-Supplementary Document-
Raising the Bar of AI-generated Image Detection with CLIP

Davide Cozzolino[1] Giovanni Poggi[1] Riccardo Corvi[1] Matthias Nießner[2] Luisa Verdoliva[1,2]

[1]University Federico II of Naples [2]Technical University of Munich

In this supplementary document, we provide a brief description of the methods used for comparison (Sec. 1), report additional ablation studies (Sec. 2), additional results (Sec. 3), and robustness analysis (Sec. 4). We also show some experiments carried out on social networks in a few-shot scenario (Sec. 5). Finally, we enlarge our initial dataset with additional synthetic generators and carry out further experiments on generalization (Sec. 6).

1. Reference methods

In our experimental analysis we included the following methods:

(a) **Wang et al.** [24] is a CNN detector based on ResNet50 and represents a reference in the research community. This work also introduced the large dataset (LSUN/ProGAN) extensively adopted for model training in subsequent works.

(b) **Gragnaniello et al.** [11] proposes a simple modification to the ResNet50 architecture which allows to better preserve low-level forensic traces and is trained on the same dataset introduced in [24].

(c) **Mandelli et al.** [17] relies on the ensemble of five EfficientNet-B4 networks trained on different datasets. At test-time the scores of randomly selected patches are aggregated: if at least one patch is detected as synthetic, then the entire image is classified as synthetic.

(d) **PatchForensics** [4] develops a fully-convolutional classifier based on local patches with limited receptive fields over an XceptionNet backbone.

(e) **Liu et al.** [16] is a detector that exploits the inconsistency between real and fake images in the representations of the learned noise patterns. This spatial information is combined with frequency information to improve the classification.

(f) **LGrad** [21] works on the gradients extracted through a pre-trained CNN model in order to filter out the content of the image and transform a data-dependent problem into a transformation-model dependent problem.

(g) **Corvi et al.** [6] performs strong augmentation to gain robustness and increase generalization and is trained on a large dataset of latent diffusion models.

(h) **Ojha et al.** [20] uses a large dataset of real and synthetic images to train a simple classifier working on pre-trained CLIP features.

(i) **DIRE** [25] is based on the idea that synthetic images can be reconstructed better than real images by a pre-trained model. To this end a ResNet-50 is trained in two different ways, using ADM images (DIRE 1) and StyleGAN images (DIRE 2), respectively.

(j) **NPR** [22] works on residual images computed as the difference between the original image and its interpolated version. The classifier is a ResNet-50 trained on only 4 classes of the ProGAN dataset.

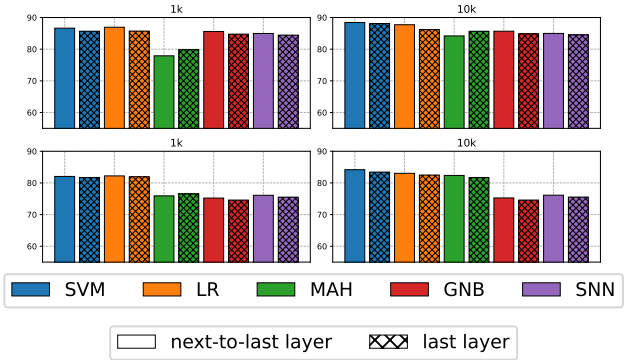

Figure 1. Average AUC for our approach considering different classifiers and different features on the original dataset (top) and after random compression and resizing (bottom). Left: 1k images, Right: 10k images.

2. Additional ablation study

In Section 4 of the paper we describe the CLIP-based detector used in all experiments. In the proposed procedure two key design choices are made: *i)* we extract features from the next to last layer of the CLIP ViT L/14 architecture; and *ii)* we use a SVM classifier, trained on a limited set of reference features. Here we test alternative solutions, that is:

- extracting features from the last layer of the architecture;
- using other classifiers: Logistic Regression (LR), Maha-

Method	Families of Generators			
	GAN AUC/Acc	Diffusion AUC/Acc	Comm. Tools AUC/Acc	Average AUC/Acc
Wang et al.	92.1 / 69.4	55.8 / 50.3	50.2 / 49.8	66.0 / 56.5
PatchFor.	84.9 / 53.6	76.4 / 50.4	50.5 / 50.0	70.6 / 51.3
Grag. et al.	95.8 / **90.1**	70.8 / 57.0	41.8 / 43.6	69.5 / 63.6
Mand. et al.	88.9 / 81.6	55.8 / 53.7	22.6 / 32.9	55.7 / 56.1
Liu et al.	**99.0** / **89.7**	79.9 / 69.5	30.8 / 42.9	69.9 / 67.4
Corvi et al.	72.7 / 52.1	91.5 / 75.1	82.1 / 62.8	82.1 / 63.3
LGrad	87.8 / 76.4	69.2 / 60.4	48.3 / 52.1	68.4 / 63.0
Ojha et al.	96.1 / 85.4	82.0 / 63.4	73.8 / 68.3	84.0 / 72.4
DIRE-1	65.9 / 64.4	71.6 / 72.8	50.2 / 49.9	62.5 / 62.4
DIRE-2	65.1 / 60.1	70.4 / 65.8	45.3 / 49.9	60.3 / 58.6
NPR	89.5 / 79.7	82.0 / 68.1	49.3 / 50.1	73.6 / 66.0
Ours 1k	91.2 / 76.4	**92.1** / 76.2	76.6 / 54.5	86.6 / 69.0
Ours 1k+	86.2 / 74.2	89.9 / **79.9**	85.3 / **72.6**	87.1 / **75.6**
Ours 10k	92.4 / 79.1	**92.6** / 73.3	80.5 / 52.6	**88.5** / 68.3
Ours 10k+	89.3 / 74.9	**91.8** / 77.2	**87.0** / 67.3	**89.4** / 73.1

Table 1. **Comparison with SOTA methods in terms of AUC and Accuracy.** We report the mean AUC and Accuracy for each family of generators and the global average.

Method	Families of Generators			
	GAN AUC/Acc	Diffusion AUC/Acc	Comm. Tools AUC/Acc	Average AUC/Acc
Wang et al.	79.2 / 62.0	59.3 / 50.4	44.1 / 49.9	60.9 / 54.1
PatchFor.	54.2 / 50.0	64.8 / 50.3	62.9 / 50.1	60.7 / 50.1
Grag. et al.	**89.0** / 67.6	67.9 / 50.8	47.3 / 50.0	68.1 / 56.1
Mand. et al.	69.2 / 55.1	50.6 / 50.8	44.7 / 49.6	54.9 / 51.9
Liu et al.	54.4 / 51.5	56.3 / 51.1	53.0 / 50.7	54.6 / 51.1
Corvi et al.	74.0 / 55.1	77.3 / 62.1	52.1 / 50.1	67.8 / 55.8
LGrad	52.0 / 50.9	48.0 / 50.5	49.4 / 50.4	49.8 / 50.6
Ojha et al.	86.1 / **73.7**	73.7 / 59.8	49.6 / 50.6	69.8 / 61.4
DIRE-1	47.6 / 50.2	51.0 / 50.5	50.1 / 49.9	49.6 / 50.2
DIRE-2	47.7 / 49.6	52.3 / 52.5	45.3 / 49.9	48.4 / 50.7
NPR	48.7 / 50.2	50.0 / 49.8	56.0 / 50.1	51.6 / 50.0
Ours 1k	76.2 / 68.9	80.6 / 71.2	72.2 / 60.7	76.3 / 66.9
Ours 1k+	75.0 / 67.2	87.8 / 77.8	83.4 / 65.7	82.1 / 70.3
Ours 10k	76.9 / 63.6	81.1 / 63.7	73.0 / 52.4	77.0 / 59.9
Ours 10k+	78.2 / 69.0	**89.3** / **78.7**	**84.7** / **66.4**	**84.1** / **71.4**

Table 2. **AUC/Accuracy in the presence of post-processing.** All images have been randomly cropped, resized and compressed to simulate a realistic scenario on the web.

lanobis distance (MAH), Gaussian Naive Bayes classifier (GNB), Soft voting k-Nearest Neighbor (SNN) [19].

In Figure 1, we show the results in terms of average AUC over the dataset described in Section 3 (main paper) using both original images and images impaired by common post-processing steps such as recompression and resizing. For all detectors, we consider the versions 1k and 10k (1000 and 10000 reference images per class, respectively).

In all cases, the SVM classifier seems to ensure the best performance, followed closely by the Logistic Regression, while the other classifiers provide a less consistent performance. Using features from the next-to-last layer is almost always preferable to using features from the last layer, and this happens always with the SVM classifier. These results motivated our design choices.

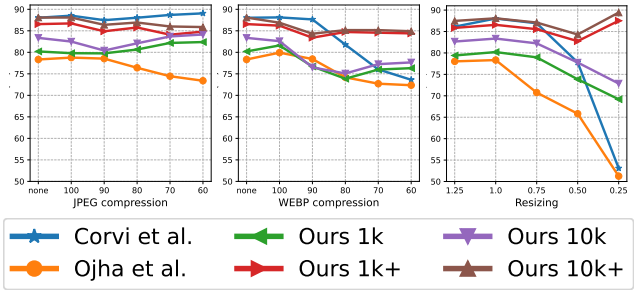

Figure 2. Robustness analysis in terms of AUC carried out on the Stable Diff. 2, SDXL, DALL·E 3, DALL·E 2, Midjourney, and Firefly generation models of the SynthBuster dataset [1].

3. AUC vs Accuracy

In this Section we present additional results on the synthetic generators analyzed in the main paper aggregated by family (GAN, Diffusion, Commercial tools). Results are not only in terms of AUC but also of balanced accuracy computed with a fixed threshold of 0.5. Indeed in a practical scenario a threshold must be set to discriminate between real and fake images, which is not a trivial task as can be seen in Table 1. In fact, in the absence of an adequate calibration procedure, a significant drop between AUC and accuracy is often observed. However, our approach provides good results even in this challenging context, especially on post-processed images (Tab. 2) where most methods are equivalent to flipping coins.

4. Additional robustness analysis

In Section 5 of the main paper we provide some results on the robustness of SoTA and proposed methods in the presence of image impairments. Due to lack of space, however, we show only some very synthetic results, averaged on all kinds of post-processed images. In Fig. 2, we analyze robustness in more detail as a function of JPEG Quality Factor, ranging from 100 to 60, and resizing scale, going from 125% to 25%. In addition, we consider also another compression method, WebP, which is gaining popularity on social networks but has been rarely considered in experimental analyses. Images are from the SynthBuster dataset [1], generated by Stable Diffusion 2, SDXL, DALL·E 2, DALL·E 3, Midjourney, and Firefly. We show results (AUC) only for the SoTA methods that proved most competitive in terms of robustness, Corvi and Ojha. For our method we consider again the versions with 1k or 10k real and fake reference images, and the corresponding versions, 1k+ and 10k+, with the same number of images including random compression and resizing.

Ojha suffers some performance losses, up to 10 points, in the presence of compression, be it JPEG or WebP, and a much more rapid decline with resizing, to the point of be-

AUC/Acc	Social network: X			
Method	DALL·E3	Midjourney	FireFly	Average
Wang et al.	22.5 / 50.0	36.7 / 50.0	65.3 / 50.0	41.5 / 50.0
PatchFor.	54.5 / 50.0	59.9 / 50.0	46.5 / 50.0	53.6 / 50.0
Grag. et al.	44.7 / 49.6	58.4 / 49.7	80.5 / 49.9	61.2 / 49.7
Mand. et al.	42.9 / 49.3	53.7 / 50.6	52.7 / 50.1	49.8 / 50.0
Liu et al.	34.5 / 49.7	38.7 / 49.7	54.9 / 49.9	42.7 / 49.8
Corvi et al.	89.3 / 86.8	98.3 / 90.8	79.1 / 52.1	88.9 / 76.6
LGrad	33.7 / 39.0	37.9 / 41.4	61.4 / 58.4	44.3 / 46.3
Ojha et al.	34.4 / 45.6	32.2 / 46.2	73.0 / 65.7	46.5 / 52.5
DIRE-1	54.3 / 52.0	60.7 / 54.3	55.1 / 51.1	56.7 / 52.5
DIRE-2	60.0 / 55.0	58.6 / 53.9	47.7 / 46.5	55.4 / 51.8
NPR	43.8 / 41.7	46.2 / 44.9	65.4 / 62.0	51.8 / 49.5
Ours 1k	69.4 / 50.9	70.2 / 53.3	69.3 / 51.4	69.6 / 51.9
Ours 1k+	82.1 / 73.3	82.8 / 74.5	83.8 / 74.9	82.9 / 74.3
Ours 10k	65.8 / 50.0	69.3 / 51.7	70.7 / 50.9	68.6 / 50.9
Ours 10k+	78.7 / 63.7	82.2 / 69.6	82.7 / 69.6	81.2 / 67.6
Ours fusion	90.0 / 85.9	97.5 / 93.8	77.8 / 57.0	88.4 / 78.9
Ours fusion+	94.7 / 87.3	97.9 / 92.1	85.4 / 74.2	92.6 / 84.5
Ours Few-Shot	98.5 / 92.5	95.8 / 87.4	92.5 / 83.5	95.6 / 87.8

Table 3. Results in terms of AUC and accuracy of proposal and SOTA methods on real and synthetic images download from X. The last row reports a few-shot experiment, where we suppose to know in advance 10+10 real/fake images from a specific generator.

Figure 3. Some examples of images downloaded from the social network X. From left to right: a real image, synthetic images from DALL·E 3 [2], Firefly [10], Midjourney [18].

coming useless in the presence of a 25% rescaling. Corvi was trained with strong augmentation and, in fact, is not affected at all by JPEG compression, while it presents acceptable losses with strong WebP compression (not seen in training), and strong resizing. The most remarkable outcome of this experiment, however, is the impressive robustness of the CLIP-based detectors. Both the 1k+ and 10k+ versions, those with augmentation, are basically insensitive to compression, no matter JPEG or WebP, and resizing. The versions without augmentation suffer some loss of performance but not nearly as dramatic as for the reference methods, and remain effective in all conditions.

5. Few-shot analysis in the wild

In this section we present an experiment in a few-shot scenario. The idea is to explore the ability of our detector to work with very limited data and adapt to a situation where only a few real and synthetic examples are available. We downloaded 500 real images and 1,500 synthetic images from X from three different generators DALL·E 3, Midjourney and Firefly (some examples can be seen in Fig. 3). To understand which generator was used to create a specific image, we relied on tags and annotations present on the social network.

Generator	modality	LSUN [27]	FFHQ [13]	ImageNet [8]	COCO [15]	Resolution
BigGAN [3]	c			✓		256^2, 512^2
EG3D [5]	u		✓			512^2
Diff.ProjectedGAN [26]	u	✓				256^2
GALIP [23]	t				✓	224^2
Taming Transf. [9]	u,c		✓	✓		256^2
DALL·E mini [7]	t				✓	256^2
DDPM [12]	u		✓			256^2
Deepfloyd-IF II stage [14]	t				✓	256^2

Table 4. Image generators used in our experiments: GAN-based, VQ-GAN-based and DM-based. The generation modalities, unconditional (u), conditional (c), and text-to-image (t), is reported in the second column. The last column reports the resolution of the images in the dataset.

In our few-shot scenario, we take 10 real images and 10 generated images as examples from a specific model and test on all the others images (note that we present average results on 1,000 runs). Results are reported in Tab. 3 in terms of AUC and accuracy. The availability of just 10+10 images of the target data provides an impressive performance boost with respect to the same method trained on a dataset much larger but not aligned with the test data (real images come from COCO, synthetic images from Latent diffusion models). It is important to underline that this is a realistic scenario, in which one is called to decide on images generated by a new method and a few sample images are available as a support. We believe that this can represent an interesting direction for the application in the wild or to easily adapt a detector to more challenging situations where some prior information is available.

To complete our analysis on this dataset we include in Tab. 3 the comparison with SoTA methods and our original proposal. We can observe that the performance of all the methods degrade which highlights the increased difficulty to handle a realistic scenario over the web. The best performance can be obtained by using our fusion approach that take the best of the low-level and high-level features and is able to achieve on average more than 90% in terms of AUC which is comparable with the few-shot analysis where some prior knowledge is available.

AUC/Acc Method	BigGAN	EG3D	Diff. Proj. GAN	GALIP	Taming Transf.	DALL·E Mini	DDPM	Deepfloyd-IF II stage	AVG
Wang et al.	92.7 / 66.1	94.4 / 59.2	89.8 / 52.1	89.7 / 57.4	54.3 / 51.7	62.5 / 51.8	31.6 / 50.1	43.1 / 50.1	69.8 / 54.8
PatchFor.	85.5 / 52.5	69.8 / 50.0	92.6 / 61.7	98.2 / 73.4	71.2 / 51.0	83.8 / 51.4	98.4 / 50.2	83.4 / 50.0	85.4 / 55.0
Grag. et al.	98.7 / 94.2	98.9 / 93.8	**100.** / 72.5	96.2 / 79.5	90.3 / 76.4	83.4 / 62.5	49.7 / 45.1	71.2 / 61.3	86.0 / 73.2
Mand. et al.	92.2 / 83.0	**100.** / **99.8**	64.5 / 49.9	77.6 / 59.1	91.8 / 84.1	83.6 / 69.8	**99.9** / **97.4**	49.2 / 48.9	82.3 / 74.0
Liu et al.	94.7 / 81.3	99.0 / 86.3	**99.5** / 84.9	94.3 / 78.4	95.4 / 78.9	98.4 / 88.1	22.8 / 49.6	97.4 / 86.8	87.7 / 79.3
Corvi et al.	83.4 / 51.8	25.2 / 50.0	96.6 / 71.2	87.7 / 50.9	**99.3** / 89.5	**99.7** / 83.8	**100.** / 90.3	68.5 / 50.8	82.5 / 67.3
LGrad	77.2 / 69.0	68.8 / 59.8	**99.5** / 90.1	56.7 / 55.1	74.1 / 64.5	67.3 / 59.9	9.8 / 16.4	75.0 / 62.5	66.1 / 59.7
Ojha et al.	**99.6** / **96.4**	92.6 / 82.5	97.4 / 75.2	98.6 / 89.9	94.1 / 84.8	97.1 / 84.9	77.7 / 68.2	60.8 / 50.0	89.7 / 79.0
DIRE-1	**99.8** / 95.3	50.1 / 50.0	49.8 / 51.6	**100.** / 96.7	73.1 / 72.4	**99.7** / 96.5	23.1 / 50.0	**99.4** / **95.8**	74.4 / 76.0
DIRE-2	98.6 / 82.4	46.1 / 50.0	50.2 / 51.4	**99.3** / 81.8	77.0 / 66.2	98.7 / 81.8	14.0 / 49.8	95.6 / 80.5	72.4 / 68.0
NPR	86.8 / 77.2	53.3 / 57.5	**100.** / **99.2**	90.7 / 77.6	80.2 / 69.2	79.0 / 73.3	92.4 / 61.6	90.9 / 76.8	84.2 / 74.1
Ours 1k	96.9 / 86.1	87.0 / 73.0	**99.3** / 70.2	**100.** / **99.7**	**99.4** / 95.0	**100.** / **99.1**	95.1 / 86.7	**99.7** / 90.7	**97.2** / **87.6**
Ours 1k+	92.0 / 80.0	76.4 / 65.5	89.6 / 67.7	**99.9** / 98.6	93.5 / 83.0	**99.6** / 95.0	94.9 / 87.4	**99.7** / **96.0**	93.2 / 84.1
Ours 10k	98.2 / 87.0	87.7 / 63.2	**99.7** / 87.4	**100.** / **99.8**	**99.7** / **97.4**	**100.** / **99.4**	93.9 / 61.9	**99.8** / 85.9	**97.4** / 85.3
Ours 10k+	93.1 / 77.5	79.9 / 58.8	92.5 / 76.0	**100.** / 97.0	94.9 / 79.5	**99.9** / 94.4	98.1 / 91.5	**99.8** / 92.2	94.8 / 83.3

Table 5. **Comparison with SOTA methods on additional data.** The results are in terms of AUC/Accuracy. The entries in bold underline the best performance for each dataset. For our approach we show four variants: trained on 1k real and 1k fake images; 10k real and 10k fake images; trained on 1k and 1k fake images but including compressed/resized images (1k+) and trained on 10k and 10k fake images but including compressed/resized images (10k+).

AUC/Acc Method	BigGAN	EG3D	Diff. Proj. GAN	GALIP	Taming Transf.	DALL·E Mini	DDPM	Deepfloyd-IF II stage	AVG
Wang et al.	82.5 / 55.5	84.7 / 52.2	80.8 / 53.2	92.2 / 59.2	66.2 / 50.6	66.7 / 50.5	69.6 / 50.2	47.9 / 49.9	73.8 / 52.7
PatchFor.	58.7 / 50.3	60.3 / 50.0	55.6 / 50.0	72.3 / 50.3	50.4 / 50.1	71.0 / 50.1	82.9 / 51.0	69.7 / 50.0	65.1 / 50.2
Grag. et al.	**97.6** / 74.4	**92.7** / 56.5	**99.2** / 65.6	**99.1** / 79.8	83.6 / 53.3	89.6 / 54.0	74.5 / 49.4	54.3 / 50.0	86.3 / 60.4
Mand. et al.	71.4 / 55.5	85.8 / **62.1**	60.3 / 49.6	78.1 / 57.3	82.3 / 59.7	68.2 / 57.4	61.1 / 58.6	49.0 / 49.8	69.5 / 56.3
Liu et al.	57.7 / 52.0	51.6 / 50.4	62.9 / 54.1	65.2 / 52.3	49.2 / 50.2	49.1 / 50.7	58.3 / 49.1	64.4 / 52.0	57.3 / 51.4
Corvi et al.	77.3 / 53.0	64.5 / 59.7	93.3 / 59.8	87.4 / 51.5	**94.4** / 76.7	97.4 / 74.8	74.7 / 59.6	70.4 / 50.9	82.4 / 60.7
LGrad	53.5 / 51.6	53.2 / 51.1	56.9 / 51.1	57.0 / 50.3	51.3 / 51.1	41.7 / 49.1	56.0 / 49.7	38.7 / 48.8	51.0 / 50.4
Ojha et al.	94.8 / **84.9**	79.3 / **61.5**	84.7 / **68.5**	95.7 / 84.5	92.9 / **81.8**	89.1 / 72.8	90.8 / 79.1	59.1 / 49.5	85.8 / 72.8
DIRE-1	47.2 / 49.5	35.2 / 49.3	35.2 / 49.6	47.4 / 50.6	47.3 / 49.7	54.6 / 51.1	38.5 / 49.4	64.6 / 53.2	46.2 / 50.3
DIRE-2	44.6 / 47.0	33.6 / 47.1	37.5 / 44.9	46.2 / 47.2	43.3 / 46.8	53.1 / 50.0	42.2 / 47.5	66.3 / 59.3	45.9 / 48.7
NPR	52.6 / 50.0	40.2 / 49.8	48.8 / 50.3	57.8 / 50.6	46.2 / 49.8	51.7 / 49.6	54.5 / 50.5	51.3 / 48.9	50.4 / 49.9
Ours 1k	84.6 / 75.8	64.8 / 60.3	73.4 / 66.7	98.3 / 92.6	85.1 / 75.8	96.2 / 88.9	76.1 / 67.7	90.0 / 79.9	83.6 / 76.0
Ours 1k+	89.6 / 79.7	60.5 / 55.1	69.6 / 61.0	**99.9** / **98.4**	86.9 / 78.0	**98.9** / 93.7	**96.4** / 84.8	**99.5** / **96.5**	87.7 / 80.9
Ours 10k	84.1 / 67.0	64.3 / 54.9	73.7 / 62.9	98.1 / 83.9	86.2 / 74.0	96.8 / 77.8	70.9 / 60.7	91.3 / 67.5	83.1 / 68.6
Ours 10k+	92.4 / 82.4	63.1 / 55.5	74.1 / 63.9	**99.9** / **98.9**	88.1 / 78.2	**99.4** / **94.9**	**96.1** / **87.1**	**99.7** / **97.0**	**89.1** / **82.2**

Table 6. **Comparison with SOTA methods on additional data in the presence of post-processing.** The results are in terms of AUC/Accuracy. The entries in bold underline the best performance for each dataset.

6. Further generalization results

In this Section we extend our generalization analysis to additional data and consider 8 more generators with 8,000 additional synthetic images for our test set (Tab. 4). In Tab. 5 we show the results in terms of AUC and accuracy on such data for the four version of the proposed CLIP-based detector and for the SoTA methods described in Section 1. We can observe that SoTA methods present a larger variability in terms of performance, with very good results on some generators and very bad on others. Instead, our method can provide consistently good performance over all the data both in terms of AUC and accuracy, with an average gain over the best reference of around 8.5% in terms of AUC and 13% in terms of Accuracy. This difference is even more noticeable on data that have undergone random compression and resizing (Tab. 6). In fact, for most of the competitors there is a dramatic performance loss, sometimes to a random guess level. Some other methods, such as, Ohja et al., preserve a good performance and the same happens for ours. Of course, in this more challenging scenario our best variant is the one that includes some form of augmentation in the reference data but even the variant trained on the original data provides satisfying results.

References

[1] Quentin Bammey. Synthbuster: Towards detection of diffusion model generated images. *IEEE OJSP*, 2023. 2

[2] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu,

and Yunxin Jiao. https://openai.com/dall-e-3, 2023. 3

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *ICLR*, 2018. 3

[4] Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. What Makes Fake Images Detectable? Understanding Properties that Generalize. In *ECCV*, pages 103–120, 2020. 1

[5] Eric Ryan Chan, Connor Zhizhen Lin, Matthew Aaron Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3d generative adversarial networks. In *CVPR*, pages 16123–16133, 2022. 3

[6] Riccardo Corvi, Davide Cozzolino, Giada Zingarini, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. On the detection of synthetic images generated by diffusion models. In *ICASSP*, pages 1–5, 2023. 1

[7] Boris Dayma, Suraj Patil, Pedro Cuenca, Khalid Saifullah, Tanishq Abraham, Phúc Lê Khàc, Luke Melas, and Ritobrata Ghosh. https://github.com/borisdayma/dalle-mini, 2021. 3

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 3

[9] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, pages 12873–12883, 2021. 3

[10] Adobe Firefly. https://www.adobe.com/sensei/generative-ai/firefly.html, 2023. 3

[11] Diego Gragnaniello, Davide Cozzolino, Francesco Marra, Giovanni Poggi, and Luisa Verdoliva. Are GAN generated images easy to detect? A critical analysis of the state-of-the-art. In *ICME*, pages 1–6, 2021. 1

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, pages 6840–6851, 2020. 3

[13] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 3

[14] Misha Konstantinov, Alex Shonenkov, Daria Bakshandaeva, Christoph Schuhmann, Ksenia Ivanova, and Nadiia Klokova. https://www.deepfloyd.ai/deepfloyd-if, 2023. 3

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, pages 740–755, 2014. 3

[16] Bo Liu, Fan Yang, Xiuli Bi, Bin Xiao, Weisheng Li, and Xinbo Gao. Detecting generated images by real images. In *ECCV*, pages 95–110, 2022. 1

[17] Sara Mandelli, Nicolò Bonettini, Paolo Bestagini, and Stefano Tubaro. Detecting GAN-generated Images by Orthogonal Training of Multiple CNNs. In *ICIP*, pages 3091–3095, 2022. 1

[18] Midjourney. https://www.midjourney.com/home, 2023. 3

[19] Harvey B Mitchell and Paul A Schaefer. A "soft" K-nearest neighbor voting scheme. *IJIS*, 16(4):459–468, 2001. 2

[20] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *CVPR*, pages 24480–24489, 2023. 1

[21] Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, and Yunchao Wei. Learning on Gradients: Generalized Artifacts Representation for GAN-Generated Images Detection. In *CVPR*, pages 12105–12114, 2023. 1

[22] Chuangchuang Tan, Huan Liu, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao Wei. Rethinking the Up-Sampling Operations in CNN-based Generative Network for Generalizable Deepfake Detection. In *CVPR*, 2024. 1

[23] Ming Tao, Bing-Kun Bao, Hao Tang, and Changsheng Xu. GALIP: Generative Adversarial CLIPs for Text-to-Image Synthesis. In *CVPR*, pages 14214–14223, 2023. 3

[24] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. CNN-generated images are surprisingly easy to spot... for now. In *CVPR*, 2020. 1

[25] Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, Hezhen Hu, Hong Chen, and Houqiang Li. DIRE for Diffusion-Generated Image Detection. *ICCV*, 2023. 1

[26] Zhendong Wang, Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Diffusion-GAN: Training GANs with Diffusion. In *ICLR*, 2023. 3

[27] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 3