Secret Lies in Color: Enhancing AI-Generated Images Detection with Color Distribution Analysis
Supplementary Materials

The organization of this appendix is as follows:

- In Section 1, we provide detailed steps of the proofs omitted in the analysis sections due to space constraints.
- In Section 2, we discuss the potential negative societal impacts that may arise from generated images in practical scenarios.
- In Section 3, we conduct additional experiments to evaluate the effectiveness of our approach in terms of robustness against image perturbations.
- In Section 4, We provide the code for the color feature extraction proposed in our work, styled in PyTorch.
- In Section 5, we summarize the potential limitations of our method and outline directions for future work to address these issues.

1. Societal Impacts

The rapid development of artificial intelligence technology, particularly in image generation, has led to a new social phenomenon: AI-generated fake images. These images can be indistinguishable from real ones and sometimes reach a level where they are almost impossible to differentiate. While this technology offers convenience, it also brings significant social issues, particularly concerning personal privacy, public trust, and social stability.

1.1. Invasion of Personal Privacy

One of the most direct impacts of AI-generated fake images is the invasion of personal privacy. For instance, many women have found nude photos of themselves circulating online, which are often created by processing their real photos using AI technology. This not only causes immense psychological pressure for the victims but can also damage their family relationships and social standing. Gabi Belle, a YouTube influencer, discovered a ”nude” photo of herself online that was actually generated by AI. This incident left her extremely distressed and even affected her daily life and career development. Similar cases are common on social media, where many women suffer from cyberbullying and personal attacks, leading to significant psychological stress.

1.2. Erosion of Public Trust

AI-generated fake images not only invade personal privacy but also erode public trust. In an era of information overload, people increasingly rely on the internet for news and information. However, the proliferation of false information undermines public trust in the information they receive. For instance, in April 2023, rumors about the possible arrest of former U.S. President Donald Trump spread online, accompanied by numerous AI-generated images of his arrest. These images, though detailed, were entirely fabricated. This event not only confused the public but also led to social unrest. Despite being later confirmed as false, the short-term impact included widespread sharing and commenting, which already caused chaos in public opinion and even prompted some supporters to take extreme actions.

1.3. Threats to Social Stability

AI-generated fake images not only affect individuals and public trust but can also pose direct threats to social stability. In May 2023, a picture showing an explosion near the Pentagon circulated rapidly on social media, causing stock market volatility and investor panic. Although it was later verified that the image was AI-generated, the social impact had already been significant. Such incidents occur globally, and the spread of false information can heighten social tensions and trigger unnecessary panic and conflict.

2. Supplementary Proof

We expand the error function $f(x) = \mathrm{erf}(x)$, As shown in Equation (5) in the main text, into a Fourier series sine-cosine form:

$$\mathcal{F}(f) = a_0 + \sum_{\substack{m \in \mathbb{Z} \\ m \neq 0}} a_m, \tag{1}$$

where a_0 is the DC component, a_m are the Fourier coefficients incorporating sine and cosine components, and the summation spans all non-zero integers m.

The expanded of a_m result is:

$$\begin{aligned} a_m = &-\frac{i\exp\left(-m\pi\left(i+2m\pi/k^2\right)\right)}{2m\pi}\cdot \\ &\left(\exp\left(2m^2\pi^2/k^2\right)\left(\operatorname{erf}(k\eta)-\operatorname{erf}(k+k\eta)\right)\right. \\ &+\exp\left(m\pi\left(2i\eta+m\pi/k^2\right)\right)\left(-\operatorname{erf}(k\eta+im\pi/k)\right. \\ &\left.\left.+\operatorname{erf}(k+k\eta+im\pi/k)\right)\right) \\ =&-\frac{i\exp\left(-m\pi i-m^2\pi^2/k^2\right)}{2m\pi}\cdot \\ &\left(\exp\left(m^2\pi^2/k^2\right)\left(\operatorname{erf}(k\eta)-\operatorname{erf}(k+k\eta)\right)\right. \\ &+\exp(2m\eta\pi i)(-\operatorname{erf}(k\eta+im\pi/k) \\ &\left.+\operatorname{erf}(k+k\eta+im\pi/k))\right) \\ =&-\frac{i(-1)^m\exp\left(-m^2\pi^2/k^2\right)}{2m\pi}\cdot \\ &\left(\exp\left(m^2\pi^2/k^2\right)\left(\operatorname{erf}(k\eta)-\operatorname{erf}(k(\eta+1))\right)\right. \\ &\left.+\left(-\operatorname{erf}(k\eta+im\pi/k)+\operatorname{erf}(k(\eta+1)+im\pi/k)\right)\right). \end{aligned} \tag{2}$$

Expanding the summation, we observe that many terms cancel out, This leads us to:

$$\begin{aligned} \sum_{\eta=-\infty}^{\infty} a_m = &\lim_{N\to\infty}-\frac{i(-1)^m\exp\left(-m^2\pi^2/k^2\right)}{2m\pi}\cdot \\ &\left(\exp\left(m^2\pi^2/k^2\right)\left(\operatorname{erf}(-kN)-\operatorname{erf}(kN)\right)\right. \\ &\left.+\left(-\operatorname{erf}(-kN+im\pi/k)+\operatorname{erf}(kN+im\pi/k)\right)\right) \\ =&\frac{i(-1)^m\exp\left(-m^2\pi^2/k^2\right)}{2m\pi}(0+2) \\ =&\frac{i(-1)^m\exp\left(-m^2\pi^2/k^2\right)}{m\pi}. \end{aligned} \tag{3}$$

Thus, we can obtain:

$$\sum_{\eta=-\infty}^{\infty}\sum_{m=1}^{\infty} a_m = \sum_{m=1,m\in\mathbb{Z}}^{\infty}\frac{i(-1)^m\exp\left(-m^2\pi^2/k^2\right)}{m\pi}. \tag{4}$$

Similarly, there are many terms in a_0 that can cancel out front and back, ultimately resulting in:

$$\begin{aligned} \sum_{\eta=-\infty}^{\infty} a_0 = &\lim_{N\to\infty}\frac{-\exp\left(-k^2N^2\right)+\exp\left(-k^2N^2\right)}{k\sqrt{\pi}} \\ &-\left[(-N)\operatorname{erf}(-kN)+N\operatorname{erf}(kN)\right] \\ =&\,0. \end{aligned} \tag{5}$$

Consequently, we derive the color gradation distribution difference $E(y)-x$ for the restored image relative to the actual image:

$$\begin{aligned} E(y)-x &= \frac{1}{2}s\sum_{m\in\mathbb{Z},m\neq 0} a_m\exp(i2\pi m\mu) \\ &= s\sum_{m=1,m\in\mathbb{Z}}^{\infty}\frac{(-1)^m\exp\left(-m^2\pi^2/k^2\right)}{m\pi}\sin(2\pi m\mu). \end{aligned} \tag{6}$$

2.1. Legal and Ethical Challenges

AI-generated fake images present significant legal and ethical challenges. Victims face numerous difficulties in seeking justice, as these images are AI-generated and lack original source material, making it hard for victims to prove their innocence. Additionally, existing laws and ethical frameworks lag behind technological advancements, complicating efforts to effectively combat such illegal activities. While some countries and regions have begun to develop relevant policies, the enforcement and effectiveness of these measures remain uncertain.

Despite the remarkable technical capabilities demonstrated by AI-generated virtual images, the potential negative impacts they may cause must also be brought to our attention. Whether it is the invasion of personal privacy, the reduction of public trust, or the potential risks to social stability, these issues require our high attention and active search for solutions. Promoting the development of more precise AI-generated content recognition technology is particularly crucial for creating a safer and more reliable information environment.

3. Additional Experiments

To evaluate the impact of forgery adaptation on the robustness of our model, we implement several common image perturbation techniques on the test images, aligning with those used in the FatFormer study. Specifically, we apply four types of perturbations: random cropping, Gaussian blurring, JPEG compression, and adding Gaussian noise, each with a 50% probability. Tests are conducted on two datasets: one composed of images generated by GANs (the Forensic Synthetics dataset) and another featuring images from diffusion models (the GenImage dataset). Additionally, we compare our results with those of the FatFormer, UniFD, and LGrad models. Table 1 summarizes these test results.

When tested against various types of noise, our proposed method has demonstrated high stability and consistency, maintaining a high recognition accuracy. This achievement is primarily attributed to the innovative combination of color features and image features we have adopted, which together construct a robust and generalized forgery detection model. This aspect is analyzed in the network chapter.

Perturbation	Method	ACC$_{GAN}$	AP$_{GAN}$	ACC$_{Diffusion}$	AP$_{Diffusion}$
Original Image	LGrad	86.1	91.6	61.8	62.0
	UniFD	89.1	98.2	68.3	78.4
	FatFormer	**98.2**	99.5	74.1	82.3
	Ours	**98.2**	**99.6**	**95.3**	**98.7**
Gaussian blurring	LGrad	78.5	83.2	57.3	58.4
	UniFD	83.4	92.0	68.3	78.4
	FatFormer	97.7	98.4	73.1	81.0
	Ours	**98.9**	**99.6**	**95.1**	**98.4**
Random cropping	LGrad	83.5	90.8	53.1	52.4
	UniFD	84.8	94.1	62.9	72.0
	FatFormer	97.3	**98.8**	72.0	79.9
	Ours	**97.7**	98.6	**94.8**	**98.2**
JPEG compression	LGrad	77.4	85.2	58.8	59.9
	UniFD	85.5	93.7	64.8	76.7
	FatFormer	96.7	99.1	72.7	81.2
	Ours	**98.1**	**99.2**	**93.7**	**98.1**
Gaussian noising	LGrad	76.1	84.4	56.7	57.8
	UniFD	81.3	92.6	69.2	80.1
	FatFormer	94.4	97.8	74.3	84.9
	Ours	**98.0**	**99.2**	**94.4**	**98.0**

Table 1. Performance comparison under different perturbations.

4. Code for Color Features Extraction

```
def quantize_image(original_image,
    num_levels):
    """
    Quantizes the input image to a
        specified number of levels using
        rounding.
    """
    scaled_image = original_image * (
        num_levels - 1)
    rounded_image = torch.round(
        scaled_image)
    quantized_image = (rounded_image / (
        num_levels - 1)).clamp(0, 1)

    return quantized_image

def add_gaussian_noise(image_tensor):
    """
    Adds Gaussian noise to the given image
        tensor.
    """
    noisy_image = util.random_noise(
        image_tensor.cpu().numpy(), mode=’
        gaussian’, clip=True)

    noisy_tensor = torch.tensor(noisy_image
        , dtype=torch.float32, device=
        image_tensor.device)
    return noisy_tensor

def process_image_batch(original_images,
    num_quantization_levels,
    repetition_count):
    """
    Processes a batch of images by
        quantizing them, adding Gaussian
        noise, and averaging the results
        over multiple repetitions.

    """
    quantized_original_images =
        quantize_image(original_images,
        num_quantization_levels)

    accumulated_processed_images = torch.
        zeros_like(original_images, dtype=
        torch.float32)

    # Perform the noise addition and
        quantization process multiple times
    for _ in range(repetition_count):
        noisy_images = add_gaussian_noise(
            original_images)
        quantized_noisy_images =
            quantize_image(noisy_images,
            num_quantization_levels)s
        accumulated_processed_images +=
            quantized_noisy_images

    # Compute the average
    averaged_processed_images =
        accumulated_processed_images /
        repetition_count

    return averaged_processed_images -
        original_images
```

Listing 1. The code for the color feature extraction proposed in our work, styled in PyTorch

5. Limitations and Future Work

Our research results indicate that the use of color features can significantly reduce bias during the training process and enhance the model's generalization ability. However, when tested on the FakeART dataset, although our method surpasses existing baseline models, there is still considerable room for improvement. Notably, the detection of cross-domain generated works remains an urgent challenge to be addressed. Moreover, when dealing with generated works that contain large areas of black and white, the discriminative features in the color space may be weakened, thereby affecting the detection performance. We plan to explore more robust features in future research and combine multiple discriminative features to develop an efficient and robust detection system capable of adapting to various practical application scenarios.