# Exploring Diffusion-Generated Image Detection Methods

**Aristotelis Dimitriou**

aristotelis.dimitriou@epfl.ch

10.01.2024

École
polytechnique
fédérale
de Lausanne

# Introduction to Deepfakes

**Advancement in AI & Public Data**

- Growth in deep learning, especially GANs and diffusion models
- Access to large-scale dataset

**Result**

- Rise into realistic fake content
  → Potential for misuse
  → Need for reliable detection

# Introduction to Deepfakes

Main Face Manipulation Techniques:

1. Entire Face Synthesis: New, realistic faces.
2. Attribute Manipulation: Changing features.
3. Identity Swap: Face replacement.
4. Expression Swap: Altering expressions.

**→ In this project we focused on Entire Face Synthesis for static images**

*[1]: Tolosana et al. (2020)*

*Entire Face Synthesis* | *Attribute Manipulation* | *Identity Swap* | *Expression Swap*

# State of the Art (SoTA)
# Deepfake Detection

## Technological Approaches

- Primarily based on Convolutional Neural Networks (CNNs) models

## Effectiveness

- High effectiveness within the same generative model family (e.g. classifiers trained on ProGAN tend to successfully detect StyleGAN fakes)
- *Primary Challenge: Lack of generalizability across different families of generative models (e.g. GANs vs Diffusion models)*

## Datasets

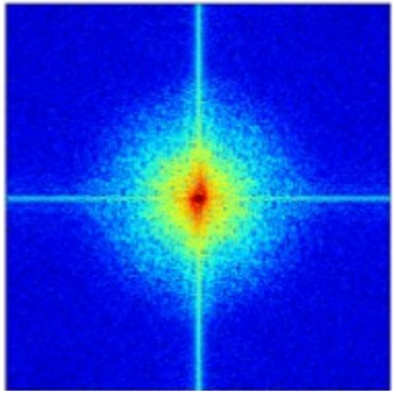
- *GAN based:* ProGAN [6], CycleGAN [7], BigGAN [8], StyleGAN [8]…
- *Diffusion based:* LDM [9], PNDM [10], DDIM [11], DDPM [12]…
- *Real:* LSUN [13], LAION [14], CelebAHQ [15]…

# Zhang et al. 2019
# Artifact identification

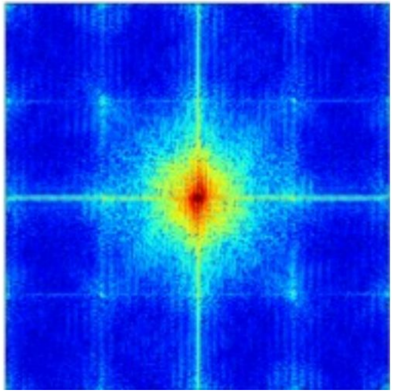
*[5]: Zhang et al. (2019)*

- Insightful research, identifying specific artifacts in the frequency domain.
- Observed periodic grid-like patterns in GAN model frequency spectra, caused by upsampling
- These findings are very interesting as they suggest that frequency domains could be potentially the key to generalization

- This serves as a foundation for our research.



Real



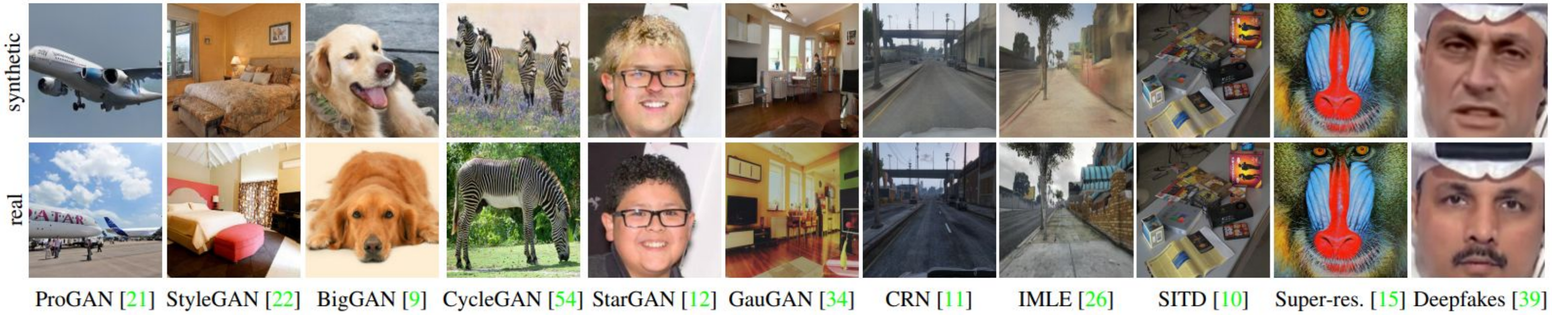
GAN

# Wang et al. 2020
# One of the best performing SoTA model

**Idea:** create a “universal” detector for telling apart real images from these generated by a CNN, regardless of architecture or dataset used.

→ With proper pre-/post-processing and data augmentation, training on data generated only by ProGAN can generalize very well for all CNN-based deepfakes

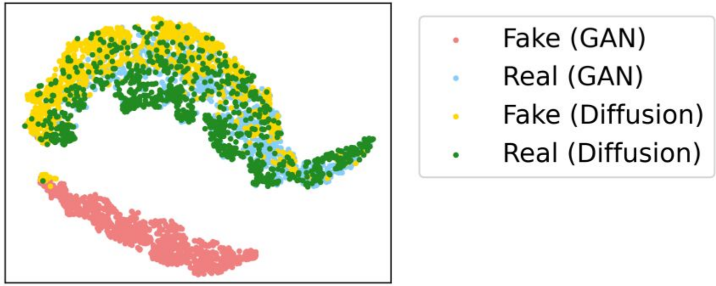
**Classifier:** ResNet50 (pre-trained on ImageNet)




*[2]: Wang et al. (2020)*

# Ojha et al. 2023 Universal Fake Detect

[4]: Ojha et al. (2023)



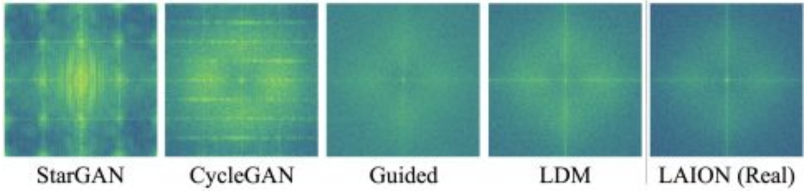

t-SNE visualization of real and fake images associated with two types of generative models. The feature space used is of a classifier trained to distinguish Fake (GAN) from Real (GAN).

## Identified Classification Issue

- Real class acts as a 'sink' class
- Include all fake images that are not from the same model as the training dataset's model

## Spectrum Differences

- Distinct spectrum characteristics in GAN vs. Diffusion model and Real images.
- These differences could explain the classification issue




*[4]: Ojha et al. (2023)*
*These spectra are obtained by applying a high-pass and subtracting the median blurred image before applying an FFT*

# Ojha et al. 2023
# Approach

**Idea:** classify in a feature space that hasn't learned to distinguish between the two classes, ensuring unbiased feature recognition for both classes.

**Feature Space** is defined by a vision transformer trained for the task of image-language alignment, CLIP: ViT/14

→ Trained on 400M images (internet-scale dataset)

→ Models general details, as well as low-level details of an image

**Classifiers**

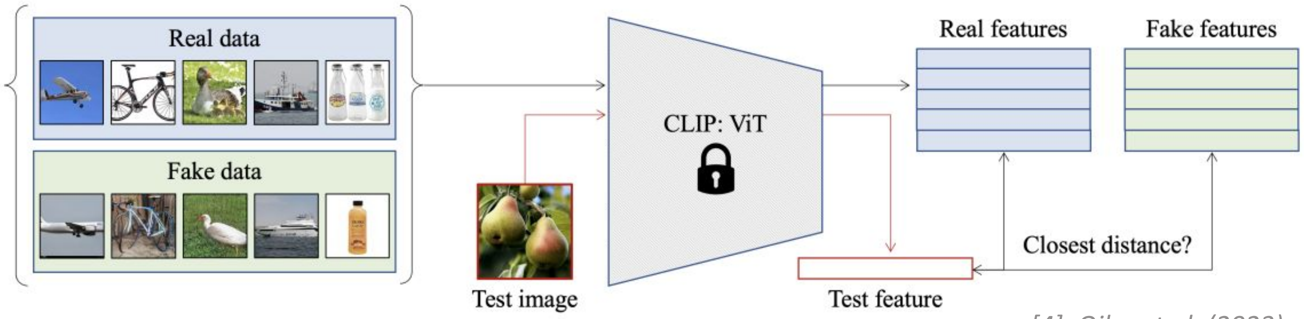
- k-Nearest Neighbors
- Linear

# Ojha et al. 2023 Approach

**Idea:** classify in a feature space that hasn't learned to distinguish
betw ... both
class

**Featu** ... k of
imag

- ...
- ... ge



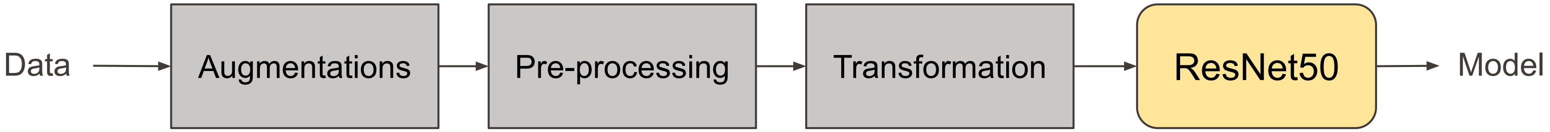

*[4]: Ojha et al. (2023)*

**Classifiers**

- k-Nearest Neighbors
- Linear

# Our proposed method

- To address this generalization challenge, we build on Wang et al.'s simple architecture and their augmentation techniques.
- The core of our proposition lies in investigating the effect of various **pre-processing** techniques and **frequency transformations** on deepfake detection performance.
- These methods will be evaluated both individually and in combination.



# Datasets

Real:

- CelebA → 1024x1024 JPEG (30k images)

Diffusion:

- PNDM → 256x256 PNG (40k images)
- DDIM → 256x256 PNG (40k images)
- DDPM → 256x256 PNG (40k images)
- LDM → 256x256 PNG (40k images)

GAN:

- ProGAN → 256x256 PNG (50k images)

*All models were **trained using ProGAN and PNDM**, having LDM, DDIM, DDPM as our generalization domain*

<u>Baselines:</u>

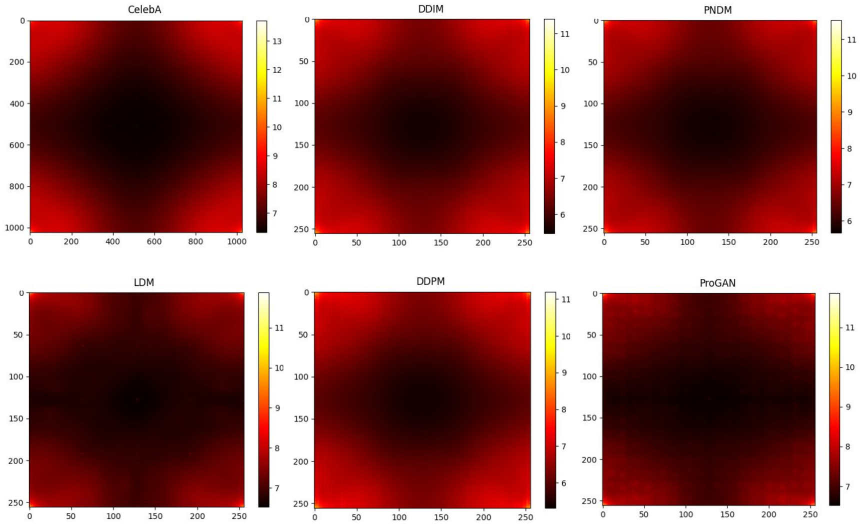
- Wang et al. is also trained with PNDM and ProGAN
- Ojha et al. uses a different approach for the prediction using a feature space extracted with non-face images (keep in mind)

# Frequency Analysis
# Fast Fourier Transform

**Analysis of average FFT**, *following Zhang et al.'s approach*

Similarly to what many other studies have found we found that:

- GAN datasets showcase unique grid-like patterns
- Diffusion model datasets have very similar average FFT spectra with the ones from real datasets.



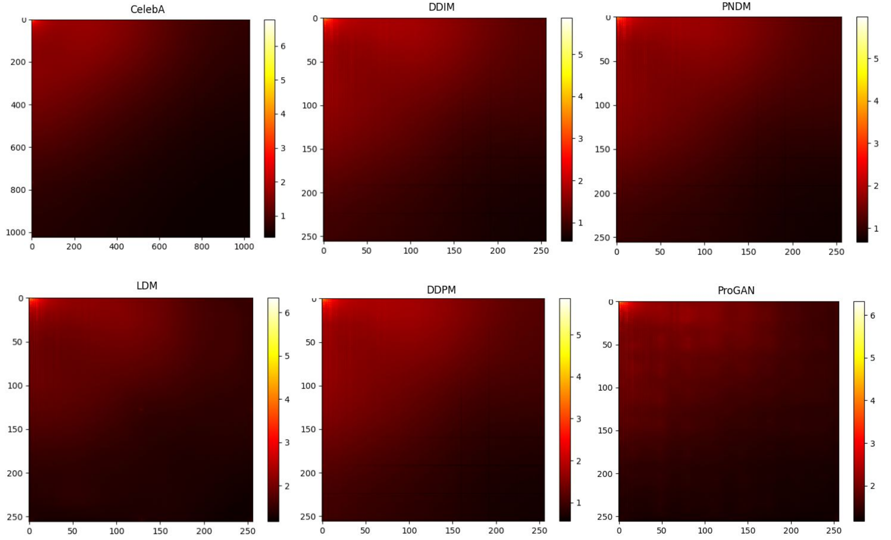

*→ Our results match what has been previously found in other studies!*
*(see Zhang et al. [5], Wang et al. [2], Ojha et al. [4])*

# Frequency Analysis
# Discrete Cosine Transform

**Analysis of average DCT**, *similar to the approach used for the FFT*

We can see that:

- Dominant low-frequency contributions in the upper-left corner.
- Gradual decrease in contribution towards higher frequencies (lower right corner).
- GAN datasets display again a grid-like pattern




→ *Our results match what has been previously found in other studies! (see Frank et al. [17])*

# Frequency Analysis Results

Both the FFT and DCT used in our experiments fed into the ResNet50 were obtained by calculating the transform for each RGB channel and then concatenating them

- FFT achieves extremely good results
- DCT however achieves performances that are close to chance

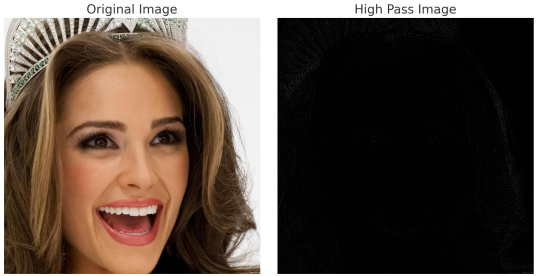
**Trained with PNDM and ProGAN**

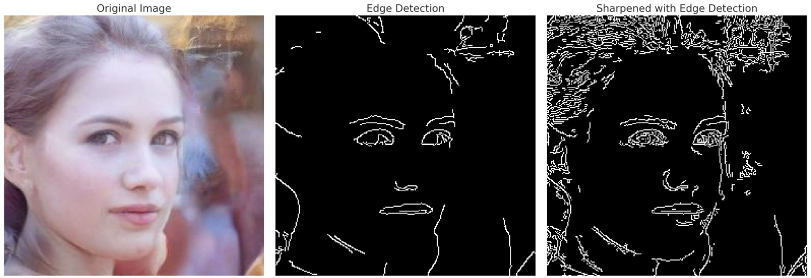
| Category | Options | | | Test | |
|---|---|---|---|---|---|
| | Augmentations | Pre-Processing | Transforms | Datasets | Acc./AP |
| Ojha [4] | (Does not apply) | (Does not apply) | (Does not apply) | PNDM | 0.6845/0.7849 |
| | | | | DDIM | 0.6855/0.7847 |
| | | | | DDPM | 0.5875/0.6543 |
| | | | | LDM | 0.8295/0.9405 |
| | | | | ProGAN | 0.8035/0.9078 |
| Wang [5] | BlurJPEG(0.5) | None | None | PNDM | 0.7337/0.9925 |
| | | | | DDIM | 0.7323/0.9734 |
| | | | | DDPM | 0.7291/0.9371 |
| | | | | LDM | 0.2533/0.4007 |
| | | | | ProGAN | 0.7671/0.9977 |
| Ours | BlurJPEG(0.5) | None | FFT | PNDM | 0.9983/0.9999 |
| | | | | DDIM | 0.9836/0.9997 |
| | | | | DDPM | 0.9736/0.9995 |
| | | | | LDM | 0.9973/1.0000 |
| | | | | ProGAN | 0.9986/1.0000 |
| Ours | BlurJPEG(0.5) | None | DCT | PNDM | 0.5553/0.8839 |
| | | | | DDIM | 0.5363/0.7936 |
| | | | | DDPM | 0.5311/0.7899 |
| | | | | LDM | 0.5300/0.8550 |
| | | | | ProGAN | 0.5944/0.9069 |

# Pre-Processing

Three pre-processing techniques were explored:

- Low Pass (Median Blur)
- High Pass (Median Blur Subtraction)
- Sharp Edge Detection (Sharpen Image → Canny Edge Detection)

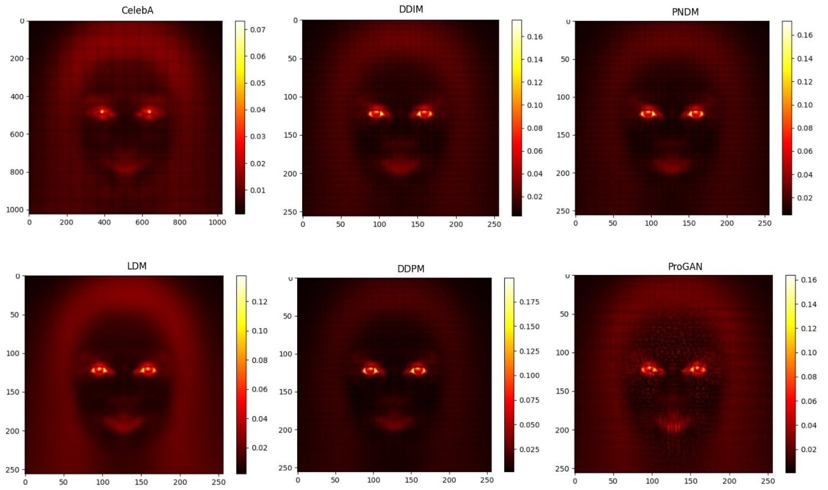



# Pre-Processing High-Pass

The heatmaps were obtained by applying the high-pass on all images and subsequently averaging all images of the dataset

In order to identify potential unique characteristics able to identify each generative model, frequency heatmaps were generated:

- There is a clear distinction between all 3 generative
- The GAN model displays a very peculiar wave-like pattern on the face
- Diffusion models have very well defined eyes
- Real images are more blurred due to greater variability of the position of the face characteristics

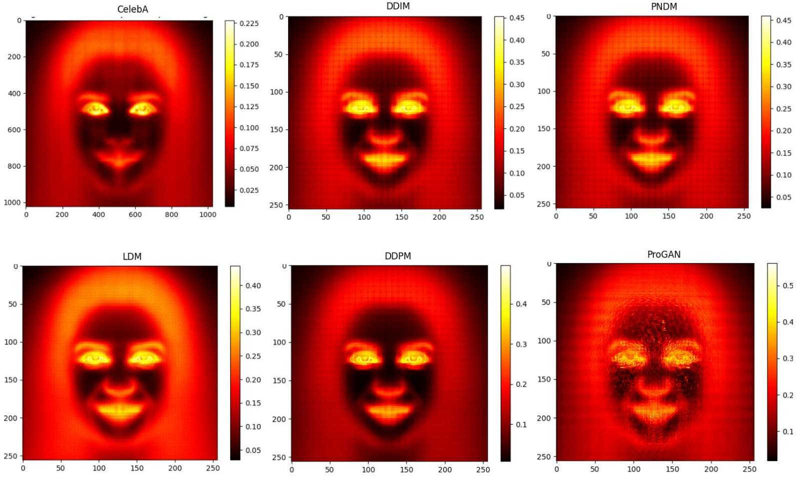


# Pre-Processing
# Sharp Edge Detection

The heatmaps were obtained by applying the sharp edge on all images and subsequently averaging all images of the dataset

A very similar approach was taken with the sharp edge pre-processing, using frequency heatmaps:

- The results yielded similar results with the high-pass frequency heatmaps
- GAN model still has an interesting pattern on the face
- Diffusion models have very pronounced eyes and mouths
- Real images still have high variability but somewhat pronounced eyes




*We can hope that these identifying characteristics will be picked by our model and separate the 3 generative models*

# Pre-Processing Results

Unfortunately we do not observe significant increase in the performances in any of the three pre-processings:

- High pass yields decent results across all datasets (except LDM)
- All other techniques underperformed and were nowhere near our baselines

**Trained with PNDM and ProGAN**

| Category | Options | | | Test | |
|---|---|---|---|---|---|
| | Augmentations | Pre-Processing | Transforms | Datasets | Acc./AP |
| Ojha [4] | (Does not apply) | (Does not apply) | (Does not apply) | PNDM | 0.6845/0.7849 |
| | | | | DDIM | 0.6855/0.7847 |
| | | | | DDPM | 0.5875/0.6543 |
| | | | | LDM | 0.8295/0.9405 |
| | | | | ProGAN | 0.8035/0.9078 |
| Wang [5] | BlurJPEG(0.5) | None | None | PNDM | 0.7337/0.9925 |
| | | | | DDIM | 0.7323/0.9734 |
| | | | | DDPM | 0.7291/0.9371 |
| | | | | LDM | 0.2533/0.4007 |
| | | | | ProGAN | 0.7671/0.9977 |
| Ours | BlurJPEG(0.5) | High Pass | None | PNDM | 0.7651/0.9992 |
| | | | | DDIM | 0.7590/0.9761 |
| | | | | DDPM | 0.7380/0.9346 |
| | | | | LDM | 0.3670/0.5343 |
| | | | | ProGAN | 0.7947/0.9999 |
| Ours | BlurJPEG(0.5) | Low Pass | None | PNDM | 0.6599/0.9778 |
| | | | | DDIM | 0.6560/0.9516 |
| | | | | DDPM | 0.6544/0.9048 |
| | | | | LDM | 0.1489/0.3759 |
| | | | | ProGAN | 0.7036/0.9954 |
| Ours | BlurJPEG(0.5) | Edge | None | PNDM | 0.5964/0.8801 |
| | | | | DDIM | 0.5901/0.8647 |
| | | | | DDPM | 0.5337/0.7432 |
| | | | | LDM | 0.2864/0.4923 |
| | | | | ProGAN | 0.6679/0.9921 |
| Ours | BlurJPEG(0.5) | Sharp Edge | None | PNDM | 0.5883/0.8741 |
| | | | | DDIM | 0.5880/0.8591 |
| | | | | DDPM | 0.5361/0.7417 |
| | | | | LDM | 0.2951/0.4925 |
| | | | | ProGAN | 0.6614/0.9904 |

# Combination of 2 Techniques
# FFT + Pre-processing

Trained with PNDM and ProGAN

- We can observe very good results for the combination FFT + Low-Pass, however they are a downgrade from the FFT
- The combination FFT + High-Pass which looked promising, gave us very mediocre results
- FFT + Sharp-Edge also had very bad performances

| Category | Options | | | Test | |
|---|---|---|---|---|---|
| | Augmentations | Pre-Processing | Transforms | Datasets | Acc./AP |
| Ojha [4] | (Does not apply) | (Does not apply) | (Does not apply) | PNDM | 0.6845/0.7849 |
| | | | | DDIM | 0.6855/0.7847 |
| | | | | DDPM | 0.5875/0.6543 |
| | | | | LDM | 0.8295/0.9405 |
| | | | | ProGAN | 0.8035/0.9078 |
| Wang [5] | BlurJPEG(0.5) | None | None | PNDM | 0.7337/0.9925 |
| | | | | DDIM | 0.7323/0.9734 |
| | | | | DDPM | 0.7291/0.9371 |
| | | | | LDM | 0.2533/0.4007 |
| | | | | ProGAN | 0.7671/0.9977 |
| Ours | BlurJPEG(0.5) | High Pass | FFT | PNDM | 0.5954/0.8241 |
| | | | | DDIM | 0.5369/0.4149 |
| | | | | DDPM | 0.4841/0.3919 |
| | | | | LDM | 0.5400/0.4993 |
| | | | | ProGAN | 0.6456/0.9034 |
| Ours | BlurJPEG(0.5) | Low Pass | FFT | PNDM | 0.9577/0.9980 |
| | | | | DDIM | 0.9509/0.9943 |
| | | | | DDPM | 0.9400/0.9910 |
| | | | | LDM | 0.8186/0.9355 |
| | | | | ProGAN | 0.9554/0.9965 |
| Ours | BlurJPEG(0.5) | Sharp Edge | FFT | PNDM | 0.5477/0.4813 |
| | | | | DDIM | 0.5359/0.4707 |
| | | | | DDPM | 0.4730/0.4346 |
| | | | | LDM | 0.4894/0.4715 |
| | | | | ProGAN | 0.6274/0.6531 |
| Ours | BlurJPEG(0.5) | None | FFT | PNDM | 0.9983/0.9999 |
| | | | | DDIM | 0.9836/0.9997 |
| | | | | DDPM | 0.9736/0.9995 |
| | | | | LDM | 0.9973/1.0000 |
| | | | | ProGAN | 0.9986/1.0000 |

# Combination of 2 Techniques DCT + Pre-processing

All of the results obtained with the DCT were very average, despite improving on the individual DCT performance, being somewhat consistent across all datasets

Trained with PNDM and ProGAN

| Category | Options | | | Test | |
|---|---|---|---|---|---|
| | Augmentations | Pre-Processing | Transforms | Datasets | Acc./AP |
| Ojha [4] | (Does not apply) | (Does not apply) | (Does not apply) | PNDM | 0.6845/0.7849 |
| | | | | DDIM | 0.6855/0.7847 |
| | | | | DDPM | 0.5875/0.6543 |
| | | | | LDM | 0.8295/0.9405 |
| | | | | ProGAN | 0.8035/0.9078 |
| Wang [5] | BlurJPEG(0.5) | None | None | PNDM | 0.7337/0.9925 |
| | | | | DDIM | 0.7323/0.9734 |
| | | | | DDPM | 0.7291/0.9371 |
| | | | | LDM | 0.2533/0.4007 |
| | | | | ProGAN | 0.7671/0.9977 |
| Ours | BlurJPEG(0.5) | High Pass | DCT | PNDM | 0.6866/0.9912 |
| | | | | DDIM | 0.624/0.8888 |
| | | | | DDPM | 0.5794/0.8045 |
| | | | | LDM | 0.6281/0.9232 |
| | | | | ProGAN | 0.7229/0.9907 |
| Ours | BlurJPEG(0.5) | Low Pass | DCT | PNDM | 0.5684/0.8397 |
| | | | | DDIM | 0.5686/0.7719 |
| | | | | DDPM | 0.5691/0.8045 |
| | | | | LDM | 0.5139/0.4562 |
| | | | | ProGAN | 0.5739/0.5125 |
| Ours | BlurJPEG(0.5) | Sharp Edge | DCT | PNDM | 0.7/0.7508 |
| | | | | DDIM | 0.6999/0.7377 |
| | | | | DDPM | 0.6619/0.6544 |
| | | | | LDM | 0.6451/0.7013 |
| | | | | ProGAN | 0.7501/0.8772 |
| Ours | BlurJPEG(0.5) | None | DCT | PNDM | 0.5553/0.8839 |
| | | | | DDIM | 0.5363/0.7936 |
| | | | | DDPM | 0.5311/0.7899 |
| | | | | LDM | 0.5300/0.8550 |
| | | | | ProGAN | 0.5944/0.9069 |

# Conclusion

- Identified that generative models generate images with facial characteristics in similar locations, through frequency heatmaps
- GAN model introduce high-frequency artifacts within the face of the generated images
- With our models we managed to achieve interesting results in two cases:
  - Individual FFT
  - FFT + Low-Pass
- Would be interesting to look at the performance of these models on unseen GAN models, or to evaluate other combination of training sets

# Future Research

- Assess how the current models generalize to previously unseen GAN-generated images.
- Evaluate the impact of various training dataset combinations on model effectiveness.
- Examine the influence of different image pre-processing techniques on model performance.
- Train distinct models using varied inputs (original, pre-processed, transformed), and integrate their feature maps prior to the Fully Connected layer to enhance the classification feature map.

# References


[1]: Tolosana et al. (2020) : https://arxiv.org/abs/2001.00179
[2]: Wang et al. (2020): https://arxiv.org/pdf/1912.11035
[3]: Ricker et al. (2023): https://arxiv.org/pdf/2210.14571
[4]: Ojha et al. (2023): https://arxiv.org/pdf/2302.10174
[5]: Zhang et al. (2019): https://arxiv.org/pdf/1907.06515
[6]: ProGAN: https://arxiv.org/abs/1710.10196
[7]: CycleGAN: https://arxiv.org/abs/1703.10593
[8]: BigGAN: https://arxiv.org/abs/1809.11096
[9]: StyleGAN: https://arxiv.org/abs/1812.04948
[10]: LDM: https://arxiv.org/abs/2112.10752
[11]: PNDM: https://arxiv.org/abs/2202.09778
[12]: DDIM: https://arxiv.org/abs/2010.02502
[13]: DDPM: https://arxiv.org/abs/2006.11239
[14]: LSUN: https://arxiv.org/abs/1506.03365
[15]: LAION: https://arxiv.org/abs/2311.13028
[16]: CelebA: https://ieeexplore.ieee.org/document/7410782
[17]: Frank et al. (2023): https://arxiv.org/abs/2003.08685

Multimedia Signal Processing Group

EPFL

https://mmspg.epfl.ch/

# Thank you!

Aristotelis Dimitriou

aristotelis.dimitriou@epfl.ch

École
polytechnique
fédérale
de Lausanne

10.01.2024

# Wang et al. 2020 Results

| Family | Name | Training settings | | | | | Individual test generators | | | | | | | | | | | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Train | Input | No. Class | Augments Blur | Augments JPEG | Pro-GAN | Style-GAN | Big-GAN | Cycle-GAN | Star-GAN | Gau-GAN | CRN | IMLE | SITD | SAN | Deep-Fake | **mAP** |
| Zhang et al. [50] | Cyc-Im | CycleGAN | RGB | – | | | 84.3 | 65.7 | 55.1 | 100. | 99.2 | 79.9 | 74.5 | 90.6 | 67.8 | 82.9 | 53.2 | 77.6 |
| | Cyc-Spec | CycleGAN | Spec | – | | | 51.4 | 52.7 | 79.6 | 100. | **100.** | 70.8 | 64.7 | 71.3 | 92.2 | 78.5 | 44.5 | 73.2 |
| | Auto-Im | AutoGAN | RGB | – | | | 73.8 | 60.1 | 46.1 | 99.9 | **100.** | 49.0 | 82.5 | 71.0 | 80.1 | 86.7 | 80.8 | 75.5 |
| | Auto-Spec | AutoGAN | Spec | – | | | 75.6 | 68.6 | 84.9 | **100.** | **100.** | 61.0 | 80.8 | 75.3 | 89.9 | 66.1 | 39.0 | 76.5 |
| Ours | 2-class | ProGAN | RGB | 2 | ✓ | ✓ | 98.8 | 78.3 | 66.4 | 88.7 | 87.3 | 87.4 | 94.0 | 97.3 | 85.2 | 52.9 | 58.1 | 81.3 |
| | 4-class | ProGAN | RGB | 4 | ✓ | ✓ | 99.8 | 87.0 | 74.0 | 93.2 | 92.3 | 94.1 | 95.8 | 97.5 | 87.8 | 58.5 | 59.6 | 85.4 |
| | 8-class | ProGAN | RGB | 8 | ✓ | ✓ | 99.9 | 94.2 | 78.9 | 94.3 | 91.9 | 95.4 | 98.9 | 99.4 | 91.2 | 58.6 | 63.8 | 87.9 |
| | 16-class | ProGAN | RGB | 16 | ✓ | ✓ | 100. | 98.2 | 87.7 | 96.4 | 95.5 | **98.1** | **99.0** | **99.7** | 95.3 | 63.1 | 71.9 | 91.4 |
| | No aug | ProGAN | RGB | 20 | | | 100. | 96.3 | 72.2 | 84.0 | **100.** | 67.0 | 93.5 | 90.3 | 96.2 | **93.6** | **98.2** | 90.1 |
| | Blur only | ProGAN | RGB | 20 | ✓ | | 100. | 99.0 | 82.5 | 90.1 | **100.** | 74.7 | 66.6 | 66.7 | **99.6** | 53.7 | 95.1 | 84.4 |
| | JPEG only | ProGAN | RGB | 20 | | ✓ | 100. | 99.0 | 87.8 | 93.2 | 91.8 | 97.5 | **99.0** | 99.5 | 88.7 | 78.1 | 88.1 | **93.0** |
| | Blur+JPEG (0.5) | ProGAN | RGB | 20 | ✓ | ✓ | 100. | 98.5 | **88.2** | 96.8 | 95.4 | **98.1** | 98.9 | 99.5 | 92.7 | 63.9 | 66.3 | 90.8 |
| | Blur+JPEG (0.1) | ProGAN | RGB | 20 | † | † | 100. | **99.6** | 84.5 | 93.5 | 98.2 | 89.5 | 98.2 | 98.4 | 97.2 | 70.5 | 89.0 | 92.6 |

*[2]: Wang et al. (2020)*

- Indeed the ProGAN classifier performs very well on CNN-based models
- Significant improvements with respect to the baseline

# Wang et al. 2020
# Tested on Diffusion models

*[3]: Ricker et al. (2023)*

| AUROC / Pd@5% / Pd@1% | Wang et al. [51] | |
|---|---|---|
| | Blur+JPEG (0.5) | Blur+JPEG (0.1) |
| ProGAN | **100.0 / 100.0 / 100.0** | **100.0 / 100.0 / 100.0** |
| StyleGAN | 98.7 / 93.7 / 81.4 | 99.0 / 95.5 / 84.4 |
| ProjectedGAN | 94.8 / 73.8 / 49.1 | 90.9 / 61.8 / 34.5 |
| Diff-StyleGAN2 | 99.9 / 99.6 / 97.9 | 100.0 / 99.9 / 99.3 |
| Diff-ProjectedGAN | 93.8 / 69.5 / 43.3 | 88.8 / 54.6 / 27.2 |
| Average | 97.4 / 87.3 / 74.3 | 95.7 / 82.4 / 69.1 |
| DDPM | 85.2 / 37.8 / 14.2 | 80.8 / 29.6 / 9.3 |
| IDDPM | 81.6 / 30.6 / 10.6 | 79.9 / 27.6 / 7.8 |
| ADM | 68.3 / 13.2 / 3.4 | 68.8 / 14.1 / 4.0 |
| PNDM | 79.0 / 27.5 / 9.2 | 75.5 / 22.6 / 6.3 |
| LDM | 78.7 / 24.7 / 7.4 | 77.7 / 24.3 / 6.9 |
| Average | 78.6 / 26.8 / 9.0 | 76.6 / 23.7 / 6.8 |

The performance of this classifier clearly deteriorates upon evaluating on diffusion models

# Wang et al. 2020
# Reproduction of Results

| Model | AP (My Results) | AP (Wang et al.) |
|---|---|---|
| ProGAN | 100.0 | 100.0 |
| StyleGAN | 98.5 | 98.5 |
| BigGAN | 88.2 | 88.2 |
| CycleGAN | 96.8 | 96.8 |
| StarGAN | 95.4 | 95.4 |
| GauGAN | 98.1 | 98.1 |
| CRN | 98.9 | 98.9 |
| IMLE | 99.5 | 99.5 |
| SITD | 92.7 | 92.7 |
| SAN | 63.9 | 63.9 |
| DeepFake | 66.3 | 66.3 |
| *Overall mAP* | 90.8 | 90.8 |

*Average Precision Results (their datasets): Using **Blur+JPEG(0.5)***

Surprisingly the results of their model underperforms for all datasets especially on ProGAN on which it should perform best

| Model | Acc./AP |
|---|---|
| PNDM | 42.7/44.2 |
| DDIM | 42.7/42.5 |
| DDPM | 42.4/39.4 |
| LDM | 43.2/45.5 |
| ProGAN | 37.7/48.2 |

*Average Precision Results (our datasets): Using **Blur+JPEG(0.5)***

# Ojha et al 2023 Results

*[4]: Ojha et al. (2023)*

"Generalization Domain"

| Detection method | Variant | Generative Adversarial Networks | | | | | | Deep fakes | Low level vision | | Perceptual loss | | Guided | LDM | | | Glide | | | DALL-E | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Pro-GAN | Cycle-GAN | Big-GAN | Style-GAN | Gau-GAN | Star-GAN | | SITD | SAN | CRN | IMLE | | 200 steps | 200 w/ CFG | 100 steps | 100 27 | 50 27 | 100 10 | | mAP |
| Trained deep network [50] | Blur+JPEG (0.1) | **100.0** | 93.47 | 84.5 | **99.54** | 89.49 | 98.15 | 89.02 | 73.75 | 59.47 | 98.24 | 98.4 | 73.72 | 70.62 | 71.0 | 70.54 | 80.65 | 84.91 | 82.07 | 70.59 | 83.58 |
| | Blur+JPEG (0.5) | **100.0** | 96.83 | 88.24 | 98.29 | 98.09 | 95.44 | 66.27 | 86.0 | 61.2 | **98.94** | **99.52** | 68.57 | 66.0 | 66.68 | 65.39 | 73.29 | 78.02 | 76.23 | 65.93 | 81.52 |
| | ViT:CLIP (B+J 0.5) | 99.98 | 93.32 | 83.63 | 88.14 | 92.81 | 84.62 | 67.23 | **93.48** | 55.21 | 88.75 | 96.22 | 55.74 | 52.52 | 54.51 | 52.2 | 56.64 | 61.13 | 56.64 | 62.74 | 73.44 |
| Patch classifier [10] | ResNet50-Layer1 | 98.86 | 72.04 | 68.79 | 92.96 | 55.9 | 92.06 | 60.18 | 65.82 | 52.87 | 68.74 | 67.59 | 70.05 | 87.84 | 84.94 | 88.1 | 74.54 | 76.28 | 75.84 | 77.07 | 75.28 |
| | Xception-Block2 | 80.88 | 72.84 | 71.66 | 85.75 | 65.99 | 69.25 | 76.55 | 76.19 | 76.34 | 74.52 | 68.52 | 75.03 | 87.1 | 86.72 | 86.4 | 85.37 | 83.73 | 78.38 | 75.67 | 77.73 |
| Co-occurence [35] | - | 99.74 | 80.95 | 50.61 | 98.63 | 53.11 | 67.99 | 59.14 | 68.98 | 60.42 | 73.06 | 87.21 | 70.20 | 91.21 | 89.02 | 92.39 | 89.32 | 88.35 | 82.79 | 80.96 | 78.11 |
| Freq-spec [53] | CycleGAN | 55.39 | **100.0** | 75.08 | 55.11 | 66.08 | **100.0** | 45.18 | 47.46 | 57.12 | 53.61 | 50.98 | 57.72 | 77.72 | 77.25 | 76.47 | 68.58 | 64.58 | 61.92 | 67.77 | 66.21 |
| **Ours** | NN, $k=1$ | **100.0** | 98.14 | 94.49 | 86.68 | 99.26 | 99.53 | **93.09** | 78.46 | 67.54 | 83.13 | 91.07 | 79.31 | 95.84 | 79.84 | 95.97 | 93.98 | 95.17 | **96.05** | 88.51 | 90.32 |
| | NN, $k=3$ | **100.0** | 98.13 | 94.46 | 86.67 | 99.25 | 99.53 | 93.03 | 78.54 | 67.54 | 83.13 | 91.06 | 79.26 | 95.81 | 79.78 | 95.94 | 93.94 | 95.13 | 94.60 | 88.47 | 90.22 |
| | NN, $k=5$ | **100.0** | 98.13 | 94.46 | 86.66 | 99.25 | 99.53 | 93.02 | 78.54 | 67.54 | 83.12 | 91.06 | 79.25 | 95.81 | 79.78 | 95.94 | 93.94 | 95.13 | 94.60 | 88.46 | 90.22 |
| | NN, $k=9$ | **100.0** | 98.13 | 94.46 | 86.66 | 99.25 | 99.53 | 91.67 | 78.54 | 67.54 | 83.12 | 91.06 | 79.24 | 95.81 | 79.77 | 95.93 | 93.93 | 95.12 | 94.59 | 88.45 | 90.14 |
| | LC | **100.0** | 99.46 | **99.59** | 97.24 | **99.98** | 99.60 | 82.45 | 61.32 | **79.02** | 96.72 | 99.00 | **87.77** | **99.14** | **92.15** | **99.17** | **94.74** | **95.34** | 94.57 | **97.15** | **93.38** |

Noticeable improvements over the best performing baseline when evaluating on unseen generative models

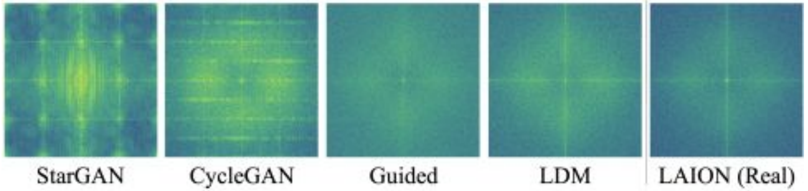
*Best performing baseline:* **+9.8 mAP** overall and **+19.49 mAP** across unseen diffusion & autoregressive models.

# Ojha et al 2023
# Performance on our datasets

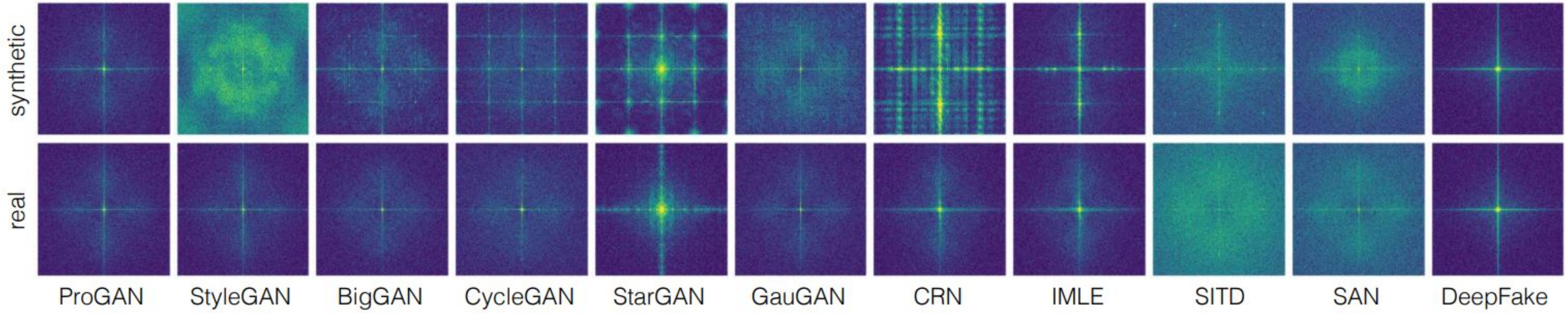
| Model | Acc./AP |
|---|---|
| PNDM | 68.5/78.5 |
| DDIM | 68.6/78.5 |
| DDPM | 58.8/65.4 |
| LDM | 83.0/94.1 |
| ProGAN | 85.4/90.8 |

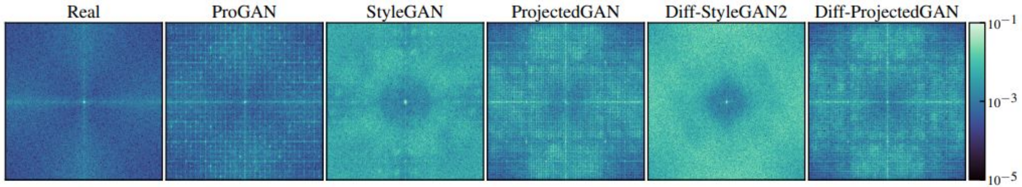
The cross-model performance, observed in the paper is verified through testing on our own datasets, were we observe similar performances

[4]: Ojha et al. (2023)

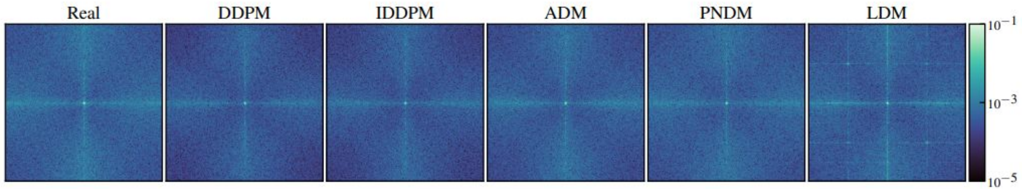



[2]: Wang et al. (2020)

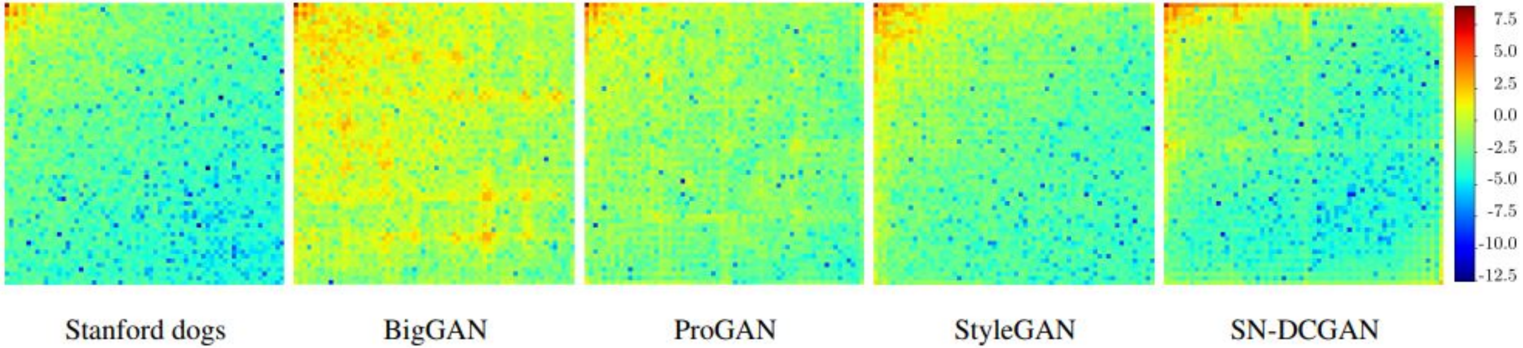


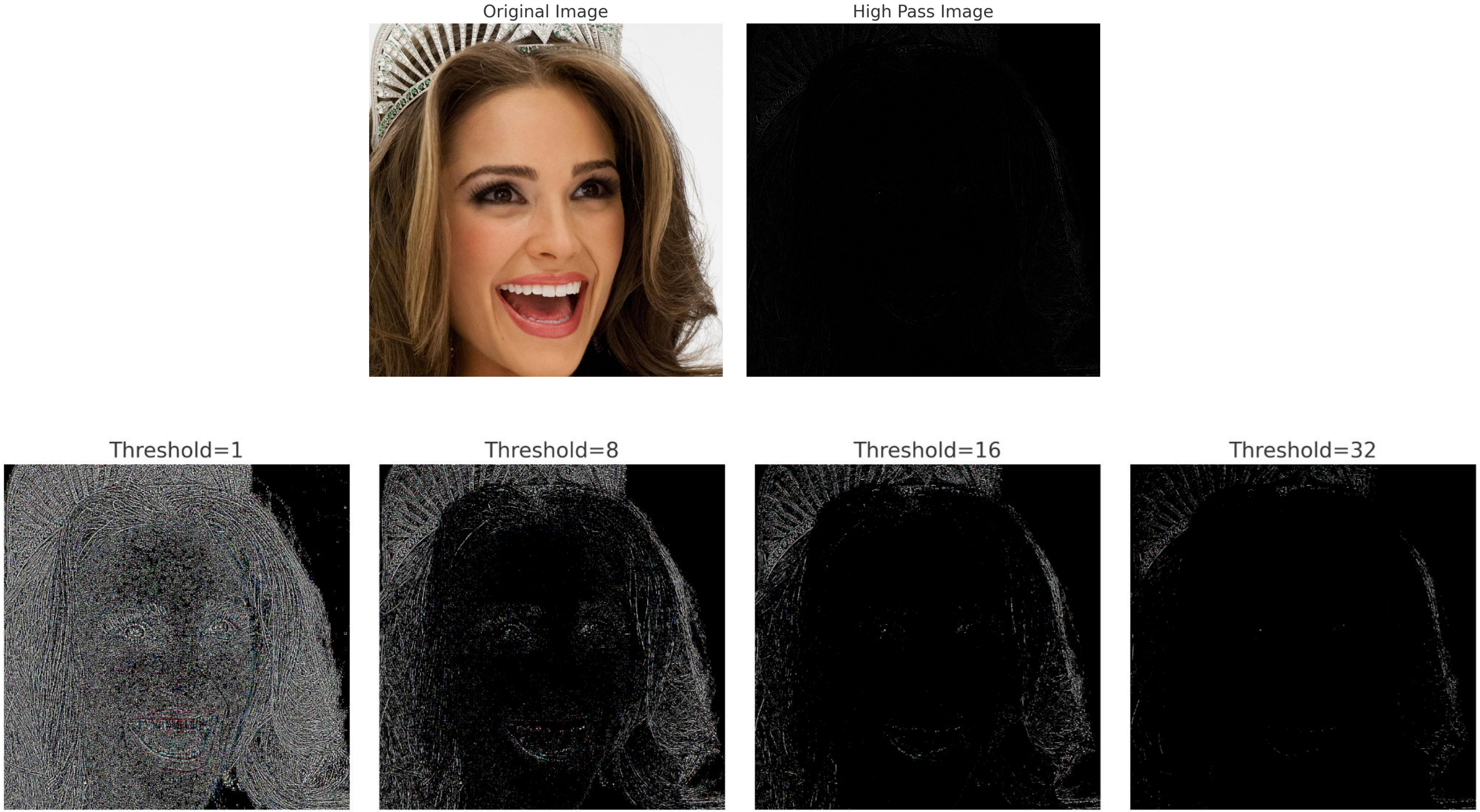




(a) GANs



*[17]: Frank et al. (2023)*




Original Image
High Pass Image
Threshold=1
Threshold=8
Threshold=16
Threshold=32