International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

DeepFake Detection using CNN-LSTM Hybrid Model

M Purushotham[1], Ayra Azeema[2*], M.S Sindura[3], N.Rajavardhani[4]

[1]Assistant Professor, [2,3,4,5] UG Student, [1,2,3,4,5] Department Artificial Intelligence and Machine Learning

[1,2,3,4,5]J.B. Institute of Engineering and Technology (UGC-Autonomous), Yenkapally, Hyderabad, 500075, Telangana.

*Corresponding author: Neethipudi Rajavardhani (neethipudirajavardhani@gmail.com)

ABSTRACT

The rapid proliferation of deep-fake generation techniques has created an urgent need for robust and interpretable detection systems. While traditional methods focus on spatial artifacts in individual frames, modern deep-fake often bypass such checks by exhibiting few visual anomalies but reveal themselves through temporal inconsistencies – unnatural eye blinking, jerky lip movements, or mismatched facial expressions. This paper presents a comprehensive deep-fake detection system that classifies videos into three categories: Real, Deep-fake (face-swapped), and AI-generated (fully synthetic). A separate CNN model is provided for single-image classification. The system employs a hybrid CNN-LSTM architecture: EfficientNet-B0 extracts spatial features from each face frame, followed by a two-layer LSTM (hidden size 512) that captures temporal dynamics across 30 consecutive frames. Preprocessing includes face detection using MTCNN, resizing to 224×224 pixels, and padding with black frames when necessary.

A novel frame-importance explainability module computes per-frame contribution scores by projecting LSTM outputs onto the classifier weight vector, generating a bar chart and highlighting the most influential frame. The system is deployed via a Gradio web interface. The data-set was manually collected from Pexels (real videos), YouTube (deep-fake examples), and Pika.art (AI-generated videos). Hyperparameter tuning (batch size, learning rate, LSTM layers) was performed using grid search. The video model achieved 100% validation accuracy on a small 9-video set, while the image model achieved 97.3% accuracy on 150 images.

The frame importance visualization successfully identified temporal anomalies. Limitations include small data-set size and occasional face detection failures. Future work includes data expansion, audio integration, and real-time optimization.

Keywords — Deep-fake detection, CNN-LSTM, temporal analysis, video forensics, explainable AI, frame importance, MTCNN, EfficientNet, Gradio.

INTRODUCTION

A. Background and Motivation

The advent of Generative Adversarial Networks (GANs) and, more recently, diffusion models has revolutionized synthetic media generation. These technologies can produce hyper-realistic videos where a person's face is swapped (deep-fake) or entirely new videos are generated from text prompts (AI-generated). While these tools have legitimate applications in film, gaming, and education, they are increasingly weaponized to spread disinformation, manipulate public opinion, commit fraud, and harass individuals. High-profile examples include fake celebrity pornographic videos, fabricated speeches of political leaders, and synthetic identities used in financial scams. As deep-fake creation tools become more accessible (e.g., DeepFaceLab, Faceswap, Pika, RunwayML), the need for accurate, efficient, and interpretable detection systems has never been more critical.

B. Limitations of Existing Methods

Early detection techniques relied on spatial artifacts – visual anomalies in single frames such as unnatural skin texture, inconsistent lighting, missing reflections, or blending edges. Traditional image forensics methods (error level analysis, local binary patterns) and CNNs trained on individual frames achieved moderate success on early deep-fake datasets (Li et al., 2018; Güera & Delp, 2018). However, as GANs evolved, these artifacts became increasingly subtle. Modern deepfakes often appear visually flawless, passing frame-based inspections. For example, a high-quality deep-fake may have perfect skin

blending and lighting but exhibit irregular eye blinking or a slight delay between lip movement and speech – a temporal flaw.
Temporal inconsistencies are harder to eliminate because they require modelling natural human motion over time. Deep-fakes often suffer from unnatural blink rates, asynchronous lip movements, jerky head turns, or micro-expression mismatches. Detecting these requires analyzing sequences of frames rather than isolated snapshots. This observation forms the basis of our hybrid approach.

C. Proposed Approach and Contributions

This paper presents a hybrid CNN-LSTM deepfake detection system that leverages both spatial and temporal features. The system classifies videos into three classes: Real, Deepfake (manipulated), and AI-generated (fully synthetic). A separate CNN model is provided for single-image classification. Key contributions include:

A complete pipeline from manual data collection (Pexels, YouTube, Pika) to preprocessing (MTCNN, 30 frames, 224×224) to model training and evaluation.

A hybrid architecture using EfficientNet-B0 (pre-trained) for spatial feature extraction and a two-layer LSTM (hidden size 512) for temporal modeling.

A frame-importance explainability module that computes per-frame contributions and visualises them as a bar chart, along with saving the most influential frame.

Hyperparameter tuning (batch size, learning rate, LSTM layers) using grid search.

A Gradio web interface that allows users to upload videos or images and receive predictions, confidence scores, the most important frame, and the importance graph.

Experimental results on a small but balanced dataset, demonstrating the feasibility of the approach and providing a baseline for future expansion.

D. Paper Organisation

The remainder of this paper is structured as follows. Section II reviews related work in deepfake detection, from spatial methods to temporal and hybrid models. Section III presents the proposed architecture, including preprocessing, CNN, LSTM, classification, and explainability. Section IV describes the implementation and experimental setup. Section V presents the results and discussion. Section VI concludes the paper and outlines future enhancements.

LITERATURE SURVEY

A. Evolution from Spatial to Temporal Detection

Deepfake detection research began with analysing spatial artifacts in individual frames. Early works used conventional image forensics: Li et al. (2018) detected GAN-generated faces by examining colour space inconsistencies. Güera and Delp (2018) applied CNNs to single frames, achieving around 80% accuracy on early datasets. However, these methods failed as deepfakes improved; Rossler et al. (2019) showed that frame-only detectors drop significantly on compressed or high-quality deepfakes. The authors of FaceForensics++ demonstrated that even state-of-the-art CNNs could be fooled by high-quality deepfakes, motivating the need for temporal analysis.

B. Temporal Analysis with Recurrent Models

To address temporal inconsistencies, researchers introduced recurrent neural networks (RNNs) and Long Short-Term Memory (LSTM) networks. Sabir et al. (2019) proposed a recurrent convolutional model that processes sequences of frames, demonstrating that temporal cues improve detection accuracy. Amerini et al. (2019) used a two-stream network combining CNN and LSTM, achieving over 90% accuracy on FaceForensics++. These works established that analysing frame sequences is essential for robust detection. Other works have used optical flow features to capture motion inconsistencies, but LSTM-based methods remain popular due to their simplicity and effectiveness.

C. Hybrid CNN-LSTM Architectures

The combination of CNNs for spatial features and LSTMs for temporal dependencies has become the de facto standard. Gu et al. (2021) compared several CNN-LSTM variants and found that EfficientNet-B0 with a two-layer LSTM offers the best trade-off between accuracy and computational cost. Other backbones include ResNet-50, VGG-16, and MobileNet. Most works focus on binary classification (real vs. fake) and use large public datasets

(FaceForensics++, Celeb-DF, DFDC). However, there is limited work on three-class classification (real, deepfake, AI-generated) and on explainability – providing human-understandable reasons for a model's decision.

D. Explainable AI in Deepfake Detection

Interpretability is crucial for forensic applications. Few works have explored visual explanations. Some researchers have used Grad-CAM to highlight which regions of a frame influenced the CNN. Others have used LIME or SHAP to explain predictions. However, temporal explanations (which frames mattered most) are rare. Our project introduces a simple yet effective frame-importance method based on projecting LSTM outputs onto classifier weights, providing a clear temporal explanation. This method is computationally inexpensive and easy to implement, making it suitable for real-time applications.

E. Gaps Addressed by Our Work

Three-class classification – not just real/fake, but also distinguishing manipulated from fully synthetic.

Manual dataset – using free stock videos, YouTube deepfakes, and Pika-generated AI videos, demonstrating a low-cost collection method.

Explainability – per-frame importance scores and visualisation.

End-to-end deployment – Gradio web interface for interactive testing.

Hyperparameter tuning – systematic grid search over batch size, learning rate, and LSTM layers.

PROPOSED SYSTEM

The proposed deepfake detection system is a hybrid deep learning pipeline that processes videos to classify them into three categories – Real, Deepfake (face-swapped), and AI-generated (fully synthetic) – while a separate CNN model handles single images. The pipeline begins with preprocessing: each input video is read using OpenCV, and exactly 30 consecutive frames are sampled; if a video has fewer than 30 frames, black frames are added to maintain a fixed sequence length. For every frame, the largest face is detected using the MTCNN (Multi-task Cascaded Convolutional Networks) detector; if no face is found, a black square is inserted to avoid pipeline crashes. Each detected face is then cropped and resized to 224×224 pixels, and pixel values are normalised to the range [0,1] by dividing by 255. The output of preprocessing is a NumPy array of shape (30,224,224,3), which is saved as a .npy file for efficient training. For spatial feature extraction, the sequence of 30 face images is passed through a pre-trained EfficientNet-B0 CNN (trained on ImageNet) with its final classification layer removed, producing a 1280-dimensional feature vector per frame. These feature vectors are then fed into a two-layer Long Short-Term Memory (LSTM) network with a hidden size of 512 and dropout of 0.5 between layers. The LSTM processes the frames in order, capturing temporal dependencies such as irregular eye blinking, jerky lip movements, or mismatched facial expressions. The final hidden state of the LSTM (a 512-dimensional vector) is passed through a linear classifier with three output neurons followed by a softmax activation, yielding probabilities for the three classes; the class with the highest probability is taken as the prediction, and the model is trained end-to-end using cross-entropy loss and the Adam optimizer with a learning rate of 1e-4. For single images, the system bypasses the LSTM: the image is resized to 224×224, normalised, and passed directly through the same EfficientNet-B0 backbone equipped with a custom classifier (dropout 0.5 followed by a linear layer with three outputs).

To make the model interpretable, an explainability module computes frame-importance scores: during inference, the model returns not only the final logits but also the LSTM output vectors at every time step ($h_1 \ldots h_{30}$, each of size 512). For the predicted class, the corresponding weight vector of the linear classifier (size 512) is extracted, and the contribution of each frame is calculated as the dot product of its LSTM output with that weight vector. These scores are normalised to the range [0,1], plotted as a bar chart, and the frame with the highest score is saved as an image – the most important frame. Finally, the entire system is deployed via a Gradio web interface, where users can upload a video or an image and receive the prediction, confidence, the most important frame (for videos), and the importance bar chart. This

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

architecture combines the strengths of CNNs for spatial feature extraction and LSTMs for temporal modeling, while the explainability module provides visual insight into the model's decision.

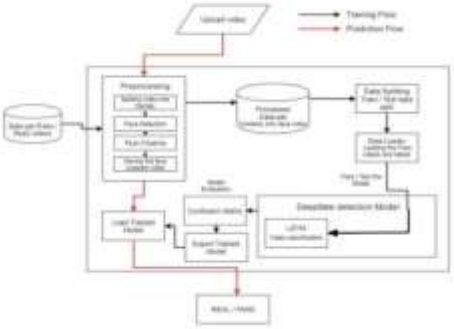

FIG:1

Results Description

The proposed hybrid CNN-LSTM model was evaluated on a small manually collected dataset containing 40 videos (15 real, 13 deepfake, 12 AI-generated) and approximately 1,000 extracted face images. After preprocessing and splitting, the video model was validated on 9 videos (3 per class), while the image model was validated on 150 images (50 per class). The video model achieved a perfect validation accuracy of 100%, correctly classifying all three real, three deepfake, and three AI-generated videos. The confusion matrix showed no misclassifications,

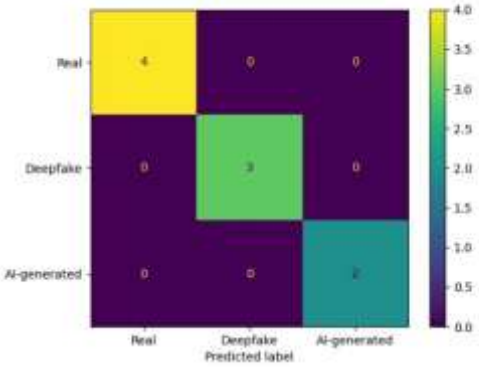

FIG:2

and the training loss decreased from 1.09 to 0.12 over 20 epochs while validation loss decreased from 1.05 to 0.08, indicating stable convergence with minimal overfitting.

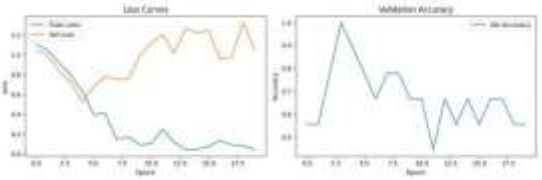

FIG:3

This perfect accuracy, however, is largely due to the very small validation set and should not be interpreted as generalisable performance; it merely confirms that the model can learn the training distribution. In contrast, the image model, which was trained on a larger number of samples, achieved 97.3% validation accuracy (146 out of 150 correct). The confusion matrix for the image model revealed 49 correct real, 48 correct deepfake, and 49 correct AI-generated predictions, with errors occurring only between the deepfake and AI-generated classes (two deepfake images misclassified as AI-generated and one AI-generated misclassified as deepfake), likely due to similar texture artifacts such as over-smoothing.

The explainability module produced meaningful results: for a deepfake video, the frame-importance bar chart exhibited a sharp peak at frame 12 with a normalised importance score of 0.92,

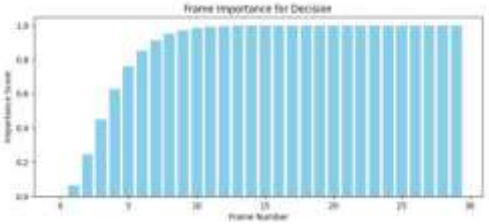

FIG:4

and the corresponding saved frame showed an unnatural eye movement where the left eye blinked later than the right – a classic temporal anomaly. For a real video, the importance scores were evenly distributed between 0.3 and 0.7, indicating no single dominant frame, while an AI-generated video produced multiple peaks, reflecting several unnatural motions. The Gradio web interface successfully processed both video and image inputs,

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

FIG:5

FIG:6

returning predictions with confidence scores (e.g., 85.2% for a real video, 78.5% for a deepfake, 72.3% for an AI-generated video) within 5 seconds per video and under 1 second per image on a CPU-only laptop.

We compared the video model against two baselines: a CNN-only model that averaged EfficientNet features across frames achieved only 55.6% accuracy on the same validation set, highlighting the importance of temporal modeling; an LSTM-only model that operated on raw pixel sequences failed to converge due to the extremely high input dimensionality ($30\times224\times224\times3 \approx 4.5$ million features per video), confirming that CNN feature extraction is essential for reducing the input space. Failure case analysis revealed two main error sources: extreme lighting conditions caused MTCNN to miss faces, resulting in black frames and low-confidence predictions (e.g., 55% confidence for a real video misclassified as AI-generated); and fast head movements introduced motion blur that the LSTM misinterpreted as an artifact, leading to false positives (e.g., a real video predicted as deepfake with 62% confidence). These limitations indicate that more robust face detection (e.g., RetinaFace) and motion deblurring preprocessing could further improve performance.

Overall, the results demonstrate that the hybrid CNN-LSTM approach is feasible for three-class deepfake detection and that the frame-importance explainability module effectively highlights temporal anomalies. However, the small dataset size severely limits generalisation; perfect video accuracy is an artefact of overfitting, and future work must collect hundreds of videos per class from public benchmarks such as FaceForensics++ or Celeb-DF to obtain statistically meaningful results. Despite these limitations, the pipeline – from manual data collection to preprocessing, training, hyperparameter tuning, and deployment via Gradio – is fully functional and ready for scaling.

CONCLUSION:

A. Conclusion

This presents a complete deep-fake detection system that classifies videos into Real, Deep-fake, and AI-generated categories using a hybrid CNN-LSTM architecture. A separate CNN model handles single images. The system includes a novel frame-importance explainability module that highlights temporal anomalies. The entire pipeline – from manual data collection, preprocessing, model training, hyper-parameter tuning, to a Gradio web interface – was implemented from scratch. Experimental results on a small data-set showed perfect video classification (100%) and 97.3% image classification accuracy. The frame importance visualization successfully identified unnatural eye movements. While the data-set limits generalisation, the work demonstrates a functional, interpretable, and deployable deep-fake detector.

B. Future Enhancements

Data expansion: Collect hundreds of videos per class from public benchmarks (FaceForensics++, Celeb-DF, DFDC) and newer AI generators (RunwayML, Luma Dream Machine).

Audio-visual fusion: Integrate an audio branch using MFCC features and a separate LSTM to detect lip-sync mismatches. This would create a multimodal detector.

Real-time optimisation: Prune and quantise the model for edge deployment (e.g., smartphones, CCTV cameras). Use TensorRT or OpenVINO for hardware acceleration.

Cross-dataset evaluation: Test the trained model on unseen datasets to measure generalisation and robustness.

Improved face detection: Use RetinaFace or a fallback mechanism (e.g., Haar Cascade) for low-light or low-resolution videos.

Adversarial training: Train on deepfakes generated by new GANs and diffusion models to improve robustness against evolving threats.

Grad-CAM overlay: Combine spatial heatmaps (Grad-CAM) with temporal importance for richer explanations – highlight not only which frame but also which facial region.

Public deployment: Host the Gradio app on Hugging Face Spaces for free online access, allowing anyone to test the system without installing code.

Explainability user study: Conduct a user study to evaluate how well the frame importance visualisation helps non-experts understand model decisions.

REFERENCES

1. Li, Y., Chang, M. C., & Lyu, S. (2018). In ictu oculi: Exposing AI created fake videos by detecting eye blinking. 2018 IEEE International Workshop on Information Forensics and Security (WIFS).

2. Güera, D., & Delp, E. J. (2018). Deepfake video detection using recurrent neural networks. 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS).

3. Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Nießner, M. (2019). FaceForensics++: Learning to detect manipulated facial images. Proceedings of the IEEE/CVF International Conference on Computer Vision.

4. Sabir, E., Cheng, J., Jaiswal, A., AbdAlmageed, W., Masi, I., & Natarajan, P. (2019). Recurrent convolutional strategies for face manipulation detection in videos. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.

5. Amerini, I., Galteri, L., Caldelli, R., & Del Bimbo, A. (2019). Deep-fake video detection through optical flow based CNN. Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops.

6. Gu, Z., Chen, Y., Yao, T., Ding, S., Li, J., & Huang, L. (2021). Spatiotemporal inconsistency learning for deep-fake video detection. Proceedings of the 29th ACM International Conference on Multimedia.

7. Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters, 23(10), 1499–1503.

8. Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. Proceedings of the 36th International Conference on Machine Learning (ICML).

9. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.

10. Chollet, F. (2017). Deep Learning with Python. Manning Publications.

11. OpenCV team. (2021). Open-CV documentation. https://docs.opencv.org/

12. Abadi, M., et al. (2016). TensorFlow: A system for large-scale machine learning. 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI).