AI-GenBench: A New Ongoing Benchmark for AI-Generated Image Detection

AI-GenBench: A New Ongoing Benchmark for AI-Generated Image Detection

Lorenzo Pellegrini*§, Davide Cozzolino†, Serafino Pandolfini*, Davide Maltoni*,
Matteo Ferrara*, Luisa Verdoliva†, Marco Prati‡, Marco Ramilli‡

*Dipartimento di Informatica - Scienza e Ingegneria (DISI)
Università di Bologna, Cesena, Italy

†Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione (DIETI)
Università degli Studi di Napoli Federico II, Naples, Italy

‡IdentifAI, Italy

***Abstract*—The rapid advancement of generative AI has revolutionized image creation, enabling high-quality synthesis from text prompts while raising critical challenges for media authenticity. We present AI-GenBench, a novel benchmark designed to address the urgent need for robust detection of AI-generated images in real-world scenarios. Unlike existing solutions that evaluate models on static datasets, AI-GenBench introduces a temporal evaluation framework where detection methods are incrementally trained on synthetic images, historically ordered by their generative models, to test their ability to generalize to new generative models, such as the transition from GANs to diffusion models. Our benchmark focuses on high-quality, diverse visual content and overcomes key limitations of current approaches, including arbitrary dataset splits, unfair comparisons, and excessive computational demands. AI-GenBench provides a comprehensive dataset, a standardized evaluation protocol, and accessible tools for both researchers and non-experts (e.g., journalists, fact-checkers), ensuring reproducibility while maintaining practical training requirements. By establishing clear evaluation rules and controlled augmentation strategies, AI-GenBench enables meaningful comparison of detection methods and scalable solutions. Code and data are publicly available to ensure reproducibility and to support the development of robust forensic detectors to keep pace with the rise of new synthetic generators[1]**

***Index Terms*—AI-generated image detection, Generative models, Forensic benchmark.**

I. Introduction

In recent years, the field of image generation has experienced rapid progress, marked by the development of robust and flexible tools based on diffusion models. These tools are capable of producing high-quality images from general conditional inputs, such as text, enabling professionals to leverage AI for innovative and creative applications in design, marketing, and entertainment. However, the potential misuse of these technologies raises significant ethical and social concerns, including the dissemination of disinformation and infringements of intellectual property rights [1]–[3]. Consequently, there is an urgent need to develop effective methods for distinguishing real images from those generated by AI.

Numerous methods have been proposed for the task of distinguishing AI-generated images from real ones. To understand their generalization ability, these methods are often trained on images from a single generator and then tested on images from other generators. However, a more realistic scenario involves evaluating generalization in an online setting, where the model is trained incrementally by incorporating new generators while preserving the historical order of their release dates [6]. To address this, we present AI-GenBench, a novel benchmark designed to evaluate and advance models capable of distinguishing AI-generated images from real ones. To ensure research reproducibility and foster innovation in the field, we will make datasets and code publicly available, and propose an evaluation protocol that facilitates the validation of new models. A key objective is to provide accessible and user-friendly tools for nonprofessionals, such as journalists, investigators, and content moderators, enabling them to easily assess the authenticity of digital information.

AI-GenBench focuses on high-quality, realistic images, such as those frequently shared or published on social networks, while deliberately excluding non-realistic content such as drawings, cartoons, low-resolution or noise-corrupted images. Unlike many existing benchmarks [4], [8], AI-GenBench is not limited to faces or human subjects but encompasses a broad range of visual content, thereby reflecting the diversity of real-world applications. AI-GenBench is not just another benchmark; it is specifically designed to address the critical limitations of existing benchmarks and datasets in the field. These limitations include arbitrary training and validation splits, which can result in biased or unreliable outcomes, unfair comparisons between methods due to inconsistent evaluation protocols, and high computational resource demand. These limitations will be discussed in detail in the following sections.

Identifying images from known generators (i.e., those used during training) is relatively straightforward. However, the

§Corresponding author: Lorenzo Pellegrini (l.pellegrini@unibo.it)

[1]https://github.com/MI-BioLab/AI-GenBench.

TABLE I: SoTA Forensic Benchmarks.

Year	Acronym	Content	Test data		Training data		Available Online	Temporal ordering
			#Real / #Fake	#gen.	#Real / #Fake	#gen.		
2021	[4] ForgeryNet	Face	290K / 290K	15	1.2M / 1.2M	15	✔	✘
2023	[5] DeepArt	Artworks	2.3K / 2.4K	5	62K / 71K	5	✘	✘
2023	[6] Epstein2023	General	22K / 116K	14	202K / 454K	14	✘	✔
2024	[7] SIDBench	General	46K / 52K	16	360K / 360K	1	✔	✘
2024	[8] DiffusionFace	Face	6K / 120K	11	24K / 480K	11	✔	✘
2024	[9] DiFF	Face	2.4K / 54K	13	484K / 21K	13	✔	✘
2024	[10] GenImage	Object	50K / 400K	8	1.3M / 10.4M	8	✔	✘
2024	[11] Park2024	General	40K / 71K	23	560K / 560K	2	✘	✘
2024	[12] ImagiNet	General	20K / 20K	8	80K / 80K	8	✔	✘
2024	[13] WildFake	General	203K / 536K	23	811K / 2.1M	23	✘	✘
2024	[14] Fake2M	General	139K / 308K	11	1M / 2.3M	3	✔	✘
2025	AI-GenBench (Ours)	General	36K / 36K	36	144K / 144K	36	✔	✔

real challenge lies in generalizing to images produced by unseen generators. A fundamental aspect of AI-GenBench is its temporal evaluation framework, where models are trained on generators from an earlier time period and validated on generators from a subsequent period. This methodology assesses the model's ability to generalize to novel generation techniques, such as the transition from GANs to diffusion models. The process is repeated by shifting the time intervals, thereby ensuring continuous adaptation to evolving models. Although this idea was introduced by Adobe researchers in [6], there is no established benchmark for the community. Figure **??** provides a conceptual overview of the benchmark, illustrating how detection models are progressively updated and evaluated as new generators emerge. This process is explained in detail in subsequent sections.

To identify the most effective deepfake detection methods, it is essential to establish clear and fair rules and to train these methods using identical datasets and augmentation strategies. The latter can significantly affect model performance, particularly in realistic scenarios; thus, it is important that the same amount of augmentation is applied across all tested models. Furthermore, the training set is designed with scalability in mind, ensuring that a model of moderate complexity can be trained on a workstation equipped with a recent GPU in about 24 hours. This approach minimizes resource demands and enables participants with limited hardware capabilities to compete on an equal footing. In real-world applications, we expect that the best-performing models can be scaled up by retraining them on larger datasets prior to deployment.

Overall, we propose a benchmark designed to enhance the detection of AI-generated content by providing a robust, fair, and scalable framework for evaluation. By addressing the limitations of existing benchmarks, our benchmark enables researchers to develop more effective and generalizable models. Additionally, nonprofessionals will have access to simple yet effective tools for analyzing the authenticity of digital media.

II. Related work

In recent years, several benchmarks have been proposed for the detection of AI-generated images, accompanied by a variety of forensic detectors and datasets. In the following we will review the most relevant ones.

a) Benchmarks: Table I presents a comprehensive list of benchmarks for synthetic image detection. Some initial benchmarks only focus on GAN-based synthetic generators [4] and are primarily concerned with face images, such as ForgeryNet [4], DiffusionFace [8], and DIFF [9], or on artworks [5]. Although these analyses offer valuable insights, their applicability is currently restricted to specific image categories.

Several generic benchmarks have been introduced with the goal of providing large-scale and diverse training datasets to enhance model generalization [10], [12], [13], or offering open-source frameworks to facilitate the integration of new models [7]. Others benchmarks focus on conducting extensive analyses of publicly available datasets [11], including evaluations of human performance in detecting synthetic images [14]. Additionally, several papers, while not explicitly presenting benchmarks, propose datasets that are widely adopted in the forensics community, including both GAN-based [15] and diffusion-based images [16]–[19].

The experimental analysis conducted in the aforementioned benchmarks primarily focus on studying the generalization ability by including synthetic generators during testing that were not present during training. However, in realistic scenarios, new generators are released continuously. Therefore, it is essential to investigate generalization ability under these conditions, i.e., training on older generators and testing on newer ones based on the historical release dates of the generation models. This type of analysis was first conducted in [6], revealing a significant decrease in performance when generators with major architectural changes emerge. While this work demonstrated the value of such analysis, it did not establish a standardized benchmark for the research community, which is a critical gap that our work directly addresses. Furthermore, the analysis in [6] is limited to 14 generative models, while

we expand this scope by incorporating 36 distinct generative models.

b) Detection Methods: Initial approaches for distinguishing synthetic from real images primarily relied on CNN-based architectures trained on large datasets [3]. These methods demonstrate strong performance when test conditions perfectly match the training distribution. However, they suffer from two main limitations in practical scenarios: robustness to common image impairments, such as re-compression, resizing, or cropping that frequently occur during online sharing, and generalization to unseen generative architectures [20]. To enhance robustness, a golden rule is to include carefully designed data augmentation during training. This approach not only strengthens the ability to handle image distortions but also improves generalization capabilities [15]. Relying on pre-trained models is also very effective, especially if large pre-trained vision-language models are used. In this respect, it is possible to achieve very good performance even by relying on one single generator during training [16].

Some methods propose to modify the architectures in order to better exploit low-level and/or high-level forensic traces [21], [22] while others focus on improving the training strategy [12], [23] or by simulating the generator artifacts [24], [25]. Another path towards improving generalization is the use of few-shot or incremental learning strategies, as done in [26]–[28]. These are interesting approaches, but require some images from the new models, which may not be available in the most challenging scenarios. Instead, we think that a more practical and realistic framework is to regularly re-train the detector by preserving the temporal order of the synthetic generator release date [6]. This approach allows us to fully leverage the forensic traces of known synthetic generators which will very likely be similar to the newer generators. Indeed, it is reasonable to believe that artificial fingerprints [29] from one generator likely enable classifiers to generalize across entire families of models, not just individual ones [15]. We will experimentally show that this is the case and that major changes in the underlying generative architecture can be an issue only as soon as a completely new model is released.

III. The Benchmark

In this section, we present the design and implementation of AI-GenBench, a novel benchmark for evaluating synthetic media detection methods. Our framework consists of two key components: a temporally ordered dataset, organized by the chronological release dates of generative models, and a standardized evaluation and training protocol that ensures fair and reproducible comparisons. The dataset allows us to assess the generalization ability of classifiers to detect new, unseen architectures under the same training/validation split, controlled augmentation strategies, and computational constraints to simulate practical deployment conditions.

A. Dataset

The dataset comprises 180K synthetic images generated by 36 different generators, each contributing 5K images.

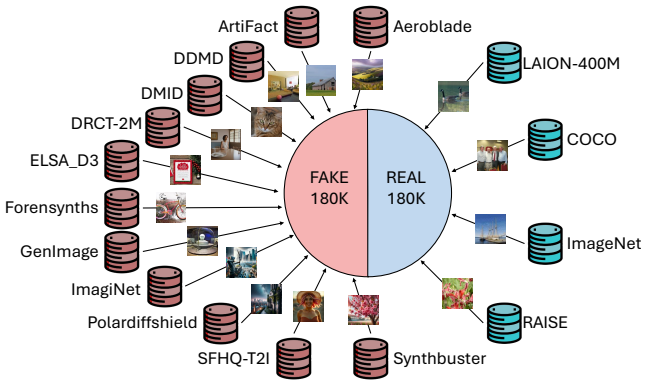

Fig. 1: A scheme of the content of the proposed dataset. Different data repositories of both synthetic and real images are merged to create a unified dataset containing images created using the major generative methods of the last seven years.

Additionally, the dataset includes 180K real images from four different sources: ImageNet (ILSVRC2012) [30], COCO2017 [31], LAION-400M [32], and RAISE [33]. To maximize diversity, synthetic images are obtained from various synthetic image repositories, listed alphabetically: Aeroblade [34], ArtiFact [35], Towards the Detection of Diffusion Model Deepfakes (DDMD) [36], Diffusion Model Image Detection (DMID) [17], DRCT-2M [37], ELSA_D3 [23], ForenSynths [15], SFHQ-T2I [38], GenImage [10], ImagiNet [12], Polardiffshield [39], and Synthbuster [18]. A general overview of the dataset and the included repositories is presented in Figure 1.

We selected these data sources because they are publicly available and have been widely used in scientific research. For each generator, our dataset includes images taken from multiple sources, as these feature overlapping sets of generators. Although the original repositories have different structures and metadata, our dataset presents a unified structure (using the Arrow format). To facilitate the setup of the dataset, we provide the code used to download the files from the various sources and organize them into a common format.

The complete list of the 36 synthetic image generators $g_i, i = 0..35$ included in our dataset is reported in Table II. The dataset is divided into training and evaluation sets, following an $80\% - 20\%$ proportion, resulting in 288K images for the training set and 72K images for the evaluation set. In particular, for each generator, 4K images are used for training and 1K for evaluation. Since the dataset is balanced, real images are also split according to the same $80\% - 20\%$ proportion.

B. Evaluation and metrics

To evaluate the performance of different detection methods, we assign the 36 generators to nine chronological sliding windows, denoted $w_j, j = 0..8$, each containing four generators $w_j = \{g_{j\times 4+t}, 0 \leq t \leq 3\}$. The deepfake detection methods are trained progressively: at each step k, the model

TABLE II: Full list of generators included in the benchmark, ordered by release date. The content of each chronological sliding window is highlighted.

w_j	g_i	Generator	Release date
0	0	CycleGAN [40]	2017-03
	1	Cascaded Refinement Networks [41]	2017-07
	2	ProGAN [42]	2017-10
	3	StarGAN [43]	2017-11
1	4	SN-PatchGAN [44]	2018-06
	5	BigGAN [45]	2018-09
	6	IMLE [46]	2018-11
	7	StyleGAN1 [47]	2018-12
2	8	GauGAN [48]	2019-03
	9	StyleGAN2 [49]	2019-12
	10	DDPM [50]	2020-06
	11	CIPS [51]	2020-11
3	12	VQGAN [52]	2020-12
	13	GANsformer [53]	2021-03
	14	ADM [54]	2021-05
	15	StyleGAN3 [55]	2021-06
4	16	LaMa [56]	2021-09
	17	FaceSynthetics [57]	2021-09
	18	ProjectedGAN [58]	2021-11
	19	Palette [59]	2021-11
5	20	VQ-Diffusion [60]	2021-11
	21	Denoising Diffusion GAN [61]	2021-12
	22	Glide [62]	2021-12
	23	Latent Diffusion [63]	2021-12
6	24	Midjourney [64]	2022-02
	25	MAT [65]	2022-03
	26	Diffusion GAN (ProjectedGAN) [66]	2022-06
	27	Diffusion GAN (StyleGAN2) [66]	2022-06
7	28	Stable Diffusion 1.4 [63]	2022-08
	29	Stable Diffusion 1.5 [63]	2022-10
	30	Stable Diffusion 2.1 [63]	2022-12
	31	DeepFloyd IF [67]	2023-04
8	32	Stable Diffusion XL 1.0 [68]	2023-07
	33	DALL-E 3 [69]	2023-09
	34	FLUX 1 Dev [70]	2024-08
	35	FLUX 1 Schnell [70]	2024-08

is trained on all the generators within the sliding windows $w_j, j \leq k$. This approach simulates a realistic scenario where detectors are periodically retrained to keep into account the latest technological advances. As new generators are published, the model can learn from all existing generators while future ones remain unseen. The detector's ability to detect images from past and future generators is evaluated using various metrics. Given a detector trained at step k and a set of performance indicators (e.g., accuracy, AUROC) we can evaluate its performance on different subsets of evaluation images:

- *Next Period*: the evaluation set includes only the generators belonging to w_{k+1}. This metric is particularly important as it measures the detector's ability to generalize to unseen generators, which will become available in the near future.
- *Past Period*: the evaluation set includes the generators belonging to $w_j, j \leq k$. This scenario evaluates how well the detector performs on the generators it has already encountered.
- *Whole Period*: the evaluation set includes the generators belonging to both the past and next time windows ($w_j, j \leq k+1$).

Algorithm 1 AI-GenBench training and evaluation workflow. In the current version, the parameters ts, am, and n are set to 9, 4, and 1, respectively.

1: **Inputs:**
2: T_j - fake and real training images of sliding win. w_j
3: E_j - fake and real evaluation images of sliding win. w_j
4: ts - number of training steps
5: am - augmentation multiplier for training
6: n - number of epochs
7: **Body:**
8: $TS \leftarrow \{\}$ ▷ Training Set
9: $ES \leftarrow DetermAugment(E_0, 1)$ ▷ Evaluation Set
10: **for** $k = 0$ **to** $ts - 1$ **do** ▷ k-th training step
11: $TA \leftarrow CustomAugment(T_k, am)$ ▷ Augm. Train
12: $TS \leftarrow TS \cup TA$ ▷ Current Training Set
13: $m_k \leftarrow$ Model trained for n epochs on TS
14: Evaluate model m_k on ES ▷ *Past Period*
15: **if** $k < ts - 1$ **then**
16: $EN \leftarrow DetermAugment(E_{k+1}, 1)$ ▷ Next Eval.
17: Evaluate model m_k on EN ▷ *Next Period*
18: $ES \leftarrow ES \cup EN$ ▷ Whole Eval. Set
19: Evaluate model m_k on ES ▷ *Whole Period*
20: **end if**
21: **end for**

For all the above metrics, a plot can be drawn (e.g. Figures 2, 3 and 4) showing the performance trend across the sliding windows. As compact ranking indicators for the leaderboard, we propose using the average Area Under Receiver Operating Characteristic (AUROC) curve and the average accuracy (across all steps) in the *Next Period* scenario. These indicators are chosen because they provide valuable insights into the detector's ability to generalize to unseen generators that are being released.

C. Training and evaluation details

To ensure a fair comparison among methods, a standardized training and evaluation workflow has been defined. The workflow pseudocode is reported in Algorithm 1. In the current version of the benchmark, T_j includes 4K train images from each of the 4 generators associated to w_j (see Table II) and 4K real train images. To this purpose, the whole set of real images is initially partitioned in groups and each group is associated to a generator. Analogously, E_j includes 1K evalution images from each of the generators associated to w_j and 1K real evaluation images. To train and evaluate a new method on AI-GenBench, the only steps that need modification are the custom training augmentation (line 11) and the method training (line 13). Hereafter, more details are provided.

1) Augmentation: the training augmentation pipeline (function $CustomAugment$) can be customized by the method itself, as this is often a key distinguishing feature. However, the number of (diverse) augmented images presented to the model is fixed and determined by the multiplier am. In this version of the benchmark, we use $am = 4$, meaning that each

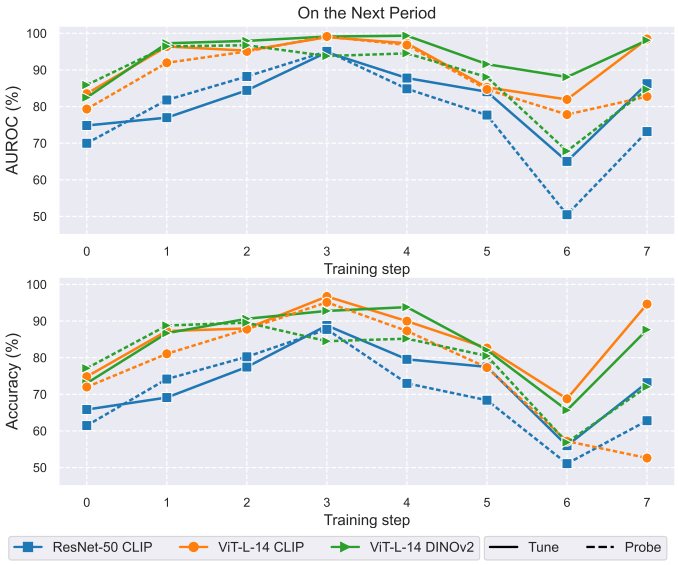

Fig. 2: AUROC and Accuracy for different models and training strategies on generators in the *Next Period* on resized images.

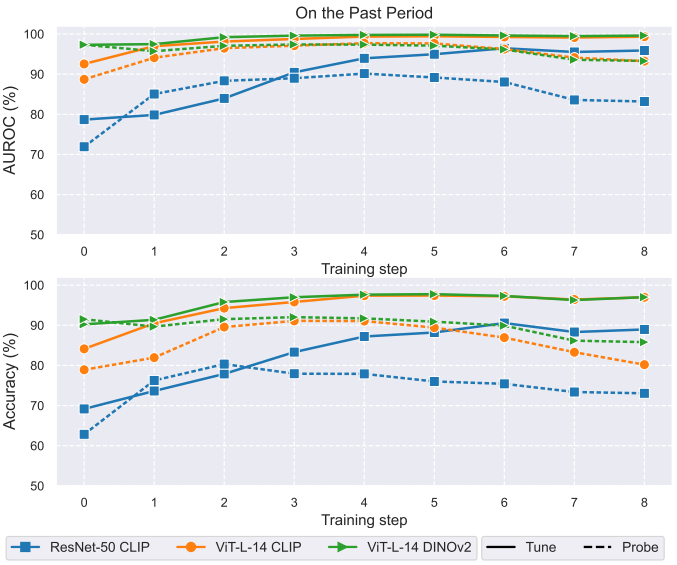

Fig. 3: AUROC and Accuracy for different models and training strategies on generators in the *Past Period* on resized images.

training image is replaced with four augmented versions of itself. This rule is enforced to ensure a fair evaluation with respect to the computational resources: by limiting the amount of augmentation allowed during training, the focus is placed on the effectiveness of the detection method itself. Without this limitation, methods relying on more computational resources could achieve better results by leveraging more extensive augmentations. In contrast, the evaluation augmentation (function $DetermAugment$) is deterministic and cannot be customized (here the multiplier is 1). The goal of the augmentation pipeline is to produce images that closely resemble those typically published on media platforms or shared via instant messaging applications. The pipeline includes a combination of compression, blurring, noise addition, variable resizing, and cropping steps. These transformations disrupt most of the noise patterns found in generated images while still producing good quality images that could reasonably be accepted and re-shared by human users.

2) Additional data and pretrained models: training on additional data or using models pre-trained on the same problem is not allowed. However, general-purpose open-weight foundation models can be used as backbones.

3) Supporting codebase, leaderboard and reproducibility: a codebase will be released to facilitate the download and setup of the dataset, as well as the execution of the training and evaluation workflow described in Algorithm 1. A plug-in architecture has been designed to simplify the addition of new methods through the customization of a few functions. The codebase also includes baseline methods for comparison and a default training augmentation pipeline. A leaderboard will be maintained for each version of the benchmark. Methods added to the leaderboard must include an accompanying white paper and scripts, ensuring full reproducibility of the results.

TABLE III: Comparison of the average AUROC and Accuracy for Resize and Multicrop Experiments on the Next Period.

Model	Mode	AUROC		Accuracy	
		Resize	Multicrop	Resize	Multicrop
ResNet-50 CLIP	Tune	81.77	76.51	73.42	67.22
ResNet-50 CLIP	Probe	77.66	69.08	69.85	63.33
ViT-L/14 CLIP	Tune	92.04	87.66	**85.28**	76.83
ViT-L/14 CLIP	Probe	88.47	68.74	76.25	63.02
ViT-L/14 DINOv2	Tune	**94.24**	**93.44**	84.09	**78.64**
ViT-L/14 DINOv2	Probe	88.47	88.74	79.34	78.09

IV. Performance of Baseline Methods

We trained and evaluated a set of baseline methods using the workflow described in Algorithm 1. These methods are based on well-known pre-trained vision models: i) *ResNet-50 CLIP* by OpenAI [71], ii) *ViT-L/14 CLIP* from LAION models[2], and iii) *ViT-L/14 DINOv2* [72]. The input size for all the models is 224x224. For each model, we explored two distinct training strategies: *fine-tuning* and *linear probing*. The fine-tuning strategy (hereafter referred to as *tune*) involves training the entire model, while the linear probing strategy (hereafter referred to as *probe*) involves training only the final classification layer built on top of a frozen backbone. In both cases, at each training step, the models have been trained for a single epoch ($n = 1$).

Additionally, we investigated two approaches for converting images of variable sizes into the fixed-size input images required by the used models: i) *resize*, where images are resized to a fixed size during both training and evaluation; and ii) *multi-cropping*, where a single random crop is used during training, while five crops (the center and the four corners) are employed during evaluation, with their predictions

[2]laion/CLIP-ViT-L-14-CommonPool.XL-s13B-b90K

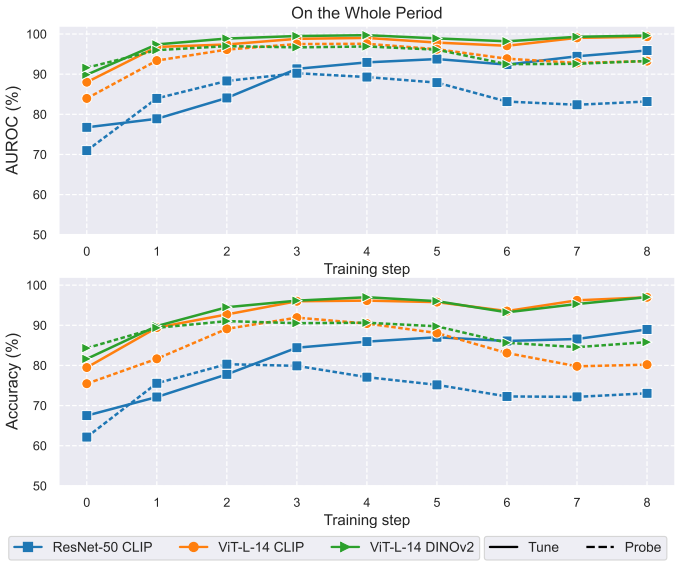

Fig. 4: AUROC and Accuracy for different models and training strategies on generators in the *Whole Period* on resized images.

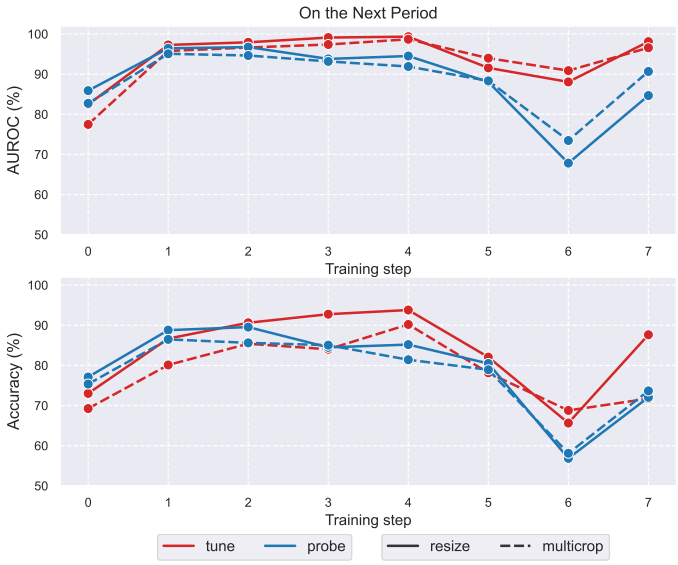

Fig. 5: AUROC and Accuracy for the *ViT-L/14 DINOv2* variants on generators in the *Next Period*.

aggregated via averaging. The resize strategy better preserves the semantic content of the images, at the cost of altered resolution and aspect ratio, which may attenuate or remove artifacts introduced by generative methods. In contrast, the multi-cropping strategy analyzes the images at their original resolution, at the cost of using only partial content information. This results in a total of 12 experiments: 3 models $\times$ 2 training modes $\times$ 2 image adaptation mechanisms. At the start of the training process (line 13 of Algorithm 1), for each step $k > 0$, we reload the weights from the model trained on the previous step m_{k-1}, rather than reloading weights from the original pre-training. This approach simulates an incremental improvement of the detection model over time.

Figures 2, 3, and 4 show the AUROC and Accuracy trends when using image resizing for *Next Period*, *Past Period* and *Whole Period* scenarios, respectively. It is evident that *Next Period* scenario is more challenging since detectors are evaluated only on unseen generators and technological changes can determine significant accuracy drops. On the contrary, in the *Whole Period* scenario, the increasing weight of known generation techniques leads to a more stable improvement.

To be more specific, in Figure 2, two noticeable performance drops are observed. A significant drop occurs at step 6 when detectors trained on $w_i, i = 0 \ldots 6$ are assessed on w_7. Referring to Table II, this drop coincides with the model's first evaluation on the *Stable Diffusion (1.4-2.1 + DF-IF)* generators, which is a significant milestone in the evolution of image generation techniques. A minor drop is observed at step 4 (training on $w_i, i = 0 \ldots 4$ and evaluation on w_5), corresponding to the introduction of *VQ-Diffusion*, *Denoising Diffusion GAN*, *Glide*, and *Latent Diffusion*, which are precursors to the next generation of robust diffusion techniques.

If the focus is moved to the comparison of the different baseline methods, besides Figure 2 and 4 we can consider Table III reporting the average AUROC and Accuracy. We observe that: i) larger models outperform smaller ones; ii) fine-tuning the entire backbone consistently leads to better generalization compared to freezing it during training; iii) resize is often preferable with respect to multi-crop. Specific details for *ViT-L/14 DINOv2* variants are provided in Figure 5.

Table IV reports the training times of the baseline methods on a workstation with a single GPU, showing that the entire training process (following the Algorithm 1 workflow) of the most resource-demanding model can be completed in about 1 day. For smaller models, such as the ResNet-50, data loading and augmentation (executed on the CPU) create a performance bottleneck, resulting in equal times for tune and probe.

TABLE IV: Training times of baselines according to Algorithm 1. The reference system is equipped with an *Intel i9-10900X* CPU and a *NVIDIA GeForce RTX 3080 Ti* GPU.

Model	Mode	Training Time
ResNet-50 CLIP	Tune	4.87h
ResNet-50 CLIP	Probe	4.87h
ViT-L/14 CLIP	Tune	19.87h
ViT-L/14 CLIP	Probe	6.35h
ViT-L/14 DINOv2	Tune	26h
ViT-L/14 DINOv2	Probe	7.5h

V. Conclusions

AI-GenBench is introduced as a novel, ongoing benchmark for detecting AI-generated images in real-world scenarios. This benchmark will provide datasets, tools, and leaderboards to researchers and practitioners, facilitating the training, evaluation, and fair comparison of their detection methods. As we write this paper, new generation techniques are rapidly gaining popularity, and we plan to incorporate these techniques

in future versions of the benchmark, including additional temporal sliding windows to keep the evaluation up-to-date with the latest technological advancements. In the future, we aim to consider local alteration and generation techniques, such as image inpainting, and extend the evaluation framework to include videos. The baseline methods presented in this paper are based on simplified design choices, which we intend to reconsider and refine. This reassessment will help determine if these design choices can benefit existing detectors, thereby distilling guidelines that may support the development of novel approaches.

Acknowledgment

We acknowledge, for the first author, the support of the European funds from the Emilia-Romagna Region under the Fse+ 2021-2027 programme. In addition, this work has received funding from the European Union under the Horizon Europe vera.ai project, Grant Agreement number 101070093, and was partially supported by SERICS (PE00000014) under the MUR National Recovery and Resilience Plan, funded by the European Union - NextGenerationEU.

References

[1] Z. Epstein, A. Hertzmann, L. Herman, *et al.*, "Art and the science of generative AI: A deeper dive," *Science*, vol. 380, 2023.

[2] C. Barrett, B. Boyd, E. Burzstein, *et al.*, *Identifying and Mitigating the Security Risks of Generative AI*. Now Foundations and Trends, 2024.

[3] L. Lin, N. Gupta, Y. Zhang, *et al.*, "Detecting multimedia generated by large AI models: A survey," *arXiv preprint arXiv:2204.06125*, 2024.

[4] Y. He, B. Gan, S. Chen, *et al.*, "ForgeryNet: A Versatile Benchmark for Comprehensive Forgery Analysis," in *CVPR*, 2021, pp. 4360–4369.

[5] Y. Wang, Z. Huang, and X. Hong, "Benchmarking deepart detection," *arXiv preprint arXiv:2302.14475*, 2023.

[6] D. C. Epstein, I. Jain, O. Wang, and R. Zhang, "Online Detection of AI-Generated Images," in *ICCV Workshops*, Oct. 2023, pp. 382–392.

[7] M. Schinas and S. Papadopoulos, "SIDBench: A Python framework for reliably assessing synthetic image detection methods," in *ACM International Workshop on Multimedia AI against Disinformation*, 2024, pp. 55–64.

[8] Z. Chen, K. Sun, Z. Zhou, *et al.*, "DiffusionFace: Towards a comprehensive dataset for diffusion-based face forgery analysis," *arXiv preprint arXiv:2403.18471*, 2024.

[9] H. Cheng, Y. Guo, T. Wang, L. Nie, and M. Kankanhalli, "Diffusion facial forgery detection," in *ACM Multimedia*, 2024, pp. 5939–5948.

[10] M. Zhu, H. Chen, Q. Yan, *et al.*, "GenImage: A Million-Scale Benchmark for Detecting AI-Generated Image," *NeurIPS*, vol. 36, pp. 77 771–77 782, 2023.

[11] D. Park, H. Na, and D. Choi, "Performance Comparison and Visualization of AI-Generated-Image Detection Methods," *IEEE Access*, 2024.

[12] D. Boychev and R. Cholakov, "ImagiNet: A Multi-Content Dataset for Generalizable Synthetic Image Detection via Contrastive Learning," *arXiv preprint arXiv:2407.20020*, 2024.

[13] Y. Hong and J. Zhang, "WildFake: A Large-scale Challenging Dataset for AI-Generated Images Detection," *arXiv preprint arXiv:2402.11843*, 2024.

[14] Z. Lu, D. Huang, L. Bai, *et al.*, "Seeing is not always believing: Benchmarking human and model perception of ai-generated images," *NeurIPS*, vol. 36, 2024.

[15] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "CNN-generated images are surprisingly easy to spot... for now," in *CVPR*, 2020, pp. 8695–8704.

[16] U. Ojha, Y. Li, and Y. J. Lee, "Towards universal fake image detectors that generalize across generative models," in *CVPR*, 2023, pp. 24 480–24 489.

[17] 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, pp. 1–5.

[18] Q. Bammey, "Synthbuster: Towards detection of diffusion model generated images," *IEEE Open Journal of Signal Processing*, vol. 5, pp. 1–9, 2024.

[19] G. Cazenavette, A. Sud, T. Leung, and B. Usman, "FakeInversion: Learning to Detect Images from Unseen Text-to-Image Models by Inverting Stable Diffusion," in *CVPR*, Jun. 2024, pp. 10 759–10 769.

[20] D. Tariang, R. Corvi, D. Cozzolino, G. Poggi, K. Nagano, and L. Verdoliva, "Synthetic Image Verification in the Era of Generative AI: What Works and What Isn't There Yet," *IEEE Security & Privacy*, vol. 22, pp. 37–49, 2024.

[21] C. Koutlis and S. Papadopoulos, "Leveraging Representations from Intermediate Encoder-blocks for Synthetic Image Detection," in *ECCV*, 2024, pp. 394–411.

[22] A. Sarkar, H. Mai, A. Mahapatra, S. Lazebnik, D. A. Forsyth, and A. Bhattad, "Shadows Don't Lie and Lines Can't Bend! Generative Models don't know Projective Geometry... for now," in *CVPR*, 2024.

[23] L. Baraldi, F. Cocchi, M. Cornia, L. Baraldi, A. Nicolosi, and R. Cucchiara, "Contrasting Deepfakes Diffusion via Contrastive Learning and Global-Local Similarities," in *ECCV*, 2024, pp. 199–216.

[24] A. S. Rajan, U. Ojha, J. Schloesser, and Y. J. Lee, "On the effectiveness of dataset alignment for fake image detection," *arXiv preprint arXiv:2410.11835*, 2024.

[25] F. Guillaro, G. Zingarini, B. Usman, A. Sud, D. Cozzolino, and L. Verdoliva, "A Bias-Free Training Paradigm for More General AI-generated Image Detection," *arXiv preprint arXiv:2412.17671*, 2024.

[26] F. Laiti, B. Liberatori, T. D. Min, and E. Ricci, "Conditioned prompt-optimization for continual deepfake detection," in *ICPR*, 2024.

[27] C. Li, Z. Huang, D. P. Paudel, *et al.*, "A continual deepfake detection benchmark: Dataset, methods, and essentials," in *WACV*, 2023, pp. 1339–1349.

[28] J. Tian, C. Yu, X. Wang, *et al.*, "Dynamic Mixed-Prototype Model for Incremental Deepfake Detection," in *ACM MM*, 2024, pp. 8129–8138.

[29] R. Corvi, D. Cozzolino, G. Poggi, K. Nagano, and L. Verdoliva, "Intriguing properties of synthetic images: From generative adversarial networks to diffusion models," in *CVPR Workshops*, 2023, pp. 973–982.

[30] O. Russakovsky, J. Deng, H. Su, *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[31] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft coco: Common objects in context," in *ECCV*, 2014, pp. 740–755.

[32] C. Schuhmann, R. Vencu, R. Beaumont, *et al.*, "LAION-400M: Open Dataset of CLIP-Filtered 400 Million Image-Text Pairs," *arXiv preprint arXiv:2111.02114*, 2021.

[33] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato, "RAISE: a raw images dataset for digital image forensics," in *ACM MMSys*, 2015, pp. 219–224.

[34] J. Ricker, D. Lukovnikov, and A. Fischer, "AEROBLADE: Training-Free Detection of Latent Diffusion Images Using Autoencoder Reconstruction Error," in *CVPR*, 2024, pp. 9130–9140.

[35] M. A. Rahman, B. Paul, N. H. Sarker, Z. I. A. Hakim, and S. A. Fattah, "ArtiFact: A Large-Scale Dataset with Artificial and Factual Images for Generalizable and Robust Synthetic Image Detection," in *ICIP*, 2023, pp. 2200–2204.

[36] J. Ricker, S. Damm, T. Holz, and A. Fischer, "Towards the detection of diffusion model deepfakes," in *VISAPP*, 2024, pp. 446–457.

[37] B. Chen, J. Zeng, J. Yang, and R. Yang, "DRCT: Diffusion Reconstruction Contrastive Training towards Universal Detection of Diffusion Generated Images," in *ICML*, 2024.

[38] D. Beniaguev, *Synthetic faces high quality - text 2 image (sfhq-t2i) dataset*, 2024. DOI: 10.34740/kaggle/dsv/9548853.

[39] Q. Bammey, *Positional learning for reliable ai-generated images detection*.

[40] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks," in *ICCV*, Oct. 2017.

[41] Q. Chen and V. Koltun, "Photographic Image Synthesis With Cascaded Refinement Networks," in *ICCV*, Oct. 2017.

[42] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive Growing of GANs for Improved Quality, Stability, and Variation," in *ICLR*, 2018.

[43] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *CVPR*, 2018, pp. 8789–8797.

[44] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. Huang, "Free-Form Image Inpainting With Gated Convolution," in *ICCV*, 2019, pp. 4470–4479.

[45] A. Brock, J. Donahue, and K. Simonyan, "Large Scale GAN Training for High Fidelity Natural Image Synthesis," in *ICLR*, 2019.

[46] K. Li, T. Zhang, and J. Malik, "Diverse Image Synthesis From Semantic Layouts via Conditional IMLE," in *ICCV*, 2019, pp. 4219–4228.

[47] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," *IEEE TPAMI*, vol. 43, no. 12, pp. 4217–4228, 2021.

[48] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic Image Synthesis With Spatially-Adaptive Normalization," in *CVPR*, 2019, pp. 2332–2341.

[49] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and Improving the Image Quality of StyleGAN," in *CVPR*, 2020, pp. 8107–8116.

[50] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," in *NeurIPS*, 2020.

[51] I. Anokhin, K. Demochkin, T. Khakhulin, G. Sterkin, V. Lempitsky, and D. Korzhenkov, "Image Generators with Conditionally-Independent Pixel Synthesis," in *CVPR*, 2021, pp. 14 273–14 282.

[52] P. Esser, R. Rombach, and B. Ommer, "Taming Transformers for High-Resolution Image Synthesis," in *CVPR*, 2021, pp. 12 868–12 878.

[53] D. A. Hudson and L. Zitnick, "Generative Adversarial Transformers," in *ICML*, 2021, pp. 4487–4499.

[54] P. Dhariwal and A. Nichol, "Diffusion models beat GANs on image synthesis," in *NeurIPS*, 2021.

[55] T. Karras, M. Aittala, S. Laine, *et al.*, "Alias-free generative adversarial networks," in *NeurIPS*, 2021.

[56] R. Suvorov, E. Logacheva, A. Mashikhin, *et al.*, "Resolution-Robust Large Mask Inpainting With Fourier Convolutions," in *WACV*, Jan. 2022, pp. 2149–2159.

[57] E. Wood, T. Baltrušaitis, C. Hewitt, S. Dziadzio, T. J. Cashman, and J. Shotton, "Fake It Till You Make It: Face Analysis in the Wild Using Synthetic Data Alone," in *ICCV*, Oct. 2021, pp. 3681–3691.

[58] A. Sauer, K. Chitta, J. Müller, and A. Geiger, "Projected GANs converge faster," in *NeurIPS*, 2021.

[59] C. Saharia, W. Chan, H. Chang, *et al.*, "Palette: Image-to-Image Diffusion Models," in *SIGGRAPH*, 2022.

[60] S. Gu, D. Chen, J. Bao, *et al.*, "Vector quantized diffusion model for text-to-image synthesis," in *CVPR*, 2022, pp. 10 686–10 696.

[61] Z. Xiao, K. Kreis, and A. Vahdat, "Tackling the generative learning trilemma with denoising diffusion GANs," in *ICLR*, 2022.

[62] A. Q. Nichol, P. Dhariwal, A. Ramesh, *et al.*, "GLIDE: Towards Photorealistic Image Generation and Editing

with Text-Guided Diffusion Models," in *ICML*, vol. 162, 2022, pp. 16 784–16 804.
[63] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," in *CVPR*, 2022, pp. 10 674–10 685.
[64] Midjourney, *Midjourney*, https://www.midjourney.com, 2022.
[65] W. Li, Z. Lin, K. Zhou, L. Qi, Y. Wang, and J. Jia, "Mat: Mask-aware transformer for large hole image inpainting," in *CVPR*, 2022, pp. 10 748–10 758.
[66] Z. Wang, H. Zheng, P. He, W. Chen, and M. Zhou, "Diffusion-GAN: Training GANs with diffusion," in *ICLR*, 2023.
[67] DeepFloyd Lab, StabilityAI, *DeepFloyd IF*, https://stability.ai/news/deepfloyd-if-text-to-image-model, 2023.
[68] D. Podell, Z. English, K. Lacey, *et al.*, "SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis," in *ICLR*, 2024.
[69] J. Betker, G. Goh, L. Jing, and T. Brooks, *Improving Image Generation with Better Captions*, https://cdn.openai.com/papers/dall-e-3.pdf, 2023.
[70] Black Forest Labs, *Flux*, https://github.com/black-forest-labs/flux, 2024.
[71] A. Radford, J. W. Kim, C. Hallacy, *et al.*, "Learning Transferable Visual Models From Natural Language Supervision," in *ICML*, 2021, pp. 8748–8763.
[72] M. Oquab, T. Darcet, T. Moutakanni, *et al.*, "DINOv2: Learning Robust Visual Features without Supervision," *Transactions on Machine Learning Research*, 2024, ISSN: 2835-8856.