Supplementary Materiels: DSDNet

Pin-Hung Kuo[1], Jinshan Pan[2], Shao-Yi Chien[1], and Ming-Hsuan Yang[3,4,5]

[1]Graduate Institute of Electronics Engineering, National Taiwan University
[2]Nanjing University of Science and Technology
[3]Google Research [4]University of California, Merced [5]Yonsei University

1 Overview

We present additional results in this supplementary document. First, we show the detailed network configurations in Section 2. Second, we provide further detailed analysis on the effect of the proposed critical components in our method (Section 4). Then we show more qualitative comparisons of the state-of-the-art methods and our models (Section 5). Finally, Section 6 shows some failure cases and analyzes the limitations of the proposed method.

2 Detailed Network Configurations

We show the detailed network configurations in Table 1.

Table 1: Architecture of the proposed DSDNet. "conv","tconv", "Max-Pool" and "Avg-Pool" denote the convolutional layer, transposed convolutional layer, Max Pooling layer and Average Pooling layer, respectively. We describe the parameters of these layers. We use 1×1 kernels in "HypNet" as the bottleneck of the Inception Net [12].

		$\mathcal{N}_F$	$\mathcal{N}_G$	Maxout	CGNet	NLNet	HypNet
conv	kernel	7×7	7×7	3×3	3×3	3×3	3×3 1×1
tconv	kernel stride				3×3 2×2		
Max-Pool	kernel stride			$4 \times 1 \times 1$ $4 \times 1 \times 1$	3×3 2×2		
Avg-Pool	kernel					3×3	
ReLU					✓	✓	✓
SoftPlus	β threshold					1 20	
Sigmoid							✓

3 Robustness to Variant Kernel Sizes

We evaluate our method on SET5 using 100 kernels with sizes from 13×13 pixels to 35×35 pixels. Fig. 1 shows that our method performs better than the evaluated methods even when the blur kernel size is large. The non-E2E methods DRUNet and IRCNN and the E2E methods DWDN are compared with ours.

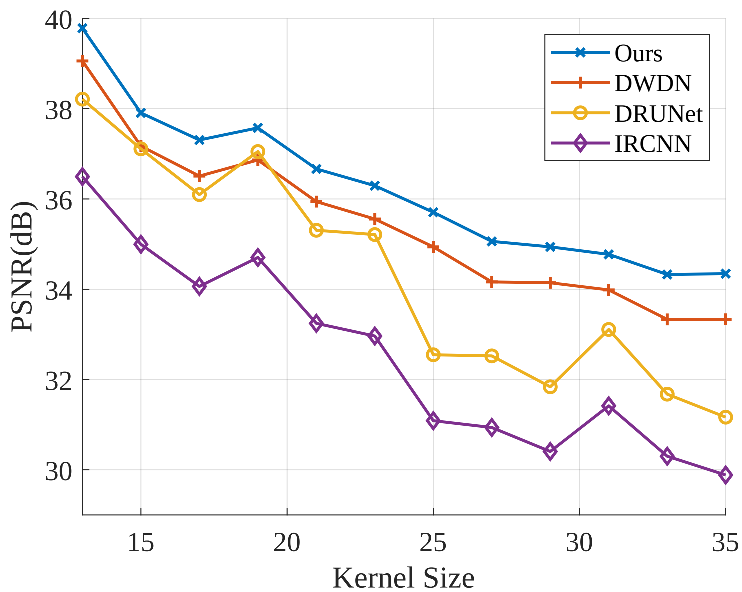
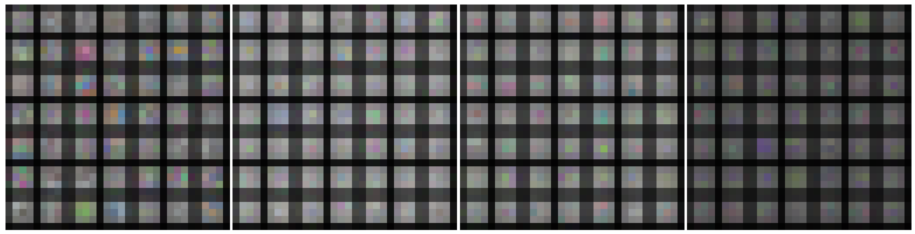


Fig. 1: PSNR versus kernel size.

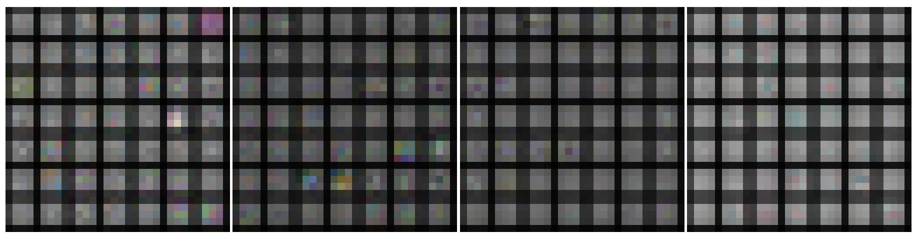
4 More Ablation Studies

We show some intermediate results in Fig. 2, 3 and 4 based on the full-weighted model to better understand what the proposed network learns. In Fig. 2, we show the learned **F** and **G**. We use the BABY image from SET5 [1] as an example and compute the feature maps. Fig. 3 and Fig. 4 show the regularization-related and data-related feature maps, respectively.

In Table 2, we show the complete ablation study in PSNR and SSIM on the datasets LEVIN [6], BSD100 [7] and SET5 [1] based on the heavy-weighted model consistently with Section 5.5 of the manuscript. The manuscript states that we include the ablation studies w.r.t. HypNet and NLNet in the supplemental material. In addition, we replace the Maxout layers with soft-shrinkage functions (e.g., ℓ_1-norm in Table 2 and Figure 5) and compare them to cascade



(a) **F** of the first stage (b) **F** of the second stage (c) **F** of the third stage (d) **F** of the latest stage



(e) **G** of the first stage (f) **G** of the second stage (g) **G** of the third stage (h) **G** of the latest stage

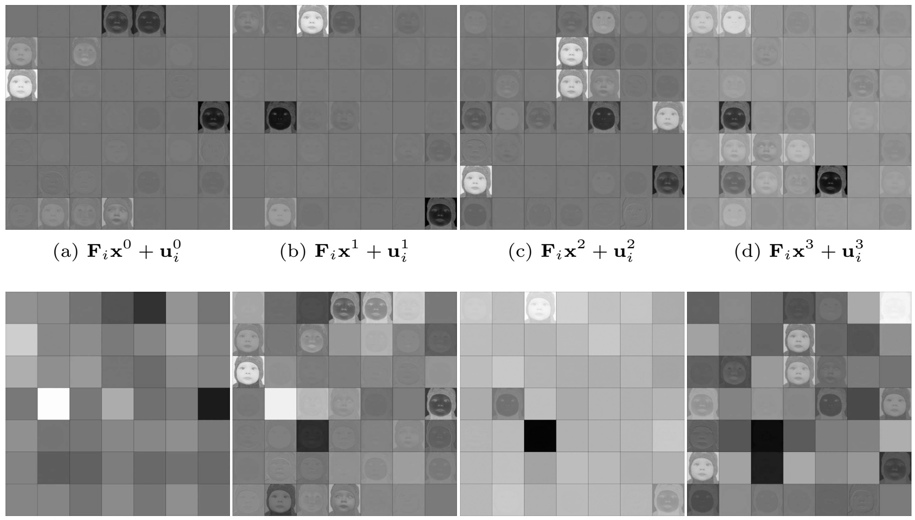
Fig. 2: Learned filters of $\mathcal{N}_F$ and $\mathcal{N}_G$.

shrinkage fields (CSF) [11] to demonstrate the effectiveness of Maxout layers against conventional shrinkage functions.

Table 2: Average PSNR(dB)/SSIM of the deblurring results with Gaussian noise using different methods. We highlight the **best** results. "CSF" denotes the result of [11], "w/o $\mathbf{F}, \mathbf{G}$", "ReLU", "RBF", "CG", "CG†", "FFT" and "FFT†" are the extended results of Table 5 of the manuscript; "w/o HypNet" and "w/o NLNet" are results of the models without these parts, respectively; "ℓ_1-norm" replaces the Maxout layers with soft-shrinkage functions, i.e., the R_i's and R_j's in Equation (8) of the manuscript are all in the form of ℓ_1-norm regularization.

Dataset	noise	CSF [11] PSNR / SSIM	w/o **F**,**G** PSNR / SSIM	ReLU PSNR / SSIM	RBF PSNR / SSIM	CG PSNR / SSIM	CG† PSNR / SSIM	FFT PSNR / SSIM	FFT† PSNR / SSIM	ℓ_1-norm PSNR / SSIM	w/o HypNet PSNR / SSIM	w/o NLNet PSNR / SSIM	DSDNet(Heavy) PSNR / SSIM
Levin [6]	1%	30.22 / 0.888	27.70 / 0.742	35.85 / 0.961	35.89 / 0.961	31.24 / 0.905	34.97 / 0.954	32.45 / 0.936	33.02 / 0.944	35.88 / 0.961	35.85 / 0.960	35.98 / 0.962	**36.07 / 0.962**
	3%	27.37 / 0.792	20.07 / 0.382	32.53 / 0.922	32.54 / 0.920	29.15 / 0.830	31.82 / 0.911	29.86 / 0.881	30.72 / 0.900	32.43 / 0.919	32.36 / 0.918	32.54 / 0.921	**32.59 / 0.922**
	5%	25.99 / 0.737	16.10 / 0.229	30.67 / 0.889	30.69 / 0.888	26.54 / 0.708	30.01 / 0.874	28.14 / 0.832	29.25 / 0.864	30.59 / 0.887	30.47 / 0.884	30.70 / 0.889	**30.71 / 0.889**
BSD100 [7]	1%	28.46 / 0.807	27.16 / 0.738	31.28 / 0.887	31.76 / 0.894	28.08 / 0.795	31.43 / 0.886	30.27 / 0.869	30.89 / 0.883	31.78 / 0.895	31.72 / 0.894	31.77 / 0.895	**31.83 / 0.896**
	3%	26.25 / 0.702	20.72 / 0.412	28.32 / 0.788	28.95 / 0.815	27.02 / 0.734	28.70 / 0.807	27.84 / 0.774	28.41 / 0.803	28.92 / 0.815	28.87 / 0.813	28.93 / 0.815	**28.97 / 0.816**
	5%	25.22 / 0.648	17.19 / 0.275	27.03 / 0.737	27.85 / 0.777	25.60 / 0.649	27.62 / 0.769	26.58 / 0.712	27.44 / 0.767	27.83 / 0.777	27.78 / 0.776	27.85 / 0.778	**27.87 / 0.779**
Set5 [1]	1%	29.75 / 0.842	26.66 / 0.683	32.78 / 0.898	32.98 / 0.901	29.07 / 0.834	32.39 / 0.893	31.30 / 0.876	32.03 / 0.892	33.02 / 0.900	32.97 / 0.899	33.03 / 0.901	**33.11 / 0.902**
	3%	26.94 / 0.752	19.40 / 0.354	30.05 / 0.844	30.14 / 0.846	27.67 / 0.761	29.51 / 0.834	28.45 / 0.800	29.32 / 0.832	30.07 / 0.844	29.95 / 0.842	30.10 / 0.844	**30.16 / 0.847**
	5%	25.62 / 0.701	15.76 / 0.233	28.60 / 0.806	28.65 / 0.808	27.64 / 0.720	28.00 / 0.793	26.93 / 0.752	27.82 / 0.787	28.62 / 0.806	28.47 / 0.803	28.65 / 0.806	**28.65 / 0.808**

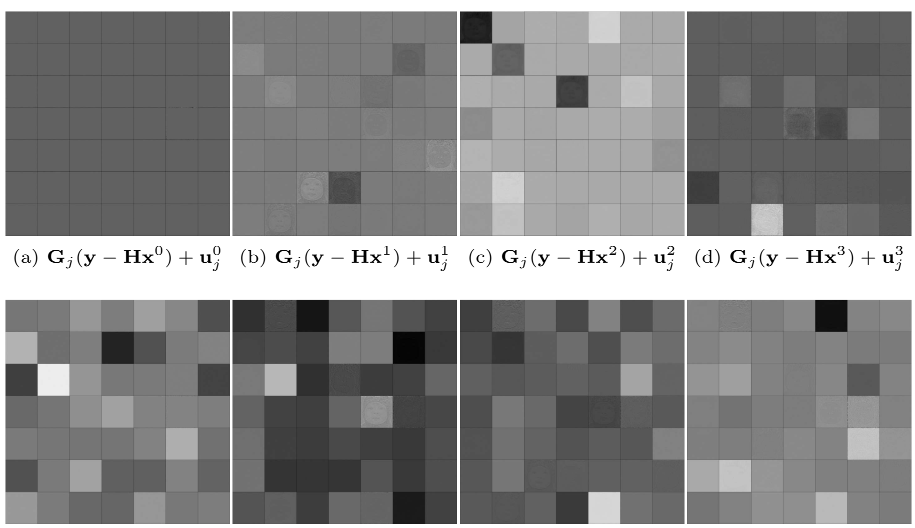
Fig. 5 shows the visual comparisons of the baseline methods, where this work generates a clearer image that is visually close to the ground truth.



(a) $\mathbf{F}_i\mathbf{x}^0 + \mathbf{u}_i^0$ (b) $\mathbf{F}_i\mathbf{x}^1 + \mathbf{u}_i^1$ (c) $\mathbf{F}_i\mathbf{x}^2 + \mathbf{u}_i^2$ (d) $\mathbf{F}_i\mathbf{x}^3 + \mathbf{u}_i^3$

(e) $\mathbf{v}_i^1$ (f) $\mathbf{v}_i^2$ (g) $\mathbf{v}_i^3$ (h) $\mathbf{v}_i^4$

Fig. 3: The regularization-related feature maps before (top row) and after (bottom row) the Maxout layers. As regularization-related terms are learned to keep the structure of images, we can observe the strong edges in the figure. The Maxout layers enhance the edges.



(a) $\mathbf{G}_j(\mathbf{y} - \mathbf{H}\mathbf{x}^0) + \mathbf{u}_j^0$ (b) $\mathbf{G}_j(\mathbf{y} - \mathbf{H}\mathbf{x}^1) + \mathbf{u}_j^1$ (c) $\mathbf{G}_j(\mathbf{y} - \mathbf{H}\mathbf{x}^2) + \mathbf{u}_j^2$ (d) $\mathbf{G}_j(\mathbf{y} - \mathbf{H}\mathbf{x}^3) + \mathbf{u}_j^3$

(e) $\mathbf{z}_j^1$ (f) $\mathbf{z}_j^2$ (g) $\mathbf{z}_j^3$ (h) $\mathbf{z}_j^4$

Fig. 4: Data-related feature maps before (top row) and after (bottom row) the Maxout layers. As data-related terms are learned to eliminate the reconstruction error and noise, we can observe the sparsity in the figure. The Maxout layers enhance the sparsity.

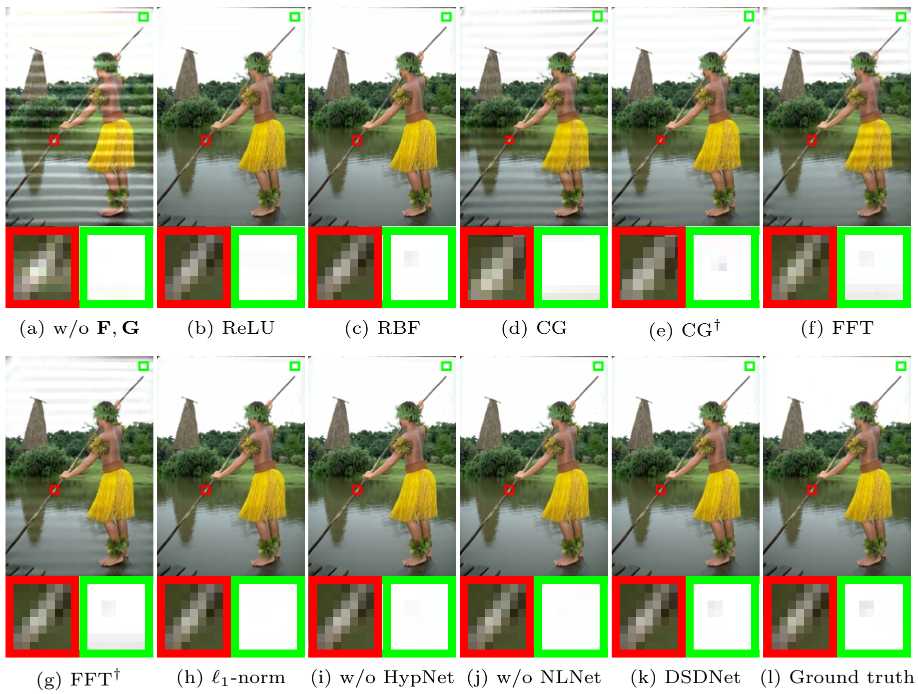


Fig. 5: Qualitative ablation study on the BSD100 dataset [7] with 1% Gaussian noise. We note that the method without using the learned filters generates significant artifacts, as shown in (a). The method using RBFs generates over-smoothed results, as shown in (c). The ReLU does not discriminatively keep the most valuable features, so it does not restore the details well (see (b)). The results of original CG in (d) and (f) still contain ringing artifacts. As mentioned in the manuscript, FFT-based methods (f) and (g) generate ring artifacts even with edge taper, where (g) is more severe than (e). The models without HypNet and NLNet do not effectively restore the patterns as shown in the green boxes in (i) and (j). (h) shows the result of soft-shrinkage functions which do not preserve details well. (b), (c) and (h) demonstrate the effectiveness of Maxout layers. The proposed method (k) generates a better result that is visually close to the ground truth.

5 More Qualitative Evaluation

In this section, we present more visual comparisons of the proposed method and state-of-the-art ones.

(a) Blurry input (b) IRCNN [14] (c) ADM_UDM [4] (d) KerUNC [8]

(e) VEM [9] (f) DWDN [2] (g) SVMAP [3] (h) DRUNet [13]

(i) DSDNet (Light) (j) DSDNet (Heavy) (k) DSDNet (Full) (l) Ground truth

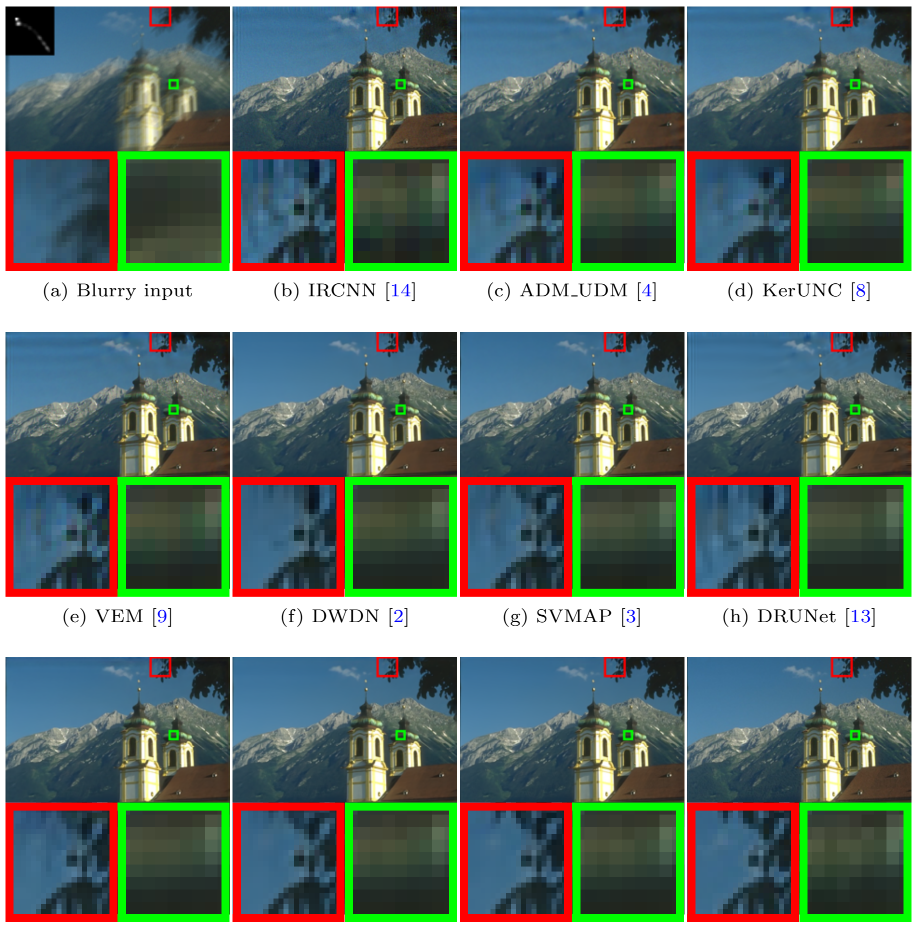
Fig. 6: Qualitative evaluation of 1% Gaussian noise on BSD100 [7]. The result by the IRCNN [14] contains some artifacts. The other evaluated methods do not effectively restore the structural details. In contrast, our methods generate clearer images.

(a) Blurry input (b) IRCNN [14] (c) ADM_UDM [4] (d) KerUNC [8]

(e) VEM [9] (f) DWDN [2] (g) SVMAP [3] (h) DRUNet [13]

(i) DSDNet (Light) (j) DSDNet (Heavy) (k) DSDNet (Full) (l) Ground truth

Fig. 7: Qualitative evaluation of 1% Gaussian noise on BSD100 [7]. Our models generate sharper texture both in the red and green boxes than other methods.

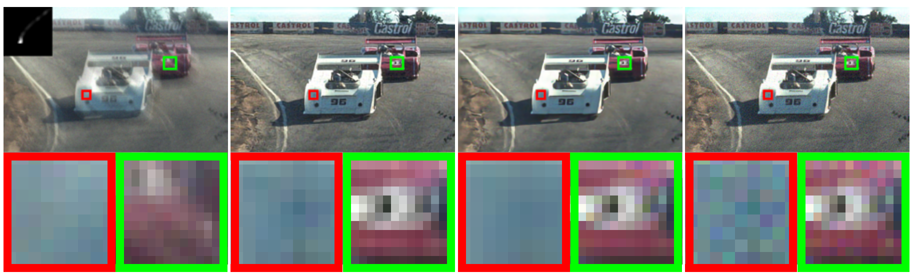


(a) Blurry input (b) IRCNN [14] (c) ADM_UDM [4] (d) KerUNC [8]

(e) VEM [9] (f) DWDN [2] (g) SVMAP [3] (h) DRUNet [13]

(i) DSDNet (Light) (j) DSDNet (Heavy) (k) DSDNet (Full) (l) Ground truth

Fig. 8: Qualitative evaluation of 1% Gaussian noise on BSD100 [7]. Only the red boxes of our (j) and (k) are out of artifacts. Besides, only the green boxes of these two models keep the sharp details.

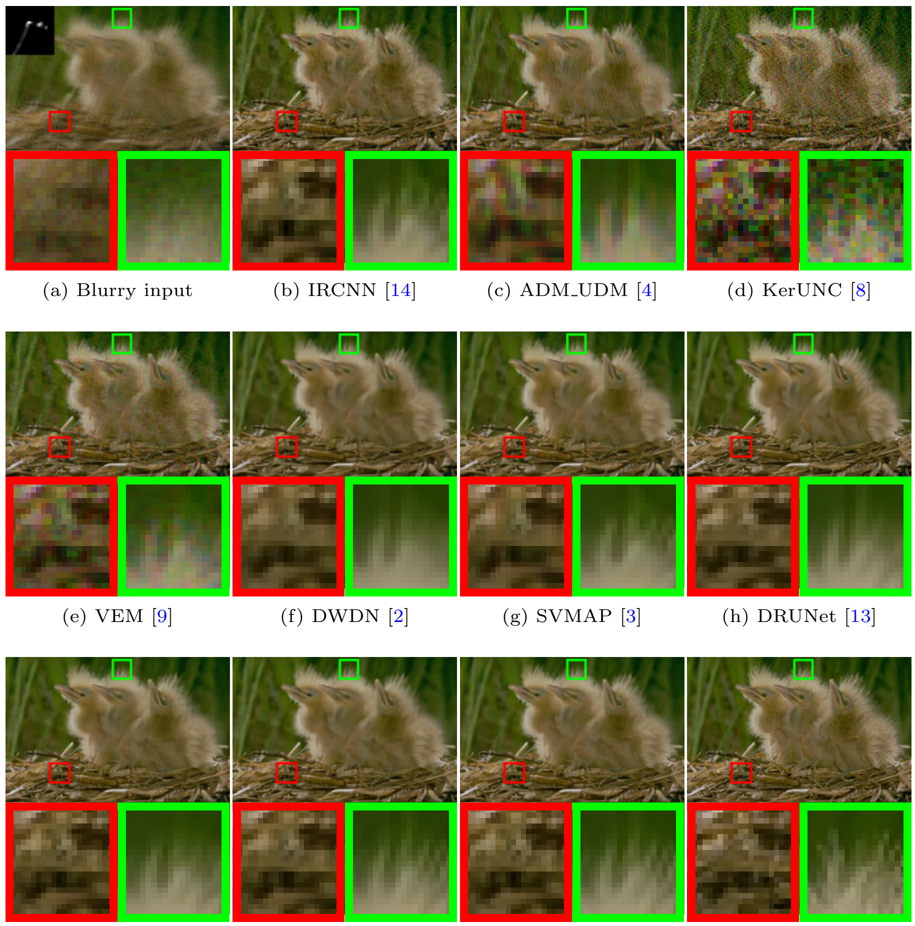


(a) Blurry input (b) IRCNN [14] (c) ADM_UDM [4] (d) KerUNC [8]

(e) VEM [9] (f) DWDN [2] (g) SVMAP [3] (h) DRUNet [13]

(i) DSDNet (Light) (j) DSDNet (Heavy) (k) DSDNet (Full) (l) Ground truth

Fig. 9: Qualitative evaluation of 3% Gaussian noise on BSD100 [7]. The contents in (k) are clearer than other methods.

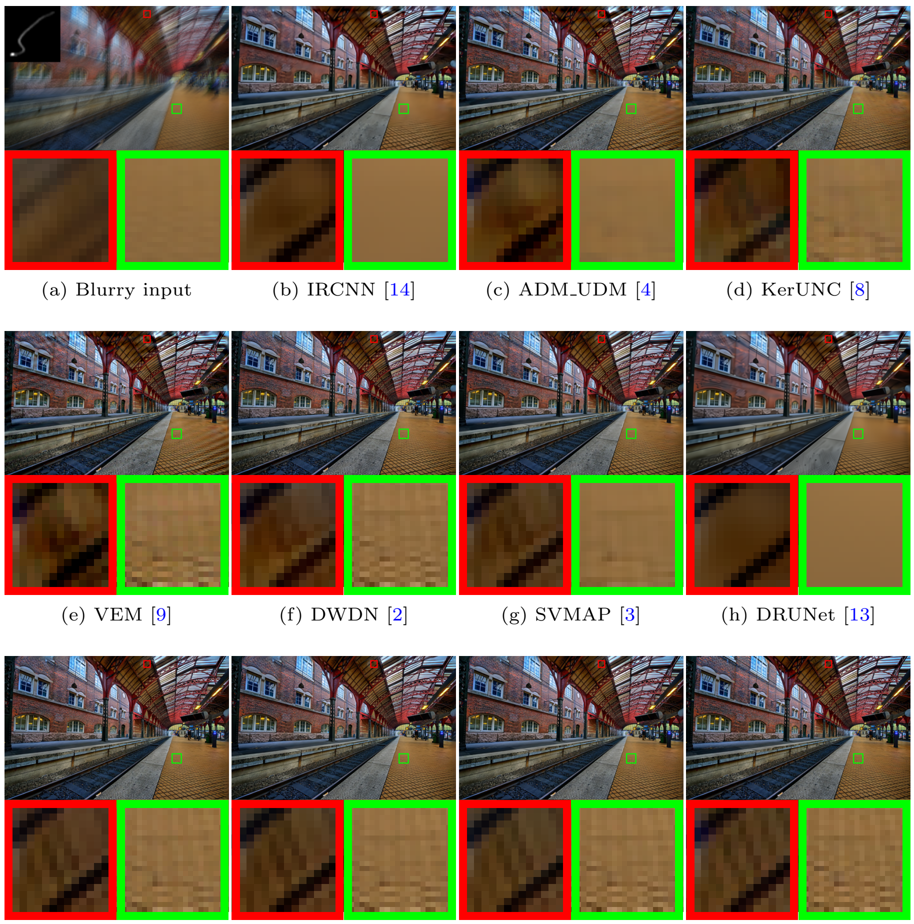


(a) Blurry input (b) IRCNN [14] (c) ADM_UDM [4] (d) KerUNC [8]

(e) VEM [9] (f) DWDN [2] (g) SVMAP [3] (h) DRUNet [13]

(i) DSDNet (Light) (j) DSDNet (Heavy) (k) DSDNet (Full) (l) Ground truth

Fig. 10: Qualitative evaluation of 5% Gaussian noise on BSD100 [7]. The texture in (j) and (k) are sharper than other methods, while that of our light-weighted model (i) is very close to state-of-the-art methods (g) and (h).

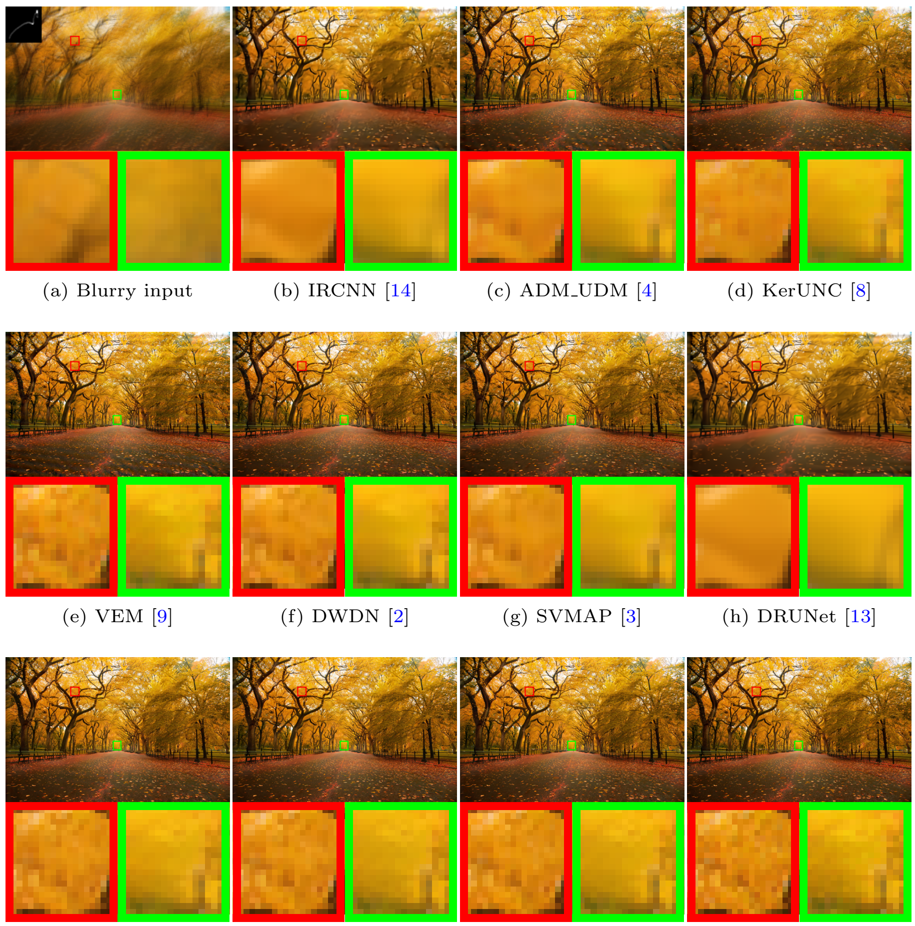


(a) Blurry input (b) IRCNN [14] (c) ADM_UDM [4] (d) KerUNC [8]

(e) VEM [9] (f) DWDN [2] (g) SVMAP [3] (h) DRUNet [13]

(i) DSDNet (Light) (j) DSDNet (Heavy) (k) DSDNet (Full) (l) Ground truth

Fig. 11: Qualitative evaluation on Manmade of Lai [5]. Only DWDN [2] (f) and our methods reconstruct the texture in green boxes, while the red box of (f) contains artifacts as other methods.



(a) Blurry input (b) IRCNN [14] (c) ADM_UDM [4] (d) KerUNC [8]

(e) VEM [9] (f) DWDN [2] (g) SVMAP [3] (h) DRUNet [13]

(i) DSDNet (Light) (j) DSDNet (Heavy) (k) DSDNet (Full) (l) Ground truth

Fig. 12: Qualitative evaluation on Natural of Lai [5]. The evaluated methods do not restore the details well, as shown in (b)-(h). In contrast, our method restores much clearer images with finer details.

(a) Blurry input (b) IRCNN [14] (c) ADM_UDM [4]

(d) KerUNC [8] (e) VEM [9] (f) DWDN [2]

(g) SVMAP [3] (h) DRUNet [13] (i) DSDNet (Full)

Fig. 13: Qualitative evaluation on a real case uses the kernel estimated by [10]. The evaluated methods do not restore the details well, as shown in (b)-(h). In contrast, our method restore much clearer images with finer details.

(a) Blurry input (b) IRCNN [14] (c) ADM_UDM [4]

(d) KerUNC [8] (e) VEM [9] (f) DWDN [2]

(g) SVMAP [3] (h) DRUNet [13] (i) DSDNet (Full)

Fig. 14: Qualitative evaluation on a real case uses the kernel estimated by [10]. The evaluated methods mix the white texts with surrounding colors in (b) - (h). In contrast, our method restores much sharper texts with less mixed color.

6 Limitation

In Section 5.6 of the manuscript, we have pointed out that our method is less effective when the blurry images contain significant saturation. Because the linear convolution model used in the degradation process does not hold for saturation, methods based on such a degradation model will generate the results with significant artifacts, as shown in Fig. 15(b).

As the pixels in the saturated regions usually have higher intensity values, clipping these pixels and ignoring them in the deblurring process would help the performance improvement (Fig. 15(c)). Since deblurring images with saturated regions is an important task, future work will consider jointly handling saturated areas and image deblurring in a principled way.

(a) Blurry input (b) Overflow without clipping (c) Clipped

Fig. 15: A failure example. Images containing dark and saturated pixels may cause overflow, as shown in (b). Clipping these pixels and ignoring them in the deblurring process generates a good result, as shown in (c).

References

1. Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.L.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In: BMVC (2012) 2, 3
2. Dong, J., Roth, S., Schiele, B.: Deep wiener deconvolution: Wiener meets deep learning for image deblurring. In: NeurIPS (2020) 6, 7, 8, 9, 10, 11, 12, 13, 14
3. Dong, J., Roth, S., Schiele, B.: Learning spatially-variant map models for non-blind image deblurring. In: CVPR. pp. 4886–4895 (2021) 6, 7, 8, 9, 10, 11, 12, 13, 14
4. Ko, H.C., Chang, J.Y., Ding, J.J.: Deep priors inside an unrolled and adaptive deconvolution model. In: ACCV (2020) 6, 7, 8, 9, 10, 11, 12, 13, 14
5. Lai, W.S., Huang, J.B., Hu, Z., Ahuja, N., Yang, M.H.: A comparative study for single image blind deblurring. In: CVPR. pp. 1701–1709 (2016) 11, 12
6. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Understanding and evaluating blind deconvolution algorithms. In: CVPR. pp. 1964–1971 (2009) 2, 3
7. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: ICCV. pp. 416–423 (2001) 2, 3, 5, 6, 7, 8, 9, 10
8. Nan, Y., Ji, H.: Deep learning for handling kernel/model uncertainty in image deconvolution. In: CVPR. pp. 2388–2397 (2020) 6, 7, 8, 9, 10, 11, 12, 13, 14
9. Nan, Y., Quan, Y., Ji, H.: Variational-em-based deep learning for noise-blind image deblurring. In: CVPR. pp. 3626–3635 (2020) 6, 7, 8, 9, 10, 11, 12, 13, 14
10. Pan, J., Sun, D., Pfister, H., Yang, M.H.: Blind image deblurring using dark channel prior. In: CVPR. pp. 1628–1636 (2016) 13, 14
11. Schmidt, U., Roth, S.: Shrinkage fields for effective image restoration. In: CVPR. pp. 2774–2781 (2014) 3
12. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015) 1
13. Zhang, K., Li, Y., Zuo, W., Zhang, L., Van Gool, L., Timofte, R.: Plug-and-play image restoration with deep denoiser prior. IEEE TPAMI (2021) 6, 7, 8, 9, 10, 11, 12, 13, 14
14. Zhang, K., Zuo, W., Gu, S., Zhang, L.: Learning deep cnn denoiser prior for image restoration. In: CVPR. pp. 3929–3938 (2017) 6, 7, 8, 9, 10, 11, 12, 13, 14