Detecting Deepfake Audio/Video in Real-Time Video Calls Using AI Models

Mr. Aditya Navnath Phalke[1], Mr. Dnyaneshwar Balu Lokhande[2]
[1]*Department of Master of Computer Application, JCEI's Jaihind Institute of Management and Research, Vadgaon Sahani*
[2]*Assistant Professor, Department of Master of Computer Application, JCEI's Jaihind Institute of Management and Research, Vadgaon Sahani*

***Abstract*—Generative artificial intelligence has enabled realistic synthetic audio and video, creating risks of impersonation in live video calls. Detecting manipulated media during a call differs from offline analysis because the detector must operate under latency, bandwidth, compression, and incomplete-evidence constraints. This paper reviews visual, acoustic, and audio-visual deepfake detection research and proposes an evaluation-ready framework for streaming assessment. The design combines a sliding-window media buffer, separate audio and visual analysis branches, an audio-visual consistency branch, quality-aware score fusion, temporal aggregation, and an uncertainty-aware warning mechanism. FaceForensics++, FakeAVCeleb, ASVspoof 2021, and DeepfakeBench are identified as complementary resources for future evaluation. The proposed study defines tests for compression, frame loss, timestamp offsets, cross-dataset generalization, false alarms, latency, throughput, and resource use. No original experimental results are reported: the architecture and evaluation protocol remain proposals that require implementation and empirical validation. The intended contribution is a reproducible framework for assessing the trade-off between detection quality and responsiveness in live communication.**

***Index Terms*—adversarial robustness, audio deepfake detection, multimodal fusion, real-time inference, streaming video analysis.**

I. INTRODUCTION

Live video communication is widely used in remote work, education, telemedicine, recruitment, and financial verification. At the same time, face-swapping systems and neural voice cloning make it increasingly practical to impersonate a person through synthetic audio, video, or both. Existing benchmarks, including FaceForensics++, have helped standardize evaluation of manipulated facial imagery, but benchmark accuracy alone does not establish that a detector can respond reliably during a live call [1]. Surveys of facial manipulation research describe a broad range of manipulation methods and persistent challenges in generalizing detectors across datasets and manipulation types [2].

Real-time detection must balance evidence quality against time-to-warning. A detector may need to analyze only a subset of frames, process short audio windows, cope with variable bitrate and packet loss, and avoid an ever-growing inference queue. A nominal target such as sub-100-ms processing for an individual update should be treated as a deployment objective to measure, not as an established capability of the proposed system. The appropriate latency budget depends on the window length, update interval, hardware, and conferencing environment.

This paper is a literature-informed system proposal. It reviews complementary resources, states an evaluation protocol, and defines a multimodal architecture that can be implemented and tested. The paper does not claim a measured accuracy improvement or a verified conferencing-platform integration.

II. PROBLEM STATEMENT

The core problem is to identify signs of audio or video manipulation from continuously arriving media while maintaining an acceptable delay and avoiding excessive false alarms. Offline detectors often assume that a complete clip is available for processing. In a live call, the input may instead contain compressed

frames, missing or duplicated frames, temporarily absent faces, low signal-to-noise ratio, silence, and non-zero audio-video timestamp offsets. Each condition can reduce the reliability of individual modality scores. A system must therefore distinguish suspicious evidence from cases in which the available media are not suitable for assessment.

A second problem is generalization. A model may learn artifacts associated with generators or compression conditions in its training set and fail on unseen generation methods or communication pipelines. A third problem is operational: even a detector with strong classification performance may be unsuitable for interactive use if its processing latency, queue length, or false-alert rate is too high. The study therefore treats classification performance, streaming behavior, uncertainty handling, and resource consumption as separate evaluation dimensions.

III. OBJECTIVES

- To review the operating challenges of real-time audio and video deepfake detection in live calls.
- To compare relevant visual, speech anti-spoofing, and multimodal datasets and evaluation resources.
- To specify a streaming architecture that combines modality-specific evidence with audio-video consistency and media-quality signals.
- To define a reproducible evaluation protocol covering detection quality, latency, false alarms, generalization, and computational cost.
- To identify research limitations involving unseen generators, degraded media, privacy, and the absence of experimental validation.

IV. LITERATURE REVIEW

Research on deepfake detection spans frame-level visual artifacts, temporal patterns, frequency-domain evidence, and speech anti-spoofing. Rössler et al. introduced FaceForensics++, a benchmark covering several facial manipulation methods and compression settings [1]. Tolosana et al. reviewed facial manipulation categories, available datasets, and detection challenges, emphasizing that the field evolves alongside generation techniques [2]. These resources inform visual-branch development but do not, by themselves, define a live-call evaluation protocol.

For multimodal research, FakeAVCeleb includes real and manipulated audio-video combinations, including synthesized speech designed to align with manipulated video [3]. ASVspoof 2021 provides evaluation resources for spoofed and deepfake speech under varied conditions [4]. DeepfakeBench addresses inconsistencies between visual detector comparisons by providing a common benchmark framework and evaluation protocols [5]. Frequency analysis has also been investigated to expose artifacts in images generated by generative adversarial networks (GANs), although image-domain findings should not be assumed to transfer unchanged to audio or compressed live video [6].

Table I summarizes the relevance and limitations of selected resources. Together, the literature supports separate modality analysis and controlled benchmarking; empirical evidence is still required to determine how those elements should be combined for streaming use.

TABLE I. SUMMARY OF REVIEWED RESEARCH

Study / resource	Focus or contribution	Relevance and limitation
Rössler et al. [1] (2019)	FaceForensics++ benchmark for facial manipulations and compression.	Supports visual-branch benchmarking; it does not test a joint live-call pipeline.
Tolosana et al. [2] (2020)	Survey of face manipulation methods, datasets, and detection approaches.	Provides field context; survey findings do not establish streaming latency.
Khalid et al. [3] (2021)	FakeAVCeleb pairs audio-video manipulations and includes lip-synchronized synthetic speech.	Supports multimodal tests; recorded clips do not establish real-time performance.

Study / resource	Focus or contribution	Relevance and limitation
Liu et al. [4] (2022)	ASVspoof 2021 evaluates spoofed and deepfake speech across multiple conditions.	Informs audio evaluation; it does not cover face manipulation or audio-video fusion.
Yan et al. [5] (2023)	DeepfakeBench standardizes visual-detector comparisons over multiple datasets.	Supports reproducibility; it is not an end-to-end live conferencing assessment.
Frank et al. [6] (2020)	Studies frequency-domain artifacts in GAN-generated images.	Motivates frequency analysis; transfer to video and audio requires separate testing.

V. RESEARCH GAP

The reviewed resources address important pieces of the detection problem, but the paper identifies a gap between benchmark evaluation and end-to-end streaming assessment. In particular, the available evidence does not establish that one configuration can simultaneously generalize across generators, tolerate communication degradation, meet a specified latency budget, and maintain an acceptable false-alarm burden.

The proposed evaluation focuses on three gaps:

- Streaming latency versus detection quality: measure how window duration, update frequency, frame sampling, and dropped media affect both detection metrics and time-to-warning.
- Cross-generator and cross-dataset generalization: hold out sources or manipulation methods where feasible, rather than relying only on random splits from a single dataset.
- Uncertainty and adversarial robustness: test whether degraded or unavailable evidence is reported as indeterminate, and assess susceptibility to perturbations and distribution shifts.

VI. PROPOSED METHODOLOGY

The methodology has two stages: literature-based system design and a planned experimental validation stage. The architecture below is a proposal, not an implemented system. Dataset access, authorized integration, model selection, thresholds, and all performance claims must be verified during future experiments.

A. Overall architecture

The proposed detector receives decoded audio and video through an authorized conferencing integration or a controlled replay environment. It aligns timestamps, estimates media quality, and places incoming samples into a sliding-window buffer. Three branches estimate audio-manipulation evidence, visual-manipulation evidence, and audio-video inconsistency. A quality-aware fusion stage aggregates usable evidence, after which temporal smoothing and persistence rules control risk states and user warnings. If the required evidence is unavailable, the detector should return an indeterminate state instead of interpreting absence of evidence as proof of authenticity.

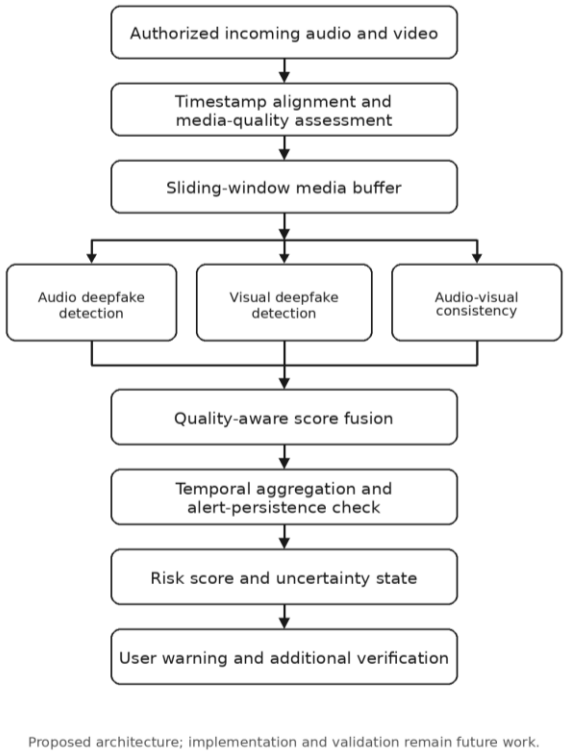


Fig. 1. Proposed pipeline for real-time audio-video deepfake detection.

B. Dataset selection and preparation

Three dataset resources are proposed for complementary evaluation: FaceForensics++ for facial manipulation detection [1], FakeAVCeleb for joint audio-video manipulation conditions [3], and ASVspoof 2021 for spoofed and deepfake speech [4]. DeepfakeBench can support reproducible visual-model comparisons [5]. Use of each dataset must follow its access conditions, license, and documented protocols. These resources should not be treated as interchangeable, and results from different tasks should be reported separately where labels and evaluation settings differ.

The multimodal test set should distinguish four conditions: (1) genuine audio and genuine video; (2) manipulated audio with genuine video; (3) genuine audio with manipulated video; and (4) manipulated audio and manipulated video. Training, validation, and test partitions should be separated by source recording and, where feasible, by participant identity. Overlapping windows from the same source recording must not be distributed across partitions, because doing so can lead to overly optimistic estimates of generalization.

C. Streaming simulation

Recorded clips should be supplied incrementally rather than processed as complete files. The planned experiments will vary video resolution and compression, audio encoding and background noise, frame drops and temporary freezing, audio-video timestamp offsets, observation-window duration, and processing hardware. An initial configuration may use two-second observation windows updated every half-second. These values are starting experimental settings only; they are not validated requirements. Several window lengths and update intervals should be compared to expose the trade-off between early warning and sufficient evidence.

D. Video analysis branch

1. Detect and track the visible participant face when a suitable face is present.
2. Extract face regions from sampled frames and apply a selected visual manipulation detector.
3. Aggregate frame-level scores over the current observation window.
4. Record face visibility, resolution, compression, and other image-quality indicators.

Candidate detectors should be selected through reproducible benchmark comparison, using resources such as FaceForensics++ and DeepfakeBench [1], [5]. If a face is absent or unsuitable for analysis, the branch should report that evidence is unavailable instead of assigning a low manipulation score. A low score and a missing score are different states and must remain distinguishable in evaluation.

E. Audio analysis branch

1. Identify speech-active portions of the observation window.
2. Normalize audio to the input format required by the selected model.
3. Extract spectral features or learned speech representations.
4. Estimate the likelihood of synthetic or converted speech and record quality indicators.

Audio experiments should follow the protocols associated with the selected ASVspoof 2021 resources [4]. Compression, background noise, silence, and unseen sources should be included in validation. Silence or insufficient speech should yield an unavailable-evidence state; it should not automatically be interpreted as genuine speech.

F. Audio-video consistency branch

This supplementary branch examines temporal correspondence between speech and visible mouth movement. It first estimates or compensates for audio-video timestamp offsets, then evaluates whether usable audio and video provide coherent temporal evidence. FakeAVCeleb can support experiments involving lip-synchronized manipulated media [3]. However, consistent lip synchronization cannot establish authenticity, since manipulated audio and video may be generated to agree. The branch should therefore contribute one evidence score rather than act as a standalone authenticity test.

G. Quality-aware score fusion

Let $s_m(t)$ denote the manipulation score from modality m in window t, $q_m(t)$ its evidence-quality or availability weight, and w_m a validation-selected modality weight. The audio-video consistency branch is represented by an inconsistency score, so that larger scores consistently indicate greater manipulation evidence. A baseline fusion rule is:

$$S_fuse(t) = \Sigma_m w_m q_m(t) s_m(t) / \Sigma_m w_m q_m(t)$$

The sum is taken over available modalities. Weights should be selected using validation data only. If the denominator is zero or usable evidence falls below a pre-defined minimum, the system should report insufficient evidence rather than a fused score. The weighted output should not be described as a probability unless a separate calibration procedure supports that interpretation. A learned fusion model may be compared with this baseline in a controlled ablation study.

H. Temporal aggregation and alerting

Window-level scores may fluctuate because of noise, compression, and intermittent evidence. One candidate smoother for evaluation is:

$\bar{S}(t) = \alpha\, S_fuse(t) + (1 - \alpha)\, \bar{S}\,(t - 1),\ 0 < \alpha \leq 1$

The smoothing coefficient, decision threshold, and persistence requirement must be selected on validation data and reported with the results. A warning may require a threshold crossing that persists across multiple windows. The user-facing output should distinguish three states: low observed manipulation risk, potential manipulation detected, and insufficient evidence. “Low observed risk” does not certify identity; a potential-manipulation alert should prompt additional verification instead of an automatic accusation.

I. Evaluation metrics

Classification quality and streaming behavior should be reported separately. Metrics should be calculated on held-out test data, with macro or class-wise reporting where class imbalance makes a single aggregate value misleading. Table II lists the proposed measures.

TABLE II. PROPOSED EVALUATION MEASURES

Dimension	Metric	Purpose
Classification	Precision, recall, F1-score	Measure detection quality and class imbalance effects.
Ranking	Receiver operating characteristic area under the curve (ROC-AUC) and precision-recall area under the curve (PR-AUC)	Assess score discrimination across thresholds.
Speech anti-spoofing	Equal error rate (EER); other task-specific measures where appropriate	Support comparison with audio anti-spoofing protocols [4].
False alarms	False-positive rate and false alerts per call-hour	Quantify disruption during genuine conversations.
Latency	Median and 95th-percentile processing latency; time-to-first-warning	Measure typical, tail, and user-visible delay.
Streaming throughput	Windows processed per second, queue growth, dropped frames	Determine whether processing keeps pace with incoming media.
Generalization	Held-out dataset or generator performance	Measure transfer beyond familiar training conditions.
Resources	CPU/GPU use and memory; energy where measurable	Assess feasibility on the specified deployment hardware.

For every latency claim, report the hardware, input configuration, window duration, update interval, actual model-processing time, and time required to accumulate enough evidence for a warning.

A detector that uses a two-second observation window cannot produce a complete-window assessment before that window has arrived, even if model inference itself is fast.

A real-time claim also requires that the incoming-media queue does not grow continuously under the stated test conditions.

J. Ablation studies

The planned ablations will compare: (1) audio-only detection; (2) video-only detection; (3) audio and video score fusion; (4) fusion with audio-video consistency; (5) fusion with quality-aware weighting; and (6) fusion with temporal aggregation. Each comparison should use the same partitioning and, where possible, comparable compute budgets. These tests will determine whether each component improves detection and streaming performance or merely increases computational cost.

K. Privacy and deployment considerations

Where feasible, inference should occur locally and raw call media should not be retained beyond the period required for authorized evaluation. Data collection must use informed consent and documented access, storage, retention, and deletion procedures. Access to a conferencing platform must be obtained through an authorized integration; this paper does not claim universal access to third-party call streams. Any deployment should expose uncertainty clearly and should not treat a model score as independent proof of a participant's identity.

VII. RESULTS AND DISCUSSION

A. Literature-based findings

No original experimental results are reported in this paper. The reviewed studies support several design choices, but they do not demonstrate that the proposed architecture is effective in live calls. FaceForensics++ motivates testing visual manipulation under multiple compression conditions [1]; FakeAVCeleb supports tests across genuine and manipulated audio-video combinations [3]; ASVspoof 2021 informs speech anti-spoofing evaluation under varied conditions [4]; and DeepfakeBench motivates reproducible visual-model comparisons [5]. These findings are literature-derived design implications, not experimental outcomes for the proposed detector.

B. Streaming performance and false alarms

The planned experiments should test whether longer observation windows improve detection enough to justify the additional warning delay. Frame sampling and compact models may reduce computational cost but can also reduce recall for short-lived artifacts. Results should report classification metrics together with time-to-first-warning, median and 95th-percentile latency, throughput, and queue growth under specified hardware and media settings. A genuine real-time claim requires that processing keeps pace with incoming media without sustained queue growth. The evaluation should also distinguish suspicious signals from poor-quality evidence. Timestamp misalignment, a missing face, silence, heavy compression, and network degradation should be included to measure false alarms and indeterminate outputs. Thresholds should reflect the intended use: routine meetings may require a low false-alarm rate, while security-sensitive workflows may prioritize recall and require a separate identity-verification step. In all cases, a model score is not independent proof of identity.

VIII. CONCLUSION

This paper reviewed relevant deepfake-detection resources and proposed a multimodal streaming framework combining audio and visual analysis, audio-video consistency, quality-aware fusion, temporal aggregation, and uncertainty-aware alerts. It defined a future evaluation protocol covering detection quality, latency, false alarms, throughput, generalization, and resource use. The architecture has not been implemented or experimentally validated, so no accuracy or real-time performance claim is made. Recorded datasets cannot fully model natural live calls, and generator or network shifts or coordinated audio-video synthesis may evade a detector. Future work should implement the pipeline, run controlled ablation and degradation tests, evaluate unseen generators, and verify authorized integration on specified hardware.

REFERENCES

[1] Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to detect manipulated facial images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 1–11.

[2] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, "Deepfakes and beyond: A survey of face manipulation and fake detection," *Inf. Fusion*, vol. 64, pp. 131–148, 2020, doi: 10.1016/j.inffus.2020.06.014.

[3] H. Khalid, S. Tariq, M. Kim, and S. S. Woo, "FakeAVCeleb: A novel audio-video multimodal deepfake dataset," in *Proc. NeurIPS Datasets and Benchmarks Track*, 2021.

[4] X. Liu, X. Wang, M. Sahidullah, J. Patino, H. Delgado, T. Kinnunen, M. Todisco, J. Yamagishi, N. Evans, A. Nautsch, and K. A. Lee, "ASVspoof 2021: Towards spoofed and deepfake speech detection in the wild," arXiv:2210.02437, 2022.

[5] Z. Yan, Y. Zhang, X. Yuan, S. Lyu, and B. Wu, "DeepfakeBench: A comprehensive benchmark of deepfake detection," arXiv:2307.01426, 2023.

[6] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, "Leveraging frequency analysis for deep fake image recognition," in *Proc. 37th Int. Conf. Mach. Learn. (ICML), Proc. Mach. Learn. Res.*, vol. 119, 2020, pp. 3247–3258.