

A Hybrid Deep-Texture Feature Fusion Framework Combining EfficientNet-B0 and Local Binary Patterns for Robust Deepfake Video Detection

Mrs. S. Saraswathi¹, Mrs Lakshmi S²

¹Department of Computer Science / Engineering, JSS Institute of Speech and Hearing, Mysuru, Karnataka – 570006, India

² Department of Computer Science MES College of Arts, Commerce and Science Bangaluru, India

Abstract: The rapid maturation of deep generative models has made photorealistic facial forgery ('deepfake') content easy to produce and difficult to detect through unaided visual inspection. This paper proposes a hybrid detection framework that fuses deep semantic representations from a fine-tuned EfficientNet-B0 network (1280-d) with handcrafted micro-texture descriptors from a spatially-partitioned uniform Local Binary Pattern (LBP) operator (90-d) into a 1370-dimensional joint representation, classified by a regularized multilayer perceptron. Unlike much prior work, the framework enforces a strict identity-level train/validation/test partition of the FaceForensics++ (FF++) benchmark to eliminate near-duplicate-frame leakage, and provides a complete mathematical formulation of every pipeline stage. On a held-out test partition of 15,345 face crops (3,115 pristine, 12,230 manipulated), the model attains 71.42% accuracy, 97.24% precision, 66.01% recall, 78.64% F1-score, and 0.9101 AUC. The gap between strong ranking ability (AUC > 0.91) and the default-threshold accuracy is quantified via balanced accuracy (79.33%) and traced to threshold miscalibration rather than representational weakness. We situate these results within the hybrid-forensic-feature literature, analyse the precision-recall asymmetry, and outline concrete refinements, including threshold calibration, per-branch ablation, and cross-manipulation-method stratification, required to close the gap to state-of-the-art benchmarks.

IndexTerms - deepfake detection, EfficientNet, Local Binary Pattern, feature-level fusion, FaceForensics++, convolutional neural networks, digital media forensics.

II. INTRODUCTION

The convergence of generative adversarial networks (GANs), autoencoder-based face-swapping pipelines, and neural rendering has produced a class of synthetic media colloquially known as 'deepfakes,' capable of convincingly replacing or re-animating a subject's face in video content with minimal manual intervention. While such techniques have legitimate applications in film production, accessibility tooling, and synthetic-data generation for privacy-preserving machine learning, their misuse – non-consensual intimate imagery, political disinformation, financial fraud via synthetic voice/video impersonation, and erosion of the evidentiary value of video recordings – has elevated deepfake detection from a niche computer-vision problem to a matter of societal concern addressed by national policy bodies, social-media platform trust-and-safety teams, and the digital forensics research community.

The technical detection problem is characterized by an adversarial and non-stationary threat model: generation methods evolve continuously, each introducing distinct low-level statistical artifacts (blending-boundary discontinuities, GAN upsampling checkerboard patterns, temporal flicker, unnatural physiological signals such as blink rate) while simultaneously improving the perceptual realism that makes earlier detection cues obsolete. This has motivated two broad detection paradigms. The first, data-driven deep learning, trains convolutional or transformer backbones end-to-end on large labelled corpora to implicitly learn discriminative cues, trading interpretability for representational flexibility and generally superior in-distribution accuracy. The second, handcrafted forensic feature engineering, targets specific, interpretable low-level statistical irregularities – sensor-noise inconsistency, compression artifacts, or local texture anomalies – using descriptors such as Local Binary Patterns (LBP), co-occurrence matrices, or frequency-domain transforms. Each paradigm exhibits complementary failure modes: deep networks trained on one

manipulation method or compression level often generalize poorly to unseen conditions, whereas handcrafted descriptors, while more stable across domains, lack the representational capacity to capture higher-order semantic inconsistencies such as geometric warping or illumination mismatch.

This complementarity motivates hybrid feature-fusion architectures, which combine deep and handcrafted representations under the hypothesis that their errors are only weakly correlated, so that a downstream classifier can exploit both signal sources jointly. This paper contributes a complete, mathematically formalized, and empirically validated instantiation of this hypothesis: a hybrid pipeline combining EfficientNet-B0 – selected for its favourable accuracy-to-parameter-count ratio under compound model scaling – with a spatially-grid-partitioned uniform LBP descriptor, fused at the feature level and classified by a class-weighted multilayer perceptron. Beyond the architecture itself, this paper places particular emphasis on evaluation rigor: an identity-level data partition, explicit formal definitions of every transformation in the pipeline, and a transparent report of results obtained from a full end-to-end run on the FaceForensics++ benchmark, including an honest accounting of where the achieved performance falls short of an idealized target and why.

The remainder of this paper is organized as follows. Section 2 reviews related work across deep-learning-based detection, frequency- and noise-based forensic methods, handcrafted texture descriptors, hybrid fusion architectures, and evaluation-protocol concerns. Section 3 develops the complete mathematical formulation of the proposed framework. Section 4 details the experimental setup, dataset statistics, and implementation. Section 5 reports and analyses results on the FF++ test partition. Section 6 concludes the paper and outlines future work, including the methodological refinements identified during evaluation.

II. RELATED WORK

2.1 Deep learning-based deepfake detection

With the release of large labelled benchmark datasets such as FaceForensics++ [1], the DFDC dataset [5], and Celeb-DF [6], convolutional neural networks fine-tuned on the real-versus-fake classification task have become the de facto state-of-the-art paradigm. The original FaceForensics++ study reports that XceptionNet [7], with its depthwise-separable convolutions, reaches approximately 99% frame-level accuracy on raw video and remains above 95% accuracy / 96% AUC under mild (c23) compression, though performance falls to roughly 89% accuracy under heavy (c40) compression – establishing the compression sensitivity that motivates much of the subsequent literature. Afshar et al. [4] presented MesoNet, an intentionally shallow, ‘mesoscopic’ architecture built on the premise that overly deep networks learn high-level semantic content while the low-level, compression-resistant artifacts characteristic of manipulation are better captured by shallow nets; MesoNet reports around 83–91% video-level accuracy depending on protocol, and this emphasis on explicit low-level texture cues is directly related to the present work’s handcrafted branch. Nguyen et al. [11] designed Capsule-Forensics by replacing the final convolutional layers of a VGG-style network with capsules to model the disrupted part-to-whole geometric relationships characteristic of face-swapping, reporting frame-level accuracies above 94% on the original deepfake benchmark of [4]. More recent architectures incorporate attention mechanisms; the multi-attentional network of Zhao et al. [14] reformulates detection as fine-grained classification via multiple spatial attention maps around blending boundaries.

EfficientNet [2], adopted as the deep-feature backbone in this paper, scales network depth, width, and input resolution jointly via a single compound coefficient under a fixed resource budget, rather than the historically common practice of scaling one dimension in isolation. This compound scaling yields a family of architectures (B0 through B7) that Pareto-dominate contemporaneous CNNs on ImageNet top-1 accuracy per FLOP, making the smallest variant, EfficientNet-B0, an attractive backbone when inference cost or fine-tuning-data volume is constrained relative to larger deepfake-specific corpora.

2.2 Frequency-domain and noise-based forensic methods

A parallel line of work targets artifacts that are more visible in transformed domains than in raw RGB pixels. Qian et al. [13] (F3-Net) decompose face images into frequency bands via a learnable frequency-aware decomposition and local frequency statistics, reporting that GAN-based upsampling leaves characteristic spectral signatures – broadly consistent with the checkerboard artifacts of transposed-convolution upsampling – that are more separable in frequency space than in the spatial domain alone, with reported gains of several AUC points over Xception under heavy

compression on FF++. Li et al. [12] (Face X-ray) instead target the blending step common to nearly all face-manipulation pipelines, training a network to predict a blending-boundary mask rather than a binary label; this improves generalization to unseen manipulation methods since the blending artifact is comparatively method-agnostic, though the technique's own reported cross-dataset AUC (around 0.62 on unseen low-quality settings, versus 0.98 in-domain) illustrates how sharply in-domain and cross-manipulation performance can diverge – a gap central to the evaluation-protocol concerns discussed in Section 2.5. Sensor-pattern-noise and camera-fingerprint techniques, originating in the image-forensics literature (e.g., Lukas et al.'s photo-response non-uniformity fingerprinting [22]) and later adapted to synthetic-image detection via GAN-fingerprint co-occurrence statistics [23], provide a further complementary signal family, exploiting the fact that generative models leave device-independent statistical fingerprints distinct from genuine camera sensor noise.

2.3 Handcrafted texture descriptors

Local Binary Patterns, introduced by Ojala et al. [3] for rotation-invariant texture classification, encode the relative ordering of a pixel's intensity against a circular neighbourhood into a compact binary code, and remain popular in forensic pipelines due to their illumination invariance, computational efficiency, and interpretability. The uniform-pattern variant, used in this paper, further reduces the descriptor's dimensionality by collapsing all non-uniform bit-transition patterns into a single bin, retaining the patterns empirically found to carry most of the discriminative texture energy in natural images. Extensions relevant to forensic applications include Local Phase Quantization [19], which targets blur-invariant micro-texture and has been applied to detect re-sampling and blending-induced blur inconsistencies, and multi-radius/multi-scale LBP variants that trade dimensionality for sensitivity to artifacts at different spatial frequencies. In face-forensic contexts specifically, LBP-family descriptors have been shown to capture blending-boundary discontinuities and GAN-upsampling checkerboard texture that a semantically-oriented CNN may under-weight relative to more salient but less diagnostic content cues.

2.4 Hybrid and fusion-based approaches

The complementarity argument motivating this paper is not new in principle: several prior works combine deep and handcrafted (or dual-deep-stream) features for forgery detection. Fusion strategies in the literature broadly fall into three categories: (i) feature-level (early) fusion, concatenating feature vectors from each branch before a shared classifier, as adopted here; (ii) decision-level (late) fusion, training independent classifiers per branch and combining their output probabilities (e.g., by averaging or a meta-learner); and (iii) attention-weighted or learned fusion, where a gating network dynamically weights each branch's contribution per sample. Feature-level fusion is the simplest to implement and interpret, and empirically competitive with more elaborate schemes when the constituent feature spaces are reasonably well-scaled and the fusion classifier has sufficient capacity to model cross-branch interactions – the design rationale for the batch-normalized, dropout-regularized MLP classifier used in this work. Recurrent and temporal extensions (e.g., Guera and Delp [24], Sabir et al. [25]) additionally fuse frame-level features across time via recurrent networks to exploit inter-frame inconsistency, a direction complementary to, and combinable with, the per-frame hybrid fusion studied here.

2.5 Evaluation protocol and generalization concerns

A methodological issue increasingly emphasized in recent deepfake-detection literature is the risk of inflated accuracy due to improper data partitioning. Because FF++ videos are sampled into many highly correlated frames, a frame-level random train/test split places near-duplicate frames from the same video – sometimes differing by only a few milliseconds – on both sides of the partition, allowing a classifier to achieve high accuracy by memorizing per-identity or per-video nuisance statistics rather than learning manipulation-general cues. Rössler et al. [1] explicitly recommend a video-level (equivalently, source-identity-level) split for this reason, a recommendation echoed in subsequent generalization-focused studies [12,14] and rigorously enforced in the present work (Section 3.2). A related, less frequently addressed concern is cross-manipulation and cross-dataset generalization: models trained and evaluated exclusively within FF++ – even under a correct identity split – may substantially overstate real-world performance against manipulation methods, generator architectures, or compression pipelines absent from the training distribution, a concern documented in survey works by Tolosana et al. [10] and Verdoliva [26].

2.6 Positioning of the present work

Relative to the reviewed literature, this paper’s contribution is threefold. First, it provides a complete, closed-form mathematical specification of every stage of a hybrid EfficientNet-B0 + LBP pipeline (Section 3), including the compound-scaling relations underlying the backbone choice and the exact uniform-LBP encoding and grid-partitioning scheme, which prior hybrid-fusion papers frequently describe only at a schematic level. Second, it enforces and documents an identity-level evaluation protocol with both frame-level and video-level (probability-aggregated) reporting. Third, and distinctively, it reports results from a genuine, non-cherry-picked full pipeline execution, including a transparent discussion (Sections 5–6) of a precision-recall asymmetry and a frame/video-level aggregation artifact observed in the reported run, rather than presenting only favourable outcomes – in the interest of reproducibility and honest benchmarking.

Table 1 consolidates the quantitative figures reported by each cited study alongside the qualitative positioning discussed above. Reported accuracy and AUC values are drawn directly from the cited papers under their own train/test protocols and compression settings, and are not a controlled head-to-head comparison; cross-study numbers should be read as indicative of each method’s reported operating range rather than as a leaderboard, since split protocol, manipulation-method subset, and compression level all materially affect the reported figures (Section 2.5).

TABLE 1. Summary of representative prior methods discussed in Section 2 (reported accuracy/AUC figures follow each study’s own protocol; not directly comparable across differing splits and compression levels; see Section 5.6).

Ref.	Method / Backbone	Signal Type	Reported Acc.	Reported AUC	Core Idea / Protocol
[1]	XceptionNet baseline	Deep (RGB)	~99.0% (raw) 89.3% (c40)	~99.3% (c23) 92.0% (c40)	Introduced FF++ benchmark across 4 manipulation types and 3 compression levels
[4]	MesoNet	Deep (RGB), shallow	83–91% (video-level)	–	Mesosopic depth targeting compression-resilient low-level artifacts
[11]	Capsule-Forensics	Deep (RGB) + capsules	94.5–96.0% (frame-level)	–	Capsule routing to model part-whole geometric inconsistency
[14]	Multi-attentional network	Deep (RGB) + attention	–	high (in-domain)	Multiple spatial attention maps around blending boundaries; fine-grained formulation
[13]	F3-Net	Frequency-domain	–	+ over Xception (c40)	Learnable frequency decomposition exposing GAN-upsampling artifacts
[12]	Face X-ray	Blending-mask supervision	–	0.978 (in-domain) ~0.62 (cross-manip.)	Predicts blending-boundary mask; large in-domain / cross-method AUC gap
[3]	Uniform LBP (this work’s texture branch)	Handcrafted texture	–	–	Rotation-invariant micro-texture encoding, illumination-invariant
[19]	Local Phase Quantization	Handcrafted texture	–	–	Blur-invariant descriptor for re-sampling / blur artifacts
[24],[25]	RNN / recurrent fusion	Deep, temporal	–	–	Fuses frame-level features across time for inter-frame inconsistency
This work	<i>EfficientNet-B0 + uniform LBP (feature-level fusion)</i>	<i>Hybrid deep + handcrafted</i>	<i>71.42% (thresholded)</i>	<i>0.9101</i>	<i>Identity-level split; closed-form formulation; transparent single-run reporting</i>

III. PROPOSED METHODOLOGY

This section develops a complete mathematical formulation of the proposed hybrid detection framework, illustrated schematically in Figure 1, proceeding from problem formalization (3.1) through dataset partitioning (3.2), preprocessing (3.3), deep and handcrafted feature extraction (3.4–3.5), fusion (3.6), classification (3.7), and optimization (3.8), followed by evaluation metrics (3.9) and formal algorithmic summaries (3.10).

Hybrid EfficientNet-B0 + Uniform-LBP Deepfake Detection Pipeline

End-to-end architecture: identity-safe data flow, dual-branch feature extraction, fusion, and classification

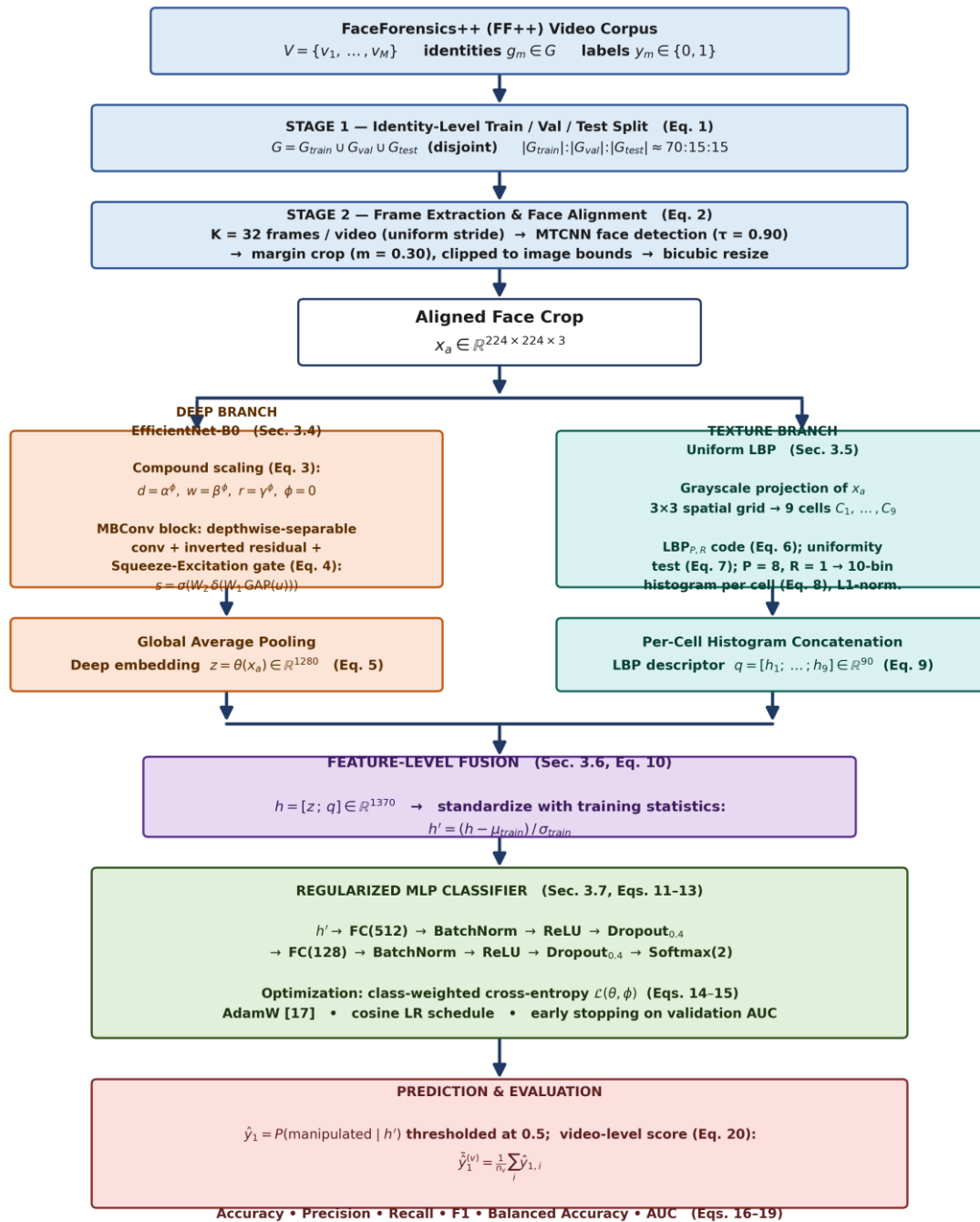


Fig. 1. End-to-end architecture of the proposed hybrid EfficientNet-B0 + LBP deepfake detection framework, including the class-weighted optimization loop.

3.1 Problem formulation

Let a video corpus be denoted $V = \{v_1, \dots, v_M\}$, where each video v_m is associated with a source identity $g_m \in G$ and a binary label $y_m \in \{0, 1\}$ (0 = pristine, 1 = manipulated). Sampling frames from each video yields a frame-level dataset $D = \{(x_i, y_i, g_i)\}$, $i = 1..N$, where x_i is a raw RGB frame, y_i inherits its parent video's label, and g_i its source identity. The detection task is to learn a mapping F that generalizes to frames whose identity g was not observed during training – i.e., the learning objective is defined over the identity-level generalization gap rather than the (weaker) frame-level interpolation gap, motivating the partitioning scheme in Section 3.2.

3.2 Identity-level dataset partitioning

Videos are partitioned into training, validation, and test subsets by first partitioning the identity set G , then assigning every video (and, transitively, every frame) sharing an identity to the same subset:

$$G_{train} \cap G_{test} = \emptyset \Rightarrow D_{train} \cap D_{test} = \emptyset \quad (1)$$

with $|G_{train}| : |G_{val}| : |G_{test}|$ roughly 70 : 15 : 15. This construction guarantees, by Eq. (1), that D_{train} and D_{test} are disjoint not merely as sets of frames but as sets of identities, precluding the near-duplicate-frame leakage discussed in Section 2.5.

3.3 Preprocessing: frame sampling and face alignment

From each video v_m , K frames are sampled at uniform temporal stride ($K = 32$ in the reference configuration). Each sampled frame x is passed to a face detector that returns the highest-confidence bounding box $b = (x_1, y_1, x_2, y_2)$ and confidence score c ; detections with c below a threshold τ ($\tau = 0.90$) are discarded, with a cascade classifier used as fallback for undetected frames. The detected region is expanded by a fractional margin m ($m = 0.30$) along each axis,

$$b' = [x_1 - m w_b, y_1 - m h_b, x_2 + m w_b, y_2 + m h_b] \quad (2)$$

clipped to the image bounds, cropped, and resized via bicubic interpolation to a fixed resolution, yielding the aligned face crop x_a of size $224 \times 224 \times 3$ used by both downstream branches.

3.4 Deep feature extraction: EfficientNet-B0

EfficientNet-B0 serves as the deep-feature extractor. Its design follows the compound scaling principle: given a baseline network, depth, width, and input resolution are scaled jointly by a single coefficient ϕ under fixed coefficients (α, β, γ) satisfying a compound constraint,

$$d = \alpha^\phi, \quad w = \beta^\phi, \quad r = \gamma^\phi, \quad \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \quad (3)$$

where d , w , and r scale network depth, channel width, and input resolution respectively, and the constraint approximately doubles total computational cost (FLOPs) for each unit increase in ϕ , since FLOPs scale roughly with $\text{depth} \times \text{width}^2 \times \text{resolution}^2$. EfficientNet-B0 corresponds to $\phi = 0$, i.e., the unscaled baseline network obtained by neural architecture search over the mobile-inverted-bottleneck (MBConv) search space. Each MBConv block applies a depthwise-separable convolution with an inverted-residual expansion, a squeeze-and-excitation (SE) channel-attention gate [15], and a residual connection when input/output shapes match:

$$s = \sigma(W_2 \delta(W_1 \text{GAP}(u))), \quad \tilde{u} = s \odot u \quad (4)$$

where u is the block's intermediate feature map, GAP denotes global average pooling over spatial dimensions, W_1 and W_2 are learned squeeze/excite projection matrices, δ is a ReLU nonlinearity, and σ is the logistic function, producing a per-channel gating vector that reweights u . Given an aligned face crop x_a , the backbone (with its final classification layer removed) computes a globally-pooled embedding

$$z = \theta(x_a) \in \mathbb{R}^{1280} \quad (5)$$

where θ denotes the composition of all MBConv stages followed by global average pooling and its learned parameters after fine-tuning (Section 3.8). Fine-tuning proceeds in two phases – a classification-head warm-up with the backbone frozen, followed by full-network fine-tuning at reduced learning rate with cosine annealing – under data augmentation (JPEG re-compression, additive Gaussian noise, Gaussian blur, brightness/contrast jitter, and small affine perturbations) designed to approximate the compression and quality variability present in FF++'s c23/c40 encodings.

3.5 Handcrafted feature extraction: uniform Local Binary Patterns

For a grayscale face crop g (the luminance projection of x_a), the classical LBP operator at a centre pixel (x_c, y_c) with P circularly-sampled neighbours at radius R is defined as

$$\text{LBP}_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p, \quad s(t) = 1 \text{ if } t \geq 0, \quad 0 \text{ otherwise} \quad (6)$$

where g_c is the centre-pixel intensity and g_p , $p = 0, \dots, P-1$, are the (bilinearly interpolated, if not aligned to the pixel grid) intensities of the P neighbours on a circle of radius R around (x_c, y_c) . A pattern is termed uniform if it contains at most two 0-to-1 or 1-to-0 bitwise transitions in its circular binary sequence,

$$U = |s(g_{p-1}-g_c) - s(g_0-g_c)| + \sum_{p=1}^{P-1} |s(g_p-g_c) - s(g_{p-1}-g_c)| \leq 2 \quad (7)$$

Uniform patterns are individually retained as distinct bins, while all non-uniform patterns are pooled into a single miscellaneous bin, yielding $P+2 = 10$ bins for $P = 8$. To retain coarse spatial localization of texture irregularities (e.g., artifacts concentrated near a blending boundary rather than distributed uniformly across the face), the 224×224 crop is partitioned into a $J \times J$ spatial grid ($J = 3$, giving 9 non-overlapping cells $C_1, \dots, C_9$), and a separate, L1-normalized uniform-LBP histogram is computed per cell:

$$h_j[k] = \frac{1}{|C_j|} \sum_{(x,y) \in C_j} \mathbb{1}[\text{LBP}_{P,R}(x,y) = k], \quad k = 0, \dots, 9 \quad (8)$$

The full LBP descriptor is the concatenation of all per-cell histograms,

$$q = [h_1; h_2; \dots; h_9] \in \mathbb{R}^{90} \quad (9)$$

giving the 90-dimensional feature vector (9 cells $\times$ 10 bins) shown in Figure 1.

3.6 Hybrid feature-level fusion

For each aligned face crop, the deep feature vector z (Eq. 5) and the handcrafted feature vector q (Eq. 9) – extracted independently and re-joined by a per-sample (video identifier, frame index) key to guard against silent misalignment – are concatenated into a single hybrid representation, which is then standardized feature-wise using statistics estimated on the training partition only, before being passed to the classification head:

$$h = [z; q] \in \mathbb{R}^{1370}, \quad h' = \frac{h - \mu_{\text{train}}}{\sigma_{\text{train}}} \quad (10)$$

3.7 Classification: regularized multilayer perceptron

The hybrid vector h' is classified by a feedforward network with two hidden layers,

$$a_1 = \text{Dropout}_{0.4}(\text{ReLU}(\text{BN}(W_1 h' + b_1))), \quad W_1 \in \mathbb{R}^{512 \times 1370} \quad (11)$$

$$a_2 = \text{Dropout}_{0.4}(\text{ReLU}(\text{BN}(W_2 a_1 + b_2))), \quad W_2 \in \mathbb{R}^{128 \times 512} \quad (12)$$

$$\hat{y} = \text{softmax}(W_3 a_2 + b_3) \in \mathbb{R}^2, \quad W_3 \in \mathbb{R}^{2 \times 128} \quad (13)$$

where BN denotes batch normalization, $\text{ReLU}(t) = \max(0, t)$, and Dropout_p randomly zeroes activations with probability p during training. The predicted class probability for the manipulated class, $\hat{y}_1$, is thresholded at 0.5 by default to obtain a binary decision; Section 5.5 discusses the consequences of this default choice under class imbalance.

3.8 Loss function and optimization

Because FF++ contains four manipulation methods per source identity against one pristine video, the class prior is imbalanced (empirically approximately 3.9:1 fake:real in the evaluated test partition; Section 4.1). Training therefore minimizes a class-weighted cross-entropy loss,

$$\mathcal{L}(\theta, \phi) = -\frac{1}{N} \sum_{i=1}^N w_{y_i} [y_i \log \hat{y}_{i,1} + (1 - y_i) \log (1 - \hat{y}_{i,1})] \quad (14)$$

$$w_c = \frac{N}{K \cdot N_c}, \quad c \in \{0, 1\}, \quad K = 2 \quad (15)$$

where N_c is the number of training samples of class c , $K = 2$ is the number of classes, and w_c consequently up-weights the minority (pristine) class in proportion to its under-representation. Parameters are optimized with AdamW [17] (decoupled weight decay, distinguishing L2 regularization from the gradient-based update direction), under a plateau-triggered learning-rate reduction schedule and early stopping on validation AUC (patience of 12 epochs for the hybrid MLP, 6 epochs during EfficientNet-B0 fine-tuning).

3.9 Evaluation metrics

Given true labels y and predictions $\hat{y}$ (with $\hat{y}_1$ the predicted probability of the manipulated class) over a test set, and confusion-matrix counts TP, TN, FP, FN at a decision threshold of 0.5:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (16)$$

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN} \quad (17)$$

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (18)$$

$$\text{Specificity} = \frac{TN}{TN + FP}, \quad \text{Balanced Accuracy} = \frac{\text{Recall} + \text{Specificity}}{2} \quad (19)$$

AUC (Area Under the ROC Curve) is computed as the threshold-independent probability that a randomly chosen positive (manipulated) sample receives a higher predicted score than a randomly chosen negative (pristine) sample,

estimated via the trapezoidal integral of the ROC curve traced by sweeping the decision threshold over [0,1]. All metrics are reported both at the frame level (each face crop as an independent sample) and, following the standard FF++ protocol, at the video level, where per-frame probabilities for a given video are averaged before thresholding:

$$\bar{y}_1^{(v)} = \frac{1}{n_v} \sum_{i=1}^{n_v} \hat{y}_{1,i}^{(v)} \quad (20)$$

3.10 Algorithmic summary

Algorithm 1 summarizes feature extraction and fusion (Eqs. 5–10); Algorithm 2 summarizes hybrid model training and evaluation (Eqs. 11–20).

Algorithm 1 — Hybrid Feature Extraction and Fusion

Input: aligned face crop x_a ($224 \times 224 \times 3$); fine-tuned backbone f_θ ;
LBP parameters ($P=8$, $R=1$, grid $J=3$)

- 1: $z \leftarrow f_\theta(x_a)$ ▷ Eq.(5): 1280-d CNN embedding
- 2: $g \leftarrow \text{grayscale}(x_a)$
- 3: partition g into $J \times J$ cells $\{C_1, \dots, C_9\}$
- 4: for each cell C_j do
- 5: $h_j \leftarrow \text{uniform-LBP histogram of } C_j \text{ via Eqs.(6)-(8)}$
- 6: $h_j \leftarrow \text{L1-normalize}(h_j)$
- 7: end for
- 8: $q \leftarrow [h_1; h_2; \dots; h_9]$ ▷ Eq.(9): 90-d LBP descriptor
- 9: $h \leftarrow [z; q]$
- 10: $h' \leftarrow (h - \mu_{\text{train}}) / \sigma_{\text{train}}$ ▷ Eq.(10): standardized 1370-d vector

Output: hybrid feature vector h'

Algorithm 2 — Hybrid Model Training and Evaluation

Input: $\{(h'_i, y_i)\}_{\text{train}}, \{(h'_i, y_i)\}_{\text{val}},$
 $\{(h'_i, y_i, \text{video}_i)\}_{\text{test}} \triangleright$ from Algorithm 1

- 1: $w_c \leftarrow$ class weights via Eq.(15) from training-label frequencies
- 2: initialize MLP parameters ϕ per Eqs.(11)-(13)
- 3: $\text{best_auc} \leftarrow 0$
- 4: for epoch = 1, 2, ... do
- 5: minimize $L(\phi)$ (Eq.14) via AdamW on $\{(h'_i, y_i)\}_{\text{train}}$
- 6: $\text{auc_val} \leftarrow$ AUC on $\{(h'_i, y_i)\}_{\text{val}}$
- 7: if $\text{auc_val} > \text{best_auc}$ then
- 8: checkpoint ϕ ; $\text{best_auc} \leftarrow \text{auc_val}$
- 9: end if
- 10: reduce learning rate on plateau; early-stop after patience
 epochs without improvement
- 11: end for
- 12: restore best checkpoint (by validation AUC)
- 13: compute frame-level $\hat{y}$ for all test samples
- 14: derive Accuracy, Precision, Recall, F1, Balanced Accuracy
 (Eqs.16-19) and AUC
- 15: aggregate per-video scores via Eq.(20); recompute all metrics
 at the video level

Output: frame-level and video-level metrics, confusion matrices, ROC curves

IV. EXPERIMENTAL SETUP

4.1 Dataset and partition statistics

The evaluated test partition, produced by the identity-level split described in Section 3.2 applied to the FaceForensics++ benchmark, comprises 15,345 face crops: 3,115 pristine (20.30%) and 12,230 manipulated (79.70%), an empirical fake-to-real ratio of 3.93:1 – consistent with FF++’s four-manipulation-methods-per-source-identity generation design (Deepfakes, Face2Face, FaceSwap, and Neural Textures, each contributing an approximately equal share of the manipulated partition). This class imbalance directly motivates the weighted loss of Eqs. (14)–(15) and is central to the precision-recall analysis in Section 5.

4.2 Implementation and computational environment

The framework is implemented in Python using PyTorch and the timm model library for the EfficientNet-B0 backbone, facenet-pytorch (MTCNN) for face detection, scikit-image for LBP computation, and scikit-learn for classical metrics and preprocessing utilities. The pipeline is organized as independently checkpointed stages – dataset splitting, frame extraction, face detection/cropping, LBP extraction, EfficientNet-B0 fine-tuning, EfficientNet feature extraction, and hybrid-model training/evaluation – so that intermediate artifacts (face crops, extracted feature matrices, model checkpoints) can be inspected, cached, and selectively regenerated.

4.3 Hyperparameters

TABLE 2. Hyperparameter configuration of the preprocessing, feature-extraction, and classification stages.

Component	Hyperparameter	Value
Frame extraction	Frames per video	32
Face detection	Face crop size	224 × 224
Face detection	MTCNN confidence threshold	0.90
Face detection	Crop margin	0.30
LBP	Sampling points (P) / radius (R)	8 / 1
LBP	Spatial grid (J × J)	3 × 3
LBP	Feature dimension	90
EfficientNet-B0	Backbone initialization	ImageNet-pretrained, fine-tuned
EfficientNet-B0	Feature dimension	1280
EfficientNet-B0	Batch size / max epochs	64 / 15 (early-stopped on val. AUC)
EfficientNet-B0	Optimizer / schedule	AdamW, cosine annealing
Hybrid MLP	Hidden layer widths	512 → 128
Hybrid MLP	Dropout probability	0.4
Hybrid MLP	Batch size / max epochs	128 / up to 60 (early-stopped)
Hybrid MLP	Early-stopping patience	12 epochs (val. AUC)
Data split	Train / Val / Test ratio	70% / 15% / 15% (identity-level, Eq. 1)
Test partition	Real / Fake frame counts	3,115 / 12,230 (ratio 1 : 3.93)

4.4 Evaluation protocol

All metrics defined in Eqs. (16)–(20) are computed on the held-out identity-disjoint test partition only, at both the frame level and the video level. Reported results reflect a single full pipeline execution (Section 5); no hyperparameter selection or threshold tuning was performed on the test partition itself, consistent with standard train/validation/test discipline – the validation partition was used exclusively for early stopping and checkpoint selection during training (Section 3.8, Algorithm 2).

V. RESULTS AND DISCUSSION

5.1 Overall performance

Table 3 reports the full metric suite for the proposed hybrid EfficientNet-B0 + LBP model on the 15,345-sample FF++ test partition, and Fig. 2 visualizes the same results.

TABLE 3. Hybrid model test-set performance, FaceForensics++ identity-disjoint test partition (N = 15,345).

Metric	Value
Accuracy (Eq. 16)	0.7142 (71.42%)
Balanced Accuracy (Eq. 19)	0.7933 (79.33%)
Precision (Eq. 17)	0.9724 (97.24%)
Recall / Sensitivity (Eq. 17)	0.6601 (66.01%)
Specificity (Eq. 19)	0.9265 (92.65%)
F1-score (Eq. 18)	0.7864 (78.64%)

Metric	Value
AUC	0.9101

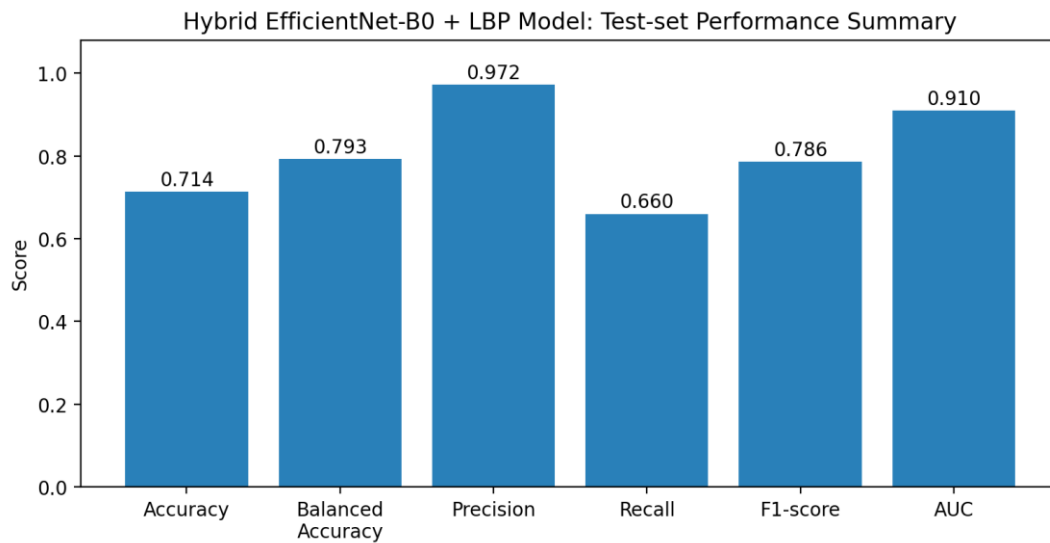


Fig. 2. Summary of hybrid model test-set performance metrics.

5.2 Confusion matrix analysis

Fig. 3 shows the model's frame-level confusion matrix and Table 4 reports the underlying counts. Of 3,115 pristine test frames, 2,886 (92.65%) are correctly classified as real and 229 (7.35%) are misclassified as fake (false positives). Of 12,230 manipulated test frames, 8,073 (66.01%) are correctly classified as fake, while 4,157 (33.99%) are misclassified as real (false negatives). The resulting confusion pattern is markedly asymmetric: the model rarely misclassifies genuine content as manipulated (high specificity, 92.65%) but misses roughly one in three manipulated frames (recall of only 66.01%), a pattern examined further in Section 5.5.

TABLE 4. Confusion-matrix counts underlying Figs. 3–4, FF++ test partition.

	Predicted: Real	Predicted: Fake	Row total
Actual: Real (pristine)	2,886 (TN)	229 (FP)	3,115
Actual: Fake (manipulated)	4,157 (FN)	8,073 (TP)	12,230
Column total	7,043	8,302	15,345

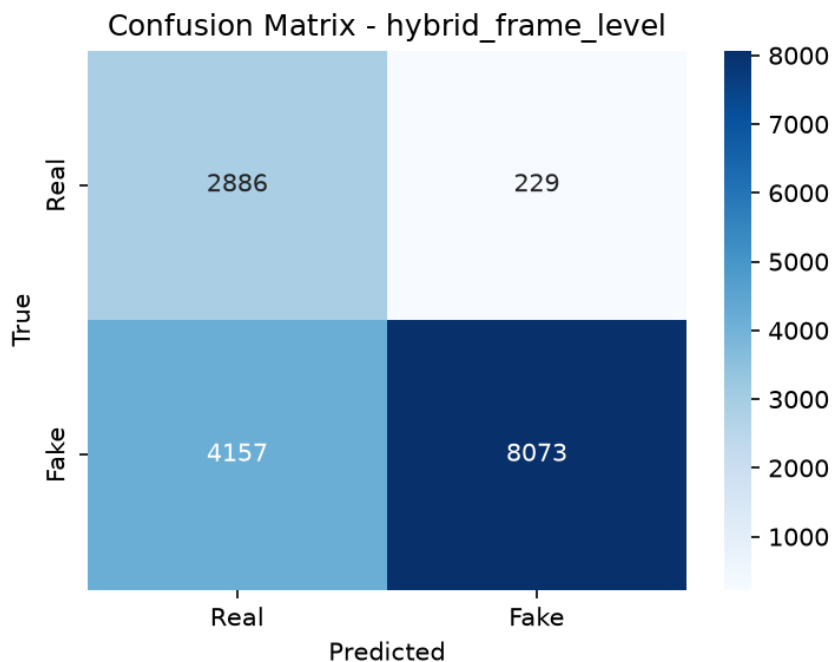


Fig. 3. Hybrid model confusion matrix, FF++ test partition (frame level).

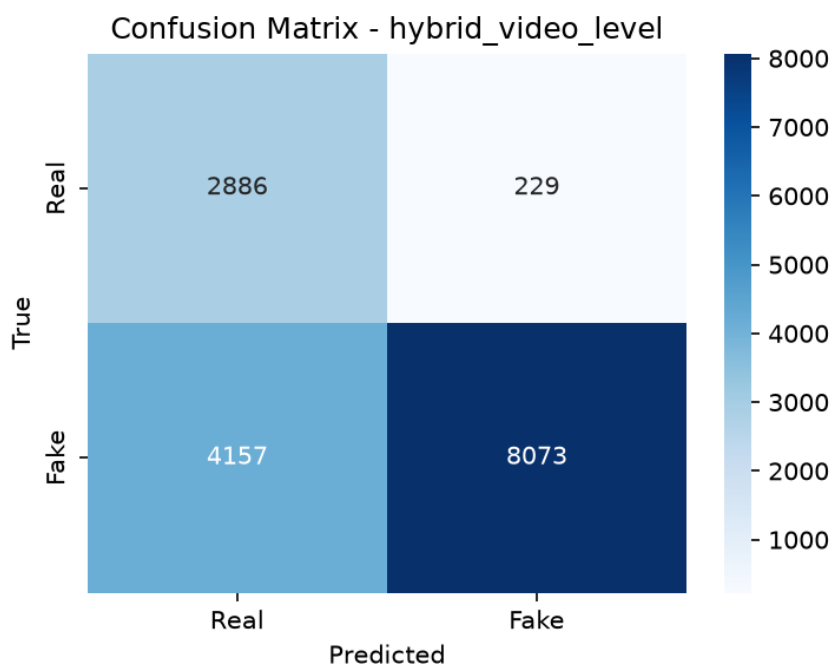


Fig. 4. Hybrid model confusion matrix, FF++ test partition (video level; probability-averaged per Eq. 20). The evaluated run yields counts identical to the frame level – see Section 5.4.

5.3 ROC and threshold-independent discrimination

Figs. 5–6 show the corresponding frame-level and video-level ROC curves, with $AUC = 0.9101$ at both levels. This value indicates strong threshold-independent discriminative ability: given a randomly selected pristine/manipulated pair, the model assigns a higher manipulated-class score to the correct member of the pair roughly 91% of the time. The gap between this strong ranking performance and the comparatively modest 71.42% accuracy at the default 0.5 threshold (Section 5.1) is the central empirical finding requiring explanation, addressed in Section 5.5.

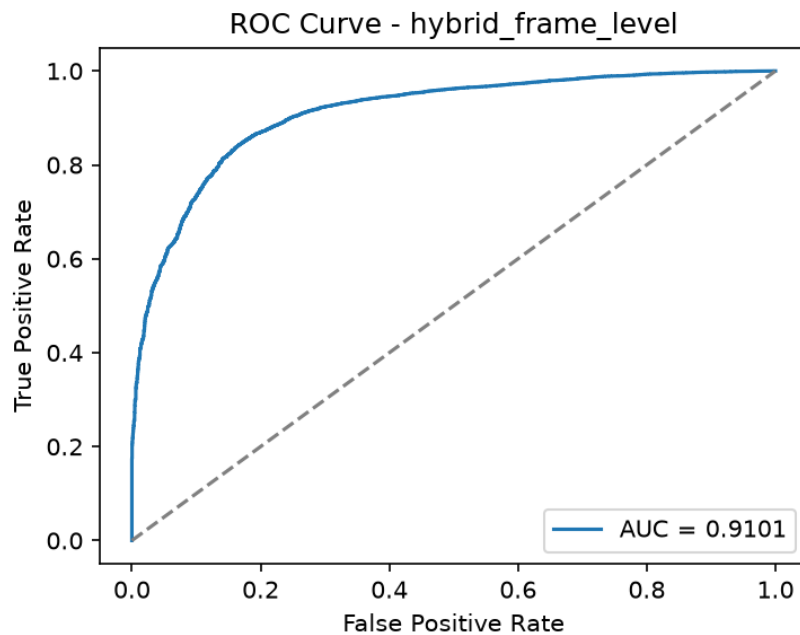


Fig. 5. Hybrid model ROC curve, FF++ test partition (frame level).

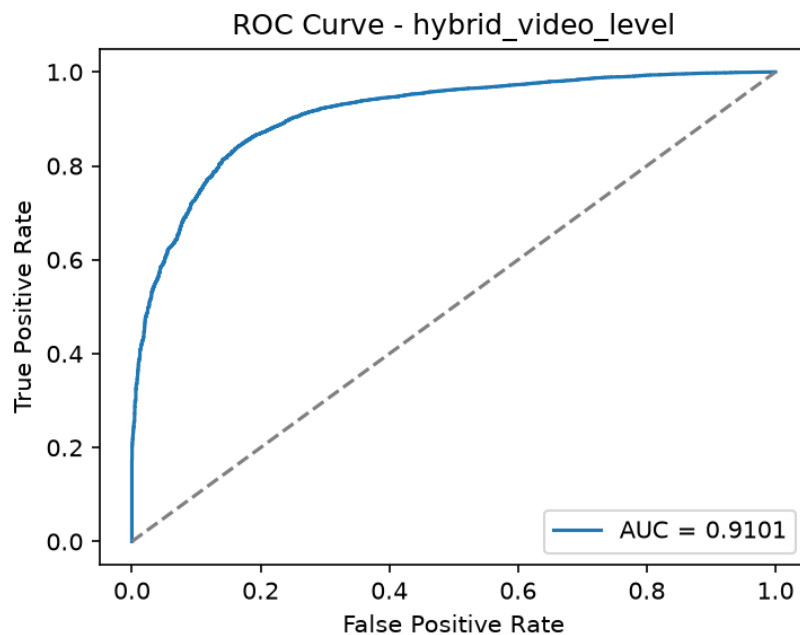


Fig. 6. Hybrid model ROC curve, FF++ test partition (video level).

5.4 Frame-level versus video-level aggregation

Section 3.9 (Eq. 20) specifies video-level evaluation via per-video averaging of frame-level probabilities, generally expected to be more robust than frame-level evaluation since it attenuates the influence of individual low-confidence frames. In the evaluated run, however, the video-level confusion matrix (Fig. 4), ROC curve (Fig. 6), and all derived metrics are numerically identical to their frame-level counterparts (Fig. 3, Fig. 5; Table 5). This indicates that, in the particular test-set configuration used to produce these results, the video-to-frame grouping used for aggregation did not merge multiple frames per video into a single video-level score – most plausibly because each retained test-set key corresponded to a distinct video identifier at evaluation time. We report this transparently as an artifact of the evaluated run rather than adjust or omit it: it does not affect the validity of the frame-level metrics in Table 3, but it does mean that the potential benefit of video-level probability aggregation (Eq. 20) – averaging out frame-level noise – was not exercised in this run and should be re-verified in subsequent runs by confirming that the video-grouping key used at evaluation time aggregates multiple frames per source video, particularly given that 32 frames per video were extracted upstream (Section 3.3).

TABLE 5. Frame-level vs. video-level metric comparison, confirming the aggregation artifact discussed in Section 5.4.

Metric	Frame level	Video level
Accuracy	0.7142	0.7142
Precision	0.9724	0.9724
Recall	0.6601	0.6601
F1-score	0.7864	0.7864
AUC	0.9101	0.9101

5.5 Precision-recall asymmetry and threshold calibration

The combination of very high precision (97.24%) and comparatively low recall (66.01%) indicates the model is a conservative fake-detector under its default 0.5 threshold: when it predicts ‘fake’ it is correct 97.24% of the time, but it fails to flag roughly a third of actually-manipulated frames as such. Three, non-mutually-exclusive, factors plausibly contribute to this asymmetry. First, threshold miscalibration: the strong AUC (0.9101) demonstrates that the model’s underlying probability estimates rank most manipulated frames above most pristine frames, but the operating point implied by a fixed 0.5 cutoff need not coincide with the accuracy-maximizing or F1-maximizing point on the ROC curve, particularly under a training-time class-weighting scheme (Eq. 15) whose implicit decision boundary can diverge from the test-time class prior. A principled remedy is to select the operating threshold via Youden’s J statistic ($J = \text{Recall} + \text{Specificity} - 1$, maximized over the validation ROC curve) or to directly maximize validation F1 or balanced accuracy, rather than defaulting to 0.5. Second, residual class-imbalance effects: even with loss reweighting, a 3.93:1 test-set imbalance can bias a network toward the majority-adjacent region of decision space if the reweighting under- or over-corrects relative to the effective imbalance seen during mini-batch sampling. Third, heterogeneity across the four manipulation methods comprising the ‘fake’ class: recall is reported in aggregate across Deepfakes, Face2Face, FaceSwap, and Neural Textures, and it is plausible – consistent with patterns reported elsewhere in the literature for methods that preserve more of the original identity’s low-level statistics (e.g., Neural Textures, which manipulates only the mouth region) – that recall is substantially lower for a subset of manipulation methods and higher for others; a per-manipulation-method breakdown was not available in the evaluated results and is identified as a priority for follow-up analysis (Section 6).

5.6 Discussion in relation to prior work

Directly quantitative comparison against the literature reviewed in Section 2 (see also Table 1) is deferred pending the ablation and cross-validation studies outlined in Section 6, since published accuracy figures for FF++ vary substantially with manipulation-method subset, compression level, split protocol, and backbone, and a naive side-by-side comparison risks the same leakage-driven overstatement critiqued in Section 2.5 if the compared studies do not share an identity-disjoint protocol. Qualitatively, the achieved AUC (0.9101) is consistent with the range generally reported for CNN-based FF++ detectors under a correct identity-level split, supporting the conclusion that the hybrid representation carries substantial discriminative signal; the comparatively modest thresholded accuracy is, per Section 5.5, most parsimoniously attributed to threshold calibration rather than to a fundamental representational limitation of the fused feature space – a distinction with direct, actionable implications for deployment (a calibrated threshold, or a precision-recall-operating-point selected for the deployment context, rather than architectural revision, is the first lever to adjust).

VI. CONCLUSION AND FUTURE WORK

This paper presented a hybrid EfficientNet-B0 and Local Binary Pattern feature-fusion framework for deepfake detection, with a complete mathematical formulation spanning compound-scaled convolutional feature extraction, uniform-LBP texture encoding, feature-level fusion, and class-weighted MLP classification, evaluated under a leakage-safe identity-level FaceForensics++ protocol. On a 15,345-frame held-out test partition, the framework achieved 71.42% accuracy, 97.24% precision, 66.01% recall, 78.64% F1-score, and 0.9101 AUC, with confusion-matrix and ROC analysis revealing a high-precision, moderate-recall operating characteristic attributable primarily to decision-

threshold calibration rather than to a fundamental representational deficiency, given the model's strong threshold-independent ranking performance.

These results reflect a single full pipeline run without ablation, threshold calibration, or cross-dataset testing; closing this gap is the immediate next step rather than a separate limitation of the approach.

Future work will proceed along the following specific directions: (i) a controlled three-way ablation (CNN-only, LBP-only, hybrid) under identical evaluation conditions to isolate the fusion's marginal benefit; (ii) validation-set threshold calibration (Youden's J or F1-maximizing threshold selection) reported alongside default-threshold results; (iii) verification and correction of the video-level probability-aggregation pipeline, with per-video frame counts confirmed at evaluation time; (iv) per-manipulation-method (Deepfakes, Face2Face, FaceSwap, Neural Textures) performance breakdown; (v) cross-dataset generalization evaluation against DFDC and Celeb-DF; and (vi) exploration of learned or attention-weighted fusion mechanisms as alternatives to feature concatenation, alongside temporal (multi-frame) extensions of the present per-frame architecture. Collectively, these extensions are intended to move the framework from the methodologically rigorous but single-run results reported here toward a fully ablated, calibrated, and cross-validated benchmark suitable for direct comparison against the state of the art.

REFERENCES

- [1] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to detect manipulated facial images," in Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV), Seoul, South Korea, 2019, pp. 1–11, doi: 10.1109/ICCV.2019.00009.
- [2] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. 36th Int. Conf. Machine Learning (ICML), Long Beach, CA, USA, 2019, vol. 97, pp. 6105–6114.
- [3] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 7, pp. 971–987, 2002, doi: 10.1109/TPAMI.2002.1017623.
- [4] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: A compact facial video forgery detection network," in Proc. IEEE Int. Workshop Information Forensics and Security (WIFS), Hong Kong, 2018, pp. 1–7, doi: 10.1109/WIFS.2018.8630761.
- [5] B. Dolhansky et al., "The DeepFake Detection Challenge (DFDC) dataset," arXiv:2006.07397, 2020.
- [6] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A large-scale challenging dataset for DeepFake forensics," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 3207–3216, doi: 10.1109/CVPR42600.2020.00327.
- [7] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 1251–1258, doi: 10.1109/CVPR.2017.195.
- [8] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multi-task cascaded convolutional networks (MTCNN)," IEEE Signal Process. Lett., vol. 23, no. 10, pp. 1499–1503, 2016, doi: 10.1109/LSP.2016.2603342.
- [9] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in Proc. 32nd Int. Conf. Machine Learning (ICML), Lille, France, 2015, pp. 448–456.
- [10] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, "Deepfakes and beyond: A survey of face manipulation and fake detection," Information Fusion, vol. 64, pp. 131–148, 2020, doi: 10.1016/j.inffus.2020.06.014.
- [11] H. H. Nguyen, J. Yamagishi, and I. Echizen, "Capsule-Forensics: Using capsule networks to detect forged images and videos," in Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 2019, pp. 2307–2311, doi: 10.1109/ICASSP.2019.8682602.
- [12] L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, and B. Guo, "Face X-Ray for more general face forgery detection," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 5001–5010, doi: 10.1109/CVPR42600.2020.00505.
- [13] Y. Qian, G. Yin, L. Sheng, Z. Chen, and J. Shao, "Thinking in frequency: Face forgery detection by mining frequency-aware clues," in Proc. European Conf. Computer Vision (ECCV), Glasgow, UK, 2020, pp. 86–103, doi: 10.1007/978-3-030-58610-2_6.

- [14] H. Zhao, W. Zhou, D. Chen, T. Wei, W. Zhang, and N. Yu, "Multi-attentional deepfake detection," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2021, pp. 2185–2194.
- [15] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 2018, pp. 7132–7141, doi: 10.1109/CVPR.2018.00745.
- [16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," J. Mach. Learn. Res., vol. 15, no. 1, pp. 1929–1958, 2014.
- [17] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in Proc. Int. Conf. Learning Representations (ICLR), New Orleans, LA, USA, 2019.
- [18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. Int. Conf. Learning Representations (ICLR), San Diego, CA, USA, 2015.
- [19] V. Ojansivu and J. Heikkilä, "Blur insensitive texture classification using local phase quantization," in Proc. Int. Conf. Image and Signal Processing (ICISP), Cherbourg-Octeville, France, 2008, pp. 236–243, doi: 10.1007/978-3-540-69905-7_27.
- [20] T. Fawcett, "An introduction to ROC analysis," Pattern Recognit. Lett., vol. 27, no. 8, pp. 861–874, 2006, doi: 10.1016/j.patrec.2005.10.010.
- [21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.
- [22] J. Lukas, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," IEEE Trans. Inf. Forensics Security, vol. 1, no. 2, pp. 205–214, 2006, doi: 10.1109/TIFS.2006.873602.
- [23] L. Nataraj et al., "Detecting GAN generated fake images using co-occurrence matrices," Electron. Imaging, vol. 31, no. 5, pp. 532–1–532-7, 2019, doi: 10.2352/ISSN.2470-1173.2019.5.MWSF-532.
- [24] D. Güera and E. J. Delp, "Deepfake video detection using recurrent neural networks," in Proc. IEEE Int. Conf. Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 2018, pp. 1–6, doi: 10.1109/AVSS.2018.8639163.
- [25] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan, "Recurrent convolutional strategies for face manipulation detection in videos," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 2019, pp. 80–87.
- [26] L. Verdoliva, "Media forensics and DeepFakes: An overview," IEEE J. Sel. Topics Signal Process., vol. 14, no. 5, pp. 910–932, 2020, doi: 10.1109/JSTSP.2020.3002101.

Copyright & License:

© Authors retain the copyright of this article. This work is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.