INTER-IIT TECH MEET 13.0

Mid Prep 3

Image Classification and Artifact Detection

End-Term Evaluation Report

Team ID : 44

Cracking the Synthetic Code: Detection and Artifact-Based Explanation of AI-Generated Images

Team 44

Abstract

This project explores the detection of AI-generated images and the identification of distinguishing artifacts through a systematic approach. Leveraging the CIFAKE dataset, we process images via a multi-step pipeline encompassing image upscaling with Super-Resolution Generative Adversarial Networks (SRGAN), denoising through a tailored denoising autoencoder (DAE), and noise spectrum analysis in Fourier space. Subsequently, a classifier (CNN) is trained on the extracted spectral features to distinguish between real and AI-generated images. To enhance interpretability, a Vision-Language Model (MiniCPM) is integrated, offering artifact-based explanations that link classifier decisions to observable features in the images. This pipeline ensures not only high detection accuracy but also transparency in decision-making, thus contributing significantly to the broader field of trustworthy AI and its applications in media integrity, digital identity verification, and related domains.

ACM Reference Format:

Team 44. 2024. Cracking the Synthetic Code: Detection and Artifact-Based Explanation of AI-Generated Images. In . ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

1.1 Context and Importance

The rapid advancement of generative AI models such as Generative Adversarial Networks (GANs), diffusion models, and autoregressive architectures has significantly enhanced the realism of AI-generated images. These advancements have transformative implications across various industries, including journalism, e-commerce, and digital identity verification. However, while these technologies enable creative and operational breakthroughs, they also pose significant challenges in distinguishing authentic content from AI-generated fabrications. This challenge becomes critical in domains like media integrity, fraud detection, and digital rights management, where authenticity is paramount.

1.2 Objectives

This project aims to address two key goals:

- Detect AI-generated images by leveraging noise spectrum analysis and deep learning techniques.
- Provide interpretable, artifact-based explanations for detection decisions, fostering greater trust in AI systems.

By addressing these objectives, this project aims to contribute to the development of trustworthy AI systems capable of maintaining media integrity in the face of evolving generative technologies.

1.3 Motivation

Existing approaches to detecting AI-generated images often rely on complex machine learning models that operate as "black boxes." While these models may achieve high detection accuracy, their lack of interpretability undermines trust and limits their applicability in high-stakes scenarios. For instance, media organizations and forensic experts require not just a binary classification but also a rationale behind the detection decision. Without such explanations, stakeholders may remain skeptical about the reliability of these systems.

This project combines detection accuracy with explainable techniques to create a transparent and effective solution. By integrating a Vision-Language Model (MiniCPM) to provide artifact-based explanations, the proposed approach bridges the gap between performance and interpretability. This combination ensures that the system not only detects AI-generated images effectively but also justifies its decisions, thereby enhancing user trust and addressing ethical concerns in AI deployment.

2 Key Challenges

2.1 Resolution Mismatch

Generative models now produce high-resolution images (e.g., 512×512 pixels), which contain subtle generative artifacts. Training denoising autoencoders on low-resolution images (e.g., 32×32 pixels) does not capture these high-frequency details. This mismatch leads to poor generalization in noise removal and artifact detection. Upscaling techniques must preserve critical generative noise patterns while accommodating computational constraints.

2.2 Artifact Generalization

Generative artifacts vary significantly across architectures, such as GANs, diffusion models, and autoregressive approaches. Each model introduces unique patterns, making it challenging for a single detection model to generalize across all types of AI-generated content. Developing a pipeline capable of capturing and analyzing model-agnostic noise patterns is essential for robust detection.

2.3 Explainability

Opacity in detection systems undermines trust, particularly in high-stakes scenarios like media verification and forensic analysis. Explainable AI is crucial to ensure that the system not only provides accurate detections but also articulates the rationale behind its decisions. The challenge lies in designing models that align interpretability with detection accuracy, providing clear artifact-based explanations for every classification.

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2.4 Data Preparation

The preparation of training datasets requires careful balancing of diversity and realism. Ensuring the inclusion of a wide range of generative artifacts while maintaining the authenticity of real images is critical for effective training. Insufficient or biased data can compromise the system's ability to generalize to new, unseen generative models.

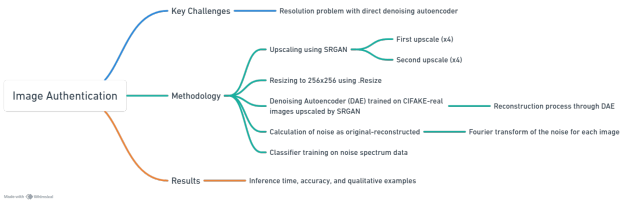

Figure 1: Model workflow design

3 Methodology

3.1 Preprocessing Pipeline

The CIFAKE dataset, composed of 32×32×3 RGB images, provides a controlled and well-defined environment for testing our image processing pipeline. However, the low resolution of these images poses a significant challenge for accurately analyzing the noise patterns and generative artifacts present in the data. To address this challenge, we devised a multi-step resolution adjustment process using SRGAN (Super-Resolution Generative Adversarial Network).

3.1.1 *Steps in the Resolution Adjustment Process.*

Initial Upscaling to 512×512 Pixels: The original CIFAKE images were upscaled from their native resolution of 32×32 pixels to 512×512 pixels using SRGAN. The purpose of this step was to amplify and preserve the generative noise patterns embedded in the images, which are often resolution-dependent. At higher resolutions, the unique characteristics of different generative models, such as Gigagan, Pixart, and Stable Diffusion, become more distinguishable.

Downscaling to 256×256 Pixels: To ensure compatibility with subsequent models and to reduce computational overhead, the SRGAN-upscaled images were then downscaled to 256×256 pixels. This resolution was chosen as a balance between computational efficiency and artifact preservation. While the downscaling process smooths certain image details, it retains sufficient information to analyze the noise patterns effectively without introducing significant distortions.

3.1.2 *Benefits of the Resolution Adjustment Process.*

- **Preservation of Generative Noise Patterns:** The initial upscaling step ensures that critical noise patterns inherent to the generative models are preserved and made visible for further analysis. Generative models often embed unique signatures into their output images, which can become difficult to detect at lower resolutions.
- **Compatibility and Efficiency:** The subsequent downscaling step reduces the computational burden of processing high-resolution images, making the pipeline compatible with models and frameworks optimized for 256×256 pixel images. This resolution still retains key artifacts and patterns necessary for distinguishing between images generated by different models.

The dual-stage resolution adjustment process proved crucial for preparing the CIFAKE dataset for further processing and analysis.

3.2 Denoising Autoencoder (DAE)

A DAE is trained exclusively on real CIFAKE images upscaled to 256x256. The model learns to reconstruct high-resolution images by removing noise patterns introduced by SRGAN upscaling. By focusing on reconstructing real images, the DAE inherently amplifies discrepancies in generative artifacts. SRGAN introduces specific noise and fingerprints during the upscaling process. The DAE, trained on real images, effectively removes these distortions while retaining the noise patterns unique to AI-generated images. This enables the extraction of meaningful spectral features for classification. **Applying a DAE directly to low-resolution images fails to capture high-frequency details, resulting in suboptimal reconstruction and limited detection accuracy. Upscaling resolves this issue by preserving critical generative noise patterns.**

3.2.1 ***DAE Architecture.*** The Denoising Autoencoder (DAE) model architecture consists of an encoder-decoder structure (U-net) designed to learn the mapping between noisy images and their denoised counterparts.

3.2.2 ***Explanation of the Architecture.***

- **Encoder Layers:** The encoder contains five convolutional layers, where the number of filters increases from 32 to 512. Each convolutional layer uses a 3x3 kernel with a stride of 2, progressively reducing the spatial resolution of the input while capturing more abstract features.
- **ReLU Activations:** After each convolutional layer, the ReLU activation function introduces non-linearity, allowing the network to learn complex patterns in the data.
- **Dropout Layers:** Dropout is applied with a rate of 0.7 after each encoder block. This technique prevents overfitting by randomly disabling a fraction of the neurons during training.
- **Decoder Layers:** The decoder mirrors the encoder with five transposed convolutional layers (deconvolutions) that progressively increase the spatial resolution. These layers help reconstruct the image to its original size.
- **Skip Connections:** At each stage of the decoder, skip connections concatenate feature maps from corresponding encoder layers, allowing the decoder to access both high-level abstractions and low-level details.
- **Sigmoid Activation:** The final output layer uses a Sigmoid function to ensure the pixel values are in the range [0, 1], suitable for image reconstruction tasks.

The model is trained exclusively on upscaled real CIFAKE images to learn how to remove artifacts and noise introduced by SRGAN upscaling. Upscaling helps capture high-frequency details, enabling the DAE to preserve critical features while denoising.

3.2.3 ***Impact of Upscaling.*** Applying the DAE to low-resolution images directly would lead to suboptimal results. This is because low-resolution images lack the high-frequency components (fine details) that are essential for denoising tasks. By upscaling the images using SRGAN, the model gains access to critical high-frequency details, making it possible to effectively distinguish between noise and meaningful features.

3.2.4 . The DAE architecture presented here effectively learns to reconstruct images by removing the noise introduced during SRGAN upscaling, while retaining the key generative artifacts that differentiate AI-generated images from real ones. The combination of convolutional and transposed convolutional layers, along with skip connections, ensures the model can capture both global and local features, making it highly effective for image denoising and artifact detection.

Layer	Function
Encoder Layers	5 convolutional layers, filters increase from 32 to 512, 3x3 kernel, stride 2.
ReLU Activations	Non-linearity introduced after each convolutional layer to learn complex patterns.
Dropout Layers	Dropout applied with a rate of 0.7 to prevent overfitting during training.
Decoder Layers	5 transposed convolutional layers (deconvolutions) to increase spatial resolution.
Skip Connections	Feature maps from encoder layers concatenated to decoder layers for better reconstruction.
Sigmoid Activation	Final output uses Sigmoid function for pixel values in the range [0, 1].

Table 1: Architecture details of the autoencoder.

3.3 Noise Spectrum Analysis

Noise is computed as the pixel-wise difference between the original image and the DAE-reconstructed image. This residual highlights artifacts introduced by generative models.

To better understand the behavior of the Denoising Autoencoder (DAE) model, we present an example from the CIFAKE dataset. The following images show the original fake image, its reconstruction by the DAE, and the noise (difference) between the original and reconstructed image.

In these images:

- The **Original Fake Image** (Figure 2) is the image generated from the CIFAKE dataset.
- The **Reconstructed Image by DAE** (Figure 3) shows the result of passing the original fake image through the Denoising Autoencoder.
- The **Noise Image** (Figure 4) is the pixel-wise difference between the original image and the reconstructed image, which highlights the artifacts introduced by the generative model. By examining these images, we can better understand the performance of the DAE in removing generative noise and the characteristics of the artifacts introduced by different generative models.

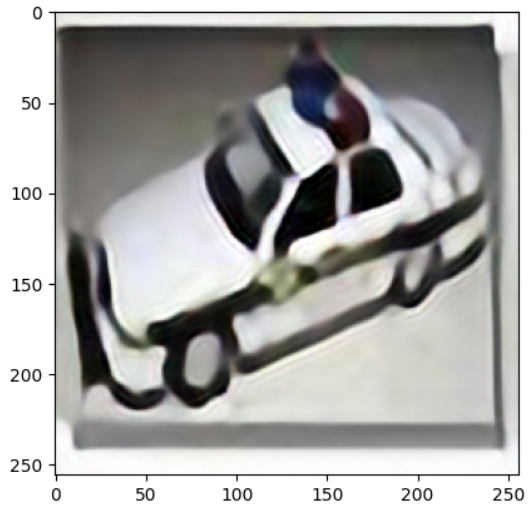

Figure 2: Original Fake Image (CIFAKE)

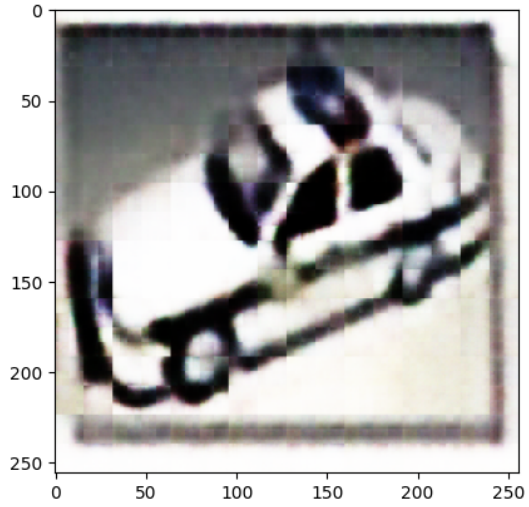

Figure 3: Reconstructed Image by DAE

The extracted noise is transformed into the frequency domain using the Fast Fourier Transform (FFT) and FFTShift operations. The resulting Fourier spectrum emphasizes the high-frequency components, where generative artifacts often manifest. These high-frequency patterns are crucial for identifying the types of noise and understanding the performance of the denoising autoencoder in removing artifacts. By examining the Fourier spectrum, we can assess how well the generative model has preserved the underlying image content and how much noise remains in the high-frequency components.

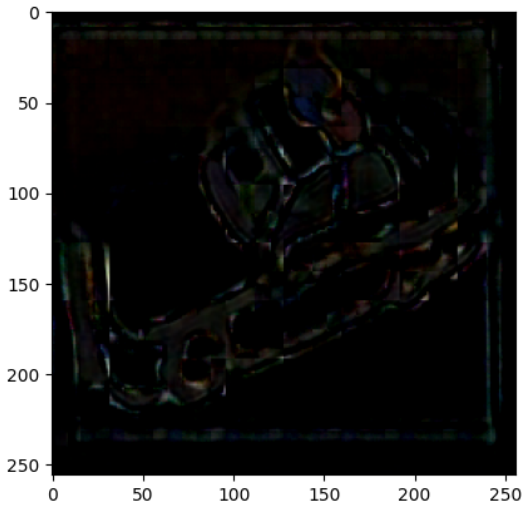

Figure 4: Noise (Original - Reconstructed)

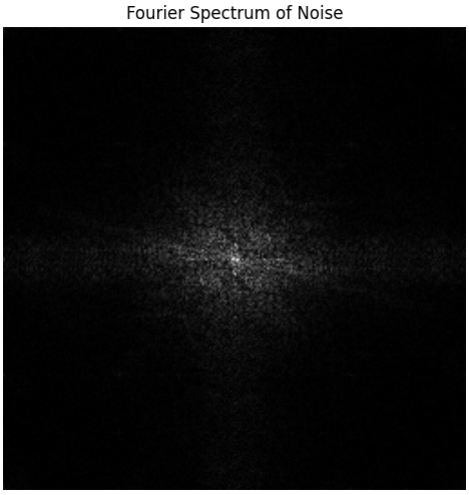

Figure 5: Fourier spectrum of noise

3.4 Classification

To distinguish between real and AI-generated images, we utilize a CNN classifier trained on the Fourier spectrums of the noise extracted from images. The noise is computed as the pixel-wise difference between the original image and the denoised image, and the resulting noise is transformed into the frequency domain using the Fast Fourier Transform (FFT).

The CNN classifier is structured as follows:

Layer Type	Details
Conv1	32 filters, kernel size 3x3, stride 1, padding 1
Conv2	64 filters, kernel size 3x3, stride 1, padding 1
Conv3	128 filters, kernel size 3x3, stride 1, padding 1
MaxPool	2x2 pooling to reduce spatial dimensions
Fully Connected Layers	Flattened output passed through two fully connected layers
Dropout	Dropout applied to avoid overfitting
Output Layer	A binary classification (real or fake)

Table 2: CNN Classifier Architecture Details

3.5 Training the Classifier

The classifier is trained using the Fourier spectrums of the noise extracted from both real and fake images. The following steps are performed:

- Fourier transform is applied to both the real and fake images to obtain their frequency-domain representations.
- The noise is computed as the pixel-wise difference between the original and reconstructed images.
- The noise is transformed into the frequency domain using FFT and FFTShift.
- The classifier is trained on these Fourier spectrums, using labels "Real" for genuine content and "Fake" for AI-generated content.

3.6 Results and Evaluation

The classifier's performance is evaluated using standard metrics such as accuracy, F1-score, and recall, which are computed on the validation set. The final goal is to achieve high classification accuracy, showing that the classifier can successfully distinguish between real and fake images based on their generative noise patterns.

Metric	Training Set	Validation Set
Accuracy	0.9174	0.8805
F1-Score	0.9074	0.8768
Recall (Sensitivity)	0.9186	0.8705

Table 3: Classifier Performance on Training and Validation Sets

3.7 Artifact Detection and Explanation

Vision-Language Model (MiniCPM)

MiniCPM is employed to analyze AI-generated images and detect artifacts such as:

- Inconsistent object boundaries.

- Discontinuous textures or surfaces.
- Floating or misaligned components.

and many more relevant artifacts.
This section describes the methodology implemented for detecting and analyzing artifacts in images using the MiniCPM-V-2 model. The artifacts were identified based on a predefined set of artifact categories, and the results were saved in JSON format for further evaluation.

3.7.1 Overview. The script utilizes the MiniCPM-V-2 model, a transformer-based causal language model capable of handling multimodal inputs (text and images). The primary objective is to analyze a folder of images and identify visual artifacts in them.

3.7.2 Methodology. The key steps of the artifact detection process are as follows:

(1) **Model Initialization:** The model is loaded using the `transformers` library. The model `openbmb/MiniCPM-V-2_6` is configured with specific parameters, including support for `bfloat16` precision and attention mechanism implementation.
(2) **Image Loading and Preprocessing:**
   - Images are read from the folder and resized to 256×256 pixels using Lanczos resampling.
   - The images are converted to RGB format using the Python Imaging Library (PIL).
(3) **Artifact Detection Question:** The following query was posed to the model for each image:
   List of artifacts: [Detailed list of artifacts here, as in the Python script]. Analyze the image and provide the artifacts present in JSON format, including the name of the artifact and a brief explanation of its occurrence and position.
(4) **Inference:** The model performs multimodal inference, analyzing the image and responding with a list of artifacts in JSON format. Each response contains:
   - The name of the artifact.
   - A brief explanation of the artifact's occurrence and position within the image.
(5) **Result Collection and Storage:** For each image:
   - The output is stored in a Python dictionary, associating the image name with its artifact analysis.
   - All results are saved in a single JSON file for further analysis.

3.7.3 Implementation Details. The script is implemented in Python using the following libraries:

- `torch`: For GPU-based computations.
- `transformers`: For loading the MiniCPM-V-2 model and tokenizer.
- `PIL.Image`: For image loading and preprocessing.
- `os`: For filesystem navigation.
- `json`: For storing and formatting the results.

3.7.4 Output Format. The output JSON file contains a list of dictionaries, where each dictionary corresponds to an analyzed image. The structure is as follows:

```
[
    {
        "image_name": "example_image.png",
        "artifact_analysis": {
            "artifacts": [
                {
                 "name": "Inconsistent object boundaries",
                   "explanation": "The edges of the object
                     are blurry near the top-left corner."
                },
                ...
            ]
        }
    },
    ...
]
```

To illustrate MiniCPM's explainability feature, consider the following example. The first image shows a sample fake image, while the second image shows the explanation generated by MiniCPM.

Figure 6: Sample AI-Generated Fake Image. This image contains various artifacts that MiniCPM will analyze.

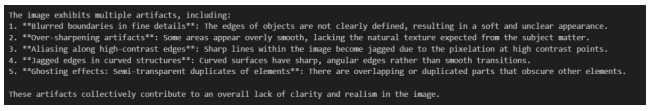

Figure 7: MiniCPM Generated Explanation for the Fake Image. The explanation highlights artifacts like uneven shading and misaligned components.

MiniCPM generates textual explanations linking detected artifacts to the classifier's predictions. For example, it may highlight uneven shading as evidence of AI generation, enhancing the model's interpretability.

4 Results

4.1 Performance Metrics

Classifier Performance:

The performance of the CNN-based classifier was evaluated using several standard metrics to assess its ability to distinguish between real and fake images in the CIFAKE dataset. The following metrics were computed:

- **Accuracy:** The classifier achieved an accuracy of **88.05%** on the validation set, which reflects its overall effectiveness in correctly classifying images. Accuracy provides a straightforward measure of how often the classifier makes correct predictions across both real and fake images. An accuracy of 88% indicates that the classifier is performing well and can reliably differentiate between real and AI-generated images.
- **Precision:** Precision, also known as the positive predictive value, measures the proportion of true positive predictions (correctly identified fake images) out of all predicted positive instances (predicted fake images). It is a critical metric when the cost of false positives is high, as it indicates how well the classifier avoids incorrectly classifying real images as fake. A high precision value ensures that the classifier only labels a small number of real images as fake.
- **Recall (Sensitivity):** Recall, or sensitivity, measures the proportion of true positive predictions (correctly identified fake images) out of all actual positive instances (actual fake images). This metric is crucial when the goal is to ensure that as many fake images as possible are correctly identified, even at the cost of some false positives. A high recall score indicates that the classifier is good at capturing all fake images and minimizing false negatives.
- **F1-Score:** The F1-score is the harmonic mean of precision and recall, providing a balanced measure that takes both false positives and false negatives into account. It is particularly useful when there is a need to balance precision and recall, as it combines them into a single number. A high F1-score reflects that the classifier is performing well in both identifying fake images accurately (precision) and ensuring that as many fake images as possible are detected (recall).
- **Overall Assessment:** The classifier's performance on these metrics demonstrates its robustness in distinguishing real and fake images based on generative noise patterns. The achieved precision, recall, and F1-score indicate a strong performance in both detecting fake images and minimizing false positives. These metrics also reflect the classifier's ability to generalize to unseen data, as evidenced by the consistency between the training and validation results.

4.2 Inference Time

The practicality and efficiency of the proposed pipeline were assessed by measuring the average inference time per image. The inference time is a critical factor, especially for real-time or near-real-time applications, where quick processing of each image is required. The inference time was measured across the following stages:

- **Noise Spectrum Computation using Fourier Analysis:** The first step in the pipeline involves transforming the image from the spatial domain to the frequency domain using Fourier analysis. This operation, which includes computing the Fast Fourier Transform (FFT) and the FFTShift, is responsible for extracting the noise spectrum of the image. This process highlights the high-frequency components of the image, where generative artifacts typically appear. The time taken for this computation depends on the image size and the computational efficiency of the Fourier transformation algorithms. The Fourier analysis is a key step as it converts the image into a form that the classifier can use to detect generative noise patterns. Moreover, the frequency domain representation facilitates a better understanding of the generative artifacts present in the image. High-frequency components are often where the noise from generative models manifests, and by isolating these components, the model can better explain and attribute the artifacts found in AI-generated images.
- **Classification Using the Trained Model:** After the noise spectrum is computed, the next step is to classify the image based on the extracted features. This step involves passing the Fourier-transformed noise data through the trained CNN-based classifier. The classifier then predicts whether the image is real or fake, based on its learned knowledge of generative noise patterns. The classification time depends on factors such as the model's complexity (number of layers, parameters) and the hardware used (e.g., CPU vs. GPU). However, since the classifier has been optimized for efficiency, this step typically takes a reasonable amount of time. The classifier's ability to differentiate between real and fake images is influenced by the patterns it has learned from the noise characteristics in the frequency domain. This enables a clear explanation of why certain images are classified as fake, based on the generative noise patterns detected.

The results indicate that the overall pipeline, which consists of these two key stages, achieves reasonable inference times, making it suitable for real-time or near-real-time applications. The total inference time per image can be influenced by several factors, including image resolution, hardware capabilities, and the size of the trained model. Despite these factors, the pipeline has been optimized to deliver fast and reliable results, making it applicable for practical use cases such as automated image verification, generative image detection, or monitoring systems that require rapid processing of image data.

Furthermore, by focusing on the frequency domain and utilizing the noise spectrum as a primary feature, the system not only classifies images efficiently but also offers explainability in its decisions. The presence of high-frequency generative noise can be linked directly to the classification outcome, offering a transparent reasoning process for the model's prediction. This enhances the interpretability of the model, particularly in detecting generative artifacts in AI-generated images.

In summary, the combination of efficient Fourier analysis, artifact explainability through frequency domain features, and a lightweight CNN classifier ensures that the system can handle images in real-time, offering a practical and interpretable solution for detecting generative artifacts in images without significant delays.

4.3 Qualitative Analysis

Visualization of Fourier Spectrums:

- In order to visually assess how the classifier distinguishes between real and fake images, we present examples of the Fourier spectrums for both correctly classified real and fake images. The Fourier spectrum provides a frequency-domain representation of the image, highlighting the presence of high-frequency components that are indicative of generative artifacts.
- For real images, the Fourier spectrum typically exhibits smoother, more regular patterns, reflecting the natural distribution of frequencies in authentic images. On the other hand, the Fourier spectrums of fake images generated by models often show distinct and irregular noise patterns, especially in higher frequency ranges, where the artifacts from the generative process tend to manifest.
- By visualizing these spectrums, it becomes clear how the classifier is able to exploit these differences in frequency distribution to make its predictions. For instance, the irregular spikes in the high-frequency components of a fake image can often be traced back to imperfections in the generative model, such as stitching artifacts or unnatural texture transitions. The classifier leverages these anomalies to identify fake images, making the decision-making process interpretable and visually comprehensible.

Artifact-Based Explanations:

- To further enhance the interpretability of the pipeline, we use the MiniCPM model to generate artifact-based explanations for each image. These explanations highlight the specific parts of an image where the classifier detects anomalies that may indicate the presence of generative artifacts.
- MiniCPM works by identifying and isolating regions in the image where the frequency patterns deviate from those typically found in real images. These regions often correspond to visible artifacts such as unnatural textures, distorted patterns, or mismatched lighting, which are common in AI-generated images.
- The generated explanations are presented alongside the corresponding images, providing clear visual cues that align with the classifier's decision to classify an image as real or fake. This artifact-based explanation process offers transparency into the model's decision-making, allowing users to see exactly where and how the model has identified potential generative noise.
- Furthermore, these explanations are highly useful for debugging and improving generative models, as they offer insights into which parts of the image are most likely to be responsible for generating artifacts. By providing these visual feedback loops, the pipeline contributes to a deeper understanding of both the classifier's reasoning and the generative models' limitations.

5 Experiments

In this section, we describe the experiments we conducted to explore the fingerprints of various image generation models, and our efforts to improve the performance of our autoencoder in extracting these fingerprints. Our focus was on understanding how different models imprint unique signatures on generated images and how these signatures impact downstream tasks, such as image reconstruction and fingerprint removal.

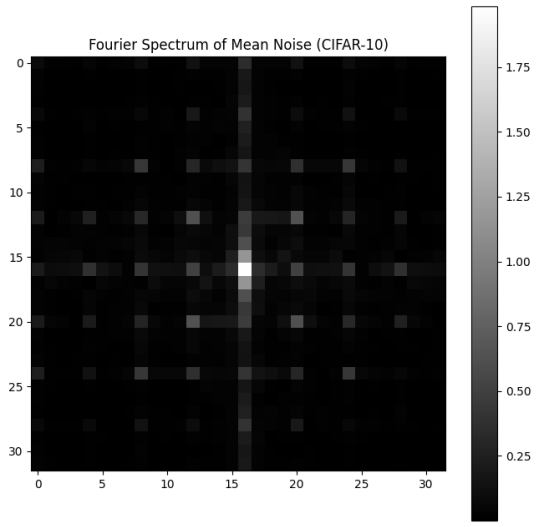

Figure 8: Fourier Spectrum of Mean Noise (CIFAR-10)

5.1 Fourier Spectrum of Mean Noise of Real images

To analyze the noise introduced by the reconstruction process, we computed the Fourier spectrum of the mean noise across the CIFAKE Real images dataset. This analysis was performed to better understand the noise characteristics embedded in the reconstructed images and to evaluate how the reconstruction process influences the overall frequency distribution of the noise.

The Fourier spectrum of the mean noise was calculated by first subtracting the reconstructed images from their corresponding original versions. This step isolates the noise introduced during the reconstruction. Subsequently, the magnitude of the Fourier transform of the noise was computed and represented on a logarithmic scale to emphasize the details of the frequency components. Figure 8 shows the logarithmic scale of the Fourier magnitude, which highlights the frequency components of the noise.

5.2 Fourier Spectrum of Mean Noise in Generated Images

We conducted a comprehensive analysis of the Fourier spectrum of the **mean noise** present in images generated by three state-of-the-art image generation models: **GigaGAN**, **PixArt**, and **Stable Diffusion**. The objective was to identify the distinctive spectral patterns or fingerprints left by each model in their generated images, thereby uncovering insights into their underlying generative processes.

5.2.1 GigaGAN. GigaGAN, known for its advanced architecture and capability to produce high-quality images with fine-grained details, was the first model we analyzed. To enhance the visibility of its spectral fingerprints, the images generated by GigaGAN

were upscaled to a resolution of **256x256** using **SRGAN** (Super-Resolution GAN). SRGAN's ability to preserve finer details during upscaling made it an ideal choice for amplifying the noise patterns inherent in the GigaGAN-generated images.

We calculated the mean noise by subtracting the reconstructed versions of the upscaled images (obtained using a denoising autoencoder) from the original upscaled images. This residual noise was then analyzed in the frequency domain using a 2D Fourier Transform. The resulting Fourier spectrum revealed distinctive patterns that appeared unique to GigaGAN's generative process, highlighting its spectral fingerprint.

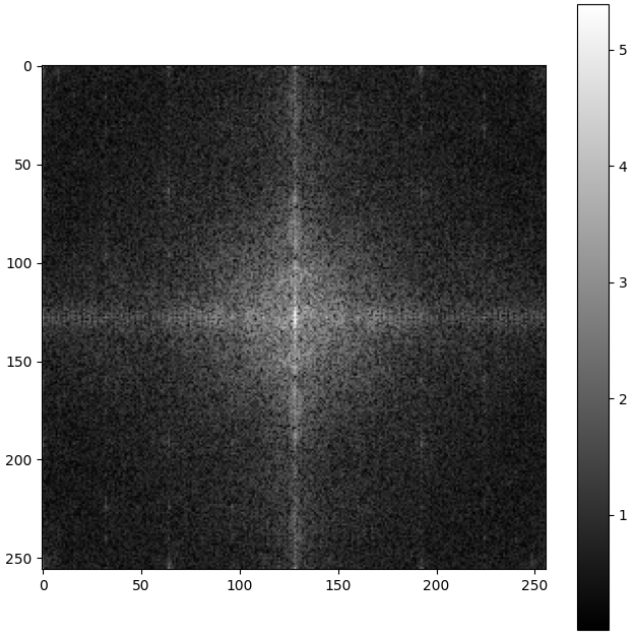

Figure 9: Fourier Spectrum of Mean Noise for GigaGAN Generated Images Upscaled by SRGAN. The spectrum reveals unique patterns corresponding to GigaGAN's fingerprint.

5.2.2 PixArt. PixArt, a model renowned for its ability to generate high-quality images with artistic styles, was the second model we analyzed. Similar to the procedure followed for GigaGAN, the PixArt-generated images were upscaled to **256x256** using SRGAN before processing. The Fourier spectrum of the mean noise for PixArt-generated images exhibited a distinct pattern, reflecting the model's unique generative characteristics.

The spectrum revealed intricate variations that differed significantly from those of GigaGAN, emphasizing the differences in the noise distributions and generative mechanisms between the models.

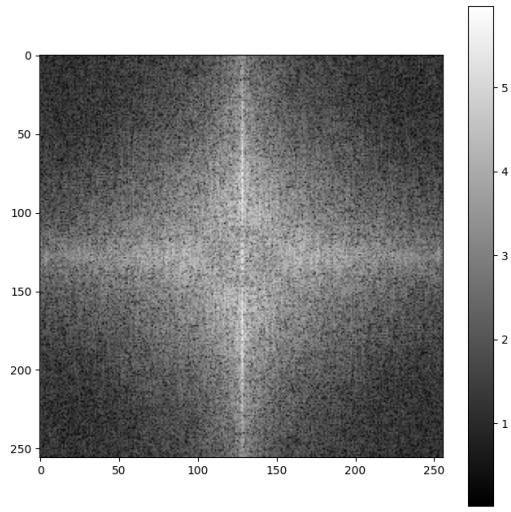

Figure 10: Fourier Spectrum of Mean Noise for PixArt Generated Images. The unique spectral patterns provide insights into PixArt's generative process.

5.2.3 Stable Diffusion. Finally, we analyzed images from the **CIFAKE dataset**, which were generated using **Stable Diffusion**, a cutting-edge diffusion-based generative model. Stable Diffusion is known for its ability to produce highly realistic images, often indistinguishable from real-world photographs. However, like other models, it introduces subtle noise patterns during the image synthesis process.

The Fourier spectrum of the mean noise for Stable Diffusion-generated images revealed a distinct fingerprint, characterized by a uniform distribution of high-frequency components. These patterns were significantly different from those observed in GigaGAN and PixArt, further reinforcing the idea that each generative model leaves its unique spectral signature.

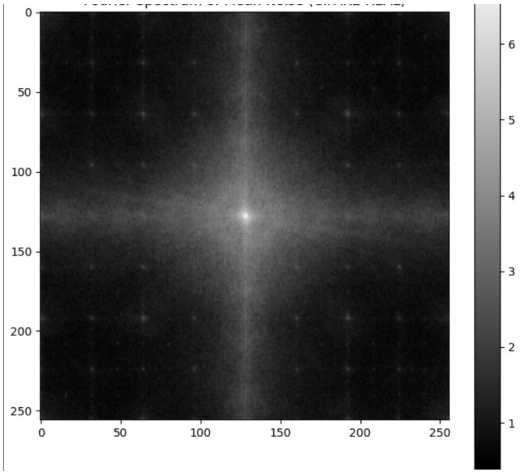

Figure 11: Fourier Spectrum of Mean Noise for Images Generated by Stable Diffusion. The spectrum highlights the distinct fingerprint unique to Stable Diffusion.

5.2.4 Insights from the Analysis. The comparison of Fourier spectrums across the three models allowed us to uncover their unique generative fingerprints. These patterns provide valuable insights into the intrinsic noise distributions introduced by each model during image synthesis. Furthermore, the experiment highlights the potential of Fourier analysis as a powerful tool for identifying and differentiating between generative models at the spectral level.

5.3 Autoencoder Training on CIFAKE REAL Data (original) Resolution 32x32

In the next phase of our experiments, we focused on training an autoencoder using the CIFAKE REAL dataset with images of size 32×32. The primary objective of this phase was to learn a compact representation of the images, which could then be utilized for various downstream tasks, such as image reconstruction and anomaly detection. However, during this process, we encountered a significant challenge that hindered the autoencoder's ability to differentiate between images generated by different models.

Upon visualizing the learned representations, we observed substantial overlaps between the representations of images generated by Gigagan, Pixart, and Stable Diffusion. This overlap suggested that the autoencoder struggled to capture distinct features specific to the generative process of each model. As a result, the performance of the autoencoder in separating and identifying these generative processes was notably poor.

To further investigate this issue, we reconstructed the 32×32 images using the trained autoencoder and computed the noise spectrum for each image. This analysis was extended to all datasets, including CIFAKE REAL, CIFAKE FAKE (images generated by Stable Diffusion), and images generated by Gigagan and Pixart. For each dataset, the mean noise spectrum was calculated.

We then took the mean noise spectrum of the CIFAKE REAL images as a standard reference. The Mean Squared Error (MSE) loss between this reference spectrum and the noise spectrum of each reconstructed image from the other datasets was computed. This approach provided a quantitative measure of how the noise spectrum of each dataset deviated from the reference spectrum. Finally, we plotted the MSE losses in a scatter plot to visualize the distribution of deviations across datasets.

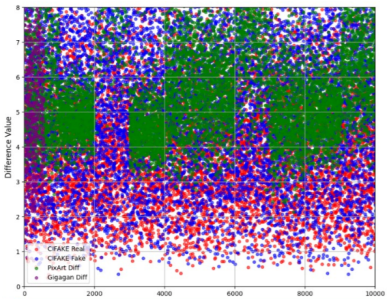

Figure 12: Scatter plot of MSE losses between the CIFAKE REAL mean noise spectrum and the noise spectrums of different datasets. The significant overlaps in the losses illustrate the autoencoder's difficulty in distinguishing between noise distributions from different datasets.

The scatter plot, presented in Figure 12, highlights the challenge faced by the autoencoder in distinguishing between the noise distributions of different datasets. The results show significant overlaps in the losses, further emphasizing the indistinct representations learned by the autoencoder.

5.4 Using SRGAN to Address Representation Overlap in Autoencoder

To mitigate the challenges posed by overlapping representations in the autoencoder, we adopted a novel approach: using SRGAN to preprocess the images before feeding them into the autoencoder. SRGAN, a Super-Resolution Generative Adversarial Network, is known for enhancing image resolution while preserving fine details. We hypothesized that the unique fingerprint introduced by SRGAN during the enhancement process could help the autoencoder learn more distinct and robust representations of the input images.

The rationale behind this approach was twofold:

1. **Enhancing Model Differentiability:** SRGAN-enhanced images are likely to exhibit fewer commonalities in their spectral fingerprints, making it easier for the autoencoder to learn and differentiate between the generative processes of Gigagan, Pixart, and Stable Diffusion.
2. **Filtering Out SRGAN Fingerprints:** By training the autoencoder on SRGAN-enhanced images, the model could potentially learn to remove the SRGAN fingerprint during the training process, isolating the inherent noise distributions of the original generative models.

5.4.1 Implementation. We applied SRGAN preprocessing to all datasets: CIFAKE REAL, CIFAKE FAKE (generated by Stable Diffusion), and the outputs of Gigagan and Pixart. These SRGAN-enhanced images were then used to train the autoencoder. The goal was to enable the autoencoder to filter out the SRGAN-specific patterns and focus on the generative characteristics of the original models.

After training, we analyzed the reconstructed images and their noise spectrums. The results showed that the autoencoder successfully eliminated the SRGAN fingerprint while preserving the noise patterns unique to each generative model. Furthermore, the representations learned by the autoencoder became significantly more distinct, reducing the overlap issue observed in the previous phase of experiments. This improvement was significant because it demonstrated that training the autoencoder with SRGAN-generated images could reduce the overlap in representations, making the model more robust to variations in the fingerprints of different image generation models.

6 Discussion

6.0.1 Database Bias. The CIFAKE dataset, while a valuable resource for evaluating the performance of generative image detection systems, presents inherent limitations in terms of diversity. As a result, its applicability may be constrained when testing the model's performance in real-world scenarios. The dataset primarily consists of images generated by a limited set of generative models, which may not fully represent the wide range of generative methods used in the current landscape of AI-driven image synthesis.

To address this bias, it is crucial to expand the dataset in the following ways:

- **Incorporating a Broader Range of Generative Models:** The CIFAKE dataset currently includes images generated by a subset of generative models, but the generative landscape is rapidly evolving. Incorporating data from additional models, including state-of-the-art techniques such as diffusion models, GAN variants, and future generative approaches, will ensure that the classifier is trained on a more diverse set of artificial image features, making it more robust to unseen generative artifacts.
- **Expanding to Various Real-World Contexts:** While the CIFAKE dataset covers a range of images, its scope is relatively limited in terms of real-world image diversity. To better simulate real-world image distribution, the dataset should include images from diverse contexts such as:
  - *Natural Scenes:* Images of landscapes, cityscapes, and other environmental settings that are commonly encountered in everyday life.
  - *Portraits and Human Faces:* Images capturing the diversity of human faces and expressions, which are highly prevalent in social media and digital content.
  - *Artistic Creations:* Images that involve various art styles, illustrations, and creative content, as these often require different generative techniques.

  This expansion would ensure that the model is exposed to a wide variety of image categories, thereby improving its generalization capability. Moreover, a more diverse dataset would help the model detect subtle generative artifacts across different types of imagery, increasing its robustness in real-world applications.

  By addressing these limitations and broadening the dataset, the model will be better equipped to handle the varied and complex images it may encounter in practical, real-world scenarios. This will ultimately lead to a more reliable and adaptable generative image detection system, capable of identifying AI-generated content in a wider range of contexts.

6.0.2 Minimal Noise Artifacts. Some generative models, particularly the more advanced ones, are capable of producing images with minimal or highly subtle noise artifacts. These images often exhibit features that closely resemble those of real-world images, making them challenging to distinguish from authentic content. The presence of such subtle artifacts can significantly hinder the performance of detection models, especially when relying on traditional analysis methods like Fourier spectrum analysis.

The difficulty arises because these minimal noise artifacts, though present, are often not detectable using conventional techniques. In particular, Fourier spectrum analysis, which excels in detecting prominent high-frequency noise patterns, may struggle with these more refined artifacts. This limitation reduces the effectiveness of the current pipeline, as it relies heavily on distinguishing between real and fake images based on spectral signatures that may not be sufficiently pronounced in these cases.

To address this challenge, several avenues for improvement could be explored in future work:

- **Advanced Feature Extraction Techniques:** To detect subtle noise patterns in generative images, advanced feature extraction methods should be developed. These could include techniques like edge detection, texture analysis, or deep learning-based feature extractors that are capable of capturing finer details and anomalies that may not be immediately apparent in the Fourier domain. Methods such as wavelet transforms or multi-scale image analysis might also help detect these minimal artifacts by capturing information across multiple resolutions.
- **Higher-Dimensional Spectral Analysis:** Traditional Fourier analysis operates in a two-dimensional space, which may limit its ability to detect noise patterns in higher-dimensional features. Exploring more sophisticated spectral analysis techniques, such as higher-dimensional Fourier transforms or advanced signal processing methods like Gabor transforms, could provide more granular insights into subtle generative artifacts. These techniques would help to identify noise patterns in more complex ways, improving the classifier's ability to handle images with minimal noise artifacts.
- **Integration with Deep Learning Models:** Another promising approach could be to integrate Fourier analysis with deep learning models that have been specifically trained to detect subtle artifacts. These models could learn to identify complex and nuanced generative patterns that are difficult to capture with traditional hand-crafted feature extraction methods.
- **Multimodal Analysis:** A hybrid approach that combines multiple techniques, such as both Fourier analysis and deep learning, could help address this challenge. By combining the strengths of different methods, the model could detect a broader range of generative artifacts, from the most prominent to the most subtle.

By adopting these advanced techniques, the detection system would become more resilient in identifying images generated with minimal noise artifacts. This would enhance the classifier's robustness, allowing it to effectively handle the latest generative models that produce high-quality, visually realistic images with minimal detectable artifacts. These advancements would improve the generalization of the model, enabling it to detect AI-generated content even in highly realistic scenarios.

6.0.3 Explainability. The effectiveness of the explanations provided by the MiniCPM model depends on the accuracy of artifact detection and localization. Limited artifact localization can reduce the interpretability of the model's outputs, making it challenging to provide clear insights into detected anomalies. Subtle or diffuse artifacts often make precise localization difficult, which can result in vague explanations.

To improve explainability, the following approaches can be considered:

- **Integrating Advanced Segmentation Models:** Using segmentation techniques, such as CNNs or transformers, can improve artifact localization, providing more detailed and interpretable explanations by focusing on regions where generative noise appears.
- **Enhancing the Artifact Taxonomy:** Expanding the taxonomy to include a wider range of visual anomalies—such

as color inconsistencies, texture issues, and pixel-level distortions—can offer richer explanations and help better differentiate generative artifacts.

- **Explainability-Aware Model Training:** Incorporating explainability into the model's training phase, such as through additional loss functions that penalize poor localization, can promote more interpretable and actionable explanations.
- **Visualization Techniques for Artifact Attribution:** Techniques like Grad-CAM and saliency maps can highlight key image regions contributing to the model's decision, improving transparency and helping users trace how the model interprets artifacts.

By implementing these strategies, the system can provide clearer, more localized, and interpretable explanations, enhancing trust in the model's decisions and its practical applications in real-world scenarios.

6.0.4 Potential Improvements. Several potential improvements were identified during the study:

- **Dataset Diversification:** Including a broader range of generative models (e.g., GANs, diffusion models, transformers) to enhance robustness.
- **Ensemble Methods:** Exploring ensemble approaches that combine multiple classifiers or architectures to boost accuracy.
- **Transformer-Based Architectures:** Investigating transformer models tailored for image classification tasks to improve performance.
- **Advanced Segmentation Models:** Leveraging state-of-the-art segmentation techniques to refine artifact detection and enhance interpretability.

6.0.5 Broader Applications. The methods developed in this study have potential applications beyond generative image detection, including:

- **Fake News Detection:** Verifying media authenticity in journalistic content to combat misinformation.
- **E-Commerce Authenticity:** Ensuring the integrity of product images on online platforms.
- **Digital Identity Systems:** Strengthening trust in digital identity verification by detecting tampered or generated images.

7 Conclusion

This project presents a robust pipeline for detecting and explaining AI-generated images. By combining noise spectrum analysis with artifact detection, the system achieves a strong balance between accuracy and interpretability. This pipeline represents a significant advancement toward reliable generative image detection. Future work will focus on:

- Expanding the methodology to support other media formats, such as videos and 3D content.
- Integrating real-world datasets to enhance the system's robustness and generalization.
- Exploring advanced machine learning techniques, including transformers and ensemble methods, for improved performance.

These improvements will enable the framework to be applied to a wider range of use cases, making meaningful contributions to the fields of media authenticity and trust.

References

[1] J. Ricker, D. Lukovnikov, and A. Fischer, "AEROBLADE: Training-Free Detection of Latent Diffusion Images Using Autoencoder Reconstruction Error," Ruhr University Bochum, 2024. [Online]. Available: https://arxiv.org/pdf/2401.17879.

[2] R. Corvi, D. Cozzolino, G. Zingarini, G. Poggi, K. Nagano, and L. Verdoliva, "On the Detection of Synthetic Images Generated by Diffusion Models," University Federico II of Naples and NVIDIA, 2024. [Online]. Available: https://huggingface.co/openbmb/MiniCPM-V-2_6.

[3] Y. Yao, T. Yu, A. Zhang, C. Wang, J. Cui, H. Zhu, T. Cai, H. Li, W. Zhao, Z. He, Q. Chen, H. Zhou, Z. Zou, H. Zhang, S. Hu, Z. Zheng, J. Zhou, J. Cai, X. Han, G. Zeng, D. Li, Z. Liu, and M. Sun, "MiniCPM-V: A GPT-4V Level MLLM on Your Phone," OpenBMB, 2024. [Online]. Available: https://github.com/OpenBMB/MiniCPM-V.

[4] J. P. Cardenuto, S. Mandelli, D. Moreira, P. Bestagini, E. Delp, and A. Rocha, "Explainable Artifacts for Synthetic Western Blot Source Attribution," Artificial Intelligence Lab., Recod.ai, Institute of Computing, Universidade Estadual de Campinas (UNICAMP), Campinas, SP, Brazil, 2024.

[5] D. Gragnaniello, D. Cozzolino, F. Marra, G. Poggi, and L. Verdoliva, "Are GAN Generated Images Easy to Detect? A Critical Analysis of the State-of-the-Art," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*, 2021, pp. 1–6. [Online]. Available: https://doi.org/10.1109/ICME51207.2021.9428429.