Internship Report
Image Deepfake Detection with Reference

Yingwei TANG

July 2025

1 Introduction

In the past decade, the development of artificial intelligence has promoted a variety of intelligent applications. Among them, the generative AI, popular in recent years, is undoubtedly one of the most popular artificial intelligences that has directly entered people's lives.

In the field of vision, advances in generative models have brought great progress in synthesizing images(and generating videos). These models can generate very realistic images that are difficult for the human eye to distinguish, such as the Stable Diffusion series[1] and the commercial model Midjourney[2], etc. With the latest models, people can guide the generation of fake images through text descriptions prompts, reference real images, etc., which further enhances the freedom of image generation. However, everything has two sides. The convenience of generating images has also led to the proliferation of fake images. Difficult-to-distinguish fake images have appeared on the Internet and social software, causing people to worry about the widespread spread of false information. Current state-of-the-art (SOTA) generative models have exhibited the ability to generate images that pose a significant challenge to human perception and discrimination. Humans can only achieve an accuracy rate of 61.3%[3] when discriminating between real images and AI-generated fake images. Therefore, there is an urgent need to develop an effective detector that can accurately identify fake images generated by these advanced generative models.

The earliest image detection focused on portraits and faces[4], because fake images were a big problem in the political field at that time, and the level of generative models was not enough to synthesize credible images without main objects. With the advancement of generative models, images of various structures can be generated with high fidelity, which has expanded detection to all possible general image fields. Early research [5] discovered that the upsampling process in Generative Adversarial Network (GAN [6]) leaves periodic artifacts in the spatial or frequency domain of the generated images, allowing for effective detection of low-quality generated images by checking these specific traces. However, with the introduction and development of diffusion-based models[7], synthetic artifacts have been alleviated a lot and it is difficult to directly use them as an indicator to distinguish the true from the false. This

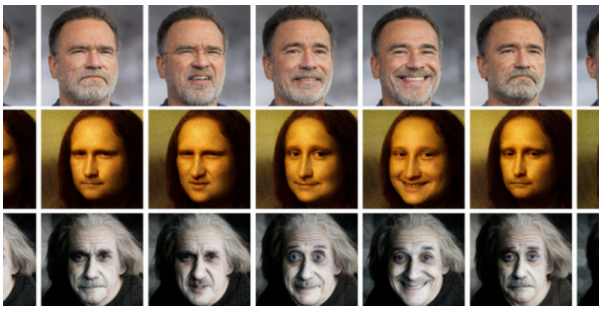


Figure 1: Deepfake

has led to the emergence of detection methods based on supervised learning neural networks, which learn common features of generated images by training on large datasets of real and fake images.

Nevertheless, the amount of data and computational cost required to train a deep neural network model are very high. In today's world where various generative models emerge in an endless stream, many supervised learning detectors that are not cross-generator generalizable enough are difficult to apply in many occasions where there are weak hardware conditions, their cost-effectiveness is too low. At the same time, in addition to using neural network models to learn feature spaces, are there other more direct and interpretable aspects that can be studied to distinguish real images from fake images? This question may help us develop a method for detection in an untrained manner without using or using a small amount of large deep neural networks (for example, only using pre-trained high-performance multimodal or visual large models).

In actual application scenarios, images that cause trouble to people generally have a theme or main meaning, which we can call the category or class of the image, and images of the same class generally have many real images for us to query. On the other side, we usually can't know the generator's identity when we are attacked by fake images (what we called "zero-day"). With this "real data base versus test input" situation, we naturally think of another field of visual inspection: anomaly detection. An exploration inspired by anomaly detection (or out-of-distribution detection, OOD), for solutions such as statistical distribution and time-frequency information analysis incorporating (or not) the encoding capabilities of pre-trained large models, is the research goal of this paper.

We start by analyzing the difference in statistical information between real and fake images, and study whether there is a gap between real and fake images on certain metrics. Secondly, we focus on whether we can analyze the frequency information of the image and use some transformations to obtain more information for detection. Finally, inspired by some image anomaly detection algorithms proposed in recent years, we apply some of the indicators mentioned in them to guide the final detection method, and we make a comparison between our new methods with supervised-learning methods.

As a result, we find that our method using dinov2's cls token as input embedding of Mahalanobis-distance scoring algorithm is comparable to some SOTA traing-base or traing-free methods.

We summarize our main contributions as follows:

1 We do a large study about different ways to understand what information gap between real and fake images is. We study and analyze various distribution statistics and spatial-frequency transforms.

2 Inspired by anomaly detection, we develop some new methods based on research and have a comparison after experiments. At least one method is comparable to existing SOTA methods.

2 Background

2.1 Generative models

Generative models are a class of machine learning models designed to generate new data samples that resemble a given dataset (always the real-world data). Among the most prominent are large language models (LLMs)[8] based on the Transformer architecture, which predict the next token in a sequence given its context and can generate coherent and contextually relevant text across a wide range of tasks. Generative Adversarial Networks (GANs)[6] consist of a generator and a discriminator in a minimax game, where the generator learns to produce realistic data (such as images) that can fool the discriminator, which tries to distinguish real from generated samples. Diffusion models[7], a new powerful generation architecture, start from random noise and gradually refine it through a learned denoising process to produce high-quality generation. Recently, multimodal large models[9] have emerged, capable of understanding and generating content across multiple data types, such as text, images, audio, or video, by jointly modeling different modalities through shared latent representations. These models leverage large-scale pretraining and cross-modal attention mechanisms to perform tasks like image captioning, text-to-image synthesis, or visual question answering, pushing the boundaries of generative AI across domains.

2.2 Image Generation

GANs and diffusion models are mainstream techniques for image generation.

GAN has brought significant quality improvements in image generation in the past decades. BigGAN[10] is a representative method in GAN family. BigGAN makes three contributions, including architectural changes, sampling techniques, and instabilities reduction techniques. BigGAN applies orthogonal regularization to the generator to improve training stability.

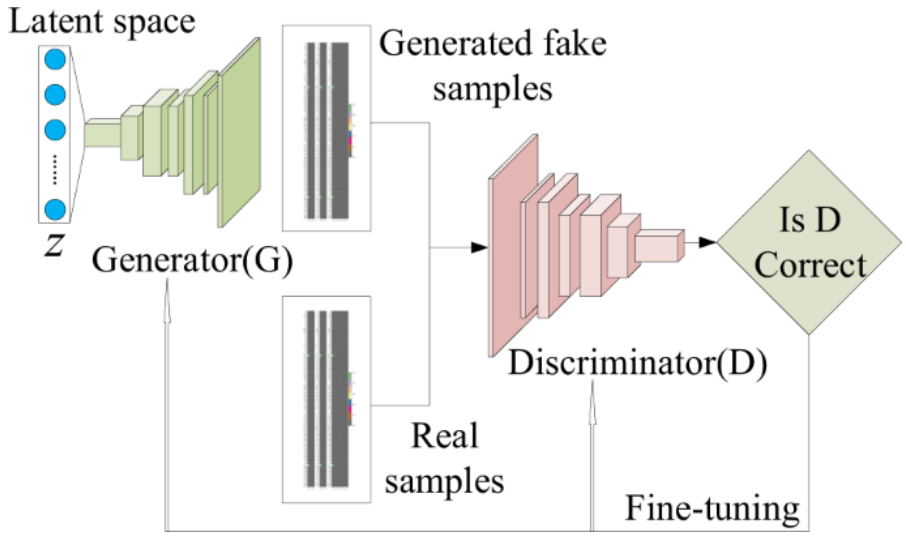


Figure 2: GAN structure

Diffusion Model has recently achieved remarkable performances in image synthesis. In diffusion-based models, models such as the Denoising Diffusion Probabilistic Model (DDPM [7]) and LDM [11] have shown impressive results in generating high-quality images. In the advanced domain of conditional image generation, which refers to generating images with better control based on specific input conditions (such as text descriptions or semantic labels), the ablative diffusion model (ADM [12]) achieves an efficient text-to-image generation architecture by removing the self-attention mechanism, proposes a diffusion model which achieves better sample quality than GANs. GLIDE [13] is a diffusion model for text-conditional image synthesis. VQDM [14] proposes a latent-space method that eliminates the undirectional bias with previous methods and incorporates a mask-and-replace diffusion mechanism to alleviate the accumulation of errors. Diffusion-based Transformer (DiT [15]) replaces U-Net in LDM with Transformer and uses Transformer's ability to capture global context to improve the quality of text-to-image generation, at the same time, it also gets the advantage of good scalability of the transformer architecture. These methods give rise to popular text-to-image generation tools such as Stable Diffusion [1] and Midjourney [2]. Wukong [16] is a large-scale text-to-image generative model based on the diffusion model. This model is trained on the largest Chinese open-source multimodal dataset, the Wukong dataset, making it particularly suitable for Chinese language processing.

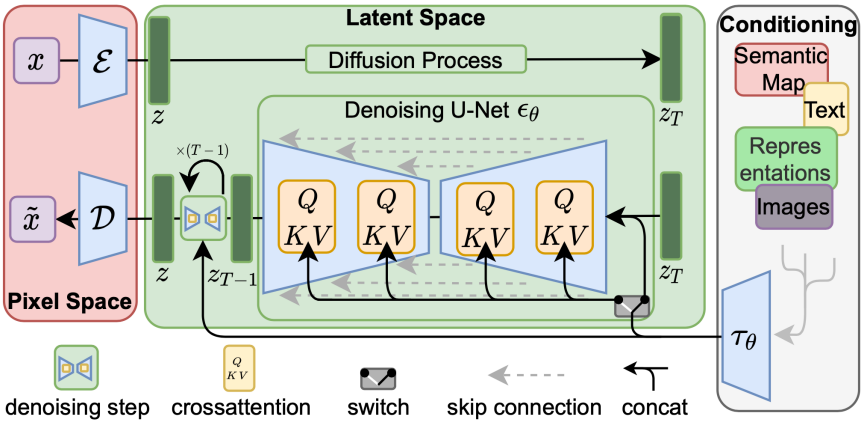


Figure 3: Diffusion structure

2.3 Fake Image Detection

Early efforts focused on leveraging hand-crafted features, such as color cues, saturation cues, and co-occurrence features, to identify machine-edited images. However, these features are no longer reliable indicators, as modern generative models have largely overcome these limitations. Another successful strategy is to analyze images in the frequency domain, where the generated images exhibit distinguishable artifacts. However, these artifacts are only evident in the upsampling model and cannot be used to detect images generated by diffusion models [17].

In recent years, learning-based approaches are the most popular. Networks based on CNN or transformer structure are trained on dataset of real and fake images to be detectors. This method utilizes the encoding ability of high-perfomance NN models.

Then, more research on the feature space learned by models and more observations on fake images are following. In reconstruction-based methods, DIRE [18] finds that diffusion models can reconstruct diffusion-generated images more accurately than real images, utilizing the reconstruction error to train the detector. Diffusion Reconstruction Contrastive Learning (DRCT) [19], enhances the generalizability of the existing detectors and the performance of reconstruction-based methods. Faced with an increasing number of generators and their distinct forensic traces, continuous learning becomes a choice for detector. E3 [20], a new approach with an expert knowledge fusion network that can update synthetic image detectors to accurately detect newly emerging generators, while requiring only minimal amounts of training data to be retained in a memory buffer.

Furthermore, in order to reduce the computational cost, training-free methods are proposed. AEROBLADE [21] detects generated images solely based on the reconstruction error of the image passing through an autoencoder. Nevertheless, it is only effective for images generated by LDM using similar autoencoders, and its generalizability remains a challenge. In order to address the problem of zero-day deepfake, multimodal large models come into people's attention. The concept of "fact checking" [22], adapted from fake news detection, is proposed to detect fake media(images and videos), by computing truth score (similarity function) using off-the-shelf features got from pretrained multimodal large models.

2.4 Image Anomaly Detection - Out-of-distribution (OOD) Detection

Humans are able to detect heterogeneous or unexpected patterns in a set of homogeneous natural images. This task is known as anomaly or novelty detection and has a large number of applications, among which visual industrial inspections. Anomaly detection is a binary classification between the normal and the anomalous classes. Out-of-distribution (OOD) detection consists of identifying whether a given test sample significantly deviates from the known information of in-distribution (ID) data.

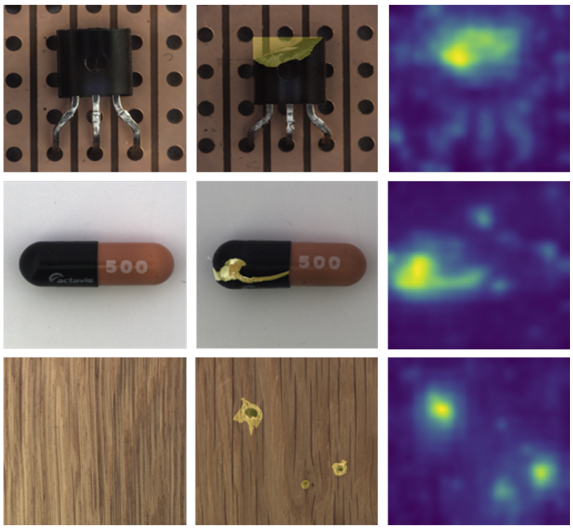


Figure 4: Image Anomaly Detection

In adversarial discriminators trained with Batch Normalization (BN), real and adversarial samples form distinct domains with unique batch statistics. DisCoPatch [23] is proposed, an unsupervised Adversarial Variational Autoencoder (VAE) framework that harnesses this mechanism. At the same time, based on assumptions of some probability models, several measures of data distribution are used for anomaly detection. Other methods optimize the utilization of the ground truth basis and directly use min-max distance based on the nearest neighbors calculation.

3 Dataset

3.1 Original Large Dataset

In order to have a acceptable generalizability cross different image generators for our analysis and research on new methods, we must do our work on images synthesized by different generators in GAN and Diffusion families.

GenImage [24] dataset is the final choice after checking many deepfake or anomaly image datasets. GenImage employs all the real images in ImageNet, image generation in GenImage leverages 1000 distinct labels in ImageNet, ensuring a near-equal distribution of real and generated images across each class. GenImage comprises 2,681,167 images, segregated into 1,331,167 real and 1,350,000 fake images. The real images are subdivided into 1,281,167 images for training and 50,000 for testing. With ImageNet [25] providing 1000 distinct image classes, it generates 1350 images for each class, out of which 1300 are allocated for training and the remaining 50 for testing. To address the problem of detecting images generated by SOTA generators, it employs eight generative models for image generation, namely BigGAN, GLIDE, VQDM, Stable Diffusion V1.4, Stable Diffusion V1.5, ADM, Midjourney, and Wukong. Each generator produces nearly the same number of images for each class, with 162 images for training and 6 for testing, with the exception of Stable Diffusion V1.5, which generates 166 images for training and 8 for testing. A combination of the fake images generated by a generator and their corresponding real images can be considered as a subset, such as Stable Diffusion V1.4 subset. The real images are not shared across subsets.

The generated images are shown in Figure 5. It can be observed that overall the generated images are similar to the real images in ImageNet. Animals and plants succeed in keeping the appearance of the target object consistent, e.g., samoyed with a similar appearance, and they differ in movement, perspective, and background. The appearance of the targets also varies for objects such as candles and

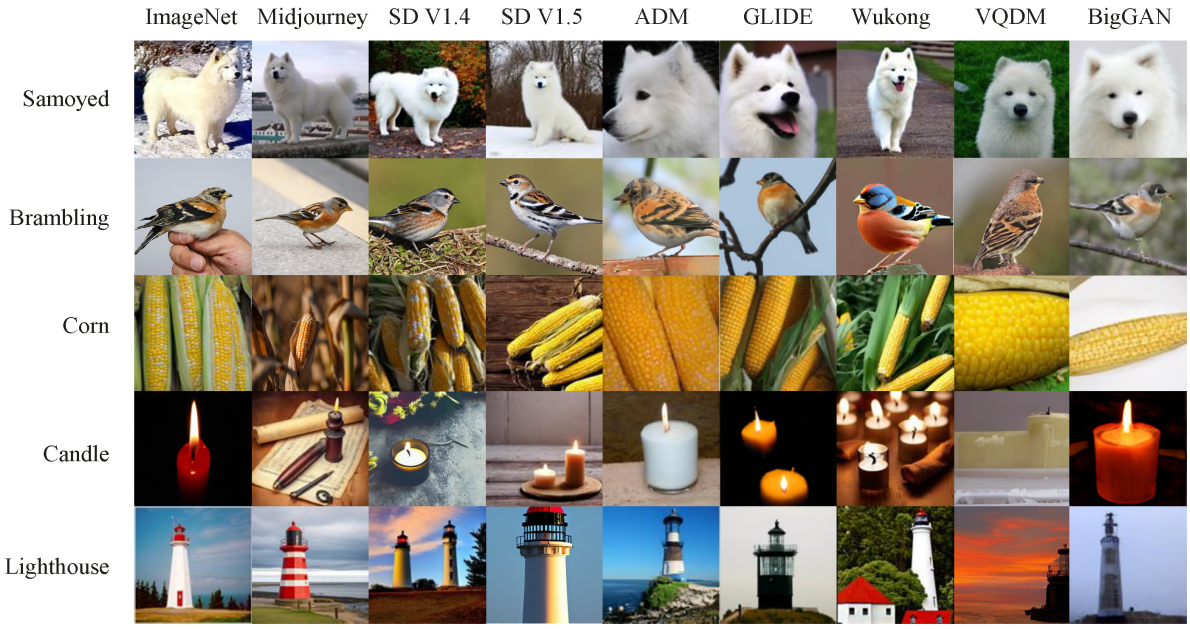


Figure 5: GenImage Dataset

lighthouses. Therefore, the generated images have a high degree of variability and reasonableness.

3.2 Extrated Small Dataset

Without a high-performance hardware environment, we cannot perform large-scale analysis, testing, and research on the original large dataset, so we choose to extract a small dataset that inherits its diversity and is representative, and use it for preliminary and mid-term work.

We randomly selected 10 classes that have both real images and generated images (because we found that one of the 1000 classes has no real ImageNet images) from 1000 (in fact 999) classes, and selected a subset of three generators: BigGAN, Midjourney, and Stable Diffusion V1.5, to form a 10-class set with three groups of 162 fake images per class, plus two groups of 162 real images per class. The real images are randomly extracted from the real images in the training set according to the category. For convenience, we call an image folder "cls(class id)/[bgan, midj, sd15, nature, nature2]". Among them, two groups of real images are extracted because "nature" is used as ground truth, and "nature2" is combined with the fake image set to calculate metrics such as receiver operating characteristic curve (ROC).

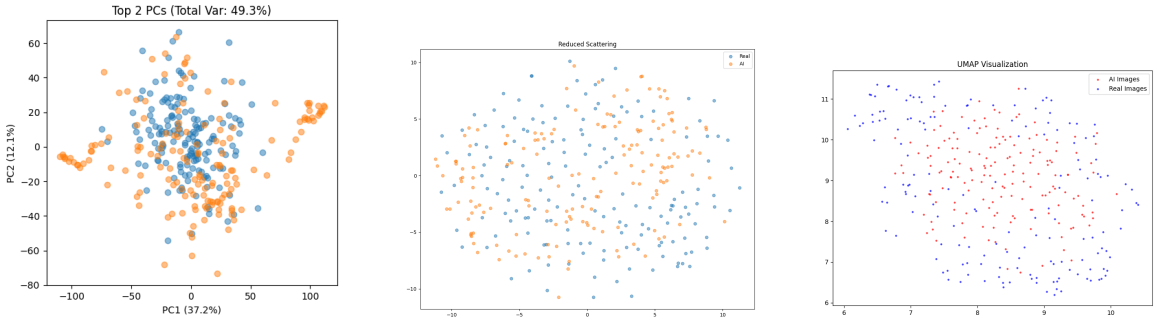
4 Preliminary Analysis

In the first step, we hope to analyze the real image data and fake image data through different metrics and different embedder models to understand the difference in the data itself and compare different direct distinction methods.

Regarding the choice of embedder models, based on studying many articles, we observed that models from clip and dino-v2 families are very common in most studies. So we also use openai/clip ViT-B/32 multimodal model and facebook/dino-v2-base image model. We will refer to them as clip and dinov2 below.

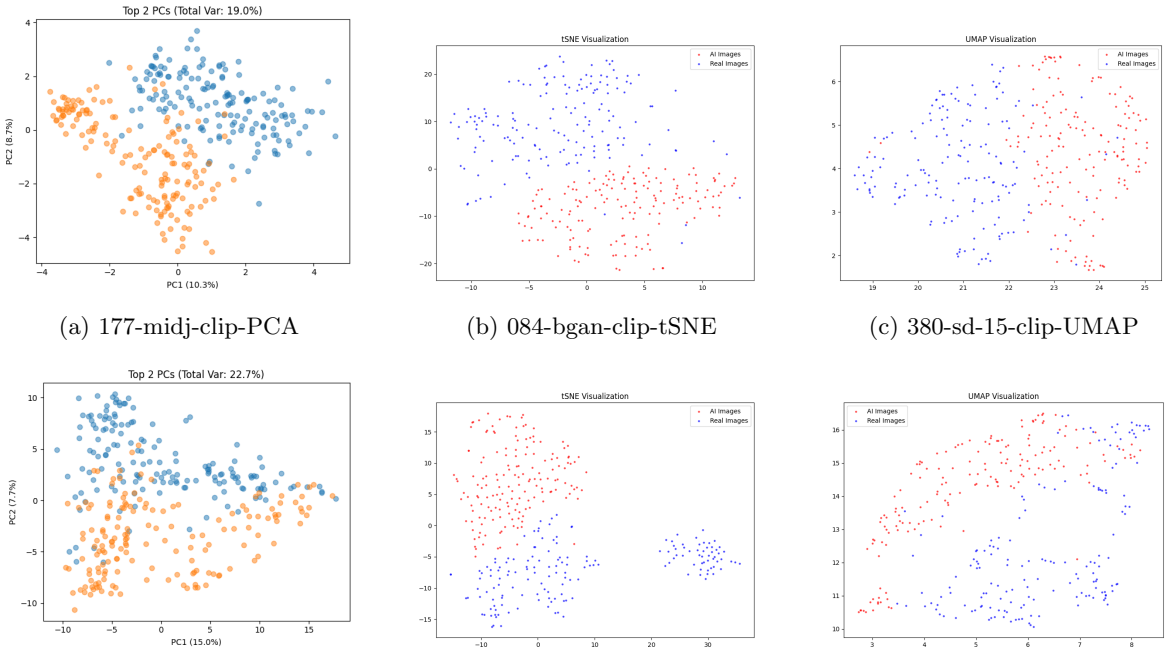
4.1 Seperation by Dimension Reduction

Naively, we tried three dimensionality reduction methods: PCA, tSNE, UMAP by performing dimensionality reduction on raw image data, clip embeddings and dinov2 embeddings.



(a) 177-midj-PCA (b) 084-bgan-tSNE (c) 380-sd-15-UMAP

Figure 6: Dimension reduction on original data



(a) 177-midj-clip-PCA (b) 084-bgan-clip-tSNE (c) 380-sd-15-clip-UMAP

(d) 177-midj-dinov2-PCA (e) 084-bgan-dinov2-tSNE (f) 380-sd-15-dinov2-UMAP

Figure 7: Dimension Reduction after embedding

Here we only present a few examples in Figure 6 and Figure 7 , but the conclusion is validated for the majority of classes: direct dimension reduction on raw image data can't separate the fake images from real images, dimension reduction techniques on clip and dinov2 embeddings of images data have much better results. It proves that embedder models are helpful in representing more image information.

It is generally believed that real images are in certain low-dimensional manifolds in high-dimensional space, and false images generated by generative large models are essentially in relatively limited areas in high-dimensional space. Their manifolds do not necessarily match, and multimodal or visual large

models can understand (encode) these images again in high-dimensional space, making it possible to distinguish them.

Besides, we can see that results depend more on classes than on generators, showing that different image contents have different features in high-dimensional space which hinders the development of universal detection methods.

4.2 MMD/W2D/FD - Distance Metrics

Metrics based on distribution statistics are very suitable for our task, because they can analyze and compare distributions, which people use to construct tests (statistical or not) to determine if two samples are drawn from different distributions.

Maximum Mean Discrepancy (MMD) [26] is a statistical measure used to quantify the difference between two probability distributions based on samples drawn from them. It is widely used in areas such as domain adaptation, generative modeling, and hypothesis testing. The MMD can be computed in quadratic time, although efficient linear time approximations are available. The idea is: We test whether distributions p and q are different on the basis of samples drawn from each of them, by finding a well behaved (e.g., smooth) function which is large on the points drawn from p, and small (as negative as possible) on the points from q. We use as our test statistic the difference between the mean function values on the two samples; when this is large, the samples are likely from different distributions. We call this test statistic the Maximum Mean Discrepancy (MMD). The process is to map the distributions into a reproducing kernel Hilbert space (RKHS) using a kernel function, and then compute the distance between their mean embeddings in that space. A smaller MMD indicates that the two distributions are more similar.

Given two distributions P and Q, and a reproducing kernel Hilbert space $\mathcal{H}$ with kernel $k(\cdot, \cdot)$, the squared Maximum Mean Discrepancy is defined as:

$$\mathrm{MMD}^2(P, Q) = \|\mathbb{E}_{x\sim P}[\phi(x)] - \mathbb{E}_{y\sim Q}[\phi(y)]\|^2_{\mathcal{H}} \tag{1}$$

This can be expanded as:

$$\mathrm{MMD}^2(P, Q) = \mathbb{E}_{x,x'\sim P}[k(x, x')] + \mathbb{E}_{y,y'\sim Q}[k(y, y')] - 2\mathbb{E}_{x\sim P, y\sim Q}[k(x, y)] \tag{2}$$

Let $X = \{x_1, \ldots, x_m\} \sim P$ and $Y = \{y_1, \ldots, y_n\} \sim Q$. The unbiased empirical estimate of the squared Maximum Mean Discrepancy is:

$$\mathrm{MMD}^2_u(X, Y) = \frac{1}{m(m-1)} \sum_{\substack{i,j=1 \\ i\neq j}}^{m} k(x_i, x_j) + \frac{1}{n(n-1)} \sum_{\substack{i,j=1 \\ i\neq j}}^{n} k(y_i, y_j) - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} k(x_i, y_j) \tag{3}$$

Wasserstein-2 Distance (W2D) (also known as the Earth Mover's Distance in some contexts) is a metric used to measure the distance between two probability distributions over a metric space. Specifically, it quantifies the minimum cost of transporting "mass" to transform one distribution into another, where the cost is proportional to the squared Euclidean distance the mass is moved. The Wasserstein-2 distance belongs to the family of optimal transport distances and is particularly useful in generative modeling, domain adaptation, and distributional robustness. Unlike some other divergence measures (e.g., KL divergence), W2D remains well-defined even when the distributions do not overlap. When the distributions are represented by empirical samples, the Wasserstein distance can be approximated by solving an optimal transport problem.

Let μ and ν be two probability distributions defined over a metric space $\mathcal{X} \subseteq \mathbb{R}^d$. The squared Wasserstein-2 distance between μ and ν is defined as:

$$W_2^2(\mu, \nu) = \inf_{\gamma\in\Gamma(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{X}} \|x - y\|^2 \, d\gamma(x, y) \tag{4}$$

where $\Gamma(\mu, \nu)$ is the set of all joint distributions γ on $\mathcal{X} \times \mathcal{X}$ with marginals μ and ν.

Given empirical distributions $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n}\delta_{x_i}$ and $\hat{\nu} = \frac{1}{n}\sum_{i=1}^{n}\delta_{y_i}$, x and y i.i.d. sampled from μ and ν respectively, the empirical Wasserstein-2 distance is:

$$W_2^2(\hat{\mu}, \hat{\nu}) = \min_{\pi \in \Pi_n} \frac{1}{n} \sum_{i=1}^{n} \|x_i - y_{\pi(i)}\|^2 \tag{5}$$

where Π_n is the set of all permutations over $\{1, 2, \ldots, n\}$.

Fréchet Distance (FD) is a measure of similarity between two probability distributions. Under the assumption that both distributions are multivariate Gaussians, the Fréchet Distance has a closed-form solution. It is especially popular in evaluating generative models, such as GANs, where it is used to compare the distribution of generated samples with that of real samples. In this context, it is often referred to as the **Fréchet Inception Distance (FID)** when computed using deep feature embeddings from pretrained networks. The FD considers both the difference in means and the difference in covariances between two distributions, making it sensitive to both shape and location of the distributions.

Let $\mathcal{N}(\mu_r, \Sigma_r)$ and $\mathcal{N}(\mu_g, \Sigma_g)$ be multivariate Gaussian distributions. The squared Fréchet Distance is:

$$\mathrm{FD}^2 = \|\mu_r - \mu_g\|^2 + \mathrm{Tr}\left(\Sigma_r + \Sigma_g - 2\left(\Sigma_r^{1/2}\Sigma_g\Sigma_r^{1/2}\right)^{1/2}\right) \tag{6}$$

Given empirical means $\hat{\mu}_r, \hat{\mu}_g$ and covariances $\hat{\Sigma}_r, \hat{\Sigma}_g$, the empirical Fréchet Distance is:

$$\mathrm{FD}^2 = \|\hat{\mu}_r - \hat{\mu}_g\|^2 + \mathrm{Tr}\left(\hat{\Sigma}_r + \hat{\Sigma}_g - 2\left(\hat{\Sigma}_r^{1/2}\hat{\Sigma}_g\hat{\Sigma}_r^{1/2}\right)^{1/2}\right) \tag{7}$$

4.3 Frequency Domain

In addition to large embedder models, we wonder also whether there are more methods to mine information from images. Refer to articles of CNNSpot [27] and GenImage [24], frequency domain information is interesting for dinstinguish the fake images from real ones.

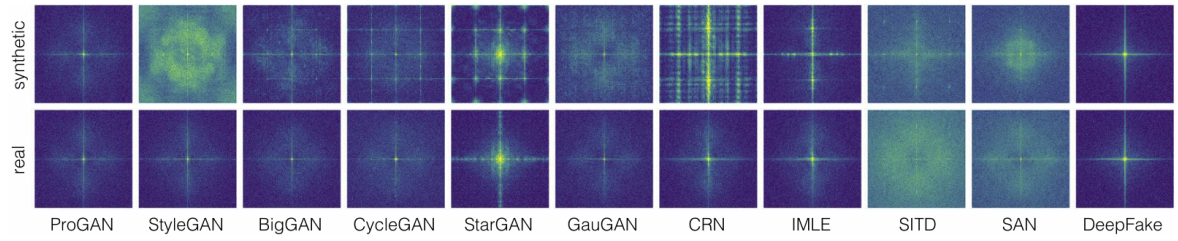


Figure 8: CNNSpot Frequency Visualization

CNNSpot visualize the average frequency spectra from each dataset to study the artifacts generated by CNNs, as shown in Figure 8. They perform a simple form of high-pass filtering (subtracting the image from its median blurred version) before calculating the Fourier transform, as it provides a more informative visualization. It must be noted that there are many interesting patterns visible in these visualizations. While the real image spectra generally look alike (with minor variations due to differences in the datasets), there are distinct patterns visible in images generated by different CNN models. Furthermore, the repeated period patterns in these spectra may be consistent with aliasing artifacts. The most effective unconditional GANs (BigGAN, ProGAN) contain relatively few such artifacts. Also, DeepFake images does not contain obvious artifacts.

In Figure 9, GenImage visualize the average spectral spectra of real images and fake images from different generators. Following CNNSpot, they use the discrete Fourier transform to investigate artifacts

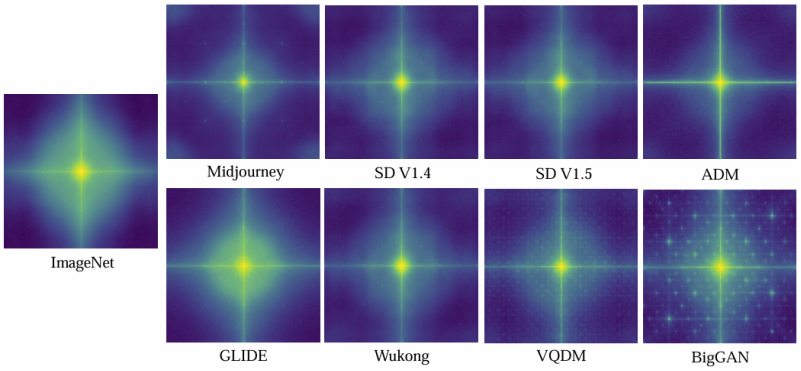


Figure 9: GenImage Frequency Visualization

generated by the generative model. The noise residuals R can be computed by the formula R = X - F(X), where X is the input image, and F() is the denoising filter DCNN. For each image source, they average the noise residuals of 1000 randomly selected images and use the Fourier transform of the results for spectral analysis. The results show that for GAN, artifacts are shown in the form of a regular grid. Real images from ImageNet contain few artifacts, as well as diffusion models. These results reflect that images from the diffusion models are closer to the real image than BigGAN, and therefore diffusion models present a greater challenge for detection. Richer et al. [28] observe that the diffusion models do not produce grid-like artifacts in the frequency spectrum, but exhibit a systematic mismatch towards higher frequencies. The hypothesis is that less weight is attached to the higher frequencies during training since matching lower frequencies is more important to the perceived quality of generated images.

Inspired of these results, we try to mine furthermore frequency information of images, especially on high-frequency part which is found more relative to difference between fake and real images and more prone to defects in generation.

Wavelet Transform is a powerful signal processing technique that decomposes a signal into components at multiple scales and locations using localized basis functions called wavelets. Unlike the traditional Fourier transform, which only provides frequency information, the wavelet transform captures both time (or spatial) and frequency characteristics, making it particularly useful for analyzing non-stationary signals such as images, audio, or video. In image processing, the 2D **Discrete Wavelet Transform (DWT)** is widely used for compression, denoising, and texture analysis. It recursively applies low-pass and high-pass filtering along both rows and columns of an image to decompose it into subbands that represent different frequency components and orientations (e.g., approximation, horizontal detail, vertical detail, and diagonal detail).

Let $I[m, n]$ be a 2D image, and let $h[n]$ and $g[n]$ be 1D low-pass and high-pass filters, respectively. The 2D discrete wavelet transform produces four subbands as follows:

$$\begin{aligned}
\mathrm{cA}[i,j] &= \sum_m \sum_n I[m,n] \cdot h[2i-m] \cdot h[2j-n] \\
\mathrm{cH}[i,j] &= \sum_m \sum_n I[m,n] \cdot h[2i-m] \cdot g[2j-n] \\
\mathrm{cV}[i,j] &= \sum_m \sum_n I[m,n] \cdot g[2i-m] \cdot h[2j-n] \\
\mathrm{cD}[i,j] &= \sum_m \sum_n I[m,n] \cdot g[2i-m] \cdot g[2j-n]
\end{aligned} \tag{8}$$

As showed in Figure 10, we compute 2D DWT on small dataset. cA is the approximate coefficient which represents the lowest frequency information. cH,cV,cD represent respectively the horizon-

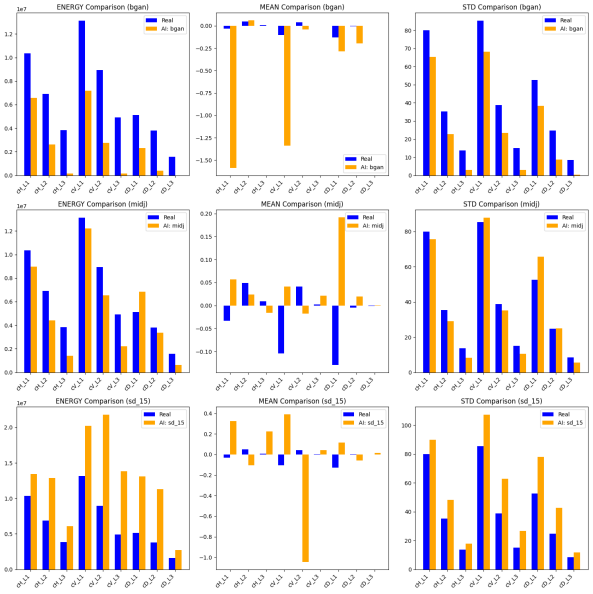


Figure 10: DWT result of class 489

tal,vertical and diagonal coefficients of higher frequency information which we care about. We calculate the mean, std and energy(sum of square) of cH,cV,cD on different levels of DWT as metrics. There seem to be some regularities, for example, std and energy show an obvious magnitude relationship at the first level, but different relationships appear in the three generative models we compared alone. We find that the trends of different coefficients are consistent in one class-generator pair but not cross pairs, which proves the reasonability of detection per class and that we can't directly use DWT information to make general (inter-generator) detector.

Wavelet Scattering Transform (WST) is a hierarchical, translation-invariant signal representation that combines the strengths of the wavelet transform and deep convolutional networks, but with mathematically guaranteed stability and interpretability. Proposed by Stéphane Mallat [29], WST extracts multiscale, deformation-stable features from signals, particularly useful in image and audio processing. It builds feature representations by cascading wavelet transforms with nonlinear modulus operators and averaging, capturing localized energy across multiple scales and orientations. Unlike traditional learned CNN filters, WST uses fixed wavelet filters (e.g., Morlet or Gabor), ensuring stability to small deformations and robustness to translation. The result is a feature extractor that resembles a deep network but is non-learned, mathematically grounded, and interpretable.

Let $x(u) \in L^2(\mathbb{R}^2)$ be a 2D input image. The zeroth-order scattering coefficient is the low-pass average:

$$S_0 x(u) = x * \phi_J(u) \tag{9}$$

The first-order scattering coefficients are:

$$S_1 x(\lambda_1, u) = |x * \psi_{\lambda_1}| * \phi_J(u) \tag{10}$$

The second-order scattering coefficients are:

$$S_2 x(\lambda_1, \lambda_2, u) = ||x * \psi_{\lambda_1}| * \psi_{\lambda_2}| * \phi_J(u) \tag{11}$$

Here, $*$ denotes convolution, $|\cdot|$ is the complex modulus, and $\lambda = (j, \theta)$ indexes scale and orientation of the wavelets ψ. Higher-order coefficients can be defined recursively in the same fashion. The complete wavelet scattering representation is the collection of all $S_m x$ for orders m = 0, 1, 2, ..., typically truncated at order 2 for practical applications.

4.4 Comparison of Representations

We have three representations as proposals: clip embeddings, dinov2 embeddings and wst (Wavelet Scattering Transform) vectors. Clip preprocesses images to be 224*224, dinov2 to be 224*224 and we preprocess images to be 256*256 for WST.

The former two embeddings have dimensions 512 and 768. For WST, we choose J(Scales' levels)=3, L(Default setting for orientation numbers)=8, the result coefficients are of dimension 217*32*32, then we pool and flatten the results to be of 217*2*2 = 868 dimension. In fact, J influences the level of frequency information got after the wst, higher J means deeper recursion of wavelet transform and higher level of frequency information structure extracted.

Then we calculate the three distances on these three representations, two kernels are chosen to test MMD: RBF(Gaussian) kernel and Laplacian kernel, the Gaussian assumption is retained to compute FD. The results showed here are relative mean differences between baseline(nature with nature2) and three generators' distances(three generator-nature pairs) for comparison: FD and W2D use $((mean(generators)/3) - baseline)/baseline$, MMD use $mean(generators)/3$ directly because we observe that ground truth distance can be approximately seen as 0 (lower than 10^{-4}).

It must be stated that we do p-test to verify our results. The p-value measures the probability of observing a test statistic as extreme as the one obtained, assuming the null hypothesis is true. We get p-value lower than 0.001 for all distance computations, which means that our results are strong.

Embedder	FD	MMD-Laplacian	MMD-RBF	W2D
Clip	1.591	0.088	0.128	0.152
Dino	1.450	0.081	0.120	0.264
WST-3	8.286	0.092	0.055	0.403

Table 1: Comparison of different embedders across various distribution distances.

From Table 1, we can see that RBF kernel shows better performance in representing data difference. Fake images have distribution deviation from real images on all representations, which means that methods based on these representations are feasible and reasonable.

5 Exploration for Solutions

After analysis of statistical distances on different subsets of class, generator and representations, we would like to further study how to apply existing architectures or algorithms in the field of applied mathematics to the obtained representations by pretrained large models in order to achieve a feasible and acceptable detection method. We find that the field of image anomaly detection is a good source of inspiration. In the advanced researches, several ideas are considered as useful for deepfake detection.

Conventionally, clip model outputs the cls token vector of a ViT passed at last through a FC layer; dinov2 model outputs the average pooled vector of the last hidden layer of a ViT, nased as "mean". Besides, we use wst3 to present Wavelet Scattering Transform with J = 3 and use an adaptive-average-pool layer to flatten its output, to get shape N*D vectors consistent with other methods.

5.1 Random Projection Outlyingness

When working with high-dimensional data, dimension reduction can translate into finding the right projections for the data, finding good projections can be costly and therefore leads to considering the possible contribution of random projections (RPs). RPs have been commonly used to conduct anomaly detection with the projected data representations. Using a great number of random projections, a stochastic approximation of the depth function can be obtained. A projection depth function associates a depth attribute to each data point available in a dataset, without explicitly estimating the underlying

probability density function. Such depth directly translates into an ordering of the data points, from the most normal to the most outlying one.

A random projection can be used to bring the data points with a d-dimensional representation on a single dimension representation space. Multiple random projections can thus be used to obtain multiple representations of data points on a single dimension, allowing the computation of normalized distances to the dataset center point for the dataset samples on each projection. These normalized distances lead to the Random Projection Outlyingness (RPO) [30].

The outlyingness score of a point $x \in \mathbb{R}^d$ with respect to a dataset $X \in \mathbb{R}^{d \times n}$ is defined by:

$$O(x; p, X) = \max_{u \in \mathbb{U}} \frac{\left|u^T x - \text{MED}(u^T X)\right|}{\text{MAD}(u^T X)} \tag{12}$$

where $u \in \mathbb{R}^d$ is a projection vector sampled from $\mathbb{U} \subset \mathbb{S}^{d-1}$ a set of p random unit-norm projection directions in the unit sphere $\mathbb{S}^{d-1} = \{x \in \mathbb{R}^d : \|x\|_2 = 1\}$, $\text{MED}(u^T X)$ is the median of projected data, $\text{MAD}(u^T X)$ is the median absolute deviation of the projected data.

The fundamental argument behind the interest for random projections to work on high-dimensional data is the Johnson-Lindenstrauss lemma [31], which guarantees the relative stability of the distance separating two data points between the input data space and the projected latent representations. In fact, as we have only 10^2 data points, dimension reduction can't be realized under strict JL lemma calculation, so we only make naive implementation of some related projections. After researching on commonly used projectors, we choose two most cited projector in other studies: Normal Random Projector (GaussianProjector) and Sparse Random Projector (SparseProjector). GaussianProjector uses a dense projection matrix whose entries are drawn i.i.d. from a standard normal distribution N(0,1); SparseProjector uses a matrix with mostly zero entries and a small number of non-zero values (e.g. -1,0,+1) sampled from a sparse distribution.

5.2 Mahalanobis Distance - PaDiM

Patch Distribution Modeling (PaDiM) [32] makes use of a pretrained convolutional neural network (CNN) for embedding extraction and each patch position is described by a multivariate Gaussian distribution, PaDiM takes into account the correlations between different semantic levels of a pretrained CNN.

In details, for all real images and one test image, it patchifies the activation outputs of different layers of CNN and merges them. It performs statistical calculations on all real images to obtain the mean vector and covariance matrix of each patch, and finally calculates the Mahalanobis distance between the test image and the real image set based on these two coefficients. In this way, an anomaly map can be obtained. The patch with the largest distance (score) corresponds to the most problematic area, and this score is also regarded as the score of the entire image.

Let $x \in \mathbb{R}^d$, and let $\mu \in \mathbb{R}^d$ be the mean vector and $\Sigma \in \mathbb{R}^{d \times d}$ be a positive-definite covariance matrix of a multivariate Gaussian distribution. The Mahalanobis distance is defined as:

$$D_M(x) = \sqrt{(x - \mu)^\top \Sigma^{-1} (x - \mu)} \tag{13}$$

This measures how many standard deviations the point x is from the mean μ under the covariance structure defined by Σ.

5.3 Memory Coreset - PatchCore

Similarly to PaDiM, usage of mid-level network patch features allows PatchCore [33] to operate with minimal bias towards ImageNet classes on a high resolution, while a feature aggregation over a local neighbourhood ensures retention of sufficient spatial context.This results in an extensive memory bank allowing PatchCore to optimally leverage available nominal context at test time. Finally, for practical applicability, PatchCore additionally introduces greedy coreset subsampling for nominal feature banks

as a key element to both reduce redundancy in the extracted, patch-level memory bank as well as significantly bringing down storage memory and inference time.

In details, to avoid losing more localized nominal information or being too localized, PatchCore makes use of mid-level layers' output of CNN and do a local neighbourhood aggregation to increase receptive field size and robustness. Then memory bank unions all these patch features from nominal image set and do a a minimax facility location coreset selection to get a representative coreset.

To construct a representative coreset $\mathcal{M}_C \subset \mathcal{M}$ from the original memory bank $\mathcal{M}$, we solve the following minimax facility location problem:

$$\mathcal{M}_C^* = \arg \min_{\mathcal{M}_C \subset \mathcal{M}} \max_{m \in \mathcal{M}} \min_{n \in \mathcal{M}_C} \|m - n\|_2 \tag{14}$$

To further reduce coreset selection time, in this paper they mention also the Johnson-Lindenstrauss theorem and random linear projections to randomly reduce data dimension.

At last, with the coreset of memory bank, a weighted score of test image is calculated by the maximum distance (score) between test patch features in its patch collection to each respective nearest neighbour in coreset. For a test image x^{test}, with its corresponding patch feature collection $\mathcal{P}(x^{\text{test}})$, we define:

For a test image x^{test}, with its corresponding patch feature collection $\mathcal{P}(x^{\text{test}})$, we define:

$$m^{\text{test},*},\, m^* = \arg \max_{m^{\text{test}} \in \mathcal{P}(x^{\text{test}})} \arg \min_{m \in \mathcal{M}} \|m^{\text{test}} - m\|_2 \tag{15}$$

$$s^* = \|m^{\text{test},*} - m^*\|_2 \tag{16}$$

where: $\mathcal{P}(x^{\text{test}})$: Set of patch-level features extracted from x^{test}; $m^{\text{test},*}$: The patch feature from $\mathcal{P}(x^{\text{test}})$ with the highest distance to its nearest neighbor in $\mathcal{M}$; m^*: Nearest neighbor of$m^{\text{test},*}$ in the memory bank $\mathcal{M}$.

To account for the neighborhood structure of m^*, we compute the final reweighted anomaly score s as:

$$s = \left(1 - \frac{\exp\left(\|m^{\text{test},*} - m^*\|_2\right)}{\sum_{m \in \mathcal{N}_b(m^*)} \exp\left(\|m^{\text{test},*} - m\|_2\right)}\right) \cdot s^* \tag{17}$$

where: $\mathcal{N}_b(m^*)$: The set of b nearest neighbors of m^* in $\mathcal{M}$; The exponential weighting increases s when m^* lies far from its own neighbors, indicating it is itself a rare occurrence.

5.4 Implementation, Ablation Experiments and Conclusion

We experiment our methods on small extracted dataset mentioned before. As all our designs of methods are threshold-judgement indicators, we use AUROC (higher better), AUPRC (higher better) and FPR-95 (lower better) as metrics for evaluation. It must be noted that here we get FPR-95 as the FPR value of threshold[idx] where threshold[idx] is the first threshold to let TPR>= 0.95. This measurement is by index, not by interpolation. It means that the FPR-95 we get will be biased upwards, estimating the value in a conservative way.

	AUROC	AUPRC	FPR-95
wst3-RPO Sparse	0.457	0.515	0.926
wst3-RPO Gaussian	0.480	0.510	0.941
dinov2-RPO Sparse-mean	**0.625**	**0.658**	**0.852**
dinov2-RPO Sparse-cls token	**0.620**	**0.606**	**0.705**
dinov2-RPO Gaussian-mean	0.623	0.652	0.860
clip-RPO Sparse	0.518	0.568	0.937
clip-RPO Gaussian	0.513	0.558	0.922

Figure 11: RPO Experiments

Inspired by [30], we implement RPO with different projectors to run test on different representations. As in Figure 11, we find that sparse projector is a bit better than gaussian projector, with dinov2 direct output embeddings the best representation in RPO trials. As a result, in the following implementations, we set sparse projector as the random projection matrix.

Inspired by [32], we also try to use statistical Mahalanobis distance and idea of patchification for our solution. As first trial, omitting patch settings and random dimension reduction, we calculate Gaussian statistical (mean vector and covariance matrix) of ground truth images on two models' direct output and wst3 output, then use Mahalanobis distance as score for a test image. Then we adjust models' output to recover the original PaDiM method: we extract all patch tokens except cls token from the last patchified output of ViT for clip and dinov2, then do the random dimension reduction by SparseProjector. And we calculate the Mahalanobis distance for every patch to get the max score of a test image.

Inspired by [33], considering that the ViT structure already encodes global relationship between patch tokens and that it doesn't consist of convolutional sturcture, we can skip feature aggregation over local neighbourhood. At first trial we implement their greedy coreset selection for direct output of clip, dinov2 and wst3, calculate the score in the same way as in paper on one-vector outputs' coreset memory bank. Then we adjust models' output to realize original PatchCore method: we extract all patch tokens except cls token, do random dimension reduction by SparseProjector for greedily extract the memory coreset, then calculate the score in the original dimension. Instead of using the same score weighting way as in original algorithm, we use softmax for there are numeric instability cases when we experiment.

Reference to studies in [27] and [24], mentioned in part 4.3, the difference is more notable in high frequency domain than in lower domain, so we try also wst with J=2 (less scale levels, less redundant information extrated), we use wst2's direct output with same flatten strategy as wst3 (only differs in scales of pooling, because J=2 has different numbers of orientation and spatial dimension). We implement mahalanobis-distance algorithm on wst2 output, find that there's no improvement.

We also try to combine frequency information with large model's encoding ability. As J=2 means smaller dimension of wst's output which makes dimension reduction easier and less information loss, we try to encode images' information by wst2 then feed them in clip model to get a latent representation of the frequency coefficients, or "frequency features". As wst2's output doesn't meet the requirements of clip's input, after wst2, we do PCA on ground truth set output and use the same fixed vectors to do PCA for test set output, and we reshape them into 3*256*256 where we can introduce the preprocessor of clip model. The experiment's result shows that this combined method is better than other clip methods, but unable to surpass the minimum demand of 0.65 AUROC nor the performance of dinov2 families.

As we talked in the beginning of section 5, we take mean vector of dinov2's output (all tokens) as direct output, but how about only the cls token? We implement also mahalanobis-distance algorithm with it and find an acceptable improvement.

Results of all experiments above are showed in Figure 12. Considering also with RPO results, we can

	AUROC	AUPRC	FPR-95
wst3-Memory coreset	0. 525	0. 547	
wst3-Mahalanobis	0. 471	0. 506	
wst2-Mahalanobis	0. 421	0. 476	0. 960
wst2 & clip-Mahalanobis	0. 547	0. 568	0. 916
dinov2-PatchCore	0. 593	0. 642	
dinov2-PaDim	0. 507	0. 555	0. 934
dinov2-Memory Coreset-mean	0. 310	0. 434	
dinov2-Mahalanobis-mean	0. 676	0. 725	
dinov2-Mahalanobis-cls token	**0. 687**	**0. 746**	**0. 795**
clip-PatchCore	0. 389	0. 461	
clip-PaDim	0. 375	0. 437	0. 984
clip-Memory coreset	0. 467	0. 529	
clip-Mahalanobis	0. 514	0. 586	

Figure 12: Experiments

say that dinov2 presents a better performance than other representations. In fact, dinov2's patch size is 16, which is smaller than clip's 32. For the same input size of 224, it will be divided into 196 patches, far more than clip's 49. This allows dinov2 to learn and model images more finely and pay more attention to local details. Reasons can also be found in [34], where authors visualized the heatmap (lefter part of Figure 13) of different vision models by GradCAM on some images, the results demonstrate that supervised models (ResNet 50) focus primarily on the main objects directly relevant to the classification result. In contrast, self-supervised models, particularly DINOv2, exhibit a more holistic perspective, capturing a broader understanding of the image content. Furthermore, they investigate the sensitivity of real and fake images to small perturbations, with a plot of the cosine similarity landscape shown in righter part of Figure 13. Their findings reveal that, compared to real images, AI-generated images exhibit higher sensitivity to small perturbations when using models like DINOv2, which adopts a more global view. This phenomenon is not so obvious in ResNet 50 and CLIP perhaps because DINOv2 uses self-supervised learning on images only, while ResNet 50 uses image labels for supervised learning and CLIP uses image captions for weakly supervised learning.

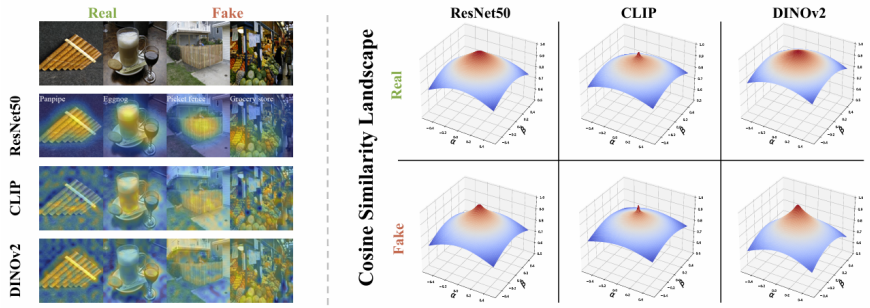


Figure 13: Visualization

In conclusion, we state that dinov2 model with cls token as outpput shows a better performance in most methods, wst combined with large model shows good potential. In all methods, mahalanobis-

distance algorithm and RPO Sparse are the best, when PatchCore, Memory Coreset, Padim have almost the same level of performance.

Then we are going to compare best solutions of our exploration, dinov2-Mahalanobis-cls and dinov2-RPO-Sparse-cls, on the whole GenImage dataset with State-of-the-art (SOTA) Methods.

6 State-of-the-art Methods

In our research, paper of GenImage (2023) [24], RIGID (2024) [34] and DRCT [35] cite the SOTA Detection Methods proposed in recent years (majority after 2020). We present them here, the corresponding reference links can be found in [24], [34] and [35].

Method	Testing Subset								Avg
	Midjourney	SD V1.4	SD V1.5	ADM	GLIDE	Wukong	VQDM	BigGAN	Acc.(%)
ResNet-50 [9]	59.0	72.3	72.4	59.7	73.1	71.4	60.9	66.6	66.9
DeiT-S [28]	60.7	74.2	74.2	59.5	71.1	73.1	61.7	66.3	67.6
Swin-T [17]	**61.7**	**76.0**	**76.1**	61.3	**76.9**	**75.1**	**65.8**	**69.5**	**70.3**
CNNSpot [31]	58.2	70.3	70.2	57.0	57.1	67.7	56.7	56.6	61.7
Spec [38]	56.7	72.4	72.3	57.9	65.4	70.3	61.7	64.3	65.1
F3Net [23]	55.1	73.1	73.1	**66.5**	57.8	72.3	62.1	56.5	64.6
GramNet [18]	58.1	72.8	72.7	58.7	65.3	71.3	57.8	61.2	64.7

Figure 14: GenImage Benchmark: Eight models trained on eight generators are tested on one generator, and their average accuracy is each data point in the testing subset column.

Methods introduced in GenImage Benchmark Among traditional approaches, image classifiers remain the most straightforward detectors, utilizing architectures such as CNN-based ResNet50, Transformer-based DeiT-S, and Swin-T for binary image classification. These models provide solid baseline methods for image forgery detection tasks.

In the early stages, research focused primarily on detectors that leveraged facial features. For example, F3Net explores both the partitioning of frequency components and the discrepancies in frequency statistics between real and fake images to detect facial forgeries. GramNet relies on global texture features to enhance robustness and generalization in fake face detection. While these models are less effective when applied beyond the facial domain, they offer valuable insights that can inspire future design choices.

Spec takes frequency spectrum representations as input and introduces GAN-like artifacts into real images without relying on specific GAN-generated data for training. CNNSpot, based on ResNet-50, functions as a binary classifier enhanced with tailored pre- and post-processing and targeted data augmentation strategies. However, existing methods still struggle to perform well on datasets containing a mix of GAN- and diffusion-generated images, highlighting the need for further advancements.

Methods introduced in RIGID paper Among learning-based methods, Wang et al. showed that even a simple classifier trained on ProGAN-generated images, when combined with Gaussian blur and JPEG compression as augmentations, could generalize to previously unseen GAN-generated images. Gragnaniello et al. enhanced detection performance further by applying a broader range of data augmentations. Corvi et al. later extended Wang's method to handle diffusion-generated images. However, these training-based approaches often face challenges in generalization and demand high computational resources for training, prompting increased interest in training-free detection techniques.

AEROBLADE identifies generated images by measuring reconstruction errors from an autoencoder. RIGID, on the other hand, leverages the varying sensitivity of real and fake images to minor perturbations. Specifically, it's observed that noise perturbations lead to gradual feature changes in real images,

AUC/AP (%)	Training Samples	Diffusion				GAN			VAE		Average
		ADM	ADMG	LDM	DiT	BigGAN	GigaGAN	StyleGAN XL	RQ-Transformer	Mask GIT	
Wang	720 000	*65.96/66.75*	*65.56/66.59*	*67.82/69.43*	61.97/64.25	*83.15/84.76*	*71.19/69.96*	*66.63/66.06*	60.66/61.67	*65.43/66.97*	*67.60/68.43*
Gragnaniello	400 000	60.21/59.91	59.45/59.71	61.61/61.37	56.67/56.56	59.62/58.49	53.63/52.35	51.58/52.35	56.49/54.34	53.70/52.68	56.99/56.24
Corvi	400 000	63.94/63.85	65.55/65.19	62.18/60.83	56.64/55.23	61.91/59.95	50.15/49.18	48.48/48.05	63.21/60.48	61.19/59.51	59.25/58.03
DIRE	80 000	57.79/56.67	57.09/56.80	61.47/62.15	53.21/53.52	49.63/50.00	50.00/51.14	52.91/53.87	53.17/52.41	49.93/51.57	53.91/54.24
AEROBLADE	Training Free	52.20/53.65	59.24/57.93	62.97/61.96	**72.98/73.65**	50.07/50.94	55.21/54.87	51.17/52.85	*70.23/69.36*	59.80/58.71	59.32/59.33
RIGID	Training Free	**87.75/86.06**	**83.50/81.46**	**81.50/80.23**	*72.07/69.55*	**93.86/93.57**	**89.29/87.92**	**85.94/84.75**	**93.39/93.11**	**92.65/91.91**	**86.67/85.40**

Figure 15: RIGID Evaluation: The AUC and AP of different AI-generated image detectors on IMAGENET. A higher value indicates better performance. The **bolded** values are the best performance, and the *underlined italicized* values are the second-best performance.

resulting in smoother gradients, while generated images respond more sharply, producing steeper gradients. Although the perturbations are subtle, they act as effective probes for both texture-rich and texture-poor regions, making them particularly useful for detecting generated content.

Methods introduced in DRCT paper The newer CNN-based model family ConvNeXt (Conv-B) can be a good image classifier with its advanced performance in traditional visual domain. For the detection of diffusion-based generated images, researchers utilized a multimodal fusion technique with CLIP as the backbone network, and found robustness in using BLIP-generated captions as input to a text model. Someone also employed a large pre-trained CLIP model as a feature extractor with a nearest neighbor classifier, achieving promising generalization (UnivFD). Diffusion Reconstruction Error (DIRE) method detects diffusion-based generated images by leveraging the insight that real images cannot be accurately reconstructed by diffusion models.

Inspired by the idea of contrastive training, Diffusion Reconstruction Contrastive Training(DRCT) enhances the generalizability of detectors by training them with hard samples under appropriate guidance. The core idea is that if a classifier can distinguish hard-to-detect generated images from real images, it is also likely to generalize well in identifying easier samples. DRCT framework creates hard samples by reconstructing real images, which can produce high-quality near-real images that have almost the same appearance as the real images but contain subtle and imperceptible traces left by the generation model. Training existing detection methods with these hard samples is expected to improve their generalizability, enabling them to effectively detect the traces left by those image generation models not covered by the training set.

Method	Midjourney	SDv1.4	SDv1.5	ADM	GLIDE	Wukong	VQDM	BigGAN	Avg.
CNNSpot	84.92	99.88	99.76	53.48	53.80	99.68	55.50	49.93	74.62
F3Net	77.85	98.99	99.08	51.20	54.87	97.92	58.99	49.21	73.51
CLIP/RN50	83.30	99.97	99.89	54.55	57.37	99.52	57.90	50.00	75.31
GramNet	73.68	98.85	98.79	51.52	55.38	95.38	55.15	49.41	72.27
De-fake	79.88	98.65	98.62	71.57	78.05	98.42	78.31	74.37	84.73
Conv-B	83.55	**99.99**	**99.92**	51.75	56.27	99.92	58.41	50.00	74.98
UnivFD	91.46	96.41	96.14	58.07	73.40	94.53	67.83	57.72	79.45
DIRE	50.40	**99.99**	**99.92**	52.32	67.23	**99.98**	50.10	49.99	71.24
DRCT/Conv-B (ours)	**94.63**	99.88	99.82	61.78	65.92	99.91	74.88	58.81	82.08
DRCT/UnivFD (ours)	91.50	95.01	94.41	**79.42**	**89.18**	94.67	**90.03**	**81.67**	**89.49**

Figure 16: DRCT Evaluation: Accuracy (ACC,%) comparisons of DRCT and other generated image detectors. All methods were trained on GenImage/SDv1.4 and evaluated on different testing subsets. The Diffusion Reconstruction Model of DIRE and DRCT is SDv1.

7 Experiments

Considering hardware condition and code release of various SOTA methods, we implement CNNSpot, DIRE, AEROBLADE these SOTA methods on the whole GenImage validation subset. We implement our two methods in whole GenImage validation fake subset and an artificial GenImage validation real subset.

7.1 Settings

There are 8 generators: BigGAN, GLIDE, VQDM, Stable Diffusion V1.4, Stable Diffusion V1.5, ADM, Midjourney, and Wukong, we name them [bgan, glide, vqdm, sd-14, sd-15, adm, midj, wukong].

GenImage validation set comprises 8 generators subset, each divided into real and fake subset. There are 6 images generated for every class in 1000 classes in all generators' fake subset, except sd-15 8 images. So in total 50 images per class.

For ground truth real images set of our methods, we extract 240 real images per class. For the reason that real images in GenImage validation subset are not divided in classes, we use random extraction of GenImage train subset to make a comparable real image part of validation set, with same total numbers and same proportion of every generator: 6 images for every generator exept 8 images for sd-15.

For example, if we test our dinov2-RPO-Sparse-cls method for class 000 in adm generator subset, we compute ground truth data points on 240 images extracted, and then score the mixed test data of 6 real images and 6 fake images.

7.2 Implementation

We implement CNNSpot with its official checkpoint release: CNNSpot-blur-jpg-prob0.1; ResNet classifier released by DIRE with checkpoint: DIRE-imagenet-adm; AEROBLADE with official released pipeline: three LDM Autoencoders are CompVis/stable-diffusion-v1-1, stabilityai/stable-diffusion-2-base and kandinsky-community/kandinsky-2-1, with lpips-vgg-2 as distance metric. CNNSpot has predicted probability of being fake images in its final output; ResNet has also probabilities as output; AEROBLADE has maximum negative distances between the original image and the reconstructed image generated by the three LDM AutoEnco as distance metricders as output. We have true labels and these probabilities or distances as scores, then get evaluation metrics.

Our first method uses cls token from last hidden layer of dinov2 model's output, calculates Mahalanobis distances between test images and ground truth statisticals as scores, and get evaluation metrics; second method use the same input but with RPO algorithm, projector is chosen as SparseProjector. It must be noted here that we use now facebook/dinov2-with-registers-base instead of dinov2-base, because it is reported as an advanced version of original dinov2-base.

There is an important point that AEROBLADE costs more time when running on the same validation set, because of reconstructing-then-computing mechanism. AEROBLADE need almost 150x time than direct classifiers, while our methods cost almost the same time as direct predictors. However, this observation is not under a very strict setting (capability of our hardware can't support same batch size for every method), still AEROBLADE shows an obvious time-cost property.

7.3 Comparison and Analysis

	GENERATOR (AUROC % / AUPRC % / FPR95 %)								
METHOD	ADM	BigGAN	GLIDE	Midjourney	SDv1.4	SDv1.5	VQDM	Wukong	Average
ResNet	53.1 / 52.0 / 93.7	68.5 / 63.1 / 81.9	54.2 / 53.3 / 96.6	51.3 / 50.5 / 97.0	47.5 / 47.4 / 95.9	47.7 / 47.3 / 95.6	59.6 / 61.7 / 96.1	37.2 / 40.9 / 99.1	52.4 / 52.0 / 94.5
CNNSpot	**73.6** / **71.2** / 72.1	**92.5** / **90.0** / **24.5**	70.6 / 66.3 / 71.5	56.4 / 56.0 / 92.6	60.6 / 56.9 / 86.6	60.8 / 57.3 / 86.0	65.8 / 62.1 / 81.4	55.4 / 52.9 / 91.3	67.0 / 64.1 / 75.7
AEROBLADE	64.0 / 67.5 / 94.3	80.3 / 76.6 / 59.2	**91.7** / **92.2** / 44.6	**80.9** / **81.1** / 69.0	56.0 / 54.9 / 90.0	56.3 / 55.0 / 89.7	49.1 / 52.0 / 98.3	62.4 / 60.7 / 87.1	67.6 / 67.5 / 79.0
Mahalanobis(ours)	69.6 / 68.1 / **58.2**	86.6 / 84.6 / 28.5	81.2 / 79.2 / **33.1**	62.8 / 63.2 / **60.2**	**63.7** / **64.4** / **54.6**	**63.5** / **63.6** / **56.6**	**88.4** / **86.8** / **27.6**	**75.2** / **74.6** / **40.2**	**73.9** / **73.1** / **44.9**
RPO(ours)	59.8 / 60.4 / 72.7	76.4 / 75.7 / 47.3	74.9 / 74.1 / 43.1	59.3 / 61.3 / 64.9	59.7 / 62.3 / 61.3	59.0 / 61.2 / 65.1	80.8 / 80.0 / 42.9	72.0 / 72.5 / 46.5	67.7 / 68.4 / 55.5

Figure 17: Final Comparison: The **bolded** values are the best performance, and the underlined values are the second-best performance.

The results are shown in figure 17, we can see that our methods get the best and the second best average performance cross different generators' subset. In details, CNNSpot shows best performance on ADM and BigGAN val subset, while AEROBLADE performs best on GLIDE and Midjourney val subset. CNNSpot can't distinguish images of advanced generative models such as SDv1.4 or SDv1.5, AEROBLADE present an obvious bias for several generators. That's because CNNSpot uses large pretrained CNN model, which is corresponding to GAN's netwok structure and has a very SOTA encoding ability, but it struggles to detect images generated by advanced diffusion-based generative models; AEROBLADE, as a reconstruction-error-based method, its performance heavily depends on which LDM's AutoEncoder it uses, so when AEROBLADE's AE meets the generator's, the result will be significantly improved.

In these released methods, training-based classifiers are trained on smaller dataset than GenImage or single generator's dataset, training-free method uses AutoEncoders from diffusion family. Our methods are already competitive. In fact, looking in details, our methods perform relatively steadily in the detection of various generators (stable in the top three), demonstrating strong generalization capabilities across generators, and also proving some advantages of anomaly detection-based detection with reference.

8 Conclusion and Future Work

In this project, we conducted a lot of investigations and comparative studies on statistics, distance metrics, and frequency analysis methods for deepfake detection, and learned from anomaly detection and made a lot of fusion and innovation methods. Among them, the more excellent methods have been verified to have comparable capabilities with mainstream detectors in recent years. However, our project still lacks a lot of ablation studies (e.g. with respect to ground truth reference set's size) to strictly study the importance of each component, strictly prove some of our speculations and strictly screen out the best solutions in our experiments. Besides, we need also to have some dimension reduction methods(like fixed-PCA but should be more convenient) with solid theoretical support, now projectors of JL lemma family are not suitable for our situation, because there are too few data points.

We are still working on it: the improvement of previous work mainly focuses on ablation studies and to implement more SOTA methods to be compared such as DRCT; in terms of subsequent expansion, we will focus on the intervention of better large models (such as SigLIP), and the use of language part of multimodal models for auxiliary recognition (calculating cosine-similarity) and other methods.

References

[1] 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 (CVPR), pages 10684–10695, June 2022.

[2] Midjourney. https://www.midjourney.com/home/. 2022.

[3] Zeyu Lu, Di Huang, Lei Bai, Xihui Liu, Jingjing Qu, and Wanli Ouyang. Seeing is not always believing: A quantitative study on human perception of ai-generated images. arXiv preprint arXiv:2304.13023, 2023.

[4] Hao Dang, Feng Liu, Joel Stehouwer, Xiaoming Liu, and Anil K Jain. On the detection of digital face manipulation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition, pages 5781–5790, 2020.

[5] Keshigeyan Chandrasegaran, Ngoc-Trung Tran, and Ngai-Man Cheung. A closer look at fourier spectrum discrepancies for cnn-generated images detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 7200–7209, 2021.

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

[7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. Advances in neural information processing systems, 33:6840–6851, 2020.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszko reit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In NeurIPS, 2017.

[9] Alec Radford, Jong Wook Kim, Chris Hallacy et al., Learning Transferable Visual Models From Natural Language Supervision. arXiv preprint arXiv:2103.00020 (2021).

[10] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096, 2018.

[11] 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.

[12] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. Advances in Neural Information Processing Systems, 34:8780–8794, 2021.

[13] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741, 2021.

[14] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10696–10706, 2022.

[15] Peebles, W. and Xie, S. Scalable diffusion models with transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4195–4205, 2023a.

[16] Wukong. https://xihe.mindspore.cn/modelzoo/wukong. 2022.

[17] Riccardo Corvi, Davide Cozzolino, Giada Zingarini, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. On the detection of synthetic images generated by diffusion models. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5. IEEE, 2023.

[18] Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, Hezhen Hu, Hong Chen, and Houqiang Li. Dire for diffusion-generated image detection. arXiv preprint arXiv:2303.09295, 2023.

[19] Baoying Chen, Jishen Zeng, Jianquan Yang, Rui Yang. DRCT: Diffusion Reconstruction Contrastive Training towards Universal Detection of Diffusion Generated Images. Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

[20] Aref Azizpour, Tai D. Nguyen, Manil Shrestha, Kaidi Xu, Edward Kim, Matthew C. Stamm. E3: Ensemble of Expert Embedders for Adapting Synthetic Image Detectors to NewGenerators Using Limited Data. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).

[21] Jonas Ricker, Denis Lukovnikov, and Asja Fischer. Aeroblade: Training-free detection of latent diffusion images using autoencoder reconstruction error. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2024.

[22] Tal Reiss, Bar Cavia, Yedid Hoshen. DETECTING DEEPFAKES WITHOUT SEEING ANY. ICLR 2024 Conference.

[23] Francisco Caetano, Christiaan Viviers, Luis A. Zavala-Mondragón et al. DisCoPatch: Batch Statistics Are All You Need For OOD Detection, But Only If You Can Trust Them. arXiv:2501.08005v1 (2025).

[24] Zhu, M., Chen, H., Yan, Q., Huang, X., Lin, G., Li, W., Tu, Z., Hu, H., Hu, J., and Wang, Y. Genimage: A million-scale benchmark for detecting ai-generated image. ArXiv, abs/2306.08571, 2023.

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

[26] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch et al. A Kernel Two-Sample Test. Journal of Machine Learning Research 13 (2012) 723-773.

[27] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 8695–8704, 2020.

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

[29] Joan Bruna and Stéphane Mallat (CMAP, Ecole Polytechnique, Palaiseau, France). Invariant Scattering Convolution Networks. arXiv:1203.1513 (2012).

[30] Martin Bauw, Santiago Velasco-Forero, Jesus Angulo, Claude Adnet, Olivier Airiau. Deep Random Projection Outlyingness for Unsupervised Anomaly Detection. ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning.

[31] William, B. J. and Lindenstrauss, J. Extensions of lipschitz mapping into hilbert space. Contemporary mathematics, 26(189-206):323–341, 1984.

[32] Thomas Defard, Aleksandr Setkov, Angelique Loesch, Romaric Audigier (Université Paris-Saclay). PaDiM: a Patch Distribution Modeling Framework for Anomaly Detection and Localization. arXiv:2011.08785v1 (2020).

[33] Karsten Roth, Latha Pemula, Joaquin Zepeda et al. Towards Total Recall in Industrial Anomaly Detection. arXiv:2106.08265v2 (2022).

[34] Zhiyuan He, Pin-Yu Chen, Tsung-Yi Ho. RIGID: A Training-Free and Model-Agnostic Framework for Robust AI-Generated Image Detection. arXiv:2405.20112v1 (2024).

[35] Baoying Chen, Jishen Zeng, Jianquan Yang, Rui Yang. DRCT: Diffusion Reconstruction Contrastive Training towards Universal Detection of Diffusion Generated Images. Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.