AI-Generated Image Detection

Aaron Weissberg
RWTH Aachen

Bohdana Zlotenko
Université Paris-Saclay

Emirhan Bilgiç
Université Paris-Saclay

Abstract

The rapid advancements in generative models have made it increasingly challenging to distinguish between real and AI-generated images. In this work, we compare three different AI-Image detection methods using a robust evaluation framework. We conduct comprehensive experiments across multiple datasets and generative models, ensuring a thorough validation of our approaches. We show that statistical and especially spectral image features can be effectively used for detecting AI-generated images, even without deep learning. The project is available at: https://github.com/emirhanbilgic/AI-Generated-Image-Detection.

1. Introduction

The rise of powerful generative models, such as DALLE [7], Stable Diffusion [8], and BigGAN [2], has introduced new challenges in detecting AI-generated images. These models can produce highly realistic images, making it difficult to differentiate them from authentic ones.

To address this, we design an evaluation framework consisting of three experimental setups: (1) in-domain experiments, where real and fake images are compared within specific datasets, (2) in-method experiments, analyzing performance against individual generative models, and (3) a comprehensive real vs. fake study, combining multiple datasets and generative methods. Our results provide valuable insights into the strengths and limitations of existing detection techniques.

We use three main methods to classify the three main experimental setups: (1) Statistical Feature Extractions (2) Bag-of-Visual-Words and (3) Spectral-Based Detection.

2. Related Work

The field of AI-generated image detection lacks standardized datasets akin to those available for deepfake detection [4, 10].

Deep learning methods dominate the field of AI-generated image detection, with most approaches leveraging neural networks to identify artifacts left by generative models [4, 9]. Recent advancements have introduced different techniques:

- **Synthbuster**: This method focuses on detecting frequency artifacts in images generated by diffusion models, utilizing spectral analysis and a high-pass cross-difference filter to distinguish real from synthetic images [1].

- **CLIP-based Detection**: Leveraging features from the pre-trained vision-language model CLIP, this approach achieves robust generalization across diverse generative models and post-processing scenarios with minimal training data. Notably, this method has been validated using a dataset that we will adopt in our work, as presented in the CVPRW 2024 paper [4].

Other notable works in this domain include methods that exploit pixel-level inconsistencies or statistical irregularities introduced by generative processes [5, 6]. Some studies analyze noise patterns or compression artifacts to differentiate between real and synthetic images [9].

However, the methods have generalization problems across different datasets. Additionally, since they are deep learning-based, they require significant resources. We aim to demonstrate that this problem can be solved more effectively without using deep learning.

In our work, we employ a spectral feature extraction method that is an adjusted version of the one presented in Synthbuster [1], combining it with a basic classifier: the Random Forest. By adopting a non-deep-learning approach, we seek to provide an efficient solution. Additionally, we will utilize the dataset introduced in the state-of-the-art CLIP-based Detection paper [4] to evaluate our approach.

3. Methodology

To avoid spurious correlations from different image sizes we resize all images to 512 by 512.

3.1. Statistical Feature Extraction

This method involves extracting statistical features from images for classification. The extracted features include corner

detection metrics, keypoint densities, gradient-based statistics, edge density, and color channel statistics. These features are computed using the following steps:

3.1.1. Corner Detection Features

Harris Corner Detection is used to identify corners in grayscale images. Two features are extracted:

- **Number of Corners**: Total number of detected corners.
- **Corner Density**: Ratio of detected corners to the total number of pixels.

3.1.2. Keypoint Detection Features

Keypoints are detected using SIFT and ORB algorithms. For each method:

- **Number of Keypoints**: Total number of detected keypoints.
- **Keypoint Density**: Ratio of detected keypoints to the total number of pixels.

3.1.3. Gradient-Based Statistics

Image gradients are computed using the Sobel operator, and the following features are derived:

- **Gradient Mean**: Average gradient magnitude.
- **Gradient Standard Deviation**: Standard deviation of gradient magnitudes.
- **Gradient Maximum**: Maximum gradient magnitude.

3.1.4. Edge Density

The Canny edge detection algorithm is used to compute the proportion of edge pixels relative to the total number of pixels.

3.1.5. Color Channel Statistics

For each RGB channel, intensity histograms are analyzed to extract:

- **Mean**: Average intensity value.
- **Standard Deviation**: Spread of intensity values.
- **Skewness**: Asymmetry in intensity distribution.
- **Kurtosis**: Tailedness of intensity distribution.

3.1.6. Feature Aggregation

All extracted features are combined into a single feature vector for each image. This vector includes:

- Corner detection metrics (*number of corners*, *corner density*).
- Keypoint metrics (*SIFT number*, *SIFT density*, *ORB number*, *ORB density*).
- Gradient statistics (*gradient mean*, *gradient standard deviation*, *gradient maximum*).
- Edge density.
- Color statistics (*mean*, *standard deviation*, *skewness*, and *kurtosis* for RGB channels).

These features are then used as input for Random Forest.

3.2. Bag-of-Visual-Words

This method follows a traditional bag-of-visual-words approach.

After reading the images in greyscale, we start by using the Scale-invariant feature transform (SIFT) algorithm to detect key points and extract descriptor feature vectors from them. We cluster the vectors with minibatch K-means to create our vocabulary of visual words. Based on this, each image is represented as a bag-of-words histogram.

The histogram values are then used as a feature vector to classify the images with a Random Forest.

3.3. Spectral-Based Detection

It is a well-studied fact that synthetically generated images contain artifacts that appear periodically. This is especially true for GAN-generated images but also to a lesser extent for diffusion-based models[3]. [1] presents a simple procedure to make these artifacts detectable using a strong high-pass filter followed by a fast Fourier transform.

The author proposes using the cross-difference filter to make reveal the image generation artifacts. Cross-difference was originally introduced for artifacts that occur during JPEG compression. Its value at some pixel $[i, j]$ can be written as the absolute difference between the two diagonals of a 2 by 2 kernel with its upper left corner placed at said pixel, namely

$$CD[i,j] = |I[i,j] - I[i+1,j] - I[i,j-1] + I[i+1,j+1]| \,. \tag{1}$$

As stated in the last section, this filter acts as a high-pass filter, reducing the low-frequency components of the image, thereby making the high-frequency artifacts more apparent. Note that we do this and the following operation for each color channel independently.

The next step is to apply a two-dimensional FFT to the filtered image (to each color channel). We also normalize the result with mean and standard deviation over the image. In the resulting image the frequency artifacts are now directly visible, as we discuss in detail in section 4.3.

To extract features from an individual image spectral representation we sample the image value at the lattice points with period image size$/8$. So in our case, since we have quadratic images of size 512 by 512, any pixel with either component being 0, 64, 128, 192, 256, 320, 384, 448, or 512. Since we have found in our exploratory analysis that the exact position can sometimes be off by a pixel, we actually take the maximum value over a 3 by 3 kernel centered at the lattice point. Since we do this for each color channel and also subtract the trivial middle point, we get a $3 \cdot (9 \cdot 9 - 1) = 240$ dimensional vector.

As a classifier, we use a standard Random Forest to which we feed the feature vector.

4. Experiments

4.1. Dataset

We use the dataset from the CVPR workshop paper [4]. Due to memory constraints we do not use the whole dataset, but the part that was originally used for testing. When we refer to the dataset in the following sections, we are referring to this subset specifically.

The dataset contains 4000 real images from different datasets, namely Common Objects in Context (COCO), Flickr-Faces-HQ (FFHQ), ImageNet and Large Scale Scene Understanding (LSUN). It also contains synthetic images created with 20 different models with 1000 examples each. The collection includes images from both GAN based models like BigGAN or StyleGAN 2 and from diffusion based models like Stable Diffusion 2 and DALLE 2.

In terms of image content, most of the images stem from more general datasets like ImageNet or COCO but there are also datasets of images of faces like FFHQ and LSUN.

(a) Real Image (b) DALLE2 (c) DALLE3

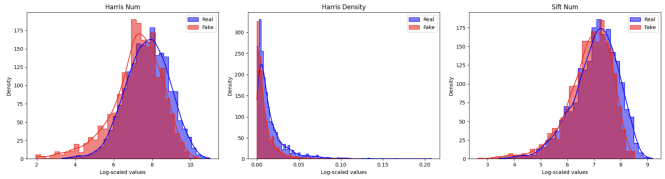
(d) SD2 (e) SDXL (f) BIGGAN

Figure 1. Examples from the Dataset.

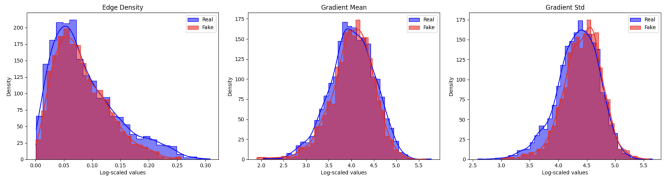
To get a comprehensive idea we will conduct three types of experiments. First, we will compare real images from a specific dataset with synthetic ones whose models were trained on that same dataset (in-domain experiments). This has the advantage of avoiding spurious correlations that stem from different image content (e.g. comparing real images of faces to synthetic ones displaying cars). We will also do experiments only using synthetic images stemming from a specific model type to judge which method our approaches can handle better (in-method experiments). Lastly, we will do a more comprehensive experiment, combining multiple real and multiple synthetic subdatasets (comprehensive or mix experiments).

4.2. Exploratory Data Analysis

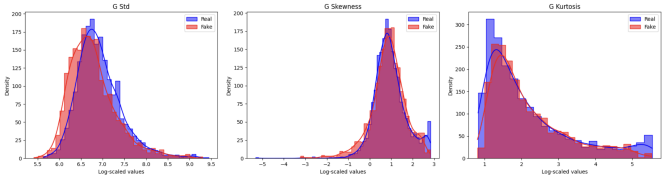
To understand the different characteristics of the datasets, we created histograms of different features used in our first method. We ultimately saw that this statistical approach was able to outperform the bag of words method.



(a) Comparison of Harris Number, Harris Density, and SIFT Number



(b) Comparison of Edge Density, Gradient Mean, and Gradient Std.



(c) Comparison of Std., Skewness, and Kurtosis of Green Channel

Figure 2. Comparison of Different Characteristics of Datasets used in the first In-Method Experiment (Real vs. DALLE2/3).

As shown in Figure 2, the histograms provide an overview of the dataset characteristics. While most features exhibit overlap, they can still prove useful during the training of a Random Forest model. In different dataset configurations, certain features appear to have less overlap, and these features are typically more informative (having higher feature importance). For the complete histograms of all dataset configurations, please refer to our GitHub page: https://github.com/emirhanbilgic/AI-Generated-Image-Detection/blob/main/experiments_Statistical_Feature_Extraction_BoW.ipynb.

4.3. FFT Visualizations

Like in [1] we visualize the artifacts by averaging the resulting FFT images over datasets of different real and generated images. To make the peaks in the resulting images more apparent we apply a morphological dilation operator (of kernel size five), resulting in the images in figure 3.

With the real images from the FFHQ (Flickr-Faces-HQ) dataset, we get a uniform image with no apparent peaks besides from the center one. If we now compare this to the

result for Stable Diffusion 2, we can see a clear difference. Here peaks appear in a regular quadratic pattern, with a period of image size$/8$ in both directions.

We see a similar effect for all diffusion-based images with different peaks in the lattice being more or less visible depending on the method. For example, in the DALLE 3 image, only the peaks with a period of image size$/4$ are visible. Interestingly for DALLE 2 (specifically it is 2.1), we cannot see artifacts that are as obvious. In the original paper, the corresponding image for DALLE 2 only showed some artifacts on the horizontal axis (in their experiments and ours, the method still performed well in this case).

For the GAN-based synthetic images our result appears even more different from the real images. While we do see peaks at the same lattice points, there are also many appearing with a smaller period (the image is quite noisy but around image size$/32$).

Interestingly, both for COCO and ImageNet we see some artifacts on the horizontal and vertical central axis, plus one above and below the horizontal axis middle point each. These artifacts stem from image compression as the images in our ImageNet data are .jpegs while those in coco are .jpgs.

Overall, for the images for which we have an equivalent in the original paper, our results look quite similar. Since the authors used a different dataset this emphasizes that the observations are really an effect of the used methods themselves, not the datasets.

4.4. In-Domain Experiments

These experiments focus on evaluating our method within specific datasets.

FFHQ: Fake vs. Real We assess the performance of our approach in distinguishing real images from fake ones within the FFHQ dataset.

COCO: Fake vs. Real We evaluate our method on the COCO dataset to determine its effectiveness in differentiating between real and AI-generated images.

4.5. In-Method Experiments

In this set of experiments, we analyze the performance of our model against specific generative methods.

Real vs. DALLE 2/3 (COCO-style) We compare real images against those generated by DALLE 2 and DALLE 3, focusing mainly on images similar to the COCO dataset.

Real vs. SDXL/SD2 (COCO-style) This experiment examines how well our model can differentiate real images from those generated by Stable Diffusion XL (SDXL) and Stable Diffusion 2 (SD2), particularly within the COCO-like domain.

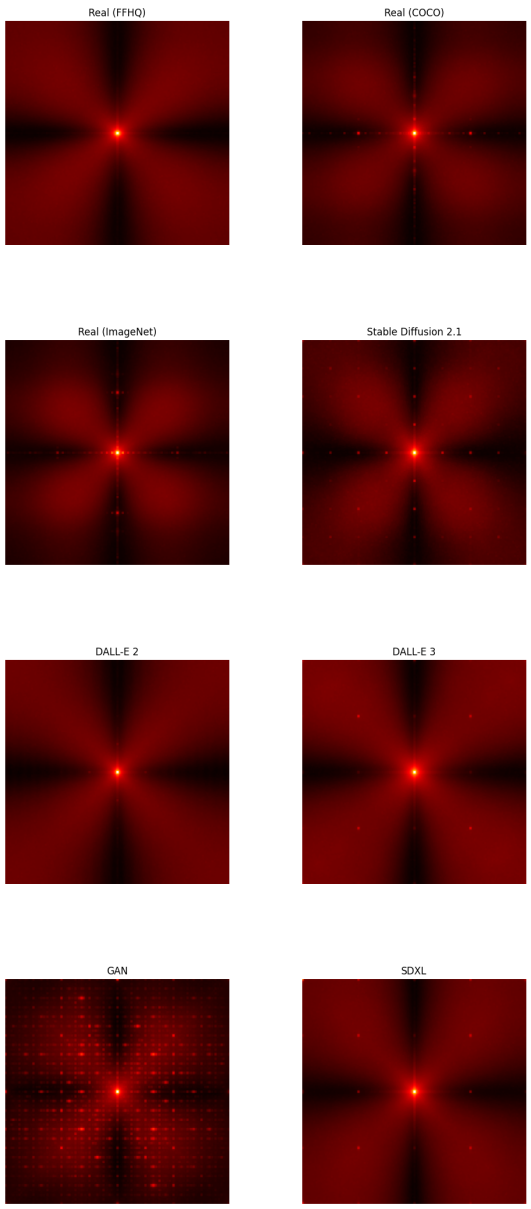


Figure 3. Comparison of the visualizations from the spectral approach for different image generation methods.

4.6. Comprehensive Real vs. Fake Experiments

We conduct a broader experiment involving multiple datasets and generative methods.

Real (COCO + ImageNet) vs. Fake (DALLE 3 + Stable Diffusion + BigGAN) This final experiment evaluates our method's ability to distinguish real images from a combination of generative models, including DALLE 3, Stable Diffusion, and BigGAN, using real images from COCO and ImageNet.

5. Evaluation

Category	Experiment	F1	Acc	Recall
In-Domain	FFHQ - SFE	0.72	0.73	0.73
	FFHQ - BoVW	0.74	0.63	0.85
	FFHQ - SBD	**0.87**	**0.88**	**0.88**
	COCO - SFE	0.75	0.75	0.75
	COCO - BoVW	0.65	0.73	0.55
	COCO - SBD	**0.99**	**0.99**	**0.99**
In-Method	DALLE 2/3-SFE	0.72	0.72	0.72
	DALLE 2/3-BoVW	0.45	0.72	0.32
	DALLE 2/3-SBD	**0.97**	**0.97**	**0.97**
	SDXL/SD2-SFE	0.77	0.77	0.77
	SDXL/SD2-BoVW	0.44	0.69	0.31
	SDXL/SD2-SBD	**0.98**	**0.98**	**0.98**
Mix	Combined-SFE	0.73	0.71	0.80
	Combined-BoVW	0.41	0.70	0.91
	Combined-SBD	**0.96**	**0.96**	**0.96**

Table 1. Performance comparison using F1 Score, Accuracy, and Precision, across different experimental settings. SFE stands for Statistical Feature Extraction, BoVW stands for Bag of Visual Words, and SBD stands for Spectral Based Detection.

The results for the experiments are collected in table 1. We start with the In-Domain results. In both cases SBD outperformed the other two methods by a wide margin, doing particularly well in the COCO experiment. The other two methods are relatively close with SFE being slightly better than BoVW.

SBD also performed worse on the FFHQ than the COCO task, suggesting that faces might be more difficult for the model.

For the In-Method results, SBD also performs very well. Similarly, SFE and BoVW have comparable accuracies, but the latter has a much lower F1 and recall score, suggesting the model is classifying many real images as synthetic (0 means synthetic in our labeling).

For the comprehensive task, we get a similar picture.

Overall the SBD method clearly outperforms the other two. We can not see a significant drop in performance between methods or when combining domains and methods but the model did perform worse (but still very good) on FFHQ.

The SFE outperformed BoVW, with the latter being imbalanced towards classifying real images as fake.

6. Conclusion

In this work, we introduced an evaluation framework to assess the effectiveness of real vs. AI-generated image detection methods. Through a series of in-domain, in-method, and comprehensive experiments, we analyzed the performance of different detection approaches across multiple datasets and generative models. Our findings highlight that images' visual, statistical and especially spectral features are useful for detecting AI-generated images and that achieving high performance in this task does not require large deep learning models.

7. Limitations and Future Directions

While we did test the methods in various settings and combinations of datasets, ultimately a larger scale evaluation would be desirable. While we specifically designed our experiments to avoid problems stemming from spurious correlations, it can still not be ruled out fully that they may have played a role in some pairings.

Also note that we did a limited hyperparameter tuning, since we did not aim for maximally possible performance. It is quite likely that different hyperparameters or other classifiers besides Random Forest (e.g. gradient boosting methods) would have yielded better results in some cases.

In terms of future directions, the most obvious next step would be to extend our task to a multi-class setting in which one additionally tries to predict the model with which the image was created. Especially for the spectral approach, this seems fruitful since the frequency patterns do look different. It might also be interesting to investigate how the configuration of the different generators might affect the frequency patterns of the artifacts.

References

[1] Quentin Bammey. Synthbuster: Towards detection of diffusion model generated images. *IEEE Open Journal of Signal Processing*, 2023. 1, 2, 3

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations (ICLR)*, 2018. 1

[3] R. Corvi, D. Cozzolino, G. Poggi, K. Nagano, and L. Verdoliva. Intriguing properties of synthetic images: From generative adversarial networks to diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 973–982, 2023. 2

[4] Davide Cozzolino, Giovanni Poggi, Riccardo Corvi, Matthias Nießner, and Luisa Verdoliva. Raising the bar of ai-generated image detection with clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4356–4366, 2024. 1, 3

[5] Diego Gragnaniello, Francesco Marra, and Luisa Verdoliva. Detection of ai-generated synthetic faces. In *Handbook of Digital Face Manipulation and Detection: From Deepfakes to Morphing Attacks*, pages 191–212. Springer International Publishing, Cham, 2022. 1

[6] Chenhao Kong, Aoran Luo, Shuaiqiang Wang, Haojie Li, Anderson Rocha, and Alex C. Kot. Pixel-inconsistency mod-

eling for image manipulation localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 1

[7] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint*, arXiv:2204.06125v1, 2022. 1

[8] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 1

[9] Ziqi Sha, Yuxuan Tan, Minghui Li, Michael Backes, and Yang Zhang. Zerofake: Zero-shot detection of fake images generated and edited by text-to-image generation models. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 4852–4866, 2024. 1

[10] Tianyi Zhang. Deepfake generation and detection, a survey. *Multimedia Tools and Applications*, 81(5):6259–6276, 2022. 1