Supplementary File to "Pixel-level Non-local Image Smoothing with Objective Evaluation"

Zhi-Ang Liu[1], Ying-Kun Hou[2], Xian-Tong Zhen[3,4], Jun Xu[1,*], Ling Shao[3,4], Ming-Ming Cheng[1]
[1]TKLNDST, College of Computer Science, Nankai University, Tianjin, China
[2]School of Information Science and Technology, Taishan University, Tai'an, China
[3]Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, UAE
[4] Mohamed bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi, UAE

In this supplementary file, we provide:

- detailed Haar transformation and inverse Haar transformation;
- more comparisons of different image smoothing methods on the datasets of NKS, [1], [6], [9].

I. Detailed Haar Transformation and Inverse Haar Transformation

We perform standard Haar transformation and inverse Haar transformation with no modification. Moreover we set $q = 4$, $m = 16$ in all experiments, so the similar pixels matrix $\boldsymbol{S} \in \mathbb{R}^{4\times 16}$ could be represented by columns as $\boldsymbol{S} = [\boldsymbol{s}_1^4, ..., \boldsymbol{s}_{16}^4] \in \mathbb{R}^{4\times 16}$. The Haar transformation includes horizontal and vertical transformation. We first apply the horizontal transformation. Specifically, we multiply the similar pixels matrix $\boldsymbol{S} \in \mathbb{R}^{4\times 16}$ and the horizontal transformation matrix $\boldsymbol{H}_r \in \mathbb{R}^{16\times 16}$:

$$
\begin{aligned}
\boldsymbol{t}_i^4 &= \frac{1}{\sqrt{16}}(\sum_{j=1}^{8}\boldsymbol{s}_j^4 + (-1)^{i-1}\sum_{j=9}^{16}\boldsymbol{s}_j^4), \text{ when } i = 1, 2;\\
\boldsymbol{t}_i^4 &= \frac{1}{\sqrt{8}}(\sum_{j=8(i-3)+1}^{8(i-3)+4}\boldsymbol{s}_j^4 - \sum_{j=8(i-3)+5}^{8(i-2)}\boldsymbol{s}_j^4), \text{ when } i = 3, 4;\\
\boldsymbol{t}_i^4 &= \frac{1}{\sqrt{4}}(\sum_{j=4(i-5)+1}^{4(i-5)+2}\boldsymbol{s}_j^4 - \sum_{j=4(i-5)+3}^{4(i-5)+4}\boldsymbol{s}_j^4), \text{ when } i = 5, ..., 8;\\
\boldsymbol{t}_i^4 &= \frac{1}{\sqrt{2}}(\boldsymbol{s}_{2(i-9)+1}^4 - \boldsymbol{s}_{2(i-9)+2}^4), \text{ when } i = 9, ..., 16.
\end{aligned} \tag{1}
$$

We stack the all these column vectors to form $\boldsymbol{T} = [\boldsymbol{t}_1^4, ..., \boldsymbol{t}_{16}^4] \in \mathbb{R}^{4\times 16}$. We then represent $\boldsymbol{T}$ by rows as $\boldsymbol{T}^4 = [\boldsymbol{t}^{1\top}, ..., \boldsymbol{t}^{4\top}]^\top \in \mathbb{R}^{4\times 16}$, and perform vertical Haar transformation. Specifically, we multiply the matrix $\boldsymbol{T}^4 \in \mathbb{R}^{4\times 16}$ and the vertical transformation matrix $\boldsymbol{H}_l \in \mathbb{R}^{4\times 4}$:

$$
\begin{aligned}
\hat{\boldsymbol{t}}^1 = \frac{1}{\sqrt{4}}\sum_{i=1}^{4}\boldsymbol{t}^i,\ \hat{\boldsymbol{t}}^2 = \frac{1}{\sqrt{4}}(\sum_{i=1}^{2}\boldsymbol{t}^i - \sum_{i=3}^{4}\boldsymbol{t}^i),\\
\hat{\boldsymbol{t}}^3 = \frac{1}{\sqrt{2}}(\boldsymbol{t}^1 - \boldsymbol{t}^2),\ \hat{\boldsymbol{t}}^4 = \frac{1}{\sqrt{2}}(\boldsymbol{t}^3 - \boldsymbol{t}^4).
\end{aligned} \tag{2}
$$

After the thresholding step, we could get the thresholded representation matrix $\widetilde{\boldsymbol{T}} \in \mathbb{R}^{4\times 16}$. We next perform inverse vertical Haar transformation and inverse horizontal Haar transformation. We first apply the inverse vertical transformation. Specifically, we multiply the inverse vertical transformation matrix $\boldsymbol{H}_{il} \in \mathbb{R}^{4\times 4}$ and the thresholded representation matrix $\widetilde{\boldsymbol{T}}^4 \in \mathbb{R}^{4\times 16}$:

$$
\begin{aligned}
\widetilde{\boldsymbol{t}}^1 &= \frac{1}{\sqrt{4}}(\hat{\boldsymbol{t}}^1 + \hat{\boldsymbol{t}}^2) + \frac{1}{\sqrt{2}}\hat{\boldsymbol{t}}^3,\\
\widetilde{\boldsymbol{t}}^2 &= \frac{1}{\sqrt{4}}(\hat{\boldsymbol{t}}^1 + \hat{\boldsymbol{t}}^2) - \frac{1}{\sqrt{2}}\hat{\boldsymbol{t}}^3,\\
\widetilde{\boldsymbol{t}}^3 &= \frac{1}{\sqrt{4}}(\hat{\boldsymbol{t}}^1 - \hat{\boldsymbol{t}}^2) + \frac{1}{\sqrt{2}}\hat{\boldsymbol{t}}^4,\\
\widetilde{\boldsymbol{t}}^4 &= \frac{1}{\sqrt{4}}(\hat{\boldsymbol{t}}^1 - \hat{\boldsymbol{t}}^2) - \frac{1}{\sqrt{2}}\hat{\boldsymbol{t}}^4.
\end{aligned} \tag{3}
$$

*Corresponding author is Jun Xu (email: nankaimathxujun@gmail.com).

We stack all these row vectors to form $\widetilde{\boldsymbol{T}}^4 = [(\widetilde{\boldsymbol{t}}^1)^\top, ..., (\widetilde{\boldsymbol{t}}^4)^\top]^\top \in \mathbb{R}^{4\times 16}$. We then represent $\widetilde{\boldsymbol{T}}$ by columns as $\widetilde{\boldsymbol{T}} = [\widetilde{\boldsymbol{t}}_1^4, ..., \widetilde{\boldsymbol{t}}_{16}^4] \in \mathbb{R}^{4\times 16}$, and perform inverse horizontal Haar transformation. Specifically, we multiply the matrix $\widetilde{\boldsymbol{T}} \in \mathbb{R}^{4\times 16}$ and the inverse horizontal transformation matrix $\boldsymbol{H}_{ir} \in \mathbb{R}^{16\times 16}$:

$$
\begin{aligned}
\widetilde{\boldsymbol{s}}_1^4 &= \frac{1}{\sqrt{16}}(\widetilde{\boldsymbol{t}}_1^4 + \widetilde{\boldsymbol{t}}_2^4) + \frac{1}{\sqrt{8}}\widetilde{\boldsymbol{t}}_3^4 + \frac{1}{\sqrt{4}}\widetilde{\boldsymbol{t}}_5^4 + \frac{1}{\sqrt{2}}\widetilde{\boldsymbol{t}}_9^4, \\
\widetilde{\boldsymbol{s}}_2^4 &= \frac{1}{\sqrt{16}}(\widetilde{\boldsymbol{t}}_1^4 + \widetilde{\boldsymbol{t}}_2^4) + \frac{1}{\sqrt{8}}\widetilde{\boldsymbol{t}}_3^4 + \frac{1}{\sqrt{4}}\widetilde{\boldsymbol{t}}_5^4 - \frac{1}{\sqrt{2}}\widetilde{\boldsymbol{t}}_9^4, \\
\widetilde{\boldsymbol{s}}_3^4 &= \frac{1}{\sqrt{16}}(\widetilde{\boldsymbol{t}}_1^4 + \widetilde{\boldsymbol{t}}_2^4) + \frac{1}{\sqrt{8}}\widetilde{\boldsymbol{t}}_3^4 - \frac{1}{\sqrt{4}}\widetilde{\boldsymbol{t}}_5^4 + \frac{1}{\sqrt{2}}\widetilde{\boldsymbol{t}}_{10}^4, \\
\widetilde{\boldsymbol{s}}_4^4 &= \frac{1}{\sqrt{16}}(\widetilde{\boldsymbol{t}}_1^4 + \widetilde{\boldsymbol{t}}_2^4) + \frac{1}{\sqrt{8}}\widetilde{\boldsymbol{t}}_3^4 - \frac{1}{\sqrt{4}}\widetilde{\boldsymbol{t}}_5^4 - \frac{1}{\sqrt{2}}\widetilde{\boldsymbol{t}}_{10}^4, \\
\widetilde{\boldsymbol{s}}_5^4 &= \frac{1}{\sqrt{16}}(\widetilde{\boldsymbol{t}}_1^4 + \widetilde{\boldsymbol{t}}_2^4) - \frac{1}{\sqrt{8}}\widetilde{\boldsymbol{t}}_3^4 + \frac{1}{\sqrt{4}}\widetilde{\boldsymbol{t}}_6^4 + \frac{1}{\sqrt{2}}\widetilde{\boldsymbol{t}}_{11}^4, \\
\widetilde{\boldsymbol{s}}_6^4 &= \frac{1}{\sqrt{16}}(\widetilde{\boldsymbol{t}}_1^4 + \widetilde{\boldsymbol{t}}_2^4) - \frac{1}{\sqrt{8}}\widetilde{\boldsymbol{t}}_3^4 + \frac{1}{\sqrt{4}}\widetilde{\boldsymbol{t}}_6^4 - \frac{1}{\sqrt{2}}\widetilde{\boldsymbol{t}}_{11}^4, \\
\widetilde{\boldsymbol{s}}_7^4 &= \frac{1}{\sqrt{16}}(\widetilde{\boldsymbol{t}}_1^4 + \widetilde{\boldsymbol{t}}_2^4) - \frac{1}{\sqrt{8}}\widetilde{\boldsymbol{t}}_3^4 - \frac{1}{\sqrt{4}}\widetilde{\boldsymbol{t}}_6^4 + \frac{1}{\sqrt{2}}\widetilde{\boldsymbol{t}}_{12}^4, \\
\widetilde{\boldsymbol{s}}_8^4 &= \frac{1}{\sqrt{16}}(\widetilde{\boldsymbol{t}}_1^4 + \widetilde{\boldsymbol{t}}_2^4) - \frac{1}{\sqrt{8}}\widetilde{\boldsymbol{t}}_3^4 - \frac{1}{\sqrt{4}}\widetilde{\boldsymbol{t}}_6^4 - \frac{1}{\sqrt{2}}\widetilde{\boldsymbol{t}}_{12}^4, \\
\widetilde{\boldsymbol{s}}_9^4 &= \frac{1}{\sqrt{16}}(\widetilde{\boldsymbol{t}}_1^4 - \widetilde{\boldsymbol{t}}_2^4) + \frac{1}{\sqrt{8}}\widetilde{\boldsymbol{t}}_4^4 + \frac{1}{\sqrt{4}}\widetilde{\boldsymbol{t}}_7^4 + \frac{1}{\sqrt{2}}\widetilde{\boldsymbol{t}}_{13}^4, \\
\widetilde{\boldsymbol{s}}_{10}^4 &= \frac{1}{\sqrt{16}}(\widetilde{\boldsymbol{t}}_1^4 - \widetilde{\boldsymbol{t}}_2^4) + \frac{1}{\sqrt{8}}\widetilde{\boldsymbol{t}}_4^4 + \frac{1}{\sqrt{4}}\widetilde{\boldsymbol{t}}_7^4 - \frac{1}{\sqrt{2}}\widetilde{\boldsymbol{t}}_{13}^4, \\
\widetilde{\boldsymbol{s}}_{11}^4 &= \frac{1}{\sqrt{16}}(\widetilde{\boldsymbol{t}}_1^4 - \widetilde{\boldsymbol{t}}_2^4) + \frac{1}{\sqrt{8}}\widetilde{\boldsymbol{t}}_4^4 - \frac{1}{\sqrt{4}}\widetilde{\boldsymbol{t}}_7^4 + \frac{1}{\sqrt{2}}\widetilde{\boldsymbol{t}}_{14}^4, \\
\widetilde{\boldsymbol{s}}_{12}^4 &= \frac{1}{\sqrt{16}}(\widetilde{\boldsymbol{t}}_1^4 - \widetilde{\boldsymbol{t}}_2^4) + \frac{1}{\sqrt{8}}\widetilde{\boldsymbol{t}}_4^4 - \frac{1}{\sqrt{4}}\widetilde{\boldsymbol{t}}_7^4 - \frac{1}{\sqrt{2}}\widetilde{\boldsymbol{t}}_{14}^4, \\
\widetilde{\boldsymbol{s}}_{13}^4 &= \frac{1}{\sqrt{16}}(\widetilde{\boldsymbol{t}}_1^4 - \widetilde{\boldsymbol{t}}_2^4) - \frac{1}{\sqrt{8}}\widetilde{\boldsymbol{t}}_4^4 + \frac{1}{\sqrt{4}}\widetilde{\boldsymbol{t}}_8^4 + \frac{1}{\sqrt{2}}\widetilde{\boldsymbol{t}}_{15}^4, \\
\widetilde{\boldsymbol{s}}_{14}^4 &= \frac{1}{\sqrt{16}}(\widetilde{\boldsymbol{t}}_1^4 - \widetilde{\boldsymbol{t}}_2^4) - \frac{1}{\sqrt{8}}\widetilde{\boldsymbol{t}}_4^4 + \frac{1}{\sqrt{4}}\widetilde{\boldsymbol{t}}_8^4 - \frac{1}{\sqrt{2}}\widetilde{\boldsymbol{t}}_{15}^4, \\
\widetilde{\boldsymbol{s}}_{15}^4 &= \frac{1}{\sqrt{16}}(\widetilde{\boldsymbol{t}}_1^4 - \widetilde{\boldsymbol{t}}_2^4) - \frac{1}{\sqrt{8}}\widetilde{\boldsymbol{t}}_4^4 - \frac{1}{\sqrt{4}}\widetilde{\boldsymbol{t}}_8^4 + \frac{1}{\sqrt{2}}\widetilde{\boldsymbol{t}}_{16}^4, \\
\widetilde{\boldsymbol{s}}_{16}^4 &= \frac{1}{\sqrt{16}}(\widetilde{\boldsymbol{t}}_1^4 - \widetilde{\boldsymbol{t}}_2^4) - \frac{1}{\sqrt{8}}\widetilde{\boldsymbol{t}}_4^4 - \frac{1}{\sqrt{4}}\widetilde{\boldsymbol{t}}_8^4 - \frac{1}{\sqrt{2}}\widetilde{\boldsymbol{t}}_{16}^4.
\end{aligned}
\tag{17}
$$

We stack all these column vectors together and form the smoothed similar pixel matrix $\widetilde{\boldsymbol{S}} = [\widetilde{\boldsymbol{s}}_1^4, ..., \widetilde{\boldsymbol{s}}_{16}^4] \in \mathbb{R}^{4\times 16}$.

II. More Comparisons of Different Image Smoothing Methods

Here, we conduct more comparisons of different image smoothing methods on the datasets of NKS, [1], [6], [9]. In Figures 1-5, we compare PSNR, SSIM [4], FSIM [7], and visual quality of different methods on image smoothing on the dataset of NKS. In Figures 6-26, we compare the visual quality of different methods on image smoothing on the datasets of [1], [6], [9]. The comparison rsults demonstrate that the PNLS method achieves better visual quality than the other image smoothing methods.

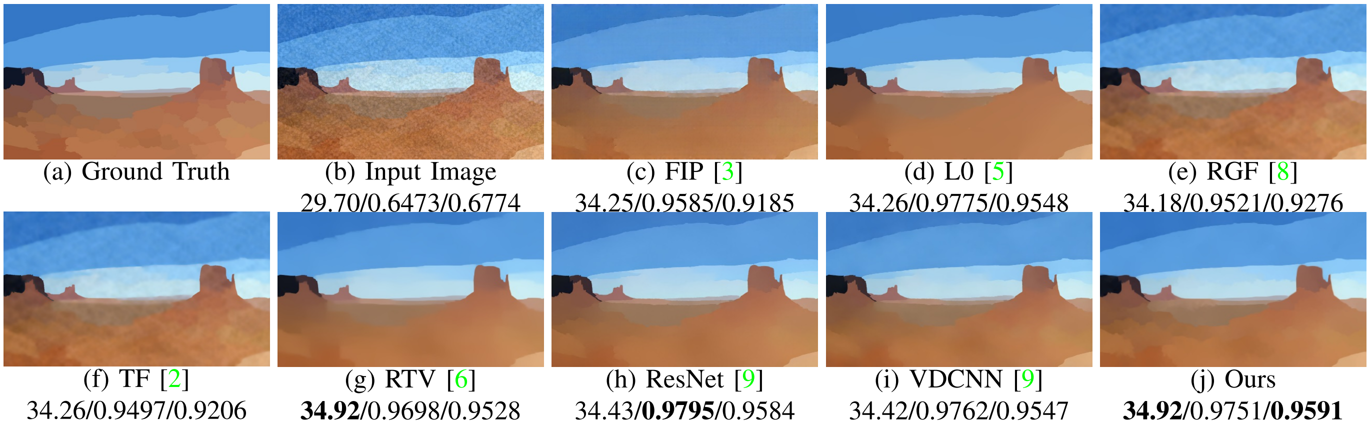


Fig. 1. Comparison of smoothed images and PSNR(dB)/SSIM/FSIM results by different methods on the image "*S03_T07*" from our NKS dataset. The best results are highlighted in **bold**.

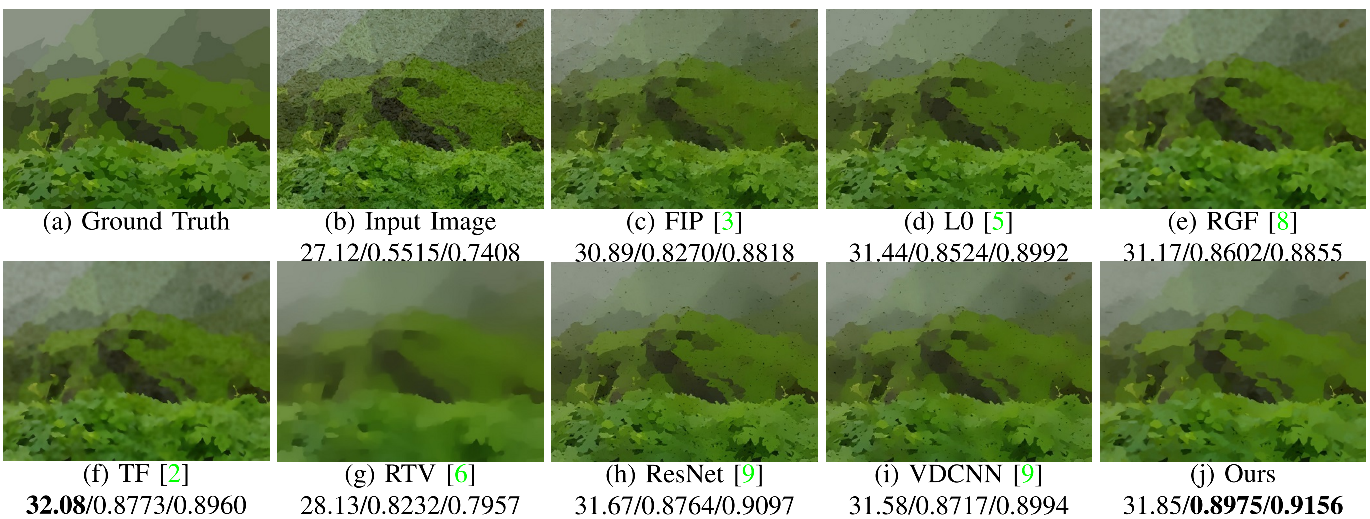


Fig. 2. Comparison of smoothed images and PSNR(dB)/SSIM/FSIM results by different methods on the image "*S07_T02*" from our NKS dataset. The best results are highlighted in **bold**.

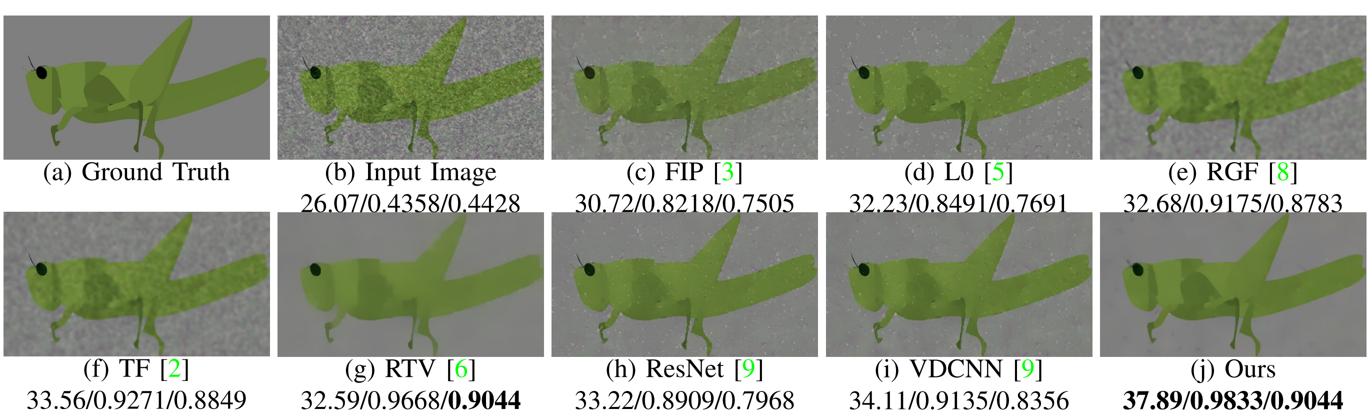
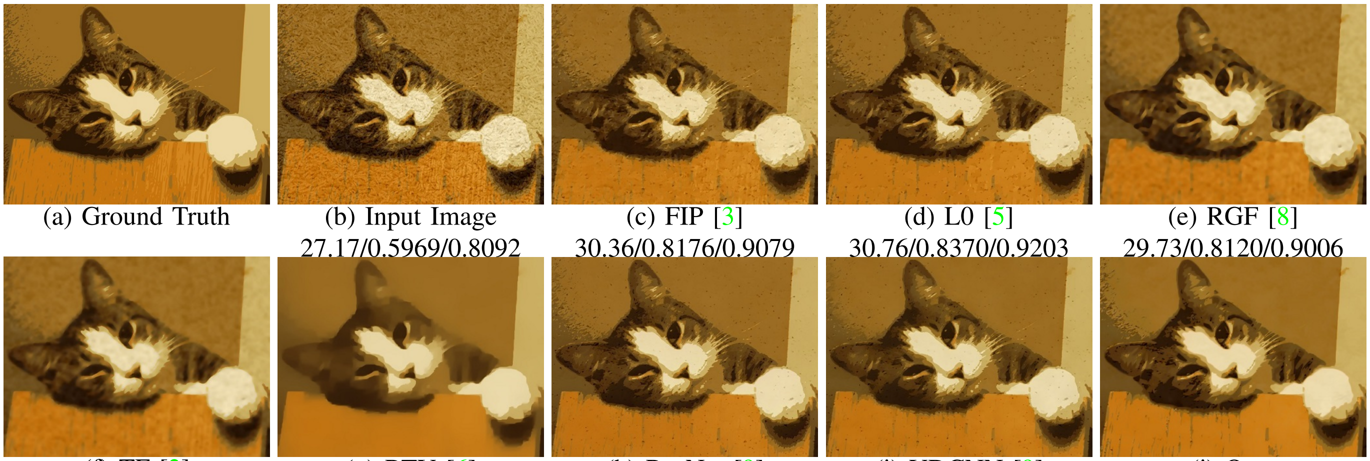
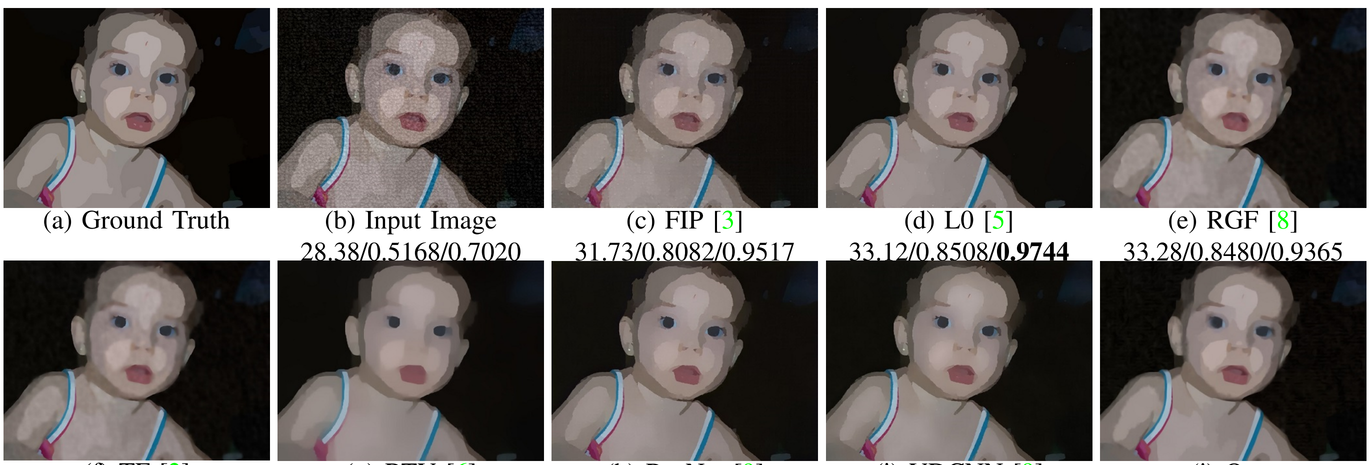


Fig. 3. Comparison of smoothed images and PSNR(dB)/SSIM/FSIM results by different methods on the image "*S09_T09*" from our NKS dataset. The best results are highlighted in **bold**.



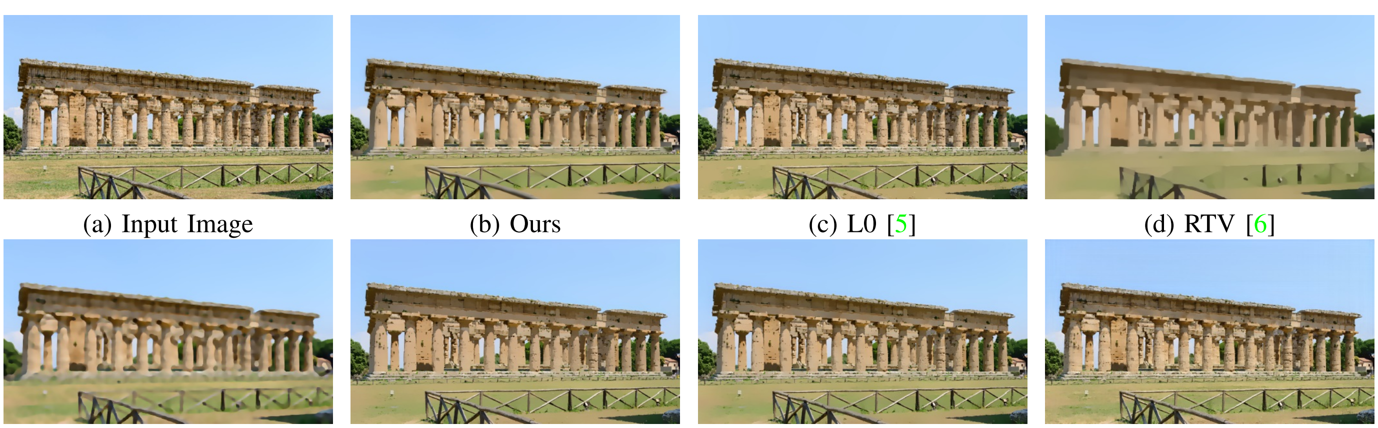
(a) Ground Truth
(b) Input Image 27.17/0.5969/0.8092
(c) FIP [3] 30.36/0.8176/0.9079
(d) L0 [5] 30.76/0.8370/0.9203
(e) RGF [8] 29.73/0.8120/0.9006
(f) TF [2] 30.15/0.8256/0.8994
(g) RTV [6] 28.00/0.7975/0.8374
(h) ResNet [9] 31.64/0.8847/0.9392
(i) VDCNN [9] 31.25/0.8726/0.9277
(j) Ours **32.18/0.8859/0.9417**

Fig. 4. Comparison of smoothed images and PSNR(dB)/SSIM/FSIM results by different methods on the image "*S10_T02*" from our NKS dataset. The best results are highlighted in **bold**.



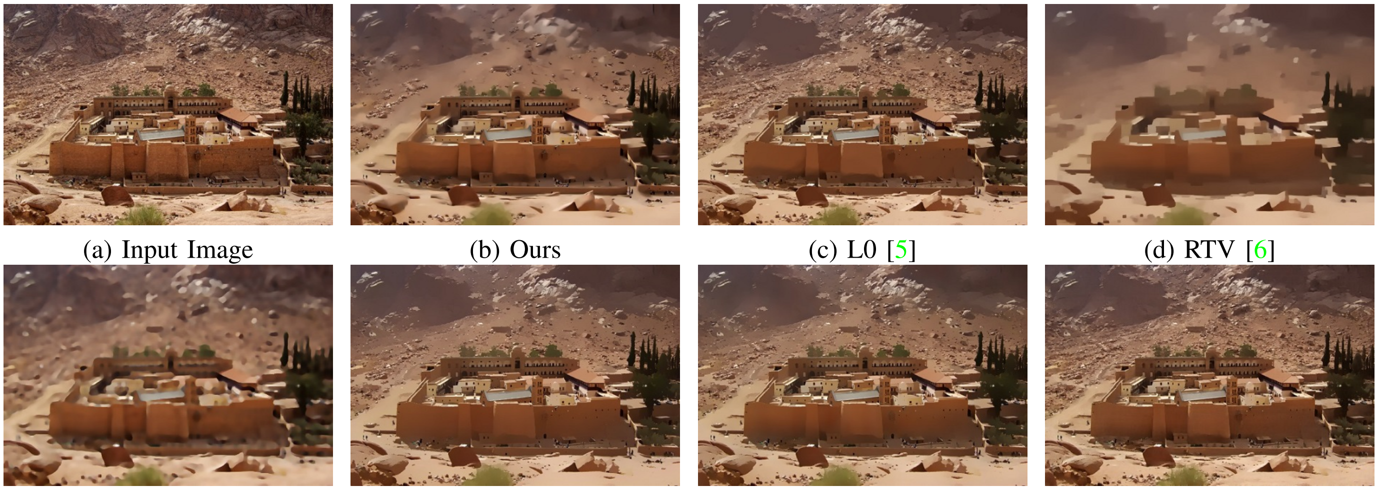
(a) Ground Truth
(b) Input Image 28.38/0.5168/0.7020
(c) FIP [3] 31.73/0.8082/0.9517
(d) L0 [5] 33.12/0.8508/**0.9744**
(e) RGF [8] 33.28/0.8480/0.9365
(f) TF [2] 33.39/0.8572/0.9290
(g) RTV [6] 30.41/0.8278/0.9351
(h) ResNet [9] 30.56/0.8035/0.9604
(i) VDCNN [9] 31.72/0.8253/0.9530
(j) Ours **33.78/0.8755**/0.9385

Fig. 5. Comparison of smoothed images and PSNR(dB)/SSIM/FSIM results by different methods on the image "*S14_T06*" from our NKS dataset. The best results are highlighted in **bold**.



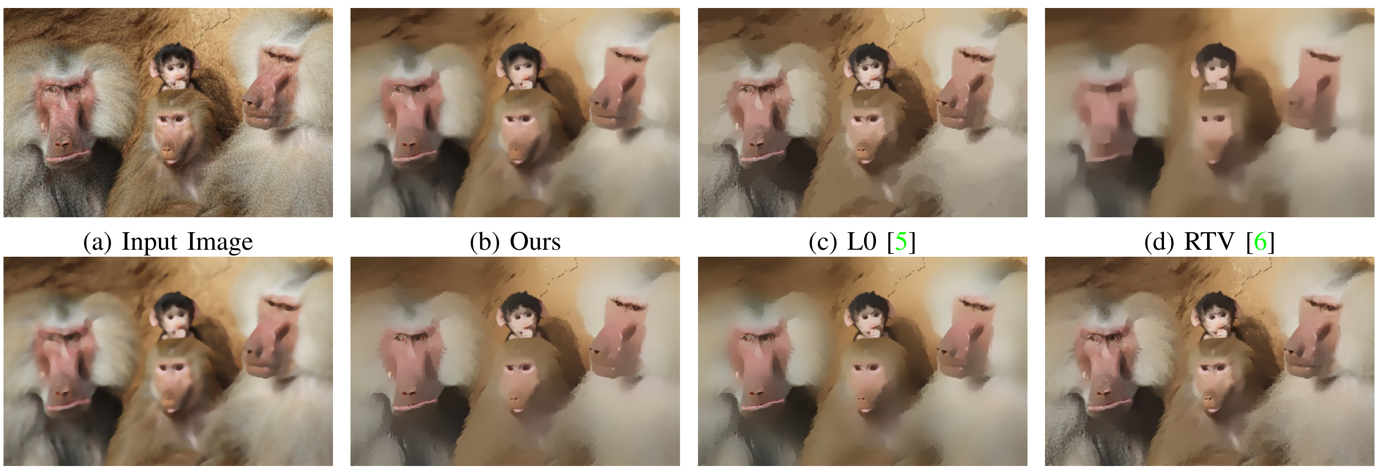
(a) Input Image
(b) Ours
(c) L0 [5]
(d) RTV [6]
(e) RGF [8]
(f) ResNet [9]
(g) VDCNN [9]
(h) FIP [3]

Fig. 6. Comparison of smoothed images by different methods on the image "*0073*" from the DIV2K dataset [1].



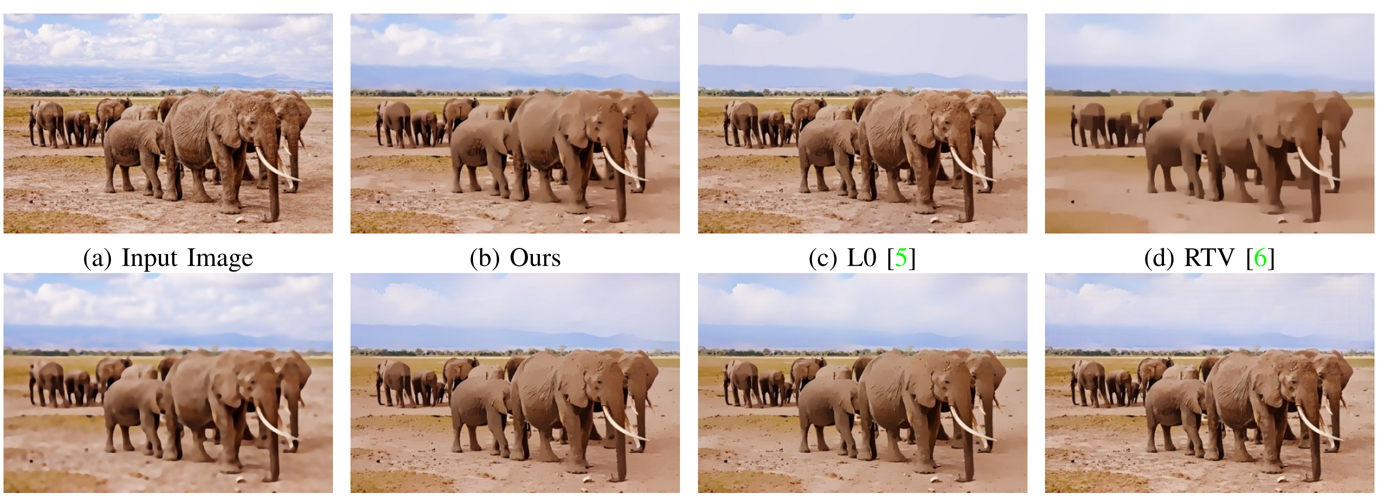
(a) Input Image (b) Ours (c) L0 [5] (d) RTV [6]
(e) RGF [8] (f) ResNet [9] (g) VDCNN [9] (h) FIP [3]

Fig. 7. Comparison of smoothed images by different methods on the image "*0102*" from the DIV2K dataset [1].



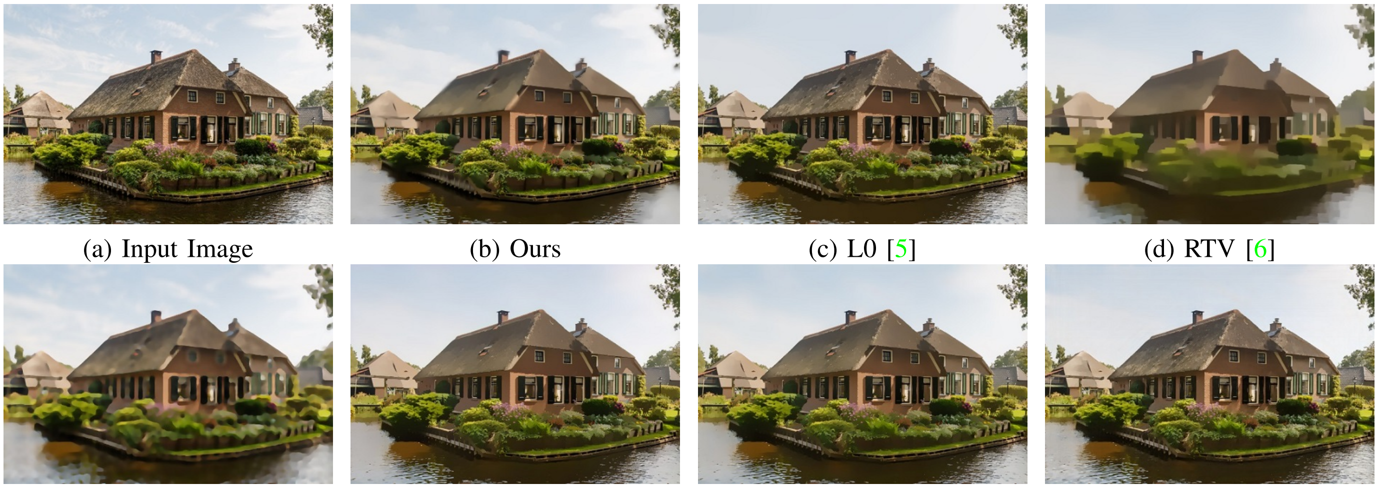
(a) Input Image (b) Ours (c) L0 [5] (d) RTV [6]
(e) RGF [8] (f) ResNet [9] (g) VDCNN [9] (h) FIP [3]

Fig. 8. Comparison of smoothed images by different methods on the image "*0105*" from the DIV2K dataset [1].



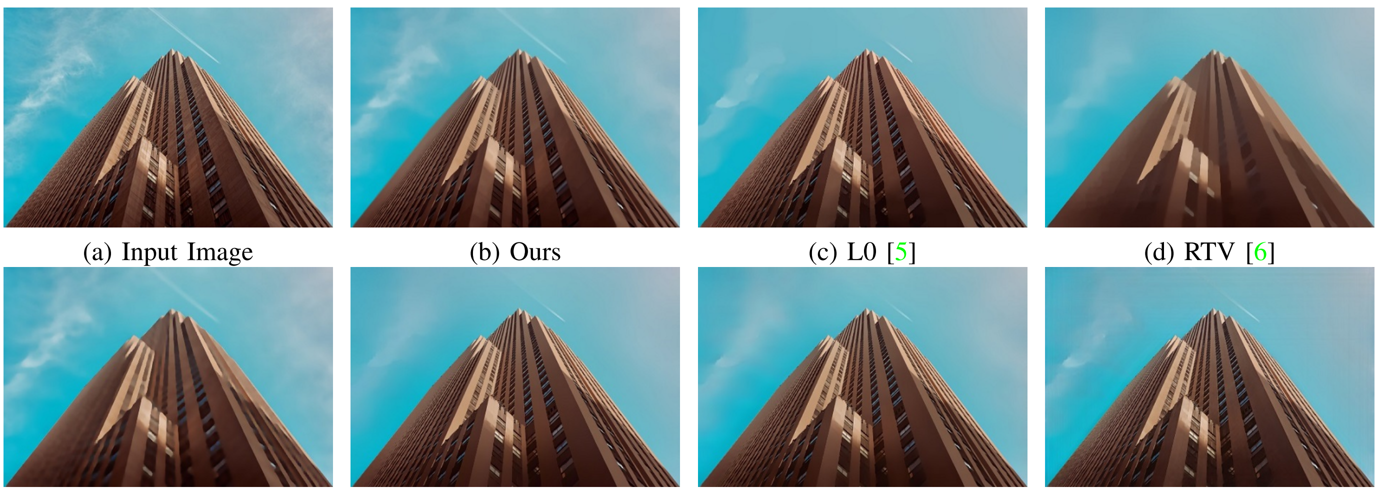
(a) Input Image (b) Ours (c) L0 [5] (d) RTV [6]
(e) RGF [8] (f) ResNet [9] (g) VDCNN [9] (h) FIP [3]

Fig. 9. Comparison of smoothed images by different methods on the image "*0117*" from the DIV2K dataset [1].



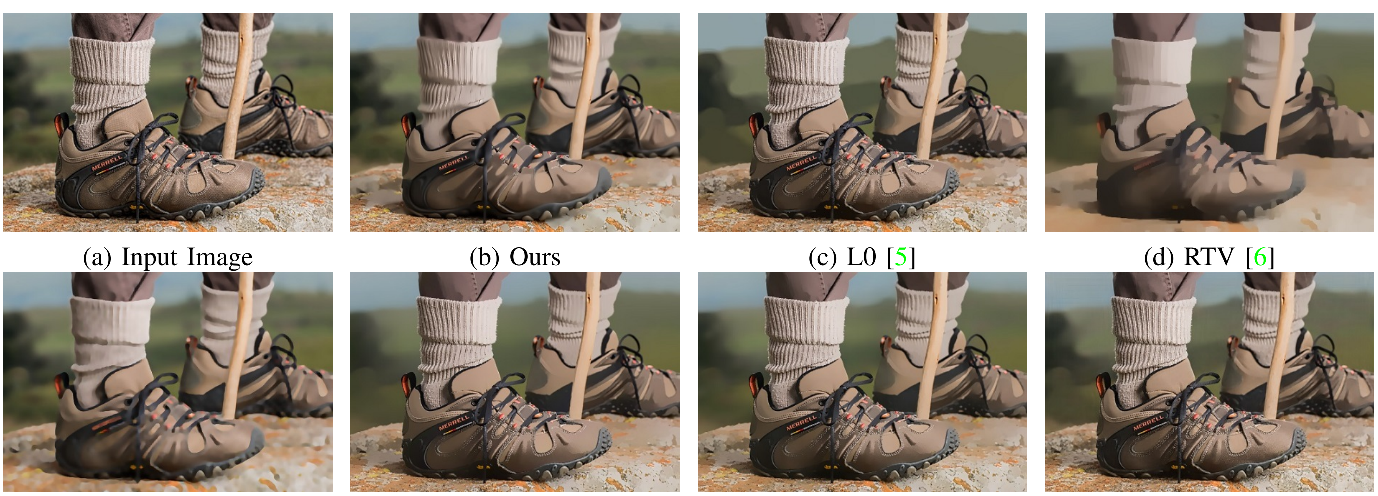
(a) Input Image (b) Ours (c) L0 [5] (d) RTV [6]
(e) RGF [8] (f) ResNet [9] (g) VDCNN [9] (h) FIP [3]

Fig. 10. Comparison of smoothed images by different methods on the image "*0146*" from the DIV2K dataset [1].



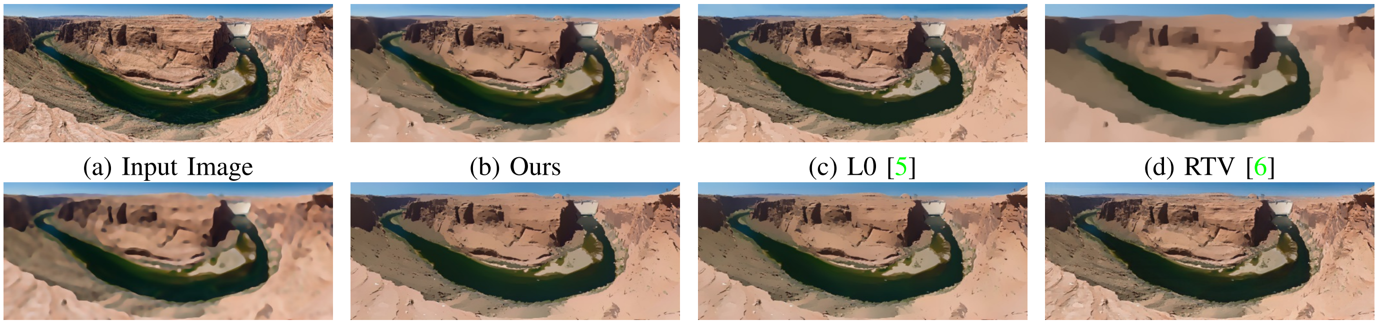
(a) Input Image (b) Ours (c) L0 [5] (d) RTV [6]
(e) RGF [8] (f) ResNet [9] (g) VDCNN [9] (h) FIP [3]

Fig. 11. Comparison of smoothed images by different methods on the image "*0154*" from the DIV2K dataset [1].



(a) Input Image (b) Ours (c) L0 [5] (d) RTV [6]
(e) RGF [8] (f) ResNet [9] (g) VDCNN [9] (h) FIP [3]

Fig. 12. Comparison of smoothed images by different methods on the image "*0166*" from the DIV2K dataset [1].



(a) Input Image (b) Ours (c) L0 [5] (d) RTV [6]

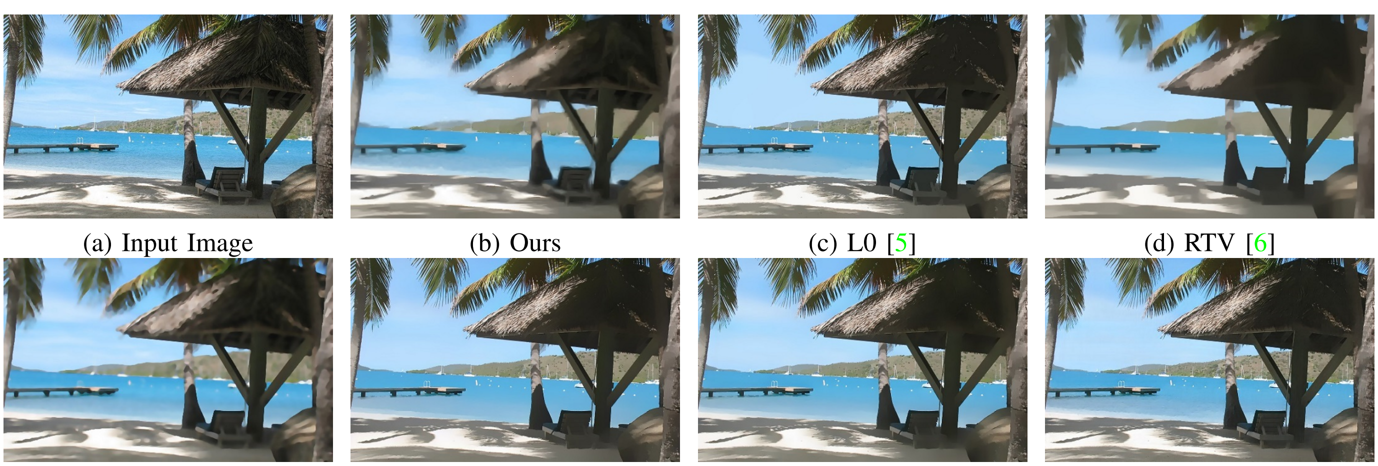
(e) RGF [8] (f) ResNet [9] (g) VDCNN [9] (h) FIP [3]

Fig. 13. Comparison of smoothed images by different methods on the image "*0205*" from the DIV2K dataset [1].

(a) Input Image (b) Ours (c) L0 [5] (d) RTV [6]

(e) RGF [8] (f) ResNet [9] (g) VDCNN [9] (h) FIP [3]

Fig. 14. Comparison of smoothed images by different methods on the image "*0404*" from the DIV2K dataset [1].

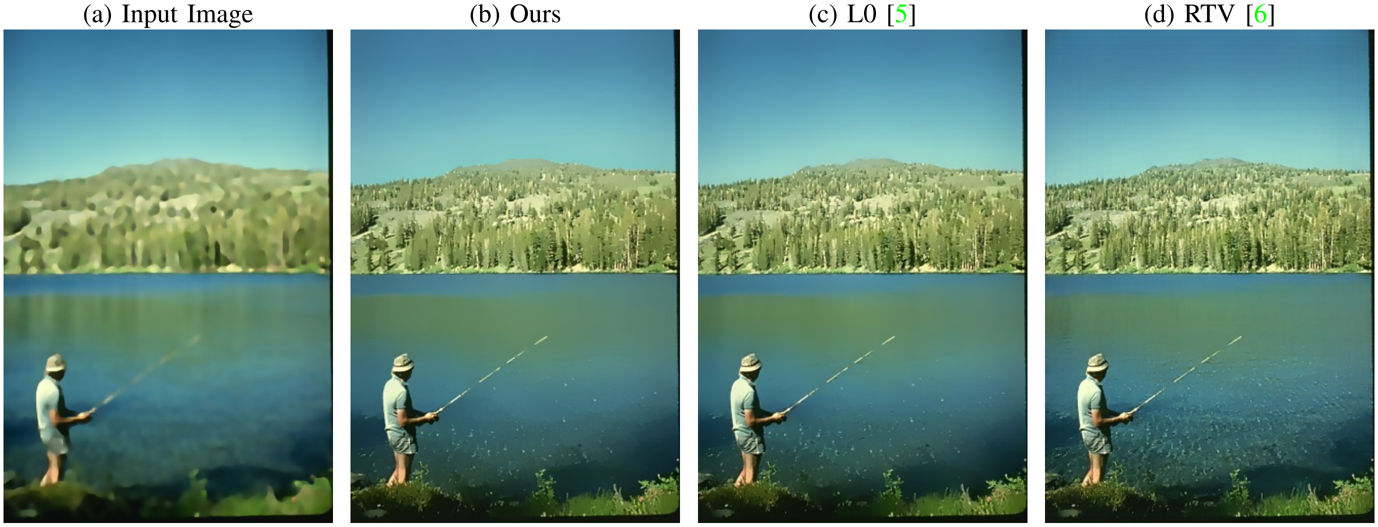


(a) Input Image (b) Ours (c) L0 [5] (d) RTV [6]

(e) RGF [8] (f) ResNet [9] (g) VDCNN [9] (h) FIP [3]

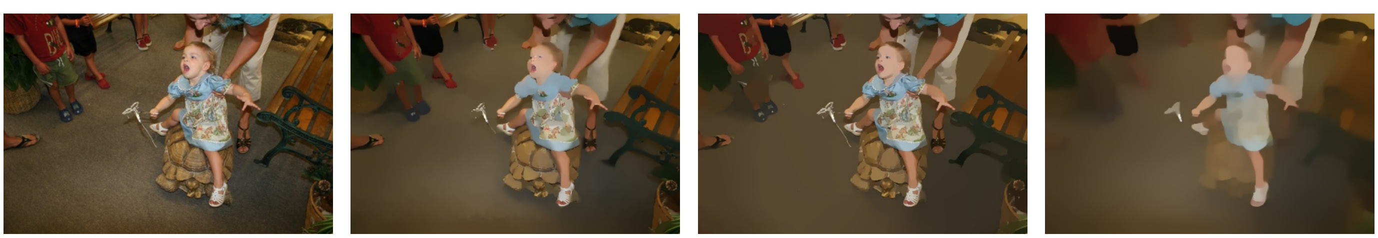
Fig. 15. Comparison of smoothed images by different methods on the image "*0094*" from the dataset in [9]

(a) Input Image (b) Ours (c) L0 [5] (d) RTV [6]



(e) RGF [8] (f) ResNet [9] (g) VDCNN [9] (h) FIP [3]

Fig. 16. Comparison of smoothed images by different methods on the image "*0115*" from the dataset in [9]



(a) Input Image (b) Ours (c) L0 [5] (d) RTV [6]

(e) RGF [8] (f) ResNet [9] (g) VDCNN [9] (h) FIP [3]

Fig. 17. Comparison of smoothed images by different methods on the image "*0169*" from the dataset in [9]


(a) Input Image

(b) Ours

(c) L0 [5]

(d) RTV [6]

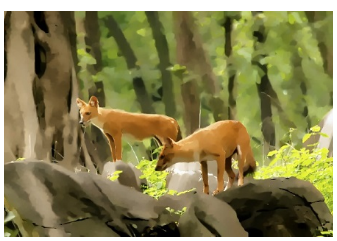
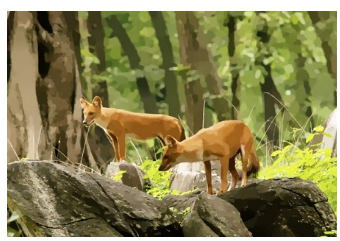
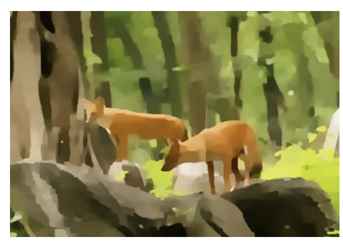
(e) RGF [8]

(f) ResNet [9]

(g) VDCNN [9]

(h) FIP [3]

Fig. 18. Comparison of smoothed images by different methods on the image "*0314*" from the dataset in [9].


(a) Input Image

(b) Ours

(c) L0 [5]

(d) RTV [6]

(e) RGF [8]

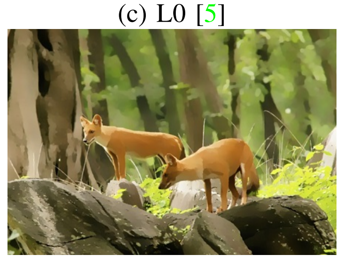
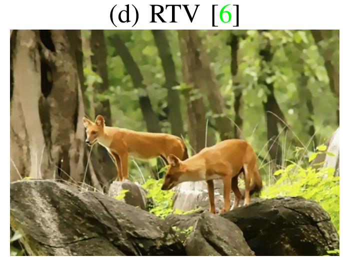
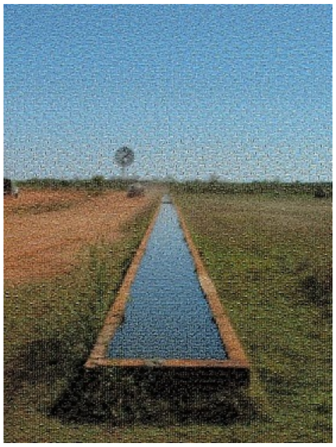
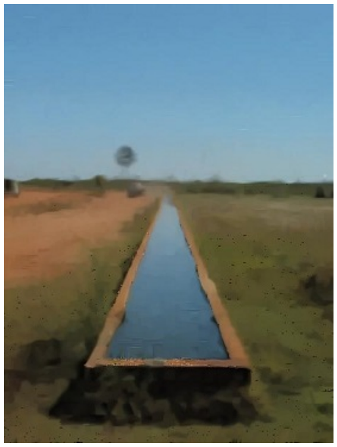
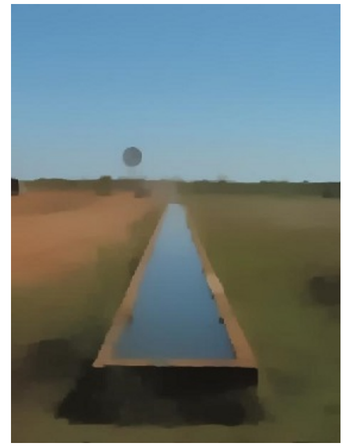
(f) ResNet [9]

(g) VDCNN [9]

(h) FIP [3]

Fig. 19. Comparison of smoothed images by different methods on the image "*0334*" from the dataset in [9].


(a) Input Image

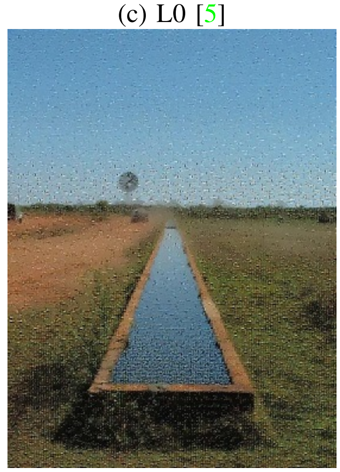

(b) Ours


(c) L0 [5]


(d) RTV [6]


(e) RGF [8]


(f) ResNet [9]


(g) VDCNN [9]

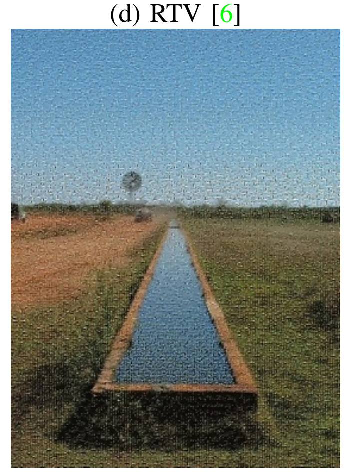
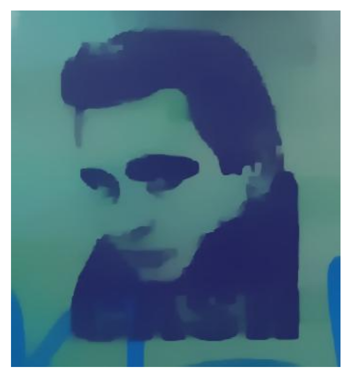

(h) FIP [3]

Fig. 20. Comparison of smoothed images by different methods on the image "*02_23*" from the dataset in [6]

(a) Input Image

(b) Ours

(c) L0 [5]


(d) RTV [6]

(e) RGF [8]

(f) ResNet [9]

(g) VDCNN [9]

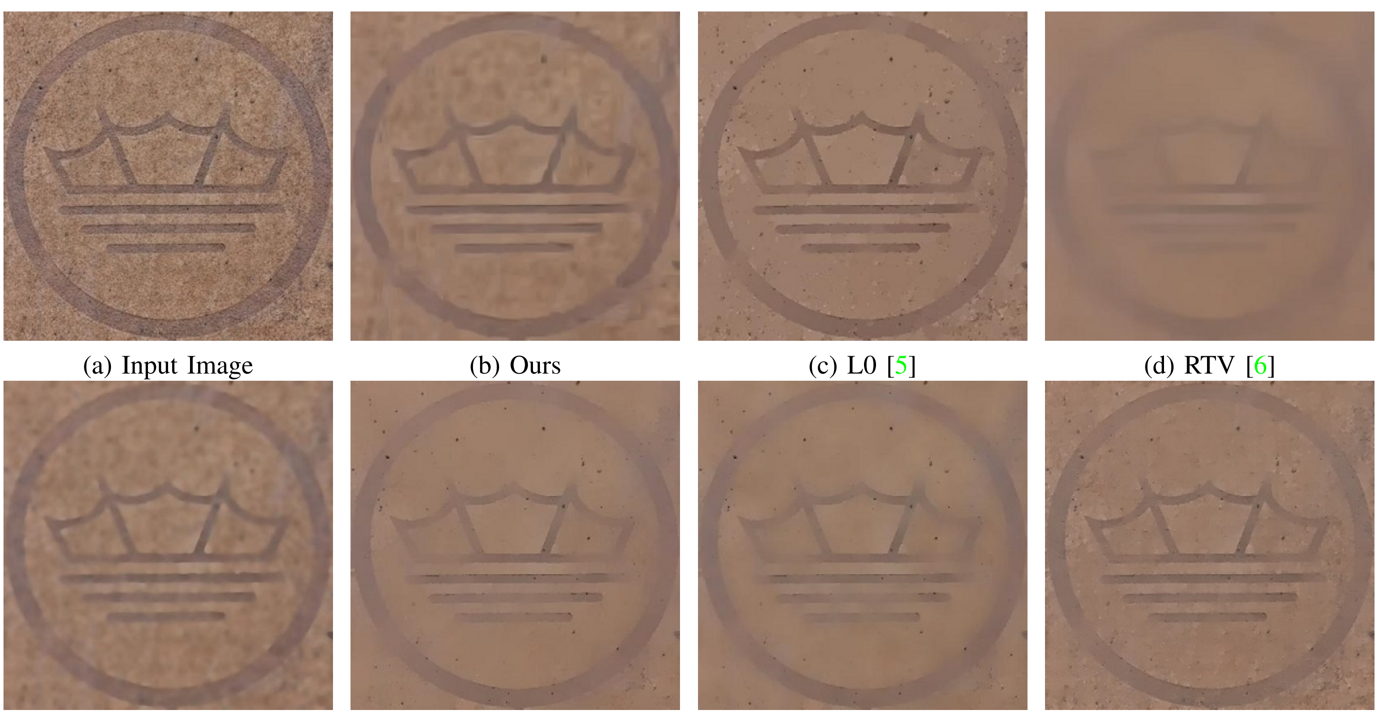
(h) FIP [3]

Fig. 21. Comparison of smoothed images by different methods on the image "*03_11*" from the dataset in [6]

(a) Input Image (b) Ours (c) L0 [5] (d) RTV [6]

(e) RGF [8] (f) ResNet [9] (g) VDCNN [9] (h) FIP [3]

Fig. 22. Comparison of smoothed images by different methods on the image "*11_07*" from the dataset in [6]



(a) Input Image (b) Ours (c) L0 [5] (d) RTV [6]

(e) RGF [8] (f) ResNet [9] (g) VDCNN [9] (h) FIP [3]

Fig. 23. Comparison of smoothed images by different methods on the image "*11_08*" from the dataset in [6]

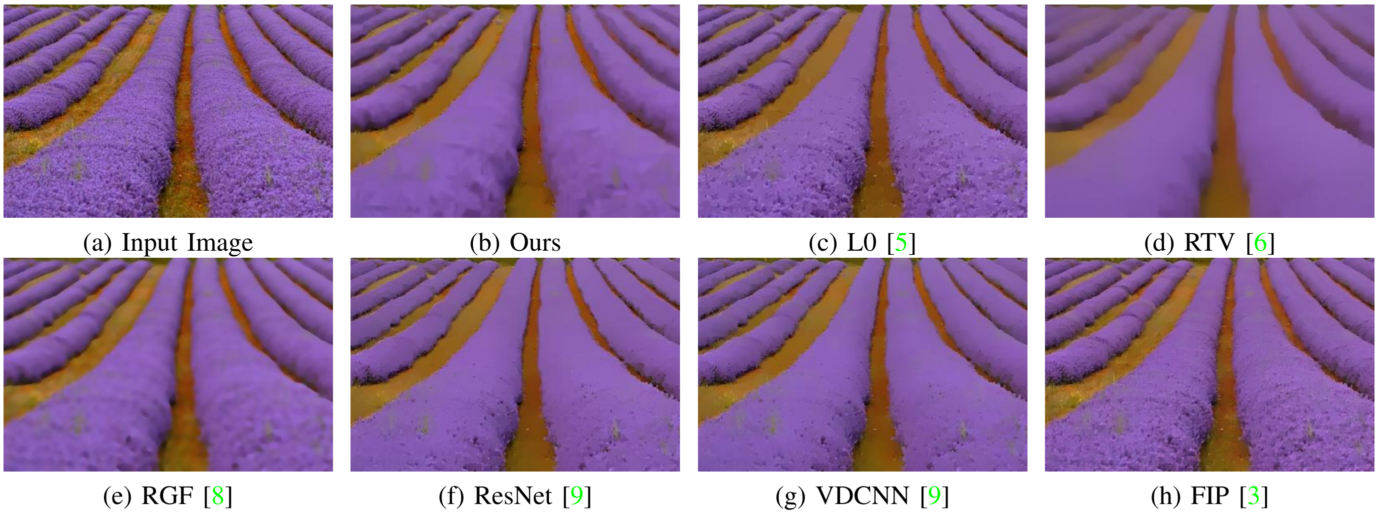


Fig. 24. Comparison of smoothed images by different methods on the image "*11_17*" from the dataset in [6]

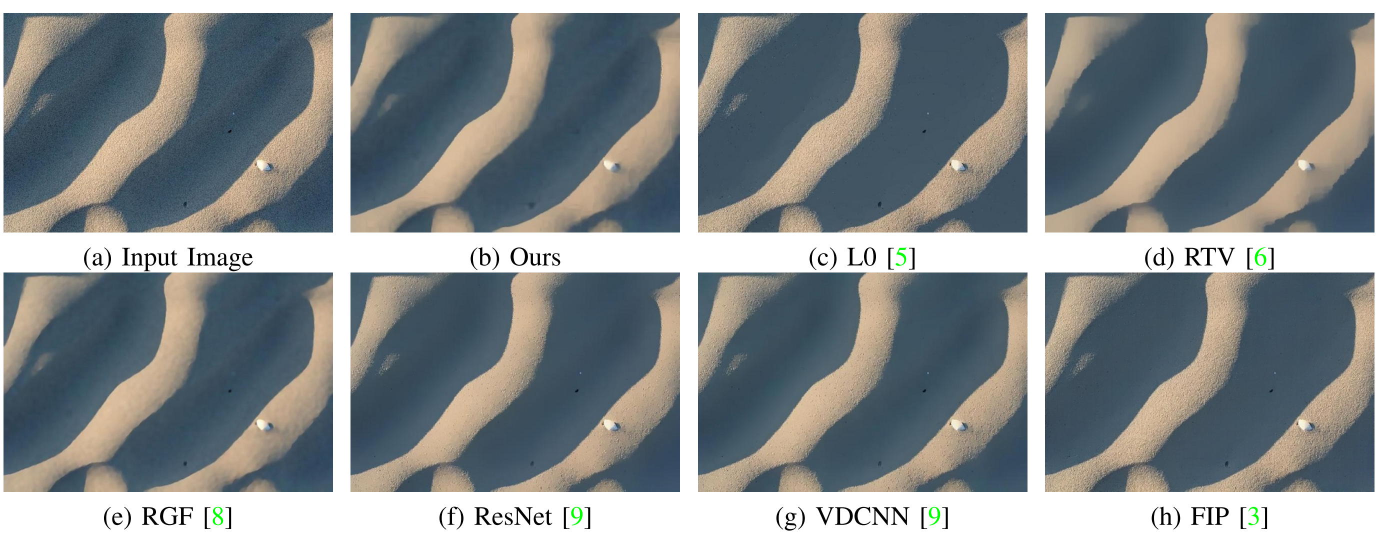


Fig. 25. Comparison of smoothed images by different methods on the image "*11_26*" from the dataset in [6]

(a) Input Image

(b) Ours

(c) L0 [5]

(d) RTV [6]

(e) RGF [8]

(f) ResNet [9]

(g) VDCNN [9]

(h) FIP [3]

Fig. 26. Comparison of smoothed images by different methods on the image "*12_53*" from the dataset in [6]

REFERENCES

[1] E. Agustsson and R. Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *in Proc. IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, July 2017. 1, 2, 4, 5, 6, 7

[2] L. Bao, Y. Song, Q. Yang, H. Yuan, and G. Wang. Tree filtering: Efficient structure-preserving smoothing with a minimum spanning tree. *IEEE Trans. Image Process.*, 23(2):555–569, 2013. 3, 4

[3] Q. Chen, J. Xu, and V. Koltun. Fast image processing with fully-convolutional networks. In *in Proc. IEEE Int. Conf. Comput. Vis.*, pages 2497–2506, 2017. 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

[4] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004. 2

[5] L. Xu, C. Lu, Y. Xu, and J. Jia. Image smoothing via l_0 gradient minimization. *ACM Trans. Graph.*, 30(6):174, 2011. 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

[6] L. Xu, Q. Yan, Y. Xia, and J. Jia. Structure extraction from texture via relative total variation. *ACM Trans. Graph.*, 31(6):139:1–139:10, 2012. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

[7] L. Zhang, L. Zhang, X. Mou, and D. Zhang. Fsim: A feature similarity index for image quality assessment. *IEEE Trans. Image Process.*, 20(8):2378–2386, 2011. 2

[8] Q. Zhang, X. Shen, L. Xu, and J. Jia. Rolling guidance filter. In *in European Conference on Computer Vision.*, pages 815–830, 2014. 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

[9] F. Zhu, Z. Liang, X. Jia, L. Zhang, and Y. Yu. A benchmark for edge-preserving image smoothing. *IEEE Trans. Image Process.*, 28(7):3556–3570, 2019. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13