MaskSim: Detection of synthetic images by masked spectrum similarity analysis

Yanhao Li[1] Quentin Bammey[1] Marina Gardella[2] Tina Nikoukhah[1]
Jean-Michel Morel[3] Miguel Colom[1] Rafael Grompone von Gioi[1]

[1]Université Paris-Saclay, ENS Paris-Saclay, CNRS, Centre Borelli, Gif-sur-Yvette, 91190 France
[2]Instituto de Matemática Pura e Aplicada, Rio de Janeiro, Brazil
[3]City University of Hong Kong, Department of Mathematics, Kowloon, Hong Kong

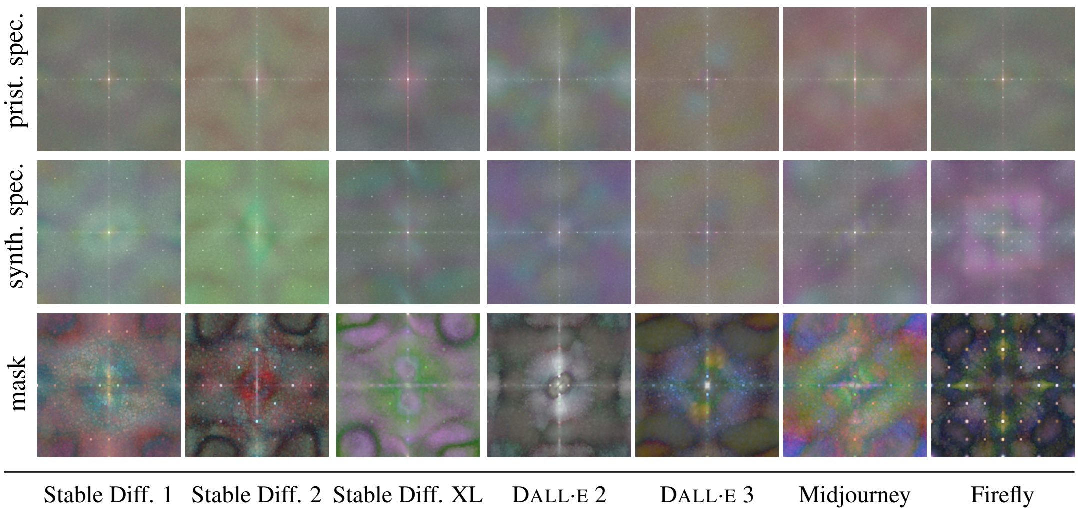


Figure 1. We extract spectral traces left by generative models to distinguish real and AI-generated images. For each generative model (columns), we compute an average FFT magnitude spectrum for real (top row) and generated (middle row) images, and learn a mask to amplify the specific frequencies that help make the distinction. The preprocessing before computing the spectrum is trained separately on each model, hence the masks being different for each model. The three-channel masks are visualized as RGB images (see Fig. 3).

Abstract

Synthetic image generation methods have recently revolutionized the way in which visual content is created. This opens up creative opportunities but also presents challenges in preventing misinformation and crime. However, these methods leave traces in the Fourier spectrum that are invisible to humans, but can be detected by specialized tools. This paper describes a semi-white-box method for detecting synthetic images by revealing anomalous patterns in the spectral domain. Specifically, we train a mask to enhance the most discriminative frequencies and simultaneously train a reference pattern that resembles the patterns produced by a given generative method. The proposed method produces explainable results with state-of-the-art performances and highlights cues that can be used as forensic evidence. Code is available at `https://github.com/li-yanhao/masksim`.

1. Introduction

The emergence of synthetic images represents a paradigm shift in the landscape of visual content creation, ushering in both innovative possibilities and significant challenges for society. Synthetic images, often generated through advanced techniques such as Generative Adversarial Networks (GANs) or Diffusion Models (DMs), have the potential to revolutionize various industries, including entertainment, design, and marketing. However, alongside these opportunities, the leap of synthetic images has given rise to substantial threats to society. One of the foremost concerns is their use as fake evidence. Indeed, synthesized content can convincingly depict events or individuals who never existed. It is therefore very important to characterize their nature and to detect them automatically, to cope with visual disinformation in social networks, and also to serve for authenticity verification in court.

Recent progress in image generation has increased dramatically the quality of synthetic images [20, 62, 65, 66], with more and more new models being released continuously. Although the detection of synthetic images is attainable when they come from a known generation source, generalization to images of unknown sources remains poor. Previous works [10, 13, 33, 72] studying the generalization of synthetic image detection suggest that different generative models result in related and identifiable artifacts, showing the possibility of training a detector on one generator and generalizing to another. Nevertheless, the detection performance strongly relies on the artifact similarity between the images used for training and for inference, which still remains a challenging problem. In addition, the inherent opacity of neural networks does not bring in the transparent cues needed to support forensic conclusions.

The Fourier spectrum of synthetic images may contain cues enabling their detection with generalization ability. As observed in [24], generative models show systematic shortcomings in replicating high-frequency characteristics of pristine images. Generally, such models privilege the reconstruction on some specific frequencies over the rest and fail to correctly reproduce spectral distributions [21]. Several studies [12, 77] demonstrate that the upsampling operations in the decoder of a generative model leave distinctive patterns that are traceable in the frequency domain. This has also been observed in many other studies [13, 33, 46, 75]. Similar to the photo response non-uniformity (PRNU) of cameras [49], these patterns can be seen as the fingerprints of generative models. Although acknowledging the presence of frequency domain artifacts in generative models, only a few existing methods [4, 75] in the literature have attempted to detect images of DMs in the frequency domain.

In this paper, we propose a semi-white-box method to detect synthetic images by revealing the abnormal spectrum patterns left by DMs. More specifically, we enhance the synthesis artifacts using a Convolutional Neural Network (CNN) denoising filter [76], then we train a mask to amplify the abnormal patterns in the spectrum and, simultaneously, we train a reference pattern that resembles the amplified patterns of the generation model, see Fig. 1. The rationale is that generation models share similar spectral patterns related to their limited decoding capacity. The trainable mask and reference pattern explicitly reveal the artifacts in the spectrum of each generative model, so as to provide explainable forensic evidence. Overall, the proposed methods establishes a new state of the art in synthetic image detection, and paves the way towards explainable and generalizable AI-generated image detection.

2. Related works

2.1. Synthetic Image Generation

The landscape of synthetic image generation has undergone a revolution with the emergence of generative deep-learning frameworks such as Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), and Diffusion Models (DMs). GANs [32] are generative models that transform low-dimensional randomly sampled latent vectors into photorealistic images. Such models are trained using adversarial learning. Well known GANs for image generation include DCGAN [60], BigGAN [8], GauGAN [56], ProGAN [38], StyleGAN [40], StyleGAN-2 [41], StyleGAN-3 [39] and EG3D [11].

While GANs have significantly shaped the realm of image generation, their prominence has recently been eclipsed by DMs [69]. These models conceptualize the distribution of data as a diffusion process, progressively altering the image using a straightforward prior and gradually restoring it to the desired distribution. Noteworthy among these is the Ablated Diffusion Model (ADM) [17], which has surpassed the capabilities of both GANs and VAEs in the field of image generation. This marks a turning point in the evolution of DMs. Concurrently, transformer models [71] have experienced a surge in applications within computer vision. This surge is largely attributed to the emergence of CLIP [59], a model proficient in embedding both images and text into a shared space. Leveraging this capability, Latent Diffusion models [64], Stable Diffusion models (SD) [65], Glide [54], CogView [18], Make-A-Scene [27], DALL·E [62] and Imagen [66] have extended the scope of diffusion models to generate images from text prompts within a latent feature space. This development represents a significant advancement in the capabilities of image generation, enhancing both the diversity and photorealism of synthesized images.

However, the rapid progress in image generation has given rise to societal concerns, particularly the menace of deepfakes, which represent significant security risks. The imperative to develop robust techniques for detecting synthetic images and mitigating their potential misuse cannot be emphasized enough.

2.2. Synthetic image detection

The primary focus of this paper is the detection of synthetic images. Such an area has recently emerged as a hit research field alongside the rapid progress in realistic image generation. AutoGAN [77] employs a classifier in the spectral domain to identify synthetic images based on their frequency artifacts. Dzanic et al. [24] showed the systematic shortcomings of deep networks in replicating correctly the high-frequency modes, and proposed to use a K-nearest neighbor classifier based on the frequency spectrum for detection. PatchForensics [10] delved into the distinctive proper-

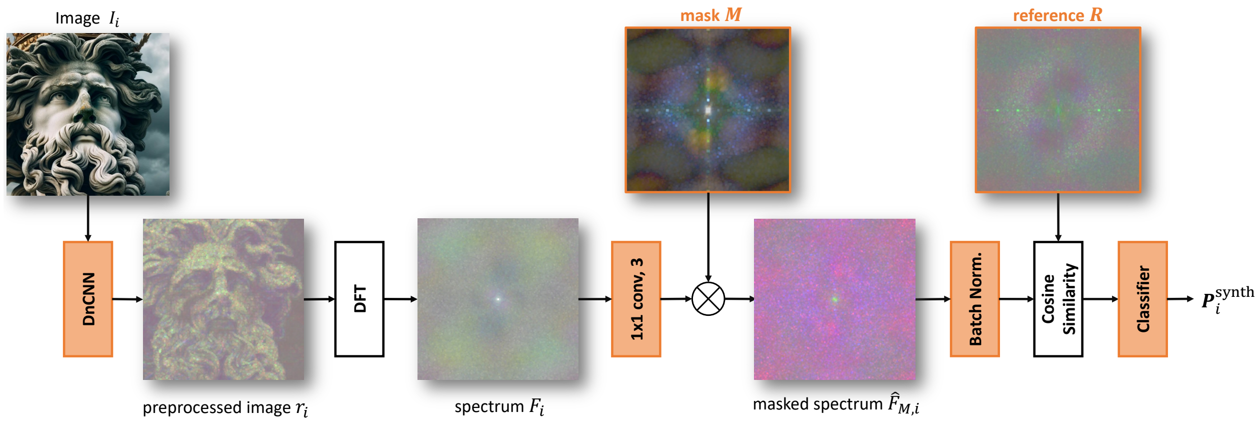


Figure 2. Flowchart of our proposed method for computing the spectrum similarity between a reference and an image within a mask, and predicting the synthesis probability accordingly. The cropped image is preprocessed by a filter, transformed by DFT, enhanced by a 1x1 convolution layer, element-wise multiplied with a mask, and compared with a spectrum reference to compute their similarity. The similarity score is subsequently used for computing the synthesis probability through a simple logistic regression classifier. The modules with trainable parameters are colored in orange.

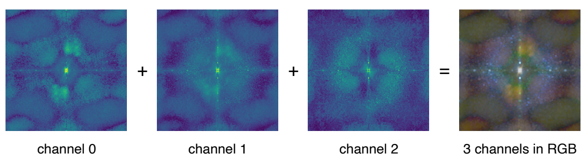


Figure 3. The 3-channel feature map visualized by RGB image.

ties of counterfeit images, especially face images, that make them detectable. It discerns patterns that generalize across various model architectures, datasets, and training modifications. McCloskey and Albright [51] leveraged the observation that the intensity values of synthetic images seldom reach saturation. He et al. [37] proposed a detection framework to re-synthesize tested images and extract visual cues for GAN-generated images detection. Wang et al. [72] and Gragnaniello et al. [33] trained CNNs to discriminate between pristine and GAN-generated images. Liu et al. [46] proposed a simple classifier using noise patterns. Mandelli et al. [50] suggested incorporating several CNN classifiers trained in an orthogonal scheme as an ensemble detector of GAN-generated images. However, these studies were conducted before the widespread adoption of DMs and text-to-image techniques.

Some recent methods have been proposed to specifically tackle the detection of images generated by diffusion models. Corvi et al. [13] retrained the existing architecture of Gragnaniello et al. [33] on DM-generated images. Ojha et al. [55] trained a network to distinguish pristine and fake images in the latent domain of a CLIP-trained architecture [19]. Similarly, Cozzolino et al. [14] discovered that with only a handful of example images from a single generative model a CLIP-based detector exhibits a surprising generalization ability and high robustness across different architectures. Zhang et al. [75] proposed a deep learning approach using the information in the frequency domain. However, they only trained and tested on images generated by SD. DIRE [73], on the other hand, proposed a novel image residual which measures the error between an input image and its reconstructed version by a pre-trained diffusion model; then, a simple binary classifier makes the decision based on such residuals. Yan et al. [74] proposed to disentangle method-specific and common synthetic features using a multi-task learning strategy. The detection scheme presented in Artifact [61] tackles the generalization problem using a multi-class scheme. CIFAKE [7] processed binary detection with a CNN classifier and explored useful features for detection via Gradient Class Activation Mapping [68]. Lorenz et al. [48] conducted a thorough investigation of the multi-local intrinsic dimensionality method, originally developed for detecting adversarial examples, and validated its detection capability for synthetic images. Arruda evaluated the effectiveness of ConvNeXt [47] and Learned Noise Patterns extraction originated from [46] for detecting synthetic images. Recently, Epstein et al. [26] stated that a classifier regularly retrained on new generators, has the opportunity to detect future, unreleased models, as long as they are architecturally similar. Synthbuster [4] proposed to highlight the artifacts left by the diffusion process in the Fourier transform of a residual image and to use manually selected frequency components for synthetic image detection. As the analysed frequency components are manually selected, the method may need to be adapted for newer, different methods.

3. Proposed method

As pointed out by Corvi et al. [13], synthetic images generated by GANs or DMs have specific fingerprints that depend on the architecture and the parameters of the generative network. Such fingerprints can be seen in the frequency spectrum of the image residual [12, 13]. Following this observation, we aim at extracting the characteristic fingerprint of each generative method. Still, not all the frequencies provide informative clues. Furthermore, the behaviour of some frequencies could be shared by synthetic and pristine images. Our goal therefore is to find the peculiar regions of the frequency domain containing the most distinctive artifacts left behind by generative models.

To this end, we train a multiplicative mask to amplify the spectra of the synthetic images in certain zones that provide the most informative cues for each generation model (see Fig. 1 with the color-map explained in Fig. 3). Besides, we also train a reference pattern to which the spectra of synthetic images should be similar. Fig. 2 summarizes the workflow of our approach.

Given an input image I_i in three channels, a preprocessing filter f is applied to the image to enhance the artifacts. Previous research has shown that, with a denoiser such as DnCNN [76], the synthesis artifacts are better exposed in the frequency domain. Following this idea, we also adopt DnCNN as the preprocessing filter, and we fine-tune it along with other modules during training. We denote by

$$r_i = f(I_i; \theta) \in \mathbb{R}^{3\times h\times w} \tag{1}$$

the processed image, where θ denotes the filter parameters to be fine-tuned during training and h and w is the height and the width of r_i, respectively. We use the YCbCr color space in order to be coherent with the space where the image compression (e.g. JPEG and WEBP) is processed.

Then, the image spectrum is computed as

$$F_i = \log \left|\mathrm{DFT}(r_i)\right|, \tag{2}$$

where DFT is the Discrete Fourier Transform applied separately to each channel, and $|\cdot|$ computes the magnitude on each pixel. In practice, we use the differentiable FFT algorithm available in PyTorch [57] to compute the DFT.

The spectrum is then enhanced by a 3-channel 1x1 convolution layer, and is element-wise multiplied with a trainable mask $M \in [0,1]^{3\times h\times w}$. The aim is to focus on the frequencies that contribute the most to discriminating pristine and synthetic images, while neglecting uninformative frequencies. A batch normalization is applied to normalize the multiplied spectrum:

$$\hat{F}_{M,i} = \mathrm{BatchNorm}\left(\mathrm{Conv1x1}(F_i) \odot M\right), \tag{3}$$

where $\odot$ is the Hadamard product. Again, for 3-channel images, a separate normalization is applied to each single channel. This normalization step is helpful to amplify the difference between the respective similarities of pristine and fake images.

A second trainable element is the reference spectrum $R \in \mathbb{R}^{3\times h\times w}$, used to compare with each enhanced spectrum. The reference spectrum is channel-wisely normalized by centering as

$$\hat{R} = R - \overline{R} \in \mathbb{R}^{3\times h\times w}, \tag{4}$$

where $\overline{R} \in \mathbb{R}^3$ is the channel-wise means of R. Then, we compute the cosine similarity between the enhanced image spectrum $\hat{F}_{M,i}$ and the normalized reference spectrum $\hat{R}$:

$$\mathrm{CosSim}(\hat{F}_{M,i}, \hat{R}) = \frac{\hat{F}_{M,i} \cdot \hat{R}}{||\hat{F}_{M,i}||_2 \; ||\hat{R}||_2}, \tag{5}$$

where $\cdot$ is the dot product between two vectorized matrices and $||\cdot||_2$ is the L^2 norm.

The objective is to maximize the similarity score for synthetic images and to minimize it for pristine images. Note that the cosine similarity can be negative, and allows the model to learn a pattern that is negatively correlated to the pristine spectra. This can lead to overfitting to the pristine spectra during training, while we expect M and R to only learn the synthetic patterns. Indeed, a dataset might have bias related to the used cameras and the data processing. If we minimize the similarity scores on pristine spectra of the training set in the negative range, the model might also learn the patterns of the bias of the pristine image dataset. To prevent this, we use the absolute cosine similarity during training for the pristine spectra, so that the model is trained to output the similarity scores close to 0 for pristine images.

The similarity score of image I_i during training is given by

$$\mathrm{sim}_i := \mathrm{CosSim}(\hat{F}_{M,i}, \hat{R}) \cdot y_i \tag{6}$$

$$+ \left|\mathrm{CosSim}(\hat{F}_{M,i}, \hat{R})\right| \cdot (1 - y_i) \tag{7}$$

where y_i is the label associated to image I_i, equal to 0 for pristine images and to 1 for synthetic images from the target model. The uniform similarity score without absolute operation is used during evaluation:

$$\mathrm{sim}_i := \mathrm{CosSim}(\hat{F}_{M,i}, \hat{R}). \tag{8}$$

We compute a similarity score for each of the three channels and take the average as the final similarity outcome.

The logistic regression classifier is adopted to predict the probability that the image I_i belongs to the family represented by R from its similarity score sim_i. Since we designed the similarity score to be close to 1 for synthetic images and close to 0 for pristine images, the predicted probability of synthesis should increase with the similarity score. Taking this into consideration, the classifier is constructed as:

$$P_i^{\text{synth}} = \mathrm{sigmoid}(e^a \, \mathrm{sim}_i + b). \tag{9}$$

where a and b are two trainable parameters.

The whole network is trained using the cross entropy loss

$$L = -\sum_{i=1}^{N} \left[y_i \log(P_i^{\text{synth}}) + (1 - y_i) \log(1 - P_i^{\text{synth}}) \right], \tag{10}$$

where P_i^{synth} is the predicted probability for the image I_i.

The same procedure is repeated to obtain one set of parameters for each generative model.

4. Experiments

We evaluated the proposed method with synthetic images from Synthbuster [4] and PolarDiffShield [3] datasets and pristine images from Mit-5k [9], Raise [15], the curated subset of HDR-Burst [35], Dresden [31] and a subset of COCO [45] dataset. Both Synthbuster [4] and PolarDiffShield [3] datasets cover 7 diffusion models: Stable Diffusion (SD)-1, SD-2, SD-XL, DALL·E 2, DALL·E 3, Midjourney and Firefly, with 1000 images per model.

The Mit-5k dataset contains 5000 processed images saved in TIFF format. The Raise-1k [15] dataset contains 1000 processed images saved in TIFF format. The HDR-Burst [35] dataset contains 153 raw images, which underwent the default processing pipeline provided by Adobe Lightroom[1] and were saved in TIFF format. The Dresden [31] dataset contains 1488 raw images, which were processed with libraw [1] and demosaiced with several demosaicing methods: AICC [22, 23], RI [42], MLRI [43], ARI [53], CDMCNN [25, 70], CS [30], GBTF [58], Alternating Projections [28], HA [34], LMMSE [29] and bilinear demosaicing, as explained in [2, 5, 6]. The used subset of COCO dataset contains 5000 JPEG images.

We used all the pristine images of Mit-5k and Dresden, and half of the pristine images of COCO for training. The validation was processed on HDR-Burst and the other half of COCO. The synthetic images from PolarDiffShield were used both for training and validation. The mixture of different datasets helped prevent the detection model from overfitting on the specific characteristics of the limited camera models and image processing pipelines used for creating the datasets. Balanced resampling was adopted between pristine and synthetic images during training. The test was done using synthetic images from Synthbuster and pristine images from Raise-1k. The Synthbuster [4] dataset was constructed using text prompts matching the Raise-1k [15] images, thus the synthetic images from Synthbuster are semantically similar to the Raise-1k pristine images. Therefore, the test was not biased on the semantics. Detailed data scheme is showed in Tab. 1.

[1] Lightroom version: 7.1.2 arm64 (Dec. 10, 2023)

As the raw pristine images are generally much larger and the COCO images are smaller than the synthetic images, we cropped each pristine image in the maximum square shape and resized to 1024×1024 in order to eliminate the frequency discrepancy due to the resolution difference between pristine images and synthetic images. The dimension of the input image was set to 512×512, thus random cropping at 512×512 was applied to all the training images. JPEG compression with random quality factors between 65 and 100 was also applied to both the images during training in order to enhance the detection robustness to different levels of JPEG compression.

	nb. images	training	validation	test
Mit-5k [9]	5000	✓		
Dresden [31]	1488	✓		
COCO [45]	5000	✓	✓	
HDR-Burst [35]	153		✓	
Raise-1k [15]	1000			✓
PolarDiffShield [3]	1000 per class	✓	✓	
Synthbuster [4]	1000 per class			✓

Table 1. The data scheme for training, validation and test. The top part is for pristine image datasets, and the bottom part is for synthetic image datasets.

Our detector was trained respectively on the images of each diffusion model, resulting in one detector per model $\{\mathcal{D}_m : I_i \mapsto P_i^{\text{synth}}\}$ where m indexes the different diffusion models, I_i an image and P_i^{synth} the probability of synthesis. Our method was studied on three criteria:

- **generalization ability of single detector**: the performance of detecting the images from all the classes with one single detector $\mathcal{D}_m$. Here the detector trained with synthetic images from SD-2 was chosen as it shows the best performance when testing on all the classes of images;
- **generalization ability of merged detector**: the performance of the generalized detector trained on all the classes of synthetic images except the tested class m, for which we merge the detectors of all the classes except m by taking the maximum predicted probability $\mathcal{D}^m_{\text{general}} = \max_{n \neq m} \mathcal{D}_n$, and test the generalized detector $\mathcal{D}^m_{\text{general}}$ on the images from the class m;
- **generic detection ability**: the performance of the generic detector $\mathcal{D}_{\text{generic}}$ merged by taking the maximum predicted probability of all the detectors: $\mathcal{D}_{\text{generic}} = \max_m \mathcal{D}_m$.

We compared our method to the detection methods of UFD [55], Wang [72], Corvi [13], Grag [33], PatchFor [10], and Synthbuster [4]. All of the compared detectors except Synthbuster [4] were trained on other types of synthetic images different from those for test. The Synthbuster detector was trained on synthetic images from PolarDiffShield.

AUC / ACC (%)	SD-1	SD-2	SD-XL	DALL·E 2	DALL·E 3	Midjourney	Firefly	AVG
UFD [55]	67.0 / 54.9	83.1 / 71.2	75.7 / 62.8	<u>90.6</u> / <u>77.0</u>	43.3 / 46.8	50.6 / 48.4	<u>94.5</u> / **84.6**	72.1 / 63.7
Wang [72]	51.5 / 50.0	63.9 / 50.8	60.6 / 50.1	69.5 / 50.3	19.8 / 49.9	38.8 / 49.9	85.3 / 51.2	55.6 / 50.3
Corvi [13]	**100.0 / 99.6**	**99.5 / 97.2**	**98.9** / 80.4	48.8 / 49.9	54.9 / 49.7	**99.8 / 95.0**	86.2 / 52.4	84.0 / 74.9
Grag [33]	85.1 / 56.7	81.0 / 60.5	52.5 / 49.9	69.3 / 50.1	23.3 / 49.8	49.0 / 50.1	**96.1** / 74.0	65.2 / 55.9
PatchFor [10]	55.1 / 50.2	71.3 / 50.1	37.6 / 50.1	42.9 / 50.1	46.4 / 50.0	43.1 / 49.8	39.0 / 49.3	47.9 / 49.9
ours, SD-2	89.4 / 75.5	<u>99.1</u> / <u>95.9</u>	96.6 / **90.0**	68.2 / 55.4	<u>90.2</u> / 75.3	96.4 / <u>90.9</u>	76.0 / 64.0	<u>88.3</u> / <u>79.4</u>
ours, generalized	85.3 / 77.1	77.2 / 68.8	95.0 / 85.9	70.2 / 60.2	89.9 / <u>81.2</u>	97.1 / 87.4	82.4 / 73.6	85.3 / 76.3
ours, generic	<u>97.9</u> / <u>87.9</u>	98.2 / 88.2	<u>97.9</u> / <u>88.2</u>	**96.7 / 87.6**	**96.4 / 88.1**	<u>98.0</u> / 88.0	86.0 / <u>78.2</u>	**96.2 / 86.6**

Table 2. The AUC / ACC (%) of the compared methods, our method trained on SD-2, the merged detector trained on all the classes of synthetic images except the one being tested (generalized), and the merged detector trained on all the classes (generic) for detecting JPEG-compressed synthetic images of different classes. The images were compressed at random qualities between 65 and 100. Fixed threshold at 0.5 was used to calculate the accuracy (ACC) scores. The last column shows the average score over the seven classes for each method. The best and the second best results of each column are highlighted in bold and by underlining, respectively.

AUC (%)	w/o proc.	Q=90	Q=80	Q=70
UFD [55]	76.7	76.4	72.5	69.9
Wang [72]	52.1	54.8	55.8	56.6
Corvi [13]	82.5	81.2	84.6	86.2
Grag [33]	68.8	64.1	64.4	65.6
PatchFor [10]	29.7	50.1	49.0	48.7
Synthbuster [4]	**98.5**	<u>92.6</u>	<u>91.7</u>	<u>91.3</u>
ours, SD-2	90.9	90.5	88.3	87.0
ours, generalized	89.5	88.6	85.6	83.6
ours, generic	<u>98.3</u>	**97.9**	**96.6**	**95.5**

Table 3. The AUC (%) over all the tested classes of synthetic images for images without post-processing and JPEG-compressed images at quality factors Q for 90, 80 and 70.

4.1. Detection on JPEG-compressed images

The first evaluation was done on JPEG-compressed images at various quality factors between 65 and 100. We compared the generalization ability of our detector trained on SD-2. the generalization ability of our merged detector, and the generic detection ability of the merged detector with the other methods. Here Synthbuster detector [4] was excluded as it was originally designed for fixed JPEG compression quality. The area under the ROC curve (AUC) and the accuracy were adopted as performance metrics. The results are shown in Tab. 2.

We observe that, in average, our method outperforms the compared methods. Nevertheless, Corvi et al. performs better on the family of Stable Diffusion (SD) models and Midjourney, while our method generalizes much better to DALL·E 2 and DALL·E 3 than Corvi et al. Note that the detection method of Corvi et al. was trained on a large number of pristine images from COCO [45], ImageNet [16] and UCID [67] and on 200K synthetic images from the Latent Diffusion model [63], while our method was only trained on 9K pristine images and 1K synthetic images per diffusion model. The Latent Diffusion model and the family of SD models have very similar network architectures, their resulting images thus feature similar artifacts. This shows that using a larger dataset could help increase the detection ability for images with very similar artifacts, but might decrease the generalization ability for images with less similar artifacts.

We further evaluated the method using JPEG-compressed images at fixed quality factors 90, 80 and 70. The average AUC over all the classes of synthetic images was computed for each detection method and each JPEG quality, shown in Tab. 3. Note that both Synthbuster detector and our generic detector have seen all the tested classes of images during training. Besides, Synthbuster detector was trained separately for images without post-processing or compressed images at fixed qualities, while our method was trained on a mix of both unprocessed images and images compressed at various qualities. As it can be seen, the generic version of our method is slightly worse than Synthbuster detector for unprocessed images but outperforms all the other methods for compressed images. As for the other compared methods, which have never seen the tested classes of synthetic images, it is fairer to compare them with our generalized detector. It is shown that both of our detectors are superior to the compared methods at all the compression settings. The better generalization ability of our detector to unseen types of images can be attributed to the higher sensibility it has to the peak frequency artifacts, and also to the variety of types of synthetic images used for training our generalized detector.

4.2. Robustness to WebP compression

Even though our detectors were trained on JPEG-compressed images, they are also robust to WebP compression. We evaluated our method on WebP-compressed images at fixed quality factors 90, 80 and 70 and at random quality factors between 65 and 100, with results shown in Tab. 4. The difference in the score obtained by each variant of our method for WebP-compressed images with respect to the JPEG-compressed ones at the same quality is

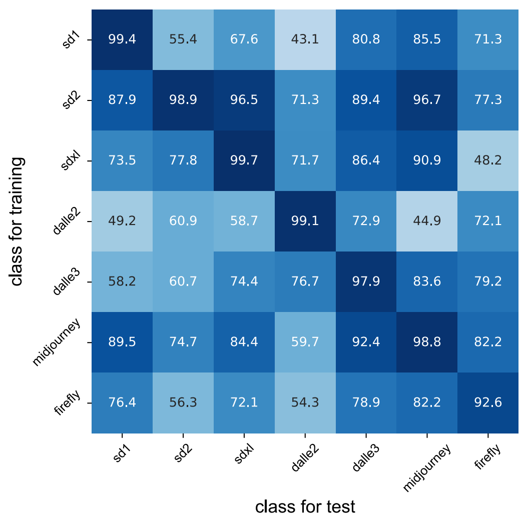


Figure 4. Generalization ability measured in AUC (%) across different classes of synthetic images. The tested images were compressed by JPEG at quality factors between 65 and 100. Each box stands for performance of the detector trained on one class (in rows) of synthetic images and tested on another class (in columns).

shown in brackets. It can be seen that our method trained on JPEG-compressed images is also applicable to detecting WebP-compressed images, with only a slight drop of performance.

AUC (%)	Q=90	Q=80	Q=70	mixed Q
our, SD-2	88.4 (-2.1)	88.2 (-0.1)	87.4 (+0.4)	88.0 (-0.3)
our, generalized	84.0 (-4.6)	83.9 (-1.7)	82.3 (-1.3)	83.0 (-2.3)
our, generic	96.5 (-1.4)	95.9 (-0.7)	94.4 (-1.1)	95.9 (-0.3)

Table 4. The AUC (%) over all the tested classes of synthetic images for WebP-compressed images at fixed quality factors Q for 90, 80 and 70 and at mixed quality factors between 65 and 100. Each value in brackets shows the performance difference between WebP and JPEG for the same detector variant and the same compression. The detection performance is only slightly dropped from JPEG to WebP.

4.3. Cross validation

We further evaluated the detection ability of our method across different classes of images, by studying the performance of the detector trained on a class of synthetic images and tested on another class, with the cross detection results shown in Fig. 4. The images for training and testing were compressed by JPEG at quality factors between 65 and 100. It is observed that the detection performs well in general for the in-class detection task where the training and test classes are the same. An exception is observed for Firefly images, which might be due to overfitting on the limited Firefly images in the training set. We can also see the generalization abilities when training and testing on different classes of images. In particular, our method trained on SD-2 has the best generalization performance to unseen classes.

4.4. Qualitative analysis

Furthermore, a qualitative analysis was conducted by studying several successfully classified examples of pristine and synthetic images. Fig. 5 shows each original image in the first row, the spectrum of its residuals after DnCNN preprocessing in the second row, and the similarity map in the third row. The similarity map is computed by $\frac{\hat{F}_{M,i}\odot\hat{R}}{||\hat{F}_{M,i}||_2\cdot||\hat{R}||_2}$ where $\hat{F}_{M,i}$ is the enhanced spectrum and $\hat{R}$ is the reference spectrum in Eq. 5. The reference and mask correspond to the detector trained on SD-2. For the tested image of SD-2, it can be seen that values on the peak frequencies contribute a lot to the overall similarity score, and the contributing values are located at different peak positions. As for the images of other classes, only a part of peak frequencies contribute to the overall similarity score, and the contributing frequency components can vary from class to class.

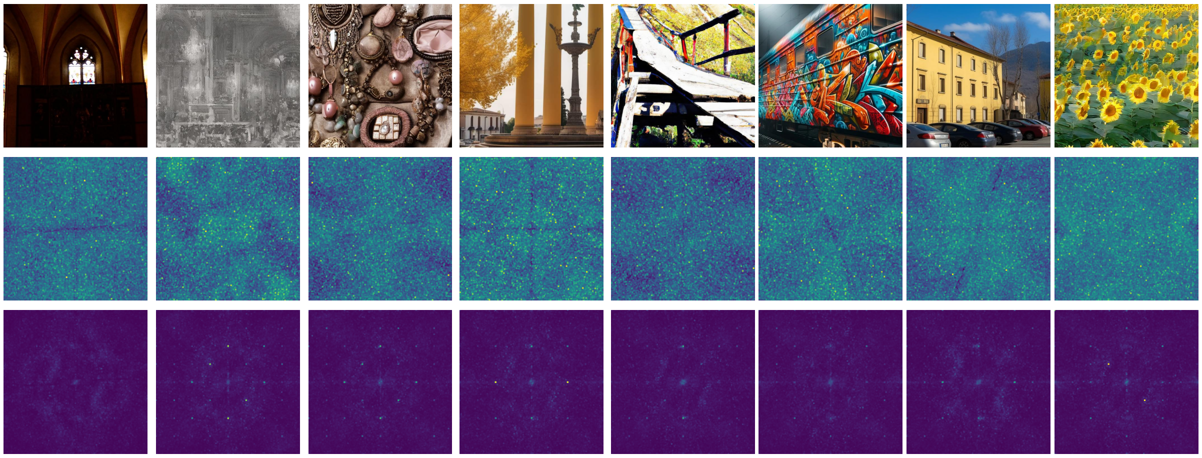
4.5. Implementation details

The pretrained DnCNN denoiser was used and finetuned during the training. All the modules of our model were jointly trained, with the learning rate at 1×10^{-4} for DnCNN and 1×10^{-3} for the rest of the modules. The ADAM optimizer [44] was used with exponential decay rate at 0.99. The batch size was 8, the image size was 512×512, and the number of epochs was 50. The training time of one detector using 2 NVIDIA A100 GPUs and 8 CPUs at 3.1 GHz was 3 hours. During training the model resulting in the smallest validation loss was selected as the final model.

The proposed detector contains 1.8M parameters in total, including 0.4M parameters for DnCNN denoiser, and 0.7M parameters for the mask and the spectrum reference, respectively. The inference time is 0.03 second per image on single NVIDIA A100 GPU.

5. Discussion

The proposed method is complementary to other methods. Even though Synthbuster [4] detector shows slightly better performance than our method on unprocessed synthetic images, it relies on the manual selection of informative frequency components, while our method automatically learns them. Our method is more portable to future generative models. Compared to the CNN-based detection methods [10, 13, 33, 72], our method allows us to make a straightforward analysis on what it has learned for detection, and gives a better understanding of the specific traces of each type of synthetic images. Compared to the methods by Ojha et al. [55] and Cozzolino et al. [14] that transform an image to a low-dimensional latent space, our method works in a high-dimensional Fourier space and is thus more sensitive to the subtle traces left by the imperfect decoding during image synthesis.



pristine | Stable Diff. 2 | Stable Diff. 1 | Stable Diff. XL | DALL·E 2 | DALL·E 3 | Midjourney | Firefly

Figure 5. Example results of successfully classified images of different classes by the detector trained on SD-2. Top: cropped input image; middle: spectrum of its residual; bottom: the similarity map between the spectrum enhanced by the mask of SD-2 and the reference spectrum of SD-2. All the similarity maps share the same range of colormap for better comparison. Both the spectra and similarity maps are visually enhanced by dilation. One can see that other classes of synthetic images also contain a part of synthesis artifacts revealed by the detector trained on SD-2.

Still, the proposed method is limited by several factors. First, it is unable to deal with rescaled synthetic images whose synthesis artefacts are shifted in the frequency domain depending on the rescaling ratio. The fixed mask and reference therefore fail to reveal the frequencies of the artifacts at unfixed positions in the spectrum. Second, our method only works for entirely synthetic images, while the images inpainted by generative models are beyond its detection capacity. Third, the proposed method is not yet available for practical use as it does not give a trustworthy decision with default threshold at 0.5. A thorough analysis on the post-validation of the outputs is necessary for a reliable detection with controlled number of false alarms, which will be further studied in future work. Finally, the robustness to various post-processings such as recompression and image enhancements is to be analyzed.

In addition, the Fourier spectrum adopted by our method assumes that the image is periodic, which actually is not true. As a result, the contrast from one border to the other leads to undesirable horizontal and vertical artifacts in the Fourier spectrum (see Fig. 1) that mix with the synthesis artifacts, making the detection more difficult. These undesirable artifacts can potentially be cancelled out by approaches such as the periodic-plus-smooth decomposition [52].

Last but not least, some preliminary experiments showed that the method without preprocessing by DnCNN denoiser results in similar detection performance. The introduction of DnCNN aims at suppressing the textures of an image and revealing the low-level synthesis artifacts, at the cost of more difficulty of its training. This will require an in-depth study of the optimal component for preprocessing and its customized training strategy.

6. Conclusion

We proposed a method for detecting synthetic images by revealing abnormal frequencies. This involved learning a mask to amplify abnormal informative frequencies and learning a spectrum reference to compare with the amplified spectrum. Experiments showed that our method is comparable to but more general than Synthbuster [4] and outperforms all others. The proposed method is robust to both JPEG and WEBP compression. As a semi-white-box method, its learned mask and reference enable us to clearly interpret which frequency components contribute to the final decision. This characteristic not only facilitates interpretable detections but also paves the way for a more generalized approach to identifying synthetic images.

Acknowledgements

This work was supported by grants from ANR (APATE, ANR-22-CE39-0016), Horizon Europe VERA.AI (No. 101070093), Région Île-de-France and UDOPIA (ANR-20-THIA-0013), and was performed using computational resources from the "Mésocentre" computing center of Université Paris-Saclay, CentraleSupélec and École Normale Supérieure Paris-Saclay supported by CNRS and Région Île-de-France (https://mesocentre.universite-paris-saclay.fr/). Centre Borelli is also a member of Université Paris Cité, SSA and INSERM.

References

[1] Libraw library, copyright © 2008-2019 libraw llc, https://www.libraw.org. 5

[2] Quentin Bammey. A contrario mosaic analysis for image forensics. In *Advanced Concepts for Intelligent Vision Systems (ACIVS)*. Springer, August 2023. 5

[3] Quentin Bammey. Positional learning for reliable ai-generated images detection, 2023. `https://github.com/qbammey/polardiffshield`. 5

[4] Quentin Bammey. Synthbuster: Towards detection of diffusion model generated images. *IEEE Open Journal of Signal Processing*, 5:1–9, 2024. `https://doi.org/10.1109/OJSP.2023.3337714`. 2, 3, 5, 6, 7, 8

[5] Quentin Bammey, Rafael Grompone von Gioi, and Jean-Michel Morel. An adaptive neural network for unsupervised mosaic consistency analysis in image forensics. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14182–14192, 2020. 5

[6] Quentin Bammey, Rafael Grompone von Gioi, and Jean-Michel Morel. Forgery detection by internal positional learning of demosaicing traces. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 328–338, January 2022. 5

[7] Jordan J Bird and Ahmad Lotfi. Cifake: Image classification and explainable identification of ai-generated synthetic images. *arXiv preprint arXiv:2303.14126*, 2023. 3

[8] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis, 2019. 2

[9] Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. Learning photographic global tonal adjustment with a database of input / output image pairs. In *The Twenty-Fourth IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 5

[10] Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. What makes fake images detectable? understanding properties that generalize. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*, pages 103–120. Springer, 2020. 2, 5, 6, 7

[11] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J. Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16123–16133, June 2022. 2

[12] Riccardo Corvi, Davide Cozzolino, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. Intriguing properties of synthetic images: from generative adversarial networks to diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 973–982, 2023. 2, 4

[13] 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 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023. 2, 3, 4, 5, 6, 7

[14] Davide Cozzolino, Giovanni Poggi, Riccardo Corvi, Matthias Nießner, and Luisa Verdoliva. Raising the bar of ai-generated image detection with clip. *arXiv preprint arXiv:2312.00195*, 2023. 3, 7

[15] Duc-Tien Dang-Nguyen, Cecilia Pasquini, Valentina Conotter, and Giulia Boato. Raise: A raw images dataset for digital image forensics. In *Proceedings of the 6th ACM Multimedia Systems Conference*, pages 219–224, 2015. 5, 1

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6, 1

[17] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 2

[18] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34:19822–19835, 2021. 2

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[20] Ruoyi Du, Dongliang Chang, Timothy Hospedales, Yi-Zhe Song, and Zhanyu Ma. Demofusion: Democratising high-resolution image generation with no $$$. *arXiv preprint arXiv:2311.16973*, 2023. 2

[21] Ricard Durall, Margret Keuper, and Janis Keuper. Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7890–7899, 2020. 2

[22] J. Duran and A. Buades. Self-similarity and spectral correlation adaptive algorithm for color demosaicking. *IEEE TIP*, 23(9):4031–4040, Sept 2014. 5

[23] J. Duran and A. Buades. A Demosaicking Algorithm with Adaptive Inter-Channel Correlation. *IPOL*, 5:311–327, 2015. 5

[24] Tarik Dzanic, Karan Shah, and Freddie Witherden. Fourier spectrum discrepancies in deep network generated images. *Advances in neural information processing systems*, 33:3022–3032, 2020. 2

[25] Thibaud Ehret and Gabriele Facciolo. A Study of Two CNN Demosaicking Algorithms. *Image Processing On Line*, 9:220–230, 2019. `https://doi.org/10.5201/ipol.2019.274`. 5

[26] David C Epstein, Ishan Jain, Oliver Wang, and Richard Zhang. Online detection of ai-generated images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 382–392, 2023. 3

[27] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *European Conference on Computer Vision*, pages 89–106. Springer, 2022. 2

[28] Pascal Getreuer. Gunturk-altunbasak-mersereau alternating projections image demosaicking. *Image Processing on Line*, 1:90–97, 2011. 5

[29] Pascal Getreuer. Zhang-wu directional lmmse image demosaicking. *Image Processing On Line*, 1:117–126, 2011. 5

[30] Pascal Getreuer. Image Demosaicking with Contour Stencils. *Image Processing On Line*, 2:22–34, 2012. https://doi.org/10.5201/ipol.2012.g-dwcs. 5

[31] Thomas Gloe and Rainer Böhme. The'dresden image database'for benchmarking digital image forensics. In *Proceedings of the 2010 ACM symposium on applied computing*, pages 1584–1590, 2010. 5

[32] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. 2

[33] D. Gragnaniello, D. Cozzolino, F. Marra, G. Poggi, and L. Verdoliva. Are gan generated images easy to detect? a critical analysis of the state-of-the-art. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2021. 2, 3, 5, 6, 7

[34] John F Hamilton Jr and James E Adams Jr. Adaptive color plan interpolation in single sensor color electronic camera, May 13 1997. US Patent 5,629,734. 5

[35] Samuel W. Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T. Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 35(6), 2016. 5

[36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[37] Yang He, Ning Yu, Margret Keuper, and Mario Fritz. Beyond the spectrum: Detecting deepfakes via re-synthesis. In *Thirtieth International Joint Conference on Artificial Intelligence*, pages 2534–2541. IJCAI, 2021. 3

[38] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 2

[39] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 852–863. Curran Associates, Inc., 2021. 2

[40] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[41] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[42] Daisuke Kiku, Yusuke Monno, Masayuki Tanaka, and Masatoshi Okutomi. Residual interpolation for color image demosaicking. In *2013 IEEE International Conference on Image Processing*, pages 2304–2308. IEEE, 2013. 5

[43] Daisuke Kiku, Yusuke Monno, Masayuki Tanaka, and Masatoshi Okutomi. Minimized-laplacian residual interpolation for color image demosaicking. In *Digital Photography X*, volume 9023, page 90230L. International Society for Optics and Photonics, 2014. 5

[44] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[45] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 5, 6

[46] Bo Liu, Fan Yang, Xiuli Bi, Bin Xiao, Weisheng Li, and Xinbo Gao. Detecting generated images by real images. In *European Conference on Computer Vision*, pages 95–110. Springer, 2022. 2, 3

[47] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022. 3

[48] Peter Lorenz, Ricard L Durall, and Janis Keuper. Detecting images generated by deep diffusion models using their local intrinsic dimensionality. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 448–459, 2023. 3

[49] J. Lukas, J. Fridrich, and M. Goljan. Digital camera identification from sensor pattern noise. *IEEE Trans. on Information Forensics and Security*, 1(2):205–214, 2006. 2

[50] Sara Mandelli, Nicolò Bonettini, Paolo Bestagini, and Stefano Tubaro. Detecting gan-generated images by orthogonal training of multiple cnns. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 3091–3095, 2022. 3

[51] Scott McCloskey and Michael Albright. Detecting gan-generated imagery using saturation cues. In *2019 IEEE international conference on image processing (ICIP)*, pages 4584–4588. IEEE, 2019. 3

[52] Lionel Moisan. Periodic plus smooth image decomposition. *Journal of Mathematical Imaging and Vision*, 39:161–179, 2011. 8

[53] Yusuke Monno, Daisuke Kiku, Masayuki Tanaka, and Masatoshi Okutomi. Adaptive residual interpolation for color image demosaicking. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 3861–3865. IEEE, 2015. 5

[54] 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, 2022. 2

[55] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24480–24489, June 2023. 3, 5, 6, 7

[56] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Gaugan: semantic image synthesis with spatially adaptive normalization. In *ACM SIGGRAPH 2019 Real-Time Live!*, pages 1–1. 2019. 2

[57] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 4

[58] Ibrahim Pekkucuksen and Yucel Altunbasak. Gradient based threshold free color filter array interpolation. In *2010 IEEE International Conference on Image Processing*, pages 137–140. IEEE, 2010. 5

[59] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. 2

[60] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2

[61] Md Awsafur Rahman, Bishmoy Paul, Najibul Haque Sarker, Zaber Ibn Abdul Hakim, and Shaikh Anowarul Fattah. Artifact: A large-scale dataset with artificial and factual images for generalizable and robust synthetic image detection. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 2200–2204, 2023. 3

[62] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. 2

[63] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 6

[64] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 2

[65] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. Stable diffusion, 2022. 2

[66] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 2

[67] Gerald Schaefer and Michal Stich. Ucid: An uncompressed color image database. In *Storage and retrieval methods and applications for multimedia 2004*, volume 5307, pages 472–480. SPIE, 2003. 6

[68] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 3

[69] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. 2

[70] Runjie Tan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Color image demosaicking via deep residual learning. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 793–798. IEEE, 2017. 5

[71] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 2

[72] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A. Efros. Cnn-generated images are surprisingly easy to spot… for now. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8692–8701, 2020. 2, 3, 5, 6, 7

[73] Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, Hezhen Hu, Hong Chen, and Houqiang Li. Dire for diffusion-generated image detection, 2023. 3

[74] Zhiyuan Yan, Yong Zhang, Yanbo Fan, and Baoyuan Wu. Ucf: Uncovering common features for generalizable deepfake detection. *ArXiv*, abs/2304.13949, 2023. 3

[75] Junbin Zhang, Yixiao Wang, Hamid Reza Tohidypour, and Panos Nasiopoulos. Detecting stable diffusion generated images using frequency artifacts: A case study on disney-style art. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 1845–1849, 2023. 2, 3

[76] 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, 2017. 2, 4

[77] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. Detecting and simulating artifacts in gan fake images. In *2019 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2019. 2