Beyond Generation: A Diffusion-based Low-level Feature Extractor for Detecting AI-generated Images

Nan Zhong[1], Haoyu Chen[1], Yiran Xu[2], Zhenxing Qian[2]*, and Xinpeng Zhang[2]
[1] City University of Hong Kong, [2] Fudan University
{nzhong20,yrxu23,zxqian,zhangxinpeng}@fudan.edu.cn,{haoychen3-c}@my.cityu.edu.hk

Abstract

The prevalence of AI-generated images has evoked concerns regarding the potential misuse of image generation technologies. In response, numerous detection methods aim to identify AI-generated images by analyzing generative artifacts. Unfortunately, most detectors quickly become obsolete with the development of generative models. In this paper, we first design a low-level feature extractor that transforms spatial images into feature space, where different source images exhibit distinct distributions. The pretext task for the feature extractor is to distinguish between images that differ only at the pixel level. This image set comprises the original image as well as versions that have been subjected to varying levels of noise and subsequently denoised using a pre-trained diffusion model. We employ the diffusion model as a denoising tool rather than an image generation tool. Then, we frame the AI-generated image detection task as a one-class classification. We estimate the low-level intrinsic feature distribution of real photographic images and identify features that deviate from this distribution as indicators of AI-generated images. We evaluate our method against over 20 different generative models, including those in GenImage and DRCT-2M datasets. Extensive experiments demonstrate its effectiveness on AI-generated images produced not only by diffusion models but also by GANs, flow-based models, and their variants.

1. Introduction

Image generation has rapidly evolved over the past years. Numerous generative models such as autoregressive models [20], variational autoencoders [19], normalizing flows [18], generative adversarial networks (GANs) [12, 16], and diffusion models [8, 13, 14] have been proposed. Especially, diffusion-based models bring image generation into a new era. Users without expertise can easily access high-quality AI-generated images and content editing by commercial APIs like Midjourney [1], which utilizes diffusion-based techniques. Although the advancement of image generation offers significant benefits for productivity, its widespread application raises concerns about malicious usages like misinformation dissemination. Therefore, it is imperative to develop methods for detecting AI-generated images to ensure a trustworthy cyberspace environment.

Recently, a flurry of detection methods [2, 27, 28, 51, 52, 56] have developed to identify AI-generated images. Existing methods frame the AI-generated image detection as a binary classification. The training set usually consists of real photographic images and one generative model (e.g., ProGAN [16]). Then, they evaluate their performance on various unseen generative models. Since the types of generative models and their variants are too vast to be encompassed in a finite training set, generalization is the primary requirement for detectors. Most studies aim to extract a universal artifact, which is absent from real photographic images, across different generative models to achieve generalization. For instance, LGrad [46] employs the gradient maps of images produced by a pre-trained classification model as the universal fingerprint of generative models. Then, they conduct binary classification for gradient maps instead of spatial images. However, the generative artifacts significantly vary with the development of generation models. In the diffusion era, most detectors [46, 51] fail to identify those high-quality AI-generated images. As a result, many studies [2, 28, 52] delve into the inner mechanism of diffusion models and extract generative artifacts specialized for diffusion-generated images.

In this paper, we frame the AI-generated image detection as a one-class classification [3, 4] (also known as anomaly detection) instead of a binary classification paradigm. Since generative models will evolve with the development of image generation, it is not suitable to identify AI-generated images from generative artifacts. Specifically, we focus on extracting the low-level intrinsic features of real photographic images through a novel elaborated low-level image feature extractor. Images whose features deviate from the distribution of real images can be regarded as AI-generated.

*Corresponding author.

The cornerstone of our approach is a pre-trained low-level image feature extractor, which transforms spatial images into a low-dimensional feature space. In this space, images from different sources, such as various generative models and real photographic images, exhibit distinct distributions. In the computer vision community, many studies focus on visual content understanding like the well-known CLIP [37], which is trained on large-scale image-text pairs. These feature extractors aim to represent high-level image features like semantic information. However, high-level image features of AI-generated images are very close to those of photographic images with the development of generative models. Low-level features like noise patterns [26], which are imperceptible to humans, play a crucial role in image forensics. Zheng *et al*. [55] and Huh *et al*. [15] utilize camera metadata (EXIF tags) to supervise the feature extractor in learning low-level image features. However, most photographic images lack camera metadata which may be erased in the post-processing. Instead of EXIF tag supervision, we design a novel pretext for the low-level image feature extractor. Given a photographic image, we first add noise with different intensities and obtain multiple corrupted images. Then, we adopt a denoising module to restore corrupted images. In this phase, the pre-trained diffusion model is used as a denoising tool rather than an image generation tool. The primary task of the feature extractor is to distinguish original photographic images and denoised images, which differ only at the pixel level. The main contributions are as follows

- We design a novel pretext task and develop a low-level image feature extractor, which can discern the differences between various source images.
- We frame the AI-generated image detection as a one-class classification from the perspective of low-level features of real photographic images.
- Extensive experiments demonstrate that our method works effectively against a wide array of generative models, including GANs, diffusion models, flow-based models, and their variants.

2. Related Works

Generative models. Image generation has witnessed rapid progress over the past few years. Introduced by Ho *et al*. [14], Denoising Diffusion Probabilistic Models have emerged at the forefront of image generation. Diffusion models [44] employ Stochastic Differential Equation (SDE) to describe the process of transforming a real photographic image into a noisy image by gradually adding small amounts of Gaussian noise in a stepwise manner. During the generation phase, these models utilize the reverse SDE to synthesize images from random noise. ADM [8] was the first to achieve superior synthetic image quality on the ImageNet [7] dataset compared to GANs, marking a significant milestone in the evolution of image generation techniques based on diffusion models. In the follow-up studies, many studies focus on sampling efficiency [45], network architecture [21], and controllability [54]. Latent diffusion models [40] significantly improve the capacity of diffusion models to produce high-resolution synthetic images. They also facilitate the incorporation of text conditioning inputs through a cross-attention mechanism. The widely used open-source model stable diffusion [40] has served as a catalyst for the viral spread of AI-generated images across social media platforms. Users without expertise can obtain high-quality AI-generated images with arbitrary prompts through commercial APIs like Midjourney [1].

AI-generated image detection. In the early studies, many researchers focus on designing hand-crafted features including statistical irregularities [35], visual artifacts [50], and physical and geometric inconsistencies [32] in images. Subsequently, the focus shifts to end-to-end learning using convolutional neural networks (CNNs), moving away from heuristic feature design. Wang *et al*. [51] employs some simple data augmentation including JPEG compression and Gaussian blur to improve the generalization of the classifier. Frank *et al*. [11] find that generative models leave significant abnormality in the frequency domain and then identify AI-generated images from the frequency perspective. The majority of existing methods aim to find a universal generative artifact across various generative models. For example, LGrad [46] adopts the gradient map of the image with respect to a pre-trained classification model as the fingerprint of AI-generated images. LNP [23] extracts the high-frequency noise as the universal fingerprint. With the advent of diffusion-based generators, many detectors struggle to identify images generated with high visual quality. As a result, some detectors specifically designed for diffusion methods have been proposed. They utilize the inherent characteristics of the diffusion models. DIRE [52] extracts the reconstruction error of the image over a pre-trained diffusion model as the artifact of diffusion-based images. Then, they conduct binary classification for the reconstruction error maps instead of the original images. DRCT [5] enhances the generalizability of detectors for diffusion-generated images by focusing on hard sample classification, treating real photographic images reconstructed by a pre-trained diffusion model as hard samples.

Unlike previous methods relying on artifacts from generative models to identify AI-generated images, we extract intrinsic features of real photographic images using our novel low-level feature extractor. We model the distribution of photographic image features and frame the detection of AI-generated images as an anomaly detection task. Images whose low-level features do not align with the distribution of photographic image features are flagged as AI-generated.

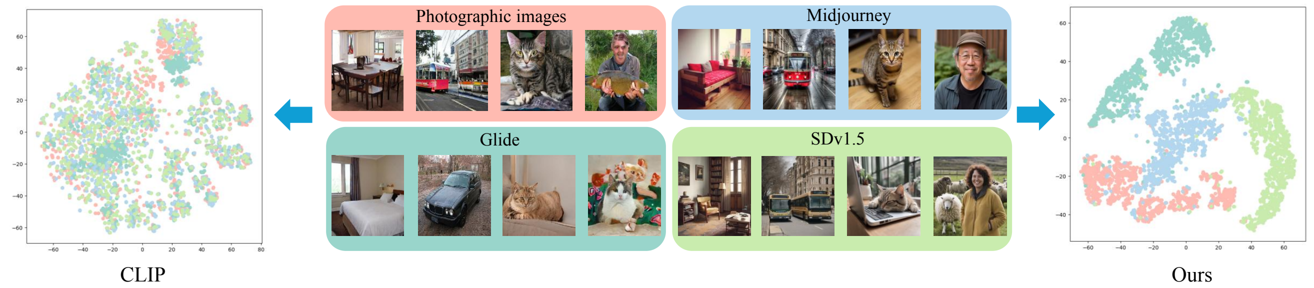

Figure 1. The feature distribution visualization through t-SNE [49]. We employ four sources of images including real photographic images, Midjourney [1], Glide [31], and SDv1.5 [40]. Our low-level feature extractor transforms spatial images into a well-separated space that reflects their respective sources.

3. Method

3.1. Motivation

Large pre-trained models for image visual understanding, such as the image encoder of CLIP [37], have become off-the-shelf tools within the computer vision community. Many downstream tasks [42] closely related to the semantic information of images significantly benefit from these pre-trained backbones. Ojha *et al*. [33] first introduced the image encoder of CLIP as the backbone of the detector to identify AI-generated images. However, CLIP primarily focuses on the high-level features of images like visual content. As the quality of AI-generated images improves, it becomes increasingly difficult to distinguish them from real photographic images based on visual content. Therefore, we focus on analyzing low-level image features that are often imperceptible to the human eye. Due to the inherent characteristics of the upsampling convolution layer, which can lead to spectral aliasing effects [10], generative models often struggle to mimic the low-level features of photographic images.

In this paper, we aim to design a foundational backbone for understanding low-level image features. The low-level feature extractor should be capable of distinguishing between two images that exhibit only pixel-level differences, which may appear nearly identical to the human eye. To achieve this goal, we design a novel pretext task which leads the feature extractor to focus on the pixel-level difference. Given an original photographic image, we generate multiple corrupted variants by applying noise at varying intensities. Then, we employ a pre-trained denoising tool to restore these corrupted images. Intuitively, the visual content of denoised images is very similar to that of the original images, but pixel-level differences may still exist. The pretext task of the low-level image feature extractor is to calculate the concrete pixel-level difference and noise intensity. Fig. 1 shows the feature visualization comparison between our low-level image feature extractor and well-known CLIP over common generative models and real photographic images. We find that the features derived from the same generative models are concentrated and distinct from those of other models and photographic images. This observation sheds light on that the pre-trained low-level feature extractor can be used as a backbone to identify AI-generated images.

3.2. Pretext Task Design

Fig. 2 illustrates the basic concept of our pretext task. The crux of our approach is to construct a set of images including real photographic images and several denoised images. The denoised images closely resemble their corresponding real images, differing only at the pixel level. We denote the real photographic images as x_0 and noisy image x_t. As Fig. 2 shows, we first add varying levels of noise to the x_0. The process of adding noise can be expressed as

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \tag{1}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and $\bar{\alpha}_t$ controls the noise intensity. This process is the same as the diffusion models, in which $\bar{\alpha}_t$ is between 0 and 1. The value of $\bar{\alpha}_t$ is close to 0 with the increasing of t. We select different t to obtain various noise images x_t. Then, we aim to restore these corrupted images x_t to clean counterpart x_0. We utilize a pre-trained diffusion model to denoise the corrupted images. In other words, we treat the pre-trained diffusion model as a plug-and-play denoising module to restore noise images x_t. Specifically, we adopt a pre-trained denoising U-Net network $\boldsymbol{\epsilon_\theta}(x_t, t)$ to predict the noise $\boldsymbol{\epsilon}$ at the time step t. The denoising procedure of diffusion is iterative. The denoised image at the time step $t-1$ can be expressed as

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\left(\frac{x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x_t,t)}{\sqrt{\bar{\alpha}_t}}\right) + \sqrt{1-\bar{\alpha}_{t-1}}\epsilon_\theta(x_t,t). \tag{2}$$

This noise reverse process is proposed by DDIM [43]. The final denoised image can be obtained through iterative noise estimation and updating. We initially add varying levels of noise to the original image by controlling $\bar{\alpha}_t$. A large t indicates a greater initial noise level that corrupts the original

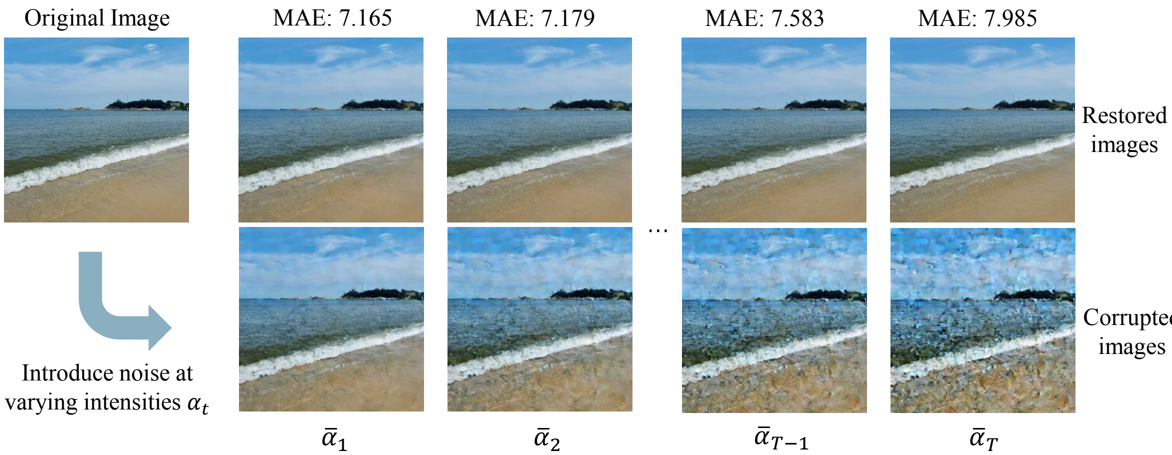

Figure 2. The illustration of our pretext task. We initially introduce noise to corrupt the original image. Subsequently, we denoise these corrupted images to create a set of images which only differ at the pixel level.

images. We denote the final denoised image as x_t^{rec}. Intuitively, the difference between x_0 and x_t^{rec} increases as the noise intensity rises with large t. The core pretext task of the low-level feature extractor is to distinguish the difference between original images and their corresponding denoised images x_t^{rec} with different noise intensities. In practical implementation, the denoising module should be effective enough to restore a corrupted image as closely as possible to the original, ensuring minimal pixel-level differences. We explore many denoising tools in the field of computer vision and find that a pre-trained stable diffusion [40] is a suitable denoising module. The schedule of $\bar{\alpha}_t$ is also based on stable diffusion. Since stable diffusion is trained in the latent space, we conduct Gaussian noise corruption and diffusion-based restoration in the latent space through a pre-trained VAE [13]. In the stable diffusion, 1000 levels of noise intensities are considered, corresponding to t from 1 to 1000. Our method chooses T different noise intensities. We denote τ as the subset of our selected t.

The training objective for the feature extractor consists of two components. First, we calculate the distance between the original images and the various denoised images using mean absolute error. In addition, we consider the autoencoder reconstruction error loss, where the original image is encoded into a latent feature and then decoded back into a spatial image without adding noise. VAE reconstruction leaves artifacts on the final synthetic images resulting in the pixel-level subtle difference from the original image. We adopt regression learning for the feature extractor, and the formulation can be expressed as

$$\ell_{regression} = \| g(f(\boldsymbol{x};\boldsymbol{\theta});\boldsymbol{\phi}) - tg_{mae}(\boldsymbol{x}) \| , \tag{3}$$

where $f(\cdot;\boldsymbol{\theta})$ and $g(\cdot;\boldsymbol{\phi})$ denote the feature extractor and regression head (a simple fully-connected layer), respectively. $tg_{mae}(\cdot)$ represents the mean absolute error value of image $\boldsymbol{x}$ compared to its corresponding original image.

Besides regression learning, we also design a classification task for the feature extractor. We assign different noise intensities, associated with varying t, as labels for the corresponding images. The number of categories is equal to the length of the subset τ plus two, which corresponds to the original image and VAE-reconstructed image. For the classification task, we adopt the cross-entropy loss with the softmax activation function. The formulation can be expressed as

$$\ell_{cle} = -\sum_{j=1}^{C} y_j \log \left(\frac{\exp(g(f(\boldsymbol{x};\boldsymbol{\theta}),\boldsymbol{\varphi})_j)}{\sum_{k=1}^{C} \exp(g(f(\boldsymbol{x};\boldsymbol{\theta}),\boldsymbol{\varphi})_k)} \right), \tag{4}$$

where C denotes the category number. Here, $g(\cdot;\boldsymbol{\varphi})$ denotes a classification head. The final training objection function for the feature extractor is formulated as

$$\ell = \ell_{cle} + \ell_{regression}, \tag{5}$$

where we simultaneously consider the regression learning and classification task. Both regression and classification heads can be removed in downstream tasks.

3.3. AI-generated Image Detection

The pre-trained low-level feature extractor is capable of discerning the subtle pixel-level difference between two images, which is crucial to some image forensic-related tasks. In this subsection, we expand the low-level image feature extractor to the detection of AI-generated images. As Fig. 1 shows, the feature representations of images from different sources including real photographic images and images generated by different models exhibit a clustering phenomenon. The feature distributions of images generated by the same model are more concentrated, while the distributions of images from different generative models are widely separated. Therefore, we adopt the Gaussian Mixture Model [39] to model the distribution of feature representations of real photographic images. Gaussian Mixture Model is a straightforward probabilistic model that is widely utilized in various fields, including clustering and density estimation. The formulation of a GMM can be expressed as follows

$$p_{\text{gmm}}(\boldsymbol{f} \mid \boldsymbol{\lambda}) = \sum_{k=1}^{K} w_k \mathcal{N}\left(\boldsymbol{f} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right), \tag{6}$$

where $\boldsymbol{f}$ denotes the low-level feature representation, w_k denotes the weight of the k-Gaussian component, the pair of $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ denotes the mean and covariance of the k-Gaussian component, respectively. All learnable parameters including w_k, $\boldsymbol{\mu}_k$, and $\boldsymbol{\Sigma}_k$ are collectively representated by $\boldsymbol{\lambda}$. We adopt the Expectation-Maximization algorithm [30] to optimize the learnable parameters $\boldsymbol{\lambda}$.

In the detection phase, we first extract the low-level feature $\boldsymbol{f}$ of the suspicious image and calculate its log-likelihood score $log(p_{\text{gmm}}(\boldsymbol{f} \mid \mathcal{M}))$ (referred to as likelihood) over the pre-trained Gaussian mixture Model $\mathcal{M}$, which is trained on feature representations of real photographic images. If the suspicious image belongs to the real photographic images, its likelihood value with respect to $\mathcal{M}$ will be high. Conversely, if it is an AI-generated image, its likelihood value will be significantly low. Therefore, we can set a suitable threshold to separate real photographic images and AI-generated images. This threshold is determined based on the training set of real photographic images. The threshold requires to be lower than most likelihood scores of real photographic images to maintain a low false alarm rate.

4. Experimental Results

4.1. Experimental Setups

Datasets. For training the low-level image feature extractor, we randomly select 12,000 real photographic images from the YFCC100M [48] dataset. Regarding AI-generated image detection evaluation, we adopt two common datasets, namely GenImage [57] and DRCT-2M [5]. GenImage contains eight generative models, including some commercial APIs like Midjourney. The real image component of the GenImage is derived from ImageNet [7]. DRCT-2M collects sixteen types of diffusion-based images, which can be grouped into two categories: ten types of stable diffusion models and their multiple variants, and the remaining images generated using diffusion-based image-to-image models [53]. The real image component of the DRCT-2M is derived from the MSCOCO dataset [22].

Selection of noise intensity $\bar{\alpha}_t$. During the training phase of stable diffusion v1.4 [40], 1000 levels of noise intensities $\bar{\alpha}_t$ are utilized. In the generation phase, many studies use the DDIM scheduler [43] to speed up the sampling process to just a few steps. In our approach, we set the number of sampling steps to 50. For each photographic image, we add 9 different levels of noise intensities, corresponding to $\bar{\alpha}_1$ through $\bar{\alpha}_9$. For example, in the case of $\bar{\alpha}_5$, the DDIM scheduler requires 5 steps to restore the image. Besides denoised images, we also include images reconstructed by the VAE module without adding noise as a separate category. Overall, there are 11 categories for the feature extractor, including original photographic images as one category.

Implementation details. The network architecture of our low-level feature extractor is based on a ViT-like model [9]. Since the primary contribution of our paper lies in the design of the pretext task, detailed specifications of the network architecture are provided in the supplementary materials. During the training phase of the feature extractor, we utilize the Adam optimizer with a fixed learning rate of 10^{-5}. For distribution modeling, we employ the scikit-learn toolbox to implement a Gaussian Mixture Model. The number of Gaussian components is set to 6. We use 10,000 real photographic images to estimate the distribution of low-level features. Subsequently, we sort the likelihood scores of the images in descending order and establish a threshold to determine whether a suspicious image belongs to the category of photographic images. This threshold is set at the 99.95th percentile of the likelihood scores for the training samples.

Baselines. We select 12 representative approaches as our baselines, encompassing a variety of top-tier conferences. They are, CNNSpot [51], F3Net[36], CLIP/RN50 [37], GramNet [24], De-fake [41], Conv-B [25], UnivFD [33], DIRE [52], LNP [23], LGrad [46], NPR [47], and DRCT [5]. Among baseline approaches, DIRE and DRCT require a pre-trained diffusion model for AI-generated image detection. We assume that they can utilize stable diffusion v1.4, which aligns with our approach. We adopt accuracy as the metric to evaluate the detection performance.

4.2. AI-generated Image Detection Evaluation

Comparisons on GenImage. We first evaluate our method and baselines on the GenImage dataset. All baselines are trained on the SDv1.4 subset of the GenImage. Our approach requires only real photographic images to model the distribution of their low-level features. In this case, we do not need to access the AI-generated images of the training set. As a further step, we use the pre-trained feature extractor as a parameter initialization strategy, which we subsequently fine-tune on the SDv1.4 subset of GenImage. Then, we model low-level feature distribution based on the joint of the original pre-trained and fined-tuned feature extractor. For each feature, we compute the likelihood values from both models and flag suspicious images as AI-generated ones if both likelihood values are simultaneously greater

Method	Midjourney	SDv1.4	SDv1.5	ADM	Glide	Wukong	VQDM	BigGAN	Avg.
CNNSpot	84.92	99.88	99.76	53.48	53.80	99.68	55.50	49.93	74.62
F3Net	77.85	98.99	99.08	51.20	54.87	97.92	58.99	49.21	73.51
CLIP/RN50	83.30	99.97	99.89	54.55	57.37	99.52	57.90	50.00	75.31
GramNet	73.68	98.85	98.79	51.52	55.38	95.38	55.15	49.41	72.27
De-fake	79.88	98.65	98.62	71.57	78.05	98.42	78.31	74.37	84.73
Conv-B	83.55	**99.99**	**99.92**	51.75	56.27	99.92	58.41	50.00	74.98
UnivFD	91.46	96.41	96.14	58.07	73.40	94.53	67.83	57.72	79.45
DIRE	50.40	**99.99**	**99.92**	52.32	67.23	**99.98**	50.10	49.99	71.24
LNP	60.30	99.72	99.64	49.86	49.88	99.52	49.85	49.88	69.80
LGrad	84.40	99.21	99.09	59.23	83.86	98.19	57.23	61.63	80.40
NPR	80.94	99.70	99.51	60.27	77.00	98.41	54.53	63.03	79.20
DRCT	91.50	95.01	94.41	79.42	89.18	94.67	90.03	81.67	89.49
Ours-basic	**95.51**	96.06	96.33	**95.88**	**96.00**	96.36	**95.95**	**95.71**	**95.97**
Ours-ft	95.03	95.32	95.53	95.29	95.17	95.58	95.10	95.15	95.27
DRCT-m	82.82	92.33	91.87	75.18	87.44	92.23	89.12	87.38	87.67
Ours-m	92.65	92.83	93.16	93.02	93.10	93.55	93.08	93.17	93.07

Table 1. The comparison of detection accuracy between our method and the baselines on the GenImage. Ours-ft denotes the pre-trained feature extractor has been fine-tuned on the training set of GenImage. Ours-basic denotes the pre-trained feature extractor that has not undergone fine-tuning. Ours-m and DRCT-m denote the cross-database scenario.

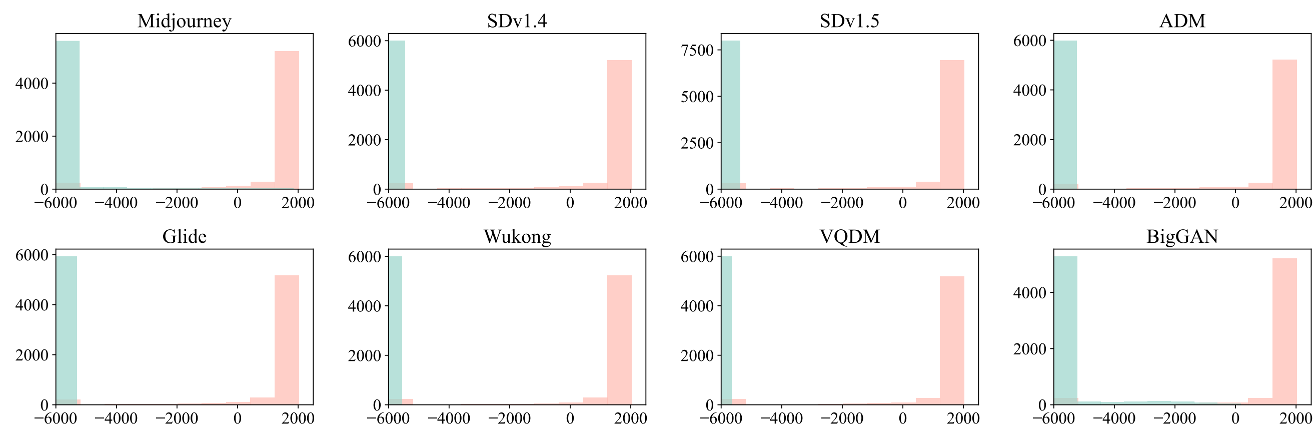

Figure 3. The illustration of likelihood score distribution between AI-generated and photographic images. To enhance visualization, likelihood scores below -6000 are limited to 2000.

than their thresholds. We name two schemes of our approach: ours-basic and ours-ft. Table 1 presents the specific experimental results. Our methods outperform all baselines with a clear margin. All methods achieve very high accuracy on images generated by models SDv1.4 and SDv1.5. The reason is that these images are close to those in the training set. However, most baselines struggle to identify images generated by BigGAN. The generative artifacts produced by BigGAN differ significantly from the images generated by diffusion models in the training set, leading to a sharp degradation in performance. In contrast, our method focuses on the low-level features of real photographic images. Our detection methodology is based on anomaly detection rather than binary classification. Even though synthetic images are generated by various generative models, their low-level feature distributions are different from the feature distribution of photographic images. This explains the superior generalization capability of our methods. Furthermore, we observe that fine-tuning the pre-trained feature extractor does not lead to an improvement. We speculate that the detection accuracy saturates on the GenImage for our feature extractor. This accounts for the ineffectiveness of fine-tuning for the final performance.

Besides numerical evaluation, we also visualize the likelihood comparison between AI-generated and photographic images. As Fig. 3 shows, the likelihood scores of AI-generated images are considerably lower than those of photographic images. Consequently, we can accurately identify AI-generated images by a fixed threshold.

Comparisons on DRCT-2M. The training set of the DRCT-2M is also the SDv1.4 subset. Similar to the evaluation on the GenImage, we evaluate our method in two

Method	SD Variants						Turbo Variants		LCM Variants		ControlNet Variants			DR Variants			Avg.
	LDM	SDv1.4	SDv1.5	SDv2	SDXL	SDXL-Refiner	SD-Turbo	SDXL-Turbo	LCM-SDv1.5	LCM-SDXL	SDv1-Ctrl	SDv2-Ctrl	SDXL-Ctrl	SDv1-DR	SDv2-DR	SDXL-DR	
CNNSpot	99.87	99.91	99.90	97.55	66.25	86.55	86.15	72.42	98.26	61.72	97.96	85.89	82.84	60.93	51.41	50.28	81.12
F3Net	99.85	99.78	99.79	88.66	55.85	87.37	68.29	63.66	97.39	54.98	97.98	72.39	81.99	65.42	50.39	50.27	77.13
CLIP/RN50	99.00	99.99	99.96	94.61	62.08	91.43	83.57	64.40	98.97	57.43	99.74	80.69	82.03	65.83	50.67	50.47	80.05
GramNet	99.40	99.01	98.84	95.30	62.63	80.68	71.19	69.32	93.05	57.02	89.97	75.55	82.68	51.23	50.01	50.08	76.62
De-fake	92.10	99.53	99.51	89.65	64.02	69.24	92.00	93.93	99.13	70.89	58.98	62.34	66.66	50.12	50.16	50.00	75.52
Conv-B	**99.97**	**100.00**	**99.97**	95.84	64.44	82.00	80.82	60.75	99.27	62.33	**99.80**	83.40	73.28	61.65	51.79	50.41	79.11
UnivFD	98.30	96.22	96.33	93.83	91.01	93.91	86.38	85.92	90.44	88.99	90.41	81.06	89.06	51.96	51.03	50.46	83.46
DIRE	98.19	99.94	99.96	68.16	53.84	71.93	58.87	54.35	**99.78**	59.73	99.65	64.20	59.13	51.99	50.04	49.97	71.23
LNP	49.50	99.45	99.45	51.26	56.37	71.24	**99.32**	**99.44**	98.96	65.30	66.07	71.02	69.81	50.71	50.65	49.82	71.77
LGrad	96.79	97.87	97.93	87.80	54.52	77.07	98.11	98.20	97.80	59.75	94.58	92.93	90.67	49.44	49.36	49.44	80.70
NPR	94.92	97.73	97.78	82.96	67.52	79.72	95.81	95.85	97.68	80.32	94.45	92.96	90.22	49.24	49.41	49.01	83.63
DRCT	96.74	96.26	96.33	94.89	96.24	93.46	93.43	92.94	91.17	95.01	95.60	92.68	91.95	**94.10**	**69.55**	**57.43**	90.49
Ours-basic	97.37	97.60	97.54	**97.65**	**97.64**	**97.67**	97.62	97.42	97.61	**97.64**	50.33	50.14	52.43	92.75	52.31	50.08	82.74
Ours-ft	97.13	97.13	97.13	97.13	97.13	97.13	97.13	97.13	97.13	97.13	97.08	**96.81**	**94.73**	92.26	51.79	49.73	**90.86**

Table 2. The comparison of detection accuracy between our method and the baselines on the DRCT-2M. Ours-ft can identify all diffusion-generated images in the top ten subsets of DRCT-2M with a 100% recall rate. The real component of DRCT-2M shares the same photographic images from MSCOCO resulting in the same detection accuracy as 97.13%.

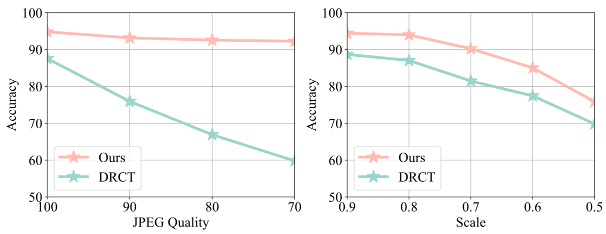

Figure 4. Comparison of robustness between ours and DRCT. We adopt the average accuracy of eight generative models in GenImage as the performance metrics.

	DRCT	Ours	97-P	75-P
DDPM [14]	83.43	**99.97**	-12869	-30540
GLOW [18]	100	**99.95**	-56310	-121336
StarGAN [6]	99.9	**99.95**	-277178	-385297
StyleGAN2 [17]	**83.53**	75.52	2187	-4047
GauGAN [34]	72.08	**99.98**	-60665	-128474
DALLE2 [38]	89.53	**99.90**	-17384	-47721

Table 3. The detection performance (recall) of other generative models. The specific threshold is -621.73. 97-P and 75-top denote the top 97% and 75% likelihood scores of the test set.

cases. Table 2 presents the specific experimental results. The results are consistent with those from the GenImage. All baselines achieve high accuracy when the test set closely aligns with the training SDv1.4. Our methods still achieve the best results compared to baselines. In this case, fine-tuning is crucial to the final detection performance. Ours-ft achieves remarkable performance across various diffusion models except for SDv2-DR and SDXL-DR. They are challenging test sets, in which images are reconstructed by diffusion models from real images rather than random noises. Additional experimental results related to recall and precision analyses are available in the supplementary materials.

Robustness Evaluation. In the practical application, images often undergo benign post-processing, such as JPEG compression, on social media platforms. Therefore, AI-generated image detection approaches should be resilient to such benign post-processing. We employ two common post-processing including JPEG compression and downsampling to evaluate the robustness of our method. In our intuition, such post-processing does not affect the visual content but may alter the low-level features of images resulting in the degradation of detectors. We select the DRCT, whose performance is the closest to ours, as a reference. The quality factor (QF) of JPEG compression varies from 100 to 70. In terms of downsampling, we reduce the resolution of images to r times its original size, where r varies from 0.9 to 0.5. Fig. 4 shows the experimental results on the GenImage dataset. Although our method degrades with the increase of benign image distortion, it consistently outperforms DRCT in all cases.

Comparisons on cross-database generalization. To further validate the generalization of our method. We conduct a cross-database evaluation, in which we train detectors on the training set of DRCT-2M and evaluate the detection performance on GenImage. We employ real images from the training set of DRCT-2M (MSCOCO) to model the distribution of low-level features of photographic images. As aforementioned description, the real image component of DRCT-2M and GenImage is derived from MSCOCO and ImageNet, respectively. Table 1 shows the experimental results. We find the performance of our method only slightly degrades in this case. This experiment shows that our methods correctly flag images from ImageNet as real images, even though the real images of the test set (ImageNet) are not aligned with those of the training set (MSCOCO). Meanwhile, the AI-generated images in the GenImage set are accurately identified.

Detection on other generative models. Since our low-level feature extractor is trained using the pre-trained latent diffusion model stable diffusion v1.4, it is crucial to

	Ours	w/o Reg	w/o Cls	BF-CNN	ADM
Midjourney	**95.51**	94.58	94.34	70.31	65.42
SDv1.4	**96.06**	95.15	95.99	60.30	89.69
SDv1.5	**96.33**	95.11	96.04	60.32	90.03
ADM	95.88	94.91	95.63	**96.35**	94.33
Glide	96.00	94.83	95.93	**96.90**	65.24
Wukong	**96.36**	95.22	96.14	69.20	92.73
VQDM	95.59	94.80	**95.78**	87.01	91.82
BigGAN	**95.71**	94.59	95.46	85.92	87.65
Avg.	**95.97**	94.90	95.66	78.36	84.61

Table 4. The ablation studies of the pretext task. The terms "w/o Reg" and "w/o Cls" indicate the removal of regression learning and classification learning, respectively.

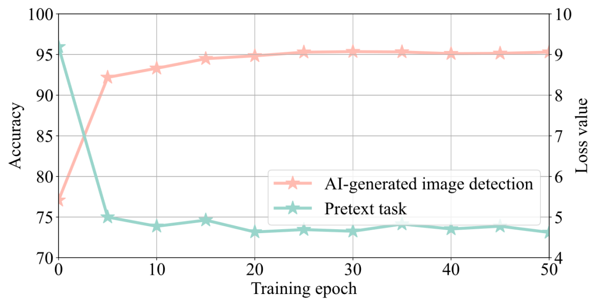

Figure 5. The relationship between the pretext task and AI-generated image detection.

evaluate its generalization against other generators, such as non-latent diffusion models, flow-based models, and GANs. This experiment demonstrates that our feature extractor recognizes universal low-level features of images rather than overfitting to the artifacts of stable diffusion. Table 3 presents the recall rates for various AI-generated images. The Gaussian Mixture Model used in our method is trained on real photographic images from MSCOCO. Images with likelihood scores below our threshold are flagged as AI-generated. The threshold is determined by the likelihood score of photographic images in the training set. We adopt DCRT as a baseline, which also employs a pre-trained SDv1.4 during the training phase of the detector. Our method achieves remarkable success in identifying nearly all AI-generated images. The likelihood scores for these images are significantly lower than the threshold established for photographic images.

4.3. Ablation Studies

We first analyze the relationship between the pretext task and AI-generate image detection accuracy. Fig. 5 illustrates the loss values for the pretext task and detection accuracy over the training epochs of the feature extractor. The results show a strong correlation between the two tasks. As training epochs progress, the capability of the feature extractor to discern low-level features improves, leading to an increase in detection accuracy.

Then, we focus on the low-level feature extractor construction. In the feature extractor training phase, we utilize a pre-trained stable diffusion model as an off-the-shelf tool to create multiple pairs of similar images that differ only at the pixel level. We employ two alternative denoising tools: a pre-trained CNN-based denoising network BF-CNN [29] designed to eliminate common Gaussian noise and an early diffusion model ADM [8]. We introduce varying intensities of Gaussian noise to the spatial domain of the original images and obtain the corresponding denoised versions using the pre-trained BF-CNN and ADM. The results are presented in Table 4. While these two denoising tools are less effective than our approach, they still demonstrate a certain level of effectiveness. We speculate that the denoising capabilities of BF-CNN and ADM are less effective than those of stable diffusion. This disparity leads to significant differences between the original and restored images beyond the pixel level. Consequently, the feature extractor is unable to effectively learn the low-level features of the images.

In terms of the objection function of our pretext, we utilize cross-entropy loss and regression loss. Table 4 shows the ablation studies of these two parts. Our approach, which relies solely on either regression or noisy classification learning, is effective against AI-generated images. Whether using regression learning or noise classification, the feature extractor can focus on pixel-level image differences. However, the combination of regression and classification learning slightly outperforms each individually.

Additional ablation studies on the number of photographic images for feature distribution estimation, the number of Gaussian components, and threshold settings are provided in the supplementary materials.

5. Conclusions

In this paper, we propose a novel low-level image feature extractor designed to capture pixel-level differences between images. The crux of the feature extractor lies in the pretext task design. We utilize a pre-trained diffusion model as a denoising tool to create multiple pairs of original and denoised images that only differ at the pixel level. The pre-trained feature extractor transforms spatial images into a well-separated space, in which different sources including real photographic and AI-generated images exhibit different distributions. Drawn inspiration from the feature separability, we frame the AI-generated image detection as a one-classification task. We model the feature distribution of real photographic images by the Gaussian Mixture Model. Consequently, images whose low-level features deviate from the estimated distribution are flagged as AI-generated. Extensive experiments demonstrate that our approach effectively identifies AI-generated images from various generative models.

References

[1] Midjourney. https://www.midjourney.com/home/, 2023. Accessed: 2023-07-01. 1, 2, 3

[2] George Cazenavette, Avneesh Sud, Thomas Leung, and Ben Usman. FakeInversion: Learning to detect images from unseen text-to-image models by inverting stable diffusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10759–10769, 2024. 1

[3] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019. 1

[4] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41 (3):1–58, 2009. 1

[5] Baoying Chen, Jishen Zeng, Jianquan Yang, and Rui Yang. DRCT: Diffusion reconstruction contrastive training towards universal detection of diffusion generated images. In *International Conference on Machine Learning*, 2024. 2, 5

[6] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018. 7

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 2, 5

[8] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, pages 8780–8794, 2021. 1, 2, 8

[9] 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 16×16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 5

[10] Ricard Durall, Margret Keuper, and Janis Keuper. Watch your up-convolution: CNN based generative deep neural networks are failing to reproduce spectral distributions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7890–7899, 2020. 3

[11] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *International Conference on Machine Learning*, pages 3247–3258, 2020. 2

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 1

[13] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10696–10706, 2022. 1, 4

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, pages 6840–6851, 2020. 1, 2, 7

[15] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A Efros. Fighting fake news: Image splice detection via learned self-consistency. In *European Conference on Computer Vision*, pages 101–117, 2018. 2

[16] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, pages 213–238, 2018. 1

[17] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 7

[18] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224, 2018. 1, 7

[19] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*, 2014. 1

[20] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pages 4743–4751, 2016. 1

[21] Yanyu Li, Huan Wang, Qing Jin, Ju Hu, Pavlo Chemerys, Yun Fu, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. SnapFusion: Text-to-image diffusion model on mobile devices within two seconds. In *Advances in Neural Information Processing Systems*, pages 20662–20678, 2024. 2

[22] 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 *European Conference on Computer Vision*, pages 740–755, 2014. 5

[23] 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, 2022. 2, 5

[24] Zhengzhe Liu, Xiaojuan Qi, and Philip HS Torr. Global texture enhancement for fake face detection in the wild. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8060–8069, 2020. 5

[25] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022. 5

[26] Jan Lukas, Jessica Fridrich, and Miroslav Goljan. Digital camera identification from sensor pattern noise. *IEEE Transactions on Information Forensics and Security*, 1(2): 205–214, 2006. 2

[27] Yunpeng Luo, Junlong Du, Ke Yan, and Shouhong Ding. LaRE^2: Latent reconstruction error based method for diffusion-generated image detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17006–17015, 2024. 1

[28] Ruipeng Ma, Jinhao Duan, Fei Kong, Xiaoshuang Shi, and Kaidi Xu. Exposing the fake: Effective diffusion-generated images detection. *arXiv preprint arXiv:2307.06272*, 2023. 1

[29] Sreyas Mohan, Zahra Kadkhodaie, Eero P Simoncelli, and Carlos Fernandez-Granda. Robust and interpretable blind image denoising via bias-free convolutional neural networks. *arXiv preprint arXiv:1906.05478*, 2019. 8

[30] Todd K Moon. The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60, 1996. 5

[31] 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*, pages 16784–16804, 2022. 3

[32] James F O'brien and Hany Farid. Exposing photo manipulation with inconsistent reflections. *ACM Transactions on Graphics*, 31(1):4:1–4:11, 2012. 2

[33] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24480–24489, 2023. 3, 5

[34] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019. 7

[35] Alin C Popescu and Hany Farid. Exposing digital forgeries by detecting traces of resampling. *IEEE Transactions on Signal Processing*, 53(2):758–767, 2005. 2

[36] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *European conference on Computer Vision*, pages 86–103, 2020. 5

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763, 2021. 2, 3, 5

[38] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 7

[39] Douglas Reynolds. Gaussian mixture models. *Encyclopedia of Biometrics*, 741(659-663), 2009. 5

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 2, 3, 4, 5

[41] Zeyang Sha, Zheng Li, Ning Yu, and Yang Zhang. De-fake: Detection and attribution of fake images generated by text-to-image generation models. In *2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 3418–3432, 2023. 5

[42] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can CLIP benefit vision-and-language tasks? *arXiv preprint arXiv:2107.06383*, 2021. 3

[43] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 3, 5

[44] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 2

[45] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, pages 32211–32252, 2023. 2

[46] Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, and Yunchao Wei. Learning on gradients: Generalized artifacts representation for GAN-generated images detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12105–12114, 2023. 1, 2, 5

[47] Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao Wei. Rethinking the up-sampling operations in CNN-based generative network for generalizable deepfake detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28130–28139, 2024. 5

[48] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 5

[49] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9 (11), 2008. 3

[50] Alin C Popescu and Hany Farid. Exposing digital forgeries in color filter array interpolated images. *IEEE Transactions on Signal Processing*, 53(10):3948–3959, 2005. 2

[51] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. CNN-generated images are surprisingly easy to spot... for now. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8695–8704, 2020. 1, 2, 5

[52] Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, Hezhen Hu, Hong Chen, and Houqiang Li. DIRE for diffusion-generated image detection. In *IEEE/CVF International Conference on Computer Vision*, pages 22445–22455, 2023. 1, 2, 5

[53] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 5

[54] Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K Wong. Uni-ControlNet: All-in-one control to text-to-image diffusion models. In *Advances in Neural Information Processing Systems*, pages 11127–11150, 2024. 2

[55] Chenhao Zheng, Ayush Shrivastava, and Andrew Owens. Exif as language: Learning cross-modal associations between images and camera metadata. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6945–6956, 2023. 2

[56] Nan Zhong, Yiran Xu, Sheng Li, Zhenxing Qian, and Xinpeng Zhang. Patchcraft: Exploring texture patch for efficient ai-generated image detection. *arXiv preprint arXiv:2311.12397*, 2023. 1

[57] Mingjian Zhu, Hanting Chen, Qiangyu Yan, Xudong Huang, Guanyu Lin, Wei Li, Zhijun Tu, Hailin Hu, Jie Hu, and Yunhe Wang. GenImage: A million-scale benchmark for detecting ai-generated image. In *Advances in Neural Information Processing Systems*, pages 77771–77782, 2023. 5