

RESEARCH

Open Access

DIPPAS: a deep image prior PRNU anonymization scheme



Francesco Picetti* , Sara Mandelli, Paolo Bestagini, Vincenzo Lipari and Stefano Tubaro

Abstract

Source device identification is an important topic in image forensics since it allows to trace back the origin of an image. Its forensics counterpart is source device anonymization, that is, to mask any trace on the image that can be useful for identifying the source device. A typical trace exploited for source device identification is the photo response non-uniformity (PRNU), a noise pattern left by the device on the acquired images. In this paper, we devise a methodology for suppressing such a trace from natural images without a significant impact on image quality. Expressly, we turn PRNU anonymization into the combination of a global optimization problem in a deep image prior (DIP) framework followed by local post-processing operations. In a nutshell, a convolutional neural network (CNN) acts as a generator and iteratively returns several images with attenuated PRNU traces. By exploiting straightforward local post-processing and assembly on these images, we produce a final image that is anonymized with respect to the source PRNU, still maintaining high visual quality. With respect to widely adopted deep learning paradigms, the used CNN is not trained on a set of input-target pairs of images. Instead, it is optimized to reconstruct output images from the original image under analysis itself. This makes the approach particularly suitable in scenarios where large heterogeneous databases are analyzed. Moreover, it prevents any problem due to the lack of generalization. Through numerical examples on publicly available datasets, we prove our methodology to be effective compared to state-of-the-art techniques.

Keywords: Deep image prior, Image anonymization, Image forensics, PRNU

1 Introduction

Source device identification is a well-studied problem in the multimedia forensics community [1–4]. Indeed, identifying the source camera of an image helps trace its origin and verify its integrity. Many state-of-the-art techniques tackle this problem by relying on photo response non-uniformity (PRNU), which is a unique characteristic noise pattern left by the device on each acquired content [1]. Given a query image and a candidate device, it is possible to infer whether the device shot the image with a cross-correlation test between a noise trace extracted from the image and the device PRNU [5].

Despite the effort put into developing robust PRNU-based source attribution techniques, the forensic community has also focused on studying the possibility of

removing PRNU traces from images. On one hand, determining at which level PRNU can be removed is essential to study the robustness of PRNU-based forensic detectors and possibly improve them. On the other hand, linking a picture to its owner is undesirable when privacy is a concern. For example, photojournalists conducting legit investigations may anonymize their shots to avoid being threatened.

For these reasons, counter-forensic methods that enable deleting or reducing PRNU traces from images have been proposed in the literature. We can broadly split the developed techniques into two families. The first family requires the knowledge of the PRNU pattern to be deleted, and we refer to them as PRNU-aware methods. This is the case of [6–8], which propose different iterative solutions to delete a known PRNU from a given picture. Specifically, [6] proposes an adaptive PRNU-based image denoising, removing an estimate of the PRNU from

*Correspondence: francesco.picetti@polimi.it
Dipartimento di Elettronica, Informazione e Biongegneria - Politecnico di Milano, Piazza Leonardo da Vinci, 32, 20133 Milano, Italy

each image. The authors of [7] estimate the best subtraction weight that minimizes the cross-correlation between the PRNU and the trace extracted from the image to be anonymized. Recently, [8] applies a convolutional neural network (CNN), which exploits the source PRNU to hinder its traces from a query image. The network is used as a parametric operator, which iteratively overfits the given pair of image and PRNU, imposing a minimization of their correlation.

The second family of methods works by blindly modifying pixel values to make the underlying PRNU unrecognizable. For instance, [9] shows that multiple image denoising steps can help attenuate the PRNU, even though this may not be enough to ultimately hinder its traces from images [10]. Alternatively, [11] applies seam-carving to change pixel locations, and [12] exploits patch-match techniques to scramble pixel positions. More recently, [13] proposes an inpainting-based method that deletes and reconstructs image pixels such that final images are anonymized with respect to their source PRNU.

In this manuscript, we propose an image anonymization tool leveraging a combination of a global optimization strategy (i.e., operating on the entire image) and local post-processing operations. In particular, global optimization on the entire image is performed exploiting a CNN. Given an image to be anonymized and a reference PRNU trace to be removed, the proposed network iteratively generates multiple images where PRNU traces are attenuated, still maintaining high visual quality. Differently from most CNN works, the proposed network does not need a training step. In fact, the used CNN exploits the deep image prior (DIP) paradigm [14], thus acting as a framework to solve an inverse problem: estimate PRNU-free images from the picture under analysis and a reference PRNU trace. The proposed CNN takes a random noise realization as input and iterates until it is capable of generating PRNU-free representations of a selected picture. The analyst can decide when to stop CNN iterations by simply checking the trade-off between the quality of the generated images and the reached anonymization level. Then, we propose to aggregate the CNN output images through a post-processing step that works at the image local level. In doing so, we achieve a remarkable enhancement concerning both image quality and anonymization level at the expense of little additional computational cost.

The developed anonymization scheme is validated on 1200 color images of the well-known Dresden Image Database [15] and 600 color images of the Vision Source Identification Dataset [16]. We address the anonymization problem on both uncompressed images (i.e., images selected from the Dresden Dataset) and JPEG-compressed images (i.e., Dresden and Vision Datasets). For the sake of comparison with state-of-the-art techniques, we test our methodology both when an estimate of the device

PRNU is available (i.e., in a PRNU-aware scenario) and when the device PRNU can be estimated only from the query image itself (i.e., in a PRNU-blind scenario). The results show that the proposed technique hinders PRNU-based detectors, especially when the actual device PRNU is available.

The contributions of this paper can be summarized as follows:

- We propose the first application of the DIP paradigm to an image forensic problem, to the best of our knowledge.
- We adapt the DIP denoising pipeline to work in case of known multiplicative noise.
- We propose a PRNU-aware image anonymization technique that outperforms the state-of-the-art on the Dresden Dataset and is the runner-up on Vision.
- We propose a PRNU-blind image anonymization technique that reduces PRNU traces around image edges in contrast to other methods in the literature.
- We propose a method that allows forensic analysts to select the trade-off between image quality and anonymization capability depending on the working scenario.

The rest of the paper is organized as follows. In Section 2, we provide the reader with the background concepts helpful in understanding the core of the proposed methodology. In Section 3, we present the details of the proposed scheme. In particular, we first define the inverse problem we aim at solving, then we devise a processing pipeline to obtain the target PRNU-free image. Finally, we describe the architecture design along with the optimization strategies. In Section 4, we describe the experimental setup; in Section 5, we discuss all the achieved results, compared with state-of-the-art solutions. Eventually, in Section 6, we draw our conclusions.

2 Background and problem statement

In this section, we introduce some background concepts helpful to understand the rest of the paper. First, we introduce photo response non-uniformity (PRNU) and its use in source device identification. Then, we present the adopted methodology known as deep image prior (DIP) [14], which has been recently proposed as a paradigm to solve diverse inverse problems like image denoising. Eventually, we formulate the source device anonymization problem faced in this paper.

2.1 Photo response non-uniformity

Photo response non-uniformity (PRNU) is a characteristic noise fingerprint introduced in all images acquired by a device. Specifically, the PRNU \mathbf{K} has the form of a zero-mean pixel-wise multiplicative noise. According to

the well-known model proposed in [1, 2], a generic image \mathbf{I} shot by a digital device can be described as:

$$\mathbf{I} = \mathbf{I}_0 (1 + \gamma \mathbf{K}) + \Theta, \tag{1}$$

where \mathbf{I}_0 is the sensor output in the absence of noise, γ is the weight of the PRNU contribution, and Θ includes all the additional independent random noise components. The PRNU \mathbf{K} can be estimated by collecting a set of images shot by the device, following the method proposed in [1, 2].

PRNU \mathbf{K} is commonly used to solve source device identification problem, that is, given a query image \mathbf{I} and a candidate device, understanding if the device shot that image or not. One way to solve the problem is to compute the normalized cross-correlation (NCC) [1] between a noise residual \mathbf{W} [2] extracted from the query image \mathbf{I} and the PRNU \mathbf{K} of the candidate device, pixel-wise scaled by \mathbf{I} . Referring to [1], we can define the NCC as:

$$\text{NCC}(\mathbf{W}, \mathbf{IK}) = \frac{\sum_{i,j} [\mathbf{W}]_{i,j} \cdot [\mathbf{IK}]_{i,j}}{\|\mathbf{W}\|_F \cdot \|\mathbf{IK}\|_F}, \tag{2}$$

where $\|\cdot\|_F$ is the Frobenius norm, while $[\mathbf{W}]_{i,j}$ and $[\mathbf{IK}]_{i,j}$ are the terms in position (i, j) of \mathbf{W} and \mathbf{IK} , respectively. If $\text{NCC}(\mathbf{W}, \mathbf{IK})$ is greater than a predefined threshold, the image can be attributed to the device with a certain confidence.

2.2 Deep image prior

A generic image restoration problem is usually solved through the minimization of an objective function of the form

$$J(\mathbf{x}) = E(\mathbf{x}; \mathbf{I}) + \lambda R(\mathbf{x}), \tag{3}$$

where $E(\mathbf{x}; \mathbf{I})$ is a task-dependent misfit function conditioned by the (corrupted) input image \mathbf{I} , and $R(\mathbf{x})$ is a regularization term designed to tackle the ill-posedness and ill-conditioning of the inverse problem. To avoid confusion with the rest of notation used in the paper, we refer at \mathbf{x} as a generic image which the cost function is evaluated for. λ is a weight setting the trade-off between honoring the data and imposing the desired a-priori features [17]. The restored image $\hat{\mathbf{x}}$ is then obtained as:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} J(\mathbf{x}). \tag{4}$$

The data misfit $E(\mathbf{x}; \mathbf{I})$ is usually quite simple to devise, depending on the desired task. The design of a regularization term $R(\mathbf{x})$ can be challenging because it should capture the features of the desired image.

Deep image prior (DIP) has been proposed as an alternative solution with respect to standard regularization [14]. The objective function to minimize is recast as:

$$J(\phi) = E(\mathbf{x}; \mathbf{I}) = E(f_{\phi}(\mathbf{z}); \mathbf{I}), \tag{5}$$

where $f_{\phi}(\cdot)$ is a CNN represented as a parametric non linear function, ϕ are the parameters of this function (i.e., the weights of the CNN), and \mathbf{z} is a random noise realization. The value $f_{\phi}(\mathbf{z})$ is associated with the output image to the network; thus, it is related to a random noise realization \mathbf{z} and to the CNN parameters ϕ . Notice that there is not an explicit regularization term: the CNN architecture itself plays the role of the prior. Through its convolutional layers, the CNN captures the inner structure and self-similarities of the desired uncorrupted image from the input corrupted one and constrains the solution space. In other words, instead of minimizing the objective function in the space of the image as in (4), DIP performs the search in the space of the CNN parameters ϕ . This dramatically changes the shape of the objective function, driving the solution to honor both the data misfit and the deep features captured on the corrupted input image. The restored image is then obtained as:

$$\hat{\mathbf{x}} = f_{\hat{\phi}}(\mathbf{z}), \tag{6}$$

where

$$\hat{\phi} = \arg \min_{\phi} J(\phi). \tag{7}$$

For the specific case of image denoising, the DIP objective function presented in (5) is customary set to the ℓ_2 distance between the output image to the CNN for a given combination of noise realization \mathbf{z} and network parameters ϕ , defined as $f_{\phi}(\mathbf{z})$, and the input image \mathbf{I} [18–20]. For the sake of notation, from now on, we refer to the CNN output image $f_{\phi}(\mathbf{z})$ as \mathbf{I}_{ϕ} . Therefore, (5) becomes:

$$J(\phi) = \|\mathbf{I}_{\phi} - \mathbf{I}\|_2^2. \tag{8}$$

One may ask why a CNN that is designed to reconstruct a generic image should perform denoising while its goal is set to fit the input (noisy) image \mathbf{I} . The main reason is the different behavior of signal and noise components throughout the iterative optimization [21]. If the minimization is led to convergence, the result will indeed fit the noisy image. Still, the authors of [14] have shown that parametrizing the optimization via the weights of a CNN generator distorts the search space so that in the minimization process, the signal fits faster than the noise. Therefore, [14] proposes to perform denoising by early stopping the iterative minimization. For example, in a forensic scenario, the analyst can stop the optimization when some task-specific average metrics reach a desirable value.

2.3 Problem formulation

This paper focuses on the forensics counterpart of the source device identification problem, that is, performing source device anonymization. Specifically, given an image, we aim at hindering PRNU traces left on the image to

make it impossible to associate the image with its source device. Meanwhile, the anonymization process should not compromise the visual quality of the anonymized image. This translates into fulfilling two main goals:

1. The NCC between the anonymized image and the actual source PRNU should be lower than a predefined threshold.
2. The peak-signal-to-noise-ratio (PSNR) between the anonymized image and the original image should assume high values.

To this purpose, we propose the Deep Image Prior PRNU Anonymization Scheme (DIPPAS), which is an anonymization method based on a DIP optimization framework followed by a local post-processing and assembly operations studied on purpose. In the next section, we present the proposed strategy, discussing the central intuitions behind the approach.

3 Proposed methodology

The proposed method for image anonymization works in two main steps: a DIP-based approach is adapted to generate multiple images with limited PRNU traces; these images are combined into a single one through a post-processing scheme.

To illustrate all the details of the proposed approach, in this section, we start showing the theoretical DIP-based framework chosen for our specific problem. Then, we explain how it is possible to generate an image using this framework. We then report the details of the developed post-processing scheme that merge multiple images into one. Finally, we describe the employed CNN architecture.

3.1 DIP-based image generation

Considering the model (1) of a generic image \mathbf{I} acquired by a digital device, the anonymization task consists in estimating the ideal PRNU-free image \mathbf{I}_0 . Indeed, \mathbf{I}_0 is completely uncorrelated from the device PRNU, and it has a reasonably good visual quality. We aim at this goal by combining the DIP denoising paradigm of (8) with the PRNU-based image modeling proposed in (1). $\mathbf{I}_\phi = f_\phi(\mathbf{z})$ being the output of the CNN for a given parameter configuration ϕ and \mathbf{z} a noise realization, the functional to be minimized becomes:

$$J(\phi) = \|\mathbf{I}_\phi (1 + \gamma \mathbf{P}) - \mathbf{I}\|_{\mathbb{F}}^2. \quad (9)$$

We define \mathbf{P} as the device fingerprint that can be either the estimated PRNU pattern (i.e., $\mathbf{P} = \mathbf{K}$) or the noise residual \mathbf{W} extracted from \mathbf{I} as suggested in [2].

The former situation is a PRNU-aware scenario (e.g., users want to anonymize their pictures and know the reference PRNU). In this case, the proposed scheme makes

use of the PRNU \mathbf{K} as the fingerprint \mathbf{P} ; hence, (9) becomes:

$$J(\phi) = \|\mathbf{I}_\phi (1 + \gamma \mathbf{K}) - \mathbf{I}\|_{\mathbb{F}}^2. \quad (10)$$

The latter situation is a PRNU-blind scenario (e.g., a website wants to store anonymized images uploaded by users, but each reference PRNU is not known at server-side). In this case, the fingerprint \mathbf{P} we inject in the inverse problem is the noise residual \mathbf{W} extracted from \mathbf{I} itself [2]; hence, (9) becomes:

$$J(\phi) = \|\mathbf{I}_\phi (1 + \gamma \mathbf{W}) - \mathbf{I}\|_{\mathbb{F}}^2. \quad (11)$$

Notice that the term $\mathbf{I}_\phi (1 + \gamma \mathbf{P})$ emulates the image modeling shown in (1) (correctly if $\mathbf{P} = \mathbf{K}$, approximately if $\mathbf{P} = \mathbf{W}$). The more $\mathbf{I}_\phi (1 + \gamma \mathbf{P})$ approaches \mathbf{I} in terms of Frobenius norm, the more \mathbf{I}_ϕ will represent a reasonably better estimate of the ideal PRNU-free image \mathbf{I}_0 , apart from independent random noise contributions. Given these premises, the estimated image \mathbf{I}_ϕ is a good candidate for the anonymization of the input image \mathbf{I} .

In a nutshell, the proposed strategy is depicted in Fig. 1. Starting from image \mathbf{I} , we extract the device noise residual \mathbf{P} either in PRNU-aware or PRNU-blind scenario. Then, we generate the image \mathbf{I}_ϕ following the DIP paradigm, i.e., imposing the fingerprint-injected image $\mathbf{I}_\phi (1 + \gamma \mathbf{P})$ to be as similar as possible to the known image \mathbf{I} . By minimizing the functional (9), we estimate the image \mathbf{I}_ϕ with attenuated fingerprint traces.

3.2 Generation of PRNU-attenuated images

As we previously reported, the proposed DIP process can generate multiple images with attenuated PRNU traces. Referring to Fig. 1, the generation pipeline is the following:

1. The CNN input tensor \mathbf{z} is a realization of a zero-mean white Gaussian noise, with the same size of image \mathbf{I} . We found uniform distributions to be less effective; we are convinced the Gaussian noise is able to excite the deep prior layers in a broader range of values, producing better images.
2. We optimize the weights of the CNN by minimizing (9). The optimization is performed over the generator weights ϕ . Notice that the PRNU injection weight γ acts as a trainable architecture layer, so γ is estimated directly during the inversion. Specifically, γ is clamped to be positive, as negative γ values are not model representative.
3. At each minimization step, we generate an image \mathbf{I}_ϕ , which is saved only if the PSNR with respect to the original image \mathbf{I} is above a certain threshold τ_{PSNR} . This is done to guarantee a sufficiently good visual quality for the generated image.

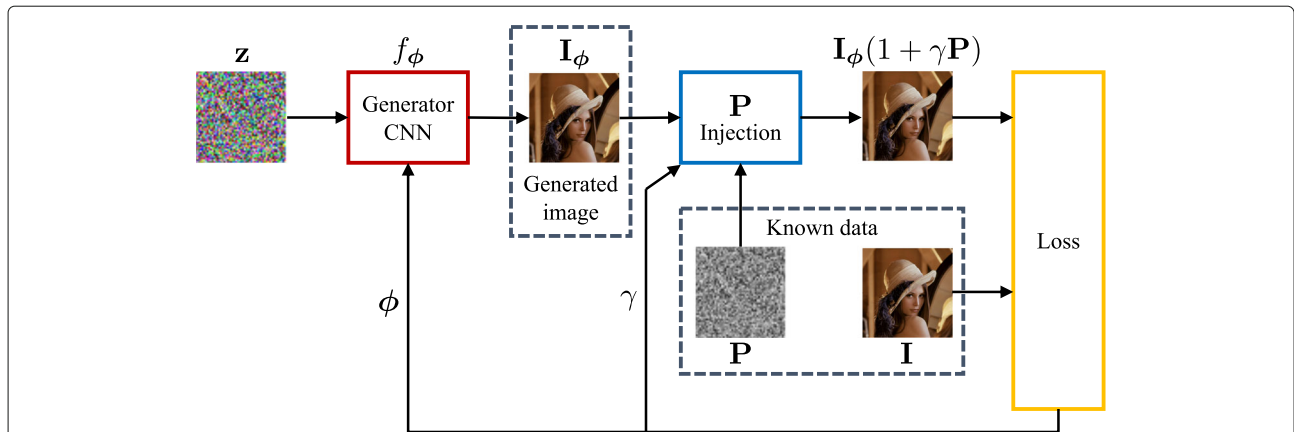


Fig. 1 The proposed DIP-based image generation process. The known data are the acquired image \mathbf{I} and the device fingerprint \mathbf{P} . During the inversion, the traces of \mathbf{P} are attenuated by injecting \mathbf{P} into the image \mathbf{I}_ϕ generated by the CNN. In the PRNU-aware setup, $\mathbf{P} = \mathbf{K}$; in the PRNU-blind setup, $\mathbf{P} = \mathbf{W}$, a noise residual extracted from \mathbf{I} [2]

The minimization process ends when the PSNR between the generated image \mathbf{I}_ϕ , and the initial image \mathbf{I} overcomes a threshold of 39 dB. The maximum number of iterations is anyway fixed to 10,000. In doing so, after the DIP process ends, a pool of M generated images $\mathbf{I}_\phi^{(m)}, m \in [1, M]$ with $\text{PSNR} \geq \tau_{\text{PSNR}}$ has been collected. For the sake of clarity, Figs. 2 and 3 report one example of the inversion process, showing the evolution of the CNN-generated images, together with their PSNRs with respect to the original image and NCCs with the source device PRNU.

3.3 Local post-processing and assembly

Notice that the DIP minimization functional Eq. (9) imposes a constraint on the Frobenius norm of the difference between the fingerprint-injected image and the initial image. This constraint represents a global constraint as it does not explicitly focus on local pixel areas. Recalling our final goals (i.e., maximizing the PSNR and minimizing the NCC of the anonymized image), while the PSNR is a global metric as it considers the entire image and not local areas, the NCC can strongly depend on spe-

cific local regions of the image, which can correlate with the device PRNU in diverse fashions.

Moreover, as DIP iterations increase, the minimization process risks injecting an excessive amount of PRNU traces in the generated images. This is noticeable from the examples depicted in Figs. 2 and 3: when iterations' number grows, the PSNR of the generated images improves but their NCC can slightly grow as well, resulting in worse anonymization.

Given these premises, we propose to further optimize our solution by investigating the M generated images on their local areas. We improve upon these results with a very straightforward methodology to generate one final anonymized image out of the M previously generated ones by locally optimizing the cross-correlation with the reference device fingerprint \mathbf{P} .

Specifically, Fig. 4 depicts the proposed pipeline: we divide each available image $\mathbf{I}_\phi^{(m)}$ and the reference fingerprint \mathbf{P} into N_b non overlapping squared blocks of $B \times B$ pixels. Image and fingerprint blocks are defined as $[\mathbf{I}_\phi^{(m)}]_b$ and $[\mathbf{P}]_b, b \in [1, N_b]$, respectively. Notice that, for each block geometric position b , we have M available

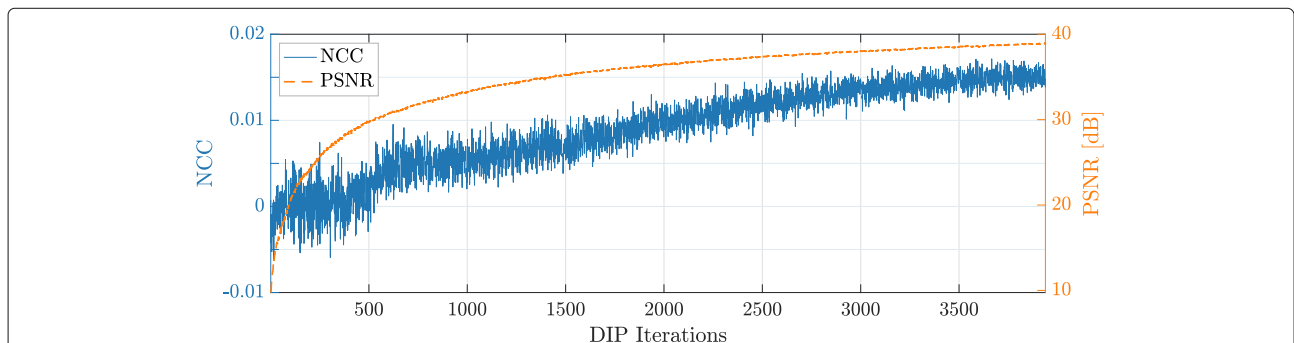


Fig. 2 NCC and PSNR behavior of generated images as a function of DIP iterations. Some generated images are depicted in Fig. 3

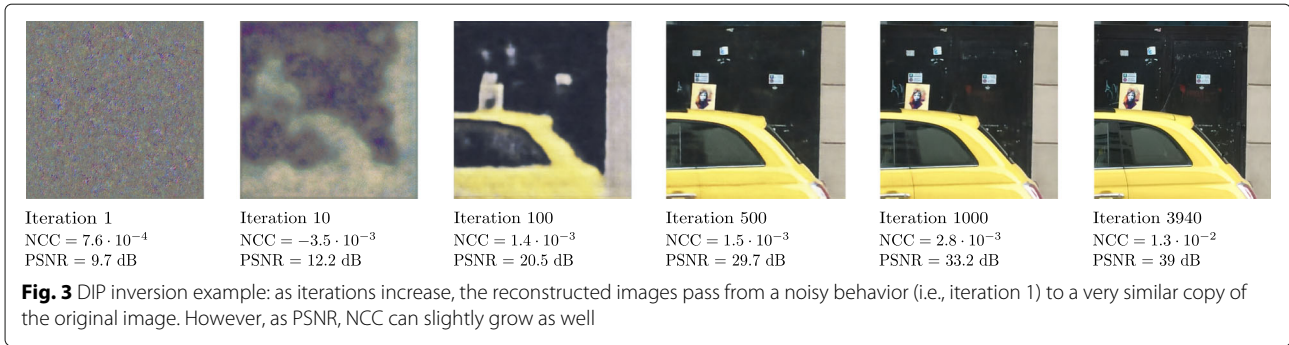


image blocks associated with the M image realizations produced during the DIP iterations. For each block position b , three main steps follow:

1. We compute the NCCs between the M image blocks $[\mathbf{I}_\phi^{(m)}]_b, m \in [1, M]$ and the fingerprint block $[\mathbf{P}]_b$ as in (2).
2. The available M blocks are ordered accordingly to their resulting NCCs: first, we select the blocks with negative NCC and increasing absolute value; secondly, we select the blocks with positive NCC and increasing absolute value. In doing so, blocks with low absolute NCC and negative NCC are given higher priority than blocks returning bigger NCCs.
3. Following the order specified above, we average the first L blocks pixel by pixel, ending up with a $B \times B$ final reconstructed block.

The final anonymized image $\hat{\mathbf{I}}$ is estimated by assembling the results obtained for each single block position b and color channel.

3.4 CNN architecture

The U-Net is a convolutional autoencoder (i.e., a CNN aiming at reconstructing a processed version of its input)

characterized by the so-called skip-connections and originally introduced for medical image processing [22]. If properly trained according to the standard deep learning paradigm, it proves very effective for multidimensional signal processing tasks such as denoising [23, 24], interpolation [25, 26], segmentation [27, 28], inpainting [29], and domain-specific post-processing operators [30, 31].

More recently, the multi-resolution U-Net [32] has been proposed for multimodal medical image segmentation, based on the consideration that the targets of interest have different shapes and scales. Working at different scales can be strongly beneficial if we want to capture self-similarities of natural images as a prior. Preliminary experiments on a small dataset led us to adopt such architecture rather than traditional U-Net and its derivations analyzed in [14].

Therefore, we propose an ad hoc multi-resolution U-Net (shown in Fig. 5) that can be summarized as follows:

1. Convolutional layers are replaced by the so-called multi-resolution blocks shown in the bottom-right portion of Fig. 5. These blocks approximate multi-scale features of the inception block [33] while limiting the number of parameters of the network, which is critical when employing it as a deep prior.

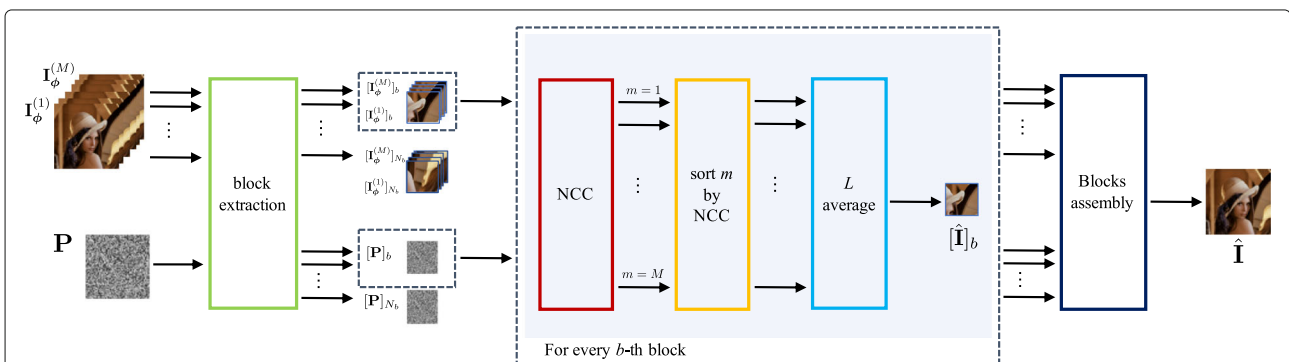


Fig. 4 Assembly of the anonymized image. For each generated image $\mathbf{I}_\phi^{(m)}$, N_b blocks are extracted from the image and the fingerprint \mathbf{P} . Fixing a block position b , we compute the NCC between each pair of blocks $[\mathbf{I}_\phi^{(m)}]_b, [\mathbf{P}]_b, m \in [1, M]$. Then, we order the M image blocks according to their resulting NCCs, and we average the first L blocks pixel by pixel, obtaining the estimated block $[\hat{\mathbf{I}}]_b$. We follow this pipeline for each block position b , eventually assembling the results and estimating the anonymized image $\hat{\mathbf{I}}$

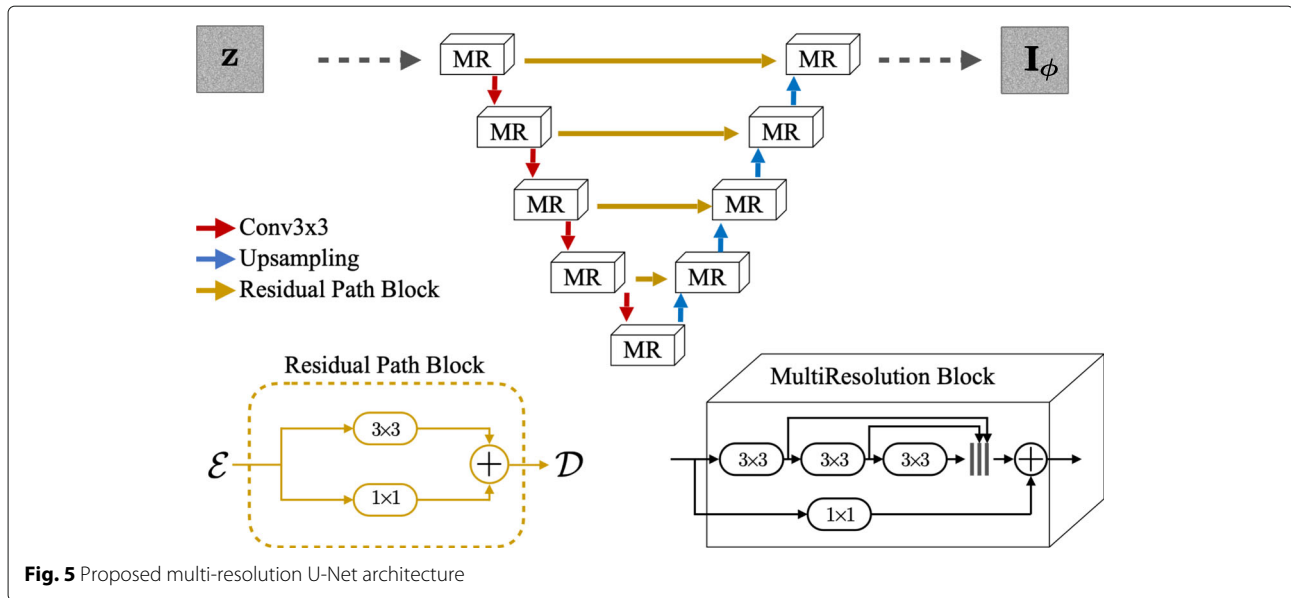


Fig. 5 Proposed multi-resolution U-Net architecture

Every block is a chain of three convolutions; the final output is the sum of the input, scaled by a learned factor, and the stack of the three partial outputs.

2. Skip connections, which are the distinctive feature of the U-Net, are replaced by residual path blocks shown in the bottom-left portion of Fig. 5, as proposed in [32]. \mathcal{E} is the output of an encoding layer, and \mathcal{D} is concatenated to the corresponding decoding layer.
3. Downsampling is achieved by 3×3 convolutions with stride 2×2 . Upsampling is performed by nearest neighbour interpolation. Batch normalization and LeakyReLU activation follow every convolution apart from the last one (i.e., the CNN output) that is activated by a sigmoid.

Notice that, even though the result is the output of a CNN, the DIP method does not exploit the typical deep learning paradigm where a training phase is performed over a specifically designed set of data. In particular, only the query image is used in the reconstruction process, and the CNN implicitly assumes the role of prior information that exploits correlations in the image to learn its inner structure.

4 Experiments

This section describes the used datasets, the experimental setup, and the evaluation metrics.

4.1 Datasets

We resort to two well-known datasets commonly used for investigating PRNU-related problems on images. The first dataset is the Dresden Image Database [15], which collects both uncompressed and compressed images from

more than 50 various devices. Following the same procedure done in past works proposed in literature [12, 13], we select images from 6 different camera instances, precisely Nikon D70, Nikon D70s, and Nikon D200, two devices each. The second dataset is the recently released Vision Dataset [16], which includes JPEG compressed images captured from 35 devices. Among the pool of available models, we collect images from 6 different camera vendors, precisely from devices named as D12, D17, D19, D21, D24, and D27 in [16].

The PRNU fingerprint of each device is computed by collecting all the available flat-field images shot by the device and following the maximum likelihood estimation proposed in [2]. Concerning the Dresden Dataset, we exploit never-compressed Adobe Lightroom images to compute the PRNU, as it reasonably is the most accurate way to estimate the device fingerprint. Indeed, JPEG compression can create blockiness artifacts that may hinder PRNU estimation [2]. Every device includes 25 homogeneously lit flat-field images for the PRNU estimation. For the Vision Dataset, each device fingerprint has been computed on more than 95 JPEG flat-content images.

The images to be anonymized are selected from natural images; precisely, we pick 100 natural images per device. Regarding the Dresden Dataset, two different sub-sets can be extracted. For every device, we select 100 never-compressed Adobe Lightroom images, together with other 100 taken from the pool of JPEG compressed images. We end up with three distinct datasets comprising 600 images each: the Dresden uncompressed dataset, called \mathcal{D}_u ; the Dresden compressed dataset, defined as \mathcal{D}_c ; and the Vision (compressed) dataset, \mathcal{V} .

4.2 State-of-the-art solutions

We select the most recent anonymization methods proposed in the literature as state-of-the-art solutions.

Among the pool of PRNU-aware methods, we implement the method proposed in [6], being the most recent and cited contribution. We do not compare our solution with the PRNU-aware strategy recently proposed in [8], as its performance drops significantly whenever the used image denoising operator during cross-correlation tests is the commonly used one suggested in [1, 2]. Since in our proposed strategy we follow the methodology devised in [1, 2] for image denoising and cross-correlation, a comparison with [8] would be unfair.

Regarding PRNU-blind strategies, the most recent contribution is that proposed by us in [13], which demonstrates to outperform results of [12] in a PRNU-blind scenario. For the implementation of [13], we consider the parameter configurations achieving the best anonymization results, i.e., the strategies defined as $\ell_1^{(3)}$ and $\ell_1^{(5)}$ in the original paper.

Moreover, to show that simple denoising does not achieve good anonymization performances [10], we implement the well-known DnCNN denoiser [23] which represents a modern data-driven solution among image denoising strategies.

4.3 Experimental setup

Considering what was done in the past state-of-the-art [13], our experiments process images of 512×512 pixels with 512 features extracted at the first MultiRes block. To do so, we center-crop all the images and the computed PRNUs to a common resolution of 512×512 pixels. The optimization is performed through the ADAM algorithm with a learning rate of 0.001. At each iteration, we perturb the CNN input noise \mathbf{z} with additive white Gaussian noise with standard deviation 0.1 to strengthen the convergence. This way, the network is more robust with respect to the specific noise realization, and it is forced to learn higher-level features. Without any specific code optimization, we reach a computation speed of 5 iterations per second on an Nvidia Tesla V100 GPU, requiring 8 GB of GPU memory.

Concerning the proposed local post-processing and assembly in Section 3.3, notice that the amount of M available images at the output of DIP process changes accordingly to the input image and the PSNR requirement. Typical values are between 500 and 2500 images. The computational impact of the post-processing is mainly due to the Wiener filter that estimates the noise residual on all the M images. It is worth noticing that the vast majority of images need few iterations (less than 3000 iterations over 10,000 possible cycles, on average) to achieve the threshold of 39 dB chosen to stop the inversion. We consider multiple parameter configurations to include a

sufficiently broad pool of investigation cases. The block size B can be chosen among $B = [32, 64, 128, 256, 512]$ pixels, and the maximum number of averaged blocks can vary as well, being $L = [1, 5, 10, 25, 50, 75, 100]$. Notice that the case $B = 512$ corresponds to select the full image, without dividing it into blocks. It is worth noticing that the configuration $\{L = 1, B = 512\}$ coincides with the absence of the local area post-processing proposed in Section 3.3.

4.4 Evaluation metrics

After the generation of the anonymized image $\hat{\mathbf{I}}$, we compute the NCC between $\hat{\mathbf{I}}$ and the source device PRNU \mathbf{K} , together with the PSNR between $\hat{\mathbf{I}}$ and the original image \mathbf{I} . These values are the used metrics for evaluating the results and comparing them with state-of-the-art. The lower the achieved NCC together with a high PSNR, the better the image anonymization performance.

To summarize the results related to the achieved NCCs, we make use of receiver operating characteristic (ROC) curves related to the source device identification problem. Given a fixed device PRNU \mathbf{K} , NCCs of anonymized images taken with that device are defined as the positive set, while NCCs of images shot by other cameras are the negative set. Anonymization performance is evaluated through the area under the curve (AUC), as done in [8, 12, 13]. Our goal is to reduce the AUC of the curves, thus making the PRNU-based identification not working, at the same time maintaining high values of PSNR.

Finally, we propose a synthetic metric to summarize the trade-off between the image quality, measured by PSNR, and the anonymization level reached in terms of AUC. Specifically, the proposed trade-off metric \mathcal{C} is computed as follows:

$$\mathcal{C} = \text{PSNR} (1 - 2|\text{AUC} - 0.5|), \quad (12)$$

where $|\cdot|$ is the absolute value, the term inside the brackets is 0 if the AUC is 1 (i.e., no anonymization) or 1 if the AUC is 0.5 (i.e., perfect anonymization). This anonymization measure is then scaled by the visual quality factor, i.e., the PSNR. The higher the metric, the better the quality/anonymization trade-off.

5 Results and discussion

In this section, we provide the numerical results achieved with our experimental campaign that demonstrate the capability and limitations of our methodology. First, we deploy our method when the reference PRNU of the device is available at the analyst (i.e., $\mathbf{P} = \mathbf{K}$). Then, we show that our method can also be applied in the case of blind anonymization when the PRNU \mathbf{K} is unknown (i.e., $\mathbf{P} = \mathbf{W}$). The results are compared with state-of-the-art techniques to highlight the pros and cons.

5.1 PRNU-aware anonymization

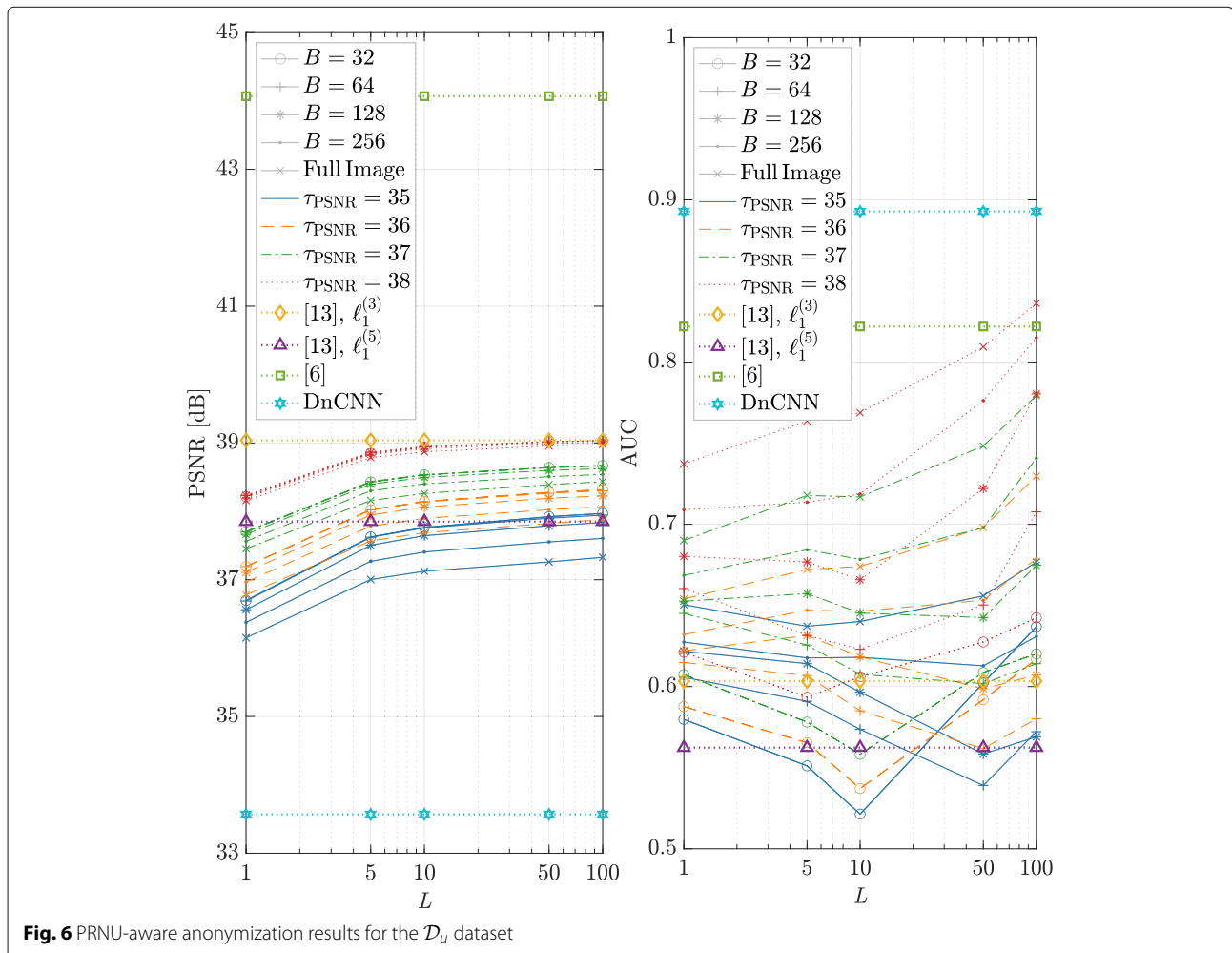
In this scenario, the actual PRNU of the source device to be anonymized is known. The results are shown in terms of PSNR and AUC of the ROC curves for the three investigated datasets. Figures 6, 7, and 8 refer to datasets \mathcal{D}_u , \mathcal{D}_c , and \mathcal{V} , respectively. For the sake of clarity in the following discussion, the results for $L = 25$ and $L = 75$ are not depicted in these plots. It is important to notice that the results must be analyzed by watching PSNR and AUC concurrently. Indeed, high PSNR is a good result only if paired with low AUC. We, therefore, privilege solutions providing a good PSNR/AUC trade-off. To ease the readability of the reported results, we separately analyze the performance of each PRNU-anonymization method in brief paragraphs.

5.1.1 Proposed DIPPAS method

For all the investigated datasets, the proposed method is able to achieve PSNRs greater than 38 dB, provided that a sufficiently high threshold τ_{PSNR} is chosen. Also, in

terms of AUCs, the proposed method can cover a wide range of possibilities, according to the chosen block size B and the amount of averaged blocks L . It is worth noticing that the application of local post-processing presented in Section 3.3 allows an improvement of the results in all the three scenarios. The DIP approach alone, which corresponds to configuration $\{L = 1, B = 512\}$, often shows low PSNRs and too high AUCs. Instead, by working on smaller image blocks, both PSNR and AUC improve, even without averaging multiple blocks together (i.e., with $L = 1$).

In general, the smaller the block size B , the better the PSNR achieved, even though this behavior seems to attenuate for high values of τ_{PSNR} . Besides, the more the amount of averaged blocks L , the better the achieved PSNR. Regarding Dresden-related datasets, middle values of L seem to work better for achieving good AUCs, while dataset \mathcal{V} requires higher values of L for lowering the AUC. The achieved AUCs are better on Dresden-related datasets, i.e., Figs. 6 and 7. For these two datasets, none of



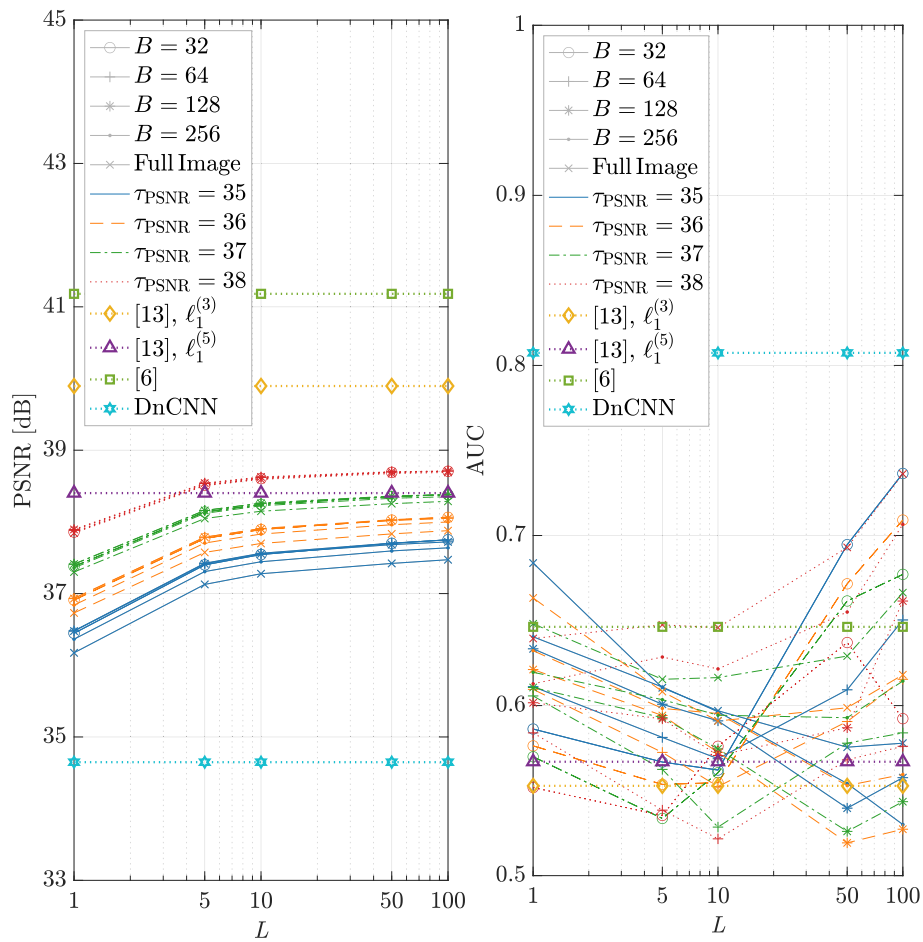


Fig. 7 PRNU-aware anonymization results for the \mathcal{D}_c dataset

the state-of-the-art works outperforms the best DIPPAS results, while the \mathcal{V} dataset seems to be more challenging to be anonymized.

5.1.2 Proposed method in [6]

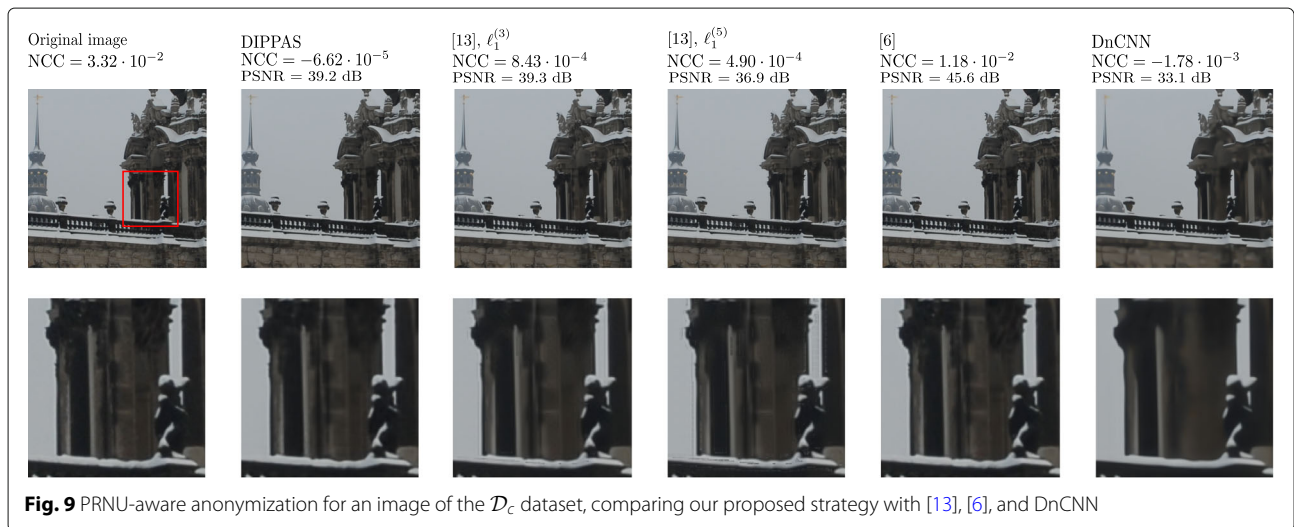
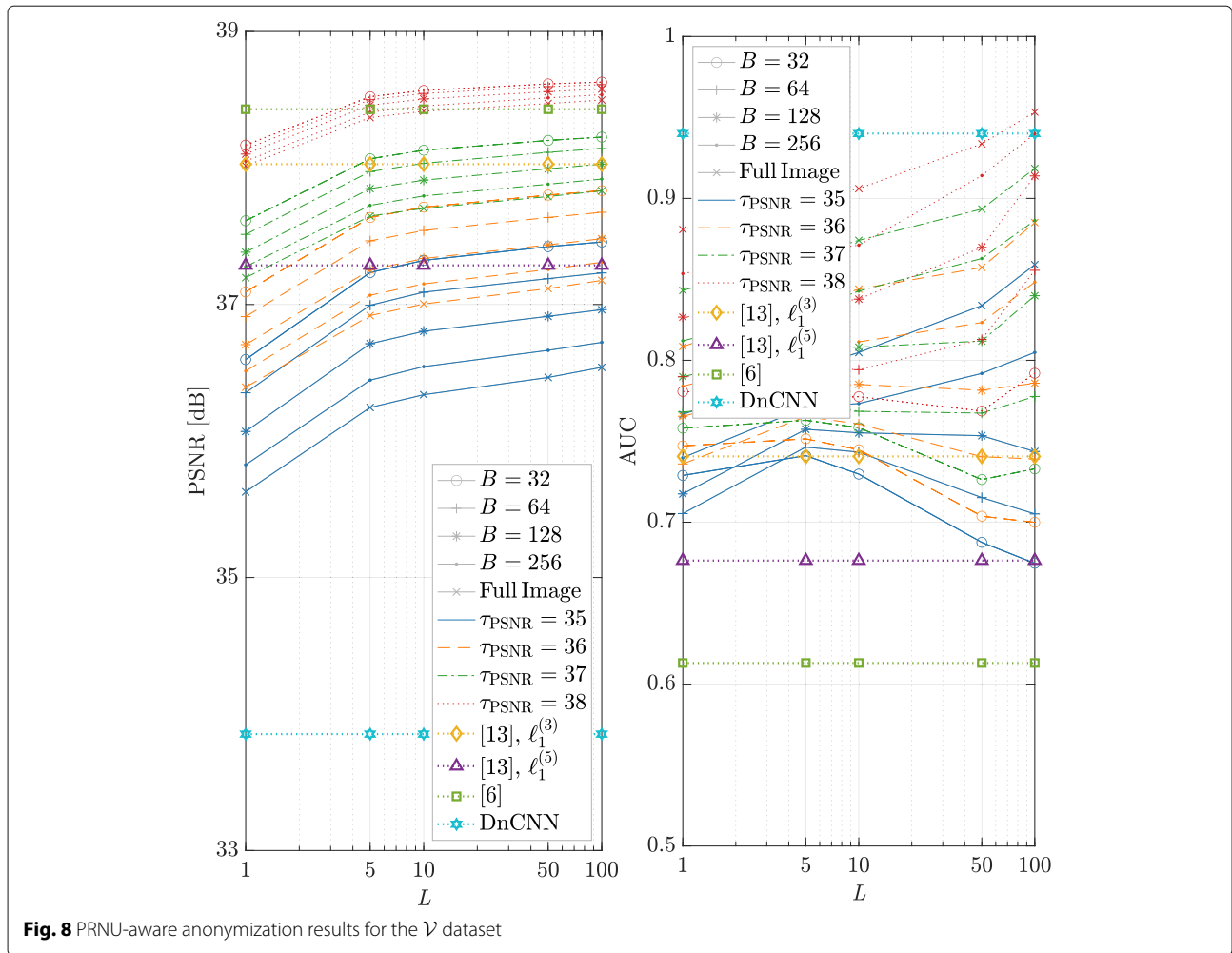
The solution provided by [6] achieves the best results in terms of PSNR for all three datasets. However, notice that the corresponding AUCs show inferior results if compared with DIPPAS and [13] for Dresden-related datasets. Concerning the dataset \mathcal{V} shown in Fig. 8, the AUC obtained by [6] seems to outperform every proposed strategy.

We think this different behavior can be explained by the diverse nature of Vision dataset with respect to Dresden. Indeed, in dataset \mathcal{V} the device PRNU is estimated directly from JPEG-compressed images, while in Dresden-based datasets the PRNU is estimated from uncompressed ones. As a matter of fact, the PRNU estimated from JPEG-compressed images can present artifacts due to JPEG compression, which can also contribute

to hinder the subtle sensor traces left on images [2]. As a consequence, anonymizing JPEG-compressed images with respect to the PRNU estimated from uncompressed data can be slightly more complicated than anonymizing JPEG images with respect to the PRNU estimated from JPEG data. In this vein, the strategy proposed by [6] seems to work in a very accurate way only if the device PRNU is estimated from JPEG-compressed images.

5.1.3 Proposed method in [13]

The proposed strategy in [13] achieves acceptable values of PSNRs in all the considered datasets, actually comparable to those achieved by DIPPAS. The resulting AUCs show satisfying values as well, except for the Vision-related dataset, where [13] seems to suffer more with respect to Dresden-related datasets, following a similar trend to that previously shown by the DIPPAS method. Regardless, notice that DIPPAS can outperform the AUCs achieved by [13] in all three datasets.



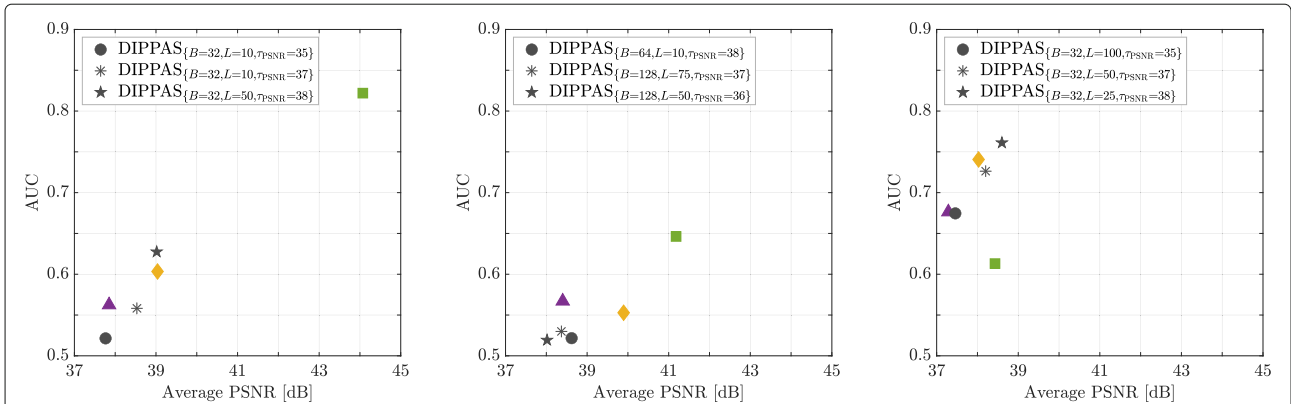


Fig. 10 PRNU-aware anonymization results, reported in terms of AUC as a function of average PSNR achieved by three DIPPAS configurations for the \mathcal{D}_u dataset (left), the \mathcal{D}_c dataset (center): the \mathcal{V} dataset (right). We compare our proposed strategy with [13] $\ell_1^{(3)}$ (yellow \diamond), [13] $\ell_1^{(5)}$ (purple \blacktriangle), and [6] (green \blacksquare)

5.1.4 Proposed method in [23]

The DnCNN solution proposed in [23] shows small values of PSNRs in all the experiments. Furthermore, the achieved results are too high AUCs, which are unacceptable for good image anonymization. The DnCNN results seem to confirm that this simple image denoiser cannot accurately delete PRNU traces [10], leading to poor anonymization performances.

Figure 9 reports an example of anonymization performed over an image of \mathcal{D}_c dataset. We depict the original image and its anonymized versions exploiting DIPPAS and the methods of [13], [6], and DnCNN [23]. Specifically, we choose the best performing DIPPAS parameter configuration in terms of both PSNR and AUC, i.e., $\{\tau_{\text{PSNR}} = 38, B = 64, L = 10\}$; we select this configuration by referring to results shown in Fig. 7. Zooming in the images (red squared area), we can visually notice that the best results are obtained by DIPPAS and [6]; the DnCNN results in a heavily smoothed image, while [13] introduces some edge artifacts. The method devised in [6] can halve the original NCC, while DIPPAS dramatically scales the original NCC value by a factor of 0.002.

To summarize the previously reported results, Fig. 10 shows the behavior of AUC as a function of the average

PSNR achieved by DIPPAS in three selected parameter configurations. We compare our results with state-of-the-art as well. The best working condition consists of high PSNR and low AUC; however, a trade-off exists between good visual quality and high anonymization. Such trade-off is expressed by a synthetic metric in Eq.(12), whose values for the three datasets are reported in Tables 1, 2, and 3. It is possible to notice that DIPPAS provides the best trade-off on the Dresden dataset and the second-best one on Vision. In this latter scenario, the best trade-off is provided by [6], which, however reports inferior results on the Dresden dataset.

5.2 PRNU-blind anonymization

In this scenario, the actual device PRNU is unknown. Therefore, the reference device fingerprint used during the DIP inversion and the blocks assembly corresponds to the noise residual extracted from the image, i.e., $\mathbf{P} = \mathbf{W}$. As previously done, we report the PSNR and AUC of the ROC curves for all three investigated datasets. Figures 11, 12, and 13 depict the results for datasets \mathcal{D}_u , \mathcal{D}_c , and \mathcal{V} , respectively. For the sake of clarity in the following discussion, the results for $L = 25$ and $L = 75$ are not depicted in these plots.

Table 1 Trade-off metric \mathcal{C} for the PRNU-aware anonymization on \mathcal{D}_u

Setup	\mathcal{C}
DIPPAS $_{B=32,L=10,\tau_{\text{PSNR}}=35}$	36.15
DIPPAS $_{B=32,L=10,\tau_{\text{PSNR}}=37}$	34.04
DIPPAS $_{B=32,L=50,\tau_{\text{PSNR}}=38}$	29.07
[13], $\ell_1^{(3)}$	30.97
[13], $\ell_1^{(5)}$	33.13
[6]	15.69

Table 2 Trade-off metric \mathcal{C} for the PRNU-aware anonymization on \mathcal{D}_c

Setup	\mathcal{C}
DIPPAS $_{B=64,L=10,\tau_{\text{PSNR}}=38}$	36.95
DIPPAS $_{B=128,L=75,\tau_{\text{PSNR}}=37}$	36.07
DIPPAS $_{B=128,L=50,\tau_{\text{PSNR}}=36}$	36.55
[13], $\ell_1^{(3)}$	35.68
[13], $\ell_1^{(5)}$	33.26
[6]	29.13

Table 3 Trade-off metric \mathcal{C} for the PRNU-aware anonymization on \mathcal{V}

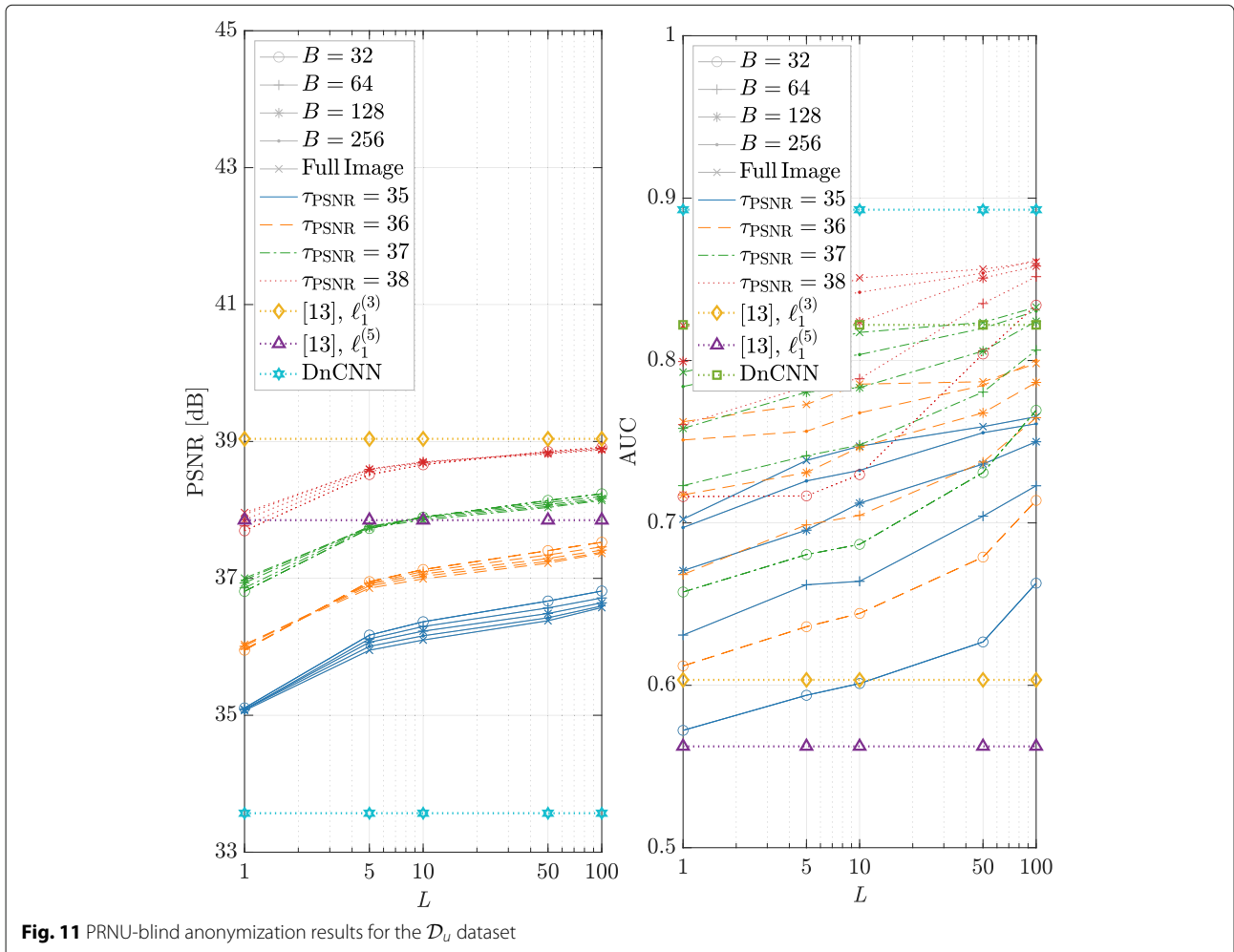
Setup	\mathcal{C}
DIPPAS $_{B=32,L=100,\tau_{PSNR}=35}$	24.38
DIPPAS $_{B=32,L=50,\tau_{PSNR}=37}$	20.91
DIPPAS $_{B=32,L=25,\tau_{PSNR}=38}$	18.44
[13], $\ell_1^{(3)}$	19.72
[13], $\ell_1^{(5)}$	24.14
[6]	29.74

In terms of PSNR, on the Vision Dataset, we can outperform [13], while for the Dresden Dataset, we achieve slightly lower results. Moreover, DIPPAS achieves slightly higher AUCs than state-of-the-art solutions. Notice that the best AUC values are obtained for $L = 1$, i.e., without performing block averaging.

We think this less effective anonymization with respect to the previous PRNU-aware scenario can be due to the

assumption done during the DIP inversion (9) to estimate the anonymized image. As a matter of fact, the DIP paradigm leverages the PRNU-based image modeling reported in (1). Whenever the PRNU estimate \mathbf{K} is unknown and the noise residual \mathbf{W} is used instead, as reported in (11), the model is not clearly satisfied and the DIP solution will be sub-optimum. For this reason, in a PRNU-blind scenario, DIPPAS is still able to achieve good performance in terms of PSNR, but the AUC is not as good as in the PRNU-aware scenario.

From these results, it may seem that DIPPAS cannot achieve better results than [13], but an important point has to be noticed. Indeed, [13] applies different processing to edges and flat regions, thus removing PRNU traces in concentrated local areas. Precisely, notice that method [13] works in two separate steps: (i) estimate an anonymized version of the image exploiting inpainting techniques and (ii) substitute a denoised version of the edges extracted from the original image into the anonymized image to enhance the output visual quality. In light of these considerations, we think that the edge



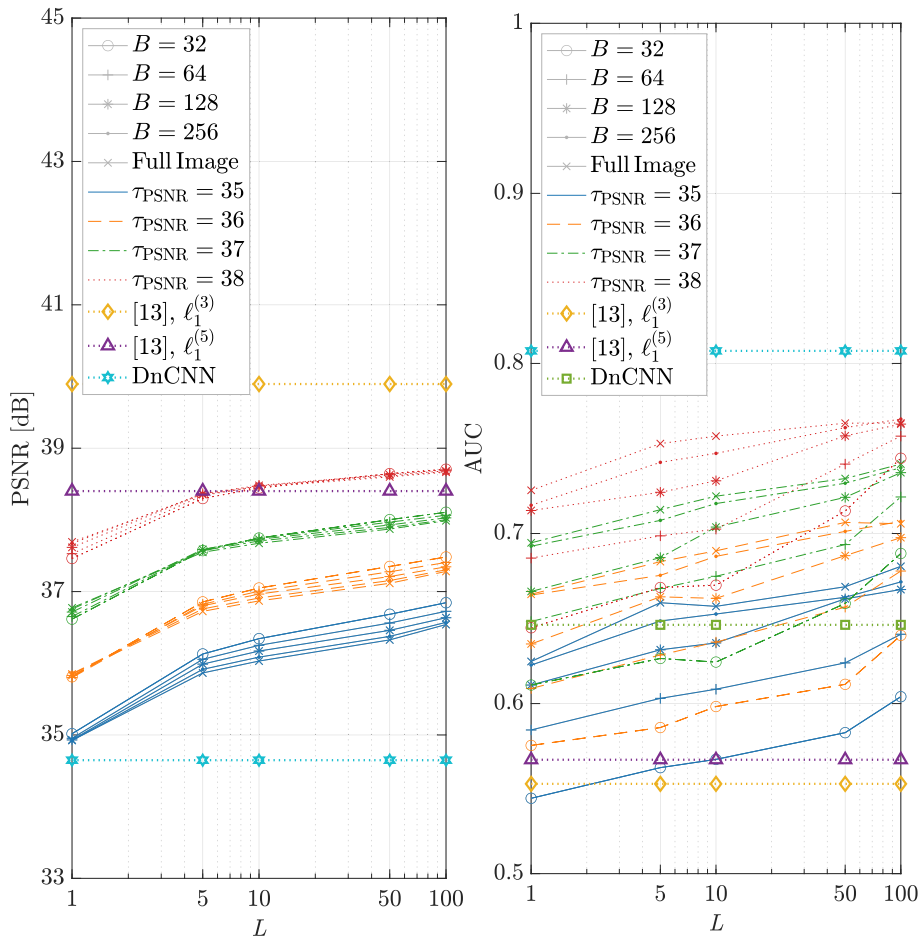


Fig. 12 PRNU-blind anonymization results for the \mathcal{D}_c dataset

processing operation performed on the output image can be the weak link in the proposed pipeline of [13]. Indeed, image edges only undergo two successive steps of BM3D denoising algorithm [34]; thus, they reasonably contain enough PRNU traces for performing source attribution, as suggested in [10].

Therefore, we compare DIPPAS and [13] only along the image edges, extracted following the same pipeline proposed in [13]. For every dataset, we evaluate the DIPPAS results for a parameter configuration which returns the nearest PSNR value to that achieved by [13]. For instance, looking at Fig. 11, dataset \mathcal{D}_u is evaluated for $\{B = 32, L = 100, \tau_{\text{PSNR}} = 38\}$ if compared to $[13], \ell_1^{(3)}$; we use $\{B = 512, L = 10, \tau_{\text{PSNR}} = 37\}$ when comparing to $[13], \ell_1^{(5)}$.

We compare the results in terms of relative change of AUC evaluated over image edges with respect to the AUC achieved on the full image. In a nutshell, the relative change in AUC can be computed as $(\text{AUC}_{\text{edges}} - \text{AUC})/\text{AUC}$, being AUC the metrics associated to the full image. Table 4 reports the results. Notice that the relative

AUC change maintains a coherent behavior for all three datasets. On one side, [13] always reports a positive relative change; on the other side, DIPPAS presents a negative relative change.

DIPPAS results are coherent with what happens on natural images when compared with the PRNU in a reduced region (e.g., only along the edges). Indeed, the NCC drops as the image content is reduced. Thus, the AUC of the source attribution problem decreases. On the contrary, the accuracy of [13] evaluated only along image edges firmly drops as the NCC increases, with a consequent AUC growth. The previously reported consideration can explain this phenomenon, that is, [13] performs only denoising along the edges, and this is usually not enough to hinder PRNU traces [10].

As a consequence, the anonymization algorithm proposed in [13] can be easily spotted and defeated just by analyzing image edges. In contrast, the suggested DIPPAS solution does not present this drawback. Even if an analyst only uses edges for PRNU-based attribution, the image

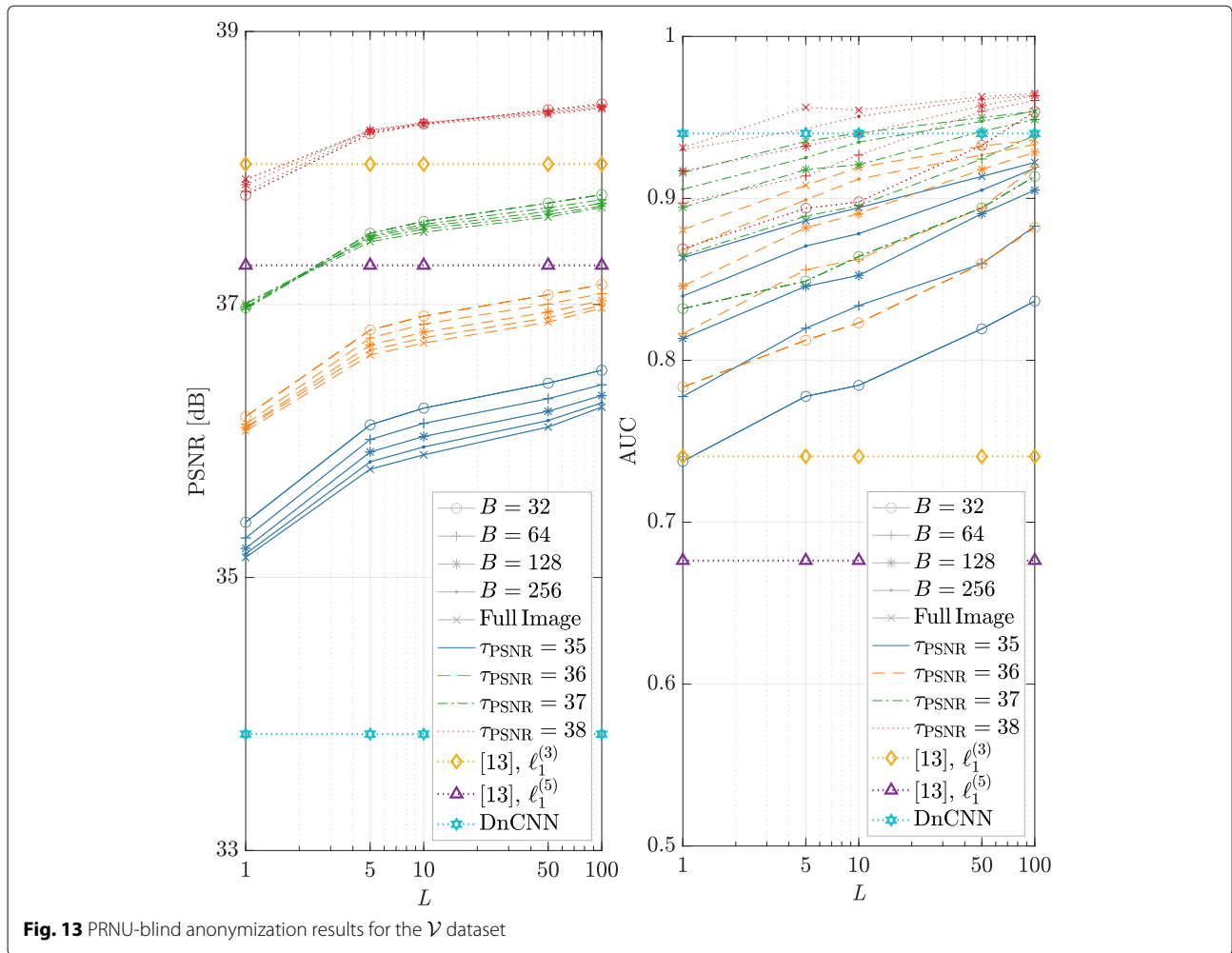


Fig. 13 PRNU-blind anonymization results for the \mathcal{V} dataset

would look anonymized. This is an additional advantage of the DIPPAS technique.

5.3 Computational cost

In the following, and the revised manuscript, we report the computation time of single image anonymization, taken from \mathcal{D}_u dataset, in the case of PRNU-aware scenario.

The machine is a dual-socket workstation equipped with Intel Xeon Gold 6246 CPUs, 256 GB of memory, and a single Nvidia Tesla V100 PCIe GP-GPU. The operative system is Ubuntu Server 20.04 LTS. Although the developed code is parallelized over all the available cores, we report the CPU time in user mode.

First of all, the image generation through deep priors took 4262 iterations to reach the PSNR stopping threshold of 42 dB, for a total time of $\approx 850s$ (0.2s/it).

Having set $\tau_{PSNR} = 38$ dB, the number of generated images is $M = 1355$. Table 5 reports the computation time of the noise extraction step (first row) and the post-processing step (subsequent rows) for two val-

ues of block size $B = 512$ (i.e., full image), and $B = 64$ (i.e., 16 blocks per image). It is worth noticing that, for both values of B , extracting the noise through the Wiener filter takes an order of magnitude more than the NCC sorting and merging operations. Moreover, varying L does not significantly impact the computation time, which is practically affected by input/output operations only.

Skipping the post-processing means computing the NCC (with $B = 512$) for images that have a $PSNR \geq \tau_{PSNR}$ and taking the image associated with the minimum absolute value of the NCC. Indeed, randomly selecting an image such that $PSNR \geq \tau_{PSNR}$ does not guarantee that the associated NCC will be enough to fool a device attribution test, as depicted in Fig. 2.

Either way, the bottleneck of the proposed methodology is the NCC computation, which the final image building depends on. This limitation can be overcome by incorporating a reliable NCC term in the loss function. In the light of these results, we can state that post-processing helps the anonymization.

Table 4 AUC relative change on image edges, for PRNU-blind anonymization

NCC area	[13], $\ell_1^{(3)}$	DIPPAS	NCC area	[13], $\ell_1^{(5)}$	DIPPAS
\mathcal{V}_{edges}	+11.6%	-9.1%	\mathcal{V}_{edges}	+22.3%	-9.9%
$\mathcal{D}_{u_{edges}}$	+10.9%	-17.8%	$\mathcal{D}_{u_{edges}}$	+19.1%	-19.1%
$\mathcal{D}_{c_{edges}}$	+8.7%	-17.6%	$\mathcal{D}_{c_{edges}}$	+6.1%	-17.7%

6 Conclusions

In this manuscript, we propose a source device anonymization scheme that leverages the deep image prior (DIP) paradigm to attenuate PRNU traces in natural images, paired with a post-processing scheme that exploits multiple images. With this method, a CNN learns to generate images from noise realizations iteratively. Specifically, at each DIP iteration: (i) the CNN generates an image; (ii) we inject the device PRNU into this image; (iii) we minimize the distance between the input query image and the PRNU-injected image. In doing so, we can generate multiple images with a strongly attenuated PRNU pattern and high visual quality. Finally, we devise an efficient post-processing operation for assembling the final anonymized image from the CNN outputs realized at different iterations.

We compare our method against state-of-the-art anonymization schemes through numerical examples. In particular, when the PRNU of the device is available, we achieve our best results. Our scheme can be generalized to the case of blind anonymization, i.e., when the device PRNU is unknown, and only a noise residual can be extracted from the query image and then injected into the DIP generation process.

Not surprisingly, our method suffers when the injected noise is quite different from the source device PRNU. However, it still proves interesting when compared to state-of-the-art solutions if we consider the homogeneity of PRNU removal effect. Indeed, we are capable of removing PRNU traces on all image regions, whereas the considered baseline leaves image edges mainly non-anonymized.

Our future work will be devoted to investigating the possibility of starting from a pre-trained network to speed-up convergence. Moreover, we will focus on a better inversion model to be used in case of blind PRNU removal.

Table 5 CPU time spent in user mode [s]

	B = 512	B = 64
Noise extraction	187	554
Post-processing, $L = 1$	18	26
Post-processing, $L = 10$	19	26
Post-processing, $L = 50$	18	27
Post-processing, $L = 100$	19	27

Acknowledgements

This work was supported by the PREMIER project, funded by the Italian Ministry of Education, University, and Research within the PRIN 2017 program. This material is based on research sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL) under agreement number FA8750-20-2-1004. The US government is authorized to reproduce and distribute reprints for governmental purposes, notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors. They should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA and AFRL or the US government. Hardware support was generously provided by the NVIDIA Corporation.

Authors' contributions

All authors contributed to the design of the proposed method and writing the manuscript. All authors read and approved the final manuscript.

Funding

The authors have no funding to be declared.

Availability of data and materials

All the numeric experiments shown in this manuscript rely on publicly available datasets [15, 16]. The codes have been released on GitHub: https://github.com/polimi-ispl/dip_prnu_anonymizer.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 9 June 2021 Accepted: 23 January 2022

Published online: 14 February 2022

References

1. J. Lukáš, J. Fridrich, M. Goljan, Digital camera identification from sensor pattern noise. *IEEE Trans. Inf. Forensics Secur. (TIFS)*. **1**, 205–214 (2006)
2. M. Chen, J. Fridrich, M. Goljan, J. Lukas, Determining image origin and integrity using sensor noise. *IEEE Trans. Inf. Forensics Secur. (TIFS)*. **3**, 74–90 (2008)
3. M. Kirchner, C. Johnson, in *IEEE International Workshop on Information Forensics and Security*. SPN-CNN: boosting sensor-based source camera attribution with deep learning (IEEE, New York, 2019)
4. S. Mandelli, D. Cozzolino, P. Bestagini, L. Verdoliva, S. Tubaro, CNN-based fast source device identification. *IEEE Signal Process. Lett.* **27**, 1285–1289 (2020)
5. M. Chen, J. Fridrich, M. Goljan, J. Lukas, in *SPIE Electronic Imaging (EI)*. Source digital camcorder identification using sensor photo-response nonuniformity (SPIE, Bellingham, 2007)
6. A. Karaküçük, A. E. Dirik, Adaptive photo-response non-uniformity noise removal against image source attribution. *J Digit. Investig.* **12**, 66–76 (2015)
7. H. Zeng, J. Chen, X. Kang, W. Zeng, in *IEEE International Conference on Image Processing (ICIP)*. Removing camera fingerprint to disguise photograph source (IEEE, New York, 2015)
8. N. Bonettini, L. Bondi, D. Güera, S. Mandelli, P. Bestagini, S. Tubaro, E. J. Delp, in *2018 26th European Signal Processing Conference (EUSIPCO)*. Fooling PRNU-based detectors through convolutional neural networks (IEEE, New York, 2018), pp. 957–961
9. K. Rosenfeld, H. T. Sencar, in *IS&T/SPIE Electronic Imaging (EI)*. A study of the robustness of PRNU-based camera identification (SPIE, Bellingham, 2009)
10. J. Bernacki, On robustness of camera identification algorithms. *Multimedia Tools Appl.* **80**, 921–942 (2021). <https://link.springer.com/article/10.1007%2Fs11042-020-09133-9>
11. A. E. Dirik, H. T. Sencar, N. Memon, Analysis of seam-carving-based anonymization of images against PRNU noise pattern-based source attribution. *IEEE Trans. Inf. Forensics Secur. (TIFS)*. **9**, 2277–2290 (2014)
12. J. Entrieri, M. Kirchner, in *IS&T Electronic Imaging (EI)*. Patch-based desynchronization of digital camera sensor fingerprints (IS&T, Springfield, 2016)

13. S. Mandelli, L. Bondi, S. Lameri, V. Lipari, P. Bestagini, S. Tubaro, in *IEEE International Conference on Image Processing (ICIP)*. Inpainting-based camera anonymization (IEEE, New York, 2017)
14. D. Ulyanov, A. Vedaldi, V. Lempitsky, in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Deep image prior (IEEE, New York, 2018)
15. T. Gloe, R. Böhme, The Dresden image database for benchmarking digital image forensics. *J Dig. Forensic Pract.* **3**, 150–159 (2010)
16. D. Shullani, M. Fontani, M. Luliani, O. A. Shaya, A. Piva, Vision: a video and image dataset for source identification. *EURASIP J. Inf. Secur.* **15** (2017). <https://jis-eurasipjournals.springeropen.com/articles/10.1186/s13635-017-0067-2>
17. P. Moulin, J. Liu, Analysis of multiresolution image denoising schemes using generalized gaussian and complexity priors. *IEEE Trans. Inf. Theory.* **45**(3), 909–919 (1999). <https://doi.org/10.1109/18.761332>
18. T. Tirer, R. Giryes, Back-projection based fidelity term for ill-posed linear inverse problems. *IEEE Trans. Image Process.* **29**, 6164–6179 (2020). <https://doi.org/10.1109/TIP.2020.2988779>
19. W. Dong, P. Wang, W. Yin, G. Shi, F. Wu, X. Lu, Denoising prior driven deep neural network for image restoration. *IEEE Trans. Patt. Anal. Mach. Intell.* **41**(10), 2305–2318 (2019). <https://doi.org/10.1109/TPAMI.2018.2873610>
20. B. Lin, X. Tao, J. Lu, Hyperspectral image denoising via matrix factorization and deep prior regularization. *IEEE Trans. Image Process.* **29**, 565–578 (2020). <https://doi.org/10.1109/TIP.2019.2928627>
21. H. K. Aggarwal, M. P. Mani, M. Jacob, Modl: model-based deep learning architecture for inverse problems. *IEEE Trans. Med. Imaging.* **38**(2), 394–405 (2019). <https://doi.org/10.1109/TMI.2018.2865356>
22. O. Ronneberger, P. Fischer, T. Brox, in *International Conference on Medical Image Computing and Computer-assisted Intervention*. U-Net: convolutional networks for biomedical image segmentation (Springer, Cham, 2015), pp. 234–241
23. K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, Beyond a gaussian denoiser: residual learning of deep CNN for image denoising. *IEEE Trans. Image Process. (TIP)*. **26**(7), 3142–3155 (2017). <https://doi.org/10.1109/TIP.2017.2662206>
24. C. Cruz, A. Foi, V. Katkovnik, K. Egiazarian, Nonlocality-reinforced convolutional neural networks for image denoising. *IEEE Signal Process. Lett.* **25**(8), 1216–1220 (2018). <https://doi.org/10.1109/LSP.2018.2850222>
25. F. Kong, F. Picetti, V. Lipari, P. Bestagini, S. Tubaro, in *SEG Technical Program Expanded Abstracts 2020*. Deep prior-based seismic data interpolation via multi-res U-Net, (2020), pp. 3159–3163. <https://doi.org/10.1190/segam2020-3426173.1>
26. S. Mandelli, F. Borra, V. Lipari, P. Bestagini, A. Sarti, S. Tubaro, in *SEG Technical Program Expanded Abstracts 2018*. Seismic data interpolation through convolutional autoencoder, (2018), pp. 4101–4105. <https://doi.org/10.1190/segam2018-2995428.1>
27. Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, J. Liang, Unet++: redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Trans. Med. Imaging.* **39**(6), 1856–1867 (2020). <https://doi.org/10.1109/TMI.2019.2959609>
28. R. E. Jurdi, C. Petitjean, P. Honeine, F. Abdallah, Bb-unet: U-Net with bounding box prior. *IEEE J. Sel. Top. Signal Process.* **14**(6), 1189–1198 (2020). <https://doi.org/10.1109/JSTSP.2020.3001502>
29. O. Sidorov, J. Y. Hardeberg, in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. Deep hyperspectral prior: single-image denoising, inpainting, super-resolution, (2019), pp. 3844–3851. <https://doi.org/10.1109/ICCVW.2019.00477>
30. F. Picetti, V. Lipari, P. Bestagini, S. Tubaro, Seismic image processing through the generative adversarial network. *Interpretation.* **7**(3), 15–26 (2019). <https://doi.org/10.1190/INT-2018-0232.1>
31. F. Devoti, C. Parera, A. Lieto, D. Moro, V. Lipari, P. Bestagini, S. Tubaro, in *SEG Tech. Program Expanded Abstr.* 2019. Wavefield compression for seismic imaging via convolutional neural networks, (2019), pp. 2227–2231. <https://doi.org/10.1190/segam2019-3216395.1>
32. N. Ibtehaz, M. S. Rahman, Multiresunet: rethinking the U-Net architecture for multimodal biomedical image segmentation. *Neural Netw.* **121**, 74–87 (2020)
33. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Going deeper with convolutions (IEEE, New York, 2015)
34. K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3D transform-domain collaborative filtering. *IEEE Trans. Image Process. (TIP)*. **16**, 2080–2095 (2007)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)