

Supplementary File to “Non-local Image Smoothing with Objective Evaluation”

Zhi-Ang Liu¹, Ying-Kun Hou², Jun Xu^{1,*}, Xian-Tong Zhen^{3,4}, Ling Shao^{3,4}, Ming-Ming Cheng¹

¹TKLNDST, College of Computer Science, Nankai University, Tianjin, China

²School of Information Science and Technology, Taishan University, Tai'an, China

³Inception 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 $\mathbf{S} \in \mathbb{R}^{4 \times 16}$ could be represented by columns as $\mathbf{S} = [\mathbf{s}_1^4, \dots, \mathbf{s}_{16}^4] \in \mathbb{R}^{4 \times 16}$. The Haar transformation includes horizontal and vertical transformation. We first apply the horizontal transformation. Specifically, we multiply the similar pixels matrix $\mathbf{S} \in \mathbb{R}^{4 \times 16}$ and the horizontal transformation matrix $\mathbf{H}_r \in \mathbb{R}^{16 \times 16}$:

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

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

$$\begin{aligned} \hat{\mathbf{t}}^1 &= \frac{1}{\sqrt{4}} \sum_{i=1}^4 \mathbf{t}^i, \quad \hat{\mathbf{t}}^2 = \frac{1}{\sqrt{4}} \left(\sum_{i=1}^2 \mathbf{t}^i - \sum_{i=3}^4 \mathbf{t}^i \right), \\ \hat{\mathbf{t}}^3 &= \frac{1}{\sqrt{2}} (\mathbf{t}^1 - \mathbf{t}^2), \quad \hat{\mathbf{t}}^4 = \frac{1}{\sqrt{2}} (\mathbf{t}^3 - \mathbf{t}^4). \end{aligned} \quad (2)$$

After the thresholding step, we could get the thresholded representation matrix $\tilde{\mathbf{T}} \in \mathbb{R}^{4 \times 16}$. 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 $\mathbf{H}_{il} \in \mathbb{R}^{4 \times 4}$ and the thresholded representation matrix $\tilde{\mathbf{T}}^4 \in \mathbb{R}^{4 \times 16}$:

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

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

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

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

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

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 6-24, we compare the visual quality of different methods on image smoothing on the datasets of [1], [6], [9]. The comparison results demonstrate that the PNLS method achieves better visual quality than the other image smoothing methods.

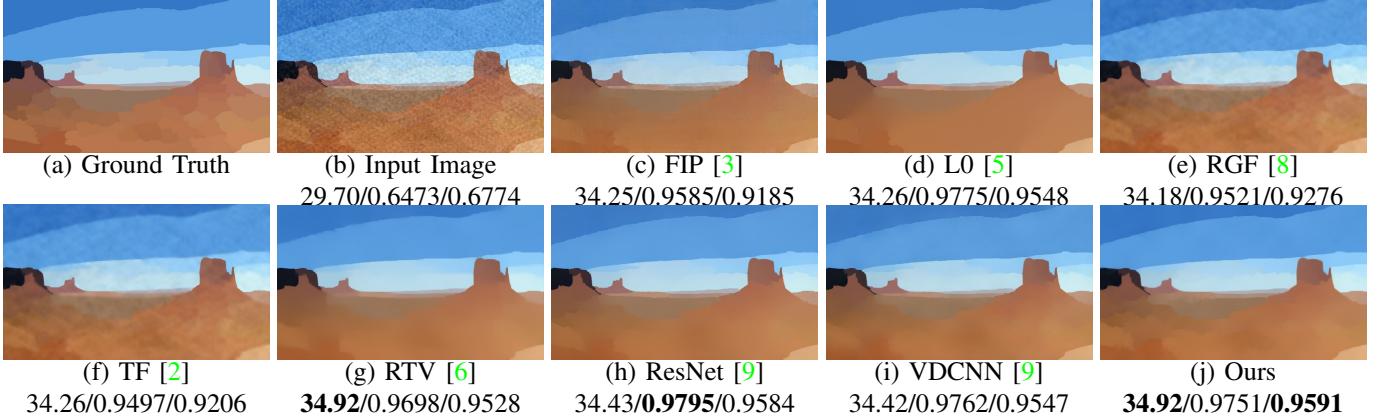


Fig. 1. Comparison of smoothed images and PSNR(dB)/SSIM/FSIM results by different methods on the image “S03_T07” from our NKS dataset. The best results are highlighted in **bold**.

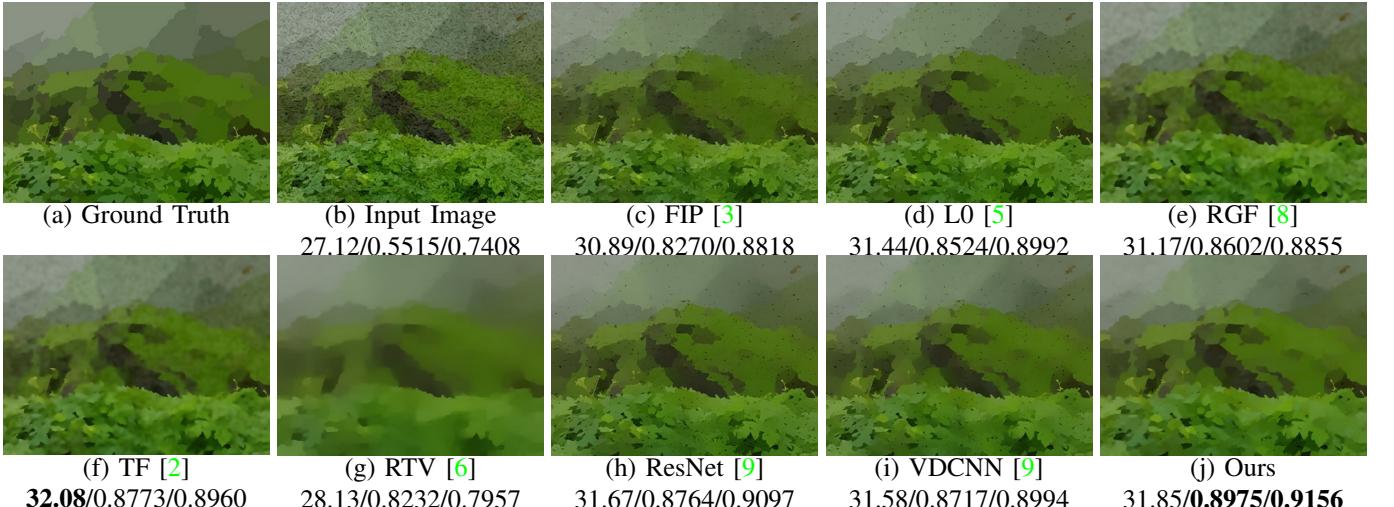


Fig. 2. Comparison of smoothed images and PSNR(dB)/SSIM/FSIM results by different methods on the image “S07_T02” from our NKS dataset. The best results are highlighted in **bold**.

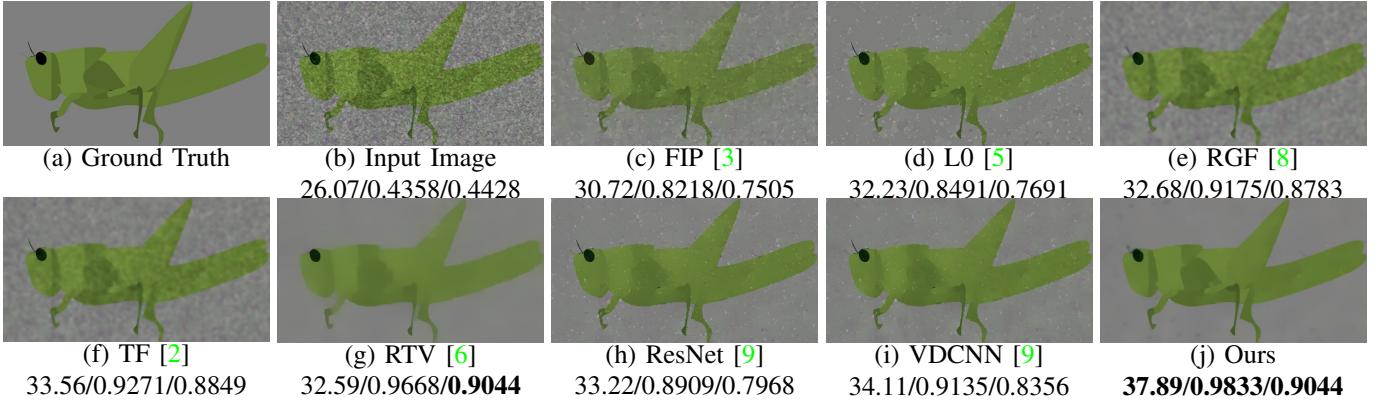


Fig. 3. Comparison of smoothed images and PSNR(dB)/SSIM/FSIM results by different methods on the image “S09_T09” from our NKS dataset. The best results are highlighted in **bold**.

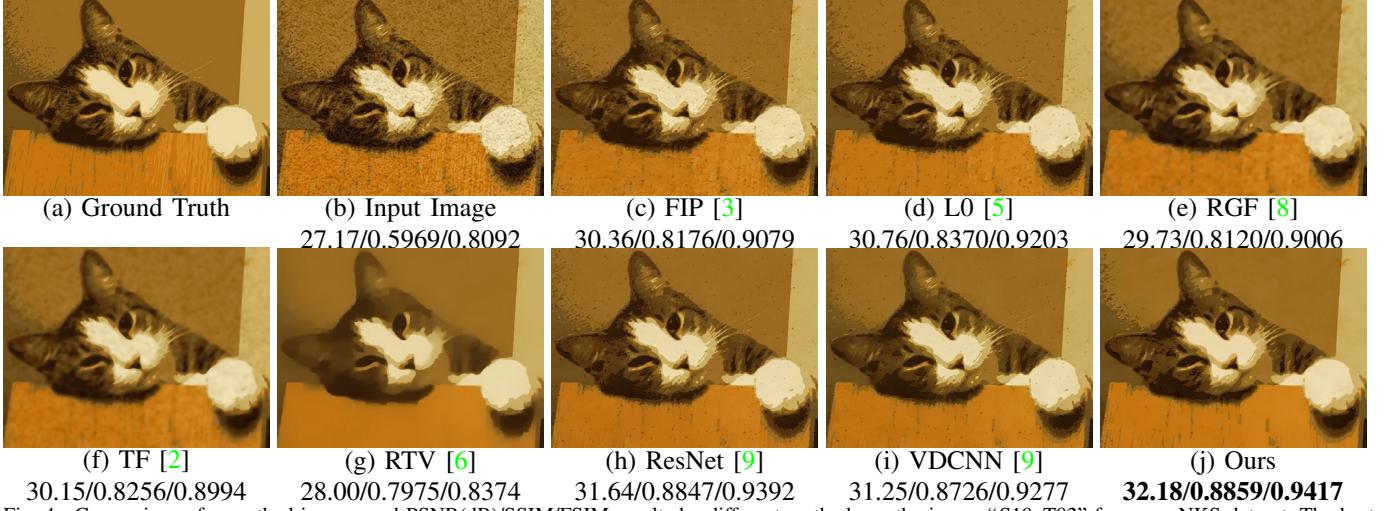


Fig. 4. Comparison of smoothed images and PSNR(dB)/SSIM/FSIM results by different methods on the image “S10_T02” from our NKS dataset. The best results are highlighted in **bold**.

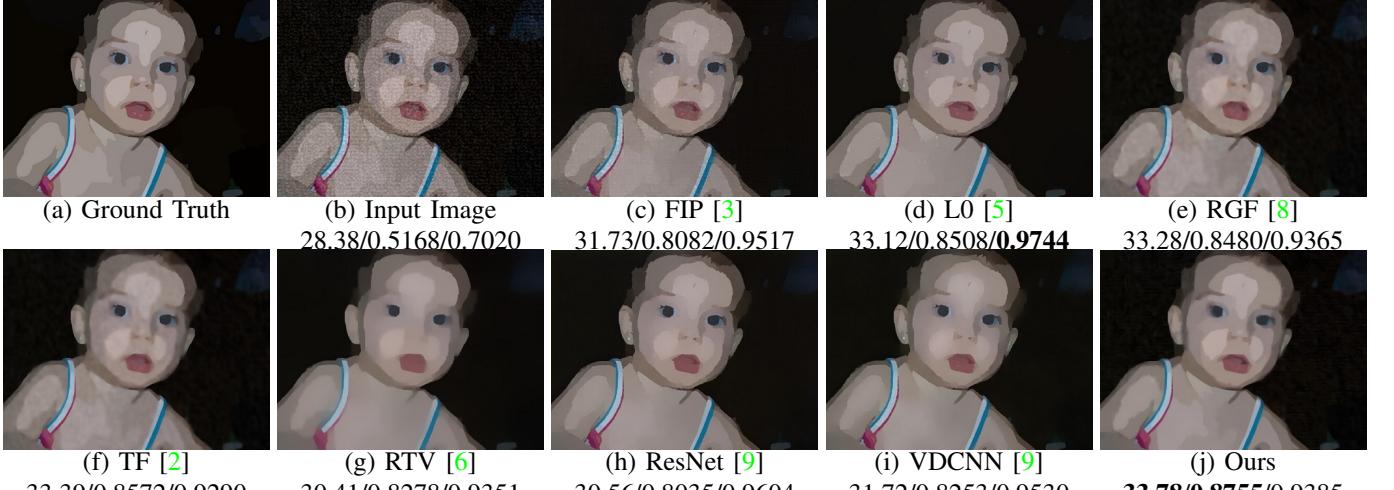


Fig. 5. Comparison of smoothed images and PSNR(dB)/SSIM/FSIM results by different methods on the image “S14_T06” from our NKS dataset. The best results are highlighted in **bold**.

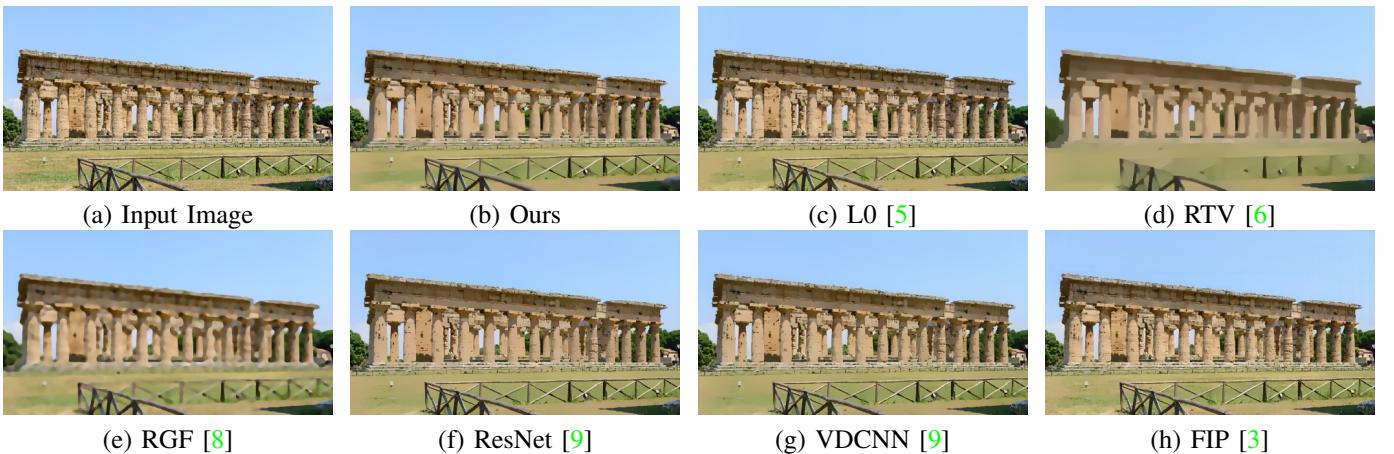


Fig. 6. Comparison of smoothed images by different methods on the image “0073” from the DIV2K dataset [1].

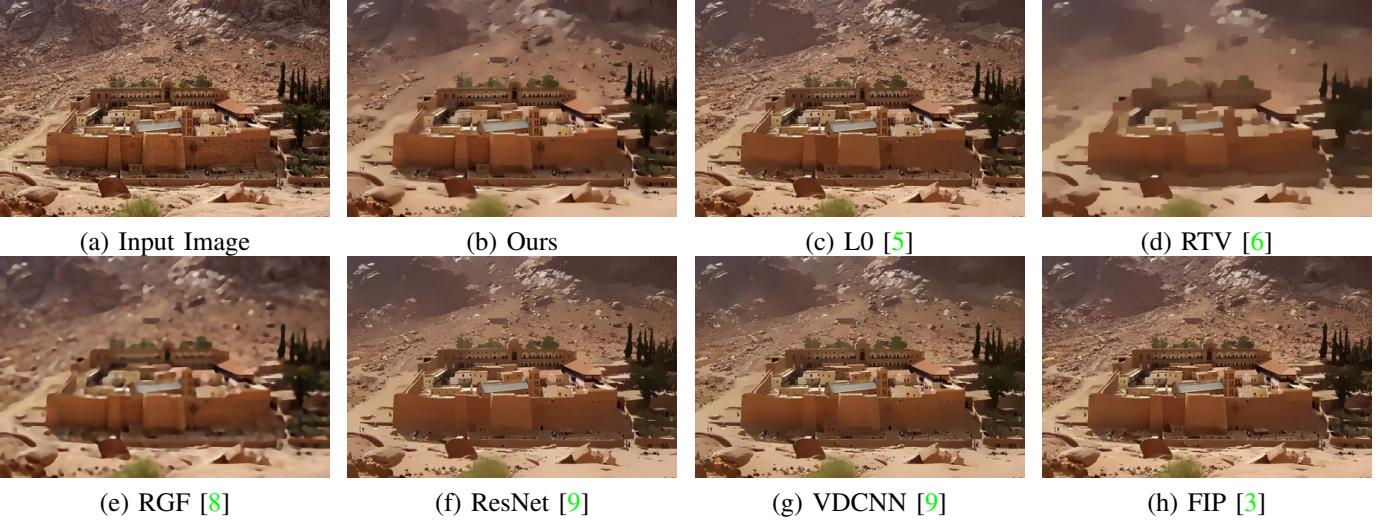


Fig. 7. Comparison of smoothed images by different methods on the image “0102” from the DIV2K dataset [1].

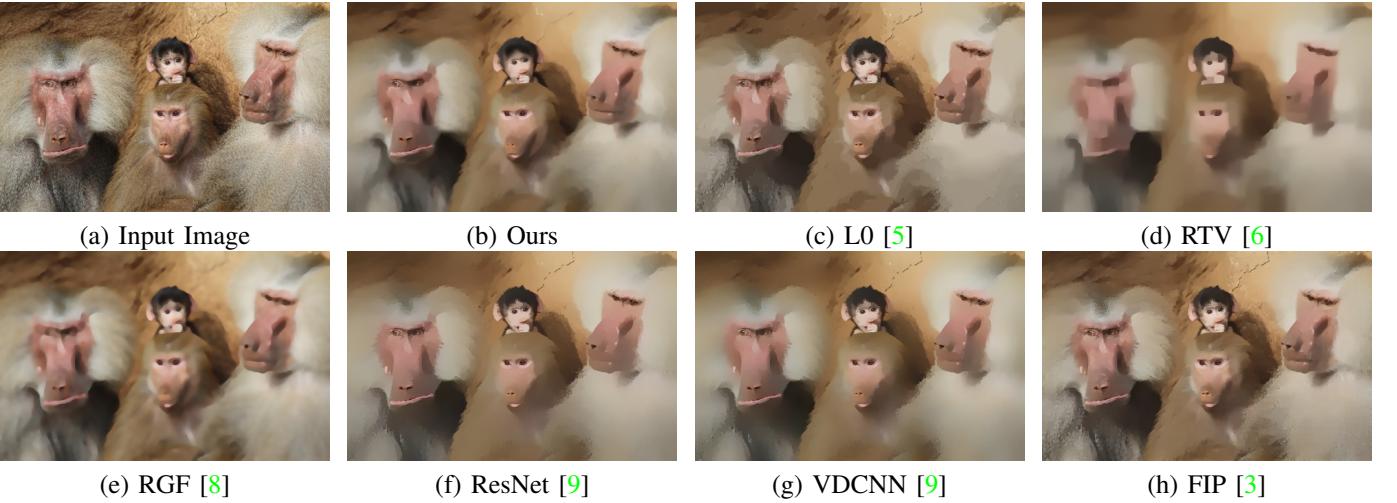


Fig. 8. Comparison of smoothed images by different methods on the image “0105” from the DIV2K dataset [1].

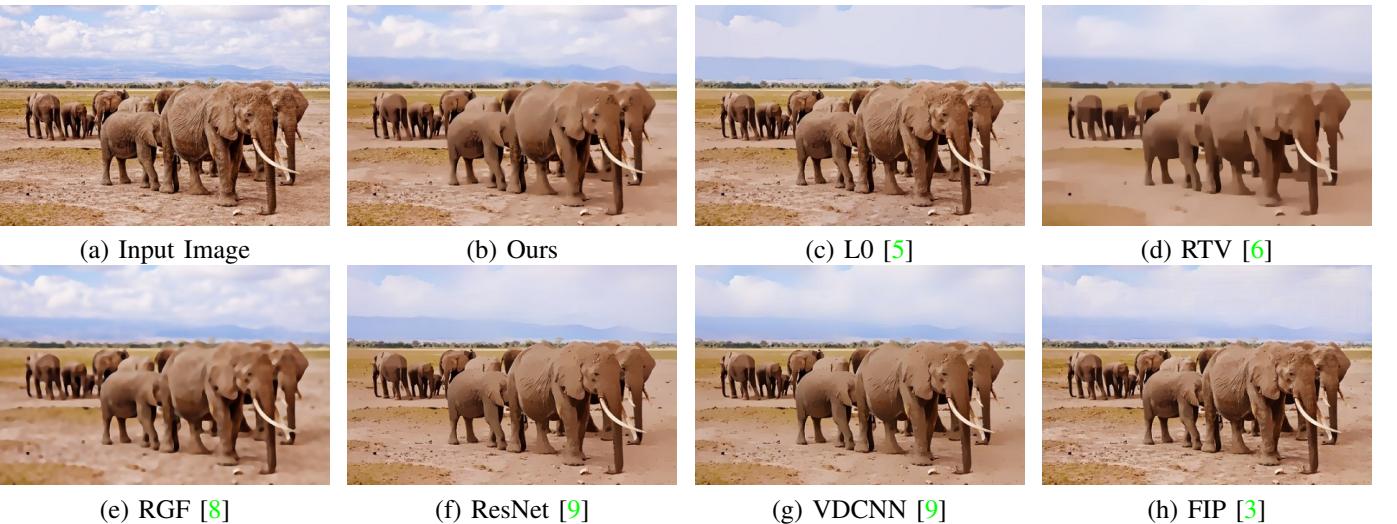


Fig. 9. Comparison of smoothed images by different methods on the image “0117” from the DIV2K dataset [1].

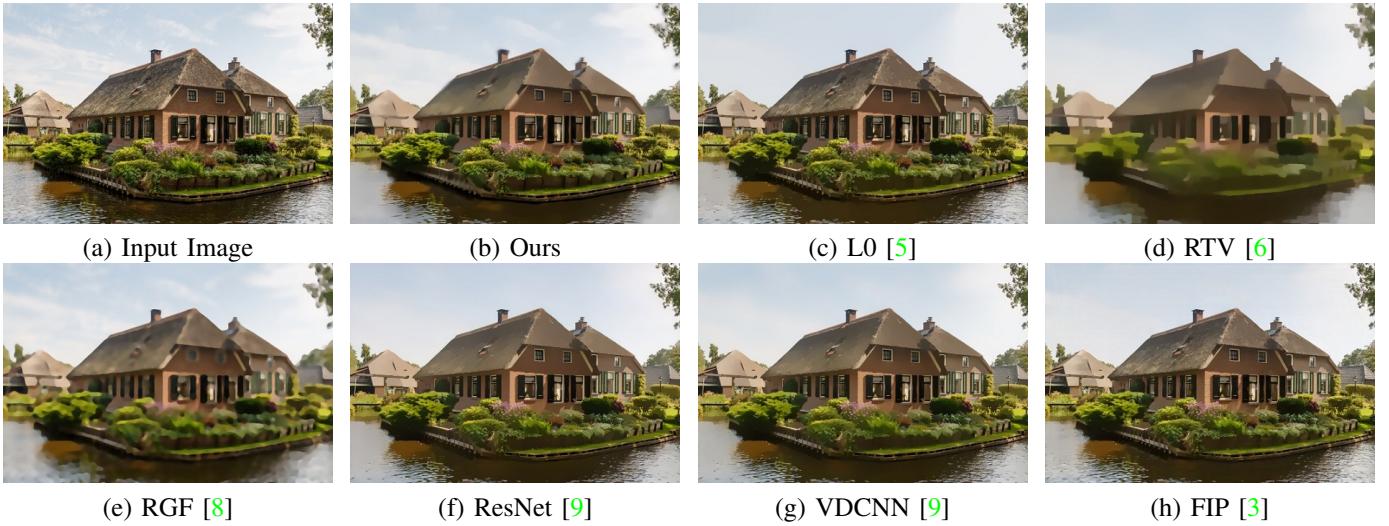


Fig. 10. Comparison of smoothed images by different methods on the image “0146” from the DIV2K dataset [1].

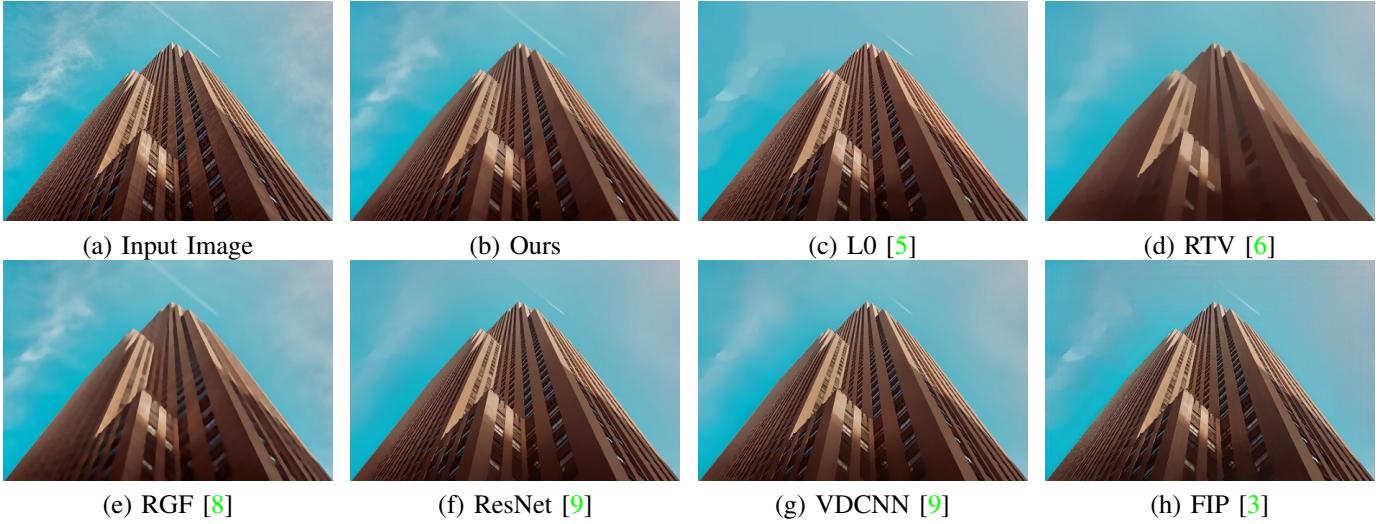


Fig. 11. Comparison of smoothed images by different methods on the image “0154” from the DIV2K dataset [1].

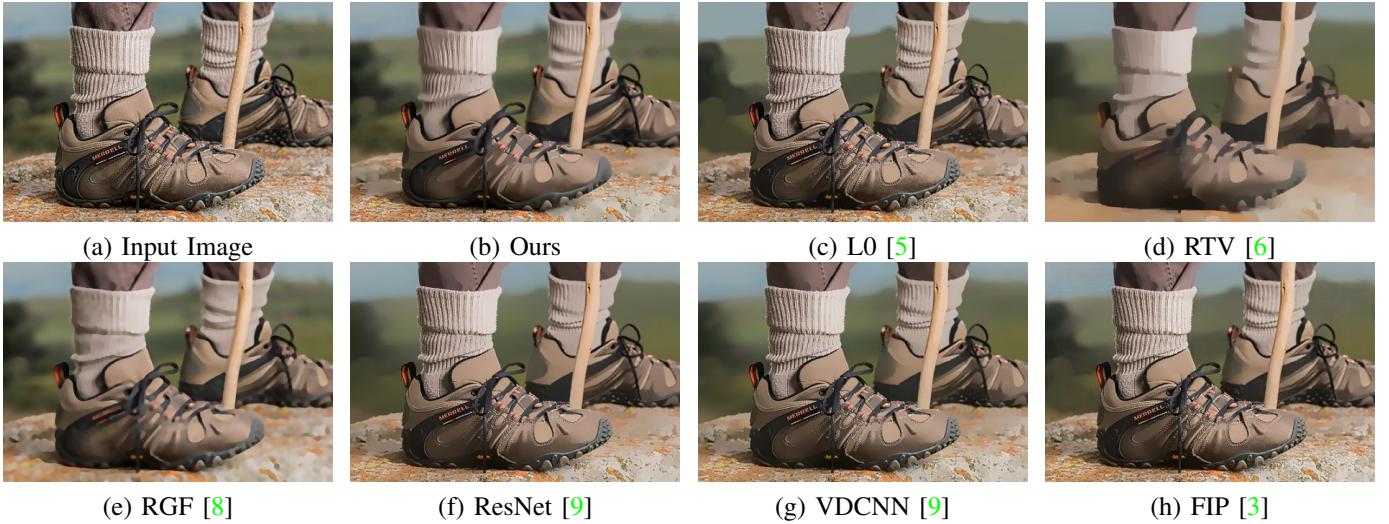


Fig. 12. Comparison of smoothed images by different methods on the image “0166” from the DIV2K dataset [1].

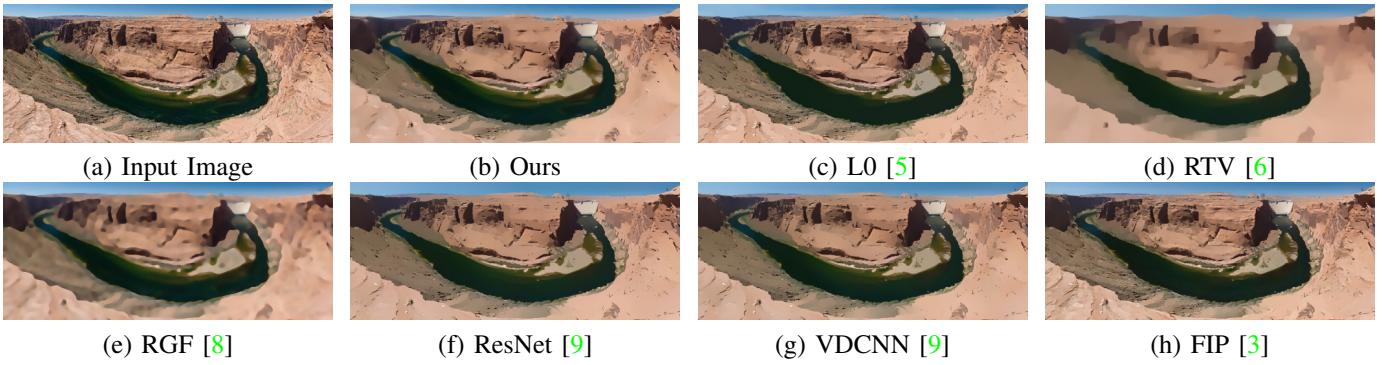


Fig. 13. Comparison of smoothed images by different methods on the image “0205” from the DIV2K dataset [1].

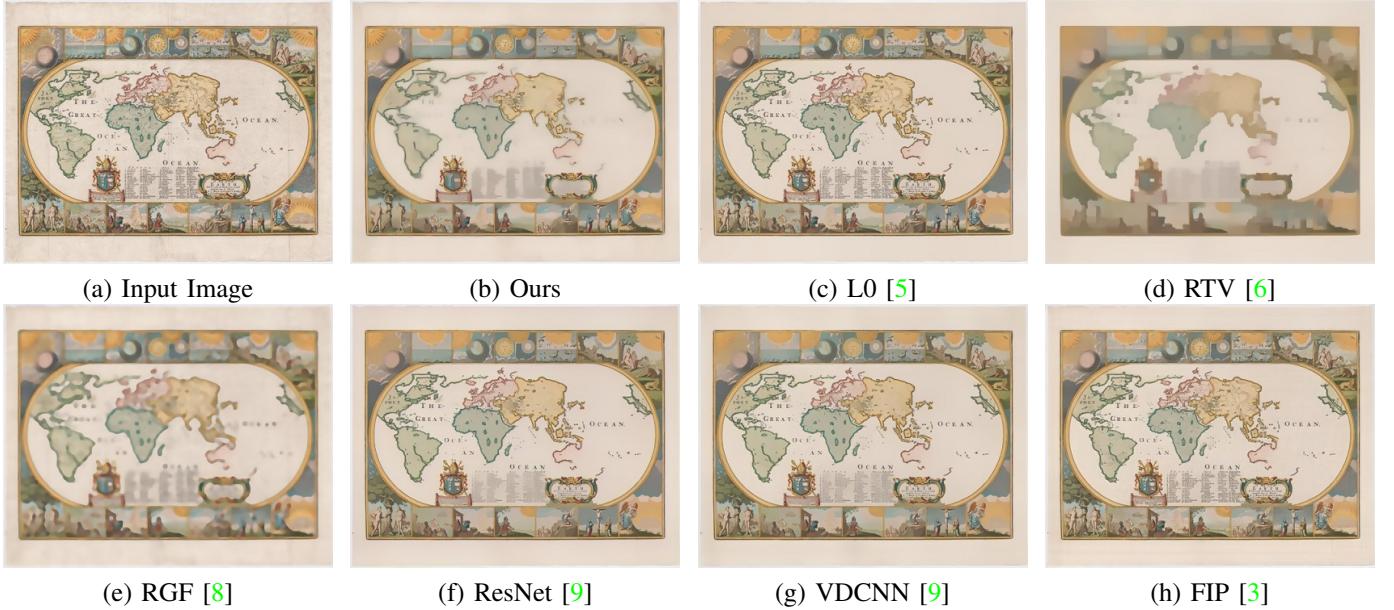


Fig. 14. Comparison of smoothed images by different methods on the image “0404” from the DIV2K dataset [1].

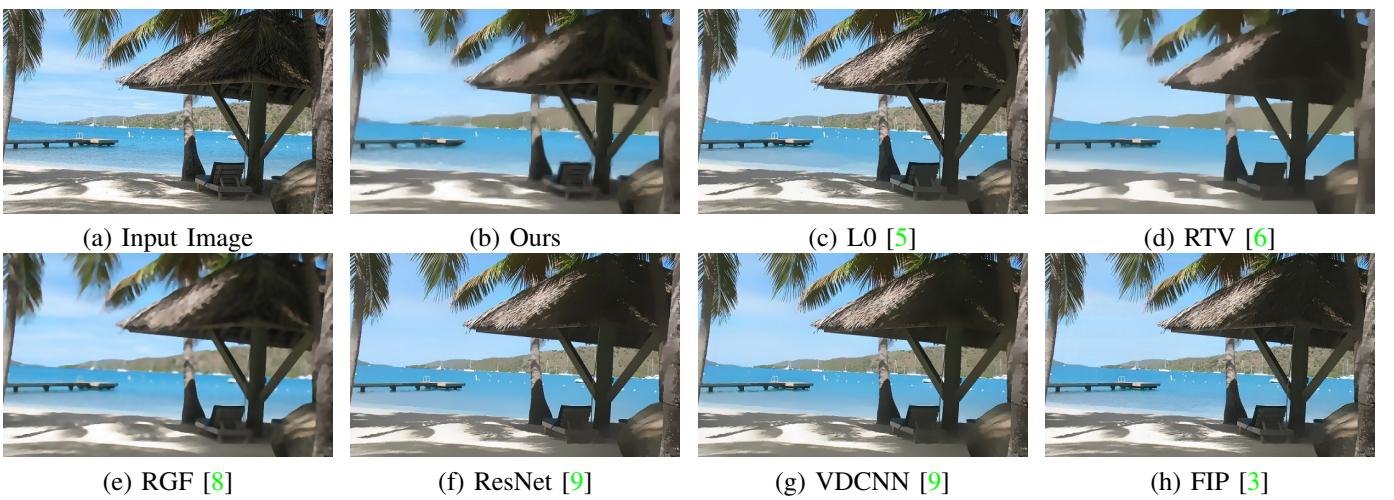


Fig. 15. Comparison of smoothed images by different methods on the image “0094” from the dataset in [9].

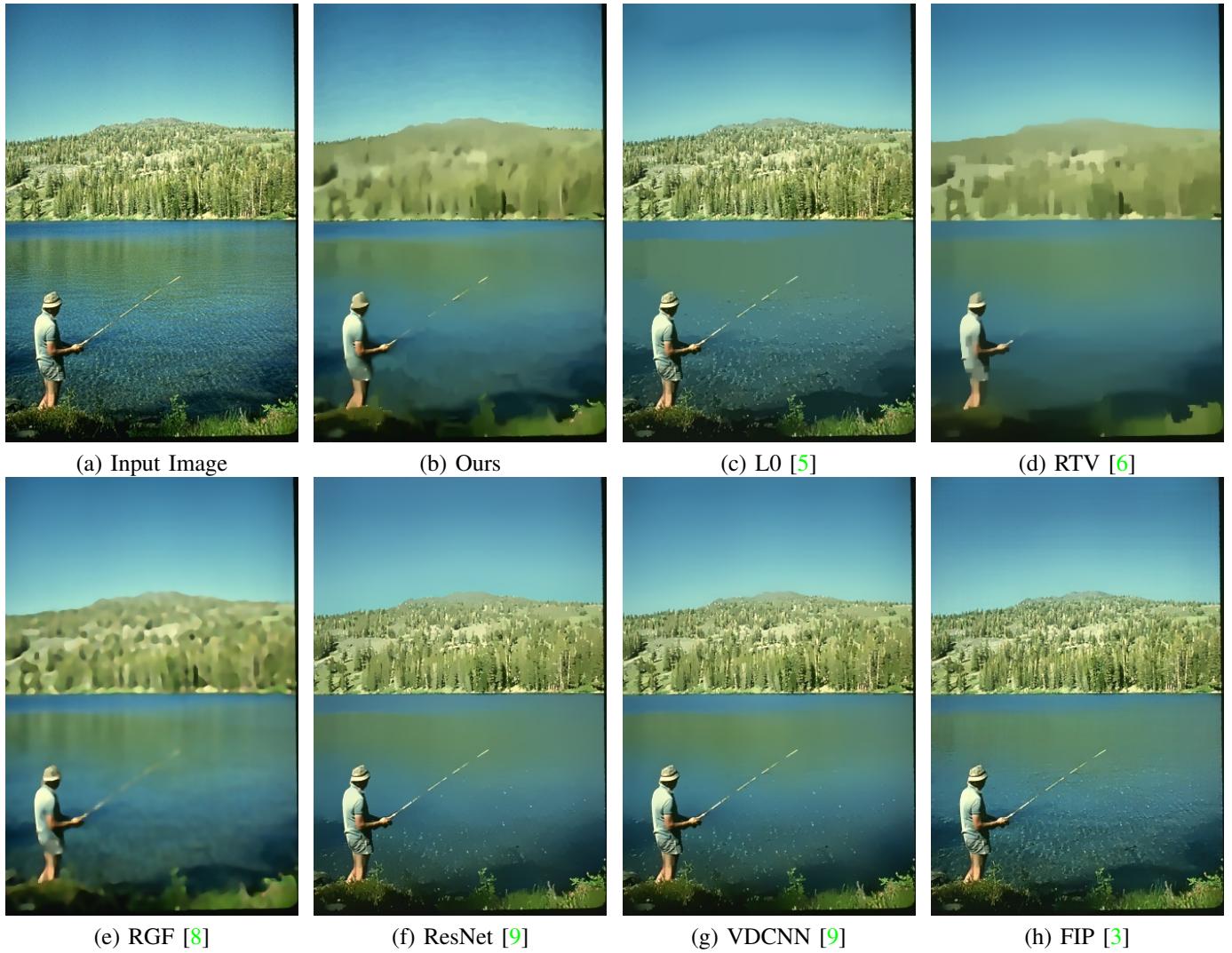


Fig. 16. Comparison of smoothed images by different methods on the image “0115” from the dataset in [9]

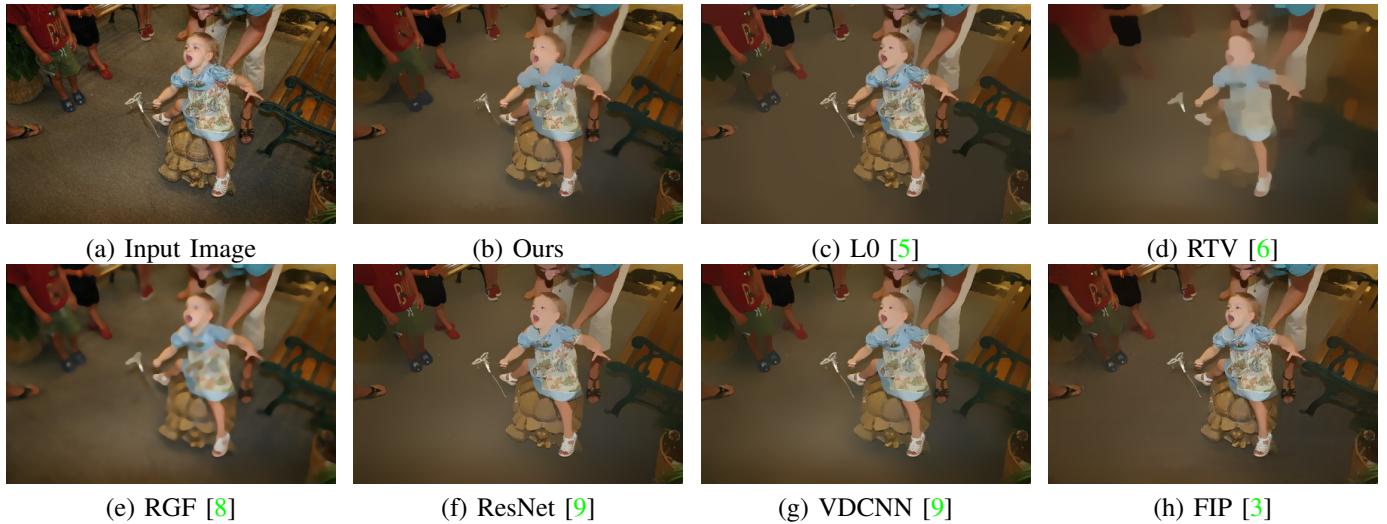


Fig. 17. Comparison of smoothed images by different methods on the image “0169” from the dataset in [9]

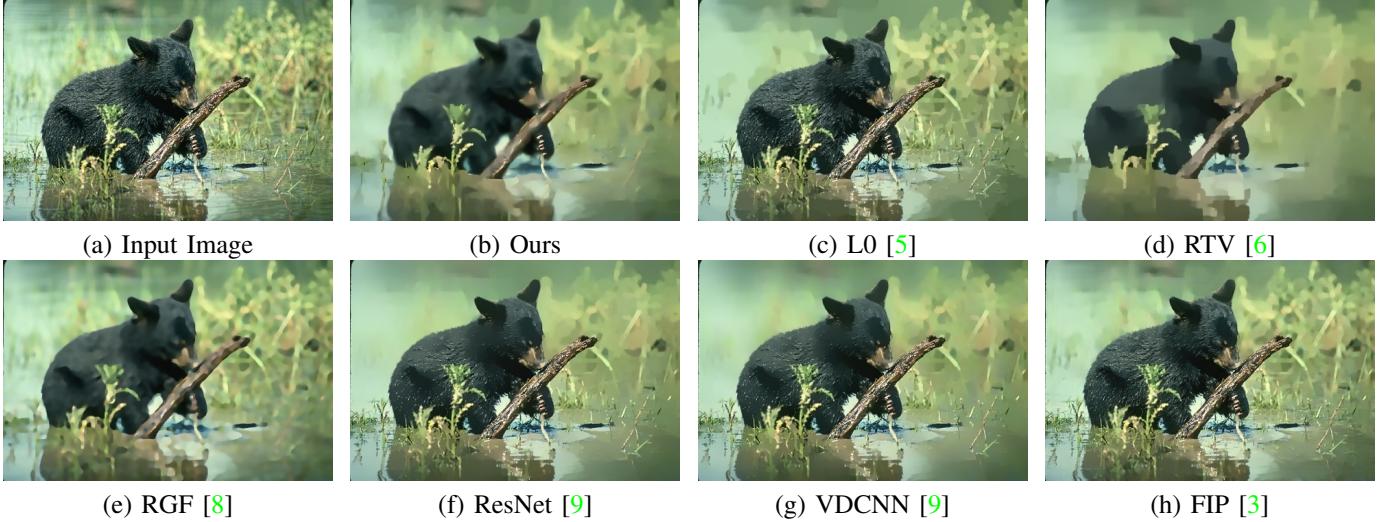


Fig. 18. Comparison of smoothed images by different methods on the image “0314” from the dataset in [9].

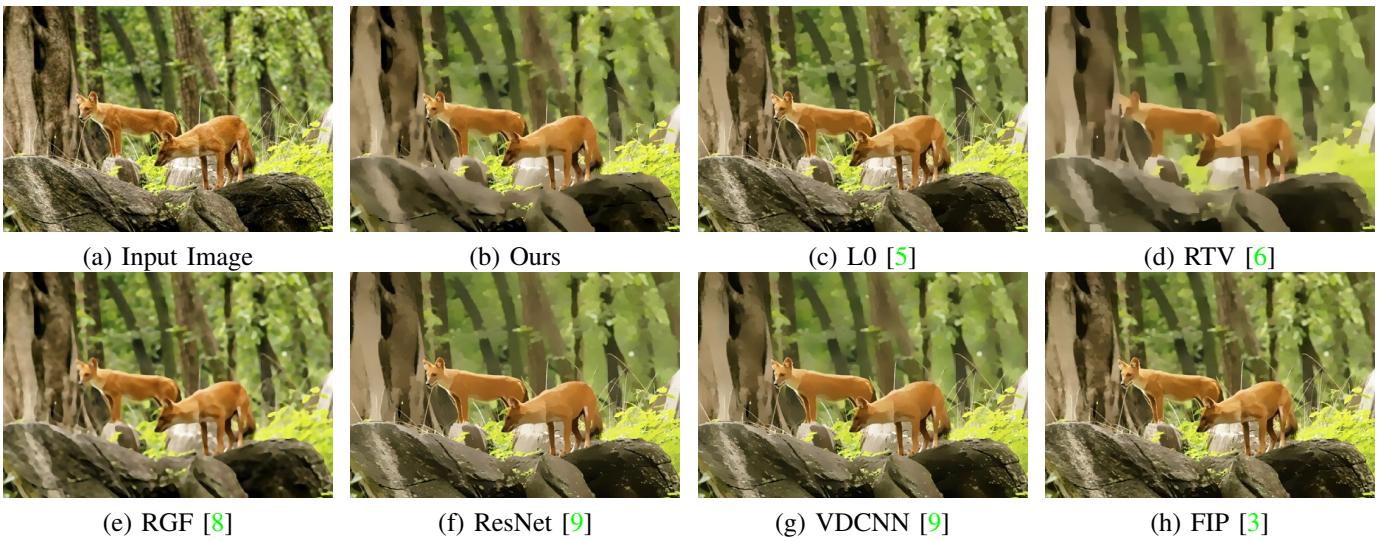


Fig. 19. Comparison of smoothed images by different methods on the image “0334” from the dataset in [9].

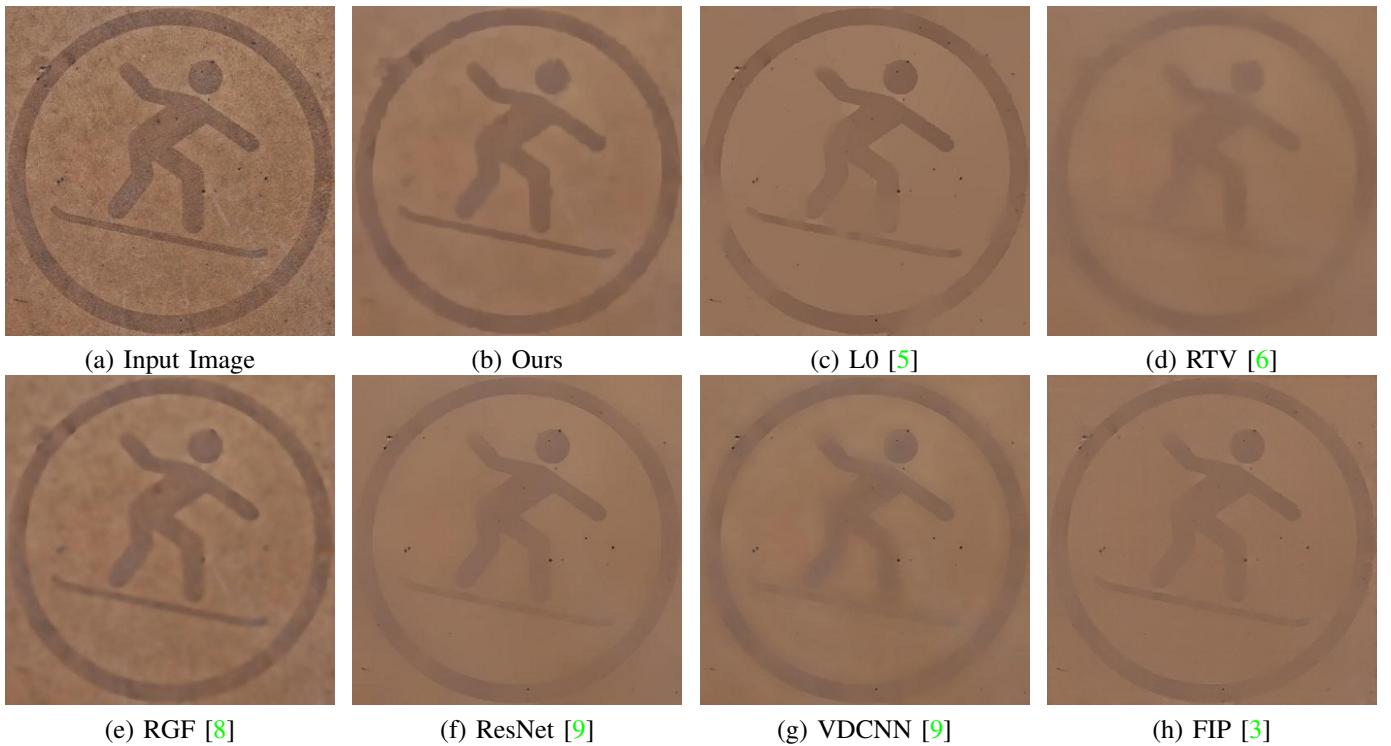


Fig. 20. Comparison of smoothed images by different methods on the image "11_07" from the dataset in [6]

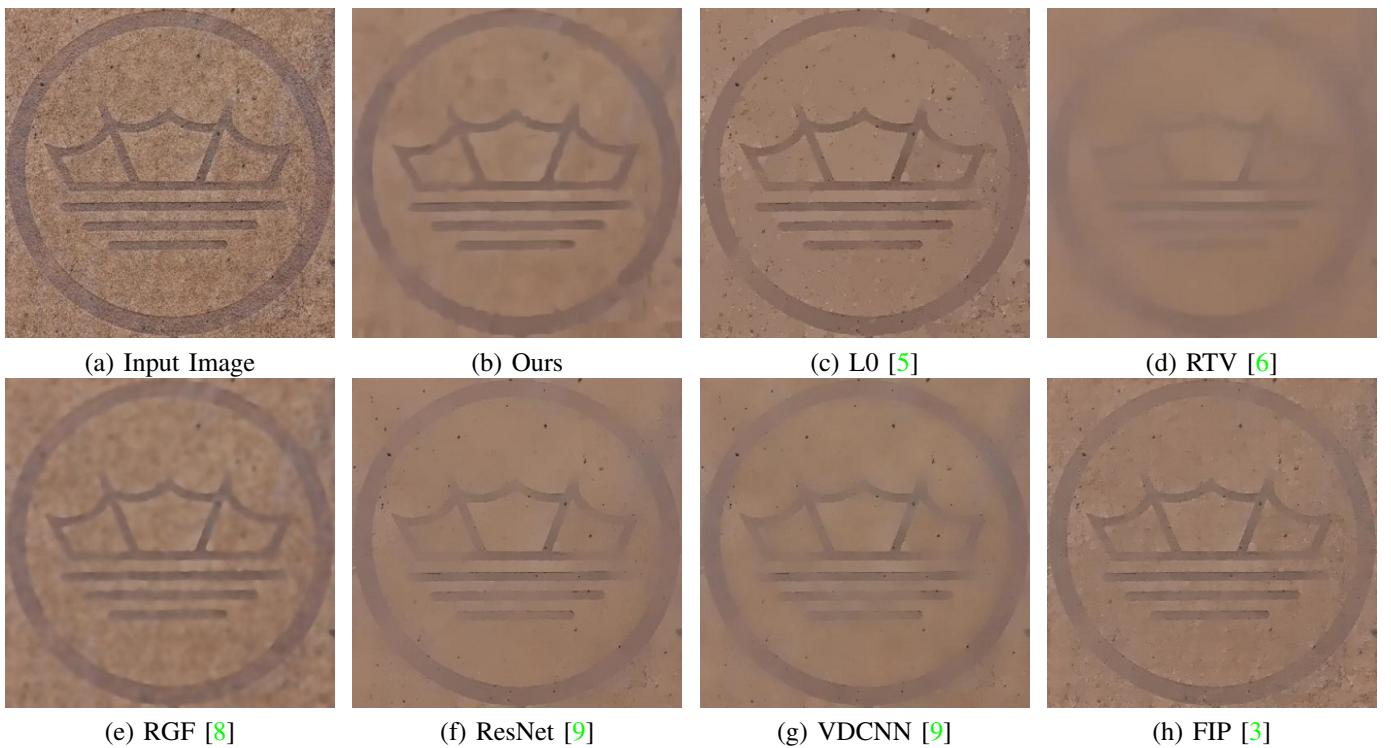


Fig. 21. Comparison of smoothed images by different methods on the image "11_08" from the dataset in [6]

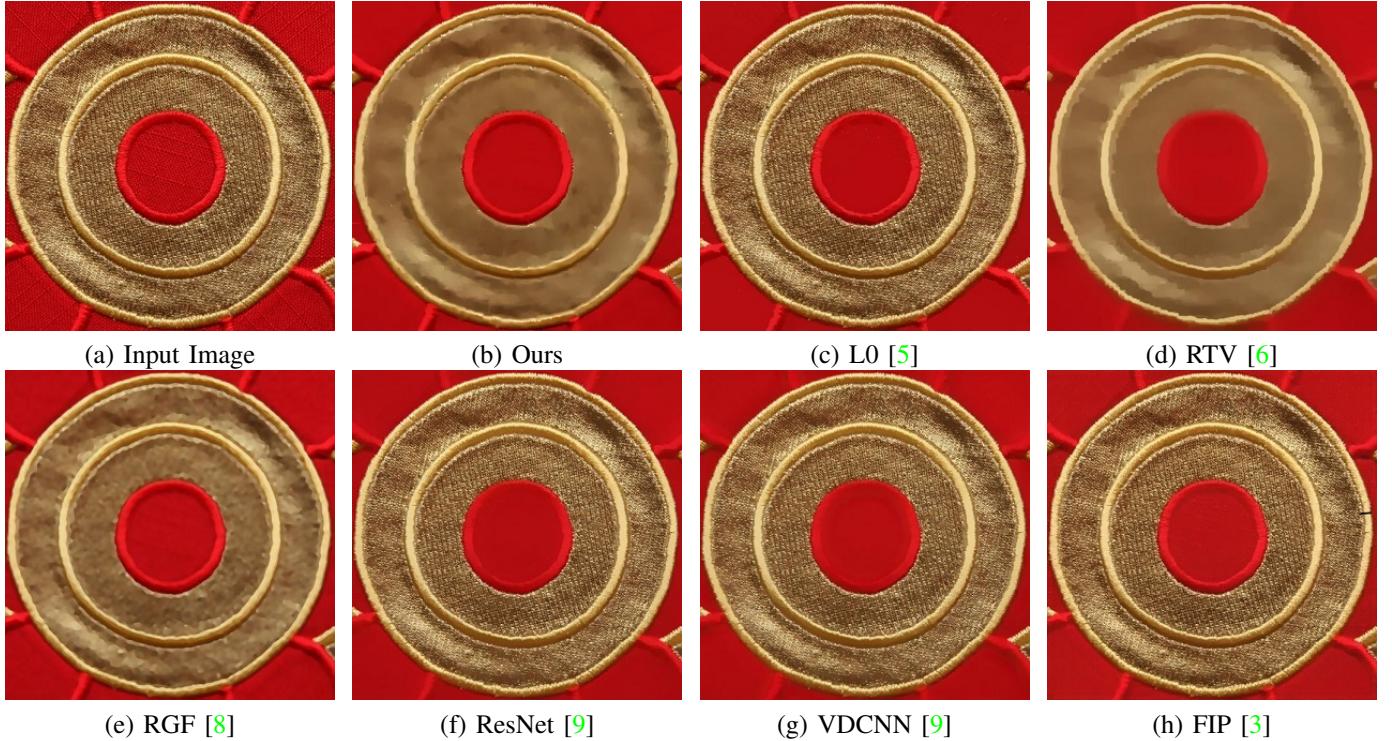


Fig. 22. Comparison of smoothed images by different methods on the image “11_11” from the dataset in [6]

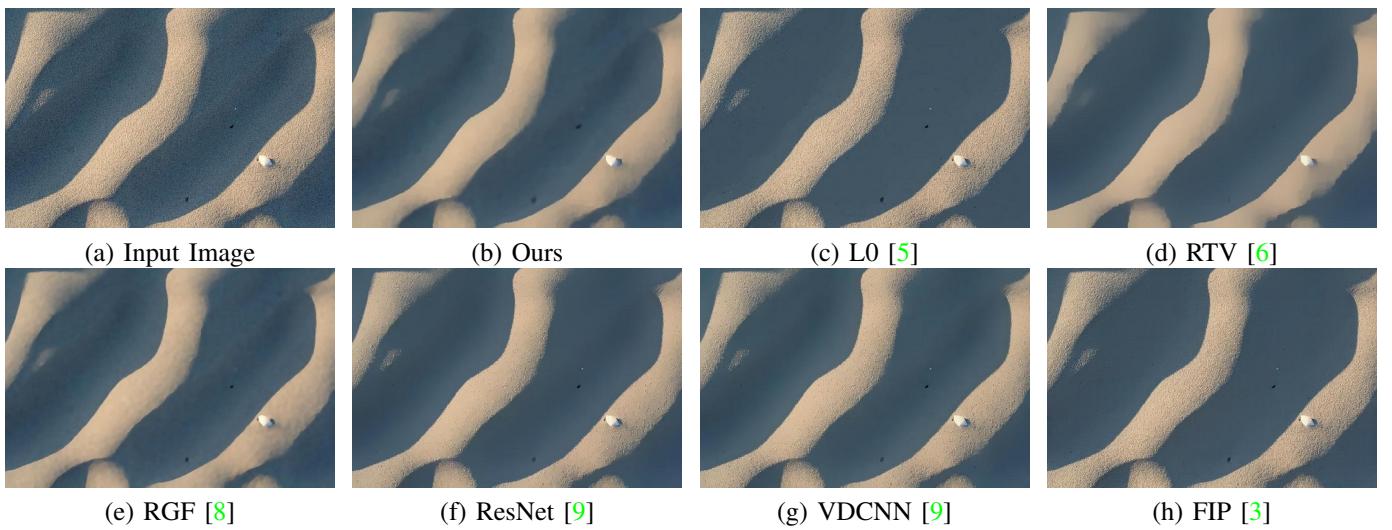


Fig. 23. Comparison of smoothed images by different methods on the image “11_26” from the dataset in [6]

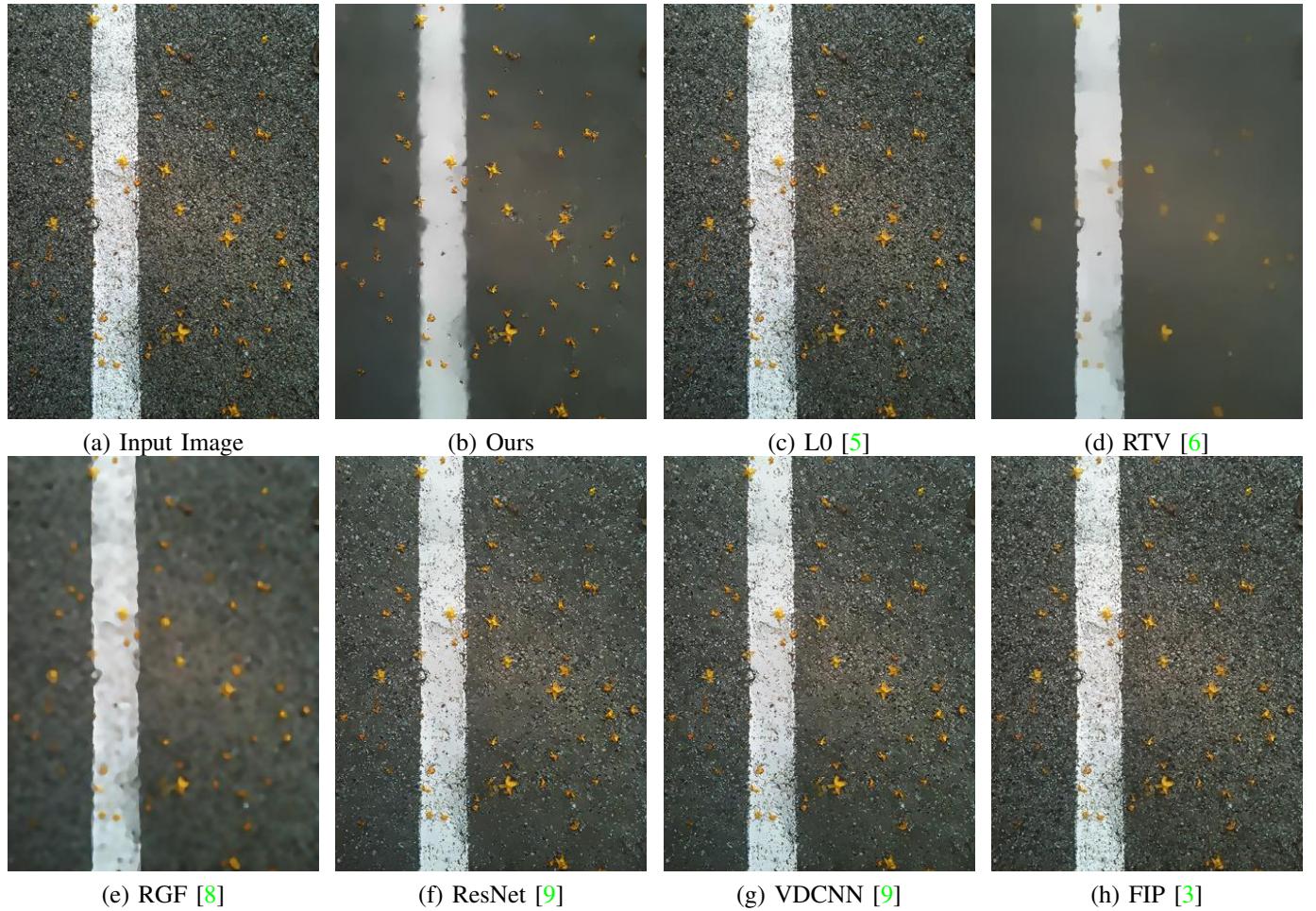


Fig. 24. Comparison of smoothed images by different methods on the image “12_53” from the dataset in [6]

REFERENCES

- [1] E. Agustsson and R. Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 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 Transactions on Image Processing*, 23(2):555–569, 2013. [3](#), [4](#)
- [3] Q. Chen, J. Xu, and V. Koltun. Fast image processing with fully-convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2497–2506, 2017. [3](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#)
- [4] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. [2](#)
- [5] L. Xu, C. Lu, Y. Xu, and J. Jia. Image smoothing via l_0 gradient minimization. *ACM Transactions on Graphics*, 30(6):174, 2011. [3](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#)
- [6] L. Xu, Q. Yan, Y. Xia, and J. Jia. Structure extraction from texture via relative total variation. *ACM Transaction on Graphics*, 31(6):139:1–139:10, 2012. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#)
- [7] L. Zhang, L. Zhang, X. Mou, and D. Zhang. Fsim: A feature similarity index for image quality assessment. *IEEE transactions on Image Processing*, 20(8):2378–2386, 2011. [2](#)
- [8] Q. Zhang, X. Shen, L. Xu, and J. Jia. Rolling guidance filter. In *ECCV*, pages 815–830, 2014. [3](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#)
- [9] F. Zhu, Z. Liang, X. Jia, L. Zhang, and Y. Yu. A benchmark for edge-preserving image smoothing. *IEEE Transactions on Image Processing*, 28(7):3556–3570, 2019. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#)