

LOTA: Bit-Planes Guided AI-Generated Image Detection

Hongsong Wang^{1,2,*†}, Renxi Cheng^{3†}, Yang Zhang⁴, Chaolei Han³, Jie Gui^{3,5,6*}

¹School of Computer Science and Engineering, Southeast University, Nanjing 210096, China

²Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, China

³School of Cyber Science and Engineering, Southeast University, Nanjing 210096, China

⁴School of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China

⁵Purple Mountain Laboratories, Nanjing 210000, China

⁶Engineering Research Center of Blockchain Application, Supervision And Management (Southeast University), Ministry of Education, China

{hongsongwang, renxi, chaoleihan, guijie}@seu.edu.cn, yangzhang@szu.edu.cn

Abstract

*The rapid advancement of GAN and Diffusion models makes it more difficult to distinguish AI-generated images from real ones. Recent studies often use image-based reconstruction errors as an important feature for determining whether an image is AI-generated. However, these approaches typically incur high computational costs and also fail to capture intrinsic noisy features present in the raw images. To solve these problems, we innovatively refine error extraction by using bit-plane-based image processing, as lower bit planes indeed represent noise patterns in images. We introduce an effective bit-planes guided noisy image generation and exploit various image normalization strategies, including scaling and thresholding. Then, to amplify the noise signal for easier AI-generated image detection, we design a maximum gradient patch selection that applies multi-directional gradients to compute the noise score and selects the region with the highest score. Finally, we propose a lightweight and effective classification head and explore two different structures: noise-based classifier and noise-guided classifier. Extensive experiments on the GenImage benchmark demonstrate the outstanding performance of our method, which achieves an average accuracy of **98.9%** (**11.9%** \uparrow) and shows excellent cross-generator generalization capability. Particularly, our method achieves an accuracy of over 98.2% from GAN to Diffusion and over 99.2% from Diffusion to GAN. Moreover, it performs error extraction at the millisecond level, nearly a hundred times faster than existing methods. The code is at <https://github.com/hongsong-wang/LOTA>.*

1. Introduction

With the rapid development of generative models, especially Generative Adversarial Networks (GANs) [15] and Diffusion models [32], AI-generated images are becoming more and more realistic, and it is even difficult for people to distinguish the difference between real and AI-generated images. These AI-generated images may also be used for illegal purposes [4, 20], such as spreading unreal information or harmful content, which may mislead or harm the public. Therefore, there is an urgent need for a robust technique for distinguishing AI-generated images from real ones.

Early deep learning-based deepfake methods focus on the detection of GAN-generated images. However, recent works [9, 31] find that with the emergence of diffusion models, detection performance declines significantly when GAN-based detection methods are applied to diffusion-generated images. With regard to detecting diffusion-generated images, many works are based on reconstructed image errors, *e.g.*, DIRE [38], SeDID [28], LaRE² [27], ESSP [7], ZED [10]. For example, DIRE [38] computes the error between the raw and reconstructed images, and considers the image with smaller error to be AI-generated. SeDID [28] computes the loss error for a given step in the forward and reverse process of Diffusion, and regards the image with smaller error as AI-generated. These approaches extract error signals through a multi-step DDIM sampling process, which is not only inefficient but also susceptible to introducing random noise at different steps.

Least Significant Bit (LSB)-based steganography is a simple yet effective technique that embeds a secret message in pixel values while minimizing perceptible distortions [18]. LSB-based steganography can be easily extended to multiple bit-planes, with lower ones prioritized

*Corresponding authors. [†]Equal Contribution

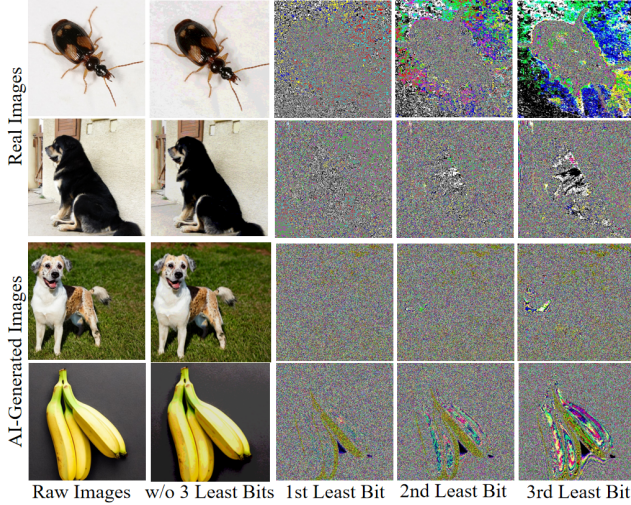


Figure 1. **Comparison of least bit-planes between real images and AI-generated images.** We extract the 1st, 2nd and 3rd least bit-planes from both types of images, separately. We find that images of low bit-planes of real images and AI-generated images have different noise patterns and distributions that can be used for distinguishing between them. Low bit-planes of fake images contain artifacts that are invisible in RGB images.

to preserve perceptual quality and local properties checked when using higher bit-planes, while the k -least significant bits can also be utilized for steganography [14, 21]. However, most LSB-based methods are primarily limited to the fields of image steganography and steganalysis.

Bit-planes based approaches have the potential to address AI-generated image detection, as they are capable of exploiting subtle differences in pixel values and detecting artifacts that are typically absent in natural images. We visualize and compare the least bit planes for both real and AI-generated RGB images in Figure 1. It can be seen that, although removing the least significant three bit-planes has almost no impact on the visual appearance of both real and AI-generated RGB images, differences still exist in the bit-plane images between real and AI-generated RGB images. Compared to real images, the brightness of the least significant bit-planes in AI-generated images is low, or the brightness distribution is irregular. Moreover, low bit-planes of fake images contain artifacts. One possible reason is that current image generation models lack the capability to generate visually imperceptible details. Since there are no such works on AI-generated image detection, we aim to fill this gap and leverage imperceptible bit-planes for this purpose.

To this end, we introduce a simple yet effective approach called LOW-bit pAtch (LOTA) for detecting AI-generated images. LOTa consists of three key modules: Bit-planes Guided Noisy Image Generation (BGNIG), Maximum Gradient Patch Selection (MGPS) and classification head. BG-

NIG takes lower bit-planes, which contains noise, to extract the error image and achieves high efficiency and accuracy. To enhance the brightness of the noisy image, we explore two normalization methods: scaling and thresholding. To further amplify the noise signal for subsequent detection, the MGPS calculates the noise score using multi-directional gradients and selects the patch with the highest score. Finally, we introduce Noise-Based Classifier (NBC) and Noise-Guided Classifier (NGC). The NBC is a simple convolutional neural network based solely on the noise image, while the NGC uses noise patches to effectively guide fake detection from the raw image. We conduct extensive experiments on GenImage [42], where images are generated by eight different generators. Compared to existing mainstream methods, our approach is more effective, faster, and more generalizable. The main contributions are as follows:

- **Novel solution for AI-generated image detection:** We innovatively address AI-generated image detection based on bit-planes, and propose an efficient approach for noisy representation extraction.
- **Efficient pipeline design:** We propose a simple yet effective pipeline with three modules: noise generation, patch selection and classification. We design a heuristic strategy called maximum gradient patch selection and introduce two effective classifiers: noise-based classifier and noise-guided classifier. Our approach operates at millisecond level, nearly a hundred times faster than current methods.
- **Exceedingly superior performance:** Extensive experiments demonstrate the effectiveness of LOTa, which achieves **98.9%** ACC on GenImage, showing great cross-generator generalization capability and outperforming existing mainstream methods by more than **11.9%**.

2. Related Work

Recent advances in AI-generated image detection have focused on exploiting artifacts across different domains. We briefly review works belonging to the following categories. **Spatial Domain-Based Methods:** Most detecting methods are based on the spatial domain. Early studies primarily analyze pixel-level texture patterns and geometric inconsistencies. Wang *et al.* [37] demonstrate that CNN-generated images tend to exhibit distinguishable artifacts that can be detected. Subsequent studies extend this to real-image priors [24] and generalized gradient artifacts [34]. With the rise of Diffusion models [32], DIRE [38] considers the reconstruction error as an essential metric for detecting generated images, and GLFF [19] fuses global and local features to capture multi-scale inconsistencies. Recently, DRCT [6] realizes universal detection through contrastive reconstruction. ESSP [7] attains outstanding performance by using single-patch analysis. Additionally, geometric inconsistencies [33] and zero-shot frameworks [10] further extend spatial domain analysis.

Frequency Domain-Based Methods: Frequency analysis reveals artifacts often imperceptible in pixel space. The seminal work of [40] identified GAN-specific frequency artifacts. Later, Dzanic *et al.* [13] focus on high-frequency features to better simulate real images. Chandrasegaran *et al.* [5] validate these findings for existing CNN-based generative models. Corvi *et al.* [8, 9] extend frequency analysis to Diffusion models by exploring distinct fingerprints and differences. Recent frequency masking techniques [12] further enhance generalization across different generators. These methods face challenges in handling high-resolution images and adaptive generation strategies.

Multi-Domain Feature Fusion-Based Methods: Traditional methods have difficulties coping with increasingly high-quality generated images, so several studies try to integrate complementary signals from multiple domains. Yu *et al.* [39] combine texture and semantic features to detect manipulated faces, and Luo *et al.* [27] combine reconstructed error features and latent space features to detect generated images. Efficiency is also prioritized. Lanzino *et al.* [22] employ binary neural network by combining frequency-domain features, local texture features and pixel-domain features. Loporoni *et al.* [23] fuse RGB and depth features to exploit 3-Dimension inconsistencies. These methods demonstrate that multi-domain fusion enhances robustness against evolving generation techniques.

Image Error-Based Methods: Regarding the detection of diffusion-generated images, several recent methods rely on error computation, focusing on the differences between reconstructed and raw images. DIRE [38] calculates reconstructed image errors to distinguish generated images from real ones. SeDID [28] extracts the loss error for a given processing step in the Diffusion process. LaRE² [27] first computes noise image within the diffusion-based framework, then introduces both spatial and channel feature refinement to enhance feature learning. Chen *et al.* [7] exploit the noise pattern of an image for detection and confirm that more noise indicates a higher possibility of being real images. Cozzolino *et al.* [10] calculate the differences between expected and actual coding cost of an image for detection. Different from these approaches, we attempt to extract the noise signal contained within the image itself.

3. Method

We introduce LOTA for AI-generated image detection. LOTA comprises three subsequent modules: Bit-Planes Guided Noisy Image Generation, Maximum Gradient Patch Selection and classification head. An illustration of LOTA is provided in Figure 2, with details described below.

3.1. Bit-Planes Guided Noisy Image Generation

The reconstruction error images, such as those from DIRE [38], LaRE² [27] and ESSP [7], extract error maps

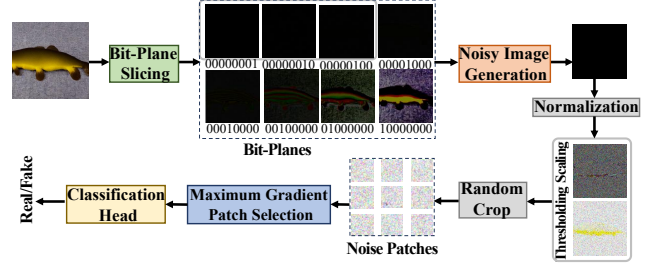


Figure 2. **Overview of our method.** First, we decompose the image into 8 bit-planes, and compose least bit-planes to generate the noise representation. Second, we crop the noise image into several patches, and select the patch with the highest gradient-based score. Finally, a classification head is applied.

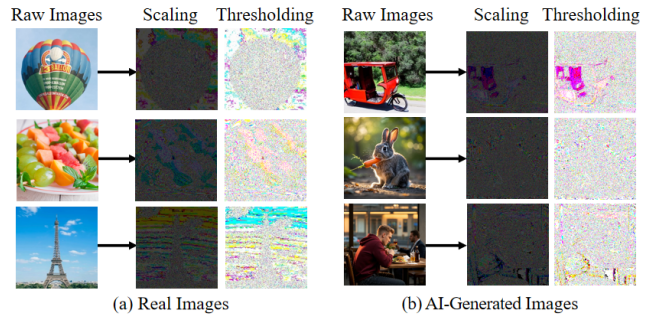


Figure 3. **Visualizations of generated noisy images by scaling and thresholding in the BGNIG module.** We compare corresponding noise images for real and AI-generated images. Both scaling and thresholding methods effectively extract the noise patterns of images. The brightness distribution of noisy images from real images is relatively regular. However, noisy images from synthetic images contain several regions with artifacts.

or noise patterns, that can be effectively utilized for detecting generated images. Inspired by this, we use noise images for generated image detection. However, instead of relying on reconstruction error, we exploit the noise map inherently contained in the image.

A bit-plane of an image consists of the bits at a specific position in the binary representation of each pixel. Since a gray-scale image is typically represented with eight bits per pixel, it contains eight bit-planes in total. Then a RGB image can be seen to be composed of three gray-scale images of each channel. Let x^c be the RGB image of the channel c , where $c \in \{R, G, B\}$, and let x_k^c denote its corresponding k -th bit-plane of the channel c , where $0 \leq k \leq 7$, then the image of this channel can be decomposed as:

$$x^c = \sum_{k=0}^7 2^k \cdot x_k^c. \quad (1)$$

Dividing an image into multiple bit planes is known as bit-plane slicing. Higher-order bit-planes contain visual infor-

mation such as textures and colors, while lower-order bit-planes preserve details including contours and noises.

To extract noise patterns in images, we select the three lowest-order bit-planes of each channel, namely x_2^c , x_1^c , and x_0^c , to generate a low-bit image. The lowest-order bit-planes of the image are composed with addition operations. Specifically, the following formulation is applied:

$$z^c = 2^2 \cdot x_2^c + 2 \cdot x_1^c + x_0^c, \quad (2)$$

where z^c denotes the composed low-bit image for each channel of the RGB image.

Since the pixel values in z^c range from 0 to 7, normalization needs to be applied before extracting image features. Two distinct methodologies are employed: scaling and thresholding.

Scaling: The min-max normalization is used to scale these values to [0, 255]:

$$\tilde{z}^c = 255 \cdot \frac{z^c - z_{\min}^c}{z_{\max}^c - z_{\min}^c}, \quad (3)$$

where \tilde{z}^c denotes the c -th channel of normalized noise \tilde{z} .

Thresholding: Since the values in the lower bit planes are sparse, we mitigate this issue by directly setting all values greater than 0 to 255. This approach is called thresholding, which enhances the brightness of the normalized image. The formulation of thresholding is:

$$\tilde{z}_{i,j}^c = \begin{cases} 0, & \text{if } z_{i,j}^c = 0, \\ 255, & \text{if } z_{i,j}^c > 0, \end{cases} \quad (4)$$

where $z_{i,j}^c$ represents the element at the i -th row and j -th column of z^c in Eq. (2).

Figure 3 shows visualizations of noisy images generated by two different approaches for both real and AI-generated images. We observe that for real images, the brightness distribution of noisy images is relatively regular, with visible object contours and some texture information. In contrast, for AI-generated images, the brightness distribution of noisy images is relatively chaotic, making it difficult to discern the contours of objects in original images.

3.2. Maximum Gradient Patch Selection

Though the noise pattern in low-bit images serves as a critical feature for distinguishing real and generated images, it still contains much useless information that may interfere with detection. To further extract the intrinsic feature which can distinguish real and generated images in essence, we introduce Maximum Gradient Patch Selection (MGPS) to select the most informative patch from an image for further detection.

For the low-bit images \tilde{z} , we randomly divide them into non-overlapping patches. Then, we design a divergence-based score function to measure the sparsity of image gradients in different directions. Let \tilde{z}_p be the noisy patch,

where p denotes the index of the patch numbers, the score g_p is computed as:

$$g_p = \|\tilde{z}_p * g_x\|_1 + \|\tilde{z}_p * g_y\|_1 + \|\tilde{z}_p * g_{xy}\|_1 + \|\tilde{z}_p * g_{yx}\|_1, \quad (5)$$

where $*$ represents the image convolution operation, $\|\cdot\|_1$ denotes $L1$ norm of the matrix, g_x, g_y, g_{xy} and g_{yx} are convolution kernels described as:

$$g_x = \begin{bmatrix} -1 & 1 \end{bmatrix}, \quad g_y = \begin{bmatrix} 1 & -1 \end{bmatrix}^T = g_x^T, \\ g_{xy} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad g_{yx} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

The first two terms of the score represent horizontal and vertical gradients, while the latter two terms represent the diagonal gradients. Since the patch denotes image noise, the scores with large values often correspond to regions with excessive high-frequency variations, which are likely dominated by noise or structural details rather than image content. For AI-generated images, these high-divergence regions might indicate artifacts caused by imperfect generative models.

Thus, we select the noise patch with the highest g_p score:

$$\tilde{z}_{p^*} = \arg \max_p g_p, \quad (6)$$

where p^* denotes the index of the best patch.

It should be noted that although both our approach and ESSP [41] select a simple patch, our MGPS differs from ESSP in three key aspects. First, we compute the gradient-based score instead of the texture diversity score. Second, we formulate the score function using a concise and efficient image convolution operation. Third, we choose the patch with the highest score instead of the lowest one.

3.3. Classification Head

After obtaining the selected patch of noise image, we present two methods as classifiers for AI-generated image detection. The structures of the two classifiers are illustrated in Figure 4, with details described below.

Noise-Based Classifier: A straightforward approach is to use a convolutional neural network pre-trained on ImageNet as the classifier for low-bit patch images. Since the patch is small, it needs to be resized to the standard size of 256×256 before being fed into the convolutional classifier.

Noise-Guided Classifier: Current studies mainly focus on finding a more effective error extraction method [7, 10, 28, 38], but neglect the information of raw images. So we align the raw and error maps from the spatial perspective to provide more reliable information for classification.

For the raw image x , we first put it through the image encoder (e.g., ResNet-50) to get the feature map \tilde{x} , then the pooling, flatten and flatten are respectively used to get

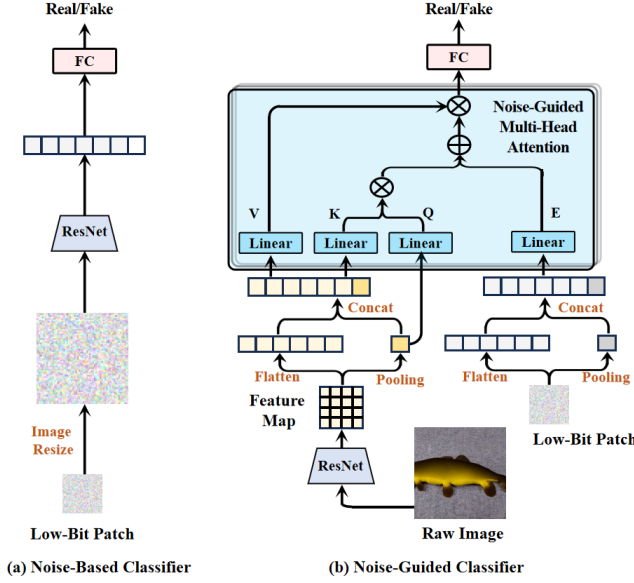


Figure 4. **Structure of the classifier.** Two different classifiers are applied. For the noise-based classifier, the low-bit patch is directly fed into the ResNet. For the noise-guided classifier, the low-bit patch and the feature map of raw images are combined by using the noise-guided multi-head attention.

query Q , key K and value V for the spatial attention. For the error patch \tilde{z}_{p^*} , it is flattened and projected to get error E for the spatial attention. So the Noised-Guided Multi-Head Attention can be defined as:

$$U = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} + E \right) V, \quad (7)$$

where $\text{softmax}(\cdot)$ is the activation function, d_k is the dimension of the tensor K . Based on this, we further apply the multi-head Attention according to Transformer [36].

The output vector is then followed by a fully connected layer with binary cross-entropy loss to distinguish between real and generated images.

4. Experiments

4.1. Dataset and Implementation Details

Dataset and Evaluation Metrics: We evaluate the proposed method on GenImage dataset [42], which employs ImageNet dataset as real images, and incorporates eight mainstream GAN and Diffusion generators (including BigGAN [3], Midjourney [1], Wukong [2], Stable Diffusion V1.4 [32], Stable Diffusion V1.5 [32], ADM [11], GLIDE [29], and VQDM [16]) to generate AI-generated images. The dataset comprises a total of 1,331,167 real images and 1,350,000 generated images. The data corresponding to each generator are split into training and testing subsets. For each classifier, training was conducted on the training sub-

sets, followed by comprehensive evaluation across all eight testing subsets. Following existing works for AI-generated image detection [7, 27, 38], we adopt accuracy (ACC) and average precision (AP) as evaluation metrics. The threshold for computing accuracy is 0.5.

Implementation Details: Before Maximum Gradient Patch Selection (MGPS), the noise image is resized to a resolution of 256×256 . In MGPS, a 32×32 patch is selected. For noise-guided classifier, the patch is directly fed into the classifier, combined with corresponding raw images. For Noise-Based Classifier, the patch is first resized to 256×256 . ResNet-50 is used as the image encoder for the classifier. During training, the learning rate is 0.0001, and the batch size is 64. The maximum number of training epochs is 30, with Adam as the optimizer.

4.2. Experimental Results

Analysis of Experimental Results: We train our model on eight training subsets of GenImage and evaluate each trained model on all eight testing subsets in Table 1. The default LOTA uses thresholding when generating low-bit images and employs noise-based classification. The two other variants, LOTA-*scl*. and LOTA-*ngc*, use scaling and noise-guided classifier, respectively. We compare the noise image generation approaches of scaling and thresholding, and find that thresholding achieves slightly higher average accuracy compared to scaling. From Figure 5, we find that results of the three subsets (e.g., Wukong, Stable Diffusion V1.4, and Stable Diffusion V1.5) are highly correlated, which may be attributed to the high correlation of the generators.

Comparison with State-of-the-Arts: In Table 2, the proposed LOTA achieves an average accuracy of 98.9%, while its two variants, LOTA-*scl*. and LOTA-*ngc*, achieve 98.7% and 93.2%, respectively. All three significantly outperform current mainstream methods. Specifically, for error extraction based methods, LOTA shows improvements of approximately 11.9% over ESSP [7], 19.6% over LaRE² [27], and 25.5% over DIRE [38].

Comparison of Cross-Generator Generalization: Results of cross-generator generalization are compared in Figure 5. Existing methods based on error extraction predominantly exhibit darker colors along the main diagonal, indicating their optimal performance only when training and testing generators are identical. While LaRE² [27] and ESSP [7] show improved cross-generator generalization with darker off-diagonal entries, their capability remains confined to homologous generators (e.g., Diffusion model generators including Stable Diffusion V1.4 [32], Stable Diffusion V1.5 [32], ADM [11], and GLIDE [29]). In contrast, LOTA demonstrates uniformly distributed color patterns across all rows and columns, reflecting its superior cross-generator generalization. Notably, even the lightest-colored column representing Midjourney achieves higher detection

Table 1. **Detailed results on the GenImage dataset.** The model is trained on eight subsets of GenImage and tested on the corresponding subsets. The default LOTA employs thresholding during noisy image generation and Noise-Based Classifier for the classification head. The two other variants, LOTA-*scl.* and LOTA-*ngc*, use scaling and noise-guided classifier, respectively.

Train Subset	Method	BigGAN	Midjourney	Wukong	SD V1.4	SD V1.5	ADM	GLIDE	VQDM	Avg.
BigGAN	LOTA	100	91.3	99.8	99.9	99.9	99.4	98.2	99.4	98.6
	LOTA- <i>scl.</i>	100	91.4	99.8	99.9	99.9	99.5	98.3	99.5	98.6
	LOTA- <i>ngc</i>	100	82.0	63.5	62.2	62.7	84.8	83.8	88.4	76.3
Midjourney	LOTA	98.9	98.8	99.0	99.0	98.9	99.0	99.0	99.0	99.0
	LOTA- <i>scl.</i>	98.9	98.8	98.1	98.9	98.9	99.0	99.1	98.0	98.9
	LOTA- <i>ngc</i>	97.4	99.7	99.7	99.7	99.7	99.8	99.3	99.3	98.6
Wukong	LOTA	100	93.2	100	99.9	100	99.8	99.9	99.7	99.1
	LOTA- <i>scl.</i>	100	91.6	99.8	99.9	99.9	99.5	98.3	99.5	98.6
	LOTA- <i>ngc</i>	85.7	91.3	99.9	100	100	100	100	100	97.9
SD V1.4	LOTA	100	91.3	99.8	99.9	99.9	99.4	98.2	99.4	98.5
	LOTA- <i>scl.</i>	100	91.6	99.8	99.9	99.9	99.5	98.3	99.5	98.6
	LOTA- <i>ngc</i>	70.4	93.6	100	100	100	100	100	100	95.6
SD V1.5	LOTA	100	93.1	100	100	100	99.7	100	99.7	99.1
	LOTA- <i>scl.</i>	100	91.6	99.8	99.9	99.9	99.5	98.3	99.5	98.6
	LOTA- <i>ngc</i>	94.9	92.3	100	100	100	100	100	100	96.8
ADM	LOTA	100	91.2	99.8	99.9	99.9	99.4	98.2	99.4	98.5
	LOTA- <i>scl.</i>	100	91.6	99.8	99.9	99.9	99.5	98.4	99.5	98.6
	LOTA- <i>ngc</i>	71.1	91.0	99.9	99.9	99.9	100	100	100	95.2
GLIDE	LOTA	100	94.0	100	100	100	99.8	100	99.8	99.2
	LOTA- <i>scl.</i>	99.2	96.4	99.3	99.2	99.1	99.2	99.2	99.2	98.9
	LOTA- <i>ngc</i>	90.5	85.6	92.0	91.2	91.1	99.3	99.2	99.4	92.1
VQDM	LOTA	99.9	92.7	100	99.9	100	99.6	99.8	99.6	99.0
	LOTA- <i>scl.</i>	100	91.6	99.8	99.9	99.9	99.5	98.3	99.5	98.6
	LOTA- <i>ngc</i>	94.5	87.3	91.9	92.3	92.8	100	100	100	92.7

Table 2. **Comparison of averaged accuracy against existing methods on the GenImage dataset.** Models are trained and tested on eight subsets of GenImage, and the average accuracy is reported. LOTA-*scl.* and LOTA-*ngc* denote the variants that use scaling and noise-guided classifier, respectively. ‘*’ means the results are reproduced by ourselves.

Method	Error-Based	BigGAN	Midjourney	Wukong	SD V1.4	SD V1.5	ADM	GLIDE	VQDM	Avg.
CNNSpot [37]	✗	56.6	58.2	67.7	70.3	70.2	57.0	57.1	56.7	61.7
F3Net [30]	✗	56.5	55.1	72.3	73.1	73.1	66.5	57.8	62.1	64.6
GramNet [25]	✗	61.2	58.1	71.3	72.8	72.7	58.7	65.3	57.8	64.7
Spec [40]	✗	64.3	56.7	70.3	72.4	72.3	57.9	65.4	61.7	65.1
ResNet-50 [17]	✗	66.6	59.0	71.4	72.3	72.4	59.7	73.1	60.9	66.9
DeiT-S [35]	✗	66.3	60.7	73.1	74.2	74.2	59.5	71.1	61.7	67.6
Swin-T [26]	✗	69.5	61.7	75.1	76.0	76.1	61.3	76.9	65.8	70.3
DIRE* [38]	✓	56.7	59.7	74.6	74.7	74.7	68.8	69.3	68.8	73.4
LaRE ² * [27]	✓	74.0	66.4	85.5	87.5	87.3	66.6	81.3	84.4	79.4
ESSP* [7]	✓	78.3	80.8	93.5	94.2	84.4	82.1	92.1	91.0	87.0
LOTA- <i>scl.</i>	✓	99.8	93.1	99.5	99.7	99.7	99.4	98.5	99.3	98.7
LOTA- <i>ngc</i>	✓	88.1	90.4	93.4	93.2	93.3	98.0	97.8	98.4	93.2
LOTA	✓	99.9	93.2	99.8	99.8	99.8	99.5	99.2	99.5	98.9

accuracy than results from other mainstream methods.

4.3. Ablation Studies and Analysis

We conduct ablation studies to validate the effectiveness of each module. We train models on the Stable Diffusion V1.5 [32] subset and test on all 8 subsets. By default, we use the LOTA variant with a noise-guided classifier as our baseline, denoted as LOTA-*ngc*. For simplicity, the subsets of BigGAN, Midjourney, Wukong, SD V1.4, SD V1.5, ADM, GLIDE, and VQDM are abbreviated as Big, Mid,

Wuk, SD4, SD5, ADM, GLI, and VQD, respectively.

Ablation Studies: To validate the effectiveness of each module, we respectively remove each module proposed in our method: 1) w/o BGNIG: we employ raw images instead of low-bit images for detection. 2) w/o MGPS: we do not crop images into patches, and use images of original size. 3) w/o NBC/NGC: we replace the noise-based or noise-guided classifier with a fully-connected layer. As shown in Table 3, it exhibits a clear accuracy decline (especially testing on BigGAN and Midjourney) when using

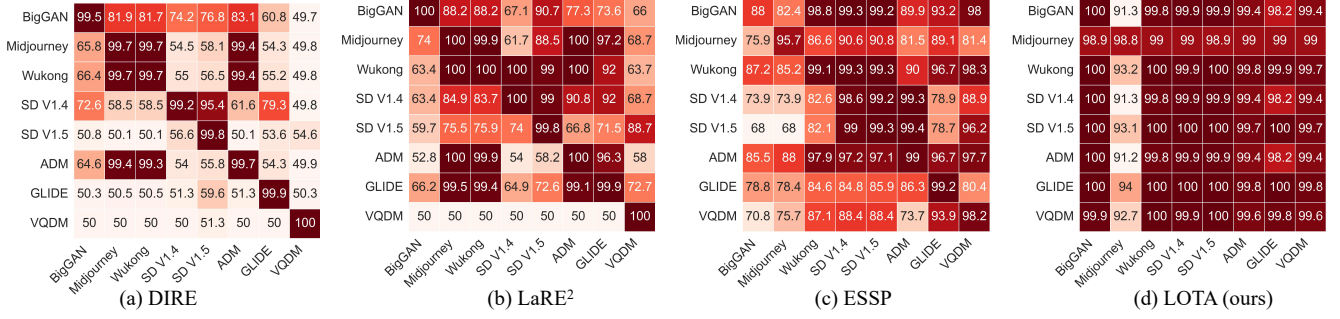


Figure 5. **Comparison of cross-generator generalization with existing methods.** DIRE [38], LaRE² [27] and ESSP [7] are selected as comparison methods. These models and ours are trained on eight training subsets and tested on eight testing subsets in the GenImage. The noise-based classifier is applied on both training and testing. The vertical axis represents the training subsets, and the horizontal axis represents the test subsets. Darker colors indicate higher accuracy.

Table 3. **Ablation.** We conduct ablation studies by removing the modules of BGNIG, MGPS and NBC/NGC, respectively.

Ablations	Big	Mid	Wuk	SD4	SD5	ADMGLI	VQD	Avg.
LOTA- <i>ngc</i>	94.9	92.3	100	100	100	100	100	96.8
w/o BGNIG	77.6	92.8	92.5	80.5	99.8	99.8	88.7	87.5
w/o MGPS	82.2	81.1	94.6	96.8	96.6	77.3	96.7	90.9
w/o NBC/NGC	52.9	52.4	58.1	55.1	55.2	50.5	51.7	50.8

Table 5. **Impact of Patch Selection Strategy.** We select patches randomly, by highest score, or by lowest score.

Selection	Big	Mid	Wuk	SD4	SD5	ADMGLI	VQD	Avg.
Random	69.9	98.1	99.8	99.8	99.8	99.4	98.6	93.1
Min	79.0	91.9	99.0	99.2	99.2	96.1	95.3	89.6
Max	94.9	92.3	100	100	100	100	100	96.8

raw images only, which demonstrates that low-bit planes extraction effectively exploits the intrinsic patterns of images and supplements the capability of cross-generator generalization. We also find there is a significant accuracy drop when NBC and NGC classifier are discarded, indicating the feature refinement capability of our proposed classifier.

Impact of Patch Size: The selection of patch size is critical for further detection, as a too large patch would introduce texture or useless information, and a too small patch would weaken the noise pattern for detection. So we vary the selection of patch size, including 16×16 , 32×32 , 48×48 , and 64×64 . As shown in Table 4, as the patch size increases, the average accuracy first increases and then decreases. When using the patch of size 32×32 , the average accuracy is much higher than other cases. Apart from this, we also find that fluctuations in average accuracy are due to variations in BigGAN subset (a heterologous generator), so the patch size also influences the cross-generator generalization. So we choose the patch of size 32×32 to maximize the cross-generator generalization.

Impact of Patch Selection Strategy: In the MGPS module, we choose the patch with the highest gradient-based

Table 4. **Impact of Patch Size.** We evaluate different size of patch, e.g., 16×16 , 32×32 , 48×48 , and 64×64 , for the MGPS module.

Patch Size	Big	Mid	Wuk	SD4	SD5	ADMGLI	VQD	Avg.
16×16	50.2	95.7	99.9	99.9	99.9	99.9	99.9	92.7
32×32	94.9	92.3	100	100	100	100	100	96.8
48×48	83.8	78.0	100	100	100	100	100	91.4
64×64	53.6	84.0	100	100	99.9	100	100	90.8

Table 6. **Impact of Classifier.** We choose a simple FC layer, noise-based and noise-guided classifier as the final classifier.

Classifier	Big	Mid	Wuk	SD4	SD5	ADMGLI	VQD	Avg.
FC	52.9	52.4	58.1	55.1	55.2	50.5	51.7	50.8
NBC	100	93.1	100	100	100	99.7	100	99.1
NGC	94.9	92.3	100	100	100	100	100	96.8

score for subsequent processes based on the assumption that higher score indicates greater high-frequency variations for real images or more obvious artifacts for generated images. To validate this, we modify the selection method by using the maximum score, minimum score, and random selection. As shown in Table 5, the selection based on the minimal score gets the lowest accuracy, while the selection based on the maximum score gets the highest accuracy. Consequently, we choose the patch with the highest score to exploit the intrinsic patterns of noisy images.

Impact of Classifier: We compare our Noise-Based Classifier (NBC) and Noise-Guided Classifier (NGC) with a simple Fully Connected (FC) layer, where the error map is directly sent to a FC layer. As shown in Table 6, FC itself cannot effectively extract useful information from low-bit images, achieving an average ACC of only 52.6%. In contrast, the NBC and NGC classifiers fully leverage the noise patterns in low-bit images to distinguish AI-generated images from real ones, attaining average ACCs of 99.1% and 96.8%, respectively.

Impact of the Number of Bit-Planes: We choose the three least bit-planes to generate the noise image based on the

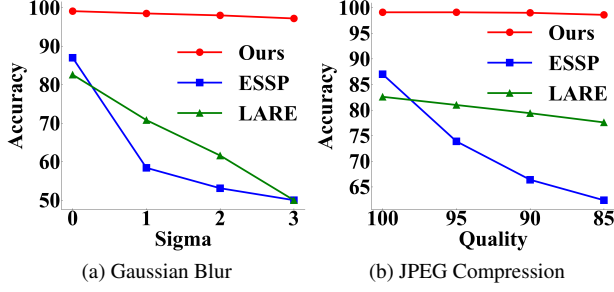


Figure 6. **Robustness to Image Degradation.** Gaussian blur with $\sigma = 0,1,2,3$ and JPEG compression ($quality = 100\%,95\%,90\%,85\%$) are applied to LaRE² [27], ESSP [7] and our LOTA, the results demonstrate the robustness of ours to unseen perturbations.

Table 7. **Impact of Bit-Planes.** We generate the noise image by combining different numbers of bit-planes, ranging from 0 to 5.

Bit-Planes	Big	Mid	Wuk	SD4	SD5	ADM	GLI	VQD	Avg.
0	78.8	88.3	100	100	100	99.8	99.4	100	91.5
0~1	93.6	77.0	100	100	100	99.9	99.9	100	95.1
0~2	94.9	92.3	100	100	100	100	100	100	96.8
0~3	85.5	97.0	99.9	100	99.9	100	100	100	95.7
0~4	89.2	96.3	99.9	100	99.9	99.9	100	100	93.4
0~5	91.4	96.1	98.8	99.3	99.1	85.6	82.6	79.0	86.4

assumption that lower-order bit-planes preserve more noise patterns, which are the critical indicator for distinguishing AI-generated images from real ones. To validate this point, we choose different numbers of bits to generate an image: from 0-bit to 5-bit. As shown in Table 7, as the number of combined planes increases, the average accuracy firstly increases to 96.8% when the three lowest bit-planes are composed, and then decreases. Fluctuation is especially obvious when testing on the subsets of BigGAN and Midjourney. This suggests that lower bit-planes contain less excessive high-frequency variation, making it difficult to detect key features, while higher bit-planes introduce some visual features that obscure certain noise patterns, thereby impairing the cross-generator generalization capability.

Analysis of Computational Efficiency: For time consumption of error extraction and deepfake image classification, DIRE [38] uses 20 steps to build an error map, totally consuming 2 seconds per image, and LaRE² [27] uses 1 step, totally consuming 0.26 seconds per image. Even the existing fastest ESSP [7] consumes 31.99 milliseconds to process an image. As shown in Table 8, leveraging bit-plane based operations, LOTA uses only one step to generate an error image, which operates at the millisecond level (1.52 milliseconds for error extraction and 4.00 milliseconds in total), demonstrating nearly a hundred times faster than existing methods in error extraction. Regarding the number of parameters, several mainstream methods rely on large pre-

Table 8. **Comparison in Computation Efficiency.** We compare our methods with different classifiers with other mainstream methods based on error extraction, considering two dimensions: 1) Time consumption for error extraction and classification—our method operates at the millisecond level, nearly a hundred times faster than other methods. 2) Model parameters—our method is efficient and lightweight, requiring significantly fewer parameters than methods that rely on large pre-trained models, which introduce substantial computational overhead.

Method	Time		Params	
	Error Extraction	Total	Error Extraction	Total
DIRE [38]	1.99 s	2 s	644.8M	688.3M
LaRE ² [27]	250 ms	260 ms	1066.2M	1165.8M
ESSP [7]	25.10 ms	31.99 ms	7.1M	30.7M
LOTA-NBC	1.52 ms	4.00 ms	0	23.6M
LOTA-NGC	1.52 ms	4.71 ms	0	28.4M

trained models (e.g., Diffusion), resulting in models with abundant parameters. Our model has 23.6M parameters for deepfake image detection, a breakthrough that significantly enhances its practical deployment potential.

Robustness to Image Degradation: To further demonstrate the robustness to unseen perturbations and degradations, we train our model and comparison models on Stable Diffusion V1.5 [32], and test on 8 subsets which are processed by Gaussian blur and JPEG compression to different extents. Following [37] and [38], we choose Gaussian blur ($\sigma = 0,1,2,3$) and JPEG compression ($quality = 100\%,95\%,90\%,85\%$) when testing, the results are shown in Figure 6. We observe that when the image degrades, existing methods, like LaRE² [27] and ESSP [7], encounter a huge decline. Especially when the σ of the Gaussian blur is set to 2 and 3, the two models nearly degenerate into random guess classifiers. In contrast, our method is very stable against these interference, demonstrating the strong robustness to unseen perturbations and degradations.

5. Conclusion

In this paper, we propose an effective and efficient AI-generated image detection method called LOTA. LOTA innovatively leverages bit-planes for noisy image generation and fake detection. It comprises three decoupled and concise modules: Bit-plane Guided Noisy Image Generation (BGNIG), Maximum Gradient Patch Selection (MGPS) and classification head. Extensive experiments on the GenImage benchmark demonstrate the outstanding performance and strong cross-generator generalization capability of our method. Our approach is highly robust to unseen perturbations and degradations. It contains only 23.6 million parameters and operates at the millisecond level, making it nearly a hundred times faster than existing methods.

Acknowledgments

This work was supported by National Science Foundation of China (62172090, 62302093, 62176163), Jiangsu Province Natural Science Fund (BK20230833), Start-up Research Fund of Southeast University under Grant RF1028623097, Shenzhen Higher Education Stable Support Program General Project (Grant 20231120175215001), Scientific Foundation for Youth Scholars of Shenzhen University, and Big Data Computing Center of Southeast University.

References

- [1] Midjourney. <https://www.midjourney.com/home/>, 2022. 5
- [2] Wukong. <https://xihe.mindspore.cn/modelzoo/wukong>, 2022. 5
- [3] Andrew Brock et al. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2018. 5
- [4] Biwei Cao, Qihang Wu, Jiuxin Cao, Bo Liu, and Jie Gui. External reliable information-enhanced multimodal contrastive learning for fake news detection. In *AAAI*, pages 31–39, 2025. 1
- [5] Keshigeyan Chandrasegaran, Ngoc-Trung Tran, and Ngai-Man Cheung. A closer look at fourier spectrum discrepancies for cnn-generated images detection. In *CVPR*, pages 7200–7209, 2021. 3
- [6] Baoying Chen, Jishen Zeng, Jianquan Yang, and Rui Yang. Drcr: Diffusion reconstruction contrastive training towards universal detection of diffusion generated images. In *ICML*, 2024. 2
- [7] Jiaxuan Chen, Jieteng Yao, and Li Niu. A single simple patch is all you need for ai-generated image detection. *arXiv preprint arXiv:2402.01123*, 2024. 1, 2, 3, 4, 5, 6, 7, 8
- [8] Riccardo Corvi, Davide Cozzolino, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. Intriguing properties of synthetic images: from generative adversarial networks to diffusion models. In *CVPR*, pages 973–982, 2023. 3
- [9] Riccardo Corvi, Davide Cozzolino, Giada Zingarini, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. On the detection of synthetic images generated by diffusion models. In *ICASSP*, pages 1–5. IEEE, 2023. 1, 3
- [10] Davide Cozzolino, Giovanni Poggi, Matthias Nießner, and Luisa Verdoliva. Zero-shot detection of ai-generated images. In *ECCV*, pages 54–72. Springer, 2024. 1, 2, 3, 4
- [11] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 2021. 5
- [12] Chandler Timm Doloriel and Ngai-Man Cheung. Frequency masking for universal deepfake detection. In *ICASSP*, pages 13466–13470. IEEE, 2024. 3
- [13] Tarik Dzanic, Karan Shah, and Freddie Witherden. Fourier spectrum discrepancies in deep network generated images. In *NeurIPS*, pages 3022–3032, 2020. 3
- [14] Omar Elharrouss, Noor Almaadeed, and Somaya Al-Maadeed. An image steganography approach based on k-least significant bits (k-lsb). In *IEEE international conference on informatics, IoT, and enabling technologies (ICIOT)*, pages 131–135. IEEE, 2020. 2
- [15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 1
- [16] 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 *CVPR*, pages 10696–10706, 2022. 5
- [17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6
- [18] Neil F Johnson and Sushil Jajodia. Exploring steganography: Seeing the unseen. *Computer*, 31(2):26–34, 1998. 1
- [19] Yan Ju, Shan Jia, Jialing Cai, Haiying Guan, and Siwei Lyu. Glff: Global and local feature fusion for ai-synthesized image detection. *IEEE Transactions on Multimedia*, 26:4073–4085, 2023. 2
- [20] Felix Juefei-Xu, Run Wang, Yihao Huang, Qing Guo, Lei Ma, and Yang Liu. Countering malicious deepfakes: Survey, battleground, and horizon. *IJCV*, 2022. 1
- [21] Eiji Kawaguchi and Richard O Eason. Principles and applications of bpcs steganography. In *Multimedia systems and applications*, pages 464–473. SPIE, 1999. 2
- [22] Romeo Lanzino, Federico Fontana, Anxhelo Diko, Marco Raoul Marini, and Luigi Cinque. Faster than lies: Real-time deepfake detection using binary neural networks. In *CVPR*, pages 3771–3780, 2024. 3
- [23] Giorgio Leporoni, Luca Maiano, Lorenzo Papa, and Irene Amerini. A guided-based approach for deepfake detection: Rgb-depth integration via features fusion. *Pattern Recognition Letters*, 181:99–105, 2024. 3
- [24] Bo Liu, Fan Yang, Xiuli Bi, Bin Xiao, Weisheng Li, and Xinbo Gao. Detecting generated images by real images. In *ECCV*, 2022. 2
- [25] Zhengzhe Liu, Xiaojuan Qi, and Philip HS Torr. Global texture enhancement for fake face detection in the wild. In *CVPR*, pages 8060–8069, 2020. 6
- [26] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 6
- [27] Yunpeng Luo, Junlong Du, Ke Yan, and Shouhong Ding. Lare²: Latent reconstruction error based method for diffusion-generated image detection. In *CVPR*, pages 17006–17015, 2024. 1, 3, 5, 6, 7, 8
- [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, 3, 4
- [29] 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. *arXiv preprint arXiv:2112.10741*, 2021. 5

- [30] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *ECCV*, pages 86–103. Springer, 2020. [6](#)
- [31] Jonas Ricker, Simon Damm, Thorsten Holz, and Asja Fischer. Towards the detection of diffusion model deepfakes. *arXiv preprint arXiv:2210.14571*, 2022. [1](#)
- [32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. [1](#), [2](#), [5](#), [6](#), [8](#)
- [33] Ayush Sarkar, Hanlin Mai, Amitabh Mahapatra, Svetlana Lazebnik, David A Forsyth, and Anand Bhattad. Shadows don’t lie and lines can’t bend! generative models don’t know projective geometry... for now. In *CVPR*, pages 28140–28149, 2024. [2](#)
- [34] Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, and Yunchao Wei. Learning on gradients: Generalized artifacts representation for gan-generated images detection. In *CVPR*, 2023. [2](#)
- [35] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. [6](#)
- [36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. [5](#)
- [37] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *CVPR*, pages 8695–8704, 2020. [2](#), [6](#), [8](#)
- [38] Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, Hezhen Hu, Hong Chen, and Houqiang Li. Dire for diffusion-generated image detection. In *CVPR*, pages 22445–22455, 2023. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#)
- [39] Yang Yu, Rongrong Ni, Wenjie Li, and Yao Zhao. Detection of ai-manipulated fake faces via mining generalized features. *ACM TOMM*, 18(4):1–23, 2022. [3](#)
- [40] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. Detecting and simulating artifacts in gan fake images. In *WIFS*, pages 1–6. IEEE, 2019. [3](#), [6](#)
- [41] Nan Zhong, Yiran Xu, Zhenxing Qian, and Xinpeng Zhang. Patchcraft: Exploring texture patch for efficient ai-generated image detection. *arXiv preprint arXiv:2311.12397*, 2023. [4](#)
- [42] 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. *NeurIPS*, 2023. [2](#), [5](#)