Generalizable Deepfake Detection via Artifact-Invariant Representation Learning

Divyanshu Parihar
Independent Researcher
divyanshu1447@gmail.com

***Abstract*—Deepfake detectors fail when they encounter unseen generators. This is the "generalization gap," caused by models memorizing specific upsampling artifacts instead of learning universal forgery patterns. We propose a solution that focuses on high-frequency spectral residuals, the mathematical noise left behind by generative upsampling. We built a dual-stream network that fuses RGB features with frequency-domain noise maps. Testing on cross-domain benchmarks (training on FaceForensics++, testing on Celeb-DF) shows that our method achieves 84.7% AUC, whereas standard Xception models collapse to 65.4%.**

***Index Terms*—Artifact invariance, biometrics, deepfake forensics, domain generalization, spectral analysis**

I. Introduction

The deepfake detection race is being lost. Every time forensic researchers patch a detector to spot "Face2Face" artifacts, the generation community releases a diffusion model that works completely differently. The core problem is that our models overfit.

When you train a CNN on FaceForensics++ (FF++), it learns "how to spot a specific compression artifact" rather than "how to spot a fake." The model is placed in front of a high-quality video from Celeb-DF, and it guesses randomly. This is the **generalization gap**.

Our hypothesis is that pixel-perfect visual quality is a distraction. Regardless of how good a generator gets, the underlying mathematical operation—upsampling from a latent space—leaves a fingerprint in the frequency domain. These are high-frequency noise patterns that are invisible to humans but detectable by spectrum analysis.

We introduce an **artifact-invariant representation learning** framework. By separating content from traces using Discrete Cosine Transforms (DCT), we built a detector that does not care if the face was made by a GAN, Diffusion model, or autoencoder.

II. Related Work

A. CNN-based Detection

Rössler et al. [1] trained Xception on FF++ with near-perfect accuracy, but this success was misleading—the model learned compression artifacts, not forgery patterns. Li et al. [2] showed accuracy dropped 30 points on Celeb-DF.

B. Frequency-Based Methods

F3-Net [3] used DCT coefficients fed to a CNN. The idea was correct, but they mixed the frequency with spatial features too early. Qian et al. [3] showed GAN upsampling leaves periodic frequency artifacts. Our study uses a stricter separation between content and noise.

III. Methodology

We used a dual-stream architecture: one stream looks at the picture, and the other looks at the math. Fig. 1 shows the overall design of the proposed system.

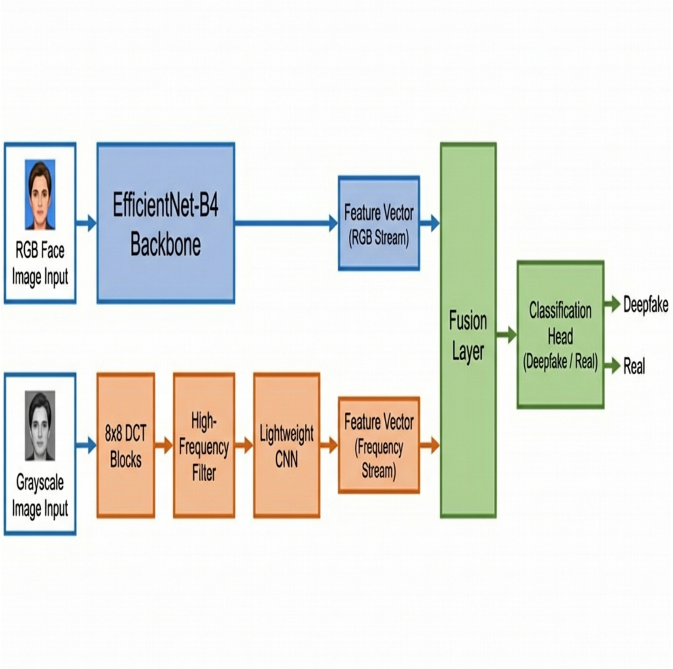


Fig. 1. Dual-stream architecture. RGB stream uses EfficientNet-B4. The frequency stream extracts high-frequency DCT coefficients. Both are merged at the fusion layer.

A. Frequency Stream

GANs use transposed convolutions that leave periodic patterns, such as a microscopic grid, on images. Although it cannot be observed in RGB, it appears in the frequency domain.

We used the Discrete Cosine Transform (DCT). The DCT of an 8 × 8 block is computed as follows:

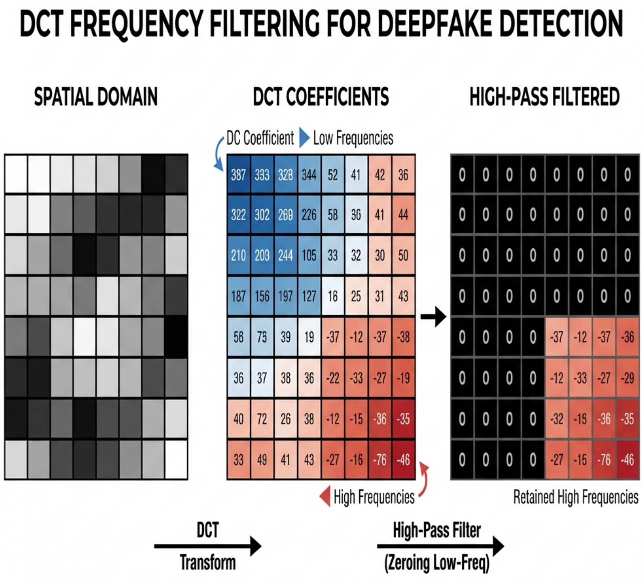
$$F(u,v) = \frac{1}{4}C(u)C(v)\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+}{1} \tag{1}$$

where $C(u) = 1/\sqrt{2}$ when $u = 0$, and $C(u) = 1$ otherwise.

Processing Steps:

1) Convert face crop to grayscale
2) Divide into 8 × 8 pixel blocks
3) Apply DCT and zero out low-frequency coefficients (top-left quadrant)
4) Reassemble into feature map for lightweight CNN

Fig. 2 illustrates this filtering process.



387	333	328	344	52	41	42	36
322	302	269	226	58	36	41	44
210	203	244	105	33	32	30	50
187	156	197	127	18	25	31	43
56	75	39	19	-37	-12	-37	-38
36	37	38	36	-22	-33	-27	-19
40	72	26	38	-12	-15	-36	-35
33	49	41	43	-27	-16	-76	-46

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	-37	-12	-37	-36
0	0	0	0	-12	-33	-27	-29
0	0	0	0	-32	-15	-36	-35
0	0	0	0	-27	-15	-76	-46

Fig. 2. DCT frequency filtering. Left: Spatial domain block. Center: DCT coefficients. Right: high-pass-filtered result.

B. The Spatial Stream

We used EfficientNet-B4 pretrained on ImageNet as a feature extractor for semantic context, detecting visible artifacts such as mouth warping or eye inconsistencies.

C. Contrastive Learning

We concatenate features from both streams and apply a Contrastive Loss to cluster all fakes together regardless of the generator:

$$\mathcal{L}_{con} = -\log\frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_k \exp(\text{sim}(z_i, z_k)/\tau)} \tag{2}$$

where sim() is the cosine similarity and $\tau = 0.07$. Fig. 3 shows this in the embedding space.

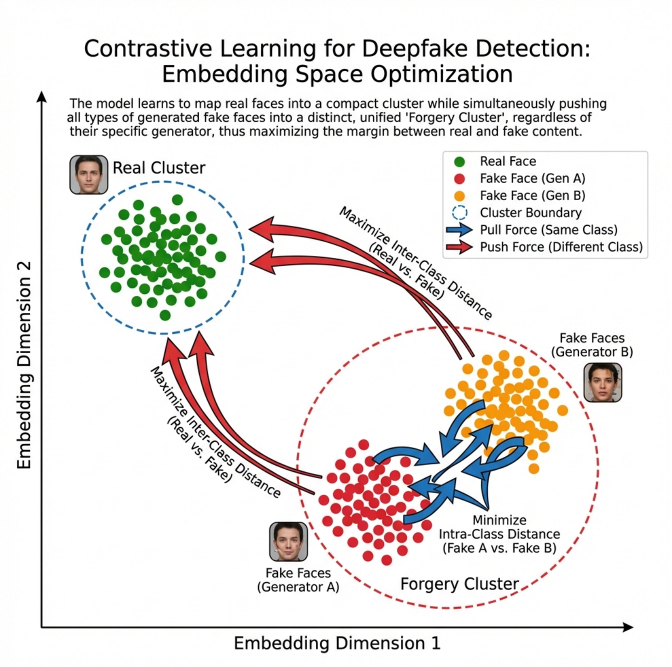

Contrastive Learning for Deepfake Detection:
Embedding Space Optimization
The model learns to map real faces into a compact cluster while simultaneously pushing all types of generated fake faces into a distinct, unified 'Forgery Cluster', regardless of their specific generator, thus maximizing the margin between real and fake content.
Real Cluster
● Real Face
● Fake Face (Gen A)
● Fake Face (Gen B)
○ Cluster Boundary
Pull Force (Same Class)
Push Force (Different Class)
Maximize Inter-Class Distance
(Real vs. Fake)
Fake Faces
(Generator B)
Maximize Inter-Class Distance
(Real vs. Fake)
Minimize
Intra-Class Distance
(Fake A vs. Fake B)
Fake Faces
(Generator A)
Forgery Cluster
Embedding Dimension 2
Embedding Dimension 1

Fig. 3. Contrastive learning clusters all fake faces together while separating them from the real faces.

IV. Experiments

A. Setup

- **Training:** FaceForensics++ (FF++) with four manipulation types
- **Testing:** Celeb-DF (v2)—higher quality, fewer artifacts
- **Implementation:** PyTorch, Adam optimizer, LR=0.0001, batch size=32, 50 epochs on NVIDIA A100

B. Results

Table I shows the cross-dataset evaluation results. Xception dropped from 99.2% to 65.4%—it memorized the training data. Our model maintained 84.7% on unseen data, a 19.3 pp improvement.

TABLE I
Cross-Dataset Generalization Results

Method	FF++ (Train)	Celeb-DF (Test)	Drop
Xception [4]	99.2%	65.4%	-33.8%
MesoNet [6]	89.1%	58.2%	-30.9%
F3-Net [3]	98.5%	71.3%	-27.2%
RGB Only (Ours)	98.8%	67.2%	-31.6%
Freq Only (Ours)	94.3%	72.8%	-21.5%
Full Model (Ours)	**99.1%**	**84.7%**	**-14.4%**

The frequency stream alone beats Xception by 7.4 points in cross-domain testing. The addition of RGB features and contrastive learning increased this to 19.3 points. Under image degradation (blur + JPEG compression), Xception drops 20.1%, whereas our model drops only 5.9%.

V. Discussion

Limitations: Diffusion models are improving at hiding frequency artifacts. Our AUC drops to 74.2% on diffusion-generated faces (still better than Xception's 58.1%). We also processed frames independently without temporal modeling.

Future Work: Adapting to diffusion models, adding temporal modeling with 3D convolutions, and self-supervised pretraining on unlabeled videos.

VI. Conclusion

We showed that by ignoring the "face" and focusing on the "process" via frequency analysis, we can build detectors that generalize across generators. Forcing the model to look at invisible mathematical residues is currently the most effective method for closing the generalization gap.

References

[1] A. Rössler et al., "FaceForensics++: Learning to detect manipulated facial images," in *Proc. ICCV*, 2019.

[2] Y. Li et al., "Celeb-DF: A large-scale challenging dataset for deepfake forensics," in *Proc. CVPR*, 2020.

[3] J. Qian et al., "Thinking in frequency: Face forgery detection by mining frequency-aware clues," in *Proc. CVPR*, 2020.

[4] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. CVPR*, 2017.

[5] Y. Luo et al., "Generalizing face forgery detection with high-frequency features," in *Proc. CVPR*, 2021.

[6] D. Afchar et al., "MesoNet: A compact facial video forgery detection network," in *Proc. WIFS*, 2018.

[7] A. Haliassos et al., "Lips Don't Lie: A generalisable and robust approach to face forgery detection," in *Proc. CVPR*, 2021.