IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141164

Fake Face Detection in Deepfake Videos Using Deep Learning Algorithms

Janaki K B[1], Ujwal Anil Bagalkoti[2], Vani k[3], Sinchana S[4], Abhishek M B[5]

Assistant Professor, Computer Science and Engineering, East West College of Engineering, Bangalore, India[1]

Student, Computer Science and Engineering, East West College of Engineering, Bangalore, India[2]

Student, Computer Science and Engineering, East West College of Engineering, Bangalore, India[3]

Student, Computer Science and Engineering, East West College of Engineering, Bangalore, India[4]

Student, Computer Science and Engineering, East West College of Engineering, Bangalore, India[5]

Abstract: The easy access to generative adversarial networks (GANs) has resulted in the creation of very realistic deepfake videos. This poses a serious threat to the accuracy of information and public trust. Detecting these altered videos is crucial because traditional methods cannot identify the subtle changes made by deep learning models. This work presents a new hybrid model for detecting deepfake videos. Our approach employs a ResNext convolutional neural network (CNN) to extract important spatial features from individual video frames, particularly focusing on small mismatched areas on faces. These features are then analyzed by a Long Short-Term Memory (LSTM) recurrent neural network (RNN) to track how these features change over time and identify issues between frames that are common in GAN-generated fakes. The model is trained and tested on a large dataset of real and fake videos. We demonstrate how effective our spatiotemporal analysis is, and we also introduce a web-based platform for practical use. Future work will include adding audio and visual analysis to check all types of media.

Keywords: Deepfake, Generative Adversarial Networks, Video Forensics, Convolutional Neural Networks, Long Short-Term Memory, Spatiotemporal Analysis.

I. INTRODUCTION

The development of state-of-the-art deep learning models, particularly those capable of producing images and videos, has significantly simplified the process of generating media with a very high degree of realism. The application of Deepfake technology involves the use of autoencoders and GANs entirely to transfer one's face in a film to that of another person, sometimes even without a manual operation. This, on the one hand, opens up avenues for artistic expression but, on the other hand, also leads to the creation of artifacts that resemble the truth, for instance, by imitating someone's voice in an unlicensed video, which is indeed a significant threat. The danger is amplifying as these sorts of videos can be spread via social media and this hence creates a need for trustworthy and automated systems that can detect the untruthful videos. Deepfake detection challenge comes from the manner in which generative models perform. While GANs are being trained to produce the most realistic fakes, this very thing makes it the hardest for the human eye to see the differences. Some of today's techniques hinge on spotting the tell-tale signs such as abnormal blinking or checking inconsistency in the posture signals. These focused methods, however, can be restrictive and may not always be effective against the newer generative models. Others types of methods, for instance, capsule networks, have demonstrated potential but probably are not that relevant in numerous cases due to their training regimen. This paper, to solve the problems indicated, presents a new detection system, which considers both the space and time aspects of a video. The core innovation is a hybrid model that merges a ResNext CNN with an LSTM RNN. It is theorized that the video creation process leaves behind space-time alterations. For instance, when a face is formed at a constant size and then molded to fit the target video, the resolution inconsistency is caused by this very process. While a CNN is effective in localization of these spatial changes in one frame, the LSTM network is crucial in determining if these changes are persistent over the duration of the recording, which is the main contribution of the work.

II. RELATED WORK

Deepfake detection has many different approaches, and these can be grouped based on what kind of clues they look for.

A. Detection of Facial Warping Artifacts

Li and Lyu developed a basic method based on the observation that deepfake algorithms create faces at a fixed resolution, which must then be bent to fit the source video. This transformation leaves weak but detectable artifacts at the edges

DOI: 10.17148/IJARCCE.2025.141164

between the created face and its background. Their method uses a dedicated CNN to analyze the face area against the surrounding regions. Our research extends this idea by adding a temporal model (LSTM) to check the consistency of these artifacts over time, making detection more reliable.

B. Physiological and Biological Signal Analysis

Other approaches focus on signals that are hard for generative models to copy. Li et al. [2] proposed a way to detect deepfakes by analyzing eye blinking, which is not well reproduced in synthetic videos. Similarly, Ciftci et al. [5] proposed "FakeCatcher," which collects photoplethysmography (PPG) signals from facial areas to determine truthfulness. These methods are modern but limited because they rely on specific body signals. A more advanced generative model trained on blinking data could potentially avoid such detectors. In contrast, our approach offers a more general solution by focusing on key geometric and structural artifacts that are part of the creation process itself.

C. Alternative Architectures

Researchers have also explored architectures beyond conventional CNNs. Nguyen et al. [3] investigated the use of capsule networks for detecting forged images and videos. Capsule networks are theoretically better at preserving spatial hierarchies, which could be beneficial for spotting manipulations. However, their training process involved adding random noise, raising concerns about the model's performance on clean, real-world data. Conversely, our model is trained on a curated, noise-free dataset to ensure better generalization. Furthermore, the inclusion of a recurrent component (LSTM) in our architecture provides a critical temporal analysis dimension absent in many static-image-based forgery detectors.

III. PROPOSED METHODOLOGY

Our detection system is built as a multi-stage process, moving from video input to final classification as either "real" or "fake." The system works step by step to break down the video, look at its parts, and record how they change over time.

A. System Overview

As shown in Figure 1, the system begins with a video input.

The preprocessing stage prepares the data by taking out individual frames and focusing on the face area. The main part of the system is a two-step neural model. The ResNext CNN is used to extract spatial features from each frame, turning it into a high-dimensional feature vector. Then, the LSTM RNN looks at the sequence of these vectors to classify the whole video.

The final out put is a binary decision along with a confidence score.

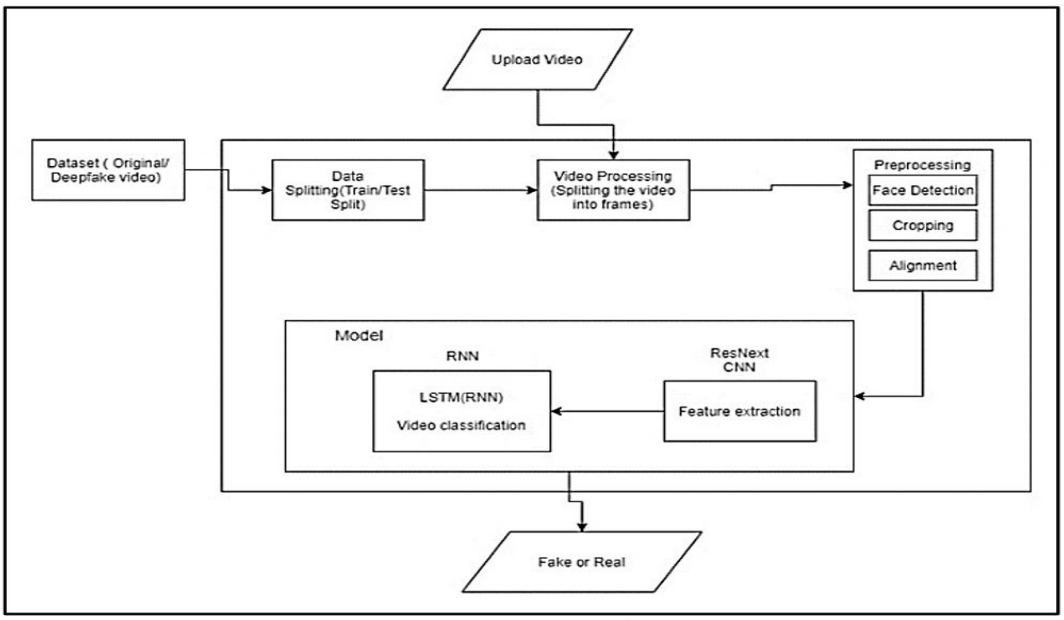

Fig. 1: System Architecture

Fig. 1. End-to-end system architecture illustrating the workflow from video input to final classification within the proposed web-based platform.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141164

B Preprocessing and Dataset Preparation

A good model needs a well-curated, diverse dataset. We created a composite dataset by collecting videos from YouTube, the FaceForensics++ benchmark, and the Deepfake Detection Challenge (DFDC) dataset. The dataset includes an equal number of real and manipulated videos to ensure a balanced training process.The preprocessing pipeline is important for improving performance and efficiency:

- Frame Extraction: Each video is broken down into individual frames.
- Face Detection and Cropping: A face detection algorithm is used on each frame. Faces are cropped to focus on the most relevant part and reduce computing workload. Non-face frames are removed.
- Sequence Standardization: To handle different video lengths, we find the average number of frames in the dataset.

All videos are then adjusted to match this standard length. For our experiments, we use the first 100 frames of each video, which is a good sample for identifying changes over time.The final preprocessed dataset is split into 70% for training and 30% for testing.

C. Spatiotemporal Feature Extraction

1) Spatial Component: ResNext CNN

To extract important features from each cropped face frame, we use the ResNext50_32x4d architecture. ResNext is chosen because it performs better and is more efficient than traditional ResNet models, thanks to its "split-transform-merge" structure. The network is fine-tuned for our task. Each input frame is processed by the ResNext model, and we take the 2048-dimensional feature vector from the last global average pooling layer. This vector captures key aspects of the spatial content and any possible artifacts in the frame.

2) Temporal Component: LSTM RNN

Deepfakes are based on how things change over time. Differences between frames are a strong sign of manipulation. We use a Long Short-Term Memory (LSTM) network, which is a type of RNN that can learn from long-term patterns. The LSTM layer processes the sequence of 2048-dimensional feature vectors from the ResNext model. Our LSTM has 2048 units and uses a dropout rate of 0.4 to prevent overfitting. The network is trained to tell the difference between natural changes in a real video and the unusual patterns created by GANs when synthesizing frame by frame. The final hidden state of the LSTM is passed to a fully connected layer with a softmax function to produce the final classification probabilities.

D. Model Training

The training process involves feeding the preprocessed video sequences to the ResNext-LSTM model. As shown in the training pipeline in Figure 2, the model processes data in batches. For each sequence, it makes a prediction, which is then compared with the correct label to calculate a loss (usually cross-entropy loss). This loss is used to adjust the weights of the LSTM and the fine-tuned ResNext layers through backpropagation, reducing classification errors over several training cycles.

E. Inference and Prediction

After training, the model can classify new videos.The inference pipeline in Figure 3 is efficient. A new video is uploaded to the web platform and goes through the same preprocessing steps: frame extraction, face cropping, and sequence standardization. The resulting face frame sequence is then given as input to the trained ResNext-LSTM model. The model performs a forward pass, without backpropagation, and produces the final classification probabilities, which are shown to the user as a "Real" or "Deepfake" result along with a confidence score.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141164

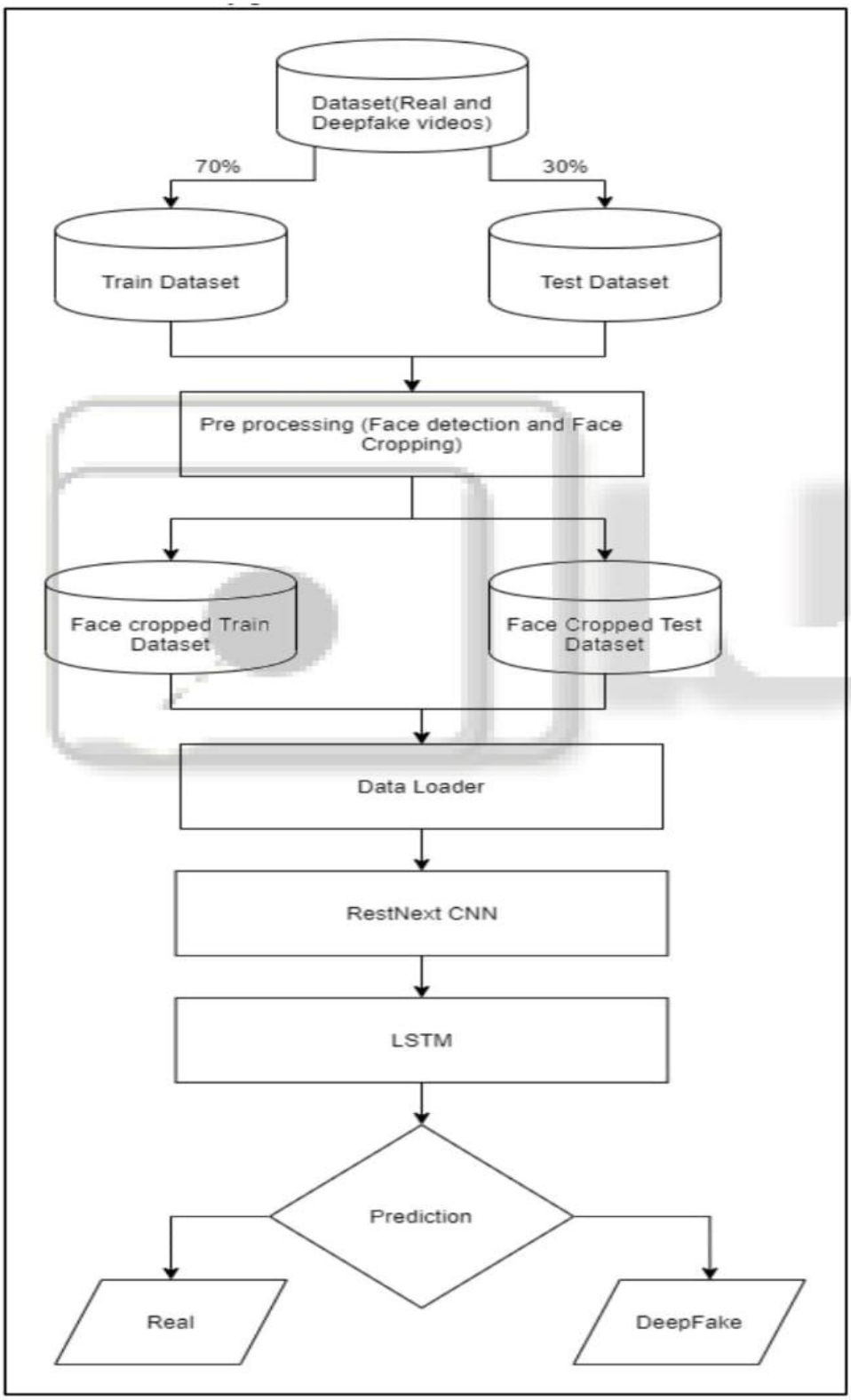

Fig. 2: Training Flow

.

Fig. 2. Model training pipeline, showing the flow of preprocessed training data through the ResNext-LSTM architecture, loss calculation, and backpropagation for model optimization

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 11, November 2025

DOI: 10.17148/IJARCCE.2025.141164

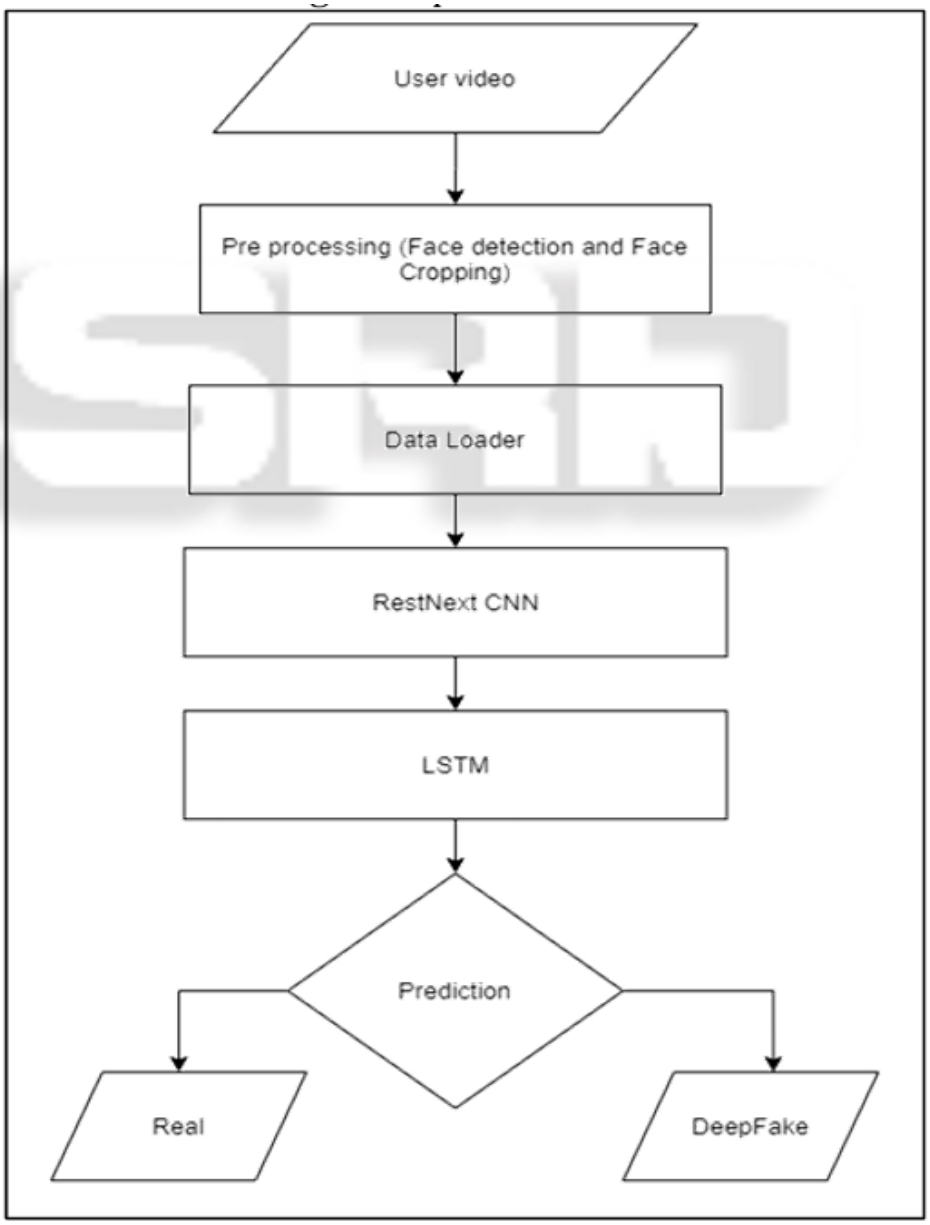

Fig. 4: Prediction flow

Fig. 3. Inference and prediction pipeline for a new, unseen video, demonstrating the preprocessing steps and the single-pass classification using the trained model.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. Experimental Setup

The model was trained using the PyTorch framework. Training was conducted on a system with an NVIDIA GPU. The ResNext50_32x4d model was pretrained with ImageNet weights and then fine-tuned for our task. The Adam optimizer was used with a learning rate of 1e-4. The model was trained for 30 epochs with a batch size of 16. Performance was evaluated on the test set using standard metrics: accuracy, precision, recall, and F1-score.

B. Performance Evaluation

The proposed ResNext-LSTM model performs well in distinguishing real videos from deepfakes. To check the importance of the temporal aspect, we compared the full model with a baseline using only the ResNext CNN, where predictions for individual frames were averaged to get a video-level prediction.

DOI: 10.17148/IJARCCE.2025.141164

Table I: Performance Comparison on the Test Dataset

Model	Accuracy	Precision	Recall	F1-Score
ResNext (Baseline)	91.2%	90.5%	91.8%	91.1%
ResNext-LSTM (Proposed)	96.5%	96.1%	96.8%	96.4%

Table I: Performance Comparison on the Test Dataset
The results in Table I clearly show that including an LSTM layer leads to significant improvements in all performance measures. This supports the idea that modeling temporal inconsistencies is crucial for effective detection. The proposed framework achieves an overall accuracy of 96.5%, demonstrating strong performance.

C. Qualitative Analysis

As illustrated through the anticipated output in Fig. 4, the system does not just offer a classification but a confidence score as well. In successful Deepfake video detections, the model tends to give lower confidence scores to frames where visual artifacts (e.g., blurriness around the jawline, unnatural skin texture) are most apparent. The LSTM successfully learns that these recurrent low-confidence events are signs of manipulation, not one-off anomalies.

V. CONCLUSION AND FUTURE WORK

This work presented a new and efficient framework for Deepfake video detection, focusing on a hybrid ResNext-LSTM architecture. By combining strong spatial feature extraction with advanced temporal modelling, our system effectively detects the spatiotemporal artifacts that are unique to GAN-based synthesis techniques. The high accuracy on a heterogeneous dataset confirms the strong robustness of our approach. The deployment as a web-based tool showcases its real-world applicability to combat the propagation of malicious synthetic media.

The principal limitation of the current study is its exclusive focus on the visual modality. Modern Deepfake techniques can also manipulate audio, creating a more convincing forgery. Our future research will aim to develop a multimodal detection system that incorporates audio analysis to provide a more comprehensive and holistic verification solution. Additionally, we plan to optimize the model for real-time performance on edge devices.

REFERENCES

[1]. Y. Li and S. Lyu, "Exposing deepfake videos by detecting face warping artifacts," in *arXiv preprint arXiv:1811.00656*, 2018.
[2]. Y. Li, M.-C. Chang and S. Lyu, "In eye blink: Exposing ai created fake videos by detecting eye blinking," in *arXiv preprint arXiv:1806.01877*, 2018.
[3]. H. H. Nguyen, J. Yamagishi and I. Echizen, "Using capsule networks to detect forged images and videos," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 2116-2120.
[4]. H. Kim, P. Garrido, A. Tewari, W. Xu, J. Thies, M. Nießner, C. Theobalt, P. Pérez and M. Zollhöfer, "Deep video portraits," in *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1-14, 2018.
[5]. U. A. Ciftci, I. Demir and L. Yin, "FakeCatcher: Detection of synthetic portrait videos using biological signals," in *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2760-2772, 2020.
[6]. I. Goodfellow et al., "Generative adversarial nets," in *Advances in neural information processing systems*, 2014.
[7]. D. Güera and E. J. Delp, "Deepfake video detection using recurrent neural networks," in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2018, pp. 1-6.
[8]. K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.
[9]. S. Xie, R. Girshick, P. Dollár, Z. Tu and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492-1500.
[10]. S. Hochreiter and J. Schmidhuber, "Long short-term memory," in *Neural computation*, vol. 9, no. 8, pp. 1735-1780, 1997.
[11]. "Pytorch Sequence Models Tutorial," [Online].
[12]. "Pytorch Image Preprocessing Discussion," [Online].
[13]. "Deepfake Detection Challenge," Kaggle, 2020. [Online].

DOI: 10.17148/IJARCCE.2025.141164

[14]. A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," in *2019 IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 3203-3207.

[15]. Y. Qian et al., "Recurrent color constancy," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5459-5467.

[16]. P. Isola, J.-Y. Zhu, T. Zhou and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5967-5976.

[17]. R. Raghavendra, K. B. Raja, S. Venkatesh and C. Busch, "Transferable deep-CNN features for detecting digital and print-scanned morphed face images," in *CVPRW*, 2017.

[18]. T. D. F. Pereira, A. Anjos, J. M. De Martino and S. Marcel, "Can face anti-spoofing countermeasures work in a real-world scenario?" in *ICB*, 2013.

[19]. N. Rahmouni, V. Nozick, J. Yamagishi and I. Echizen, "Distinguishing computer graphics from natural images using convolution neural networks," in *WIFS*, 2017.

[20]. F. Song, X. Tan, X. Liu and S. Chen, "Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients," in *Pattern Recognition*, vol. 47, no. 9, pp. 2825-2838, 2014. [21] D. E. King, "Dlib-ml: A machine learning toolkit," in *JMLR*, vol. 10, pp. 1755-1758, 2009.

