Fooling PRNU-Based Detectors Through Convolutional Neural Networks

Nicolò Bonettini, Luca Bondi,
Sara Mandelli, Paolo Bestagini, Stefano Tubaro
Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano, Milan, Italy

David Güera,
Edward J. Delp
School of Electrical and Computer Engineering
Purdue University, West Lafayette, USA

***Abstract*—In the last few years, forensic researchers have developed a wide set of techniques to blindly attribute an image to the device used to shoot it. Among these techniques, those based on photo response non uniformity (PRNU) have shown incredibly accurate results, thus they are often considered as a reference baseline solution. The rationale behind these techniques is that each camera sensor leaves on acquired images a characteristic noise pattern. This pattern can be estimated and uniquely mapped to a specific acquisition device through a cross-correlation test. In this paper, we study the possibility of leveraging recent findings in the deep learning field to attack PRNU-based detectors. Specifically, we focus on the possibility of editing an image through convolutional neural networks in a visually imperceptible way, still hindering PRNU noise estimation. Results show that performing such an attack is possible, even though an informed forensic analyst can reduce its impact through a smart test.**

I. Introduction

Image source attribution techniques are widely studied in multimedia forensics under two different but complementary aspects: camera model identification and camera device identification [1].

Camera model identification aims at finding which camera brand/model shot a specific picture. Techniques that exploit different digital traces left on the captured images have been developed for this task. These solutions can make use of the traces left by color filter array (CFA) interpolation [2], [3], histogram equalization [4], statistical descriptors paired with machine-learning classifiers [5], [6], as well as convolutional neural networks (CNNs) [7], [8]. By contrast, camera device identification techniques are mainly based on Photo-Response Non Uniformity (PRNU), i.e., a characteristic of image acquisition sensors left on photographs as a noise pattern [9], [10] that can be estimated for each specific sensor. If an image needs to be attributed to a camera, a correlation test is run between a noise trace extracted from the image and different candidate PRNUs. PRNU is also robust against editing operations such as compression, cropping and resizing [10], making camera attribution possible even in complex situations.

Image source attribution has proved to be powerful enough to bind a specific picture to the device that shot it. Increased privacy concerns make the need for effective anonymization methods even more pressing. Studying the boundaries of image anonymization can enable analysts to be aware of the robustness of camera attribution methods in the presence of malicious attacks. For these reasons, device anonymization techniques tailored to remove or hinder PRNU traces have been developed. These techniques can be divided into two different families: i) methods that require the knowledge of camera PRNU [11], [12]; ii) blind methods based solely on the image under analysis [13], [14], [15].

In this paper we explore the possibilities offered by CNNs in terms of camera device anonymization based on the knowledge of the reference PRNU. An image-wise anonymization loop is built upon a CNN-based noise extractor. An autoencoder inspired fully-convolutional neural network is trained as anonymization function via back-propagation, exploiting the possibilities offered by a recently introduced CNN-based denoising method [16].

The proposed use of a CNN is different from the typical one. Instead of training a CNN on many images to learn a generalizable method, we "overfit" the proposed CNN on each single image to be anonymized. In other words, we consider the CNN as a parametric operator. We build a loss function to be minimized in order to estimate the CNN parameters. The CNN training is seen as an iterative way of minimizing the CNN loss for each given image.

Our results on 600 raw images from the Dresden Image Database [17] show that image anonymization is possible without hindering image quality. However, depending on the denoising operator and specific correlation test adopted by the forensic analyst, it is still possible to attenuate the attack power.

II. Background and Problem Formulation

In this section, we introduce the problem of image device attribution, and we provide a formal definition of the goal of this work, i.e., image device anonymization.

Device attribution Given an image $\mathbf{I}$ and a generic noise extraction function $\mathcal{N}$, we define $\mathbf{W} = \mathcal{N}(\mathbf{I})$ as the noise residual extracted from $\mathbf{I}$. Given a camera device characterized

This material is based on research sponsored by DARPA and Air Force Research Laboratory (AFRL) under agreement number FA8750-16-2-0173. The U.S. 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 and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA and Air Force Research Laboratory (AFRL) or the U.S. Government.

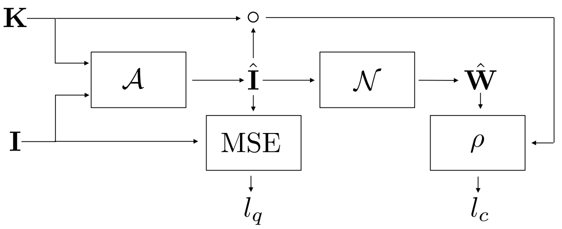


Fig. 1: Architecture of the proposed system. An anonymization function $\mathcal{A}$ is fed with the input image $\mathbf{I}$ and the relative camera PRNU $\mathbf{K}$. The anonymized image $\hat{\mathbf{I}}$ is used to compute a quality loss l_q based on the Mean Squared Error (MSE) between $\hat{\mathbf{I}}$ and $\mathbf{I}$. The noise residual $\hat{\mathbf{W}}$, extracted through a noise extraction function $\mathcal{N}$ from $\hat{\mathbf{I}}$, is used together with the camera PRNU $\mathbf{K}$ to determine a correlation loss l_c.

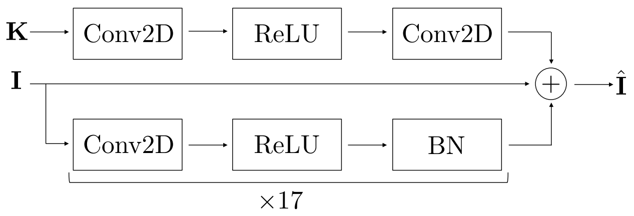


Fig. 2: Structure of the proposed CNN-based anonymization function $\mathcal{A}$. The input image $\mathbf{I}$ is processed through a set of 17 convolutional layers (Conv2D) followed by ReLU non-linearity and Batch Normalization (BN). The reference PRNU $\mathbf{K}$ is processed with two convolutional layers separated by a ReLU non-linearity. The output anonymized image $\hat{\mathbf{I}}$ results from the sample-wise algebraic sum of the input image $\mathbf{I}$ and the two fully-convolutional branches.

by a PRNU fingerprint $\mathbf{K}$, we predict that $\mathbf{I}$ has been taken by the same camera model if [9]:

$$\rho\left(\mathbf{W}, \mathbf{K} \circ \mathbf{I}\right) > \tau, \tag{1}$$

where $\circ$ represents the Hadamard (sample-wise) product, ρ is a normalized cross-correlation function and τ is a threshold set in order to bound false-detection probability below a confidence value α.

Device anonymization In order to anonymize an image $\mathbf{I}$, we propose an anonymization function $\mathcal{A}\left(\mathbf{I}, \mathbf{K}\right)$ that generates an anonymized version of $\mathbf{I}$, namely $\hat{\mathbf{I}} = \mathcal{A}\left(\mathbf{I}, \mathbf{K}\right)$. As shall be clear from the next section, the design of $\mathcal{A}$ is such that the Peak Signal-to-Noise Ratio (PSNR) between $\mathbf{I}$ and $\hat{\mathbf{I}}$ is greater than a reference value while the normalized cross-correlation between the noise residual $\hat{\mathbf{W}}$ extracted from $\hat{\mathbf{I}}$ and $\mathbf{K}$ is minimized. In an optimal case, it will result that $\rho\left(\hat{\mathbf{W}}, \mathbf{K} \circ \hat{\mathbf{I}}\right) < \tau$, so that it is not possible anymore to bind the anonymized image $\hat{\mathbf{I}}$ to its camera device with confidence α.

III. Proposed Image Anonymization Method

The proposed anonymization method is based on the idea of minimizing a cost function made up of two components: i) a measure of the difference between the input image $\mathbf{I}$ and its anonymized version $\hat{\mathbf{I}}$; ii) the cross-correlation between the anonymized noise residual $\hat{\mathbf{W}}$ and the camera PRNU $\mathbf{K}$.

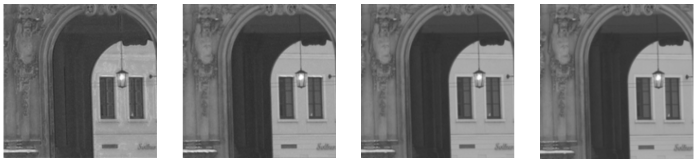
Figure 1 shows the overall working scheme. An image $\mathbf{I}$ and the corresponding camera PRNU $\mathbf{K}$ are fed as input to the anonymization function $\mathcal{A}$. The output of $\mathcal{A}$ is an anonymized version of $\mathbf{I}$, namely $\hat{\mathbf{I}}$. The Mean Square Error (MSE) between $\mathbf{I}$ and $\hat{\mathbf{I}}$ is computed and stored into l_q, the first component of the global cost function. The anonymized image $\hat{\mathbf{I}}$ is fed as input to the noise extraction function $\mathcal{N}$ and the output $\hat{\mathbf{W}}$ is correlated with the sample-wise product between $\mathbf{K}$ and $\hat{\mathbf{I}}$ to get l_c, the second component of the global cost function. The global cost function l is then defined as $l = (1 - \beta) \cdot l_q + \beta \cdot l_c$, where β is a weighting parameter tailored at balancing the trade-off between image quality and anonymization performance.

In the depicted scheme, $\mathcal{N}$ is a fixed noise extractor, whereas $\mathcal{A}$ is a denoising function learned independently on every pair $(\mathbf{I}, \mathbf{K})$ provided as input. We require both $\mathcal{N}$ and $\mathcal{A}$ to support gradient computation so that it is possible to learn via back-propagation the parameters of $\mathcal{A}$ as a function of the overall cost function l.

To satisfy the gradient computation capability for $\mathcal{N}$ we resort to DnCNN [16], a fully-convolutional neural network that shows noise extraction capabilities comparable with the Wavelet-based filtering approach commonly used for PRNU-based image source attribution. DnCNN is composed by a set of 17 convolutional layers composed by 64 filters each with kernel size equal to 3 and padding 1, each followed by ReLU non linearity and batch normalization. The fully-convolutional nature of the network does not require as input a fixed size image and produces as output a noise residual with the same size of the input image.

As for the choice of $\mathcal{A}$, we exploit an autoencoder structure similar to DnCNN, as depicted in Figure 2. The input image $\mathbf{I}$ is processed by a set of 17 convolutional layers (Conv2D), each followed by ReLU non-linearity and batch normalization (BN). The reference PRNU $\mathbf{K}$ is fed to a convolutional layer, followed by a ReLU and yet another convolutional layer. The final anonymized image $\hat{\mathbf{I}}$ results from the sum of the two convolutional processing branches together with the input image itself. The weights of the convolutional layers and the parameters of batch normalization for $\mathcal{A}$ are learned for every single image via back-propagation, driven by the global cost function l.

The image-wise anonymization process follows as from Algorithm 1. An input Image $\mathbf{I}$, a reference PRNU $\mathbf{K}$ and a minimum desired Peak Signal-to-Noise Ratio (PSNR_{min}) are provided as input. The loss weighting factor β is initialized at 0.1. At every iteration the anonymized image $\hat{\mathbf{I}}$ is first computed, together with the MSE loss l_q and the PSNR P with respect to the original image. Then the noise extraction function $\mathcal{N}$ is used to extract a noise residual $\hat{\mathbf{W}}$ from the anonymized image and compute the cross-correlation loss l_c. The global loss l is computed according to the weighting factor β. As all operations in $\mathcal{A}$ are differentiable, it is possible to back-propagate the error and modify $\mathcal{A}$ parameters to minimize



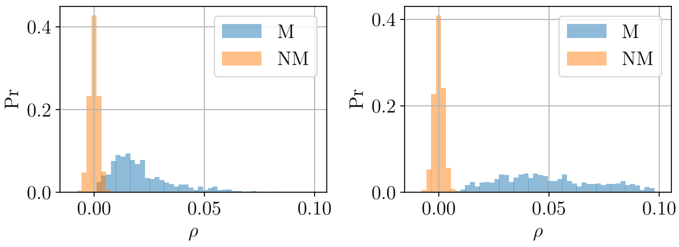
$i = 30$ $\rho = 0.122$ $P = 27.3$

$i = 410$ $\rho = 0.012$ $P = 34.3$

$i = 970$ $\rho = 0.004$ $P = 42.2$

$\mathbf{I}$

Fig. 3: Iterations of the proposed algorithm on a sample image. From left to right the evolution of $\hat{\mathbf{I}}$ at $i = \{30, 310, 970\}$ with cross-correlation ρ decreasing and PSNR P increasing. The rightmost picture is the original image $\mathbf{I}$.



(a) $\mathcal{N}_{\text{dn}}$

(b) $\mathcal{N}_{\text{wl}}$

Fig. 4: Distribution of normalized cross-correlation values on pristine images using DnCNN ($\mathcal{N}_{\text{dn}}$) and Wavelet ($\mathcal{N}_{\text{wl}}$) noise extractors, for matching image-PRNU pairs (M) and non-matching pairs (NM).

loss with any iterative optimization algorithm (e.g., stochastic gradient descent in our implementation). Once every 300 iterations if the PSNR value P is smaller than the desired minimum value PSNR_{min} the weighting factor β is reduced by a factor 4, to raise the importance of the MSE loss l_q over the cross-correlation loss l_c. If the current PSNR is greater than the desired minimum and the cross-correlation loss is small enough (i.e., $l_c < 10^{-4}$), the current anonymized image $\hat{\mathbf{I}}$ is returned and the optimization stops. At most 3000 iterations of the algorithm are performed, in order to bound the required anonymization time if the early stop condition is not met. A sample of the evolution of $\hat{\mathbf{I}}$, ρ and P over the iteration is provided in Figure 3.

IV. Experimental Setup

To state the effectiveness of the proposed approach, we resort to the same dataset and metrics used in [15], [14]. The dataset is composed of 600 raw natural images, demosaicked with Adobe Lightroom, randomly selected from 6 cameras (Nikon D70, Nikon D70s, Nikon D200, two devices each) from the Dresden Image Database [17]. All the images are cropped in their center to a fixed size of 512×512 pixels. We evaluate the anonymization performance by using two different noise extraction functions: i) the DnCNN function used as noise extractor within the anonymization scheme, denoted as $\mathcal{N}_{\text{dn}}$; ii) the Wavelet-based noise exaction function [18] commonly used in PRNU-based works, denoted as $\mathcal{N}_{\text{wl}}$. As for the use of DnCNN as noise extractor, we resort to the pre-trained model available from [16]. We resort to Pytorch [19] as Deep Learning and CNN framework.

The reference PRNU $\mathbf{K}$ for each device is estimated from 25 raw flatfield images from the same database, according to $\mathcal{N}_{\text{wl}}$ as from [9]. All the 600 images are anonymized by varying the PSNR_{min} parameter in the set of values $\{37, 38, 39, 40, 41\}$. Each anonymized image is stored as an uncompressed PNG file, thus being quantized to 8-bit as in real case scenario. For each value of PSNR_{min} we observe the distribution of the obtained PSNR values. Noise residuals are extracted with $\mathcal{N}_{\text{dn}}$ and $\mathcal{N}_{\text{wl}}$ for each anonymized image and than correlated with the 6 camera PRNUs. For each PSNR_{min} we compute a Receiver-Operating-Characteristic (ROC) by varying the value of τ, the threshold used in the cross-correlation test to detect an image as being shot from a specific camera. From each ROC we extract both the True-Positive Rate value at a False Alarm probability $\alpha = 0.01$ (TPR@0.01) and the Area Under Curve (AUC). Small AUC values indicate good anonymization performance. Small TPR@0.01 values indicate that when accepting a small false-alarm probability it is not possible to bind the picture to its camera device.

Algorithm 1 Image-wise anonymization process

```
Require: $\mathbf{I}$, $\mathbf{K}$, $\text{PSNR}_{\text{min}}$
  $\beta \leftarrow 0.1$
  for $i$ in $\{1, \dots, 3000\}$ do
      $\hat{\mathbf{I}} \leftarrow \mathcal{A}(\mathbf{I}, \mathbf{K})$
      $l_q \leftarrow \text{MSE}(\mathbf{I}, \hat{\mathbf{I}})$
      $P \leftarrow \text{PSNR}(\mathbf{I}, \hat{\mathbf{I}})$
      $\hat{\mathbf{W}} \leftarrow \mathcal{N}(\hat{\mathbf{I}})$
      $l_c \leftarrow \mid \rho(\hat{\mathbf{W}}, \mathbf{K} \circ \hat{\mathbf{I}}) \mid$
      $l = (1 - \beta) \cdot l_q + \beta \cdot l_c$
      $\mathcal{A} \leftarrow \text{BackPropagate}(\mathcal{A}, l)$
      if $\text{mod}(i, 300) = 0 \wedge P < \text{PSNR}_{\text{min}}$ then
          $\beta \leftarrow \beta/4$
      end if
      if $P > \text{PSNR}_{\text{min}} \wedge l_c < 10^{-4}$ then
          return $\hat{\mathbf{I}}$
      end if
  end for
  return $\hat{\mathbf{I}}$
```

V. Results

In this section we report a set of results to test and validate the proposed pipeline.

Validation of Denoising Operator First, we need to asses whether DnCNN ($\mathcal{N}_{\text{dn}}$) can be used as a reasonable approximation for the widespread Wavelet ($\mathcal{N}_{\text{wl}}$) noise extractor tailored to PRNU matching and camera device identification. Figure 4 shows the distribution of normalized cross-correlation values (ρ) when $\mathcal{N}_{\text{dn}}$ and $\mathcal{N}_{\text{wl}}$ are used as noise extractors from pristine images. In both cases the reference PRNU ($\mathbf{K}$) is computed with the Wavelet filter. We can notice that for both noise extractors the discriminability between matching (M) and non-matching (NM) image-camera pairs is preserved, with a slight superimposition of the two distribution for DnCNN. Figure 5 shows the difference in terms of Receiver-Operating-Characteristic between $\mathcal{N}_{\text{dn}}$ and $\mathcal{N}_{\text{wl}}$ on pristine images. The

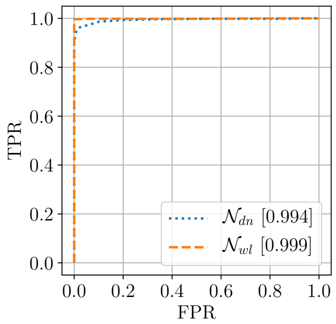


Fig. 5: Comparison between $\mathcal{N}_{\text{wl}}$ and $\mathcal{N}_{\text{dn}}$ as noise residual extractors in terms of Receiver-Operating-Characteristic. The Area Under Curve reported between squared brackets shows almost equivalent performance in terms of detection.

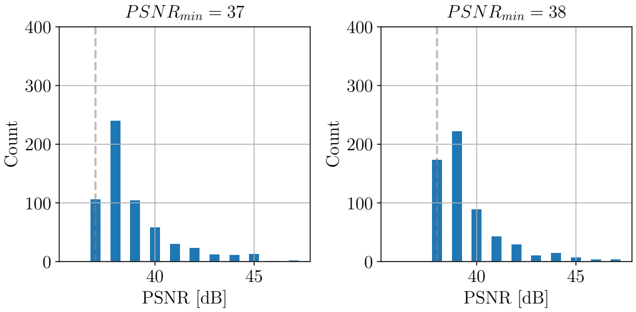
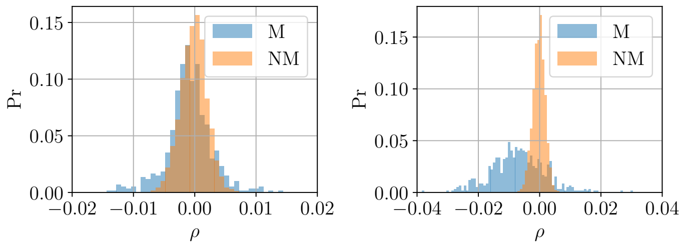


Fig. 6: Real PSNR distribution when varying PSNR_{min} in $\{37, 38\}$. The real PSNR values are always greater or equal the the minimum value (vertical dashed gray line). The same behavior is obtained for different PSNR_{min} values.



(a) $\mathcal{N}_{\text{dn}}$ (b) $\mathcal{N}_{\text{wl}}$

Fig. 7: Distribution of normalized cross-correlation values on anonymized images using DnCNN ($\mathcal{N}_{\text{dn}}$) and Wavelet ($\mathcal{N}_{\text{wl}}$) noise extractors, for matching image-PRNU pairs (M) and non-matching pairs (NM).

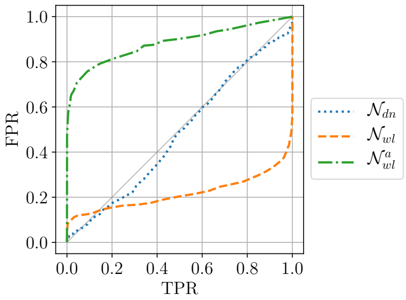


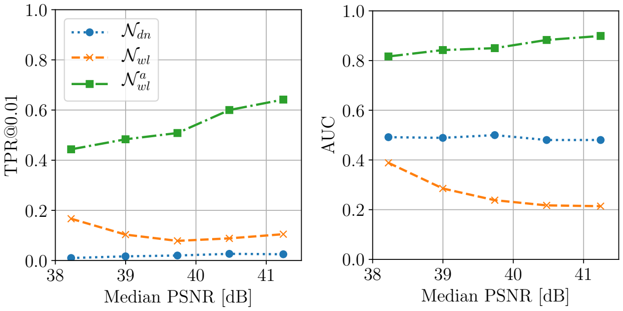
Fig. 8: Receiver-Operating-Characteristic for $\text{PSNR}_{\text{min}} = 40$.

values of AUC reported in the legend show how DnCNN detection performance are slightly lower than the ones of Wavelet, but still above 0.99. This test confirms that DnCNN is able to extract PRNU-based residual information from images, thus justifying its use within our anonymization pipeline.

Minimum PSNR Requirement As the proposed algorithm uses the PSNR_{min} as a driving criteria for the minimum accepted image quality, we are interested in checking whether this criteria is actually met in an experimental fashion. In facts, it might happen that the anonymization loop reaches the maximum number of iterations but the PSNR between $\mathbf{I}$ and $\hat{\mathbf{I}}$ is still smaller than PSNR_{min}. Figure 6 reports the histograms of PSNR values obtained for various values of PSNR_{min}. It is possible to notice that for every choice of PSNR_{min} the actual values of PSNR are always greater or equal to the minimum bound. This confirms that the proposed iterative method is able to reach convergence in terms of the imposed minimum PSNR requirement.

Image Anonymization When it comes to verify the effectiveness of the proposed pipeline in reducing PRNU-based device identification, we first compute the distribution of matching and non-matching normalized cross-correlation (ρ) values obtained from anonymized images with noise residuals extracted with DnCNN ($\mathcal{N}_{\text{dn}}$) and Wavelet ($\mathcal{N}_{\text{wl}}$). Figure 7a shows how the distributions of matching and non-matching ρ values, obtained when noise residuals are extracted from $\hat{\mathbf{I}}$ through $\mathcal{N}_{\text{dn}}$, are superimposed. This makes practically impossible to bind an anonymized images to the device it comes from. This means that the proposed anonymization pipeline is working in the proper way, thus it has minimized the cross-correlation between the reference PRNU $\mathbf{K}$ and the noise residual extracted through $\mathcal{N}_{\text{dn}}$. As we wish to evaluate the effect of the proposed method when the Wavelet-based noise extractor is used on $\hat{\mathbf{I}}$, Figure 7b shows the distribution of matching and non-matching ρ values obtained when noise residuals are extracted with $\mathcal{N}_{\text{wl}}$. We can immediately spot two differences with respect to the $\mathcal{N}_{\text{dn}}$ extractor: i) the mean of the matching values is not anymore zero, but it is shifted toward negative values; ii) the variance of matching cross-correlations is way higher than the variance of non-matching cross-correlations. A forensic investigator acting in a blind way, without the knowledge of the proposed anonymization pipeline, might use the cross-correlation test definition at Eq. (1) to asses whether an image $\hat{\mathbf{I}}$ under investigation comes from a camera whose PRNU is $\mathbf{K}$. However, a smart investigator would also perform another test, evaluating the absolute value of the normalized cross-correlation, thus building a symmetric test $|\rho(\mathbf{W}, \mathbf{K} \circ \mathbf{I})| > \tau$. In the plots, we refer to the results obtained with the standard Wavelet detector with $\mathcal{N}_{\text{wl}}$, while the results obtained with the Wavelet symmetric detector are denoted as $\mathcal{N}^{\text{a}}_{\text{wl}}$.

Figure 8 shows the ROC on anonymized images detection



(a) (b)

Fig. 9: True-Positive Rate at a fixed False-Alarm probability $\alpha = 0.01$ (a) and Area Under Curve (b) when varying PSNR_{min}.

for $\text{PSNR}_{\text{min}} = 40$. If $\mathcal{N}_{\text{dn}}$ is used to extract the noise residual from $\hat{\mathbf{I}}$ we get almost perfect anonymization performance. This confirms that the anonymization loop, based on the minimization of the cross-correlation value between $\mathbf{K} \circ \hat{\mathbf{I}}$ and $\hat{\mathbf{W}}$ extracted through $\mathcal{N}_{\text{dn}}$, is effectively working as expected. When noise residuals are extracted from $\hat{\mathbf{I}}$ through the Wavelet-based function and the unidirectional test in Eq. (1) is used ($\mathcal{N}_{\text{wl}}$), the detection performance are severely affected. However, resorting to the symmetric detector ($\mathcal{N}^{\text{a}}_{\text{wl}}$) shows that in fact the detection performances are affected, but are not as bad as when the asymmetrical detector is used.

A final result is shown in Figure 9, where two standard metrics in anonymization are presented. Figure 9a and Figure 9b respectively report the True-Positive rate at a fixed False-Alarm rate of 0.01 and the Area Under Curve for several median PSNR values. Each point is obtained by setting PSNR_{min} to $\{37, 38, 39, 40, 41\}$. The almost zero TPR@0.01 value for $\mathcal{N}_{\text{dn}}$ and the almost constant 0.5 value for AUC are assessing that the anonymization cycle is working properly if the noise extraction function used in the anonymization loop is the same as the one used for analysis purposes. When a different noise extraction function is used and a forensics investigator is aware of the attack ($\mathcal{N}^{\text{a}}_{\text{wl}}$) the anonymization is not guaranteed anymore.

VI. Conclusions

In this paper we proposed a method to anonymize images by removing PRNU traces in a scenario in which the specific PRNU to be removed is assumed to be known. The proposed solution makes use of a CNN in an uncommon fashion. Indeed, the CNN is seen as a parametric operator. CNN training is used to estimate CNN parameters by minimizing a loss function on a single image. From a different perspective, the proposed method works by overfitting a specific CNN to each input image.

From the adversarial forensic point-of-view, results show an interesting aspect. If the denoising operators used for PRNU testing and within the anonymization network match (i.e., DnCNN is used), images are strongly anonymized. If the analyst makes use of a different denoising operator for PRNU testing (i.e., the Wavelet-based one), anonymization may or may not be effective depending on the use correlation test. In reality, denoising operator matching is not needed by an attacker, given that the analyst is not informed about the possibility of an attack. If analysts know about possible attacks, they can use the symmetric test to avoid being completely fooled.

References

[1] M. Kirchner and T. Gloe, "Forensic Camera Model Identification," in *Handbook of Digital Forensics of Multimedia Data and Devices*. 2015.

[2] S. Bayram, H. Sencar, N. Memon, and I. Avcibas, "Source camera identification based on CFA interpolation," in *IEEE International Conference on Image Processing (ICIP)*, 2005.

[3] X. Zhao and M. C. Stamm, "Computationally efficient demosaicing filter estimation for forensic camera model identification," *IEEE International Conference on Image Processing (ICIP)*, pp. 151–155, 2016.

[4] S.-H. Chen and C.-T. Hsu, "Source camera identification based on camera gain histogram," *IEEE International Conference on Image Processing (ICIP)*, pp. 429–432, 2007.

[5] C. Chen and M. C. Stamm, "Camera model identification framework using an ensemble of demosaicing features," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2015.

[6] F. Marra, G. Poggi, C. Sansone, and L. Verdoliva, "Evaluation of residual-based local features for camera model identification," in *New Trends in Image Analysis and Processing – ICIAP 2015 Workshops*. Springer International Publishing, 2015.

[7] A. Tuama, F. Comby, and M. Chaumont, "Camera model identification based on machine learning approach with high order statistics features," *IEEE European Signal Processing Conference (EUSIPCO)*, pp. 1183–1187, 2016.

[8] L. Bondi, L. Baroffio, D. Güera, P. Bestagini, E. J. Delp, and S. Tubaro, "First steps toward camera model identification with convolutional neural networks," *IEEE Signal Processing Letters (SPL)*, vol. 24, no. 3, pp. 259–263, 2017.

[9] J. Lukáš, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 1, pp. 205–214, 2006.

[10] M. Goljan and J. Fridrich, "Camera identification from cropped and scaled images," in *SPIE Electronic Imaging (EI)*, 2008.

[11] A. Karaküçük and A. E. Dirik, "Adaptive photo-response non-uniformity noise removal against image source attribution," *Journal of Digital Investigation*, vol. 12, pp. 66–76, 2015.

[12] H. Zeng, J. Chen, X. Kang, and W. Zeng, "Removing camera fingerprint to disguise photograph source," in *IEEE International Conference on Image Processing (ICIP)*, 2015.

[13] A. E. Dirik, H. T. Sencar, and N. Memon, "Analysis of seam-carving-based anonymization of images against PRNU noise pattern-based source attribution," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 9, pp. 2277–2290, 2014.

[14] J. Entrieri and M. Kirchner, "Patch-based desynchronization of digital camera sensor fingerprints," in *IS&T Electronic Imaging (EI)*, 2016.

[15] S. Mandelli, L Bondi, S. Lameri, V. Lipari, P. Bestagini, and S. Tubaro, "Inpainting-based camera anonymization," in *IEEE International Conference on Image Processing (ICIP)*, 2017.

[16] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing (TIP)*, vol. 26, no. 7, pp. 3142–3155, 2017.

[17] T. Gloe and R. Böhme, "The dresden image database for benchmarking digital image forensics," *Journal of Digital Forensic Practice*, vol. 3, pp. 150–159, 2010.

[18] M. K. Mihcak, I. Kozintsev, K. Ramchandran, and P. Moulin, "Low-complexity image denoising based on statistical modeling of wavelet coefficients," *IEEE Signal Processing Letters (SPL)*, vol. 6, pp. 300–303, 1999.

[19] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," *NIPS, Autodiff Workshop*, 2017.