Deep Image Fingerprint: Towards Low Budget Synthetic Image Detection and Model Lineage Analysis

Sergey Sinitsa, Ohad Fried
Reichman University

Abstract

The generation of high-quality images has become widely accessible and is a rapidly evolving process. As a result, anyone can generate images that are indistinguishable from real ones. This leads to a wide range of applications, including malicious usage with deceptive intentions. Despite advances in detection techniques for generated images, a robust detection method still eludes us. Furthermore, model personalization techniques might affect the detection capabilities of existing methods. In this work, we utilize the architectural properties of convolutional neural networks (CNNs) to develop a new detection method. Our method can detect images from a known generative model and enable us to establish relationships between fine-tuned generative models. We tested the method on images produced by both Generative Adversarial Networks (GANs) and recent large text-to-image models (LTIMs) that rely on Diffusion Models. Our approach outperforms others trained under identical conditions and achieves comparable performance to state-of-the-art pre-trained detection methods on images generated by Stable Diffusion and MidJourney, with significantly fewer required train samples.

1. Introduction

Generative neural networks enable the generation of high-quality images. While this has many benefits for scientific, creative, and business purposes, it can also be used for malicious deception. As image quality continues to improve, it becomes increasingly difficult for human observers to distinguish between real and fake images without careful observation and identification of inconsistencies, especially in spite of large text-to-image-models (LTIMs) [17, 18]. Therefore, there is an urgent need for an automated tool that can both detect generated images. This work aims to provide such a method, which has demonstrated good performance on both novel and popular approaches.

Acknowledgement: This work was supported in part by the Israel Science Foundation (grant No. 1574/21).
Project page: https://sergo2020.github.io/DIF/

The advancement of image generation has been made possible by the use of Deep Neural Networks (DNNs), particularly the Convolutional Neural Networks (CNNs) subclass. CNNs provide the best of both worlds by combining the image prior [51] with the flexibility of DNNs [28]. This has led to the development of image generators, which are generative models trained to produce images given a sample from known distributions and can be conditioned on input. The most popular type of image generator families is the Generative Adversarial Network (GAN) [22, 42], which quickly gained popularity for its ability to produce high-quality and high-resolution images in various applications such as [6, 8, 32–34, 57]. Previously these models have demonstrated state-of-the-art (SOTA) results in the field of image generation.

However, the advent of diffusion models [26, 44] have introduced a new paradigm that has shown the ability to generate high-quality images surpassing those produced by GANs [14]. Diffusion models are also a type of generative models and can be easily conditioned to arbitrary input. This gave rise to a new type of image generators, the LTIMs. These models incorporate advanced image generators (e.g., [16, 26, 44]) and are capable of generating high-quality images from text captions combined with other input domains [3–5, 40, 43, 47, 53]. Despite the advancements and conceptual shifts, all the mentioned methods still depend on CNNs for image generation.

To detect generated images, there are typically two approaches: data-driven and rule-based. Data-driven methods [11, 23, 52] involve training models on large datasets of images, which are expected to generalize to unseen data. However, these methods may struggle to detect images from conceptually different generators, as shown in a later study [10]. Rule-based methods, on the other hand, rely on identifying common patterns or characteristics seen in generated images [7, 21, 31, 36–38, 54], and typically require less data. However, most of these rules have only been demonstrated on a selected set of GAN models and image domains, which makes it necessary to re-evaluate the rules for novel types of image generators. This can be a laborious process as the rules are human-devised.

Despite significant progress in synthetic images detection, the widespread adoption of LTIMs created new challenges in the field. Due to the resource-intensive nature of LTIMs' training, access to LTIMs and the amount of available generated images for research purposes is limited. As a result, data-driven approaches and some rule-based methods may not be practical, as they require a significant amount of data for training, and in some cases the training of an image generator itself. Therefore, new detectors are expected to not only perform well, but also to be trained on a small number of images.

Detecting images generated by "personalized" models poses an additional challenge. These models are fine-tuned versions of LTIMs that are trained to generate images with specific objects or in particular styles [2, 20, 46]. Despite their widespread use, the impact of personalization and fine-tuning on fingerprints has not been studied.

The proposed method achieves high detection accuracy with minimal training data (less than 512 images), leveraging CNN architecture to extract fingerprints of image generators and detect images from the same generator. Other detectors typically require hundreds of thousands of images.

Our method was tested on various image generators, including established GANs and LTIMs. It outperforms competitors trained under the same conditions and achieves comparable performance to SOTA pre-trained detectors on widely used LTIMs such as Stable Diffusion [44] and MidJourney [27], with significantly fewer required training samples. Moreover, our method proves valuable for model lineage analysis (Sec. 4.4). However, it is important to note that our method is a proof-of-concept work and has certain limitations, which are discussed in Sec. 5.

2. Related Work

Data-driven methods rely on CNN classification models. These methods aim to detect compressed images from unknown image generators by training a detector on images from a single image generator. The authors of [52] fine-tune a pre-trained Resnet-50 [25] with 720k fake and real images produced by ProGAN and apply a set of compressions during training. They achieve high average precision on images from several GAN models, but demonstrate poor accuracy (Sec. 4.2). In [23], the authors repeat the process, but with a modified Resnet50 and heavier augmentations, resulting in SOTA performance. However, after the development of LTIMs, another study [10] revealed that later model generalizes well only on the same image generator family.

One notable approach from rule-based methods involves detecting spatially-stationary and high-frequency artifacts that image generators produce within images [41]. These artifacts were observed in generated images [31, 37, 54], including those produced by LTIMs [9, 10]. Since they are unique to each trained image generator, they are referred to as *fingerprints* ($\mathcal{F}$-s). To estimate $\mathcal{F}$, a set of residuals is produced by passing each image through a denoising filter (f_D), and the residuals are then averaged. This leaves only the common deterministic pattern within the residuals — fingerprint. The image is associated with the generative model by calculating the correlation coefficient between its residual and the model's $\mathcal{F}$.

The usage of $\mathcal{F}$-s holds great potential in model lineage analysis. Marra et al. [37] demonstrated that residuals of images generated by a specific model architecture exhibit high correlation not only with the model's fingerprint but also with the fingerprints of models sharing the same architecture [37]. Yu et al. proposed a supervised method to attribute fake images to their source models, training a detector on a dataset of images generated by multiple GANs [54]. Nevertheless, this approach relies on manual supervision, requiring researchers to make educated guesses about the relationship between different models prior to training. Previous studies have not explored the relationship between models and their fine-tuned versions, or the methodology for model lineage analysis.

In addition to the above, $\mathcal{F}$-s and other artifacts are primarily observed in the spectrum space, by transforming the image with the Fast Fourier Transform (FFT). Some studies [19, 56] use this concept by training models on image spectrum samples or performing operations within Fourier space. Some attempts have also been made to synthesize a large set of $\mathcal{F}$-s for further training of detectors [30, 56]. However, while these methods demonstrate good detection accuracy for a set of image generators on average, they may perform poorly with some of them.

Another group of rule-based methods focuses on detecting color distortions in generated images [7, 21, 36, 38]. These detection methods are usually evaluated on both generated and natural images and have shown promising results. However, it should be noted that color distortions have only been demonstrated for a few image domains and GAN models.

3. Method

Our method extracts a CNN fingerprint of an image generator using a small number of generated images and applies it for the detection of other images produced by the same image generator. The method relies on the properties of CNNs, as explained below with a simple experiment (Sec. 3.1). Then we'll explain the fingerprint extraction (Sec. 3.2) and following detection process (Sec. 3.3), including implementation details (Sec. 3.4).

3.1. Deep Image Fingerprint

Deep Image Prior [51] demonstrated that CNNs incorporate an image prior in their structure. Consider an image restoration task, where given a corrupt image $X \in$

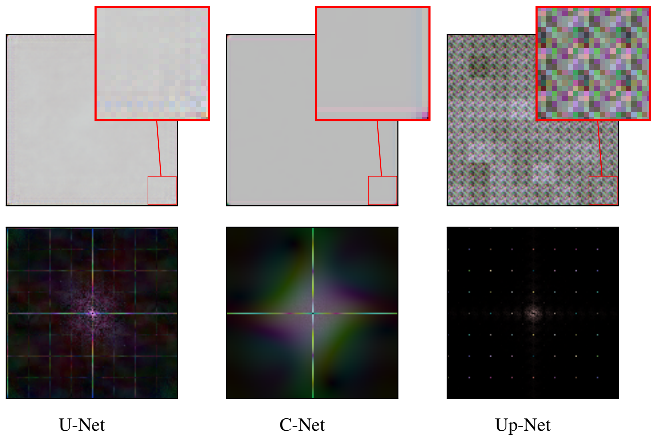

Figure 1. The reconstructed gray images (top) and their FFT log-magnitude (bottom). For previewing images are normalized. The mean value is subtracted before applying the FFT. U-Net produces mix of boundary artifacts (lines) and up-sampling artifacts (checkerboard). C-Net produces solely boundary artifacts, while Up-Net exclusively yields up-sampling artifacts. In the Up-Net model, the input noise is up-sampled after 1x1 convolution, resulting in blocks with varying gray levels.

$\mathbb{R}^{3\times H\times W}$ the goal is to obtain the restored image $\hat{Y} \in \mathbb{R}^{3\times H\times W}$ from a trained model h_θ:

$$\hat{Y} = \min_{\theta} E(h_\theta, X) + P(X) \tag{1}$$

E is a data term and P is the image prior.

For a CNN encoder-decoder model g_θ, the architecture itself serves as the image prior. Thus, the optimization task is simplified, as the search is focused on finding only the data term E within the weights (θ) space of the CNN without the need of learning P. Here the input of g_θ is a random tensor $Z \in \mathbb{R}^{C\times H\times W}$, where each element is $z_i \sim U(0, 1)$.

$$\hat{Y} = \min_{\theta} E(g_\theta(Z), X) \tag{2}$$

After the optimization, the reconstructed image is given by:

$$\hat{Y} = g_{\hat{\theta}}(Z), \quad \hat{\theta} = \operatorname*{argmin}_{\theta} E(g_\theta(Z), X) \tag{3}$$

But, following a number of observations, Eq. (3) seems to be incomplete. Images produced by CNNs include a unique model fingerprint ($\mathcal{F} \in \mathbb{R}^{3\times H\times W}$), thus:

$$\hat{Y} + \mathcal{F}(g_{\hat{\theta}}) = g_{\hat{\theta}}(Z) \tag{4}$$

To prove this statement, we perform a simple experiment: we optimize the weights of CNN (θ) U-Net [45] to produce a single gray image without any semantic content. It seems as a trivial task, yet after convergence, the model is still unable to reconstruct the image perfectly (Fig. 1).

We reveal the artifacts by simple image normalization to range [0,1]. Two main fingerprint patterns are observed: up-sampling and boundary artifacts. Previous research has primarily focused on up-sampling artifacts, resulting from interpolation and kernel overlap [15, 19, 30, 41]. In general, an up-sampling replicates signal in spectrum domain. To simulate only these artifacts, we optimize the Up-Net model: four blocks of 1x1 [49] convolutional layers, to avoid padding, followed by deconvolutional layers with kernel size of 2 and stride 2. This produces a periodic pattern in image space and dots in the spectrum domain (Fig. 1).

Boundary artifacts cause a grid-like structure in spectrum domain. When applying FFT to an image, it assumes periodicity, but if image is not periodic, a "cross" artifact will appear as a result of the image's non-periodicity [39]. Our target gray image, which is periodic, does not exhibit this artifact, but the reconstruction does. This phenomenon is rooted in image padding and the mechanism of convolution layers, which are known to impact CNN performance [1,29]. To simulate this, we optimize the C-Net model with eight convolutional layers, each having a kernel size of 3, a stride of 1, and a padding of 1. This configuration helps preserve the spatial dimensions of the input and induces artifacts. Addition of up-sampling replicates "cross" structure resulting in grid structure in spectrum domain.

Consequently, in parallel to the Deep Image Prior [51], where a CNN's structure was shown to be an image prior, here we have shown that CNN's structure is also artifact prior. As such, we term our method *deep image fingerprint* (DIF), and describe it next.

3.2. Fingerprint Extraction

We can extract fingerprints of a target model by an optimization of θ given a set of generated images and a set of arbitrary real images. The optimization is similar to the denoising procedure [51], where $\mathcal{F}$ is acquired by Eq. (3), but instead of computing mean square error loss with respect to some image we compute correlation with respect to a set of image residuals. Residual $R_i \in \mathbb{R}^{3\times H\times W}$ is defined as $R_i = f_D(X_i)$, where (X_i) is an image and f_D is a denoiser filter [37].

The goal is to produce fingerprint that is highly correlated with residuals of generated images, and non-correlated with residuals of real images. Pearson Correlation Coefficient is proved to be a good correlation metric between image residual and model's fingerprint [37, 54]. In practice we transform each input into their zero-mean and unit-norm versions, preform inner product and average values. This will be referred simply as *correlation* and denoted as $\rho(\cdot, \cdot)$.

The loss function is formulated similarly to contrastive loss in siamese setting [24]. We define a similarity factor $t_{ij} \in \{0, 1\}$, which is equal to 1 when two residuals are from the same class and 0 otherwise and correlation distance (D_{ij}) as euclidean distance between correlation values:

$$D_{ij} = \sqrt{(\rho(R_i, \mathcal{F}) - \rho(R_j, \mathcal{F}))^2} \quad (5)$$

Finally the sample loss function ($\mathcal{L}_{\mathcal{S}}$) is summarised below:

$$\mathcal{L}_{\mathcal{S}} = \frac{t_{ij} \cdot D_{ij} + (1 - t_{ij}) \cdot (m - D_{ij})}{m} \quad (6)$$

m is a hyper parameter.

3.3. Detection of Generated Images

We perform detection through hypothesis testing. After generating the $\mathcal{F}$ from the trained model ($g_{\hat{\theta}}$), the model itself (and a GPU) is no longer necessary. Then, we compute the means of the populations of real and fake correlations according to Eq. (7), denoted as $\mu_{\boldsymbol{r}}$ and $\mu_{\boldsymbol{g}}$, respectively. $N_{\boldsymbol{r}}$ and $N_{\boldsymbol{g}}$ represent the number of real and generated images in the training set. For each test image, we extract its residual $R_{\text{test}} = f_D(X_{\text{test}})$ and then perform hypothesis testing as follows: if $|\rho(R_{test}, \mathcal{F}) - \mu_{\boldsymbol{g}}| < |\rho(R_{test}, \mathcal{F}) - \mu_{\boldsymbol{r}}|$, the tested image is considered generated, otherwise, it is considered real or not produced by the target model. This procedure does not require any parameters, such as threshold.

$$\mu_{\boldsymbol{r}} = \frac{1}{N_{\boldsymbol{r}}} \sum_{i \in \boldsymbol{r}} \rho(R_i, \mathcal{F}) \ , \ \ \mu_{\boldsymbol{g}} = \frac{1}{N_{\boldsymbol{r}}} \sum_{i \in \boldsymbol{g}} \rho(R_i, \mathcal{F}) \quad (7)$$

3.4. Implementation Details

In all of our experiments (Sec. 4), we train and test the method in a similar manner. Margin is constant $m = 0.01$ and z has 16 channels. f_D is a pre-trained DnCNN model [55] that is trained separately on real images (Laion-5B dataset [48]) with sigma range of $[5, 15]$ and crop size 48×48 pixels. During the extraction procedure, f_D 's weights are not updated. The extraction model is a U-net [45]. Optimization was carried out using Adam [35], with a constant learning rate of $5e^{-4}$. During training, the fingerprint was accumulated using exponential moving averaging. We provide additional details about the selection of f_D , the training process of DnCNN, and the U-Net architecture in our supplementary materials.

4. Experiments

This study includes a series of experiments that involve a varied collection of images generated by different LTIMs and GANs. The datasets are summarized in Sec. 4.1. The detection results are presented in Sec. 4.2, which is divided into two parts: the detection of images produced by LTIMs and the detection of GAN-generated images. In Sec. 4.3, we investigate the effect of image generator training and fine-tuning on its fingerprint, and in Sec. 4.4, we analyze the relationship between selected LTIMs. Finally, in Sec. 4.5, we test the method on compressed images.

4.1. Detection Data

Our data includes generated images from a variety of LTIM and GAN models. In contrast to LTIMs, that produce multi-domain images, GANs are often limited to a specific image domain that they were trained on. Therefore, some of the GAN datasets include a number of image domains, each produced by a different model. Tab. 1 summarizes the data. The amount of real and generated images per set is equal. During the experiments we randomly split each dataset into equally sized train and test sets.

Source model	N_I	N_M
CycleGAN	2,600	6
ProGAN$_e$	8,000	20
ProGAN$_t$	80,000	20
BigGAN	4,000	1
StyleGAN	12,000	3
StyleGAN2	16,000	4
GauGAN	10,000	1
StarGAN	4,000	1
SD 1.4	6,000	1
SD 2.1	6,000	1
MJ	6,000	1
DALL·E-2	2,000	1
GLIDE	6,000	1
DALL·E-Mini	6,000	1

Table 1. Our data. We specify the source model, number of images N_I, and number of model variants N_M.

Datasets representing LTIMs were generated by us and is available online. We randomly selected real images and their corresponding captions from the Laion-5B [48] dataset. Then, we used the captions to generate corresponding images using publicly available Stable Diffusion models [44], versions 1.4 (SD 1.4) and 2.1 (SD 2.1), DALL·E-Mini [12] and GLIDE [40]. In the case of DALL·E-2 [43] we generated images with OpenAI's official API[1]. Lastly, a large set of generated images produced by MidJourney (MJ) [27] is publicly available on the Kaggle website [50], from which we select a random subset for this work.

Datasets representing GAN models were obtained from the supplementary materials of Wang et al. [52]. Our supplementary document contains a detailed review of the datasets.

4.2. Detection of Generated Images

We evaluate DIF as a generated image detector and compare it to rule-based methods and data-driven methods. We summarize the results of detecting images produced by LTIMs and GANs in Tabs. 2 and 3, respectively. To represent the number of training samples and the size of the pre-training datasets, we utilize N_S and N_D respectively.

[1]https://platform.openai.com/docs/guides/images

N_P	N_S	Method	SD 1.4	SD 2.1	MJ	DALL·E-Mini	GLIDE	DALL·E-2	Mean
0	1024	Joslin20	49.7	49.9	49.9	52.3	57.0	51.4	51.7
		Marra18	52.6	48.0	75.7	85.3	57.6	56.3	62.6
		Ning18	50.8	51.4	59.5	58.1	57.7	52.2	55.0
		Resnet-50	72.2	72.6	87.1	87.5	94.1	**88.7**	83.7
		Resnet-M	69.2	73.6	89.7	89.4	**94.3**	85.9	83.7
		DIF	**99.3**	**89.5**	**99.0**	**99.0**	90.3	79.5	**92.8**
	512	Resnet-50	70.6	70.5	90.0	85.4	**92.7**	**85.4**	82.4
		Resnet-M	68.3	71.4	81.1	85.2	91.3	83.8	80.2
		DIF	**99.2**	**86.3**	**98.8**	**98.7**	88.2	79.1	**91.7**
	256	Resnet-50	66.2	64.0	86.4	82.0	**89.0**	**81.4**	78.2
		Resnet-M	67.2	65.8	74.4	83.5	88.0	75.3	75.7
		DIF	**98.5**	**81.3**	**98.1**	**98.0**	85.9	77.7	**89.9**
	128	Resnet-50	66.2	64.0	85.9	75.1	**87.6**	75.2	75.7
		Resnet-M	68.0	66.4	72.6	81.0	80.3	71.7	73.3
		DIF	**97.7**	**75.5**	**97.3**	**97.0**	81.4	**76.1**	**87.5**
720k	1024	Wang20	63.7	61.7	75.7	78.3	74.5	74.7	71.4
		Grag21	93.5	86.9	93.5	96.1	**97.5**	**93.5**	93.5
		Corv22	**99.7**	**99.0**	**99.2**	**96.4**	96.0	91.9	**97.0**
	512	Grag21	93.0	86.0	94.7	**96.1**	**96.4**	**92.1**	93.1
		Corv22	**99.6**	**98.6**	**98.8**	95.9	95.1	89.4	**96.2**
	256	Grag21	89.9	82.9	94.1	94.9	**95.6**	**90.5**	91.3
		Corv22	**99.6**	**98.6**	**99.0**	**95.6**	94.8	89.0	**96.1**
	128	Grag21	87.9	82.2	93.4	93.5	**94.2**	**90.4**	90.3
		Corv22	**99.6**	**98.4**	**98.6**	**93.9**	89.1	77.5	**92.9**
	0	Grag21	57.0	50.2	63.1	58.0	54.3	51.2	55.6
		Corv22	**99.3**	**99.3**	**99.0**	**89.5**	**57.0**	**51.6**	**82.6**

Table 2. Classification accuracy (%). N_S and N_P are the amount of train samples and pre-train dataset size. DIF achieves best average accuracy comparing to other rule-based and non pre-trained models even when $N_S = 128$. For SD 1.4, MJ and DALL·E-Mini, DIF is also comparable with the SOTA pre-trained and fine-tuned method (Corv22).

Rule-based methods rely on fingerprint extraction. Marra18 [37] is a conventional method for extracting $\mathcal{F}$ that involves averaging over residuals. Classification is then performed using the extracted $\mathcal{F}$ as described in Sec. 4.2. Joslin20 [31] follows a similar approach to Marra18, but correlation is conducted in the Fourier domain. For these two methods and DIF, we employ the same f_D (Sec. 3.4), as with it they demonstrate superior performance. Further details can be found in the supplementary material. Ning18 [54] involves extracting fingerprints from images using a trainable auto-encoder model and classifying the images with an additional CNN model.

We also compare to data-driven methods: pre-trained Resnet-50 [25] on ImageNet [13], Wang20 [52] is Resnet-50, which is trained on ProGAN images, Grag21 [23] and Corv22 [10] utilize modified Resnet-50 architecture (Resnet-M).

The methods Wang20, Grag21, and Corv22 have previously demonstrated SOTA results in the detection of images produced by GANs and LDMs [10]. Wang20, Grag21, and Corv22 are trained on 720k images each, where half of the dataset consists of real images, while the other half is composed of images generated by ProGAN [32], StyleGAN [33] and LDM [44], respectively. Additionally, we performed fine-tuning on all the aforementioned models using N_S images. In the case of fine-tuning, the models are frozen, and only the last fully-connected layer is reinitialized and updated during training.

According to the results in Tab. 2, the proposed method achieves remarkably high performance with a low number of training samples when applied to images generated by LTIMs. DIF demonstrates higher accuracy compared to other rule-based methods and a higher mean accuracy compared to non pre-trained data-driven methods. Furthermore, when directly comparing DIF to the best fine-tuned method (Corv22), DIF performs equally well in the case of SD 1.4, MJ, and DALL·E-Mini, and outperforms all non fine-tuned models on average, even with $N_S = 128$.

For GLIDE, SD 2.1 and DALL·E-2, the detection accuracy of DIF is lower. $\mathcal{F}$ is formed by model architecture, weight initialization, and train data (Sec. 2). Refer to Fig. 2. In contrast to SD 1.4, the $\mathcal{F}_A$ of GLIDE exhibits a barely visible grid pattern that gets lost within the accumulated noise. This makes it a "weak" fingerprint, characterized by lower energy. Similar “weak” fingerprint is also observed in SD 2.1. Despite sharing architecture with SD 1.4, SD 2.1 was trained on different dataset[2] and possibly with different weight initialization.

Similar results are obtained when evaluating images generated by GANs, as shown in Tab. 3. DIF surpasses all rule-based methods and performs comparably to the best pre-trained model (Grag21) in four out of seven GAN models. The lowest detection accuracy is observed for the ProGAN$_e$ dataset, which is expected. As mentioned in Sec. 2, each trained model produces unique fingerprints, and DIF is specifically designed to detect images from a single model. For completeness, we also measure the detection accuracy for ProGAN$_t$, where DIF is trained on images of each model separately. In this case, the mean accuracy is 91.2%, with the highest accuracy reaching 96.0% and the lowest accuracy being 83.2%. Similarly to the case of SD models, we observe that not only model architecture affects it's $\mathcal{F}$. A comprehensive evaluation for ProGAN$_t$ is provided in our supplementary materials.

Summarizing the experiments, DIF exhibits strong performance with both novel and well-established image generation methods. We have demonstrated that DIF outperforms other methods trained under the same conditions on average, in the detection of images generated by both LTIMs and GANs. Additionally, DIF performs comparably to pre-trained SOTA methods (Grag21 and Corv22). It is worth noting that DIF achieves these results with just a few hundred images, while pre-training typically requires three orders of magnitude more images and additional fine-tuning. Therefore, we consider DIF to be a formidable competitor to existing methods.

4.3. Detection of Images From Fine-Tuned Models

Now consider a more challenging setting, where given a trained DIF for some image generator we aim to detect images generated by its fine-tuned version. Due to uniqueness of fingerprints (Sec. 2) images generated by these new model variations might not be detected by DIF trained on images produced by the original model.

To investigate the relation between source models and their variations we conduct an experiment using varied datasets and checkpoints. First, we train four ProGAN models ($P_A, P_B, \hat{P}_A$, and $\hat{P}_B$) for 70 epochs. P_A and P_B are trained on 2,500 “cat” class images from AFHQ [34] with random seeds A and B respectively. $\hat{P}_A$ and $\hat{P}_B$ are trained with the same random seeds, but on 2,500 “wild” class images from AFHQ. Next, for each model we use five checkpoints at epochs 20, 32, 40, 52 and 70, and generate 2,500 images for each. Finally, we construct 20 datasets by adding real images from the corresponding train set of ProGANs to each set of generated images. This results in 20 ProGAN models with 20 datasets.

We preform a cross-detection on $P_A, P_B, \hat{P}_A$ and $\hat{P}_B$ and for brevity summarize results for P_A and P_B in Fig. 3. The full cross-detection map, including comparison to cross-correlation of fingerprints, is available in our supplementary materials. During cross-detection we attempt to classify images produced by some image generator with DIF which is trained on images from another image generator. Observe the symmetric relation within the same model: for checkpoints of epochs 40, 52 and 70 cross-detection accuracy is high and symmetric. Other relations include low accuracy and/or asymmetric, which is exactly what is expected for DIF trained on unique $\mathcal{F}$-s. We may conclude that the model's $\mathcal{F}$ changes during training. However, as the model converges, these changes become insignificant, resulting in high cross-detection accuracy for DIF.

We conduct an additional experiment where we measure cross-detection on images from a number of fine-tuned/“personalized” stable diffusion models [46], which involve updates of image decoder weights. The models are: our custom fine-tuned SD 1.4 with a small set of images and two downloaded models of stable diffusion v1.5 and v2.0 fine-tuned on a large set of anime images[3] and robot images[4]. Models denoted as SD 1.4S, SD 1.5A and SD 2.0R respectively. For each model we generated 1000 images from the same caption set that was used with SD 1.4, including style/object keywords specific to each model. Then train DIF for each with 512 real and 512 generated images (Sec. 4.2). Upon analyzing the results presented in Fig. 4, we can conclude that the relationships observed in previous experiments are maintained even when the model is fine-tuned with new data. Consequently DIF will also perform well on images generated by fine-tuned models.

4.4. Model Lineage Analysis

We can use our extracted fingerprints to detect the lineage of trained models. In Fig. 5 we show cross-detection results for LTIMs. We observe that SD 1.4 and MJ produce high-cross detection accuracy, thus MJ is likely to be a fine-tuned version of SD 1.4. Indeed, while this is not public knowledge, we found evidence that our analysis is correct[5].

In contrast, SD 2.1 does not retain such a relation with both SD 1.4 and MJ, therefore we can conclude that this model was trained from scratch. Indeed, this was confirmed by the SD 2.1 developers[3].

4.5. Robustness

To test our method's robustness to image compression, we use two models: SD 1.4 and GLIDE, representing strong

[2] https://github.com/Stability-AI/stablediffusion

[3] https://huggingface.co/DGSpitzer/Cyberpunk-Anime-Diffusion

[4] https://huggingface.co/nousr/robo-diffusion-2-base

[5] https://tokenizedhq.com/midjourney-model/

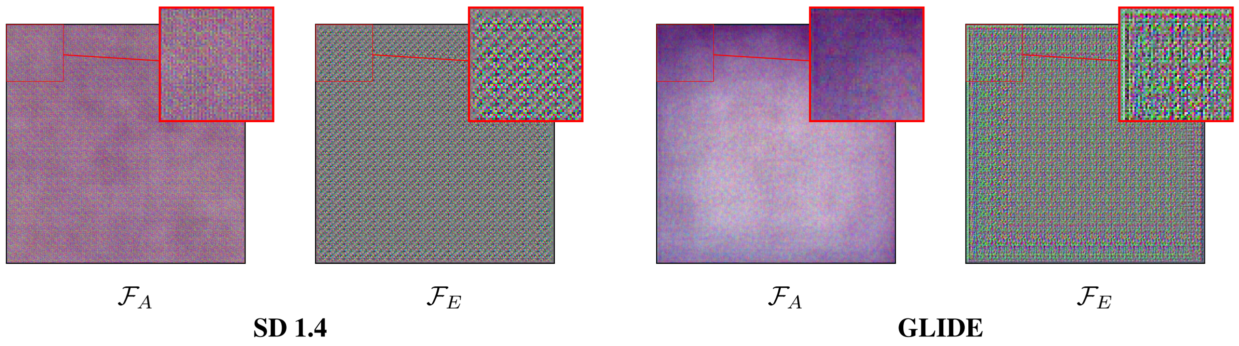

Figure 2. Fingerprints of SD 1.4 and GLIDE. Per each model two types of fingerprint are shown: fingerprint by residual averaging ($\mathcal{F}_A$) and extracted by DIF ($\mathcal{F}_E$). Observe the $\mathcal{F}_A$ of each model: SD 1.4 demonstrates strong grid-like pattern, whereas GLIDE shows none - GLIDE has a "weak" fingerprint. In contrast to $\mathcal{F}_A$, $\mathcal{F}_E$ reveals clear patterns of SD 1.4 and GLIDE.

N_P	N_S	Method	CycleGAN	StyleGAN	StyleGAN2	StarGAN	BigGAN	GauGAN	ProGAN$_e$	Mean
0	1024	Joslin20	51.6	72.5	52.4	68.6	50.8	53.9	50.2	57.1
		Marra18	58.8	82.6	50.9	92.5	54.8	51.5	48.5	62.8
		Ning18	49.5	61.7	57.2	64.1	57.7	55.5	50.2	56.6
		DIF	**94.4**	**96.6**	**91.5**	**99.9**	**96.9**	**91.8**	**57.7**	**89.8**
720k	1024	Wang20	91.7	94.1	94.0	95.6	85.7	95.0	**100**	92.6
		Corv22	93.0	94.8	92.6	98.3	93.0	95.9	95.1	94.7
		Grag21	**98.0**	**100**	**100**	**100**	**98.2**	**98.5**	**100**	**98.8**
	0	Wang20	84.6	76.5	72.2	84.7	59.4	82.9	**100**	91.9
		Corv22	50.6	59.8	51.2	45.7	51.9	46.3	51.2	51.0
		Grag21	**93.5**	**100**	**100**	**99.9**	**96.5**	**90.9**	**99.9**	**97.2**

Table 3. Classification accuracy (%). N_S and N_P are the amount of train samples and pre-train dataset size. DIF achieves best result comparing to rule-based methods and is on par with pre-trained models. Low accuracy with DIF for ProGAN$_e$ is expected as it is a mix of images generated from 20 different models.

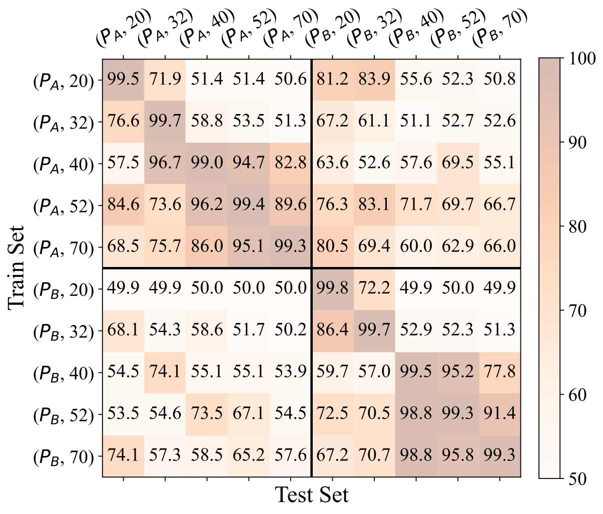

Figure 3. Cross-detection accuracy in percents. Each grid characterized by model and epoch. Observe clusters for epochs 40,52,70.

and "weak" $\mathcal{F}$-s, respectively. We created four additional datasets for each model: images compressed using JPEG at quality levels of 75 (J75) and 50 (J50), resized (R) images, and blurred (B) images. Uncompressed images are denoted as U. The resized images were down-sampled to half of their original size and then up-sampled back to their original size using nearest-neighbor interpolation. We applied blur with a sigma value of 3, resulting in heavy smoothing.

We train DIF separately on each compression set for each model, following the procedure outlined in Sec. 4.2, using 256 training images each time. To emulate a more realistic scenario in which the compression type is not known, we also train the model on a mixed dataset where images are randomly compressed during the training procedure.

The detection accuracy for each compression type in reported in Figs. 6a and 6b for SD 1.4 and GLIDE, respectively. We observe that the blur significantly reduces the extraction and detection capabilities of the method because the $\mathcal{F}$ signal resides on higher frequencies of the image. In other cases, the accuracy is reduced, but this depends on the characteristics of the fingerprint.

We hypothesize that the detection of compressed images

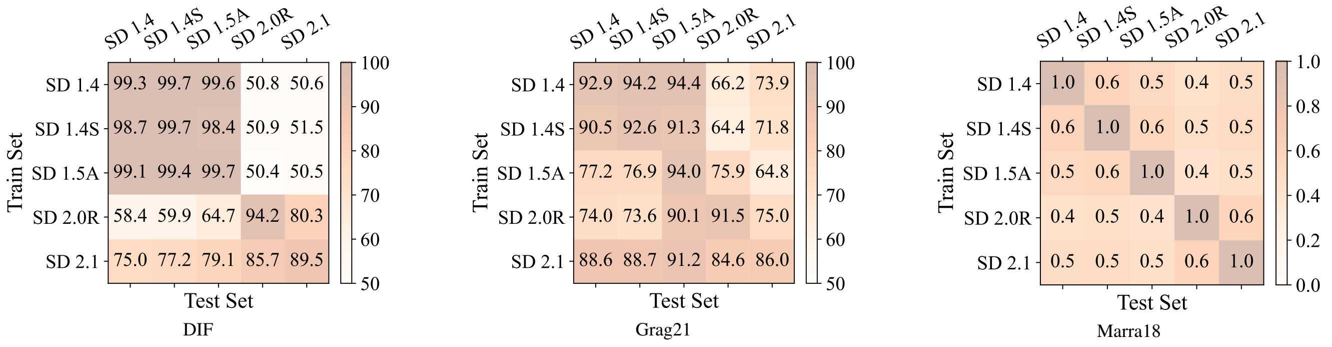

Figure 4. Model lineage analysis of SD models with different detection methods. For DIF and Grag21 we show cross-detection (%) and for Marra18 cross-correlation of fingerprints. Notably, only DIF exhibits clusters of SD 1.x and SD 2.x with high and symmetric cross-detection.

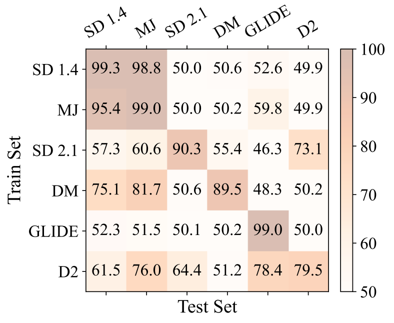

Figure 5. Model lineage analysis of LTIMs with DIF by cross-detection (%). DM and D2 denote DALL·E-Mini and DALL·E-2, respectively. Relation between SD 1.4 and MJ is similar to relation between SD 1.x models.

(Fig. 4).

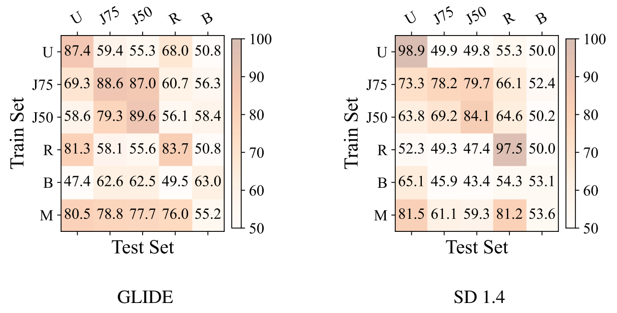

Figure 6. Detection and cross-detection accuracy per compression for images generated by GLIDE and SD 1.4.

is influenced by the original pattern of $\mathcal{F}$. For example, we can observe different detection behaviors for resized images in both models. DIF trained on GLIDE's resized images can easily detect uncompressed images, but in the same scenario, where DIF is trained on images produced by SD 1.4, it is unable to detect uncompressed images. Additionally, we can barely observe symmetry in cross-detection accuracy per model. This is a clear sign of changes within $\mathcal{F}$ that result from compression.

The above is only a brief analysis of the effect of compression on $\mathcal{F}$. It appears to be a complex topic that requires a more comprehensive investigation, which we leave for future work.

5. Conclusions and Future Work

This study provides a twofold contribution: the development of new methods for synthetic image detection and the establishment of a methodology for model lineage analysis. We have shown that CNNs naturally exhibit image artifacts, which we leverage in our method called DIF. DIF extracts fingerprints from generated images, allowing us to detect images that come from the same model or its fine-tuned versions. Our method achieves high detection accuracy, surpassing methods trained under the same conditions and performing similarly to pre-trained state-of-the-art detectors for generated images from popular models. Remarkably, we achieve these results using a small number of generated images (up to 512), while other detectors require significantly more samples for training.

In terms of model lineage analysis, we employ cross-detection as a means to trace fine-tuned generative models. Notably, our analysis reveals that MidJourney is indeed a fine-tuned variant of the Stable Diffusion 1.x model.

However, we identified several drawbacks that require further investigation. Some image generators produce weak fingerprints, which are challenging for DIF. Furthermore, the method performs poorly on certain compression methods and blurred images. We believe that the effect of compression on fingerprint extraction and detection requires additional attention and is a topic for future research.

References

[1] Bilal Alsallakh, Narine Kokhlikyan, Vivek Miglani, Jun Yuan, and Orion Reblitz-Richardson. Mind the pad – {cnn}s can develop blind spots. In *International Conference on Learning Representations*, 2021. 3

[2] Omri Avrahami, Kfir Aberman, Ohad Fried, Daniel Cohen-Or, and Dani Lischinski. Break-a-scene: Extracting multiple concepts from a single image. *arXiv preprint arXiv:2305.16311*, 2023. 2

[3] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *ACM Trans. Graph.*, 42(4), jul 2023. 1

[4] Omri Avrahami, Thomas Hayes, Oran Gafni, Sonal Gupta, Yaniv Taigman, Devi Parikh, Dani Lischinski, Ohad Fried, and Xi Yin. Spatext: Spatio-textual representation for controllable image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18370–18380, June 2023. 1

[5] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18208–18218, June 2022. 1

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv*, 2019. 1

[7] Keshigeyan Chandrasegaran, Ngoc-Trung Tran, Alexander Binder, and Ngai-Man Cheung. Discovering transferable forensic features for cnn-generated images detection. *arXiv*, 2022. 1, 2

[8] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018. 1

[9] Riccardo Corvi, Davide Cozzolino, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. Intriguing properties of synthetic images: from generative adversarial networks to diffusion models, 2023. 2

[10] Riccardo Corvi, Davide Cozzolino, Giada Zingarini, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. On the detection of synthetic images generated by diffusion models. *arXiv*, 2022. 1, 2, 5

[11] Davide Cozzolino, Diego Gragnaniello, Giovanni Poggi, and Luisa Verdoliva. Towards universal gan image detection. *arXiv*, 2021. 1

[12] Boris Dayma, Suraj Patil, Pedro Cuenca, Khalid Saifullah, Tanishq Abraham, Phuc Le Khac, Luke Melas, and Ritobrata Ghosh. Dall·e mini, 7 2021. 4

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 5

[14] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *arXiv*, 2021. 1

[15] Ricard Durall, Margret Keuper, and Janis Keuper. Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions. *arXiv*, 2020. 3

[16] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. *arXiv*, 2020. 1

[17] Hany Farid. Lighting (in)consistency of paint by text. *arXiv*, 2022. 1

[18] Hany Farid. Perspective (in)consistency of paint by text. *arXiv*, 2022. 1

[19] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. *arXiv*, 2020. 2, 3

[20] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion, 2022. 2

[21] Manjary P Gangan, Anoop K, and Lajish V L. Distinguishing natural and computer-generated images using multi-colorspace fused efficientnet. *arXiv*, 2021. 1, 2

[22] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv*, 2014. 1

[23] 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. *arXiv*, 2021. 1, 2, 5

[24] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742, 2006. 3

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv*, 2015. 2, 5

[26] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv*, 2020. 1

[27] David Holz. Midjourney, 2012. 2, 4

[28] Kurt Hornik, Maxwell B. Stinchcombe, and Halbert L. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989. 1

[29] Carlo Innamorati, Tobias Ritschel, Tim Weyrich, and Niloy Jyoti Mitra. Learning on the edge: Investigating boundary filters in cnns. *International Journal of Computer Vision*, 128:773–782, 2019. 3

[30] Yonghyun Jeong, Doyeon Kim, Youngmin Ro, Pyounggeon Kim, and Jongwon Choi. Fingerprintnet: Synthesized fingerprints for generated image detection. In *ECCV*, 2022. 2, 3

[31] Matthew Joslin and Shuang Hao. Attributing and detecting fake images generated by known gans. In *2020 IEEE Security and Privacy Workshops (SPW)*, pages 8–14, 2020. 1, 2, 5

[32] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv*, 2018. 1, 5

[33] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4217–4228, 2021. 1, 5

[34] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8107–8116, 2020. 1, 6

[35] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 4

[36] Haodong Li, Bin Li, Shunquan Tan, and Jiwu Huang. Identification of deep network generated images using disparities in color components. *Signal Processing*, 174:107616, sep 2020. 1, 2

[37] Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. Do gans leave artificial fingerprints? *arXiv*, 2018. 1, 2, 3, 5

[38] Scott McCloskey and Michael Albright. Detecting gan-generated imagery using color cues. *arXiv*, 2018. 1, 2

[39] Lionel Moisan. Periodic plus smooth image decomposition. *Journal of Mathematical Imaging and Vision*, 39:161–179, 2011. 3

[40] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, 2021. 1, 4

[41] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016. 2, 3

[42] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015. 1

[43] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. 1, 4

[44] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *arXiv*, 2021. 1, 2, 4, 5

[45] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *arXiv*, 2015. 3, 4

[46] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *arXiv*, abs/2208.12242, 2022. 2, 6

[47] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv*, 2022. 1

[48] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022. 4

[49] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. 3

[50] Iulia Turc and Gaurav Nemade. Midjourney user prompts and generated images (250k), 2022. 4

[51] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Deep image prior. *CoRR*, abs/1711.10925, 2017. 1, 2, 3

[52] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A. Efros. Cnn-generated images are surprisingly easy to spot... for now. *arXiv*, 2019. 1, 2, 4, 5

[53] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *arXiv*, 2022. 1

[54] Ning Yu, Larry Davis, and Mario Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. *arXiv*, 2018. 1, 2, 3, 5

[55] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, Jul 2017. 4

[56] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. Detecting and simulating artifacts in GAN fake images. *arXiv*, 2019. 2

[57] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv*, 2017. 1