Deepfake Detection using Machine Learning: An Analysis of the Effects of Image Degradation

Nishka Batra[1]

Received April 19, 2026
Accepted October 9, 2026
Electronic access November 15, 2026

Deepfake videos present increasing risks for misinformation, identity fraud, and manipulation of public opinion. These videos create significant risks for security. However, detection models usually perform poorly when videos are degraded by real-world conditions like compression and noise. This study investigates how video degradation affects the performance of deepfake detection models. A Convolutional Neural Network Long Short-Term Memory (CNN-LSTM) detection architecture was trained on the FaceForensics++ C23 dataset, using original videos as the real class and NeuralTextures videos as the fake class. Degradations which used blur, noise, and compression were applied to create real-world conditions. Four experiments were conducted to see how the model performs under clean and degraded training and testing conditions. When both training and testing data were clean, the model had a 91.0% accuracy and a ROC-AUC of 0.9546. However, when that same clean-trained model was tested on degraded videos, the performance dropped substantially, resulting in an accuracy of 54.5% and a ROC-AUC of 0.5668. When both training and testing data were degraded, the model had an accuracy of 55.5% and a ROC-AUC of 0.6092. When the degraded-trained model was tested on clean videos, it had an accuracy of 58.5% and a ROC-AUC of 0.5970. Overall, these results show that training detection models with degraded data changes the model's behavior but does not fully restore performance under degraded real-world conditions.

Keywords: Deepfake Detection, Machine Learning, Image Degradation, CNN-LSTM, FaceForensics++, computer vision

Introduction

The rise of artificial intelligence has come with many applications as well as ethical concerns. One such application is deepfakes, which can be described as AI-generated videos, photos, or audio that make people appear to do or say things that never truly happened.[1] While many individuals may not realize it, deepfakes are everywhere, including social media platforms, news sites, and entertainment media. They can be used to spread misinformation and create many security risks. They often allow identity impersonation, fraud, and the manipulation of public opinion. In 2023, face swap attacks increased by 704%.[2] These are a subset of deepfake attacks. As they increase in number, this demonstrates the increased need for detection technology.

Machine learning allows computers to learn patterns from data and use those patterns to make predictions on new examples. In deepfake detection, supervised learning is commonly used because models are trained on labeled real and fake videos. The model learns visual and temporal patterns from the training data and is then tested on unseen videos to evaluate whether it can generalize beyond the examples it has already seen.

Deepfakes function by training AI models on datasets so that they can learn patterns and then reproduce those into new content. Oftentimes, neural networks are used to train them on huge amounts of video data. Models such as Generative Adversarial Networks (GANs) use competing generator and discriminator networks to learn how to produce increasingly realistic synthetic content.[3] Through these processes, deepfakes come in several forms, including audio, video, and image-based deepfakes. While all these present risks, this paper specifically focuses on video deepfakes.

As deepfake generation technology has advanced, deepfake detection has become increasingly difficult. The increasing realism of deepfakes has made it more difficult for humans to distinguish between what is real and what is fake. Additionally, these models are quickly changing, allowing deepfake creators to quickly adapt based on new developments. For this reason, manually identifying deepfakes is no longer reliable, creating a need for automated detection systems.

To address these challenges, researchers have created multiple different detection methods. Many of these methods depend on deep-learning based models, which teach computers to recognize patterns from large amounts of data without being explicitly programmed. Other detection approaches are feature-based or statistical approaches which rely on human

[1] *Mount Si High School, Washington, United States*

© The National High School Journal of Science 2026

selected visual clues. These different models exist because there are several tradeoffs between models including accuracy vs. complexity and interpretability vs. performance.

As previously discussed, there are different categories of deepfake detection methods. Deepfake detection models are often categorized based on their interpretability.[4] Black-box models are AI systems which are hard to interpret but have extremely high accuracy. On the other hand, white-box models are more transparent but have lower accuracy. It may seem like the best choice is the black-box model because of the higher accuracy, but all these factors are important for researchers when choosing a model because deepfake detection involves user trust, explainability, and ethical concerns. In this work, the proposed architecture is mainly a black-box deep learning model. However, the steps of the model are still explainable. ResNet-50 looks at the visual details in each video frame, the LSTM looks at how those details change across frames, and the final classifier decides whether the video is real or fake.

Many deepfake detection systems rely on neural networks, specifically convolutional neural networks (CNNs). CNNs work particularly well for image and video analysis because they can automatically learn hierarchical visual patterns such as edges, textures, shapes, and facial structures.[5] Overall, CNNs have been widely used in computer vision tasks, including deepfake detection. Other types of models include recurrent neural networks, which process sequences of frames over time, and autoencoders, which learn compressed representations of input data. These models can be used in different parts of deepfake generation and detection, depending on the design of the system.

Many researchers have worked on projects related to the detection of deepfakes using various methodologies.

Baocai et al. looked at the robustness of face-swap detection models under common distortions.[6] These included JPEG compression, Gaussian noise, motion blur, and bitrate reduction. This study is directly connected to the present work because several of these distortions overlap with the degradation types that are used in this experiment. Their work shows that strong performance in clean or controlled training conditions may not replicate in real-world video conditions.

Lagsoun et al. proposed a compression-aware framework for deepfake detection.[7] They specifically focused on improving detection under high-compression conditions. Because videos shared online are often compressed during uploading, streaming, or transmission, this is important for real-world applications. Compression can remove details that deepfake models rely on to identify manipulated videos. This supports the need for detection systems that can be reliable even after video quality is decreased.

Tian et al. proposed GFADE, which is a method focused on improving deepfake detection generalization through feature adaptation.[8] Overall, it addresses the broader problem of models failing under domain shift because they learn features that are too specific to one dataset or manipulation type. Feature adaptation methods are particularly useful because they try to make the learned representation more stable across different testing conditions. This connects to the present study because degraded videos also change the features that the detector analyzes, which causes the model to perform differently than during original training.

Batagelj et al. looked at cross-dataset deepfake detection using self-supervised learning.[9] The work showed that self-supervised strategies can help improve generalization across datasets by helping the model learn more transferable features instead of depending only on supervised labels from one dataset. This is specifically important because models can overfit to a training dataset, which is relevant because the present work focuses on how well the model generalizes when the testing condition changes.

Lu Ebrahimi investigated how accuracy can overestimate how models actually perform in the real-world.[10] This fact occurs because online content is often compressed to make it easier to transfer. In their experiment, they used similar data to the current study, focusing on the FaceForensics++ dataset at raw, C23, and C40 quality levels. They applied several degradations like JPEG compression, Gaussian noise, Gaussian blur, average blur, resizing, and combined distortions. For their method, they utilized a robustness-assessment framework with stochastic degradation augmentation during model training. As their conclusion, they discovered that deepfake detectors can lose substantial performance after common image/video transformations. However, training with controlled degradations can help with robustness, but augmentation can reduce performance on clean images. This paper is particularly relevant because it evaluates the dataset used in this study, specifically, FaceForensics++ quality variants, including C23 and C40.

Li et al. focuses on deepfake videos that have been posted on social media, where they often go through compression.[11] They evaluated low-quality deepfake-video settings and documented their exact datasets before formal use. They used a spatial restored detection framework, including a visual-restoration and mapping stage, high frequency recovery, and a degradation-oriented approach. They found that low resolution and heavy compression can remove the needed visual features. The authors report that a restoration-aware detector can outperform conventional approaches on real-world videos. This is extra relevant because it shows how compression is a strong threat to a ResNet-50 extractor because CNNs depend on several visual features.

Gao et al. proposed high-frequency enhancement for highly compressed deepfakes.[12] Specifically, a high-frequency enhancement network that combines local and global high fre-

© The National High School Journal of Science 2026

quency representations was proposed. They found that compression removes visual artifacts from the image, but explicitly enhancing high-frequency information improves detection. This paper supports the mechanism behind a likely decline in my model's accuracy after JPEG compression.

Zhang et al. use a Multi-View Inconsistency Measurement, which uses a combination of noise-inconsistency stream and temporal-inconsistency stream.[13] They then detect and localize deepfake manipulations by combining different kinds of inconsistency. The data they use includes FFIW and associated video deepfake testing, testing the degradations of Gaussian blur, Gaussian noise, JPEG compression, and video compression. They report that combining noise inconsistencies with cross frame temporal inconsistencies improves robustness compared to using a single evidence source. This is extremely relevant because it is a direct comparison to the CNN-LSTM model used in this paper. The architecture of this paper focuses on spatial and temporal information specifically.

Balafrej et al. examines a more practical detector that is computationally efficient and resilient to real-world trans formations.[14] Specifically, they use a steganalysis-inspired SR-Net detector tested with RGB, YCbCr, and DCT-domain representations to analyze Celeb-DFv2 and DFDC data. The pipeline of this model includes face detection, dynamic frame sampling, and data augmentation. The degradations they utilized in training are JPEG compression, Gaussian noise, Gaussian blur, resizing, grayscale conversion, color changes, and brightness adjustments. Overall, they found that YCbCr representation achieved the strongest reported AUC values in the paper. This shows that degradation-oriented training can improve models across datasets. This is useful to this paper because the model used could be trained with realistic degradation augmentation in the future.

Zhang et al. use a generalizable detector, called IncoTune, which utilizes inconsistent cues instead of dataset-specific artifacts.[15] They used degradations of JPEG compression at quality levels of 75, 50, and 25 on the FaceForensics++, Celeb-DF, DFD, DFDC, and DFDCP datasets. In the reported experiment, the AUC declined from 91.66% on clean data to 70.15% at the strongest compression level. This demonstrates that preserving deepfake-detection performance is extremely difficult for several different methods. This is useful to understand for this paper because it shows excellent direct quantitative evidence for the JPEG-compression hypothesis.

Xu et al. compares the robustness of deepfake detectors under spatial and frequency-domain perturbations.[16] The authors utilize several models including ResNet-50, XceptionMinimal, and FreqNet, using data focusing on deepfake benchmarks used for those models. Several degradations are used, including JPEG compression at quality values from 20 to 100, Gaussian blur with σ = 1,2,4, resizing from 0.4 to 1.0 scale, frequency masking, and a combined resize–blur–JPEG transformation. Their main findings are that the Gaussian blur produced the largest difference among the tested model responses. ResNet-50 is what showed comparatively stable prediction-score behavior in the study's analysis. This is extremely useful for this study because the paper uses ResNet-50.

Clapten and Balaji target compressed/degraded video deepfake detection by modeling block-level video coding behavior and temporal artifacts using data from the Celeb-DF dataset.[17] They utilized IP-GTA Net: block-level 8×8 intra prediction reconstruction, hybrid convolutional autoencoder, MobileNetV3 feature extraction, gated convolutional recurrent units, and temporal attention. They ultimately found 90.52% accuracy and 0.9051 F1 and on Celeb-DF, exceeding comparisons with other models. This is extremely useful because it is a close architecture comparison because both approaches combine CNN visual features and recurrent temporal modeling.

Lagsoun et al. treats compressed deepfake detection as the main classification problem.[18] They utilize compressed deepfake data and use a genetic-algorithm and LASSO-based features selection followed by deep-learning classification. The authors report original compressed-deepfake detection experiments and propose selected features intended to remain useful despite compression. As can be seen, compressed deepfake detection is a recognized research problem.

Coccomini et al. ask whether modern deep-learning architectures trained on one type of forgery generalized to other manipulation types and datasets.[19] The data used is ForgeryNet for training/analysis and DFDC Preview for cross dataset evaluation. The models they used are EfficientNetV2 M, Vision Transformer, and Swin Transformer. During the experiment, they included random image compression and Gaussian-noise augmentation. They found that cross-dataset performance differed sharply from within-dataset performance. The study's reported Swin-Small model achieved 71.2% AUC in one broader-training cross-dataset configuration, showing that architecture choice and training diversity affect transfer. A limitation in this study is that for the model that is tested only on FaceForensics++ C23 may learn dataset-specific features, so a separate cross-dataset test is required before claiming general real-world robustness.

Pu et al. focus on improving deepfake-video detection by learning at both frame and video levels.[20] They used data from multiple video deepfake benchmarks with cross-dataset evaluation, using a dual-level model combining frame-level classification with video-level prediction and an AUC-oriented loss. They found that the system improves robustness by exploiting both local face artifacts and video-level evidence. This is highly relevant to this study because the ResNet-50 + LSTM structure depends on frame-level spatial evidence plus sequential temporal information, so this paper provides a scholarly

© The National High School Journal of Science 2026

precedent for combining the two levels.

Zhao et al. improve transfer between low-definition and high-definition deepfake video conditions.[21] They used low definition and high-definition deepfake video domains using semantic reduction and cross-domain training. The study evaluated the low-definition/high-definition mismatch directly and proposed training intended to reduce sensitivity to definition changes. This provides direct support for discussing resolution as a form of domain shift. Even if the underlying fake generation process is similar, the input distribution changes after downsampling, compression, or re-encoding.

Li et al. looked at how deepfake detection models can work on different sets of data without depending on certain features.[22] They used features and frequency guidance from masks and training that helps models work across areas. This helps signs of fake content from things like image compression or differences in how pictures are taken. This is helpful for the paper because it shows how changing the type of data a model is trained on can make CNNs less effective. When a model learns patterns that only work for data sets, like C23 or FaceForensics++ it can't find features that always show if something is a deepfake.

Petmezas et al. compare CNN and Transformer deepfake detectors under cross-dataset video-testing conditions.[23] They used FaceForensics++, Celeb-DF, DeepFakeDetection, and ReenactFaces datasets on the models ResNet-50, ResNet-101, CNN–LSTM, CNN with attention, Vision Transformer, and TimeSformer. They directed cross-dataset testing, limited target domain fine-tuning, and temporal-window analysis. They found that generalization depends strongly on both model architecture and training/testing domain. Fine-tuning and temporal context can improve transfer, but no model is uniformly robust in every cross-dataset setting. This is extremely relevant because it includes both ResNet-50 and CNN-LSTM systems.

Agrawal et al. compared multiple spatiotemporal archi tectures under realistic video deepfake conditions.[24] They compared the models of 3D CNN, 3D ResNet, temporal con volutional network, and variational autoencoder on the data of FaceForensics++, DFDC, and Celeb-DF. They found that crossdataset detection remained challenging. The study's highest reported cross-dataset test accuracy was 64.68% for its 3D CNN, underscoring the gap between controlled benchmarking and transfer to heterogeneous data. This is important for this study because a high FaceForensics++ test score may be valid for that dataset, but it cannot automatically establish performance under another dataset, manipulation method, or compression pipeline.

Ismail et al. combined spatial facial evidence and temporal video evidence in a unified deepfake detector.[25] They used the model of YOLO face detection, HOG-CNN and modified XceptionNet features, feature fusion, and stacked GRUs for temporal classification on the datasets of FaceForensics++ C23 and Celeb-DF. The approach shows that combined spatial and temporal evidence can improve video deepfake detection and supports mixed-source training. This is useful to this paper because it is a close conceptual precursor to the ResNet-50 + LSTM pipeline.

Lal et al. enhanced video deepfake detection through attention-weighted spatial features and LSTM-based temporal modeling.[26] It used face extraction, ResNeXt-50 visual features, soft attention, LSTM temporal sequence analysis, and softmax classification on deepfake video data. The proposed ResNeXt attention-LSTM model reported 92.88% accuracy on the RAW FaceForensics++ condition in the paper's evaluation. This is a direct structural comparison to the model used in this study: ResNeXt-50/ResNet-50 backbone features are aggregated by an LSTM. Temporal modeling is expected to capture blink, movement, warping, flicker, or inter-frame inconsistency information that a single-frame CNN may miss.

Rafique et al. detects deepfake images using error-level analysis and deep-learning features.[27] They use the method of Error-Level Analysis (ELA), CNN feature extraction, then k-nearest-neighbor and support-vector-machine classification on deepfake-image data. The authors report 89.5% accuracy for their best ResNet + k-NN configuration. This source is useful for explaining that JPEG encoding may sometimes create forensic evidence, such as inconsistent local compression errors.

Cunha et al. improve video deepfake detection by optimizing deep neural-network model configurations.[28] They use particle swarm optimization to tune/enhance deep neural-network design on video deepfake benchmarks. The authors report original comparative performance experiments showing the benefit of optimized model configurations. This is useful because it demonstrates how neural networks can be used for deepfake detection, which is utilized in this study.

Rajesh and Maruthupermal use multiple facial cues and attention mechanisms to identify deepfakes more reliably.[29] They test EfficientNet-B7 with attention, combining texture, face shape, landmarks, eye-blink behavior, and motion-related information on deepfake benchmark data. The article presents original experiments suggesting that multi-cue attention-based analysis can improve deepfake identification. This is useful to see the contrast between detectors relying mainly on ResNet visual features and LSTM sequences compared to multi-cue approaches.

Jiang et al. (2025) proposed a face-forgery detection and localization framework that combines high-frequency and texture-based features to improve reliability on compressed media.[30] Using FaceForensics++, Celeb-DF, and DFDC, the study reported performance improvements of 1% to 10% over comparison methods and emphasized that heavy JPEG compression can substantially reduce the visibility of manipulation

© The National High School Journal of Science 2026

artifacts.

An important part of training deepfake detection models is access to large datasets. Many researchers have used the FaceForensics++ Dataset. This dataset includes videos created from multiple different models such as autoencoders and GAN based models. This dataset has been used widely by researchers because it includes a variety of real-world conditions, including differences in lighting and other distractors. It is important to include real-world conditions when training models in controlled settings because it helps evaluate whether deepfake detectors remain successful when tested realistically.

Tables I–III show a comprehensive comparison of the related works investigated in this study.

A major challenge in deepfake detection is that real-world videos are often degraded due to compression during transmission. These degradations can remove or distort the visual cues that deepfake detection models rely on. Prior research has explored the effects of image degradation using CNN-based and other machine learning approaches. Building on this work, this study utilizes a model architecture that combines spatial feature extraction and temporal pattern analysis. Specifically, it focuses on video-based deepfake detection and the impact of degradation. To simulate real-world conditions, degradations such as blur, noise, and compression are applied. This study aims to evaluate how video degradation affects the performance and generalization of deepfake detection models using a CNN-LSTM architecture. It asks: How does visual degradation affect deepfake detection performance, and to what extent can degraded training improve the model robustness?

Methodology

This study employs a comparative experimental design to evaluate the impact of domain shift caused by video degradation on deepfake detection performance. Four experiments were conducted to test the model under clean and degraded training and testing conditions. In the first experiment, the model was both trained and tested on clean videos as a baseline. In the second experiment, the same clean-trained model was tested on degraded videos to understand domain shift. In the third experiment, the model was both trained and tested on degraded videos to understand whether degraded training would improve degraded-video performance. In the fourth experiment, the degraded-trained model was tested on clean videos to see whether degraded training could still generalize to clean data. This design helps compare model performance under matched and mismatched data conditions.

Fig. 1 shows the overall pipeline used in this study, from dataset selection and degradation to face detection, model prediction, and evaluation.

In this study, the publicly available deepfake dataset of FaceForensics++ was utilized.[31] This dataset contains 1000 original (real) videos and 6000 deepfake videos. To ensure consistency in manipulation type, the NeuralTextures subset was selected. This folder consists of deepfakes that focus on repetitive patterns to be able to generate hyper-realistic skin textures, clothing, and backgrounds.

Due to computational constraints, a balanced subset of the dataset was used for experimentation. 800 videos were selected for training and 200 videos for testing, with an equal number of real and deepfake samples to maintain class balance. The training set contained 400 real videos and 400 fake videos, while the test set had 100 real videos and 100 fake videos. These videos were chosen using a random balanced split with a fixed random seed of 42. Real and fake videos were shuffled separately. In addition to that, the train and test sets were checked to make sure that no video appeared in both sets. Ensuring that the dataset is balanced prevents the model from favoring the majority class, preventing shortcut learning and improving the model's generalization on new data. The exact train and test video ID lists were saved as supplementary files for reproducibility, including separate lists for real and fake videos in each split.

To better approximate real-world conditions that deepfake detection models see, such as social media compression, low bandwidth, and poor recording quality, image degradation was applied using Python and OpenCV-based image processing operations. The specific degradations were Gaussian blur, additive noise, and JPEG compression, which together represent common types of quality loss in online videos. Gaussian blur was applied using kernel sizes of 7, 9, or 11, with a randomly selected kernel size per video. Gaussian noise was added with a standard deviation which was randomly selected between 18 and 45 for each video. JPEG compression was also applied with a quality value randomly selected between 15 and 45 for each video. Since these values were selected using the fixed random seed of 42, it allows the degradation process to be reproducible. Each video kept the same degradation parameters across the sampled frames so that the video had a consistent degradation condition. The videos from the FaceForensics++ dataset were processed into frames for degradation as input. These frames were then resized and normalized to ensure that they were consistent. Degradations were applied frame-by-frame, with random parameter values chosen for each video so that the videos were degraded but still recognizable. This was done to prevent the model from being overfit into a single corruption level. The corruption order was Gaussian blur, followed by Gaussian noise, followed by JPEG compression. For Gaussian blur, sigma was randomly selected between 1.2 and 3.0 for each video. The degradation process used uint8 image frames with pixel values in the 0–255 range.

After the degradation of the videos was completed, the

© The National High School Journal of Science 2026

Table 1 Summary of Qualifying Peer-Reviewed Primary Research Studies Currently Included in the Related Work (Part I)

Study	Media Type	Method / Model Used	Dataset	Key Findings	Strengths	Limitations
Baocai et al. (2026) [6]	Video	Multiple detectors tested under visual and transmission distortions	Face-swap deepfake videos	JPEG compression, blur, noise, and bitrate reduction reduce detection performance	Directly studies real-world degradation	Focused on face-swap detection models
Lagsoun et al. (2025) [7]	Video	Compression-aware deepfake detection framework	Deepfake video benchmarks	Compression-aware design improves robustness under quality loss	Directly focused on compression	Less focused on blur and noise
Tian et al. (2025) [8]	Image/Video	Generalized feature adaptation and discrimination	Multiple deepfake datasets	Feature adaptation improves generalization across changing conditions	Strong domain-generalization focus	Not narrowly focused on degradation
Batagelj et al. (2025) [9]	Video	Multitask self-supervised learning	Multiple deepfake datasets	Self-supervised learning improves cross-dataset robustness	Strong generalization approach	More complex training process
Lu & Ebrahimi (2024) [10]	Image/Video	Robustness-assessment framework with degradation augmentation	FaceForensics++ raw, C23, C40	Common transformations reduce performance; controlled degradation training can improve robustness	Directly evaluates the same dataset quality variants	Augmentation can reduce clean-image performance
Li et al. (2023) [11]	Video	Spatial Restored Detection Framework	Low-quality social-media deepfake videos	Low resolution and heavy compression remove useful visual features	Directly studies low-quality compressed videos	Uses a specialized restoration framework
Gao et al. (2024) [12]	Image/Video	High-frequency enhancement network	Highly compressed deepfake content	High-frequency enhancement improves detection after compression	Directly studies compression	Primarily focused on high-frequency information
Zhang et al. (2025) [13]	Video	Multi-View Inconsistency Measurement	FFIW and deepfake video data	Combining noise and temporal inconsistencies improves robustness	Direct spatial and temporal comparison	Uses a different architecture
Balafrej et al. (2024) [14]	Image/Video	SRNet with RGB, YCbCr, and DCT representations	Celeb-DFv2 and DFDC	YCbCr achieved the strongest reported AUC values	Tests many realistic degradations	Different datasets and architecture

Table 2 Summary of Qualifying Peer-Reviewed Primary Research Studies Currently Included in the Related Work (Part II)

Study	Media Type	Method / Model Used	Dataset	Key Findings	Strengths	Limitations
Zhang et al. (2026) [15]	Image/Video	IncoTune	FF++, Celeb-DF, DFD, DFDC, DFDCP	AUC declined from 91.66% clean to 70.15% under strongest compression	Strong quantitative JPEG-compression evidence	Uses a vision-language approach
Xu et al. (2026) [16]	Image	ResNet-50, XceptionMinimal, FreqNet	Deepfake benchmarks	Gaussian blur caused large changes in detector responses	Directly evaluates ResNet-50	Primarily image-level analysis
Clapten and Balaji (2025) [17]	Video	IP-GTA Net with MobileNetV3 and recurrent temporal modeling	Celeb-DF	Reported 90.52% accuracy and 0.9051 F1	Close CNN plus recurrent architecture comparison	Different backbone and dataset
Lagsoun et al. (2026) [18]	Video	GA-LASSO feature selection and classification	Compressed deepfake data	Selected features remain useful despite compression	Treats compression as the primary problem	Different feature-selection approach
Coccomini et al. (2023) [19]	Video	EfficientNetV2-M, Vision Transformer, Swin Transformer	ForgeryNet and DFDC Preview	Cross-dataset performance differed sharply from within-dataset performance	Strong generalization evaluation	Not focused on CNN-LSTM
Pu et al. (2022) [20]	Video	Deep dual-level network	Multiple deepfake benchmarks	Frame-level and video-level evidence improve robustness	Strong spatial and video-level comparison	Different architecture
Zhao et al. (2025) [21]	Video	Semantic reduction and cross-domain training	Low- and high-definition deepfake domains	Cross-domain training reduces sensitivity to definition changes	Directly addresses quality-based domain shift	Focuses on definition mismatch
Li et al. (2025) [22]	Image/Video	Domain generalization and mask-guided supervision	Multiple deepfake datasets	Generalized features reduce dataset-specific dependence	Strong domain-generalization focus	Not specifically focused on blur/noise/JPEG

© The National High School Journal of Science 2026

Table 3 Summary of Qualifying Peer-Reviewed Primary Research Studies Currently Included in the Related Work (Part III)

Study	Media Type	Method / Model Used	Dataset	Key Findings	Strengths	Limitations
Petmezas et al. (2026) [23]	Video	ResNet-50, ResNet-101, CNN-LSTM, ViT, TimeSformer	FF++, Celeb-DF, DFD, ReenactFaces	Generalization depends on architecture and training/testing domain	Directly includes ResNet-50 and CNN-LSTM	Cross-dataset rather than degradation-focused
Agrawal et al. (2026) [24]	Video	3D CNN, 3D ResNet, TCN, VAE	FF++, DFDC, Celeb-DF	Cross-dataset detection remained challenging	Tests multiple spatiotemporal architectures	Does not isolate degradation effects
Ismail et al. (2022) [25]	Video	YOLO, HOG-CNN, XceptionNet, stacked GRUs	FF++ C23 and Celeb-DF	Combined spatial and temporal evidence improves video detection	Close conceptual comparison to CNN-LSTM	Uses different feature extractors
Lal et al. (2025) [26]	Video	ResNeXt-50, attention, LSTM	Deepfake video data	Proposed ResNeXt-attention-LSTM model reported 92.88% accuracy on RAW FF++	Very close structural comparison	Different dataset and attention design
Rafique et al. (2023) [27]	Image	ELA, CNN, k-NN/SVM	Deepfake image data	Best ResNet + k-NN configuration reported 89.5% accuracy	Shows JPEG encoding can create forensic evidence	Image rather than video detection
Cunha et al. (2024) [28]	Video	Particle Swarm Optimization with deep neural networks	Video deepfake benchmarks	Optimization improved model configurations	Shows importance of model optimization	Not specifically focused on degradation
Rajesh & Maruthuperumal (2026) [29]	Video	EfficientNet-B7 with attention and multiple facial cues	Deepfake benchmark data	Multi-cue attention improves deepfake identification	Uses texture, landmark, blink, and motion cues	Different architecture
Jiang et al. (2025) [30]	Image/Video	High-frequency and texture features	FF++, Celeb-DF, DFDC	Heavy JPEG and H.264 compression reduce manipulation artifacts	Direct connection to compression robustness	Also focuses on localization

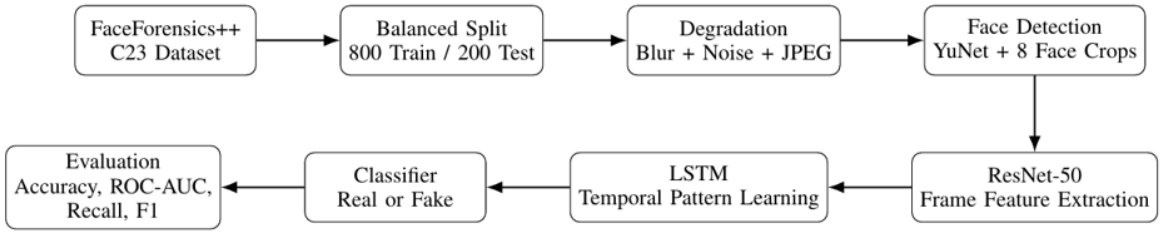


Figure. 1 Overview of the deepfake detection pipeline in this study.

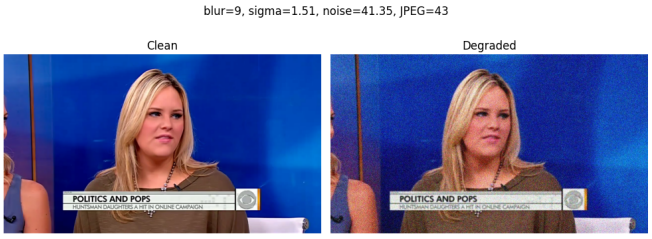


Figure. 2 Example of a clean frame and its degraded version after blur, Gaussian noise, and JPEG compression.

videos were prepared for model input and evaluation on the predefined training and testing sets. Next, it was important to convert the raw videos into a consistent format to input into the model. For this reason, every video was turned into a fixed-length sequence of face frames. This process was done using OpenCV, which allows individuals to read, write, and manipulate videos and images. The library also allows for face and pattern recognition in videos. The frames of the degraded videos were sampled at a fixed interval of every fourth frame to reduce repetitiveness among the videos. Since OpenCV reads videos in BGR, once they passed through that process, every one of these sampled frames was converted back to RGB to be passed through deep learning models. From the sampled frames, the faces from the videos were detected and cropped so that the focus of the model was on the faces where signs of deepfakes usually come through. Face detection was performed using the YuNet face detector, a lightweight CNN-based face detector available in OpenCV.[32] YuNet detects faces by predicting bounding boxes and confidence scores for faces in each frame. The highest-confidence face detection in each frame was selected and cropped before being resized to the model's input resolution. Each one of the videos was

© The National High School Journal of Science 2026

represented as a sequence of 8 face crops. If fewer than 8 faces were detected, the last detected face was repeated. If no face was detected in a frame, a center-cropped region of the image was used to maintain consistent input data. This process made sure that every video had the same input shape for the deep learning model. The same frame sampling, face detection, cropping, resizing, and normalization processes were used on both clean and degraded conditions. This was done so that the performance differences could clearly be attributed to degradation and the training condition instead of inconsistent preprocessing.

The deepfake detector model was built using two main parts known as a CNN, specifically ResNet-50, and a Long Short-Term Memory (LSTM). The CNN was used to analyze the sampled individual frames of the videos, and the LSTM was used to analyze how frames change over time. To extract important visual features from each frame, a ResNet-50 model was used. This is a 50-layer deep CNN introduced by He et al. (2016), which was originally trained on the ImageNet dataset.[5] Using the pretrained model helped the system start with an understanding of common visual patterns, making it stronger for image and video classification tasks. The ResNet-50 model turned each face image into a set of numbers describing the patterns, boundaries, facial structure details, and lighting inconsistencies. So, instead of working with pixels from the image, the model would work with these features that were already understood. This set of numbers is called a 2048-dimensional feature vector which was then passed to the LSTM model to evaluate. An LSTM is built to handle sequences since it processes data in a step-by-step manner.[33] After it has received the 8 feature vectors in a video from ResNet, it takes each one, which is 2048 numbers, and reads them in order. An LSTM functions by looking at the current frame's features, remembering important things from previous frames, deciding what to keep and what to forget, and updating its internal memory. This memory has a size of 384 numbers which represents patterns it has noticed so far and any inconsistencies. This 384-number summary is then fed into a classifier which examines the summary and determines if the video is real or fake. An attention pooling layer was used after the LSTM to help the model focus more on the most informative frames before making the final classification. The final model was a pretrained ResNet-50 feature extractor, a two-layer LSTM with a hidden size of 384, an attention pooling layer, and a binary classification head.

The cross-entropy loss was calculated when training the model, which is useful for binary classification tasks. It measures how wrong the model's predictions are compared to the true labels. Loss was calculated after the model predicted the category of each video. The AdamW optimizer then updated the model's weights to reduce the error.[34] AdamW was chosen because it combines adaptive learning rates with weight decay, which can help improve training stability and reduce overfitting. The model was trained with a learning rate of 3 $\times$ 10^{-4} and a weight decay of 2 $\times$ 10^{-4}. Cross-entropy loss with label smoothing was used for classification. A warmup plus cosine decay learning-rate schedule was used to stabilize early training and gradually reduce the learning rate over later epochs.

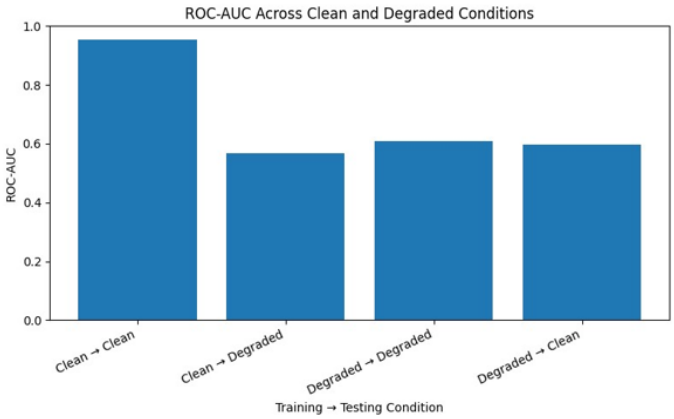


Figure. 3 ROC-AUC comparison across clean and degraded training/testing conditions.

Instead of training the model on the entire dataset all at once, the model was trained using smaller batches of videos. The batch size was four because using relatively small batches helped reduce GPU memory usage while processing sequences of video frames. The model went through the dataset multiple times since it was trained for 12 epochs. The experiments were trained using Google Colab with a CUDA-enabled GPU. Mixed precision training was used when available to reduce memory usage and improve training speed while keeping the same model structure and training procedure.

Two training conditions were used in this study: clean training and degraded training. Both the clean-trained model and degraded-trained model were each individually tested on degraded and clean test videos. This produced four experimental conditions for comparing matched and mismatched training and testing domains.

Results

To evaluate how the model performs, standard classification metrics commonly used in deepfake detection and image classification tasks were used. These include accuracy and ROCAUC, recall, and F1-score. Accuracy is the overall correctness of the model, recall measures how many samples from a class the model correctly identified, and ROC-AUC measures how well the model separates real and fake videos across clas-

© The National High School Journal of Science 2026

sification thresholds. The metrics apart from accuracy were included because accuracy by itself may hide whether the model performs differently on real and fake videos. Class predictions were produced using a binary classification threshold of 0.5.

Four experiments were evaluated to measure the effect of degradation on deepfake detection performance. The clean training/clean testing condition was used as the baseline. The clean training/degraded testing condition was used to see the effect of domain shift. The degraded training/degraded testing condition was used to test whether degraded training improved performance on degraded videos. The degraded training/clean testing condition was used to test whether the degraded training generalized back to clean videos.

The clean-training/clean-testing condition showed the strongest performance, with 91.0% accuracy and a ROC-AUC of 0.9546. The model correctly identified 86 out of 100 real videos and 96 out of 100 fake videos, showing that the CNNLSTM architecture can distinguish between real and fake videos when testing and training conditions are both clean.

However, when the same clean-trained model was tested on degraded videos, its performance dropped substantially. Accuracy went from 91.0% to 54.5%, and ROC-AUC decreased from 0.9546 to 0.5668. Fake recall also dropped from 0.9600 to 0.5300. This shows that the model missed many fake videos under degraded conditions. It suggests that blur, noise, and compression reduced the model's ability to detect manipulated videos.

Training directly on degraded videos did not substantially improve the model. The degraded-training/degraded-testing condition had an accuracy of 55.5% and a ROC-AUC of 0.6092. Although ROC-AUC improved slightly compared with the clean-training/degraded-testing condition, fake recall dropped to 0.3300. This means that the degraded-trained model became better at identifying real videos but worse at detecting fake videos.

The degraded-training/clean-testing condition achieved 58.5% accuracy and a ROC-AUC of 0.5970. This result shows that degraded training also did not generalize strongly back to clean videos. Overall, the results show that video degradation caused a major performance drop, and degraded-only training was not able to fully restore reliable deepfake detection performance.

Discussion

The results show that video degradation created a major domain shift for the deepfake detection model. The clean-training/clean-testing condition performed the best, with 91.0% accuracy and a ROC-AUC of 0.9546. This shows that the CNN-LSTM architecture was able to learn useful differences between real and NeuralTextures videos when the training and testing conditions matched. However, when the same clean-trained model was tested on degraded videos, its performance dropped to 54.5% accuracy and a ROC-AUC of 0.5668. This substantial drop in performance shows that the model did not generalize well when blur, noise, and compression were added to the videos.

This drop in performance is likely because degradation changes or removes the visual details that deepfake detection models rely on. In clean videos, the model can use facial textures, boundaries, lighting inconsistencies, and other visual information to distinguish real and fake samples. When degradation is applied, some of those details become harder to detect. Blur can soften facial boundaries, noise can add random patterns, and JPEG compression can remove texture and high-frequency details that detectors use to identify manipulation artifacts.[10] As a result, the model trained only on clean videos struggled to recognize fake videos under degraded conditions.

The degraded-training/degraded-testing condition showed only limited recovery. Its ROC-AUC increased from 0.5668 in the clean-training/degraded-testing condition to 0.6092, but its overall accuracy remained low at 55.5%. This means that training on degraded videos helped the model slightly separate real and fake videos, but it did not fully restore strong performance. One important pattern was that real recall increased to 0.7800, while fake recall dropped to 0.3300. This shows that the degraded-training model became more likely to correctly identify real videos, but it struggled more with detecting fake videos.

The degraded-training/clean-testing condition also had limited performance, with 58.5% accuracy and a ROC-AUC of 0.5970. This shows that degraded training did not generalize strongly back to clean videos either. In other words, training with only degraded data did not create a model that worked well across clean and degraded conditions. Instead, the model's behavior changed depending on the training condition, but neither degraded condition reached the performance of the clean baseline.

These findings are also consistent with prior work showing that real-world compression and visual degradation can reduce the features that deepfake detectors depend on. They also show why robustness is difficult in deepfake detection. A model that performs well in a controlled clean setting might not perform well once the videos are compressed, blurred, or noisy. This is especially important because videos shared online commonly undergo compression, re-encoding, or other quality changes before they are viewed or analyzed.[7,10,11]

These results also connect more directly to the prior studies in the literature review. Baocai et al. showed that common visual and transmission distortions such as JPEG compression, Gaussian noise, motion blur, and bitrate reduction can reduce deepfake detection performance.[6] This matches the substantial drop seen when the clean-trained model in the present study was tested on degraded videos. Lagsoun et al. sim-

© The National High School Journal of Science 2026

Table 4 Summary of Model Performance Across Clean and Degraded Training/Testing Conditions

Condition	Accuracy	ROC-AUC	Real Recall	Fake Recall	Real F1	Fake F1
Clean → Clean	0.9100	0.9546	0.8600	0.9600	0.9053	0.9143
Clean → Degraded	0.5450	0.5668	0.5600	0.5300	0.5517	0.5381
Degraded → Degraded	0.5550	0.6092	0.7800	0.3300	0.6367	0.4258
Degraded → Clean	0.5850	0.5970	0.6100	0.5600	0.5951	0.5744

ilarly focused on compression-aware deepfake detection and showed the importance of designing models that remain reliable after video quality is reduced.[7] Tian et al. addressed the broader problem of domain shift by developing feature-adaptation methods intended to improve generalization across changing conditions.[8] Batagelj et al. also showed that learning more transferable features can improve cross-dataset robustness.[9] Together, these studies support the present finding that strong performance in clean conditions does not necessarily transfer to degraded or otherwise changed testing conditions.

Overall, the results show that simply adding degraded training data is not enough to solve the robustness problem. More diverse training strategies may be needed, such as using multiple degradation levels, combining clean and degraded data more carefully, adding more manipulation types, or using larger datasets. The results support the idea that deepfake detection systems should be evaluated under realistic conditions instead of only clean benchmark settings.

Future work should test larger datasets, additional manipulation types, mixed clean-and-degraded training, and separate degradation levels to better understand how robust deepfake detectors can be improved.

Several limitations should be acknowledged. First, this study used 800 training videos and 200 test videos, which may limit how strongly the findings can be generalized. The experiments were also conducted using one fixed train/test split and one random seed, so future work should repeat the experiments across multiple seeds and report confidence intervals. Second, only NeuralTextures was used as the deepfake manipulation method for the fake class. Because of this, the findings may not fully extend to other manipulation techniques such as FaceSwap, Face2Face, DeepFakes, or FaceShifter. Third, the degradation process focused on combined blur, noise, and JPEG compression rather than separate blur-only, noise-only, compression-only, or severity-level experiments. Future work should test different degradation levels separately to better understand which distortions have the strongest effect. Fourth, this study did not compare the CNN-LSTM model against simpler baselines such as frame-average ResNet-50, smaller CNNs, or classical feature-based detectors. Finally, this study did not include saliency maps, failure-case frame examples, precisionrecall curves, or bootstrap confidence intervals. Future work could address these limitations by using larger datasets, testing multiple manipulation methods, comparing against baseline models, applying a wider range of realistic video degradations, and including more interpretability analysis. Future work could also include making the training, degradation, and evaluation scripts available to improve reproducibility.

Conclusion

This study investigated how the degradation of video data affects deepfake video detection using machine learning models. Four experiments were conducted using the same ResNet-50 and LSTM architecture across clean and degraded training and testing conditions. The results show that the model performed strongest when trained and tested on clean videos, achieving an accuracy of 91.0% and a ROC-AUC of 0.9546. However, when the clean-trained model was tested on degraded videos, performance dropped, leading to an accuracy of 54.5% and a ROC-AUC of 0.5668. This demonstrates that video degradation is a major barrier to deepfake detection.

Training on degraded videos only partially improved performance under degraded conditions. The degraded-training/degraded-testing condition reached an accuracy of 55.5% and a ROC-AUC of 0.6092, while the degraded-training/clean-testing condition reached an accuracy of 58.5% and a ROC-AUC of 0.5970. These results show that degraded training changed the model's behavior but did not fully restore reliable performance. Models trained only on clean footage may struggle to detect deepfakes in real-world environments where videos are often compressed, blurred, noisy, or otherwise degraded. These findings highlight the importance of evaluating deepfake detection models under realistic conditions and developing training strategies that improve robustness across both clean and degraded videos.

References

1 M. Westerlund, "The emergence of deepfake technology: A review," Technology Innovation Management Review, vol. 9, no. 11, pp. 40–52, 2019.

2 R. Hutchinson, "Deepfakes are the new frontier of cyberattacks: 5 steps to fight back." https://www.forbes.com/councils/forbestechcouncil/2025/01/06/deepfakes-are-the-new-frontier-of-cyberattacks-5-steps-to-fight-back/, 2025.

© The National High School Journal of Science 2026

3 I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Advances in Neural Information Processing Systems, vol. 27, 2014.

4 S. S. Makubhai, G. R. Pathak, and P. R. Chandre, "Exploring the trade-offs between blackbox and explainable ai: A comparative study," in Proceedings of the 7th International Conference on Computing, Communication, Control and Automation, IEEE, 2023.

5 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, pp. 770–778, 2016. doi: https://doi.org/10.1109/CVPR.2016.90.

6 L. Baocai, F. B. Azzali, and N. F. b. N. M. Farid, "Evaluating the real-world robustness of face-swap detection models under compression and noise," Frontiers in Artificial Intelligence, vol. 9, p. 1835651, 2026. doi: https://doi.org/10.3389/frai.2026.1835651.

7 A. M. Lagsoun, M. Oujaoura, and M. Hedabou, "Compression-aware hybrid framework for deep fake detection in low-quality video," IEEE Access, vol. 13, pp. 131980–131997, 2025. doi: https://doi.org/10.1109/ACCESS.2025.3592358.

8 Z. Tian and J. Yi, "Gfade: Generalized feature adaptation and discrimination enhancement for deepfake detection," PeerJ Computer Science, vol. 11, p. e2879, 2025. doi: https://doi.org/10.7717/peerj-cs.2879.

9 B. Batagelj, A. Kronovšek, V. Štruc, and P. Peer, "Robust cross-dataset deepfake detection with multitask self-supervised learning," ICT Express, vol. 11, no. 5, pp. 858–862, 2025. doi: https://doi.org/10.1016/j.icte.2025.02.011.

10 Y. Lu and T. Ebrahimi, "Assessment framework for deepfake detection in real-world situations," EURASIP Journal on Image and Video Processing, vol. 2024, p. 6, 2024. doi: https://doi.org/10.1186/s13640-024-00621-8.

11 Y. Li, S. Bian, C. Wang, K. Polat, A. Alhudhaif, and F. Alenezi, "Exposing low-quality deepfake videos of social network service using spatial restored detection framework," Expert Systems with Applications, vol. 231, p. 120646, 2023. doi: https://doi.org/10.1016/j.eswa.2023.120646.

12 J. Gao, Z. Xia, G. L. Marcialis, C. Dang, J. Dai, and X. Feng, "Deepfake detection based on high-frequency enhancement network for highly compressed content," Expert Systems with Applications, vol. 249, p. 123732, 2024. doi: https://doi.org/10.1016/j.eswa.2024.123732.

13 B. Zhang, Q. Yin, W. Lu, and X. Luo, "Deepfake detection and localization using multi-view inconsistency measurement," IEEE Transactions on Dependable and Secure Computing, vol. 22, no. 2, pp. 1796–1809, 2025. doi: https://doi.org/10.1109/TDSC.2024.3472064.

14 I. Balafrej and M. Dahmane, "Enhancing practicality and efficiency of deepfake detection," Scientific Reports, vol. 14, p. 31084, 2024. doi: https://doi.org/10.1038/s41598-024-82223-y.

15 L. Zhang, S. Peng, M. Xu, and T. Lu, "Towards generalizable deepfake detection: An inconsistency-aware vision–language model tuning approach," Sensors, vol. 26, no. 14, p. 4512, 2026. doi: https://doi.org/10.3390/s26144512.

16 H. Xu, "Frequency-aware robustness analysis of deepfake detection models," Journal on Artificial Intelligence, vol. 8, no. 1, pp. 153–167, 2026. doi: https://doi.org/10.32604/jai.2026.078014.

17 J. E. Clapten and V. Balaji, "A gated temporal attention based intra prediction framework for robust deepfake video detection," Scientific Reports, vol. 15, p. 38540, 2025. doi: https://doi.org/10.1038/s41598-025-22333-3.

18 A. M. Lagsoun, O. Khouili, A. Bekkari, N. Boudra, M. Oujaoura, A. Jraifi, et al., "Compressed deepfake detection via ga-lasso selection of deep features and machine learning models," Scientific Reports, vol. 16, p. 19708, 2026. doi: https://doi.org/10.1038/s41598-025-34733-6.

19 D. A. Coccomini, R. Caldelli, F. Falchi, and C. Gennaro, "On the generalization of deep learning models in video deepfake detection," Journal of Imaging, vol. 9, no. 5, p. 89, 2023. doi: https://doi.org/10.3390/jimaging9050089.

20 W. Pu, J. Hu, X. Wang, Y. Li, S. Hu, B. Zhu, R. Song, Q. Song, X. Wu, and S. Lyu, "Learning a deep dual-level network for robust deepfake detection," Pattern Recognition, vol. 130, p. 108832, 2022. doi: https://doi.org/10.1016/j.patcog.2022.108832.

21 C. Zhao, C. Wang, Z. Song, G. Hu, L. Wang, and D. Miao, "Multi-definition deepfake detection via semantics reduction and cross-domain training," Pattern Recognition, vol. 163, p. 111469, 2025. doi: https://doi.org/10.1016/j.patcog.2025.111469.

22 J. Li, Y. Hu, B. Liu, H. She, and C.-T. Li, "Deepfake detection with domain generalization and mask-guided supervision," Pattern Recognition, vol. 165, p. 111622, 2025. doi: https://doi.org/10.1016/j.patcog.2025.111622.

23 G. Petmezas et al., "Cross-dataset video deepfake detection using transformer and cnn architectures," Machine Vision and Applications, vol. 37, p. 52, 2026. doi: https://doi.org/10.1007/s00138-026-01809-w.

24 P. Agrawal, D. Pathak, V. Madaan, et al., "Spatiotemporal deep learning for real-time video-based deepfake detection using 3dcnn, 3dresnet, tcn, and vae," Scientific Reports, vol. 16, p. 18200, 2026. doi: https://doi.org/10.1038/s41598026-49090-1.

25 A. Ismail, M. Elpeltagy, M. S. Zaki, et al., "An integrated spatiotemporal-based methodology for deepfake detection," Neural Computing and Applications, vol. 34, pp. 21777–21791, 2022. doi: https://doi.org/10.1007/s00521-02207633-3.

26 K. Lal, S. Shiwani, and G. C. Gandhi, "Deepfake video deception detection using visual attention-based method," Scientific Reports, vol. 15, p. 40089, 2025. doi: https://doi.org/10.1038/s41598-025-23920-0.

27 R. Rafique, R. Gantassi, R. Amin, et al., "Deep fake detection and classification using error-level analysis and deep learning," Scientific Reports, vol. 13, p. 7422, 2023. doi: https://doi.org/10.1038/s41598-023-34629-3.

28 L. Cunha, L. Zhang, B. Sowan, et al., "Video deepfake detection using particle swarm optimization improved deep neural networks," Neural Computing and Applications, vol. 36, pp. 8417–8453, 2024. doi: https://doi.org/10.1007/s00521-024-09536-x.

29 S. T. Rajesh and S. Maruthupermal, "Generalizable deepfake detection framework using hybrid convolution-based efficientnetb7 with attention mechanism," Expert Systems with Applications, vol. 297, p. 129357, 2026. doi: https://doi.org/10.1016/j.eswa.2025.129357.

30 Q. Jiang, S. Liu, S. Miao, et al., "Robust manipulated media localization and detection based on high frequency and texture features," Discover Computing, vol. 28, p. 10, 2025. doi: https://doi.org/10.1007/s10791-025-09500-w.

31 A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1–11, 2019. doi: https://doi.org/10.1109/ICCV.2019.00009.

32 W. Wu, H. Peng, and S. Yu, "Yunet: A tiny millisecond-level face detector," Machine Intelligence Research, vol. 20, no. 5, pp. 656–665, 2023.

33 S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997. doi: https://doi.org/10.1162/neco.1997.9.8.1735.

34 I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in International Conference on Learning Representations, 2019.

© The National High School Journal of Science 2026