Hybrid Forensic-Neural Fusion for Deepfake Image Detection Using Multi-Signal Analysis

Dr. B. Suribabu[1], Tamanampudi Sai Jahnavi[2], Korukonda Gamya Sri[3], Gollakoti Jyothi[4], Saladi Lakshmi Kantham[5], Pappula Krishna Veni[6], Dr. Y. Venkat[7]

[1]*Associate Professor, Department of Artificial Intelligence and Machine Learning, Srinivasa Institute of Engineering and Technology*

[2,3,4,5,6]*UG Scholar, Department of Artificial Intelligence and Machine Learning, Srinivasa Institute of Engineering and Technology*

[7]*Professor, Department of Artificial Intelligence and Machine Learning, Srinivasa Institute of Engineering and Technology*

***Abstract*—Fabricated facial images have emerged as a significant concern across journalism, legal proceedings, and digital identity verification. While existing detection systems perform effectively under controlled conditions, they often fail when evaluated on images generated by previously unseen manipulation techniques. This limitation reduces their reliability in real-world scenarios, where the source and method of manipulation are typically unknown.**

This paper proposes a detection approach that integrates two complementary analysis paradigms that have traditionally been studied independently: deep neural inference and classical image forensics. Three forensic signals are extracted from each input, namely compression error patterns across multiple quality levels, discontinuities at JPEG block boundaries, and inconsistencies in noise variance. These features are combined with the output of a pretrained convolutional neural network using a weighted score fusion strategy. In addition, a conservative correction mechanism is introduced to reduce false positives, particularly in genuine images.

Experimental evaluation on the Face Forensics++ and Celeb-DF v2 datasets achieves an accuracy of 91.3% and an AUC-ROC of 0.957. When tested on previously unseen data, the proposed model exhibits an accuracy drop of only 7.7 percentage points, which is approximately half the degradation observed in comparable single-model approaches. The entire pipeline operates efficiently on a standard CPU without requiring GPU acceleration and is implemented as both a web-based application and a real-time webcam detection system.

***Index Terms*—Deepfake Detection, Image Forensics, Error Level Analysis, Score Fusion, Convolutional Neural Networks, Face Forensics++, Celeb-DF, JPEG Artifact Analysis, Digital Media Integrity**

I. INTRODUCTION

A few years ago, spotting a fabricated face in a photograph or video was relatively straightforward. The telltale signs were visible to anyone paying attention: edges that did not quite match, lighting that fell at the wrong angle, unnatural movement around the eyes or mouth. That is no longer true. The generation tools available today produce synthetic faces that are, in many cases, visually indistinguishable from genuine photographs, and the gap between what can be created and what can be reliably detected has grown wider with each passing year.

The consequences of this gap are not theoretical. Fabricated images have been used in political disinformation campaigns, in the production of non-consensual intimate imagery, and in attempts to manipulate legal evidence [1]. The problem has drawn attention from technology researchers, policymakers, and platform operators alike, and yet a practical, deployable solution that works reliably outside of benchmark conditions remains elusive. The difficulty lies in the nature of the detection challenge itself. Neural network-based detectors, which have dominated published benchmarks over the past several years, learn to recognize the specific patterns left

behind by a particular generation method. When that method changes or when the image has been compressed, resized, or processed after creation, those patterns are altered or erased, and the detector's performance degrades sharply [2]. Classical forensic methods, which analyze properties of the image itself rather than the output of a specific generator, are more robust to these changes, but struggle to match the accuracy of trained models on well-controlled datasets.

This paper describes a detection framework built on the observation that these two approaches fail in different circumstances. By combining them at the output level, merging their scores after independent inference rather than redesigning either from scratch, it is possible to preserve the discriminative power of the neural component while adding the cross-domain stability of the forensic signals. No retraining is required when switching to a different neural model, which means the framework can be updated as better models become available without rebuilding the forensic layer.

II. LITERATURE SURVEY

Research on deepfake detection has grown substantially since the release of the FaceForensics++ benchmark [2], which gave the community a shared standard for comparing methods. Despite considerable progress, the central challenge - building a detector that transfers well to unseen generation methods and image conditions - has not been fully solved. The approaches developed so far fall into three broad categories.

Neural Network-Based Detectors

The most widely cited approaches in this category use deep convolutional networks trained end-to-end on labelled datasets. XceptionNet [2] established an early baseline with near-perfect accuracy on high-quality images, but its performance dropped substantially under compression, a well-documented sensitivity that limits its usefulness for images distributed through social media platforms. FaceXRay [3] shifted the learning target from generation artifacts to blending boundaries, which improved cross-dataset results but left the model blind to fakes that did not involve visible blending. Zhao et al. [4] incorporated a multi-attentional mechanism to focus learning on locally inconsistent regions, achieving 88.7% on FaceForensics++ at the cost of considerably higher compute requirements. Vision Transformer architectures [5] have pushed benchmark accuracy above 90%, but require extensive labeled data and hardware resources that are not available in most practical settings.

Forensic Signal-Based Approaches

Classical forensic methods analyze properties of the image that are independent of how the manipulation was performed. Error Level Analysis [6] identifies regions with anomalous compression histories, a useful signal because manipulated areas typically carry different JPEG compression patterns than the unaltered portions of the image. Laplacian-based noise analysis [7] targets the high-frequency residuals introduced by camera sensor noise, which synthetic faces tend to lack or render inconsistently. Block DCT artifact analysis examines the discontinuity pattern at JPEG 8×8 grid boundaries, which GAN-generated images frequently disrupt. These methods are attractive because they are not tuned to any specific generator, but each individually achieves only moderate accuracy on challenging benchmarks, typically in the range of 70-78%.

Hybrid and Fusion Approaches

The potential of combining neural and forensic approaches has been noted in the literature for some time. Verdoliva [8] provides an overview of media forensics and observes that multi-modal systems consistently outperform single-signal detectors. The difficulty with most published hybrid systems is that they require all components to be trained jointly, which means the system must be rebuilt from scratch whenever a better neural backbone becomes available. The approach proposed here avoids this limitation by fusing scores at the output stage, leaving each component architecturally independent. Any model that produces a scalar fake probability can be substituted into the neural slot without modifying the forensic layer.

Table 1: Comparison with existing deepfake detection methods

Method	Dataset	Accuracy	AUC	Key Limitation
XceptionNet [2]	FF++ c40	81.0%	0.892	Compression sensitive
FaceXRay [3]	Celeb-DF	79.5%	0.874	Blending artifacts only
Multi-Attn CNN [4]	FF++	88.7%	0.931	High compute requirement
ViT-B/16 [5]	FF++	90.1%	0.948	Dataset-specific tuning
ELA Only [6]	Mixed	71.3%	0.762	Low standalone accuracy
Proposed HFNF	FF++ + Celeb-DF	91.3%	0.957	Lighting-sensitive heatmap

III. METHODOLOGY

System Overview

The proposed Hybrid Forensic-Neural Fusion (HFNF) pipeline takes a single image as input and produces a fake probability score along with a binary label - Real or Fake. Processing moves through five stages in sequence: face detection, neural inference, forensic signal extraction, score fusion, and adaptive thresholding. Because the neural backend is treated as a replaceable module, the system does not need to be retrained when a better model becomes available.

Image Preprocessing

All input images are resized to 224×224 pixels before inference. For standard neural models, pixel values are normalized to the [0, 1] range. For Hugging Face ViT-compatible models (identified by the tensor name 'pixel values'), normalization maps to [−1, 1], using mean = 0.5 and standard deviation = 0.5. Face detection uses the OpenCV Haar Cascade frontal face detector. A 20% margin is added around the detected bounding box to include the forehead, chin, and side regions that are commonly altered in face-swap manipulations. When no face is detected, the full image is processed as a fallback, with a corresponding reduction in expected accuracy of approximately 6-8 percentage points.

Neural Inference

The inference engine supports three model formats: ONNX Runtime (recommended), TensorFlow/Keras (keras, .h5), and Torch Script (.pt). Input channel ordering - NCHW or NHWC - is detected automatically from the tensor shape. All raw model outputs are converted to a fake probability in [0, 1] using the following normalization:

$P\_model = \text{sigmoid}(z)$, if $z \in (-\infty, 0) \cup (1, \infty)$

$P\_model = \text{SoftMax}([z_0, z_1])$ [1], if output is a 2-class logit vector

This normalization ensures consistent score scaling regardless of which backend or model architecture is used.

Forensic Signal Extraction

Error Level Analysis (ELA)

When a JPEG image is re-compressed, regions that have already been compressed once settle at a lower error level than regions that have been altered - because altered regions carry a different compression history. ELA exploits this property by computing the pixel-wise absolute difference between the original image and a re-compressed version of it. To reduce sensitivity to any single quality setting, the analysis is performed at three JPEG quality levels: Q = {95, 90, 85}.

$ELA\_q(I) = |I - \text{decompress}(\text{compress}(I, q))| / 255$

Three summary statistics are derived from the resulting error map: the mean error (ELA_mean), the 95th percentile (ELA_P95), and the 99th percentile (ELA_P99). Together these capture both the typical level of compression inconsistency across the image and its worst-case local extremes - a combination that is more discriminative than any single statistic alone.

3.4.2 JPEG Block-Boundary Artifact Score

JPEG compression divides images into 8×8-pixel blocks and compresses each independently. Authentic images retain a consistent and subtle discontinuity pattern at these block edges. GAN-generated faces disrupt this pattern, either because they were created outside the JPEG encoding pipeline or because regional manipulations introduce incompatible compression histories at boundaries. The block score B measures the mean absolute intensity difference across all horizontal and vertical 8-pixel grid boundaries in the grayscale image G:

$B = (1/2) \times [\text{mean}(|G[i, 8k] - G[i, 8k-1]|) + \text{mean}(|G[8k, j] - G[8k-1, j]|)] / 18$

The normalization constant 18 is empirically calibrated to the boundary contrast range observed in facial images from the training set, ensuring the score falls within a consistent range across different image types.

Laplacian Noise Inconsistency

Camera sensors introduce a characteristic high-frequency noise pattern into photographs - a fingerprint that is spatially consistent across the image. GAN-synthesized faces often lack this property: some regions are over-smoothed while others carry noise patterns that are statistically inconsistent with the surrounding image. The Laplacian operator highlights these high-frequency components, and the variance of the resulting map provides a scalar measure of noise consistency:

$N = \text{clip}(\text{Var}(\nabla^2 G) / 800, 0, 1)$, where $\nabla^2 G = \text{Laplacian}(G)$

The divisor 800 is a normalization factor derived empirically to scale the score to [0, 1] across the range of values observed in the training data.

Forensic Score Composition

Each of the three signals is computed separately for the detected face region and for the full image. The individual signal scores are combined into a single forensic score for each region, and the higher of the two regional scores is retained - a conservative choice that ensures any suspicious region, regardless of whether it falls inside or outside the face crop, contributes to the final decision:

$S = \text{clip}(0.20 \times \text{ELA\_mean} + 0.30 \times \text{ELA\_P95} + 0.20 \times \text{ELA\_P99} + 0.15 \times B + 0.15 \times N, 0, 1)$

$P_{forensic} = \text{clip}(0.50 \times \max(S_{face}, S_{full}) + 0.30 \times \text{artifact\_peak} + 0.20 \times \text{artifact\_consistency}, 0, 1)$

Score Fusion

The neural probability and the forensic probability are combined through a fixed weighted sum. The neural weight ($w_1 = 0.65$) reflects the higher average discriminative power of the trained model on in-distribution images. The forensic weight ($w_2 = 0.35$) contributes robustness in cross-domain scenarios where the neural model's confidence is unreliable:

$P_{final} = \text{clip}(0.65 \times P_{model} + 0.35 \times P_{forensic}, 0, 1)$

Conservative Override Rule

To reduce false accusations on genuine images, a correction is applied when the forensic signals are strongly consistent with authenticity but the neural model is only mildly confident about a fake verdict:

If (forensic label = Real) AND (forensic confidence ≥ 0.85) AND ($P_{model} < 0.80$):

then $P_{final} = \min(P_{final}, 0.45)$

This rule does not override strong neural detections - it applies only when the neural confidence is below the 0.80 threshold, which corresponds to cases where the model is uncertain enough that the forensic signal should carry greater weight.

Hard Detection Flags

To prevent the system from missing clearly manipulated images when the neural model is uncertain, six threshold-based flags are evaluated. If any two or more trigger simultaneously, the label is forced to Fake and the final score is set to at least 0.70:

ELA P99 value (face or full image) ≥ 0.22

Block boundary score (face or full image) ≥ 0.52

Noise inconsistency score (face or full image) ≥ 0.42

Combined condition: artifact peak ≥ 0.48 and artifact consistency ≥ 0.34

These thresholds were derived through grid search over the FaceForensics++ validation split and verified on Celeb-DF v2 data without further adjustment.

3.9 Adaptive Threshold Calibration

Rather than applying a fixed decision threshold, the system searches for the optimal value T* by maximizing balanced accuracy over a labeled calibration set. Balanced accuracy treats false positives and false negatives equally, which prevents the threshold from shifting toward whichever class is more common in the calibration data:

$\text{BalAcc}(T) = [\text{TPR}(T) + \text{TNR}(T)] / 2$

$T^* = \text{argmax over } T \in [0.40, 0.90], \text{step} = 0.01$

The threshold $T^* = 0.62$, derived from the FF++ validation split, was used for all experiments reported in this paper.

IV. SYSTEM IMPLEMENTATION

Module Structure

This separation allows the neural backend to be replaced by updating a single file, without modifying the forensic analysis or fusion components.

Table 2: Software module summary

Module	File	Function	Technology
Model Loader	download_model.py	Downloads pretrained weights	urllib.request
Inference Engine	model_backend.py	Runs neural model, normalizes output	ONNX Runtime, TF, PyTorch
Forensic Analyzer	forensic.py	ELA, block score, noise score, heatmap	OpenCV, NumPy
Web Interface	app.py	Score fusion, UI, result display	Gradio
Threshold Tuner	calibrate_threshold.py	Finds optimal T* via balanced accuracy	NumPy, OpenCV
Live Detector	realtime.py	Webcam frame analysis with overlay	OpenCV VideoCapture

Implementation Specifications

Table 3: Implementation details

Component	Specification
Language	Python 3.9+
Neural Inference	ONNX Runtime 1.16+ (CPU)
Image Processing	OpenCV 4.8+, NumPy 1.24+
Web Interface	Gradio 4.x
Input Resolution	224 × 224 pixels
ELA Quality Levels	95, 90, 85 (three-scale averaging)
Face Detector	Haar Cascade frontal face (OpenCV built-in)
Threshold Search Range	[0.40, 0.90], step = 0.01
Webcam Frame Rate	Every 4th frame analyzed (configurable)
Test Hardware	Intel Core i5 CPU, 8 GB RAM, no GPU

V. RESULTS ON FACEFORENSICS++

Table 4: Performance on Face Forensics++ c23 split

Method	Accuracy	Precision	Recall	F1-Score	AUC-ROC
XceptionNet [2]	87.9%	0.876	0.881	0.878	0.921
EfficientNet-B4	88.3%	0.881	0.879	0.880	0.928
Multi-Attn CNN [4]	88.7%	0.884	0.888	0.886	0.931
ViT-B/16 [5]	90.1%	0.899	0.903	0.901	0.948
Forensic Only (Ours)	74.2%	0.738	0.751	0.744	0.792
HFNF (no override)	89.7%	0.895	0.891	0.893	0.951
HFNF Full (Ours)	91.3%	0.912	0.908	0.910	0.957

HFNF Full achieves the highest result across all six metrics. The gain over the nearest competitor, ViT-B/16, is modest in absolute terms - 1.2 percentage points in accuracy and 0.009 in AUC-ROC - but the two systems are operating close to the benchmark ceiling. The more meaningful comparison is on cross-dataset data, where the difference is considerably larger.

Cross-Dataset Generalization

Table 5: Cross-dataset results (trained on FF++, tested on Celeb-DF v2)

The cross-dataset results are the strongest evidence for the value of the forensic component. XceptionNet loses 14.4 percentage points when moved from FF++ to Celeb-DF v2 - a decline that reflects its dependence on GAN-specific patterns learned from the training data. HFNF loses only 7.7 points.

Method	Accuracy	F1-Score	AUC-ROC	Drop from FF++
XceptionNet [2]	73.5%	0.741	0.798	−14.4 pp
ViT-B/16 [5]	78.2%	0.779	0.841	−11.9 pp
HFNF Full (Ours)	83.6%	0.831	0.889	−7.7 pp

The likely reason is that ELA and noise analysis are sensitive to properties of the image - compression history and sensor noise structure - that remain distinguishable between real and fake images regardless of which generation method was used. This

is precisely the cross-domain robustness the forensic component was intended to provide.

Table 6: Per-component processing time (Intel Core i5, CPU only)

Component	Time (ms)
ONNX Neural Inference	38 ms
ELA — three scales, two regions	22 ms
Block-boundary score	4 ms
Noise inconsistency score	3 ms
Fusion and threshold	< 1 ms
Total per image	~ 68 ms
Real-time (every 4th frame)	~ 17 ms effective

The system processes a single image in approximately 68 milliseconds on a standard CPU with no GPU required. This is sufficient for offline batch processing, web application use, and real-time analysis of webcam feeds at standard frame rates.

VI. DISCUSSION

The improvement over standalone neural models come from structural complementarity rather than raw model size or additional training data. Neural detectors are weakest on compressed images because the JPEG artifacts mask the subtle texture signatures that GANs leave behind. ELA is, in a sense, the opposite: it becomes more sensitive under compression, because the differential error between an authentic region and a manipulated one grows as the image is re-encoded. When the neural component is uncertain - when its output probability is close to the decision threshold - the forensic signals frequently provide the additional information needed to make a correct decision. This is reflected clearly in the ablation: the full fusion achieves a lower false-positive rate than either component alone.

The false-positive rate in deepfake detection deserves more attention than it typically receives in published evaluations. In a journalism or legal context, labeling a genuine photograph as a deepfake carries real consequences - it can damage reputations, compromise evidence, or undermine trust in the system itself. Our analysis of the cases where real images were incorrectly flagged found a consistent pattern: most were professionally retouched portraits or images that had been compressed and re-uploaded multiple times, both of which produce artifact patterns superficially similar to GAN outputs. The conservative override rule was designed specifically to handle these cases, and the ablation data shows it works: the FP rate drops from 12.1% to 7.6% at a cost of 0.8% in recall - a trade-off that is clearly favourable in most deployment scenarios.

Three observations from deployment testing are worth noting for anyone considering using this system in practice. First, accuracy is noticeably lower on images where the face detection step fails - partial faces, unusual angles, or heavily cropped images. The fallback to full-image analysis recovers some performance, but a gap of 6-8 percentage points remains on these cases. Second, extreme lighting conditions - overexposure or very dark images - can produce anomalous ELA signals on genuine photographs, contributing to residual false positives. Third, the 68ms processing time is adequate for most practical use cases, but the real-time webcam mode analyzes only every fourth frame and should be treated as a screening indicator rather than a forensic-grade verdict.

VII. CONCLUSION

This paper presented a deepfake image detection system that combines the pattern-recognition capability of pretrained neural networks with the domain-independent reliability of classical image forensics. The key design decision - fusing scores after independent inference rather than training the components jointly - makes the system practically flexible: any pretrained model can be substituted into the neural slot without rebuilding or retraining the forensic layer.

The experimental results support the approach across two dimensions. On Face Forensics++, the proposed framework achieves 91.3% accuracy and an AUC-ROC of 0.957, outperforming the best standalone neural baseline. On Celeb-DF v2, tested with no fine-tuning on that data, the accuracy drop is 7.7 percentage points - compared to 14.4 points for XceptionNet, the most widely cited alternative. The conservative override rule reduces the false-positive rate to 7.6% without meaningfully affecting recall. The system runs at approximately 68 milliseconds per image on a standard CPU and is delivered as working, deployable software.

The limitations are real and acknowledged: forensic signal effectiveness will decline as generative models

improve, face detection failures cause measurable accuracy gaps on unusual inputs, and the current design does not leverage temporal consistency across video frames. Addressing these limitations - through adaptive fusion weights, frequency-domain signal extensions, and video-level analysis - forms the direction of future work.

VIII. FUTURE WORK

Adaptive fusion weights: The current weights (0.65/0.35) were determined by grid search on a single dataset. A meta-learning layer that adapts these weights based on estimated compression level, detection confidence, and image resolution could improve results on edge cases.

Frequency-domain features: DCT coefficient statistics and spectral analysis can detect GAN fingerprints that are invisible in the spatial domain. Adding these signals would extend the system's coverage to generation methods that currently evade spatial artifact detection.

Video-level temporal analysis: Single-image analysis misses consistency signals available across frames - identity embedding stability, optical flow anomalies, and blink rate irregularities. Extending the pipeline to video would add a significant detection dimension.

Adversarial robustness: As GAN training increasingly incorporates detection-awareness, evaluating this framework against adversarial crafted deepfakes would provide a more realistic picture of its real-world ceiling.

Dual explainability: Integrating GRAD-CAM from the neural backend alongside the forensic heatmap would provide two complementary spatial explanations - one semantic, one signal-based - improving utility in high-stakes review scenarios.

Mobile deployment: INT8 quantization of the ONNX model would enable real-time inference on mobile and embedded hardware, expanding the practical deployment range considerably.

REFERENCES

[1] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, "DeepFakes and beyond: A survey of face manipulation and fake detection," Inf. Fusion, vol. 64, pp. 131–148, 2020, doi: 10.1016/j.inffus.2020.06.014.

[2] Rossler et al., "Face Forensics++: Learning to detect manipulated facial images," in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), 2019, pp. 1–11.

[3] L. Li et al., "Face X-ray for more general face forgery detection," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2020, pp. 5001–5010.

[4] H. Zhao et al., "Multi-attentional deepfake detection," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2021, pp. 2185–2194.

[5] Y. Zheng et al., "Exploring temporal coherence for more general video face forgery detection," in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), 2021, pp. 15044–15054.

[6] H. Farid, "Image forgery detection: A survey," IEEE Signal Process. Mag., vol. 26, no. 2, pp. 16–25, 2009, doi: 10.1109/MSP.2008.931079.

[7] X. Pan, X. Zhang, and S. Lyu, "Exposing image splicing with inconsistent local noise variances," in Proc. IEEE Int. Conf. Comput. Photography (ICCP), 2012, pp. 1–10.

[8] L. Verdoliva, "Media forensics and DeepFakes: An overview," IEEE J. Sel. Topics Signal Process., vol. 14, no. 5, pp. 910–932, 2020.

[9] Y. Li et al., "Celeb-DF: A large-scale challenging dataset for DeepFake video forensics," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2020, pp. 3207–3216.

[10] Goodfellow et al., "Generative adversarial nets," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 27, pp. 2672–2680, 2014.

[11] Y. Qian et al., "Thinking in frequency: Face forgery detection by mining frequency-aware clues," in Proc. Eur. Conf. Comput. Vis. (ECCV), LNCS 12357, pp. 86–103, 2020.