Fingerprinting Deep Image Restoration Models
(Supplemental Material)

1. Details of LGP and Color Histograms in Fingerprint Feature Comparison

The LGP operator [6, 7] assigns an integer code to each image pixel based on its neighboring local structure. Let $y_{\boldsymbol{c}}$ denote the pixel value at the spatial location $\boldsymbol{c}$. Consider a circle of radius R centered at $\boldsymbol{c}$ and take P sampling points along on the circle with a fixed order. The pixel values of those sampling points, denoted by $y_0, y_1, \cdots, y_{P-1}$, are obtained via bi-linear interpolation wherever necessary. Let $g_p = |y_p - y_c|$ and $\bar{g} = \frac{1}{P}\sum_{p=0}^{P-1} g_p$. The LGP code is defined as

$$\mathrm{LGP}_{P,R} = \sum_{p=0}^{P-1} s(g_p - \bar{g})2^p, s(x) = \begin{cases} 1, & x \geq 0, \\ 0, & x < 0. \end{cases} \tag{1}$$

The LGP code is indeed a binary string in the form of an integer. Such a bit string will be circularly shifted w.r.t. image rotation and may be sensitive to noise. Thus, borrowing the idea of uniform rotation-invariant transform [11], we enhance rotational invariance by taking the minimum value under bit-wise cyclic shifting and reduce noise sensitivity by eliminating the patterns with frequent bit-wise jumps. This leads to a uniform rotation-invariant version of LGP:

$$\mathrm{LGP}_{P,R}^{\mathrm{ri}} = \begin{cases} \min_k \mathcal{S}_k(\mathrm{LGP}_{P,R}), \text{ if } \mathcal{U}(\mathrm{LGP}_{P,R}) \leq u_0, \\ P+1, \text{otherwise}, \end{cases} \tag{2}$$

where $\mathcal{S}_k$ denotes the circular bit-wise right shift on the input by k times, and $\mathcal{U}$ is a uniformity measure that counts the number of bit-wise transitions from 0 to 1 or vice versa. The LGP is applied with $P = 10$, $R = 2$, $u_0 = 2$ and it results in a 12-dimensional LGP histogram. An 18-dimensional color (RGB) histogram is also used and thus we finally have a 30-dimensional feature vector of a fingerprint image.

2. Determining Value of σ for Model Ownership Verification

The reason we set $\sigma = 0.015$ is two-fold. First, similar to [33], we simply assume $\boldsymbol{h}_{\text{sou}}(j), \boldsymbol{h}_{\text{sus}}(j) \sim \mathcal{N}(\mu_0, \sigma_0^2)$ to facilitate hypothesis test. So $\boldsymbol{e}(j) \sim \mathcal{N}(0, \sigma^2)$ with $\sigma = \sqrt{2}\sigma_0$. Considering $\boldsymbol{h}_{\text{sou}}$ is implemented by a 30-dim normalized vector where $\boldsymbol{h}_{\text{sou}}(j)$ is around $1/30 = 0.033$ when it is uniformly distributed, we assume $\mu_0 = 0.033$ and $\sigma_0 = 0.011$ so that $\mu_0 \pm 3\sigma \in [0, 1]$. Here $\mu_0 \pm 3\sigma$ is considered due to the 3-sigma rule in statistics. Then we set σ to 0.015 which is around $\sqrt{2}\sigma_0$. Second, as $\boldsymbol{h}_{\text{sou}}(j), \boldsymbol{h}_{\text{sus}}(j) \in [0, 1]$, the Gaussian distribution of $\boldsymbol{e}(j)$ should be truncated into $[-1, 1]$. To approximate the truncated Gaussian distribution, one way is ensuring Pr[-1$\leq$ $\boldsymbol{e}(j)$ $\leq$1]$\approx$1, and $\sigma = 0.015$ satisfies it.

3. Details of Source Models

Denoising models Restormer, Nei2Nei, and DBSN are trained with synthetic noisy images, and DnCNN, NAFNet, and SimBase are trained with real-world noisy images. Specifically, Restormer is trained using synthetic noisy images from the BSD68 dataset [10] with white Gaussian noise whose level is drawn from the range $[0, 50]$. Note that BSD68 is often used a test set in existing literature, but here we use it as training data for evaluating the performance of fingerprinting. DnCNN is trained using the SIDD dataset [1]. The other four denoising models are trained using the data used in their own works.

SR models We use the pre-trained models released online for all the models. Among them, EDSR, RRDBNet, and RNAN are provided by [5], and the other three models are obtained from their official websites.

Independent Restormer models Restormer #1 is trained using synthetic noisy images from the BSD68 dataset of [10] with white Gaussian noise whose standard deviation is drawn from the range $[0, 50]$. Restormer #2∼#5 are trained using synthetic noisy images from the DIV2K [2], Flickr2K [8], WED [9] and BSD500 [3] datasets, with white Gaussian noise whose levels (*i.e.*, standard deviations) are set to 15, 25, 50, and drawn from the range $[0, 50]$, respectively. Restormer # 6 is trained on the real-world noisy images from the SIDD dataset [1].

4. Implementation Details for Additional Restoration Tasks

Image Deblurring The operator $\mathcal{D}_{\mathcal{T}}$ for image deblurring is defined as

$$\mathcal{D}_{\mathcal{T}}(\mathbf{X}) := \mathbf{K} \otimes \mathbf{X} + \mathbf{N},$$

where $\mathbf{K}$ denotes a blur kernel and $\mathbf{N}$ denotes the noise. For defocus blurring models, we define $\mathbf{K}$ as a 3×3 Gaussian kernel with standard deviation of 1 and draw $\mathbf{N}$ from $\mathcal{N}(0, 15/255)$. The λ is set 0.05 for fingerprint extraction. For motion deblurring models, we define $\mathbf{K}$ as a 9×9 vertical linear motion kernel and draw $\mathbf{N}$ from $\mathcal{U}(0, 0.1)$. The λ is set 0.1 for fingerprint extraction. See Figure 1 for the fingerprints extracted from three models of motion deblurring.

Low-light Image Enhancement We use an exponential transformation of power 3 and a min-max normalization for simulating low-light changes. Therefore, the operator $\mathcal{D}_{\mathcal{T}}$ for low-light image enhancement is defined as

$$\mathcal{D}_{\mathcal{T}}(\mathbf{X}) := \text{Norm}(\mathbf{X}^3),$$

where $\text{Norm}(\mathbf{X}) = (\mathbf{X} - \min(\mathbf{X}))/(\max(\mathbf{X}) - \min(\mathbf{X}))$.

Image Deraining The operator $\mathcal{D}_{\mathcal{T}}$ for image deraining is define as

$$\mathcal{D}_{\mathcal{T}}(\mathbf{X}) := \mathbf{X} + \mathbf{R},$$

where $\mathbf{R}$ denotes the synthetic rain layer. Following existing work, we generate the synthetic rain layer by convolving motion blur kernels with some points randomly sampled from a uniform distribution with a threshold of 0.995. The synthesized rain layer is then scaled down by 0.1 to reduce the intensities. The extracted fingerprints are shown in Figure 1, which exhibit distinctive patterns and remain similar after the pruning and quantization attacks.

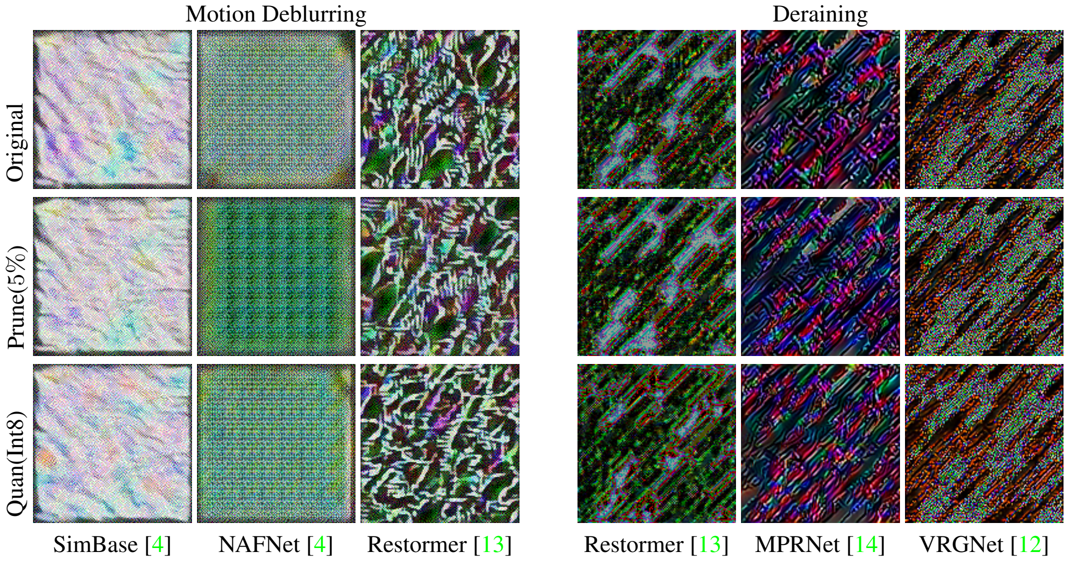


Figure 1: Fingerprints extracted from different image DNN models of two tasks.

5. Sensitivity Analysis on Initial Critical Images

To investigate the sensitivity of our fingerprinting approach to different initial critical images $\mathbf{S}^{(0)}$ sampled from a Gaussian distribution, we using different seeds in the Gaussian random generator to obtain different instances of $\mathbf{S}^{(0)}$ for calculating the fingerprints. As shown in Figure 2 on four models, the patterns of fingerprints are consistent across different instances of $\mathbf{S}^{(0)}$ for the same model. Moreover, we evaluate the robustness under pruning, fine-tuning, and quantization attacks on two models, with different instances of $\mathbf{S}^{(0)}$. The extracted fingerprints are shown in Figure 3. We can also observe that the changes of initial critical images have little impact on the extracted fingerprints under different attacks.

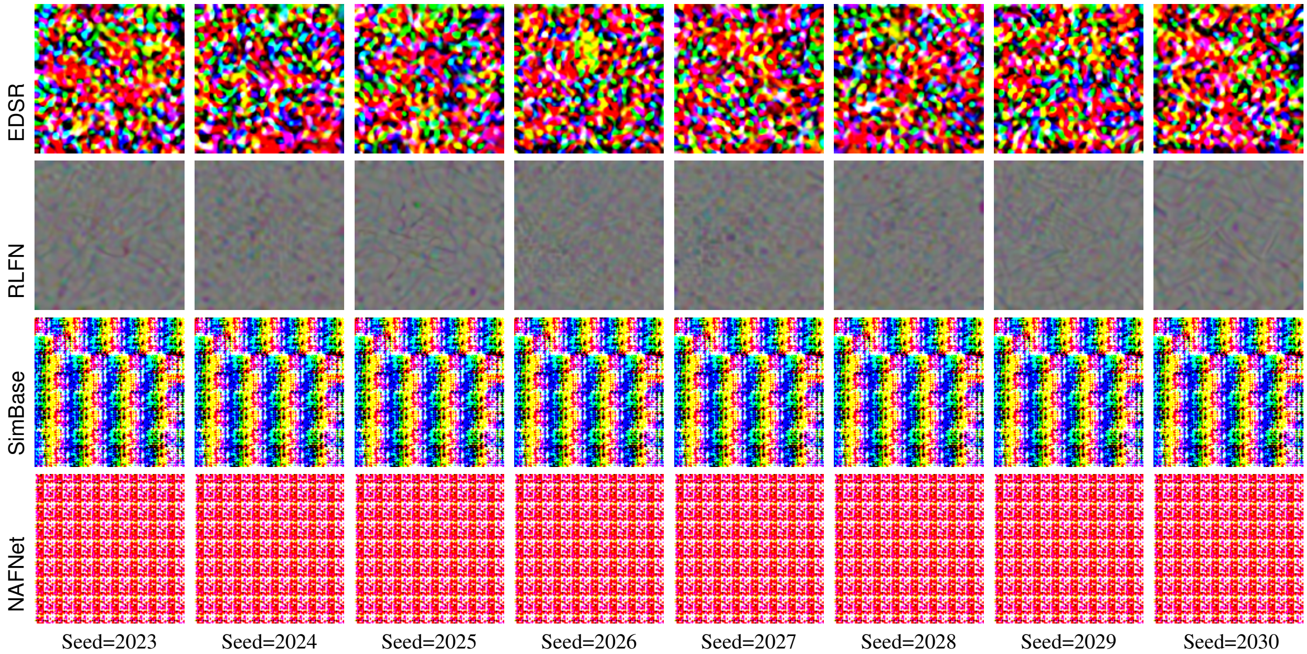
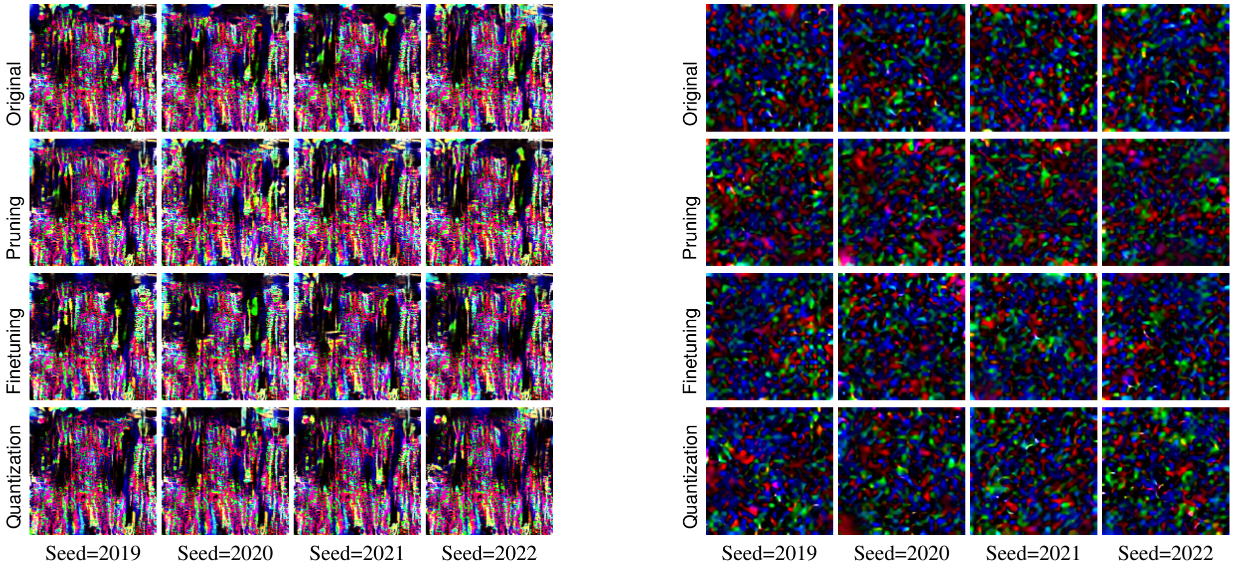


Figure 2: Fingerprints calculated using different instances of $\mathbf{S}^{(0)}$ obtained via different seeds.



(a) Fingerprints on Nei2Nei.

(b) Fingerprints on MobileSR.

Figure 3: Fingerprints calculated using different initialization seeds under various attacks.

6. Robustness Analysis under Finetuning Attacks with Significant Model Performance Decrease

The main paper has shown that our proposed fingerprinting approach is robust under the finetuning attack with 500 iterations (steps). We further examine the robustness under more iterations of finetuning, including 1.7k, 3.4k and 6.8k iterations. As the number of iterations increases, the performance of the attacked models changes more significantly. See Table 1 for the performance differences of five denoising models under finetuning with different numbers of iterations. For instance, the performances of all the models change 2.12dB in average under the finetuning with 6.8k iterations. Such significant changes may make the attacked models inapplicable in practice.

The extracted fingerprints are shown in Figure 4. Our approach produces consistent critical images for all source models under attacks with 1.7k iterations. The extracted fingerprints for SimBase, DBSN, Nei2Nei, and Restormer also keep similar under the attacks with 3.4k or 6.8k iterations. However, for NAFNet, the extracted fingerprint presents similar texture patterns but shows a different color compared to the original one under the finetuning attacks with 3.4k or 6.8k iterations. Note that in these case, NAFNet suffers from a significant PSNR drop of 1.6dB and 2.9dB, respectively. In conclusion, our approach is robust under finetuning attacks with reasonable performance changes, but may fail under extreme attacks that cause significant performance degradation of the model.

Table 1: PSNR difference(dB) of some denoising model under finetuning with different numbers of iterations.

#Iteration	SimBase	DBSN	Nei2Nei	NAFNet	Restormer	Avarage
1700	0.82	2.44	0.06	0.86	0.50	0.94
3400	1.58	2.55	0.57	1.64	0.81	1.43
6800	3.21	2.67	0.76	2.93	1.04	2.12

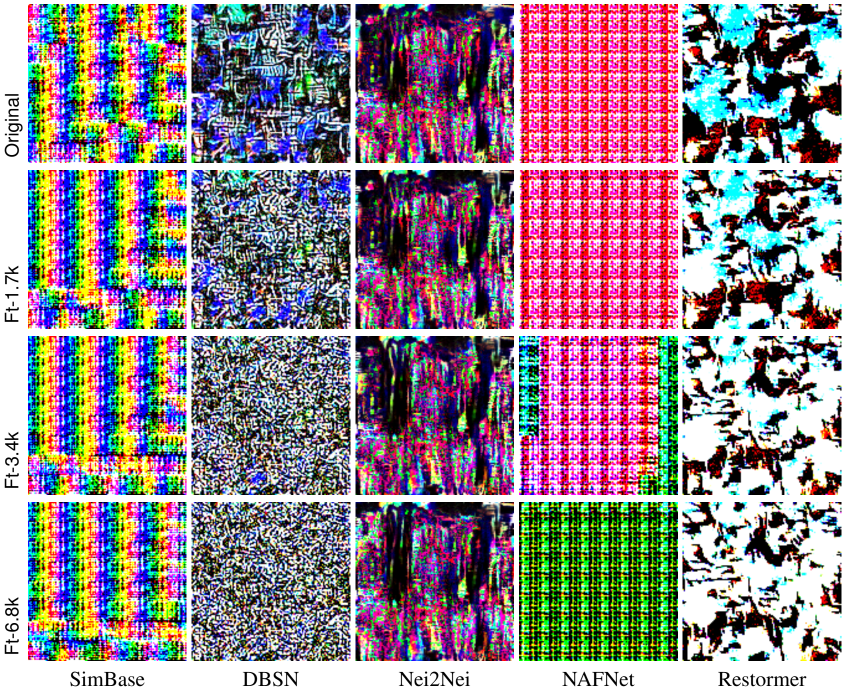


Figure 4: Fingerprints calculated from the denoising models under finetuning attacks with different numbers of iterations.

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, pages 1692–1700, 2018. 1, 2

[2] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of IEEE/CVF International Conference on Computer Vision Workshops*, pages 126–135, July 2017. 2

[3] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2010. 2

[4] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple Baselines for Image Restoration, Aug. 2022. 2

[5] Jinjin Gu and Chao Dong. Interpreting Super-Resolution Networks With Local Attribution Maps. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9199–9208, 2021. 1

[6] Bongjin Jun, Inho Choi, and Daijin Kim. Local transform features and hybridization for accurate face and human detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1423–1436, 2012. 1

[7] Bongjin Jun and Daijin Kim. Robust face detection using local gradient patterns and evidence accumulation. *Pattern Recognition*, 45(9):3304–3316, 2012. 1

[8] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced Deep Residual Networks for Single Image Super-Resolution. In *Proceedings of IEEE/CVF International Conference on Computer Vision Workshops*, pages 136–144, 2017. 2

[9] Kede Ma, Zhengfang Duanmu, Qingbo Wu, Zhou Wang, Hongwei Yong, Hongliang Li, and Lei Zhang. Waterloo exploration database: New challenges for image quality assessment models. *IEEE Transactions on Image Processing*, 26(2):1004–1016, 2016. 2

[10] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of IEEE International Conference on Computer Vision*, volume 2, pages 416–423, 2001. 1

[11] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002. 1

[12] Hong Wang, Zongsheng Yue, Qi Xie, Qian Zhao, Yefeng Zheng, and Deyu Meng. From rain generation to rain removal. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14791–14801, 2021. 2

[13] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient Transformer for High-Resolution Image Restoration. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5728–5739, 2022. 2

[14] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-Stage Progressive Image Restoration. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14821–14831, 2021. 2