Pattern Recognition Letters 179 (2024) 73–79

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

# Adversarial mimicry attacks against image splicing forensics: An approach for jointly hiding manipulations and creating false detections

Giulia Boato [a,*], Francesco G.B. De Natale [a,e], Gianluca De Stefano [a,d], Cecilia Pasquini [a,c], Fabio Roli [b]

[a] *Department of Information Engineering and Computer Science, University of Trento, Italy*
[b] *Department of Informatics, Bioengineering, Robotics, and Systems Engineering, University of Genova, Italy*
[c] *Center for Cybersecurity, Fondazione Bruno Kessler, Italy*
[d] *CISPA Helmholtz Center for Information Security, Germany*
[e] *CNIT (Consorzio Nazionale Interuniversitario per le Telecomunicazioni), Italy*

ARTICLE INFO

Editor: Maria De Marsico

*Keywords:*
Adversarial multimedia forensics
Gray-box attack
Image manipulation hiding
False forgery creation
Image splicing detection

ABSTRACT

The term “mimicry attack” has been coined in computer security and used in adversarial machine learning: an attacker observes what a machine-learning system has learned and adjusts the malicious input so that it mimics a benign input. In this paper we extend this concept to image forensics, to allow an attacker modifying a manipulated image so that it appears pristine when analyzed by a target forensic detector. Recent work has shown that such attacks can be executed against detectors based on deep networks for hiding image tampering. We do more than that: our mimicry attack can force the target detector to identify arbitrary fictitious manipulations, while hiding the true ones. Accordingly, the user of the forensic detector is completely misled. From a methodological viewpoint, the proposed attack artificially alters the detector-specific intermediate representations according to the pixel distribution in the manipulated image, by applying a gradient-based optimization process. Experimental tests on different data sets and detectors demonstrate that our approach succeeds in jointly hiding manipulated areas and arbitrarily adding new ones, favorably comparing with the state-of-the-art in the first task.

## 1. Introduction

An ever-increasing share of the population has the means, tools and capabilities to produce and manipulate high-quality media contents, casting a long shadow on our ability to discern the truth from the false. This amount of potentially unreliable information can be easily shared and disseminated over the web, thus clearly posing a challenge from a social standpoint.

In the last decade, much effort in image forensics has been devoted to develop general purpose forgery detectors in digital images (e.g., Noiseprint [1], Exif-SC [2], Spliceradar [3]) possibly based on Deep Learning (DL), thanks to the increasing availability of large datasets of manipulated contents. While this has led to unparalleled performance, it also opened the path to adversarial attacks against image forensics tools, due to security limitations of learning-based systems [4]. Recently, the work in [5] demonstrated how to handcraft perturbations on manipulated images, so that they are not detected by some of the aforementioned DL-based methods.

In this vein, this work expands this idea by introducing a mimicry attack able to jointly fulfill a two-fold objective: avoiding the actual forgery to be exposed by forensic detectors *and* introducing arbitrary areas that are instead wrongly detected as forgeries, while retaining the quality and the semantics of the originally manipulated image. This is achieved through a gray-box gradient-based iterative process, executed against a given forensic detector.

The effect of such attack against an exemplary detector is visualized in Fig. 1: in the manipulated image (top-left) the person on the left has been added to the original picture, as represented in the binary ground truth tampering map. An arbitrary target tampering map pointing to another person within the picture is then fixed by the attacker (top-right): the iterative mimicry attack slightly modifies the manipulated image, so as to force the forensic detector to miss the actual forgery and reveal the false one.

The practical relevance of the proposed attack is twofold: when applied to individual images, it allows an adversary introducing specific

[*] Corresponding author.
*E-mail addresses:* giulia.boato@unitn.it (G. Boato), francesco.denatale@unitn.it (F.G.B. De Natale), gianluca.de-stefano@cispa.de (G. De Stefano), c.pasquini@fbk.eu (C. Pasquini), fabio.roli@unige.it (F. Roli).

https://doi.org/10.1016/j.patrec.2024.01.023
Received 4 July 2023; Received in revised form 13 November 2023; Accepted 23 January 2024
Available online 24 January 2024

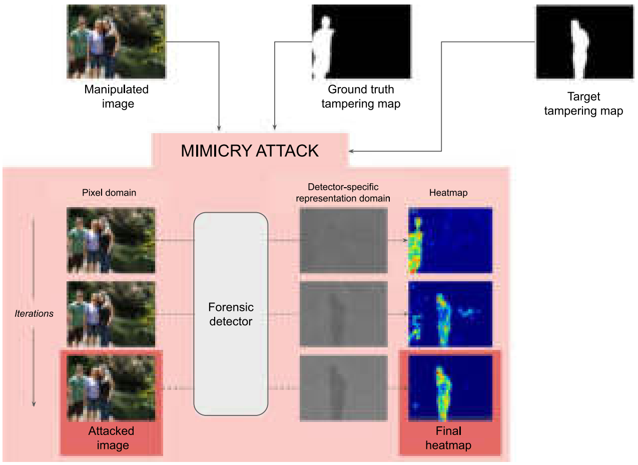


**Fig. 1.** Visualization of the proposed attack's effect.

semantic biases in the forensic detection; when applied at large-scale within the inputs of a forensic tool, it would bring to massive false-alarms, thus casting doubts on the reliability of the detection system with respect to its prescribed purpose. This would cause a similar effect to what is known as *DoS* (denial of service) attack in computer security.

Experimental tests demonstrated the efficacy of the proposed approach on several benchmark datasets and forensic detectors. In particular, we studied for the first time the effectiveness of stacked attacks (i.e., multiple attacks applied subsequently), as a mean to increase attack transferability between detectors and delving into the relationship between their different feature domains.

The paper is structured as follows: we review the literature on adversarial forensics in Section 2 and detail the proposed attack in Section 3. Section 4 describes the setup of our experimental evaluation, while results are reported and discussed in Section 5. Conclusions are drawn in Section 6.

## 2. Background concepts and previous work

In this section we summarize the state of the art on image splicing forensics, and on attacks to forensics detectors, which are the basis for our proposed adversarial attack.

*Image splicing detectors:* different splicing detection systems have been proposed, relying on a number of feature representations, calibration strategies and learning machinery. Although they inherently differ in terms of strategies employed, we can provide a general unified view of their overall pipeline and the different steps performed to obtain a tampering map. Fig. 2 reports a representation of how these detectors work in their operational phase, thus after the design, training, and system deployment for image forensic applications.

The input is the image under investigation, which is manipulated according to a binary *ground truth tampering map* where each pixel is labeled as pristine (0) or forged (1), visualized at the bottom-right corner in Fig. 2. The latter is unknown to the analyst, which aims at estimating such map as output of the detector. This is performed by extracting relevant local features through a function $F : \mathbb{R}^N \rightarrow \mathbb{R}^M$, which is applied on either patches or single pixels with dimensionality $N$ and provides as output an $M$-dimensional feature vector. Typically, $F$ is determined through a previous differentiable learning process. A post-processing phase of the extracted features provides an intermediate representation and may include non-differentiable operators. This serves as input to clustering procedures, where features are analyzed statistically with the goal of identifying and locating anomalies in their distribution, supposedly corresponding to local manipulations. The output of this phase is generally a pixel-wise or patch-wise heatmap of scores, expressing the probability that the pixel or patch has been manipulated. Finally, the heatmap can be binarized to obtain a tampering map providing a hard decision on image authenticity. Performance metrics are typically evaluated by measuring the similarity between such map and the groundtruth.

*Attacks to image forensics:* Adversarial attacks aiming at misleading the predictions of image manipulation detectors have been investigated for more than a decade, under the name of *anti-* or *counter-forensics* [6]. Typically, their goal is to conceal the traces that single or classes of manipulations leave in the image signal. It has been demonstrated that effective strategies can be devised to hide traces of resampling operations [7], multiple compression [8,9], contrast enhancement [10], median filtering [11]. Also, techniques to synthetize the statistics of authentic data have been proposed, such as the insertion of CFA patterns [6] and PRNU noise [12]. Anti-forensics techniques evolved over years together with the design of new and more effective forensic detectors, as surveyed in [13]. In fact, attacks have been proposed against both model-based detectors and machine learning-based detectors [14]. Recently, deep learning detection pipelines are dominant and adversarial strategies against them have been explored for both manipulation detection [15,16] and source identification tasks [17,18], by adapting approaches proposed in the field of adversarial machine learning [4,19].

When focusing on the splicing localization problem, the most relevant approach is presented in [5], where the authors adapt the attack procedure proposed on their previous work [20] to deal with forensic splicing detectors based on deep learning. This is the closest contribution to our work, as the iterative process applied to process individual patches has several commonalities. However, our framework presents a higher flexibility as target representations are determined on a spatial basis, thus allowing to arbitrarily create fictitious manipulated areas.

## 3. Threat model and proposed attack strategy

We here describe the adversarial framework and the considered threat model, and we formalize the proposed attack.

### 3.1. Threat model

Different attacks can be conceived and performed to compromise the effectiveness of detection systems structured as shown in Fig. 2. By following the convention introduced in [4], we consider the following threat model:

*Attacker's knowledge:* We consider the case where the attacker knows only the feature extractors $F(\cdot)$, while being agnostic to the following feature post-processing, clustering, and binarization steps. The attack is not crafted by exploiting the knowledge of the end-to-end analysis procedure but only with respect to the feature extraction part, with the goal of compromising the subsequent steps as well: the attack is *gray-box* or with *limited knowledge* [21].

*Attacker's goal:* The attacker's goal is to violate the integrity of the splicing detection system: for a given input manipulated image, the attacker aims at modifying it so that the splicing detector outputs an arbitrarily-defined *target tampering map*. In particular, differently from previous works, we address the case where the attacker attempts to jointly hide the truly manipulated area *and* introduce a new fictitious manipulation in the target tampering map, in order to further mislead the analyst. Thus, we introduce the concepts of *ground truth* (GT) forgery and *decoy* (D) forgery: the former is the set of pixel locations labeled as forged in the ground truth tampering map (corresponding to the truly manipulated area), while the latter is the set of pixel locations that the attacker wants to be recognized as manipulated by the detector. The target tampering map is then obtained by setting as forged (1)

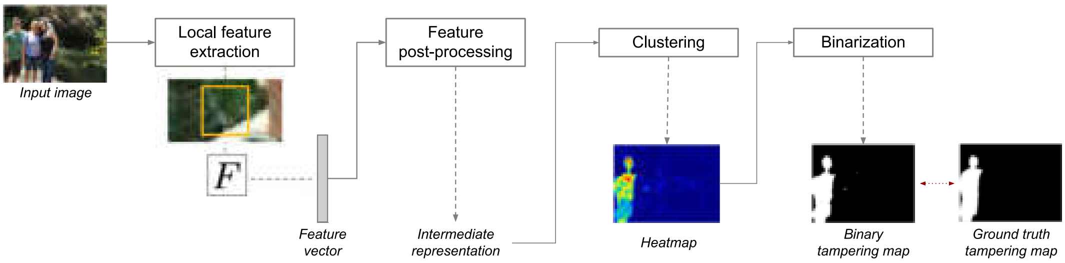


**Fig. 2.** Common pipeline of image splicing detectors.

**Table 1**
Components of the two splicing detectors considered.

| Method | Local feature extraction | Feature post processing | Clustering | Definition of $\mathbf{t}_p$, $\mathbf{t}_f$ |
|---|---|---|---|---|
| Noiseprint [1] | $F(\cdot)$: noise residual extracted by a neural network (backbone architecture proposed in [22]). *Training strategy:* Siamese architecture where patches from the same camera and pixel location are forced to have similar feature representations. | Noise residuals gets quantized pixel-wise on a truncated interval, and local co-occurrences on four pixels are computed, as in [23]. Intermediate local feature vectors with limited dimensionality are obtained. | Unsupervised clustering is performed on the intermediate feature vectors through the expectation–maximization (EM) with Gaussian Mixture Model. This provides a heatmap highlighting the forged and pristine region. | $\mathbf{t}_p$: $N$-dimensional vector containing zeros.<br>$\mathbf{t}_f$: average of the feature representations $F(\mathbf{x}_i)$ with $i \in \mathcal{I}_f$ scaled by a factor 5 |
| EXIF-SC [2] | $F(\cdot)$: a ResNet50 architecture. *Training strategy:* Siamese architectures where patches coming from images with the same EXIF metadata are forced to have similar feature representations. | For a number of patches, a consistency measure with all other patches is computed. This leads to intermediate response maps, where pristine regions ideally have low consistency with forged regions. | The Mean Shift algorithm is applied in order to find the most consistent mode among all patch response maps and produce a single heatmap, where pristine and manipulated regions are highlighted. | $\mathbf{t}_p$: average of the feature representations $F(\mathbf{x}_i)$ with $i \in \mathcal{I}_p$ scaled by a factor of 10<br>$\mathbf{t}_f = -\mathbf{t}_p$ |

the pixels corresponding to the region D, and as pristine (0) all the remaining pixels.[1]

*Attacker's capability:* The attack is exploratory, as the adversary has no access to the training phase, but attacks pre-trained models in their operational phase. In doing so, the attacker has access to the manipulated image and can modify its pixel values before it is given in input to the detector.

### 3.2. Proposed attack strategy

From the given manipulated image $\mathbf{X}$, a set of non-overlapping vectorized patches $\{\mathbf{x}_i\}_{i\in\mathcal{I}}$ of size $N$ is extracted, where the set $\mathcal{I}$ indexes also the corresponding binary patches in the tampering maps. The following subsets of $\mathcal{I}$ can then be defined with respect to the two different tampering maps:

- $\mathcal{I}_p, \mathcal{I}_f$: $\mathcal{I}_p$ corresponds to patches in the ground truth tampering map where all pixels are pristine, $\mathcal{I}_f$ where all of them are forged;
- $\mathcal{I}_p^T, \mathcal{I}_f^T$ : $\mathcal{I}_p^T$ corresponds to patches in the target tampering map where the majority of pixels are pristine, $\mathcal{I}_f^T$ where the majority of them are forged; in this case, $\mathcal{I}_p^T \dot\cup \mathcal{I}_f^T = \mathcal{I}$.

The attack procedure processes each $\mathbf{x}_i$ with the goal of moving its feature representation $F(\mathbf{x}_i)$ as close as possible to a *target representation* $t(\mathbf{x}_i)$ in terms of Euclidean distance. The function $t : \mathbb{R}^N \to \mathbb{R}^M$ that associates to a generic patch $\mathbf{x}_i$ its target representation is defined as follows:

$$t(\mathbf{x}_i) = \begin{cases} \mathbf{t}_p & \text{if } i \in \mathcal{I}_p^T \\ \mathbf{t}_f & \text{if } i \in \mathcal{I}_f^T \end{cases} \tag{1}$$

The vectors $\mathbf{t}_p$ and $\mathbf{t}_f$ are derived from the manipulated image by properly combining the image patches corresponding to the indices in $\mathcal{I}_p$ and $\mathcal{I}_f$, respectively. This operation may be performed differently according to the specific detector attacked.

Once the target representations are fixed, we use the iterative procedure described in [5] to modify the image patches. At the $k$th iteration, an image $\mathbf{X}^{(k)}$ is obtained from $\mathbf{X}^{(k-1)}$ through the addition of a perturbation. For each $i \in \mathcal{I}$ we use the Euclidean distance between the feature representation of a patch $F(\mathbf{x}_i^{(k)})$ and its target representation $t(\mathbf{x}_i^{(k)})$ as loss function and compute its gradient with respect to the patch components:

$$\mathbf{g}_i^{(k)} = \nabla_{\mathbf{x}}\left[\frac{1}{2}||t(\mathbf{x}_i^{(k)}) - F(\mathbf{x}_i^{(k)})||_2^2\right], \quad i \in \mathcal{I} \tag{2}$$

Then, those local gradients are combined to obtain

$$\mathbf{G}^{(k)} = \bigoplus_{i\in\mathcal{I}} \mathbf{g}_i^{(k)}, \tag{3}$$

where $\bigoplus$ indicates a spatial recombination operation in which gradients $\mathbf{g}_i$ are placed at the coordinates of the corresponding patches $\mathbf{x}_i$, so that $\mathbf{G}^{(k)}$ has the same spatial size of $\mathbf{X}$.

The attacked image is obtained as:

$$\mathbf{X}^{(k)} = \mathbf{X}^{(k-1)} - \alpha\frac{\mathbf{G}^{(k)}}{||\mathbf{G}^{(k)}||_\infty} \tag{4}$$

$\alpha$ is a non-negative scaling factor that controls the infinity norm of the perturbation. The procedure is initialized with $\mathbf{X}^{(0)} = \mathbf{X}$ and stops when the maximum number of iterations $K$ is reached.

The iterative process used for moving individual patches towards their assigned target representation in the feature space (formulas (2)–(4)) is then the same as in [5], which was in turn adapted from [20]. However, the attack devised in [5] defines a single target representation derived from pristine patches, which is enforced on all patches. Here, due to the two-fold objective of concealing the forgery and introducing a decoy one, different patches are pushed towards one of the two target representations $\mathbf{t}_p$ and $\mathbf{t}_f$, depending on their spatial location. Moreover, as detailed in the next section, $\mathbf{t}_p$ and $\mathbf{t}_f$ are crafted differently for individual detectors, so as to increase the attack effectiveness.

[1] D and GT can be either be fully disjoint, partially overlapping or fully contained in each other.

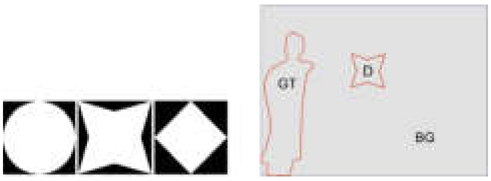


**Fig. 3.** Objects used as decoy forgeries and partition of the image pixels.

## 4. Experimental setup

### 4.1. Data and forensic detectors

We tested our attack against two representative image splicing detectors, namely Noiseprint [1] and EXIF-SC [2]. For completeness, we report in Table 1 a brief explanation of the different components of the two methods, following the pipeline reported in Fig. 2. We also report the strategy we adopted to define the target representations $\mathbf{t}_p$ and $\mathbf{t}_f$ used for applying the attack described in Section 3.2. We set $\alpha = 5$ and $K = 50$ as in [5].

We consider two datasets: Columbia [24], consisting in 183 authentic images and 180 containing splices with sizes varying from 757 × 568 to 1152 × 768 pixels, and DSO-1 [25], containing 200 pictures of size 2048 × 1536, half authentic and half with a splice. All images come with a GT tampering map indicating the truly manipulated area.

### 4.2. Attack protocol and performance indicators

For each image, we have run the attack as described in Section 3.2, according to the definition of $\mathbf{t}_p$ and $\mathbf{t}_f$ reported in Table 1. The target tampering maps are created by removing the GT forgery and placing one of the three objects reported in Fig. 3 (left panel) as decoy forgery in such a way that they do not overlap with the GT forgery. The objects have been chosen so as to simulate forgeries with small size and varied shapes. In fact, in all cases the object is $0.2 \cdot h$ high, where $h$ is the height of the attacked image, and occupies a fraction $d$ of the image pixel count, where $d \approx 0.02$. This results in the image pixels being partitioned into 3 areas: the GT forgery, the D forgery, and the background (BG), as shown in Fig. 3 (right panel).

In order to measure the effectiveness of the attack with respect to the capabilities of the splicing detectors, we employ different performance indicators on the output a heatmap, that we normalize into the interval $[0, 1]$. Previous works [1,5] measured the effectiveness of the attack through *threshold-based* metrics, where the heatmap is binarized according to a certain threshold and individual pixels are considered as either forged or pristine. However, such process strongly relies on the threshold value, which is chosen by the forensic analyst and unknown a priori to the attacker, thus entailing a range of corresponding metrics values depending on the heuristics adopted in choosing the threshold. In order to better characterize the impact of the attack on the detector outputs, we then propose to employ also *threshold-free* indicators, which are computed from the outcome heatmap regardless of the threshold choice and express statistical properties of its values.

All indicators are summarized in Table 2, and are computed per image before being averaged over the datasets. Among the threshold-free indicators, we compute median values of the heatmap in the three different areas (med$_{\mathrm{BG}}$, med$_{\mathrm{D}}$, med$_{\mathrm{GT}}$), which are then used to compute "visibility" indicators expressing the distance between the two forgery areas D and GT with respect to the background area (V$_{\mathrm{D}}$ = med$_{\mathrm{D}}$-med$_{\mathrm{BG}}$, V$_{\mathrm{GT}}$ = med$_{\mathrm{GT}}$-med$_{\mathrm{BG}}$). Also, we compute pixel-wise two separate Receiver Operating Curves (ROC), one with respect to the target tampering map and the other to the GT tampering map. We then report the corresponding Areas Under the Curves (AUC$_{\mathrm{D}}$ and AUC$_{\mathrm{GT}}$, respectively) as indicators of pixel separability between the two different forgery areas and the rest of the image.

**Table 2**
Performance indicators and threshold choices.

| Threshold-free indicators | |
|---|---|
| med$_{\mathrm{BG}}$ | Median value of the heatmap in the BG area |
| med$_{\mathrm{D}}$ | Median value of the heatmap in the D area |
| med$_{\mathrm{GT}}$ | Median value of the heatmap in the GT area |
| V$_{\mathrm{D}}$ | Difference between med$_{\mathrm{D}}$ and med$_{\mathrm{BG}}$ (visibility of D wrt BG) |
| V$_{\mathrm{GT}}$ | Difference between med$_{\mathrm{GT}}$ and med$_{\mathrm{BG}}$ (visibility of GT wrt BG) |
| AUC$_{\mathrm{D}}$ | Area Under the Curve wrt the target map |
| AUC$_{\mathrm{GT}}$ | Area Under the Curve wrt the GT map |

| Threshold-based indicators | | Threshold choices | |
|---|---|---|---|
| F1$_{\mathrm{D}}$ | $F1$-score wrt target map | $\tau_{\mathrm{GT}}$ | Yielding the higher F1$_{\mathrm{GT}}$ |
| F1$_{\mathrm{GT}}$ | $F1$-score wrt GT map | $\tau_{\mathrm{D}}$ | Yielding the higher F1$_{\mathrm{D}}$ |
| MCC$_{\mathrm{D}}$ | $MCC$-score wrt D map | $\tau_{\mathrm{OTSU}}$ | Fixed with Otsu's method [26] |
| MCC$_{\mathrm{GT}}$ | $MCC$-score wrt GT map | $\tau_{\mathrm{MP}}$ | Midpoint of heatmap value range |
| dr$_{\mathrm{BG}}$ | Detection rate in the BG area | $\tau_d$ | (1-$d$)-quantile of heatmap values |
| dr$_{\mathrm{D}}$ | Detection rate in the D area | $\tau_{0.2}$ | 0.8-quantile of heatmap values |
| dr$_{\mathrm{GT}}$ | Detection rate in the GT area | | |

Regarding threshold-based indicators, we report the widely used F1-score again computed with respect to the two different maps, thus getting F1$_{\mathrm{D}}$ and F1$_{\mathrm{GT}}$, and the detection rate (intended as the ratio of pixels detected as forged) within the three different areas for different choices of the threshold, specified in Table 2. Those correspond to different strategies of the forensic analyst in interpreting the output heatmap: $\tau_{\mathrm{GT}}$ corresponds to the best result (in terms of F1) achievable by the forensic analyst in detecting the truly manipulated area. Conversely, $\tau_{\mathrm{D}}$ yields the best achievable result for the attacker, who wants a high fidelity to the target tampering map. However, differently from the attacker who knows both tampering maps, the forensic analyst has no (or perhaps partial) a priori knowledge of the GT map and will select a threshold according to other criteria. Thus, in order to better characterize the attack effectiveness, we report the results obtained by selecting the threshold with other strategies, such as the widely known Otsu's method [26] ($\tau_{\mathrm{OTSU}}$) and the value corresponding to the midpoint of the values range ($\tau_{\mathrm{MP}}$, in our case equal to 0.5). We also consider thresholds related to the distribution of heatmap pixels values, where a percentage of them yielding the higher values is detected as forged. In particular, we use the $(1 - d)$-quantile ($\tau_d$) and 0.8-quantile ($\tau_{0.8}$): in the first case, a number of pixels equal to ones in the $D$ forgery are selected, while in the second case 1/5 of the pixels are selected.

## 5. Results

### 5.1. Attack evaluation against individual detectors

We evaluate the effectiveness of the attack on the considered datasets and with respect to the different detectors. The results are reported in Fig. 4. For the different combinations of dataset and detector, we report the threshold-less and the threshold-based indicators. In order to assess the impact of the attack, we also report the threshold-less indicators of the manipulated images prior to the attack (*No attack*). An example of manipulated image before and after the attack is also reported.

First, we observe that the AUC$_{\mathrm{GT}}$ values before the attack are quite high in all cases, thus showing that the detectors are generally effective on images that are manipulated but not attacked. However, they decrease dramatically after the application of the attack, which hinders the detection of the GT forgery. This is observable also in the values of med$_{\mathrm{D}}$ and med$_{\mathrm{GT}}$ after the attack: while the latter decreases and gets close to the value of med$_{\mathrm{BG}}$, the former increases so that the D forgery is prominent in the heatmap. When analyzing threshold-based indicators, we notice that for the threshold $\tau_{\mathrm{GT}}$ (i.e., the best case for the analyst in terms of F1), we have that dr$_{\mathrm{GT}} \geq 0.85$, thus the GT forgery is largely detected. However, at the same time dr$_{\mathrm{BG}} \geq 0.50$, thus showing that more than half of the background pixels are also wrongly detected

**Performance of EXIF-SC on DSO-1 dataset**

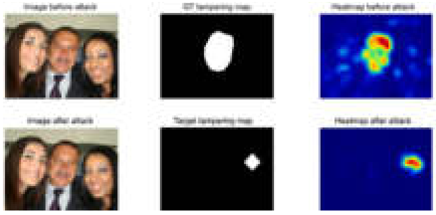
Threshold-less indicators

| | $\text{med}_{\text{BG}}$ | $\text{med}_{\text{D}}$ | $\text{med}_{\text{GT}}$ | $\text{V}_{\text{D}}$ | $\text{V}_{\text{GT}}$ | $\text{AUC}_{\text{D}}$ | $\text{AUC}_{\text{GT}}$ |
|---|---|---|---|---|---|---|---|
| *No attack* | 0,13 | 0,15 | 0,47 | 0,02 | 0,34 | 0,48 | 0,82 |
| *After attack* | 0,04 | 0,46 | 0,06 | 0,39 | 0,02 | 0,93 | 0,66 |

Threshold-based indicators

| | $\text{F1}_{\text{D}}$ | $\text{F1}_{\text{GT}}$ | $\text{MCC}_{\text{D}}$ | $\text{MCC}_{\text{GT}}$ | $\text{dr}_{\text{BG}}$ | $\text{dr}_{\text{D}}$ | $\text{dr}_{\text{GT}}$ |
|---|---|---|---|---|---|---|---|
| $\tau_{\text{GT}}$ | 0,08 | 0,29 | 0,12 | 0,12 | 0,60 | 0,96 | 0,85 |
| $\tau_{\text{D}}$ | 0,49 | 0,05 | 0,50 | -0,03 | 0,06 | 0,72 | 0,07 |
| $\tau_{\text{OTSU}}$ | 0,38 | 0,07 | 0,40 | -0,02 | 0,05 | 0,69 | 0,06 |
| $\tau_{\text{MP}}$ | 0,36 | 0,02 | 0,36 | -0,03 | 0,01 | 0,42 | 0,01 |
| $\tau_d$ | 0,41 | 0,01 | 0,40, | -0,03 | 0,01 | 0,41 | 0,01 |
| $\tau_{0.2}$ | 0,16 | 0,24 | 0,25 | 0,13 | 0,14 | 0,89 | 0,36 |

Single-image example:



**Performance of EXIF-SC on Columbia dataset**

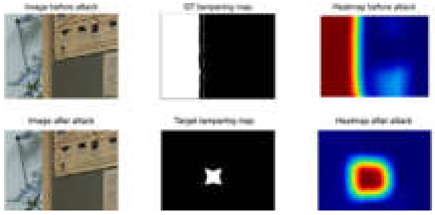
Threshold-less indicators

| | $\text{med}_{\text{BG}}$ | $\text{med}_{\text{D}}$ | $\text{med}_{\text{GT}}$ | $\text{V}_{\text{D}}$ | $\text{V}_{\text{GT}}$ | $\text{AUC}_{\text{D}}$ | $\text{AUC}_{\text{GT}}$ |
|---|---|---|---|---|---|---|---|
| *No attack* | 0,08 | 0,10 | 0,74 | 0,02 | 0,66 | 0,34 | 0,94 |
| *After attack* | 0,02 | 0,82 | 0,05 | 0,80 | 0,03 | 0,98 | 0,64 |

Threshold-based indicators

| | $\text{F1}_{\text{D}}$ | $\text{F1}_{\text{GT}}$ | $\text{MCC}_{\text{D}}$ | $\text{MCC}_{\text{GT}}$ | $\text{dr}_{\text{BG}}$ | $\text{dr}_{\text{D}}$ | $\text{dr}_{\text{GT}}$ |
|---|---|---|---|---|---|---|---|
| $\tau_{\text{GT}}$ | 0,06 | 0,48 | 0,09 | 0,16 | 0,52 | 1,00 | 0,9 |
| $\tau_{\text{D}}$ | 0,59 | 0,01 | 0,62 | -0,10 | 0,02 | 0,84 | 0,01 |
| $\tau_{\text{OTSU}}$ | 0,31 | 0,04 | 0,40 | -0,16 | 0,08 | 1,00 | 0,03 |
| $\tau_{\text{MP}}$ | 0,37 | 0,02 | 0,44 | -0,15 | 0,06 | 0,91 | 0,01 |
| $\tau_d$ | 0,48 | 0,00 | 0,47 | -0,07 | 0,09 | 0,48 | 0,00 |
| $\tau_{0.2}$ | 0,17 | 0,19 | 0,27 | -0,04 | 0,14 | 0,99 | 0,19 |

Single-image example:



**Performance of Noiseprint on DSO-1 dataset**

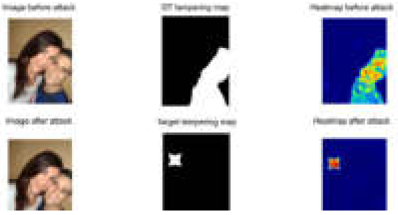
Threshold-less indicators

| | $\text{med}_{\text{BG}}$ | $\text{med}_{\text{D}}$ | $\text{med}_{\text{GT}}$ | $\text{V}_{\text{D}}$ | $\text{V}_{\text{GT}}$ | $\text{AUC}_{\text{D}}$ | $\text{AUC}_{\text{GT}}$ |
|---|---|---|---|---|---|---|---|
| *No attack* | 0,03 | 0,04 | 0,33 | 0,01 | 0,29 | 0,46 | 0,91 |
| *After attack* | 0,02 | 0,64 | 0,06 | 0,61 | 0,04 | 0,97 | 0,68 |

Threshold-based indicators

| | $\text{F1}_{\text{D}}$ | $\text{F1}_{\text{GT}}$ | $\text{MCC}_{\text{D}}$ | $\text{MCC}_{\text{GT}}$ | $\text{dr}_{\text{BG}}$ | $\text{dr}_{\text{D}}$ | $\text{dr}_{\text{GT}}$ |
|---|---|---|---|---|---|---|---|
| $\tau_{\text{GT}}$ | 0,10 | 0,37 | 0,14 | 0,21 | 0,60 | 0,96 | 0,92 |
| $\tau_{\text{D}}$ | 0,83 | 0,07 | 0,83 | 0,01 | 0,01 | 0,90 | 0,07 |
| $\tau_{\text{OTSU}}$ | 0,74 | 0,12 | 0,75 | 0,05 | 0,03 | 0,88 | 0,12 |
| $\tau_{\text{MP}}$ | 0,73 | 0,04 | 0,74 | 0,00 | 0,00 | 0,68 | 0,03 |
| $\tau_d$ | 0,81 | 0,03 | 0,80 | -0,01 | 0,00 | 0,81 | 0,02 |
| $\tau_{0.2}$ | 0,17 | 0,33 | 0,27 | 0,24 | 0,13 | 0,96 | 0,49 |

Single-image example:



**Performance of Noiseprint on Columbia dataset**

Threshold-less indicators

| | $\text{med}_{\text{BG}}$ | $\text{med}_{\text{D}}$ | $\text{med}_{\text{GT}}$ | $\text{V}_{\text{D}}$ | $\text{V}_{\text{GT}}$ | $\text{AUC}_{\text{D}}$ | $\text{AUC}_{\text{GT}}$ |
|---|---|---|---|---|---|---|---|
| *No attack* | 0,08 | 0,10 | 0,34 | 0,02 | 0,27 | 0,40 | 0,85 |
| *After attack* | 0,00 | 0,67 | 0,01 | 0,66 | 0,01 | 1,00 | 0,55 |

Threshold-based indicators

| | $\text{F1}_{\text{D}}$ | $\text{F1}_{\text{GT}}$ | $\text{MCC}_{\text{D}}$ | $\text{MCC}_{\text{GT}}$ | $\text{dr}_{\text{BG}}$ | $\text{dr}_{\text{D}}$ | $\text{dr}_{\text{GT}}$ |
|---|---|---|---|---|---|---|---|
| $\tau_{\text{GT}}$ | 0,06 | 0,46 | 0,10 | 0,11 | 0,61 | 1,00 | 0,93 |
| $\tau_{\text{D}}$ | 0,88 | 0,00 | 0,87 | -0,09 | 0,00 | 0,90 | 0,00 |
| $\tau_{\text{OTSU}}$ | 0,83 | 0,00 | 0,83 | -0,09 | 0,01 | 0,89 | 0,00 |
| $\tau_{\text{MP}}$ | 0,78 | 0,00 | 0,79 | -0,07 | 0,00 | 0,70 | 0,00 |
| $\tau_d$ | 0,86 | 0,00 | 0,86 | -0,08 | 0,00 | 0,86 | 0,00 |
| $\tau_{0.2}$ | 0,17 | 0,31 | 0,27 | 0,10 | 0,12 | 1,00 | 0,31 |

Single-image example:

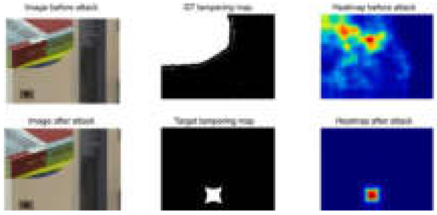


**Fig. 4.** Performance metrics of the attack for different datasets and detectors. For each combination, threshold-less and threshold-based indicators are reported. For a selected image the image before and after attack is reported, together with the GT and target tampering maps and the resulting heatmaps.

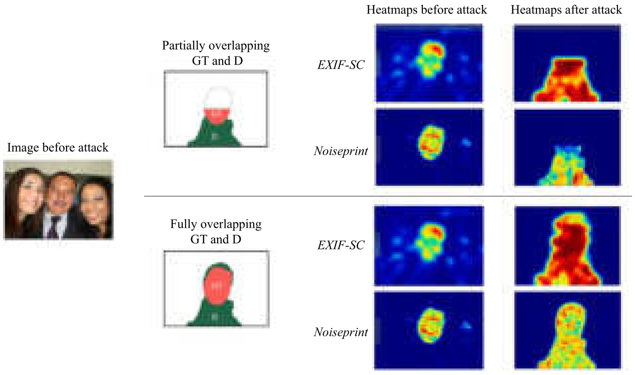


**Fig. 5.** Samples of attacked images with overlapping GT and D areas.

as forged. In other words, the result is affected by an extremely high false alarm rate. For all the other threshold choices, the GT forgery is essentially missed, while a large portion of the D forgery is detected. For $\tau_{0.2}$, the GT forgery is partially detected but, again, the false alarm rate on the background increases dramatically, thus obtaining a poorly informative estimated tampering map. Also, the D forgery is anyway detected with higher precision.

For completeness, we also report in Fig. 5 qualitative examples of attacked images where the GT and the D areas are partially or fully overlapping. In these cases, part or all the pixels in D are already forged and thus possibly closer to the target representation already. It can be observed that after the attack the target tampering map is accurately highlighted by both the detectors.

### 5.2. Attack evaluation in cross-detector scenarios

We now study experimental scenarios where the analyst uses a detector different from the attacked one. This corresponds to a transferability analysis of the attack between the two detectors. Moreover, we introduce the analysis of stacked attacks: this is the case of images sequentially attacked against two detectors, and tested by either of them. Stacked attacks represent a possible strategy for an attacker who attempts to deceive multiple known detectors.

We report the results in Table 3 and, for the sake of clarity, we first split them according to the detector used for the forensic analysis.

**Table 3**
Results of the attack in cross-detector settings. Threshold-less indicators are reported column-wise for both detectors when applied on images which underwent different attacks, reported row-wise in each subtable.

| Type of attack | Performance of EXIF-SC on DSO-1 dataset | | | | | | | Performance of EXIF-SC on Columbia dataset | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\mathrm{med}_{BG}$ | $\mathrm{med}_{D}$ | $\mathrm{med}_{GT}$ | $V_D$ | $V_{GT}$ | $\mathrm{AUC}_D$ | $\mathrm{AUC}_{GT}$ | $\mathrm{med}_{BG}$ | $\mathrm{med}_{D}$ | $\mathrm{med}_{GT}$ | $V_D$ | $V_{GT}$ | $\mathrm{AUC}_D$ | $\mathrm{AUC}_{GT}$ |
| *EXIF-SC (aligned)* | 0,04 | 0,46 | 0,06 | 0,39 | 0,02 | 0,93 | 0,66 | 0,02 | 0,82 | 0,05 | 0,80 | 0,03 | 0,98 | 0,64 |
| *Noiseprint + EXIF-SC (stacked)* | 0,04 | 0,44 | 0,06 | 0,41 | 0,02 | 0,93 | 0,67 | 0,02 | 0,80 | 0,05 | 0,78 | 0,02 | 0,98 | 0,61 |
| *EXIF-SC + Noiseprint (stacked)* | 0,04 | 0,44 | 0,07 | 0,40 | 0,03 | 0,93 | 0,69 | 0,02 | 0,81 | 0,05 | 0,79 | 0,02 | 0,98 | 0,63 |
| *Noiseprint (misaligned)* | 0,13 | 0,20 | 0,48 | 0,07 | 0,36 | 0,52 | 0,82 | 0,09 | 0,14 | 0,71 | 0,05 | 0,62 | 0,44 | 0,93 |
| *No attack* | 0,13 | 0,15 | 0,47 | 0,15 | 0,34 | 0,48 | 0,82 | 0,08 | 0,10 | 0,74 | 0,02 | 0,66 | 0,34 | 0,94 |

| Type of attack | Performance of Noiseprint DSO-1 dataset | | | | | | | Performance of Noiseprint Columbia dataset | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\mathrm{med}_{BG}$ | $\mathrm{med}_{D}$ | $\mathrm{med}_{GT}$ | $V_D$ | $V_{GT}$ | $\mathrm{AUC}_D$ | $\mathrm{AUC}_{GT}$ | $\mathrm{med}_{BG}$ | $\mathrm{med}_{D}$ | $\mathrm{med}_{GT}$ | $V_D$ | $V_{GT}$ | $\mathrm{AUC}_D$ | $\mathrm{AUC}_{GT}$ |
| *Noiseprint (aligned)* | 0,02 | 0,64 | 0,06 | 0,61 | 0,04 | 0,97 | 0,68 | 0,00 | 0,67 | 0,01 | 0,66 | 0,01 | 1,00 | 0,55 |
| *EXIF-SC + Noiseprint (stacked)* | 0,02 | 0,70 | 0,01 | 0,68 | −0,01 | 0,99 | 0,52 | 0,00 | 0,70 | 0,00 | 0,70 | 0,00 | 1,00 | 0,61 |
| *Noiseprint + EXIF-SC (stacked)* | 0,03 | 0,40 | 0,09 | 0,37 | 0,06 | 0,92 | 0,77 | 0,03 | 0,56 | 0,07 | 0,53 | 0,04 | 0,97 | 0,63 |
| *EXIF-SC (misaligned)* | 0,07 | 0,07 | 0,24 | 0,00 | 0,16 | 0,47 | 0,82 | 0,12 | 0,15 | 0,26 | 0,03 | 0,14 | 0,50 | 0,70 |
| *No attack* | 0,32 | 0,04 | 0,33 | 0,01 | 0,29 | 0,46 | 0,91 | 0,08 | 0,10 | 0,34 | 0,02 | 0,27 | 0,40 | 0,85 |

Please note that the first and last line of each sub-table coincide with the results discussed in the previous section.

We can observe that the misaligned attack is in general poorly effective for both detectors. In fact, when comparing the indicators with respect to the *No attack* scenario, we notice that the impact of the attack is rather limited: $V_{GT}$ and $\mathrm{AUC}_{GT}$ are only sightly decreased, while $V_D$ and $\mathrm{AUC}_D$ are also essentially unaltered. In general, passing from the aligned to misaligned scenario leads to a rather reduced effectiveness of the attack, thus suggesting that the statistical properties captured by the two detectors differ significantly.

Differently, stacked attacks perform significantly better. When looking at the case of EXIF-SC, we can observe that the two possible stacked attacks yield very similar results, and both are quite close to the aligned case. The order of attacks has instead some impact on the performance of Noiseprint.

In order to further delve into the interaction between different attacks and detectors, we jointly report in Table 4 selected performance metrics obtained with EXIF-SC (horizontal axis) and Noiseprint (vertical axis) on differently attacked images, labeled as in the legend. In particular, the top row refers to the $\mathrm{AUC}_D$ and the bottom row to the $\mathrm{AUC}_{GT}$, expressing the distinguishability of the D and GT pixels, respectively, from the rest of the image. It can be seen that in terms of $\mathrm{AUC}_D$, both stacked attacks (regardless of the order) retain the effectiveness of individual aligned attacks and fool both detectors, in a consistent manner among the two datasets. Regarding $\mathrm{AUC}_{GT}$, more diverse effects are observed, especially with respect to the Noiseprint detector. In fact, attacking first against Noiseprint (dark blue marker) and then against EXIF-SC (light yellow marker) does fool EXIF-SC, but causes to restore the performance of the Noiseprint detector. A similar effect is observed also for the EXIF+Noiseprint attack on the Columbia dataset: the effectiveness on Noiseprint is lower with respect to the aligned case. However, further experiments have shown that by simply increasing scaling factor of the attack against Noiseprint (see Table 1, top-right cell) mitigates this effect.

### 5.3. Comparison to [5]

As mentioned in Section 2, the closest approach to our work in the state-of-the-art-literature is represented by the LOTS attack proposed in [5], on which we have built for defining our attack procedure. Therefore, we report for completeness a performance comparison of the two attack pipelines (in our own implementation) against both the Noiseprint and EXIF-SC detectors. Since LOTS does not aim at introducing false detections, the subject of this comparison is only the ability of hiding the GT forgery. Thus, only the metrics related to this task are considered ($\mathrm{med}_{GT}$, $V_{GT}$, $\mathrm{AUC}_{GT}$). The results are reported in Table 5. In general, we can see that the proposed attack pipeline better hides the GT when compared to the one in [5], as attested by consistently lower scores across both datasets and detectors. In addition, our attack also effectively introduces the decoy forgery at the prescribed location.

**Table 4**
Crossed analysis of $\mathrm{AUC}_D$ (top row) and $\mathrm{AUC}_{GT}$ (bottom row) values for different attacks over the two detectors. The horizontal (vertical) axis correspond to the AUC values obtained with EXIF-SC (Noiseprint).

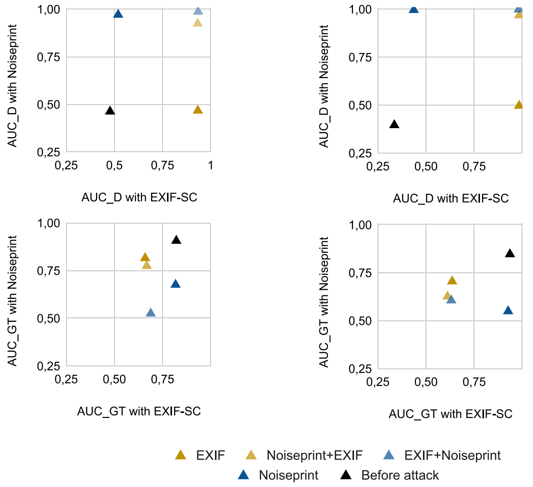


**Table 5**
Performance metrics of the LOTS attack for different datasets and detectors, and gap with respect to the proposed attack.

| | EXIF-SC on DSO-1 dataset | | | EXIF-SC on Columbia dataset | | |
|---|---|---|---|---|---|---|
| | $\mathrm{med}_{GT}$ | $V_{GT}$ | $\mathrm{AUC}_{GT}$ | $\mathrm{med}_{GT}$ | $V_{GT}$ | $\mathrm{AUC}_{GT}$ |
| *Attack in [5]* | 0,16 | 0,08 | 0,71 | 0,31 | 0,20 | 0,79 |
| *Gap wrt ours* | −0,10 | −0,06 | −0,05 | −0,26 | −0,17 | −0,15 |

| | Noiseprint on DSO-1 dataset | | | Noiseprint on Columbia dataset | | |
|---|---|---|---|---|---|---|
| | $\mathrm{med}_{GT}$ | $V_{GT}$ | $\mathrm{AUC}_{GT}$ | $\mathrm{med}_{GT}$ | $V_{GT}$ | $\mathrm{AUC}_{GT}$ |
| *Attack in [5]* | 0,27 | 0,18 | 0,83 | 0,24 | 0,10 | 0,69 |
| *Gap wrt ours* | −0,21 | −0,14 | −0,15 | −0,23 | −0,09 | −0,14 |

## 6. Conclusion

We have proposed a mimicry attack against splicing forgery detectors which hides traces of manipulation while jointly introducing fictitious ones. Experimental validation shows the effectiveness of the approach against different detectors and benchmarking datasets, while discussing transferability issues among feature domains. Future work include assessing a wider variety of detectors and further strengthening attack generalization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgments

This work was partially supported by the project SERICS, Italy (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU, by the project PREMIER funded by the Italian Ministry of Universities and Research (MUR), Italy under Grant 2017Z595XS, and by the Defense Advanced Research Projects Agency (DARPA), Italy under Agreement No. HR00112090136.

## References

[1] D. Cozzolino, L. Verdoliva, Noiseprint: A CNN-based camera model fingerprint, IEEE Trans. Inf. Forensics Secur. (2020).

[2] M. Huh, A. Liu, A. Owens, A.A. Efros, Fighting fake news: Image splice detection via learned self-consistency, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018.

[3] A. Ghosh, Z. Zhong, T.E. Boult, M. Singh, SpliceRadar: A learned method for blind image forensics, in: IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2019.

[4] B. Biggio, F. Roli, Wild patterns: Ten years after the rise of adversarial machine learning, Pattern Recognit. (2018).

[5] A. Rozsa, T.E. Boult, Z. Zhong, Adversarial attack on deep learning-based splice localization, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, 2020.

[6] R. Böhme, M. Kirchner, Counter-forensics: Attacking image forensics, in: H.T. Sencar, N. Memon (Eds.), Digital Image Forensics: There Is more To a Picture than Meets the Eye, Springer New York, New York, NY, 2013.

[7] M. Kirchner, R. Böhme, Hiding traces of resampling in digital images, IEEE Trans. Inf. Forensics Secur. (2008).

[8] M.C. Stamm, S.K. Tjoa, W.S. Lin, K.J.R. Liu, Anti-forensics of JPEG compression, in: IEEE International Conference on Acoustics, Speech and Signal Processing, 2010.

[9] C. Pasquini, G. Boato, JPEG compression anti-forensics based on first significant digit distribution, in: IEEE International Workshop on Multimedia Signal Processing, MMSP, 2013.

[10] G. Cao, Y. Zhao, R. Ni, H. Tian, Anti-forensics of contrast enhancement in digital images, in: ACM Workshop on Multimedia and Security, 2010.

[11] D. Dang-Nguyen, I. Gebru, V. Conotter, G. Boato, F. De Natale, Counter-forensics of median filtering, in: 2013 IEEE 15th International Workshop on Multimedia Signal Processing, MMSP, 2013.

[12] M.C. Stamm, X. Zhao, Anti-forensic attacks using generative adversarial networks, in: H.T. Sencar, L. Verdoliva, N. Memon (Eds.), Multimedia Forensics, 2022.

[13] M. Barni, M.C. Stamm, B. Tondi, Adversarial multimedia forensics: Overview and challenges ahead, in: 2018 26th European Signal Processing Conference, EUSIPCO, 2018, pp. 962–966.

[14] Z. Chen, B. Tondi, X. Li, R. Ni, Y. Zhao, M. Barni, A gradient-based pixel-domain attack against SVM detection of global image manipulations, in: IEEE Workshop on Information Forensics and Security, 2017.

[15] D. Gragnaniello, F. Marra, G. Poggi, L. Verdoliva, Analysis of adversarial attacks against CNN-based image forgery detectors, in: European Signal Processing Conference, EUSIPCO, 2018.

[16] W. Li, B. Tondi, R. Ni, M. Barni, Increased-confidence adversarial examples for deep learning counter-forensics, in: International Workshop on Pattern Recognition, 2021.

[17] D. Cozzolino, J. Thies, A. Rossler, M. Niesner, L. Verdoliva, SpoC: Spoofing camera fingerprints, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPRW, 2021.

[18] C. Chen, X. Zhao, M.C. Stamm, Generative adversarial attacks against deep-learning-based camera model identification, IEEE Trans. Inf. Forensics Secur. (2019).

[19] N. Papernot, P. McDaniel, A. Sinha, M.P. Wellman, SoK: Security and privacy in machine learning, in: 2018 IEEE European Symposium on Security and Privacy, EuroS&P, 2018.

[20] A. Rozsa, M. Günther, T.E. Boult, LOTS about attacking deep features, in: 2017 IEEE International Joint Conference on Biometrics, IJCB, 2017.

[21] M. Barni, M.C. Stamm, B. Tondi, Adversarial multimedia forensics: Overview and challenges ahead, in: 2018 26th European Signal Processing Conference, EUSIPCO, 2018.

[22] 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. 26 (7) (2017).

[23] D. Cozzolino, G. Poggi, L. Verdoliva, Splicebuster: A new blind image splicing detector, in: IEEE International Workshop on Information Forensics and Security, WIFS, 2015.

[24] T.-T. Ng, S.-F. Chang, A data set of authentic and spliced image blocks, in: ADVENT Technical Report #203-2004-3 Columbia University, 2004.

[25] T. Carvalho, F.A. Faria, H. Pedrini, R. da S. Torres, A. Rocha, Illuminant-based transformed spaces for image forensics, IEEE Trans. Inf. Forensics Secur. 11 (4) (2016).

[26] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9 (1) (1979).