Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.081091

ARTICLE

Seeing through Deepfakes: An Explainable Multi-Task Detection Framework with Deep Learning and Large Language Models

Jiyeong Park[1], Sercan Yeşilköy[1], Doyeon Lim[1], Huiryeong Park[1], Eunseo Lee[1], Mohsen Ali Alawami[1,*] and Ki-Woong Park[2,*]

[1]Division of Computer Engineering, Hankuk University of Foreign Studies, Yongin, Republic of Korea
[2]Department of Information Security, Sejong University, Seoul, Republic of Korea

*Corresponding Authors: Mohsen Ali Alawami. Email: mohsencomm@hufs.ac.kr; Ki-Woong Park. Email: woongbak@sejong.ac.kr

Received: 23 February 2026; Accepted: 24 June 2026; Published: 13 August 2026

ABSTRACT: The recent increase in deepfake content has significantly increased cyber threats. Although numerous deepfake detection technologies have achieved high accuracy, there are limits to clarifying the rationale behind their detection decisions. To bridge the gap, in our study, we leverage the combination of Explainable Artificial Intelligence (XAI) and Large Language Models (LLMs) to deliver clear, consistent, and understandable interpretations of deepfake detection outcomes. To do that, we integrate XAI and LLMs to visually represent detection rationales and automatically generate coherent natural-language explanations. During the implementation of our method, we developed a multi-task learning framework based on a Convolutional Neural Network (CNN) combined with a Long Short-Term Memory (LSTM) architecture, trained simultaneously for binary classification (real or fake) and multi-class classification (identifying specific deepfake techniques) using the FaceForensics++ dataset. The CNN architecture considered for this model included ResNeXt50-32x4d, EfficientNet-b0, and Xception, with EfficientNet-b0 ultimately selected as the optimal detection model based on superior performance metrics on the FaceForensics++ dataset. EfficientNet-b0 achieved the best performance, with an accuracy of 95.14% for binary classification and 94.29% for multi-class classification. Additionally, a web-based interface was developed to allow intuitive inspection of the detection results, combining visual and textual explanations to enhance interpretability.

KEYWORDS: Deepfake detection; multi-task learning; XAI; LLM; CNN-LSTM architecture

1 Introduction

The term “Deepfake” is a portmanteau of deep learning and fake, using artificial intelligence to create a fake video or image that is difficult to distinguish from real material. According to the World Economic Forum [1], deepfake content has increased by more than 900 percent over the past three years. Additionally, the Avast Threat Labs [2] reported that AI technologies that are abused, including deepfakes, have significantly increased in cyber threats. Deepfake videos rapidly spread across online platforms, such as YouTube and Instagram, making it harder to block their widespread diffusion. To minimize its potential harm effectively, reliable deepfake detection methods are essential.

However, most existing research based on deepfake detection has simply focused on its binary classification, whether it’s real or fake. Feng et al. [3] suggested the method using a trained Xception backbone to learn contrastive features that push fake faces and real faces apart. Then, make a binary classification whether real or fake by 2D features. Li et al. [4], on the other hand, used a “face X-ray” approach that

Copyright © 2026 The Authors. Published by Tech Science Press. This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

shows blending boundaries by analyzing inherent noise and error-level artifacts across synthetic composites. Since this method is training by only real images, it shows its result only by whether it's real or fake. The high accuracy means it's likely a real image. Zhao et al. [5] used a multi-attentional deepfake detection method, which performs fine-grained classification using multiple spatial attention heads, a textural feature enhancement block, and a regional independence loss with attentional-guided augmentation. Wang and Deng [6] and Rössler et al. [7] represent deepfake detection, showing the results for real or fake.

Although prior studies have achieved strong performance in deepfake detection using CNNs, transformers, or physiological cues, they still share several common limitations. These methods leave users without sufficient evidence to support their results, which undermines the reliability of their outcomes. First, existing explanation methods are primarily limited to visual interpretations, often relying on heatmap visualizations without structured interpretation, which may not be sufficient for users to fully understand model decisions. While visual explanations using Gradient-weighted Class Activation Mapping (Grad-CAM) are effective in presenting results, they still leave users to interpret the outcomes on their own, which can lead to interpretations that differ from the intended meaning. Second, there is a lack of region-level quantitative interpretability, making it difficult to identify which facial regions contribute most to the prediction. This also makes it challenging to systematically record and evaluate the results in a quantitative manner. Third, most approaches do not provide natural-language explanations grounded in model evidence, making it difficult to clearly communicate the underlying causes of deepfake detection results to others. Fourth, existing methods offer limited practical inspection tools, making it difficult for users to explore and understand model decisions in an integrated manner.

To address these limitations, our study integrates XAI and LLMs into deepfake detection systems, enabling users to better understand model decisions through both visual and textual explanations grounded in model evidence. Our approach extends interpretability by incorporating Region of Interest (ROI)-based facial region quantification. This enables users to identify which facial regions contribute most to the prediction. Furthermore, it translates region-level evidence into natural-language rationales using LLMs. In this research, we propose an explainable deepfake detection framework that incorporates advanced CNN and LSTM models to effectively classify manipulated videos. Specifically, the pre-processing stage uses face detection with the Multi-task Cascaded Convolutional Networks (MTCNN) algorithm, followed by precise facial cropping using OpenCV. The model is applied to prominent benchmark datasets, including FaceForensics++ [7], DeeperForensics [8], and Celeb-DF [9]. In addition, the processed data are systematically divided into training and testing subsets to enable experimental evaluation. Our approach integrates pre-trained CNN models such as ResNeXt50-32x4d, Xception, and EfficientNet-b0 to extract critical spatial features from face images. These features are then sequentially analyzed by an LSTM network designed to capture the temporal dependencies inherent in video sequences.

Furthermore, the adoption of XAI techniques, specifically Grad-CAM, provides visual interpretations of the model's predictions, aiming to improve transparency. In addition, LLM generates textual explanations by converting region-level evidence into natural-language rationales. The entire detection system is conveniently delivered through a web-based interface, facilitating an accessible visualization and enhanced interpretability for end-users. The web interface was developed to provide more detailed explanations about the results. It shows the original video as well as the video that shows the cropped faces with its Grad-CAM on it. The interface allows users to inspect visual evidence associated with the prediction and may assist in interpreting the classification results.

We emphasize that the novelty of our study does not lie in proposing a new CNN or LSTM backbone. Rather, it lies in integrating multi-task spatio-temporal detection with ROI-quantified Grad-CAM evidence

and LLM-based explanation generation within a unified inspection framework. Unlike conventional deepfake detection studies that mainly focus on predicting accuracy, the proposed system framework is designed to simultaneously address detection performance, visual interpretability, and human-understandable explanation generation in a single end-to-end pipeline. By comparing CNN-LSTM-based temporal feature learning, region-level XAI analysis, and structured prompt-based LLM explanations, the proposed system provides an explainable deepfake analysis process that integrates detection and interpretation modules within a unified framework.

The main contributions of our study are summarized as follows. First, we propose a unified and explainable multi-task deepfake detection framework that jointly performs binary deepfake detection and manipulation-type classification by integrating CNN-LSTM spatio-temporal modeling. Second, we develop a structured explainability pipeline that combines Grad-CAM visualization, ROI-level activation, quantification, and prompt-based LLM generation to transform model evidence into human-understandable natural-language rationales. Third, to support interpretability for end-users, we implement a prototype user-facing inspection interface that integrates prediction results, visual explanations, ROI statistical analysis, and textual explanations into a single deepfake analysis system with a practical prototype-level inference time.

To achieve this, we combine XAI with LLMs to visualize detection reasons and generate clear natural-language explanations. Experiments and qualitative analyses, including pairwise comparison of generated explanations, suggest that the proposed framework can provide more informative and evidence-grounded explanations for end-users while maintaining competitive deepfake detection performance. Our results show that the framework can detect deepfakes with an accuracy of 95.14% for binary classification and an Area Under the Curve (AUC) of 0.993 using the FaceForensics++ dataset.

The remainder of this paper is structured as follows. In Section 2, we review related works of recent deepfake detection approaches. We present the framework design overview in Section 3 and describe details of the methodology, modules, and technologies employed throughout the work in Section 4. Section 5 shows experiment settings and demonstrates a comparative analysis of deepfake detection evaluation performance and validates the rationale for detection decisions. We discuss the limitations and practical implications in Section 6. We finally provide our conclusion and future directions of our study in Section 7.

2 Related Work

With the growing realism and accessibility of Deepfake generation technologies, detecting manipulated videos has become a crucial challenge. As deep learning-based synthesis methods become more robust, distinguishing fake content from authentic videos—especially for non-expert users—becomes increasingly difficult. In this section, we examine existing research efforts related to Deepfake detection, with a particular focus on approaches that overlap with our proposed framework, which incorporates CNN + LSTM modeling, explainability via XAI, and LLM integration.

Early physiological-based detection methods, such as In Ictu Oculi [10] and DeepVision [11], leverage irregularities in human eye blinking patterns, exploiting temporal inconsistencies not easily replicated in generated videos. These methods highlight the usefulness of physiological signals as indicators of forgery. Motion-based strategies, including Deepfake Video Detection through Optical Flow-based CNN [12], apply optical flow analysis to detect unnatural movements across frames. Similarly, Learning Self-Consistency for Deepfake Detection [13] introduces temporal coherence checks across facial dynamics to spot inconsistencies. Several CNN-based methods have explored spatial artifacts introduced during face manipulation. For example, Multi-attentional Deepfake Detection [5] employs attention mechanisms to enhance feature learning, while Lips Don't Lie. Ref. [14] proposes a robust pipeline that focuses on mouth movements as key indicators. Deepfake Detection by Analyzing Convolutional Traces [15] identifies specific noise patterns

left by CNN architectures used in generation. Cross-domain robustness has become a central challenge. Liu et al. [16] use residual federated learning to generalize across datasets, while [17,18] focus on improving detection performance across compression rates and video domains. Jeong et al. [19] introduce frequency perturbations to improve robustness, and Lai et al. [20] employs a contrastive learning strategy to tackle compression-specific degradation. Transformer-based models, such as DFDT [21] and M2TR [22], leverage the representational power of Vision Transformers and multi-scale attention mechanisms, respectively, for end-to-end detection. These models have shown impressive performance on benchmark datasets. From a sequence modeling perspective, Deepfake video detection using Recurrent Neural Networks [23] represents one of the earlier attempts to model temporal dependencies using RNNs, while our method builds on this by combining CNNs with LSTMs. Several works focus on identity-driven and semantic consistency. Cozzolino et al. [24] utilizes identity mismatches introduced during generation as classification cues and Yang et al. [25] explores the relational structure of facial regions to identify subtle inconsistencies. Explainability has gained increasing attention in recent studies. Explainable Deepfake detection using visual interpretability methods [26] and unmasking Deepfake faces [27] adopt saliency and cost-sensitive techniques to provide human-understandable insights into model predictions.

Explainable Deepfake detection using CNN and CapsuleNet [28] introduces capsule networks for deepfake detection by modeling spatial relationships between facial features. Dang et al. [29] present a comprehensive deepfake detection framework that generalizes across multiple manipulation techniques. It provides a scalable and extensible architecture, making it suitable for adaptable deepfake detection systems.

Most recently, Wang et al. [17] analyzes artifacts in synthesized images and shows how CNN-based detectors can identify them. It provides insights into model interpretability and detection transparency, supporting explainable deepfake detection. The emergence of vision-language models introduces new directions for explainable and generalizable detection. Recent surveys have highlighted the growing role of LLMs in cybersecurity applications, including threat detection, incident response, cyber-resilience, explainability, and decision support. Patel et al. [30] provided a comprehensive survey of explainable deepfake detection methods and identified interpretability as a key requirement for the practical deployment of deepfake forensic systems. This observation further motivates the integration of LIME-based visual explanations and LLM-generated textual rationales in our framework. Aleem et al. [31] propose an explainable deepfake detection framework using multiple CNN models to enhance interpretability that focuses on generating visual explanations to understand model decisions and improves transparency, but mainly relies on spatial (frame-level) features.

Despite the increasing use of XAI techniques in deepfake detection, existing approaches still exhibit several important limitations. Methods such as Grad-CAM primarily highlight salient regions in the input but do not provide clear semantic explanations of the model's decisions. As a result, the generated heatmaps can be difficult to interpret, particularly for non-expert users, and often lack precise, region-level justification. Moreover, many studies rely on qualitative visual explanations without offering quantitative measures or translating visual evidence into human-readable textual descriptions. These limitations reduce the practical usability of XAI methods and highlight the need for more comprehensive and interpretable explanation frameworks.

Table 1 provides a comparison of existing deepfake detection methods based on their model architectures, temporal modeling capabilities, explainability techniques, and cross-dataset performance.

Table 1: Summary of related works on deepfake detection.

Study	Method	Temporal	XAI	LLM/VLM	Binary	Multi	Cross-Dataset	Deployment
[10]	CNN + LSTM	LSTM	✗	✗	✓	✗	✗	✗
[12]	CNN	None	✗	✗	✓	✗	✗	✗
[11]	Attention	None	✗	✗	✓	✗	✗	✗
[5]	3D CNN	Implicit	Occlusion	✗	✓	✗	✗	✗
[16]	Federated Learning	None	Grad-CAM	✗	✓	✗	✓	✗
[21]	Vision Transformer	Transformer	✗	✗	✓	✗	✗	✗
[25]	Relation Learning	None	Grad-CAM	✗	✓	✗	✗	✗
[15]	KNN/SVM/LDA	None	✗	✗	✓	✗	✗	✗
[19]	CNN (ResNet50)	None	✗	✗	✓	✗	✗	✗
[22]	CNN + Transformer	Transformer	✗	✗	✓	✗	✗	✗
[26]	CNN	None	LRP	✗	✓	✓	✗	✗
[27]	CNN	None	Grad-CAM	✗	✓	✓	✗	✗
[18]	CNN + RL	None	✗	✗	✓	✓	✓	✗
[28]	CNN + Capsule + LSTM	LSTM	Grad-CAM	✗	✓	✗	✗	✗
[29]	CNN	None	✗	✗	✓	✓	✗	✗
[31]	CNN	None	Grad-CAM	✗	✓	✓	✗	✗
[32]	Multi-modal CNN	None	Rationales	GPT-based	✓	✓	✗	✗
[33]	CNN + LSTM + Transformer	LSTM	✗	✗	✓	✓	✓	✗
[20]	Prompt Learning	None	✗	LLM-based	✓	✗	✗	✗
[13]	Self-consistency	None	✗	✗	✓	✗	✗	✗
[34]	Transformer	Attention	Attention Map	✗	✓	✓	✓	✗
[35]	VLM-based	None	Multimodal Reasoning	LLaVA/LLM	✓	✗	✓	✗
[36]	CNN + Transformer	Spatio-temporal Attention	✗	✗	✓	✓	✓	✗
Our study	CNN + LSTM	LSTM	Grad-CAM	OLLAMA	✓	✓	✓	✓

Note: [✓] denotes the presence or support of a specific capability, while [✗] denotes its absence or lack of support.

In contrast, our study integrates CNN + LSTM for spatio-temporal learning, while this uses only CNN. Also, we combine XAI and LLM-based textual explanations, not only visual explanations. In addition, we perform multi-task learning for both binary and multi-class classification, which is not addressed in their work. Bharati et al. [32] introduce a multi-model explainable framework designed for forensic and legal contexts that focuses on producing human-interpretable rationales for deepfake detection and emphasizes explainability for decision support rather than model efficiency. On the other hand, our study is a single unified CNN-LSTM framework, unlike Bharati's multi-model ensemble-based approach. Furthermore, we integrate LLMs for automatic explanation generation, which they do not use. One key distinction is that our framework is designed as a practical prototype that integrates detection and explanation modules rather than solely for forensic analysis. Petmezas et al. [33] propose a hybrid model combining CNN + LSTM + Transformer for deepfake detection using 3D facial modeling (3DMM) and captures both short- and long-term temporal dependencies. Compared with our lightweight model, their model is computationally heavy (Transformer + 3DMM), and we focus on explainability (XAI + LLM), which is not their main focus. In addition, our framework targets multi-task classification + interpretability, not only accuracy. DeFaX [34] introduces a cross-attention fusion framework for robust and explainable deepfake detection. The method integrates hierarchical spatial representations using transformer-based attention mechanisms to capture subtle forgery artifacts. The framework also improves interpretability through attention-based

localization and demonstrates strong cross-dataset robustness. Yu et al. [35] explore the use of large vision-language models (LVLMs) for explainable and generalizable deepfake detection. The framework combines visual prompt embeddings, forgery localization, and LLM-based reasoning to generate interpretable textual explanations. Experimental results demonstrate improved cross-dataset generalization and multimodal reasoning capability. Thakre et al. [36] propose a cross-attentive spatio-temporal fusion framework that jointly models spatial and temporal forgery patterns. The model uses transformer-based attention to dynamically fuse temporal and spatial representations for robust deepfake detection. The method achieves intra-domain and cross-domain performance across multiple deepfake benchmarks.

In summary, recent studies have explored hybrid deep learning architectures and explainable AI for deepfake detection, including CNN-based models, CNN–LSTM frameworks, and Transformer-based approaches for capturing spatial and temporal features. While these methods provide a strong foundation, several important limitations remain.

First, many approaches primarily focus on spatial features and lack effective temporal modeling, limiting their ability to capture dynamic inconsistencies in deepfake videos. Second, although XAI techniques such as Grad-CAM are widely used, they mainly generate visual heatmaps without structured or human-readable explanations, making interpretation difficult for non-expert users. Third, most existing works focus on binary classification, with limited support for multi-task learning that handles both binary and multi-class detection. Finally, many models are computationally complex and do not sufficiently consider lightweight and real-time deployment scenarios.

To address these limitations, our study proposes a unified and lightweight framework that integrates CNN–LSTM for spatio-temporal modeling, XAI for visual interpretability, and LLM-based modules for generating human-readable explanations. In addition, the proposed approach supports multi-task learning for both binary and multi-class classification while maintaining practical efficiency for real-world deployment. This design directly aligns with and addresses the research gaps identified in the Introduction.

3 Proposed System Design

3.1 Framework Architecture

In this research, an explainable deepfake detection framework is provided, which goes beyond simply providing detection outcomes by supplying evidence for each decision. Fig. 1 illustrates the overall system, which consists of five stages: data preprocessing, deepfake detection, visual explanation based on XAI, textual explanation based on LLM, and an accessible result visualization interface. The LLM was not additionally trained or fine-tuned on Grad-CAM outputs. Instead, Grad-CAM activation summaries were converted into structured text prompts containing region names, activation percentages, and predicted labels. The LLM then generated natural-language explanations based on these structured prompts.

The proposed system framework begins with collecting the real and deepfake videos as the dataset, which includes FaceForensics++ [7], DeeperForensics [8], and Celeb-DF [9]. Specifically, FaceForensics++ is utilized as the primary training dataset, while Celeb-DF and DeeperForensics are employed as external test datasets. For the preprocessing process, faces will be recognized by the MTCNN algorithm [37] and cropped to 224 × 224 using OpenCV. According to this, the system will generate cropped face datasets for both training and testing. In the model training phase, the model leverages pretrained CNN architectures such as ResNeXt50-32x4d, Xception, or EfficientNet-b0 to extract spatial features, which are then fed into an LSTM to capture temporal dynamics.

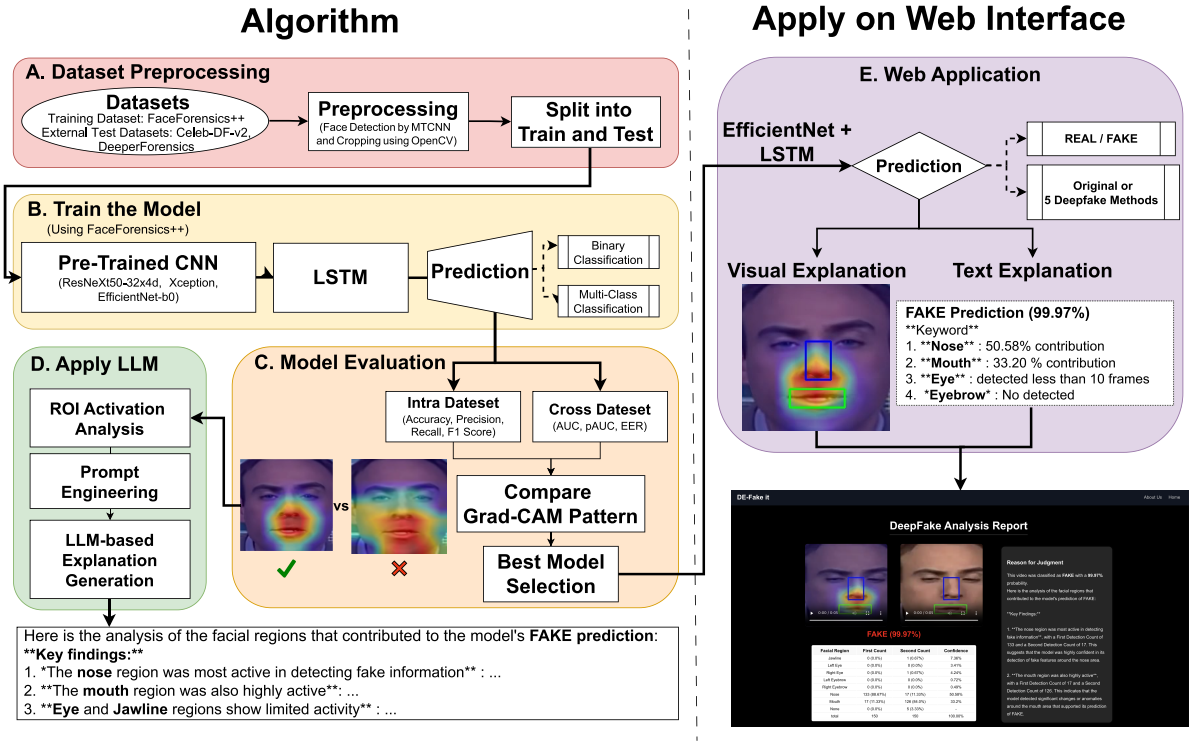


Figure 1: Architecture of the proposed deepfake detection framework, combining CNN-LSTM classification, LLM-based explanation, and a web interface.

Eq. (1) defines the CNN-based spatial feature extraction stage of the framework. Given an input frame x_t, the CNN extracts discriminative facial representations f_t that capture spatial forgery artifacts such as texture inconsistencies, blending traces, and manipulation patterns. These extracted spatial features serve as input to the subsequent LSTM model for temporal dependency modeling across video frames.

$$f_t = \mathrm{CNN}(x_t) \tag{1}$$

Eq. (2) models temporal dependencies between consecutive video frames using the LSTM network. Since deepfake artifacts may vary across frames and exhibit temporal inconsistencies, the hidden state h_t enables the framework to capture sequential contextual information beyond individual frame-level analysis. This temporal modeling improves robustness against frame-level noise and enhances video-level deepfake classification performance.

$$h_t = \mathrm{LSTM}(f_t, h_{t-1}) \tag{2}$$

Performance of the trained model is evaluated using classification metrics such as accuracy, precision, recall, and F1 score for intra-dataset evaluation, and AUC, partial Area Under the Curve (pAUC), and Equal Error Rate (EER) for cross-dataset scenarios. Based on the model's prediction, the input video is classified as either "real" or "fake". Complementing this, an XAI technique, specifically Grad-CAM, is applied to visualize model-relevant facial regions. Grad-CAM visualizations are further quantified through ROI-based activation analysis, and the resulting regional evidence is used to generate textual explanations via LLM. Finally, the web system provides the prediction results together with visual, quantitative, and textual information for both original videos and videos manipulated with various deepfake techniques.

The datasets used in our study, including their attributes, size, and usage, are summarized in Table 2. Specifically, FaceForensics++ c23 was utilized for model development, covering the training, validation, and internal testing phases. In contrast, Celeb-DF and DeeperForensics were used exclusively for external cross-dataset testing to examine the model's cross-dataset performance.

Table 2: Dataset summary and usage in our study.

Dataset	Purpose	Classes	Videos	Real/Fake	Usage (Train/Val/Int/Ext)
FaceForensics++ (c23) [7]	Model Development	6	7000	2000/5000	Train, Val, Internal Test
Celeb-DF (v2) [9]	External Testing	2	6228	589/5639	External Cross-dataset Test
DeeperForensics [8]	External Testing	2	2000	1000/1000	External Cross-dataset Test

3.2 Website Interface Design

The detected results and supporting rationales must be clear to the users to improve their understanding. Specifically, the web interface is included as a practical demonstration layer that allows users to inspect predictions, heatmaps, and textual explanations in one place. It is not the primary methodological contribution, but rather a deployment-oriented component supporting interpretability. To provide enhanced interpretability and an accessible visualization interface, a system comprising a React front-end (https://react.dev/) and Django back-end (https://www.djangoproject.com/) is provided as shown in Fig. 2. As the user uploads a video, preprocessing on the server begins, and each frame is analyzed by the detection model. For both real and fake videos, the Grad-CAM is applied, then a visual explanation based on XAI is created within a natural language LLM description to present both visual and textual rationales to facilitate intuitive understanding for the user.

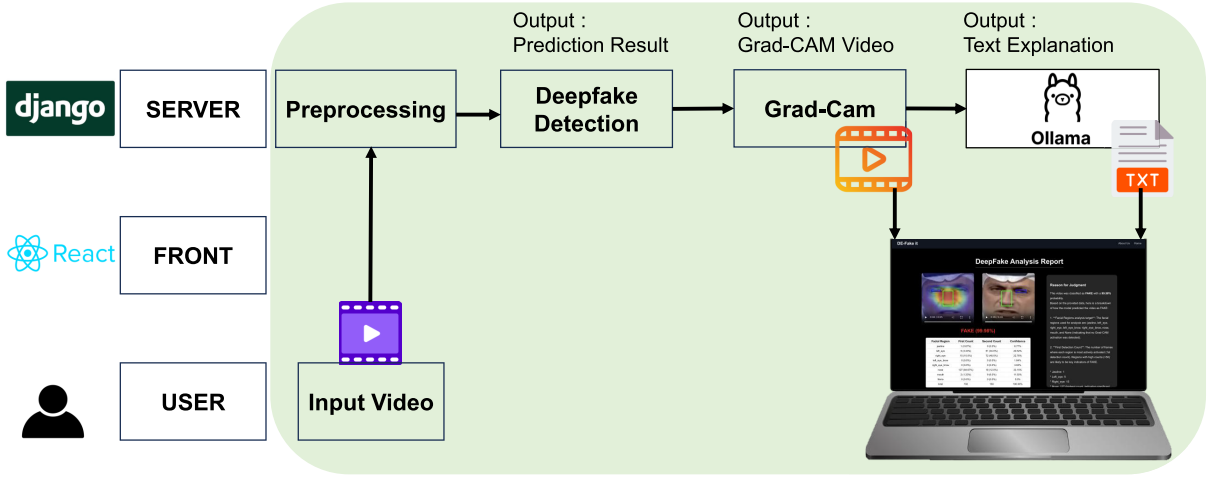


Figure 2: Overview of the deepfake web server processing, showing video input, prediction, visual (Grad-CAM), and text explanation outputs.

The Web interface guides users through the process of uploading a video and viewing the corresponding results. On the start page, users can upload a video for classification. Then, after pressing the "detect" button, the system presents the analysis results on the result page. At the top of the result page, two videos are displayed. These are the videos that have been cropped from the original uploaded video using face detection techniques. The displayed videos include Grad-CAM visualizations and ROI activation overlays. Below the

videos, the binary classification, whether the uploaded video is real or fake will be shown using color-coded labels. Then the textual explanation created by the LLM is shown at the bottom of the page. It provides detailed explanations of why the video has been classified as fake or real, so that users can understand the reasons behind the results.

3.3 Data Flow and Main Processes

When a video is used as input, frames are extracted and face detection is performed. The detected face regions are then resized by 224 × 224. Preprocessed frames are saved on the server in Motion Joint Photographic Experts Group (MJPEG) format. Preprocessed videos are then put into deepfake detection model and return prediction results indicating whether the video is fake or real. After prediction, Grad-CAM generates heatmaps overlaid on original frames, and ROI Activation highlights two regions with the highest activation scores using bounding boxes. Extracted data, such as Grad-CAM scores and ROI Activation values, are used as inputs to the LLM to generate textual explanations automatically. These results are then compiled and presented to the user through the developed website interface. The current explanation module is primarily grounded in region-level activation statistics rather than fine-grained pixel-level semantic descriptors. Expanding the system to produce more detailed visual attribute explanations is left for future work.

4 Overall Deepfake Methodology

4.1 Main Framework Components

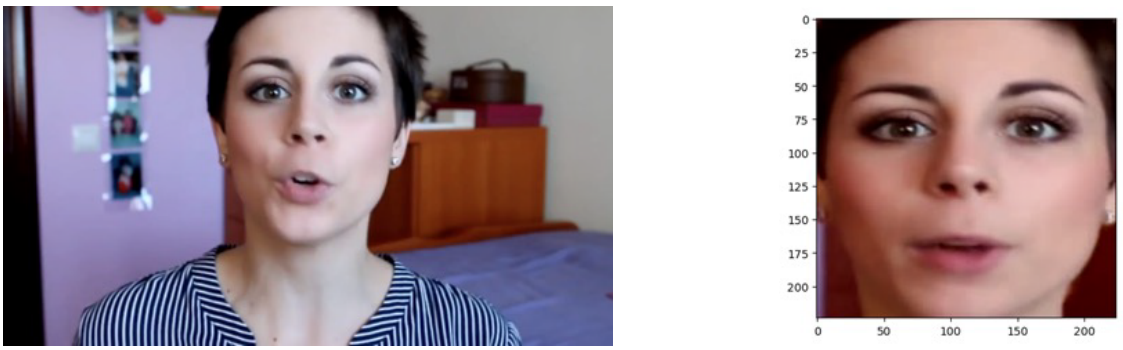
4.1.1 Data Preprocessing

To provide suitable input representations for deepfake detection, we designed a preprocessing pipeline that generates face-centric time series input data. This pipeline consists of five stages and aims to simultaneously obtain numerical and temporal stability for CNN and LSTM-based models. In particular, we adopted the MTCNN, which can detect facial regions and simultaneously extract key landmarks such as eyes, nose, and mouth. MTCNN consists of a lightweight cascaded architecture composed of P-Net, R-Net, and O-Net, and demonstrates robust performance across various face sizes, angles, and expressions. These characteristics enable a stable and precise generation of face-centric time series data.

We summarize the five stages of the preprocessing as follows:

1. Frame extraction: We sequentially extracted 150 frames from each video. This process enabled the generation of fixed-length inputs.
2. Face detection: MTCNN was used to detect faces in each frame.
3. Background removal: Based on the bounding box coordinates returned from MTCNN, the first detected face was selected and the corresponding rectangular region was cropped.
4. Resizing: The cropped images were resized to 224 × 224 pixels and fed into CNN-based models with a consistent size.
5. MJPEG reconstruction: We reconstructed preprocessed face images into MJPEG format.

Fig. 3 illustrates the comparison of preprocessing results between the original video and the preprocessed face region obtained using MTCNN and the resizing process. This preprocessing pipeline was implemented in Python using the following open-source libraries. For face detection, we used the MTCNN implementation provided by the facenet-pytorch library (https://github.com/timesler/facenet-pytorch), while frame extraction, cropping, resizing, and video encoding were performed using OpenCV (https://github.com/opencv/opencv-python).



(**a**) Original Video (**b**) Preprocessed Video

Figure 3: An illustrative comparison of preprocessing results. (**a**) Frame extracted from the original video. (**b**) Corresponding preprocessed face region obtained using MTCNN, resized to 224 × 224 pixels, and centered for input into the detection model.

4.1.2 Deepfake Detection Models

The combination of CNN and LSTM has already been shown to be effective for deepfake detection in previous studies [38,39]. CNN excels at feature extraction by identifying frame-level anomalies such as texture distortions and discontinuous boundaries characteristic of deepfake content. Additionally, LSTM is well-suited for capturing temporal inconsistencies such as irregular eye blinking or unnatural mouth movements. Therefore, several CNN architectures were evaluated to determine the most effective backbone for the proposed framework. Specifically, ResNeXt50-32x4d, Xception, and EfficientNet-b0 were selected as representative CNN architectures due to their proven effectiveness in capturing spectral artifacts and visual inconsistencies in deepfake detection applications.

ResNeXt builds on the ResNet residual connection architecture by introducing a new dimension called cardinality, referring to the number of parallel transformations [40]. Unlike traditional methods that increase the model capacity by deepening or widening the network, ResNeXt increases representational power by grouping multiple smaller transformations in parallel. Specifically, ResNeXt50-32x4d denotes a 50-layer deep model with a cardinality of 32 and a bottleneck width of 4, offering a balanced trade-off between model complexity and computational efficiency. The architecture maintains the simplicity and modularity of ResNet by reusing the residual block structure while integrating grouped convolutions, making it easy to scale. This design choice allows ResNeXt to process feature maps more diversely, enhancing its ability to capture subtle variations and patterns in visual data.

In the context of deepfake detection, we particularly used ResNeXt50-32x4d because of its grouped convolution mechanism that enables the network to extract fine-grained spatial anomalies such as facial contour distortions, artifacts near boundary regions, and inconsistent skin textures. Therefore, ResNeXt is a strong candidate for real-time deepfake detection systems because it has shown improved generalization and better performance on benchmark datasets with only a marginal increase in computational cost. Moreover, when combined with temporal modeling such as LSTMs or with XAI modules, the features extracted by ResNeXt50-32x4d serve as a robust backbone for multi-task learning tasks, including binary and multi-class classification of deepfake generation methods.

Xception extends the Inception architecture by replacing the traditional Inception modules with depthwise separable convolutions [41,42]. This technique decouples the learning of spatial and channel-wise features, where a depthwise convolution is first applied independently to each input channel, followed by a point-wise (1 × 1) convolution to capture cross-channel interactions. This factorization significantly reduces

the number of parameters and computational cost compared to standard convolutions, while preserving model expressiveness and performance. Unlike Inception, which employs multiple convolutional filters of varying sizes in parallel, Xception relies entirely on depthwise separable convolutions in a linear stack of modules, making the architecture simpler, easier to optimize, and more elegant. In the domain of deepfake detection, Xception has demonstrated outstanding performance due to its ability to capture fine-grained textures and subtle spatial inconsistencies such as color blending mismatches, inconsistencies around facial landmarks (e.g., eyes, lips, and jawlines), localized artifacts introduced by Generative Adversarial Networks (GANs), and abnormal pixel-level patterns.

Xception is widely used in deepfake detection with frame-based and video-level models to extract features that significantly contribute to both binary classification (real/fake) and multi-class classification (identifying techniques such as Deepfakes, Face2Face, or NeuralTextures). Moreover, the Xception model can be deployed for real-time scenarios such as mobile deepfake detectors on benchmark datasets like FaceForensics++ and Celeb-DF.

EfficientNet-b0 introduces a compound scaling method that optimally balances model depth, width, and resolution, simultaneously improving performance and efficiency [43]. It applies a compound coefficient (ϕ) to uniformly scale all three dimensions according to fixed ratios, which makes it computationally efficient compared to the traditional approaches that arbitrarily scale only one network dimension (e.g., CNN). Specifically, EfficientNet-b0 demonstrates high performance despite its lightweight design, making it highly adaptable to various image resolutions and ideal for real-time or mobile environments. The EfficientNet family consists of models from b0 to b7, where b0 serves as the baseline model. Additionally, EfficientNet-b0 includes key features such as Mobile Inverted Bottleneck Convolution (MBConv) layers, which incorporate depthwise separable layers for reduced computation. Another feature is the squeeze-and-excitation (SE) optimization to reduce parameter count and adjust channel weights for feature recalibration.

Before training, each file must be confirmed whether it is not damaged after preprocessing to ensure the quality of the input data. Training is performed with a batch size of 8 and two sub-processes. The model is trained by branching into binary classification that distinguishes between fake and real, and multi-class classification that classifies deepfake techniques. The loss is calculated as the sum of the two classifications to simultaneously optimize the two classifications, and the accuracy is tracked for the binary classification that is relatively more important.

4.1.3 Visual Explanation Based on XAI

Deep learning models possess a "black box" characteristic due to their complex internal structures and non-linear operations, making it difficult to interpret their prediction processes. To improve interpretability, the proposed framework employs Grad-CAM and ROI Activation analysis as XAI-based visualization techniques. Grad-CAM is used to identify the most discriminative facial regions contributing to the prediction, while ROI Activation quantitatively measures activation levels across predefined facial regions to provide structured regional evidence for the explanation generation process.

Grad-CAM is class-discriminative, which highlight regions most relevant for a particular class. To do that, Grad-CAM leverages gradient information from the last convolutional layer of CNNs and visually emphasizes the regions with high activation, indicating areas that contribute most significantly to the model's prediction.

Grad-CAM localization maps are computed as:

$$L^{Grad-CAM} = ReLU\left(\sum_k \alpha_k A^k\right) \tag{3}$$

where A^k represents feature maps and α_k denotes the importance weights derived from gradients.

Eq. (3) computes a class-discriminative localization map by weighting the final convolutional feature maps according to their gradients with respect to the predicted class. In this framework, it is used to identify facial regions that contributed most strongly to the real/fake prediction. The resulting heatmap is later aggregated into ROI-level activation scores, which are used as structured evidence for the LLM explanation module. Grad-CAM works with a wide range of CNN-based architectures (ResNet, Xception, etc.). Unlike older methods (like CAM), Grad-CAM does not require re-training or architectural modifications.

In addition to Grad-CAM visualization, ROI Activation analysis is performed to quantify activation levels across different facial regions based on the generated heatmaps. When Grad-CAM visualizations span the entire face, it can be challenging to pinpoint specific reasoning areas. ROI Activation calculates activation scores for facial regions, such as eyes, nose, and mouth, using the face-alignment library (https://pypi.org/project/face-alignment/).

4.1.4 Text Explanation Based on LLM

To provide textual explanations of the deepfake detection results, a prompt-engineered Llama 3 model running on Ollama was employed. After the model prediction has been completed, information extracted from all frames is used as input for the LLM. Inputs include the deepfake method, highly activated facial areas identified by Grad-CAM, areas identified through ROI Activation, and specific activation levels for facial regions. These structured inputs are converted into prompt templates that guide the explanation generation process. Formally, the prompt generation process can be interpreted as a mapping function:

$$P = f(R, A, M) \tag{4}$$

where R denotes the ROI activation statistics, A represents Grad-CAM attention patterns, and M indicates the predicted manipulation class. The overall explanation pipeline integrating Grad-CAM-based visual evidence, ROI-level activation aggregation, and prompt-based LLM explanation generation is summarized in Fig. 4.

Algorithm: XAI-Driven LLM Explanation Pipeline
1. Compute Grad-CAM heatmaps from selected frames
2. Map activation scores to predefined facial ROIs
3. Aggregate ROI activation statistics across frames
4. Construct structured prompt input
5. Generate natural-language explanation using the LLM

Figure 4: Pipeline for generating LLM-based explanations from XAI-derived regional evidence.

This approach provides detailed and consistent explanations. As illustrated in Fig. 4, this structured pipeline enables the LLM to generate detailed and consistent explanations grounded in model-derived regional evidence.

4.2 Binary and Multi-Class Classification

A multi-task learning structure simultaneously performing binary and multi-class classification was implemented for deepfake detection. The binary classification distinguishes between real and fake classes, while the multi-class classification includes FaceForensics++'s five deepfake methods and the original class. The overall loss function is defined as $loss = loss_{\text{bin}} + loss_{\text{method}}$.

$$L = \lambda_1 L_{\text{bin}} + \lambda_2 L_{\text{method}} \tag{5}$$

Eq. (5) defines the overall multi-task optimization objective used during model training. The framework simultaneously performs binary deepfake detection and manipulation-type classification; therefore, the total loss combines both objectives using weighted balancing parameters (λ_1) and (λ_2). This formulation enables the model to jointly learn general forgery characteristics and method-specific manipulation patterns within a unified training framework.

$$\text{class\_weight}_i = \frac{N}{C \cdot n_i} \tag{6}$$

The training process primarily monitored binary classification accuracy ($\text{Accuracy}_{\text{bin}}$), while multi-class accuracy ($\text{Accuracy}_{\text{method}}$) was recorded for post-analysis. Due to a class imbalance where the fake class data outnumbered real class data, we applied a weighting of (0.7:1.75) to the loss calculation. The weight was calculated using Eq. (6), following Scikit-learn's "compute_class_weight" function, where N is the total number of samples, C is the number of classes, and n_i is the number of samples in class i. Eq. (6) computes class weights to mitigate dataset imbalance during optimization. Since manipulated and original samples are not equally distributed across training datasets, the weighted loss function prevents the model from becoming biased toward majority classes. These class weights are incorporated into the cross-entropy loss to improve learning stability and balanced classification performance.

Fig. 5 shows the pipeline process of conditional CAM based on the prediction results from the model. In detail, the model ultimately produces two classification predictions, and we use these results for cross-validation within the XAI framework and conditionally apply them to the Grad-CAM visualization. If the prediction is "REAL" from binary classification and "Original" from multi-class classification, indicating a non-artificially generated video, CAM is deleted. In contrast, if the prediction is "FAKE" in the binary classification and one of the deepfake methods (not "Original" class) in the multi-class classification, CAM is amplified by a factor of 1.5. After the CAM is normalized to a range between 0 and 1, it is overlaid onto the frame.

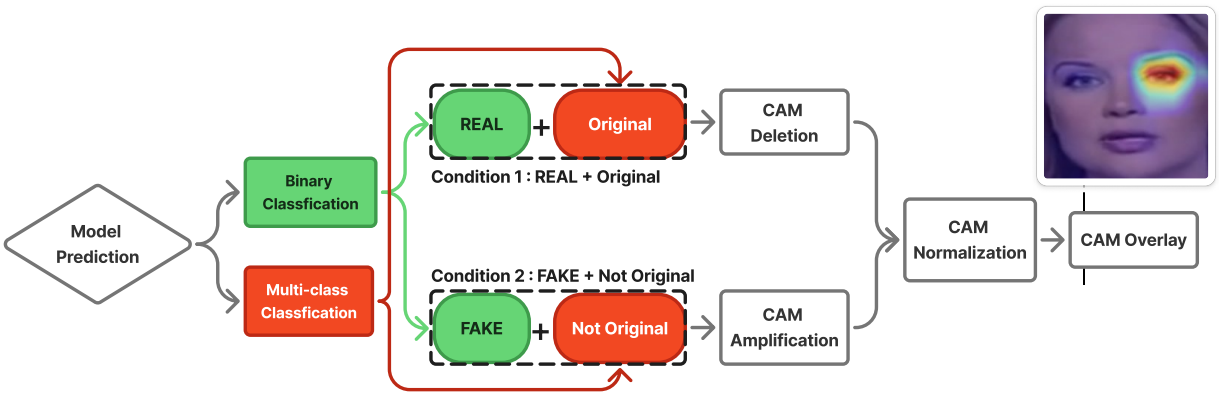


Figure 5: Conditional CAM processing pipeline based on model prediction.

4.3 LLM Prompt Tuning Method

To generate reliable explanations using an LLM, the quality of the data used for prompt tuning is critical. We first analyze the Grad-CAM activations and quantify the importance of each facial region. We captured seven facial regions, such as eyes, nose, and mouth, using a face alignment library and first counted the top-2 most activated facial regions per frame. To analyze which facial regions significantly contributed to the model's fake prediction, we define four metrics as follows.

The total number of times each region was ranked first or second was then calculated across all frames. The first detection rate and second detection rate, defined in Eqs. (7) and (8), respectively, are obtained by dividing these counts by the total number of frames. A high detection rate indicates that the corresponding region was frequently activated as the main reason for being judged as fake. Eq. (9) defines the weighted activation score, which accumulates the activation score across all frames, using the model's predicted probability of being fake as a weight. To enable relative comparison between regions, the contribution score in Eq. (10) is calculated by normalizing the weighted activation scores across all regions. Since this measure reflects both spatial activation and prediction confidence, it provides a clearer indication of each region's contribution to the fake prediction.

These aggregated ROI activation scores provide structured regional evidence that summarizes spatial attention patterns across frames and serve as key inputs for the subsequent explanation generation stage.

$$\text{FirstDetectionRate}_r = \frac{\text{FirstDetectionCount}_r}{N} \times 100 \tag{7}$$

$$\text{SecondDetectionRate}_r = \frac{\text{SecondDetectionCount}_r}{N} \times 100 \tag{8}$$

$$\text{WeightedActivationScore}_r = \sum_{i=1}^{N} \text{CAMScore}_i \cdot \text{ROIScore}_{i,r} \tag{9}$$

$$\text{Contribution}_r = \frac{\text{WeightedActivationScore}_r}{\sum_{r' \in \mathcal{R}} \text{WeightedActivationScore}_{r'}} \times 100 \tag{10}$$

The statistical data calculated from the above formula were input to the LLM. We differentiated the prompts based on whether the prediction was REAL or FAKE. In the case of REAL, Eqs. (7) and (8) are mainly referenced to explain why the video is difficult to view as a deepfake. In contrast, for the FAKE case, all four Eqs. (7)–(10) were referenced to explain which facial part showed signs of forgery. Additionally, the method's prediction result from multi-class classification was used to supplement the explanation. Each LLM-generated response was returned with numbered points for readability. The description of all the notations used in the ROI contribution analysis is summarized in Table 3.

Table 3: Notations used in the ROI contribution analysis.

Symbol	Description
r	Facial region (e.g., mouth, nose, eyes, etc.)
N	Total number of video frames
$\text{FirstDetectionCount}_r$	Number of frames where region r had the highest activation
$\text{SecondDetectionCount}_r$	Number of frames where region r had the second highest activation
CAM Score_i	Predicted probability of frame i being fake

(Continued)

Table 3 (continued)

Symbol	Description
$\text{ROIScore}_{i,r}$	Average Grad-CAM activation of region r in frame i
$\text{WeightedActivationScore}_r$	Cumulative weighted activation score for region r over all frames
Contribution_r	Normalized contribution (%) of region r to the final prediction

The datasets used in our study, including their attributes, size, and specific usage for model development and external cross-dataset testing, are summarized in Table 4.

Table 4: Dataset sample distribution. The second FaceForensics++ c23 entry corresponds to the processed dataset used for training and evaluation.

Dataset	Real Videos	Fake Videos	Total Videos	Role in Experiment	Notes
FaceForensics++ c23 [7]	1000	5000	6000	Source dataset	Before balancing
FaceForensics++ c23	2000	5000	7000	–	Real samples increased
Training set	1436	3604	5040	Training	80% of development set
Validation set	364	896	1260	Validation	20% of development set
Internal test set	200	500	700	Internal Test	Held-out set
Celeb-DF-v2 [9]	589	5639	6228	Cross-dataset test	Short videos (<150 frames) excluded
DeeperForensics [8]	1000	1000	2000	Cross-dataset test	Balanced real and fake samples

5 Experimental Setup and Results

In this section, we present the details of the experimental setup, including the dataset and the methods used for evaluation. We then provide the results and demonstrate the performance validation and outcomes of the proposed framework.

5.1 Experimental Setup

5.1.1 Dataset

We use FaceForensics++ c23 [7] for the model training, while the DeeperForensics [8] and Celeb-DF-v2 [9] were used as external cross-dataset benchmarks to measure the model's cross-dataset performance. These datasets are publicly available for research use, subject to their respective access policies. The FaceForensics++ c23 dataset is publicly available through its official repository (https://github.com/ondyari/FaceForensics). Similarly, the DeeperForensics-1.0 dataset can be accessed through its official repository (https://github.com/EndlessSora/DeeperForensics-1.0), while the Celeb-DF-v2 dataset is available through its official repository (https://github.com/yuezunli/celeb-deepfakeforensics). FaceForensics++ c23 is a high-quality compressed dataset that is advantageous for learning fine-grained features. It contains 1000 real videos and 5000 manipulated videos generated using five deepfake methods: Deepfakes, Face2Face, FaceSwap, NeuralTextures, and FaceShifter. 150 frames were extracted from each video, and face detection was applied to retain only frames containing detected faces. Because the dataset is imbalanced, with significantly fewer real videos than manipulated ones, we increased the number of real samples. For longer real videos, we generated additional samples by extracting frames from non-overlapping temporal segments within the same

video. As a result, the final dataset of FaceForensics++ c23 used in our study contains 2000 real videos and 5000 manipulated videos, yielding a total of 7000 video samples after balancing and preprocessing. These were randomly split at the video level into 6300 development videos and 700 internal test videos. The 6300 development videos were further randomly partitioned into 5040 training videos and 1260 validation videos, corresponding to an 80:20 split. All splits were performed at the video level to avoid frame leakage, as summarized in Table 4.

To examine cross-dataset performance under selected external test conditions, we evaluated the model using Celeb-DF [9] and a balanced subset of DeeperForensics [8]. For Celeb-DF, all videos were used except for a single video with fewer than 150 frames, as short clips may lack sufficient temporal information for reliable evaluation. For DeeperForensics, we constructed a balanced subset by selecting 1000 real and 1000 fake videos. This was done to ensure a fair evaluation by removing the influence of class imbalance and focusing on the model's ability to distinguish between real and manipulated videos.

5.1.2 Hyperparameter Settings

Table 5 summarizes the hyperparameter settings used for training all models in our study, including input size, number of frames per video, optimizer, learning rate, batch size, number of epochs, loss function, and regularization strategies.

Table 5: Hyperparameter settings used in the study for Model A (ResNeXt50-LSTM), Model B (Xception-LSTM), and Model C (EfficientNet-LSTM).

Setting	Model A (ResNeXt50-LSTM)	Model B (Xception-LSTM)	Model C (EfficientNet-LSTM)
Input size	224 × 224	224 × 224	224 × 224
Frames per video	10	10	10
Optimizer	Adam	Adam	Adam
Learning rate	1e−4	1e−4	1e−4
Batch size	8	8	8
Epochs	30	30	30
Loss function	CrossEntropy	CrossEntropy	CrossEntropy
Train/Val split	80:20	80:20	80:20
LSTM hidden size	2048	2048	2048
Dropout	0.5	0.5	0.5
Early Stopping	No	No	No
Learning Rate Scheduler	ReduceLROnPlateau (patience = 3, factor = 0.5)	ReduceLROnPlateau (patience = 3, factor = 0.5)	ReduceLROnPlateau (patience = 3, factor = 0.5)
Hardware	NVIDIA Tesla T4	NVIDIA Tesla T4	NVIDIA Tesla T4

In detail, all video frames were resized to 224 × 224 and sampled into sequences of 10 frames for training. Cross-entropy loss was used for both binary and multi-class classification tasks, and the final loss was defined as the sum of the two losses. The dataset was split into training and validation sets with a ratio of 80:20, and a dropout layer with a rate of 0.5 was applied to improve generalization. Rather than using early stopping, a ReduceLROnPlateau scheduler was used to adaptively adjust the learning rate based on validation loss. The best-performing model was selected based on the highest validation AUC and saved during training.

5.1.3 Evaluation Approaches

The evaluation framework was designed to assess the proposed framework from three perspectives: model classification performance, the effectiveness of Grad-CAM-based visual explanations, and the consistency of LLM-generated textual explanations. Model performance evaluation was conducted using intra-dataset and cross-dataset experiments to analyze classification accuracy and cross-dataset performance. In addition, the Grad-CAM evaluation was performed to investigate the activation patterns and ROI-based attention distributions associated with manipulated facial regions. Finally, the generated LLM-based textual explanations were qualitatively verified to examine their consistency with the Grad-CAM-derived regional evidence.

The model performance evaluation was conducted by an intra-dataset test and a cross-dataset test. In the intra-dataset test, the performance was evaluated based on top-1 accuracy and macro average values of Precision, Recall, and F1-score using FaceForensics++. Additionally, T-SNE was employed to visually assess the feature distribution and class separability. For cross-dataset evaluation, DeeperForensics and Celeb-DF were used as external test sets, and the model was evaluated based on AUC, pAUC at 10% False Positive Rate (FPR), and EER to measure cross-dataset performance.

To further analyze the interpretability of the proposed framework, Grad-CAM was used to visualize the regions that influence the model's prediction of fake samples. First, the difference in Grad-CAM activation patterns between real and fake samples was visually assessed. Fake videos showed high activation in regions that are commonly associated with forgery artifacts. To enable quantitative comparison among models and the five deepfake methods in the FaceForensics++ dataset, we selected the top-1 frame with the highest probability of being fake. These frames were used to compare both the visualization results and the corresponding activation levels. Additionally, we quantified the model's attention using ROI-based activation scores. We focused on facial ROIs such as the eyes, nose, and mouth, and calculated the average activation scores within these areas based on the Grad-CAM heatmaps. Using these values, we conducted a quantitative comparison to evaluate Grad-CAM.

To evaluate the consistency of the explanation pipeline, the generated LLM-based textual explanations were qualitatively verified using ROI-based activation statistics extracted from Grad-CAM visualizations. In detail, we calculated the average ROI-based activation scores across all frames, and these statistical values are used to construct structured prompts provided to the LLM. This approach enables the LLM to produce natural language descriptions of the model outputs based on ROI-based activation patterns. For example, if the mouth region shows the highest activation score, the prompt includes specific statistics, such as being detected 70 times, to guide the LLM to focus its explanation on that area. We use the LLM as a post-hoc explanation generator rather than a predictive model. Accordingly, we do not apply quantitative evaluation metrics. Instead, we qualitatively examine the consistency of the explanations by examining whether the regions mentioned in the explanations correspond to those with the highest activation scores. This qualitative analysis suggests that the generated explanations are generally consistent with the ROI-based Grad-CAM evidence, particularly for fake samples where manipulated facial regions are more prominently highlighted.

5.2 Results

5.2.1 Model Performance Evaluation

We conducted experiments in binary and multi-class classification on the FaceForensics++ internal test data, comprising 700 test samples across five deepfake generation methods. Table 6 presents the classification performance of the models. Both Xception and EfficientNet-b0 demonstrated high performance in binary

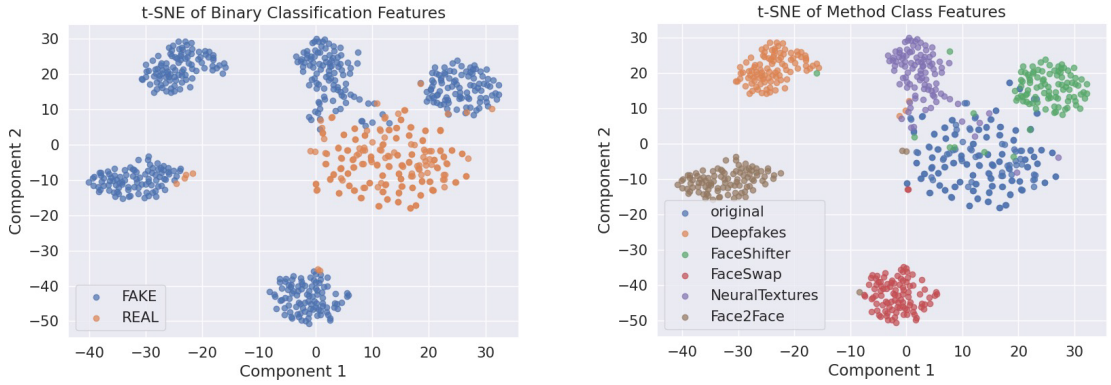
classification, achieving a top-1 accuracy of 95.14% for binary classification. However, EfficientNet-b0 is superior in multi-class classification with a top-1 accuracy of 94.29%.

Table 6: Classification results (Accuracy, F1, Precision, and Recall) for binary and multi-class tasks on FaceForensics++ dataset.

Model	Classification	Accuracy	Macro F1	Macro Precision	Macro Recall
ResNeXt50-32x4d	Binary	0.94	0.93	0.94	0.91
Xception		**0.95**	**0.94**	**0.95**	0.93
EfficientNet-b0		**0.95**	**0.94**	0.93	**0.95**
ResNeXt50-32x4d	Multi-class	0.93	0.80	0.80	**0.81**
Xception		0.93	0.80	0.80	0.80
EfficientNet-b0		**0.94**	**0.81**	**0.82**	**0.81**

Note: Bold values indicate the best performance among the compared models for each metric.

As illustrated in Fig. 6a, 6t-SNE shows that EfficientNet-b0 successfully separates real and fake videos into distinct clusters. Notably, the fake cluster splits into several subgroups, indicating that the model captures variations among different types of fake generation methods, rather than treating them as a single class. In Fig. 6b, each cluster aligns closely with one of the five deepfake generation techniques present in the FaceForensics++ dataset, indicating that the model effectively learns method-specific features for multi-class classification. Both classifiers use the same feature space, but different label interpretations (binary vs. multi-class) reveal distinct decision boundaries. These results highlight that EfficientNet-b0 provides the most reliable performance across both tasks.



(**a**) Binary classification

(**b**) Multi-class classification

Figure 6: t-SNE visualizations of EfficientNet-B0 features.

To examine cross-dataset performance, the model was tested on 700 samples from FaceForensics++, 6228 from Celeb-DF v2, and 2,000 from DeeperForensics. As shown in Tables 6 and 7, EfficientNet-b0 consistently demonstrated high binary classification performance, with an overall accuracy of 95.14% and an AUC of 0.993. For cross-dataset testing trained on FaceForensics++ and tested on an unseen dataset, EfficientNet-b0 achieved better performance than the other models, with an AUC of 0.696, pAUC of 0.129, and EER of 0.358 using the Celeb-DF dataset. On the DeeperForensics, ResNeXt50-32x4d achieved the highest AUC of 0.841 and the lowest EER of 0.239. Additionally, Xception showed competitive performance

in specific metrics, achieving the lowest EER of 0.040 on FaceForensics++ and the highest pAUC of 0.474 on DeeperForensics. These results suggest that EfficientNet-b0 provides stable performance across datasets, while ResNeXt50 and Xception exhibit strengths in specific datasets and evaluation metrics.

Table 7: Detection performance (AUC, pAUC, and EER) of ResNeXt50, Xception, and EfficientNet-b0 on FaceForensics++, Celeb-DF, and DeeperForensics datasets.

Model	Training Dataset	Evaluation Datasets								
		FaceForensics++			Celeb-DF			DeeperForensics		
		AUC	pAUC	EER	AUC	pAUC	EER	AUC	pAUC	EER
ResNeXt50-32x4d	FaceForensics++	0.991	0.743	0.050	0.607	0.085	0.416	**0.841**	0.456	**0.239**
Xception	FaceForensics++	0.992	0.945	**0.040**	0.664	0.121	0.380	0.815	**0.474**	0.257
EfficientNet-b0	FaceForensics++	**0.993**	**0.950**	0.055	**0.696**	**0.129**	**0.358**	0.827	0.374	0.248

Note: Bold values indicate the best performance for each metric among the compared models. Higher values indicate better performance for AUC and pAUC, while lower values indicate better performance for EER.

5.2.2 XAI Results

This subsection presents the XAI-based analysis results obtained using Grad-CAM visualization and ROI Activation scoring. Grad-CAM was employed to identify the facial regions that most strongly influenced the deepfake prediction process, while ROI Activation analysis was used to quantitatively measure the activation intensity across specific facial regions to support structured interpretation and LLM-based explanation generation.

The Grad-CAM visualization using the feature maps of the final selected EfficientNet-b0 + LSTM detection model is shown in Fig. 7. Grad-CAM highlights the regions in each frame that most significantly influenced the model's prediction. Red regions indicate higher influence on the prediction of a fake label. The model predominantly identified deepfake artifacts in specific facial areas, such as the eyes (a), nose (b), and lips (c), which consistently exhibited manipulation characteristics.

(**a**) Eye (**b**) Nose (**c**) Mouth

Figure 7: Grad-CAM visualizations highlighting discriminative facial regions: (**a**) eye, (**b**) nose, and (**c**) mouth.

We selected the top-1 frame with the highest prediction confidence for each deepfake video derived from the same original source, as shown in Fig. 8. A visual comparison of the Grad-CAM results between EfficientNet-b0 and ResNeXt50-32x4d reveals distinct differences in their attention patterns. EfficientNet-b0 tends to focus on localized facial regions, such as the eyes, nose, and mouth, indicating that it determined the

fake by detecting detailed features. In contrast, ResNeXt50-32x4d exhibits broader activation across the face, suggesting that it focuses on more global regions when it makes decisions. Furthermore, when calculating the average of activation for each frame, EfficientNet-b0 generally produces lower activation than ResNeXt50-32x4d. This suggests that EfficientNet-b0 detects the fake using more compact and concentrated regions. Such focused attention not only enhances the visual interpretability of the result, but also facilitates the transformation of XAI results into structured quantitative data through ROI-based activation scoring, as described in the following section.

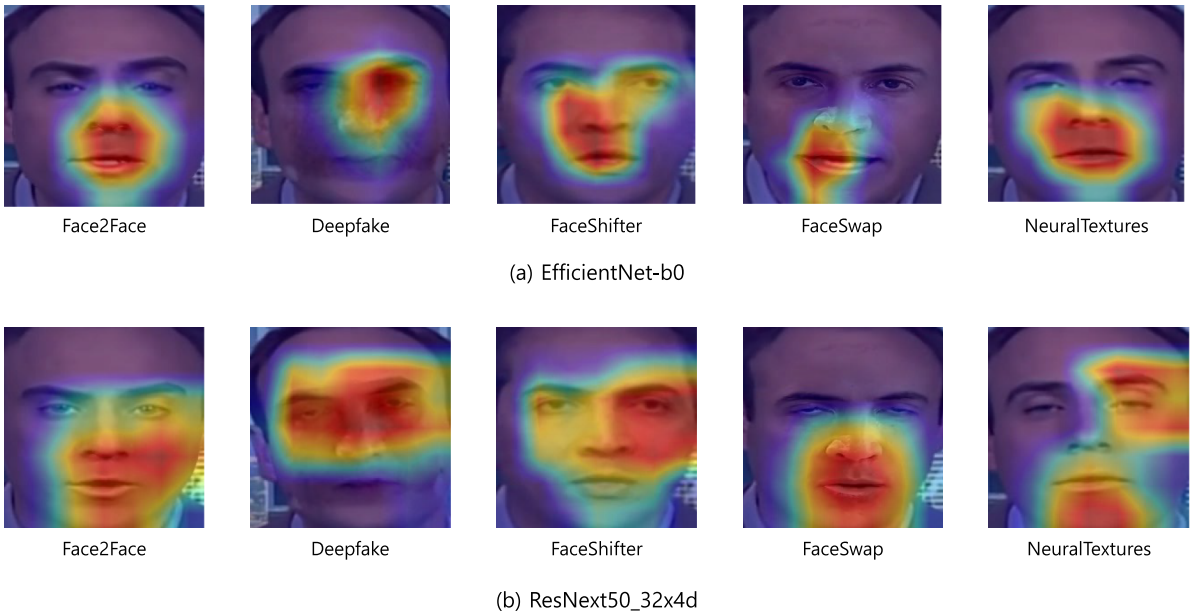
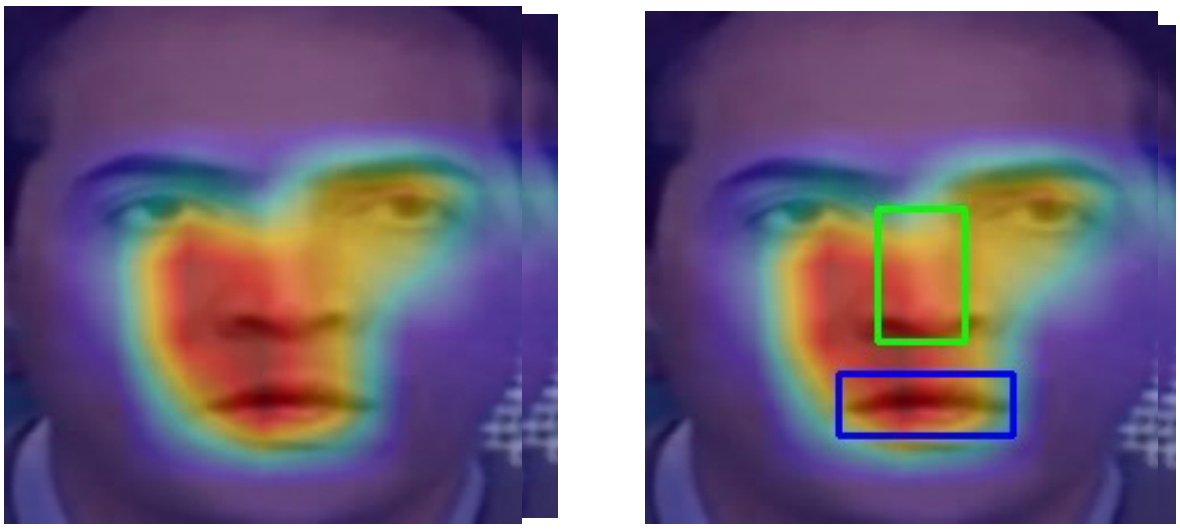


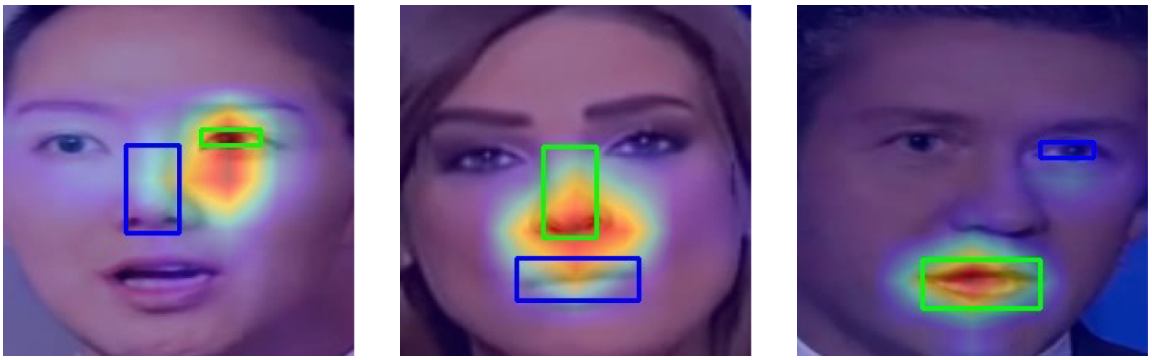
Figure 8: Grad-CAM visualizations of fake samples using (**a**) EfficientNet-b0 and (**b**) ResNeXt50-32x4d.

To transform Grad-CAM visualization results into structured regional evidence for LLM, ROI Activation analysis was performed using facial landmark-based region extraction. In detail, we use the Face Alignment library to detect specific facial regions. We detect the eyes, nose, mouth, jawline, and eyebrows as key facial ROIs, and calculate the average activation intensity of each region using the Grad-CAM heatmap. Fig. 9 illustrates how Grad-CAM results are aligned with facial landmarks. The right-hand image overlays detected ROIs such as the nose, mouth, and eyes, offering a region-wise interpretation of the model's focus. This structure facilitates the generation of interpretable reasoning using LLMs. Fig. 10 presents three example cases showing the Grad-CAM visualization, predicted label, confidence score, and ROI-based activation percentages. These scores quantify how much attention the model pays to each facial region when making decisions. For instance, in sample (a), the right eye accounts for 78.6% of activation, indicating potential visual artifacts the model picked up on. In sample (c), the mouth region dominates with 59.02% activation, implying irregularities in lip movement were key to detection.



(a) (b)

Figure 9: Comparison of Grad-CAM visualizations. (**a**) Original Grad-CAM heatmap. (**b**) Grad-CAM with ROI activation analysis highlighting facial regions using bounding boxes.



actual : fake
prediction : fake
confidence : 99.04%

roi activation score (%) :
jawline : 1.61%
left eye : 19.63%
right eye : 78.6%
left eyebrow : 8.30%
right eyebrow : 22.84%
nose : 27.16%
mouth : 6.4%

actual : fake
prediction : fake
confidence : 99.99%

roi activation score (%) :
jawline : 14.67%
left eye : 2.22%
right eye : 5.83%
left eyebrow : 0.07%
right eyebrow : 0.09%
nose : 55.38%
mouth : 34.13%

actual : fake
prediction : fake
confidence : 99.90%

roi activation score (%) :
jawline : 3.58%
left eye : 0.28%
right eye : 12.73%
left eyebrow : 1.26%
right eyebrow : 4.16%
nose : 9.42%
mouth : 59.02%

(a) (b) (c)

Figure 10: Examples of Grad-CAM visualizations and ROI activation scores for correctly detected deepfake videos. (**a**) Right eye region exhibits the highest activation. (**b**) Nose region exhibits the highest activation. (**c**) Mouth region exhibits the highest activation. Green and blue bounding boxes indicate the regions with the highest and second-highest activation scores, respectively.

In addition to these primary activations (highlighted in green), we also collected the secondary responses (highlighted in blue), which provide supplementary but meaningful cues. For example, in sample (a), the nose shows 27.16% activation, suggesting texture artifacts around the nose region. In sample (b), the mouth accounts for 34.13% activation, complementing the dominant nose region (55.38%) and reinforcing the detection of lip synchronization issues. Similarly, in sample (c), while the mouth is the most influential region, the right eye still contributes 12.73% activation, indicating that the model captured inconsistencies not only in lip movements but also in eye details. This ROI-wise activation representation acts as a structured, interpretable signal that can be leveraged by LLMs to generate natural language explanations for deepfake classification.

5.2.3 Large Language Models (LLM)

We use OLLAMA to generate natural language explanations based on structured prompts derived from statistical analysis of prediction probabilities and ROI activation scores. As mentioned in Section 4.3, we construct the prompt based on activation and prediction statistics for each frame, and input it into the LLM. The generated explanations are presented for representative REAL and FAKE cases.

For REAL cases shown in Fig. 11, the explanation focuses on the number of frames that did not detect activations and returned "None". Most facial regions exhibit very low detection counts, typically between 0 and 10, while the "None" category dominates with over one hundred counts. The generated explanation correctly reflects this pattern by emphasizing the lack of strong activations and concluding that the video is likely REAL.

REAL Explanation:
Here's the analysis of the Grad-CAM ROI activation statistics for a video that is predicted to be **REAL**:
Key Findings:
1. ****Low detection counts for facial regions**:** The detection counts for each facial region (jawline, left/right eye, left/right eyebrow, nose, mouth) are generally low, **with most values ranging from 0 to 10**. This suggests that the model is not strongly activating any specific features in these regions.
2. ****High 'None' detection count**:** The 'None' category has a very **high detection count of 118**, indicating that most frames in the video do not have any notable activations in the facial regions. This is consistent with the overall REAL prediction, as fake videos tend to have more pronounced and deliberate features in the face region.
3. **Lack of significant activations:** The low detection counts for facial regions and the high 'None' count suggest that there are no significant activations or features in these regions that could indicate manipulation.
Conclusion:
The analysis suggests that **the video is likely REAL due to the lack of notable activations or features in the facial regions.** The low detection counts and high 'None' count support this conclusion, indicating that the video does not exhibit any strong cues that would suggest manipulation.

Figure 11: REAL explanation generated by LLM.

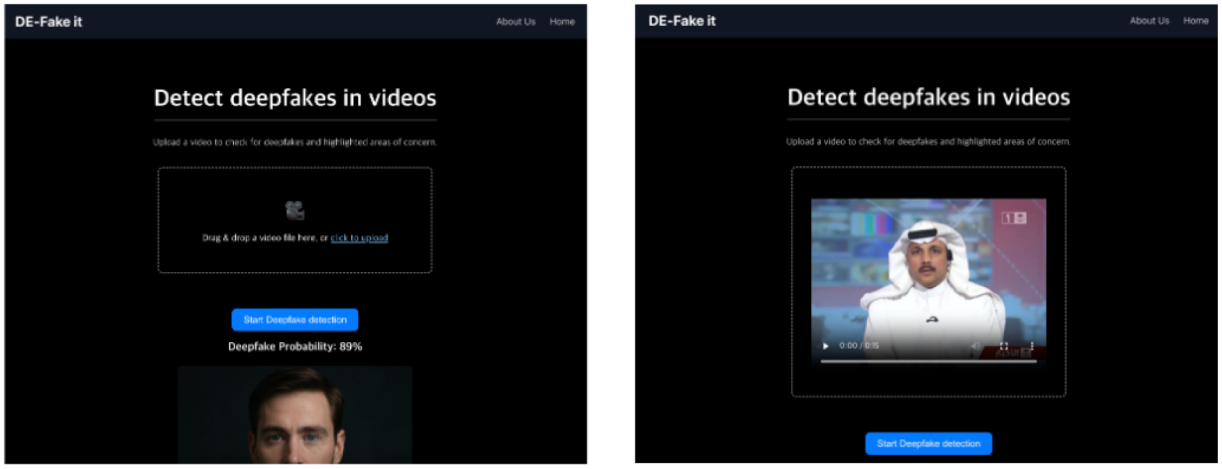
For FAKE cases shown in Fig. 12, the explanation highlights the facial regions with the highest activation counts. As illustrated in the figure, the mouth and jawline exhibit the highest detection counts, with detection counts of 70 and 34, respectively, and the generated explanation correspondingly emphasizes these regions as key contributors to the classification. In contrast, regions with minimal activation, including the nose with a count of 4 and the left eyebrow with a count of 8, have limited influence on the deepfake classification outcome. Furthermore, since our model also returns multi-class classification results, this information is also included in the generated explanation. The explanation refers to manipulation patterns consistent with Face2Face, which aligns with the strong activation observed in the mouth and jawline regions. These examples qualitatively verify that the ROI-based activation statistics and prediction results are properly reflected in the generated explanations. This correspondence supports the consistency between the structured inputs and the generated descriptions.

FAKE Explanation :
Here's the analysis of the facial regions that contributed to the model's prediction of **FAKE**:
Key Takeaways:
1. ****Mouth region is highly activated****: The mouth region was detected as the most active region **(70 times)** in the Grad-CAM analysis, which suggests that this region played a crucial role in the model's decision to classify the video as FAKE.
2. **Jawline and Mouth regions are highly correlated**: Both the jawline and mouth regions were detected multiple times (34 and 70 times, respectively), indicating a strong correlation between these two regions in the detection of fake information.
3. ****Left Eye Brow region is relatively inactive****: The left eye brow region was only detected **8 times**, which suggests that this region did not play a significant role in the model's prediction of FAKE.
Other Observations:
4. **Right Eye and Left Eye regions are moderately active**: These two regions were detected 13 and 10 times, respectively, indicating some level of activation but not as high as the mouth or jawline regions.
5. **Nose region is relatively inactive**: The nose region was only detected 4 times, which suggests that this region did not contribute significantly to the model's prediction of FAKE.
conclusion:
Based on the Grad-CAM analysis, it appears that the mouth and jawline regions were highly active in the detection of fake information, suggesting that these areas may have been manipulated or altered in some way. The left eye brow region was relatively inactive, which may indicate that this area was not targeted for manipulation. Overall, the results suggest that **the model is highly sensitive to changes in the mouth and jawline regions** when detecting FAKE videos.
Note: **Face2Face** is a deepfake technique that can be used to manipulate facial expressions, so **it's possible that this technique was used to alter the mouth and jawline regions of the original video.**

Figure 12: FAKE explanation generated by LLM.

5.2.4 System Deployment Interface

To demonstrate the deepfake detection system and provide an accessible visualization interface for end users, we developed the user interface "DE-Fake it". Fig. 13 illustrates the video upload process within the website. The left panel shows the interface before uploading a video, with a drag-and-drop upload area. The right panel displays the interface after a video is uploaded, including a preview player. In cases where videos cannot be directly processed on the client side, they are transmitted to the server, processed using FFmpeg, and then rendered for playback within the interface.



(**a**) Before the upload (**b**) After the upload

Figure 13: Website interface example: (**a**) before the upload and (**b**) after the upload.

Figs. 14 and 15 illustrate example outputs of the interface. For visual explanation, we provide videos with Grad-CAM and ROI activation overlays on the original frames. These overlays allow users to intuitively identify which facial regions the model focuses on during prediction. The most highly activated region is

marked with a green bounding box, while the second most activated region is indicated by a blue bounding box. The interface also presents frame-level predictions in a tabular format and provides natural language explanations generated by an LLM. These components are designed to help users interpret the model's predictions and better understand its behavior.

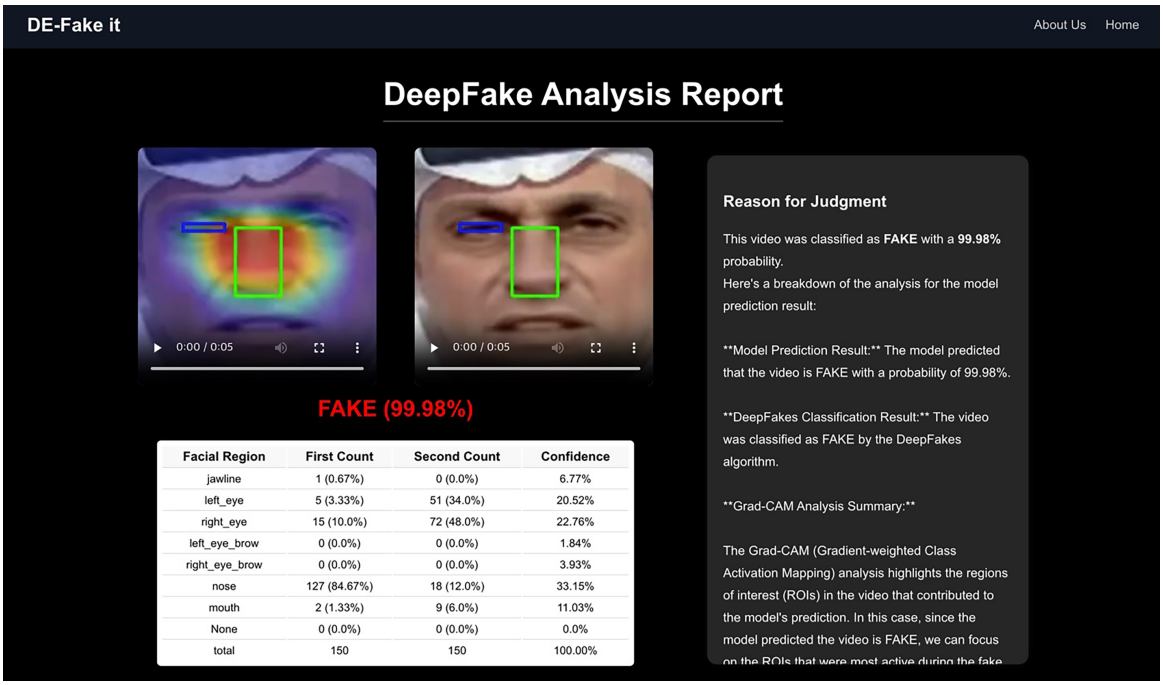


Figure 14: Website result page—FAKE.

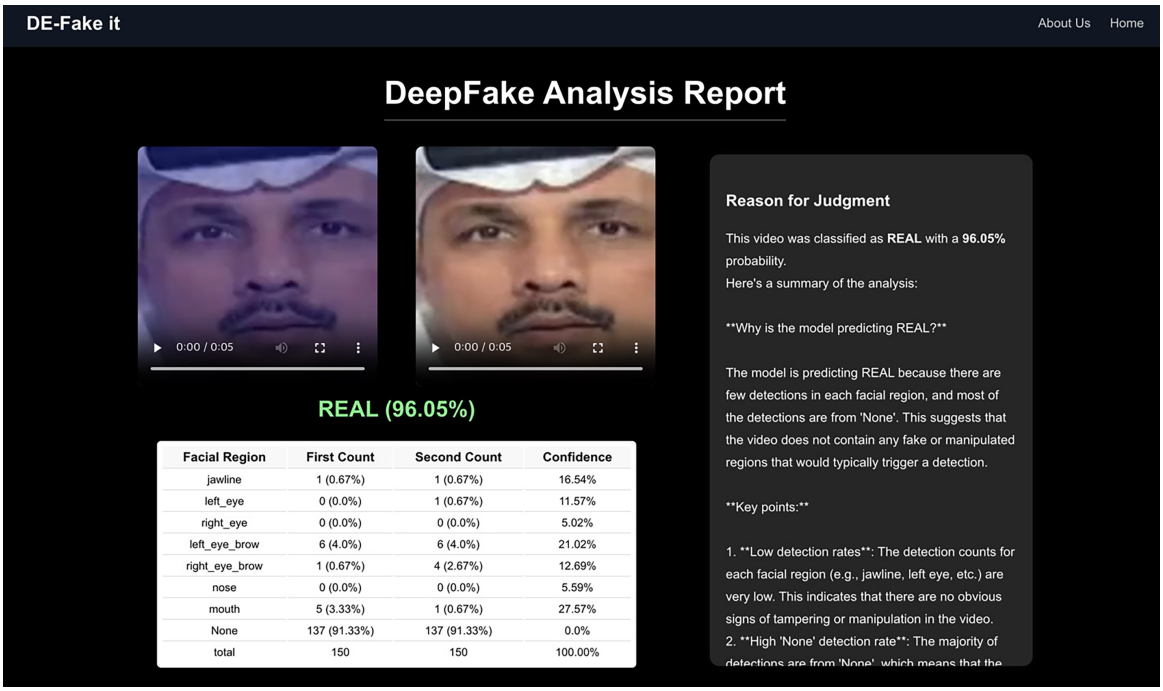


Figure 15: Website result page—REAL.

To quantitatively analyze the runtime characteristics of the proposed interface, we conducted runtime measurements using an 11-s video with a resolution of 640 × 480 at 30 FPS. The pipeline was executed sequentially in a local prototype environment on a MacBook with an Apple M2 Pro chip and 32 GB memory, using PyTorch with MPS acceleration, OpenCV, FFmpeg, and Ollama-based Llama3. Each experiment was repeated five times under the same environment, and the average runtime per video was calculated, as summarized in Table 8. The preprocessing and Grad-CAM visualization stages require frame-level image processing, resulting in average runtimes of 3.01 and 3.92 s, respectively. The Grad-CAM visualization time includes Grad-CAM heatmap generation, overlay rendering, bounding-box visualization, and ROI activation analysis. In contrast, the deepfake inference stage required only 0.26 s on average. The LLM-based explanation generation stage required 10.93 s on average and accounted for the largest portion of the overall runtime.

Table 8: Runtime analysis of the proposed framework.

Metric	Result
Average preprocessing time per video	3.01 s
Average inference time per video	0.26 s
Average Grad-CAM generation time per video	3.92 s
Average LLM explanation generation time per video	10.93 s
Average total processing time per video	18.12 s

These results suggest that the proposed framework can provide prediction outputs, Grad-CAM/ROI visual evidence, and LLM-based explanations through an integrated prototype interface in a user-accessible format. Although the LLM-based explanation generation stage required the longest runtime, this delay mainly originated from the external Ollama-based Llama3 model used for natural language explanation generation. Overall, the measured runtime suggests that the system can present prediction results, explainable visualizations and natural language explanations within a practical prototype-level processing time, although it is not intended to demonstrate real-time performance.

While the interface is intended to improve accessibility, formal usability testing has not yet been conducted and is planned for future work. In our study, the System Deployment Interface focuses on establishing a pipeline that enables users to upload videos and intuitively access the outputs. In future work, the system can be deployed in real-world environments to collect user feedback, which can be used to evaluate the usability and interpretability of the interface.

5.3 Ablation Study

To evaluate the contribution of each component in the proposed framework, we conduct an ablation study across two aspects. First, we assess detection performance by progressively incorporating architectural modules: CNN backbone, the temporal LSTM module, and the multitask learning head. Additionally, we provide a supplementary analysis of explanation quality across three variants: Binary Only, Binary + Method, and Binary + Method + ROI. We adopted a G-Eval-style pairwise evaluation in which LLM judges assessed the quality of explanations generated from different information configurations.

5.3.1 Detection Performance Ablation

Table 9 shows the detection performance of three model variants on FaceForensics++ under a five-epoch training setting for comparative ablation analysis. The CNN-only model uses frame-level spatial

features without temporal modeling, and achieves a binary accuracy of 0.54. This indicates that spatial features alone are insufficient for reliable deepfake detection. Adding the LSTM module improves the binary detection performance, increasing the binary accuracy from 0.54 to 0.87 and macro F1 score from 0.54 to 0.83, representing the best binary detection performance among all variants. The multitask variant jointly learns binary real/fake classification and manipulation-method classification. Although binary accuracy and macro F1 are slightly lower than CNN + LSTM, the multitask model achieves the highest AUC of 0.91 among all variants, indicating strong discrimination ability in ranking real and fake samples. In addition, the multitask head provides method-level information about the likely deepfake generation technique, which serves as a critical input to the subsequent explainability module. Therefore, the multitask objective was designed not only to optimize binary classification performance, but also to enhance explainability.

Table 9: Detection performance ablation.

Variant	Binary Acc.	AUC	Macro F1
CNN only	0.54	0.88	0.54
CNN + LSTM	**0.87**	0.90	**0.83**
CNN + LSTM + multitask	0.85	**0.91**	0.80

Note: Bold values indicate the best performance for each evaluation metric.

5.3.2 Explanation Quality Ablation

To evaluate the contribution of different contextual information to explanation quality, we define three explanation variants: Binary only, Binary + Method, and Binary + Method + ROI. The Binary only setting provides only the REAL/FAKE prediction result to the LLM explanation module. The Binary + Method setting additionally includes the predicted manipulation method, while the Binary + Method + ROI setting further incorporates ROI-based quantitative evidence derived from Grad-CAM activations. Thus, this ablation focuses on whether adding method prediction and ROI-based evidence improves the quality of textual explanations.

Table 10 presents the results of the pairwise preference evaluation across three explanation variants. Explanations were generated using Llama3, and preference judgments were conducted by two independent judges, Llama3 and GPT-4o. To reduce potential single-model bias in the evaluation process, preference judgments were independently performed by both LLM judges. The evaluation was conducted only on correctly classified samples, consisting of 60 videos in total, with 30 real and 30 fake samples. For each video and each pair of explanation settings, two explanations were presented to an LLM judge, which selected the explanation that better supported the model's REAL/FAKE decision. The evaluation was conducted using predefined explanation-quality criteria: decision justification, faithfulness to the provided evidence, specificity of evidence, completeness of reasoning, and interpretability for non-expert users.

As shown in Table 10, both judges preferred Binary + Method over Binary only in all cases. When comparing Binary + Method with Binary + Method + ROI, ROI-enhanced explanations were preferred by Llama3 and GPT-4o in 75.0% and 81.7% of the samples, respectively. This preference was more pronounced for fake samples, where ROI-enhanced explanations were preferred in 90.0% of cases by Llama3 and 100.0% by GPT-4o. For real samples, the preference was weaker, with ROI-enhanced explanations preferred in 60.0% and 63.3% of cases by Llama3 and GPT-4o, respectively. These results indicate that method prediction provides useful contextual information beyond the binary prediction alone. In addition, ROI statistics further improve the perceived explanation quality, especially for fake samples. This suggests that localized activation evidence is particularly helpful when explaining manipulated videos, where region-specific artifacts are more

relevant. In contrast, the smaller gain for real samples is reasonable because real videos are characterized by the absence of manipulation, which reduces the usefulness of region-specific ROI evidence. As a result, Binary + Method and Binary + Method + ROI show a smaller difference for real predictions. Overall, the proposed ROI-based quantitative evidence improves the informativeness of LLM-generated explanations, particularly for fake predictions.

Table 10: Pairwise preference evaluation of explanation settings using LLM-based judgment.

Comparison A	Comparison B	Preferred Setting	LLM Judge	Overall	Fake	Real
Binary only	Binary + Method	B preferred	Llama3	60/60 (100%)	30/30 (100%)	30/30 (100%)
		B preferred	GPT-4o	60/60 (100%)	30/30 (100%)	30/30 (100%)
Binary + Method	Binary + Method + ROI	B preferred	Llama3	45/60 (75%)	27/30 (90%)	18/30 (60%)
		B preferred	GPT-4o	49/60 (82%)	30/30 (100%)	19/30 (63%)
Binary only	Binary + Method + ROI	B preferred	Llama3	60/60 (100%)	30/30 (100%)	30/30 (100%)
		B preferred	GPT-4o	60/60 (100%)	30/30 (100%)	30/30 (100%)

Note: Evaluation was conducted using correctly classified samples only ($N = 60$, including 30 real and 30 fake videos).

6 Discussion

The proposed framework demonstrates several strengths compared with existing deepfake detection approaches. First, the multi-task learning architecture enables simultaneous binary classification and manipulation-type classification, improving both detection performance and interpretability. Second, the integration of Grad-CAM-based visual explanations with ROI-level activation analysis provides structured regional evidence rather than relying solely on qualitative heatmap visualization. Furthermore, the use of an LLM as a post-hoc explanation tool enables the model outputs to be translated into natural-language rationales, supporting interpretation of the detection results at the user level.

Compared with prior deepfake detection studies that primarily focus on classification accuracy, the proposed approach emphasizes explainability by linking structured model outputs with interpretable explanations. This design allows the framework to provide both prediction outcomes and supporting evidence in multiple complementary modalities, improving transparency of the decision-making process. Therefore, we clarify that the contribution of our study is not centered on designing a new CNN or LSTM architecture from scratch. Instead, the main advancement lies in the unified integration of multiple components into a single explainable deepfake analysis framework. Specifically, the proposed system combines spatio-temporal feature learning through CNN-LSTM modeling, ROI-level Grad-CAM-based visual interpretation, and LLM-driven textual explanation generation. This pipeline enables not only accurate detection performance, but also improved interpretability and human-understandable inspection capabilities, which distinguish the proposed approach from conventional detection-only frameworks.

In detail, our study is the integration of XAI and LLM to enhance interpretability. The system uses Grad-CAM to generate visual explanations, highlighting the facial regions most influential to the prediction. Beyond simple heatmap visualization, we introduced a ROI activation analysis, which quantifies attention scores for predefined facial regions (e.g., eyes, mouth, and nose). These statistical scores are then structured into prompts for the LLM, allowing it to generate more accurate, data-driven natural language explanations tailored to each prediction. This ensures that the explanations remain grounded in model evidence and improve interpretability.

A key characteristic of our explanation module is that it relies on structured text prompts derived from Grad-CAM summaries—such as region names and activation percentages—rather than direct fine-tuning

of the LLM on visual heatmaps. While this approach ensures computational efficiency, it inherently limits the explanations to region-level statistics. Consequently, the system cannot currently provide fine-grained, pixel-level semantic descriptors, a challenge we aim to address in future work by expanding the model's descriptive granularity.

From a practical perspective, the proposed framework can support human decision-making in applications such as media verification, digital forensics support, and identity authentication systems, where interpretable detection results are essential. By presenting predictions together with their supporting evidence, the framework helps users inspect model decisions more easily, thereby increasing confidence in the results. In addition, the integration of region-level analysis and natural-language explanations reduces ambiguity in interpreting model outputs, leading to more consistent and well-grounded interpretations across users. Furthermore, natural-language explanations improve accessibility by helping non-expert users better understand model outputs, especially when visual heatmaps are difficult to interpret. Finally, the combination of visual, numerical, and textual explanations provides complementary perspectives, offering a more comprehensive view of the model's behavior.

However, several limitations should be acknowledged. First, although external testing was included, model development relied on a single primary training dataset, which may limit the diversity of manipulation artifacts learned during training. Second, our study focuses on qualitative verification of alignment between LLM-generated explanations and model outputs through an LLM-based pairwise comparison of explanation quality, but does not include human-subject validation. Third, although a web-based interface was developed, it has not yet been deployed in real-world environments, and no formal user study has been conducted to evaluate the usability of the system and the usefulness of its outputs. Finally, the ROI-based analysis is designed to provide structured region-level interpretability using predefined facial regions, which may limit its ability to capture more fine-grained manipulation artifacts. Future work will explore more fine-grained visual-semantic explanations to enhance interpretability and better align model outputs with human-understandable reasoning.

7 Conclusion and Future Work

This paper introduced a novel, explainable, and user-facing deepfake detection framework that integrates deep learning with interpretability techniques to improve both performance and transparency. The proposed system utilizes a multi-task learning architecture based on CNN and LSTM models to simultaneously perform binary classification (real vs. fake) and multi-class classification (specific deepfake methods). The model was developed using FaceForensics++ and externally tested on Celeb-DF and DeeperForensics to examine performance under selected cross-dataset settings.

To improve interpretability and user-facing functionality, the entire framework is deployed as a web-based service. The interface enables users to upload a video and receive results that include the classification outcome, visual explanations (Grad-CAM overlays), and detailed LLM-generated textual rationales. This multi-modal feedback design aims to support both detection and interpretation by presenting visual and textual rationales that help users inspect the model's decisions. Among the models tested, EfficientNet-b0 combined with LSTM achieved the highest performance, with a binary classification accuracy of 95.14, a multi-class accuracy of 94.29, and an AUC of 0.993 on the FaceForensics++ test set.

The framework may serve as a prototype-level decision-support tool for domains such as media verification, forensic analysis support, and identity verification. In social media moderation, the proposed system could be explored as an API-based tool for identifying potentially manipulated content and providing visual rationales. For journalism and fact-checking, it may support the inspection of digital media where reporters may inspect viral evidence and generate structured LLM reports before public release. Lastly, the

proposed approach could be considered for integration into identity verification systems, such as banking e-KYC, where visual and textual explanations may help support the analysis of facial inconsistencies during the authentication process.

As future work, subsequent research could explore domain adaptation techniques or contrastive pre-training strategies in order to improve generalization to other video sources. Additionally, while Grad-CAM and ROI-based activation scoring are informative visual cues, the current mechanism of interpretability can be optimized. Grad-CAM focuses on large regions of activation but does not compute uncertainty or causal contribution explicitly. Therefore, incorporating tools like SHAP or Layer-wise Relevance Propagation (LRP) could improve interpretability, especially for marginal cases. Lastly, though the prompt-tuned LLM rationales improved user interpretability, they are presently derived from static ROI activation scores. Future work can leverage temporal patterns or affect-based cues across frames to enrich LLM inputs in order that more contextual and stable textual rationales can be facilitated.

Acknowledgement: None.

Funding Statement: Our study was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Ministry of Science and ICT (MSIT) under the Information Security Core Technology Development Program (Project No. RS-2026-25519773, 30%; RS-2024-00438551, 10%), the Realistic Content Core Technology Development Program (Project No. RS-2023-00228996, 20%), and the Information Technology Research Center (ITRC) Program (Project No. IITP-2026-RS-2021-II211816, 10%), and by the National Research Foundation of Korea (NRF) grant funded by the MSIT under the Basic Research Program (Project No. RS-2026-25481431, 30%).

Author Contributions: The authors confirm their contribution to the paper as follows: Study conception and system design: Jiyeong Park; Data collection, methodology, coding, and results: Jiyeong Park, Sercan Yeşilköy, Doyeon Lim, Huiryeong Park, and Eunseo Lee; Supervision, manuscript preparation, and revisions: Mohsen Ali Alawami; Review and funding: Ki-Woong Park. All authors reviewed and approved the final version of the manuscript. During the preparation of this manuscript, AI-assisted language editing tools were used only to improve grammar, clarity, and readability. The authors reviewed and edited all content and take full responsibility for the final manuscript.

Availability of Data and Materials: The datasets used in our study are publicly available through their official repositories: 1. FaceForensics++ (c23): https://github.com/ondyari/FaceForensics; 2. DeeperForensics-1.0: https://github.com/EndlessSora/DeeperForensics-1.0; 3. Celeb-DF: https://github.com/yuezunli/celeb-deepfakeforensics. These repositories provide dataset descriptions, access instructions, and download procedures. The FaceForensics++ c23 compressed version used for training and evaluation in our study was obtained according to the official repository guidelines, while DeeperForensics-1.0 and Celeb-DF were used as external datasets for evaluating the generalization capability of the proposed framework. All experiments reported in our study were conducted using publicly available datasets to support transparency, accessibility, and reproducibility.

Ethics Approval: Not applicable. This study did not involve human participants, human subjects, personal data collection, or animal experiments. All datasets used in this work are publicly available research datasets and were used in accordance with their respective licenses and usage policies.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Tolosana R, Vera-Rodriguez R, Fierrez J, Morales A, Ortega-Garcia J. DeepFakes and beyond: a survey of face manipulation and fake detection. Inf Fusion. 2020;64:131–48. doi:10.1016/j.inffus.2020.06.014.

2. Mirsky Y, Lee W. The creation and detection of deepfakes: a survey. ACM Comput Surv (CSUR). 2021;54(1):1–41. doi:10.1145/3425780.
3. Feng D, Lu X, Lin X, Yang H, Pasupa K, Leung ACS, et al. Deep detection for face manipulation. In: Yang H, Pasupa K, Leung ACS, Kwok JT, Chan JH, King I, editors. Neural information processing. Cham, Switzerland: Springer International Publishing; 2020. p. 316–23.
4. Li L, Bao J, Zhang T, Yang H, Chen D, Wen F, et al. Face X-ray for more general face forgery detection. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA. p. 5000–9.
5. Zhao H, Wei T, Zhou W, Zhang W, Chen D, Yu N. Multi-attentional deepfake detection. In: Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun 20–25; Nashville, TN, USA. p. 2185–94.
6. Wang C, Deng W. Representative forgery mining for fake face detection. In: Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun 20–25; Nashville, TN, USA. p. 14918–27.
7. Rössler A, Cozzolino D, Verdoliva L, Riess C, Thies J, Niessner M. FaceForensics++: learning to detect manipulated facial images. In: Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV); 2019 Oct 27–Nov 2; Seoul, Republic of Korea. p. 1–11.
8. Jiang L, Li R, Wu W, Qian C, Loy CC. DeeperForensics-1.0: a large-scale dataset for real-world face forgery detection. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA. p. 2886–95.
9. Li Y, Yang X, Sun P, Qi H, Lyu S. Celeb-DF: a large-scale challenging dataset for deepfake forensics. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA. p. 3204–13.
10. Li Y, Chang MC, Lyu S. In Ictu Oculi: exposing AI created fake videos by detecting eye blinking. In: Proceedings of the 2018 IEEE International Workshop on Information Forensics and Security (WIFS); 2018 Dec 11–13; Hong Kong, China. p. 1–7.
11. Jung T, Kim S, Kim K. DeepVision: deepfakes detection using human eye blinking pattern. IEEE Access. 2020;8:83144–54. doi:10.1109/access.2020.2988660.
12. Amerini I, Galteri L, Caldelli R, Del Bimbo A. Deepfake video detection through optical flow based CNN. In: Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW); 2019 Oct 27–28; Seoul, Republic of Korea. p. 1205–7.
13. Zhao T, Xu X, Xu M, Ding H, Xiong Y, Xia W. Learning self-consistency for deepfake detection. In: Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV); 2021 Oct 10–17; Montreal, QC, Canada. p. 15003–13.
14. Haliassos A, Vougioukas K, Petridis S, Pantic M. Lips don't lie: a generalisable and robust approach to face forgery detection. In: Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun 20–25; Nashville, TN, USA. p. 5037–47.
15. Guarnera L, Giudice O, Battiato S. Deepfake detection by analyzing convolutional traces. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW); 2020 Jun 14–19; Seattle, WA, USA. p. 2841–50.
16. Liu D, Dang Z, Peng C, Zheng Y, Li S, Wang N, et al. FedForgery: generalized face forgery detection with residual federated learning. IEEE Trans Inf Forensics Secur. 2023;18:4272–84. doi:10.1109/tifs.2023.3293951.
17. Wang SY, Wang O, Zhang R, Owens A, Efros AA. CNN-generated images are surprisingly easy to spot. . .for now. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA. p. 8692–701.
18. Nadimpalli AV, Rattani A. On improving cross-dataset generalization of deepfake detectors. In: Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW); 2022 Jun 19–20; New Orleans, LA, USA. p. 91–9.

19. Jeong Y, Kim D, Ro Y, Choi J. Frepgan: robust deepfake detection using frequency-level perturbations. In: Proceedings of the AAAI Conference on Artificial Intelligence; 2022 Feb 28–Mar 1; Vancouver, BC, Canada. p. 1060–8.
20. Lai CY, ting Hsu C, Hsu CC, Lin CW. Prompt-guided multi-modal contrastive learning for cross-compression-rate deepfake detection. In: Proceedings of the 35th British Machine Vision Conference 2024, BMVC 2024; 2024 Nov 25–28; Glasgow, UK.
21. Khormali A, Yuan JS. DFDT: an end-to-end deepfake detection framework using vision transformer. Appl Sci. 2022;12(6):2953. doi:10.3390/app12062953.
22. Wang J, Wu Z, Ouyang W, Han X, Chen J, Jiang YG, et al. M2TR: multi-modal multi-scale transformers for deepfake detection. In: Proceedings of the 2022 International Conference on Multimedia Retrieval. ICMR'22; 2022 Jun 27–30; Newark, NJ, USA. New York, NY, USA: Association for Computing Machinery; 2022. p. 615–23.
23. Güera D, Delp EJ. Deepfake video detection using recurrent neural networks. In: Proceedings of the 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS); 2018 Nov 27–30; Auckland, New Zealand. p. 1–6.
24. Cozzolino D, Rössler A, Thies J, Nießner M, Verdoliva L. ID-reveal: identity-aware deepfake video detection. In: Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV); 2021 Oct 10–17; Montreal, QC, Canada. p. 15088–97.
25. Yang Z, Liang J, Xu Y, Zhang XY, He R. Masked relation learning for deepfake detection. IEEE Trans Inf Forensics Secur. 2023;18:1696–708. doi:10.1109/tifs.2023.3249566.
26. Malolan B, Parekh A, Kazi F. Explainable deep-fake detection using visual interpretability methods. In: Proceedings of the 2020 3rd International Conference on Information and Computer Technologies (ICICT); 2020 Mar 9–12; San Jose, CA, USA. p. 289–93.
27. Mahmud F, Abdullah Y, Islam M, Aziz T. Unmasking deepfake faces from videos using an explainable cost-sensitive deep learning approach. In: Proceedings of the 2023 26th International Conference on Computer and Information Technology (ICCIT); 2023 Dec 13–15; Cox's Bazar, Bangladesh. p. 1–6.
28. Nguyen HH, Yamagishi J, Echizen I. Capsule-forensics: using capsule networks to detect forged images and videos. In: Proceedings of the ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2019 May 12–17; Brighton, UK. p. 2307–11.
29. Dang H, Liu F, Stehouwer J, Liu X, Jain AK. On the detection of digital face manipulation. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA. p. 5780–9.
30. Patel K, Sutariya M, Parmar P. A comprehensive survey on explainable deepfake detection: techniques, challenges, and future directions. In: Proceedings of the 2025 International Conference on Artificial Intelligence and Machine Vision (AIMV); 2025 Aug 16–17; Gandhinagar, India. p. 1–6.
31. Aleem M, Umair M, Zubair M, Ibrahim R, Naseem MT, Raza MM, et al. Seeing through the fake: explainable AI with multiple CNNs for deepfake detection. IEEE Access. 2026;14:131–62. doi:10.1109/access.2025.3649128.
32. Bharati N, Wong P, Mostéfaoui SK, Kbaier D, Collie J. Explainable deepfake detection: a multi-model framework with human-interpretable rationales for legal investigation purposes. Mach Learn Appl. 2026;23:100819. doi:10.1016/j.mlwa.2025.100819.
33. Petmezas G, Vanian V, Konstantoudakis K, Almaloglou EE, Zarpalas D. Video deepfake detection using a hybrid CNN-LSTM-Transformer model for identity verification. Multimed Tools Appl. 2025;84(33):40617–36. doi:10.1007/s11042-024-20548-6.
34. Al-Imran M, Sheikh MS, Kirtonia U, Arthi NT, Ripon S. DeFaX: a cross-attention fusion framework for robust and explainable deepfake detection. IEEE Access. 2025;13:213962–79. doi:10.1109/access.2025.3645769.
35. Yu P, Fei J, Gao H, Feng X, Xia Z, Chang CH. Unlocking the capabilities of large vision-language models for generalizable and explainable deepfake detection. In: Proceedings of the 42nd International Conference on Machine Learning; 2025 Jul 13–19; Vancouver, QC, Canada, San Diego, CA, USA: PMLR, Vol. 267, p. 72925–43.
36. Thakre A, Nagwekar O, Talekar V, Santra Biswas A. CAST: cross-attentive spatio-temporal feature fusion for deepfake detection. Knowl-Based Syst. 2026;338:115560. doi:10.1016/j.knosys.2026.115560.

37. Zhang K, Zhang Z, Li Z, Qiao Y. Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Process Lett. 2016;23(10):1499–503. doi:10.1109/lsp.2016.2603342.
38. Tipper S, Atlam HF, Lallie HS. An investigation into the utilisation of CNN with LSTM for video deepfake detection. Appl Sci. 2024;14(21):9754. doi:10.3390/app14219754.
39. Al-Dulaimi OAHH, Kurnaz S. A hybrid CNN-LSTM approach for precision deepfake image detection based on transfer learning. Electronics. 2024;13(9):1662. doi:10.3390/electronics13091662.
40. Xie S, Girshick R, Dollár P, Tu Z, He K. Aggregated residual transformations for deep neural networks. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017 Jul 21–26; Honolulu, HI, USA. p. 5987–95.
41. Xception CF. Deep learning with depthwise separable convolutions. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017 Jul 21–26; Honolulu, HI, USA. p. 1800–7.
42. Alkurdi DA, Cevik M, Akgundogdu A. Advancing deepfake detection using xception architecture: a robust approach for safeguarding against fabricated news on social media. Comput Mater Contin. 2024;81(3):4285–305. doi:10.32604/cmc.2024.057029.
43. Tan M, Le Q. EfficientNet: rethinking model scaling for convolutional neural networks. In: Proceedings of the 36th International Conference on Machine Learning; 2019 Jun 9–15; Long Beach, CA, USA. San Diego, CA, USA: PMLR; 2019. p. 6105–14.