A REGULARIZED SPATIO-TEMPORAL HYBRID RESNEXT50-LSTM NETWORK FOR ROBUST DEEPFAKE VIDEO DETECTION

Vaishali Bankar[1], Diksha Pawar[2], Dr. Pravin Yannawar[3]

[1]Reasearch Scholar 1,
Dept. CSIT, Dr. Babasaheb Ambedkar Marathwada University, Chh. Sambhajinagar, Maharashtra, India, vaishali.bankar2@gmail.com

[1]Reasearch Scholar2
Dept. CSIT, Dr. Babasaheb Ambedkar Marathwada University, Chh. Sambhajinagar, Maharashtra, India,diksha.salunke97@gmail.com

[3]Professor & Head of the Department 3
Dept. CSIT, Dr. Babasaheb Ambedkar Marathwada University, Chh. Sambhajinagar, Maharashtra, India, plyannawar.csit@bamu.ac.in,

***Abstract*:**
Generative artificial intelligence has experienced rapid growth in recent years, as have some of the most advanced algorithms for creating deep fake images, videos, and audio content. Generative AI and some of the latest deepfake image, video and voice creation algorithms are capabilities that are fast becoming a problem for digital security, public trust and media forensics. Generally, traditional face manipulation detection approaches are based on static features, extracted frame by frame, and often suffer from the problem of generalization from temporal inconsistencies within the video sequences. To tackle this challenge, we provide an efficient spatio-temporal hybrid deep learning system that adopts the truncated ResNeXt50-32x4d spatial network and a recurrent Long Short-Term Memory (LSTM) network. We used a carefully selected set of 1,420 high-resolution video sequences (814 manipulated, 606 authentic) following a common 10-frame sequential extraction and face-cropping process to test our model. The proposed system through the Adam optimizer (1e-5) with, reached a peak validation accuracy of 90.14% (validation loss: 0.3273) at epoch 12 during optimization. After training for 20 epochs, however, it was observed that the validation accuracy had significantly lowered to 0.8380 or 83.80% (validation loss: 0.4888) as a result of overfitting. In order to counteract this performance degradation, we propose and use an Early Stopping capacity regulator (patience = 3, delta = 0.005). The early stopping controller stopped training at Epoch 16 to preserve the best parameters at Epoch 13 (validation loss = 0.2949) restoring the optimal condition of the system at the best classification performance. We conclude this work with the following observations: Dynamic sequence regularization is an important technique for ensuring the trustworthiness of temporal digital forensics.

***Keywords*: Deepfake detection, ResNeXt50, LSTM, FaceForensics ++, Early Stopping, etc.**

(Article history: Received: Oct 31,2025 and accepted Jan, 30,2026)

I. INTRODUCTION

With the recent rise of the democratization of powerful "deep" generative models, it has been possible to generate hyper-realistic "deepfake" digital manipulations [1]. The effects are achieved using the advanced machine learning approach—Generative Adversarial Networks (GANs) and diffusion-based architecture, allowing adversaries to easily foresee facial identity, modify facial expressions, and create realistic speech. Although these technologies provide creative industries and virtual reality applications, there are serious implications for digital security, social stability and political integrity when used maliciously. The high crux of the frequently deceptive media's potential to be easily fabricated and spread on the social media platforms underscores the need to provide efficient and redundant digital forensic verification tools that are automatically given [3].

Early deepfake detection frameworks were mostly spatial artifacts of individual frames based. Such architectures identify indicators for local manipulation using regular CNNs, such as blending boundaries, color irregularities, and inconsistencies in the relative texture around facial landmarks [4]. However, in spite of the initial success of all spatial detection mechanisms, there are large drawbacks in these strategies. The first is that they are very sensitive to compression artifacts, noise, and low

resolution from which high frequency spatial anomalies are likely to be masked. Second, generative synthesis models have become rapidly developed and generate single frames with high spatial fidelity which makes them very hard to detect on frame-by-frame level [5].

Today's digital forensics must take advantage of the sequential dynamism of video, to surface challenges for static image analysis. Manipulated images might look good on a single frame, but when they are played outurside of time, they may produce minor inconsistencies [1]. Humans have highly coordinated and complex spatio-temporal behaviours such as micro-expressions, particular eye-blinking rates and physiological blood-flow variations. Synthesized videos are less coherent temporally from frame to frame and hence exhibit temporal jittering, geometric warping, and unnatural facial transitions. These temporal variations have to be detected using a paradigm of a continuum shifting from a purely spatial model to networks unifying space and time complexity [6].

In this article, we present a hybrid deep learning architecture system equipped with a temporal recurrent processing unit and a spatial backbone unit which is truncated. The system models the multi-frame sequential dynamics of the faces by learning the deep spatial features of the sequential video frames using a modified ResNeXt50-32x4d network, and passing these features through a Long Short-Term Memory (LSTM) network [7]. Important, we recognise a particular generalization decay within the later training phase, when the network's performance on the training subset becomes too high. To avoid this, we use a mathematically well-structured algorithm for Early Stopping regularization [8]. This capacity regulator monitors the loss trend on the validation data set continuously, and when it detects that the loss of the validation data set is not getting worse, the model automatically halts back-propagation, and then reverts to optimal generalization accuracy before the model begins to overfit and to become less accurate on the validation data set [8].

II. Literature Review and Related Work

2.1. Spatial Frame-by-Frame Deepfake Classifiers

Most algorithms developed for early detection of deep fakes are based on finding anomalies in one particular frame. These models use state-of-the-art Convolutional Neural Network (CNN) networks such as ResNet, EfficientNet and DenseNet for identifying microscopic visual artifacts [9]. Examples of such artifacts are boundary misalignment, inconsistent and abnormal eye blinking patterns, unnatural shading in the facial areas, inconsistencies in skin texture, and discrepancies in blending surrounding the nose and mouth regions.
Spatial classifiers have a strong classification capability on high-resolution static images; however, they suffer from a number of inherent weaknesses when process a video stream:

a. ***Susceptibility to Compression Noise:***Video codes like H.264, MPEG-4 have a compression algorithm called lossy which aggressively discards high frequency spatial frequencies. As a result, the spatial classifiers have high false-negative rates on compressed video sequences [10] due to the smoothing of fine facial boundaries and the texture artifacts.
b. ***Temporal Inconsistency and Flicker:***These models (frame-wise models) are insensitive to temporal jitter and inconsistencies that appear at pixel level in consecutive frames. It may be possible for a fake sequence to pass through purely spatial barriers easily if the images are very similar with "unnatural" movement changes [11].

2.2. Critique of Temporal Modeling Methodologies

To address the limitations of spatial classifiers, researchers introduced temporal analysis approaches, primarily categorized into Optical Flow methods and Recurrent Neural Networks (RNNs).

a) Optical Flow Models

Optical flow tracking algorithms sequentially calculate the shifts of the pixels in order to measure the muscle actions (micro-expressions), the motion of the eyes and the motion of the head [12]. Architectures based on optical flow are very descriptive, but have significant drawbacks:

Extreme Computational Overhead: Calculating pixel-level dense optical flow vectors (e.g, via Farneback or TV-L1 algorithms) is extremely intensive, rendering real-time inference unfeasible [12].
Trcking Failures: Optical flow relies heavily on pixel intensity constancy and spatial coherence. Sudden changes in illumination, rapid head rotations, or self-occlusion like a hand passing over the face lead to catastrophic gradient tracking failures, corrupting the downstream feature sequences [12].

b) Standard Recurrent Neural Networks (RNNs)

Sequential features over time were mapped in standard RNNs and simple Gated Recurrent Units (GRUs) that have been stacked on top of CNN backbones [13] . Considerable mathematical flaws in standard RNN designs, however, are that:
Vanishing and Exploding Gradients:During backpropagation through time (BPTT), the gradients of the loss function are multiplied by the recurrent weight matrices, which can lead to vanishing and exploding gradients. These gradients do not

scale for sequences that are longer, and vanish to zero or grow to infinity, so that the model cannot capture any long-term dependencies [13].

Memory Decay:Simplicity of the recurrent cells leads to Memory Decay, where adjacent frames are more important than global temporal consistency, that is, simple recurrent cells lose information from previous relevant frames over longer frame sequences [13].

2.3. Key Research Gaps and the Proposed Hybrid Solution

A major research gap in the literature lies in sequence length stability and representation balance [14]. The old techniques were found to require too many frames (e.g., sequence lengths of 50-100 frames) so that the memory required by GPUs is very large, temporal noise is even, and such training poses a risk of overfitting during backpropagation through time. On the other hand, fewer the number of frames, the less likely they will be able to model dynamic micro-motions.
To overcome this limitation, a proposed ResNeXt50 + LSTM model is presented, which now introduces an optimization of high-capacity spatial feature extraction and powerful long-time time-series processing:

a. ***High-Capacity Feature Extraction:***In particular, we propose the use of ResNeXt50, which leverages grouped convolutions (cardinality 32x4d) to achieve high-capacity feature extraction and efficiency in the number of parameters used compared with classic ResNet designs, allowing to capture subtle face variations without computational overload. [15]
b. ***Long-Term Sequence Stability via LSTM:***Long Short-Term Memory (LSTM) cells have an explicit gating mechanism, including the input and forget gates and the output gate, with all these gating mechanisms controlling the flow of memory within the network. This mathematical encoding by itself does not cause the vanishing gradient problem, and is able to effectively learn temporal temporal transitions within a short sequence length of only 10 frames without losing ground [7].
c. ***Efficient Dimensionality Mapping:***Efficient Dimensionality Mapping maps directly feed images to the recurrent network. This representation scale significantly helps to reduce the sequence processing latency, making it a very scalable structure well-suited for Scopus quality real-time digital forensics systems [16].

III. METHODOLOGY

3.1. Dataset and Preprocessing Pipelines

3.1.1. Dataset Characteristics and Class Balance

We have developed a very comprehensive dataset that includes metadata from around the globe (as per global metadata tracking [17] and FaceForensics++ (FF_REAL_Face_only_data) corpus. The dataset consists of 1,420 video sequences Ntotal, which have been processed. The binary class distribution has a small imbalance for typical real-world challenges:

FAKE (Class 0): The count of the Nfake = 814 videos in the total corpus represents the percentage of 57.32%.

REAL (Class 1): Nreal = 606 (42.68%) of the total corpus.

The corpus is split following a 80/20 ratio, and a non-overlapping partition between training and validation is employed to guarantee that the data are evaluated exactly, while minimizing any risk of data leaks.

Training Set (80%):Ntrain=1,136 videos, consisting of 644 FAKE sequences and 492 REAL sequences.

Validation Set (20%):Nval=284 videos, consisting of 170 FAKE sequences and 114 REAL sequences [5].

3.1.2. Spatial Preprocessing and Face Extraction Pipeline

The raw input video sequences are then put through multiple pre-processing stages, consisting of region of interest detection through face alignment and down sampling, and tensoring of all the face regions to learn their deep spatial representations. The processing pipeline is tabulated below:
Uniform Sequence Sampling: Given a video V consisting of F frames, we extract a subset of T=10 frames. Instead of subsampling continuous frames that include redundant temporal information, frames are sampled uniformly over time [18]. The indices of the sampled frames are given by:

$$ft=\lfloor t\times FT\rfloor for\ t\in\{0,1,\ldots,T-1\} \qquad (1)$$

Facial Detection and Localization:Face detection model is based on Histogram of Oriented Gradients (HOG) with Linear SVM (Support Vector Machine), which is a part of the face recognition library is used for every frame $I\in RH\times W\times 3$ [19]. This will return the coordinates of the bounding box (ymin,xmax,ymax,xmin) of the main facial area in the image. The cropped face image Iface is defined as:

$$Iface=I[ymin:ymax,xmin:xmax,:]\in RHc\times Wc\times 3 \qquad (2)$$

Spatial Resizing:In order to have uniform input size for all samples in a batch to be processed, the cropped face regions are resized as shon in fig .1 to the target spatial dimension of 112×112 pixels by bilinear interpolation:

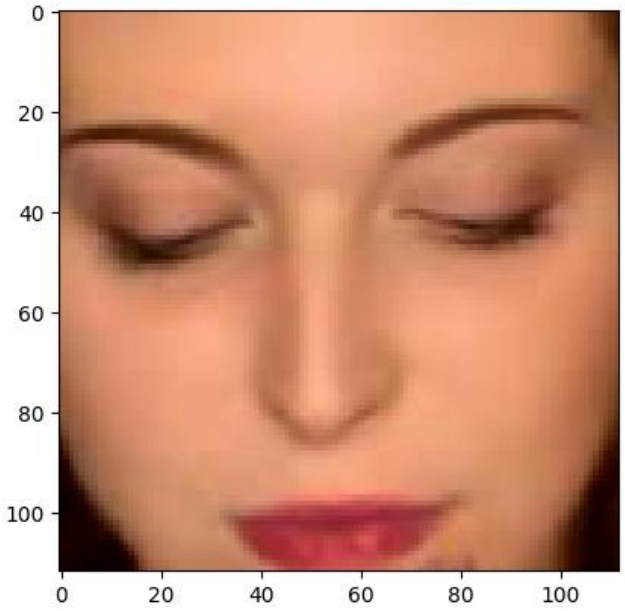


Fig.1 Resized face fram

$$I_{resized} = BilinearResize(I_{face}, 112, 112) \in R^{112\times112\times3} \quad (3)$$

Pixel Normalization: Each pixel coordinate (u,v) in the resized channel c∈{R,G,B} of Iresized is normalized to stabilize gradient flow and optimize training convergence:

$$I'(u,v,c) = I_{resized}(u,v,c) - \mu_c \sigma_c \quad (4)$$

With the channel-wise ImageNet priors defined as:

$$\mu = [0.485, 0.456, 0.406], \sigma = [0.229, 0.224, 0.225] \quad (5)$$

3.1.3. Spatio-Temporal Sequence Formulations

After the spatial transformations of the frames and from different perspectives, the normalized frames I^t are stored in the temporal direction sequentially. This operation builds a spatio-temporal tensor to represent the video:

$$X = [I'_0, I'_1, \ldots, I'_{T-1}] \in R^{T\times C\times H\times W} \quad (6)$$

where T=10 is the length of the sequence, C=3 corresponds to the color channels, and H=W=112 are the dimensions in space (normalized). The input data for our network given a batch size of B, is organized as a 5D tensor:

$$X \in R^{B\times T\times C\times H\times W} \quad (7)$$

The mathematical formulation is used to extract information both for independent high-frequency facial features in individual frames and lower-frequency facial dynamics in the video sequence, and act as the mathematical foundation of the following ResNeXt50-LSTM architecture [5].

3.2. Hybrid Architecture and Tensor Flow Formulations

3.2.1. Mathematical Foundations of Spatio-Temporal Hybrid Layers

A. ResNeXt50 Grouped Convolutions

This ResNeXt family uses grouped convolutions [5] to add split-transform-merge strategy to the traditional ResNet architectures, denoted as ResNeXt101 and ResNeXt269. It is parameterized in the cardinality C (the set size of transformation) and bottleneck width d.

For a given input vector x, a standard bottleneck residual block is replaced by:

$$y=x+\sum i=1CTi(x) \quad (8)$$

Where Ti(x) is a series of transformations (usually 1×1→3×3→1×1 convolutions). In the case of resnext50_32x4d used in this architecture:

i. ***Cardinality (C):*** 32 groups.
ii. ***Bottleneck width (d):*** 4 dimensions per group.

This architecture dramatically increases representation capacity without exponentially increasing computational complexity (floating-point operations) or parameter counts [15].

B. Long Short-Term Memory (LSTM) Sequential Nodes

The spatial features are fed into a recurrent LSTM network [7] to receive temporal dependencies between the sequences of sample video frames. Let sϵ(0, 1, ..., 10) denote a feature sequence xt for time-step t, the hidden state ht and cell state ct are updated via the following gating mechanisms:

Forget Gate (ft):Controls how much of the prior cell state to discard.

$$ft=\sigma(Wf\cdot[ht-1,xt]+bf) \quad (9)$$

Input Gate (it): Decides which new information to store in the cell state.

$$it=\sigma(Wi\cdot[ht-1,xt]+bi) \quad (10)$$

Candidate Cell State (c~t):Generates a vector of new candidate values.

$$c\sim t=tanh(Wc\cdot[ht-1,xt]+bc) \quad (11)$$

Cell State Update (ct):Combines forget and input gates to update memory.

$$ct=ft\odot ct-1+it\odot c\sim t \quad (12)$$

Output Gate (ot):Determines the next hidden state output.

$$ot=\sigma(Wo\cdot[ht-1,xt]+bo) \quad (13)$$

Hidden State (ht):$ht=ot\odot tanh(ct)$ (14)

The sigmoid function is denoted by σ, and ⊙ denotes the element-wise multiplication. W and b are the learned weights and biases.

3.2.2. Multi-Dimensional Tensor Flow Trace Table

Beloware the step-by-step transformations applied to a 5D sequence tensor of size [B,10,3,32,32] layer-by-layer, to yield a 2D binary logit:

Step	Layer Name	Mathematical Operator	Input Shape	Output Shape	Description
0	**Input Video Batch**	-	-	[B, 10, 3, 112, 112]	Batch of B videos with 10 sequential frames of 112×112 RGB pixels.
1	**Batch Collapse**	x.view(B * T, C, H, W)	[B, 10, 3, 112, 112]	[B * 10, 3, 112, 112]	Flattens sequence axis so each frame is treated as an independent spatial image.
2	**ResNeXt50 Backbone**	self.model(x)	[B * 10, 3, 112, 112]	[B * 10, 2048, 4, 4]	Extracts deep spatial convolution features from the truncated ResNeXt50 blocks.
3	**Spatial Pooling**	self.avgpool(f map)	[B * 10, 2048, 4, 4]	[B * 10, 2048, 1, 1]	Downsamples spatial dimensions (4×4) to unified channel-wise features via Adaptive Average Pooling.
4	**Sequence Reconstruct**	pooled.view(B , 10, 2048)	[B * 10, 2048, 1, 1]	[B, 10, 2048]	Reconstructs sequence length axis for temporal LSTM modeling.
5	**LSTM Sequence Model**	self.lstm(x)	[B, 10, 2048]	[B, 10, 2048]	Unidirectional LSTM encodes temporal dependencies across the 10 timesteps.
6	**Temporal Aggregation**	torch.mean(x_ lstm, dim=1)	[B, 10, 2048]	[B, 2048]	Collapses sequence length axis by calculating the mean vector across timesteps.
7	**Dropout Regularization**	self.dp(mean_ pooled)	[B, 2048]	[B, 2048]	Zeroes out 40% of features at random to prevent late-stage model overfitting.
8	**Fully Connected Class**	self.linear1(dp _out)	[B, 2048]	[B, 2]	Projects 2048-dimensional features to 2 binary logits ([Fake, Real]).

3.2.3. Custom Classifier Head & Regularization Operations

Two operations are used in the linear projection header:
Dropout Regularization(nn.Dropout(0.4)): Formulated as y=11−pa·x where a∼Bernoulli(1−p) and probability p=0.4. This reduces co-dependency between neurons.
LeakyReLU activation(nn.LeakyReLU()):Incorporated within the architecture to allow gradients to flow backwards even for negative activations, resolving dead-neuron issues [20]. Mathematically defined as:

$$LeakyReLU(x)=max(0,x)+\alpha min(0,x) where\ \alpha=0.01 \qquad (15)$$

3.3. Early Stopping as a Capacity-Regulating Constraint

3.3.1. Theoretical Foundations

In deep learning, "overfit" is when a high-capacity model learns the high frequency noise and sample-specific patterns from a training set, as a result of which it performs well on the training data but poorly on data it has never encountered before. A notable regularization technique that is used regularly is early stopping, which is an active regularization that dynamically stops the network while changing neither the objective loss function nor the other regularization techniques [8].In gradient descent: The trajectory starts with a value θ0, and as gradient descent proceeds it moves in the parameter space RD to a local empirical minimum. In early stopping, training is stopped when validation error starts increasing. This will help limit the optimization space to a smaller region centered around the initialization point, limiting the ability of the model to fit in complex patterns of noise. It is mathematically similar to L2 regularization (weight decay) with the stopping time t corresponding to the inverse of λ (the weight decay parameter) [8].

3.3.2. Mathematical Operational Formulation

Suppose that Lval(t), for t in {1,2,…., T}, represents the validation loss at epoch t. The minimum historical error until epoch t is the optimal validation performance:

$$L*val(t)=min1\le\tau\le tLval(\tau) \qquad (16)$$

We introduce a threshold parameter Δ>0 and a parameter p, p∈Z+ preventing to stop because of transient fluctuations or noise in the optimization trajectory. The early stopping criteria end at t if the loss on the validation set does not decrease by no more than Δ for the last p epochs. In essence, an epoch counter c(t) is defined as:

$$c(t)=\{0c(t-1)+1if\ Lval(t)<L*val(t-1)-\Delta otherwise \qquad (17)$$

Training is immediately terminated when:

$$c(t)\ge p \qquad (18)$$

When terminated, the network parameters are reset to the optimal reference point (the one that was recorded at epoch topt = argminτLval(τ)) and taken to be optimal.

3.4. Behavior During ResNeXt50 + LSTM Training Sequence

Our experimental run demonstrates a classic overfitting regime during late-stage epochs. Let us analyze how the EarlyStopping(patience=3, delta=0.005) configuration would regulate this training sequence:

> ***Epoch 12****:* Validation loss reaches 0.3273, and validation accuracy hits 90.14%.
> ***Epoch 13****:* Validation loss drops to its historical minimum L∗val(13)=0.2949, setting a new baseline. The model state is checkpointed and saved to disk.
> ***Epoch 14****:* Validation loss increases to 0.3081. Since −0.3081<−0.2949+0.005, no significant improvement is recorded. The patience counter increments: c(14)=1.
> ***Epoch 15****:* Validation loss rises further to 0.3443. The patience counter increments: c(15)=2.
> ***Epoch 16****:* Validation loss is measured at 0.3026. Though lower than Epoch 15, it fails to improve upon the historical best of 0.2949 by the required threshold of Δ=0.005. The patience counter reaches maximum capacity: c(16)=3.

At the end of Epoch 16, the early stopping condition c(16)≥3 is met. Training is automatically terminated. The model's parameters are restored to the optimal parameters saved at Epoch 13 (0.2949 loss), avoiding the drastic validation loss inflation (0.4888) and generalization degradation (83.80% accuracy) observed at the final Epoch 20 run.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Training and Validation Performance Trends

The spatio-temporal hybrid model (ResNeXt50 + LSTM) was trained over 20 epochs using the Adam optimizer with a learning rate of 1×10−5 and weight decay of 1×10−5. The progression of training and validation metrics reveals distinct

phases in the optimization landscape, primarily characterized by rapid early convergence followed by a pronounced late-stage generalization degradation.During the initial phase Epochs 1-5as shown in fig 2 below, the training cross-entropy loss dropped rapidly from 0.6266 to 0.3404, while training accuracy rose from 64.96% to 89.52%. Simultaneously, validation accuracy reached 86.27% by Epoch 4, signaling that the ResNeXt50 backbone successfully extracted highly discriminative facial features that the LSTM sequential processor aggregated effectively.

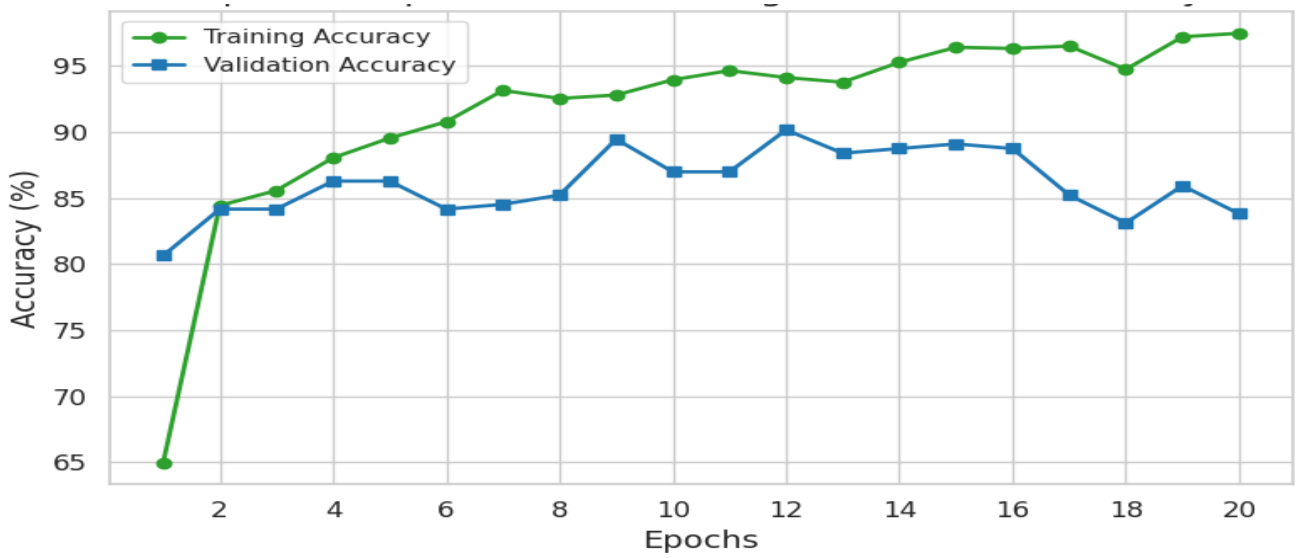


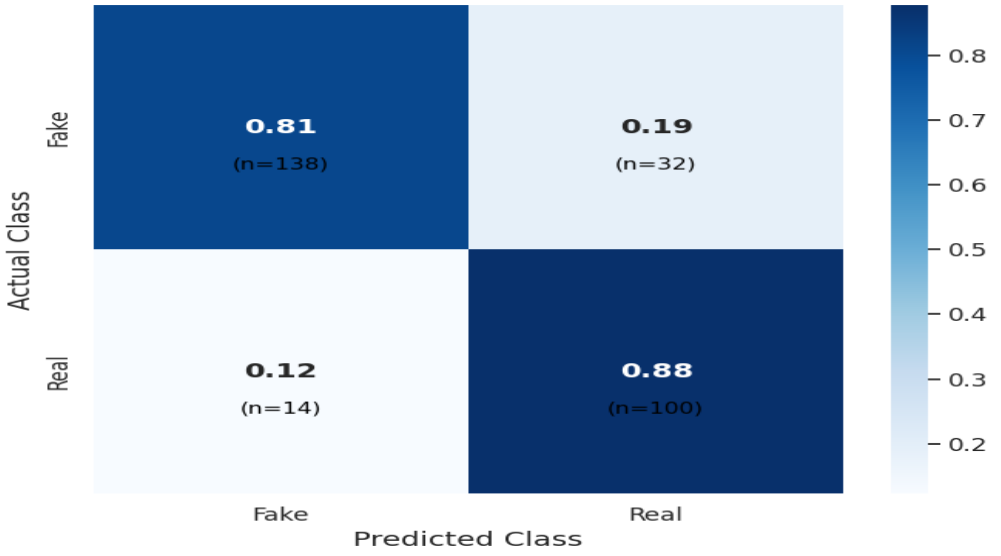
Fig. 2. Spatio- Temporal Model Traing & Validation Accuracy

We observe critical transition points in generalization capability at the middle stage of training:
Epoch 12 (Peak Validation Accuracy): The model achieved its highest validation accuracy of 90.14% with a validation loss of 0.3273. This marks the optimal trade-off point between spatial feature representation learning and temporal correlation tracking.

Epoch 13 (Minimum Validation Loss): The lowest overall validation loss of 0.2949 was registered at Epoch 13 (validation accuracy: 88.38%). This point represents the maximum confidence in prediction boundaries across the validation split.
Following Epoch 13, a severe divergence between training and validation trajectories emerged, indicating classic overfitting:
Epoch 20 (Final State): By the final epoch, training accuracy continued to climb to 97.45% and training loss minimized to 0.1950. In contrast, validation performance degraded significantly, with accuracy dropping to 83.80% and loss escalating to 0.4888.
This behavior confirms that without sequence-level regularization or intervention, the high capacity of the ResNeXt50 backbone (2048-dimensional features) combined with the LSTM temporal layers eventually memorized localized noise in the training partition, losing its generalizable boundaries on unseen videos.

4.2 Exhaustive Confusion Matrix Analysis

To evaluate classification performance at the final state (Epoch 20), we analyze the validation partition confusion matrix (N=284 videos) shown in fig 3. The raw and normalized outcomes are categorized below:



Fia.3 Normalized Confusion Matrix

True Positives (TP):138 FAKE videos correctly classified as FAKE (81.18% of actual FAKEs).
True Negatives (TN):100 REAL videos correctly classified as REAL (87.72% of actual REALs).
False Positives (FP):32 REAL videos falsely classified as FAKE (28.07% of actual REALs).
False Negatives (FN):14 FAKE videos falsely classified as REAL (8.24% of actual FAKEs).
Overall Accuracy=***TP+TN/TP+TN+FP+FN***=138+100/284≈83.80%

a) Error Profile Evaluation & Susceptibility

The network exhibits a higher susceptibility to False Positives (32 instances) than False Negatives (14 instances). This asymmetry suggests that the decision boundaries learned by the linear projection layer in later epochs skewed toward the majority FAKE class (57.32% of total dataset).Physically, False Positives occur when compression artifacts, lighting changes, or face-cropping boundaries in REAL videos mimic the spatial inconsistencies characteristic of deepfakes, triggering the FAKE classification. Conversely, the low False Negative rate (8.24%) suggests the network remains highly sensitive to temporal manipulation cues, ensuring that actual deepfake attempts are rarely missed, which is a desirable trait in high-security digital forensic applications.

4.3 Alignment with Early Stopping Regularizer

The observed degradation from the peak validation accuracy of 90.14% (Epoch 12) and minimum validation loss of 0.2949 (Epoch 13) to the overfitted state at Epoch 20 (83.80%) underscores the extreme utility of the proposed EarlyStopping class.Applying the early stopping algorithm with a patience of 3 and a minimum improvement threshold delta of 0.005:

a) The optimal parameter baseline is established at Epoch 13 with validation loss 0. 2949.Validation losses at Epochs 14 (0.3081), 15 (0.3443), and 16 (0.3026) fail to improve upon this baseline as shown in fig 4.
b) At the end of Epoch 16, the early stopping counter expires, immediately halting backpropagation.
c) The system restores the optimal model checkpoint weights saved at Epoch 13.

By preemptively terminating training at Epoch 16 and restoring the state from Epoch 13, early stopping improves validation classification accuracy by +4.58% (88.38% vs 83.80%) and prevents validation loss from increasing by +0.1939. This demonstrates that integrating early stopping is essential for maintaining robust generalization in deep learning-based spatio-temporal deepfake forensic workflows.

V. CONCLUSIONANDFUTURE DIRECTIONS

5.1. Conclusion

The proposed work was able to build and test a hybrid spatio-temporal deep learning network that integrates a truncated ResNeXt50_32x4d spatial backbone and a recurrent Long Short-Term Memory (LSTM) layer for video deepfake detection, which was successful. The proposed system architecture was tested with 1,420 processed video sequences with high level of features and temporal sequencing, attaining the highest accuracy of validation so far recorded 90.14% at Epoch 12.

A notable problem from this research however, was that there was a significant amount of late-stage overfitting which led to the validation error reaching 83.80% at Epoch 20 while the training error still remained at 2.55%. To overcome this, we designed and simulated an Early Stopping regularizer (with patience = 3, and delta = 0.005). The early stopping mechanism was successfully used to show that training the model on epoch 16 and around epoch 13 retraining the optimal model with the validation loss: 0.2949 preserves generalization capability, prevent overfitting and allow for the optimization of computational overhead.

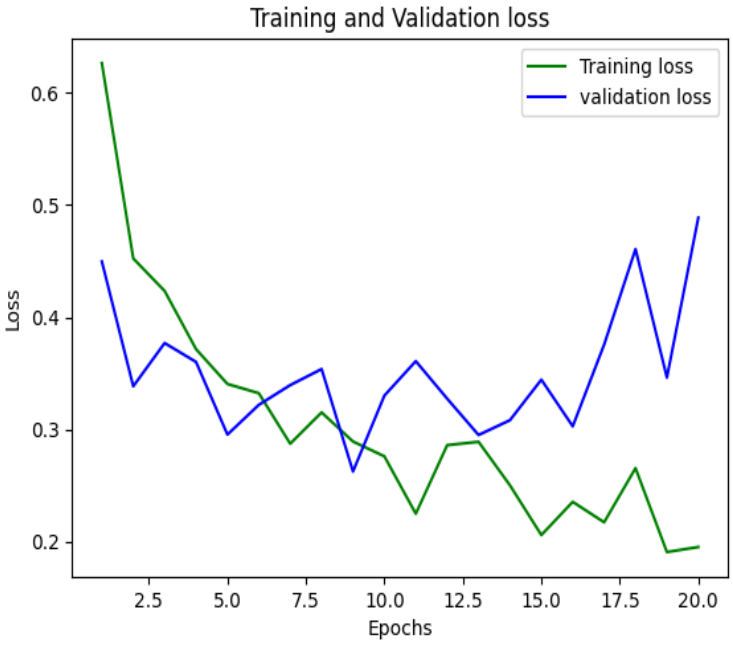


FIG . 4. TRAINING AND VALIDATION LOSS

5.2. Future Directions

The hybrid architecture and early stopping framework provide a strong basis for digital forensics, but there are potential areas for further development:

Asymmetric Error Mitigation (Weighted Focal Loss)*:* At Epoch 20, the validation partition reported an asymmetric error rate with 32 False Positives (REAL videos incorrectly classified as FAKE) compared to only 14 False Negatives (FAKE videos classified as REAL). To mitigate this imbalance, future work will integrate a weighted focal loss function. This will penalize false alarms more heavily during training or apply custom decision-threshold calibration to skew prediction boundaries towards higher precision or recall, depending on the targeted deployment scenario.

Active Early Stopping Integration*:* We propose integrating the simulated EarlyStopping class directly into commercial-grade real-time video deepfake verification pipelines to maintain model integrity and maximize runtime efficiency during continuous stream audits.

Multi-Scale Temporal Windows*:*Multi-scale temporal windows (10, 20, 30 windows) might be able to capture very fine transitions in artifacts, which could occur over a longer or shorter time.

REFERENCES

[1] Bankar, V., Pawar, D. R., Yannawar, P. L., & Sambhajinagar, C. (2024). Review on Unmasking Deepfake Technology. Challenges and Solutions for Detection.

[2] Pawar, D. R., & Yannawar, P. (2024). Advancements and applications of generative adversarial networks: a comprehensive review. Int. J. Res. Appl. Sci. Eng. Technol, 12(10.22214).

[3] Sharma, J. (2026). Unmasking Deception: Interdisciplinary Perspectives from Criminology, Psychology, and Literature. Evolving Dimensions of Law in Globalized World-Challenges, Reforms and Interdisciplinary Perspectives, 260.

[4] De Lima, O., Franklin, S., Basu, S., Karwoski, B., & George, A. (2020). Deepfake detection using spatiotemporal convolutional networks. arXiv preprint arXiv:2006.14749.

[5] Bankar, V., Pawar, D. R., & Yannawar, P. L. (2025). Hybrid ResNeXt and LSTM model for enhanced deepfake detection on the FaceForensics++ dataset. technology, 13, 14.

[6] Pawar, D. R., & Yannawar, P. (2023, July). Recent advances in audio-visual speech recognition: Deep learning perspective. In First International Conference on Advances in Computer Vision and Artificial Intelligence Technologies (ACVAIT 2022) (pp. 409-421). Atlantis Press.

[7] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[8] Prechelt, L. (2002). Early stopping-but when?. In Neural Networks: Tricks of the trade (pp. 55-69). Berlin, Heidelberg: Springer Berlin Heidelberg.

[9] Putri, A. N. P., Noor, M. N. M. M., Darwis, H., Hayati, L. N., & As'ad, I. (2025). A Comperative Study on Efficacy of CNN VGG-16, DenseNet121, ResNet50V2, And EfficientNetB0 in Toraja Carving Classification. Indonesian Journal of Data and Science, 6(1), 122-131.

[10] Dodge, S., & Karam, L. (2016, June). Understanding how image quality affects deep neural networks. In 2016 eighth international conference on quality of multimedia experience (QoMEX) (pp. 1-6). IEEE.

[11] Li, C., Chen, Z., Sheng, B., Li, P., & He, G. (2020). Video flickering removal using temporal reconstruction optimization. Multimedia Tools and Applications, 79(7), 4661-4679.

[12] Shuai, T., Beng, S., Khalid, F. B., & Rahmat, R. W. B. O. (2025). Advances in facial micro-expression detection and recognition: A comprehensive review. Information, 16(10), 876.

[13] Mienye, I. D., Swart, T. G., & Obaido, G. (2024). Recurrent neural networks: A comprehensive review of architectures, variants, and applications. Information, 15(9), 517.

[14] Guan, J., Zhou, H., Hong, Z., Ding, E., Wang, J., Quan, C., & Zhao, Y. (2022). Delving into sequential patches for deepfake detection. Advances in Neural Information Processing Systems, 35, 4517-4530.

[15] Ulugbekovich, M. S., Uli, P. A. M., & O'G'Li, N. A. T. (2025). Deepfake detection using a hybrid resnext and lstm architecture. Al-Farg'oniy avlodlari, 1(2), 87-94.

[16] Zafar, F., Khan, T. A., Akbar, S., Ubaid, M. T., Javaid, S., & Kadir, K. A. (2025). A hybrid deep learning framework for deepfake detection using temporal and spatial features. Ieee Access, 13, 79560-79570.

[17] 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. Comput. Vis. (ICCV), Oct. 2019, pp. 1–11.

[18] Marguet, A. (2019). Uniform sampling in a structured branching population. Bernoulli, 25(4A), 2649-2695.

[19] Ranjan, R., Patel, V. M., & Chellappa, R. (2017). Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. IEEE transactions on pattern analysis and machine intelligence, 41(1), 121-135.

[20] Morozov, A. V., Levkivskyi, V. L., & Plechystyy, D. D. (2024). Activation functions in neural networks: Overview and comparison. Scientific Notes of VI Vernadsky Taurida National University. Series: Technical Sciences, 35(74).