Exploring the Adversarial Robustness of AI-generated Image Detectors

Thomas Lazzerini, Samuele Cappelletti, Martina D'Angelo
University of Trento

***Abstract*—Nowadays, there exists many image generators, so it is important to discriminate between synthetic images and real ones. In this regard, many detectors has been proposed but, at the same time, also many attacks to this detectors has been developed to interfere with their detection capabilities. In this report we present a general overview about some detection techniques and some recently developed attack techniques.**

I. Introduction

Synthetic images are now flooding the real world, from online dating sites to social media, fake profiles and scams are everywhere. While some images are funny and harmless, others could be exploited by malicious users [1]. In relation to this, in the image forensic field there is a continuous fight between *image detectors* and *adversarial attacks*. On one hand, the detectors try to discriminate synthetic images from real ones, while, on the other hand, the attacks try to trick the detectors by manipulating the images, both real and synthetic. In order to detect synthetic images, we have two main types of techniques: the *high-level forensic techniques* and the *low-level forensic techniques*. The former focuses on physical inconsistencies [2][3] and on repeated and uniform patterns [4], both of which are mostly visible to the human eye. While, the latter focuses on pixel-level artifacts, almost invisible to the human eye. Since image generators are becoming smarter every day and are able to avoid generating high-level artifacts, we prefer to rely on the low-level ones.

II. Detectors

In this section we provide an overview about some detection techniques. In particular, a smart use of *CLIP* to extract the feature vectors from the images [5] and some considerations about the generalization of *GAN* detectors over *Diffusion Models* [6].

A. CLIP-Based Detector

Many SoTA detectors perform well at detecting images generated by generators of the same family as the ones used during training. However, their performance drops significantly when dealing with images produced by different types of generators. For example, a detector trained on images from GAN-based generators is good with other GANs but struggles with images generated by diffusion-based models.

On the other hand, the CLIP-based detector proposed by [5] works well in detecting images generated by any type of generator, both with and without augmentations (e.g., cropping, resizing, compression, etc.). Thanks to the fact that the CLIP features achieve an excellent generalization and robustness even with a few examples (e.g., 1k or 10k), the CLIP-based has a significant performance improvement in the out-of-distribution scenario.

The CLIP-based method consists in collecting N real images with their corresponding captions and using these to generate N synthetic images. Then, CLIP is used to extract feature vectors from the $N + N$ images and feed them to a linear SVM classifier to perform the classification.

The evaluation of this CLIP-based method, reported in [5], suggests good generalization properties over different generator families (*GAN*, *Diffusion* and *Commercial Tools*), both in the the presence of post-processed images and not.

B. Detection of Images by Diffusion Models

Lately, *Diffusion Models* gained the spotlight in the image generation community, allowing for unmatched test-to-image photorealism and diversity. Most SoTA detectors exploits low-level artifacts, introduced during the generation phase by GAN generators. The study in [6] suggests that similar traces can be found also in DM-generated images.

The study in [6] also provides interesting evaluation results, comparing the performances of several SoTA detectors over different GAN and DM generators both in ideal case (uncompressed images) and real case (compressed and resized using the guidelines in [7]). These evaluations highlight how performances vary significantly between the models, due to the differences in their artifacts, therefore suggesting generalization difficulties (e.g., in classifying DM images with GAN training and vice versa). Despite these difficulties, the inclusion of DM generated images during training and the application of a calibration procedure, like the one suggested by [8], may help the generalization over similar architectures, despite not providing reliable results on out-of-training artifacts.

III. Attacks

Despite the powerful detectors at our disposal, there exists many users that aim at attacking such detectors, in order to hide traces of their forgeries or also to introduce traces typical of generated images, to disguise real content as synthetic. In this section we provide an overview of some newly developed attacking techniques.

A. White Black

Evaluating image detectors reveals their vulnerabilities to both white-box and black-box attacks, significantly reducing their effectiveness under various attach scenarios. SoTA classifiers are able to achieve almost perfect results in terms of AUC when trained on a single generator [9], yet remains susceptible to adversarial perturbations [10]. The attacks can be categorized into *white-box attacks*, where full access to the classifier's parameters is available, and *black-box attacks*, where only the classifier type is known.

The work in [10] introduced several attack types targeting the classifier in [9]. Among these, we can find *image-specific attacks*, which modify input images with perturbations, and *universal attacks*, which create a single adversarial perturbation applicable across various images. An interesting example regards the *black-box transfer attack*, which uses adversarial examples from a surrogate model to impair the performance of a more robust classifier.

The evaluation provided in [10] shows that these attack strategies are highly effective and can significantly reduce the AUC performance of the detector.

Insights into attack transferability [10] reveal that attacks effective on one model often struggle against others. Transferability is notably successful within the same family of detectors, such as CNN to CNN or CLIP to CLIP, but less so between different families (e.g., CNN and CLIP). While both CNN and CLIP models are vulnerable to white-box attacks, CLIP models demonstrate greater robustness, particularly against fake-to-real attacks. The low transferability of adversarial attacks suggests that distinct model architectures process images differently: CNN-based detectors focus on medium-to-high frequencies and isotropic spectra, while CLIP-based detectors rely on low-frequency patterns and cross-shaped spectra. This architectural divergence contributes to the limited effectiveness of attacks across model types, indicating that successful defenses must consider these fundamental differences in image processing.

The forger holds a strategic advantage, needing to devise only one successful attack, while the defender must guard against all potential threats. Notably, detectors trained on ImageNet [11] are particularly vulnerable; forensic classifiers require perturbations approximately ten times smaller than those needed to deceive ImageNet classifiers, possibly due to JPEG artifacts present in the training data [10]. Two effective defenses have emerged:

- Adversarial Training: This technique involves continuously training the classifier on adversarial examples generated from previous iterations, enhancing its robustness.
- Randomized Smoothing: This method adds significant Gaussian noise to each pixel, making it provably impossible for small perturbations to alter the classifier's output.

Forensic classifiers must integrate an adversarial model into their defenses that extends beyond standard techniques like recompression, resizing, blurring, or adding white noise. This comprehensive approach is essential for improving resilience against increasingly sophisticated attacks.

B. Mimicry attack against image splicing forensic

As stated in [12], this *mimicry adversarial attack* can be used to hide image manipulation while forcing the detector to detect arbitrary ones by applying a gradient based optimization approach. Applied at large scale, this would cause high false-alarms, producing an effect similar to *DoS* attacks while undermining the reliability of the target detector.

The attack strategy proposed in [12] involves splitting the image in uniform patches and use these to compute a target representation for both the *pristine patch* t_p, computed from the pristine patches, and the *forged patch* t_f, computed from the forged patches. The function used for computing such target representations needs to be defined for each detector for the attack to be effective, this due to the fact that different detectors exploit different features. Once the targets have been computed, a gradient-based iterative approach is applied the each patch of the manipulated image, in order to make the patch feature representation more similar to the respective target's feature representation. A visualization of such iterative approach can be seen in Fig. 1

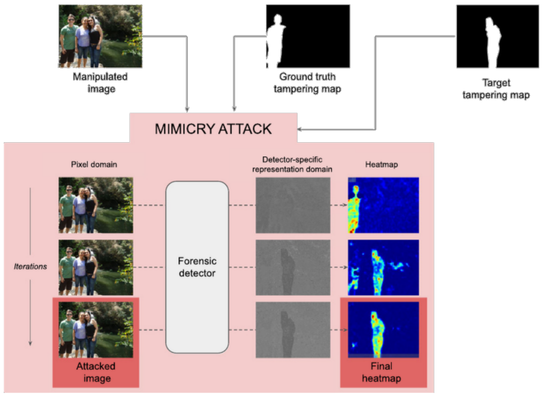

Fig. 1. Visualization of the mimicry attack strategy and its effects proposed in [12]. The "*ground truth*" tampering map represents the real forgery, while the "*target*" tampering map represent the arbitrary forgery the attacker wants the detector to output

The evaluation results reported in [12] suggests this attack is highly effective, both in hiding the real forgery and also highlighting a "decoy" forgery arbitrarily introduced. Two image detectors were tested, *Noiseprint* [13] and *EXIF-SC* [14], over two different datasets, *Columbia* [15] and *DSO-1* [16]. Several threshold-based and threshold-less metrics have been tested, with the latter being more important from the attacker point of view since the threshold values are unknown to him.

Another interesting result presented in [12] regards the *cross-detector* scenario, in which the attack is performed targeting a specific detector but then another is used in the evaluation. Also, *stacked attacks* are considered, in which an image is sequentially attacked against different detectors. An evaluation in these regards reveal mixed results: a misaligned attack in not effective, while the performances of a stacked attack are highly dependant on both the order of the attacks and the detector used in the evaluation. Nevertheless, this is an interesting scenario open for further studies.

C. *Image Laundering with Stable Diffusion*

Differently from ”classic” diffusion models, like *Latent Diffusion*, *Stable Diffusion* models allow the users to provide an initial image as input [17] [18] [19] [20]. This image will be superimposed with noise and modified by the model according to the textual prompt. The weight of such modifications can be set via a dedicated strength parameter in the range $[0, 1]$.

Processing images in such pipeline using a strength parameter equal to 0 produces outputs with the maximum similarity to the inputs: the image is encoded and decoded right away, without any denoising step. As suggested in [21], this process could be exploited by malicious users in order to mask real content as synthetic. In fact, the encoding/decoding is sufficient to introduce enough artifacts into the real image to make it synthetic in the eyes of numerous detectors. This practice is known as *image laundering*.

The study in [21], proposes a two-step architecture, visualized in Fig. 2, as solution to efficiently discriminate between real, fully synthetic and laundered images. Such architecture is inspired by [22], in which the image is split into multiple random patches, a score is assigned to each one and the average o the highest scores provide the global score of the image: a positive score suggest a synthetic image, while a negative score a real one. Despite the good results, this backbone architecture alone is unfit for the laundered image detection task, hence the introduction of the 2 steps: the first step discriminate real from synthetic images, while the second step discriminate fully synthetic from laundered images.

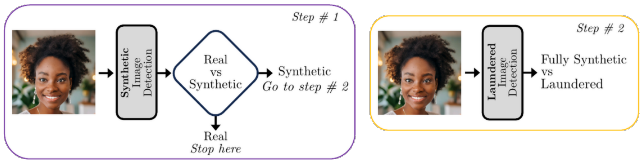
Step # 1
Synthetic Image Detection
Real vs Synthetic
Synthetic
Go to step # 2
Real
Stop here
Step # 2
Laundered Image Detection
Fully Synthetic vs Laundered

Fig. 2. Visualization of the 2 step architecture proposed in [21] for the laundered image classification task. The first step discriminate real images from synthetic (both fully and laundered) ones, while the second step discriminate fully synthetic images from laundered ones

The evaluation provided in [22] about such 2 step pipeline are good: the first step reach a good separability between real and synthetic image, while the second step reach almost perfect results over multiple models and multiple metrics, with only minor decreases in performances when post-processing operations like JPEG compression and resize are applied.

IV. Experiment

In this first phase of the project, our team executed a preliminary experiment, to asses the capability of the laundering attack, from the section III-C, on the dataset *TrueFake* provided by the *MMLAB* team.

The first phase of the experiment consisted in recovering 25 real images and, given their large size, extract 4 patches of size 1024×1024 from each of them, for a total of 100 real patches. Next, such patches were laundered, with a *denoising* parameter of 0, using the model *sd_xl_base_1.0* [23]. Lastly, a total of 100 fully synthetic images, generated by *Stable Diffusion XL*, were collected from the *TrueFake* dataset in equal quantity from each category available.

This small dataset was submitted to the 2 step pipeline from [21], visualized in Fig. 3, obtaining interesting results. The first step, as can be seen in Fig. 3, yielded similar results as [21], with a good separability around threshold 0. On the other hand, the second step, in Fig. 4, yielded results slightly different from [21], in particular the fully synthetic images had a average score of about 1, where [21] reported good separability at threshold 0.

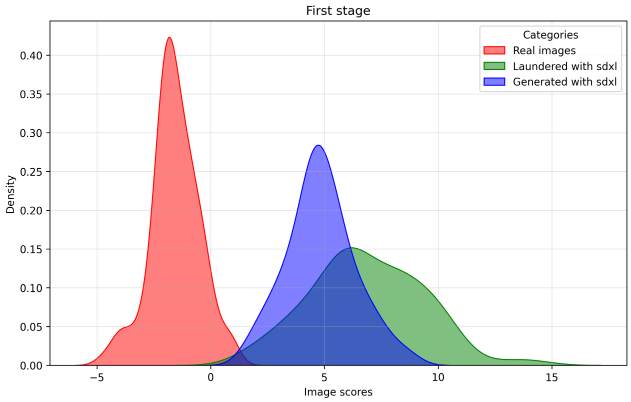

Fig. 3. Results of the first step of the pipeline from [21] using images from *MMLAB TrueFake* dataset

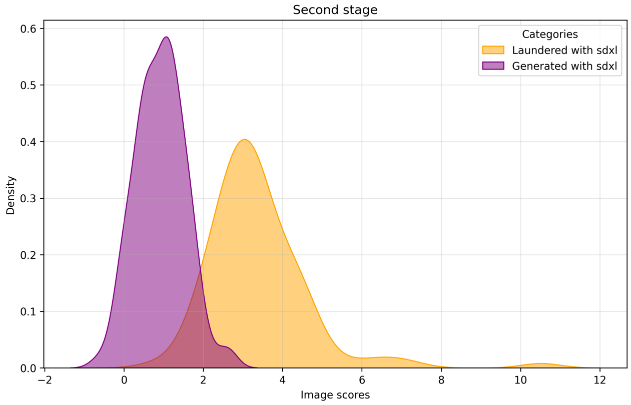

Fig. 4. Results of the second step of the pipeline from [21] using images from *MMLAB TrueFake* dataset

V. Conclusions

References

[1] N. Carlini and H. Farid, “Evading deepfake-image detectors with white- and black-box attacks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 658–659.

[2] H. Farid, “Lighting (in) consistency of paint by text,” *arXiv preprint arXiv:2207.13744*, 2022.

[3] H. Farid, "Perspective (in) consistency of paint by text," *arXiv preprint arXiv:2206.14617*, 2022.

[4] S. Mundra, G. J. A. Porcile, S. Marvaniya, J. R. Verbus, and H. Farid, "Exposing gan-generated profile photos from compact embeddings," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 884–892.

[5] D. Cozzolino, G. Poggi, R. Corvi, M. Nießner, and L. Verdoliva, "Raising the bar of ai-generated image detection with clip," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 4356–4366.

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

[7] R. Corvi, D. Cozzolino, K. Nagano, and L. Verdoliva, "IEEE Video and Image Processing Cup," https://grip-unina.github.io/vipcup2022/, 2022.

[8] J. Platt, "Probabilistic outputs for support vector machines and comparison to regularized likelihood methods," *Advances in Large Margin Classiers*, 1999.

[9] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "Cnn-generated images are surprisingly easy to spot... for now," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[10] V. De Rosa, F. Guillaro, G. Poggi, D. Cozzolino, and L. Verdoliva, "Exploring the adversarial robustness of clip for ai-generated image detection," *arXiv preprint arXiv:2407.19553*, 2024.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. I. Fei-Fei, "A large-scale hierarchical image database. computer vision and pattern recognition, 2009. cvpr 2009," in *IEEE Conference on*, pp. 248–255.

[12] G. Boato, F. G. De Natale, G. De Stefano, C. Pasquini, and F. Roli, "Adversarial mimicry attacks against image splicing forensics: An approach for jointly hiding manipulations and creating false detections," *Pattern Recognition Letters*, vol. 179, pp. 73–79, 2024.

[13] D. Cozzolino and L. Verdoliva, "Noiseprint: A cnn-based camera model fingerprint," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 144–159, 2019.

[14] M. Huh, A. Liu, A. Owens, and A. A. Efros, "Fighting fake news: Image splice detection via learned self-consistency," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 101–117.

[15] T.-T. Ng, S.-F. Chang, and Q. Sun, "A data set of authentic and spliced image blocks," *Columbia University, ADVENT Technical Report*, vol. 4, 2004.

[16] T. Carvalho, F. A. Faria, H. Pedrini, R. d. S. Torres, and A. Rocha, "Illuminant-based transformed spaces for image forensics," *IEEE transactions on information forensics and security*, vol. 11, no. 4, pp. 720–733, 2015.

[17] Computer Vision and Learning LMU Munich, *Stable Diffusion*, 2022 (accessed June 20, 2024), https://github.com/CompVis/stable-diffusion.

[18] S. AI, *Stable Diffusion Version 2*, 2022, https://github.com/Stability-AI/stablediffusion.

[19] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach, "Sdxl: Improving latent diffusion models for high-resolution image synthesis," *arXiv preprint arXiv:2307.01952*, 2023.

[20] A. Sauer, D. Lorenz, A. Blattmann, and R. Rombach, "Adversarial diffusion distillation," *arXiv preprint arXiv:2311.17042*, 2023.

[21] S. Mandelli, P. Bestagini, and S. Tubaro, "When synthetic traces hide real content: Analysis of stable diffusion image laundering," *arXiv preprint arXiv:2407.10736*, 2024.

[22] S. Mandelli, N. Bonettini, P. Bestagini, and S. Tubaro, "Detecting gan-generated images by orthogonal training of multiple cnns," in *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2022, pp. 3091–3095.

[23] stabilityai, *stable-diffusion-xl-base-1.0*, 2023 (accessed November 25, 2024), https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0.