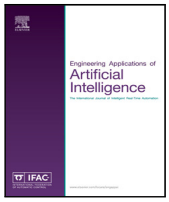
Engineering Applications of Artificial Intelligence 124 (2023) 106594

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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



Black-box attack against GAN-generated image detector with contrastive perturbation

Zijie Lou, Gang Cao *, Man Lin

State Key Laboratory of Media Convergence and Communication, Communication University of China, Beijing 100024, China
School of Computer and Cyber Sciences, Communication University of China, Beijing 100024, China

ARTICLE INFO

Keywords:
Black-box attack
Anti-forensic
GAN-generated image detector
GAN fingerprint
Encoder–decoder network
Contrastive perturbation

ABSTRACT

The emergence of visually realistic GAN-generated facial images has raised concerns regarding potential misuse. In response, effective forensic algorithms have been developed to detect such synthetic images in recent years. However, the vulnerability of such forensic detectors to adversarial attacks remains an important issue that requires further investigation. In this paper, we propose a new black-box attack method against GAN-generated image detectors. It involves contrastive learning strategy to train an encoder–decoder anti-forensic network with a contrastive loss function. GAN-generated and corresponding simulated real images are constructed as positive and negative samples, respectively. By leveraging the trained attack model, we can apply imperceptible perturbation to input synthetic images for removing GAN fingerprint to some extent. GAN-generated image detectors may be deceived consequently. Extensive experimental results demonstrate that the proposed attack effectively reduces the accuracy of three state-of-the-art detectors on six popular GANs, while also achieving high visual quality of the attacked images. The source code will be available at https://github.com/ZXMMD/BAttGAND.

1. Introduction

Nowadays, it becomes increasingly easy to manipulate the face of a real person in an image or even to automatically synthesize non-existent faces. This is largely due to the remarkable progress made in deep learning technologies, especially Generative Adversarial Networks (GAN) (Goodfellow et al., 2020), which have facilitated the creation of synthetic content that is quite realistic. Synthesizing fake facial images has become relatively easy with the availability of accessible open software and mobile applications, such as FaceApp (FaceApp, 2017). Many different GANs, such as ProGAN (Karras et al., 2017), StarGAN (Choi et al., 2018a), StarGAN2 (Choi et al., 2020), StyleGAN (Karras et al., 2019), StyleGAN2 (Karras et al., 2020), and StyleGAN3 (Karras et al., 2021a), have been proposed in recent years. Such GANs are capable of generating extremely realistic facial images, which raise concerns about the potential misuse of such images for malicious purposes. These include identity theft for fraudulent activities and posing a threat to social security.

To counter the potential threat posed by GANs, numerous forensic methods have been proposed to detect GAN-generated facial images. Some of such approaches rely on specific facial traces, such as differences in iris color (Matern et al., 2019), which were left behind by early GAN architectures. For instance, some face images generated by ProGAN exhibit differences in the color of the left and right eye. Most recent GAN-generated image detection methods rely on deep learning and significantly outperform the handcraft feature-based ones. Marra et al. (2018) demonstrate that the off-the-shelf deep neural networks, such as Xception (Chollet, 2017), Inception (Szegedy et al., 2016) and DenseNet (Iandola et al., 2014), could achieve excellent detection performance after being pre-trained on ImageNet and trained on GAN-generated and real images. Sheng-Yu et al. (2020) propose an effective GAN-generated image detector based on ResNet50 (He et al., 2016) backbone network. This approach applies strong sample enhancement techniques, such as compression and blur, during model training to improve the detector's generalization ability and robustness. In Gragnaniello et al. (2021), the generalization performance of GAN-generated image detectors is further boosted by inserting an initial residual layer and removing the downsampling in the first layer. It is significant to evaluate the security and reliability of such GAN-generated image detectors in real-world applications, where malicious attacks are likely to occur.

In recent years, adversarial sample techniques have caused a new threat to GAN-generated image detectors, which may be deceived or suffer performance degradation due to anti-forensic attacks (Na et al., 2022; Carlini and Farid, 2020; Szegedy et al., 2013; Papernot et al., 2016; Zhao and Stamm, 2021; Xie et al., 2022; Huang et al., 2020; Neves et al., 2020). Na et al. (2022) employ GAN with Limited Queries

* Corresponding author at: State Key Laboratory of Media Convergence and Communication, Communication University of China, Beijing 100024, China.
E-mail address: gangcao@cuc.edu.cn (G. Cao).

https://doi.org/10.1016/j.engappai.2023.106594
Received 6 November 2022; Received in revised form 29 March 2023; Accepted 2 June 2023
Available online xxxx
0952-1976/

to generate unrestricted adversarial examples to deceive classification models. Specifically, they manipulate the latent vectors by accessing the top-1 final decision of a classification model to deceive it. Carlini and Farid (2020) propose generating adversarial samples using Box-constrained L-BFGS (Szegedy et al., 2013) or JSMA (Papernot et al., 2016) to attack GAN-generated image detectors in the white-box scenario. Similarly, Zhao and Stamm (2021) achieve a white-box attack by synthesizing forensic traces associated with real images through an anti-forensic generator. However, such attack methods require full knowledge of the detector, including network structure and internal parameters, which are almost inaccessible in real-world applications. Consequently, a few black-box attack methods have also been proposed. Xie et al. (2022) present an end-to-end deep dithering model that eliminates generative artifacts in various GAN-generated images. The attacked images are produced by adding dithering noise to GAN-generated images, rather than generating them entirely from scratch. The FakePolisher method (Huang et al., 2020) achieves shallow reconstruction of fake images by a learned linear dictionary, which could reduce the artifacts introduced during image synthesis to some extent. Neves et al. (2020) train an anti-forensic generator in one-class mode, where only real images are used during the training phase. Inherent characteristics of such real images are captured by an autoencoder and then injected into the GAN-generated images. Although this attack method has the advantage of training without GAN-generated samples and yielding high-quality attacked images, the success rate of attacking still requires improvement. Furthermore, since the target of the attack is GAN-generated images, it might be better to use both real and fake image samples with the same visual content under two-class supervised training.

To enhance the performance and applicability, in this paper, we propose a novel black-box attack against GAN-generated image detectors. Contrastive perturbation is learned by deep supervised training on a lightweight yet effective encoder–decoder network with GAN-generated images and their corresponding simulated real counterparts. Following the completion of training, the GAN fingerprint can be removed by introducing this perturbation to input GAN-generated images. Notably, our attack method requires only access to the input–output of a forensic detector, rather than complete information. Specifically, we perform supervised training by constructing pairs of GAN-generated and simulated real images, which enjoy the same visual appearance. In our study, we view the simulated real image as a label for the input GAN-generated image. To train the anti-forensic generator, we aim to make the output as close as possible to this label while simultaneously moving away from the input GAN-generated image.

The rest of this paper is organized as follows. The proposed black-box attack scheme is presented detailedly in Section 2, followed by extensive experimental results and discussions in Section 3. We draw the conclusions in Section 4.

2. Proposed black-box attack

In this section, we provide a comprehensive description of our proposed attack strategy against GAN-generated image detectors. Our attack scheme aims to fulfill two primary requirements for a successful attack: firstly, the attacked image should have a high probability of evading GAN-generated image detectors, and secondly, it should maintain the same visual appearance as the corresponding GAN-generated image. Since GAN-generated image detectors generally work by capturing generation fingerprint, GAN-generated images will be attacked by removing their inherent GAN fingerprint via an autoencoder network. Such a network is effectively trained by our proposed GAN-generated image-orient contrastive perturbation method. To construct a novel training sample set, we carefully create numerous pairs of GAN-generated and simulated real images.

2.1. Attack via encoder-decoder network

An overview of the proposed black-box attack model is illustrated in Fig. 1. As inspired by the prior work (Neves et al., 2020), the attack is implemented by a lightweight yet effective encoder–decoder network, which includes an encoder E followed by a decoder D as

$$E : I \rightarrow Y, \quad D : Y \rightarrow I^A. \tag{1}$$

Here, the latent feature Y is extracted from an input GAN-generated image I by the encoder E. Then the attacked image I^A is reconstructed from such a latent feature by the decoder D.

The meaning of each layer of the network is shown in the dotted box in the bottom of Fig. 1. Different colors represent different operating layers. The number in the color block representing the convolutional layer indicates the number of convolution kernels. In the encoder E, four convolutional layers each followed by ReLU activations (Agarap, 2018) are used to extract features from $I \in \mathbb{R}^{3\times H\times W}$, where $H \times W$ denotes the spatial dimension. The fourth convolutional layer is a dilated convolutional layer, and the size of all the convolution kernels is 3×3. To reduce the feature map size, we employ three max-pooling layers progressively, resulting in a feature map size of $C \times 28 \times 28$, where the channel number C of the latent feature Y is set to 32 by default, and the pooling stride is set to 2×2. Drawing inspiration from previous seminal networks (Zamir et al., 2021), we utilize a multi-stage stacking structure for the en/decoder modules. This structure aids in extracting a pyramidal structure of hierarchical multi-scale features, which are valuable in reconstructing input images.

The decoder D is responsible for progressively recovering a high-resolution representation of reconstructed images. A dilated transposed convolutional layer is employed firstly to restore the channel number of the feature map to 128. Three max-unpooling layers are then utilized to gradually increase the size of the feature maps, resulting in a three-channel color image, with the pooling stride set to 2×2. To avoid the checkerboard artifacts (Odena et al., 2016) that can arise from transposed convolution, a convolutional layer followed by a max-unpooling layer is used to upsample the input feature maps in the first three transposed convolution layers. In the final stage, only a transposed convolutional layer is necessary to generate an attacked image with the same size as the input, and the size of all the transposed convolution kernels set to 3×3.

2.2. Contrastive perturbation training

In this subsection, we propose a contrastive perturbation-based method to train the encoder–decoder network. In terms of intrinsic attributes, the attacked image I^A is expected to approach a real image and keep away from the input GAN-generated image I. Such a goal could be achieved by supervised learning from pairs of visually indistinguishable real and GAN-generated image samples, which share the same visual content and appearance. However, as pointed out in Xie et al. (2022), such pairs of image samples could not be collected directly since the GAN-generated images are typically random and different from real-world photograph images at pixel level. As a result, we have to recur to image simulation methodology. Different from the recent work (Xie et al., 2022), we propose to simulate the real images, instead of GAN-generated images, by removing the generation artifacts from GAN-generated images. Such a simulation strategy could directly address the attacking target, i.e., GAN-generated images, from which the output I^A would be far away.

For a candidate GAN-generated image I, let its corresponding simulated real image be denoted by I^R, which owns indistinguishable visual appearance with I. In order to form saliently contrastive labels, I and I^R are expected to be classified as GAN-generated and real images respectively by GAN-generated image detectors at a high probability.

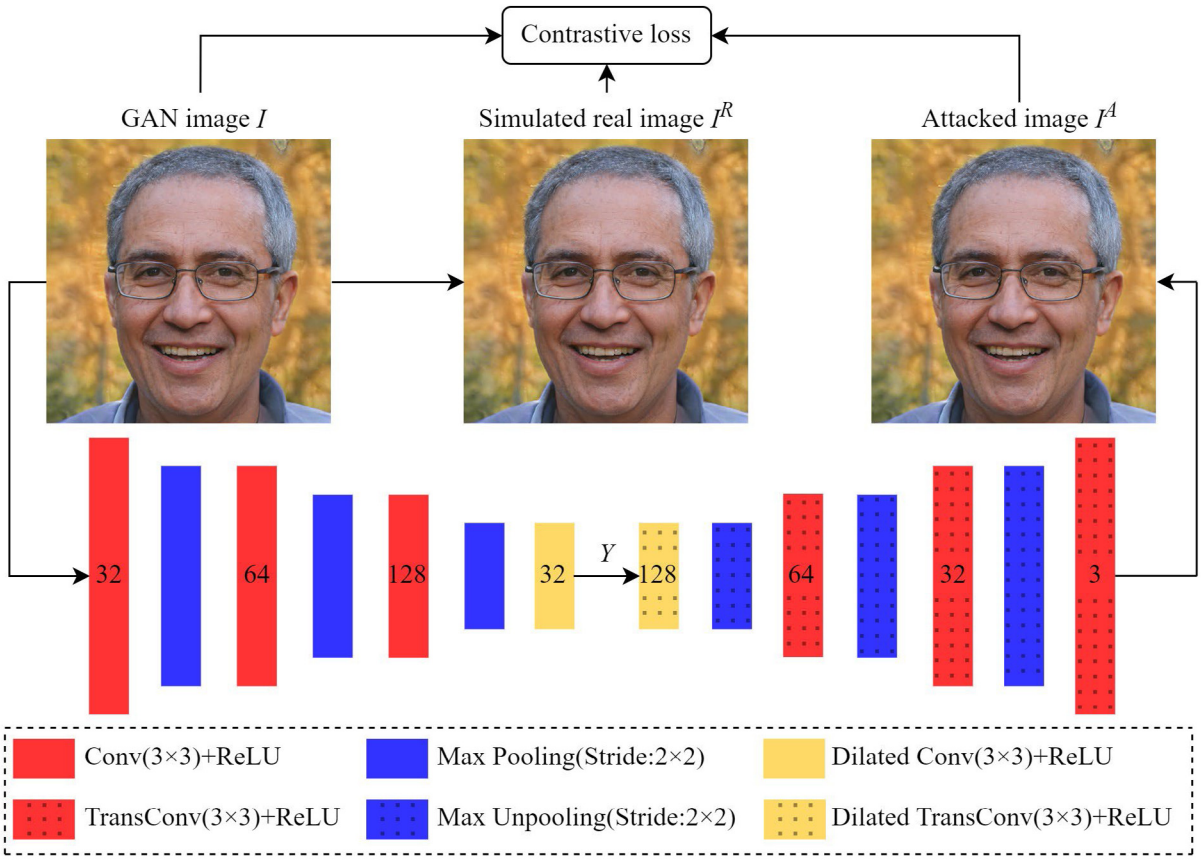


Fig. 1. Network architecture of the proposed attack scheme against GAN-generated image detectors. I represents the GAN-generated image, I^R represents the simulated real image paired with I, and I^A represents the attacked image.

Table 1
Algorithm for generating simulated real image samples.

Input: A GAN-generated Image I
Output: A Simulated Real Image Sample I^R

P_{GAN} = Detector(I)// Run Detector on input image I
if $P_{GAN} > T_1$ then
  I^R = GANPrintR(I)// Run GANPrintR on I to get I^R
  P_{GAN}= Detector(I^R)// Run Detector on output image I^R
  if $1-P_{GAN} > T_2$ then
    return I^R // Return output image I^R as the final output

In model training, generation of the attacked image I^A is guided by a contrastive loss function defined as

$$loss = \frac{1}{N}\sum_{i=1}^{N}\frac{|I^A - I^R|}{|I^A - I|} \tag{2}$$

where N denotes the number of GAN-generated and simulated real image pairs in the training sample set. To minimize the loss function, the resulting I^A reconstructed by the autoencoder should be close to I^R and away from I gradually. Since GAN fingerprint has been removed from the simulated real image I^R but intactly exists in the GAN-generated image I, the attacked image I^A would be misclassified as real by GAN-generated image detectors.

After training, the contrastive perturbation θ between the input I and output I^A can be learned by the anti-forensics model as

$$I^A = I + \theta \tag{3}$$

It implies that a GAN-generated image can be attacked by adding such contrastive perturbation θ, which is enforced by the autoencoder network. It should be mentioned that mean absolute error (MAE) or mean square error (MSE) are unsuitable to be used as loss function. Although they enable the model to rapidly converge and generate images with satisfactory visual quality, it is difficult to eliminate GAN fingerprint fully due to limited adjustment to the input image.

2.3. Generation of simulated real image samples

In this subsection, we propose an effective method for generating the simulated real image samples I^R used for model training. Table 1 presents a detailed process for generating simulated real image samples. We employed a GAN-generated image detector (Gragnaniello et al., 2021) to selectively identify images that had a high probability of being classified as GAN-generated type, with a threshold of $P_{GAN} > T_1$. Here, P_{GAN} represents the probability of being classified as a GAN-generated image by the detector, and T_1 is a predefined threshold. The GANPrintR method (Neves et al., 2020) was then utilized to eliminate the GAN fingerprint from the selected images. To generate more deceiving samples, we reused the GAN-generated image detector (Gragnaniello et al., 2021) to screen simulated real images with a lower probability of being classified as GAN-generated type, with a threshold of $P_{GAN} < 1 - T_2$. We determined that setting the experimental thresholds to $T_1 = 0.8$ and $T_2 = 0.7$ was appropriate for collecting a sufficient number of GAN-generated and simulated real images.

3. Experimental results and discussion

In this section, extensive experiments are performed to verify effectiveness of the proposed attack scheme against GAN-generated image detectors.

Table 2
Detection rate P_d of Wang detector (Sheng-Yu et al., 2020) on different types of GAN-generated images under different attack methods. GF and MF denote Gaussian low-pass and median filtering, respectively. Digitals are in percentage.

Attack methods	GAN type						
	ProGAN	StarGAN	StarGAN2	StyleGAN	StyleGAN2	StyleGAN3	Average
Without attack	100	90.95	97.73	95.45	92.42	82.70	93.21
GF (3 × 3)	94.98	81.07	92.23	72.20	71.45	84.52	82.74
GF (5 × 5)	95.32	79.12	91.27	73.82	73.30	82.75	82.60
MF (3 × 3)	95.42	82.50	93.53	68.90	71.32	85.83	82.92
MF (5 × 5)	97.58	80.92	93.40	75.32	79.08	85.73	85.34
Resizing (256 × 256)	98.97	87.03	96.85	74.83	74.55	88.65	86.81
Resizing (512 × 512)	99.95	99.37	100	89.05	89.52	95.50	95.57
GANPrintR (Huang et al., 2020)	86.37	93.78	85.60	54.52	52.07	71.42	73.96
Proposed	82.38	90.48	79.95	45.05	42.98	67.82	68.11

3.1. Datasets and GAN-generated image detectors

The training sample set consists of 14 000 pairs of GAN-generated and corresponding simulated real face images for StarGAN2, StyleGAN and StyleGAN2, respectively. Such pairs of samples are collected according to the procedure proposed in Section 2.3. The testing set is collected with total 36 000 facial images generated by 6 different GANs, where 6000 samples for each. The ProGAN, StyleGAN and StyleGAN2 images are downloaded from the public datasets shared by Nvidia Research Lab (Public database, 2019a,b; Karras et al., 2018). The StarGAN, StarGAN2 and StyleGAN3 images are created by public pre-trained generators (Choi et al., 2018b, 2019; Karras et al., 2021b). All the involved sample images are uniformly resized to 224 × 224 pixels for benefiting to implement and assess the attack schemes, which could also be adapted to other spatial resolutions by feeding proper training samples.

The following three state-of-the-art GAN-generated image detectors are attacked in tests.

(1) Wang detector (Sheng-Yu et al., 2020). It is a ResNet50 network trained on ProGAN and real images with strong data enhancement including compression and blur, which ensures generalization capability and robustness of the detector.

(2) Gragnaniello detector (Gragnaniello et al., 2021). It uses a variant Resnet50 backbone trained on ProGAN and real images with various content, such as human faces, animals and paintings. Compared with Wang detector (Sheng-Yu et al., 2020), the generalization performance is further improved by applying suitable training strategy and network architectural changes, for example, removing downsampling operation from the first layer.

(3) Kitware detector https://github.com/Kitware/generated-image-detection. It is trained on StyleGAN2 and real images. Varied image representations (raw pixels and residual images) and deep learning backbones (ResNet, EfficientNet (Tan and Le, 2019) and VGG (Simonyan and Zisserman, 2014)) have been compared experimentally and achieved approximate performance. ResNet101 is lastly adopted as backbone network.

3.2. Experimental settings and performance metrics

We use PyTorch for implementation. Our model is trained on a PC with Intel Xeon W-2245 CPU and one NVIDIA RTX 3090 GPU. We use AdamW optimizer and the kernel weights are initialized with the Xavier initializer. We set the initial learning rate to 1×10^{-3} and applied a cosine annealing strategy to decrease it to 1×10^{-6} during 200 training epochs. Due to GPU limitations, we set the batch size to 32.

As for the test sample set, the detection rates P_d of GAN-generated image detectors before and after attack are computed and compared to evaluate the attack effect. P_d is defined as the rate of accurately detected GAN-generated images in the testing set or its attacked versions. In order to evaluate the influence of attacks on visual quality of resulting images, PSNR and SSIM (Wang et al., 2004) between the naive and attacked GAN-generated images are computed.

3.3. Quantitative evaluation results

We perform quantitative statistical testing for the proposed attack scheme in this subsection. All GAN-generated images in the testing set are first processed by an anti-forensic attack method. Both original GAN-generated images and the resulting attacked images are then identified by the Wang, Gragnaniello and Kitware detectors, respectively. The following attack methods are compared detailedly.

(i) Gaussian filtering. Gaussian low-pass filtering with a 3 × 3 or 5 × 5 kernel is enforced.

(ii) Median filtering. Median filtering with a 3 × 3 or 5 × 5 kernel is enforced.

(iii) Resizing. The GAN-generated images are upsized to 256 × 256 or 512 × 512 pixels with bicubic interpolation.

(iv) GANPrintR (Neves et al., 2020). It is an autoencoder-based GAN fingerprint removal model trained merely on real face images.

Tables 2–4 show the detection rates P_d of different detectors on each type of GAN-generated images against varied attack methods. For the original GAN-generated images without attack, Wang, Gragnaniello and Kitware detectors all gain high P_d values on most types of GAN. Gragnaniello behaves the best with average P_d of 99.47%. Such results indicate high detection accuracy and good generalization capability of the detectors. Note that the generalization performance of Kitware detector is slightly weak, since its P_d for StarGAN is only 10.57%. That may attribute to the low visual quality of StarGAN-generated images.

The results indicate that Gaussian filtering (3×3, 5×5), Median filtering (3×3, 5×5), and Resizing (256×256, 512×512) have a negligible impact on detection performance, which is indicative of the robustness of the detectors. The Gragnaniello detector, for instance, exhibits an average P_d of 99.34%, 99.34%, 99.05%, 99.17%, 97.41%, 99.41% under the six aforementioned attacks, resulting in drops of 0.13%, 0.13%, 0.42%, 0.30%, 2.06%, 0.06%, respectively. In comparison to common image manipulations, the prior attack method GANPrintR (Neves et al., 2020) demonstrates superior performance. With the GANPrintR attack, the Gragnaniello detector exhibits an average P_d of 98.18%, 82.68%, 76.55%, 96.78%, 94.58%, 98.37% for six GANs.

Compared with GANPrintR, our proposed attack scheme forms more serious threat, which incurs lower detection rates for detectors. As for Wang, Gragnaniello and Kitware detectors, the average P_d values of our attack scheme are lower than those of GANPrintR by 5.85%, 2.33% and 1.03%, respectively. Meanwhile, Tables 5–6 report the visual quality of the test GAN-generated images attacked by different methods respectively. Gaussian filtering with a 3 × 3 kernel achieves average PSNR of 35.3 dB and SSIM of 0.949 on test set, which are better than that with 5 × 5 kernel. The same fact is true for Median filtering. Despite this, filtering has a negligible impact on the detection performance of the detectors. Our proposed attack scheme yields an average PSNR of 34.4 dB and SSIM of 0.945, which is comparable to GANPrintR and suggests imperceptible visual alterations. In conclusion, the quantitative assessment results demonstrate that our proposed attack scheme outperforms GANPrintR and common manipulations on all three GAN-generated image detectors, while maintaining high visual quality of the result images.

Table 3
Detection rate P_d of Gragnaniello detector (Gragnaniello et al., 2021) on different types of GAN-generated images under different attack methods. GF and MF denote Gaussian low-pass and median filtering, respectively. Digitals are in percentage.

Attack methods	GAN type						
	ProGAN	StarGAN	StarGAN2	StyleGAN	StyleGAN2	StyleGAN3	Average
Without attack	100	99.65	100	99.00	98.60	99.57	99.47
GF (3 × 3)	100	98.55	100	98.90	98.73	99.88	99.34
GF (5 × 5)	100	98.55	100	98.90	98.73	99.88	99.34
MF (3 × 3)	100	97.97	100	98.30	98.13	99.87	99.05
MF (5 × 5)	100	95.93	100	99.57	99.52	99.98	99.17
Resizing (256 × 256)	99.98	99.32	100	95.02	92.88	97.25	97.41
Resizing (512 × 512)	100	100	100	98.12	98.52	99.82	99.41
GANPrintR (Huang et al., 2020)	96.78	98.37	98.18	82.68	76.55	94.58	91.19
Proposed	93.58	97.80	94.70	79.28	74.43	93.35	88.86

Table 4
Detection rate P_d of Kitware detector https://github.com/Kitware/generated-image-detection on different types of GAN-generated images under different attack methods. GF and MF denote Gaussian low-pass and median filtering, respectively. Digitals are in percentage.

Attack methods	GAN type						
	ProGAN	StarGAN	StarGAN2	StyleGAN	StyleGAN2	StyleGAN3	Average
Without attack	97.15	10.57	96.50	97.73	99.33	68.72	78.33
GF (3 × 3)	78.82	1.85	47.30	80.85	86.30	18.98	52.35
GF (5 × 5)	78.82	1.85	47.30	80.85	86.30	18.98	52.35
MF (3 × 3)	90.63	12.22	87.60	91.67	97.05	56.02	72.53
MF (5 × 5)	79.30	9.58	60.13	80.28	88.18	42.08	59.93
Resizing (256 × 256)	96.27	5.43	88.93	94.15	98.63	59.70	73.85
Resizing (512 × 512)	95.97	3.90	84.20	93.55	98.35	52.33	71.38
GANPrintR (Huang et al., 2020)	88.02	37.98	85.75	87.32	96.32	50.75	74.36
Proposed	84.73	34.32	81.57	85.22	95.73	58.40	73.33

Table 5
PSNR (dB) of the different types of GAN-generated images altered by different attack methods. GF and MF denote Gaussian low-pass and median filtering, respectively.

Attack methods	GAN type						
	ProGAN	StarGAN	StarGAN2	StyleGAN	StyleGAN2	StyleGAN3	Average
GF (3 × 3)	31.6	40.7	38.3	31.9	31.5	37.7	35.3
GF (5 × 5)	29.7	36.8	35.1	30.0	29.6	34.7	32.7
MF (3 × 3)	31.0	41.2	38.8	31.2	30.9	37.9	35.2
MF (5 × 5)	28.6	35.1	33.6	28.9	28.5	33.3	31.4
GANPrintR (Huang et al., 2020)	32.4	38.7	38.5	31.9	31.6	38.1	35.2
Proposed	32.0	37.3	37.2	31.5	31.2	36.9	34.4

Table 6
SSIM of the different types of GAN-generated images altered by different attack methods. GF and MF denote Gaussian low-pass and median filtering, respectively.

Attack methods	GAN type						
	ProGAN	StarGAN	StarGAN2	StyleGAN	StyleGAN2	StyleGAN3	Average
GF (3 × 3)	0.920	0.988	0.976	0.917	0.921	0.975	0.949
GF (5 × 5)	0.880	0.972	0.952	0.877	0.882	0.952	0.919
MF (3 × 3)	0.892	0.986	0.972	0.889	0.895	0.970	0.934
MF (5 × 5)	0.835	0.949	0.921	0.830	0.836	0.924	0.882
GANPrintR (Huang et al., 2020)	0.933	0.979	0.977	0.914	0.920	0.978	0.950
Proposed	0.929	0.972	0.972	0.910	0.915	0.974	0.945

3.4. Qualitative evaluation results

To qualitatively evaluate the performance of different attack methods, six example images generated by different GANs are analyzed illustratively. Fig. 2 shows such GAN-generated images and their corresponding attacked versions. It can be seen that our attack scheme preserves high visual quality without leaving visible abnormal traces. GANPrintR also keeps high visual quality, while the manipulation attacks, i.e., GF and MF, cause apparent blurriness to some extent. Such visual results can also be validated consistently by the corresponding PSNR and SSIM measurements reported in Table 7.

Table 8 shows P_{GAN} of Gragnaniello detector (Gragnaniello et al., 2021) on the six example images. The GF and MF manipulations bring little decrease to the GAN-generated image detection performance, which keeps consistent with the quantitative evaluations. Compared with GANPrintR (Neves et al., 2020), our proposed attack scheme could fool the Gragnaniello detector at higher probability. For example, P_{GAN} achieves 99.99% on the unattacked example GAN-generated image *a*, and is reduced to 96.39% by GANPrintR. In contrast, the proposed attack scheme has yielded a significant reduction of P_{GAN} to 11.18%, as evidenced by our results. Such findings serve to validate the efficacy and performance superiority of our proposed attack scheme.

4. Conclusion

In this paper, we propose a new black-box attack method against GAN-generated image detectors. A novel contrastive learning strategy is adopted to train the anti-forensic model with a contrastive loss

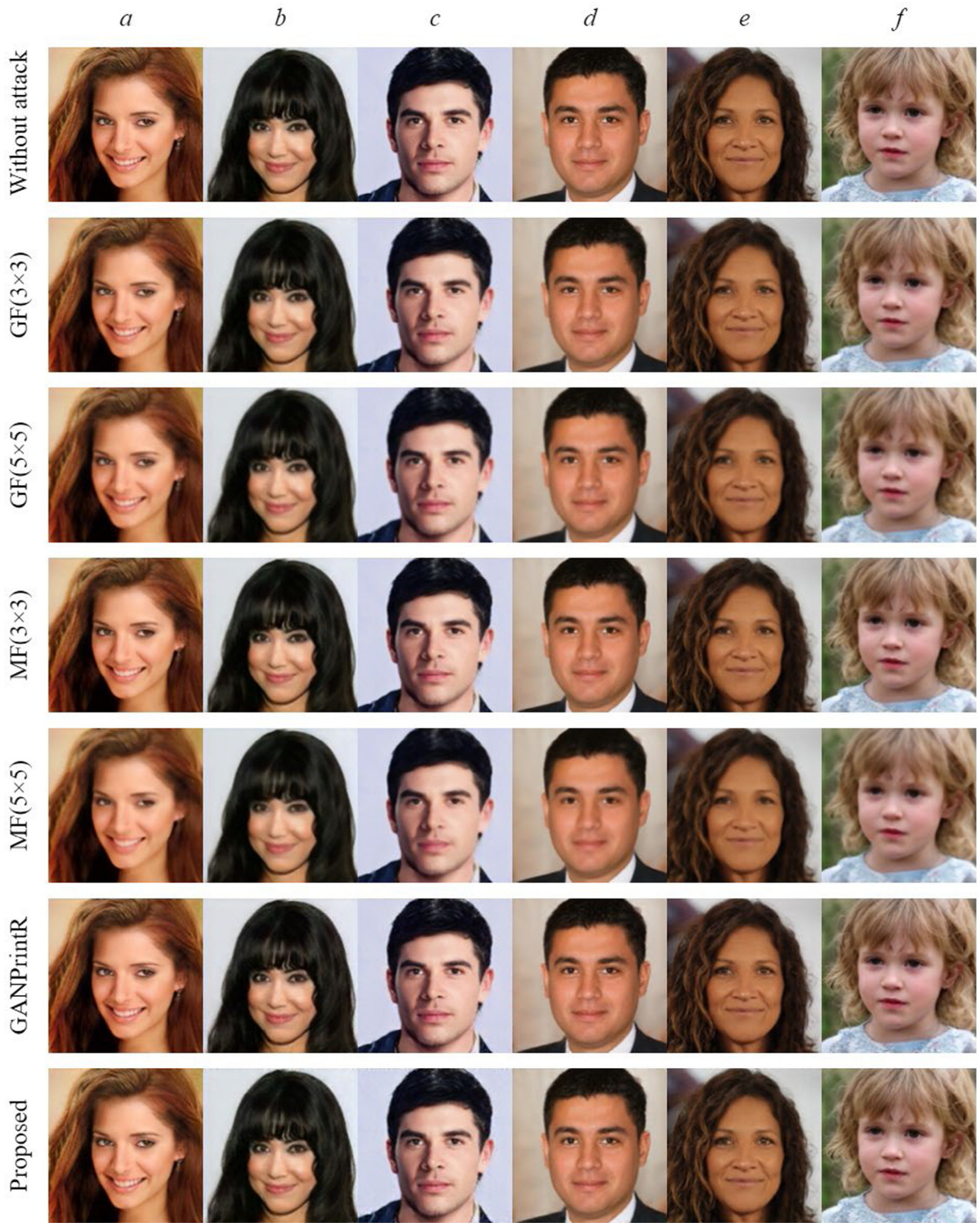


Fig. 2. Visual examples of GAN-generated images without and with different attacks. The columns *a–f* are for ProGAN, StarGAN, StarGAN2, StyleGAN, StyleGAN2 and StyleGAN3, respectively.

function. We design a lightweight network for this work. Extensive experimental results verify that our approach effectively reduces the accuracy of three state-of-the-art detectors on six popular GANs. High visual quality of the attacked images is also achieved. Novel methods for collecting sample counterparts and training network are proposed in our attack scheme, which leads to a new black-box attack pipeline against generative image detectors. It will contribute to the advancement on the security evaluation of multimedia forensics techniques. Main defects of the proposed method include the lower successful rate than white-box attacks and inadaptation for poorly generated images. Our future work will be focused on addressing such defects and improving the attacking effectiveness against various forensic detectors.

CRediT authorship contribution statement

Zijie Lou: Take part in the discussion of the work described in this paper, Conceived and designed the experiments, Performed the experiments and analyzed the data, Wrote the paper. **Gang Cao:** Take

Table 7
PSNR (dB) and SSIM of the six example GAN-generated images altered by different attack methods. GF and MF denote Gaussian low-pass and median filtering, respectively.

Attack methods	GAN-generated image											
	a		*b*		*c*		*d*		*e*		*f*	
	PSNR	*SSIM*	*PSNR*	*SSIM*	*PSNR*	*SSIM*	*PSNR*	*SSIM*	*PSNR*	*SSIM*	*PSNR*	*SSIM*
GF (3 × 3)	30.6	0.875	39.4	0.987	35.7	0.978	34.5	0.941	34.8	0.935	35.6	0.959
GF (5 × 5)	29.1	0.822	35.5	0.971	32.4	0.956	32.3	0.913	32.7	0.899	32.8	0.919
MF (3 × 3)	29.4	0.824	41.7	0.987	37.3	0.976	34.7	0.923	34.1	0.915	35.5	0.953
MF (5 × 5)	27.8	0.756	35.8	0.955	32.1	0.935	32.3	0.890	31.3	0.852	31.0	0.864
GANPrintR (Huang et al., 2020)	30.5	0.876	38.0	0.978	36.1	0.970	34.6	0.935	35.2	0.945	36.6	0.973
Proposed	30.3	0.874	36.7	0.971	34.8	0.961	34.0	0.928	34.8	0.944	35.5	0.969

Table 8
P_{GAN} of Gragnaniello detector (Gragnaniello et al., 2021) on six example GAN-generated images under different attack methods. GF and MF denote Gaussian low-pass and median filtering, respectively. Digitals are in percentage.

Attack methods	GAN-generated image					
	a	*b*	*c*	*d*	*e*	*f*
Without attack	99.99	83.91	99.99	100	100	87.26
GF (3 × 3)	100	86.62	100	100	100	99.93
GF (5 × 5)	100	80.98	100	100	100	100
MF (3 × 3)	99.99	66.66	99.99	100	100	99.71
MF (5 × 5)	100	83.53	100	100	100	100
GANPrintR (Huang et al., 2020)	96.39	83.09	93.13	59.72	62.36	59.49
Proposed	11.18	14.83	79.44	27.28	57.28	19.03

part in the discussion of the work described in this paper, Conceived and designed the experiments, Wrote the paper. **Man Lin:** Take part in the discussion of the work described in this paper, Performed the experiments and analyzed the data.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant No. 62071434), the Fundamental Research Funds for the Central Universities, China (Grant No. CUC22GZ065). All authors approved the version of the manuscript to be published.

References

Agarap, A.F., 2018. Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375.
Carlini, N., Farid, H., 2020. Evading deepfake-image detectors with white-and black-box attacks. In: Proc. IEEE/CVF Conf. on Comput. Vis. and Pattern Recognit. Workshops. pp. 658–659.
Choi, Y., Choi, M., Kim, M., et al., 2018a. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In: Proc. IEEE Conf. on Comput. Vis. Pattern Recognit.. pp. 8789–8797.
Choi, Yunjey, Uh, Youngjung, Yoo, Jaejun, Jung-WooHa, 2018b. Pre-trained stargan. https://github.com/clovaai/stargan.
Choi, Yunjey, Uh, Youngjung, Yoo, Jaejun, Jung-WooHa, 2019. Pre-trained stargan-v2. https://github.com/clovaai/stargan-v2.
Choi, Y., Uh, Y., Yoo, J., et al., 2020. Stargan v2: Diverse image synthesis for multiple domains. In: Proc. IEEE Conf. on Comput. Vis. Pattern Recognit.. pp. 8188–8197.
Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. In: Proc. IEEE Conf. on Comput. Vis. Pattern Recognit.. pp. 1251–1258.
2017. Faceapp. [Online]. Available: https://apps.apple.com/us/app/faceapp-ai-face-editor/id1180884341.
Goodfellow, Ian, et al., 2020. Generative adversarial networks. Commun. ACM 63 (11), 139–144.
Gragnaniello, Diego, et al., 2021. Are GAN generated images easy to detect? A critical analysis of the state-of-the-art. In: Proc. IEEE Int. Conf. on Multimedia Expo.. pp. 1–6.
He, K., Zhang, X., Ren, S., et al., 2016. Deep residual learning for image recognition. In: Proc. IEEE Conf. on Comput. Vis. Pattern Recognit.. pp. 770–778.
Huang, Yihao, et al., 2020. Fakepolisher: Making deepfakes more detection-evasive by shallow reconstructi-on. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 1217–1226.
Iandola, F., Moskewicz, M., Karayev, S., et al., 2014. Densenet: Implementing efficient convnet descriptor pyramids. arXiv preprint arXiv:1404.1869.
Karras, T., Aila, T., Laine, S., et al., 2017. Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196.
Karras, T., Aila, T., Laine, S., et al., 2018. Public database of progan. https://github.com/tkarras/progressive_growing_of_gans.
Karras, T., Aittala, M., Laine, S., et al., 2021a. Alias-free generative adversarial networks. In: Proc. Adv. Neural Inf. Process. Syst.. pp. 852–863.
Karras, T., Aittala, M., Laine, S., et al., 2021b. Pre-trained stylegan3. https://github.com/NVlabs/stylegan3.
Karras, T., Laine, S., Aila, T., 2019. A style-based generator architecture for generative adversarial networks. In: Proc. IEEE Conf. on Comput. Vis. Pattern Recognit.. pp. 4401–4410.
Karras, T., Laine, S., Aittala, M., et al., 2020. Analyzing and improving the image quality of stylegan. In: Proc. IEEE Conf. on Comput. Vis. Pattern Recognit.. pp. 8110–8119.
Marra, F., Gragnaniello, D., Cozzolino, D., et al., 2018. Detection of gan-generated fake images over social networks. In: Proc. IEEE Conf. on Multimedia Info. Process. and Retrieval. pp. 384–389.
Matern, F., Riess, C., Stamminger, M., 2019. Exploiting visual artifacts to expose deepfakes and face manipulations. In: Proc. IEEE Winter Applications of Computer Vision Workshops. pp. 83–92.
Na, Dongbin, Ji, Sangwoo, Kim, Jong, 2022. Unrestricted black-box adversarial attack using GAN with limited queries. In: ECCV Workshops. pp. 467–482.
Neves, J.C., Tolosana, R., Vera-Rodriguez, R., et al., 2020. Ganprintr: Improved fakes and evaluation of the state of the art in face manipulation detection. IEEE J. Sel. Top. Signal Process. 14 (5), 1038–1048.
Odena, A., Dumoulin, V., Olah, C., 2016. Deconvolution and checkerboard artifacts. Distill 1 (10), e3.
Papernot, Nicolas, et al., 2016. The limitations of deep learning in adversarial settings. In: IEEE European Symposium on Security and Privacy. pp. 372–387.
2019a. Public Database of Stylegan. Nvidia Research Lab, https://github.com/NVlabs/stylegan.
2019b. Public Database of Stylegan. Nvidia Research Lab, https://github.com/NVlabs/stylegan2.
Sheng-Yu, W., et al., 2020. Cnn-generated images are surprisingly easy to spot... for now. In: Proc. IEEE Conf. on Comput. Vis. Pattern Recognit.. pp. 8695–8704.
Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
Szegedy, C., Vanhoucke, V., Ioffe, S., et al., 2016. Rethinking the inception architecture for computer vision. In: Proc. IEEE Conf. on Comput. Vis. Pattern Recognit.. pp. 2818–2826.

Szegedy, Christian, et al., 2013. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199.

Tan, M., Le, Q., 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In: Int. Conf. on Mach. Learn.. pp. 6105–6114.

Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E., 2004. Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. 13 (4), 600–612.

Xie, Hao, et al., 2022. Evading generated-image detectors: A deep dithering approach. Signal Process. 197, 108558.

Zamir, S.W., Arora, A., Khan, S., et al., 2021. Multi-stage progressive image restoration. In: Proc. IEEE Conf. on Comput. Vis. Pattern Recognit.. pp. 14821–14831.

Zhao, X., Stamm, M.C., 2021. Making GAN-generated images difficult to spot: a new attack against synthetic image detectors. arXiv preprint arXiv:2104.12069.