Exploring the Adversarial Robustness of AI-generated Image Detectors

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

***Abstract*—TODO**

I. Introduction

Synthetic images are now flooding the real world. From online dating sites to social media, fake profiles and scams are everywhere. The problem with synthetic images is that, while some of them are funny and harmless, others could be harmful, they could be exploited by malicious users [1]. In relation to this, in the image forensic field there is a continuous fight between *fake image detectors* and *adversarial attacks*. On one hand, the detectors try to distinguish fake images from real ones, while, on the other hand, the attacks try to trick the detectors by manipulating the images (both real and fake ones). In order to detect fake images, we can exploit the traces/artifacts that fake image generators leave on the generated images. To do so, we have two main types of techniques: the *low-level forensic techniques* and the *high-level forensic techniques*. To former focuses on the pixel-level artifacts, which are almost invisible to the human eye. The latter focuses on physical inconsistencies and on repeated and uniform patterns, both of which are mostly visible to the human eye. Examples of physical inconsistencies are lighting, shadows, reflections or vanishing points inconsistencies [2][3]. While, an example of repeated and uniform patterns, typical of GAN-based image generators, is the generation of the mouth, the nose and the eyes always in the same position [4]. In general, we prefer to rely on low-level artifacts since fake image generators are becoming smarter every day, thus they are learning to generate always more realistic images, with fewer physical inconsistencies.

II. Detectors

In this section we will briefly describe a couple of fake image detectors: one uses CLIP to extract the feature vectors from the images [5] and one identifies the low-level traces/artifacts by training a GAN and a Diffusion Model [6].

A. CLIP-Based Detector

Many SoTA fake image detectors works very well in detecting fake images that are generated by an image generator of the same family of the generator that generated the images that they were trained on. But, the problem is that their performance decreases a lot when trying to detect images generated by another type of detector. For example, if the images used to train the detector were generated by a GAN-based generator, then the detector is good in detecting images generated by other GAN-based generators but, it is bad in detecting images generated by a Diffusion-based generator. Moreover, the SoTA detectors hardly generalize to new and unseen generative methods.

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.). Moreover, the performance of this CLIP-based method is similar to the one of the SoTA detectors in the in-distribution scenario but, it has a significant improvement in the out-of-distribution scenario. This is 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 method consists in: collect N real images $\{R_1, \dots, R_N\}$ with their corresponding captions $\{t_1, \dots, t_N\}$. Then, use these N captions to generate N images $\{F_1, \dots, F_N\}$ using some image generator $G(.)$, $F_i = G(t_i)$. Successively, use CLIP to extract the feature vectors $\{\mathbf{r}_1, \dots, \mathbf{r}_N\}$ and $\{\mathbf{f}_1, \dots, \mathbf{f}_N\}$ of the $N + N$ images (real + generated), $\mathbf{r}_i = CLIP(R_i)$ and $\mathbf{f}_i = CLIP(F_i)$. Finally, feed the $N + N$ feature vector to a linear SVM classifier.

In the paper they compared the results (in AUC) of the CLIP-based method with other SoTA detectors on images generated by different generators of different families: *GAN*, *Diffusion* and *Commercial Tools*, with and without post-processing on the images. In the latter, the performance of the CLIP-based method is the best one in average wrt. the performance of the other SoTA methods tested. In the case of post-processed images the results are slightly worse but, especially for the CLIP-based method, they are still good across all generators.

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. These new powerful tools are a new asset in the hands of malicious users, posing new challenges to the forensic community. Most SoTA detectors exploits low-level artifacts, not visible by a human eye, introduced during the generation phase by GAN generators. The study in [6] suggests that, as can be seen in Fig. 1, 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

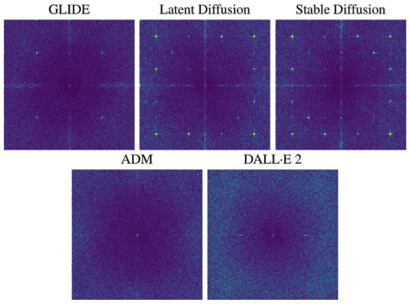

Fig. 1. Fourier transform of of the fingerprint of some DM architectures (*GLIDE* [7], *Latent Diffusion* [8], *Stable Diffusion* [9], *ADM* [10], *DALL·E 2* [11]) presented in [6]

different GAN and DM generators both in ideal case (uncompressed images) and real case (compressed and resized using the guidelines in [12]). These evaluations highlight how performances vary significantly between the models, due to the differences in their artifacts, therefore suggesting generalization difficulties (for example, in classifying a DM images with a GAN training and vice versa). Despite these difficulties, the inclusion of DM during training and performing an careful calibration procedure, like the one suggested by [13], 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 hide disguise content as fake. In the following chapter, some newly developed attacking techniques are discussed, to provide a general overview of the attacker-side.

A. Mimicry attack against image splicing forensic

As stated in [14], 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 [14] 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. 2

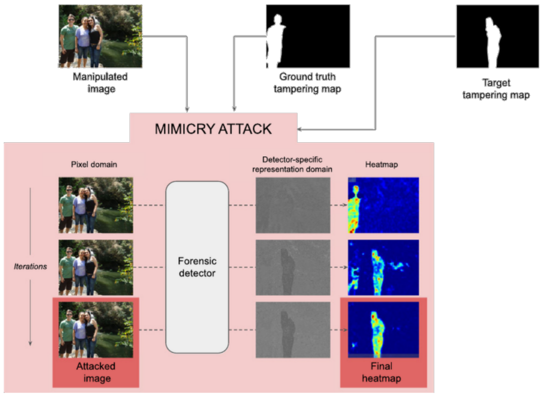

Fig. 2. Visualization of the mimicry attack strategy and its effects proposed in [14]. The "*ground truth*" tampering map represents the real forgery, while the "*target*" tampering map represent the arbitrary forgery the attacker wants the detector to output

The evaluation results reported in [14] 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* [15] and *EXIF-SC* [16], over two different datasets, *Columbia* [17] and *DSO-1* [18]. 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 [14] 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.

B. 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 [19] [20] [21] [22]. 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 [23], 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 [23], proposes a two-step architecture, visualized in Fig. 3, as solution to efficiently discriminate between real, fully synthetic and laundered images. Such architecture is inspired by [24], 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.

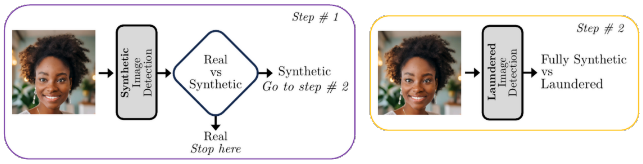
Step # 1
Synthetic Image Detection
Real vs Synthetic
Synthetic
Go to step # 2
Real
Stop here
Step # 2
Laundered Image Detection
Fully Synthetic vs Laundered

Fig. 3. Visualization of the 2 step architecture proposed in [23] 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 [24] 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.

C. White Black

Evaluating deepfake-image detectors reveals their vulnerabilities to both white- and black-box attacks, significantly reducing their effectiveness. For instance, powerful forensic classifiers can be compromised to achieve near 0% accuracy under various attack scenarios. A state-of-the-art classifier, as demonstrated by Wang et al. [25], achieves an area under the ROC curve (AUC) of 0.95 when trained on a single generator, yet remains susceptible to adversarial perturbations [26]. The attacks are categorized into two conditions:

- White-box attacks, where full access to the classifier's parameters is available.
- Black-box attacks, where only the classifier type is known, utilizing adversarial examples that transfer misclassifications across models

They introduce four attack types targeting the classifier from Wang et al. [25]:

- Image-specific attacks: These modify input images with perturbations to deceive detectors (eg. PGD, DI2-FGSM)
  - Loss-Maximizing Attack: this attack maximizes the likelihood that a fake image x perturbed by δ is misclassified as real. This attack was found to be highly effective and even with flipping the lowest-order bit of 40% of pixels for uncompressed images, the AUC reduces from 0.966 to 0.27.
- Universal Attacks: These create a single adversarial perturbation applicable across various images, reducing computational costs.
  - Universal Adversarial-Patch Attack: A universal patch overlaid on images reduces AUC from 0.966 to 0.085.
  - Universal Latent-Space Attack: Instead of altering images directly, this method modifies low-level attributes of the generative model, resulting in an AUC drop from 0.99 to 0.17.
- Black-Box Transfer Attack: This approach uses adversarial examples from a surrogate model to impair a more robust classifier's performance. By transferring adversarial examples they develop their own forensic classifier and trasfer the attack to the target classifier [25], the AUC is reduced from 0.96 to 0.22.

Insights into attack transferability [27] 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 [28] 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 [26]. 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.

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-B, 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* [29]. 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 [23], visualized in Fig. 4, obtaining interesting results. The first step, as can be seen in Fig. 4, yielded similar results as [23], with a good separability around threshold 0. On the other hand, the second step, in Fig. 5, yielded results slightly different from [23], in particular the fully synthetic images had a average score of about 1, where [23] reported good separability at threshold 0.

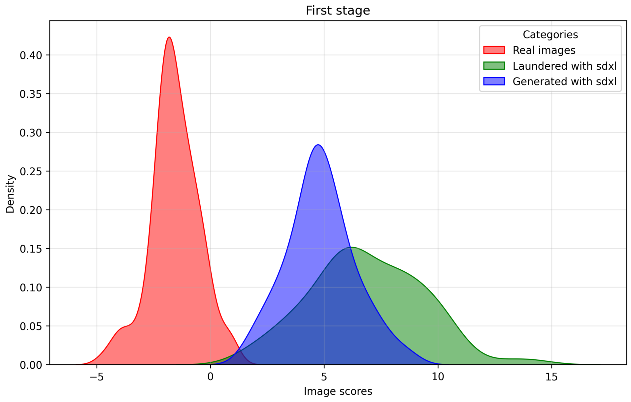

Fig. 4. Results of the first step of the pipeline from [23] using images from *MMLAB TrueFake* dataset

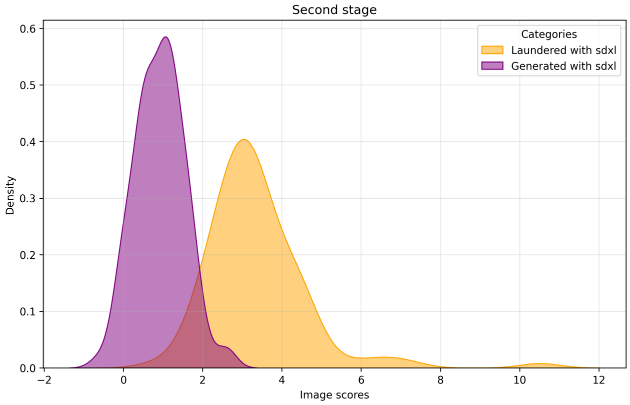

Fig. 5. Results of the second step of the pipeline from [23] using images from *MMLAB TrueFake* dataset

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] 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.

[8] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *CVPR*, 2022, pp. 10 684–10 695.

[9] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "Stable diffusion," https://github.com/CompVis/stable-diffusion, 2022.

[10] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780–8794, 2021.

[11] 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.

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

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

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

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

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

[21] 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.

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

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

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