Camera Trace Erasing

Chang Chen*[1], Zhiwei Xiong ✉ [1], Xiaoming Liu [2], and Feng Wu[1]
[1] University of Science and Technology of China [2] Michigan State University
changc@mail.ustc.edu.cn, {zwxiong,fengwu}@ustc.edu.cn, liuxm@cse.msu.edu

Abstract

Camera trace is a unique noise produced in digital imaging process. Most existing forensic methods analyze camera trace to identify image origins. In this paper, we address a new low-level vision problem, camera trace erasing, to reveal the weakness of trace-based forensic methods. A comprehensive investigation on existing anti-forensic methods reveals that it is non-trivial to effectively erase camera trace while avoiding the destruction of content signal. To reconcile these two demands, we propose Siamese Trace Erasing (SiamTE), in which a novel hybrid loss is designed on the basis of Siamese architecture for network training. Specifically, we propose embedded similarity, truncated fidelity, and cross identity to form the hybrid loss. Compared with existing anti-forensic methods, SiamTE has a clear advantage for camera trace erasing, which is demonstrated in three representative tasks. Code and dataset are available at https://github.com/ngchc/CameraTE.

1. Introduction

Noise is inevitable in digital imaging process. Camera trace is such a kind of noise that is unique to each type of imaging device. Specifically, camera trace is produced by the different response characteristics of camera sensor to light [30], and then manipulated by the in-camera processing pipeline [34]. Therefore, camera trace implicitly encodes information of camera type into the imaging results in a form of noise. Based on the analysis on camera trace, researches have proposed a variety of methods for image forensic tasks, in terms of origin identification [14, 29], tampering detection [7, 52, 53], and forgery localization [10, 32, 47], to name a few. These methods play an important role in the steady development of image-based social networks [51].

Nevertheless, there is little systematic research on the performance of these forensic methods in an adversarial case. In this paper, we address camera trace erasing in order to reveal the weakness of trace-based forensic methods. As an example shown in Fig. 1, a median filter effectively degrades the classification accuracy of a forensic method [29], yet at the cost of signal destruction. While a real-world image denoiser [48] gently removes the visible noise, the residual camera trace in the processed image is still sufficient for the classifier to make a right prediction. Therefore, it is non-trivial to effectively erase camera trace while avoiding the destruction of content signal.

*This work was partially done when Chang Chen was a visiting scholar at Michigan State University.

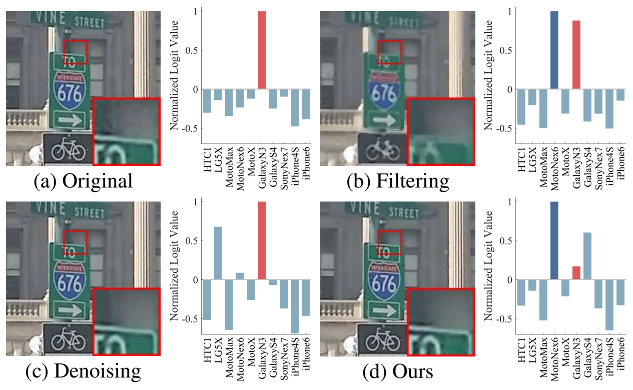


Figure 1. An example of camera trace erasing in the classification task. (a) Given an image, the state-of-the-art classifier [29] can predict the right image origin (red bar) with high confidence (measured by normalized logit value). (b) A median filter effectively erases camera trace, which misleads the prediction (dark blue bar), yet at the cost of signal destruction. (c) A real-world image denoising method [48], though gently removing the visible noise, is not effective enough to erase camera trace. (d) Our proposed method, SiamTE, effectively erases the camera trace without visible destruction of content signal. Zoom in for a better visual experience.

We propose Siamese Trace Erasing (SiamTE) to reconcile these two demands, which is implemented by a Convolutional Neural Network (CNN). Specifically, we design a novel hybrid loss on the basis of Siamese architecture [9] for network training. This hybrid loss contains three terms: embedded similarity, truncated fidelity, and cross identity. For the embedded similarity loss, we gather images captured by different types of cameras as a group of input, and measure the similarity between network outputs. Theoretically, given a suitable metric for camera trace, the similarity of images captured by different types of cameras will increase along with the decrease of camera trace. It is thus possible to measure the degree of camera trace erasing by calculating this similarity. Inspired by deep metric learning [27], we adopt the normalized Euclidean distance in a learned embedding space to be such a metric.

However, the embedded similarity loss alone is not sufficient to guide the network towards desired outputs, mainly due to the issue of over-manipulation. We then introduce a truncated fidelity loss to restrict the degree of image manipulation, by minimizing the Manhattan distance between network inputs and outputs. More specifically, we truncate the distance values below a threshold to zero, which preserves essential manipulation for camera trace erasing while avoiding potential over-manipulation. Besides the above two loss items, we further propose a cross identity loss for better disentanglement between camera trace and content signal.

To evaluate the anti-forensic performance of SiamTE, we conduct experiments on two datasets, in which images are captured by different types of cameras. We take three forensic tasks, *i.e.*, classification, clustering, and verification into consideration. In the classification task, we adopt CNN-based methods for evaluation [29]. In the clustering task, we perform K-means clustering on CNN-extracted features [3]. In the verification task, we adopt a classic forensic method using hand-crafted features [14]. Compared with the existing methods, *i.e.*, filtering, compression, denoising, deblocking, and gradient-based adversarial methods, SiamTE significantly boosts the anti-forensic performance in all three tasks, without visible destruction of content signal. It demonstrates a clear advantage of SiamTE for camera trace erasing.

Contributions of this paper are summarized as follows:

- We address a new low-level vision problem, termed as camera trace erasing, to reveal the weakness of trace-based forensic methods.
- We propose SiamTE as an advanced solution and design a novel hybrid loss based on Siamese architecture for network training.
- SiamTE achieves a significant and consistent performance improvement over existing anti-forensic methods in terms of different datasets and tasks.

2. Related Work

Image anti-forensics. In order to counter image forensics, researches have proposed a variety of anti-forensic methods to disguise the manipulation history of images (Fig. 2). Among these methods, median filter and JPEG compression attract most research interests. Kirchner *et al.* adopt a median filter to hide traces of image resampling [25]. Though effective, the median filter itself will leave a distinctive pattern, termed as streaking artifact [4, 21, 49]. As a compensation, researches propose methods for streaking artifact removal, in order to disguise the manipulation history of median filter [7, 23]. For JPEG compression, Fan *et al.* propose a variational approach to hide the traces (*i.e.*, the blocking artifact) of compression [12]. Furthermore, a dictionary-based method [2] and a GAN-based method [31] are also proposed for JPEG anti-forensics. In addition to deblocking after compression, researches propose anti-forensic methods in the JPEG compression process, by adding dithering to the transform coefficients [39, 44, 45].

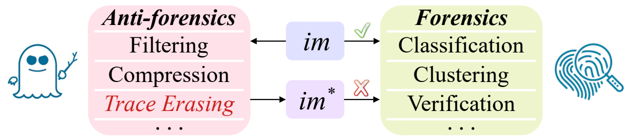


Figure 2. Image anti-forensics and image forensics. Forensics could fail when im is processed to im^* by anti-forensic methods.

Camera trace erasing can be categorized to image anti-forensics. Compared with the existing settings, we make a step forward to address a more realistic trace. Unlike the streaking artifact caused by median filter and the blocking artifact caused by JPEG compression, it is difficult to conclude a fixed pattern for camera trace, since it varies from camera to camera. Besides, we involve filtering, compression, and deblocking methods for comparison and conduct a comprehensive investigation to these anti-forensic methods in the problem of camera trace erasing.

Adversarial technique. The adversarial technique is designed to verify the security of defense system [40]. In this research direction, adversarial machine learning has attracted a lot of attention, due to the rapid development of learning-based methods [28]. Researchers have discovered that an adversarial method can easily trigger malfunctions of a trained neural network, by adding gradient-based perturbation to network inputs [15, 16, 17].

Our proposed SiamTE can be viewed as an adversarial method, since it degrades the performance of trace-based forensic methods. But different from the common solutions, SiamTE works *without* the support of gradient information, which makes it capable of handling more kinds of images, whose camera types are out of the known labels.

Real-world image denoising. Image denoising is a classic research topic in low-level computer vision. Recently, researchers have widened their focus from synthetic noise (*e.g.,* Gaussian noise) [5, 8, 11, 26, 46, 50] to the real-world ones [1, 6, 20, 34, 37, 48]. Since it is difficult to characterize a real-world noise produced by the complex and diverse in-camera processing pipelines, researcher have to define the noise-free image, in order to obtain the ground truth for evaluation. Two kinds of methods have been proposed for this definition, *i.e.*, (a) multi-frame averaging [1, 34] and (b) paired-acquisition under low-ISO value [38].

Camera trace can be viewed as one kind of real-world noise. However, the existing definitions of noise-free images are not suitable for camera trace erasing. For (a), researchers have found that a part of camera trace can still survive or even be enhanced after frame averaging [19, 30]. For (b), an image cannot get rid of camera trace even if it is captured at a low-ISO setting. Thus, we directly define camera trace based on its characteristics. To the best of our knowledge, it is the first time that a specific definition is provided to a real-world noise. Besides, we adopt two representative real-world denoising methods for comparison and demonstrate the advantage of our proposed method.

3. Siamese Trace Erasing

3.1. Problem formulation

We view a captured image (im for short) as two parts. One is camera trace (trs for short) and the other is content signal (sig for short), which can be formulated as

$$im = sig + trs. \tag{1}$$

The goal of a camera trace erasing method $F(\cdot)$ is to achieve $F(im) = sig$. As detailed in Sec. 2, existing methods cannot provide a suitable definition to either the sig part or the trs part for a single image.

To address this issue, we propose to take multiple images into consideration at the same time. Firstly, we define that camera trace should be a distinguishable part in an image. In other words, a certain kind of camera trace should be different from the other one, when these two images are captured by different types of cameras. From this definition, there shall exist a certain kind of similarity between images, which increases along with the decrease of the distinguishable part (*i.e.*, camera trace) in each image. Taking two images (im_1 and im_2) as an example, we denote $\phi(\cdot,\cdot)$ as the similarity between two images. With these notations, we have an inequation as

$$\phi(F_1(im_1), F_1(im_2)) > \phi(F_2(im_1), F_2(im_2)), \tag{2}$$

when a trace erasing method $F_1(\cdot)$ is better than $F_2(\cdot)$.

Moreover, we define that camera trace should be the *only* distinguishable part, regardless of the content signal. Ideally, when $sig_n = F(im_n), trs_n = im_n - F(im_n), n = 1, 2$, we have an equation as

$$\phi(sig_1 + trs_2, sig_2 + trs_1) = \phi(sig_1 + trs_1, sig_2 + trs_2). \tag{3}$$

We use the above formulation to define camera trace and motivate our proposed trace erasing method.

3.2. Hybrid loss guided SiamTE

Let $F_\Theta(\cdot)$ denote a parametric method for camera trace erasing, where Θ is the trainable parameters. In this paper, we adopt the CNN structure proposed in [6] as an embodiment of $F_\Theta(\cdot)$. It is worth mentioning that, the focus of this paper is not the network design. $F_\Theta(\cdot)$ can be implemented by other network structures.

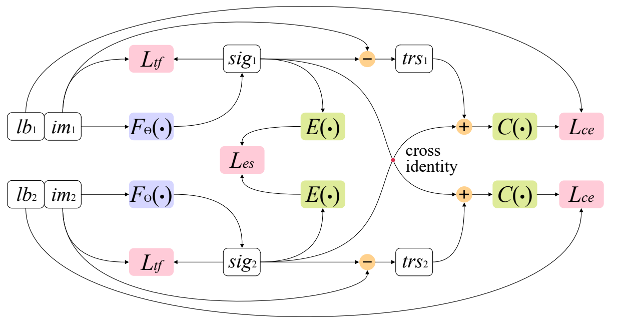


Figure 3. Flowchart of training strategy. $im_n (n = 1, 2)$ denote images captured by the n^{th} type of camera with labels $lb_n (n = 1, 2)$. sig_n and trs_n denote the estimated content signal and camera trace, respectively. $F_\Theta(\cdot)$ denotes a parametric method for camera trace erasing. $E(\cdot)$ denotes a mapping for image embedding. $C(\cdot)$ denotes a classier for image origin identification. The hybrid loss contains embedded similarity L_{es}, truncated fidelity L_{tf}, and the cross identity part with cross-entropy loss L_{ce}. **Blue**: trainable model, **Green**: fixed oracles, and **Red**: loss functions.

For the setup of network, we adopt the Siamese architecture proposed in [9]. Taking two images as an example, we illustrate the Siamese architecture in Fig. 3, where $F_\Theta(\cdot)$ is duplicated in two branches with shared parameters. Such a design can be easily generalized to multiple images, by adding more branches. Following the name of architecture, we term our proposed method as Siamese Trace Erasing (*SiamTE*). To train the network, we propose a hybrid loss as detailed below and shown in Fig. 3.

Embedded similarity loss L_{es}. Motivated by inequation (2), we propose the embedded similarity loss to guide the network training. Inspired by deep metric learning [27], we adopt a learned metric to calculate the similarity between images, by embedding images to a trace-related space with $E(\cdot)$. In this embedding space, we calculate the normalized Euclidean distance between features to obtain the similarity. The calculation process of embedded similarity loss is summarized in Algorithm 1.

Truncated fidelity loss L_{tf}. Generally, the intensity of camera trace is limited compared to content signal since it is a by-product in imaging process. Motivated by this prior knowledge, we propose truncated fidelity loss to restrict the manipulation of a camera trace method $F_\Theta(\cdot)$. Specifically, we calculate the Manhattan distance between im and $F_\Theta(im)$ to measure the degree of manipulation. To preserve essential manipulation while avoiding potential over-manipulation, we truncate the distance values below a threshold T to zero, which can be formulated as

$$L_{tf} = \begin{cases} |im - F_\Theta(im)|, & |im - F_\Theta(im)| > T \\ 0, & |im - F_\Theta(im)| \le T \end{cases}. \tag{4}$$

Cross identity loss L_{ci}. Motivated by equation (3), we propose cross identity loss for better disentanglement between camera trace and content signal. Suppose we have G

Algorithm 1 Calculation of embedded similarity loss

```
Require:
 1: $im[\cdot]$: $G$ images captured by different cameras;
 2: $F_\Theta(\cdot)$: a method for trace erasing with parameters $\Theta$;
 3: $E(\cdot)$: a learned mapping for image embedding;
 4: $N(\cdot)$: an operator for $L_2$ normalization;
 5: $S(\cdot, k)$: an operator for cyclic shift with $k$ step size;
 6: $D(\cdot, \cdot)$: an operator to calculate Euclidean distance;
 7: $M$: a margin for Euclidean distance.
Output: embedded similarity loss of $im$, $L_{es}$
 8: for each $g \in [1, G]$ do
 9:     Trace erasing: $sig[g] = F_\Theta(im[g])$;
10:     Embedding: $feat[g] = E(sig[g])$;
11:     Normalization: $feat[g] = N(feat[g])$;
12: end for
13: Initialize $L_{es}$ to zero;
14: for $k = 1$; $k < G$; $k = k + 1$ do
15:     $dist = \max(0, D(feat, S(feat, k)) - M)$;
16:     $L_{es} = L_{es} + \text{mean}(dist)$;
17: end for
18: Average: $L_{es} = L_{es}/(G - 1)$;
19: return $L_{es}$
```

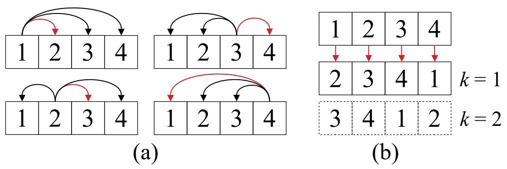


Figure 4. Calculating one-to-one operation in a parallel computing friendly way. Given G images in a group ($G = 4$ in this example), the number of sequential executions can be reduced from (a) P_G^2 (or C_G^2 when an operation is commutative) to (b) $G - 1$. With the cyclic shift $S(\cdot, k)$, the operations colored in red arrows can be calculated in parallel. For each execution, the step size of cyclic shift k traverses from 1 to $G - 1$.

types of cameras, images captured from different types of cameras are grouped as network input $im_n, n = 1, 2, ..., G$. Let trs_m denotes a camera trace produced by the m^{th} camera. The cross identity loss aims to maximize the probability of synthetic images $sig_n + trs_m, n = 1, 2, ..., G (n \neq m)$ to be identified as the ones captured by the m^{th} camera device. In our implementation, we combine an estimated trace $im_m - F_\Theta(im_m)$ with signals from the other devices $F_\Theta(im_n), n = 1, 2, ..., G(n \neq m)$, to obtain these synthetic images. Then, we forward them into a trained classifier $C(\cdot)$ to obtain feedback. The calculation process of cross identity loss is summarized in Algorithm 2.

3.3. Implementation details

In Algorithms 1 and 2, we adopt a cyclic shift operator $S(\cdot, k)$ with a step size of k, in order to calculate the one-to-one operation between multiple images in a parallel computing friendly way. We illustrate the operation of cyclic shift in Fig. 4. For the image origin classifier $C(\cdot)$, we train a ResNet [18] on KCMI+ (a dataset detailed in Sec. 4.1). The weights of convolutions in ResNet are initialized by an ImageNet pretrained model [35]. After training, we utilize the stacked convolutions in this network as the embedding function $E(\cdot)$ for L_{es}. Lastly, we linearly combine the above three loss functions to form the hybrid loss as $\lambda_1 L_{es} + \lambda_2 L_{tf} + \lambda_3 L_{ci}$, where $\lambda_n (n = 1, 2, 3)$ denote the weighting factor.

Algorithm 2 Calculation of cross identity loss

```
Require:
 1: $im[\cdot]$: $G$ images captured by different cameras;
 2: $lb[\cdot]$: the corresponding origin labels of $im[\cdot]$;
 3: $F_\Theta(\cdot)$: a method for trace erasing with parameters $\Theta$;
 4: $C(\cdot)$: a method for image origin classification;
 5: $S(\cdot, k)$: an operator for cyclic shift with $k$ step size;
 6: $L_{ce}(\cdot, \cdot)$: an operator to calculate cross-entropy loss.
Output: cross identity loss of $im$, $L_{ci}$
 7: for each $g \in [1, G]$ do
 8:     Trace extraction: $trs[g] = im[g] - F_\Theta(im[g])$;
 9: end for
10: Initialize $L_{ci}$ to zero;
11: for $k = 1$; $k < G$; $k = k + 1$ do
12:     Cross identity: $pred = C(F_\Theta(im) + S(trs, k))$;
13:     $L_{ci} = L_{ci} + \text{mean}(L_{ce}(pred, S(lb, k)))$;
14: end for
15: Average: $L_{ci} = L_{ci}/(G - 1)$;
16: return $L_{ci}$
```

4. Experiments and Results

4.1. Datasets and settings

KCMI. Kaggle Camera Model Identification (KCMI) is a dataset proposed by IEEE's Signal Processing Society [43]. In KCMI, 2,750 images are captured with 10 types of cameras. We separate 550 images from it to build a test set KCMI-550, with 55 images for each camera. For training and validation, we first retrieve and download additional 2,800 images from Flickr, which are captured with the same 10 types of cameras. Then, we combine them with the rest 2,200 images in KCMI to build KCMI+, with 500 images for each camera.

VISION. It is a large-scale dataset for source identification [42], in which images are captured by 30 types of cameras. Among these cameras, 28 are different from those in KCMI. We adopt 1,500 images from VISION to build a test set VISION-1500, with 50 images for each camera.

Settings of training. We adopt KCMI+ to train SiamTE. Images from KCMI+ are randomly copped into patches with a size of 336×336. We randomly gather 4 patches

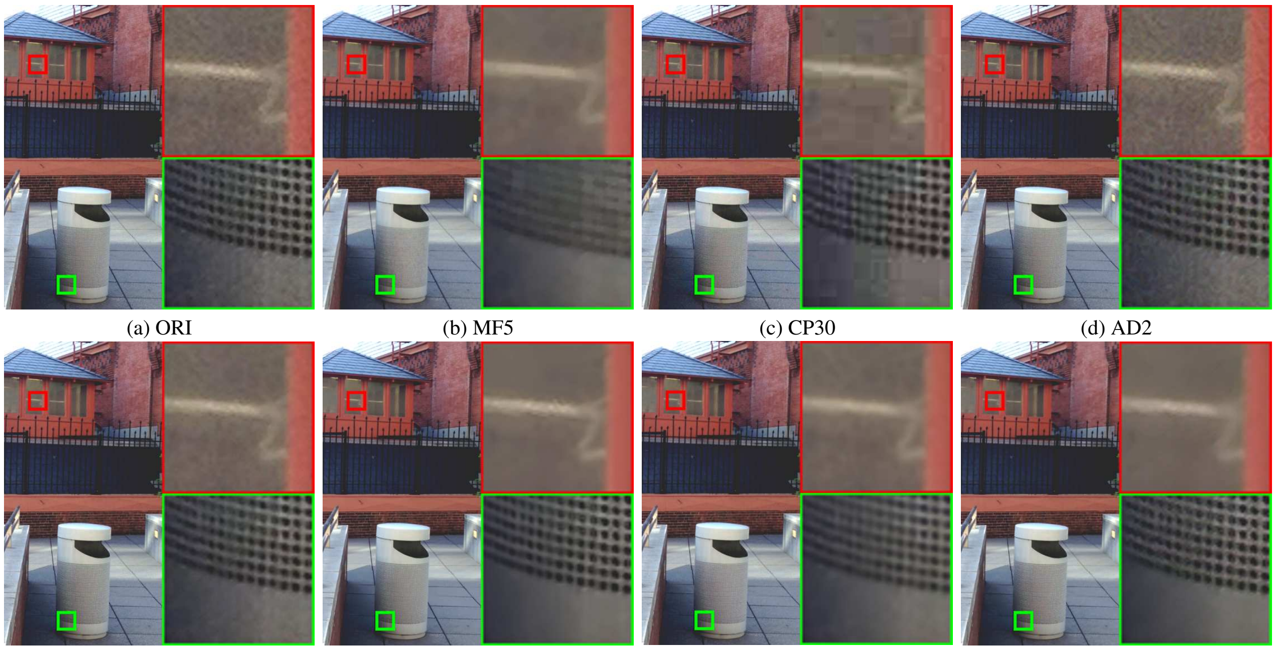

(a) ORI (b) MF5 (c) CP30 (d) AD2
(e) DB (f) DN-I (g) DN-E (h) Ours

Figure 5. Visual comparison on an image from KMCI-550.

as a group (*i.e.*, $G = 4$), in which patches are cropped from images captured by different types of cameras. 64 groups are randomly gathered into a mini-batch for the stochastic gradient descent. We adopt Adam [24] for training with the momentum factor set as 0.9. In our implementations, we set λ_1:λ_2:λ_3 as 3:1000:1 or 3:500:1 for the hybrid loss. For the hyper-parameters, we set the margin $M = 0.5$ and the threshold $T = 3$.

4.2. Image forensic tasks and metrics

Classification. In this task, given an input image, the forensic methods predict a camera type as output. We adopt two kinds of CNN-based classifiers proposed in [29] for evaluation, named as ResNet50 and DenseNet201 according to respective network structures. Since the input size of classification networks (224×224) is much smaller than that of a full image, we randomly crop 4 patches in an image as its representative, and choose the majority prediction among 4 patches as the final output. Since the classification accuracy varies with different cropped patches, we repeat each experiment 10 times and report the averaged results. KCMI-550 is adopted for evaluation.

Clustering. Besides the classification task conducted on KCMI-550, we perform clustering on VISION-1500 to evaluate the generalization ability of our proposed method. Specifically, for a SiamTE trained on KCMI+, most of camera types on VISION-1500 are *unseen*[1] during training. Thus, clustering is a more challenging task. We adopt the stacked convolutional layers in ResNet50 and DenseNet201 [29] for feature extraction, and perform K-means clustering on extracted features [3]. Since VISION-1500 contains 30 types of cameras, we set the number of clustering center $K = 30, 60$, and 90 for evaluation, respectively. In order to provide quantitative results for clustering performance, we define an accuracy for clustering. Specifically, in a group of clustered images, we assign the majority camera type to them as predictions and compare these predictions with known image origins. Therefore, such a clustering accuracy will decrease when images with different camera types are

[1]The *unseen* camera type is the one not contained in the training set (*i.e.*, *unknown* to methods), but we still know its origin.

Table 1. Quantitative comparison in the classification task.

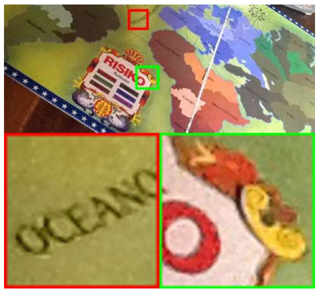
Method	Accuracy		NIQE	L_1 dist. to ORI
	ResNet50	DenseNet201		
ORI [1]	99.80 ± 0.18	99.87 ± 0.12	3.083	-
MF3 [7]	66.62 ± 2.08	75.95 ± 1.43	3.939	2.055
MF5 [7]	21.33 ± 1.09	44.33 ± 1.64	4.799	3.849
GF3 [41]	90.18 ± 0.66	93.07 ± 1.02	4.256	2.186
GF5 [41]	73.46 ± 1.56	80.18 ± 1.29	4.593	3.115
CP30 [36]	56.31 ± 1.64	58.35 ± 1.06	4.283	3.632
CP40 [36]	77.02 ± 1.51	75.27 ± 1.22	3.838	3.202
CP50 [36]	91.29 ± 0.73	86.56 ± 1.37	3.510	2.917
AD1 [16]	45.49 ± 1.43	55.53 ± 1.34	3.265	0.988
AD2 [16]	21.13 ± 1.25	32.73 ± 1.53	3.934	1.973
DB [50]	90.04 ± 1.03	92.82 ± 1.05	3.044	1.301
DN-I [48]	59.49 ± 1.79	64.87 ± 1.06	3.961	2.017
DN-E [6]	44.42 ± 1.66	56.82 ± 1.32	4.008	2.710
Ours	$\mathbf{20.42} \pm 1.19$	$\mathbf{28.11} \pm 1.76$	3.676	2.004

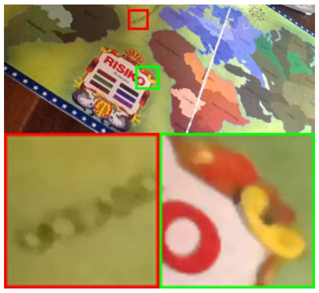
[1] Images from KCMI-550 (ORI) are adopted for evaluation.

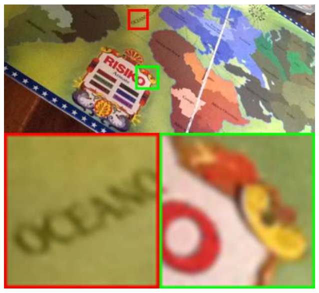
Table 2. Quantitative comparison in the clustering task. Images from VISION-1500 (ORI) are adopted for evaluation.

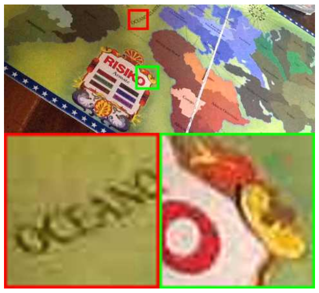
Method [1]	Accuracy						NIQE	L_1 dist. to ORI
	ResNet50 ($K = 30$) [2]	ResNet50 ($K = 60$)	ResNet50 ($K = 90$)	DenseNet201 ($K = 30$) [2]	DenseNet201 ($K = 60$)	DenseNet201 ($K = 90$)		
ORI	56.79 ± 2.08	70.27 ± 1.01	75.44 ± 1.30	59.06 ± 1.79	73.09 ± 1.27	78.77 ± 1.53	3.585	-
MF3 [7]	43.03 ± 1.36	54.42 ± 1.15	58.60 ± 1.08	45.23 ± 1.14	55.73 ± 1.13	61.79 ± 1.32	4.043	1.959
MF5 [7]	31.45 ± 1.06	38.53 ± 0.63	42.79 ± 1.15	34.97 ± 1.02	42.79 ± 0.98	48.14 ± 0.76	5.227	3.525
GF3 [41]	49.95 ± 1.74	61.68 ± 1.55	64.99 ± 1.25	51.17 ± 0.98	61.02 ± 1.74	66.19 ± 0.86	4.413	2.029
GF5 [41]	44.69 ± 1.27	51.82 ± 0.93	55.87 ± 0.56	41.47 ± 1.49	52.43 ± 1.56	57.40 ± 1.02	4.795	2.847
CP30 [36]	26.55 ± 1.06	35.32 ± 1.55	39.41 ± 0.74	25.23 ± 1.08	33.27 ± 0.80	37.68 ± 1.12	4.811	3.564
CP40 [36]	33.57 ± 1.62	43.45 ± 1.61	47.82 ± 1.60	31.44 ± 1.13	40.28 ± 0.89	44.12 ± 1.10	4.253	3.175
CP50 [36]	40.91 ± 1.30	49.62 ± 1.76	53.70 ± 0.95	35.95 ± 1.24	45.78 ± 0.84	50.87 ± 1.19	3.965	2.918
DB [50]	52.87 ± 1.61	64.13 ± 1.68	70.22 ± 1.56	53.62 ± 1.92	67.44 ± 0.90	72.44 ± 1.64	3.299	1.327
DN-I [48]	35.83 ± 1.11	46.44 ± 0.87	51.23 ± 0.73	37.06 ± 1.09	46.31 ± 1.17	51.84 ± 0.81	4.275	2.214
DN-E [6]	28.47 ± 0.89	37.83 ± 1.40	42.64 ± 0.86	28.45 ± 1.01	37.76 ± 0.71	43.71 ± 0.82	4.128	2.613
Ours	$\mathbf{23.44} \pm 0.82$	$\mathbf{33.37} \pm 1.06$	$\mathbf{37.30} \pm 0.83$	$\mathbf{22.94} \pm 0.97$	$\mathbf{31.99} \pm 1.05$	$\mathbf{37.24} \pm 0.98$	4.082	2.097

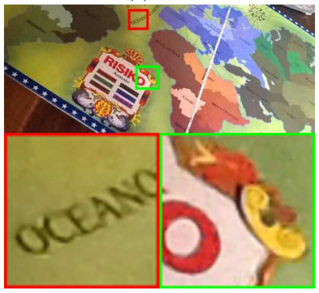
[1] AD is not involved for comparison, since it cannot generalize to images with unseen camera types, as detailed in Sec. 4.3.
[2] Features extracted by the stacked convolutional layers are adopted for K-means clustering [3].

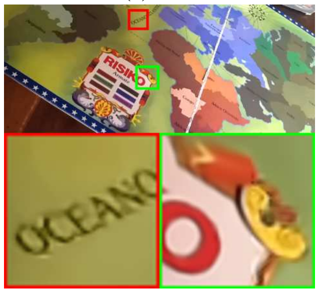

(a) ORI

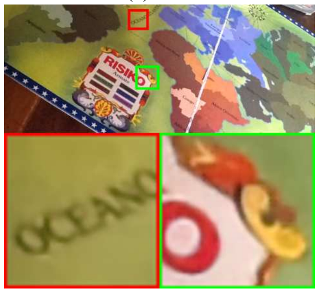

(b) MF5


(c) GF5


(d) CP30


(e) DB


(f) DN-I


(g) DN-E

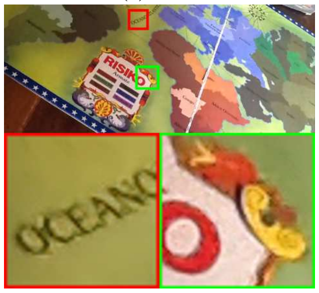

(h) Ours

Figure 6. Visual comparison on an image from VISION-1500.

wrongly clustered together. We randomly crop 128 patches in an image as its representative and repeat each experiment 10 times to obtain the averaged results.

Verification. Furthermore, we conduct evaluation in the verification task. Given two input images, the forensic method in this task predicts if they are captured by the same type of camera. We adopt KCMI+ to build fingerprints for each type of camera, by averaging the extracted noise residuals from images captured by the same type of camera. We adopt a hand-crafted method to extract the noise residual [14]. Then, we adopt Peak-to-Correlation Energy (PCE) [13] to measure the correlation between an image from KCMI-550 and its corresponding camera fingerprint. The higher PCE means a forensic method can identify the image origin with higher confidence [22].

Metrics for image assessment. In addition to the specific metrics for each task, we provide an auxiliary metric to measure the manipulation degree of a trace erasing method, by calculating the Manhattan (L_1) distance between input and output of the method. Generally, a large L_1 distance indicates that the input image has been heavily manipulated, which usually causes a destruction to content signal. While a small L_1 distance may indicate that a method fails to perform a valid manipulation on the input. Moreover, we conduct objective quality assessments using the non-reference metric NIQE [33] (a smaller value denotes a higher quality).

Table 3. Quantitative comparison in the verification task. Images from KCMI-550 (ORI) are adopted for evaluation.

Method	Peak-to-Correlation Energy											NIQE	L_1 dist. to ORI
	1	2	3	4	5	6	7	8	9	10	Avg. [1]		
ORI	647.7	390.7	1604.0	300.2	2068.2	1388.6	1170.4	11.9	2896.0	2592.8	1307.1	3.772	-
MF3 [7]	236.7	133.9	630.8	119.9	623.6	612.2	400.6	5.9	686.4	921.8	437.1	4.699	2.325
GF3 [41]	411.9	275.6	1143.9	271.1	1290.1	1066.1	897.5	9.0	1850.8	1717.5	893.3	4.589	2.521
CP50 [36]	65.7	58.4	250.3	61.1	274.4	144.0	148.7	2.5	343.9	493.5	184.3	4.070	3.104
AD2 [16]	321.4	196.5	1061.8	152.2	1118.9	637.7	643.8	7.1	1802.2	1568.0	751.0	4.855	1.994
DB [50]	399.2	267.4	1197.9	229.8	1407.9	877.0	862.0	8.2	2388.6	1740.8	937.9	3.832	1.312
DN-I [48]	226.5	147.0	553.4	58.7	339.4	418.0	311.3	5.5	183.9	471.2	271.5	4.600	2.097
DN-E [6]	263.0	198.8	854.7	209.6	989.5	584.3	682.7	6.3	1613.2	1219.6	662.2	4.801	3.007
Ours	53.7	66.9	107.1	74.1	271.9	74.9	233.4	2.1	410.6	266.5	**156.1**	4.652	2.491

[1] 10 camera types represented by serial numbers are described in the supplementary document. Images are centrally cropped to simplify the calculation.

4.3. Evaluation on anti-forensics performance

We conduct a comprehensive investigation on various existing anti-forensic methods for camera trace erasing, including median filter (MF), Gaussian filter (GF), JPEG compression (CP), gradient-based adversarial method (AD) [16], blind image deblocking (DB) [50], internal similarity based real-world denoising (DN-I) [48], and learning-based real-world denoising (DN-E) [6]. The number after MF and GF (*i.e.*, 3 and 5) denotes the kernel size of filter, the number after CP (*i.e.*, 30, 40, and 50) denotes the quality factor, and the number after AD (*i.e.*, 1 and 2) denotes the scale factor of adversarial dithering.

As described in Sec. 4.2, we adopt the L_1 distance to measure the degree of image manipulation. Taking the results of CP in Table 1 as an example, a better anti-forensic performance is achieved with a larger degree of manipulation (*e.g.,* CP30 vs. CP50). However, such a performance improvement comes at the cost of more severe signal destruction (reflected by a larger NIQE value). Thus, to measure the *efficiency* of a camera trace erasing method, we need to consider the degree of manipulation at the same time. An efficient method should achieve good anti-forensic performance with as little manipulation as possible.

According to the quantitative results listed in Tables 1, 2, 3 and visual results shown in Figs. 5 and 6, we analyze the performance of above mentioned anti-forensic methods. Median filter and JPEG compression are effective to erase camera trace, yet the processed images suffer from blurring and blocking artifacts. Gaussian filter is less effective since it cannot significantly degrade the performance of forensic methods even at a large degree of manipulation. The deblocking method [50] cannot provide a valid manipulation to erase camera trace, since it can only remove the part of blocking artifact in camera trace. The overall performance of the two real-world image denoising methods [6, 48] is relatively better than other baseline methods, yet still has a notable gap to ours.

The gradient-based adversarial method is specially designed for CNN-based classifiers [16]. It thus achieves a

Table 4. Ablation study of the hybrid loss.

Hybrid Loss			Accuracy [1]		NIQE	L_1 dist. to ORI
L_{es}	L_{tf}	L_{ci}	ResNet50	DenseNet201		
✓	✓	✓	20.42 ± 1.19	28.11 ± 1.76	3.676	2.004
	✓	✓	37.45 ± 1.19	42.65 ± 2.26	3.695	2.030
✓		✓	10.89 ± 1.11	11.71 ± 0.85	5.291	17.724
✓	✓		11.02 ± 0.43	10.42 ± 0.26	4.610	2.045

[1] Comparisons are conducted on KCMI-550 in the classification task.

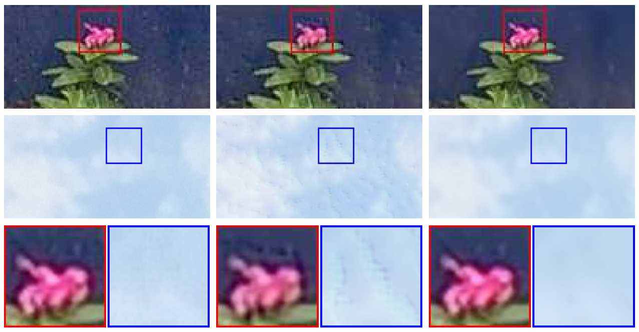


Figure 7. Visual comparison for ablation study. Two image patches from KCMI-550 are adopted for comparison.

satisfactory anti-forensic performance on CNN-based classifiers, which is on par with our proposed method (as listed in Table 1). However, the generalization ability of this adversarial method is poor. On one hand, the adversarial dithering is less effective than ours in the verification task, since a classic forensic method using hand-crafted features rather than a CNN-based one is adopted in this task (as listed in Table 3). On the other hand, the calculation of gradient is dependent on labels. Therefore, the adversarial method cannot generalize to images with unseen camera types, which makes it *not* capable of processing images in the clustering task, thus not listed in Table 2.

Compared with the above baseline methods, our proposed SiamTE is more efficient for camera trace erasing. Specifically, at the similar (or lower) degree of manipulation, SiamTE significantly reduces the classification accu-

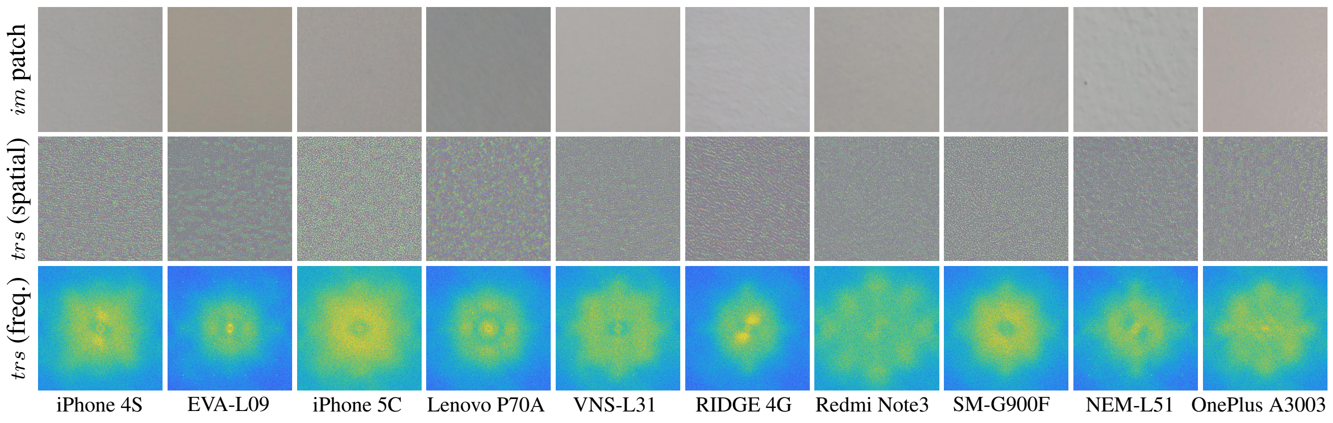


Figure 8. Visualization of camera trace extracted by SiamTE in spatial and frequency domains. Image patches in smooth areas are cropped from VISION with a size of 500×500. Brightness and contrast of camera trace in spatial domain are adjusted for a better visual experience.

racy of ResNet50 from 99.80% to 20.42%, as listed in Table 1. In the clustering task, the performance of K-means clustering is halved by SiamTE, as listed in Table 2. In the verification task, SiamTE achieves 87.2% decrease on the correlation between an image and its camera type, as listed in Table 3. From the visual comparisons conducted in Figs. 5 and 6, the perceptual quality of our results are satisfactory, which is also verified by the lower NIQE values listed in Tables 1 and 2. The comprehensive experiments demonstrate a clear advantage of SiamTE for camera trace erasing over existing anti-forensic methods.

4.4. Ablation study of hybrid loss

Our proposed hybrid loss consists of three parts: (a) embedded similarity L_{es}, truncated fidelity L_{tf}, and cross identity L_{ci}. In this section, we provide ablation study to demonstrate the function of each part. As listed in Table 4, each part of the hybrid loss has a contribution to the overall performance. Without L_{es}, the anti-forensic performance in terms of classification accuracy is significantly weakened. Without L_{tf}, the degree of manipulation loses control, which results in heavy destruction of content signal, as reflected by the large NIQE and L_1 distance. Without L_{ci}, unfavorable artifacts are introduced to the visual results, which degrade the image quality and may reveal the adversarial process, as shown in Fig. 7.

4.5. Analysis on camera trace

In this section, we analyze the extracted camera trace separately. With a trained SiamTE, we extract camera trace from KCMI+ and KCMI-550, respectively. We then train a DenseNet201 classifier on KCMI+ to identify origin from the extracted camera trace instead of the image itself. On KCMI-550, we achieve an accuracy of $93.21 \pm 0.31\%$ with camera trace, which is close to that of $99.87 \pm 0.12\%$ with original images. It verifies that the extracted camera trace contains most of the camera-distinguishable information

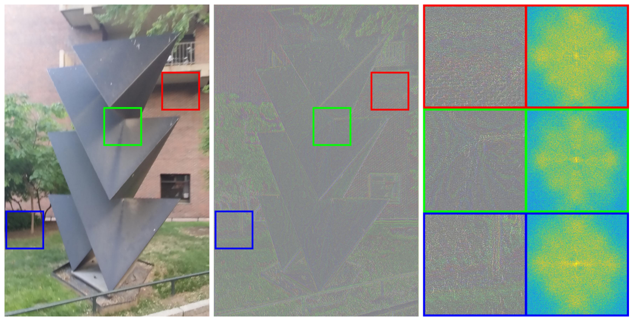


Figure 9. Visualization of camera trace in a single image captured by GalaxyN3. From left to right: original image, camera trace extracted by SiamTE, and patches in spatial and frequency domains.

from original images. We visualize the extracted camera trace in Fig. 8, which demonstrates that camera trace varies with different types of cameras. In comparison, patches in a single image have similar camera trace, as shown in Fig. 9.

5. Conclusion

We address a new low-level vision problem, termed as camera trace erasing, to reveal the weakness of trace-based forensic methods. It is of great importance to verify the security of image forensics. We conduct a comprehensive investigation to existing anti-forensic methods, and propose SiamTE as an advanced solution, which significantly boosts the anti-forensic performance in three representative tasks. We design a novel hybrid loss on the basis of Siamese architecture, which guides SiamTE to effectively erase camera trace without visible destruction of content signal.

Acknowledgement

We acknowledge funding from National Key R&D Program of China under Grant 2017YFA0700800, National Natural Science Foundation of China under Grant 61671419, and China Scholarship Council under Grant 201906340108.

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S. Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, 2018. 2

[2] Nasser Afshin, Farbod Razzazi, and Mohammad-Shahram Moin. A dictionary based approach to JPEG anti-forensics. In *International Conference on Intelligent Systems*, 2016. 2

[3] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In *ACM-SIAM Symposium on Discrete Algorithms*, 2007. 2, 5, 6

[4] Alanconrad Bovik. Streaking in median filtered images. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(4):493–503, 1987. 2

[5] Chang Chen, Zhiwei Xiong, Xinmei Tian, and Feng Wu. Deep boosting for image denoising. In *ECCV*, 2018. 2

[6] Chang Chen, Zhiwei Xiong, Xinmei Tian, Zheng-Jun Zha, and Feng Wu. Real-world image denoising with deep boosting. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2019. 2, 3, 5, 6, 7

[7] Jiansheng Chen, Xiangui Kang, Ye Liu, and Z Jane Wang. Median filtering forensics based on convolutional neural networks. *IEEE Signal Processing Letters*, 22(11):1849–1853, 2015. 1, 2, 5, 6, 7

[8] Yunjin Chen and Pock Thomas. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 39(6):1256, 2017. 2

[9] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, 2005. 1, 3

[10] Davide Cozzolino and Luisa Verdoliva. Noiseprint: A CNN-based camera model fingerprint. *IEEE Transactions on Information Forensics and Security*, 15(1):144–159, 2020. 1

[11] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007. 2

[12] Wei Fan, Kai Wang, François Cayre, and Zhang Xiong. A variational approach to JPEG anti-forensics. In *ICASSP*, 2013. 2

[13] Miroslav Goljan. Digital camera identification from images–estimating false acceptance probability. In *International Workshop on Digital Watermarking*, 2008. 5

[14] Miroslav Goljan, Jessica Fridrich, and Tomáš Filler. Large scale test of sensor fingerprint camera identification. In *Media Forensics and Security*, 2009. 1, 2, 5

[15] Ian Goodfellow, Patrick McDaniel, and Nicolas Papernot. Making machine learning robust against adversarial inputs. *Communications of the ACM*, 61(7), 2018. 2

[16] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 2, 5, 7

[17] Diego Gragnaniello, Francesco Marra, Giovanni Poggi, and Luisa Verdoliva. Analysis of adversarial attacks against CNN-based image forgery detectors. In *EUSIPCO*, 2018. 2

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4

[19] Gerald C Holst. CCD arrays, cameras, and displays. 1998. 3

[20] Amin Jourabloo, Yaojie Liu, and Xiaoming Liu. Face de-spoofing: Anti-spoofing via noise modeling. In *ECCV*, 2018. 2

[21] Xiangui Kang, Matthew C Stamm, Anjie Peng, and KJ Ray Liu. Robust median filtering forensics using an autoregressive model. *IEEE Transactions on Information Forensics and Security*, 8(9):1456–1468, 2013. 2

[22] Steven M Kay. *Fundamentals of statistical signal processing*. 1993. 5

[23] Dongkyu Kim, Han-Ul Jang, Seung-Min Mun, Sunghee Choi, and Heung-Kyu Lee. Median filtered image restoration and anti-forensics using adversarial networks. *IEEE Signal Processing Letters*, 25(2):278–282, 2018. 2

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 5

[25] Matthias Kirchner and Rainer Bohme. Hiding traces of resampling in digital images. *IEEE Transactions on Information Forensics and Security*, 3(4):582–592, 2008. 2

[26] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void-learning denoising from single noisy images. In *CVPR*, 2019. 2

[27] Brian Kulis. Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5(4):287–364, 2013. 2, 3

[28] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *ICLR*, 2016. 2

[29] Artur Kuzin, Artur Fattakhov, Ilya Kibardin, Vladimir I Iglovikov, and Ruslan Dautov. Camera model identification using convolutional neural networks. In *International Conference on Big Data*, 2018. 1, 2, 5

[30] Jan Lukáš, Jessica Fridrich, and Miroslav Goljan. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 1(2):205–214, 2006. 1, 3

[31] Yingmin Luo, Hanqi Zi, Qiong Zhang, and Xiangui Kang. Anti-forensics of JPEG compression using generative adversarial networks. In *EUSIPCO*, 2018. 2

[32] Siwei Lyu, Xunyu Pan, and Xing Zhang. Exposing region splicing forgeries with blind local noise estimation. *International Journal of Computer Vision*, 110(2):202–221, 2014. 1

[33] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2013. 5

[34] Seonghyeon Nam, Youngbae Hwang, Yasuyuki Matsushita, and Seon Joo Kim. A holistic approach to cross-channel image noise modeling and its application to image denoising. In *CVPR*, 2016. 1, 2

[35] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS Workshops*, 2017. 4

[36] William B Pennebaker and Joan L Mitchell. *JPEG: Still image data compression standard*. 1992. 5, 6, 7

[37] Tobias Plötz and Stefan Roth. Neural nearest neighbors networks. In *NeurIPS*, 2018. 2

[38] Tobias Plötz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *CVPR*, 2017. 2

[39] Zhenxing Qian and Xinpeng Zhang. Improved anti-forensics of JPEG compression. *Journal of Systems and Software*, 91:100–108, 2014. 2

[40] Shilin Qiu, Qihe Liu, Shijie Zhou, and Chunjiang Wu. Review of artificial intelligence adversarial attack and defense technologies. *Applied Sciences*, 9(5):909, 2019. 2

[41] John C Russ, Jens Rindel, and P Lord. *Forensic uses of digital imaging*. 2016. 5, 6, 7

[42] Dasara Shullani, Marco Fontani, Massimo Iuliani, Omar Al Shaya, and Alessandro Piva. Vision: a video and image dataset for source identification. *EURASIP Journal on Information Security*, 2017(1):15, 2017. 4

[43] IEEE's Signal Processing Society. Camera model identification. https://www.kaggle.com/c/sp-society-camera-model-identification. 4

[44] Matthew C Stamm and KJ Ray Liu. Anti-forensics of digital image compression. *IEEE Transactions on Information Forensics and Security*, 6(3):1050–1065, 2011. 2

[45] Matthew C Stamm, Steven K Tjoa, W Sabrina Lin, and KJ Ray Liu. Anti-forensics of JPEG compression. In *ICASSP*, 2010. 2

[46] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. MemNet: A persistent memory network for image restoration. In *ICCV*, 2017. 2

[47] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. ManTra-Net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *CVPR*, 2019. 1

[48] Jun Xu, Lei Zhang, and David Zhang. A trilateral weighted sparse coding scheme for real-world image denoising. In *ECCV*, 2018. 1, 2, 5, 6, 7

[49] Hai-Dong Yuan. Blind forensics of median filtering in digital images. *IEEE Transactions on Information Forensics and Security*, 6(4):1335–1345, 2011. 2

[50] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017. 2, 5, 6, 7

[51] Lilei Zheng, Ying Zhang, and Vrizlynn LL Thing. A survey on image tampering and its detection in real-world photos. *Journal of Visual Communication and Image Representation*, 58:380–399, 2019. 1

[52] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Two-stream neural networks for tampered face detection. In *CVPR Workshops*, 2017. 1

[53] Peng Zhou, Xintong Han, Vlad I Morariu, and Larry S Davis. Learning rich features for image manipulation detection. In *CVPR*, 2018. 1