LOCAL BINARY PATTERN IS A GOOD DESCRIPTOR FOR AI-GENERATED IMAGE DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Advancements in image generation technologies have raised significant concerns about their potential misuse, such as producing misinformation and deepfakes. Therefore, there is an urgent need for effective methods to detect AI-generated images. Recent studies reveal that texture is an important cue for fake image detection since the generator typically correlates the values of nearby pixels and cannot generate as strong texture contrast as real data. However, these methods compute the image texture only by simply summing the residual of four directions. In this paper, we find that Local Binary Pattern (LBP), which is a good texture descriptor, can capture intrinsic forgery clues. To this end, we proposed an **LBP**-based AI-generated image **D**etector, named as $\mathbf{LBPD_{based}}$. To fully leverage the LBP descriptor of images, we also proposed an LBP-guided AI-generated image detector $\mathbf{LBPD_{guided}}$, which combines LBP descriptor with image features by the proposed global refinement module from both spatial and channel perspectives. Extensive experiments validate the effectiveness of the proposed method. It also achieves state-of-the-art results across various evaluation benchmarks.

1 INTRODUCTION

The rapid development of AI generative technologies has significantly lowered the barrier for creating highly realistic fake images. As deep generative models advance and mature Goodfellow et al. (2020); Ho et al. (2020); Rombach et al. (2022); Yan et al. (2025b), the proliferation of AI-generated images has drawn considerable attention, driven by their ability to produce high-quality content with relative ease. However, these advancements also introduce significant risks. AI-generated fake images can be misused to misguide public opinion, fabricate evidence in forensic science, undermine trust in digital media, and more. Therefore, detecting such AI-generated images has become a critical and urgent topic in both academia and industry.

Early deep learning-based deepfake methods focus on detection of GAN-generated images. However, recent works Corvi et al. (2023); Ricker et al. (2022) 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 based on reconstructed image errors are proposed, such as DIRE Wang et al. (2023), AEROBLADE Ricker et al. (2024), LaRE^2 Luo et al. (2024), ESSP Chen et al. (2024c), and FIRE Chu et al. (2025). DIRE Wang et al. (2023) firstly proposes to compute the error between the raw and diffusion reconstructed images, and considers the image with smaller error to be AI-generated. LaRE^2 Luo et al. (2024) speeds up the diffusion reconstruction process by one step of the diffusion reverse process, and also proposes to use reconstruction errors to refine the image feature. FIRE Chu et al. (2025) integrates frequency decomposition into reconstruction-based detection methods, identifying the image components that contribute to higher reconstruction errors. GenSign Fei et al. (2026) ultilizes the diverse reconstruction residuals serve as supervision to learn generative signatures. However, these approaches extract error signals through a diffusion reverse process, which is not only inefficient but also susceptible to the reconstruction diffusion model.

Recently, Liu et al. (2020) found that texture is an important cue for fake image detection since the generator typically correlates the values of nearby pixels and cannot generate as strong texture contrast as real data. Subsequently, PatchCraft Zhong et al. (2023) and ESSP Chen et al. (2024c) compute the image texture by summing the residual of diagonal, counter-diagonal, horizontal, and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

vertical directions, in order to detect fake images. However, these methods compute the image texture only by simply summing the residual of four directions. In this paper, we find that the Local Binary Patterns (LBP) operator has the potential to address AI-generated image detection. The LBP operator Ojala et al. (1996) is one of the best performing texture descriptors that has been widely used in various applications, such as texture classification Ojala et al. (2002) and face recognition Ahonen et al. (2004; 2006). It has proven to be highly discriminative, and its computational efficiency, make it suitable for demanding image analysis tasks.

We visualize and compare the LBP for both real and AI-generated RGB images in Figure 1. It shows that the LBP of real images and AI-generated images have significantly different distributions. Compared to the LBP of real images, the LBP of AI-generated images has more consistent patterns. Besides, the LBP of real images contains more complicated texture components than the LBP of AI-generated images. For example, the body of the insect from the AI-generated image (The 3rd row in Figure 1) contains less complicated texture than the body of the insect from the real image (The 2nd row in Figure 1).

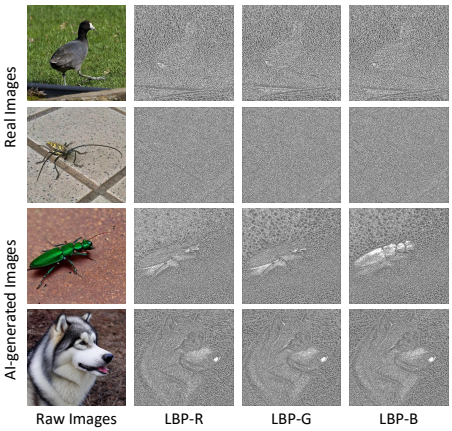

Figure 1: **Comparison of the Local Binary Patterns (LBP) between real images and AI-generated images.** The LBP-R, LBP-G, and LBP-B denote the LBP of R-channel, G-channel, and B-channel of raw images, respectively. We find that the LBP of real images and AI-generated images have significantly different distributions that can be used for distinguishing between them. **Please zoom in for a better view.**

To this end, we proposed the **LBP**-based AI-generated image **D**etector, named as $\textbf{LBPD}_{\textbf{based}}$, which directly takes the LBP images as the model input to distinguish between AI-generated images and real images. We in-depth analyze the rationale of the effectiveness of the LBP image through the histogram. To fully leverage the LBP descriptor of images, we also proposed an LBP-guided AI-generated image detector $\textbf{LBPD}_{\textbf{guided}}$, which combines the LBP features with image features by the proposed Global Refinement Module (GRM). The GRM focuses on the feature interaction for the class token of the LBP feature and the class token of the RGB feature from both spatial and channel perspectives.

Our contributions in this paper are summarized as follows:

- We find that the Local Binary Pattern (LBP) of images shows significantly different distributions between real and AI-generated images, which can be used to distinguish between them, and we in-depth analyze the rationale of the effectiveness of the LBP operator on the AI-generated image detection task.
- We propose a simple yet effective LBP-based AI-generated image detector $\textbf{LBPD}_{\textbf{based}}$. To fully leverage the LBP descriptor of images, we also proposed an LBP-guided AI-generated image detector $\textbf{LBPD}_{\textbf{guided}}$, which combines the LBP features with image features by the proposed Global Refinement Module (GRM).
- Extensive experiments on four benchmarks, GenImage Zhu et al. (2023), DRCT-2M Chen et al. (2024b), Co-Spy-Bench Cheng et al. (2025), and ForenSynths Wang et al. (2020), verify the effectiveness of the proposed method, and show that our method achieves state-of-the-art results without bells and whistles on the AI-generated image detection task.

2 Related Work

2.1 Spatial Domain-Based Methods

Most detection methods are based on the spatial domain. Recently, Tan et al. Tan et al. (2024b) proposed utilizing neighboring pixel relationships (NPR), computed through local pixel differences, as a simple yet effective cue for AI-generated detection. Zheng et al. Zheng et al. (2024) utilized

image patch shuffling to disrupt the semantic artifacts to decrease negative impact of image semantic information. Zhong et al. (2023); Chen et al. (2024c) utilize the texture information by simply summing the residual of four directions, including diagonal, counter-diagonal, horizontal, and vertical directions to detect AI-generated images. In this paper, we find that Local Binary Patterns, which is the one of the best performing texture descriptors, is capable to detect AI-generated images.

2.2 Frequency Domain-Based Methods

To tackle the subtlety of spatial artifacts in AI-generated images, frequency-domain methods analyze image frequency components, enabling more effective real vs. fake differentiation. Qian et al. Qian et al. (2020a) proposed a face forgery detection network based on frequency-aware decomposed image components and local frequency statistics. Luo et al. Luo et al. (2021) proposed a feature representation based on high-frequency noise at multiple scales and enhanced detection performance by integrating it with an attention module. Liu et al. Liu et al. (2022) utilized noise patterns in the frequency domain as feature for detecting AI-generated images. Tan et al. Tan et al. (2024a) proposed FreqNet to exploit high-frequency representation across spatial and channel dimension, toward detection generalizability.

2.3 Image Error-Based Methods

Recently, many works focus on reconstructed image errors. Wang et al. Wang et al. (2023) propose Diffusion Reconstruction Error (DIRE), which differentiates real from DM-generated images by measuring reconstruction error. Ma et al. Ma et al. (2023) enhance detection accuracy using multi-step error computation in their SeDID method. FakeInversion Cazenavette et al. (2024) detects images generated by unseen text-to-image DMs using text-conditioned inversion. AEROBLADE Ricker et al. (2024) and RDD Choi et al. (2026) utilize autoencoder reconstruction error from latent DMs for a simple, training-free approach. Different from the above approaches, we attempt to dig the forgery signal contained within the image itself.

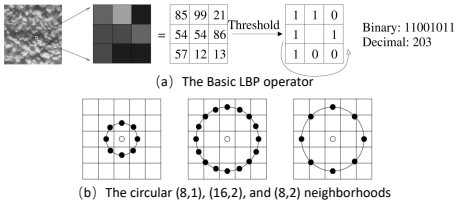

Figure 2: (a) The Basic LBP operator. (b) The circular (8,1), (16,2), and (8,2) neighborhoods Ahonen et al. (2006).

3 Method

3.1 Local Binary Patterns

The LBP operator was originally designed for texture description. The operator assigns a label to every pixel of an image by thresholding the 3×3-neighborhood of each pixel with the center pixel value and considering the result as a binary number. See Figure 2(a) for an illustration of the basic LBP operator.

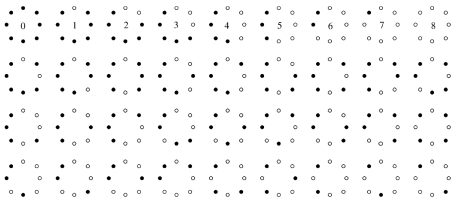

Figure 3: The 36 unique rotation invariant binary patterns that can occur in the circularly symmetric neighbor set of $LBP^{ri}_{8,R}$ Ojala et al. (2002). Please zoom in for a better view.

Default LBP. To be able to deal with textures at different scales, the LBP operator was later extended to use neighborhoods of different sizes Ojala et al. (2002). Defining the local neighborhood as a set of sampling points evenly spaced on a circle centered at the pixel to be labeled allows any radius and number of sampling points. Bilinear interpolation is used when a sampling point does not fall in the center of a pixel. In the following, the notation $LBP_{P,R}$ will be used for pixel neighborhoods which means P sampling points on a circle of radius R. See Figure 2(b) for an example of circular neighborhoods.

Rotation Invariant LBP. When the image is rotated, the value of $LBP_{P,R}$ will correspondingly change. To remove the effect of rotation, i.e., to assign a unique identifier to each rotation invariant

local binary pattern we define:

$$LBP^{ri}_{P,R} = min\{ROR(LBP_{P,R}, i) \mid i = 0, 1, ..., P-1\} \quad (1)$$

where $ROR(x, i)$ performs a circular bit-wise right shift on the P-bit number x for i times. $LBP^{ri}_{P,R}$ quantifies the occurrence statistics of individual rotation invariant patterns corresponding to certain microfeatures in the image. Figure 3 illustrates the 36 unique rotation invariant local binary patterns that can occur in the case of $P = 8$, i.e., $LBP^{ri}_{8,R}$ can have 36 different values.

Uniform LBP. Another extension to the original operator uses so called *uniform patterns* Ojala et al. (2002). A Local Binary Pattern is called uniform if it contains at most two bitwise transitions from 0 to 1 or vice versa when the binary string is considered circular. For example, 00000000, 00011110, and 10000011 are uniform patterns. Ojala et al. (2002) found that *uniform patterns* is fundamental property of texture, providing the vast majority, sometimes over 85 percent, of all $LBP^{ri}_{8,R}$ patterns present in the observed textures. In the first row of Figure 3, the labels of the *uniform patterns* are denoted inside the patterns, while the "nonuniform" patterns (the second, third, and fourth rows of Figure 3) are grouped under another "miscellaneous" label (i.e., label 9). Therefore, there are total of 10 patterns in $LBP^{ri,u2}_{8,R}$ where $u2$ stands for using only uniform patterns and labelling all remaining patterns with a single label.

Rotation Variant Uniform LBP. If we consider the rotation variant, we can get the rotation variant Uniform LBP $LBP^{u2}_{8,R}$. There are a total of $1 + 7 * 8 + 1 + 1 = 59$ patterns in $LBP^{u2}_{8,R}$. Label 0 and label 8 of the *uniform patterns* in Figure 3 remain unchanged, and will be mapped to label 0 and label 57 under the rotation variant setting, respectively. Label 1 to label 7 of the *uniform patterns* in Figure 3 will generate $7 * 8$ patterns under the rotation variant setting. The "nonuniform" patterns are also grouped under another "miscellaneous" label, named as label 58.

In conclusion, taking $P = 8$ as an exmaple, there are four kinds of LBP, named as Default $LBP_{8,R}$, Rotation invariant $LBP^{ri}_{8,R}$, Uniform $LBP^{ri,u2}_{8,R}$, and rotation variant uniform $LBP^{u2}_{8,R}$. In the following section, we take $LBP^{u2}_{8,R}$ with $R = 1$ by default setting.

3.2 Local Binary Patterns Based Detector

Why LBP is capable to address the AI-generated image detection task. From the spatial perspective, real images are produced by physical optical processes, and the relationships between pixels follow complex physical statistical laws. However, AI-generated images (such as those from GANs and diffusion models) are attempting to learn this distribution but have deficiencies in local microstructures. For example, generative models tend to smooth out minor, local mutations that do not conform to the statistics of the training data. While in real images, the surfaces of objects (such as skin and fabric) have random, irregular but physically plausible micro-textures. This is also evidenced in Liu et al. (2020) where generator typically correlates the values of nearby pixels and cannot generate as strong texture contrast as real data. On the other hand, the common up-sampling operations in popular generators can produce generalized forgery artifacts Tan et al. (2024b), such as checkerboard effect and unnatural local arrangement patterns. LBP divides the image into local neighborhoods (for example, 3x3 pixel blocks). It compares the value of the central pixel with those of the 8 neighboring pixels around it. LBP does not concern the absolute brightness (color) of the pixels, but only the local pixel relationship. It records whether this pixel point is protruding or recessed compared to its neighboring pixels, as well as the spatial arrangement of this relationship. In conclusion, LBP focuses on the local microstructures of images, where AI-generated images just right have deficiencies, as described above. Therefore, LBP is capable to address the AI-generated image detection task.

From the frequency perspective, according to the spectral bias theory Rahaman et al. (2019), deep neural networks (including the generative models) have the characteristic of favoring the learning of low-frequency components (that is, the so-called "spectral bias" or "frequency gap"). While for the real images, although the high-frequency components of real images have relatively small amplitudes, they contain rich information and have complex structures (such as the edges of objects, and the abrupt changes in texture details). Therefore, the distribution of high-frequency energy of generated images is different from that of the real images. LBP is essentially a high-pass filter (difference operator) combined with local binarization. It ignores low-frequency brightness variations and focuses on high-frequency local symbol changes. Therefore, when AI-generated images have

missing or statistically abnormal high-frequency parts in the frequency domain, the LBP feature will significantly amplify this difference.

LBP patterns between real and AI-generated images of the same content. In Figure 1, we have already shown the difference between real and AI-generated images. Here we compare the LBP patterns between real and AI-generated images of the same content from both spatial and frequency perspectives. We visualize the LBP maps for real–fake image pair, along with their differences in the pixel, LBP, and Fourier domains. Here, "fake" refers to AI-reconstructed images resembling the originals in order to keep the main content of images unchanged. As shown in Figure 4, the LBP differences (3-nd row, 2-rd col) are far more pronounced than pixel-level differences (3-st row, 1-rd col). In the frequency domain, compare with the real–fake differences of the pixel space (3-rd row, 3-rd col), the real-fake differences (3-th row, 4-rd col) in the LBP space focus more on the mid-frequency and high-frequency (i.e., the mid and high-frequency occupies more energy than low-frequency), demonstrating that current generative models focus on optimizing high-level semantic content (principal components), often overlooking low-level signals (such as complex texture components). These results show LBP's ability to amplify real–fake discrepancies.

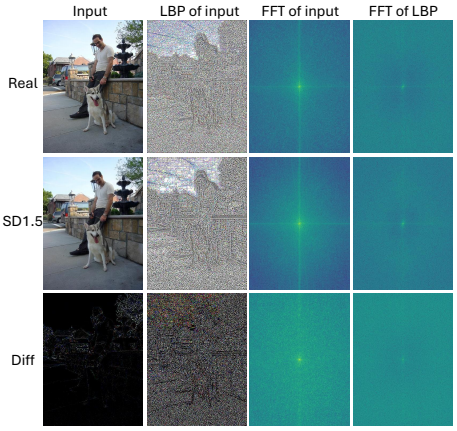

Figure 4: Visualization of LBP for real–fake image pair, along with their differences in the pixel, LBP, and Fourier domains. **Please zoom in for a better view.**

Histogram of Local Binary Patterns. We have already show the difference between real and AI-generatd images by qualitative visualization. We further give a quantitative analysis between the LBP of real and AI-generated images. To be specific, we firstly randomly sample 10000 AI-generated images from the train set of Glide, ADM, MidJourney, and BigGAN, respectively, on GenImage Zhu et al. (2023), and a total of 10000 real images from the same train sets on GenImage. Then, we plot the histogram of $LBP^{u2}_{8,1}$ in Figure 5. It shows that there are significantly different distributions of LBP between real and AI-generated images. For example, the frequency of $LBP^{u2}_{8,1} = 58$ between real and AI-generated images are significantly different in Figure 5 (a), and the frequency of $LBP^{u2}_{8,1} = 26, 27, ..., 33$ between real and AI-generated images are different in Figure 5 (b). Specifically, all $LBP^{u2}_{8,1} = 58$ in AI-generated images from Glide, ADM, MidJourney, and BigGAN are less than 0.10, while the $LBP^{u2}_{8,1} = 58$ in real images are more than 0.15. Note that we map all "nonuniform" patterns to 58 during plotting the histogram. This shows that AI-generated images have less complicated texture components than real images, since "nonuniform" patterns are more tangled as shown in Figure 3 (The second, third, and fourth rows are "nonuniform" patterns, while the first row denotes uniform patterns). Note that the histogram merely represents the frequency of LBP values. It merely indicates that there is a difference in the distribution of LBP values between the real and fake images. We did not directly use the histogram as feature to detect fake images.

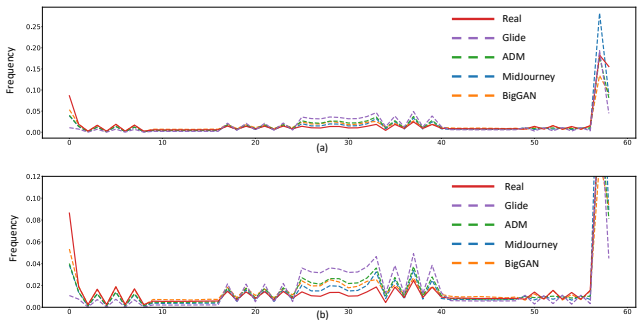

Figure 5: (a) denotes the histogram of LBP between real and AI-generated images. (b) denotes that we limit the maximum value of y-axis to 0.1 for visual convenience. The red solid line denotes real images, while the dash lines with other colors denote fake images. **Please zoom in for a better view.**

Local Binary Patterns Based Detector. Since the LBP of images shows significantly different distributions between real and AI-generated images, we propose a simple yet effective **LBP**-based

AI-generated image **D**etector, named as **LBPD$_{\text{based}}$**. As shown in Figure 6, LBPD$_{\text{based}}$ directly takes the $LBP^{u2}_{8,1}$ as the input of the image encoder (such as ResNet-34 He et al. (2016)), and outputs the probability of AI-generated images with a 2-dimensional fc layer appended at the last of the image encoder. LBPD$_{\text{based}}$ is supervised with cross-entropy loss L_{cls} during training. In experiments, we show that LBPD$_{\text{based}}$ almost achieves state-of-the-art results without bells and whistles.

3.3 Local Binary Patterns Guided Detector

3.3.1 Framework Overview

To fully leverage the LBP descriptor of images, we also proposed an **LBP**-guided AI-generated image **D**etector, named as **LBPD$_{\text{guided}}$**, which combines the LBP features with image features by the proposed Global Refinement Module (GRM).



Figure 6: Overview of the LBPD$_{\text{based}}$

As shown in Figure 7, we compute the $LBP^{u2}_{8,1}$ of input image and utilize the LBP encoder to get the LBP feature F_{LBP}. The RGB image encoder extracts RGB features F_{RGB} by taking the RGB images as input. Then, we perform global average pooling on the LBP feature F_{LBP} and RGB feature F_{RGB} to get the global LBP feature F^p_{LBP} and global RGB feature F^p_{RGB}, respectively. Finally, The GRM takes the features F_{LBP}, F^p_{LBP}, F_{RGB}, F^p_{RGB} as input to integrate features globally. The output of GRM is used to predict the probability of AI-generated images, with standard cross-entropy loss L_{cls} to be optimized.

3.3.2 Global Refinement Module

The proposed Global Refinement Module (GRM) refines features globally from both spatial and channel perspectives, as shown in Figure 8.

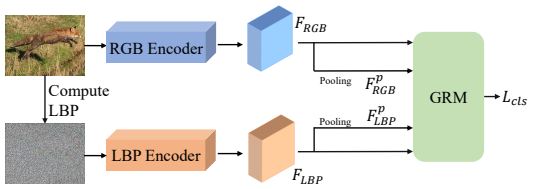

Figure 7: Overview of the proposed LBPD$_{\text{guided}}$

Spatial Refinement (SR). To refine the feature from the spatial perspective, we choose the Spatial Attention Mechanism (SAM):

$$\text{SAM}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V. \quad (2)$$

Where Q, K, V denote the query, key, and value, respectively. d_k is the channel dimension. To further enhance the feature representation, we adopt the multi-head mechanism from TransformerVaswani et al. (2017). Multi-Head Spatial Attention Module (MHSAM) is defined as:

$$\begin{aligned} \text{MHSAM}(Q, K, V) &= \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O, \\ \text{where head}_i &= \text{SAM}(QW^Q_i, KW^K_i, VW^V_i). \end{aligned} \quad (3)$$

where h is the number of heads, W^Q_i, W^K_i, W^V_i are learnable projection matrices of i-th head. W^O is the learnable projection layer of the output of MHSAM. The spatial refined feature $\bar{F}^P_{RGB}$ and $\bar{F}^P_{LBP}$ is calculated by:

$$\bar{F}^P_{RGB} = F^P_{RGB} + \text{MHSAM}(F^P_{RGB}, F_{LBP}, F_{LBP}), \quad (4)$$

$$\bar{F}^P_{LBP} = F^P_{LBP} + \text{MHSAM}(F^P_{LBP}, F_{RGB}, F_{RGB}), \quad (5)$$

where the global LBP feature F^p_{LBP} and the global RGB feature F^p_{RGB} integrates the important information from the RGB feature F_{RGB} and the LBP feature F_{LBP}, respectively.

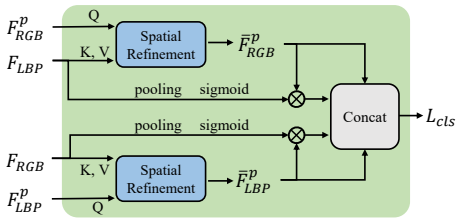

Figure 8: The proposed Global Refinement Module (GRM).

Method	Midjourney	SDv1.4	SDv1.5	ADM	GLIDE	Wukong	VQDM	BigGAN	Avg.
GramNet Liu et al. (2020)	73.6	98.8	98.7	51.5	55.3	95.3	55.1	49.4	72.2
UnivFD Ojha et al. (2023)	91.4	96.4	96.1	58.0	73.4	94.5	67.8	57.7	79.4
DIRE Wang et al. (2023)	50.4	99.9	99.9	52.3	67.2	**99.9**	50.1	49.9	71.2
LaRE^2 Luo et al. (2024)	74.0	99.9	99.9	61.7	88.5	99.9	97.2	68.7	86.2
DRCT Chen et al. (2024b)	91.5	95.0	94.4	79.4	89.1	94.6	90.0	81.6	89.4
ESSP Chen et al. (2024c)	82.6	99.2	99.3	78.9	88.9	98.6	96.0	73.9	90.6
Effort Yan et al. (2025a)	82.4	99.8	99.8	78.7	93.3	97.4	91.7	77.6	91.1
DLFE Zhong et al. (2025)	95.5	96.1	96.3	95.9	96.0	96.4	96.0	95.7	96.0
GenSign Fei et al. (2026)	**99.3**	99.8	99.7	84.3	90.4	99.8	99.5	99.8	96.6
IAPL Li et al. (2026)	95.1	99.9	99.8	85.4	95.9	99.8	98.8	98.7	96.7
LBPD$_{\text{based}}$	93.4	99.5	99.6	93.4	98.3	97.1	94.4	86.2	**95.2**
LBPD$_{\text{guided}}$	95.0	**99.9**	**99.9**	**96.4**	**99.6**	99.6	**99.5**	**99.8**	**98.7**

Table 1: Results on GenImage Zhu et al. (2023). Methods are trained on GenImage/SDv1.4 following Zhu et al. (2023). The metric is Averaged Accuracy with 0.5 as the threshold.

Methods	CNNSpot	GramNet	DeFake	UnivFD	DIRE	DRCT	DLFE	GenSign	IAPL	**LBPD$_{\text{based}}$**	**LBPD$_{\text{guided}}$**
LDM	99.9	99.4	92.1	98.3	98.2	96.7	97.1	97.4	97.7	98.9	99.7
SDv1.4	99.9	99.0	99.5	96.2	99.9	96.3	97.1	99.0	99.3	98.5	99.6
SDv1.5	99.9	98.8	99.5	96.3	99.9	96.3	97.1	97.0	98.0	98.4	99.5
SDv2	97.5	95.3	89.7	93.8	68.2	94.9	97.1	96.6	96.4	98.8	99.4
SDXL	66.3	62.6	64.0	91.0	53.8	96.2	97.1	97.0	97.5	98.6	99.3
SDXL-Refiner	86.5	80.7	69.2	93.9	71.9	93.5	97.1	96.7	97.0	98.8	99.6
SD-Turbo	86.2	71.2	92.0	86.4	58.9	93.4	97.1	96.5	96.0	98.6	98.7
SDXL-Turbo	72.4	69.3	93.9	85.9	54.3	92.9	97.1	96.2	95.5	98.7	98.6
LCM-SDv1.5	98.3	93.1	99.1	90.4	99.8	91.2	97.1	94.1	93.8	97.8	99.3
LCM-SDXL	61.7	57.0	70.9	89.0	59.7	95.0	97.1	96.2	95.2	97.0	99.1
SDv1-Ctrl	98.0	90.0	59.0	90.4	99.6	95.6	97.1	95.9	96.6	97.5	99.5
SDv2-Ctrl	85.9	75.6	62.3	81.1	64.2	92.7	96.8	96.2	95.7	98.2	99.2
SDXL-Ctrl	82.8	82.7	66.7	89.1	59.1	92.0	94.7	95.0	95.5	97.6	98.7
SDv1-DR	60.9	51.2	50.1	52.0	52.0	94.1	92.3	80.3	96.0	78.3	85.9
SDv2-DR	51.4	50.0	50.2	51.0	50.0	69.6	51.8	65.0	70.2	65.7	72.3
SDXL-DR	50.3	50.1	50.0	50.5	50.0	57.4	49.7	64.9	59.4	66.2	72.9
Avg.	81.1	76.6	75.5	83.5	71.2	90.5	90.9	91.5	92.5	**93.0**	**95.1**

Table 2: Results on DRCT-2M Chen et al. (2024b). Methods are trained on DRCT-2M/SDv1.4 following DRCT Chen et al. (2024b). The metric is Averaged Accuracy with 0.5 as the threshold. The compared methods are CNNSpot Wang et al. (2020), GramNet Liu et al. (2020), DeFake Sha et al. (2023), UnivFD Ojha et al. (2023), DIRE Wang et al. (2023), DRCT Chen et al. (2024b), DLFE Zhong et al. (2025), GenSign Fei et al. (2026), and IAPL Li et al. (2026).

Channel Refinement (CR). To refine the feature from the channel perspective, we firstly perform global averaged pooling to both F_{RGB} and F_{LBP}. Then our channel refinement is achieved by a gate mechanism:

$$\bar{F}^{P,C}_{RGB} = \text{sigmoid}\left(\text{pooling}\left(F_{LBP}\right) W_{LBP}\right) \odot \bar{F}^{P}_{RGB}, \tag{6}$$

$$\bar{F}^{P,C}_{LBP} = \text{sigmoid}\left(\text{pooling}\left(F_{RGB}\right) W_{RGB}\right) \odot \bar{F}^{P}_{LBP}, \tag{7}$$

where W_{LBP} and W_{RGB} are learnable projection layers.

Finally, $\bar{F}^{P,C}_{RGB}$, $\bar{F}^{P,C}_{LBP}$, $\bar{F}^{P}_{RGB}$, $\bar{F}^{P}_{LBP}$ are concatenated as the final feature F_{final}:

$$F_{final} = \text{Concat}(\bar{F}^{P,C}_{RGB}, \bar{F}^{P,C}_{LBP}, \bar{F}^{P}_{RGB}, \bar{F}^{P}_{LBP}). \tag{8}$$

where F_{final} is followed by one FC layer that is trained with the standard binary cross entropy loss L_{cls} to predict the probability of AI-generated images.

4 Experiment

4.1 Datasets and Implementation Details

Experiments are conducted on the GenImage, DRCT-2M, Co-Spy-Bench, and ForenSynths benchmarks. The evaluation metric is accuracy with 0.5 threshold. The operator $LBP^{u2}_{8,1}$ is used to compute the LBP images based on the RGB images. The LBP encoder is the ResNet-34 He et al. (2016) pretrained on ImageNet Deng et al. (2009). The RGB encoder network is the pre-trained Dinov2-ViT-L-14 with LoRA Hu et al. (2022) to finetune. During training, we use the Adam optimizer with the learning rate of 1×10^{-4}. The batch size is set to 32. The model is trained with PyTorch 1.13.0 on one NVIDIA RTX A6000 GPU for 10 epochs.

Methods	CNNSpot	LNP	UnivFD	DIRE	FreqNet	NPR	DRCT	CO-SPY	GenSign	IAPL	**LBPD$_{\text{based}}$**	**LBPD$_{\text{guided}}$**
LDM	77.6	84.9	79.1	66.3	74.9	84.3	79.3	95.0	95.5	96.2	95.4	98.5
SD-v1.4	90.0	95.9	80.9	83.5	69.2	90.9	81.2	92.0	93.8	94.3	95.6	98.7
SD-v1.5	90.0	96.2	80.9	83.8	68.9	91.3	81.1	91.3	94.5	94.5	95.9	98.6
SSD-1B	66.6	79.1	76.5	56.6	49.2	47.9	75.8	83.2	92.1	92.0	94.1	97.1
tiny-sd	66.4	81.5	77.0	63.6	63.6	88.4	80.0	84.8	90.8	91.5	94.6	98.1
SegMoE-SD	74.4	86.6	83.1	66.0	64.4	93.8	75.1	89.5	92.9	93.8	96.4	98.4
small-sd	70.1	80.7	77.4	68.4	65.4	89.1	81.2	85.8	92.1	92.0	95.5	98.5
SD-2-1	68.1	57.2	81.7	65.2	51.6	51.3	76.1	88.5	93.5	94.2	94.6	97.6
SD-3-medium	60.7	53.7	78.4	56.8	49.6	50.0	75.0	82.9	91.6	91.5	91.9	94.6
SDXL-turbo	88.7	83.5	84.3	73.0	66.6	83.6	80.4	95.4	94.1	95.0	95.1	98.5
SD-2	65.7	54.6	73.8	60.1	50.6	51.2	75.1	83.7	90.3	91.7	93.1	96.0
SDXL	61.8	80.3	63.6	51.9	48.8	47.7	75.2	74.1	87.6	85.0	93.0	95.9
PG-v2.5-1024	53.6	79.1	78.2	52.3	48.7	47.7	71.3	88.7	91.7	92.5	94.6	97.4
PG-v2-1024	63.5	52.5	78.5	56.6	48.8	48.4	66.6	89.1	91.3	91.4	92.4	95.9
PG-v2-512	57.9	49.4	58.9	53.6	49.1	49.1	77.6	64.9	88.5	88.0	90.3	93.3
PG-v2-256	63.3	54.8	63.0	60.3	49.4	49.7	73.5	72.9	85.0	83.9	87.4	90.2
PAXL-2-1024	56.3	53.4	80.1	54.8	49.4	51.2	71.8	93.9	89.8	90.5	91.0	94.1
PAXL-2-512	65.4	74.2	80.3	62.2	57.2	81.3	75.5	94.9	92.1	94.5	94.7	97.7
LCM-sdxl	81.5	85.5	78.0	68.6	62.3	50.9	81.1	96.2	94.1	95.9	95.4	98.5
LCM-sdv1-5	92.3	90.9	79.7	79.0	76.2	93.7	79.6	97.1	96.3	97.3	95.3	98.1
FLUX.1-sch	56.0	54.4	72.9	56.3	50.4	53.1	68.4	85.2	89.4	90.0	88.6	91.8
FLUX.1-dev	57.4	54.1	75.9	56.3	49.1	50.7	70.7	86.1	90.5	91.3	91.1	94.1
Avg.	69.4	71.9	76.5	63.4	57.4	65.7	76.0	87.1	91.7	92.1	**93.4**	**96.4**

Table 3: Results on Co-Spy-Bench Cheng et al. (2025). Methods are trained on DRCT-2M/SDv1.4 following CO-SPY Cheng et al. (2025). The metric is Averaged Accuracy with 0.5 as the threshold. The compared methods are CNNSpot Wang et al. (2020), LNP Liu et al. (2022), UnivFD Ojha et al. (2023), DIRE Wang et al. (2023), FreqNet Tan et al. (2024a), NPR Tan et al. (2024b), DRCT Chen et al. (2024b), CO-SPY Cheng et al. (2025), GenSign Fei et al. (2026), and IAPL Li et al. (2026).

4.2 Main Results

Results of LBPD$_{\text{based}}$. We compare the proposed LBPD$_{\text{based}}$ with several state-of-the art generated image detection methods. As shown in Table 1, the LBPD$_{\text{based}}$ gets 95.2% averaged accuracy on GenImage, which almost achieves the state-of-the-art results (slightly lower than IAPL Li et al. (2026) by 1.5 points). Note that the LBPD$_{\text{based}}$ outperforms other texture-based methods by a large margin (more than 4.5 points), such as GramNet Liu et al. (2020) and ESSP Chen et al. (2024c). When compared with the SOTA method IAPL Li et al. (2026) on the DRCT-2M dataset in Table 2, LBPD$_{\text{based}}$ outperforms IAPL by 0.5 points with 93.0% accuracy. Table 3 shows that LBPD$_{\text{based}}$ set new records on Co-Spy-Bench dataset with 93.4% accuracy. The results of LBPD$_{\text{based}}$ on all three datasets demonstrate that LBP is a good descriptor for AI-generated image detection task.

	real vs fake	real vs real	fake vs fake
JSD	0.1905	0.0152	0.0296
EMD	0.1562	0.0107	0.0305

Table 4: The Jensen-Shannon Divergence (JSD) and Earth Mover's Distance (EMD) for histogram of LBP between real and fake images.

Results of LBPD$_{\text{guided}}$. We compare LBPD$_{\text{guided}}$ with other methods on the GenImage, DRCT-2M, and Co-Spy-Bench datasets, respectively. Table 1, Table 2, and Table 3 show that the proposed LBPD$_{\text{guided}}$ sets new records on three benchmarks with 98.7%, 95.1%, and 96.4%, respectively. Specifically, LBPD$_{\text{guided}}$ performs significantly better than most of the methods on the more challenged ADM and GLIDE test-sets of GenImage in Table 1. The LBPD$_{\text{guided}}$ performs better than other methods all test-sets of DRCT-2M, only except for the SDv1-DR test-set, in Table 2. The LBPD$_{\text{guided}}$ outperforms all other methods by at least 4 points on Co-Spy-Bench in Table 3. Overall, the results on Table 1, 2, 3 show the superiority of the LBPD$_{\text{guided}}$.

Input	GenImage
High Frequency	85.3
NPR Tan et al. (2024b)	87.0
Texture calculation method in Zhong et al. (2023); Chen et al. (2024c)	90.2
Default $LBP_{8,1}$	93.0
Rotation Invariant $LBP^{ri}_{8,1}$	94.0
Uniform $LBP^{ri,u2}_{8,1}$	94.7
Rotation Variant Uniform $LBP^{u2}_{8,1}$	**95.2**

Table 5: Ablation of LBP based on the LBPD$_{\text{based}}$ with different input.

4.3 Ablation Study of LBPD$_{\text{based}}$

Statistical analysis to show the discrimination of LBP. We compute the Jensen-Shannon Divergence (JSD) and Earth Mover's Distance (EMD) for the histogram of LBP between real and fake images. As shown in the Table 4, since bigger JSD and EMD mean greater distinction between two distributions, this rigorous statistical quantity further shows that the LBP distributions differ between real and generated images.

Ablation of LBP. The detector is LBPD$_{\text{based}}$ with different inputs. Tan et al. (2024b) propose NPR to capture the low-level forgery artifacts in order to detect fake images. NPR achieves 87.0% on

GenImage, which is lower than LBP in Table 5. This rules out the possibility that the performance gain of LBP is simply due to the introduction of a supplementary low-level signal.

"High Frequency" in Table 5 means that we use the high frequency component of the image as model input, since there are evidences that the high frequency of the image is useful to detect fake images Luo et al. (2021); Gao et al. (2024); Tan et al. (2024a). We also compare LBP with the texture calculation method in Zhong et al. (2023); Chen et al. (2024c), which is simply summing the residual of four directions, including diagonal, counter-diagonal, horizontal, and vertical directions. Table 5 shows that LBP achieves better results than "High Frequency" and the simple texture calculation method.

LBP	RGB	SR	CR	GenImage
	✓			87.7
✓				95.2
✓	✓			96.2
✓	✓	✓		97.9
✓	✓		✓	97.5
✓	✓	✓	✓	**98.7**

Table 6: Ablation study of $\text{LBPD}_{\text{guided}}$ on GenImage. The Global Refinement Module (GRM) consists of Spatial Refinement (SR) and Channel Refinement (CR).

There are 4 types of LBP as described in Section 3.1, i.e., default $LBP_{8,1}$, rotation invariant $LBP^{ri}_{8,1}$, uniform $LBP^{ri,u2}_{8,1}$, and rotation variant uniform $LBP^{u2}_{8,1}$. Here we explore the performance of different types of LBP on GenImage. As shown in Table 5, we find that the results of different types of LBP have some differences, where the default $LBP_{8,1}$ performs worst with 93.0% accuracy, while the rotation variant uniform $LBP^{u2}_{8,1}$ performs best with 95.2% accuracy. According to the Table 5, we choose $LBP^{u2}_{8,1}$ by default.

4.4 Ablation Study of $\textbf{LBPD}_{\text{guided}}$

The effectiveness of GRM. The proposed $\text{LBPD}_{\text{guided}}$ mainly consists of the LBP encoder, the RGB encoder, and the proposed GRM that contains the SR and CR. As shown in Table 6, when concatenating both RGB and LBP features, the detector achieves 96.2% accuracy, which is better than detector that only uses LBP features (95.2%) or RGB features (87.7%). Furthermore, SR boosts performance from 96.2% to 97.9% by 1.7 points, and CR improves performance from 96.2% to 97.5% by 1.3 points, showing the effectiveness of SR and CR, respectively. Finally, SR and CR work together to improve performance from 96.2% to 98.7% by 2.5 points, indicating that SR and CR are complementary.

LBP	RGB	SR	CR	GenImage
	✓			84.3
✓				95.2
✓	✓			95.7
✓	✓	✓		96.7
✓	✓		✓	96.4
✓	✓	✓	✓	**97.2**

Table 7: Generalized to different RGB encoders. The difference between this Table and Table 6 is that RGB encoder is changed to CLIP encoder.

Different RGB encoders. To validate that the proposed method can be generalized to different RGB encoders, we change the RGB encoder from Dinov2 to the CLIP image encoder following Ojha et al. (2023); Chen et al. (2024b). The results are shown in Table 7. It shows the proposed method could still boost the performance when using the CLIP image encoder as the RGB encoder. For example, concatenating both the LBP and RGB features boosts the performance from 95.2% to 95.7% by 0.5 points, and the GRM, which consists of SR and CR, boosts the performance from 95.7% to 97.2% by 1.5 points.

4.5 Experiments in the Appendix

We include more experiments in the Appendix: (1) Results on the GAN-generated ForenSynths dataset. (2) ***Robust results of image degradation on GenImage.*** (3) More ablation study about LBP. (4) Inference time.

5 Conclusion

In this paper, we find that the Local Binary Pattern (LBP) is a good descriptor for AI-generated image detection. The LBP images, frequency spectrum, and the corresponding histograms show significantly different distributions between real and AI-generated images. Therefore, we propose the simple yet effective $\textbf{LBPD}_{\textbf{based}}$ detector. To fully leverage the LBP descriptor of images, we also proposed the $\textbf{LBPD}_{\textbf{guided}}$ detector, which combines the LBP features with image features by the GRM. Extensive experiments on four benchmarks verify the effectiveness of the proposed method.

REFERENCES

Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face recognition with local binary patterns. In *European conference on computer vision*, pp. 469–481. Springer, 2004.

Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 28(12):2037–2041, 2006.

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 *CVPR*, pp. 10759–10769, 2024.

Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. What makes fake images detectable? understanding properties that generalize. In *European conference on computer vision*, pp. 103–120. Springer, 2020.

Baoying Chen, Jishen Zeng, Jianquan Yang, and Rui Yang. Drct: Diffusion reconstruction contrastive training towards universal detection of diffusion generated images. In *Forty-first International Conference on Machine Learning*, 2024a.

Baoying Chen, Jishen Zeng, Jianquan Yang, and Rui Yang. Drct: Diffusion reconstruction contrastive training towards universal detection of diffusion generated images. In *ICML*, 2024b.

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*, 2024c.

Siyuan Cheng, Lingjuan Lyu, Zhenting Wang, Xiangyu Zhang, and Vikash Sehwag. Co-spy: Combining semantic and pixel features to detect synthetic images by ai. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 13455–13465, 2025.

Sungik Choi, Hankook Lee, Jaehoon Lee, Robin Kim, Stanley Jungkyu Choi, and Moontae Lee. A debiased reconstruction-based framework for training-free detection of ai-generated images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3254–3263, 2026.

Beilin Chu, Xuan Xu, Xin Wang, Yufei Zhang, Weike You, and Linna Zhou. Fire: Robust detection of diffusion-generated images via frequency-guided reconstruction error. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12830–12839, 2025.

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*, pp. 1–5. IEEE, 2023.

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

Jianwei Fei, Yunshu Dai, Xiaoyu Zhou, Zhihua Xia, and Alessandro Piva. Enabling supervised learning of generative signatures for generalized ai-generated images detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14084–14094, 2026.

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*, pp. 3247–3258. PMLR, 2020.

Jie Gao, Zhaoqiang Xia, Gian Luca Marcialis, Chen Dang, Jing Dai, and Xiaoyi Feng. Deepfake detection based on high-frequency enhancement network for highly compressed content. *Expert Systems with Applications*, 249:123732, 2024.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33: 6840–6851, 2020.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Yonghyun Jeong, Doyeon Kim, Seungjai Min, Seongho Joe, Youngjune Gwon, and Jongwon Choi. Bihpf: Bilateral high-pass filters for robust deepfake detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 48–57, 2022a.

Yonghyun Jeong, Doyeon Kim, Youngmin Ro, and Jongwon Choi. Frepgan: robust deepfake detection using frequency-level perturbations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 1060–1068, 2022b.

Zexi Jia, Chuanwei Huang, Yeshuang Zhu, Hongyan Fei, Xiaoyue Duan, Zhiqiang Yuan, Ying Deng, Jiapei Zhang, Jinchao Zhang, and Jie Zhou. Secret lies in color: Enhancing ai-generated images detection with color distribution analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 13445–13454, 2025.

Yiheng Li, Zichang Tan, Guoqing Xu, Zhen Lei, Xu Zhou, and Yang Yang. Towards generalizable ai-generated image detection via image-adaptive prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21262–21272, 2026.

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*, pp. 95–110. Springer, 2022.

Huan Liu, Zichang Tan, Chuangchuang Tan, Yunchao Wei, Jingdong Wang, and Yao Zhao. Forgery-aware adaptive transformer for generalizable synthetic image detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10770–10780, 2024.

Zhengzhe Liu, Xiaojuan Qi, and Philip HS Torr. Global texture enhancement for fake face detection in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8060–8069, 2020.

Yuchen Luo, Yong Zhang, Junchi Yan, and Wei Liu. Generalizing face forgery detection with high-frequency features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16317–16326, 2021.

Yunpeng Luo, Junlong Du, Ke Yan, and Shouhong Ding. Lareˆ 2: Latent reconstruction error based method for diffusion-generated image detection. In *CVPR*, pp. 17006–17015, 2024.

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.

Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996.

Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002.

Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24480–24489, 2023.

Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. *Thinking in Frequency: Face Forgery Detection by Mining Frequency-aware Clues*, pp. 86–103. Jan 2020a.

Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *ECCV*, pp. 86–103. Springer, 2020b.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International conference on machine learning*, pp. 5301–5310. PMLR, 2019.

Jonas Ricker, Simon Damm, Thorsten Holz, and Asja Fischer. Towards the detection of diffusion model deepfakes. *arXiv preprint arXiv:2210.14571*, 2022.

Jonas Ricker, Denis Lukovnikov, and Asja Fischer. Aeroblade: Training-free detection of latent diffusion images using autoencoder reconstruction error. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9130–9140, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pp. 10684–10695, 2022.

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 *Proceedings of the 2023 ACM SIGSAC conference on computer and communications security*, pp. 3418–3432, 2023.

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.

Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao Wei. Frequency-aware deepfake detection: Improving generalizability through frequency space domain learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 5052–5060, 2024a.

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 *CVPR*, pp. 28130–28139, 2024b.

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*, volume 30, 2017.

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*, pp. 8695–8704, 2020.

Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, Hezhen Hu, Hong Chen, and Houqiang Li. Dire for diffusion-generated image detection. In *CVPR*, pp. 22445–22455, 2023.

Zhiyuan Yan, Jiangming Wang, Peng Jin, Ke-Yue Zhang, Chengchun Liu, Shen Chen, Taiping Yao, Shouhong Ding, Baoyuan Wu, and Li Yuan. Orthogonal subspace decomposition for generalizable ai-generated image detection. In *Forty-second International Conference on Machine Learning*, 2025a.

Zhiyuan Yan, Junyan Ye, Weijia Li, Zilong Huang, Shenghai Yuan, Xiangyang He, Kaiqing Lin, Jun He, Conghui He, and Li Yuan. Gpt-imgeval: A comprehensive benchmark for diagnosing gpt4o in image generation. *arXiv preprint arXiv:2504.02782*, 2025b.

Chende Zheng, Chenhao Lin, Zhengyu Zhao, Hang Wang, Xu Guo, Shuai Liu, and Chao Shen. Breaking semantic artifacts for generalized ai-generated image detection. *Advances in Neural Information Processing Systems*, 37:59570–59596, 2024.

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.

Nan Zhong, Haoyu Chen, Yiran Xu, Zhenxing Qian, and Xinpeng Zhang. Beyond generation: A diffusion-based low-level feature extractor for detecting ai-generated images. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 8258–8268, 2025.

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. *Advances in Neural Information Processing Systems*, 36:77771–77782, 2023.

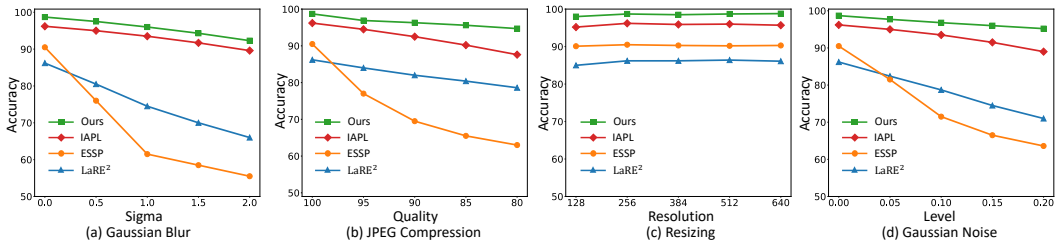

Figure 9: **Robustness to Image Degradation.** Gaussian blur, JPEG compression, Resizing, and Gaussian Noise are applied to LaRE^2 Luo et al. (2024), ESSP Chen et al. (2024c), IAPL Li et al. (2026) and our LBPD$_{\text{guided}}$, the results demonstrate the robustness of ours to unseen image perturbations.

A Appendix

A.1 More Dataset Details

Experiments are conducted on the following benchmarks: GenImage Zhu et al. (2023), DRCT-2M Chen et al. (2024a), Co-Spy-Bench Cheng et al. (2025), and ForenSynths dataset Wang et al. (2020). The training set of GenImage contains fake images generated using Stable Diffusion V1.4. For the test set, we incorporate a variety of generators, including Stable Diffusion V1.4, Stable Diffusion V1.5, GLIDE, VQDM, Wukong, BigGAN, ADM, and Midjourney. DRCT-2M includes 16 types of generated images, with 120k images for each type: 10 types of the currently available SD text-to-image models, 6 types of ControlNet and 3 types diffusion inpainting models. Co-Spy-Bench consists of 22 types text-to-image generative models: 8 types Stable Diffusion variants, 4 types Playground models, 4 types Segmind models, 2 types PixArt models, 2 types LCM models, and 2 types Flux models. ForenSynths contains 7 types of GAN-based generated images (ProGAN, StyleGAN, StyleGAN2, BigGAN, CycleGAN, StarGAN, GauGAN) and 1 Deepfake forgery images.

A.2 Results on GAN-generated Images

Methods	ProGAN	StyleGAN	StyleGAN2	BigGAN	CycleGAN	StarGAN	GauGAN	Deepfake	Mean
CNNSpot Wang et al. (2020)	91.4	63.8	76.4	52.9	72.7	63.8	63.9	51.7	67.1
FreqFD Frank et al. (2020)	90.3	74.5	73.1	88.7	75.5	99.5	69.2	60.7	78.9
PatchFor Chai et al. (2020)	97.8	82.6	83.6	64.7	74.5	100.0	57.2	85.0	80.7
F3Net Qian et al. (2020b)	99.4	92.6	88.0	65.3	76.4	100.0	58.1	63.5	80.4
BiHPF Jeong et al. (2022a)	90.7	76.9	76.2	84.9	81.9	94.4	69.5	54.4	78.6
FrePGAN Jeong et al. (2022b)	99.0	80.7	84.1	69.2	71.1	99.9	60.3	70.9	79.4
LGrad Tan et al. (2023)	99.9	94.8	96.0	82.9	85.3	99.6	72.4	58.0	86.1
UnivFD Ojha et al. (2023)	99.7	89.0	83.9	90.5	87.9	91.4	89.9	80.2	89.1
NPR Tan et al. (2024b)	99.8	96.3	97.3	87.5	95.0	99.7	86.6	77.4	92.5
FatFormer Liu et al. (2024)	99.9	97.2	98.8	99.5	**99.3**	99.8	99.4	93.2	98.4
CoD Jia et al. (2025)	99.9	98.2	98.6	93.6	99.1	99.9	**99.9**	93.4	98.2
LBPD$_{\text{based}}$	99.5	99.2	98.7	93.4	98.0	99.0	98.2	91.3	97.2
LBPD$_{\text{guided}}$	**99.9**	**99.5**	**99.1**	**99.8**	99.0	**99.9**	99.1	**94.9**	**98.9**

Table 8: Accuracy comparisons with state-of-the-art methods on ForenSynths dataset. The decision threshold is 0.5.

We compare the proposed LBPD$_{\text{guided}}$ on ForenSynths dataset Wang et al. (2020). As shown in Table 8, the proposed LBPD$_{\text{guided}}$ also achieves state-of-the-art results on GAN-generated images, when compared with other state-of-the-art methods, such as FatFormer Liu et al. (2024) and CoD Jia et al. (2025). Specifically, the proposed LBPD$_{\text{guided}}$ achieves a mean accuracy of 98.9%, demonstrating its effectiveness in GAN-generated image detection.

A.3 Robustness to Image Degradation

To further demonstrate the robustness to image perturbations and degradations, we train our model and comparison models on Stable Diffusion V1.4, and test on 8 subsets of GenImage, which are processed by Gaussian blur, JPEG compression, Resizing, and Gaussian Noise to different extents. Fol-

Input	GenImage
High Frequency	91.8
NPR Tan et al. (2024b)	93.3
Texture calculation method in Zhong et al. (2023); Chen et al. (2024c)	95.1
LBP	98.7

Table 9: Ablation of LBP based on the $\text{LBPD}_{\text{guided}}$ with different input.

P	R	GenImage
8	1	**95.2**
16	1	94.5
8	2	94.2
16	2	94.0
24	2	93.5

Table 10: Ablation of Hyper-parameters P and R in LBP based on $\text{LBPD}_{\text{based}}$.

Method	GenImage
ResNet-18	92.0
ResNet-34	**95.2**
ResNet-50	**95.3**
ResNet-101	94.9

Table 11: Ablation of different LBP encoders based on $\text{LBPD}_{\text{based}}$.

lowing Wang et al. (2020) and Wang et al. (2023), we choose Gaussian blur (σ = 0,0.5,1.0,1.5,2.0), JPEG compression (*quality* = 100%,95%,90%,85%,80%), Resizing (*resolution* = 128, 256, 384, 512, 640), and Gaussian Noise ((*level* = 0.00, 0.05, 0.10, 0.15, 0.20) when testing, the results are shown in Figure 9. We observe that when the image degrades, existing methods, like LaRE^2 Luo et al. (2024) and ESSP Chen et al. (2024c), encounter a huge decline. In contrast, the proposed $\text{LBPD}_{\text{guided}}$ shows limited performance decline against these interferences, demonstrating the robustness to unseen image perturbations and degradations.

A.4 More Ablation Study

Different inputs based on $\text{LBPD}_{\text{guided}}$. Besides Table 5 that explores different inputs based on $\text{LBPD}_{\text{based}}$, we further conduct ablation study of different inputs based on $\text{LBPD}_{\text{guided}}$. Table 9 shows that LBP acheives better results than High Frequency, NPR, and the texture calculation method in Zhong et al. (2023); Chen et al. (2024c). This further rules out the possibility that the performance gain of LBP is simply due to the introduction of a supplementary low-level signal.

Hyper-parameters of LBP. The LBP have two hyper-parameters P and R, where P denotes the sampling points on a circle of radius R. We discuss the effect of P and R with experiments on GenImage with $\text{LBPD}_{\text{based}}$ as the detector. Table 10 shows that $P = 8$ and $R = 1$ achieves the best results.

Different LBP encoders. We use ResNet-34 pretrained on ImageNet as the default LBP encoder. In Table 11, we compare different backbones of LBP encoders. It shows that ResNet-34, ResNet-50, and ResNet-101 achieve similar results on GenImage, while ResNet-18 demonstrate inferior performance. We choose ResNet-34 as the default setting.

A.5 INFERENCE TIME

The inference time of $\text{LBPD}_{\text{based}}$ and $\text{LBPD}_{\text{guided}}$ is 7.5 ms and 24.0 ms, respectively. $\text{LBPD}_{\text{guided}}$ achieves better performance than $\text{LBPD}_{\text{based}}$, while showing higher inference time. Therefore, $\text{LBPD}_{\text{based}}$ and $\text{LBPD}_{\text{guided}}$ can provide different trade-offs between performance and speed, according to different application requirements. We also want to emphasize that both $\text{LBPD}_{\text{based}}$ and $\text{LBPD}_{\text{guided}}$ achieve real-time.