Fingerprinting Deep Image Restoration Models

Yuhui Quan[1,2] Huan Teng[1,3] Ruotao Xu[1,2 *] Jun Huang[3] Hui Ji[4]
[1]School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China
[2]Pazhou Lab, Guangzhou 510335, China [3]Platform of AI, Alibaba Group, Hangzhou 311121, China
[4]Department of Mathematics, National University of Singapore, Singapore 119076

Abstract

Fingerprinting is a promising non-invasive method for protecting the intellectual property rights (IPR) of deep neural network (DNN) models. It extracts a feature called a fingerprint from a DNN model to identify its ownership. Existing fingerprinting methods focus only on classification-related models that map images to labels, while inapplicable to models for image restoration that map images to images. This paper proposes a fingerprinting framework for DNN models of image restoration. The proposed framework defines the fingerprint using a critical image, which exhibits strongly discriminative patterns and is robust to modest model modifications. Model ownership is then verified by comparing the distance of color histograms and local gradient pattern histograms of critical images between the suspect and source models. We apply the proposed framework to two representative tasks, denoising and super-resolution. It outperforms the baselines of fingerprinting and competes against existing invasive model watermarking methods.

1. Introduction

Deep learning has become a prominent tool for solving problems in computer vision, ranging from high-level image classification problems to low-level image restoration problems. However, the cost of designing and training a DNN model for specific applications has grown exponentially, leading to expenses in many areas such as hardware resources, data collection and labeling, and paying for engineers and researchers. Although sharing pre-trained DNN models has become a common practice in the community, many companies and institutes charge for the commercial usage of pre-trained models. This creates a strong incentive for adversaries to plagiarize/steal the models, using methods such as malware infection or internal leaks, to bypass the expensive training process. Consequently, both the community and companies have a strong motivation to protect the IPR of their DNN models.

One popular approach for protecting the IPR of DNN models is model watermarking (*e.g.* [44, 1, 10, 7, 53, 8, 39, 20, 54, 21]), which invasively embeds specific information called watermark into the source model and examines its existence in the suspect model for ownership verification. However, modifying model weights can potentially affect the model's utility, making it less desirable in practice.

Recently, a non-invasive approach called model fingerprinting has gained attention. Unlike watermarking, fingerprinting keeps the model intact and creates a unique feature called fingerprint from the model for identifying ownership. The ownership of a model is verified by comparing the fingerprint of the source model with that of the suspect model. While fingerprinting is still in its early stages with few existing works [3, 24, 11, 60, 33], it is gaining popularity as a non-invasive alternative to model watermarking.

1.1. Motivation

In this paper, we focus on protecting the IPR of DNN models used for low-level image restoration tasks. These types of models map degraded images or measurements to high-quality target images. Deep learning has proven to be a powerful tool for solving a variety of image restoration problems, *e.g.*, image denoising [57, 35, 36, 32], super-resolution (SR) [22, 47, 59, 43, 15], deblurring [58, 29, 40, 17, 41], and bad weather removal [34, 45, 37, 50, 51, 38].

Most existing works on IPR protection for DNN models focus on high-level vision tasks such as image classification, where the DNN outputs a label in a finite discrete set. In contrast, DNN models for image restoration output an image in a very high-dimensional space. Though there are a few works on watermarking DNN models for various image restoration tasks (*e.g.*[53, 39, 54]), these image restoration models can be vulnerable to modifications of DNN weights required in watermarking, resulting in the loss of important

*Corresponding author: Ruotao Xu (xrt@scut.edu.cn). This work is supported by the Natural Science Foundation of Guangdong Province (Grant No. 2023A1515012841, 2022A1515011755, 2022A1515011087); National Natural Science Foundation of China (Grant No. 62106077); Postdoctoral Special Foundation of China (Grant No. 2022T150219); Postdoctoral Foundation of China (Grant No. 2020M682705); and Singapore MOE AcRF Tier 1 (Grant No. A-8000981-00-00).

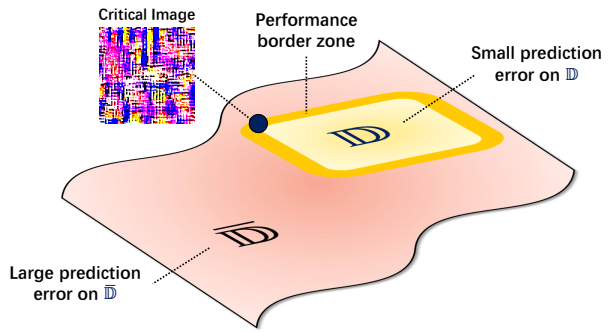


Figure 1: Main idea of our model fingerprinting approach for an image recovery DNN. We aim to find a critical image (blue point) around the DNN's performance border zone.

image details and the addition of undesired artifacts.

Fingerprinting is a more attractive option for image restoration DNNs, as it does not require any modification to model weights. However, to the best of our knowledge, no existing study has explored fingerprinting for DNN models used in image restoration tasks. The existing fingerprinting methods are designed for classification DNNs which can be fully characterized by their decision boundaries. These methods use adversarial examples to characterize the decision boundaries and generate a fingerprint for the model. In contrast, image restoration generates an image in a high-dimensional space, and its accuracy is measured by a continuous metric such as mean squared error to the ground truth. Therefore, the concept of "decision boundary" is not well-defined for image restoration models.

Due to the significant differences between image restoration and image classification, the existing fingerprinting methods for classification DNNs cannot be easily extended to image restoration DNNs. As a result, we were motivated to develop a new fingerprinting framework specifically designed for image restoration DNN models.

1.2. Main Idea

An image restoration task (*e.g.* denoising and SR) is to predict a latent image $\mathbf{X}$ from its degraded or partial measurements $\mathbf{Y} = \mathcal{D}(\mathbf{X})$, where $\mathcal{D}$ denotes the degrading/measuring process. In deep learning, a DNN model $\mathcal{M} : \mathbb{R}^{n\times m} \to \mathbb{R}^{N\times M}$ is trained to accurately predict $\mathbf{X}$ from the input $\mathcal{D}(\mathbf{X})$ in some task-related domain $\mathbb{D}$. While the prediction error for most images in $\mathbb{D}$, measured by $\|\mathbf{X} - \mathcal{M}(\mathcal{D}(\mathbf{X}))\|_2^2$, is small, there exists a complementary set $\bar{\mathbb{D}}$ where the DNN has large prediction errors.

Indeed, an image restoration DNN usually works well only for target images that lie within a low-dimensional manifold in $\mathbb{R}^{N\times M}$. It is not expected that a DNN will perform well for all points in $\mathbb{R}^{N\times M}$. For instance, consider image denoising where $\mathbf{Y} = \mathcal{D}(\mathbf{X}) = \mathbf{X} + \mathbf{N}$, with $\mathbf{N}$ representing Gaussian white noise. Let $\mathbf{X}_1$ be a constant matrix and $\mathbf{X}_2$ be a random matrix with entries randomly sampled from a normal distribution. Since adding two independent Gaussian random variables results in a new Gaussian random variable [18], $\mathbf{Y}_2$ also corresponds to a constant image corrupted by larger Gaussian white noise. Suppose a denoising model performs well on predicting $\mathbf{X}_1$ (a constant matrix) from its noisy version $\mathbf{Y}_1$. Then, the denoiser will consistently take $\mathbf{Y}_2$ as the noisy version of a constant matrix and output a constant matrix, which is far away from the ground-truth matrix $\mathbf{X}_2$ (a random matrix). In other words, a denoising model designed to work well for constant matrices may not work well for random matrices.

To summarize, just like the decision boundary in classification, there exists a "performance border zone" that separates the set of images that the restoration model can accurately restore and the set of images that it cannot. This zone encodes essential characteristics of the model, and the points close to it can be used for its characterization. Particularly, these points can be the images in the complementary set $\bar{\mathbb{D}}$ that are close to the set $\mathbb{D}$ of images that the DNN can restore accurately (*i.e.*, their prediction errors by the DNN are small). See Figure 1 for an illustration of this concept.

In this paper, we define the term "critical images" to refer to the points near the performance border zone. Suppose that the target images in $\mathbb{D}$ follow some prior, such as the sparsity prior of gradients. Then, a critical image for the model $\mathcal{M}$ is an image $\mathbf{X}$ that minimizes the prediction error $\|\mathbf{X} - \mathcal{M}(\mathcal{D}(\mathbf{X}))\|_2^2$ (i.e., $\mathbf{X}$ is close to $\mathbb{D}$) while also maximizing a penalty that reflects the prior on the images in $\mathbb{D}$ (i.e., $\mathbf{X}$ is close to $\bar{\mathbb{D}}$). This ensures that $\mathbf{X}$ is simultaneously close to both $\mathbb{D}$ and $\bar{\mathbb{D}}$, i.e., it lies near their border zone.

Empirical observations suggest that the critical images obtained from independently-trained DNNs using the proposed approach consist of strong spatial patterns that not only convey discriminative information, but are also robust to moderate modifications in model weights. Therefore, these critical images can serve as the fingerprints of a model. For model verification, we measure the distances between the fingerprints using two standard image descriptors: color histograms and local gradient patterns (LGP) [13].

1.3. Contributions

The construction of a critical image forms the foundation of our proposed fingerprinting approach for image restoration models. Our approach is applied to two representative image restoration tasks: denoising and SR. Denoising is a core problem in many image restoration tasks, and SR is one of the most successful applications of deep learning in image restoration. Extensive experiments on these two tasks have demonstrated that our proposed approach is more robust than the baseline fingerprinting methods and competitive with recent intrusive watermarking methods. See below for a summary of the main contributions:

- A fingerprinting framework is proposed for protecting the IPR of image restoration DNN models. To the best of our knowledge, this is the first study on fingerprinting DNN models of image restoration.
- Analogous to decision boundaries in classification, the concept of performance border zone is introduced to characterize image restoration DNN models.
- The concept of critical images, points in proximity to the performance border zone, is introduced to showcase discriminative and robust patterns that can serve as effective fingerprints for image restoration DNN models.

2. Related Works

2.1. Model Fingerprinting

Model fingerprinting shares a similar spirit with the zero-watermarking for digital images (*e.g*. [48]), where discriminative yet robust features are extracted from an image to represent its ownership. While it is possible to extract features directly from the weights of a DNN by treating them as digital media, this method is not robust to modifications of the model. To improve robustness, existing works [3, 24, 11, 60, 60, 33] extract features from the decision boundaries of a classification DNN, as these features are often transferable for plagiarism models but not for independently-trained models.

Cao *et al*. [3] proposed to fingerprint decision boundaries via using nearby data points defined by adversarial examples. The ownership is verified by checking whether the suspect model predicts the same labels as the source model on those data points. Lukas *et al*. [24] proposed to fingerprint the overlap of adversarial subspaces around decision boundaries between the source model and its surrogates. They synthesized conferrable adversarial examples that transfer exclusively with a target label from the source model to its surrogates. Zhao *et al*. [60] proposed to improve adversarial examples for fingerprinting by encouraging them to mimic the logits vector of a target sample randomly chosen from the target category. Peng *et al*. [33] proposed to profile decision boundaries by characterizing the universal adversarial perturbations [26]. Additionally, they trained an encoder via contrastive learning to map fingerprints from two models to a similarity score for ownership verification. Chen *et al*. [5] proposed to quantify the similarity between two models using a diverse set of testing metrics and test case generation algorithms to produce a chain of evidence for verification. This method can include many existing fingerprinting algorithms as test metrics.

All the methods discussed above are limited to classification DNNs. Our approach, however, is specifically designed for fingerprinting models of image restoration by using critical images instead of adversarial examples. It is noted that while the aforementioned methods aim to detect model plagiarism, He *et al*. [11] proposed a fingerprinting method to examine model integrity. This method identifies a small set of human-unnoticeable transformed inputs that make a model's outputs sensitive to its parameters.

2.2. Model Watermarking

Model watermarking serves a similar purpose as model fingerprinting; however, it involves embedding a code or opening a backdoor inside a DNN that could potentially harm the DNN's performance. Most existing watermarking methods are designed for classification DNNs (*e.g*. [44, 42, 56, 1, 10, 27, 4, 28, 7, 16, 53, 8, 20, 54, 21, 19, 23]), but there are also a few works that target image restoration DNN models (*e.g*. [53, 39, 31, 54]). Zhang *et al*. [53, 54] proposed to train the DNN to automatically embed an invisible watermark in its output image. The model ownership is verified by detecting the presence of watermarks in the output images of the suspect DNN. Quan *et al*. [39] proposed to train the DNN to map a random image to its naive recovered version that is unlikely to be generated by other independently well-trained DNNs. The model ownership is verified by checking whether such a mapping holds for the suspect DNN. Ong *et al*. [31] proposed to train the DNN so that an input image embedded with a visible key can lead to an output image with a visible logo. The model ownership is verified by attempting to trigger the logo via the key.

3. Methodology

3.1. Problem Statement and Overview

Threat model In a typical attack-defense scenario, an owner has trained a source model using private resources. An adversary attempts to plagiarize or steal the model. The owner is both a victim and a defender, with the goal of determining whether the suspect model is a plagiarized one. This involves verifying ownership in a white-box setting. An adversary or attacker who steals a model may modify it to avoid detection of ownership while maintaining its functionality and performance, but the access to the original training data is not given. Under this setting, we consider often-seen attacks that modify model weights, *e.g*., pruning, finetuning, and quantization.

Principles Discriminability is a fundamental property of model fingerprinting. A model fingerprint is trustworthy for protecting IPR only if it is distinguishable for different models. Given the importance of training data and open sources of DNN architecture, two models with the same architecture but trained independently using different data are considered as independent models, and their fingerprints should be distinguishable from each other. Robustness is another critical property of model fingerprinting. A model fingerprint is expected to remain nearly unchanged under various modifications (attacks) on the model. However, these two

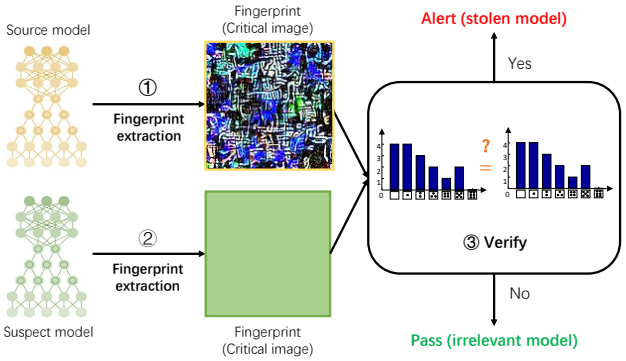


Figure 2: Proposed model fingerprinting framework.

properties often conflict with each other, and an ideal model fingerprinter should balance them carefully. It is worth noting that the fidelity of model performance, a desirable property for model watermarking, is not necessary for a non-invasive method like fingerprinting.

Framework The proposed fingerprinting method for image restoration DNN models is outlined in Figure 2, involving 3 steps. First, a fingerprint defined over critical images is extracted from the source model and registered. Second, the fingerprint of a suspect model is computed for notarization. Finally, ownership is verified by comparing the features of the two fingerprints. The remaining issues are (i) how to extract a discriminative yet robust fingerprint from a model, and (ii) how to efficiently compare two fingerprints.

3.2. Fingerprint Extraction

As discussed in Section 1.2, we use critical images to fingerprint a model. These critical images are located near the performance border zone of an image restoration DNN. Now we focus on the restoration problems for natural images which are assumed to follow a Laplacian prior. Such a prior is generally true for natural images. Specifically, the total variation (TV) $|\nabla \mathbf{X}|_1$ of an image $\mathbf{X}$ in the domain $\mathbb{D}$ is small, while an image $\overline{\mathbf{X}}$ in $\overline{\mathbb{D}}$ has a large value of $|\nabla \overline{\mathbf{X}}|_1$. Our proposed approach is also applicable to images in other domains, provided that there is some prior knowledge available for latent images.

Given a DNN model $\mathcal{M}$ trained for an image restoration task $\mathcal{T}$, we define the critical image $\mathbf{S}$ by solving the following optimization problem:

$$\mathbf{S} := \underset{\overline{\mathbf{X}} \in [0,1]^{M \times N}}{\arg\min} \; \|\overline{\mathbf{X}} - \mathcal{M}(\mathcal{D}_{\mathcal{T}}(\overline{\mathbf{X}}))\|_2^2 - \lambda \|\nabla \overline{\mathbf{X}}\|_1, \quad (1)$$

where $\mathcal{D}_{\mathcal{T}}$ denotes a degradation operator associated with the task $\mathcal{T}$, and $\lambda \in \mathbb{R}^+$ is a weight. The objective of the first term in (1) is to find an image close to the set $\mathbb{D}$. The second term encourages the image to be close to $\overline{\mathbb{D}}$. This is achieved by maximizing the value of $\|\nabla \overline{\mathbf{X}}\|_1$, which is a necessary condition for images in $\overline{\mathbb{D}}$.

The operator $\mathcal{D}_{\mathcal{T}}$ varies depending on the specific image restoration task. For instance, it corresponds to noise corruption for image denoising, downsampling for image SR, and blurring for image deblurring. For image denoising and SR, we adopt the following construction schemes for $\mathcal{D}_{\mathcal{T}}$:

$$\mathcal{D}_{\mathcal{T}}(\mathbf{X}) := \mathbf{X} + \mathbf{N}, \;\; \mathbf{N} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}), \quad \text{[Denoising]} \quad (2)$$

$$\mathcal{D}_{\mathcal{T}}(\mathbf{X}) := \mathbf{X}\downarrow_4, \quad \text{[Image SR]} \quad (3)$$

where $\mathcal{N}$ denotes normal distribution and $\downarrow_4$ denotes dyadic downsampling that reduces the image size by half. The definition of $\mathcal{D}_{\mathcal{T}}$ can be adapted to other restoration tasks.

To solve (1), we use the Adam solver [14]. However, since the problem involves a DNN, it is highly non-convex, and the output of an iterative solver depends on the initialization of $\mathbf{X}$. We use random initialization for solving (1), i.e., we set $\mathbf{S}^{(0)}$, the initial point of $\mathbf{S}$, as follows:

$$\mathbf{S}^{(0)} \sim \mathcal{N}(\mathbf{0}, \beta^2 \mathbf{I}), \;\; \beta \in \mathbb{R}. \quad (4)$$

Figure 3 illustrates the critical images calculated using different instances of $\mathbf{S}^{(0)}$ generated via (4). We observe that critical images from the same model with different $\mathbf{S}^{(0)}$ exhibit similar (homogeneous) patterns, while those from different models differ significantly from each other.

Next, we construct the fingerprint denoted by $\mathcal{F}$ via (1). To form a fingerprint of arbitrary length, one can solve (1) for multiple times using different $\mathbf{S}^{(0)}$:

$$\mathcal{F} = \{(\mathbf{S}_1^{(0)}, \mathbf{S}_1), \cdot, (\mathbf{S}_K^{(0)}, \mathbf{S}_K)\}, \quad (5)$$

where $\mathbf{S}_k$ is the critical image calculated using $\mathbf{S}_k^{(0)}$ as the initial point for solving (1). Note that different $\mathbf{S}_k$ of the same model contain the same patterns (see Figure 3) and image restoration models usually allow input (output) images of varying sizes. To obtain a similar amount of fingerprint information while avoiding multiple extractions, we can set $K = 1$ and use a single large $\mathbf{S}$ to approximate multiple $\mathbf{S}_k$ of smaller sizes. It is worth noting that solving $\mathbf{S}_k$ from (1) does not require accessing the model weights of $\mathcal{M}$, but only the gradients of $\mathcal{M}(\mathcal{D}_{\mathcal{T}}(\overline{\mathbf{X}}))$ w.r.t. $\overline{\mathbf{X}}$, in a similar spirit to federated learning.

3.3. Verification of Ownership

Verification of ownership is accomplished by comparing fingerprints between the source and suspect models. While critical images can be visually compared, automated verification is often desirable in practice. Because critical images display distinct patterns in terms of color distribution and texture, we characterize them using color histograms (a standard color descriptor) and an improved version of LGP histograms [13] (a classic texture descriptor) based on the idea of [30]. Further details can be found in the supplemental material. Our approach is learning-free and does not require collecting training data for a learning-based verifier.

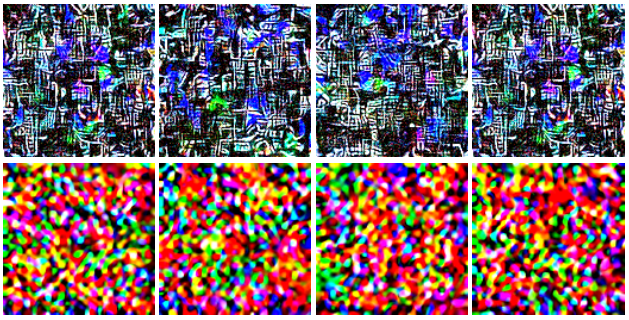


Figure 3: Critical images of DBSN [49] (upper) and EDSR [22] (bottom) calculated by using $\mathbf{S}^{(0)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ while fixing all other hyper-parameters.

Let $\boldsymbol{h}_{\text{sou}}, \boldsymbol{h}_{\text{sus}} \in \mathbb{R}^L$ be the features. *i.e.*, the concatenation of normalized color and LGP histograms, computed from the fingerprints of the source model and the suspect model, respectively. When $K > 1$, the features from multiple $\mathbf{S}_k$ are averaged. In essence, these features serve as a refined version of the fingerprints. Then the suspect model will be identified as plagiarism if

$$d = \|\boldsymbol{h}_{\text{sou}} - \boldsymbol{h}_{\text{sus}}\|_2^2 / L < \eta, \tag{6}$$

where η is a threshold (the bound of negligible error) determined by a probabilistic scheme with the same spirit of [39]. Suppose that the error $\boldsymbol{e}(j) = \boldsymbol{h}_{\text{sou}}(j) - \boldsymbol{h}_{\text{sus}}(j) \sim \mathcal{N}(0, \sigma^2), \forall j$, where $\sigma = 0.015$ (see supplemental material for our idea to determine this value). Then, the random variable $Z = \frac{\|\boldsymbol{e}\|_2^2}{\sigma^2}$ follows a Chi-squared distribution $\mathcal{X}_L^2$. By applying the p-value approach with $p < 0.05$, we can find a value of γ such that $P[Z \leq \gamma] < 0.05$, or equivalently, $P[d \leq \eta] < 0.05$. This allows us to safely reject the null hypothesis that $\boldsymbol{h}_{\text{sou}}$ and $\boldsymbol{h}_{\text{sus}}$ are similar.

4. Experiments

We assess the effectiveness of the proposed approach for two restoration tasks, namely image denoising and image SR, by measuring their discriminability and robustness.

4.1. Experimental Setup

Source models We choose six DNNs from existing literature as the source models for each task, respectively. (a) Image denoising: DnCNN [57], DBSN [49], Nei2Nei [12], Restormer [50], SimBase [6], and NAFNet [6]; (b) Image SR: EDSR [22], RRDBNet [47], RNAN [59], MobileSR [43], RFLN [15], and DRLN [2]. To comprehensively evaluate the effectiveness of fingerprinting, we consider models with varying structures, training data, and training strategies, for both tasks. We either use the pre-trained models or train them using their released codes. See more details in the supplemental material.

Implementation details In fingerprint extraction, we use a single critical image of size 128×128, for all models in both tasks. We set $\beta = 1$ in (4) for initializing $\mathbf{X}$. The learning rate of Adam for solving (1) is set to 0.1. A total number of 5000 iterations is used. In ownership verification, we form a 30-dimensional RGB+LGP histogram. The resulting threshold is $\eta = 1.39 \times 10^{-4}$. For denoising DNNs, we set $\sigma = 1$ for constructing $\mathcal{D}_{\mathcal{T}}$ in (2). For both denoising and SR, we set $\lambda = 0.001$. The seeds involved in our approach are fixed to make the whole process deterministic and reproducible.

4.2. Discriminability Analysis

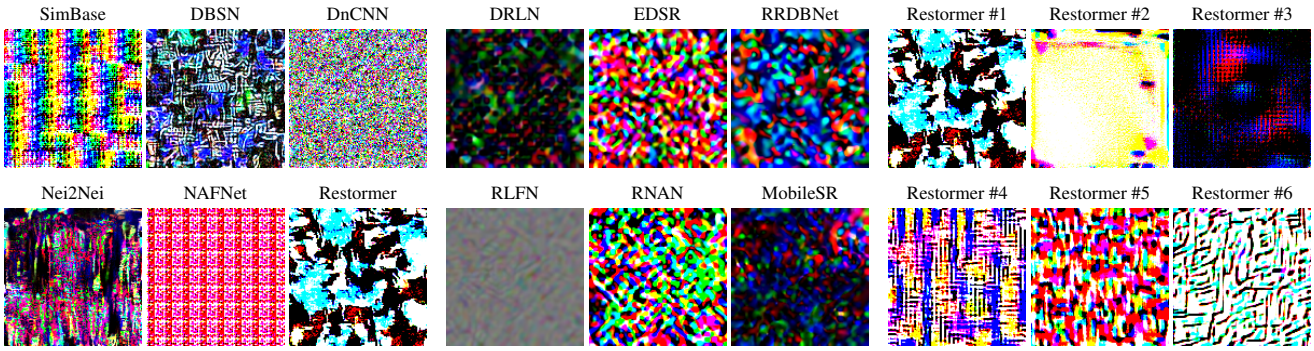
We begin by a visual inspection on the fingerprints extracted via our approach. See Figure 4 for the critical images extracted from various source models, all of which contain unique and distinguishable patterns that vary in scale, appearance, local shape, regularity and color, across different images. For instance, larger-scale patterns are visible for SimBase while the smaller-scale ones for DnCNN. The fingerprint of DnCNN exhibits random patterns, while that of NAFNet shows regular ones. Moreover, the red tone of fingerprint of NAFNet differs from the chromatic tone of SimBase. These significant visual differences demonstrate the strong discriminability of the extracted fingerprints.

The discriminability is further quantitatively assessed by computing the feature distances (6) between critical images for every pair of source models. The distance matrices in Figure 5(a) and (b) show the ratios between the resulting distances and the threshold η, for denoising DNNs and SR DNNs, respectively. In both tasks, all off-diagonal elements, *i.e.*, distances between every model pair, exceed the threshold, indicating no confusion among the models. The results indicate excellent discriminability of our extracted critical images, and the simple RGB+LGP histograms are able to effectively capture their patterns for verification.

We also test whether our approach can differentiate models with the same architecture but trained on different datasets. The Restormer model [50] is retrained on six different datasets; see the details in the supplemental material. Figure 4(c) visually compares the fingerprints of each model, and Figure 5(c) shows the resulting distance matrix. Our approach effectively differentiates the models, even those with the same architecture, with many large off-diagonal values observed in the distance matrix.

4.3. Robustness Analysis

We conduct robustness analysis using three common attacks: model pruning, model finetuning, and model quantization. A meaningful attack should only have a minor effect on the original function of a model. For analysis, we quantify the impact of an attack on the model's performance based on the resulting average PSNR gap on the recovered



(a) Fingerprints of denoising models (b) Fingerprints of SR models (c) Fingerprints of independent Restormer models

Figure 4: Fingerprints (critical images) extracted from different source models.

(a) Denoising models (b) SR models (c) Restomers trained with different datasets.

Figure 5: Visualization of pairwise feature distance of fingerprints between source models. The ratios between the distance and the threshold are shown. For better illustration, all the color bars are set as follows. The color goes from purple to blue when the value decrease from 1, while the color goes from orange to yellow when the value increase from 1.

images. Each attack has its strength controlled such that it only leads to $\leq$ 1dB change in the average PSNR value for all the models on a test set. This ensures that all attacks are not meaningless. The robustness is measured by the times of successful ownership verification under an attack.

Baselines So far, no fingerprinting method is available that is specifically designed for image restoration DNNs, making our work the first in this area. For experimental comparison, we construct two baselines by adapting existing fingerprint methods designed for classification DNNs:

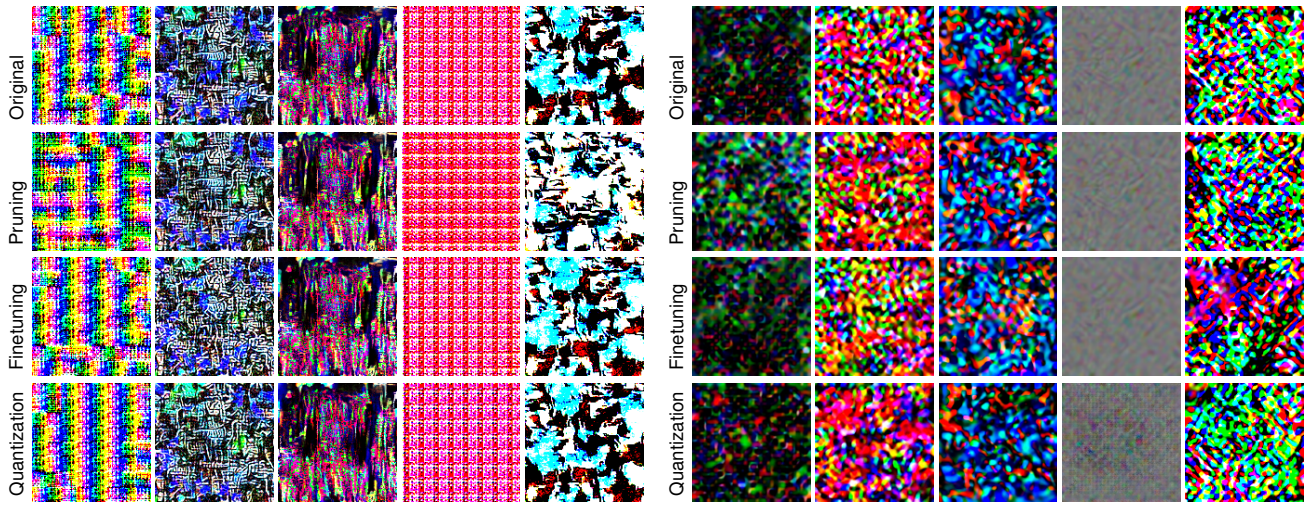
1. ProjCL: Each image restoration source model is converted to a classification DNN by adding a random projection layer with softmax activation in the end, so as to output a 100-dimensional label vector. Then, the well-known fingerprinting method IPGuard [3] is applied for the classification DNNs.
2. PoolCL: The output of each source model is downsampled to form a 10×10 image, which is then vectorized and subjected to a Softmax function, producing a 100-dimensional label vector. This converts each source model into a classification DNN, and then we fingerprint it using the IPGuard [3].

We tune the baselines' hyper-parameters, including label vector dimensions, for their discriminability. We set their verification thresholds to the minimum values that ensure all tested models pass the discrimination test. These threshold values optimize the robustness metrics, as reducing the verification threshold improves robustness.

Furthermore, we present three watermarking methods specifically designed for image restoration DNNs to compare the performance gap between non-invasive fingerprinting and invasive watermarking:

1. Zhang *et al*. [55]: The source model is retrained to embed an invisible watermark in each output.
2. Ong *et al*. [31]: The source model is retrained to composite a visible watermark region into the output when the input is composed of a specific trigger region.
3. Quan *et al*. [39]: The source model is retrained to produce a copyright image for a specific input noise.

We tune the hyper-parameters for the watermark embedding in these methods to maintain high fidelity on the PSNR performance of the source models on test data. To facilitate a more informative comparison between fingerprinting and watermarking, we create a baseline called "WeakWM" by



(a) Fingerprints on denoising models. (b) Fingerprints on SR models.

Figure 6: Fingerprints (critical images) calculated from different models under three attacks.

	Image Denoising								Image SR							
	Model	ProjCL	PoolCL	Zhang	Quan	Ong	WeakWM	Ours	Model	ProjCL	PoolCL	Zhang	Quan	Ong	WeakWM	Ours
Pruning	SimBase	✓	✓	✓	✓	✓	✗	✓	DRLN	✗	✓	✓	✓	✓	✓	✓
	DBSN	✓	✓	✓	✓	✓	✗	✓	EDSR	✗	✓	✓	✓	✓	✗	✓
	DnCNN	✗	✗	✓	✓	✓	✗	✓	MobileSR	✗	✗	✓	✓	✓	✗	✓
	NAFNet	✗	✗	✓	✓	✓	✓	✓	RLFN	✗	✓	✓	✓	✓	✗	✓
	Nei2Nei	✓	✓	✓	✓	✓	✓	✓	RNAN	✗	✗	✓	✓	✓	✗	✓
	Restormer	✓	✓	✓	✓	✓	✗	✓	RRDBNet	✗	✗	✓	✓	✓	✗	✓
Finetuning	SimBase	✓	✓	✓	✓	✓	✗	✓	DRLN	✗	✓	✓	✓	✓	✗	✓
	DBSN	✓	✓	✓	✓	✓	✓	✓	EDSR	✗	✓	✓	✓	✓	✓	✓
	DnCNN	✗	✗	✓	✓	✓	✗	✓	MobileSR	✗	✓	✓	✓	✓	✗	✓
	NAFNet	✗	✓	✓	✓	✓	✗	✓	RLFN	✗	✓	✓	✓	✓	✗	✓
	Nei2Nei	✓	✓	✓	✓	✓	✗	✓	RNAN	✗	✓	✓	✓	✓	✗	✓
	Restormer	✓	✓	✓	✓	✓	✗	✓	RRDBNet	✗	✗	✓	✓	✓	✗	✓
Quantization	SimBase	✓	✓	✓	✓	✓	✓	✓	DRLN	✗	✗	✓	✓	✓	✗	✓
	DBSN	✓	✓	✓	✓	✓	✓	✓	EDSR	✗	✓	✓	✓	✓	✓	✓
	DnCNN	✗	✗	✓	✓	✓	✗	✓	MobileSR	✗	✓	✓	✓	✓	✗	✓
	NAFNet	✗	✓	✓	✓	✓	✗	✓	RLFN	✗	✗	✓	✓	✓	✗	✓
	Nei2Nei	✓	✓	✓	✓	✓	✗	✓	RNAN	✗	✓	✓	✓	✓	✓	✓
	Restormer	✓	✓	✓	✓	✓	✗	✓	RRDBNet	✗	✗	✓	✓	✓	✗	✓
Success Rate		67%	78%	100%	100%	100%	28%	100%	Success Rate	0%	61%	100%	100%	100%	22%	100%

Table 1: Verification under attacks. Blue: fingerprinting; Purple: watermarking; ✗: failure; ✓: success.

reducing the watermark embedding strength of Quan *et al.*'s method [39] to a sufficiently low value where no significant loss in fidelity is observed. The fingerprints extracted by our approach for both tasks are shown in Figure 6, under various attacks. Verification results for different methods are listed in Table 1.

Model pruning We prune the source models by zeroing the top $p\%$ of smallest weights in the DNN's layers, where $p = 10$ for denoising and $p = 5$ for SR. The patterns are nearly the same for both tasks under model pruning. Our approach achieves successful verification on all models under all pruning ratios for both tasks, outperforming the two baselines that fails in nearly half the cases. Notably, the baselines perform much worse for SR models than for denoising models. The baseline ProjCL fails in all SR cases, suggesting that the methods designed for classification DNNs are ineffective when applied to image restoration DNNs. The watermarking methods succeed for all mod-

els, which is expected due to their invasive manner. However, when the watermark embedding strength is very weak, WeakWM fails in most cases during watermark verification.

Model finetuning We fine-tune the source models with their original tasks on the BSD68 dataset [25] for 500 iterations (steps). Little change can be observed in the patterns of the resulting critical images. Our fingerprinting approach and the three watermarking methods successfully verify all models. However, among the baselines, PoolCL fails in one denoising and one SR case, ProjCL fails in two denoising and all SR cases, and WeakWM succeeds only once in each task due to weak watermark strength.

Model quantization When Int8 quantization with simple rounding is applied to the source models, our approach still produces robust yet discriminative fingerprints. It successfully verifies all models, like the three watermarking methods. WeakWM only succeeds in one half cases. ProjCL fails in two denoising cases and all SR cases. PoolCL performs acceptably on denoising models but fails in a half of the SR cases.

4.4. Ablation Study

We form additional two baselines of our approach for further study, which are as follows:

1. w/o TV: Removing the TV (second term) in (1) for calculating the critical image.
2. MaxObj: Replacing the critical image in our approach with an image generated by maximizing the objective function in (1) instead of minimizing it. The resulting image is expected to be as smooth as possible while hard to recover, which can be seen as an adversarial sample.

For the "w/o TV" case, we observe that the resulting critical images tend to degenerate into a constant image easily for some models, meaning that its discriminability could not be guaranteed. Therefore, we only compare the success rates of MaxObj and our approach in terms of robustness. See Table 2 for the results, where our approach outperforms MaxObj. This may be because the fingerprints generated by MaxObj are not as close to the performance border zone as those generated by our approach.

Method	Compression		Finetuning		Quantization	
	Denoising	SR	Denoising	SR	Denoising	SR
MaxObj	5/6	4/6	6/6	4/6	6/6	3/6
Ours	6/6	6/6	6/6	6/6	6/6	6/6

Table 2: Success rates of baselines in ablation study.

4.5. Demos Beyond Denoising and SR

To further examine the applicability of our approach to other image restoration tasks, we provide demos on additional tasks: deblurring (including motion deblurring and defocus deblurring), deraining, and low-light enhancement. The operator $\mathcal{D}_{\mathcal{T}}$s are defined as follows:

$$\mathcal{D}_{\mathcal{T}}(\mathbf{X}) := \mathbf{K} \otimes \mathbf{X} + \mathbf{N}, \quad \text{[Deblurring]}$$
$$\mathcal{D}_{\mathcal{T}}(\mathbf{X}) := \text{Norm}(\mathbf{X}^3), \quad \text{[Low-light Enhancement]}$$
$$\mathcal{D}_{\mathcal{T}}(\mathbf{X}) := \mathbf{X} + \mathbf{R}, \quad \text{[Deraining]}$$

where $\mathbf{K}$ is a blur kernel, $\text{Norm}(\cdot)$ denotes min-max normalization, and $\mathbf{R}$ is a synthetic rain layer; see supplemental material for details. We select three DNNs with published pre-trained models for each task: IFAN [17], GKMNet [40] and NRKNet [41] for defocus deblurring; Restormer [50], SimBase [6] and NAFNet [6] for motion deblurring; Restormer [50], MPRNet [51] and VRGNet [45] for deraining; and MIRNetv2 [52], DLN [46] and ZeroDCE [9] for low-light enhancement. The results in Figure 7 (for defocus deblurring and low-light enhancement) and supplemental material (for motion deblurring and deraining) show that, the extracted fingerprints exhibit discriminative patterns, and meanwhile they remain similar under pruning and quantization attacks.

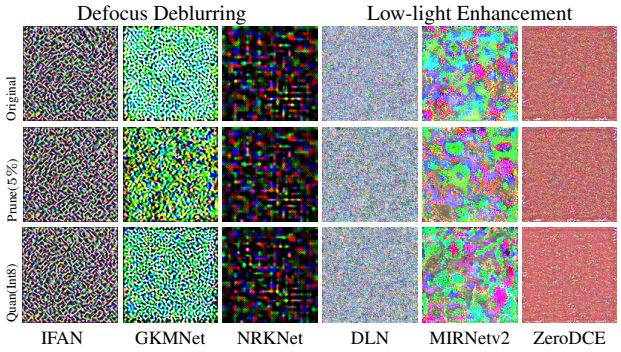


Figure 7: Fingerprints calculated from different pre-trained and attacked DNN models on other tasks. See supplemental material for the results on deraining.

5. Conclusion and Discussion

Given the prevalence of DNN models used in image restoration, protecting their IPR has become increasingly important. Toward this end, we proposed a non-intrusive fingerprinting framework for verifying the model ownership of image restoration DNNs, with its effectiveness demonstrated in two important tasks: image denoising and SR. Similar to existing white-box methods, ours requires the access to model weights. One possible future direction is exploring online model encryption techniques to extend our approach to the black-box setting. In addition, the theoretical aspects of the critical image such as uniqueness, as well as the extensions to other low-level vision DNNs, will be investigated in our future work.

References

[1] Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *USENIX Security Symposium*, pages 1615–1631, Baltimore, MD, 2018. USENIX Association. 1, 3

[2] Saeed Anwar and Nick Barnes. Densely Residual Laplacian Super-Resolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1192–1204, Mar. 2022. 5

[3] Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Ipguard: Protecting intellectual property of deep neural networks via fingerprinting the classification boundary. In *Proceedings of ACM Asia Conference on Computer and Communications Security*, pages 14–25, 2021. 1, 3, 6

[4] Huili Chen, Bita Darvish Rouhani, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. Deepmarks: A secure fingerprinting framework for digital rights management of deep learning models. In *Proceedings of International Conference on Multimedia Retrieval*, pages 105–113, 2019. 3

[5] Jialuo Chen, Jingyi Wang, Tinglan Peng, Youcheng Sun, Peng Cheng, Shouling Ji, Xingjun Ma, Bo Li, and Dawn Song. Copy, right? a testing framework for copyright protection of deep learning models. In *IEEE Symposium on Security and Privacy*, pages 824–841. IEEE, 2022. 3

[6] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple Baselines for Image Restoration, Aug. 2022. 5, 8

[7] Lixin Fan, Kam Woh Ng, and Chee Seng Chan. Rethinking deep neural network ownership verification: Embedding passports to defeat ambiguity attacks. *Advances in Neural Information Processing Systems*, 32, 2019. 1, 3

[8] Xiquan Guan, Huamin Feng, Weiming Zhang, Hang Zhou, Jie Zhang, and Nenghai Yu. Reversible watermarking in deep convolutional neural networks for integrity authentication. In *Proceedings of ACM International Conference on Multimedia*, pages 2273–2280, 2020. 1, 3

[9] Chunle Guo, Chongyi Li, Jichang Guo, Chen Change Loy, Junhui Hou, Sam Kwong, and Runmin Cong. Zero-reference deep curve estimation for low-light image enhancement. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1780–1789, 2020. 8

[10] Jia Guo and Miodrag Potkonjak. Watermarking deep neural networks for embedded systems. In *Proceedings of International Conference on Learning Representations*, pages 1–8. IEEE, 2018. 1, 3

[11] Zecheng He, Tianwei Zhang, and Ruby Lee. Sensitive-sample fingerprinting of deep neural networks. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4729–4737, 2019. 1, 3

[12] Tao Huang, Songjiang Li, Xu Jia, Huchuan Lu, and Jianzhuang Liu. Neighbor2neighbor: Self-supervised denoising from single noisy images. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, pages 14781–14790, June 2021. 5

[13] Bongjin Jun and Daijin Kim. Robust face detection using local gradient patterns and evidence accumulation. *Pattern Recognition*, 45(9):3304–3316, 2012. 2, 4

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[15] Fangyuan Kong, Mingxi Li, Songwei Liu, Ding Liu, Jingwen He, Yang Bai, Fangmin Chen, and Lean Fu. Residual Local Feature Network for Efficient Super-Resolution. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 766–776, 2022. 1, 5

[16] Erwan Le Merrer, Patrick Perez, and Gilles Trédan. Adversarial frontier stitching for remote neural network watermarking. *Neural Computing and Applications*, 32(13):9233–9244, 2020. 3

[17] Junyong Lee, Hyeongseok Son, Jaesung Rim, Sunghyun Cho, and Seungyong Lee. Iterative filter adaptive network for single image defocus deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2034–2042, 2021. 1, 8

[18] Don S Lemons. An introduction to stochastic processes in physics, 2003. 2

[19] Bowen Li, Lixin Fan, Hanlin Gu, Jie Li, and Qiang Yang. Fedipr: Ownership verification for federated deep neural network models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 3

[20] Fang-Qi Li, Shi-Lin Wang, and Alan Wee-Chung Liew. Regulating ownership verification for deep neural networks: Scenarios, protocols, and prospects. *Proceedings of International Joint Conferences on Artificial Intelligence*, 2021. 1, 3

[21] Yiming Li, Linghui Zhu, Xiaojun Jia, Yong Jiang, Shu-Tao Xia, and Xiaochun Cao. Defending against model stealing via verifying embedded external features. In *Proceedings of AAAI Conference on Artificial Intelligence*. AAAI, 2022. 1, 3

[22] 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. 1, 5

[23] Chun-Shien Lu. Sparse trigger pattern guided deep learning model watermarking. In *Proceedings of ACM Workshop on Information Hiding and Multimedia Security*, pages 33–38, 2022. 3

[24] Nils Lukas, Yuxuan Zhang, and Florian Kerschbaum. Deep neural network fingerprinting by conferrable adversarial examples. *Proceedings of International Conference on Learning Representations*, 2021. 1, 3

[25] 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. 8

[26] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, pages 1765–1773, 2017. 3

[27] Yuki Nagai, Yusuke Uchida, Shigeyuki Sakazawa, and Shin'ichi Satoh. Digital watermarking for deep neural networks. *International Journal of Multimedia Information Retrieval*, 7(1):3–16, 2018. 3

[28] Ryota Namba and Jun Sakuma. Robust watermarking of neural network with exponential weighting. In *Proceedings of ACM Asia Conference on Computer and Communications Security*, pages 228–240, 2019. 3

[29] Yuesong Nan and Hui Ji. Deep learning for handling kernel/model uncertainty in image deconvolution. In *Proceedings of IEEE/CVF conference on computer vision and pattern recognition*, pages 2388–2397, 2020. 1

[30] 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. 4

[31] Ding Sheng Ong, Chee Seng Chan, Kam Woh Ng, Lixin Fan, and Qiang Yang. Protecting intellectual property of generative adversarial networks from ambiguity attacks. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3630–3639, 2021. 3, 6

[32] Tongyao Pang, Huan Zheng, Yuhui Quan, and Hui Ji. Recorrupted-to-recorrupted: Unsupervised deep learning for image denoising. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2043–2052, June 2021. 1

[33] Zirui Peng, Shaofeng Li, Guoxing Chen, Cheng Zhang, Haojin Zhu, and Minhui Xue. Fingerprinting deep neural networks globally via universal adversarial perturbations. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13430–13439, 2022. 1, 3

[34] Rui Qian, Robby T Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. In *Proceedings of IEEE conference on computer vision and pattern recognition*, pages 2482–2491, 2018. 1

[35] Yuhui Quan, Mingqin Chen, Tongyao Pang, and Hui Ji. Self2self with dropout: Learning self-supervised denoising from single image. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020. 1

[36] Yuhui Quan, Yixin Chen, Yizhen Shao, Huan Teng, Yong Xu, and Hui Ji. Image denoising using complex-valued deep cnn. *Pattern Recognition*, 111:107639, 2021. 1

[37] Yuhui Quan, Shijie Deng, Yixin Chen, and Hui Ji. Deep learning for seeing through window with raindrops. In *Proceedings of IEEE/CVF International Conference on Computer Vision*, October 2019. 1

[38] Yuhui Quan, Xiaoheng Tan, Yan Huang, Yong Xu, and Hui Ji. Image desnowing via deep invertible separation. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(33):3133–3144, 2023. 1

[39] Yuhui Quan, Huan Teng, Yixin Chen, and Hui Ji. Watermarking deep neural networks in image processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(5):1852–1865, 2020. 1, 3, 5, 6, 7

[40] Yuhui Quan, Zicong Wu, and Hui Ji. Gaussian kernel mixture network for single image defocus deblurring. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 20812–20824. Curran Associates, Inc., 2021. 1, 8

[41] Yuhui Quan, Zicong Wu, and Hui Ji. Neumann network with recursive kernels for single image defocus deblurring. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5754–5763, 2023. 1, 8

[42] Bita Darvish Rouhani, Huili Chen, and Farinaz Koushanfar. Deepsigns: A generic watermarking framework for ip protection of deep learning models. *Proceedings of International Conference on Architectural Support for Programming Languages and Operating Systems*, 2021. 3

[43] Long Sun, Jinshan Pan, and Jinhui Tang. ShuffleMixer: An Efficient ConvNet for Image Super-Resolution, May 2022. 1, 5

[44] Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh. Embedding watermarks into deep neural networks. In *Proceedings of ACM on international conference on multimedia retrieval*, pages 269–277. ACM, 2017. 1, 3

[45] 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. 1, 8

[46] Li-Wen Wang, Zhi-Song Liu, Wan-Chi Siu, and Daniel PK Lun. Lightening network for low-light image enhancement. *IEEE Transactions on Image Processing*, 29:7984–7996, 2020. 8

[47] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. In *Proceedings of European Conference on Computer Vision*, pages 0–0, 2018. 1, 5

[48] Quan Wen, Tan-feng SUN, and Shu-xun Wang. Concept and application of zero-watermark. *ACTA ELECTONICA SINICA*, 31(2):214, 2003. 3

[49] Xiaohe Wu, Ming Liu, Yue Cao, Dongwei Ren, and Wangmeng Zuo. Unpaired Learning of Deep Image Denoising. In *Proceedings of European Conference on Computer Vision*, pages 352–368, 2020. 5

[50] 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. 1, 5, 8

[51] 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. 1, 8

[52] Syed Waqas Zamir, Aditya Arora, Salman Hameed Khan, Hayat Munawar, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Learning Enriched Features for Fast Image Restoration and Enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2022. 8

[53] Jie Zhang, Dongdong Chen, Jing Liao, Han Fang, Weiming Zhang, Wenbo Zhou, Hao Cui, and Nenghai Yu. Model watermarking for image processing networks. In *Proceedings of AAAI Conference on Artificial Intelligence*, volume 34, pages 12805–12812, 2020. 1, 3

[54] Jie Zhang, Dongdong Chen, Jing Liao, Weiming Zhang, Huamin Feng, Gang Hua, and Nenghai Yu. Deep model intellectual property protection via deep watermarking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1, 3

[55] Jie Zhang, Dongdong Chen, Jing Liao, Weiming Zhang, Huamin Feng, Gang Hua, and Nenghai Yu. Deep Model Intellectual Property Protection via Deep Watermarking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4005–4020, Aug. 2022. 6

[56] Jialong Zhang, Zhongshu Gu, Jiyong Jang, Hui Wu, Marc Ph Stoecklin, Heqing Huang, and Ian Molloy. Protecting intellectual property of deep neural networks with watermarking. In *Proceedings of Asia Conference on Computer and Communications Security*, pages 159–172, 2018. 3

[57] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. 1, 5

[58] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3929–3938, 2017. 1

[59] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual Non-local Attention Networks for Image Restoration. In *Proceedings of International Conference on Learning Representations*, Feb. 2022. 1, 5

[60] Jingjing Zhao, Qingyue Hu, Gaoyang Liu, Xiaoqiang Ma, Fei Chen, and Mohammad Mehedi Hassan. Afa: Adversarial fingerprinting authentication for deep neural networks. *Computer Communications*, 150:488–497, 2020. 1, 3

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

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

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:

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

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$\sim$#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.

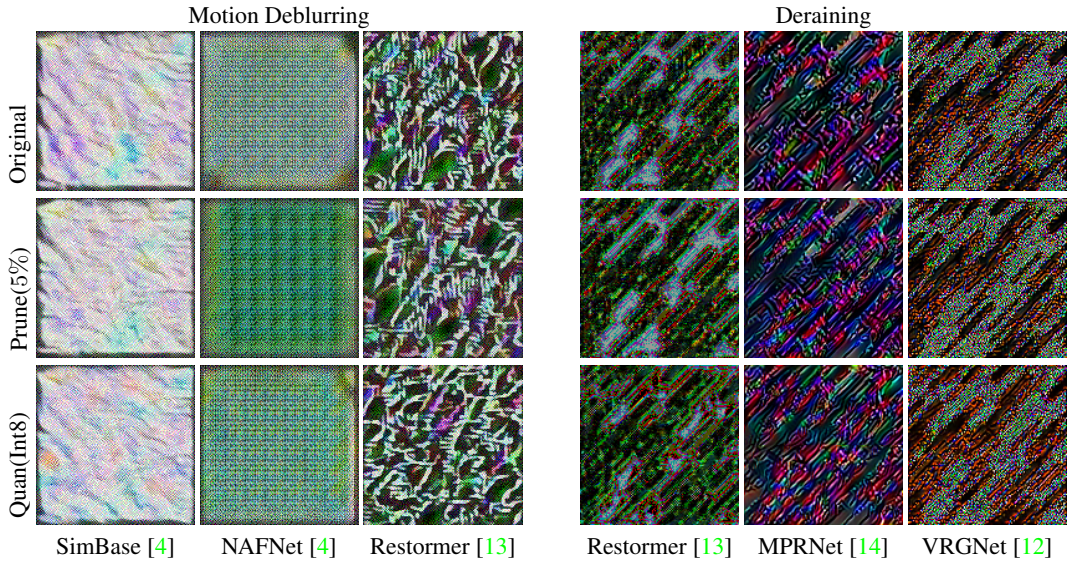


Figure 1: Fingerprints extracted from different image DNN models of two tasks.

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.

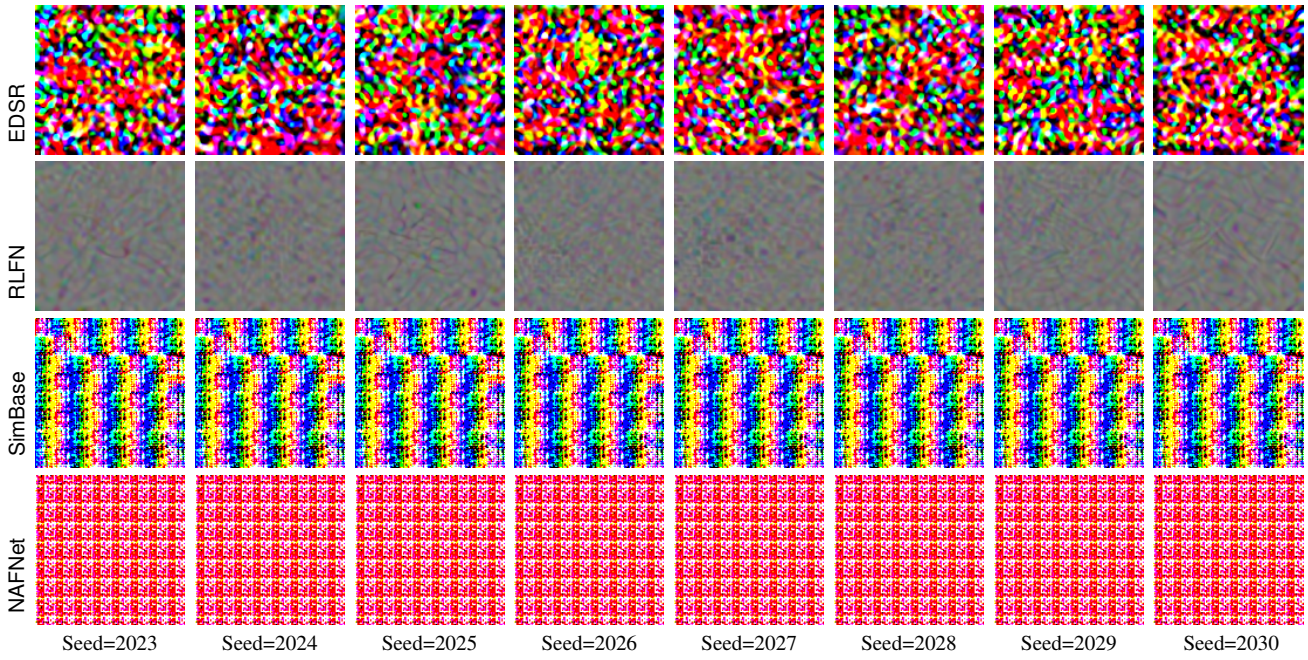
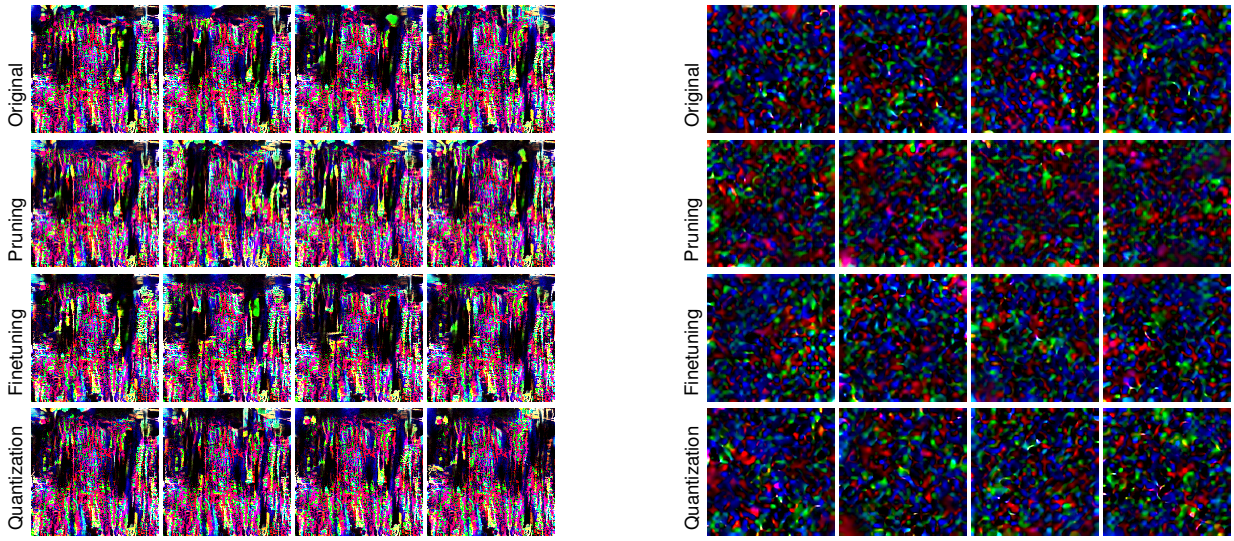


Figure 2: Fingerprints calculated using different instances of $\mathbf{S}^{(0)}$ obtained via different seeds.



(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

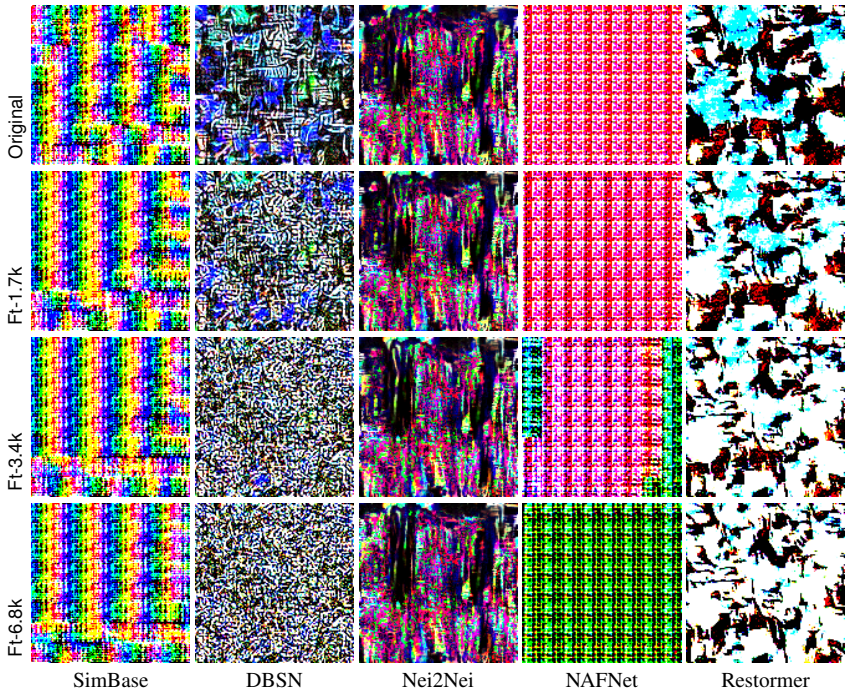


Figure 4: Fingerprints calculated from the denoising models under finetuning attacks with different numbers of iterations.

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