# FALCON-Net: Feature Aggregation of Local Patterns for AI-Generated Image Detection

Dengyong Zhang, Miao Hu, Jiaxin Chen, Changsheng Chen, *Senior Member, IEEE*, Jin Wang, *Senior Member, IEEE*, Yun Song, Gaobo Yang, Xin Liao, *Senior Member, IEEE*, and Xiangling Ding

***Abstract*—With the rapid development of generative models, the visual quality of generated images has become almost indistinguishable from real images, which poses a huge challenge to content authenticity verification. A key limitation of existing detectors is their reliance on model-specific cues, resulting in poor generalization to unseen models. Based on the observation of local differences in the generated images, we found that the generated images lack device-specific sensor noise and unnatural pixel intensity variations caused by the oversimplified generation process. These discrepancies provide important forensic cues for distinguishing between real and generated images. We propose the Feature Aggregation for Localized Context and Noise Network (FALCON-Net), which leverages these discrepancies to enhance detection capabilities. FALCON-Net integrates two complementary modules to enhance detection capabilities: the Intrinsic Noise Pattern Isolation (INP) module isolates device-specific noise patterns by analyzing high-frequency features in the frequency domain, while the Local Variation Pattern (LVP) module models the complex relationships between local pixels to capture directional intensity variations and reveal unnatural regularities in generated images. By combining these sensor-level and local structural cues, FALCON-Net identifies fundamental generative inconsistencies, ensuring robustness to post-processing and strong generalization to unseen models. Extensive experimental results show that FALCON-Net achieves the state-of-the-art performance in detecting generated images and shows good generalization ability to unseen generative models. The code is available at https://github.com/humiaomiaohaha/FALCON-Net**

***Index Terms*—AI-generated image detection, intrinsic noise pattern isolation, local variation patterns.**

Received 16 September 2025; revised 10 February 2026 and 6 April 2026; accepted 7 April 2026. Date of publication 13 April 2026; date of current version 20 April 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 62172059, Grant 62402062, Grant U22A2030, Grant 62072313, and Grant 62272160; in part by Hunan Provincial Funds for Distinguished Young Scholars under Grant 2024JJ2025; in part by Hunan Provincial Key Research and Development Program under Grant 2024AQ2027 and Grant 2025AQ2022; in part by the Natural Science Foundation of Hunan Province, China, under Grant 2025JJ60415; and in part by the Research Foundation of Education Bureau of Hunan Province, China, under Grant 25B0221. The associate editor coordinating the review of this article and approving it for publication was Prof. Davide Cozzolino. *(Corresponding author: Changsheng Chen.)*

Dengyong Zhang, Miao Hu, Jiaxin Chen, and Yun Song are with the School of Computer Science and Technology, Changsha University of Science and Technology, Changsha 410114, China (e-mail: zhdy@csust.edu.cn; 23108011670@stu.csust.edu.cn; jxchen@csust.edu.cn; songyun@csust.edu.cn).

Changsheng Chen is with the Faculty of Engineering, Shenzhen MSU-BIT University, Shenzhen 518172, China (e-mail: cschen@smbu.edu.cn).

Jin Wang and Xiangling Ding are with the School of Computer and Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201, China (e-mail: jinwang@hnust.edu.cn; xianglingding@hnust.edu.cn).

Gaobo Yang and Xin Liao are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China (e-mail: yanggaobo@hnu.edu.cn; xinliao@hnu.edu.cn).

Digital Object Identifier 10.1109/TIFS.2026.3683286

## I. Introduction

With the rapid development of generative models, such as Midjourney [1] and DALL-E3 [2], AI-generated images have achieved a visual quality that closely resembles real images, posing significant challenges to news communication and judicial authentication. This highlights the urgent need for automated detection methods that can accurately identify synthetic images and maintain the authenticity of visual content.

In recent years, deep learning-based methods have demonstrated powerful feature extraction capabilities for detecting synthetic images [3], [4], [5] and have achieved excellent detection performance on multiple generative models. These methods typically utilize convolutional neural networks (CNNs) to deeply mine global or local features of images, effectively capturing forgery traces in generated images. However, a major issue with existing deepfake detection models is their limited generalization across different data distributions, making them incapable of adapting to generative techniques not seen in the training. To address this issue, some studies have attempted to introduce large-model-based methods [6], [7], [8], [9], leveraging large-scale pre-trained models to improve cross-model generalization and adaptability to a variety of generative techniques. Nevertheless, these methods typically rely on pre-trained models from large datasets, which presents several limitations. Pre-trained large models require significant computational resources and storage space, especially during training and inference, making them difficult to deploy in resource-constrained environments. Furthermore, the performance of large models relies heavily on the large, high-quality datasets required for pre-training, which are expensive to acquire and may have skewed distributions, directly impacting the model's detection performance. Therefore, how to improve detection performance while reducing dependence on computing resources and data size remains a core challenge in current research.

To address these limitations, we propose the Feature Aggregation for Localized Context and Noise Network (FALCON-Net), which is inspired by the inherent limitations of current generative models, That is, generative models often lead to random high-frequency artifacts and lack of natural

randomness in pixel intensity variations during the generation process, which are the key differences between generated and real images. Real images usually show natural device fingerprint information and natural randomness in pixel distribution, while generated images usually show uniformity, oversimplified details, and artifacts introduced during the generation process.

Importantly, the above discrepancies are not tied to a specific generator's visible artifacts, but stem from two broader and more model-agnostic factors: (i) real images are produced by physical acquisition pipelines, leaving coherent acquisition-related traces, whereas synthetic images typically bypass such pipelines; and (ii) many generators share low-level synthesis inductive biases, which tend to produce locally more regular and less complex neighborhood intensity relationships than those formed by real imaging processes. These factors provide principled reasons for better transferability across unseen generative models.

Based on the above phenomenon, FALCON-Net introduces two complementary modules: the Intrinsic Noise Pattern Isolation (INP) module extracts high-frequency noise features in the frequency domain. The Local Variation Pattern (LVP) module models directional pixel intensity relationships. These two modules comprehensively capture the differences between real and generated images, laying a solid foundation for detection.

Specifically, INP focuses on acquisition-level residual evidence by suppressing content-dominated components and isolating high-frequency patterns that are indicative of sensor traces. Since this cue is formulated as an acquisition-consistency signal, rather than matching a generator-dependent fake noise signature, it is expected to generalize across different synthesis pipelines. In parallel, LVP characterizes dependency statistics via directional intensity-relation encoding. Rather than assuming a new generator reproduces identical local artifacts, LVP targets a distributional gap in neighborhood-relation complexity that can arise from shared synthesis mechanisms, making it more transferable across architectures and generation paradigms.

FALCON-Net further integrates a lightweight classifier to combine the features extracted by the INP and LVP modules. This classifier, built upon a pruned ResNet architecture, efficiently processes both global sensor-level and local pixel-level features, enabling accurate differentiation between real and generated images. Unlike existing methods that usually rely heavily on specific generated features, FALCON-Net focuses on these generalized properties, allowing it to effectively generalize to a variety of generative models, including unseen architectures and data distributions. The main contributions are as follows,

Based on the above phenomenon, FALCON-Net introduces two complementary modules: the Intrinsic Noise Pattern Isolation (INP) module extracts high-frequency noise features in the frequency domain. The Local Variation Pattern (LVP) module models directional pixel intensity relationships. These two modules comprehensively capture the differences between real and generated images, laying a solid foundation for detection. FALCON-Net further integrates a lightweight classifier to combine the features extracted by the INP and LVP modules. This classifier, built upon a pruned ResNet architecture, efficiently processes both global sensor-level and local pixel-level features, enabling accurate differentiation between real and generated images. Unlike existing methods that usually rely heavily on specific generated features, FALCON-Net focuses on these generalized properties, allowing it to effectively generalize to a variety of generative models, including unseen architectures and data distributions. The main contributions are as follows,

- We introduce sensor fingerprint information and pixel encoding patterns into AI-generated image detection, offering a novel perspective for cross-generator detection. By analyzing generative models' limitations during image generation, we model critical artifacts like sensor noise and unnatural pixel variations to find the most discriminative information, which are not determined by any specific single generator, significantly enhances the generalization capability of the FALCON-Net.
- We propose two complementary modules, the Intrinsic Noise Pattern Isolation (INP) and the Local Variation Pattern (LVP), to extract discriminative features from different perspectives. The INP module extracts high-frequency noise features through frequency domain analysis, effectively modeling the subtle distortions introduced by sensors in real images and capturing anomalies in the high-frequency regions of generated images. Meanwhile, the LVP module models the intensity relationships between local pixels via directional encoding, revealing the lack of natural complexity in the local patterns of generated images. This modular design allows FALCON-Net to synergistically analyze discrepancies at both the sensor and pixel-distribution levels, creating a richer source of forensic evidence by combining these distinct features.
- We conduct extensive experiments to verify the effectiveness of FALCON-Net, which not only demonstrates superior generalization ability across unseen generative models, including various GANs and diffusion models, but also exhibits strong robustness against common post-processing attacks. These results collectively highlight how FALCON-Net elegantly resolves the long-standing trade-off in the field between generalization, efficiency, and robustness.

## II. Related Work

### A. Local Feature-Based Detection

Unlike approaches that rely on global semantics or generator-specific fingerprints, many studies focus on uncovering forgery traces in local regions of images [10], [11], [12], [13], [14], [15], [16]. The core idea of these methods is that, while generated images may appear realistic at a macro level, they often reveal fundamental differences from real images in terms of pixel statistics, texture structures, or physical and physiological consistencies at a micro level. Some research emphasizes using local textures and regional relationships as key cues. Foundational work in image synthesis, such as Gatys et al. [10], [11], introduced the computation of local response correlations between feature maps in convolutional neural networks. Which effectively model and reconstruct the textures and styles of images, providing a theoretical basis for texture-based forgery detection. Building on this foundation, Chen et al. [12] divided the feature map of a single face into multiple local patches and calculated the similarity between

them to construct forgery patterns for differentiation. Hsu et al. [13] introduced a pairwise learning-based detection method, leveraging a Siamese network structure and contrastive loss to distinguish real and fake content from localized image regions.

Another category of methods focuses on identifying internal inconsistencies within a single face at a physical or physiological level. These methods often exploit specific biological traits or physical principles. For example, Jung et al. [14] proposed an algorithm, which detects forgeries by analyzing local physiological patterns such as the frequency, duration, and periodicity of eye blinking in videos, as forged videos often fail to replicate natural and subconscious blinking behaviors. Similarly, Hu et al. [15] utilized the physical principle of corneal specular highlights, extracting and comparing the reflective regions in both eyes of the same face. This method identified discrepancies in the consistency of these reflections as evidence of forgery. Yang et al. [16] exposed forgeries through inconsistent head poses. By separately estimating 3D head poses using all facial landmarks and only central facial features, the method detected forgeries if the two pose estimations exhibit significant differences. Despite their success with localized features, their applicability is often constrained to a narrow scope, primarily targeting manipulated human faces. Consequently, their effectiveness may be limited when applied to the heterogeneous nature of modern AI-generated content, which encompasses a vast range of objects and scenes. In contrast, the proposed FALCON-Net leverages its intrinsic noise pattern (INP) module and local variation pattern (LVP) module to model two complementary and foundational dimensions: sensor noise fingerprints and localized pixel variation patterns. This design not only captures the missing natural randomness and complex local patterns in generated images with greater precision but also enhances generalization and robustness against unknown generative models and various post-processing operations through adaptive feature representations.

### B. Deep Learning-Based Detection

Deep learning-based methods have become mainstream for detecting AI-generated images due to their powerful feature extraction capabilities. Marra et al. [17] leveraged image content and contextual information to improve detection on social networks. Wang et al. [18] utilized adversarial training and feature matching to enhance detection accuracy. To tackle cross-domain challenges, Tan et al. [19] learned gradient information to capture subtle structural differences, boosting generalization across generators. Zhang et al. [20] introduced unsupervised domain adaptation to adapt models to unseen data. Lim et al. [21] designed a lightweight diffusion synthesis detector to reduce computational demands, and Zhao et al. [22] proposed a model for robust fake video face detection. Deep learning methods have demonstrated remarkable performance in forgery detection. However, their robustness remains a significant challenge. Many models are highly sensitive to variations introduced by unseen generative techniques and common post-processing operations, such as image compression, blurring, and cropping. These vulnerabilities stem from their reliance on specific training data distributions and their tendency to overfit to particular forgery artifacts. Consequently, enhancing robustness to such manipulations remains a key focus in current research. Different from the above approaches, our FALCON-Net focuses on improving generalization to unseen manipulations by refining local feature extraction and highlighting discrepancies between generated and real images, achieving strong robustness and adaptability even against unknown generative techniques.

### C. Large-Language-Model-Based Detection

In recent years, the powerful feature expression ability of pre-trained vision-language models (VLMs) has provided a new solution for AI-generated image detection. Liu et al. [6] introduced a low-rank adapter (LoRA) to capture global and specific artifact features in the CLIP model while retaining pre-training knowledge to improve generalization performance. Xu et al. [7] further combined LoRA with contrastive learning to achieve stronger generalization capabilities under very few sample conditions. Beyond parameter-efficient fine-tuning, Zhuang et al. [23] proposed a prompt tuning strategy (AntiFakePrompt) to optimize textual prompts for maximizing the feature discrepancy between real and fake images, rather than modifying the model weights. From the perspective of semantic consistency, Sha et al. [24] leveraged Multimodal Large Language Models (MLLMs) to caption input images and calculate the semantic mismatch between the image and text as a forensic cue. More recently, some studies have further explored large multimodal models for explainable detection and localization: Xu et al. [25] proposed *FakeShield*, a multi-modal framework that performs authenticity prediction, generates tampered-region masks, and provides textual judgment rationales; Huang et al. [26] introduced *SIDA*, which targets social-media scenarios and jointly supports detection, localization, and explanation with large multimodal models; Zhou et al. [27] developed *AIGI-Holmes*, which adapts MLLMs for explainable and generalizable AI-generated image detection via explanation-rich datasets and a staged training pipeline. However, this type of method still has certain limitations when samples are extremely scarce.

To this end, some researchers have explored detection strategies that directly use the CLIP feature space. For example, Cozzolino et al. [8] achieved excellent generalization and robustness on unseen generators by training a linear SVM classifier with only a small number of paired real or fake samples from a single generator. Ojha et al. [28] used nearest neighbors and linear classifiers to detect images generated by diffusion models and autoregressive models; in addition, they have shown that the CLIP feature is robust to operations such as compression and cropping. Furthermore, detection methods based on generative priors have also gained attention. Wang et al. [29] utilized the reconstruction error from pre-trained large diffusion models (DIRE) to identify generated content, based on the hypothesis that generated images are easier to reconstruct by the source model than real images.

Despite the remarkable performance achieved by large models, their practical applicability is often constrained by several inherent limitations. These models typically rely on pre-trained models that demand extensive computational resources, large-scale high-quality datasets, and lengthy training and testing durations. Such requirements not only significantly increase their complexity and operational costs but also make them difficult to deploy in resource-constrained environments or real-time scenarios. In contrast, the proposed FALCON-Net offers a relatively lightweight and efficient solution.

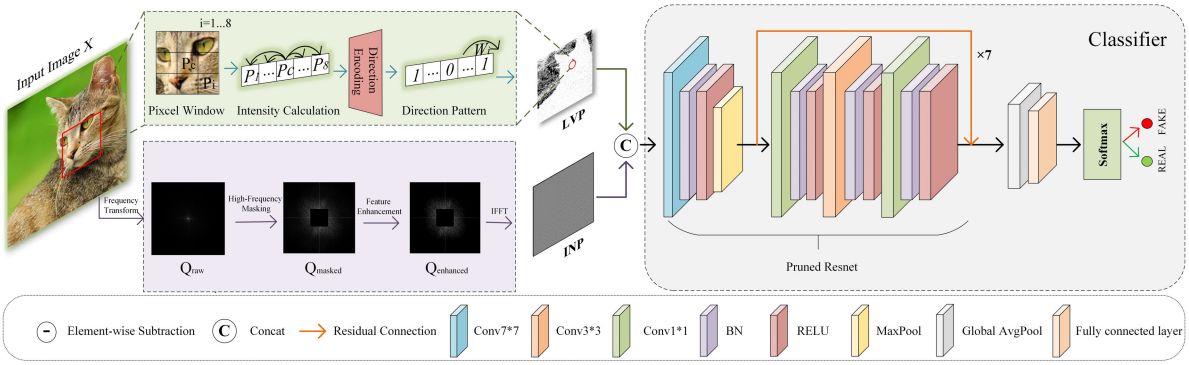



Fig. 1. Overview of the proposed FALCON-Net framework. The framework consists of three main components, which work collaboratively to detect AI-generated images, (1) The Inherent Noise Pattern (INP) module, which extracts high-frequency noise features in the frequency domain to emphasize the absence of device-specific sensor fingerprints in generated images. (2) The Local Variation Pattern (LVP) module, capturing unnatural regularities in pixel intensity distribution through directional encoding, thereby revealing the synthetic images' lack of natural complexity. (3) A lightweight classifier, constructed based on a pruned ResNet architecture. It integrates the features extracted by the INP and LVP modules and effectively classifies images as real or generated.

## III. Proposed Method

### A. Overview

As shown in Fig. 1, this paper proposes the Feature Aggregation for Localized Context and Noise Network (FALCON-Net), which is composed of the Intrinsic Noise Pattern Isolation (INP) module, the Local Variation Pattern (LVP) module, and a classifier. While many existing methods have shown promise, they often rely on specific, high-level forgery cues that can be brittle and lack generalization. For example, detectors based on physiological patterns, such as eye-blinking, are effective in specific scenarios like detecting forged human faces, but can fail when subjected to common post-processing and are inherently inapplicable to the diverse range of modern AI-generated content, which extends beyond faces to include varied objects and scenes. A heavily compressed video of a real person might exhibit degraded blinking features that become statistically indistinguishable from the artificial patterns in an uncompressed fake video, causing the detector to fail [30]. Similarly, other methods focusing on specific texture artifacts or frequency anomalies can be compromised by simple post-processing operations like blurring or noise addition. These approaches often target the byproducts of a specific generation process rather than the fundamental inconsistencies inherent to all synthetic media. To address these vulnerabilities, FALCON-Net is designed to capture more foundational and robust forensic traces that persist even after post-processing.

Inspired by the fact that real images retain subtle device fingerprints due to the non-uniformity in sensor manufacturing processes, we propose the INP module, which extracts high-frequency noise features from the frequency domain to highlight the absence of intrinsic noise pattern in generated images. By analyzing and enhancing high-frequency components, the INP module captures discriminative features that reflect the imaging processes of real sensors, effectively distinguishing them from synthetic images. On the other hand, real images exhibit complex pixel intensity relationships in local regions, which reflect natural randomness and diverse visual patterns. In contrast, AI-generated images often show uniformity and reduced variation due to the oversimplified mechanisms of generative models. To address this, we propose the LVP module, which models these local relationships using directional encoding to identify anomalies in pixel intensity distribution. This module effectively highlights the lack of natural complexity in synthetic images, making it easier to differentiate between real and generated images. These two types of features reveal the differences between generated and real images from the perspectives of device fingerprints and pixel distribution patterns. The extracted INP and LVP features are then concatenated and fed into the classifier, which combines local cues to complete the AI-generated image detection task, the proposed classifier adopts a pruned ResNet as the backbone, significantly reducing the number of parameters and computational complexity without compromising feature extraction capabilities.

### B. Intrinsic Noise Pattern Isolation Module

The unique noise patterns of imaging sensors, caused by the non-uniformity of the sensor manufacturing process, result in subtle pixel-level intensity variations in real images. These noise patterns, primarily distributed in the high-frequency regions of an image, can be regarded as a type of device fingerprint. Real images captured by physical imaging devices inherently carry this device fingerprint, reflecting the stochastic and device-specific characteristics of the sensor. In contrast, AI-generated images bypass the physical imaging process and lack the intrinsic noise characteristics introduced by such imaging devices. Although generative models may introduce pseudo-random high-frequency noise to enhance visual realism, these synthetic noise patterns are fundamentally different from the unique device fingerprints found in real images. This discrepancy arises because the optimization process of current generative models is dominated by high-energy semantic and structural components, making it difficult to capture the extremely low-magnitude, stochastic sensor noise inherent to physical hardware. Furthermore, the mathematical generation process bypasses the physical acquisition stage, thereby failing to introduce the intrinsic manufacturing imperfections found in real sensors. Even with advancements in generative technologies, accurately simulating such device-specific fingerprints remains challenging, as it requires replicating the non-uniform, stochastic patterns inherent to the sensor manufacturing process. Figure 2 shows the device fingerprint information between the real image and the generated image,

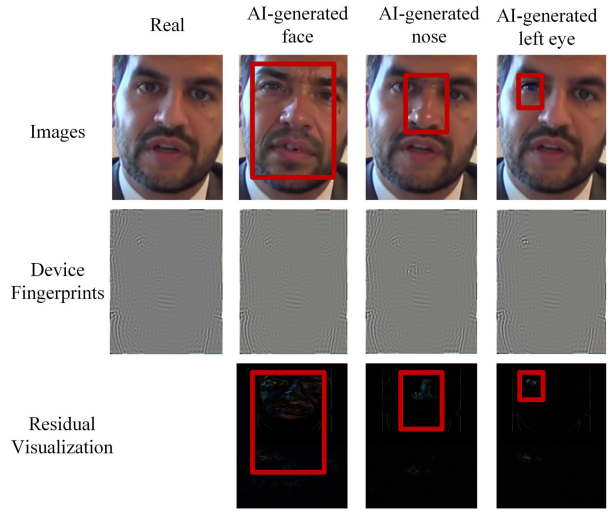



Fig. 2. Visualization diagram of the difference in device fingerprints between real images and AI-generated images.

and obtains the residual visualization results between them. It can be found that the device fingerprint in the real image is significantly different from the random device fingerprint introduced in the generation process of the AI-generated image. Thus, by leveraging this fundamental difference, we can effectively distinguish between real and generated images.

To effectively extract device fingerprint information, we design the Intrinsic Noise Pattern Isolation (INP) module. This module transforms the input image into the frequency domain via a two-dimensional Fast Fourier Transform (FFT) to yield the raw frequency spectrum $Q_{\text{raw}}(u, v)$, where $(u, v)$ represent the frequency coordinates. Since device fingerprints are predominantly embedded in high-frequency components, a two-step filtering strategy is employed to purify these key features. The process begins with the application of a fixed central low-pass suppression mask $M_{\text{fixed}}(u, v)$, to filter out strong and content-related low-frequency components. This mask zeroes out a predefined rectangular area at the center of the spectrum, which is defined as follows,

$$M_{\text{fixed}}(u, v) = \begin{cases} 0, & \text{if } (u, v) \in \text{central region} \\ 1, & \text{otherwise} \end{cases} \tag{1}$$

Applying this mask to the raw spectrum produces the masked frequency spectrum, $Q_{\text{masked}}(u, v)$, through the element-wise multiplication,

$$Q_{\text{masked}}(u, v) = Q_{\text{raw}}(u, v) \cdot M_{\text{fixed}}(u, v) \tag{2}$$

Subsequently, to enhance the signal-to-noise ratio and precisely isolate the fingerprint features, an adaptive thresholding mechanism is applied to $Q_{\text{masked}}(u, v)$. This involves computing the global mean $\mu_{\text{masked}}$ and standard deviation $\sigma_{\text{masked}}$ of its magnitudes,

$$\mu_{\text{masked}} = \frac{1}{H \cdot W} \sum_{u=1}^{H} \sum_{v=1}^{W} |Q_{\text{masked}}(u, v)| \tag{3}$$

$$\sigma_{\text{masked}} = \sqrt{\frac{1}{H \cdot W} \sum_{u=1}^{H} \sum_{v=1}^{W} (|Q_{\text{masked}}(u, v)| - \mu_{\text{masked}})^2} \tag{4}$$

where $H$ and $W$ are the spectrum's dimensions and $|Q_{\text{masked}}(u, v)|$ is the magnitude of a frequency coefficient. An adaptive threshold $T$ is then computed as,

$$T = \mu_{\text{masked}} + \alpha \cdot \sigma_{\text{masked}} \tag{5}$$

where $\alpha$ is a scalar hyperparameter that controls the sensitivity of the threshold. This adaptive threshold design provides inherent robustness against common post-processing operations such as gaussian blurring and image resizing. This robustness stems from the dynamic nature of the threshold $T$. Operations like gaussian blurring function as low-pass filters, while resizing (specifically downsampling) discards high-frequency information. Both operations attenuate the high-frequency components where device fingerprints reside. This attenuation directly reduces the overall energy in the masked spectrum, causing a decrease in both its global mean ($\mu_{\text{masked}}$) and standard deviation ($\sigma_{\text{masked}}$). As a result, the threshold $T$ dynamically adjusts downward, allowing the module to maintain sensitivity and effectively capture the weakened but still present fingerprint patterns from the degraded signal. The procedure continues by utilizing the threshold $T$ to filter the high-frequency coefficients, generating the enhanced frequency spectrum $Q_{\text{enhanced}}(u, v)$. Only coefficients with magnitudes exceeding the threshold are preserved,

$$Q_{\text{enhanced}}(u, v) = \begin{cases} Q_{\text{masked}}(u, v), & \text{if } |Q_{\text{masked}}(u, v)| > T \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

The final step involves transforming the enhanced spectrum $Q_{\text{enhanced}}(u, v)$ back to the spatial domain via an Inverse Fast Fourier Transform (IFFT) to produce the final Intrinsic Noise Pattern (INP) feature map,

$$\text{INP}(x, y) = \text{IFFT}(Q_{\text{enhanced}}(u, v)) \tag{7}$$

where $(x, y)$ are the spatial coordinates. This feature map accurately captures the sensor-specific noise patterns, purified through the two-step process, providing highly discriminative features for distinguishing between real and generated images.

### C. Local Variation Pattern Module

Local regions in real images typically exhibit complex pixel distributions and diverse variation patterns. However, due to the inherent limitations of image generators, the pixel distribution of generated images tends to be more uniform. Inspired by this observation, we propose a local variation pattern (LVP) module based on pixel intensity relationships, which aims to describe the relative variations between local pixels and reveal potential anomalies in generated images.

For each pixel $P_c$ in the input image $X(B \times C \times H \times W)$, we define a $3 \times 3$ local neighborhood window centered on $P_c$, which includes the pixel itself and its 8 surrounding neighboring pixels. To capture the local variation, we compute the intensity differences between the central pixel $P_c$ and each of its neighboring pixels. The calculation is expressed as follows,

$$\Delta I(P_c, P_n) = I(P_c) - I(P_n), \quad P_n \in \mathcal{N}(P_c) \tag{8}$$

where $I(P_c)$ and $I(P_n)$ represent the intensities of the central pixel $P_c$ and a neighboring pixel $P_n$, respectively. $\mathcal{N}(P_c)$ denotes the set of all eight neighboring pixels around $P_c$. $\Delta I(P_c, P_n)$ represents the intensity difference between the

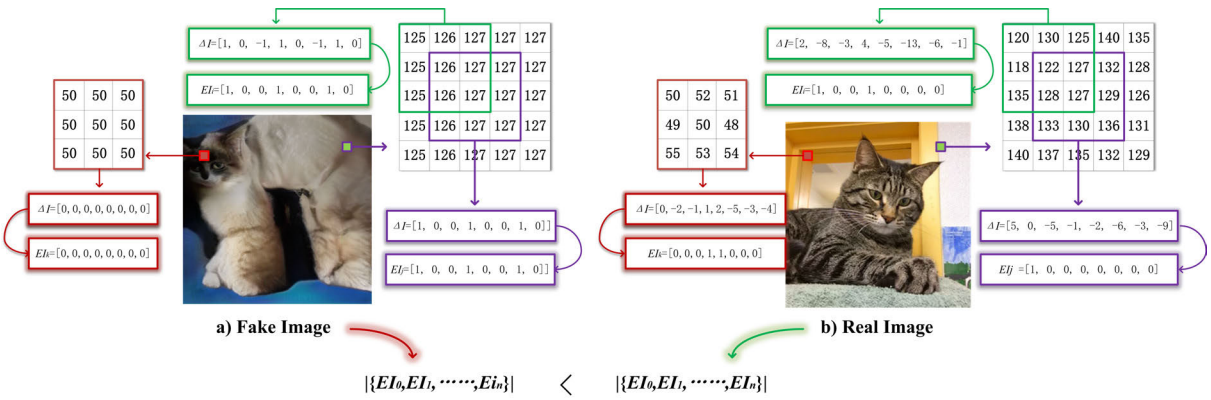



Fig. 3. Comparison of directional pixel intensity encoding patterns between real and generated images, a) is fake image generated from real image b).

central pixel and its neighboring pixels. Each difference value $\Delta I(P_c, P_n)$ is converted into a symbolic directional code to represent the variation trend of the neighboring pixel intensity relative to the central pixel.

This abstraction transforms the relationship between the central pixel and its neighboring pixels into a binary pattern, inherently reflecting the complexity and diversity of the local region. As illustrated in Fig. 3, due to the inherent limitations of the generator, local regions of the generated image lack the randomness observed in real scenarios, leading to highly similar directional encoding. In extreme cases, when large local regions in a generated image consist of identical pixel values, the directional encoding may exhibit fixed patterns, such as all zeros or all ones. Conversely, real images show highly complex textures and edges, producing more diverse binary patterns in directional encoding. As a result, the pattern complexity of generated images is significantly lower compared to that of real images. Let $E_i(p)$ denote the directional encoding of the $i^{th}$ neighboring pixel relative to pixel $p$, which maps the complexity of these patterns.

$$E_i(p) = \begin{cases} 1, & \Delta I(P_c, P_n) > 0 \\ 0, & \Delta I(P_c, P_n) \leq 0 \end{cases} \tag{9}$$

To generate the local variation pattern feature, we assign a unique weight to each direction and perform a weighted aggregation of the directional encoding for pixel $p$. To ensure the uniqueness of the feature and avoid confusion in the directional encoding results, we randomly select eight distinct real numbers $W_i$ $(i = 0, 1, \ldots, 7)$ as weights. This weight design ensures that each direction has a unique and non-repeating weight, and that there is no ambiguity between adjacent directions. Consequently, the unique feature can preserve the variation information of each direction completely.

$$LVP(p) = \sum_{i=0}^{N} E_i(p) \times W_i(N = 7) \tag{10}$$

In real images, directional variation patterns are typically highly diverse. The relative intensity relationships in different directions form a wide range of combinations, resulting in a broad distribution of feature values. While in generated images, variation patterns across multiple directions tend to be uniform, leading to a loss of directional information and producing feature values with a narrow or overly simplistic distribution. Thus, the local variation pattern feature value, designed with unique weights, effectively reveals anomalous pattern differences between real and generated images.

By repeating the above steps for each pixel $p$ in the input image $X$, the LVP feature can be extracted, which effectively encodes the variation patterns within local regions of the image, providing a reliable basis for analyzing the differences between real and generated images.

Other methods focus on extracting pixel features, such as central differential convolution (CDC) [31], which combines pixel intensity and gradient information to capture local gradient variations and enhance the detection of forged features. However, CDC focuses more on pixel-level gradient variations and lacks the ability to model high-dimensional directional relationships between pixels. Therefore, it has certain limitations in dealing with complex and diverse local anomalies in generated images. In contrast, traditional local binary pattern (LBP) relies primarily on simple threshold comparisons of pixel intensities to encode local texture features, but lacks the ability to capture diverse and fine-grained forged features. Compared to the above two methods, LVP uses directional encoding and weighted aggregation to more accurately capture the directional characteristics of pixel distribution. By introducing a unique weight assignment mechanism, it not only enhances sensitivity to local anomaly patterns but also significantly improves the robustness of feature representation. It excels at fine-grained feature modeling and generalizes well to complex generated images.

### D. Classifier

The classifier utilizes the pruned ResNet [32] as the backbone, where structural optimizations significantly reduce network parameters and computational complexity while preserving efficient feature extraction capabilities. During the feature processing stage, LVP and INP features are integrated through feature-level concatenation (concat), capturing both local variation patterns and global feature correlations. Subsequently, a Global Average Pooling (GAP) layer is employed to compress spatial information effectively, generating a compact global feature representation. Finally, the fully connected layer maps the high-dimensional features into a binary classification space to predict whether the input image is generated or real.

## IV. Experiments

### A. Experiment Setup

*1) Training Dataset:* In order to maintain consistency in our evaluation, we use the training set of ForenSynths [18] to train our model. Based on previous research [33], we select four different categories of this training set (cars, cats, chairs, and horses), each of which contains 18,000 synthetic images generated by ProGAN, as well as an equal number of real images selected from the LSUN [34] dataset of an equal number of real images.

*2) Testing Dataset:* To evaluate the generalization ability of the proposed method in real-world scenarios, we use the following four testing datasets, which consists of various real images, diverse GAN and Diffusion generated images.

- **The ForenSynths dataset** [18] includes images generated by eight models (ProGAN, StyleGAN, StyleGAN2, BigGAN, CycleGAN, StarGAN, GauGAN, and DeepFake) along with their corresponding real images.
- **The DiffusionForensics dataset** [29] contains images generated by eight diffusion models (ADM, DDPM, IDDPM, LDM, PNDM, VQ Diffusion, Stable Diffusion v1 (Sdv1), and Stable Diffusion v2 (Sdv2)), with real images sampled from the LSUN and ImageNet [35] datasets.
- **The Ojha dataset** [28] includes images generated by ADM, Glide, DALL-Emini, and LDM, with real images sourced from the LAION [36] and ImageNet datasets.
- **The Self-Synthesized dataset** [33] contains images generated through 1,000 diffusion steps (DDPM, IDDPM, and ADM) as well as synthetic content generated by Midjourney and DALLE [37], collected from the social platform Discord.

*3) Baseline:* We compare our FALCON-Net with SOTA methods, including NPR [33], MI_Net [38], CNNDetection [18], Frank et al. [39], Durall et al. [40], Patchfor [41], F3Net [42], SelfBland [43], GANdetection [44], BiHPF [45], FrePGAN [46], LGrad [19], FreqNet [47], Ojha et al. [28], FatFformer [48], SAFE [49], AIDE [50], DIRE [29], and DRCT [51]. In the experiments, we reimplement the baseline NPR [33], MI_Net [38], FreqNet [19], FatFormer [48], SAFE [49], AIDE [50], DIRE [29], DRCT [51], and C2P-CLIP [9] using its official code, since the C2P-CLIP method does not provide training code, we utilize the officially provided pre-trained weights for subsequent experiments, while other data are obtained from NPR [33].

*4) Implemental Details:* We implement the proposed FALCON-Net using the PyTorch framework and train it on an NVIDIA 3060Ti GPU. The network is trained end-to-end using the Adam optimizer with binary cross-entropy as the loss function. The learning rate is set to 0.0002, the batch size is 32, and the model is trained for 40 epochs. We use accuracy (ACC) and average precision (AP) as evaluation metrics.

### B. Ablation Study

*1) Verification of the Effectiveness of Each Module:* To evaluate the contribution of each module, we conduct experiments where the INP and LVP modules are used independently for generated image detection. We perform our evaluation on 29 sub-testsets, and as shown in Table I, the complete FALCON-Net achieves significantly better performance compared to the standalone modules. Specifically, compared to the w/o INP configuration, FALCON-Net improves the average ACC by 7.7% and AP by 5.9%. Similarly, compared to the w/o LVP configuration, FALCON-Net achieves an average improvement of 7.3% in ACC and 6.8% in AP across all datasets. These results highlight the complementary nature of the INP and LVP modules in capturing different aspects of anomalies in generated images.

TABLE I

Ablation Study of FALCON-Net on 29 Subsets Across Four Datasets (ForenSynths, DiffusionForensics, Ojha, and Self-Synthesized). w/o LVP and w/o INP Denote the Network Without the Local Variation Pattern (LVP) and Intrinsic Noise Pattern Isolation (INP) Modules, Respectively. For Comparison, FALCON-Net (CDC), (LBP), (CLBP), (RI-LBP) and (CS-LBP) Refer to Variants Where the LVP Module Is Replaced by the Central Difference Convolution, Local Binary Pattern, Completed Local Binary Pattern, Rotation Invariant LBP and Center-Symmetric LBP module, respectively

| Method | 29-subsets datasets | |
|---|---|---|
| | Mean ACC | Mean AP |
| **w/o LVP** | **85.9%** | **91.4%** |
| **w/o INP** | **86.3%** | **90.5%** |
| **FALCON-Net (CDC)** | **79.8%** | **85.4%** |
| **FALCON-Net (LBP)** | **84.9%** | **93.3%** |
| **FALCON-Net (CLBP)** | **91.6%** | **95.4%** |
| **FALCON-Net (RI-LBP)** | **90.3%** | **94.8%** |
| **FALCON-Net (CS-LBP)** | **78.6%** | **83.7%** |
| **FALCON-Net** | **93.6%** | **97.3%** |

The INP module focuses on extracting inherent noise characteristics of the sensor using frequency domain separation techniques. This enables the INP module to accurately locate pixel-level intensity fluctuations caused by the non-uniformity of the sensor manufacturing process, effectively highlighting the device fingerprint anomalies present in real images. However, relying solely on the INP module limits the model's ability to capture broader or more diverse forgery cues, such as pixel coding patterns. The LVP module, on the other hand, captures relative variation patterns between pixels through directional encoding, revealing the lack of natural complexity and diversity in generated images. Nevertheless, using only the LVP module fails to account for prominent anomalies in the high-frequency regions of generated images. By integrating both INP and LVP features, FALCON-Net captures differences between generated and real images from both edge-texture features and pixel distribution patterns, achieving significantly higher detection accuracy.

*2) Verification of the Effectiveness of Local Detail Features:* To further verify the adaptive local feature extraction capability of the LVP module, we design five alternative versions for comparison, FALCON-Net (CDC), FALCON-Net (LBP), FALCON-Net (CLBP), FALCON-Net (RI-LBP) and FALCON-Net (CS-LBP). In FALCON-Net (CDC), the LVP branch is replaced with the central difference convolution (CDC) [31] module. CDC combines pixel intensity information and gradient information to enhance local detail features through central difference operations. In FALCON-Net (LBP),

TABLE II

CROSS-GAN SOURCES EVALUATION ON THE FORENSYNTHS DATASET. THE BEST PERFORMANCE IS HIGHLIGHTED USING BOLD AND UNDERLINED TEXT, WHILE THE SECOND-BEST PERFORMANCE IS HIGHLIGHTED USING BOLD TEXT

| Method | ProGAN | | StyleGAN | | StyleGAN2 | | BigGAN | | CycleGAN | | StarGAN | | GauGAN | | DeepFake | | Mean | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ACC | AP | ACC | AP | ACC | AP | ACC | AP | ACC | AP | ACC | AP | ACC | AP | ACC | AP | ACC | AP |
| **CNNDetection (CVPR'20)** | 91.4 | 99.4 | 63.8 | 91.4 | 76.4 | 97.5 | 52.9 | 73.3 | 72.7 | 88.6 | 63.8 | 90.8 | 63.9 | 92.2 | 51.7 | 62.3 | 67.1 | 86.9 |
| **Frank (PRML'20)** | 90.3 | 85.2 | 74.5 | 72.0 | 73.1 | 71.4 | 88.7 | 86.0 | 75.5 | 71.2 | 99.5 | 99.5 | 60.2 | 77.4 | 60.7 | 49.1 | 78.9 | 76.5 |
| **Durall (CVPR'20)** | 81.1 | 74.4 | 54.4 | 52.6 | 66.8 | 62.0 | 60.1 | 56.3 | 69.0 | 64.0 | 98.1 | 98.1 | 61.9 | 57.4 | 50.2 | 50.0 | 67.7 | 64.4 |
| **Patchfor (ECCV'20)** | 97.8 | 100.0 | 82.6 | 93.1 | 83.6 | 98.5 | 64.7 | 69.5 | 74.5 | 87.2 | 100.0 | 100.0 | 57.2 | 55.4 | 85.0 | 93.2 | 80.7 | 87.1 |
| **F3Net (ECCV'20)** | 99.4 | 100.0 | 92.6 | 99.7 | 88.0 | 99.8 | 65.3 | 69.9 | 76.4 | 84.3 | 100.0 | 100.0 | 58.1 | 56.7 | 63.5 | 78.8 | 80.4 | 86.2 |
| **SelfBland (CVPR'22)** | 58.8 | 65.2 | 50.1 | 47.7 | 48.6 | 47.4 | 51.1 | 51.9 | 59.2 | 65.3 | 74.5 | 89.2 | 59.2 | 65.5 | 93.8 | 99.3 | 61.9 | 66.4 |
| **GANDetection (ICIP'22)** | 82.7 | 95.1 | 74.4 | 92.9 | 69.9 | 87.9 | 76.3 | 89.9 | 85.2 | 95.5 | 68.8 | 99.7 | 61.4 | 75.8 | 60.0 | 83.9 | 72.3 | 90.1 |
| **BiHPF (WACV'22)** | 90.7 | 86.2 | 76.9 | 75.1 | 76.2 | 74.7 | 84.9 | 81.7 | 81.9 | 78.9 | 94.4 | 94.4 | 69.5 | 78.1 | 54.4 | 54.6 | 78.6 | 77.9 |
| **FrePGAN (AAAI'22)** | 99.0 | 99.9 | 80.7 | 89.6 | 84.1 | 98.6 | 69.2 | 71.1 | 71.1 | 74.4 | 99.9 | 100.0 | 60.3 | 71.7 | 70.9 | 91.9 | 79.4 | 87.2 |
| **LGrad (CVPR'23)** | 99.9 | 100.0 | 94.8 | 99.9 | 96.0 | 99.9 | 82.9 | 90.7 | 85.3 | 94.0 | 99.6 | 100.0 | 72.4 | 79.3 | 58.0 | 67.9 | 86.1 | 91.5 |
| **Ojha (CVPR'23)** | 99.7 | 100.0 | 89.0 | 98.7 | 83.9 | 98.4 | 90.5 | 99.1 | 87.9 | 99.8 | 91.4 | 100.0 | 89.9 | 100.0 | 80.2 | 90.2 | 89.1 | 98.3 |
| **DIRE (ICCV'23)** | 99.3 | 99.9 | 90.9 | 99.5 | 87.2 | 99.5 | 79.7 | 86.4 | 84.5 | 92.4 | 99.9 | 100.0 | 82.9 | 92.6 | 68.7 | 77.9 | 86.7 | 93.6 |
| **MI_Net (AAAI'24)** | 99.2 | 100.0 | 89.1 | 96.1 | 96.9 | 99.7 | 62.8 | 60.6 | 68.6 | 74.5 | 99.7 | 100.0 | 54.7 | 50.6 | 73.5 | 81.5 | 80.6 | 82.9 |
| **NPR (CVPR'24)** | 99.9 | 100.0 | 96.4 | 99.9 | 97.0 | 99.9 | 85.3 | 91.4 | 85.6 | 98.7 | 99.8 | 100.0 | 83.0 | 84.1 | 79.6 | 85.4 | 90.8 | 94.9 |
| **Freqnet (AAAI'24)** | 99.6 | 100.0 | 90.2 | 99.7 | 88.0 | 99.5 | 90.5 | 96.0 | 95.8 | 99.6 | 85.7 | 99.8 | 93.4 | 98.6 | 88.9 | 94.4 | 91.5 | **98.5** |
| **Fatformer (CVPR'24)** | 98.9 | 99.9 | 82.6 | 96.3 | 74.1 | 92.7 | 97.5 | 99.7 | 99.5 | 100.0 | 99.5 | 100.0 | 99.6 | 100.0 | 84.2 | 95.6 | 92.0 | 98.0 |
| **DRCT/Conv-B (ICML'24)** | 99.3 | 99.9 | 80.3 | 98.1 | 98.0 | 71.9 | 96.0 | 80.4 | 95.6 | 94.0 | 98.9 | 71.9 | 93.9 | 99.5 | 51.0 | 78.1 | 80.3 | 95.4 |
| **AIDE (ICLR'25)** | 100.0 | 100.0 | 99.6 | 100.0 | 98.0 | 99.9 | 83.9 | 94.4 | 98.5 | 99.9 | 99.9 | 100.0 | 73.2 | 97.7 | 54.1 | 76.4 | 88.4 | 96.0 |
| **SAFE (KDD'25)** | 99.8 | 100.0 | 95.5 | 99.9 | 99.0 | 100.0 | 88.8 | 96.0 | 91.4 | 95.0 | 99.4 | 99.9 | 94.6 | 98.2 | 73.1 | 81.7 | **92.7** | 96.3 |
| **C2P-CLIP (AAAI'25)** | 97.3 | 100.0 | 93.8 | 98.6 | 100.0 | 100.0 | 99.1 | 100.0 | 95.6 | 99.9 | 96.4 | 99.5 | 99.2 | 100.0 | 99.6 | 100.0 | **97.6** | **99.7** |
| **FALCON-Net (our)** | 99.7 | 100.0 | 97.6 | 99.8 | 97.2 | 99.7 | 90.6 | 96.6 | 93.4 | 97.1 | 100.0 | 100.0 | 90.0 | 97.2 | 60.4 | 76.1 | 91.1 | 95.8 |

the LVP branch is replaced with the local binary patterns (LBP) [52] module, a classical method for extracting local texture features. LBP encodes texture by performing binary comparisons of pixel intensities, but its simple thresholding operation may result in less precise feature extraction, especially when handling diverse and fine-grained forgery patterns in generated images. To provide a more comprehensive comparison against advanced local pattern descriptors, we also implemented FALCON-Net (CLBP), where the LVP module is replaced by the completed local binary pattern (CLBP) [53]. As proposed in its original paper, CLBP is a significant enhancement over the traditional LBP. Unlike LBP, which only considers the sign of pixel differences, CLBP provides a more complete representation by also encoding the magnitude of these differences and the intensity of the center pixel. This allows it to capture not only the local structure but also the local contrast and texture variations, making it a substantially more powerful hand-crafted feature descriptor. To expand the comparison of LBP-family variants, we additionally implemented FALCON-Net (RI-LBP) and FALCON-Net (CS-LBP), where the LVP module is replaced by the Rotation Invariant LBP (RI-LBP) [54] and the Center-Symmetric LBP (CS-LBP) [55], respectively.

As shown in Table I, FALCON-Net (CDC) achieves an average ACC of 79.8% and an AP of 85.4%. FALCON-Net (LBP) achieves an average ACC of 84.9% and an AP of 93.3%. The FALCON-Net (CLBP) variant reaches a notably higher average ACC of 91.6% and an AP of 95.4%. This result significantly surpasses both CDC and LBP, confirms that CLBP is more effective at capturing the subtle inconsistencies in generated images. FALCON-Net (RI-LBP) achieves an average ACC of 90.3% and an AP of 94.8%, while FALCON-Net (CS-LBP) achieves a lower average ACC of 78.6% and an AP of 83.7%. In contrast, the proposed FALCON-Net with our LVP module achieves a significantly higher average ACC of 93.6% and AP of 97.3%, surpassing FALCON-Net (CLBP) by over 2% in average ACC and 1.9% in average AP. These results demonstrate that the design of the LVP module provides significant advantages in capturing subtle and diverse local anomalies in generated images. The LVP module uses directional encoding and weighted aggregation to precisely capture complex directional characteristics of pixel distributions, enhancing sensitivity to local abnormalities. In contrast, while the CDC module integrates intensity and gradient information, it focuses only on pixel-level gradient changes and lacks the ability to model higher-dimensional directional relationships. Similarly, the LBP module can extract basic local texture features but falls short in representing fine-grained and complex patterns due to its simple binary thresholding operation. Even the more advanced CLBP, while powerful, is ultimately outperformed by our LVP module. This suggests that while sophisticated hand-crafted features like CLBP are strong baselines, the LVP module's approach of using weighted aggregation of directional encodings provides a more flexible and adaptive representation, better suited for learning the specific, high-dimensional relationships that characterize AI-generated artifacts.

Consequently, the combination of the INP and LVP modules enables FALCON-Net to effectively capture both high-frequency noise features and complex local pixel relationships in generated images, resulting in superior performance compared to other designs.

### C. GAN-Sources Evaluation

To verify the generalization ability of FALCON-Net on GAN-generated images, we evaluate it on the ForenSynths dataset [18]. As shown in Table II, our FALCON-Net performs slightly lower than C2P-CLIP in average ACC, with an average ACC of 97.6%, and with an average AP of 99.7%. Some recent methods may have advantages in dealing with specific forgery features generated by certain GAN models, such as specialization in capturing high-frequency forgery clues or model features. However, this specialization usually weakens their adaptability to a wider range of models. In contrast, the design focus of FALCON-Net is to capture localized abnormal features, thereby achieving stronger cross-model generalization capabilities. This versatility enables FALCON-Net to still achieve robust performance when dealing with more unseen models. Although the detection performance on the ForenSynths dataset is slightly inferior to that of the C2P-CLIP, the average ACC and AP of FALCON-Net have reached ideal values compared with several state-of-the-art methods. These results still show that FALCON-Net exhibits strong generalization capabilities in different GAN models and can effectively reveal the essential difference between generated images and real images.

TABLE III

CROSS-DIFFUSION SOURCES EVALUATION ON THE DIFFUSIONFORENSICS DATASET. THE BEST PERFORMANCE IS HIGHLIGHTED USING BOLD AND UNDERLINED TEXT, WHILE THE SECOND-BEST PERFORMANCE IS HIGHLIGHTED USING BOLD TEXT

| Method | ADM | | DDPM | | IDDPM | | LDM | | PNDM | | VQ-Diffusion | | Sdv1 | | Sdv2 | | Mean | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ACC | AP | ACC | AP | ACC | AP | ACC | AP | ACC | AP | ACC | AP | ACC | AP | ACC | AP | ACC | AP |
| CNNDetection (CVPR'20) | 53.9 | 71.8 | 62.7 | 76.6 | 50.2 | 82.7 | 50.4 | 78.7 | 50.8 | 90.3 | 50.0 | 71.0 | 38.0 | 76.7 | 52.0 | 90.3 | 51.0 | 79.8 |
| Frank (PRML'20) | 58.9 | 65.9 | 37.0 | 27.6 | 51.4 | 65.0 | 51.7 | 48.5 | 44.0 | 38.2 | 51.7 | 66.7 | 32.8 | 52.3 | 40.8 | 37.5 | 46.0 | 50.2 |
| Durall (CVPR'20) | 39.8 | 42.1 | 52.9 | 49.8 | 55.3 | 56.7 | 43.1 | 39.9 | 44.5 | 47.3 | 38.6 | 38.3 | 39.5 | 56.3 | 62.1 | 55.8 | 47.0 | 48.3 |
| Patchfor (ECCV'20) | 77.5 | 93.9 | 62.3 | 97.1 | 50.0 | 91.6 | 99.5 | 100.0 | 50.2 | 99.9 | 100.0 | 100.0 | 90.7 | 99.8 | 94.8 | 100.0 | 78.1 | 97.8 |
| F3Net (ECCV'20) | 80.9 | 96.9 | 84.7 | 99.4 | 74.7 | 98.9 | 100.0 | 100.0 | 72.8 | 99,5 | 100.0 | 100.0 | 73.4 | 97.2 | 99.8 | 100.0 | 85.8 | 99.0 |
| SelfBland (CVPR'22) | 57.0 | 59.0 | 61.9 | 49.6 | 63.2 | 66.9 | 83.3 | 92.2 | 48.2 | 48.2 | 77.2 | 82.7 | 46.2 | 68.0 | 71.2 | 73.9 | 63.5 | 67.6 |
| GANDetection (ICIP'22) | 51.1 | 53.1 | 62.3 | 46.4 | 50.2 | 63.0 | 51.6 | 48.1 | 50.6 | 79.0 | 51.1 | 51.2 | 39.8 | 65.6 | 50.1 | 36.9 | 50.8 | 55.4 |
| LGrad (CVPR'23) | 86.4 | 97.5 | 99.9 | 100.0 | 66.1 | 92.8 | 99.7 | 100.0 | 69.5 | 98.5 | 96.2 | 100.0 | 90.4 | 99.4 | 97.1 | 100.0 | 88.2 | 98.5 |
| Ojha (CVPR'23) | 78.4 | 92.1 | 72.9 | 78.8 | 75.0 | 92.8 | 82.2 | 97.1 | 75.3 | 92.5 | 83.5 | 97.7 | 56.4 | 90.4 | 71.5 | 92.4 | 74.4 | 91.7 |
| DIRE (ICCV'23) | 64.8 | 65.3 | 77.0 | 65.1 | 67.8 | 68.5 | 89.2 | 98.8 | 76.5 | 83.5 | 57.5 | 73.0 | 72.7 | 76.0 | 87.6 | 90.3 | 74.1 | 77.6 |
| MI_Net (AAAI'24) | 83.3 | 90.0 | 62.5 | 54.6 | 50.0 | 55.7 | 99.3 | 100.0 | 56.2 | 99.1 | 99.7 | 100.0 | 95.7 | 99.6 | 99.2 | 100.0 | 80.7 | 87.4 |
| NPR (CVPR'24) | 86.3 | 98.0 | 90.7 | 98.2 | 84.3 | 92.2 | 100.0 | 100.0 | 93.7 | 100.0 | 96.6 | 99.9 | 98.9 | 100.0 | 100.0 | 100.0 | **93.8** | 98.5 |
| FreqNet (AAAI'24) | 67.2 | 93.2 | 91.4 | 99.9 | 59.0 | 98.3 | 98.9 | 100.0 | 85.2 | 99.9 | 100.0 | 100.0 | 63.9 | 98.6 | 81.8 | 98.7 | 80.9 | 98.6 |
| FatFormer (CVPR'24) | 70.7 | 93.7 | 67.2 | 78.9 | 69.3 | 96.3 | 97.3 | 100.0 | 99.3 | 100.0 | 100.0 | 100.0 | 61.7 | 97.0 | 84.3 | 98.7 | 81.2 | 95.6 |
| DRCT/Conv-B (ICML'24) | 51.9 | 62.9 | 66.1 | 84.4 | 55.0 | 88.4 | 50.2 | 65.2 | 67.3 | 96.3 | 53.5 | 85.8 | 40.0 | 78.5 | 52.8 | 92.4 | 54.6 | 81.7 |
| AIDE (ICLR'25) | 85.6 | 97.3 | 88.1 | 97.8 | 87.1 | 99.1 | 99.2 | 100.0 | 97.8 | 99.9 | 93.7 | 99.6 | 95.7 | 99.5 | 99.5 | 100.0 | 93.3 | 99.2 |
| SAFE (KDD'25) | 77.5 | 97.0 | 99.8 | 100.0 | 95.8 | 100.0 | 100.0 | 100.0 | 92.9 | 100.0 | 84.1 | 99.4 | 89.8 | 100.0 | 100.0 | 100.0 | 92.5 | **99.6** |
| C2P-CLIP (AAAI'25) | 68.8 | 95.3 | 73.5 | 76.2 | 80.7 | 94.9 | 97.3 | 99.7 | 84.2 | 97.2 | 78.9 | 99.2 | 66.7 | 94.8 | 95.8 | 99.7 | 80.7 | 94.6 |
| FALCON-Net (our) | 94.0 | 98.9 | 99.7 | 100.0 | 99.7 | 100.0 | 99.8 | 100.0 | 99.7 | 100.0 | 90.3 | 99.1 | 99.4 | 100.0 | 99.8 | 100.0 | **97.8** | **99.7** |

TABLE IV

CROSS-DIFFUSION SOURCES EVALUATION ON THE OJHA DATASET. THE BEST PERFORMANCE IS HIGHLIGHTED USING BOLD AND UNDERLINED TEXT, WHILE THE SECOND-BEST PERFORMANCE IS HIGHLIGHTED USING BOLD TEXT

| Method | DALLE | | Glide_100_10 | | Glide_100_27 | | Glide_50_27 | | ADM | | LDM_100 | | LDM_200 | | LDM_200_cfg | | Mean | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ACC | AP | ACC | AP | ACC | AP | ACC | AP | ACC | AP | ACC | AP | ACC | AP | ACC | AP | ACC | AP |
| CNNDetection (CVPR'20) | 51.8 | 61.3 | 53.3 | 72.9 | 53.0 | 71.3 | 54.2 | 76.0 | 54.9 | 66.6 | 51.9 | 63.7 | 52.0 | 64.5 | 51.6 | 63.1 | 52.8 | 67.4 |
| Frank (PRML'20) | 57.0 | 62.5 | 53.6 | 44.3 | 50.4 | 40.8 | 52.0 | 42.3 | 53.4 | 52.5 | 56.6 | 51.3 | 56.4 | 50.9 | 56.5 | 52.1 | 54.5 | 49.6 |
| Durall (CVPR'20) | 55.9 | 58.0 | 54.9 | 52.3 | 48.9 | 46.9 | 51.7 | 49.9 | 40.6 | 42.3 | 62.0 | 62.6 | 61.7 | 61.7 | 58.4 | 58.5 | 54.3 | 54.0 |
| Patchfor (ECCV'20) | 79.8 | 99.1 | 87.3 | 99.7 | 82.8 | 99.1 | 84.9 | 98.8 | 74.2 | 81.4 | 95.8 | 99.8 | 95.6 | 99.9 | 94.0 | 99.8 | 86.8 | 97.2 |
| F3Net (ECCV'20) | 71.6 | 79.9 | 88.3 | 95.4 | 87.0 | 94.5 | 88.5 | 95.4 | 69.2 | 70.8 | 74.1 | 84.0 | 73.4 | 83.3 | 80.7 | 89.1 | 79.1 | 86.5 |
| SelfBland (CVPR'22) | 52.4 | 51.6 | 58.8 | 63.2 | 59.4 | 64.1 | 64.2 | 68.3 | 58.3 | 63.4 | 53.0 | 54.0 | 52.6 | 51.9 | 51.9 | 52.6 | 56.3 | 58.7 |
| GANDetection (ICIP'22) | 67.2 | 83.0 | 51.2 | 52.6 | 51.1 | 51.9 | 51.7 | 53.5 | 49.6 | 49.0 | 54.7 | 65.8 | 54.9 | 65.9 | 53.8 | 58.9 | 54.3 | 60.1 |
| LGrad (CVPR'23) | 88.5 | 97.3 | 89.4 | 94.9 | 87.4 | 93.2 | 90.7 | 95.1 | 86.6 | 100.0 | 94.8 | 99.2 | 94.2 | 99.1 | 95.9 | 99.2 | 90.9 | 97.2 |
| Ojha (CVPR'23) | 89.5 | 96.8 | 90.1 | 97.0 | 90.7 | 97.2 | 91.1 | 97.4 | 75.7 | 85.1 | 90.5 | 97.0 | 90.2 | 97.1 | 77.3 | 88.6 | 86.9 | 94.5 |
| DIRE (ICCV'23) | 80.3 | 83.4 | 80.6 | 82.8 | 80.5 | 79.4 | 80.9 | 79.4 | 73.9 | 74.4 | 80.0 | 83.4 | 79.7 | 82.8 | 80.5 | 82.4 | 79.5 | 81.0 |
| MI_Net (AAAI'24) | 63.3 | 73.6 | 80.8 | 90.0 | 78.5 | 88.0 | 82.0 | 90.3 | 69.5 | 75.5 | 81.3 | 90.5 | 81.2 | 89.9 | 80.0 | 89.3 | 77.1 | 85.9 |
| NPR (CVPR'24) | 75.6 | 98.3 | 96.6 | 99.8 | 95.8 | 99.8 | 99.8 | 99.6 | 76.0 | 87.5 | 95.4 | 99.9 | 95.2 | 99.9 | 95.5 | 99.8 | 90.8 | 98.1 |
| FreqNet (AAAI'24) | 97.2 | 99.7 | 87.8 | 96.0 | 84.4 | 95.6 | 86.6 | 95.8 | 67.2 | 75.4 | 97.8 | 99.9 | 97.4 | 99.9 | 97.2 | 99.9 | 89.4 | 95.3 |
| FatFormer (CVPR'24) | 98.7 | 99.8 | 94.2 | 99.3 | 94.3 | 99.3 | 94.6 | 99.5 | 76.0 | 91.9 | 98.6 | 99.9 | 98.6 | 99.8 | 94.9 | 99.2 | 93.7 | 98.6 |
| DRCT/Conv-B (ICML'24) | 56.5 | 69.2 | 55.6 | 72.5 | 56.5 | 72.5 | 58.3 | 76.0 | 53.8 | 64.7 | 59.3 | 74.4 | 59.9 | 75.0 | 54.4 | 64.8 | 56.8 | 71.1 |
| AIDE (ICLR'25) | 97.5 | 99.0 | 97.8 | 99.2 | 98.1 | 99.3 | 98.4 | 99.2 | 88.5 | 97.3 | 98.6 | 99.3 | 98.2 | 99.3 | 97.4 | 99.1 | **96.8** | **99.0** |
| SAFE (KDD'25) | 82.2 | 99.6 | 95.2 | 99.8 | 94.5 | 99.8 | 94.4 | 99.8 | 74.9 | 88.7 | 92.2 | 99.9 | 92.3 | 99.9 | 95.0 | 99.9 | 90.1 | 98.4 |
| C2P-CLIP (AAAI'25) | 98.6 | 99.9 | 95.3 | 99.8 | 96.1 | 99.8 | 95.3 | 99.7 | 69.1 | 94.1 | 99.3 | 100.0 | 99.3 | 100.0 | 97.3 | 99.8 | 93.8 | **99.2** |
| FALCON-Net (our) | 97.0 | 99.6 | 94.0 | 97.4 | 96.2 | 99.1 | 92.4 | 96.8 | 87.4 | 94.9 | 97.8 | 99.7 | 97.0 | 99.6 | 97.0 | 99.6 | **94.9** | 98.3 |

### D. Diffusion-Sources Evaluation

To evaluate the performance of FALCON-Net on unseen diffusion-generated images, we test it on the DIRE [29] and Wang et al. [29] datasets. From Table III, we can find that FALCON-Net is superior to SOTA methods, with an average ACC at least 4.0% higher than comparison methods. From Table IV, FALCON-Net achieves an average ACC that is at least 1.1% higher than SOTA methods. These results demonstrates that FALCON-Net is superior adaptability and generalization to unseen diffusion models.

To address the limitations of low-resolution and less realistic images generated with fewer diffusion steps in the Ojha dataset, we perform further evaluations on the Self-Synthesis dataset [33], which includes diffusion images generated with 1000 steps. The results in Table V show that FALCON-Net achieves gains of 1.0% and 1.5% in average ACC and AP compared to SAFE [49]. These results demonstrate that even when dealing with generated images through more extended diffusion processes, FALCON-Net maintains exceptional generalization performance.

### E. Recent-AIGI Evaluation

To further evaluate the generalization ability of FALCON-Net on more recent and challenging AIGI benchmarks, we additionally test it on GenImage [56], Chameleon [50], and A-Bench [57]. All experimental settings follow the same setup as described in Sec. IV-A. The results are summarized in Table VI. From Table VI, we can observe that FALCON-Net remains competitive on GenImage, achieving 83.3% ACC and 92.2% AP, and obtaining the best ACC among the compared baselines. On the more challenging Chameleon benchmark, FALCON-Net achieves 60.3% ACC and 51.7% AP, slightly outperforming other methods in terms of ACC, which indicates its robustness under harder-to-detect generation settings. Moreover, on A-Bench, FALCON-Net achieves strong performance (87.9% ACC and 100.0% AP), surpassing the strongest baseline by 3.3% in ACC. Overall, these results demonstrate that FALCON-Net maintains strong adaptability and generalization to newer data distributions and recently emerged generative models.

### F. Comparison of Parameter Quantities

Table VII provides a detailed comparison of the parameter size and performance of our model with other SOTA methods. As shown in the table, the parameter size of our proposed model is only 1.4M, which is significantly smaller than many other approaches, such as MI_Net with 88.0M and Ojha with 304.0M. Despite its much smaller parameter size, our model achieve the highest average ACC and AP across 29 subtestsets, reaching 93.6% and 97.3%, respectively. Compared to the parameter-efficient NPR method (1.4M), our model improves the average ACC by + 3.4% and the average AP

TABLE V

CROSS-DIFFUSION SOURCES EVALUATION ON THE SELF-SYNTHESIS DATASET. THE BEST PERFORMANCE IS HIGHLIGHTED USING BOLD AND UNDERLINED TEXT, WHILE THE SECOND-BEST PERFORMANCE IS HIGHLIGHTED USING BOLD TEXT

| Method | DDPM | | IDDPM | | ADM | | Midjourney | | DALLE | | Mean | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ACC | AP | ACC | AP | ACC | AP | ACC | AP | ACC | AP | ACC | AP |
| **CNNDetection (CVPR'20)** | 50.0 | 63.3 | 48.3 | 52.7 | 53.4 | 64.4 | 48.6 | 38.5 | 49.3 | 44.7 | 49.9 | 52.7 |
| **Frank (PRML'20)** | 47.6 | 43.1 | 70.5 | 85.7 | 67.3 | 72.2 | 39.7 | 40.8 | 68.7 | 65.2 | 58.8 | 61.4 |
| **Durall (CVPR'20)** | 54.1 | 53.6 | 63.2 | 71.7 | 39.1 | 40.8 | 45.7 | 47.2 | 53.9 | 52.2 | 51.2 | 53.1 |
| **Patchfor (ECCV'20)** | 54.1 | 66.3 | 35.8 | 34.2 | 68.6 | 73.7 | 66.3 | 68.8 | 60.8 | 65.1 | 57.1 | 61.6 |
| **F3Net (ECCV'20)** | 59.4 | 71.9 | 42.2 | 44.7 | 73.4 | 80.3 | 73.2 | 80.4 | 79.6 | 87.3 | 65.5 | 72.9 |
| **SelfBland (CVPR'22)** | 55.3 | 57.7 | 63.5 | 62.5 | 57.1 | 60.1 | 54.3 | 56.4 | 48.8 | 47.4 | 55.8 | 56.8 |
| **GANDetection (ICIP'22)** | 47.3 | 45.5 | 47.9 | 57.0 | 51.0 | 56.1 | 50.0 | 44.7 | 49.8 | 49.7 | 49.2 | 50.6 |
| **LGrad (CVPR'23)** | 59.8 | 88.5 | 45.2 | 46.9 | 72.7 | 79.3 | 68.3 | 76.0 | 75.1 | 80.9 | 64.2 | 74.3 |
| **Ojha (CVPR'23)** | 69.5 | 80.0 | 64.9 | 74.2 | 81.3 | 90.8 | 50.0 | 49.8 | 66.3 | 74.6 | 66.4 | 73.9 |
| **DIRE (ICCV'23)** | 77.2 | 70.4 | 86.2 | 91.9 | 73.6 | 82.1 | 59.2 | 64.8 | 56.2 | 61.0 | 70.4 | 74.1 |
| **MI_Net (AAAI'24)** | 50.4 | 52.6 | 60.9 | 68.4 | 62.1 | 63.3 | 41.6 | 43.1 | 42.5 | 49.7 | 51.5 | 55.4 |
| **NPR (CVPR'24)** | 80.4 | 79.3 | 75.2 | 87.8 | 82.2 | 90.0 | 87.7 | 94.6 | 87.4 | 93.4 | 82.6 | 89.0 |
| **FreqNet (AAAI'24)** | 70.9 | 73.9 | 52.3 | 60.4 | 80.6 | 90.5 | 55.5 | 65.3 | 52.9 | 61.7 | 62.4 | 70.4 |
| **FatFormer (CVPR'24)** | 59.1 | 77.9 | 59.4 | 72.5 | 81.8 | 95.7 | 62.9 | 85.4 | 68.8 | 93.2 | 66.4 | 85.0 |
| **DRCT/Conv-B (ICML'24)** | 61.1 | 75.7 | 64.7 | 85.8 | 55.1 | 70.3 | 48.8 | 37.3 | 50.7 | 52.6 | 56.1 | 64.3 |
| **AIDE (ICLR'25)** | 77.1 | 80.0 | 79.8 | 90.7 | 53.4 | 55.7 | 73.7 | 80.1 | 83.1 | 95.9 | 73.7 | 80.1 |
| **SAFE (KDD'25)** | 82.1 | 93.9 | 87.5 | 80.3 | 83.4 | 95.2 | 98.2 | 99.6 | 87.3 | 94.8 | **87.7** | **92.8** |
| **C2P-CLIP (AAAI'25)** | 66.4 | 93.2 | 72.0 | 81.9 | 74.4 | 94.6 | 75.2 | 92.6 | 66.8 | 93.1 | 71.0 | 91.1 |
| **FALCON-Net (our)** | 90.5 | 96.9 | 92.2 | 97.4 | 92.5 | 97.3 | 82.8 | 91.1 | 85.6 | 88.5 | **88.7** | **94.3** |

TABLE VI

CROSS-DIFFUSION SOURCES EVALUATION ON THE RECENT AIGI DATASET. THE BEST PERFORMANCE IS HIGHLIGHTED USING BOLD AND UNDERLINED TEXT, WHILE THE SECOND-BEST PERFORMANCE IS HIGHLIGHTED USING BOLD TEXT

| Method | Genimage | | Chameleon | | A-bench | | Mean | |
|---|---|---|---|---|---|---|---|---|
| | ACC | AP | ACC | AP | ACC | AP | ACC | AP |
| **MI_Net (AAAI'24)** | 54.0 | 60.6 | 56.6 | 29.9 | 81.0 | 100.0 | 63.7 | 63.5 |
| **NPR (CVPR'24)** | 83.0 | 91.8 | 59.4 | 41.6 | 84.6 | 100.0 | **75.7** | 77.8 |
| **FreqNet (AAAI'24)** | 68.3 | 81.2 | 59.1 | 48.2 | 72.1 | 100.0 | 66.5 | 76.5 |
| **FatFormer (CVPR'24)** | 59.5 | 68.6 | 57.1 | 61.7 | 24.4 | 100.0 | 47.0 | 76.8 |
| **SAFE (KDD'25)** | 80.3 | 89.6 | 60.2 | 52.8 | 83.2 | 100.0 | 74.6 | **80.8** |
| **C2P-CLIP (AAAI'25)** | 81.9 | 95.7 | 58.0 | 46.4 | 51.2 | 100.0 | 63.7 | 80.7 |
| **FALCON-Net (our)** | 83.3 | 92.2 | 60.3 | 51.7 | 87.9 | 100.0 | **77.2** | **81.3** |

TABLE VII

PARAMETER COMPARISON OF VARIOUS METHODS ON ALL 29 SUB-TESTSETS. THE BEST PERFORMANCE IS HIGHLIGHTED USING BOLD AND UNDERLINED TEXT, WHILE THE SECOND-BEST PERFORMANCE IS HIGHLIGHTED USING BOLD TEXT

| Methods | Parameters | Mean ACC.of 29 sub-testsets | Mean AP.of 29 sub-testsets |
|---|---|---|---|
| **F3Net (ECCV'20)** | 48.9M | 78.4% | 88.0% |
| **Ojha (CVPR'23)** | 304.0M | 77.3% | 88.5% |
| **LGrad (CVPR'23)** | 46.6M | 83.5% | 92.2% |
| **MI_Net (AAAI'24)** | 88.0M | 72.4% | 79.2% |
| **FreqNet (AAAI'24)** | 1.9M | 83.0% | 92.8% |
| **NPR(CVPR'24)** | **1.4M** | 90.2% | 95.8% |
| **FatFormer (CVPR'24)** | 577.2M | 85.1% | 95.3% |
| **SAFE (KDD'25)** | 1.4M | **91.1%** | **97.2%** |
| **C2P-CLIP (AAAI'25)** | 304M | 87.3% | 96.7% |
| **Ours** | **1.4M** | **93.6%** | **97.3%** |

by + 1.5%. Furthermore, when compared to SAFE, our model outperforms it by + 2.5% in average ACC. These results demonstrate the excellent balance between parameter efficiency and performance achieved by our model. The reduced parameter size of FALCON-Net can be attributed to its lightweight architectural design and efficient feature extraction strategy. Specifically, the model employs a pruned ResNet as the classifier backbone, which significantly trims unnecessary layers and redundant parameters while retaining critical feature extraction capabilities. Moreover, instead of relying on deep and large-scale neural networks, FALCON-Net focuses on extracting discriminative features in the frequency and spatial domains, allowing the model to achieve high accuracy without requiring complex and computationally expensive layers.

### G. Robustness Evaluation

*1) Robustness to Gaussian Blur:* To validate the robustness of FALCON-Net, we evaluate its performance under Gaussian blur operations, using the DiffusionForensics dataset. Unlike traditional methods that struggle with this operation, FALCON-Net consistently demonstrates exceptional robustness and adaptability. We test two Gaussian blur kernel sizes ($7 \times 7$ and $9 \times 9$) and calculate the average performance across both parameters. The results show that FALCON-Net achieves an average accuracy (ACC) of 90.4% and an average precision (AP) of 98.3%. Compared to other methods, FALCON-Net outperforms NPR by 3.3% in ACC and 1. 1% in AP, MI_Net by 17. 2% in ACC and 15.3% in AP, SAFE by 1.2% in ACC and 0.8% in AP, and C2P-CLIP by 32.9% in ACC and 36.7% in AP. These findings highlight FALCON-Net's remarkable robustness in handling blurred images.

*2) Robustness to Resizing:* To validate the robustness of our model when dealing with image resizing, we also evaluate two resizing factors (0.5 and 1.5) and calculate the average performance. FALCON-Net achieves an average ACC of 95.3% and an average AP of 99.2%, outperforming NPR by 5.7% in ACC and 2.3% in AP. Furthermore, it significantly surpasses MI_Net with a 24.1% improvement in ACC and a 21.1% improvement in AP. FALCON-Net also outperforms SAFE by 5.7% in ACC and 1.1% in AP, while exceeding C2P-CLIP by 20.8% in ACC and 6.8% in AP. These results demonstrate FALCON-Net's exceptional adaptability to image resizing operations.

*3) Robustness to JPEG Compression:* In JPEG compression scenarios, FALCON-Net achieves an average ACC of 70.1% and an average AP of 79.1%, outperforming NPR by 6.8% in ACC and 7.7% in AP. Furthermore, FALCON-Net significantly exceeds MI_Net, achieving 16.1% and 16.4% improvements in ACC and AP, respectively. It also outperforms SAFE with a 6.8% increase in ACC and a 10.3% increase in AP. Compared to C2P-CLIP, FALCON-Net

TABLE VIII

ROBUSTNESS EVALUATION ON THE DIFFUSIONFORENSICS DATASET

| Manipulation | Parameter | Method | ADM | | DDPM | | IDDPM | | LDM | | PNDM | | VQ-Diffusion | | Sdv1 | | Sdv2 | | Mean | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | ACC | AP | ACC | AP | ACC | AP | ACC | AP | ACC | AP | ACC | AP | ACC | AP | ACC | AP | ACC | AP |
| Gaussian Blur | 7*7 | NPR (CVPR'24) | 83.3 | 93.9 | 73.8 | 95.1 | 64.4 | 96.9 | 99.3 | 100.0 | 87.2 | 99.5 | 99.8 | 100.0 | 83.7 | 97.5 | 99.4 | 100.0 | 86.4 | 97.9 |
| | | MI_Net (AAAI'24) | 71.5 | 84.1 | 62.8 | 51.8 | 50.5 | 65.6 | 94.7 | 99.6 | 52.6 | 84.3 | 99.4 | 100.0 | 71.2 | 93.6 | 98.4 | 99.9 | 75.1 | 84.9 |
| | | SAFE (KDD'25) | 86.0 | 94.1 | 88.3 | 98.5 | 81.8 | 98.8 | 97.9 | 99.9 | 93.7 | 99.5 | 85.7 | 96.7 | 99.2 | 100.0 | 98.8 | 99.9 | 91.4 | 98.4 |
| | | C2P-CLIP (AAAI'25) | 63.4 | 68.8 | 73.9 | 72.4 | 73.6 | 79.3 | 58.2 | 59.5 | 78.1 | 85.3 | 45.9 | 67.7 | 46.0 | 45.6 | 50.6 | 51.8 | 61.2 | 66.3 |
| | | FALCON-Net (our) | 86.8 | 94.9 | 87.0 | 98.1 | 83.9 | 98.8 | 99.4 | 100.0 | 96.7 | 99.8 | 90.9 | 98.0 | 99.5 | 100.0 | 99.5 | 100.0 | **92.9** | **98.7** |
| | 9*9 | NPR (CVPR'24) | 81.0 | 90.5 | 83.3 | 93.4 | 80.8 | 96.6 | 98.0 | 99.8 | 83.8 | 96.8 | 97.3 | 99.7 | 79.6 | 94.4 | 98.0 | 99.8 | 87.7 | 96.4 |
| | | MI_Net (AAAI'24) | 66.6 | 81.0 | 62.5 | 43.2 | 50.1 | 58.0 | 83.0 | 99.1 | 50.5 | 73.8 | 96.2 | 99.9 | 64.1 | 92.8 | 96.7 | 99.9 | 71.2 | 81.0 |
| | | SAFE (KDD'25) | 80.6 | 90.1 | 82.6 | 94.3 | 79.9 | 96.8 | 93.3 | 99.2 | 84.9 | 97.3 | 80.9 | 94.7 | 97.0 | 99.7 | 95.7 | 99.6 | 86.9 | 96.5 |
| | | C2P-CLIP (AAAI'25) | 58.5 | 59.7 | 67.6 | 59.3 | 65.8 | 68.8 | 46.2 | 47.9 | 70.6 | 75.3 | 39.3 | 59.9 | 41.0 | 41.5 | 40.6 | 42.2 | 53.7 | 56.8 |
| | | FALCON-Net (our) | 82.9 | 93.4 | 84.1 | 96.6 | 79.3 | 98.1 | 98.9 | 99.9 | 78.1 | 97.9 | 81.3 | 96.4 | 99.2 | 99.9 | 99.2 | 100.0 | **87.9** | **97.8** |
| Resizing | 0.5 | NPR (CVPR'24) | 80.7 | 90.4 | 81.4 | 87.4 | 66.5 | 83.7 | 97.9 | 99.8 | 82.2 | 94.6 | 98.2 | 99.9 | 87.9 | 97.3 | 98.0 | 99.9 | 86.6 | 94.1 |
| | | MI_Net (AAAI'24) | 59.2 | 68.8 | 61.8 | 39.8 | 50.1 | 50.8 | 77.8 | 95.4 | 50.7 | 64.0 | 78.2 | 95.6 | 56.5 | 85.7 | 81.3 | 96.1 | 64.5 | 74.5 |
| | | SAFE (KDD'25) | 83.2 | 93.6 | 90.3 | 96.0 | 81.7 | 94.6 | 98.6 | 100.0 | 65.2 | 89.7 | 90.9 | 98.3 | 97.5 | 99.8 | 98.6 | 100.0 | 88.2 | 96.5 |
| | | C2P-CLIP (AAAI'25) | 73.1 | 92.2 | 68.0 | 68.1 | 73.6 | 93.5 | 92.2 | 99.4 | 65.0 | 87.9 | 66.2 | 95.8 | 82.4 | 98.5 | 89.1 | 99.1 | 76.2 | 91.8 |
| | | FALCON-Net (our) | 86.2 | 94.4 | 96.5 | 99.0 | 89.0 | 98.3 | 99.0 | 100.0 | 96.3 | 99.3 | 87.9 | 97.4 | 98.6 | 99.9 | 99.0 | 100.0 | **94.1** | **98.5** |
| | 1.5 | NPR (CVPR'24)* | 85.0 | 98.9 | 91.5 | 100.0 | 74.3 | 99.2 | 99.9 | 100.0 | 92.5 | 100.0 | 100.0 | 100.0 | 97.3 | 99.9 | 99.7 | 100.0 | 92.5 | 99.7 |
| | | MI_Net (AAAI'24)* | 75.8 | 83.3 | 61.5 | 42.4 | 49.2 | 47.7 | 96.7 | 99.1 | 60.1 | 87.3 | 97.0 | 98.9 | 84.2 | 95.2 | 97.8 | 99.5 | 77.8 | 81.7 |
| | | SAFE (KDD'25) | 77.8 | 97.3 | 99.1 | 100.0 | 89.8 | 99.9 | 100.0 | 100.0 | 71.9 | 100.0 | 94.7 | 99.9 | 93.8 | 100.0 | 100.0 | 100.0 | 90.9 | 99.6 |
| | | C2P-CLIP (AAAI'25) | 67.4 | 96.3 | 64.3 | 58.0 | 62.5 | 92.6 | 91.4 | 99.9 | 72.5 | 97.5 | 77.6 | 99.7 | 63.9 | 99.4 | 83.1 | 99.8 | 72.8 | 92.9 |
| | | FALCON-Net (our) | 91.8 | 99.6 | 99.6 | 100.0 | 94.9 | 99.9 | 100.0 | 100.0 | 95.8 | 100.0 | 89.4 | 99.8 | 99.7 | 100.0 | 100.0 | 100.0 | **96.4** | **99.9** |
| JPEG Compression | 70 | NPR (CVPR'24) | 54.2 | 60.0 | 71.3 | 68.5 | 64.0 | 77.2 | 51.2 | 52.7 | 62.6 | 75.6 | 36.2 | 59.9 | 54.1 | 62.9 | 57.7 | 67.6 | 56.4 | 65.5 |
| | | MI_Net (AAAI'24) | 52.4 | 47.2 | 38.0 | 30.6 | 50.9 | 44.0 | 45.1 | 47.2 | 51.7 | 51.6 | 65.5 | 70.1 | 53.7 | 45.4 | 49.7 | 38.3 | 50.9 | 46.8 |
| | | SAFE (KDD'25) | 55.3 | 57.6 | 63.4 | 50.8 | 61.3 | 61.4 | 42.6 | 43.3 | 54.4 | 54.9 | 32.5 | 52.2 | 48.0 | 47.5 | 61.2 | 63.0 | 52.3 | 53.8 |
| | | C2P-CLIP (AAAI'25) | 51.1 | 81.0 | 63.7 | 80.2 | 52.7 | 90.0 | 50.0 | 92.2 | 57.4 | 95.5 | 36.0 | 71.3 | 50.0 | 50.8 | 50.0 | 77.4 | 51.4 | **79.8** |
| | | FALCON-Net (our) | 58.3 | 62.3 | 74.2 | 70.9 | 73.3 | 79.6 | 56.4 | 60.3 | 63.7 | 70.4 | 39.1 | 61.7 | 64.8 | 70.9 | 68.5 | 77.1 | **62.3** | 69.2 |
| | 80 | NPR (CVPR'24) | 56.0 | 59.9 | 74.5 | 72.7 | 73.2 | 83.6 | 67.2 | 76.7 | 66.1 | 76.4 | 35.9 | 57.9 | 60.3 | 69.6 | 60.8 | 70.4 | 61.8 | 70.9 |
| | | MI_Net (AAAI'24) | 52.3 | 55.6 | 62.3 | 42.5 | 51.2 | 55.5 | 60.2 | 75.3 | 50.1 | 52.7 | 40.8 | 68.2 | 65.0 | 80.8 | 59.1 | 74.5 | 55.1 | 63.1 |
| | | SAFE (KDD'25) | 54.1 | 53.7 | 70.4 | 59.6 | 66.6 | 69.0 | 65.5 | 66.2 | 52.6 | 51.5 | 35.4 | 55.8 | 58.1 | 60.2 | 63.6 | 68.6 | 58.3 | 60.6 |
| | | C2P-CLIP (AAAI'25) | 52.0 | 84.7 | 64.5 | 82.9 | 55.8 | 92.5 | 50.0 | 95.6 | 60.8 | 97.2 | 36.2 | 72.9 | 50.0 | 55.9 | 50.0 | 81.8 | 52.4 | **82.9** |
| | | FALCON-Net (our) | 56.9 | 60.5 | 77.3 | 80.2 | 76.5 | 89.2 | 78.2 | 90.3 | 60.6 | 69.7 | 46.5 | 71.0 | 76.0 | 88.3 | 63.8 | 77.4 | **67.0** | 78.3 |
| | 85 | NPR (CVPR'24) | 56.7 | 59.3 | 72.4 | 66.5 | 73.9 | 80.6 | 61.5 | 66.7 | 76.0 | 84.0 | 72.0 | 80.7 | 36.9 | 56.6 | 64.8 | 71.1 | 64.3 | 70.7 |
| | | MI_Net (AAAI'24) | 54.0 | 66.6 | 70.0 | 75.7 | 65.2 | 87.4 | 49.1 | 39.0 | 78.0 | 94.9 | 49.9 | 54.0 | 35.1 | 58.5 | 50.3 | 58.5 | 56.5 | 66.8 |
| | | SAFE (KDD'25) | 55.3 | 57.7 | 78.3 | 80.7 | 73.9 | 85.0 | 81.3 | 91.2 | 56.5 | 59.8 | 42.0 | 66.6 | 79.2 | 90.0 | 74.4 | 86.1 | 67.6 | 77.1 |
| | | C2P-CLIP (AAAI'25) | 53.4 | 85.5 | 65.2 | 80.5 | 56.2 | 92.0 | 51.3 | 95.5 | 63.9 | 97.4 | 36.4 | 76.6 | 50.0 | 57.8 | 50.2 | 86.8 | 53.3 | **84.0** |
| | | FALCON-Net (our) | 60.0 | 65.8 | 84.7 | 88.0 | 83.2 | 92.3 | 89.8 | 97.0 | 59.1 | 64.6 | 47.5 | 72.1 | 83.7 | 92.5 | 84.8 | 93.3 | **74.1** | 83.2 |
| | 90 | NPR (CVPR'24) | 59.7 | 62.7 | 78.9 | 80.1 | 84.2 | 91.3 | 79.2 | 87.2 | 80.2 | 88.8 | 80.0 | 87.8 | 36.9 | 56.8 | 66.0 | 72.8 | 70.6 | 78.4 |
| | | MI_Net (AAAI'24) | 54.0 | 66.7 | 64.3 | 62.0 | 52.5 | 75.0 | 55.8 | 87.9 | 54.4 | 78.8 | 51.0 | 66.9 | 39.4 | 73.7 | 54.9 | 82.9 | 53.3 | 74.2 |
| | | SAFE (KDD'25) | 55.9 | 56.9 | 88.5 | 92.5 | 82.9 | 92.6 | 89.9 | 96.7 | 61.8 | 70.1 | 44.6 | 69.6 | 88.9 | 96.3 | 88.6 | 95.7 | 75.1 | 83.8 |
| | | C2P-CLIP (AAAI'25) | 52.5 | 85.5 | 64.1 | 81.2 | 54.3 | 93.4 | 51.0 | 97.3 | 61.5 | 98.3 | 36.2 | 74.3 | 50.0 | 62.8 | 50.0 | 88.8 | 52.4 | 85.2 |
| | | FALCON-Net (our) | 61.4 | 67.1 | 84.2 | 88.6 | 83.5 | 93.4 | 94.8 | 99.1 | 55.5 | 65.1 | 53.9 | 79.7 | 88.8 | 96.1 | 92.0 | 97.4 | **76.8** | **85.8** |

achieves 17.7% higher ACC, though its AP is slightly lower by 3.9%. Collectively, these results further confirm FALCON-Net's outstanding performance in handling JPEG compression.

These results indicate that the strength of low-level features lies in their model-agnostic nature. Unlike high-level cues that may depend more on semantic content or generator-specific artifacts, the proposed INP and LVP modules focus on acquisition-related residual traces and local intensity-relation statistics, which are more closely tied to the intrinsic difference between real imaging pipelines and synthetic generation processes and thus transfer better across unseen generators. Meanwhile, the robustness results suggest that these cues remain effective under common post-processing operations such as Gaussian blur and resizing, where useful local structures are still largely preserved. However, the performance degradation under JPEG compression also reflects the inherent trade-off of this design: once strong lossy compression suppresses high-frequency residuals and weakens local variation patterns, part of the discriminative low-level forensic evidence is inevitably reduced.

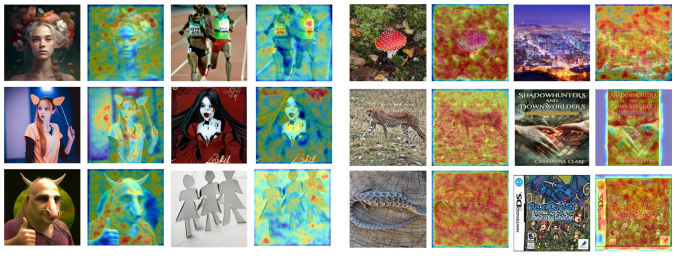
### H. Evaluation of the Hyperparameter $\alpha$ on INP Module

The hyperparameter $\alpha$ in Equation (5) is crucial for the performance of our Intrinsic Noise Pattern Isolation (INP) module, as it controls the sensitivity of the adaptive threshold T. This threshold determines which high-frequency components are preserved as part of the device fingerprint. To determine the optimal value of $\alpha$, we conduct a sensitivity analysis by evaluating the model's performance across a range of $\alpha$ values. Specifically, we test $\alpha$ from 0.00 to 4.00, and evaluate the overall performance using the mean Accuracy (ACC) and mean Average Precision (AP) across all 29 test subsets.

TABLE IX

PERFORMANCE EVALUATION WITH DIFFERENT VALUES OF THE HYPER-PARAMETER $\alpha$ IN THE INP MODULE. ALL RESULTS ARE REPORTED AS THE MEAN ACC AND MEAN AP ON THE 29 SUB-TESTSETS. THE BEST PERFORMANCE IS HIGHLIGHTED IN BOLD

| $\alpha$ | Mean ACC.of 29 sub-testsets | Mean AP.of 29 sub-testsets |
|---|---|---|
| 4.00 | 92.8% | 97.3% |
| 3.00 | 92.9% | 97.4% |
| 2.00 | 92.9% | 96.5% |
| 1.50 | 93.1% | 97.2% |
| 1.00 | 93.3% | 97.4% |
| 0.85 | 93.4% | 97.1% |
| 0.55 | 92.8% | 96.8% |
| 0.25 | 93.4% | 97.1% |
| 0.00 | 93.3% | 97.1% |
| **0.65(best)** | **93.6%** | **97.3%** |



a) CAM of Fake b) CAM of Real

Fig. 4. CAM visualization for real and generated images.

The results of this evaluation are presented in Table VIII. As the data shows, our model achieves the best performance when $\alpha$ is set to 0.65, reaching a Mean ACC of 93.6% and a Mean AP of 97.3%. We observe that the model's performance

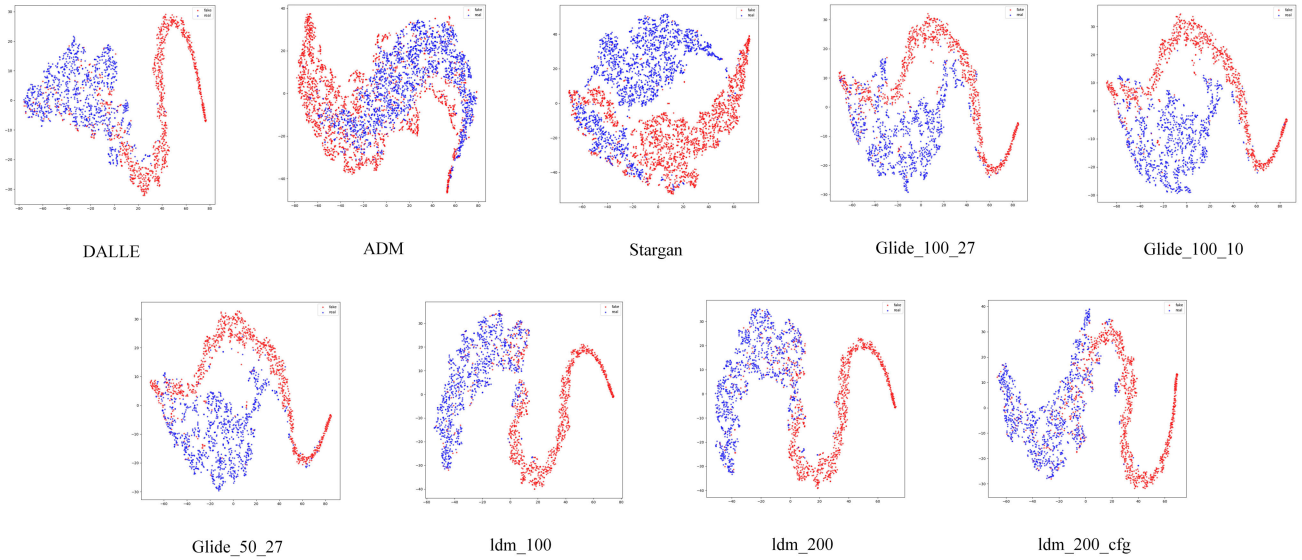



Fig. 5. t-SNE visualization of the high-dimensional features learned by FALCON-Net. Real images (blue dots) and fake images (red dots) form distinct and well-separated clusters, demonstrating that our model can effectively distinguish between the two classes.

is relatively stable within a certain range, for instance, at $\alpha = 0.85$ and $\alpha = 0.25$, the performance remains strong (93.4% ACC). However, when $\alpha$ is set to a larger value such as 2, the performance slightly degrades. In particular, when $\alpha \geq 2.00$, the Mean ACC drops to 92.8%–92.9%. This is likely because an overly large $\alpha$ leads to a high threshold T, which may filter out some subtle but important fingerprint features. Based on this empirical evidence, we select $\alpha = 0.65$ as the default value for all experiments conducted in this paper to ensure optimal performance.

### *I. Qualitative Analysis*

To further explore the inner characteristics of FALCON-Net, we conduct a qualitative analysis using Class Activation Map (CAM) visualizations [58], with data sourced from Midjourney and DALLE. As presented in Fig. 4, the CAMs for real images highlights broader regions of the image, whereas the CAMs for AI-generated images focus on more local forged areas. These observations provide strong evidence of FALCON-Net to effectively identify and localize generation artifacts, showcasing its robustness and precision in distinguishing between real and generated content.

Additionally, we visualize the feature vectors derived from the final layers of our model and the backbone model using t-SNE [59] as shown in Fig. 5. For images generated by DALLE, ADM, Stargan, Glide50_27, Glide100_10, Glide100_27, ldm_100, ldm_200, LDM_200_cfg, our model shows a smaller overlap between real and fake images in the feature space, indicating better generalization ability.

## V. Conclusion

This work proposes a novel Feature Aggregation for Localized Context and Noise Network (FALCON-Net) to expose generation artifacts in AI-generated images. Instead of pursuing ephemeral, model-specific artifacts, it establishes a new detection paradigm by targeting fundamental flaws inherent to all digital synthesis, the absence of device-specific sensor noise and the presence of unnatural local patterns. The INP module leverages frequency domain transformation and high-frequency enhancement techniques to isolate and amplify device-specific noise patterns, effectively capturing subtle discrepancies caused by the absence of physical sensor noise in generated images. Meanwhile, the LVP module employs directional encoding of intensity variations within local pixel neighborhoods, revealing unnatural regularities and reduced complexity in the spatial structures of generated images. Together, these modules work synergistically to extract and learn critical forensic cues that differentiate real and generated images. Experimental results demonstrate that FALCON-Net exhibits strong generalization in detecting various generative models and unseen data distributions. Furthermore, FALCON-Net illustrates robustness against post-processing operations. Nevertheless, FALCON-Net is less effective for partial manipulations such as DeepFake face swapping. Since the INP module extracts frequency-domain noise patterns globally, the real sensor fingerprints preserved in non-manipulated regions may obscure the fake signal from the swapped face region. Future work will explore attention mechanisms to better focus on manipulated regions.

## References

[1] M. Ruskov, "Grimm in wonderland: Prompt engineering with midjourney to illustrate fairytales," 2023, *arXiv:2302.08961*.

[2] J. Betker et al., "Improving image generation with better captions," *Comput. Sci.*, vol. 2, no. 3, p. 8, 2023. [Online]. Available: https://cdn.openai.com/papers/dall-e-3.pdf

[3] Z. Sun, N. Ruan, and J. Li, "DDL: Effective and comprehensible interpretation framework for diverse deepfake detectors," *IEEE Trans. Inf. Forensics Security*, vol. 20, pp. 3601–3615, 2025.

[4] J. Lu, J. Zhou, J. Dong, B. Li, S. Lyu, and Y. Li, "ForensicsForest family: A series of multi-scale hierarchical cascade forests for detecting GAN-generated faces," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 5106–5119, 2024.

[5] S. Tang, P. He, H. Li, W. Wang, X. Jiang, and Y. Zhao, "Towards extensible detection of AI-generated images via content-agnostic adapter-based category-aware incremental learning," *IEEE Trans. Inf. Forensics Security*, vol. 20, pp. 2883–2898, 2025.

[6] Z. Liu, H. Wang, Y. Kang, and S. Wang, "Mixture of low-rank experts for transferable AI-generated image detection," 2024, *arXiv:2404.04883*.

[7] J. Xu, Y. Yang, H. Fang, H. Liu, and W. Zhang, "FAMSeC: A few-shot-sample-based general AI-generated image detection method," *IEEE Signal Process. Lett.*, vol. 32, pp. 226–230, 2025.

[8] D. Cozzolino, G. Poggi, R. Corvi, M. Nießner, and L. Verdoliva, "Raising the bar of AI-generated image detection with CLIP," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2024, pp. 4356–4366.

[9] C. Tan et al., "C2P-Clip: Injecting category common prompt in clip to enhance generalization in deepfake detection," in *Proc. AAAI Conf. Artif. Intell.*, 2025, vol. 39, no. 7, pp. 7184–7192.

[10] L. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015.

[11] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2414–2423.

[12] S. Chen, T. Yao, Y. Chen, S. Ding, J. Li, and R. Ji, "Local relation learning for face forgery detection," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 2, pp. 1081–1088.

[13] C.-C. Hsu, Y.-X. Zhuang, and C.-Y. Lee, "Deep fake image detection based on pairwise learning," *Appl. Sci.*, vol. 10, no. 1, p. 370, Jan. 2020.

[14] T. Jung, S. Kim, and K. Kim, "DeepVision: Deepfakes detection using human eye blinking pattern," *IEEE Access*, vol. 8, pp. 83144–83154, 2020.

[15] S. Hu, Y. Li, and S. Lyu, "Exposing GAN-generated faces using inconsistent corneal specular highlights," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 2500–2504.

[16] X. Yang, Y. Li, and S. Lyu, "Exposing deep fakes using inconsistent head poses," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 8261–8265.

[17] F. Marra, D. Gragnaniello, D. Cozzolino, and L. Verdoliva, "Detection of GAN-generated fake images over social networks," in *Proc. IEEE Conf. Multimedia Inf. Process. Retr. (MIPR)*, Apr. 2018, pp. 384–389.

[18] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "CNN-generated images are surprisingly easy to spot. for now," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Apr. 2020, pp. 8695–8704.

[19] C. Tan, Y. Zhao, S. Wei, G. Gu, and Y. Wei, "Learning on gradients: Generalized artifacts representation for GAN-generated images detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 12105–12114.

[20] M. Zhang, H. Wang, P. He, A. Malik, and H. Liu, "Improving GAN-generated image detection generalization using unsupervised domain adaptation," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2022, pp. 1–6.

[21] Y. Lim, C. Lee, A. Kim, and O. Etzioni, "DistilDIRE: A small, fast, cheap and lightweight diffusion synthesized deepfake detection," 2024, *arXiv:2406.00856*.

[22] C. Zhao, C. Wang, G. Hu, H. Chen, C. Liu, and J. Tang, "ISTVT: Interpretable spatial–temporal video transformer for deepfake detection," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 1335–1348, 2023.

[23] Y. Zhuang et al., "Anti-fakeprompt: Prompt-tuned vision-language models for generalizable detection of AI-generated images," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2023.

[24] Z. Sha, Z. Li, N. Ning, and Y. He, "De-fake: Detection and attribution of fake images generated by text-to-image generation models," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2023, pp. 3418–3432.

[25] Z. Xu, X. Zhang, R. Li, Z. Tang, Q. Huang, and J. Zhang, "FakeShield: Explainable image forgery detection and localization via multi-modal large language models," 2024, *arXiv:2410.02761*.

[26] Z. Huang et al., "SIDA: Social media image deepfake detection, localization and explanation with large multimodal model," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2025, pp. 28831–28841.

[27] Z. Zhou et al., "AIGI-holmes: Towards explainable and generalizable AI-generated image detection via multimodal large language models," 2025, *arXiv:2507.02664*.

[28] U. Ojha, Y. Li, and Y. J. Lee, "Towards universal fake image detectors that generalize across generative models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 24480–24489.

[29] Z. Wang et al., "Dire for diffusion-generated image detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Aug. 2023, pp. 22445–22455.

[30] J. Hu, X. Liao, W. Wang, and Z. Qin, "Detecting compressed deepfake videos in social networks using frame-temporality two-stream convolutional network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 1089–1102, Mar. 2022.

[31] Z. Yu et al., "Searching central difference convolutional networks for face anti-spoofing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5295–5305.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[33] C. Tan, Y. Zhao, S. Wei, G. Gu, P. Liu, and Y. Wei, "Rethinking the up-sampling operations in CNN-based generative network for generalizable deepfake detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2024, pp. 28130–28139.

[34] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop," 2015, *arXiv:1506.03365*.

[35] O. Russakovsky et al., "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, pp. 211–252, Jul. 2015.

[36] C. Schuhmann et al., "LAION-400M: Open dataset of CLIP-filtered 400 million image-text pairs," 2021, *arXiv:2111.02114*.

[37] A. Ramesh et al., "Zero-shot text-to-image generation," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Jul. 2021, pp. 8821–8831.

[38] Z. Ba et al., "Exposing the deception: Uncovering more forgery clues for deepfake detection," in *Proc. AAAI Conf. Artif. Intell.*, 2024, vol. 38, no. 2, pp. 719–728.

[39] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, "Leveraging frequency analysis for deep fake image recognition," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 3247–3258.

[40] R. Durall, M. Keuper, and J. Keuper, "Watch your up-convolution: CNN based generative deep neural networks are failing to reproduce spectral distributions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7887–7896.

[41] L. Chai, D. Bau, S.-N. Lim, and P. Isola, "What makes fake images detectable? Understanding properties that generalize," in *Proc. 16th Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 103–120.

[42] Y. Qian, G. Yin, L. Sheng, Z. Chen, and J. Shao, "Thinking in frequency: Face forgery detection by mining frequency-aware clues," in *Proc. 16th Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 86–103.

[43] K. Shiohara and T. Yamasaki, "Detecting deepfakes with self-blended images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 18720–18729.

[44] S. Mandelli, N. Bonettini, P. Bestagini, and S. Tubaro, "Detecting gan-generated images by orthogonal training of multiple CNNs," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2022, pp. 3091–3095.

[45] Y. Jeong, D. Kim, S. Min, S. Joe, Y. Gwon, and J. Choi, "BiHPF: Bilateral high-pass filters for robust deepfake detection," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 48–57.

[46] Y. Jeong, D. Kim, Y. Ro, and J. Choi, "FrePGAN: Robust deepfake detection using frequency-level perturbations," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 1, pp. 1060–1068.

[47] C. Tan, Y. Zhao, S. Wei, G. Gu, P. Liu, and Y. Wei, "Frequency-aware deepfake detection: Improving generalizability through frequency space domain learning," in *Proc. AAAI Conf. Artif. Intell.*, 2024, vol. 38, no. 5, pp. 5052–5060.

[48] H. Liu, Z. Tan, C. Tan, Y. Wei, J. Wang, and Y. Zhao, "Forgery-aware adaptive transformer for generalizable synthetic image detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 10770–10780.

[49] O. Li, J. Cai, Y. Hao, X. Jiang, Y. Hu, and F. Feng, "Improving synthetic image detection towards generalization: An image transformation perspective," 2024, *arXiv:2408.06741*.

[50] S. Yan et al., "A sanity check for AI-generated image detection," 2024, *arXiv:2406.19435*.

[51] B. Chen, J. Zeng, J. Yang, and R. Yang, "DRCT: Diffusion reconstruction contrastive training towards universal detection of diffusion generated images," in *Proc. 41st Int. Conf. Mach. Learn.*, 2024.

[52] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognit.*, vol. 29, no. 1, pp. 51–59, Jan. 1996.

[53] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1657–1663, Jun. 2010.

[54] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[55] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with center-symmetric local binary patterns," in *Proc. 5th Indian Conf. Comput. Vis., Graph. Image Process.* Cham, Switzerland: Springer, 2006, pp. 58–69.
[56] M. Zhu et al., "GenImage: A million-scale benchmark for detecting AI-generated image," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 77771–77782.
[57] Z. Zhang et al., "A-bench: Are LMMs masters at evaluating AI-generated images?," 2024, *arXiv:2406.03070*.
[58] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.
[59] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, 2008.

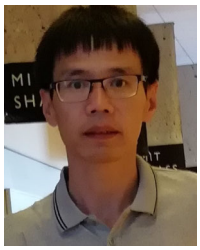
**Dengyong Zhang** received the B.S. and M.S. degrees from Changsha University of Science and Technology, Changsha, China, in 2003 and 2006, respectively, and the Ph.D. degree from Hunan University, Changsha, in 2018. He is currently a Professor with Changsha University of Science and Technology. His current research interests include digital media forensics and image processing.

**Miao Hu** received the B.E. degree in software engineering from Harbin University of Commerce, Harbin, China, in 2023. She is currently pursuing the M.S. degree with the School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, Hunan, China. Her research interests include video deepfake and image deepfake.

**Jiaxin Chen** received the B.S. degree from Central China Normal University, Wuhan, China, in 2017, and the Ph.D. degree from Hunan University, Changsha, China, in 2023. She is currently a Lecturer with Changsha University of Science and Technology, China. Her current research interests include multimedia forensics and deepfake detection. She is a member of the Technical Committee (TC) on Digital Forensics and Security of China Society of Image and Graphics.

**Changsheng Chen** (Senior Member, IEEE) received the B.E. degree in software engineering from Sun Yat-sen University, Guangzhou, China, in 2008, and the Ph.D. degree in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2013. He is currently a Professor with the Faculty of Engineering, Shenzhen MSU-BIT University, China. His research interests include multimedia forensics and security, document image analysis, and machine learning. He serves as a member the IEEE Information Forensics and Security Technical Committee (IFS-TC). He received the Best Paper Award from IEEE WIFS 2025 and the Best Paper Nomination at IEEE ICME 2024. He serves as the Area Chair for IEEE ICME and PRCV. He is an Associate Editor of IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY and IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING.

**Jin Wang** (Senior Member, IEEE) received the B.S. and M.S. degrees from Nanjing University of Posts and Telecommunications, China, in 2002 and 2005, respectively, and the Ph.D. degree from Kyung Hee University, South Korea, in 2010. He is currently a Professor with Hunan University of Science and Technology, China. He has published more than 400 international journals and conference papers. His research interests mainly include wireless ad hoc and sensor networks and network performance analysis and optimization. He is a fellow of IET.



**Yun Song** received the M.S. degree in computer science from Changsha University of Science and Technology, China, in 2008, and the Ph.D. degree in computer science from Hunan University, China, 2016. He is currently a Professor with Changsha University of Science and Technology. His research interests include image and video forensics, multimedia security, image and video processing, and deep learning.

**Gaobo Yang** received the B.S. degree from Shenyang University of Technology in 1995, the M.S. degree from East China Jiaotong University, China, in 2001, and the Ph.D. degree from Shanghai University, China, in 2004. He is currently a Professor with Hunan University, China. His research interests include image and video signal processing and digital media forensics.

**Xin Liao** (Senior Member, IEEE) received the B.E. and Ph.D. degrees in information security from Beijing University of Posts and Telecommunications in 2007 and 2012, respectively. He was a Post-Doctoral Fellow with the Institute of Software, Chinese Academy of Sciences, and also a Research Associate with The University of Hong Kong. From 2016 to 2017, he was a Visiting Scholar with the University of Maryland, College Park, USA. He is currently a Professor and a Doctoral Supervisor with Hunan University, China. His current research interests include multimedia forensics, steganography, and watermarking.

**Xiangling Ding** received the B.S. and M.S. degrees from Hunan Normal University in 2003 and 2006, respectively, and the Ph.D. degree from Hunan University in 2018. He is currently a Professor with Hunan University of Science and Technology, Xiangtan, China. His current research interests include multimedia forensics, video processing, and machine learning.