An Adversarial Approach for Anti-Detection of AI-Generated Images through Sample Generation

Meng-Luen Wu
Department of Computer Science and Information Engineering
Tamkang University
New Taipei City, Taiwan
158769@mail.tku.edu.tw

Cheng-Bing Cheng
Department of Computer Science and Information Engineering
Tamkang University
New Taipei City, Taiwan
407411361@mail.tku.edu.tw

***Abstract*—Artificial intelligence-generated images have become increasingly realistic, often rendering flaws imperceptible to the human eye. In recent years, diffusion models have gained prominence, enabling users to generate images from descriptive text. To mitigate the potential for malicious misuse, various detection methods have been developed to identify images produced by these models, demonstrating strong performance in experimental settings. However, concerns remain regarding their robustness in real-world applications, particularly against adversarial attacks. Accurately detecting such images under these conditions presents a significant challenge. In this study, we propose a novel attack method aimed at evaluating the resilience of these detection techniques. Our approach involves eliminating the frequency-domain fingerprints commonly associated with synthetic images, thereby generating adversarial samples. We further enhance the generalization capabilities of these samples to effectively challenge existing detection methods. Our experiments reveal significant vulnerabilities in these systems, highlighting the need for ongoing research to improve the detection of diffusion model-generated images and ensure their reliability in practical scenarios.**

Keywords—GAN, Adversarial Attack, Diffusion Model, Adversarial Samples

I. Introduction

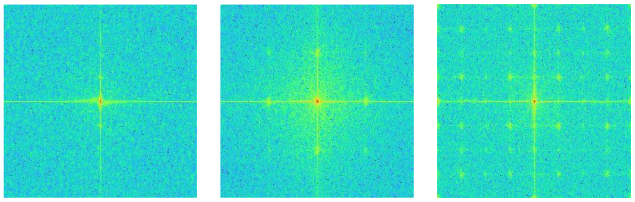
Many years ago, with the advent of Generative Adversarial Networks (GAN) [1], it became possible to generate incredibly realistic images. While some use GAN for entertainment, others exploit them maliciously, creating misleading images of celebrities or war-related weapons. These concerns have led to the development of various detection methods. Recently, new architectures like Denoising Diffusion Probabilistic Models (DDPM) [2] have emerged, significantly advancing synthetic image generation. Models such as Latent Diffusion [3], Stable Diffusion [3], GLIDE [4], and DALL·E 2 [5] can produce high-quality images. However, similar issues arise with diffusion models, necessitating detection methods to distinguish real from generated images. Previous detection methods either input images directly into neural networks or convert them into the frequency domain, as [6] found that the up-sampling step in CNNs leaves distinctive fingerprints, aiding in differentiation.

In [7], the effectiveness of past GAN detection methods on diffusion model-generated images was explored, revealing that even robust methods struggled due to differing characteristics. Several detection methods for diffusion models have been proposed, achieving high accuracy, but their robustness remains to be validated. For instance, [8] directly input images for detection, while [9] focused on the similarity of regenerated images. Additionally, [10] analyzed the differences between real images and those generated from prompts, arguing that generated images closely match their prompts. While these methods have shown over 90% accuracy, robustness is crucial, especially in real-world scenarios where images may be compressed or blurred. Testing methods often involve applying compression or Gaussian blur to assess detection effectiveness. Some methods fail under such conditions, indicating a lack of robustness.

Beyond basic testing methods, more in-depth testing involves designing specific attacks on detection methods. For instance, [11] utilized the Iterative Fast Gradient Sign Method (I-FGSM) to iteratively adjust pixel values based on detection feedback until the detection method was deceived, further testing its robustness.

Currently, basic robustness tests like compression and Gaussian blur are applied to diffusion model detection methods. However, no specific method has been designed to attack these detection methods and assess their vulnerabilities. This paper proposes a method to specifically attack diffusion model detection methods to evaluate their robustness. Our method has two main objectives: first, to successfully reduce the detection accuracy of targeted detection methods, and second, to generalize the method to unseen detection methods. Initially, we eliminate the fingerprint left in the frequency domain by the image generation process. As noted in [7], images generated by diffusion models, similar to those generated by GAN, leave detectable fingerprints. Using the filter proposed by [12], we obtained Fig. 1, which illustrates the fingerprints left by diffusion model-generated images: (a) Real Image, (b) GLIDE, (c) Latent Diffusion.

We believe that removing these obvious detection features will reduce the basis for determining image authenticity. Thus, we designed a GAN to eliminate these fingerprints in the frequency domain. The processed images are then input into the detection method to obtain a score. We iteratively adjust the pixel values and reinput the images into the detection method, repeating this process until the score diminishes to our set target.



(a) (b) (c)

Fig. 1: The fingerprints of images after Fourier transformation in the frequency domain

The objective is to generalize our method for unseen detection methods using adversarial samples, which are tailored based on feedback from the input detection model. By enhancing the pixel value adjustment process, we aim to demonstrate that successful attacks on unseen methods indicate a lack of robustness in those detection techniques.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

II. RELATED WORKS

A. Adversarial Perturbation Attack

Adversarial perturbation attacks disrupt features that distinguish real from generated images, exploiting detection methods that rely on single features. The simplest method involves image compression, which can lead to detection errors. In [13], four types of adversarial perturbation attacks are proposed: a threshold-based method misclassifies inputs by ensuring predicted probabilities fall below a set threshold; a loss function maximizes the probability of misclassification while limiting perturbation; a universal adversarial patch is added, though it is visually noticeable; and a universal feature space attack misclassifies any image with a specific perturbation, which is less detectable. In [14], a method minimizes a loss function based on classification probabilities, using the gradient sign method for perturbation, enhancing generalization for black-box attacks. This method introduces a universal perturbation that can be easily applied to achieve effective misclassification.

B. Elimination of Manipulation Fingerprints in the Frequency Domain

Elimination of manipulation fingerprints in the frequency domain, aims to eliminate operational fingerprints in the frequency domain, identified as differences between real and generated images. In [6], it was found that these fingerprints result from the up-sampling process, and removing them can mislead detection methods. In [15], a method incorporating noise addition and deep filtering was introduced, effectively disrupting fingerprints while aligning spectral distributions, though it caused quality degradation. They proposed selective noise addition using an adversarial guided map for better results. Additionally, [16] introduced SpectralGAN, which includes an extra discriminator to improve the spectral distribution of generated images, reducing detectable fingerprints and enhancing indistinguishability from real images.

C. Employing Image Filtering to Mislead Detectors

This attack method uses image filters to mislead detectors by addressing operational fingerprints broadly, not just in the frequency domain. In [17], the detectable checkerboard-like spectral fingerprint from CycleGAN-generated images was tackled by improving the CycleGAN architecture with fixed convolutional layers to eliminate up-sampling issues, effectively reducing detection accuracy. Experimental results showed that the modified CycleGAN significantly decreased detection rates compared to the original. Additionally, in FakePolisher [18], a dictionary model trained on real images created a low-dimensional space for regenerating images, which significantly reduced fingerprints without introducing new generative artifacts.

III. PROPOSED METHOD

A. System Architecture

Our proposed GAN consists of a generator that regenerates images to eliminate fingerprints in the frequency domain and a discriminator that penalizes the generator during training. Once trained, the generator removes frequency domain fingerprints from images generated by the diffusion model. The processed images are then used to generate adversarial samples using the gradient sign method. Fig. 2 illustrates the architecture of this method, which will be detailed in two parts: the elimination of frequency domain fingerprints and the generation of adversarial samples.

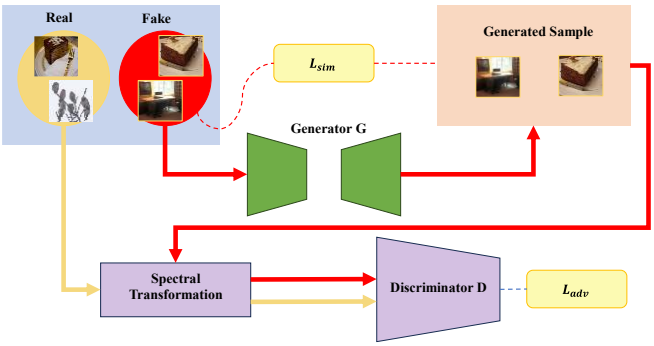


Fig. 2: Architecture of generating adversarial sample approach

B. Eliminating Frequency Domain Fingerprint

To eliminate frequency fingerprints, we employ a GAN architecture based on the design proposed in [19], which originally includes one generator and three discriminators to address spatial anomalies, spectral differences, and specific fingerprints. However, our method focuses solely on frequency domain fingerprints, utilizing only one discriminator to differentiate between the spectra of real and generated images, while maintaining the generator's original architecture. Fig. 3 illustrates our framework for eliminating frequency domain fingerprints.

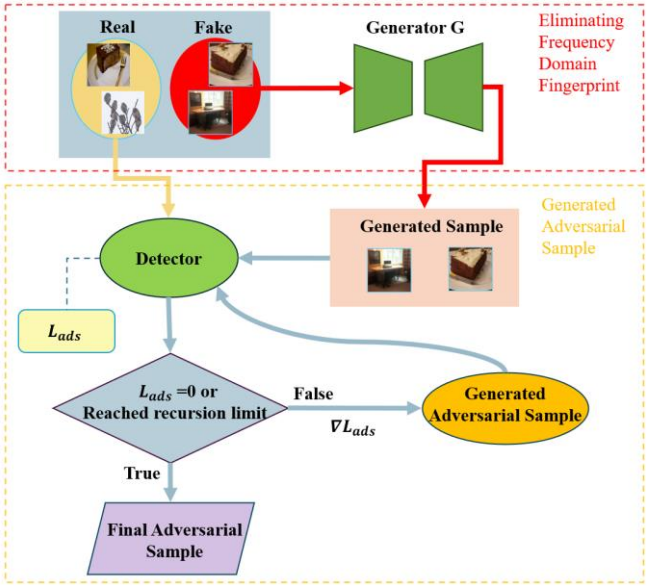


Fig. 3: Architecture of eliminating frequency domain fingerprint.

C. Generator

For the generator G, we utilize the architecture proposed by [19], based on the U-Net structure [20]. Fig. 4 illustrates this architecture, which performs three main tasks: down-sampling through feature extraction and halving the feature map size, up-sampling by enlarging and concatenating feature maps with skip connections, and addressing frequency domain fingerprints introduced during up-sampling. To prevent new fingerprints, [19] incorporated a "feature scaling layer" that replaces transpose convolutions with nearest or bilinear up-sampling, as suggested by [21]. This modification ensures that the final enlarged feature map undergoes 1×1 convolution without introducing new frequency domain fingerprints.

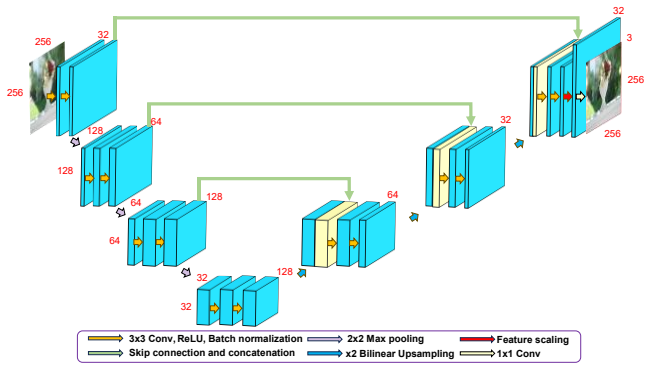


Fig. 4: Architecture of Generator

1) Discriminator

The discriminator D employs a simpler CNN architecture, as noted by [19], to prevent an imbalance that could hinder the generator's training. This design allows the discriminator to effectively learn the frequency domain fingerprints of generated images, aiding the generator in their elimination. Fig. 5 illustrates the discriminator's architecture, which convolves the frequency domain image six times before passing it through a fully connected layer. The output, which does not utilize an activation function, yields values ranging from $[-\infty, \infty]$; larger values indicate a higher likelihood of being a real image, while smaller values suggest a generated fake image. We utilize the two-dimensional Discrete Fourier Transform (2D-DFT) to convert images into the frequency domain.

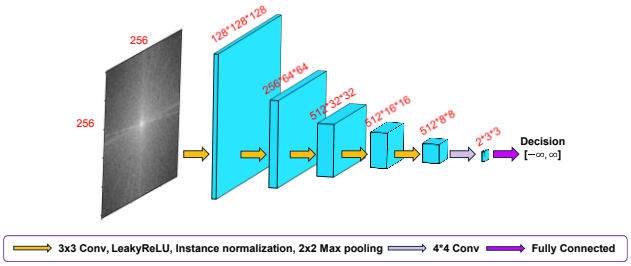


Fig. 5: Architecture of Discriminator

2) Loss Function

In training our GAN, we initially used cross-entropy as the loss function, as noted in [19]. However, this approach failed to eliminate frequency domain fingerprints, likely due to the use of a single discriminator, which provided insufficient gradient feedback to the generator. To address this, we adopted the Wasserstein GAN (WGAN) loss function, which stabilizes learning by mitigating the issues caused by overly strong discriminators. The loss functions for the generator LG and discriminator L_D are defined as follows:

$$L_G = -\mathop{\mathbb{E}}_{G(x_{fake})\sim\mathbb{P}_g}[D(G(x_{fake}))] \tag{1}$$

$$L_D = \mathop{\mathbb{E}}_{G(x_{fake})\sim\mathbb{P}_g}\left[D\left(G(x_{fake})\right)\right] - \mathop{\mathbb{E}}_{x_{real}\sim\mathbb{P}_r}[D(x_{real})] + k\mathop{\mathbb{E}}_{\hat{x}\sim\mathbb{P}_u}\|\nabla_{\hat{x}}D(\hat{x})\|^p \tag{2}$$

where P_r is the probability distribution of real images, and $\hat{x}$ is a combination of x_{real} and x_{fake}. The adversarial loss for the GAN is defined as:

$$L_{adv} = L_G + L_D\,, \tag{3}$$

To ensure the regenerated images closely resemble the originals, we incorporate Perceptual Loss $L_{sim}(G)$:

$$L_{sim}(G) = \frac{1}{H_{fea}\times W_{fea}\times 3}\left\|\mathrm{VGG}(x_{fake}) - \mathrm{VGG}(G(x_{fake}))\right\|_2^2, \tag{4}$$

This loss function utilizes the VGG network to compare feature representations, guiding the generator to produce images similar to the originals, where H_{fea} and W_{fea} denote the height and width of the feature maps, respectively.

D. Generated Adversarial Sample

We utilize the Iterative Fast Gradient Sign Method (I-FGSM) proposed by [11] for generating adversarial samples. The process begins by inputting the image into the detector, which classifies its authenticity. The detector's output is then used to compute a value based on a defined loss function, allowing us to derive the gradient for pixel value adjustments. This iterative process continues until the maximum iterations are reached or the detector is deceived. To limit adjustments and prevent noticeable distortions, we define a perturbation magnitude ϵ and apply the clipping formula:

$$Clip_{x,\epsilon}\{x'\}(h,w,z) = \min\{255, x(h,w,z)+\epsilon, max\{0, x(h,w,z)-\epsilon, x'(h,w,z)\}\} \tag{5}$$

Here, x' is the modified image, with h and w as pixel coordinates, and z representing RGB channels. For the loss function, we adopt the one from [14], as it minimizes distortion and enhances robustness, defined as:

$$L_{ads}(x') = \max\left(f(x')_o - f(x')_y, 0\right) \tag{6}$$

where $f(x')_o$ and $f(x')_y$ are the predicted values for the original and opposite classes, respectively. Our goal is to minimize L_{ads}. The I-FGSM is defined as:

$$x_0^{adv} = x\,,\quad x_{N+1}^{adv} = Clip_{x,\epsilon}\{x_N^{adv} - \alpha\,\mathrm{sign}\left(\nabla_x L_{ads}(x_N^{adv})\right)\} \tag{7}$$

In this method, pixel values are adjusted based on the gradient's direction, controlled by α, while ensuring adjustments remain within the specified perturbation magnitude ϵ.

IV. EXPERIMENTAL RESULTS

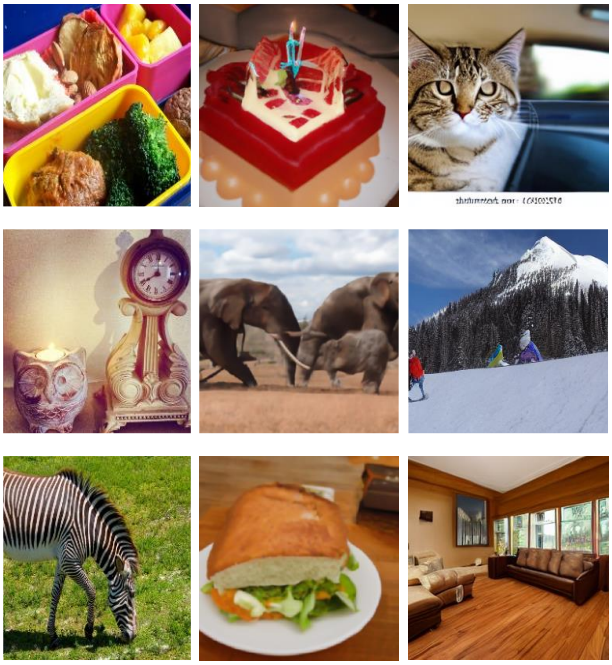
In this study, we aimed to evaluate the robustness of diffusion model detection methods against adversarial attacks. Our experiments focused on generating adversarial samples and assessing their impact on the accuracy of three prominent detection methods: UniFD, DIGBD, and SSIP. We utilized a dataset comprising 20,000 real images from the 2017 COCO dataset and 20,000 fake images generated by two diffusion models: Latent Diffusion and GLIDE.

A. Evaluation Metrics

The primary metric for evaluating the effectiveness of our adversarial samples was accuracy. We measured how well each detection method could classify images as real or fake before and after the introduction of adversarial samples.

B. Dataset Composition

The dataset was split into training, validation, and testing sets, with an 80-20 ratio for real and fake images. The detailed composition is shown in Table 1.



(a) (b) (c)

Fig. 6: Dataset used in this study: (a) 2017 COCO dataset, (b) GLIDE, (c) Latent Diffusion.

Table 1. Detailed Composition of the Real Image Dataset.

Dataset Component	Real Images	Fake Images
Training Set	14,400	14,400
Validation Set	1,600	1,600
Test Set	4,000	4,000

C. Key Experimental Findings

We conducted experiments to generate adversarial samples using the Iterative Fast Gradient Sign Method (I-FGSM). The results demonstrated a significant decrease in the accuracy of the detection methods when subjected to adversarial samples. Table 2 summarizes the accuracy of each detection method before and after the introduction of adversarial samples.

Table 2. Accuracy of Detection Methods.

Dataset Component	Accuracy Before Attack (%)	Accuracy After Attack (%)
UniFD [22]	92%	72%
DIGBD [23]	89%	69%
SSIP [24]	85%	11%

As shown in Table 2, UniFD maintained a relatively higher accuracy compared to DIGBD and SSIP, which experienced a drastic drop, particularly SSIP, which fell to 11%. This indicates that while some methods exhibit robustness, others are significantly vulnerable to adversarial attacks.

D. Conclusion of Results

The experiments highlight the effectiveness of our approach in generating adversarial samples that can deceive diffusion model detection methods. The substantial reduction in accuracy across the board underscores the need for ongoing research to enhance the robustness of these detection systems against adversarial attacks. Future work should focus on developing more resilient detection methods that can withstand such manipulations.

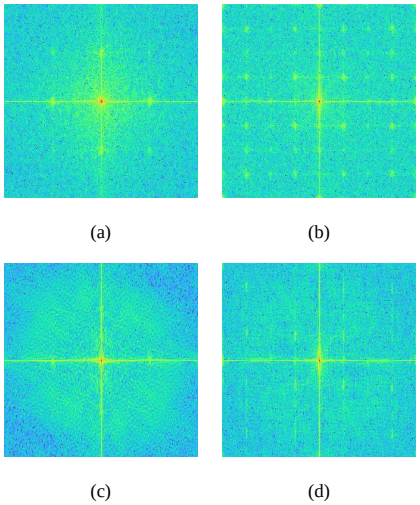
E. Eliminating Frequency Domain Fingerprint

Regarding the experiments on eliminating frequency fingerprint, we will first introduce our training setup. Previously, we discussed the dataset. During our training, we found that using the loss function from Wasserstein Divergence for GANs [25] significantly increased the training time. Therefore, we used a subset of the dataset for training.

During the actual training of the entire network, we set the batch size to 8 and used the Adam optimizer [26]. The learning rates for the discriminator and generator were set to $1.6e^{-4}$ and $1.6e^{-2}$, respectively. We also utilized a scheduler to aid in the learning process, set to reduce the learning rate by half every 5 epochs. The total training spanned 40 epochs. Training our model on a 14-core 2.7 GHz CPU and a 6GB RAM Nvidia GPU RTX 3060 took one week.

1) Presenting the Effect of Eliminating Frequency Domain Fingerprint

In our experiments, we found that although our goal was to eliminate frequency domain fingerprints, we did not achieve perfect results. However, we were able to partially remove some fingerprints. Fig. 7 shows our results: (a) Original GLIDE frequency domain fingerprints, (b) Original Latent Diffusion frequency domain fingerprints, (c) GLIDE frequency domain fingerprints after elimination, (d) Latent Diffusion frequency domain fingerprints after elimination. Despite not achieving complete elimination of the fingerprints, our subsequent testing revealed that the partial removal still had an impact on the detectors.



(a) (b)

(c) (d)

Fig. 7: The effect of eliminating frequency domain fingerprint

V. CONCLUSIONS

This study explored the vulnerabilities of diffusion model detection methods to adversarial attacks, highlighting the need for enhanced robustness in these systems. Through a series of experiments, we demonstrated the effectiveness of generating adversarial samples using the Iterative Fast Gradient Sign Method (I-FGSM). Our findings revealed that while some detection methods, such as UniFD, exhibited a

degree of resilience, others, like SSIP and DIGBD, experienced significant drops in accuracy when faced with adversarial inputs.

The results underscore the critical importance of not only achieving high accuracy in detection but also ensuring that these methods can withstand adversarial manipulations. As the landscape of AI-generated content continues to evolve, it is imperative for detection systems to adapt and improve. Future research should focus on developing more robust detection techniques that can effectively counteract adversarial attacks, thereby maintaining the integrity of image authenticity verification in an increasingly complex digital environment.

REFERENCES

[1] I. Goodfellow et al., "Generative adversarial nets," in Proc. Int. Conf. Neural Inf. Process. Syst., pp. 2672–2680, 2014.

[2] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in Proc. Int. Conf. Neural Inf. Process. Syst., pp. 6840–6851,2020.

[3] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in CVPR, pp. 10684–10695,2022

[4] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, "GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models," arXiv preprint arXiv:2112.10741, 2021.

[5] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," arXiv preprint arXiv:2204.06125v1, 2022.

[6] R. Durall, M. Keuper, and J. Keuper, ''Watch your up-convolution: CNN based generative deep neural networks are failing to reproduce spectral distributions,'' in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Seattle, WA, USA, Jun. pp. 7887–7896,2020

[7] R. Corvi, D. Cozzolino, G. Zingarini, G. Poggi, K. Nagano, and L. Verdoliva, "On the detection of synthetic images generated by diffusion models," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5,2023.

[8] L. Guarnera, O. Giudice, and S. Battiato, "Level up the deepfake detection: a method to effectively discriminate images generated by gan architectures and diffusion models," arXiv preprint arXiv:2303.00608, 2023.

[9] Z. Wang, J. Bao, W. Zhou, W. Wang, H. Hu, H. Chen, and H. Li, "Dire for diffusion-generated image detection," arXiv preprint arXiv:2303.09295, 2023.

[10] Z. Sha, Z. Li, N. Yu, and Y. Zhang, "DE-FAKE: Detection and Attribution of Fake Images Generated by Text-to-Image Diffusion Models," arXiv preprint arXiv:2210.06998, 2022.

[11] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in Proc. Int. Conf. Learning Representations, pp. 1–14, 2017.

[12] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," IEEE Trans. Image Process., vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[13] N. Carlini and H. Farid, ''Evading deepfake-image detectors with whiteand black-box attacks,'' arXiv:2004.00622, 2020.

[14] P. Neekhara, B. Dolhansky, J. Bitton, and C. C. Ferrer, "Adversarial threats to DeepFake detection: A practical perspective," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), pp. 923–932, 2021.

[15] Y. Huang et al., "FakeRetouch: Evading DeepFakes detection via the guidance of deliberate noise," arXiv:2009.09213, 2020.

[16] S. Jung and M. Keuper. Spectral Distribution aware Image Generation. arXiv preprint arXiv:2012.03110, 2020.

[17] T. Osakabe, M. Tanaka, Y. Kinoshita, and H. Kiya, "Cyclegan without checkerboard artifacts for counter-forensics of fake-image detection," in Int. Workshop Adv. Imag. Technol. 2021, vol. 11766. International Society for Optics and Photonics, Art. no. 1176609, 2021.

[18] Y. Huang et al., "FakePolisher: Making deepfakes more detectionevasive by shallow reconstruction," in Proc. 28th ACM Int. Conf. Multimedia, pp. 1217–1226, 2020.

[19] C. Liu, H. Chen, T. Zhu, J. Zhang, and W. Zhou, "Making deepfakes more spurious: Evading deep face forgery detection via trace removal attack," IEEE TDSC, 2023.

[20] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Interv., pp. 234–241, 2015.

[21] K. Chandrasegaran, N.-T. Tran, and N.-M. Cheung, "A closer look at Fourier spectrum discrepancies for CNN-generated images detection," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 7200–7209, 2021.

[22] U. Ojha, Y. Li, and Y. J. Lee, "Towards universal fake image detectors that generalize across generative models," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., pp. 24480–24489, 2023.

[23] Davide Alessandro Coccomini et al., "Detecting images generated by diffusers," arXiv preprint arXiv:2303.05275, 2023.

[24] Jiaxuan Chen, Jieteng Yao, and Li Niu., "A Single Simple Patch is All You Need for AI-generated Image Detection," arXiv preprint arXiv:2402.01123, 2024.

[25] J. Wu, Z. Huang, J. Thoma, D. Acharya, and L. Van Gool, "Wasserstein divergence for GANs," in Proc. Eur. Conf. Comput. Vis., pp. 653–668, 2018.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980, 2014.