Electrical and Electronics Engineering
Semester Project
Master Semester 3 - Autumn 2023

Exploring Diffusion-Generated Image Detection Methods

Student: Aristotelis Dimitriou

Supervised by: Yuhang Lu
Prof. Dr. Touradj Ebrahimi

January 12, 2024

Multimedia Signal Processing Group
EPFL

Abstract

This study addresses the challenge of enhancing the generalizability of deepfake detection across various generative models, particularly focusing on GANs and diffusion models. Using a ResNet50 pre-trained on ImageNet, we investigate the effectiveness of diverse pre-processing techniques and frequency transformations in distinguishing fake images generated by these two types of generative models.

Our research explores high-pass, low-pass, and sharp-edge pre-processing methods, both individually and in combination with Fast Fourier Transform (FFT) and Discrete Cosine Transform (DCT). A key finding of our research is the remarkable effectiveness of FFT as a standalone transformation, which significantly enhances detection accuracy and average precision across diverse datasets. This outcome underscores FFT's robust capability in revealing subtle, frequency-based artifacts inherent in deepfake images.

The study also compares these results with established baseline models, offering a broader perspective on the state-of-the-art in deepfake detection. Through this comprehensive analysis, we contribute to the advancement of detection techniques, highlighting the critical role of tailored pre-processing and frequency transformation strategies in enhancing the model's ability to generalize across different generative models.

The code associated with this project can be found here; `https://github.com/aristo6253/diffusion-image-detection`

Contents

1 Project description

In recent years, face manipulation and generation techniques have achieved remarkable progress. The deep learning-based tools along with open-source software have simplified the creation of forgery, raising public concern about the potential abuse for malicious purposes. Computer vision and media security experts have devoted significant efforts to address the problem of face manipulation caused by deepfake techniques. But the problem of detecting purely synthesized face images has been studies in a lesser extent. In particular, the recent popular diffusion models have shown remarkable success in image synthesis. Existing detectors struggle to detect images generated by diffusion models.

The objective of this project is to first familiarize the student with the state-of-the-art deep learning-based synthesized image detection methods and databases. Then the student will conduct frequency analysis on the diffusion-generated face images and will further explore a detection method in this direction. The student is also encouraged to explore other methods from different perspectives. At the end, the student should evaluate proposed detection method on multiple testing datasets to show both the performance and generalization ability.

In general, the following tasks shall be performed:

- Literature review of the state-of-the-art deep learning-based synthesized image detection methods and summarize their key ideas
- Set up at least one state-of-the-art synthesized image detection method as a baseline
- Conduct frequency domain analysis on diffusion model-generated face images
- Explore new detection methods and set up a comprehensive train/validation/test pipeline
- Assess the performance of the proposed detector on multiple test sets generated by different types of diffusion models
- Document the code and results and write a report on the project.

Deliverables (final report, final presentation, electronic package of all documents and software, etc.) must be submitted according to EPFL and MMSPG guidelines and schedule.

2 Introduction

The advancement of deep learning, particularly the advancement of generative models, coupled with the accessibility to very large datasets, has led to the rise of highly realistic fake media, commonly known as deepfakes. These deepfakes, specifically in the real of facial imagery, have the potential to be used in a harmless entertaining way to a serious misinfomative way posing security threats. As a result, the development of reliable detection methods for such content has become a critical area of research.

Deepfakes include several main techniques for facial manipulation:

- **Entire Face Synthesis**, which involves creating new, realistic faces from scratch;
- **Attribute Manipulation**, which changes specific features of a face;
- **Identity Swap**, commonly known as face replacement;
- **Expression Swap**, which alters the facial expressions of an individual.

This project focuses specifically on Entire Face Synthesis for static images, exploring the challenges and solutions in detecting such deepfakes.

So far, the state-of-the-art in deepfake detection has primarily relied on Convolutional Neural Networks (CNNs). These methods have proven high effectiveness within the same generative model family. For instance, classifiers trained on images generated by ProGAN tend to successfully detect fakes produced by similar models like StyleGAN. However, the main challenge in this domain is the lack of generalizability across different families of generative models, such as between GANs and diffusion models.

The goal of this project is to bridge the gap in generalizability. One of the best performing state-of-the-art models (Wang et al. [11]) was investigated, designed as a "universal" detector to differentiate real images from those generated by any CNN, regardless of its architecture or the dataset used. However its effectiveness is greatly impacted when applied to diffusion models, highlighting an important area for improvement. Our primary objective is to enhance this model's ability to generalize across different types of generative models, especially between Generative Adversarial Networks (GANs) and diffusion models.

To achieve this, the impact of various pre-processing techniques , transformations, and data augmentations were explored. The combination of such methods could potentially improve the model's ability to generalize and accurately detect deepfakes. This study is important for digital image forensics and helps make digital media more reliable and trustworthy in a time when advanced artificial intelligence is becoming more potent for misuse.

3 Related Work

3.1 GAN Artifacts in Frequency Analysis

An insightful contributor to the field of deepfake detection is the research conducted by Zhang et al. [12] in 2019. This work delves into the challenges of classifier generalization across different GAN models. They recognized that while classifiers are effective within the same GAN family, their performance significantly drops when evaluated on images generated by different GAN architectures. To tackle this they introduces a new approach, AutoGAN, designed to generate images containing upsampling artifacts typical in GANs.

The aspect of Zhang et al.'s work that is particularly pertinent to this study is their discovery of specific artifacts in the frequency domain. They observed that images produced by GAN models showcase periodic, grid-like pattern in their frequency spectra, a feature rarely seen in real images. This discovery is very interesting as it suggests that the frequency domain could potentially be the key for generalizing detection methods across various generative models.

This insight into the frequency domain artifacts serves as a foundation for our research. It hits us that by focusing on these unique spectral signatures, we might develop more robust and universally applicable detection methods, capable of effectively identifying deepfakes regardless of the underlying generative technology.

3.2 Enhancing CNN Deepfake Detection with ResNet50

The research conducted by Wang et al. [11] in 2020 not only replicated the periodic grid-like patterns identified by Zhang et al., but also expanded upon this findings to make a more universal solution for deepfake detection.

The foundation of Wang et al.'s approach is the use of a pre-trained ResNet50 [3] model, original trained on ImageNet. A big emphasis is set on the proper pre- and post-processing techniques, along with appropriate data augmentation. Their methodology has shown impressive performance in detecting CNN-generated fake images, setting a new benchmark in the field.

The proposed approach has become a widely accepted baseline in recent research on deepfake detection. In this project, we will use as well this approach as a baseline, against which the enhancements/modifications we propose are going to be compared to.

3.3 Unbiased Classification of Deepfakes via CLIP

In an innovative study, conducted in 2023, Ojha et al. [8] look into the classification challenge in deepfake detection. They made an interesting observation that the 'real' class often acts as a 'sink' class, where the detection models missclassifies all fake images not originating from the training dataset as 'real'.

Ohja et al. reproduced the unique spectra found by Zhang et al. which show a clear difference between GAN generated images and diffusion and real images whose mean spectra are very similar. They suggested that these spectral differences might be the underlying cause of the classification issues. To address this, they proposed a novel approach, using a feature space that hasn't learned to distinguish between real and fake classes. This feature space is defined by CLIP: ViT/14, a vision transformer trained on a 400 million image dataset. The ability of such model to capture a wide range of image details makes it a potentially great candidate for unbiased feature recognition.

For the classification task of this problem, simple methods like k-Nearest Neighbors and Linear Classifiers were employed, taking advantage of the strengths of the CLIP-defined feature space. This approach, which focuses on the unbiasedness of the feature space, presents a promising direction in deepfake detection.

Notably, this paper has achieved state-of-the-art results. Due to its great results, it will also be used as a baseline for this project. This will achieve a comparison of our enhancements/modifications with one of the leading models in the field.

4 Our proposed method

To address the challenge of generalizing the detection of deepfakes generated by various generative models, in this report a series of innovative approaches are explored. Our methodology builds on the simple architecture used in Wang et al.'s study, specifically using a ResNet50 model pretrained on ImageNet. Additionally we included the augmentation approach in the aforementioned study.

The core of our proposition lies in investigating the effects of various preprocessing techniques and frequency transformations on deepfake detection performance. We aim to examine these methods both individually and in combination, to understand their impact on improving detection accuracy

4.1 Augmentations

In this study, we replicate the augmentation techniques used by Wang et al. to ensure consistency and comparability of our methodology. The specific augmentations adopted were

- **Random Left-Right Flipping and Cropping:** All images in the training set were subject to random left-right flipping, introducing variability in the dataset. Additionally, these images were cropped to a resolution of 224 pixels, aligning with the input size requirements of the ResNet50 model.

- **Blur+JPEG (0.5):** Each image in the training set has a 50% chance of being both blurred and JPEG compressed. This dual augmentation, aims to simulate common image alterations that occur in real-world scenarios. The details of the above transformations are the following:

  - **Gaussian Blur:** When used the blur was applied before the cropping, with the blur intensity σ randomly chosen from a uniform distribution ranging from 0 to 3.
  - **JPEG Compression:** When used the JPEG compression was performed using two popular libraries: OpenCV [1] and the Python Imaging Library (PIL) [2]. The quality

of compression was varied, selected randomly from a uniform distribution of values between 30 and 100.

4.2 Pre-processing

4.2.1 Low-Pass

By applying a low-pass, we reduce image noise and smooth out details (removing edges), which could be helpful in the enhancement of the performance of the deepfake detectors. To achieve that we employ a median blur with a kernel of size 5. The effect of this modification can be seen in Figure 1.

Figure 1: Visualization of the effect of the low-pass filter (median blur).

4.2.2 High-Pass

The application of a high-pass emphasizes on finer details in images, potentially revealing artifacts distinct to real, GAN, and diffusion models. The high-pass filtering involves subtracting a median blur (i.e. the low-pass) from each image, highlighting those finer details. The result can be seen in Figure 2.

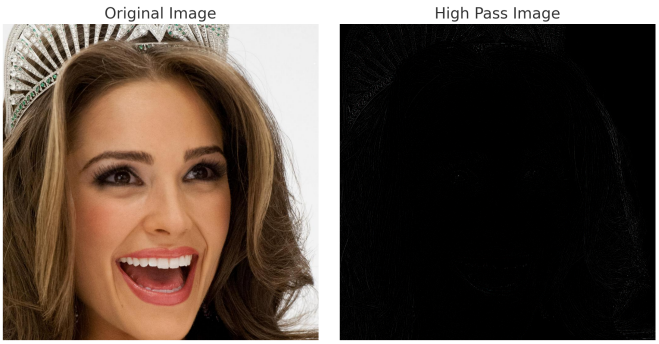

Figure 2: Visualization of the effect of the high-pass filter (median blur subtraction).

In Figure 2, the high-pass can not be easily seen, due to the low values of many of the resulting pixels, so for visual purposes the images had been binarized with various thresholds (i.e. the pixel values vary from 0 to 256 so in order to make the images binary a threshold had to be chosen within that range).

The heatmaps that can be seen in Figure 4 are generated according to the following process:

1. **High-Pass:** Each image in a given dataset is converted to a grayscale and then subjected to a high-pass (median blur subtraction)

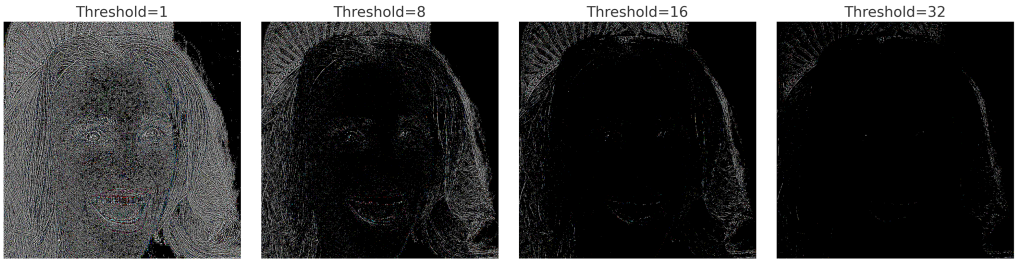

Figure 3: Visualization of thresholded high-pass.

2. **Heatmap:** Visual representation of the frequency of non-zero pixels across all images. Areas with higher frequencies of non-zero pixels appear more prominently on the heatmap.

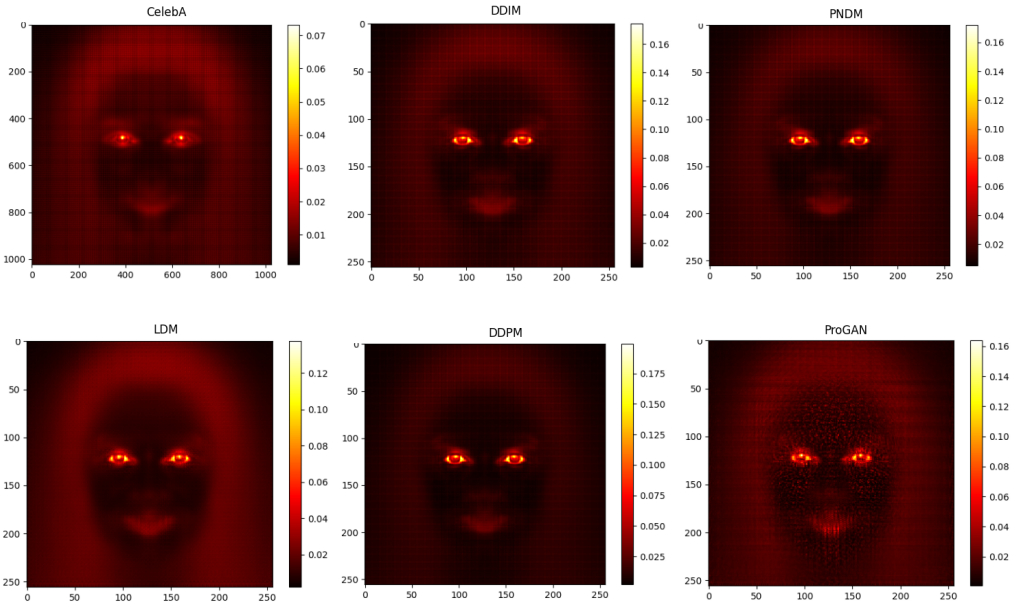

Figure 4: Frequency heatmaps of the non-thrsholded high-pass filters, for each of our datasets.

We can clearly separate the above heatmaps into three categories, just by looking at the characteristics of the "mean faces". These categories happen to exactly separate the two generative families and the real dataset. The real dataset showcases much lower probability of occurrence for each pixel probably due to the inherent variability of placement of the characteristics of each face, despite that managing to highlight prominent characteristics (e.g. eyes, mouth, nose, general face shape). Similarly to the real dataset the diffusion models (DDIM, PNDM, DDPM, LDM) have similar features being manifested, however if we look at the scale next to the heatmaps of the diffusion models the variability of these datasets are much lower than the real images (higher frequency implies lower variability). Now in the case of the GAN model we are looking at (ProGAN) we have a very peculiar wave-like patterns on the face making it very unique and identifiable. The similarities between diffusion models and real images, and the uniqeness of the GAN images could be potentially linked to the similarities found by Ojha et al. when reproducing the frequency spectra from Zhang et al. This insinuates that there are clear characteristics to one dataset depending on the model it was generated from, the question that needs to be asked is whether these differences can be identified by our detection model in order to generalize well, or are they merely present upon averaging? Additionally it would be interesting if the thresholding

performed for visualization purposes could have some positive effect on generalization.

4.2.3 Sharpened Edge Detection

The next pre-processing technique we explore is sharpened edge detection. This method involves first enhancing the sharpness of an image, in order to enhance the edge detection as it can be seen in Figures 5, 6, 7. We can see that for the fake images we have improvements in the edge detection whereas in the real images there is some deterioration but this kind of processing could improve the cross model generalization of our model (the simple edge detection will also be evaluated).

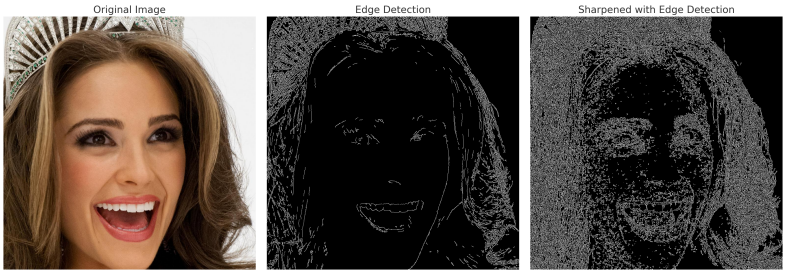

Figure 5: Real images (CelebA)

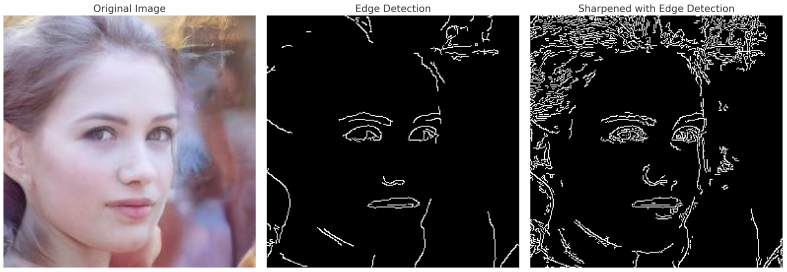

Figure 6: Diffusion model generated images (DDIM)

Figure 7: GAN generated images (ProGAN)

To follow that, the frequency heatmaps generated for the high-passes were also created for the sharp edge detection images, as they can be seen in Figure 8.

These heatmaps again seem to correctly categorize into three distinct classes separating the two generative models and the real dataset, the similarities between diffusion model generated images and real images are still present, whereas the GAN generated dataset has a very distinct characteristics on the face similarly to the high-pass heatmaps.

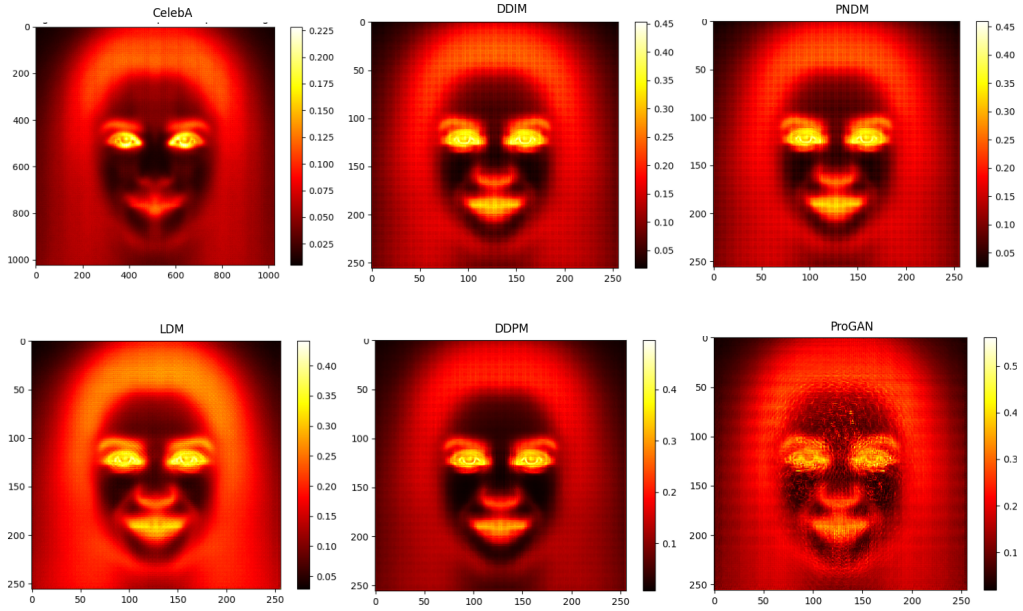

Figure 8: Frequency heatmaps of the sharp edge detection modified images, for each of our datasets.

4.3 Frequency Transformation

An important aspect of this study involves exploring frequency transformations as a way to improve generalization of deepfake detection. We will examine two such frequency transformations, the Fast Fourier Transform (FFT) and the Discrete Cosine Transform (DCT). Both of them were implemented by applying the transformation on each color channel individually and afterwards concatenating the 3 resulting spectra together.

4.3.1 Fast Fourier Transform

The Fast Fourier Transform (FFT) is a tool in digital signal processing, used for analyzing the frequency components of signals, including images. In the context of image processing, FFT transforms an image from the spatial domain to the frequency domain. This transformation allows us to analyze the image in terms of its frequency components, which can reveal unique characteristics often invisible in the spatial domain.

In the context of deepfake detection, FFT's ability to uncover subtle, repetitive patterns and anomalies becomes key. Such characteristics were found by Zhang et al. on GAN generated images, presenting grid-like patterns in the average FFT spectra of each high-pass filtered image. We reproduced these spectra for each dataset we worked with in order to reproduce these patterns and potentially uncover something new. The results can be seen in Figure 9. Similar to Wang et al. and Zhang et al., we see a distinct and repeated pattern in ProGAN, caused by the upsampling component included in the common GAN pipeline. However, this pattern is missing in the fake images from diffusion models (note that LDM displays some faint spots in its spectrum, which is intriguing), similar to images from a real distribution.

4.3.2 Discrete Cosine Transform

The Discrete Cosine Transform (DCT) is another important tool in digital signal processing, especially relevant in image analysis. DCT is used to convert images from the spatial to the

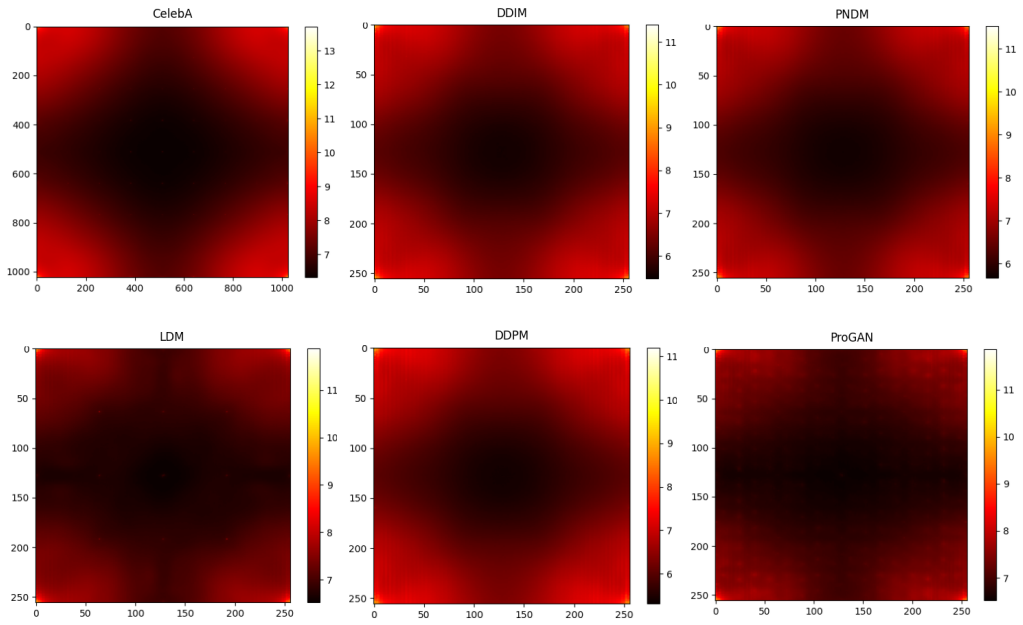

Figure 9: Average FFT spectra for each of our datasets.

frequency domain. However, contrary to FFT, DCT focuses on cosine functions, which are particularly good at separating images into parts of varying importance (frequencies).

In deepfake detection, the application of DCT can help discern subtle patterns and discrepancies not visible in the spacial domain that could be crucial in identifying fake images. In an attempt to unveil patterns like the ones found by Zhang et al. with the FFTs we perform the same process calculating the average spectra of each high-pass filtered image using the DCT, obtaining the spectra seen in Figure 10. We can see very similar characteristics with the FFT case where ProGAN has a repeated grid pattern, whereas the diffusion model and real datasets are missing this pattern.

4.4 Combination of the two

Our study proceeds by investigating the potential synergies of combining pre-processing techniques with frequency transformations (FFT and DCT) enhancing deepfake detection. The process involves the application of augmentations across all images, followed by the implementation of a selected pre-processing technique. Subsequently, we apply a frequency transformation to these pre-processed images.

In selecting combinations, we prioritize those that preserve the informational content of the images, specifically avoiding techniques that lead to excessive simplifications, such as binarization. The objective of this phase is to investigate whether the integration of these methods can more effectively differentiate real images from synthetic ones across various generative models, thereby improving the generalizability and accuracy of deepfake detection.

4.5 Datasets

For the purpose of this research, we have used six different datasets. The datasets employed in our experiments are categorized into 3 types: Real, GAN and Diffusion.

Table 1 provides a detailed summary of the datasets used, including their type, size, format and the division of the data into training, validation, and testing sets.

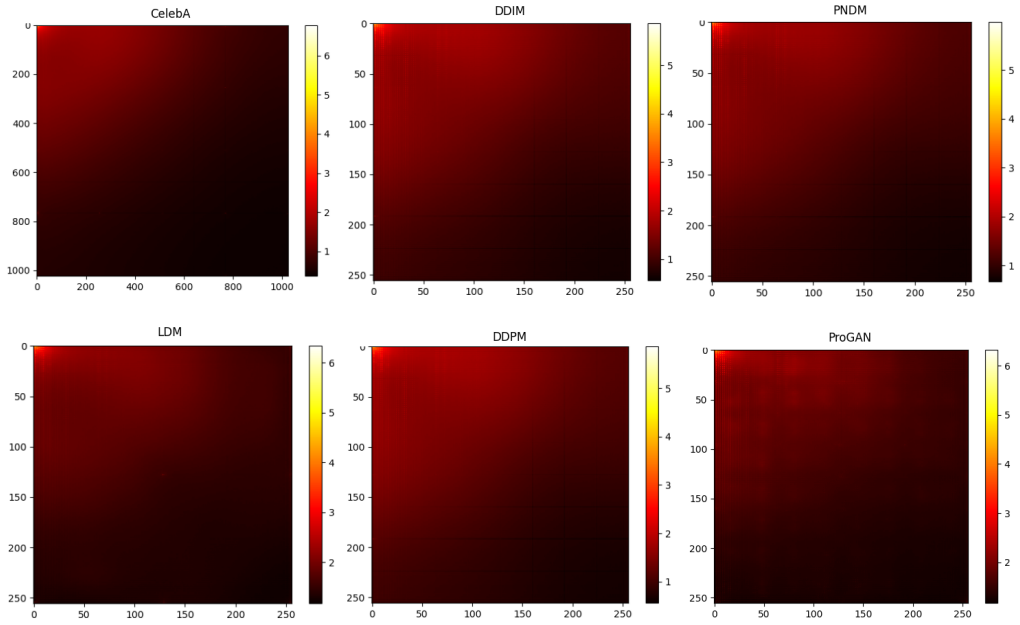

Figure 10: Average DCT spectra for each of our datasets.

Type	Dataset	Division (train/val/test)	Size	Format
Real	CelebA-HQ-img [7]	24k/3k/3k	1024x1024	JPEG
GAN	ProGAN [5]	40k/5k/5k	256x256	PNG
Diffusion	DDIM_200 [10]	32k/4k/4k	256x256	PNG
	PNDM_200 [6]	32k/4k/4k	256x256	PNG
	DDPM_200 [4]	32k/4k/4k	256x256	PNG
	LDM_200 [9]	32k/4k/4k	256x256	PNG

Table 1: Summary of datasets used in the study

4.6 Experimental Setup

In this section, we will detail the experimental setup designed to evaluate the effectiveness of various deepfake detection techniques. The central aspect of our study is to train a pre-trained ResNet50 model tailored to the specific needs of our experiments.

Training Datasets: To effectively train the ResNet50 model, we selected representative datasets from each generative family of interest. ProGAN, and PNDM, to represent GAN-generated images and diffusion models respectively.

Wang et al. Baseline: As one of our baselines, we retrained the model from Wang et al. using our selected datasets[1]. This retraining was essential to ensure that the baseline model is on equal footing with our models, considering the specific nature of our datasets. The retraining was conducted without any additional pre-processing or transformation techniques.

Ojha et al. Baseline: In contrast, the model from Ojha et al., our other baseline, was originally trained on a general, internet-scale dataset not focused exclusively on faces. Due to the vast scope and scale of their training data, it was not feasible to retrain this model using our datasets. Therefore, it serves as a baseline in its original form, providing a broader perspective on deepfake detection.

[1]The Wang et al. baseline is essentially identical to our model, the only difference is that Wang et al. uses only data augmentation, whereas we use pre-processing and/or transformations in our models (which is our point of investigation).

5 Results

5.1 Pre-Processing

Our analysis begins with an assessment of the impact of different pre-processing techniques on deepfake detection. The results are evaluated in terms of accuracy and average precision (AP). The experiments were conducted using a consistent augmentation strategy Blur+JPEG(0.5) and then applying different pre-processing methods without any frequency transformations.

The results are summarized in Table 2, comparing our models with the baseline models from Ojha et al. and Wang et al.

Category	**Options**			**Test**	
	Augmentations	**Pre-Processing**	**Transforms**	**Datasets**	**Acc./AP**
Ojha [8]	(Does not apply)	(Does not apply)	(Does not apply)	PNDM	0.6845/0.7849
				ProGAN	**0.8035**/0.9078
				DDIM	0.6855/0.7847
				DDPM	0.5875/0.6543
				LDM	**0.8295**/**0.9405**
Wang [11]	BlurJPEG(0.5)	None	None	PNDM	0.7337/0.9925
				ProGAN	0.7671/0.9977
				DDIM	0.7323/0.9734
				DDPM	0.7291/**0.9371**
				LDM	0.2533/0.4007
Ours	BlurJPEG(0.5)	High Pass	None	PNDM	**0.7651**/**0.9992**
				ProGAN	0.7947/**0.9999**
				DDIM	**0.7590**/**0.9761**
				DDPM	**0.7380**/0.9346
				LDM	0.3670/0.5343
Ours	BlurJPEG(0.5)	Low Pass	None	PNDM	0.6599/0.9778
				ProGAN	0.7036/0.9954
				DDIM	0.6560/0.9516
				DDPM	0.6544/0.9048
				LDM	0.1489/0.3759
Ours	BlurJPEG(0.5)	Edge	None	PNDM	0.5964/0.8801
				ProGAN	0.6679/0.9921
				DDIM	0.5901/0.8647
				DDPM	0.5337/0.7432
				LDM	0.2864/0.4923
Ours	BlurJPEG(0.5)	Sharp Edge	None	PNDM	0.5883/0.8741
				ProGAN	0.6614/0.9904
				DDIM	0.5880/0.8591
				DDPM	0.5361/0.7417
				LDM	0.2951/0.4925

Table 2: Results of pre-processing applied individually, for our models trained on ProGAN and PNDM, with the best performance of each dataset being in **bold**.

Key Observations:

- **High-Pass:** Our model with high-pass pre-processing showed a slight improvement in detection accuracy and AP for most datasets compared to the baseline models. This suggests that the high-pass, which emphasizes on the finer details in images, may be effective in uncovering subtle characteristics introduced by generative models. This could indicate that generative models, tend to leave behind high-frequency signatures that high-pass filtering helps to expose.

- **Low-Pass:** We can observe a performance drop with the low-pass pre-processing, particularly with the LDM dataset. The smoothing effect of the low-pass filter, could potentially obscure subtle artifacts present in fake images.

- **Edge Detection:** The performance decrease noted with the edge detection, indicates a potential challenge in using this technique in deepfake detection despite its similarities to the high-pass technique. This result might be linked to the nature of edge detection, whose goal is to identify the boundaries and contours of an image. While this can enhance certain aspects of an image, it might also lead to the loss of subtle textural information. The technique's emphasis on edges could be counter productive, so this finding suggests a more nuanced pre-processing strategy.
- **Sharp Edge:** The sharp edge pre-processing showed similar trends to low pass and edge techniques, indicating a potential decrease in performance for certain datasets. This suggests that the additional sharpening step in this method does not necessarily contribute to enhancing the model's performance

Following this analysis, we will explore the performance of thresholded high-pass pre-processing to determine its impact.

5.1.1 Thresholded High-Pass

Following the examination of various pre-processing techniques, we explored the effect of applying different threshold values in the high-pass processing. The results can be seen in Table 3.

Category	**Options**			**Test**	
	Augmentations	**Pre-Processing**	**Transforms**	**Datasets**	**Acc./AP**
Ours	BlurJPEG(0.5)	High Pass	None	PNDM	**0.7651**/0.9992
				ProGAN	**0.7947**/**0.9999**
				DDIM	**0.7590**/0.9761
				DDPM	**0.7380**/**0.9346**
				LDM	0.3670/0.5343
Ours	BlurJPEG(0.5)	High Pass (1)	None	PNDM	0.7041/**0.9988**
				ProGAN	0.7411/0.9998
				DDIM	0.6399/0.9397
				DDPM	0.6057/0.8784
				LDM	0.2251/0.4146
Ours	BlurJPEG(0.5)	High Pass (8)	None	PNDM	0.6567/0.9936
				ProGAN	0.7001/0.9994
				DDIM	0.6567/**0.9936**
				DDPM	0.6129/0.8759
				LDM	0.2071/0.4494
Ours	BlurJPEG(0.5)	High Pass (16)	None	PNDM	0.6316/0.9794
				ProGAN	0.7084/0.9979
				DDIM	0.633/0.9198
				DDPM	0.6057/0.8660
				LDM	0.3383/**0.6246**
Ours	BlurJPEG(0.5)	High Pass (32)	None	PNDM	0.6316/0.9222
				ProGAN	0.6908/0.9861
				DDIM	0.6183/0.8857
				DDPM	0.5906/0.8451
				LDM	**0.4223**/0.4146

Table 3: Results of thresholded high-pass applied individually, for our models trained on ProGAN and PNDM, with the best performance of each dataset being in **bold**.

We observe a general trend: increasing the threshold value typically results in decline both in accuracy and AP. This suggests that the use of a threshold leads to a significant information loss, resulting in worse deepfake detection performances. Interestingly, the impact of thresholding was not uniform across all datasets. For example, the LDM dataset exhibited significant variability in performance, reflecting a particular sensitivity to different threshold levels.

We can conclude that optimal threshold value is dataset-specific, showcasing that in some case it may be beneficial to add a threshold like in the case of the LDM dataset here.

5.2 Frequency Transformations

We now delve into the influence of frequency transformations, specifically, FFT and DCT, on the accuracy and AP of deepfake detection. These transformations are applied following the consistent use of the Blur+JPEG(0.5) augmentation, with no additional pre-processing.

The results, as shown in Table 4, compare the effectiveness of these transformations in our model against the baseline performances from Ojha et al. and Wang et al.

Category	**Options**			**Test**	
	Augmentations	**Pre-Processing**	**Transforms**	**Datasets**	**Acc./AP**
Ojha [8]	(Does not apply)	(Does not apply)	(Does not apply)	PNDM	0.6845/0.7849
				ProGAN	0.8035/0.9078
				DDIM	0.6855/0.7847
				DDPM	0.5875/0.6543
				LDM	0.8295/0.9405
Wang [11]	BlurJPEG(0.5)	None	None	PNDM	0.7337/0.9925
				ProGAN	0.7671/0.9977
				DDIM	0.7323/0.9734
				DDPM	0.7291/0.9371
				LDM	0.2533/0.4007
Ours	BlurJPEG(0.5)	None	FFT	PNDM	**0.9983**/**0.9999**
				ProGAN	**0.9986**/**1.0000**
				DDIM	**0.9836**/**0.9997**
				DDPM	**0.9736**/**0.9995**
				LDM	**0.9973**/**1.0000**
Ours	BlurJPEG(0.5)	None	DCT	PNDM	0.5553/0.8839
				ProGAN	0.5944/0.9069
				DDIM	0.5363/0.7936
				DDPM	0.5311/0.7899
				LDM	0.5300/0.8550

Table 4: Results of frequency transformations applied individually, for our models trained on ProGAN and PNDM, with the best performance of each dataset being in **bold**.

Key Observations:

- **FFT:** The application of FFT to our model shows a remarkable improvement in both accuracy and AP across almost all datasets. This indicates a high efficacy of FFT in enhancing the model's ability to distinguish between real and synthetic images. The significant improvements suggest that FFT is adept at uncovering subtle, yet critical, frequency-based artifacts in deepfake images.

- **DCT:** Conversely, the implementation of DCT yields notably lower performance metrics compared to FFT. The decline in accuracy and AP suggests that, unlike FFT, DCT may not be as effective in isolating deepfake indicators within these datasets. This could be due to the nature of DCT in processing image frequencies, which might not align as effectively with the specific characteristics of deepfake artifacts.

5.3 Combination

To further our understanding of deepfake detection, we explored the combined effects of pre-processing techniques with frequency transformations. The experiments involved applying Blur +JPEG(0.5) augmentation, followed by a pre-processing technique and then a frequency transformation.

The results of these combinations, as compared to the baseline models from Ojha et al. and Wang et al., are summarized in Table 5.

Category	Options			Test	
	Augmentations	**Pre-Processing**	**Transforms**	**Datasets**	**Acc./AP**
Ojha [8]	(Does not apply)	(Does not apply)	(Does not apply)	PNDM	0.6845/0.7849
				ProGAN	0.8035/0.9078
				DDIM	0.6855/0.7847
				DDPM	0.5875/0.6543
				LDM	**0.8295**/**0.9405**
Wang [11]	BlurJPEG(0.5)	None	None	PNDM	0.7337/0.9925
				ProGAN	0.7671/**0.9977**
				DDIM	0.7323/0.9734
				DDPM	0.7291/**0.9371**
				LDM	0.2533/0.4007
Ours	BlurJPEG(0.5)	High Pass	FFT	PNDM	0.5954/0.8241
				ProGAN	0.6456/0.9034
				DDIM	0.5369/0.4149
				DDPM	0.4841/0.3919
				LDM	0.5400/0.4993
Ours	BlurJPEG(0.5)	Low Pass	FFT	PNDM	**0.9577**/**0.9980**
				ProGAN	**0.9554**/0.9965
				DDIM	**0.9509**/**0.9943**
				DDPM	**0.9400**/0.9910
				LDM	0.8186/0.9355
Ours	BlurJPEG(0.5)	Sharp Edge	FFT	PNDM	0.5477/0.4813
				ProGAN	0.6274/0.6531
				DDIM	0.5359/0.4707
				DDPM	0.4730/0.4346
				LDM	0.4894/0.4715
Ours	BlurJPEG(0.5)	High Pass	DCT	PNDM	0.6866/0.9912
				ProGAN	0.7229/0.9907
				DDIM	0.624/0.8888
				DDPM	0.5794/0.8045
				LDM	0.6281/0.9232
Ours	BlurJPEG(0.5)	Low Pass	DCT	PNDM	0.5684/0.8397
				ProGAN	0.5739/0.5125
				DDIM	0.5686/0.7719
				DDPM	0.5691/0.8045
				LDM	0.5139/0.4562
Ours	BlurJPEG(0.5)	Sharp Edge	DCT	PNDM	0.7/0.7508
				ProGAN	0.7501/0.8772
				DDIM	0.6999/0.7377
				DDPM	0.6619/0.6544
				LDM	0.6451/0.7013

Table 5: Results of pre-processing applied along with frequency transformations, for our models trained on ProGAN and PNDM, with the best performance of each dataset being in **bold**.

Key Observations:

- **High-Pass + FFT:** Combining high pass preprocessing with FFT generally resulted in decreased performance across most datasets. This suggests that while each method individually enhances detection, their combination might be masking of crucial features.

- **Low-Pass + FFT:** In contrast, low pass preprocessing combined with FFT showed significant improvement in detection accuracy and AP. This indicates a complementary relationship where low pass filtering's smoothing effect synergizes well with FFT's frequency analysis.

- **High-Pass & Low-Pass + FFT:** The combination of high pass and low pass preprocessing with DCT exhibited mediocre results, suggesting that the effectiveness of DCT in conjunction with these pre-processing methods is somewhat ineffective.

5.4 Review of Results

In Table 6, we look at the best performing model in each case and comparing them with our baselines. We can clearly see that the FFT alone outperforms every other model by a large margin, with the FFT in combination with a low-pass trailing achieving very good results as well. However, our best performing pre-processing, the high-pass has similar results with the two baselines, slightly beating Wang et al's model in all datasets and Ojha et al's model being able to perform decently for the LDM datasets manages to beat it in that domain.

Category	**Options**			**Test**	
	Augmentations	**Pre-Processing**	**Transforms**	**Datasets**	**Acc./AP**
Ojha [8]	(Does not apply)	(Does not apply)	(Does not apply)	PNDM	0.6845/0.7849
				ProGAN	0.8035/0.9078
				DDIM	0.6855/0.7847
				DDPM	0.5875/0.6543
				LDM	0.8295/0.9405
Wang [11]	BlurJPEG(0.5)	None	None	PNDM	0.7337/0.9925
				ProGAN	0.7671/0.9977
				DDIM	0.7323/0.9734
				DDPM	0.7291/0.9371
				LDM	0.2533/0.4007
Ours	BlurJPEG(0.5)	High Pass	None	PNDM	0.7651/0.9992
				ProGAN	0.7947/0.9999
				DDIM	0.7590/0.9761
				DDPM	0.7380/0.9346
				LDM	0.3670/0.5343
Ours	BlurJPEG(0.5)	None	FFT	PNDM	**0.9983**/**0.9999**
				ProGAN	**0.9986**/**1.0000**
				DDIM	**0.9836**/**0.9997**
				DDPM	**0.9736**/**0.9995**
				LDM	**0.9973**/**1.0000**
Ours	BlurJPEG(0.5)	Low Pass	FFT	PNDM	0.9577/0.9980
				ProGAN	0.9554/0.9965
				DDIM	0.9509/0.9943
				DDPM	0.9400/0.9910
				LDM	0.8186/0.9355

Table 6: Best results of each of the 3 scenarios (pre-processing alone, transformation alone, combination of the two), for our models trained on ProGAN and PNDM, with the best performance of each dataset being in **bold**.

6 Discussion

This study's findings offer valuable insight in the field of deepfake detection, particularly in enhancing generalizability across various generative models. However an important note to consider is that the real dataset used was composed of 1024x1024 JPEG images, whereas the fake datasets consisted of 256x256 PNG images. This difference in the two datasets may have influenced the effectiveness of certain models, for instance the model may learn to classify JPEGs vs PNGs which is not at all what we are looking for. So the variation in image resolutions and formats requires closer examination.

Our analysis revealed that the FFT, applied independently, proved to be very effective, outperforming other combinations of pre-processing and frequency transformations. This highlights FFT's capability in capturing subtle frequency-based anomalies characteristic of synthetic images, which are often not detectable in the spatial domain.

However, the study also opens avenues for further exploration. The field of image preprocessing is vast, and numerous techniques remain untested in the context of deepfake detec-

tion. For instance, the application of Sobel filters, known for edge detection, could offer additional insights into the textural differences between real and synthetic images. Moreover, depth estimation heatmaps could provide a novel perspective in distinguishing between images, leveraging depth-related information which might be manipulated or absent in deepfakes.

While this study has made significant progress in identifying effective techniques for deepfake detection, it also highlights the vast potential for further research. Key to this future exploration is the examination of model performance against new GAN models, which will show us how current models generalize to unseen GAN-generated images. Additionally, the research should look into the impact of various training dataset combinations, assessing how these variations influence the effectiveness and adaptability of the detection models. The investigation could also extend to alternative pre-processing methods, exploring the potential enhancements these techniques may offer in terms of model performance. A particularly innovative approach will involve training multiple models using a range of inputs, including original, pre-processed, and transformed images. The feature maps from these models will be integrated before the Fully Connected layer, aiming to enrich the classification feature map. This strategy holds promise for a more comprehensive approach to deepfake detection, potentially leading to significant improvements in distinguishing GAN-generated images from authentic ones.

7 Conclusion

In conclusion, this project provided insights into the characteristics of images generated by GAN models. A key finding was the identification of a pattern where generative models consistently place facial characteristics (e.g. eyes, mouth, face contour) in similar locations, whereas real images showcase greater variability on the location of such characteristics, a phenomenon observed while analyzing frequency heatmaps. Furthermore, it was observed that GAN models introduce high-frequency artifacts within the face of the generated images. These artifacts provide a unique signature that can be leveraged for detection.

In terms of practical applications, our models proved very effective in deepfake detection, in two particular scenarios. The first scenario consisted of using the Fast Fourier Transform (FFT) alone, which proved efficient in isolating and identifying these high-frequency artifacts. The second scenario combined FFT with a Low-Pass filter. These findings not only advance our understanding of the intricacies of GAN-generated images but also highlight the potential enhancements in generalization across generative families through frequency spectra and well tailored pre-processing.

References

[1] Gary Bradski, Adrian Kaehler, et al. Open source computer vision library. `https://opencv.org/`, 2000.

[2] Alex Clark et al. Pillow (pil fork). `https://python-pillow.org/`, 2010.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020.

[5] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[6] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022.

[7] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738, 2015.

[8] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24480–24489, June 2023.

[9] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *arXiv preprint arXiv:2112.10752*, 2021.

[10] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[11] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A. Efros. Cnn-generated images are surprisingly easy to spot... for now. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8692–8701, 2020.

[12] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. Detecting and simulating artifacts in gan fake images. In *2019 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2019.

A Generative Models

Generative models are a class of algorithms in machine learning that focus on generating new data samples that resemble the training data. These models are widely used for tasks such as image generation, text-to-image synthesis etc. Two prominent types of generative models are Diffusion Models and Generative Adversarial Networks (GANs).

A.1 Diffusion Models

Diffusion models are a class of generative models that have gained popularity for their ability to produce high-quality, detailed samples. These models work by gradually adding noise to a data sample until it becomes a meaningless signal, and then learning to reverse this process to generate new data. They have been particularly successful in tasks like image and audio generation, often outperforming other generative models in terms of the quality and realism of the generated samples.

A.2 Generative Adversarial Networks (GAN) models

GANs consist of two neural networks, the generator and the discriminator, which are trained simultaneously through adversarial processes. The generator creates data samples, while the discriminator evaluates them against real data, learning to distinguish between the two. The generator's goal is to produce data so convincing that the discriminator cannot tell if it is real or fake. This adversarial training process continues until the generator produces high-quality data. They are known for their ability to produce very realistic images but can be challenging to train and are often sensitive to the choice of model architecture and training parameters.

B Performance Metrics

B.1 Accuracy

Accuracy is one of the simplest most intuitive performance metrics used in evaluating models, especially in classification tasks. In simple terms, accuracy measures how often the model makes the correct prediction. It is calculated by dividing the number of correct predictions made by the model by the total number of predictions. However, accuracy may not always present the whole picture, especially in the case where the data is unbalanced (if one class is significantly more common than the other) other metrics like the F1-score may be more suited.

B.2 Average Precision (AP)

Average precision (AP) is a metric used to assess the quality of a model's output by considering both the precision and the recall of the predictions. Precision refers to the proportion of positive identifications that were actually correct, while recall refers to the proportion of actual positives that were identified correctly.

To calculate AP, we first compute the precision at each threshold when the recall changes and then average these precision values.

$$\text{AP} = \sum_n (R_n - R_{n-1} P_n) \tag{1}$$

where P_n and R_n are the precision and recall at the nth threshold.

AP is a more comprehensive metric, especially in cases where the balance between precision and recall is important.