

# Generalizable Deepfake Detection via Artifact-Invariant Representation Learning

Divyanshu Parihar  
Independent Researcher  
divyanshu1447@gmail.com

**Abstract**—Deepfake detectors fail when they meet 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 our method achieves 84.7% AUC where standard Xception models collapse to 65.4%.

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

## I. INTRODUCTION

We are losing the deepfake detection race. 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.” Put that model in front of high-quality video from Celeb-DF, and it guesses randomly. This is the **generalization gap**.

Our hypothesis: pixel-perfect visual quality is a distraction. No matter 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 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 build 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

Rossler 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 right, but they mixed frequency with spatial features too early. Qian et al. [3] showed GAN upsampling leaves periodic frequency artifacts. Our work uses stricter separation between content and noise.

## III. METHODOLOGY

We use a dual-stream architecture: one stream looks at the picture; the other looks at the math. Fig. 1 shows the overall design.

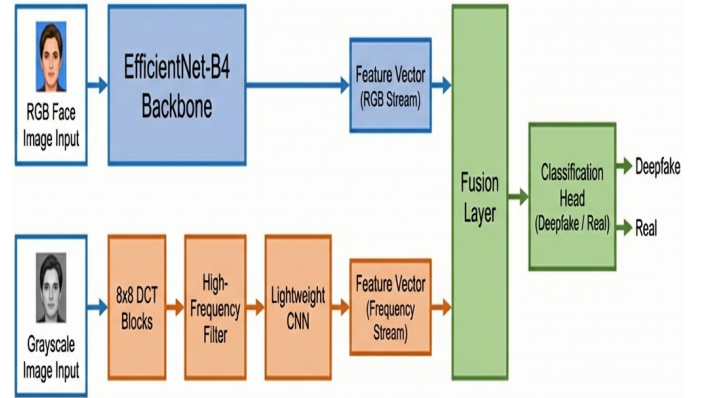


Fig. 1. Dual-stream architecture. RGB stream uses EfficientNet-B4. Frequency stream extracts high-frequency DCT coefficients. Both merge at a fusion layer.

### A. The Frequency Stream

GANs use transposed convolutions that leave periodic patterns—like a microscopic grid—on images. You cannot see it in RGB, but it appears in the frequency domain.

We use the **Discrete Cosine Transform (DCT)**. The DCT of an  $8 \times 8$  block is computed as:

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

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  $8 \times 8$  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.

### DCT FREQUENCY FILTERING FOR DEEFAKE DETECTION

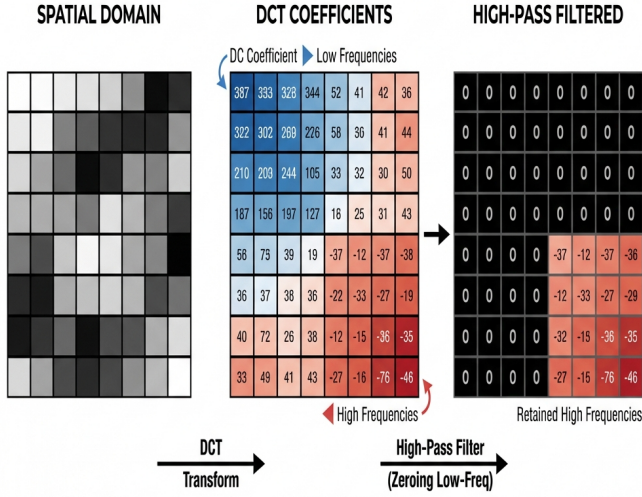


Fig. 2. DCT frequency filtering. Left: spatial domain block. Center: DCT coefficients. Right: high-pass filtered result.

#### B. The Spatial Stream

We use **EfficientNet-B4** pretrained on ImageNet as a feature extractor for semantic context—detecting visible artifacts like 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 generator:

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

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

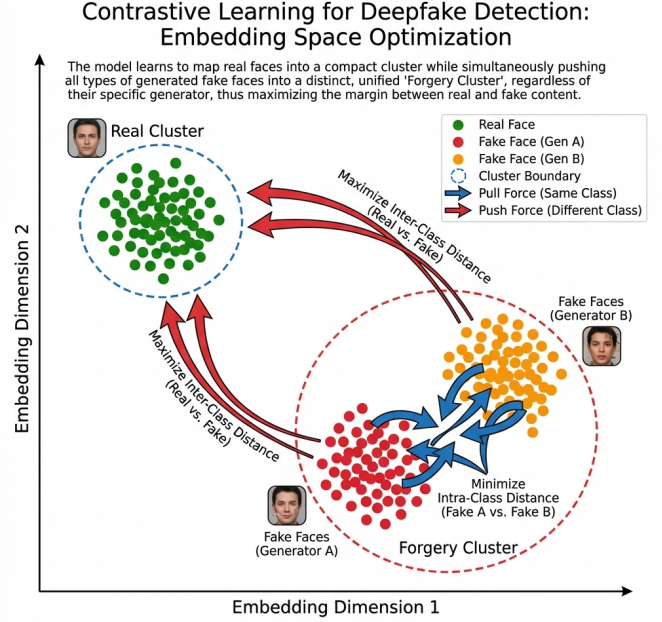


Fig. 3. Contrastive learning clusters all fakes together while separating them from real faces.

### 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 cross-dataset evaluation results. Xception drops from 99.2% to 65.4%—it memorized the training data. Our model maintains 84.7% on unseen data, a 19.3 percentage point 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%
<b>Full Model (Ours)</b>	<b>99.1%</b>	<b>84.7%</b>	<b>-14.4%</b>

The frequency stream alone beats Xception by 7.4 points on cross-domain testing. Adding RGB features and contrastive learning pushes this to 19.3 points. Under image degradation (blur + JPEG compression), Xception drops 20.1% while ours 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 process frames independently without temporal modeling.

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

## 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 way to close the generalization gap.

## REFERENCES

- [1] A. Rossler 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.