quantitative comparisons are presented in Table 5 and Table 6.

3.1 Qualitative Comparisons

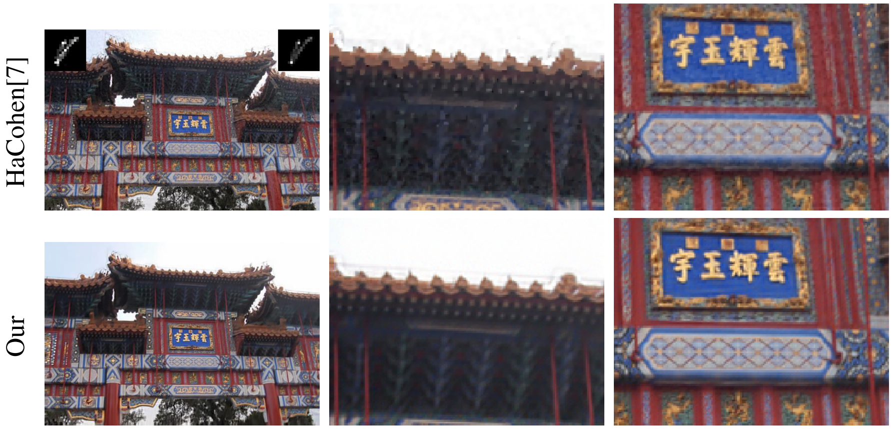


Fig. 26: Image 1, kernel 2. Top: output from HaCohen *et al.* [7], estimated PSF (top-left), ground truth PSF (top-right). Bottom: our results. Note that both methods use the estimated PSF for deconvolution; the ground truth PSF is not used.

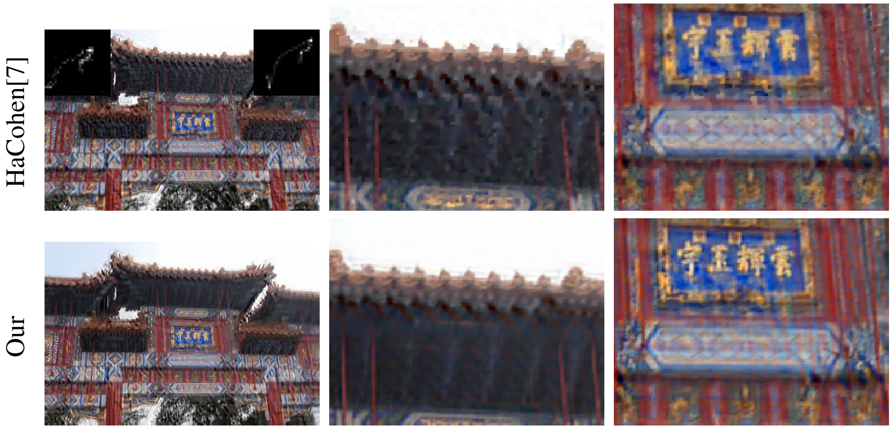


Fig. 27: Image 1, kernel 4. Top: output from HaCohen *et al.* [7], estimated PSF (top-left), ground truth PSF (top-right). Bottom: our results. Note that both methods use the estimated PSF for deconvolution; the ground truth PSF is not used.

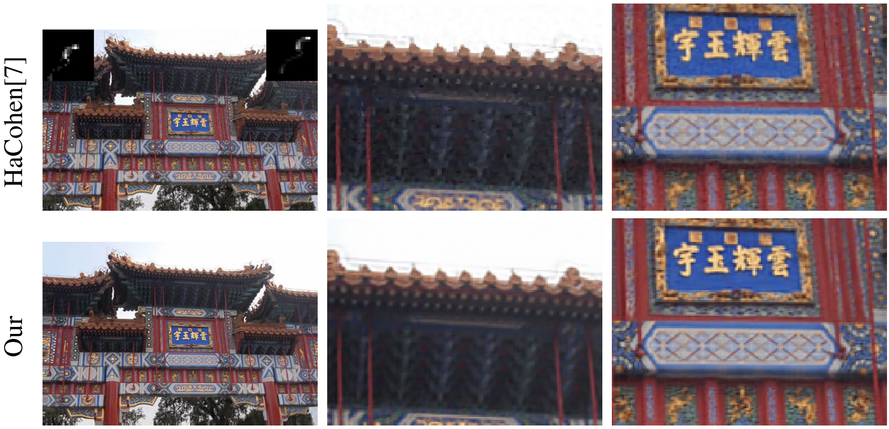


Fig. 28: Image 1, kernel 6. Top: output from HaCohen *et al.* [7], estimated PSF (top-left), ground truth PSF (top-right). Bottom: our results. Note that both methods use the estimated PSF for deconvolution; the ground truth PSF is not used.

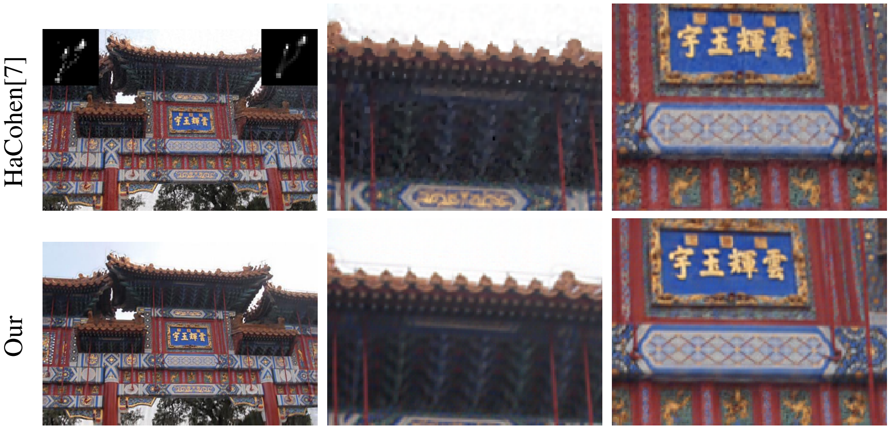


Fig. 29: Image 1, kernel 8. Top: output from HaCohen *et al.* [7], estimated PSF (top-left), ground truth PSF (top-right). Bottom: our results. Note that both methods use the estimated PSF for deconvolution; the ground truth PSF is not used.

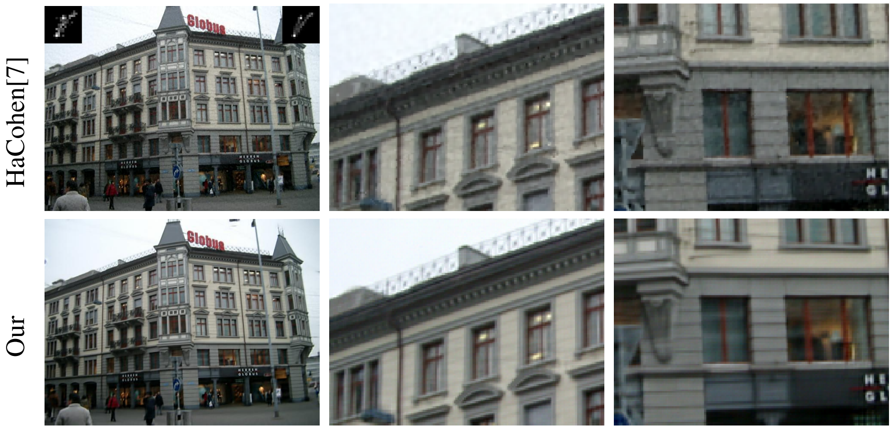


Fig. 30: Image 2, kernel 2. Top: output from HaCohen *et al.* [7], estimated PSF (top-left), ground truth PSF (top-right). Bottom: our results. Note that both methods use the estimated PSF for deconvolution; the ground truth PSF is not used.

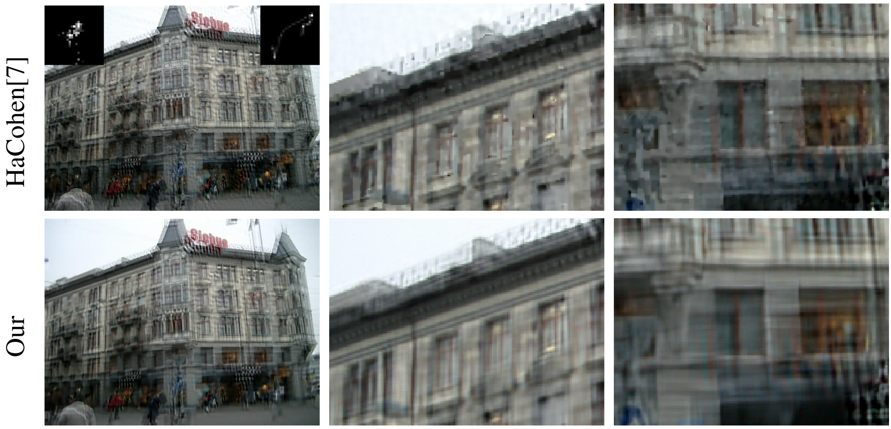


Fig. 31: Image 2, kernel 4. Top: output from HaCohen *et al.* [7], estimated PSF (top-left), ground truth PSF (top-right). Bottom: our results. Note that both methods use the estimated PSF for deconvolution; the ground truth PSF is not used.

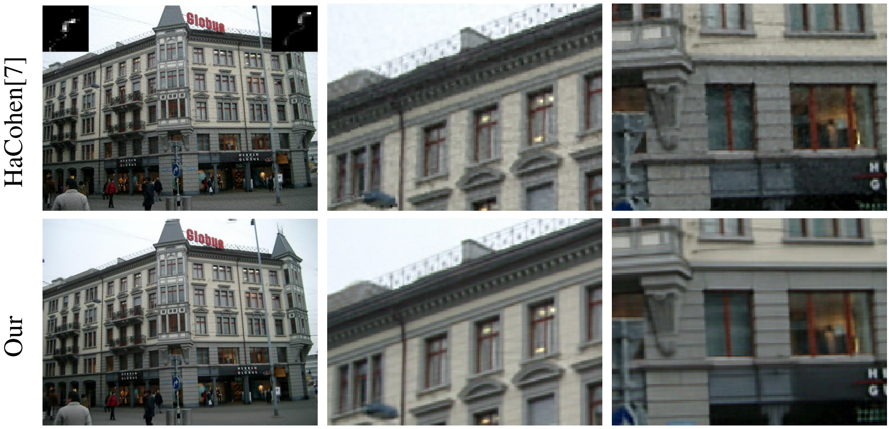


Fig. 32: Image 2, kernel 6. Top: output from HaCohen *et al.* [7], estimated PSF (top-left), ground truth PSF (top-right). Bottom: our results. Note that both methods use the estimated PSF for deconvolution; the ground truth PSF is not used.

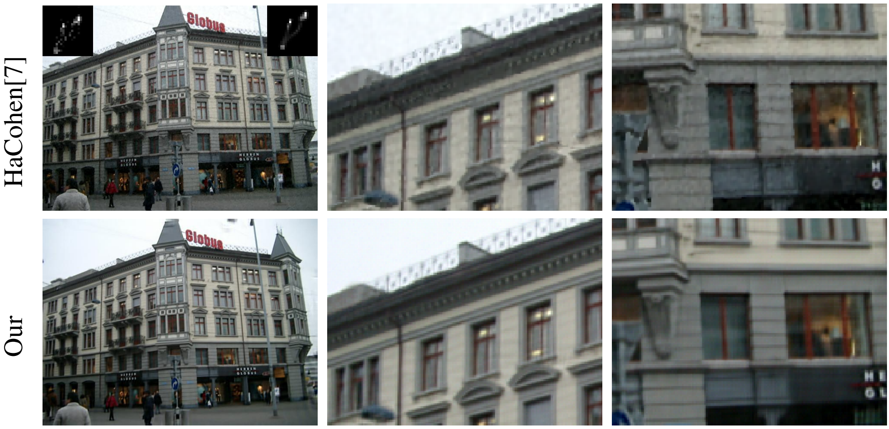


Fig. 33: Image 2, kernel 8. Top: output from HaCohen *et al.* [7], estimated PSF (top-left), ground truth PSF (top-right). Bottom: our results. Note that both methods use the estimated PSF for deconvolution; the ground truth PSF is not used.

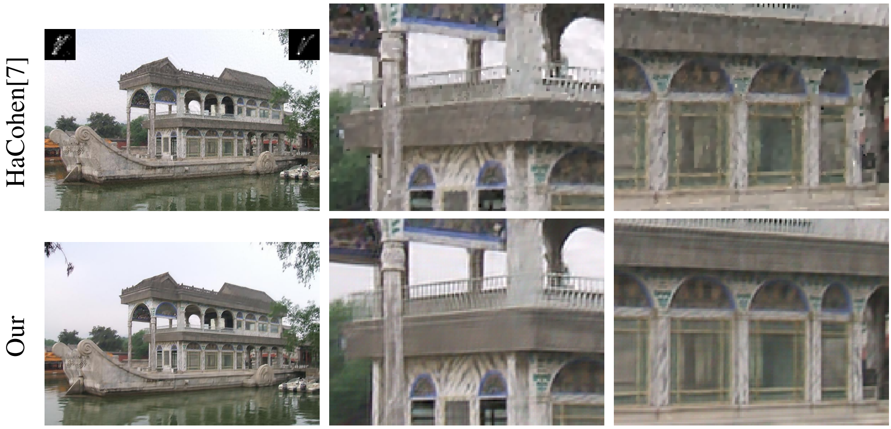


Fig. 34: Image 3, kernel 2. Top: output from HaCohen *et al.* [7], estimated PSF (top-left), ground truth PSF (top-right). Bottom: our results. Note that both methods use the estimated PSF for deconvolution; the ground truth PSF is not used.

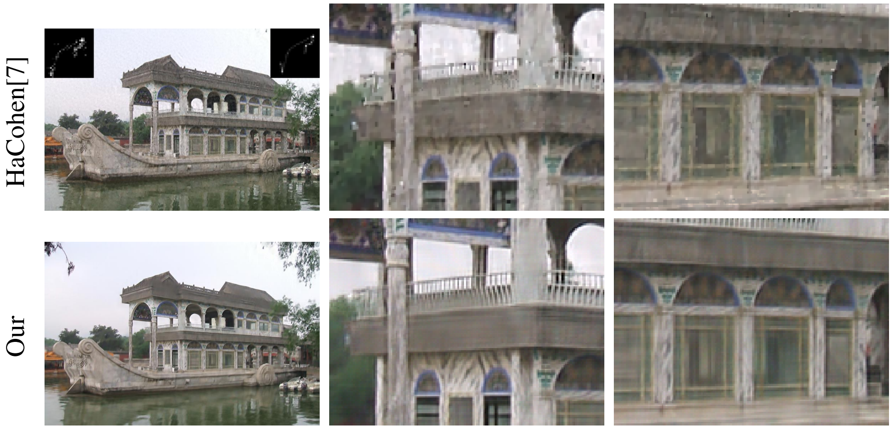


Fig. 35: Image 3, kernel 4. Top: output from HaCohen *et al.* [7], estimated PSF (top-left), ground truth PSF (top-right). Bottom: our results. Note that both methods use the estimated PSF for deconvolution; the ground truth PSF is not used.

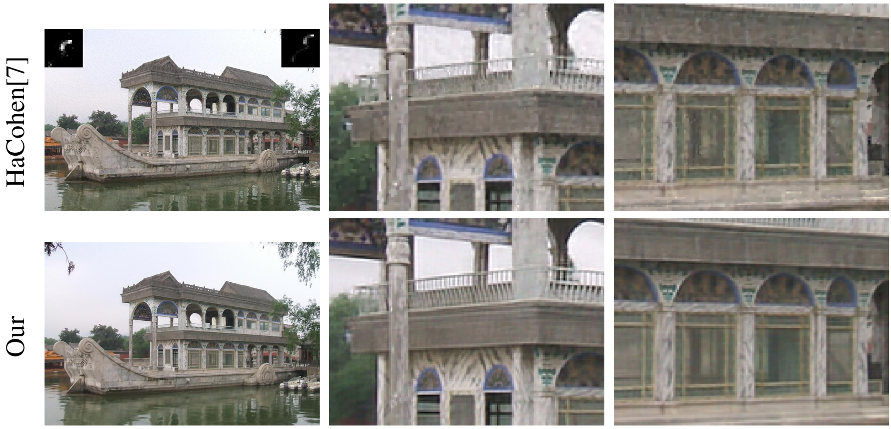


Fig. 36: Image 3, kernel 6. Top: output from HaCohen *et al.* [7], estimated PSF (top-left), ground truth PSF (top-right). Bottom: our results. Note that both methods use the estimated PSF for deconvolution; the ground truth PSF is not used.

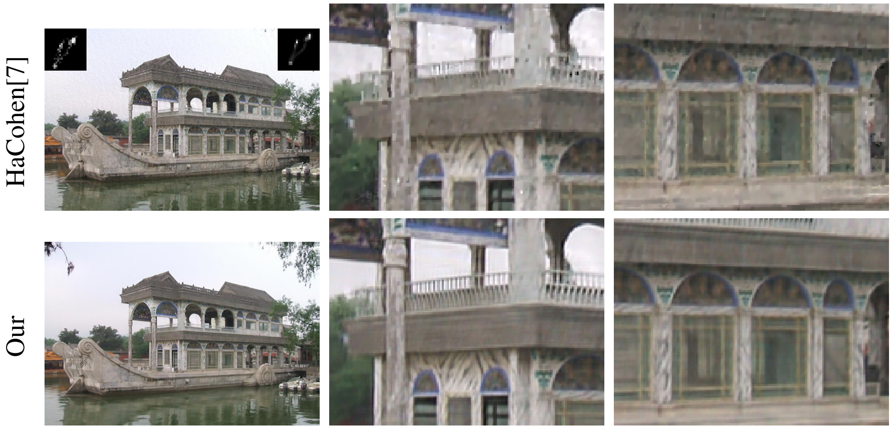


Fig. 37: Image 3, kernel 8. Top: output from HaCohen *et al.* [7], estimated PSF (top-left), ground truth PSF (top-right). Bottom: our results. Note that both methods use the estimated PSF for deconvolution; the ground truth PSF is not used.

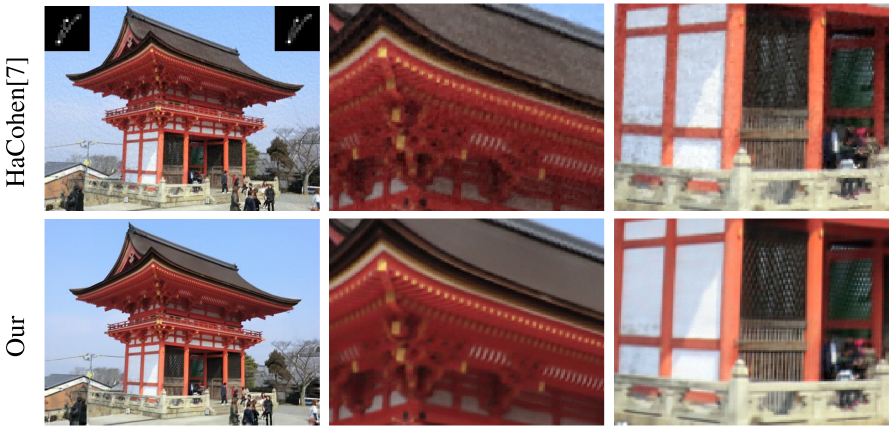


Fig. 38: Image 4, kernel 2. Top: output from HaCohen *et al.* [7], estimated PSF (top-left), ground truth PSF (top-right). Bottom: our results. Note that both methods use the estimated PSF for deconvolution; the ground truth PSF is not used.

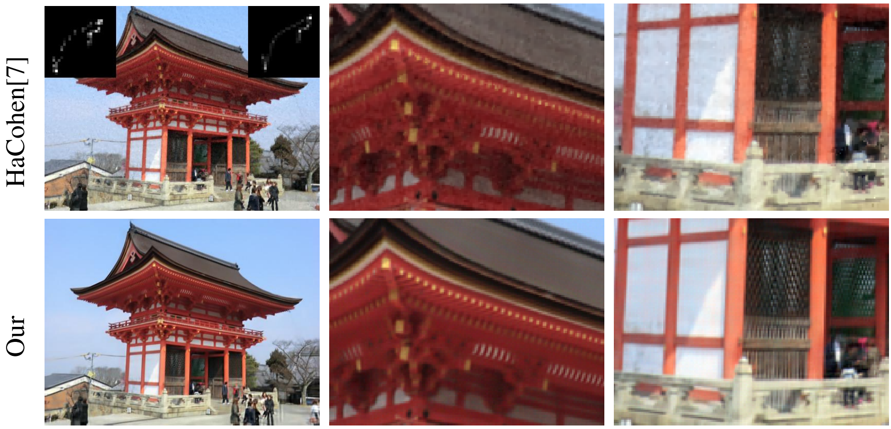


Fig. 39: Image 4, kernel 4. Top: output from HaCohen *et al.* [7], estimated PSF (top-left), ground truth PSF (top-right). Bottom: our results. Note that both methods use the estimated PSF for deconvolution; the ground truth PSF is not used.

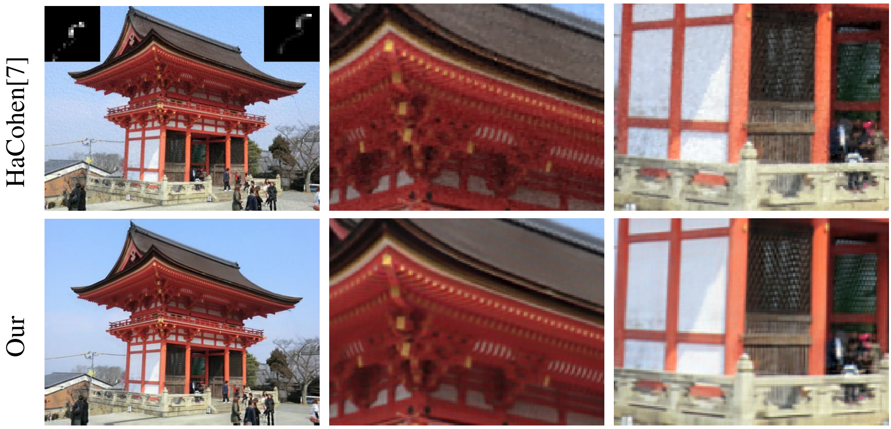


Fig. 40: Image 4, kernel 6. Top: output from HaCohen *et al.* [7], estimated PSF (top-left), ground truth PSF (top-right). Bottom: our results. Note that both methods use the estimated PSF for deconvolution; the ground truth PSF is not used.

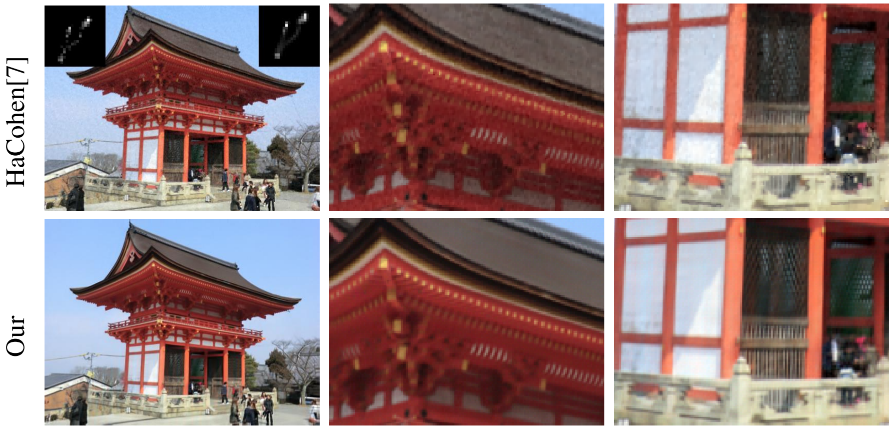


Fig. 41: Image 4, kernel 8. Top: output from HaCohen *et al.* [7], estimated PSF (top-left), ground truth PSF (top-right). Bottom: our results. Note that both methods use the estimated PSF for deconvolution; the ground truth PSF is not used.

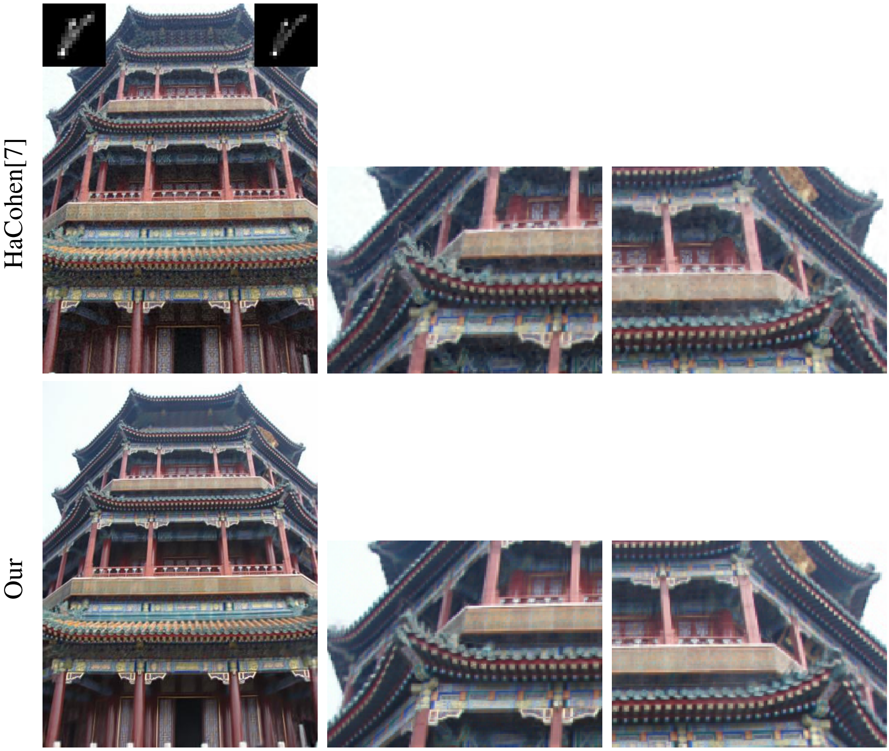


Fig. 42: Image 5, kernel 2. Top: output from HaCohen *et al.* [7], estimated PSF (top-left), ground truth PSF (top-right). Bottom: our results. Note that both methods use the estimated PSF for deconvolution; the ground truth PSF is not used.

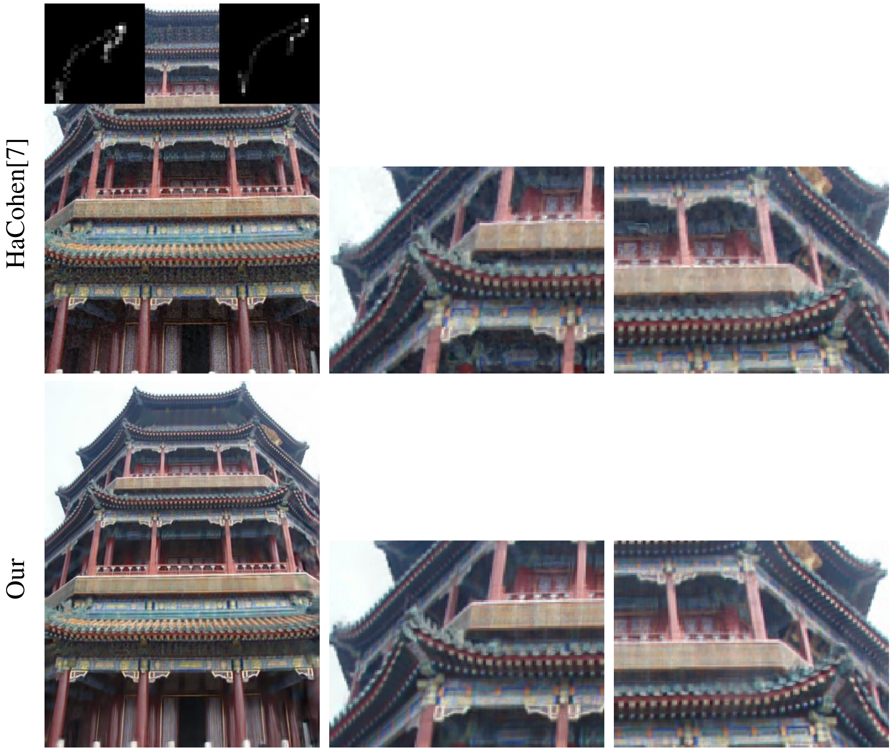


Fig. 43: Image 5, kernel 4. Top: output from HaCohen *et al.* [7], estimated PSF (top-left), ground truth PSF (top-right). Bottom: our results. Note that both methods use the estimated PSF for deconvolution; the ground truth PSF is not used.

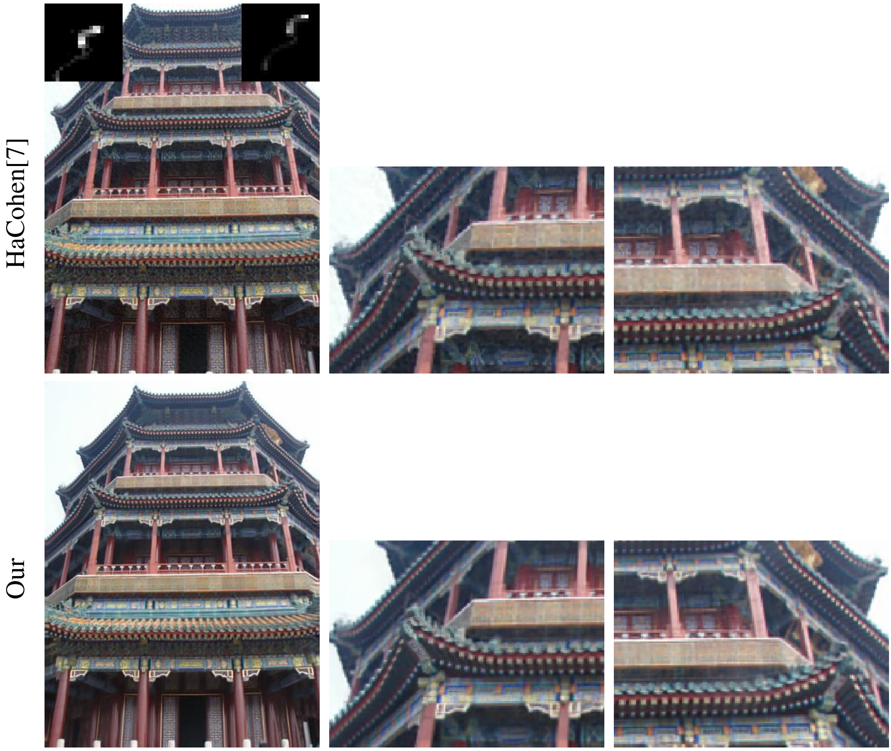


Fig. 44: Image 5, kernel 6. Top: output from HaCohen *et al.* [7], estimated PSF (top-left), ground truth PSF (top-right). Bottom: our results. Note that both methods use the estimated PSF for deconvolution; the ground truth PSF is not used.

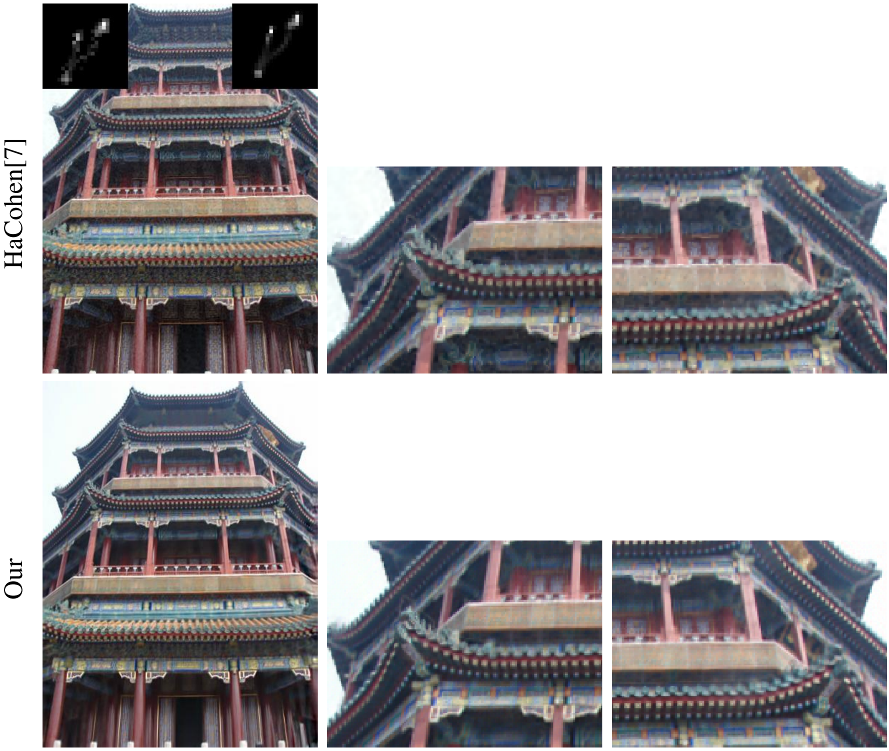


Fig. 45: Image 5, kernel 8. Top: output from HaCohen *et al.* [7], estimated PSF (top-left), ground truth PSF (top-right). Bottom: our results. Note that both methods use the estimated PSF for deconvolution; the ground truth PSF is not used.

3.2 Quantitative Comparisons

In Table 5, we show a quantitative evaluation of performance gain over HaCohen *et al.* [7] in terms of PSNR and SSIM. This is computed by considering the best possible alignment with integer offset values. Because kernel 4 is consistently estimated with large error by [7], our non-blind deconvolution performance gain for kernel 4 seems to fluctuate more than other kernels. For kernel 6, we observe slight subpixel misalignment (with respect to the groundtruth), hence producing negative PSNR gain for image 4 and 6, while maintaining positive (or close to zero) SSIM gain.

In Table 6, we report the same evaluation based on best possible subpixel alignment, linearly interpolating pixel values when needed. Under this regime, our method is able to outperform [7] 100% in terms of SSIM, and on all but one test image in terms of PSNR.

To be consistent, results in Table 5 is reported in the paper.

PSNR gain (mean= +0.5973)

	kernel 2	kernel 4	kernel 6	kernel 8
image 1	1.2577	0.0108	0.3199	1.1328
image 2	0.6379	−0.2493	0.3949	1.2702
image 3	0.7481	0.3385	0.7273	1.0560
image 4	2.2467	0.9682	−2.3442	0.1681
image 5	2.5524	0.5723	−0.5494	0.6863

SSIM gain (mean= +0.0255)

	kernel 2	kernel 4	kernel 6	kernel 8
image 1	0.0418	0.0163	0.0272	0.0323
image 2	0.0260	−0.0216	0.0182	0.0292
image 3	0.0172	0.0199	0.0296	0.0340
image 4	0.0683	0.0582	0.0098	0.0359
image 5	0.0385	0.0169	−0.0050	0.0176

Table 5: Performance gain over HaCohen *et al.* [7]. Computed with best integer alignment.

PSNR gain (mean= +0.9095)

	kernel 2	kernel 4	kernel 6	kernel 8
image 1	1.2167	0.0715	0.5364	0.9130
image 1	0.9138	1.1842	0.6209	1.2702
image 1	0.3351	0.4631	0.6375	0.5794
image 1	1.9252	1.5779	−0.1863	1.0064
image 1	2.2246	0.9046	0.8665	1.1290

SSIM gain (mean= +0.0308)

	kernel 2	kernel 4	kernel 6	kernel 8
image 1	0.0333	0.0091	0.0144	0.0309
image 2	0.0244	0.1671	0.0172	0.0292
image 3	0.0115	0.0174	0.0175	0.0162
image 4	0.0558	0.0538	0.0360	0.0374
image 5	0.0243	0.0071	0.0088	0.0052

Table 6: Performance gain over HaCohen *et al.* [7]. Computed with best subpixel alignment.

4 Additional Visualizations

In this section we provide additional visualizations for how deblur quality varies given example images at different levels of similarity. Similar visualizations for test image 1 and 2 have been presented in the paper, here we show the same for test image 3,4 and 5. For ease of comparison, we used kernel 4 from [3] for all results shown in this section.

Our experiments show that single-scale patch priors often lack the ability to insert high frequency details and mid frequency structures, regardless of whether it is globally or locally trained. Furthermore, global training does not seem to benefit from having better example images, showing similar results from left to right. Finally, we observe the best quality in deblurred results from combining locally training with a multi-scale patch pyramid setup.

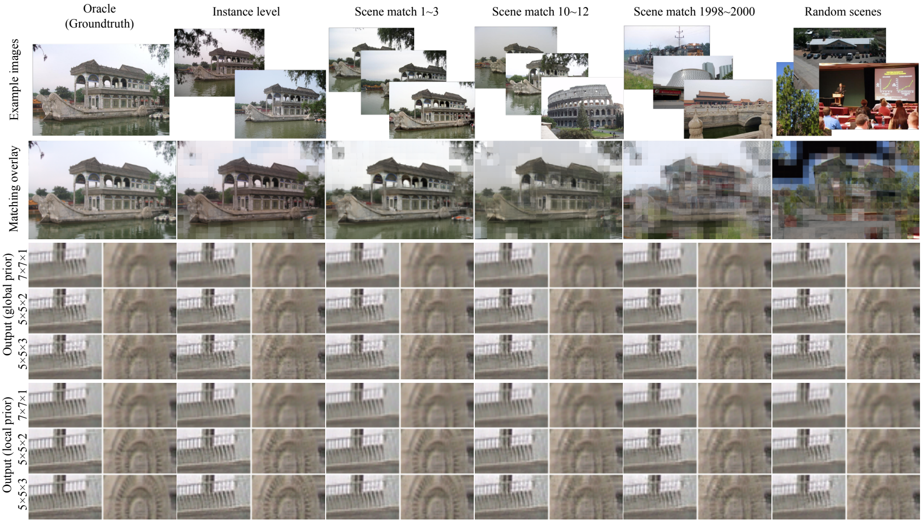


Fig. 46: Deblur performance across various example image scenarios for test image 3.

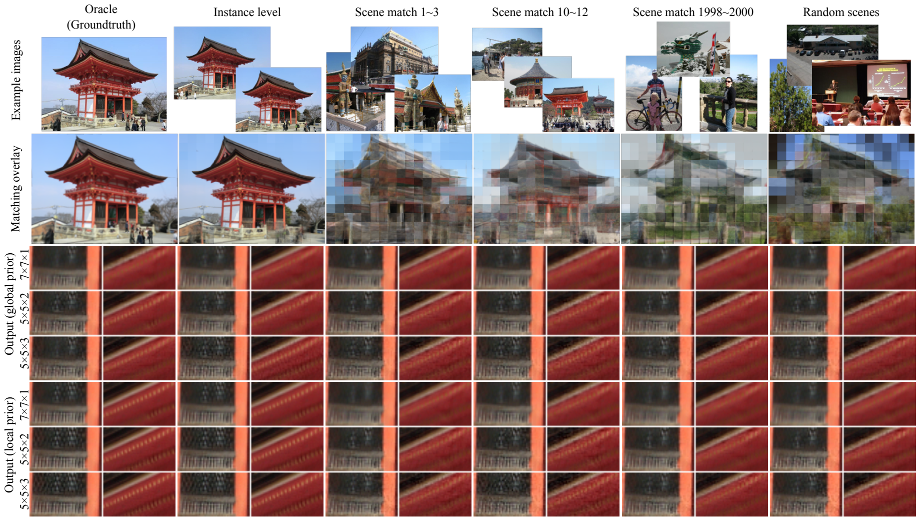


Fig. 47: Deblur performance across various example image scenarios for test image 4.

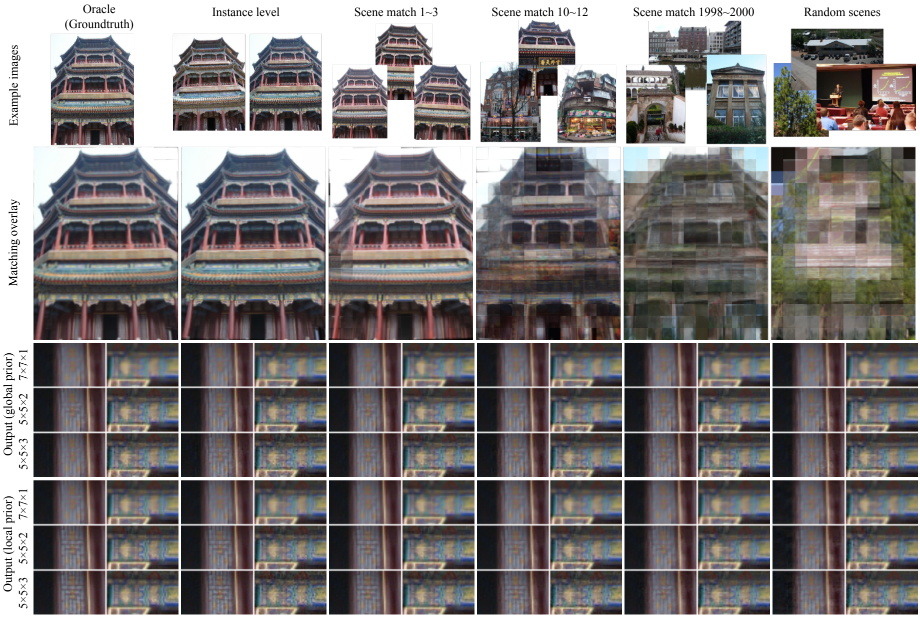


Fig. 48: Deblur performance across various example image scenarios for test image 5.

References

1. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Understanding and evaluating blind deconvolution algorithms. In: CVPR. (2009)
2. Yue, H., Sun, X., Yang, J., Wu, F.: Landmark image super-resolution by retrieving web images. In: Image Processing, IEEE Transactions on. (2013)
3. Levin, A., Fergus, R., Durand, F., Freeman, W.T.: Image and depth from a conventional camera with a coded aperture. In: ACM Transactions on Graphics. (2007)
4. Krishnan, D., Fergus, R.: Fast image deconvolution using hyper-laplacian priors. In: NIPS. (2009)
5. Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: ICCV. (2011)
6. Schmidt, U., Rother, C., Nowozin, S., Jancsary, J., Roth, S.: Discriminative non-blind deblurring. In: CVPR. (2013)
7. HaCohen, Y., Shechtman, E., Lischinski, D.: Deblurring by example using dense correspondence. In: ICCV. (2013)