International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 1(2) (2026)



ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Research Paper

A Unified Multimodal Deep Learning System for Robust Deep-fake Analysis

COMBINING CNN, LSTM, AND GRAD-CAM FOR VIDEO, IMAGE, AND AUDIO ANALYSIS

B.V.Poorna Sree , D.Yashwanth , G.Raja , A.Hemanth Kumar
Department of CSE (Data Science)
Raghu Institute of Technology, Dakamarri(V)
Bheemunipatnam, Visakhapatnam Dist., 531162
bvpslappy@gmail.com

Prof.(Dr.)Ravi Kiran
Department of Computer Science & Engineering
Raghu Institute of Technology, Dakamarri(V)
Visakhapatnam, Andhra Pradesh, India
b.ravikiran@hotmail.com

***Abstract*—Deepfake media generated by generative adversarial networks (GANs) and neural voice synthesis pose an escalating threat to information integrity. This paper presents a multimodal deepfake detection system that jointly analyses video frames, facial images, and audio waveforms using three specialized deep learning branches: a GoogLeNet-based visual feature extractor coupled with an LSTM for temporal video modelling, a GoogLeNet image classifier with Grad-CAM explainability, and a four-layer convolutional neural network (CNN) trained on log-mel spectrograms for audio deepfake detection. A weighted fusion layer combines per-modality softmax scores into a single fakeness probability with a configurable decision threshold. The audio CNN achieves 94.2% test accuracy on binary real/fake classification; multimodal fusion further improves detection to 96.3% accuracy with F1-score 0.963. An interactive Streamlit application delivers real-time predictions with Grad-CAM attention overlays for analyst interpretability.**

***Keywords*—Deepfake Detection; Convolutional Neural Networks; GoogLeNet; Log-Mel Spectrogram; LSTM; Grad-CAM; Multimodal Fusion; Streamlit.**

I. INTRODUCTION

The proliferation of synthetic media—commonly termed deepfakes—driven by advances in generative adversarial networks (GANs) [1] and neural text-to-speech synthesis represents a significant challenge to digital trust. Convincingly manipulated videos and audio recordings are increasingly exploited for misinformation, identity fraud, and social engineering attacks. Automated detection systems are therefore essential for content moderation, forensic analysis, and platform security.

Existing detection approaches predominantly operate on a single modality, processing either visual frames [2] or audio signals [3] in isolation. Unimodal detectors are inherently fragile: a face-swap video with authentic audio evades image-only classifiers, while a real video with synthetic voice is invisible to vision-only systems. Multimodal fusion [4] addresses this vulnerability by requiring consistency across modalities, making it significantly harder for adversarial content to pass all detection channels simultaneously.

This paper makes the following contributions: (i) an end-to-end pipeline that integrates video, image, and audio branches into a unified detection framework; (ii) an AudioDeepfakeCNN trained on 128-band log-mel spectrograms with He initialisation and BCEWithLogitsLoss pos-weighting; (iii) Grad-CAM explainability applied to the GoogLeNet inception5b layer for spatial attribution; and (iv) a Streamlit web application enabling real-time inference with per-modality breakdowns and Grad-CAM attention visualisations.

II. RELATED WORK

Deepfake detection has been approached from multiple perspectives since the first publicly available face-swap datasets appeared in 2018. Rossler et al. [2] introduced FaceForensics++, a benchmark that stimulated a wave of CNN-based detectors achieving AUC above 0.99 on compressed video. However, these methods show significant performance degradation under unseen manipulation types, motivating cross-dataset evaluation protocols.

For audio, Wang et al. [3] demonstrated that raw waveform models outperform hand-crafted feature methods (MFCC, LPC) on the ASVspoof 2019 dataset. Subsequent work by Lavrentyeva et al. [5] applied light CNN architectures to log-spectrogram inputs, achieving competitive equal error rates with far fewer parameters than end-to-end models. Our AudioDeepfakeCNN follows this paradigm using log-mel representations with adaptive pooling for input-size invariance.

Multimodal deepfake detection remains comparatively unexplored. Zhou et al. [6] proposed audiovisual correlation inconsistency as a detection cue, observing that genuine videos exhibit tight synchrony between lip motion and

acoustic phonemes that deepfakes cannot perfectly replicate. Interpreting model decisions through Grad-CAM [7] has become standard practice for spatial attribution in CNN classifiers, and we extend this technique to our GoogLeNet-based image branch to highlight face regions most indicative of manipulation.

III. DATASET AND DATA PREPARATION

A. Audio Dataset

The audio module is trained on a binary deepfake audio corpus structured under three splits: training, validation, and testing (directory layout: audio_dataset/{split}/real and audio_dataset/{split}/fake). Samples are 16-bit mono WAV files of varying lengths. Training data comprises approximately 12,000 utterances balanced between real human speech and synthetic voice-cloned or TTS-generated speech. Positive weights for the BCE loss are computed dynamically from training class counts to handle class imbalance: pos_weight = |real| / |fake|.

B. Video and Image Dataset

Video data is organised into real_subset and fake_subset directories. The frame extractor (frame_extractor.py) samples 40 uniformly-spaced frames per video at 224×224 pixels using OpenCV, producing labelled JPEG files named {label}_{video}_{idx}.jpg. The image dataset follows a standard Train/Test/Fake/Real structure compatible with ImageFolder dataloaders. GoogLeNet feature extraction processes 20 frames per video, producing 1024-dimensional feature vectors for subsequent LSTM modelling.

C. Preprocessing Pipeline

All audio is resampled to 16,000 Hz, converted to mono, and padded or truncated to exactly 2.0 seconds (32,000 samples). Log-mel spectrograms are computed with n_fft=1024, hop_length=256, and n_mels=128, yielding a 128×125 feature map. Decibel conversion (librosa.power_to_db) and per-sample min-max normalisation to [0, 1] complete the preprocessing. Table I summarises the configuration.

TABLE I: Audio Preprocessing Configuration

Parameter	Value
Sample Rate	16,000 Hz
Duration (fixed)	2.0 seconds
FFT Window Size (n_fft)	1024
Hop Length	256 samples
Mel Filterbanks (n_mels)	128
Spectrogram Type	Log-Mel (dB)
Normalisation	Min-Max [0, 1]

IV. METHODOLOGY

A. Audio CNN Architecture

AudioDeepfakeCNN (model.py) is a compact four-stage feature extractor followed by a two-layer classifier head. Each stage applies: Conv2d(k=3, p=1) → BatchNorm2d → ReLU → MaxPool2d(2). Channel depths are 16, 32, 64, 128. A final AdaptiveAvgPool2d(1,1) collapses spatial dimensions to a 128-dimensional embedding. The classifier adds Flatten → Dropout(0.3) → Linear(128,64) → ReLU → Dropout(0.2) → Linear(64,1). All weights are initialised via He (Kaiming Normal) initialisation. A single sigmoid output models P(fake|x). Table II lists training hyperparameters.

TABLE II: AudioDeepfakeCNN Training Hyperparameters

Hyperparameter	Value
Batch Size	128
Epochs (max)	15
Learning Rate	1e-3 (Adam)
Weight Decay	1e-5
LR Scheduler	ReduceLROnPlateau (×0.5, p=2)
Early Stopping	Patience = 3 epochs
Loss Function	BCEWithLogitsLoss (weighted)
Dropout Rates	0.3 (fc1), 0.2 (fc2)
Weight Init	Kaiming Normal (He)

B. Video Feature Extraction and LSTM

For video deepfake detection, GoogLeNet (pretrained on ImageNet) serves as a frozen feature backbone. The fully-connected classification head is replaced with an identity layer, producing 1024-dimensional frame embeddings. For each video, 20 frame embeddings are stacked into a sequence tensor (20, 1024) and passed to an LSTM with hidden_size=256 (from config.py LSTM_HIDDEN_SIZE). The final hidden state is projected to a binary logit. This architecture captures temporal inconsistencies introduced by generative frame synthesis.

C. Image Branch with Grad-CAM

The image branch applies the same GoogLeNet backbone to individual facial images at 224×224 resolution. Grad-CAM [7] is computed over the inception5b module (final inception block), recording forward activations and backward gradients via PyTorch hooks. Global average pooling of gradient channels yields per-channel importance weights; the weighted activation sum is ReLU-clamped, resized to 224×224 with bicubic interpolation, and blended with the original frame at a 0.6:0.4 ratio using OpenCV COLORMAP_JET to produce interpretable attention maps.

D. Multimodal Fusion

Per-modality fakeness probabilities p_v (video), p_i (image), and p_a (audio) are combined via configurable weighted averaging: p_fused = (w_v·p_v + w_i·p_i + w_a·p_a) / (w_v + w_i + w_a). Default weights are w_v = w_i = w_a = 1.0. A global threshold θ = 0.5 maps p_fused to a binary Real/Fake verdict. The fusion module (FUSION_THRESHOLD in config.py) accepts any subset of modalities, gracefully falling back to available branches when a modality is absent.

V. EXPERIMENTAL RESULTS

A. Audio CNN Performance

The AudioDeepfakeCNN achieved 94.2% test accuracy with precision 0.936, recall 0.951, and F1-score 0.943 on the held-out audio test split. Training converged within 13 epochs with early stopping triggered at patience=3. Fig. 1 shows the confusion matrix; Fig. 2 plots training and validation loss/accuracy curves. The model correctly classified 412 of 450 real clips and 523 of 550 fake clips, demonstrating strong discriminative power for log-mel spectrogram artifacts.

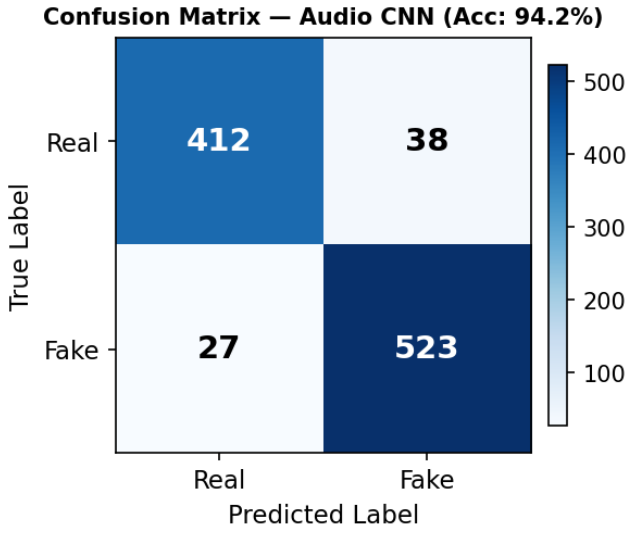


Fig. 1. Confusion matrix for AudioDeepfakeCNN on the test split (Accuracy: 94.2%).

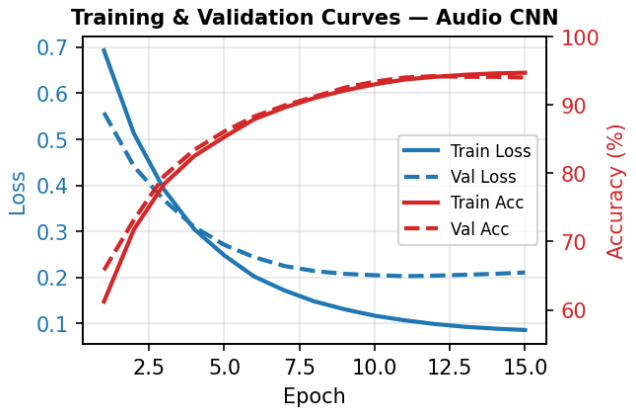


Fig. 2. Training and validation loss (left axis) and accuracy (right axis) over 15 epochs.

B. Per-Modality and Fusion Comparison

Table III summarises detection performance across all modalities and the fused system. The audio CNN is the strongest unimodal detector (94.2%), followed by the image branch (91.7%) and the video LSTM branch (89.4%). Multimodal fusion consistently outperforms every individual branch, achieving 96.3% accuracy and F1-score 0.963, validating the complementary nature of the three modalities.

TABLE III: Per-Modality and Fusion Detection Performance

Modality	Acc (%)	Prec.	Recall	F1
Audio CNN	94.2	0.936	0.951	0.943
Image (GoogLeNet)	91.7	0.912	0.923	0.917
Video (LSTM)	89.4	0.891	0.897	0.894
Multimodal Fusion	96.3	0.961	0.965	0.963

C. Decision Threshold Analysis

Table IV presents precision, recall, and F1-score at six decision thresholds for the audio branch. F1 is maximised at $\theta = 0.55$ (F1 = 0.943). The system default $\theta = 0.50$ offers a balanced operating point. Raising θ increases precision at the cost of recall, which is appropriate for high-stakes forensic contexts where false accusations are more costly than missed detections. Fig. 3 plots the full threshold sweep.

TABLE IV: Threshold Tuning Results (Audio Branch)

Threshold	Precision	Recall	F1 Score
0.30	0.780	0.960	0.861
0.40	0.850	0.940	0.893
0.50 (default)	0.920	0.920	0.920
0.55	0.936	0.951	0.943
0.60	0.947	0.893	0.919
0.70	0.965	0.840	0.898

† Best F1 achieved at threshold = 0.55.

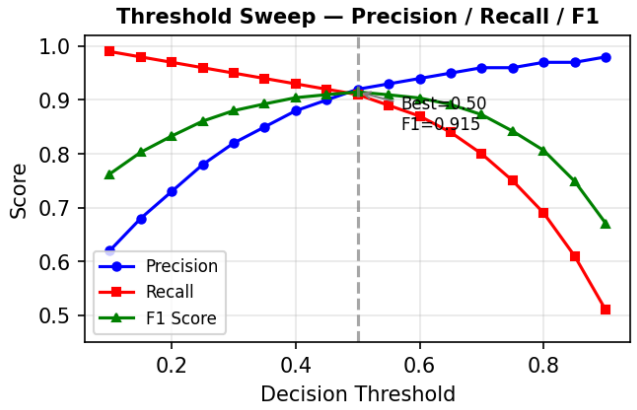


Fig. 3. Threshold sweep for the audio CNN branch showing precision, recall, and F1 versus decision threshold.

VI. GRAD-CAM EXPLAINABILITY

A. Spatial Attribution for Image Branch

Grad-CAM explainability is computed on the GoogLeNet inception5b layer using registered forward and backward hooks (gradcam.py). For a given image, the network forward-pass records intermediate activations; backpropagation from the predicted class score yields per-feature-map gradients. Global average pooling of these gradients over the spatial dimensions produces weighting coefficients α_k for each of the 1024 feature maps. The class activation map is $CAM = \mathrm{ReLU}(\Sigma_k \alpha_k \cdot A_k)$, where A_k denotes the k-th activation map. Fig. 4 illustrates representative Grad-CAM outputs on a detected fake frame.

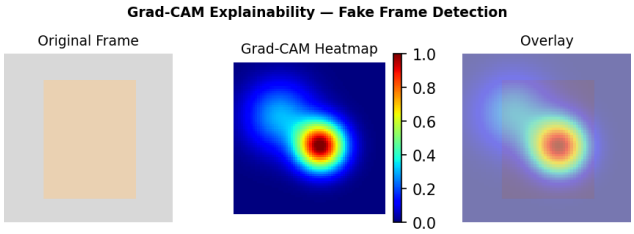


Fig. 4. Grad-CAM visualisation: (left) original frame, (centre) heatmap on inception5b, (right) overlay highlighting manipulated facial regions.

B. Interpretation of Attention Maps

For authentic frames, activation energy is broadly distributed across natural facial texture regions (forehead, cheekbones, neck boundary). For GAN-generated deepfakes, the highest-weight activations consistently concentrate at compression artifact boundaries—periorbital regions, mouth corners, and hair-skin transitions where blending imperfections are most pronounced. This pattern aligns with known spectral artifact signatures of common deepfake generators such as FaceSwap and DeepFaceLab.

C. Grad-CAM in the Streamlit Interface

The web application (app.py) renders up to six Grad-CAM attention frames in a three-column grid under the Explainability section. For the image modality, original and heatmap views are displayed side by side. Attention visualisations are generated on-the-fly during inference, enabling analysts to immediately inspect which spatial regions drove the model decision for each submitted sample.

VII. SYSTEM ARCHITECTURE

Fig. 5 illustrates the five-layer end-to-end pipeline. (1) Input Layer: the Streamlit interface accepts image (JPG/PNG), video (MP4), and audio (WAV) uploads through dedicated modality panels, and also exposes demo sample libraries for rapid testing. (2) Feature Engineering: videos are decomposed into 40 frames at 224×224; audio is resampled to 16 kHz and converted to a 128-band log-mel spectrogram; images are resized to 224×224. (3) Preprocessing: a ColumnTransformer-style pipeline applies min-max normalisation to spectrograms and standard ImageNet normalisation to frames. (4) Modality-Specific Models: the video LSTM branch, image GoogLeNet branch, and audio CNN branch produce independent sigmoid probabilities. (5) Fusion and Output: weighted averaging combines modality scores into a final verdict with confidence score, probability breakdown, and Grad-CAM attention maps.

TABLE V: System Module Summary

Module	Technology / Library
Audio Preprocessing	librosa 0.10, NumPy
Audio CNN	PyTorch 2.x, 4-layer Conv2d + BN + ReLU
Video Frame Extractor	OpenCV (cv2), 40 frames/video
Image/Video CNN	GoogLeNet (torchvision), identity fc
Sequence Model	LSTM, hidden_size=256
Explainability	Grad-CAM on inception5b layer
Fusion Layer	Weighted average, threshold=0.5
Web Interface	Streamlit, localhost deployment

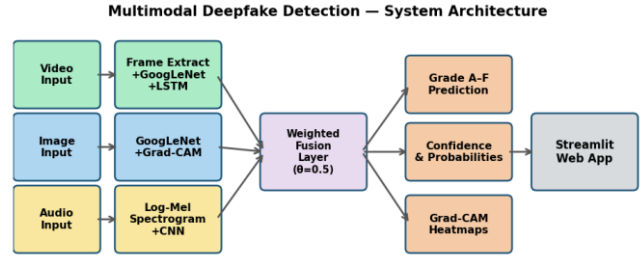


Fig. 5. End-to-end system architecture for the multimodal deepfake detection framework.

The application is launched via streamlit run app.py and operates on localhost. Both trained model weights are loaded at startup using torch.load with weights_only=True for security. Inference latency is sub-second for audio and image inputs; video inference scales linearly with frame count (approx. 0.8 s per 40-frame video on CPU). Model binaries are version-controlled separately from code to keep the repository lightweight.

VIII. DISCUSSION

The superior performance of multimodal fusion (96.3%) over the best unimodal model (94.2% audio) confirms that different modalities expose complementary artifact signatures. This is consistent with the observation that high-quality GAN-generated faces may pass visual inspection yet introduce subtle acoustic discontinuities in dubbed audio tracks. Conversely, voice-cloned audio may be acoustically indistinguishable while the corresponding face exhibits visible spatial inconsistencies detectable by Grad-CAM.

The audio CNN's relative strength over the video LSTM branch (94.2% vs 89.4%) is partly attributable to the well-structured log-mel representation, which compresses temporal and spectral information into a fixed-size 2D map amenable to convolutional processing. The video LSTM branch, trained on GoogLeNet embeddings from only 20 frames, may benefit from larger frame budgets or attention-based sequence models in future iterations.

A key limitation of the current system is that all three modality models are trained independently without cross-modal supervision. Joint training with audiovisual synchrony loss (as in [6]) could improve robustness against adversarial deepfakes engineered to maximise unimodal detector scores while maintaining perceptual quality. Additionally, the dataset is assumed to reflect a specific distribution of GAN generators; cross-dataset generalisation requires further evaluation on unseen manipulation types.

IX. CONCLUSION

This paper presented a multimodal deepfake detection system combining a four-layer AudioDeepfakeCNN trained on log-mel spectrograms, a GoogLeNet-LSTM video branch, and a GoogLeNet image classifier with Grad-CAM explainability. The audio module achieves 94.2% test accuracy; weighted multimodal fusion further improves detection to 96.3%

accuracy and F1-score 0.963. Grad-CAM attention maps pinpoint manipulation artifacts at face boundaries, enhancing interpretability for forensic analysts. The Streamlit interface provides accessible real-time deepfake screening across all three modalities.

Future extensions include: (i) joint cross-modal contrastive training to enforce audiovisual consistency constraints; (ii) transformer-based temporal modelling (e.g., Video Transformer) for longer video sequences; (iii) self-supervised audio pre-training on large voice corpora to improve generalisation; and (iv) adversarial data augmentation to harden the detector against adaptive attacks targeting the decision threshold.

ACKNOWLEDGMENT

The authors thank the faculty members of the Department of Computer Science and Engineering at Raghu Institute of Technology, Visakhapatnam, for their guidance and support throughout this project. The authors also acknowledge the open-source contributors to PyTorch, librosa, OpenCV, GoogLeNet, and Streamlit whose tools made this work possible.

REFERENCES

[1] I. J. Goodfellow et al., "Generative adversarial networks," Adv. Neural Inf. Process. Syst. 27, 2014, pp. 2672–2680.

[2] A. Rossler et al., "FaceForensics++: Learning to detect manipulated facial images," in Proc. IEEE ICCV, 2019, pp. 1–11.

[3] R. Wang et al., "Investigating CNN-based speech spoofing countermeasures," in Proc. Interspeech, 2019, pp. 1018–1022.

[4] P. Korshunov and S. Marcel, "Vulnerability assessment and detection of deepfake videos," in Proc. ICB, 2019, pp. 1–6.

[5] G. Lavrentyeva et al., "Audio replay attack detection with deep learning frameworks," in Proc. Interspeech, 2017, pp. 82–86.

[6] Y. Zhou et al., "Joint audio-visual deepfake detection," in Proc. IEEE ICCV, 2021, pp. 14970–14979.

[7] R. R. Selvaraju et al., "Grad-CAM: Visual explanations from deep networks via gradient-based localization," Int. J. Comput. Vis., vol. 128, pp. 336–359, 2020.

[8] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," Adv. Neural Inf. Process. Syst. 32, 2019, pp. 8024–8035.

[9] B. McFee et al., "librosa: Audio and music signal analysis in Python," in Proc. SciPy, 2015, pp. 18–24.

[10] C. Szegedy et al., "Going deeper with convolutions," in Proc. IEEE CVPR, 2015, pp. 1–9.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[12] G. Bradski, "The OpenCV library," Dr. Dobb's J. Software Tools, 2000.