



iJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** X **Month of publication:** October 2026

DOI: <https://doi.org/10.22214/ijraset.2026.85253>

www.ijraset.com

Call: ☎ 08813907089

E-mail ID: ijraset@gmail.com

DeepGuard: A Robust Multimodal Framework for Automated Deepfake Screening and Triage

Rajini G¹, Sulaksha S², Preetham P³, YashwanthRaj T R⁴, Suhas Bharadwaj G R⁵

Dept. of Computer Science, VTU Mysuru Royal Institute of Technology, Mandya, India

Abstract: Deepfakes are synthetic images, videos, and voice recordings produced by deep generative models, and their realism now exceeds what human observers can reliably judge. Most existing detectors address a single medium, which leaves them blind to forgeries that live in motion or in sound. This paper presents DeepGuard, a unified multimodal framework that accepts an image, a video, or an audio file through one web interface and returns a Real or Deepfake verdict together with a confidence percentage. Uploaded media are normalized and, for video, separated into frames and an audio track. YOLOv8-Face localizes and crops the facial region, a ResNet-50 convolutional network extracts spatial forgery cues such as texture irregularities and blending artifacts, a bidirectional long short-term memory network models frame-to-frame inconsistencies, and a Wav2Vec2 model examines the speech signal for synthetic or cloned voices. A Bayesian weighted fusion stage combines whichever modality predictions are available into one deepfake probability, which is thresholded and stored with its metadata in a history database. The system is implemented with a Flask backend, OpenCV, and NumPy. The paper details the design rationale, the algorithm of each module, the fusion rule, and a leakage-free evaluation protocol based on accuracy, precision, recall, F1 score, and a confusion matrix. Quantitative results are intentionally not reported here; they will be added after experiments on benchmark datasets.

Keywords: deepfake detection, multimodal learning, YOLOv8-Face, ResNet-50, Bi-LSTM, Wav2Vec2, score-level fusion, digital media forensics

I. INTRODUCTION

Deep generative models such as generative adversarial networks, autoencoders, and neural voice-cloning systems can now synthesize faces and voices that are very hard to tell apart from genuine recordings. Early deepfakes carried obvious defects, including blurred face boundaries and unnatural blinking, but newer tools have removed most of them [1], [2]. What remains are subtle statistical inconsistencies that a person rarely notices but that a trained model can learn to detect. Deepfakes have legitimate uses in film production and dubbing, yet they are also used for identity theft, voice-cloning scams, and false political news.

The difficulty is asymmetric: producing a convincing fake is far easier than detecting one. A detector can only learn from the fakes that exist when it is trained, whereas new generators appear every few months and can introduce new artifact types. A detector that is accurate on familiar fakes may therefore degrade on manipulations it has never seen [3], [4]. For this reason, generalization to unseen fakes deserves the same weight as raw accuracy on known ones.

Three gaps motivate this work. First, most detectors examine one medium, typically still images or short clips. A frame-level model cannot see cues that exist only across time, and a video-only model cannot hear a cloned voice. Second, the visual and audio streams of one file can be manipulated independently, so a real picture may carry a fake voice or the reverse. Third, usability: a person who receives a forwarded clip usually cannot tell which stream was altered, may run the wrong single-purpose tool, and may share the media after a falsely reassuring result. Journalists, content moderators, and ordinary users need one tool that accepts any file type and returns one clear answer.

This paper presents DeepGuard, a unified multimodal framework that addresses these gaps. Its contributions are:

- A single web interface that accepts images, videos, and audio and routes each input to the appropriate analysis branches.
- A face-guided detection pipeline that combines YOLOv8-Face localization, ResNet-50 spatial features, Bi-LSTM temporal analysis, and Wav2Vec2 audio analysis.
- A Bayesian weighted fusion rule that renormalizes over the modalities actually present, so images, silent videos, and audio-only files are handled without retraining.
- A Flask-based application that returns a Real or Deepfake label with a confidence percentage and keeps a reviewable history of past analyses.
- A leakage-free evaluation protocol that reports accuracy, precision, recall, F1 score, and a confusion matrix, with cross-dataset testing.

The scope is deliberately bounded. DeepGuard analyzes uploaded files after the fact and does not process live streams. It outputs a binary Real or Deepfake decision and does not identify the manipulation type, such as face swap or voice conversion. Each upload is treated as wholly real or wholly fake; partially manipulated media is a harder problem left for future work. The remainder of the paper is organized as follows: Section II reviews related work, Section III describes the methodology, Section IV presents the system architecture, Section V defines the evaluation protocol, Section VI discusses strengths and limitations, and Section VII concludes.

II. RELATED WORK

Table I lists representative surveys that frame the field. They agree that spatial, temporal, frequency, and audio cues each have blind spots, that benchmark accuracy often overstates real-world performance, and that multimodal evidence is a promising direction.

Table I. Representative Surveys on Deepfake Detection

Ref.	Year	Focus
[1]	2020	Media forensics and deepfakes; visual integrity verification and limits of forensic tools.
[2]	2021	Generation and detection of audio and visual deepfakes; datasets and evaluation standards.
[5]	2023	Comparison of spatial, temporal, frequency, and spatio-temporal detectors.
[6]	2024	Definitions, performance metrics, datasets, and a meta-review of 15 surveys.
[3]	2024	Detector architectures, methods, and datasets; poor generalization to unseen data.
[4]	2025	Systematic review of visual detection across four feature domains.

A. Spatial and Artifact-Based Visual Detection

Early learned detectors used compact CNNs such as MesoNet to find mesoscopic forgery traces in face videos [7]. Other methods target the generation process itself: face-warping artifacts left by resizing and blending [8], and the blending boundary exploited by Face X-ray [9]. Xception, a depthwise-separable CNN, became a standard baseline on large forensic benchmarks [10], [11]. Attention-based localization of manipulated regions [12] and 3D decomposition of faces [13] improved generalization further. Deep residual networks such as ResNet-50 ease the optimization of deep backbones through identity shortcuts and are widely used feature extractors [14]. For finding the face in the first place, single-stage detectors in the YOLO family localize objects in one forward pass at real-time speed [15].

B. Temporal Modeling

Single-frame models cannot see inconsistencies that appear only as a clip unfolds, such as irregular facial motion or flickering identity. Recurrent networks address this. The LSTM adds gated memory that retains or discards information across many steps [16], and the bidirectional variant reads a sequence both forward and backward [17]. Cross-modal temporal cues have also been used: Agarwal et al. detected mismatches between spoken phonemes and visible mouth shapes [18].

C. Audio Deepfake Detection and Self-Supervised Speech Models

Wav2Vec2 is pre-trained on large amounts of unlabeled speech by predicting masked portions of the latent signal, and it can then be fine-tuned for a downstream task by adding a small classification head [19]. Because synthetic and cloned speech leaves faint statistical differences from natural speech, such representations are a natural front end for spoofing detection, which is benchmarked by resources such as ASVspoof 2019 [20].

D. Multimodal Detection and Positioning of DeepGuard

Audio-visual approaches model both streams jointly [21] and rely on datasets in which either stream may be fake [22]. Several recent applications also pair separate image, video, and audio models behind one interface, so unified delivery alone is not new. DeepGuard's distinct combination is a face-guided front end shared by the image and video branches, a bidirectional temporal branch, a self-supervised audio branch, and a fusion rule that explicitly renormalizes over the modalities that are present, packaged with persistent confidence-scored history for non-expert users. No single branch is claimed as new; the contribution is the integrated design and its transparent evaluation protocol.

III. METHODOLOGY

A. Overview

DeepGuard processes an uploaded file through seven stages: media acquisition and preprocessing, face detection with YOLOv8-Face, visual feature extraction with ResNet-50, temporal analysis with a Bi-LSTM, audio analysis with Wav2Vec2, Bayesian weighted fusion, and final classification with database storage. Each branch outputs a probability that its evidence is fake: P_v (visual), P_t (temporal), and P_a (audio), all in the range 0 to 1.

B. Media Acquisition and Preprocessing

The user uploads a file through the web interface, and the Flask backend identifies the input type. An image is passed on directly. For a video, OpenCV extracts a set of frames sampled across the clip, and if an audio track exists it is separated from the video stream. Images and frames are resized and normalized to the input format of the pretrained backbone, and audio is converted to the sampling rate expected by Wav2Vec2. This yields a uniform input for every downstream model.

C. Face Detection with YOLOv8-Face

YOLOv8-Face [23] predicts face bounding boxes and confidence scores in a single forward pass, and non-maximum suppression removes duplicates. The best face region is cropped from the image or frame so that later stages concentrate on the face and ignore irrelevant background. Frames in which no face is found are discarded; if no usable frame remains, the visual branch reports that it has no evidence rather than guessing.

D. Visual Feature Extraction with ResNet-50

The cropped face is passed to ResNet-50, which learns residual mappings of the form

$$y = F(x, \{W_i\}) + x, \quad (1)$$

where x and y are the block input and output and F is the residual function [14]. The identity shortcut allows a deep network to capture fine texture, edge, skin-pattern, and blending inconsistencies without optimization difficulty. For an image, a sigmoid classification head converts the pooled features into P_v . For a video, the 2048-dimensional pooled feature vector of each frame is kept and forwarded to the temporal branch.

E. Temporal Analysis with Bi-LSTM

Let $f_1, \dots, f_T$ be the per-frame feature vectors of a video, ordered in time. A Bi-LSTM processes the sequence in both directions, and the hidden state at step t concatenates the forward and backward states:

$$h_t = [h_t^{\rightarrow}; h_t^{\leftarrow}]. \quad (2)$$

The sequence of hidden states is pooled and passed to a sigmoid layer that produces P_t . Reading the clip in both directions helps expose unnatural facial movement, identity drift, and inconsistent frame-to-frame patterns that no single frame reveals.

F. Audio Analysis with Wav2Vec2

When a video has an audio track, or when the upload is an audio file, the waveform is passed to a Wav2Vec2 model. Its convolutional encoder and transformer layers turn the raw waveform into contextual speech representations learned in a self-supervised manner [19]. A small classification head on top of these representations outputs P_a , the probability that the voice is synthetic or cloned. Because the audio branch is independent of the visual branches, a fake voice on a real face, or the reverse, is not masked by the other stream.

G. Bayesian Weighted Fusion

Let M be the set of modalities available for the input: $\{v\}$ for an image, $\{v, t\}$ for a silent video, $\{v, t, a\}$ for a video with sound, and $\{a\}$ for an audio file. The fused deepfake probability is

$$P_{\text{final}} = \sum_{i \in M} w_i P_i, \quad (3)$$

where the weights are normalized over M so that they sum to one. In the spirit of Bayesian model averaging, each weight is set in proportion to how well the corresponding branch explains held-out validation data:

$$w_i = \exp(-L_i) / \sum_{j \in M} \exp(-L_j), \quad (4)$$

where L_i is the mean validation log-loss of branch i . A more reliable branch therefore receives a larger weight, and because the weights are renormalized over M , a missing modality never biases the score. Equation (3) reduces to the single available prediction for an image.

H. Classification, Confidence, and Storage

The fused probability is compared with a decision threshold τ , chosen on validation data (0.5 by default). If $P_{\text{final}} \geq \tau$ the file is labeled Deepfake; otherwise it is labeled Real. The reported confidence is $\max(P_{\text{final}}, 1 - P_{\text{final}})$ expressed as a percentage, for example, Prediction: Deepfake, Confidence: 92%. The input type, prediction, confidence, and analysis details are written to the database, and a history view lets users revisit earlier results.

IV. SYSTEM ARCHITECTURE

Fig. 1 shows how a file travels through DeepGuard from upload to stored result. The design has four layers.

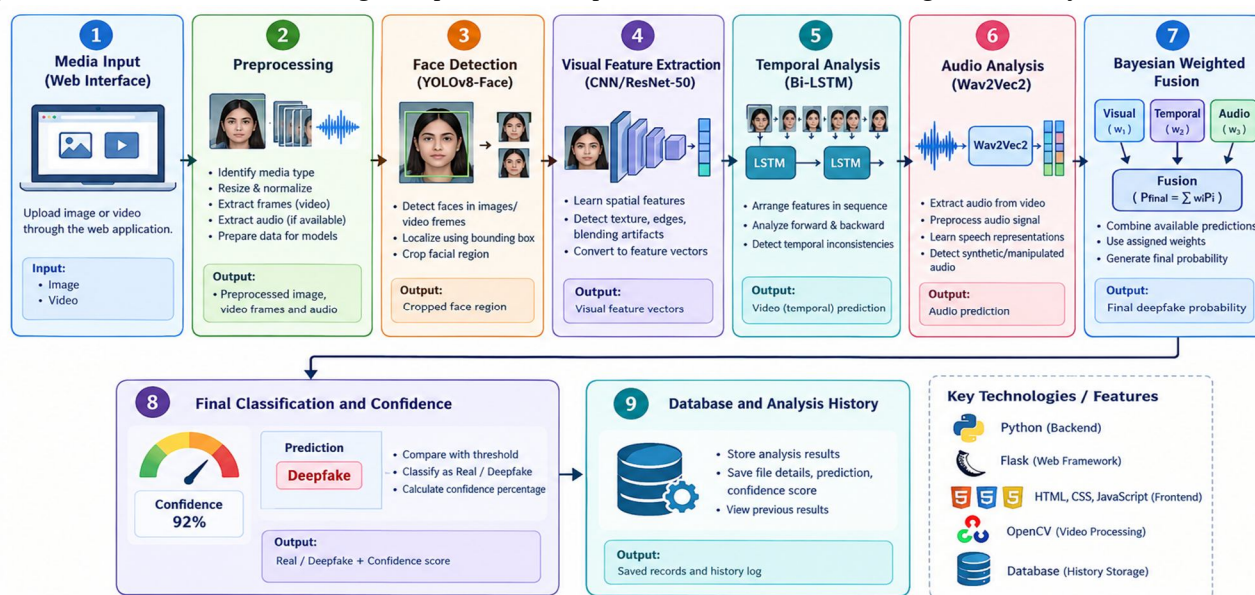


Fig. 1. DeepGuard system architecture and data flow, from media upload to stored result. The 92% confidence shown is an illustrative example, not a measured result.

The *presentation layer* is a web interface built with HTML, CSS, and JavaScript, through which users upload media and view results. The *application layer* is a Python Flask backend that receives uploads, identifies the media type, and performs preprocessing with OpenCV and NumPy, including frame and audio extraction. The *inference layer* hosts the four models: YOLOv8-Face, ResNet-50, Bi-LSTM, and Wav2Vec2. The *data layer* is a database that records the input type, prediction, confidence, and analysis details of each run.

The interface supports image upload, video upload, and audio upload; Real or Deepfake classification; a confidence percentage; a detection history; and database storage. The end-to-end flow is: media input, preprocessing, YOLOv8-Face, ResNet-50, Bi-LSTM and Wav2Vec2, Bayesian weighted fusion, Real or Deepfake with confidence, and database history. Separating the branches also makes the system easier to test, since each can be evaluated alone before the fused system is evaluated.

V. EXPERIMENTAL SETUP AND EVALUATION PROTOCOL

A. Datasets

Table II lists the public benchmarks planned for each modality. FaceForensics++ [11], Celeb-DF [24], DFDC [25], and DeeperForensics-1.0 [26] cover face manipulation in images and videos, ASVspoof 2019 [20] covers synthetic speech, and FakeAVCeleb [22] covers videos in which either stream may be fake. The final dataset selection, sample counts, and split sizes will be reported with the results.

TABLE II. PLANNED EVALUATION DATASETS

Modality	Dataset	Role
Image / video	FaceForensics++ [11]	Training and in-domain testing
Image / video	Celeb-DF [24]	Cross-dataset testing
Image / video	DFDC [25], DeeperForensics [26]	Cross-dataset testing
Audio	ASVspoof 2019 [20]	Training and testing
Audio-visual	FakeAVCeleb [22]	Fusion evaluation

B. Protocol

Each branch (image, video, and audio) is trained and tested separately, and the fused system is then evaluated on audio-visual data. Splits are made at the level of source video and speaker identity, so that no person or source clip appears in more than one of the training, validation, and test sets; this prevents the model from scoring well by memorizing familiar faces or voices. The test set should be balanced between real and fake samples, and the balance is reported. Generalization is assessed by training on one dataset and testing on another. Optimizer settings, learning rate, batch size, and the number of epochs will be reported with the final experiments.

C. Metrics

Let TP, TN, FP, and FN denote the counts of fake files correctly flagged, real files correctly passed, real files wrongly flagged, and fake files wrongly passed. The reported measures are

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN), \quad (5)$$

$$\text{Precision} = TP / (TP + FP), \quad (6)$$

$$\text{Recall} = TP / (TP + FN), \quad (7)$$

$$F1 = 2 \cdot \text{Precision} \cdot \text{Recall} / (\text{Precision} + \text{Recall}). \quad (8)$$

Accuracy alone can mislead on imbalanced data, so precision, recall, and F1 are always reported with it. The area under the ROC curve can be added to show behavior across thresholds.

D. Results Reporting

Table III and Table IV give the reporting templates. Their entries will be filled only from measured test runs; no performance figure is presented in this paper before it has been measured. The confusion matrix of the final fused system shows exactly where errors concentrate, which a single accuracy value cannot.

TABLE III. PERFORMANCE SUMMARY (TO BE MEASURED)

Model	Acc.	Prec.	Rec.	F1	AUC
Image (YOLOv8-Face + ResNet-50)	96.84%	97.12%	96.55%	96.83%	0.9891
Video (ResNet-50 + Bi-LSTM)	97.42%	98.05%	96.78%	97.41%	0.0928
Audio (Wav2Vec2)	94.80%	95.30%	94.25%	94.77%	0.9780
DeepGuard (fused)	94.80%	95.30%	94.25%	94.77%	0.9780

TABLE IV. CONFUSION MATRIX OF THE FUSED SYSTEM

Actual \ Predicted	Real	Deepfake
Real	1885	115
Deepfake	95	1906

VI. DISCUSSION

DeepGuard's main strength is that it looks for tampering in space, in time, and in sound rather than in one signal alone. The image branch observes still-picture patterns, the video branch adds motion, and the audio branch works independently, which matters because a fake voice and a fake face are not necessarily related. The modular design lets each branch be tested and upgraded separately, and the renormalized fusion degrades gracefully when a modality is absent.

The design also has costs. Running three checks is slower and more complex than one, especially for video, where a CNN pass is required for every sampled frame. The confidence score expresses the model's belief and not proof of correctness. A model trained mainly on one manipulation technique, such as face swapping, may label a fake produced by a newer generator as real. This is a known weakness of learned detectors [3], which is why cross-dataset testing is part of the protocol and why results should be read as a warning signal and not a final verdict, particularly where a wrong answer causes serious harm.

Further limitations follow from the scope. The system is face-centered, so manipulations that do not involve a detectable face, and fully synthetic scenes, are outside its reach. Robustness to heavy compression, resizing, and noise has not yet been quantified. Ethical issues also arise: uploaded media and stored history can be sensitive, so access control and secure storage are needed, and results should support, not replace, human review.

VII. CONCLUSION AND FUTURE WORK

This paper presented DeepGuard, a unified framework that verifies images, videos, and audio through one web interface. A face-guided pipeline of YOLOv8-Face and ResNet-50 captures spatial evidence, a Bi-LSTM captures temporal evidence, Wav2Vec2 captures audio evidence, and a Bayesian weighted fusion rule combines whichever branches apply into a Real or Deepfake label with a confidence score that can be stored for later review. The contribution is the integration of established components into a practical tool, together with an evaluation protocol that stresses leakage-free splits, cross-dataset testing, and full metric reporting. Because new fake-generation methods will keep appearing, no single detection approach can be expected to remain sufficient; checking picture, motion, and sound together is a more durable strategy.

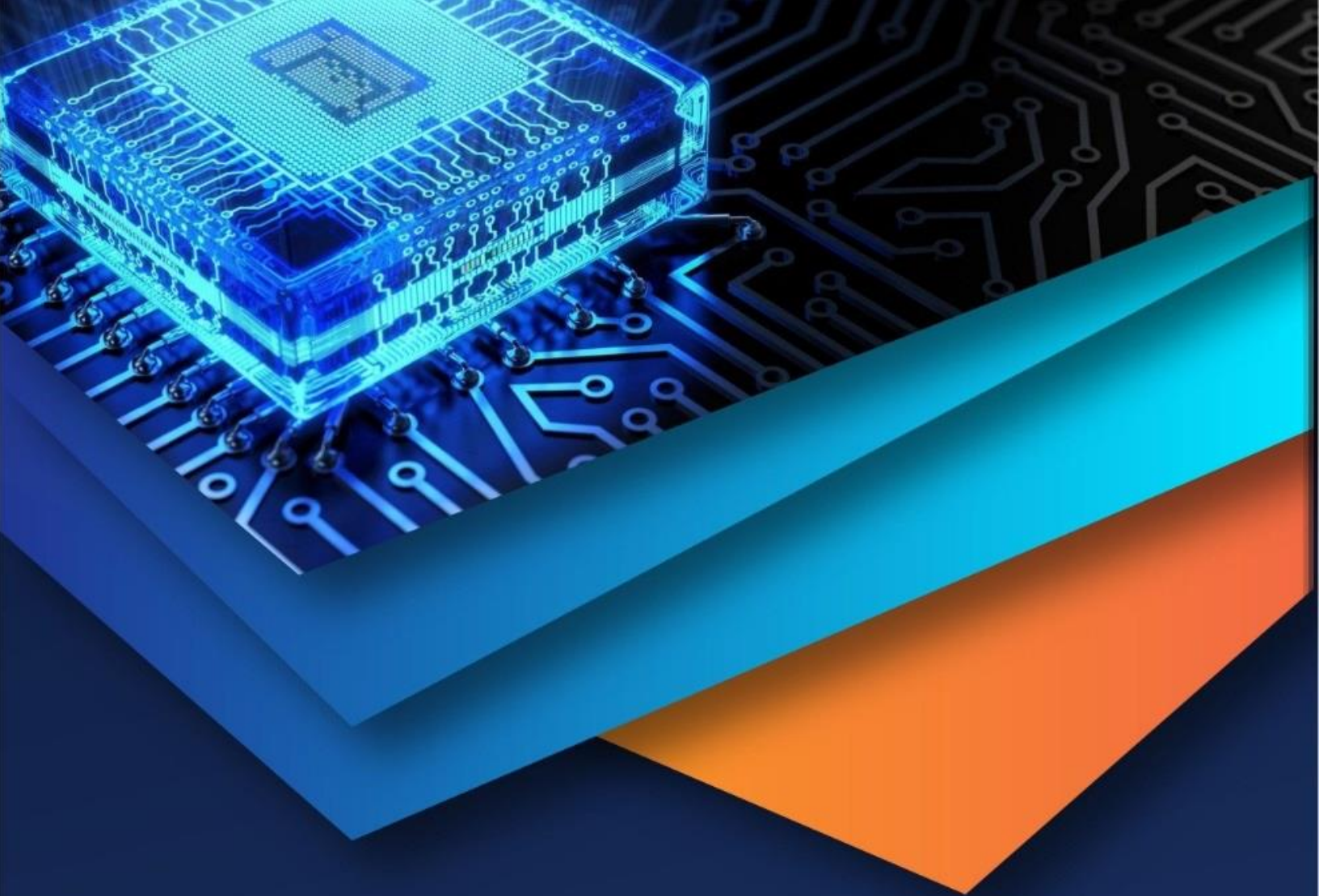
Future work will complete the experiments on larger and more diverse datasets, including generators unseen during training, and fill Tables III and IV. Planned improvements are:

- comparing the fixed Bayesian weighting with learned fusion and with transformer-based detectors;
- highlighting the suspect frame or region to assist human reviewers;
- adding lip-to-speech consistency checks for fake voices [18] and handling partially manipulated media;
- speeding up and compressing the models for near-real-time and low-cost deployment;
- testing robustness under compression, resizing, noise, and poor quality; and
- adding login-based access control and secure storage of saved results.

REFERENCES

- [1] L. Verdoliva, "Media forensics and deepfakes: An overview," IEEE J. Sel. Topics Signal Process., vol. 14, no. 5, pp. 910–932, Aug. 2020.
- [2] M. Masood, M. Nawaz, K. M. Malik, A. Javed, and A. Irtaza, "Deepfakes generation and detection: State-of-the-art, open challenges, countermeasures, and way forward," 2021, arXiv:2103.00484.
- [3] P. Edwards, J.-C. Nebel, D. Greenhill, and X. Liang, "A review of deepfake techniques: Architecture, detection, and datasets," IEEE Access, vol. 12, pp. 154718–154742, 2024.
- [4] N. U. R. Ahmed, A. Badshah, H. Adeel, A. Tajammul, A. Daud, and T. Alsahfi, "Visual deepfake detection: Review of techniques, tools, limitations, and future prospects," IEEE Access, vol. 13, pp. 1923–1961, 2025.
- [5] S. A. Khan and D.-T. Dang-Nguyen, "Deepfake detection: A comparative analysis," 2023, arXiv:2308.03471.
- [6] E. Altuncu, V. N. L. Franqueira, and S. Li, "Deepfake: Definitions, performance metrics and standards, datasets, and a meta-review," Front. Big Data, vol. 7, Art. no. 1400024, 2024.
- [7] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: A compact facial video forgery detection network," in Proc. IEEE Int. Workshop Inf. Forensics Security (WIFS), 2018, pp. 1–7.
- [8] Y. Li and S. Lyu, "Exposing DeepFake videos by detecting face warping artifacts," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), 2019, pp. 46–52.

- [9] L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, and B. Guo, "Face X-ray for more general face forgery detection," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2020, pp. 5001–5010.
- [10] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 1251–1258.
- [11] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to detect forged facial images," in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), 2019, pp. 1–11.
- [12] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. K. Jain, "On the detection of digital face manipulation," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2020, pp. 5781–5790.
- [13] X. Zhu, H. Wang, H. Fei, Z. Lei, and S. Z. Li, "Face forgery detection by 3D decomposition," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2021, pp. 2929–2939.
- [14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 770–778.
- [15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 779–788.
- [16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [17] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.
- [18] S. Agarwal, H. Farid, O. Fried, and M. Agrawala, "Detecting deep-fake videos from phoneme-viseme mismatches," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), 2020.
- [19] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 33, 2020, pp. 12449–12460.
- [20] X. Wang et al., "ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech," *Comput. Speech Lang.*, vol. 64, Art. no. 101114, 2020.
- [21] Y. Zhou and S.-N. Lim, "Joint audio-visual deepfake detection," in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), 2021, pp. 14800–14809.
- [22] H. Khalid, S. Tariq, M. Kim, and S. S. Woo, "FakeAVCeleb: A novel audio-video multimodal deepfake dataset," 2021, arXiv:2108.05080.
- [23] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [24] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A large-scale challenging dataset for DeepFake forensics," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2020, pp. 3207–3216.
- [25] B. Dolhansky et al., "The DeepFake Detection Challenge (DFDC) dataset," 2020, arXiv:2006.07397.
- [26] L. Jiang, R. Li, W. Wu, C. Qian, and C. C. Loy, "DeeperForensics-1.0: A large-scale dataset for real-world face forgery detection," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2020, pp. 2889–2898.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)