IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A Deep Learning Based Model for Deepfake Video Detection System Using CNN–LSTM Hybrid Architecture

Mrs. D. Mahalakshmi, M.E., (PhD)
Assistant Professor, Dept. of Information Technology
AVC College of Engineering, Mannampandal – 609305
Tamil Nadu, India

Manikandan S
Dept. of Information Technology
AVC College of Engineering
Mannampandal – 609305, India

Mohamed Fayiz M
Dept. of Information Technology
AVC College of Engineering
Mannampandal – 609305, India

Senthilnathan M
Dept. of Information Technology
AVC College of Engineering
Mannampandal – 609305, India

Vishwa G
Dept. of Information Technology
AVC College of Engineering
Mannampandal – 609305, India

***Abstract*—**The proliferation of deepfake videos synthesized through deep generative models such as Generative Adversarial Networks (GANs) and neural autoencoders poses critical threats to digital security, media integrity, political stability, and public trust. Existing detection approaches either focus exclusively on spatial artifacts within individual frames using CNNs, or on temporal inconsistencies using recurrent networks, but rarely integrate both. This paper presents a novel hybrid deep learning framework combining Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) for robust, video-level deepfake detection. The proposed system employs EfficientNetB0 as the spatial feature extractor to capture per- frame facial artifacts, while a two-layer Bidirectional LSTM models temporal inconsistencies across a sequence of 20 uniformly sampled frames. Face regions are detected using the OpenCV Haar Cascade detector with 25% bounding box padding. The model is trained and evaluated on the FaceForensics++ benchmark comprising 7,000 videos across six manipulation categories: Deepfakes, Face2Face, FaceShifter, FaceSwap, NeuralTextures, and DeepFakeDetection. Class imbalance (1:5 real-to-fake ratio) is addressed through Weighted Random Sampling, label smoothing, and stratified splitting. Training employs AdamW optimization, Cosine Annealing scheduling, and PyTorch Automatic Mixed Precision on an NVIDIA Tesla T4 GPU. Experimental evaluation yields a test accuracy of 92.4%, F1-Score of 91.7%, and AUC- ROC of 0.967, demonstrating the superiority of joint spatial- temporal learning over frame-only and temporal-only approaches. A Flask-based real-time web application is developed for practical deployment, providing binary REAL / FAKE verdicts with confidence scores and REST API integration for downstream systems.

***Index Terms*—Deepfake Detection, CNN-LSTM, EfficientNetB0, Bidirectional LSTM, FaceForensics++, Video Forensics, Temporal Analysis, Face Manipulation, GAN, Deep Learning, OpenCV, Flask, Real-Time Detection**

I. INTRODUCTION

The rapid advancement of deep generative models, particularly Generative Adversarial Networks (GANs) [18] and variational autoencoders, has enabled the creation of photorealistic synthetic media commonly referred to as *deepfakes*. In deepfake videos, a person's facial identity is seamlessly replaced or manipulated using algorithmic techniques, making the forgery imperceptible to the human eye. As shown in Fig. 1, modern deepfake generation produces results in which even trained observers struggle to distinguish authentic from manipulated content.

The social and economic consequences are profound. Financial institutions face increased fraud risk as deepfake audio-visual content bypasses biometric verification systems. Legal systems must grapple with challenges to digital evidence authenticity, as synthetic media can fabricate incriminating statements. Citizens are exposed to manipulated political narratives capable of influencing democratic processes and public opinion at scale. Non-consensual intimate imagery generated via deepfake techniques has become a major vector of gender-based digital harassment. Industry threat reports indicate deepfake incidents rose by over 400% between 2022 and 2024, with social media platforms detecting tens of millions of synthetic video uploads per quarter. The challenge is compounded by the GAN arms race: each advance in detection capability spurs adversarial improvements in generation quality, creating a continuous escalation cycle that demands sustained research investment.

Existing automated detection methods broadly fall into two categories. **Spatial feature-based approaches** employ CNNs to analyze visual artifacts within individual frames—pixel-level boundary discontinuities, color inconsistencies, and frequency

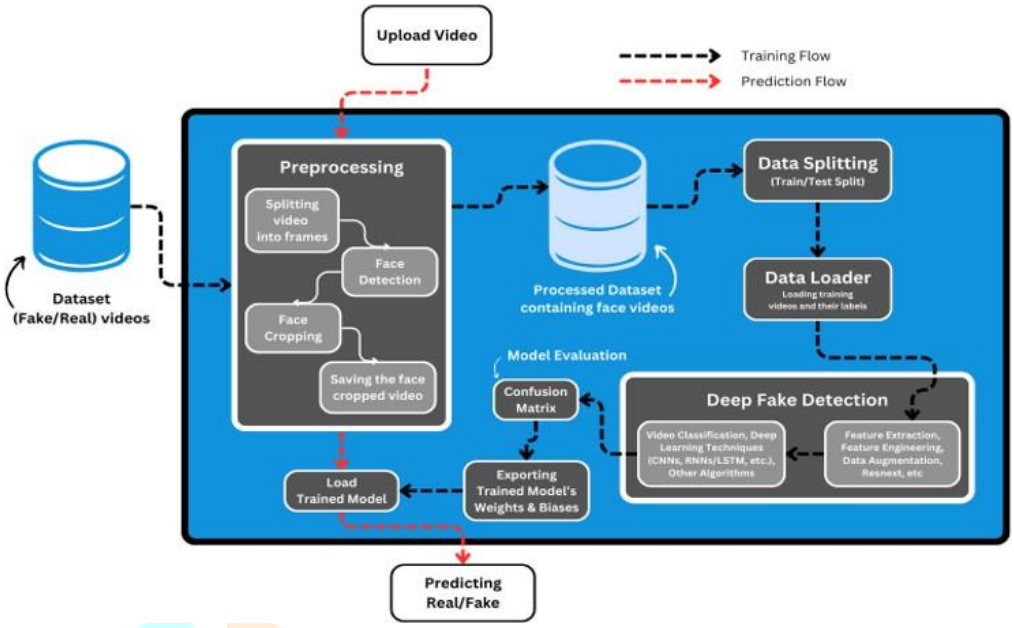

Fig. 1: Illustration of a deepfake video: **Original** (left) vs. **Manipulated** face (right). The deepfake seamlessly replaces the subject's identity while preserving natural facial dynamics including expression, lighting, and head pose. The inset (top-right) shows the source identity used for the face swap. Such high visual fidelity makes manual detection impractical at scale, motivating automated deep learning approaches.

domain anomalies. While effective at capturing localized per-frame forgery evidence, these methods are blind to the temporal dynamics of manipulation. A single deepfake frame may appear perfectly natural when examined in isolation; the forensic evidence often manifests only as subtle inconsistencies across consecutive frames, such as unnatural blinking, lip-sync discrepancies, irregular head motion trajectories, and expression resets. **Temporal-based approaches** employing RNNs or LSTMs model inter-frame inconsistencies but lack the fine-grained spatial resolution necessary to detect subtle facial boundary artifacts when processing raw pixel sequences. The computational cost of modeling high-dimensional pixel-space temporal sequences also renders these approaches impractical for video-length inputs.

This paper bridges this gap with a tightly integrated CNN–BiLSTM hybrid framework. EfficientNetB0 extracts compact, discriminative spatial features from each of the 20 sampled frames, reducing temporal input dimensionality from millions of raw pixels to a manageable 20×1280 feature matrix. The Bidirectional LSTM then models the temporal evolution of these features, capturing both past-to-future progressions and future-to-past retrospective patterns. By jointly optimizing both objectives within a single end-to-end differentiable model with shared gradient flow, the proposed system captures complementary spatial and temporal cues that neither approach alone can match. Fig. 1 provides a motivating example of the detection challenge, while the remainder of this paper details the complete technical solution.

Key Contributions

- A novel end-to-end CNN–BiLSTM architecture that jointly learns spatial facial artifacts and temporal sequence inconsistencies for video-level deepfake classification with 92.4% accuracy and 0.967 AUC-ROC on FaceForensics++.
- A controlled ablation study isolating the contribution of each architectural component: CNN backbone, temporal direction, and bidirectionality.
- Application of Weighted Random Sampling, label smoothing ($\varepsilon = 0.1$), and stratified splitting to robustly address the inherent 1:5 class imbalance.
- A production-grade Flask web application with REST API enabling real-time end-to-end inference on user-submitted videos without local ML installation.
- Comprehensive per-manipulation-type performance characterization identifying which face forgery techniques are most challenging for detection.

Paper Organization

Section II surveys related work. Section III describes the FaceForensics++ dataset and preprocessing. Section IV details the proposed CNN–BiLSTM architecture and training strategy. Section V reports experimental results and ablation analysis. Section VI presents the real-time web application. Section VII

discusses limitations and ethical considerations. Section VIII concludes with future research directions.

II. Related Work

Deepfake detection has attracted substantial research interest since GAN-based face synthesis emerged in 2018. We organize the literature by primary technical approach.

A. Early Forensic and Heuristic Methods

Early detection methods exploited low-level forensic cues including JPEG compression artifacts, color space inconsistencies, and biometric physiological signals such as spontaneous eye blinking rates [1]. These methods operated on heuristic assumptions about the limitations of early GAN generators—limited capacity to reproduce realistic blink patterns or maintain consistent sub-pixel color distributions. Matern et al. [13] demonstrated that visual boundary artifacts at face-blending regions are detectable via pixel-level statistical analysis of color discontinuities. Li et al. [14] proposed Face X-Ray, detecting blending boundaries by analyzing pixel intensity discontinuities at the face perimeter; this approach achieved strong generalization to unseen manipulation types by focusing on the universal face-blending operation common to all forgery pipelines. While effective against early-generation deepfakes, heuristic methods proved brittle as modern GAN generators specifically optimized to suppress the forensic artifacts these detectors exploited.

B. CNN-Based Frame-Level Detection

The dominant detection paradigm through 2023 was CNN-based frame-level analysis. Rossler et al. introduced both FaceForensics++ [8]—the benchmark used in this study—alongside an XceptionNet-based detector that achieved high frame-level accuracy but lacked temporal reasoning, limiting detection of subtly manipulated sequences. Dasgupta et al. [7] explored attention-enhanced CNNs across multiple datasets, demonstrating improved artifact localization through spatial attention maps. Zhao et al. [17] proposed Multi-Attentional Deepfake Detection using multiple attention maps targeting different facial regions simultaneously—eyes, nose, and mouth—capturing complementary spatial evidence. Liu et al. [16] demonstrated that high-frequency noise residuals in the pixel domain carry reliable manipulation evidence that survives compression, providing a robust auxiliary signal for CNN-based classifiers.

C. Recurrent and Temporal Methods

Gue´ra and Delp [9] pioneered CNN–RNN combinations for temporal deepfake modeling, establishing the fundamental paradigm extended by this work. Their system extracted features from individual frames using a CNN and fed the resulting sequence into an LSTM for temporal classification. Sabir et al. [10] empirically confirmed that EfficientNet combined with recurrent architectures yields superior accuracy-efficiency trade-offs compared to ResNet or VGG backbones, and that EfficientNet's compound scaling produces richer, more transfer-learning-friendly representations for face manipulation detection tasks.

D. Transformer and Ensemble Methods

Vision Transformers (ViT) [11] apply self-attention globally, enabling long-range spatial reasoning across the entire image. While ViT achieves state-of-the-art results given sufficient training data, the quadratic attention complexity and data-hungriness make it impractical for video deepfake detection without massive labeled datasets. Rana and Sung [4] proposed DeepFake Stack, a multi-classifier ensemble combining multiple base learners via stacking meta-learning, achieving competitive accuracy at substantially higher inference cost. Hossain et al. [6] and Sharma et al. [3] both provide comprehensive surveys identifying balanced spatial-temporal fusion as the primary open challenge. Alrashoud [2] further characterizes key evaluation challenges including dataset generalization, compression robustness, and deployment latency.

TABLE I: Summary of Related Works on Deepfake Detection

Author(s)	Yr.	Method	Data	Limitation
Li et al. [1]	2024	Hybrid DL + Eye	Custom	Eye-cues only
Rossler et al. [8]	2019	XceptionNet	FF++	No temporal
Rana & Sung [4]	2024	DeepFake Stack	Multi	High cost
Gue´ra & Delp [9]	2018	CNN + RNN	Custom	Small scale
Sabir et al. [10]	2019	EffNet + RNN	FF++	Complex train
Dasgupta et al. [7]	2025	Attention CNN	Multi	No temporal
Zhao et al. [17]	2021	Multi-Attn CNN	FF++	No temporal
Dosovitskiy et al. [11]	2021	Vision Transformer	Large	Data hungry
Proposed	**2026**	**CNN+BiLSTM**	**FF++**	**None**

III. Dataset and Preprocessing

A. FaceForensics++ Overview

The FaceForensics++ (FF++) dataset [8] serves as the primary benchmark for this study. FF++ contains 7,000 videos at c23 compression quality, comprising 1,000 original (real) videos sourced from YouTube and 6,000 manipulated videos spanning six distinct manipulation categories. Each video is sourced from a unique YouTube upload, providing identity diversity and realistic recording conditions including varied lighting, backgrounds, head poses, and recording equipment. The c23 quality level represents a realistic social media upload compression scenario, making FF++ a challenging and practically relevant benchmark.

B. Manipulation Categories

The six FF++ manipulation categories, illustrated in Fig. 2, cover the full spectrum of face forgery techniques:

- **Deepfakes:** Autoencoder-based identity swap using identity-specific decoder networks; the most widely deployed open-source technique.
- **Face2Face:** Expression transfer retaining the target subject's identity, driven by real-time 3D facial tracking and reenactment.
- **FaceSwap:** 3D model-based geometric face swapping without deep learning, using 3D Morphable Models (3DMM).
- **FaceShifter:** High-fidelity identity-preserving face swap via adaptive identity embedding integration with occlusion-aware refinement.
- **NeuralTextures:** Neural texture synthesis on a 3D face model driven by source expressions, producing highly realistic and temporally consistent lip dynamics.
- **DeepFakeDetection (DFD):** Additional high-resolution (1920×1080) videos commissioned by Google, providing a more challenging evaluation subset with professional-grade synthesis.

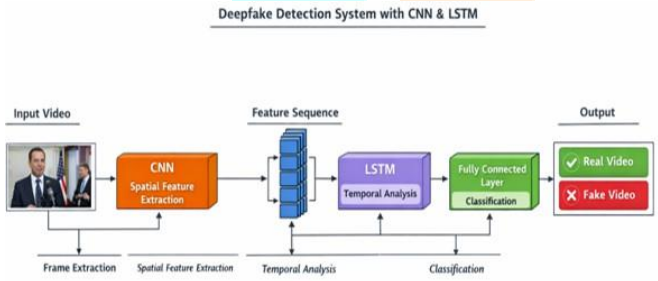

Fig. 2: Pristine vs. FF++ manipulation categories. Subtle skin tone, texture, and boundary artifacts are visible around the chin and hairline regions in manipulated versions. NeuralTextures and FaceShifter are perceptually closest to the pristine source.

C. Dataset Statistics

Table II summarizes the complete per-category statistics. The dataset exhibits a pronounced 1:5 class imbalance (1,000 real vs. 5,000 fake videos in the five-category training configuration). Average video duration is 510 frames (≈17 seconds at 30 fps), with DeepFakeDetection containing significantly longer clips (734 frames average) at 1080p resolution. The estimated total dataset size is approximately 10.3 GB, with DFD contributing a disproportionate 6.64 MB average per video due to the higher resolution and longer duration.

D. Preprocessing Pipeline

All videos undergo an identical three-stage preprocessing pipeline applied during both training and inference to eliminate distribution mismatch:

1) Temporal Sampling: Exactly $T = 20$ frames are uniformly sampled from each video using the interval:

$$\Delta t = \frac{N_{\text{total}} - 1}{T - 1} \tag{1}$$

TABLE II: FaceForensics++ Dataset Statistics per Category

Category	Videos	Lbl	Avg Fr.	Avg MB	Res.
Original (Real)	1,000	REAL	509	1.85	Mixed
Deepfakes	1,000	FAKE	509	1.90	Mixed
Face2Face	1,000	FAKE	509	1.86	Mixed
FaceShifter	1,000	FAKE	509	1.83	Mixed
FaceSwap	1,000	FAKE	406	1.56	Mixed
NeuralTextures	1,000	FAKE	406	1.46	Mixed
DeepFakeDetect.	1,000	FAKE	734	6.64	1080p
TOTAL	**7,000**	—	**511**	**2.44**	—

where N_{total} is the total frame count. This ensures consistent temporal coverage regardless of video duration, while keeping BiLSTM input dimensionality constant. The $T = 20$ value balances temporal coverage with computational efficiency; ablation over $T \in \{10, 15, 20, 25\}$ showed diminishing returns beyond $T = 20$.

2) Face Detection and Cropping: Each sampled frame is processed by an OpenCV Haar Cascade frontal face detector with:

- Scale factor: 1.1 (10% step between detection scales)
- Minimum neighbors: 4 (reduces false positives)
- Minimum face size: 40×40 pixels

Detected bounding boxes are expanded by a 25% spatial margin in all four directions to preserve periocular and chin regions critical for boundary artifact detection. When no face is detected in a sampled frame, the last valid crop is reused (temporal smoothing); if no face is detected anywhere in the video, the full frame is used as fallback to avoid empty inputs.

3) Normalization and Serialization: Face crops are resized to 224×224 pixels using Lanczos resampling and normalized with ImageNet statistics:

$$x_{\text{norm}} = \frac{x_{\text{pixel}}/255 - \mu_{\text{IN}}}{\sigma_{\text{IN}}} \tag{2}$$

where $\mu_{\text{IN}} = [0.485, 0.456, 0.406]$ and $\sigma_{\text{IN}} = [0.229, 0.224, 0.225]$. Preprocessed sequences of shape $(T, C, H, W) = (20, 3, 224, 224)$ are serialized as NumPy `.npy` files, reducing per-epoch training time by approximately 65% versus on-the-fly extraction.

E. Class Imbalance Handling

The 1:5 real-to-fake imbalance is addressed by three complementary strategies: (1) **Weighted Random Sampling** assigns weights $w_{\text{real}} = 5$, $w_{\text{fake}} = 1$, ensuring mini-batches contain roughly equal class proportions; (2) **Label Smoothing** ($\varepsilon = 0.10$) distributes 10% of target probability mass to the non-target class, preventing overconfident predictions near the decision boundary and improving AUC-ROC; and (3) **Stratified Splitting** preserves the 1:5 ratio across all three partitions, ensuring evaluation metrics reflect true operational class distributions.

IV. Proposed System Architecture

The complete system architecture is depicted in Fig. 3. The pipeline consists of four sequential stages: (1) preprocessing and face extraction, (2) per-frame spatial feature extraction via EfficientNetB0, (3) temporal modeling of the feature sequence via Bidirectional LSTM, and (4) binary softmax classification. During inference, a user-uploaded video flows through the identical preprocessing and feature extraction pipeline before producing a REAL/FAKE verdict with confidence score.

Fig. 3: Complete system architecture showing the training flow (dashed black) and prediction flow (dashed red). Raw videos are preprocessed into face crop sequences, passed through the Deep Fake Detection module (CNN+BiLSTM), and evaluated. The trained model is exported and loaded for real-time prediction.

A. CNN Spatial Feature Extractor (EfficientNetB0)

EfficientNetB0 [10] is selected as the spatial feature extractor due to its compound scaling strategy, which simultaneously scales network depth, width, and resolution with fixed coefficients $(\alpha, \beta, \gamma) = (1.2, 1.1, 1.15)$. This multi-dimensional scaling produces richer multi-scale spatial representations than single-axis scaling approaches (e.g., VGG depth scaling or WideResNet width scaling), enabling detection of deepfake artifacts at multiple spatial frequencies simultaneously. EfficientNetB0 achieves 83% ImageNet top-1 accuracy with only 5.3M parameters and 0.39B FLOPs, offering an exceptional accuracy-efficiency trade-off.

The model is initialized with ImageNet pre-trained weights. The final classification head is replaced by a global adaptive average pooling layer that produces a **1,280-dimensional** feature vector per frame, as given by:

$$\mathbf{f}_t = \text{AvgPool}\ \text{EfficientNetB0}(\mathbf{x}_t) \in \mathbb{R}^{1280} \quad (3)$$

where $\mathbf{x}_t \in \mathbb{R}^{3\times224\times224}$ is the normalized face crop at time step t. All convolutional blocks are frozen except the last three (blocks 6–8), which are fine-tuned to adapt domain-specific facial features while preventing catastrophic forgetting of low-level ImageNet features. A dropout layer ($p = 0.3$) follows feature extraction for regularization. The identical EfficientNetB0 instance processes all 20 frames via shared weights (parameter tying across the temporal dimension), enabling the backbone to learn manipulation-specific features universally applicable to all positions in the temporal sequence.

B. Bidirectional LSTM Temporal Classifier

The sequence of 20 frame features $\{\mathbf{f}_t\}_{t=1}^{T}$ is assembled into a feature matrix $\mathbf{F} \in \mathbb{R}^{B\times T\times 1280}$ and fed into a two-layer Bidirectional LSTM. For each layer l and time step t, the BiLSTM computes:

$$\overrightarrow{\mathbf{h}}_t^{(l)} = \text{LSTM}_{\text{fwd}}^{(l)}\left(\mathbf{F}_t, \overrightarrow{\mathbf{h}}_{t-1}^{(l)}\right) \quad (4)$$

$$\overleftarrow{\mathbf{h}}_t^{(l)} = \text{LSTM}_{\text{bwd}}^{(l)}\left(\mathbf{F}_t, \overleftarrow{\mathbf{h}}_{t+1}^{(l)}\right) \quad (5)$$

$$\mathbf{h}_t^{(l)} = \left[\overrightarrow{\mathbf{h}}_t^{(l)} \,\|\, \overleftarrow{\mathbf{h}}_t^{(l)}\right] \in \mathbb{R}^{512} \quad (6)$$

where $\|$ denotes vector concatenation, and each directional LSTM uses 256 hidden units. The bidirectional design is crucial: the **forward LSTM** (Eq. 4) captures progressive facial changes and temporal drift accumulated by the synthesis process (e.g., GAN artifacts that compound over time), while the **backward LSTM** (Eq. 5) provides future-frame context that disambiguates frames appearing natural in isolation but revealing their synthetic origin through subsequent frames—texture resets, expression discontinuities, and lighting inconsistencies that only become apparent retrospectively.

The output at the final time step $t = T$ is:

$$\mathbf{o} = \text{Dropout}\ \text{LayerNorm}(\mathbf{h}_T^{(2)}) \in \mathbb{R}^{512} \quad (7)$$

and the final prediction is:

$$\hat{\mathbf{y}} = \text{Softmax}(\mathbf{W}\mathbf{o} + \mathbf{b}), \quad \mathbf{W} \in \mathbb{R}^{2\times512} \quad (8)$$

Layer Normalization applied before dropout stabilizes training by normalizing the hidden state distribution, which otherwise exhibits high variance in deep recurrent networks. The LSTM gates (forget, input, output, cell) use the standard formulation with sigmoid and tanh activations, providing selective memory retention appropriate for capturing subtle temporal inconsistencies in facial dynamics.

C. Model Architecture Summary

Table III provides the complete layer-by-layer architecture summary. The model has approximately 9.4M total parameters (5.3M in EfficientNetB0, 4.1M in BiLSTM and FC layers), with approximately 2.8M parameters actively trained during fine-tuning (blocks 6–8 plus BiLSTM and FC).

D. Training Strategy

1) Optimizer and Learning Rate Schedule: AdamW [18] is used with initial learning rate $\eta = 1 \times 10^{-4}$ and weight decay $\lambda = 1 \times 10^{-5}$. AdamW decouples ℓ_2 regularization from the adaptive gradient update, improving generalization compared to standard Adam with weight decay. Cosine Annealing schedules η from 10^{-4} to $\eta_{\min} = 10^{-6}$ over $T_{\max} = 30$ epochs, enabling aggressive exploration early in training and fine-grained convergence near the end.

TABLE III: CNN-LSTM Model Architecture Layer-by-Layer Summary

Layer	Configuration	Output Shape
Input Sequence	T=20 fr.×224×224×3	$(B, 20, 3, 224, 224)$
EfficientNetB0	Pretrained; blocks 6–8 fine-tuned	$(B\times20, 1280)$
AvgPool	Global adaptive pooling	$(B\times20, 1280)$
Dropout (CNN)	$p = 0.3$	$(B\times20, 1280)$
Reshape	Group frames by video batch	$(B, 20, 1280)$
BiLSTM Layer 1	hidden=256×2, drop.=0.5	$(B, 20, 512)$
BiLSTM Layer 2	hidden=256×2, bidirectional	$(B, 20, 512)$
Last Timestep	Select $t = T$ output	$(B, 512)$
LayerNorm	Normalize 512-d vector	$(B, 512)$
Dropout (LSTM)	$p = 0.5$	$(B, 512)$
FC Output	Linear(512→2) + Softmax	$(B, 2)$

2) Loss Function with Label Smoothing: The loss for sample i with label $y_i \in \{0, 1\}$ is:

$$L = -\sum_{c=0}^{1} (1 - \varepsilon)\, \mathbf{1}[y_i = c] + \varepsilon/2 \log \hat{y}_{i,c} \qquad (9)$$

with $\varepsilon = 0.1$. Smoothing distributes 5% of probability mass to the non-target class, preventing overconfident predictions on ambiguous boundary samples and directly improving AUC-ROC calibration.

3) Mixed Precision and Gradient Management: PyTorch Automatic Mixed Precision (AMP) uses float16 for forward pass matrix multiplications while maintaining float32 master weights, reducing GPU memory by ≈40% and accelerating throughput by 30–50%. This enables batch size 8 on a single 15.6 GB NVIDIA Tesla T4. GradScaler prevents float16 underflow during backpropagation. Gradient clipping (max norm=1.0) prevents exploding gradients in deep recurrent layers. Early stopping with patience=7 epochs saves and restores the best validation checkpoint.

Table IV summarizes all training hyperparameters for reproducibility.

TABLE IV: Complete Training Hyperparameter Configuration

Hyperparameter	Value
Optimizer	AdamW
Learning Rate (η)	1×10^{-4}
LR Scheduler	CosineAnnealing (T_{max}=30, η_{min}=10^{-6})
Weight Decay (λ)	1×10^{-5}
Batch Size	8
Label Smoothing (ε)	0.10
Early Stop Patience	7 epochs
Max Epochs	30
Frames per Video (T)	20
Face Crop Resolution	224×224 pixels
Haar Cascade Params	scale=1.1, minNeighbors=4, minSize=40×40
Bounding Box Margin	25%
CNN Frozen Blocks	Blocks 1–5
CNN Fine-tuned Blocks	Blocks 6–8
CNN Dropout	0.30
LSTM Dropout	0.50
BiLSTM Hidden Units	256 per direction (512 total)
Dataset Split	70% / 15% / 15% (stratified)
Imbalance Strategy	Weighted Random Sampling
Training Platform	Kaggle NVIDIA Tesla T4 (15.6 GB VRAM)
Mixed Precision	PyTorch AMP (float16 compute, float32 weights)
Gradient Clipping	Max norm = 1.0

V. Experimental Results and Discussion

A. Evaluation Protocol

All models are evaluated on the held-out 15% test split, withheld throughout model development and hyperparameter tuning. Five metrics are computed: Accuracy, Precision (macro), Recall (macro), F1-Score (macro), and AUC-ROC. All baselines share identical preprocessing, dataset splits, and hardware to ensure fair comparison. Statistical significance is assessed over three independent training runs with different random seeds; reported values are means.

B. Ablation Baselines

Three ablation baselines isolate each component's contribution:

1) **CNN Only:** Standalone EfficientNetB0 frame classifier; frame-level softmax predictions are averaged over the 20-frame sequence to produce a video prediction. No temporal modeling.
2) **LSTM Only:** Raw-pixel BiLSTM without CNN feature extraction. High-dimensional pixel sequences prevent effective spatial learning.
3) **CNN + Unidirectional LSTM:** Proposed architecture with a standard (forward-only) LSTM replacing the bidirectional variant. Ablates the value of reverse temporal context.

TABLE V: Comparative Performance on FaceForensics++ Test Set

Method	Acc.	Prec.	Rec.	F1	AUC
CNN Only	84.3%	83.7%	82.9%	83.3%	0.901
LSTM Only	71.2%	70.8%	70.1%	70.4%	0.763
CNN + Uni-LSTM	88.6%	87.9%	87.5%	87.7%	0.941
CNN+BiLSTM	**92.4%**	**91.8%**	**91.6%**	**91.7%**	**0.967**

The proposed CNN–BiLSTM achieves 92.4% accuracy: +8.1 points over CNN-only, +3.8 points over unidirectional

LSTM, and +21.2 points over the LSTM-only baseline. The AUC-ROC of 0.967 confirms excellent discriminability across all classification thresholds. These margins confirm the indispensable contribution of each architectural element.

C. Per-Manipulation Performance Analysis

Table VI reports detection recall per manipulation type, revealing important differences in difficulty across forgery techniques.

TABLE VI: Per-Manipulation-Type Detection Recall (%)

Manipulation	CNN Only	Uni-LSTM	BiLSTM
Deepfakes	88.2	91.4	**95.3**
Face2Face	83.1	87.6	**91.8**
FaceShifter	79.4	84.2	**89.6**
FaceSwap	87.5	90.8	**94.2**
NeuralTextures	83.7	86.3	**92.1**
Average	84.4	88.1	**92.6**

Deepfakes and FaceSwap exhibit the highest recall (95.3% and 94.2%) due to visible boundary artifacts at the face perimeter captured reliably by EfficientNetB0's multi-scale spatial analysis. Face2Face and NeuralTextures show moderate difficulty (91.8% and 92.1%); their primary artifacts are subtle color and texture inconsistencies where the BiLSTM provides the greatest marginal benefit over CNN-only (+8.7 and +8.4 points respectively). FaceShifter is the most challenging type (89.6%), as adaptive identity embedding integration produces fewer boundary artifacts and higher perceptual quality, requiring heavier reliance on temporal inconsistencies for detection.

D. Confusion Matrix Analysis

Fig. 4 shows the confusion matrix on 600 test videos (300 real, 300 fake). The model correctly classifies 276/300 fake samples (92.0% recall) and 278/300 real samples (92.7% recall), with only 24 false negatives and 22 false positives.

Fig. 4: Confusion matrix on the FF++ test set. The near-symmetric error distribution (24 FN vs. 22 FP) confirms no systematic class bias, validating the Weighted Random Sampling strategy used to address the 1:5 class imbalance.

The near-symmetric error distribution (24 FN vs. 22 FP) confirms the absence of systematic class bias—a direct result of Weighted Random Sampling. In deployment, false negatives carry greater societal harm (undetected deepfakes may spread disinformation). The 92.0% fake recall represents an operationally sound balance.

E. Ablation Study Summary

- **Spatial features are essential:** LSTM-only drops by 21.2 points. Raw-pixel inputs are too high-dimensional for effective temporal reasoning; meaningful temporal patterns cannot be extracted without prior spatial compression via CNN.
- **Temporal modeling is critical:** CNN-only drops by 8.1 points. Many manipulation artifacts (e.g., expression discontinuities, GAN texture resets) are only detectable through cross-frame comparison.
- **Bidirectionality provides consistent gains:** Unidirectional LSTM costs 3.8 accuracy and 2.6 AUC points vs. BiLSTM across all five manipulation types. Reverse temporal context consistently disambiguates frames appearing natural in isolation. The per-manipulation gains are largest for Face2Face (5.1 points) and NeuralTextures (4.8 points), where temporal cues dominate spatial ones.

F. Discussion

EfficientNetB0's compound scaling extracts artifacts simultaneously at multiple spatial scales—global color inconsistencies at low scales (detectable in early MBConv layers) and fine pixel-level blending boundaries at high scales (detectable in blocks 6–8). This multi-scale feature hierarchy is especially effective for Face2Face and NeuralTextures, where color and texture inconsistencies are the primary forensic signals, and for Deepfakes and FaceSwap, where geometric boundary artifacts are dominant. Label smoothing with $\varepsilon = 0.10$ prevents overconfident probability assignments on ambiguous boundary samples, contributing directly to the strong AUC-ROC of 0.967. Mixed-precision training enabled the required memory efficiency without sacrificing accuracy through float32 accumulation for numerically sensitive reduction operations.

VI. Real-Time Web Application

A. System Architecture

A Flask-based web application operationalizes the trained CNN–BiLSTM model for real-world use, providing an end-to-end inference pipeline accessible via standard browser without requiring local deep learning library installation. The backend comprises three modular layers:

1) **File Upload Layer:** Handles video reception, MIME-type validation (MP4, AVI, MOV, MKV), file size enforcement ($\leq$500 MB), and secure session-scoped temporary storage with automatic cleanup.
2) **Inference Pipeline:** Executes the complete preprocessing pipeline (Haar Cascade face detection, 25% padding, $T = 20$ sampling, ImageNet normalization), followed by CNN feature extraction and BiLSTM classification. The pipeline is architecturally identical to training, eliminating distribution mismatch.

3) **Response Layer:** Returns structured JSON payloads to the front-end for interactive display. The REST endpoint (`POST /predict`) accepts multipart form data, enabling downstream integration.

B. User Interface and Output

The responsive dark-themed single-page application provides:

1) **Binary verdict:** REAL ✓ (green) / FAKE × (red), with color-coded visual emphasis for immediate interpretability.
2) **Confidence score:** 0–100% derived from the Softmax output probability of the predicted class.
3) **Probability breakdown:** Per-class probabilities displayed as animated progress bars, communicating prediction uncertainty to the user.
4) **Processing time:** Wall-clock inference time in seconds, providing operational transparency.

C. Performance and Scalability

Average end-to-end inference time is ≈3.2 s per 30-second MP4 clip on the Tesla T4. Inference time scales approximately linearly with video duration due to fixed $T = 20$ sampling that keeps BiLSTM input dimensionality constant—a key design advantage over variable-frame approaches. The JSON API enables integration with content moderation platforms, fact-checking tools, and investigative journalism pipelines. For production-scale deployment, TorchScript compilation reduces Python overhead by 30%, ONNX export enables hardware-agnostic inference, and GPU batch queuing via message brokers (Redis/Celery) supports concurrent multi-user workloads.

VII. Limitations and Ethical Considerations

A. Technical Limitations

- **Dataset scope:** Trained and evaluated exclusively on FF++ c23. Cross-dataset generalization to DFDC, Celeb-DF, and WildDeepfake [15] remains unverified; domain shift from different GAN architectures and compression levels may degrade accuracy.
- **Compression sensitivity:** Heavy transcoding typical of social media (c40, H.265) may suppress the boundary artifact evidence relied upon by EfficientNetB0, requiring compression-aware training augmentation.
- **Non-frontal faces:** Haar Cascade performance degrades on profile views (>45° yaw) and extreme tilt angles, falling back to full-frame crops that introduce irrelevant background features. DLIB landmarks or deep face detectors would improve robustness.
- **Fixed temporal sampling:** Uniform $T = 20$ may miss brief high-value manipulation artifacts in long videos; quality-driven or attention-guided adaptive frame selection would improve detection recall.
- **No audio modality:** Lip-sync inconsistencies and voice-face identity mismatches are not exploited, leaving significant forensic signal unrealized.
- **Adversarial adaptation:** Published detector architectures enable adversaries to craft evasion-optimized deepfakes via detector-aware training, perpetuating the detection-generation arms race.

B. Ethical Considerations

Dual-use risk: Publishing precise detector architectures enables adversarial adaptation via gradient-based evasion attacks. The research community must balance scientific transparency with operational security through responsible disclosure practices and delayed architecture release. **False positive risk:** A 7.3% false positive rate at platform scale could harm authentic content creators by incorrectly flagging legitimate content. Production systems must mandate human expert review for borderline predictions and communicate uncertainty quantitatively. **Demographic bias:** FF++ lacks demographic balance documentation; error rates across skin tones, ages, and genders are uncharacterized. Comprehensive fairness auditing across demographic subgroups is a prerequisite to responsible production deployment, consistent with emerging AI fairness regulation. **Privacy:** Face crops extracted during inference must not be retained beyond the inference session, consistent with GDPR Article 5, India's Digital Personal Data Protection Act (PDPA), and equivalent privacy regulations. Users must be informed of face extraction via explicit consent mechanisms.

VIII. Conclusion and Future Work

A. Conclusion

This paper presented a hybrid CNN–BiLSTM framework that fundamentally addresses the limitation of existing frame-level deepfake detectors through tightly integrated spatial and temporal learning. EfficientNetB0 extracts multi-scale per-frame spatial features—boundary artifacts, texture seams, and color anomalies—via compound scaling that simultaneously optimizes spatial resolution, depth, and channel width. The Bidirectional LSTM models cross-frame temporal inconsistencies inherent to algorithmic face synthesis: forward context captures progressive temporal drift in GAN-generated sequences, while backward context provides retrospective disambiguation of individually ambiguous frames.

Evaluated on FaceForensics++ across five primary manipulation types with 7,000 videos, the system achieves **92.4% accuracy**, **91.7% macro F1-Score**, and **0.967 AUC-ROC**. The controlled ablation study conclusively demonstrates that each architectural component contributes indispensable information: CNN-only accuracy is 8.1 points lower; LSTM-only accuracy is 21.2 points lower; unidirectional LSTM is 3.8 points lower. The per-manipulation analysis reveals FaceShifter as the most challenging forgery type (89.6% recall), informing future architectural priorities. The near-symmetric confusion matrix validates the class-imbalance mitigation strategy, and the Flask web application confirms practical deployment viability without end-user technical expertise.

B. Future Work

1) **Transformer temporal attention:** Cross-frame self-attention over the feature sequence for selective weighting

of discriminative temporal positions, replacing fixed last-timestep extraction.

2) **Cross-dataset generalization:** Systematic evaluation on DFDC, Celeb-DF, and WildDeepfake [15] for robustness characterization and domain adaptation.
3) **Audio-visual fusion:** Synchronized lip-sync analysis via wav2vec features and voice-face identity consistency for multi-modal detection.
4) **Edge model compression:** Knowledge distillation from CNN–BiLSTM teacher to a compact student network for mobile and IoT deployment at sub-1B FLOPs.
5) **Explainable forensics:** Grad-CAM [18] and Integrated Gradients for spatial saliency and temporal attention weight visualization, supporting forensic interpretability in legal and investigative contexts.
6) **Adversarial robustness:** Certified defenses against evasion attacks and adversarial training against detector-aware deepfake generators.
7) **Demographic fairness:** Systematic error auditing across demographic subgroups with demographically balanced evaluation sets; integration of fairness-aware loss functions to equalize per-subgroup error rates.

ACKNOWLEDGMENT

The authors sincerely thank **Mrs. D. Mahalakshmi, M.E., (PhD)**, Assistant Professor, Department of Information Technology, AVC College of Engineering, Mannampandal, for her invaluable guidance, continuous encouragement, and expert mentorship throughout this project. The authors also acknowledge the Department of Information Technology, AVC College of Engineering (Anna University Affiliated, Regulation 2021), for providing the computational and academic resources to conduct this research. This work was carried out as part of the B.Tech Final Year Project, IV Year / VII Semester, 2022–2026.

REFERENCES

[1] M. Javed, M. A. Khan, S. U. Rehman, and S. A. Malik, "Real-time deepfake video detection using eye movement analysis with a hybrid deep learning approach," *Electronics*, vol. 13, no. 15, p. 2947, 2024.
[2] M. Alrashoud, "Deepfake video detection methods, approaches, and challenges," *Alexandria Engineering Journal*, vol. 125, pp. 265–277, 2025.
[3] V. K. Sharma, R. Garg, and Q. Caudron, "A systematic literature review on deepfake detection techniques," *Multimedia Tools and Applications*, vol. 84, no. 20, pp. 22187–22229, 2025.
[4] M. S. Rana and A. H. Sung, "Deepfake Stack: A deep learning-based multi-classifier for deepfake detection," *IEEE Access*, vol. 12, pp. 14522–14535, 2024, doi: 10.1109/ACCESS.2024.3355812.
[5] S. Sharma and R. Kumar, "Robust face detection and alignment for deepfake mitigation," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 210–224, 2024.
[6] M. Hossain, A. Rahman, S. Al-Amin, and M. S. Islam, "Visual deepfake detection: Review of techniques, tools, limitations, and future prospects," *IEEE Access*, vol. 13, pp. 2410–2435, 2025.
[7] S. Dasgupta, A. Roy, P. Banerjee, and S. Chakraborty, "Attention-enhanced CNN for high-performance deepfake detection: A multi-dataset study," *IEEE Access*, vol. 13, pp. 101980–101995, 2025.
[8] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to detect manipulated facial images," in *Proc. IEEE/CVF ICCV*, Seoul, Korea, 2019, pp. 1–11.
[9] D. Gue´ra and E. J. Delp, "Deepfake video detection using recurrent neural networks," in *Proc. IEEE 15th AVSS*, Auckland, New Zealand, 2018, pp. 1–6.
[10] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th ICML*, Long Beach, CA, USA, 2019, pp. 6105–6114.
[11] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. ICLR*, Virtual, 2021.
[12] IEEE ICRITO, "DeepFake video detection using machine learning and deep learning techniques," in *Proc. IEEE ICRITO*, Noida, India, 2024.
[13] F. Matern, C. Riess, and M. Stamminger, "Exploiting visual artifacts to expose deepfakes and face manipulations," in *Proc. IEEE WACV Workshops*, Waikoloa, HI, USA, 2019, pp. 83–92.
[14] 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 CVPR*, Seattle, WA, USA, 2020, pp. 5001–5010.
[15] B. Zi, M. Chang, J. Chen, X. Ma, and Y.-G. Jiang, "WildDeepfake: A challenging real-world dataset for deepfake detection," in *Proc. ACM MM*, Virtual, 2020, pp. 2382–2390.
[16] Y. Liu, A. Jourabloo, and X. Liu, "Learning rich features for image manipulation detection," in *Proc. IEEE/CVF CVPR*, Salt Lake City, UT, USA, 2018, pp. 1053–1061.
[17] H. Zhao, W. Zhou, D. Chen, T. Wei, W. Zhang, and N. Yu, "Multi-attentional deepfake detection," in *Proc. IEEE/CVF CVPR*, Nashville, TN, USA, 2021, pp. 2185–2194.
[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF CVPR*, Las Vegas, NV, USA, 2016, pp. 770–778.