SpoC: Spoofing Camera Fingerprints

Davide Cozzolino[1] Justus Thies[2] Andreas Rössler[2] Matthias Nießner[2] Luisa Verdoliva[1]
[1]University Federico II of Naples [2]Technical University of Munich

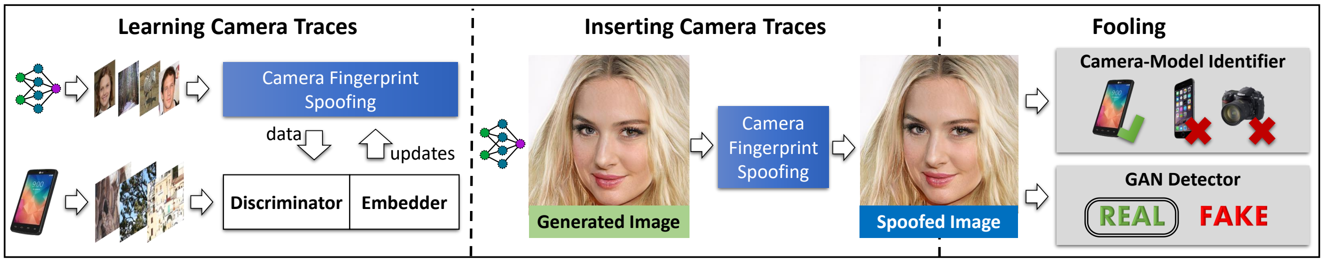

Figure 1: *SpoC* learns to spoof camera fingerprints. It can be used to insert camera traces to a generated image. Experiments show that we can fool both camera-model identifiers and GAN detectors which were not seen during training.

Abstract

Thanks to the fast progress in synthetic media generation, creating realistic false images has become very easy. Such images can be used to wrap rich fake news with enhanced credibility, spawning a new wave of high-impact, high-risk misinformation campaigns. Therefore, there is a fast-growing interest in reliable detectors of manipulated media. The most powerful detectors, to date, rely on the subtle traces left by any device on all images acquired by it. In particular, due to proprietary in-camera processes, like demosaicing or compression, each camera model leaves trademark traces that can be exploited for forensic analyses. The absence or distortion of such traces in the target image is a strong hint of manipulation. In this paper, we challenge such detectors to gain better insight into their vulnerabilities. This is an important study in order to build better forgery detectors able to face malicious attacks. Our proposal consists of a GAN-based approach that injects camera traces into synthetic images. Given a GAN-generated image, we insert the traces of a specific camera model into it and deceive state-of-the-art detectors into believing the image was acquired by that model. Likewise, we deceive independent detectors of synthetic GAN images into believing the image is real. Experiments prove the effectiveness of the proposed method in a wide array of conditions. Moreover, no prior information on the attacked detectors is needed, but only sample images from the target camera.

1. Introduction

There have been astonishing advances in synthetic media generation in the last few years, thanks to deep learning, and in particular to Generative Adversarial Networks (GANs). This technology enabled significant improvement in the level of realism of generated data, increasing both resolution and quality [55]. Nowadays, powerful methods exist for generating an image from scratch [29, 6, 31], and for changing its style [61, 30, 31] or only some specific attributes [12]. These methods are very effective, especially on faces, and allow one to easily change the expression of a person [51, 46] or to modify its identity through face swapping [42, 44]. This manipulated visual content can be used to build more effective fake news. In fact, it has been estimated that the average number of reposts for news containing an image is 11 times larger than for those without images [27]. This raises serious concerns about the trustworthiness of digital content, as testified by the growing attention to the deepfake phenomenon.

The research community has responded to this threat by developing a number of forensic detectors [53]. Some of them exploit high-level artifacts, like asymmetries in the color of the eyes, or anomalies arising from an imprecise estimation of the underlying geometry [41, 59]. However, technology improves so fast that these visual artifacts will soon disappear. Other approaches rely on the fact that any acquisition device leaves distinctive traces on each captured image [11], because of its hardware, or its signal processing suite. They allow associating a media with its acquisition device at various levels, from the type of source, to its

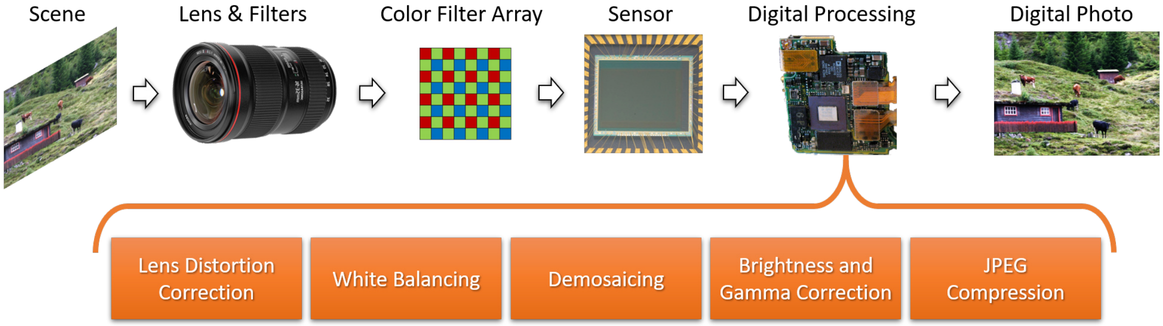

Figure 2: A digital image of a scene contains camera-related traces of the image formation process that could act as a fingerprint of a camera model. The used lenses and filters, the sensor and the manufacturer-specific digital processing pipelines result in unique patterns. These patterns can be used to identify camera models.

brand/model, to the individual device [33]. A major impulse to this field has been given by the seminal work of Lukàs et al. [35], where it has been shown that reliable device identification is possible based on the camera photo-response non-uniformity (PRNU) pattern. This pattern is due to tiny imperfections in the silicon wafer used to manufacture the imaging sensor and can be considered as a type of device fingerprint.

Beyond extracting fingerprints that contain device-related traces, it is also possible to recover camera model fingerprints [14]. These are related to the internal digital acquisition pipeline, including operations like demosaicing, color balancing, and compression, whose details differ according to the brand and specific model of the camera [33] (See Fig.2). Such differences help attribute images to their source camera, but can also be used to better highlight anomalies caused by image manipulations [18, 1]. In fact, the absence of such traces, or their modification, is a strong clue that the image is synthetic or has been manipulated in some way [36, 14]. Detection algorithms, however, must confront with the capacity of an adversary to fool them. This applies to any type of classifier, and is also very well known in forensics, where many counter-forensics methods have been proposed in the literature [3]. Indeed, forensics and counter-forensics go hand in hand, a competition that contributes to improving the level of digital integrity over time.

In this work, we propose a method to synthesize traces of cameras using a generative approach that is agnostic to the detector (i.e., not just targeted adversarial noise). We achieve this by training a conditional generator to jointly fool an adversarial discriminator network as well as a camera embedding network. To this end, the proposed method injects the distinctive traces of a target camera model in synthetic images, while reducing the original generation traces themselves, leading all tested classifiers to attribute such images to the target camera ('targeted attack').

To the best of our knowledge, this is the first work that inserts real camera signatures into synthetic imagery to fool unknown camera identifier. Indeed, previous work on fooling camera model identification, based on adversarial noise [22, 39] or pattern injection [32, 10], always considered only real images. In addition, when generating the attack, assume advance knowledge of the attacked CNN-based classifier or, at least, all camera models used to train them [22, 39, 10]. In contrast, we work in an open set scenario, and require exclusively a suitable number of images coming from the target camera model (in principle, we do not even need to know the camera model itself). There is also some recent work that specifically fool GAN detectors by adding adversarial noise [7] or by removing low-level traces in GAN images [43]. However, the scenario is different from ours, since our aim is to make a synthetic image appear as a real one so as to fool not only a GAN detector, but also a camera model identification method. Hence, our main contributions are the following:

- we devise a GAN-based approach that inserts real camera traces in synthetic images;
- we carry out targeted attacks against CNN-based camera model detectors and show that they are easily fooled by our approach;
- we carry out attacks to GAN detectors without re-training our model and show that they can be easily fooled by our approach.

2. Related work

Adversarial attacks to camera identification. Adversarial attacks are conceived to fool a classifier by adding imperceptible perturbations [19]. [22] and [39] investigate on the vulnerability of deep learning based camera model classifiers in a white box setting. They show that the attack is effective only when the image is uncompressed, while in realistic forensic applications the perturbation noise is hidden by the compression artifacts. The analyses also highlight the difficulty of transferring such attacks in the context of camera model identification. This is part of a more general behavior. Contrary to what happens in many computer vision applications [19, 34], in forensics applications, where detectors rely on tiny variations of the data, transferability for non-targeted attacks is not achieved easily [21, 2]. A different perspective is followed in [9] where, instead of introducing noise, camera traces are deleted from an image to prevent correct identification.

Adversarial attacks to GAN image detection. Synthetic images do not contain camera-related traces. However, they do contain hidden traces related to the pipeline used to generate them [40, 60]. These traces are implicitly or explicitly used to distinguish synthetic images from real ones by several CNN-based architectures [20].

Some recent works proposed different ways to attack these detectors. In [7] it has been investigated the robustness of such classifiers to adversarial attacks both in a white-box and in a black-box scenario. Experiments show that imperceptible perturbations can cause misclassification in both scenarios. A different perspective is pursued in [43], where instead of adding noise, the specific fingerprints that characterize GAN images are removed through an autoencoder-based strategy. While these papers are related to our work, there is a main difference since our objective is twofold. In fact, our strategy is able to fool a GAN detector, but at the same time it can also fool a camera model identification method. In fact, we do not only reduce GAN traces, but we introduce the typical low-level features present in a target camera. This additional feature can be used in a forensic scenario to perform targeted attacks.

Adversarial attacks using generative networks. In the literature, several papers have already addressed the problem of using generative networks to create adversarial examples. In [49] and [56] the adversarial examples are generated from scratch, with no further constraint. Other papers instead are more relevant to our scenario and use a generative approach to slightly modify an existing image [24, 45, 58, 10, 16]. In [45] the classifier under attack is supposed to be perfectly known (white-box scenario) and its gradient is used to train the generator of adversarial samples. In [24], instead, the attacker is only allowed to query the classifier and observe the predicted labels (black-box scenario). With this information, a substitute classifier is trained and its gradient is used to train the generator. The strategies of [45] for white-box scenarios and of [24] for black-box scenarios are adapted by Chen et al. [10] for the specific problem to fool a camera model classifier. In [58], instead, these approaches are further extended by including a discriminator network to distinguish between original and attacked images, pushing the generator to improve the fidelity of the attacked image to the original. To adapt to a face recognition scenario, where the number of classes is not fixed in advance, in [16] the target classifier is replaced with a face matcher based on the cosine similarity in the embedding space.

Unlike previous works, our proposal does not require knowledge of the classifier under attack [45, 58, 10], and not even of the labels output by the network in response to selected queries [24, 58, 10]. Moreover, although the use of an embedder was already proposed in [16], we use a less restrictive loss in the embedding space. In fact, while in [16] the distance of the generated example is minimized with respect to a random sample of the target class, we minimize distances to a representative anchor vector, and focus mainly on critical outliers with respect to this anchor. In addition, our discriminator does not aim to preserve perceptual quality [58, 16], but to improve the generators ability to introduce traces of the target camera model, and to remove peculiar traces of synthetic images. The objective is thus similar to [10]; however, in our work we inject camera model traces in synthetic images and not real ones. More importantly, our strategy does not need to include the camera model classifier in the generation process as done in [10], but requires only images coming from the target distribution.

3. Reference Scenario

In this section, we describe in more detail our reference scenario. We want to show that inconsistencies in camera traces in forensics cannot be reliable even when the attacker has very little knowledge about the classifier.

Targeted attack: Our goal is to make a synthetic image appear as if it was taken by a specific real camera by inserting peculiar traces of the latter. Hence, our attack is targeted, and our aim is to fool CNN-based camera model identification detectors that will recognize the generated image as if it was taken by the target camera model.

Available knowledge: We assume that the attacker has no knowledge about the specific classifier and cannot make queries to it. We only suppose that the training images are drawn from the same distribution, that is, generated by the targeted camera model. However, no knowledge is given on the other camera models on which the classifier is trained.

Visual imperceptibility: We require that the attacked images look realistic and do not present visible artifacts. The generation of realistic synthetic images is out of the scope of this work, and we simply assume they are available. How-

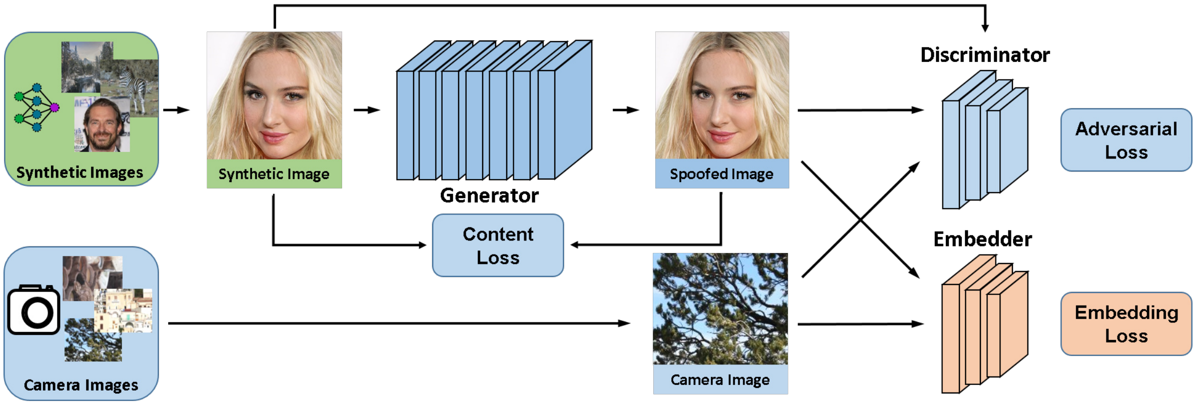

Figure 3: Our architecture is based on three components: a classical GAN setup, using a generator as well as a discriminator, and an embedding network. By applying a content loss between synthetic images as well as spoofed images, we ensure that our generator keeps the image content intact. The pre-trained embedder checks if we only generate specific camera traces. In comparison to regular GAN methods, we use all three involved images for our discriminator, i.e., synthetic, spoofed and camera typical images, and adjust the loss correspondingly.

ever, we require that the attack does not change the image content and introduces a perceptually acceptable distortion.

Processing pipeline: Images are JPEG compressed after being modified to simulate a realistic scenario. This is very important since fake content is especially harmful when it is uploaded on the internet and shared with a malicious goal to propagate false information.

4. Proposed Method

Let $\mathcal{R}_M$ be the set of real images generated by the target camera model M, and $\mathcal{S}$ a set of synthetic images generated by some software tools. We aim to process the images in $\mathcal{S}$ so as to make them forensically indistinguishable[1] from images in $\mathcal{R}_M$. That is, we want to find a transformation $T_M(\cdot)$ such that, with high probability,

$$F_i(T_M(x)) = M, \qquad x \in \mathcal{S}, \;\; i = 1, \dots, N$$

with F_i's the available camera model classifiers. Since state-of-the-art forensic classifiers focus on the traces left on all acquired images by the model-specific processing suite, such traces must be injected into the original image. We pursue this goal by training a convolutional neural network on images drawn from $\mathcal{R}_M$, such that eventually, for each $x \in \mathcal{S}, y = T_M(x)$ looks to classifiers as belonging to $\mathcal{R}_M$. Meanwhile, to remain undetected, the attack is required not to modify the semantic content of the image, and hence to minimize some suitable measure of distortion $d(x, y)$ between original and modified images.

4.1. Architecture

Our goal is to be agnostic to any signature-based camera model identifier; i.e., we want more than inserting adversarial noise to fool a specific identifier. To this end, we train a CNN through a reformulated GAN schema. Specifically, our proposal involves the use of three networks: a generator, a discriminator and an embedder. These are described in the following, and depicted in Fig.3.

Generator: The Generator, $G(\cdot) = T_M(\cdot)$, has the goal of introducing traces of a specific camera model M into the input image, while preserving the semantic content. We adopt an architecture formed by 7 convolutional layers. The output of the last layer is summed to the input image and then a hyperbolic tangent is applied to limit the values of the output in the same range used for the input images, the range is equal to $[-1, 1]$. Note that, before entering the generator, the input image is Gaussian filtered with $\sigma = 0.4$ to remove possible high-frequency imperfections, such as checkerboard artifacts.

Discriminator: In conventional GANs, the discriminator, $D(\cdot)$, is required to distinguish real from generated images, with the aim of pushing the generator to improve over time. Accordingly, we could ask the discriminator to separate the sets $\mathcal{R}_M$, real images of the target camera model, and $G(\mathcal{S})$, modified synthetic images. Instead, we train it to tell apart $\mathcal{R}_M$ from both $G(\mathcal{S})$ and $\mathcal{S}$, the set of original synthetic images. By doing so, the generator is encouraged not only to introduce traces of the target camera model, but also to remove traces peculiar of synthetic images. We use a patch-based discriminator [26] realized as a fully-convolutional network with a fixed first layer. The fixed layer takes as input the RGB image and returns 9 channels, which are the

[1] Of course, the synthetic images should also be visually realistic, but we do not address this problem, here, and assume the generation tool produces already visually plausible images.

original RGB components and their third-order horizontal and vertical derivatives. In practice, this first layer extracts the image residuals, which highlight discriminating camera traces, speeding-up the network convergence [52].

Embedder: In addition to a generator and discriminator, we use an embedder, $E(\cdot)$. This network is pre-trained offline, using images drawn from many camera models, to extract a compact 512-dimensional feature vector. It is trained using a triplet-loss [48] with the goal of obtaining the same feature vector for all images acquired by the same camera model. Therefore, it provides a compact, model-specific, representation of the image, independent of the current data. For the embedder, we use the same fixed first layer used for the discriminator.

4.2. Loss function of the generator

The goal of our scheme is to train an effective generator. To this end, we define its objective function as the sum of three losses,

$$\mathcal{L}_G = \mathcal{L}_{CNT} + \lambda_E \mathcal{L}_{EMB} + \lambda_A \mathcal{L}_{ADV} \tag{1}$$

each of which drives the generator towards a specific goal: preserving the scene content ($\mathcal{L}_{CNT}$), fooling the embedder ($\mathcal{L}_{EMB}$) and, fooling the discriminator ($\mathcal{L}_{ADV}$). These losses are detailed in the following.

Scene content preserving loss: To achieve the first goal, we use a combination of an objective distortion measure as well as a perceptual loss between input and the output of the generator:

$$\mathcal{L}_{CNT} = \mathcal{L}_{REC} + \lambda_p \mathcal{L}_{PER} \tag{2}$$

As distortion measure we use an L1 distance between the two images, $\mathcal{L}_{REC} = \mathbb{E}_{x\sim\mathcal{S}}\left[\|x - G(x)\|_1\right]$. Following [28], we define the perceptual loss $\mathcal{L}_{PER}$ as the sum of the L1 distances between the feature maps extracted by the VGG-19 network trained on ImageNet.

Embedder Loss: The second goal of the generator is to fool the embedder, namely, to ensure that feature vectors extracted from the transformed images are indistinguishable from those of real images of the target model. To this end, by averaging the feature vectors of real images, we first compute an anchor vector, $e_M = \mathbb{E}_{z\sim\mathcal{R}_M}\left[E(z)\right]$, representing the camera model in the embedding space. Then we should aim at minimizing the distance, $d(x) = \|E(G(x)) - e_M\|_1$, between feature vectors extracted from transformed images and this anchor vector. However, from the literature on triplet loss, it is well known that better results are obtained by comparing distances [48]. Therefore, we first define a reference distance $d_{ref} = \mathbb{E}_{z\sim\mathcal{R}_M}\left[\|E(z) - e_M)\|_1\right]$, and then define the loss to minimize as

$$\mathcal{L}_{DST} = \mathbb{E}_{x\sim\mathcal{S}}\left[\,|d(x) - d_{ref} + m|_+\right] \tag{3}$$

where $|x|_+ = x$ for $x > 0$ and 0 otherwise, and m is the margin of the triplet-loss fixed to 0.01 in our experiments. To further help the generator to fool the embedder, we include also a feature matching loss, $\mathcal{L}_{FM}$, proposed in literature [55] to stabilize the training of GANs, which we compute based on the feature maps extracted by the embedder of the real and the transformed images. The final embedding loss is then

$$\mathcal{L}_{EMB} = \mathcal{L}_{DST} + \lambda_f \mathcal{L}_{FM} \tag{4}$$

Adversarial Loss: The discriminator, trained in parallel with the generator, should output values close to 1 for real images, $x \in \mathcal{R}_M$, and close to zero for synthetic images, $x \in S$, both before and after being modified. Accordingly, it relies on a modified binary cross-entropy loss:

$$\begin{aligned}\mathcal{L}_D = &-\mathbb{E}_{x\sim\mathcal{R}_M}\left[\log D(x)\right] + \\ &-\tfrac{1}{2}\mathbb{E}_{x\sim\mathcal{S}}\left[\log(1 - D(G(x))) + \log(1 - D(x))\right]\end{aligned} \tag{5}$$

which pools original and modified synthetic images.

A major goal of our generator is to modify the synthetic images to fool the discriminator, i.e., to make the discriminator believe they come from the target camera model. Therefore, for the last term of Eq.(1) we adopt the standard adversarial loss based on the binary cross-entropy:

$$\mathcal{L}_{ADV} = -\mathbb{E}_{x\sim\mathcal{S}}\left[\log D(G(x))\right] \tag{6}$$

which is minimized when $D(G(x)) \to 1$, that is, synthetic images are classified as real after being modified. Thus, generator and discriminator concur to modify the synthetic images to be similar to images of the target camera model but also different from the original synthetic images.

5. Results

In this section we present the results of our experiments. Specifically, we evaluate the proposed method in terms of its ability to (a) deceive detectors of GAN-generated images into believing they are dealing with real images (see Sec. 5.2) and, (b) deceive camera model identifiers into recognizing the modified images as acquired by the target camera model (see Sec. 5.1). Special attention will be devoted to testing robustness to off-training GANs. To this end, experiments will be carried out on images generated by GAN architectures never seen in the training phase.

We consider a dataset of real images acquired by different camera models and a dataset of synthetic images generated by various GAN architectures. For the real images, we use the 10 camera models adopted in the IEEE Forensic Camera Model Identification Challenge, 400 images per model are used for training, and 50 for testing. From the test set we sample 25000-patches to test the camera model identifiers. For the synthetic images, seven GAN architectures are considered: StarGAN [12], CycleGAN [61], ProGAN [29], StyleGAN [30], RelGAN [57], bigGAN [5], and StyleGAN2 [31]. For each of the first five architectures, we take 20000 images for training and 2000 for testing. In addition, 2000 bigGAN images and 2000 StyleGAN2 images are only used for test, but not in training, in order to evaluate generalization. All experiments are carried out on 256 × 256-pixel patches. For both real images and high-resolution GAN images (generated by ProGAN, StyleGAN and StyleGAN2) patches are extracted at random locations from the whole image.

For each of the 10 camera models, a different generator-discriminator couple is trained, using only real images of the selected model. We use ADAM optimizers with a batch size of 10 and 30 patches, respectively. For both networks, the learning rate is set to 10^{-4}, and the ADAM moments to 0.5 and, 0.999. Through preliminary experiments, the loss weights are set to $\lambda_E = 1$, $\lambda_A = 0.1$, $\lambda_p = 0.001$, and $\lambda_f = 0.01$. Training stops after $50K$ iterations.

The embedder is trained in advance using an external dataset of 600 camera models and a total of 20394 images publicly available on dpreviewer.com. We also adopt ADAM for the embedder, with a batch of 80 patches, a learning rate of 10^{-4} and default moments. It is worth underlining that this dataset is different from the one used to train camera model identification methods, hence our approach does not assume a prior knowledge on the camera models used for the training step of the analyzed camera identification algorithms.

5.1. Fooling camera model ID detectors

In this section we analyze the ability of our method to deceive camera model identifiers. To this end, we consider four CNN-based target classifiers. The first two architectures, Tuama2016 [52], and Bondi2017 [4], have been proposed specifically for camera model identification. Moreover, in view of the results of the IEEE Forensic Camera Model Identification Challenge hosted on the Kaggle platform[2] we also consider two deep general-purpose architectures, Xception [13] and InceptionV3 [50], trained to work as camera model classifiers. The performance of all these classifiers on our 10-models test set, in the absence of any attack, is shown in Tab. 1. Although all methods perform well, the very deep networks clearly outperform the other classifiers. In general, most successful solutions are based on very deep networks [50, 25, 13], that perform well even when data are subjected to common post-processing operations, like re-compression.

Tuama2016	Bondi2017	Xception	InceptionV3
74.1	87.1	97.0	95.4

Table 1: Accuracy (%) of camera model identification evaluated on 25000 patches of dimension equal to 256 × 256 pixels.

		Model Classifiers			
Method	PSNR	Tuama 2016	Bondi 2017	Xcep.	Incep. V3
Proposal	31.7	**53.3**	**67.6**	**81.4**	**71.7**
only-embed.	49.7	48.1	6.76	10.8	13.2
only-discr.	31.2	47.9	63.8	66.7	70.4

Table 2: Averaged results on 50000 images attacked using 5 different camera models. Performance is measured in terms of Successful Attack Rate (SAR).

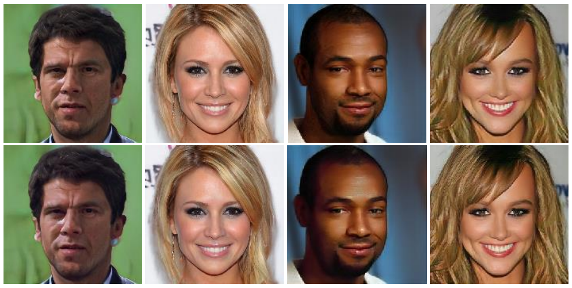

Figure 4: First line: original images from StyleGAN [30] and RelGAN [57]. Second line: attacked images with our method.

5.1.1 Ablation study

To show the importance of the embedder and the discriminator in the training scheme, we compare two variants. In the first variant, *only-embed.*, we remove the discriminator by setting $\lambda_D = 0$ for the loss of the generator (Equation 1 of the main paper). While in the second variant, *only-disc.*, we remove the embedder by setting $\lambda_E = 0$ for the loss of the generator. The other hyperparameters are not modified. Performance is measured in terms of Successful Attack Rate (SAR), the fraction of modified synthetic images that are classified as acquired by the target model. Tab. 2 shows the performance of the variants to deceive four camera-model classifiers. As can be seen this supports our choices. The *only-embed.* variant obtains the worst results with the maximum SAR of 48.13% for Tuama2016 and a SAR lower than 15% for the other classifiers, while the *only-discr.* variant performs worse for all the four camera-model classifiers.

Finally, in Fig. 4 we show some examples of attacked images, where no clear visual artifacts can be spotted.

[2] https://www.kaggle.com/c/sp-society-camera-model-identification/

			Model Classifiers			
	Method	PSNR	Tuama 2016	Bondi 2017	Xcep.	Inc. V3
Bondi2017	PGD	30.98	1.0	96.8	0.0	0.8
	TI-MI-FGSM	31.38	31.2	31.7	1.4	2.0
	GAP	31.69	20.7	67.4	3.1	22.4
	Adv-Cam-Id	30.18	24.1	89.5	54.5	58.8
Xception	PGD	33.52	24.7	6.7	98.2	0.2
	TI-MI-FGSM	30.77	35.1	3.2	87.9	6.5
	GAP	32.09	5.4	0.6	96.4	5.5
	Adv-Cam-Id	29.69	37.6	37.2	97.5	43.3
Ensemble	PGD	32.70	20.3	9.2	1.2	0.5
	TI-MI-FGSM	31.07	30.0	3.2	5.2	5.5
	GAP	31.97	25.5	4.3	12.1	6.8
	SpoC (ours)	31.41	**55.6**	**64.7**	**73.4**	**69.3**

Table 3: Averaging results on 50000 images attacked using 5 different camera models. Performance is measured in terms of a Successful Attack Rate (SAR). We compare with white-box attacks on Bondi2017 (first block) and Xception (second block) and hence discard values on such architecture for the analysis (red). We also compare with methods using an ensemble of classifiers (third block).

			Model Classifiers			
	Method	PSNR	Tuama 2016	Bondi 2017	Xcep.	Inc. V3
Bondi2017	PGD	32.33	12.2	97.5	3.5	13.5
	TI-MI-FGSM	31.29	34.1	27.9	2.1	5.1
	GAP	31.55	16.3	59.4	4.0	22.5
	Adv-Cam-Id	24.63	23.5	71.8	33.6	39.8
Xception	PGD	33.11	20.2	5.6	98.5	0.2
	TI-MI-FGSM	30.56	37.1	4.0	93.3	11.0
	GAP	31.88	3.6	1.2	90.8	2.5
	Adv-Cam-Id	28.79	34.7	40.1	98.0	40.3
Ensemble	PGD	32.39	18.4	4.4	1.6	0.2
	TI-MI-FGSM	30.93	35.0	2.1	6.4	9.4
	GAP	31.90	24.8	2.1	20.4	5.3
	SpoC (ours)	30.37	**56.8**	**48.4**	**74.2**	**66.0**

Table 4: Averaging results on 20000 images attacked using 5 different camera models and two GAN architectures ([5], [31]) outside the training set. Performance is measured in terms of a Successful Attack Rate (SAR). We compare with white-box attacks on Bondi2017 (first block) and Xception (second block) and hence discard values on such architecture for the analysis (red). We also compare with methods using an ensemble of classifiers (third block).

5.1.2 Comparison with state-of-the-art

We will compare the results of our proposal with four baseline methods for the generation of adversarial attacks: Projected Gradient Descent (PGD)[3] [37], Translation-Invariant Momentum Iterative Fast Gradient Sign Method (TI-MI-FGSM) [17], Generative Adversarial Perturbation (GAP)[4] [45], and Generative Adversarial Attack against Camera Identification (Adv-Cam-Id) [10]. Only the latter method was specifically developed to attack camera model classifiers, the other ones are more general and powerful approaches that insert adversarial noise to fool machine learning based detectors. Adversarial noise in all these cases was added to make the image recognized as if it was taken by a specific camera model.

All these methods perform white-box attacks, i.e., they are trained with reference to a known target classifier. Here, we will consider both Bondi2017 and Xception as target classifiers. However, to assess the transferability of the attack, we will evaluate performance also on all off-training classifiers. To improve transferability, we also perform training an ensemble of classifiers, as proposed in [17]; performance is evaluated on the fourth one. For a fair comparison, we set the parameters of all methods in order to obtain a PSNR of ≈31dB. We refer to the original papers for further details.

Results are computed on the central 256 × 256 crop of 10000 synthetic images, and averaged over 5 target models (Motorola DroidMaxx, Samsung GalaxyS4, Sony Nex7, iPhone 4s and, Motorola X). All images are JPEG compressed using the quantization table of the target camera model. This corresponds to a realistic setting where images are compressed before being spread over the web. Note that in the absence of any attack, the average SAR is 10% for our 10-class setting. Results are reported in Tab. 3.

Several considerations are in order. First of all, it is clear that all white-box attacks are very effective when tested on the very same classifier used during training, Bondi2017 in the first block, Xception in the second one. Such results are emphasized with red text. However, there is no reason to expect such a scenario in practice, as the defender is free to use any classifier for camera identification. Therefore, the most interesting results are those in black. On off-training classifiers, only Adv-Cam-Id, among the baselines, provides a reasonably good performance, never exceeding 60% though. Instead, SpoC performs quite well on all classifiers, with SAR going from about 56% to 73%. With respect to Adv-Cam-Id, the best baseline, it improves from 10% to 30%. Training on an ensemble of classifiers should ensure higher transferability, but the results of the third block do not seem especially encouraging, with an average SAR barely exceeding 10%.

Although, in Tab. 3, we show results on 256 × 256 crops, our method can also be applied to higher resolution images (e.g., 1024 × 1024) without re-training the network. Indeed,

[3] https://github.com/BorealisAI/advertorch

[4] https://github.com/OmidPoursaeed/Generative_Adversarial_Perturbations

	in training					out training				
	Xception	Spec	ResNet50	Patch Forensics	FFD	Xception	Spec	ResNet50	Patch Forensics	FFD
before our attack	99.87	76.06	99.38	78.90	97.83	82.12	22.02	80.55	22.77	75.94
after our attack	44.28	8.45	45.69	3.65	27.19	1.00	0.96	5.24	0.11	4.74

Table 5: True Positive Rate (TPR) for the GAN detectors before and after the proposed attack using GAN architectures considering both images inside (StarGAN [12], CycleGAN [61], ProGAN [29], StyleGAN [30], and RelGAN [57]) and outside the training-set (bigGAN [5], and StyleGAN2 [31]).

if we apply our network to the StyleGAN images at resolution 1024×1024, we still achieve a good result with an attack success rate always above 48%.

5.1.3 Generalization

Both the proposed technique and the reference techniques based on a generative network require a training phase. To test their ability to generalize, we add a further experiment, on images generated by a GAN architecture not used in the training set. In Table 4 we show the results.

GAP shows about the similar performance, measured in terms of SAR and PSNR, on data generated by architectures outside and inside the training-set. On the contrary Adv-Cam-Id presents a reduction of PSNR (until 5dB), while preserving a reasonably good SAR. Finally, comparing the SpoC with respect to the other methods, it continues to have the better results, with SAR going from 48.4% to 74.2%, with a minimal reduction in PSNR.

5.2. Fooling GAN-image detectors

Here, we show that our architecture largely succeeds in removing the peculiar features of GAN images that make them distinguishable from real images. It is important to highlight that we do not re-train our model. To this end, we challenge five CNN-based GAN-image detectors. The first one is the spectrum-based (Spec) classifier proposed in [60], which detects the frequency-domain peaks caused by the up-sampling steps of common GAN pipelines. Moreover, we consider a two general-purpose deep networks, Xception [13] and ResNet50 [23], which proved to be an effective tool for GAN image detection and deepfakes video [38, 47, 54]. We also include PatchForensics that is a fully-convolutional patch-based classifier proposed in [8] and FFD (Facial Forgery Detection) [15], a variant of Xception, that includes an attention-based layer, in order to focus on high-frequency details. All the detectors are trained on 10000 synthetic images coming from 5 GAN architectures and 4000 real images coming from 10 cameras. We used the augmentation strategy proposed in [54], that helped to increase the generalization ability of each detector. Testing is carried out on 256×256-pixel central crop of 10000 synthetic images of seen GAN architectures and 4000 synthetic images of two unseen GAN architectures. In Tab. 5 (left), we show the results of the experiment with aligned training and test. The detectors have a high true positive rate (TPR), sometimes close to 100%, that is, they detect GAN images with near certainty. They have also good performance on real images with a false positive rate (FPR), not shown in the table, always less than 1%. However, after modifying the synthetic images with our attack, the TPRs decrease to 3.65% and, 45.69%. In a second experiment, we work on the off-training images generated by the BigGAN and StyleGAN2 architecture. Results are reported in Tab. 5 (right). Some detectors (Xception, ResNet50 and FFD) keeps detecting GAN images with high accuracy, nonetheless, after modifying the images with our approach, the TPR reduces drastically less than 6%.

6. Conclusion

In this work, we proposed a GAN-based method to attack both forensic camera model identifiers and GAN detectors. Our scheme allows to inject model-specific traces into the input image such to deceive the classifier into believing the image was acquired by the desired target camera model. Moreover, the method requires no prior knowledge on the attacked network, and works even on completely synthesized images. Experimental results prove the effectiveness of the proposed method and, as a consequence, the weaknesses of current forensic detectors, calling for new approaches, less dependant on critical hypotheses, and more resilient to unforeseen attacks.

Acknowledgement. We gratefully acknowledge the support of this research by the AI Foundation, a TUM-IAS Hans Fischer Senior Fellowship, and Google Faculty Research Award. In addition, this material is based on research sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL) under agreement number FA8750-20-2-1004. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA and AFRL or the U.S. Government. This work is also supported by the PREMIER project, funded by the Italian Ministry of Education, University, and Research within the PRIN 2017 program.

References

[1] Shruti Agarwal and Hany Farid. Photo Forensics from JPEG Dimples. In *IEEE International Workshop on Information Forensics and Security*, 2017. 2

[2] Mauro Barni, Kassem Kallas, Ehsan Nowroozi, and Benedetta Tondi. On the Transferability of Adversarial Examples against CNN-based Image Forensics. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018. 3

[3] Mauro Barni, Matthew Stamm, and Benedetta Tondi. Adversarial multimedia forensics: Overview and challenges ahead. In *European Signal Processing Conference (EUSIPCO)*, pages 962–966, 2018. 2

[4] Luca Bondi, Luca Baroffio, David Güera, Paolo Bestagini, Edward Delp, and Stefano Tubaro. First steps toward camera model identification with convolutional neural networks. *IEEE Signal Processing Letters*, 24(3):259–263, 2017. 6

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis, 2018. 6, 7, 8

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. 1

[7] Nicholas Carlini and Hany Farid. Evading deepfake-image detectors with white- and black-box attacks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2804–2813, 2020. 2, 3

[8] Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. What makes fake images detectable? understanding properties that generalize. In *European Conference on Computer Vision (ECCV)*, pages 103–120, 2020. 8

[9] Chang Chen, Zhiwei Xiong, Xiaoming Liu, and Feng Wu. Camera trace erasing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2947–2956, 2020. 3

[10] Chen Chen, Xinwei Zhao, and Matthew C. Stamm. Generative adversarial attacks against deep-learning-based camera model identification. *IEEE Trans. Inf. Forensics Security, in press*, October 2019. 2, 3, 7

[11] Mo Chen, Jessica Fridrich, Miroslav Goljan, and Jan Lukàš. Determining image origin and integrity using sensor noise. *IEEE Trans. Inf. Forensics Security*, 3:74–90, 2008. 1

[12] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 6, 8

[13] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6, 8

[14] Davide Cozzolino and Luisa Verdoliva. Noiseprint: a CNN-based camera model fingerprint. *IEEE Trans. Inf. Forensics Security*, 15:144–159, 2020. 2

[15] Hao Dang, Feng Liu, Joel Stehouwer, Xiaoming Liu, and Anil K Jain. On the detection of digital face manipulation. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5781–5790, 2020. 8

[16] Debayan Deb, Jianbang Zhang, and Anil K. Jain. Advfaces: Adversarial face synthesis. In *IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–10, 2020. 3

[17] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 7

[18] Pasquale Ferrara, Tiziano Bianchi, Alessia De Rosa, and Alessandro Piva. Image forgery localization via fine-grained analysis of CFA artifacts. *IEEE Trans. Inf. Forensics Security*, 7(5):1566–1577, 2012. 2

[19] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 2, 3

[20] Diego Gragnaniello, Davide Cozzolino, Francesco Marra, Giovanni Poggi, and Luisa Verdoliva. Are GAN generated images easy to detect? A critical analysis of the state-of-the-art. In *IEEE International Conference on Multimedia and Expo (ICME)*, 2021. 3

[21] Diego Gragnaniello, Francesco Marra, Giovanni Poggi, and Luisa Verdoliva. Analysis of adversarial attacks against CNN-based image forgery detectors. In *European Signal Processing Conference*, pages 384–389, 2018. 3

[22] David Güera, Yu Wang, Luca Bondi, Paolo Bestagini, Stefano Tubaro, and Edward Delp. A Counter-Forensic Method for CNN-Based Camera Model Identification. In *IEEE CVPR Workshops*, 2017. 2

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 8

[24] Weiwei Hu and Ying Tan. Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN. *arXiv preprint arXiv:1702.05983v1*, 2019. 3

[25] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6

[26] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4

[27] Zhiwei Jin, Juan Cao, Yongdong Zhang, Jianshe Zhou, and Qi Tian. Novel visual and statistical image features for microblogs news verification. *IEEE Trans. on Multimedia*, 19(3):598–608, 2017. 1

[28] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711, 2016. 5

[29] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *International Conference on Learning Representations*, 2018. 1, 6, 8

[30] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4401–4410, 2019. 1, 6, 8

[31] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *CVPR*, pages 8110–8119, 2020. 1, 6, 7, 8

[32] Matthias Kirchner and Rainer Böhme. Synthesis of color filter array pattern in digital images. In *Media Forensics and Security*, 2009. 2

[33] Matthias Kirchner and Thomas Gloe. Forensic camera model identification. In T.S. Ho and S. Li, editors, *Handbook of Digital Forensics of Multimedia Data and Devices*, pages 329–374. Wiley-IEEE Press, 2015. 2

[34] Yanpei Liu, Xinyun Chen, Shanghai Jiao Tong, Chang Liu, and Dawn Song. Delving into tranferable adversarial examples and black-box attacks. In *International Conference on Learning Representations*, 2017. 3

[35] Jan Lukàš, Jessica Fridrich, and Miroslav Goljan. Digital camera identification from sensor pattern noise. *IEEE Trans. Inf. Forensics Security*, 1(2):205–214, 2006. 2

[36] 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. 2

[37] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 7

[38] Francesco Marra, Diego Gragnaniello, Giovanni Poggi, and Luisa Verdoliva. Detection of GAN-Generated Fake Images over Social Networks. In *IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 384–389, 2018. 8

[39] Francesco Marra, Diego Gragnaniello, and Luisa Verdoliva. On the vulnerability of deep learning to adversarial attacks for camera model identification. *Signal Processing: Image Communication*, 65, 2018. 2

[40] Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. Do GANs leave artificial fingerprints? In *IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 506–511, 2019. 3

[41] Falko Matern, Christian Riess, and Mark Stamminger. Exploiting visual artifacts to expose deepfakes and face manipulations. In *IEEE WACV Workshop on Image and Video Forensics*, 2019. 1

[42] Ryota Natsume, Tatsuya Yatagawa, and Shigeo Morishim. RSGAN: Face Swapping and Editing using Face and Hair Representation in Latent Spaces. In *ACM SIGGRAPH*, 2018. 1

[43] Joo C. Neves, Ruben Tolosana, Ruben Vera-Rodriguez, Vasco Lopes, Hugo Proena, and Julian Fierrez. Ganprintr: Improved fakes and evaluation of the state of the art in face manipulation detection. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):1038–1048, 2020. 2, 3

[44] Yuval Nirkin, Yosi Keller, and Tal Hassner. FSGAN: Subject Agnostic Face Swapping and Reenactment. In *IEEE International Conference on Computer Vision*, Oct. 2019. 1

[45] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative Adversarial Perturbations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4422–4431, 2018. 3, 7

[46] Shengju Qian, Kwan-Yee Lin, Wayne Wu, Yangxiaokang Liu, Quan Wang, Fumin Shen, Chen Qian, and Ran He. Make a Face: Towards Arbitrary High Fidelity Face Manipulation. In *IEEE International Conference on Computer Vision*, 2019. 1

[47] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics++: Learning to Detect Manipulated Facial Images. In *International Conference on Computer Vision (ICCV)*, 2019. 8

[48] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 5

[49] Yang Song, Rui Shu, Nate Kushman, and Stefano Ermon. Constructing Unrestricted Adversarial Examples with Generative Models. In *Conference on Neural Information Processing Systems (NIPS)*, 2018. 3

[50] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[51] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2Face: Real-Time Face Capture and Reenactment of RGB Videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, June 2016. 1

[52] Amel Tuama, Frédéric Comby, and Marc Chaumont. Camera model identification with the use of deep convolutional neural networks. In *IEEE Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2016. 5, 6

[53] Luisa Verdoliva. Media forensics and deepfakes: An overview. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):910–932, 2020. 1

[54] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. CNN-generated images are surprisingly easy to spot... for now. In *CVPR*, 2020. 8

[55] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional GANs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 5

[56] Xiaosen Wang, Kun He, and John E. Hopcroft. AT-GAN: A Generative Attack Model for Adversarial Transferring on Generative Adversarial Nets. *arXiv preprint arXiv:1904.07793v3*, 2019. 3

[57] Po-Wei Wu, Yu-Jing Lin, Che-Han Chang, Edward Y. Chang, and Shih-Wei Liao. Relgan: Multi-domain image-to-image translation via relative attributes. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 6, 8

[58] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating Adversarial Examples with Adversarial Networks. In *International Joint Conference on Artificial Intelligence*, 2018. 3

[59] Xin Yang, Yuezun Li, Honggang Qi, and Siwei Lyu. Exposing GAN-synthesized Faces using Landmark Locations. In *ACM Workshop on Information Hiding and Multimedia Security*, pages 113–118, 2019. 1

[60] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. Detecting and Simulating Artifacts in GAN Fake Images. In *IEEE Workshop on Information Forensics and Security (WIFS)*, 2019. 3, 8

[61] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 6, 8