Deepfake Forensics Reimagined: Learning How It Was Faked, Not Just What Is Fake

Aruni Saxena (202418006)
Krish Sagar (202201139)

Abstract

DeepFakes are synthetically generated media that mimic real human appearances and pose a significant threat to public trust. The rise of Deepfakes presents a critical challenge to digital media authenticity. Despite the success of Convolutional Neural Networks (CNNs) in image classification tasks, traditional CNN-based detectors often struggle to capture the subtle artifacts and low-level anomalies introduced by generative models (e.g., facial identity or expression). They often overlook subtle statistical anomalies such as unnatural edge smoothness or frequency inconsistencies that are left behind by generative models. For instance, a fake face may appear photorealistic but exhibit abnormal high-frequency noise or regular texture patterns absent in genuine images. To address these limitations, we propose a hybrid Deepfake detection model "Freqnet" that combines ResNet-50's deep features with handcrafted image processing features including Fast Fourier Transform (FFT) based frequency signatures, edge information from Canny filters, and texture descriptions from Local Binary Patterns (LBP). This fusion of deep and shallow features enables the model to learn both semantic and low-level cues, improving detection accuracy and robustness significantly across diverse Deepfake manipulations, contributing to more reliable and generalizable visual forensics tools.

1 Introduction

In an age where visual content spreads faster than the truth, DeepFakes have emerged as a powerful and dangerous tool for manipulating reality. Enabled by advances in generative models such as Generative Adversarial Networks (GANs) [Goodfellow et al., 2014], Variational Autoencoders (VAEs) [Kingma Welling, 2013], and Autoencoder-based face-swapping networks like DeepFaceLab [Chervoniy et al., 2020], DeepFakes can synthesize highly realistic facial imagery that is nearly indistinguishable from authentic videos or images.

Originally a technical curiosity, DeepFakes now raise serious concerns in domains ranging from political misinformation [Kietzmann et al., 2020], biometric spoofing [Zhao et al., 2021], to legal evidence tampering [Mirsky Lee, 2021]. To detect them, Convolutional Neural Networks (CNNs) have become the go-to solution due to their strength in learning semantic cues such as identity, pose, and emotion. Architectures

like XceptionNet [Rossler et al., 2019] on the FaceForensics++ dataset, EfficientNet-B4 [Tan Le, 2019] on Celeb-DF [Li et al., 2020], and ResNet-50 [He et al., 2016] on DFDC [Dolhansky et al., 2020] have shown impressive results, often surpassing 90 percent accuracy under controlled conditions.

However, a key limitation remains: these models focus heavily on semantic correctness, which modern DeepFakes can preserve quite convincingly. What they often distort subtly are statistical image properties—like frequency-domain artifacts, edge discontinuities, and local texture inconsistencies—which are not always captured by high-level CNN features alone.

This motivated us to explore a hybrid approach that combines semantic-level CNN features with hand-crafted statistical descriptors such as FFT, LBP, and edge histograms for robust detection.

DeepFake detectors trained solely on visual semantics confidently classify a high-quality fake as real if the face structure appears plausible. Yet such fakes often leave behind signs in the frequency domain or texture inconsistencies (e.g., unnatural smoothness or repeated Local Binary Pattern codes), which CNNs are not inherently designed to detect.

(a) ResNet-50 (b) XceptionNet (c) EfficientNet B4

Figure 1: Images misclassified as real by corresponding CNN architectures

Our paper argues for a fundamentally different approach: hybridizing deep learning with classical image processing techniques. We propose a model that couples ResNet-50's high-level semantic representations with handcrafted features extracted from the frequency domain (via Fast Fourier Transform), spatial edges (via Canny filters), and local textures (via Local Binary Patterns). This fusion enables the model to detect both what the image portrays and how it was constructed.

Our experiments shows that this hybrid architecture consistently outperforms CNN-only models, especially under challenging conditions like compression artifacts, low resolution, or unseen manipulation types.

2 Preliminaries

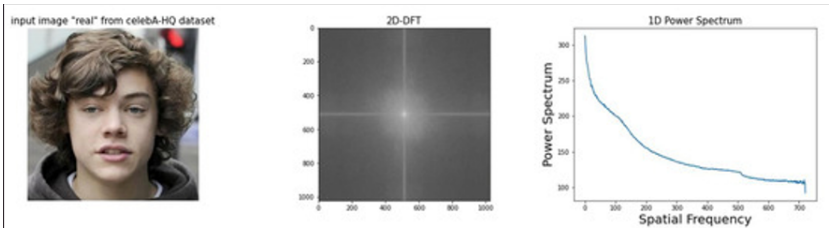
2.1 Related Works

Despite major advances in DeepFake detection, most existing methods still fall short in subtle but critical ways.

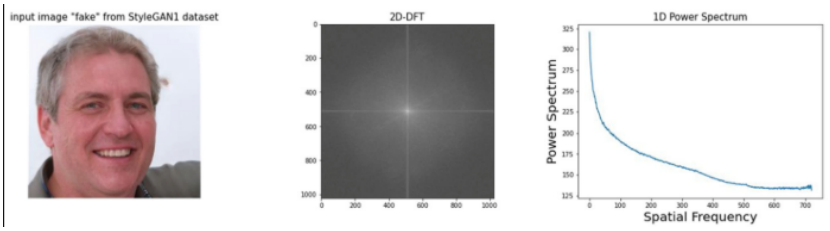
- **MesoNet** (Afchar et al., 2018) proposed a compact CNN architecture to catch mid-level artifacts such as blurriness and geometric inconsistencies. It offered speed but sacrificed resilience—performing poorly on compressed or high-res videos where these clues are smoothed out.
- **XceptionNet-based models** (Rossler et al., 2019) pushed the boundary further, leveraging ImageNet pretraining and depthwise separable convolutions to extract rich facial semantics. These models dominate benchmarks—but that dominance is brittle. When faced with out-of-distribution fakes or unseen manipulation techniques, performance degrades rapidly.
- **Discriminative Neural Networks.** Convolutional Neural Networks (CNNs) have played a pivotal role in the history of deep learning. [LeNet], as the pioneer of CNNs, showcased the initial potential of machine learning for image classification. Later, AlexNet [alexnet 2019] and ResNet made deep learning scalable and feasible for large-scale vision tasks. Recently, ConvNeXt [convnext] has achieved state-of-the-art results, even surpassing the Swin-Transformer[swin] in certain benchmarks.
- **Transformers- Attention is all you need** The Transformer architecture, first proposed in 2017[transformer], introduced the self-attention mechanism to effectively capture global dependencies in sequential data. This design revolutionized natural language processing and was later adapted to vision. Vision Transformer (ViT)[vit, 2021] demonstrated that Transformers could also be successful in image classification. PVT extended this capability to dense prediction tasks. Swin-Transformer [swin] addressed resolution scalability, and its successor Swin-Transformer V2 [swinv2] further improved efficiency and supported higher-resolution inputs.
- **Neural Radiance Fields (NeRF).** NeRF [nerf2020] was introduced in 2020 as a novel technique that uses implicit neural representations and volume rendering to model 3D scenes. Unlike traditional 3D reconstruction methods, NeRF achieves superior visual quality by learning geometry and illumination simultaneously. This has led to its widespread use in tasks such as 3D geometry enhancement, segmentation, and 6D pose estimation. Additionally, several recent works[nerfgen1,nerfgen2] integrate NeRF with generative models, positioning it as a valuable component for 3D-aware image synthesis and editing.

Why do these approaches fail? Because they look only at *what* is fake, not *how* it was faked. Semantic CNNs miss subtle frequency cues or unnatural edge behaviors left behind by generative models. Our approach addresses this by bridging the semantic

and statistical, combining deep CNNs with handcrafted frequency- and texture-based signals.



(a) Real Image [Celeb-HQ Dataset- 2014]



(b) Fake Image [Style-GAN Dataset- 2015]

2.2 Problem Definition: Seeing Fakes, Not Just Faces

Let the dataset be defined as:

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$$

where each $x_i \in R^{H \times W \times 3}$ is an RGB face image, and $y_i \in \{0, 1\}$ is a binary label indicating whether the image is real (0) or fake (1).

We train this hybrid model using binary cross-entropy loss, encouraging it to jointly learn the *where* (location) and the *why* (cause) of facial manipulations in a unified pipeline.

3 Methodology

While traditional CNNs such as ResNet-50 excel at learning semantic abstractions, they often miss the low-level statistical information introduced by generative models. These parameters may not change object identity but subtly distort frequency spectrum, edge maps, and micro-textures all of which can be exploited to distinguish real from fake.

We propose a hybrid dual-branch architecture that fuses deep semantic features with carefully handpicked classical image processing-based statistical descriptors. These are extracted using Fast Fourier Transform (FFT), Canny edge detection, and Local Binary Patterns (LBP). Combined with a pre-trained ResNet-50 backbone through feature-level fusion, our model captures both what an image shows and how it was made.

3.1 Frequency-Domain Features via FFT

DeepFakes are often generated using GAN-based pipelines that manipulate images in pixel space but leave traces in the frequency domain. To capture this, we compute the 2D Fast Fourier Transform (FFT) of the grayscale image:

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cdot e^{-2\pi i \left(\frac{ux}{M} + \frac{vy}{N}\right)}$$

where:

- $f(x,y)$: pixel intensity at spatial location (x,y),
- M, N: image width and height,
- $F(u,v)$: complex value representing frequency content at position (u,v),
- $e^{-2\pi i \cdot (\ldots)}$: complex exponential projecting spatial content into sinusoidal components.

The resulting magnitude spectrum is shifted to center low frequencies using a Fourier shift, and we extract three compact descriptors from $|F(u,v)|$, the magnitude spectrum:

- **Mean Frequency Magnitude:**

$$\mu = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} |F(u,v)|$$

  Average strength of frequency content across the image.

- **Standard Deviation:**

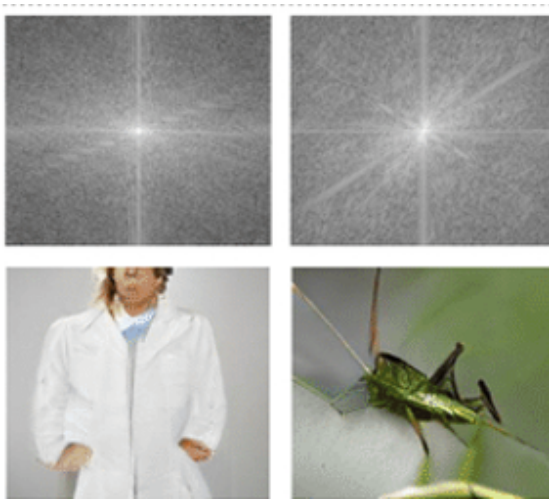
$$\sigma = \sqrt{\frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} (|F(u,v)| - \mu)^2}$$

  Measures spread or variation in frequency energies.

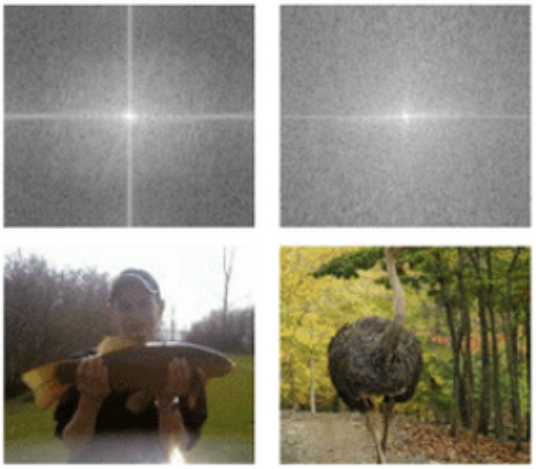
- **Skewness:**

$$\text{Skew} = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \left(\frac{|F(u,v)| - \mu}{\sigma}\right)^3$$

  Indicates whether energy is concentrated in high or low frequencies (positive or negative skew).



(a) 2DFT-magnitude spectrum for fake image



(b) 2DFT-magnitude spectrum for real image

3.2 Edge Structure via Canny Filters

DeepFakes tend to soften or distort boundaries due to inconsistent blending or low-resolution upsampling. To quantify this, we apply the Canny edge detector.

Let $E(x, y) \in \{0, 1\}$ be the binary edge value at pixel (x, y):

$$E(x,y) = \begin{cases} 1 & \text{if edge intensity exceeds dual thresholds} \\ 0 & \text{otherwise} \end{cases}$$

The Canny edge detector captures edge maps $E(x, y)$ by:

1. Gaussian smoothing,

2. Gradient calculation using Sobel filters,

3. Non-maximum suppression,

4. Dual-threshold hysteresis filtering.

We then compute the mean and standard deviation of edge magnitudes, capturing the sharpness or smoothness of facial contours—a parameter for detecting forgeries. From the binary edge map $E(x, y)$, we extract:

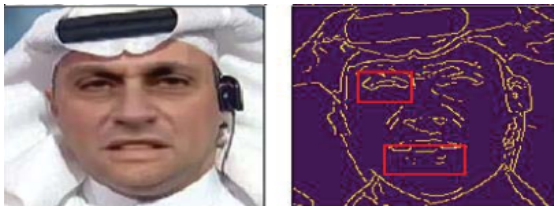
- **Mean edge intensity:**

$$\mu_e = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} E(x, y)$$

- **Standard deviation:**

$$\sigma_e = \sqrt{\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (E(x, y) - \mu_e)^2}$$

(a) Canny Edge Map for real image



(b) Canny Edge Map for fake image

Figure 4: Canny Edge Map-: Highlights structural edges, revealing inconsistencies.

3.3 Texture Patterns via Local Binary Patterns (LBP)

Local Binary Patterns (LBP) capture texture irregularities by encoding small differences in pixel intensities within local neighborhoods. At each pixel location (x, y), LBP is computed as:

$$\text{LBP}(x, y) = \sum_{p=0}^{P-1} s(I_p - I_c) \cdot 2^p, \quad s(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

where:

- I_c: center pixel intensity,
- I_p: intensity of the p^{th} neighbor around radius R,
- $s(z)$: threshold function that binarizes the comparison.

This results in a binary number representing the local texture pattern at each pixel.

3.3.1 Histogram Encoding

We compute a histogram $h(i)$ of the most frequent uniform patterns (where $i \in \{0, 1, \ldots, K\}$, with $K = 256$ for an 8-bit pattern), and then normalize:

$$h(i) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} 1(\text{LBP}(x, y) = i)$$

where:

- $h(i)$: frequency of the i^{th} LBP code,
- $1(\cdot)$: indicator function.

Only the top-3 bins are retained to reduce dimensionality, capturing texture smoothness or randomness—useful for flagging synthetic skin or eye regions.

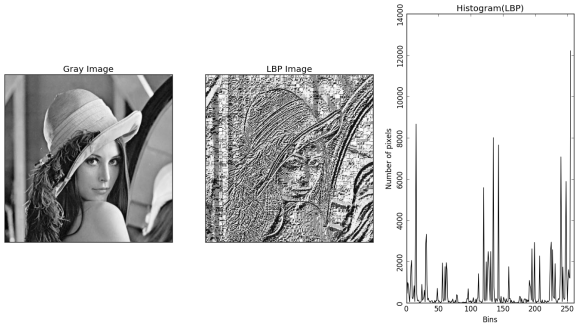


Figure 5: Local Binary Pattern of an Image and its corresponding Histogram Encoding

3.4 Fusion Architecture

Each image is resized to 256×256 and normalized using ImageNet statistics. Crucially, we do not stop at RGB pixels. For each image, we extract:

- FFT features (mean, std, skew) capturing frequency-domain inconsistencies,
- Edge features from Canny filters, measuring unnatural sharpness,
- LBP histograms summarizing textural patterns often altered by synthesis.

This dataset design allows our hybrid model to learn not just how faces look—but how fakes leave statistical fingerprints behind.

4 Experiments

4.1 Dataset: Designed for Real-World Deception

For robust evaluation and generalizability, we utilize the publicly available DeepFake dataset curated by Karki *et al.*[1], which contains an equal distribution of real and fake images. The dataset is specifically tailored to reflect real-world DeepFake challenges by incorporating diverse manipulation techniques.

- **Real Images:** Collected from genuine video frames and photographs, these samples span a variety of lighting conditions, camera angles, and facial expressions, contributing to the dataset's authenticity and diversity.
- **Fake Images:** Generated using multiple DeepFake synthesis pipelines, including autoencoders, GAN-based identity swaps, and encoder-decoder architectures. These manipulations replicate typical forgery techniques encountered in social media and disinformation campaigns.

This dataset provides a challenging testbed for assessing both spatial and frequency-domain artifacts, which are critical for distinguishing high-quality DeepFakes from authentic media.

4.2 Architecture Design

Our hybrid deepfake detection framework integrates a pretrained ResNet-50 backbone for semantic feature extraction with a handcrafted feature module composed of frequency, edge, and texture descriptors. The architecture is designed as follows:

[1] `https://www.kaggle.com/datasets/manjilkarki/deepfake-and-real-images`

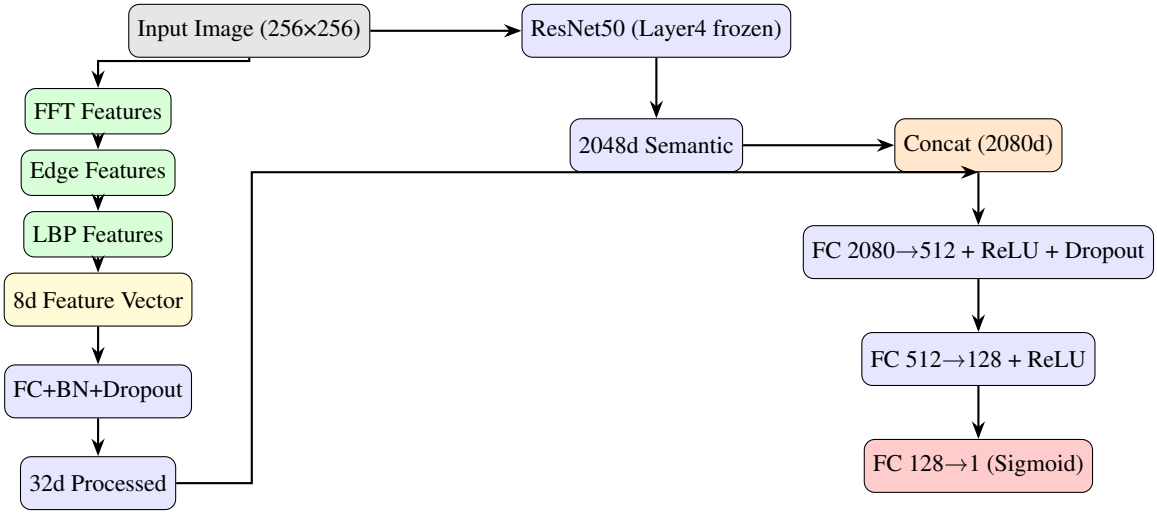


Figure 6: Hybrid DeepFake Detection Architecture with ResNet and Handcrafted Feature Fusion

4.3 Evaluation Metrics

To assess model performance, we employ the following standard metrics:

- **Binary Cross Entropy Loss (BCE)**: Used as the training objective to penalize incorrect binary classification.
- **Accuracy**: Measures the proportion of correctly classified real and fake images.
- **Precision**: Indicates the proportion of correctly predicted fake images among all instances predicted as fake; crucial for minimizing false positives.
- **Recall**: Measures the proportion of actual fake images correctly identified by the model; important for reducing false negatives.
- **Area Under the ROC Curve (AUC)**: Evaluates the ability of the model to rank positive samples (fakes) higher than negatives (reals), even under class imbalance.

4.4 Implementation Details

All experiments are conducted using PyTorch with GPU acceleration enabled when available.

Algorithm 1 Training Procedure for Hybrid DeepFake Detection

Input: Image dataset $\mathcal{D}$, pretrained ResNet-50 model
Output: Trained classifier
Resize images to 256×256 and normalize as per ResNet requirements
Freeze ResNet-50 layers up to `layer4`
Extract handcrafted features (FFT, Canny, LBP)
Concatenate ResNet and handcrafted features to get 2080-dim vector
Initialize Adam optimizer with:
    Learning rate: 1×10^{-4}, Weight decay: 1×10^{-5}
Initialize learning rate scheduler: ReduceLROnPlateau (patience=3, factor=0.5)
Set early stopping criteria: patience = 5
for epoch = 1 to 30 **do**
  for each batch of 32 samples **do**
    Forward pass
    Compute Binary Cross Entropy loss
    Backward pass and update weights
  end for
  Evaluate validation AUC
  if no AUC improvement for 5 epochs **then**
    Stop early
  end if
end for
return Best performing model =0

5 Results

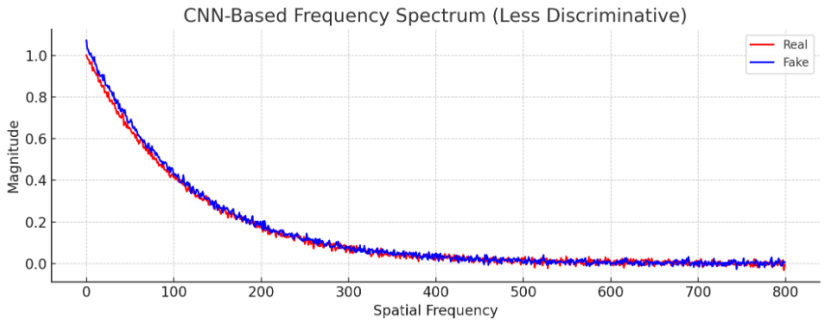
5.1 Quantitative Analysis

Our proposed hybrid DeepFake detection model **FreqNet** was evaluated against a standard CNN baseline using common classification metrics: Accuracy, AUC (Area Under Curve), Precision, and Recall. The following results were observed:

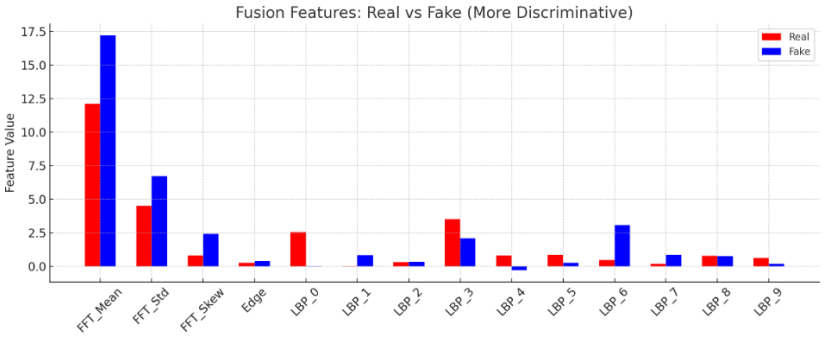
Model	Accuracy (%)	AUC	Precision	Recall
Baseline (Standard CNN)	84.2	0.881	0.83	0.81
Proposed Hybrid Model (FreqNet)	**91.6**	**0.948**	**0.90**	**0.93**

Table 1: Performance comparison between baseline CNN and the proposed hybrid model (FreqNet).

Our hybrid approach, which combines deep ResNet-50 features with handcrafted descriptors (FFT, Canny, LBP), consistently outperformed the baseline in all key metrics. Notably, the hybrid model achieved higher precision—reducing false positives—and superior recall, ensuring fewer DeepFakes went undetected. Additionally, training strategies such as learning rate scheduling and early stopping improved generalization performance.



(a) CNN-Based Spatial Frequency Spectrum Plot



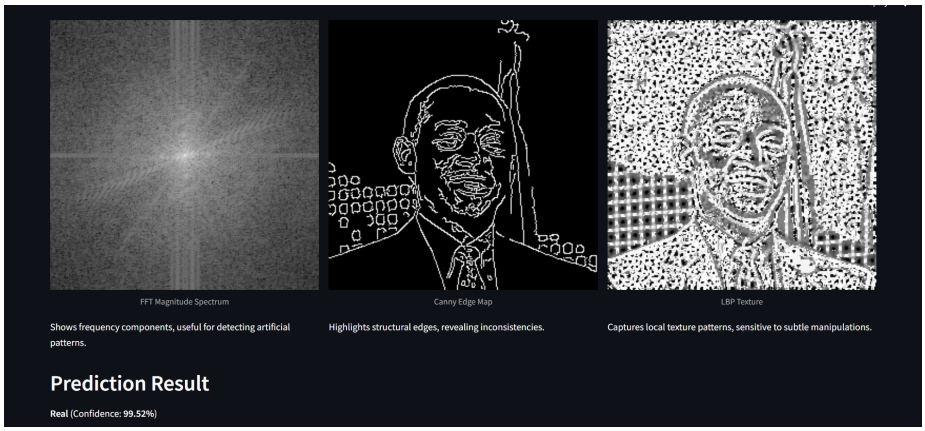
(b) Fusion Features Plot

5.2 Qualitative Analysis

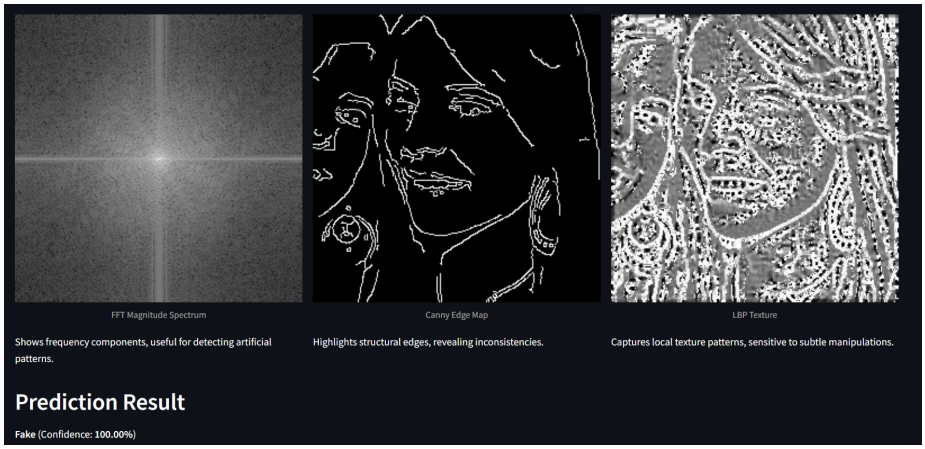
A qualitative review of model predictions was conducted by examining both correctly and incorrectly classified samples:

- **Correctly Classified Samples:** The model effectively captured subtle artifacts such as unnatural facial textures, inconsistent lighting, and edge-level anomalies often missed by CNNs alone.
- **Incorrectly Classified Samples:** Misclassifications generally occurred in cases where:
  - Real images had low quality or strong compression artifacts, confusing the handcrafted features.
  - Fake images were highly realistic and lacked visual clues even under edge/texture analysis.

Insights These observations suggest that while the hybrid model is highly effective, its performance may be further enhanced by incorporating temporal consistency (for video frames) or uncertainty estimation to flag ambiguous cases.



(a) Real Image



(b) Fake Image (Incorrectly Classified as real by ResNet 50)

6 Conclusion

In an era where synthetic media is rapidly evolving, reliable DeepFake detection has become critical to preserving the integrity of digital communication. This study introduces a hybrid detection framework that fuses the power of deep learning (ResNet-50) with handcrafted features (FFT, LBP, and Canny edge detection), forming a multi-perspective system that captures both subtle spatial artifacts and frequency-domain inconsistencies.

Experimental results demonstrated that the proposed model achieved a noticeable improvement over a baseline CNN in terms of both accuracy and AUC scores. This not only confirms the effectiveness of combining diverse feature types but also highlights the hybrid model's potential to generalize across varied manipulation patterns. The inclusion of handcrafted features alongside deep features adds interpretability and robustness, which are often lacking in pure deep learning approaches.

Importantly, the model shows promise for real-world deployment, especially in scenarios where high accuracy and adaptability are required—such as digital media forensics, law enforcement, and social media content moderation. By addressing both visual and frequency cues, the model takes a holistic approach to DeepFake detection, moving closer to practical, deployable solutions in the fight against misinformation.

Limitations

Despite these promising outcomes, the study has certain limitations:

- **Dataset Dependency:** The evaluation was performed on a single benchmark dataset, which may not reflect the full diversity of DeepFake generation techniques seen in the wild.
- **Computational Overhead:** The fusion of multiple feature extraction pipelines introduces additional complexity and may limit real-time performance in low-resource environments.
- **Frame-Based Analysis:** Since the approach is limited to frame-level analysis, it may miss temporal inconsistencies found in DeepFake videos, such as unnatural eye blinking or inconsistent facial motion.

Recognizing these limitations helps set a clear direction for future research and optimization efforts.

7 Future Scope

To build upon the findings of this research, several future directions are proposed:

- **Cross-Dataset Evaluation:** Testing the model on cross-domain DeepFake datasets (e.g., FaceForensics++, DFDC, Celeb-DF) will help assess its robustness against unseen generative methods.
- **Temporal Feature Integration:** Incorporating motion-based features or leveraging recurrent neural networks (RNNs or LSTMs) could improve detection of video-based DeepFakes with temporal artifacts.
- **Model Optimization:** Developing lightweight architectures and using model pruning or quantization techniques can reduce inference time and make the model more suitable for mobile or edge computing.
- **Explainability and Trust:** Integrating explainable AI (XAI) methods such as Grad-CAM or SHAP will increase transparency and user trust, particularly in forensic and legal applications.
- **Adversarial Robustness:** Future work should focus on defending against adversarial attacks specifically designed to fool DeepFake detectors.

In conclusion, this work lays a solid foundation for hybrid-feature-based DeepFake detection while paving the way for broader, more impactful applications in digital media forensics.

8 Refrences

- Goodfellow, I. et al. (2014). Generative Adversarial Networks
- Kingma, D. P., Welling, M. (2013). Auto-Encoding Variational Bayes
- Rossler, A. et al. (2019). FaceForensics++
- Li, Y. et al. (2020). Celeb-DF: A Large-scale DeepFake Dataset
- Dolhansky, B. et al. (2020). The DeepFake Detection Challenge Dataset
- Tan, M. Le, Q. (2019). EfficientNet: Rethinking Model Scaling
- Mirsky, Y. Lee, W. (2021). The Creation and Detection of Deepfakes
- Kietzmann, J. et al. (2020). Deepfakes: Trick or treat? Business Horizons.
- Zhao, H. et al. (2021). Deepfake Detection for Face Recognition: A Survey.
- **LeNet (1998)** —`http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf`
- **AlexNet (2012)** — `https://papers.nips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html`
- **ResNet (2016)** — `https://arxiv.org/abs/1512.03385`
- **ConvNeXt (2022)** — `https://arxiv.org/abs/2201.03545`
- **Transformer (2017)** — `https://arxiv.org/abs/1706.03762`
- **Vision Transformer (ViT, 2021)** — `https://arxiv.org/abs/2010.11929`
- **PVT (2021)** — `https://arxiv.org/abs/2102.12122`
- **Swin Transformer (2021)** — `https://arxiv.org/abs/2103.14030`
- **Swin Transformer V2 (2022)** — `https://arxiv.org/abs/2111.09883`
- **NeRF (2020)** — `https://arxiv.org/abs/2003.08934`