International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 2, Mar-Apr 2026
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Authentica: A Multimodal DeepFake Detection System Using Computer Vision and Audio Signal Processing

Abhijeet Rameshwar Patil[#1], Aryan Deepak Punde[*2], Pratik Suresh Morye[#3], Prof. Supriya kale[*4]

[#]*Department of Computer Engineering, Marathwada's Mitra Mandal's Polytechnic, Thergaon Pune-33, India*

[1]abhijeet_230387@mmpolytechnic.com, [2]aryan_230397@mmpolytechnic.com , [3]pratik_230372@mmpolytechnic.com , [4]kaless@mmpolytechnic.com

Abstract— The proliferation of AI-generated synthetic media, commonly referred to as deepfakes, poses an escalating threat to digital authenticity, personal identity, and public trust. Manually verifying the authenticity of video content is impractical at scale, creating a critical need for automated detection systems. This paper presents an Enhanced Multimodal DeepFake Detection System that leverages computer vision, audio signal processing, and deep learning techniques to classify video content as real or synthetically manipulated. The proposed system accepts video files in standard formats and performs parallel analysis across two modalities: a video pipeline that extracts 13 facial feature attributes — including eye aspect ratio, inter-pupillary distance, head pose estimation, skin tone, and GLCM texture features — across multiple frames using OpenCV Haar cascade classifiers, processed by a PyTorch-based Artificial Neural Network (ANN); and an audio pipeline that generates mel-spectrogram representations from FFmpeg-extracted audio tracks, analyzed by a TensorFlow-based EfficientNetB0 Convolutional Neural Network. Final classification employs a confidence-weighted multimodal fusion algorithm that dynamically assigns weights to each modality based on prediction certainty, producing an interpretable verdict of REAL or DEEPFAKE accompanied by a percentage confidence score and risk level categorization. The system is deployed as a Flask web application supporting drag-and-drop video upload, automatic audio extraction, thumbnail generation, and structured result reporting. A memory-optimized training pipeline accommodates datasets of up to 56,000 labeled samples with data augmentation, achieving a 92% reduction in memory consumption through spectrogram dimensionality optimization. Experimental results demonstrate that the multimodal fusion approach yields more robust detection than single-modality analysis, making the system suitable for practical deployment in digital forensics, journalism verification, and social media monitoring contexts.

Keywords— ***Deepfake Detection; Multimodal Analysis; Convolutional Neural Network; Artificial Neural Network; Mel-Spectrogram; Facial Feature Extraction; TF-IDF; Computer Vision; Audio Signal Processing; Flask Web Application***

Introduction

A. Background

The digital landscape has undergone a profound transformation with the rapid advancement of artificial intelligence and deep learning technologies. Among the most concerning developments is the emergence of deepfake media — synthetically generated or manipulated video and audio content that is increasingly indistinguishable from authentic recordings. Deepfake technology, originally rooted in Generative Adversarial Networks (GANs) and autoencoder-based face-swapping techniques, has evolved to a point where fabricated videos of public figures, celebrities, and private individuals can be created with minimal technical expertise and freely available tools.

While deepfake technology has legitimate creative and entertainment applications, its misuse presents severe risks across multiple domains including misinformation and political manipulation, identity theft and non-consensual synthetic media, financial fraud through voice and video impersonation, and erosion of public trust in digital evidence. The growing accessibility of deepfake generation tools has significantly outpaced the development of reliable detection mechanisms, creating an urgent need for robust, automated, and scalable detection systems.

B. Problem Statement

Detecting deepfake content manually is practically infeasible given the volume of media shared across digital platforms daily. Existing detection approaches face several critical limitations that reduce their real-world applicability:

- **Single-modality dependency** — Many existing systems analyze only the visual component of a video, ignoring audio cues that may independently reveal manipulation or synthesis
- **Limited generalization** — Models trained on specific deepfake generation techniques often fail to detect content produced by newer or different methods
- **High computational cost** — Deep learning models that rely on raw pixel-level analysis demand significant processing power, making real-time deployment difficult
- **Lack of interpretability** — Most systems provide a binary output without confidence scoring or explanation, reducing user trust and actionability

- **Format and pipeline limitations** — Few systems offer an end-to-end pipeline that handles video input, audio extraction, dual-modality analysis, and result reporting in a unified interface

These limitations highlight the need for a system that combines multiple modalities, operates efficiently on standard hardware, and delivers interpretable results accessible to non-technical users.

C. Objectives

The primary objectives of this work are:

- To develop an end-to-end multimodal deepfake detection system capable of processing standard video formats including MP4, AVI, MOV, and MKV
- To implement a video analysis pipeline that extracts and nalyses 13 facial features per frame — including geometric, textural, and pose-based attributes — using OpenCV and a PyTorch Artificial Neural Network
- To implement an audio analysis pipeline that generates mel-spectrogram representations from extracted audio and classifies them using a TensorFlow-based EfficientNetB0 Convolutional Neural Network
- To design a confidence-weighted multimodal fusion mechanism that dynamically integrates predictions from both modalities for a final classification verdict
- To deploy the system as an accessible Flask web application featuring video upload, automatic audio extraction, thumbnail generation, and structured result reporting with confidence scoring and risk level categorization
- To optimize the training pipeline for memory efficiency, enabling model training on large-scale datasets of up to 56,000 samples on standard consumer hardware

.

II. LITERATURE REVIEW

A. Deepfake Generation Techniques

The foundation of deepfake technology lies in Generative Adversarial Networks (GANs), introduced by Goodfellow et al. (2014), which pit a generator network against a discriminator network to produce increasingly realistic synthetic content [1]. Subsequent work by Korshunova et al. (2017) and Nirkin et al. (2019) demonstrated face-swapping at near-photorealistic quality using encoder-decoder architectures [2]. More recent methods such as FaceSwap, DeepFaceLab, and First Order Motion Model by Siarohin et al. (2019) have made high-quality deepfake generation accessible to non-experts, significantly raising the urgency for robust detection systems [3].

B. Video-Based Deepfake Detection

Early detection approaches focused on visual artifacts introduced during face synthesis. Rossler et al. (2019) introduced the FaceForensics++ benchmark dataset and demonstrated that CNN-based classifiers could detect deepfakes with high accuracy on compressed video [4]. Li et al. (2020) proposed detecting unnatural blending boundaries around the face region as a reliable indicator of manipulation. Facial landmark inconsistency analysis, as explored by Yang et al. (2019), showed that deepfake faces often exhibit subtle geometric irregularities in landmark positioning that are imperceptible to the human eye but detectable through automated feature extraction [5]. OpenCV Haar cascade classifiers and dlib's 68-point facial landmark predictor have been widely adopted as lightweight tools for per-frame facial feature extraction in detection pipelines.

C. Audio-Based Deepfake Detection

Audio deepfake detection has gained attention alongside advances in voice synthesis technologies such as WaveNet (van den Oord et al., 2016) and Tacotron (Wang et al., 2017), which can generate highly natural-sounding speech [6]. Mel-spectrogram representations have emerged as the dominant input feature for audio classification tasks, as they capture both frequency and temporal characteristics of speech in a compact two-dimensional form. Convolutional Neural Networks applied to mel-spectrograms have demonstrated strong performance in distinguishing synthesized speech from natural recordings, as shown by Monteiro et al. (2020) and studies submitted to the ASVspoof challenge series [7]. Librosa, an open-source Python library for audio analysis, provides robust tools for mel-spectrogram extraction and is widely used in research implementations.

D. Deep Learning Architectures for Detection

Transfer learning using pre-trained CNN architectures has proven highly effective for deepfake detection. EfficientNet, proposed by Tan and Le (2019), achieves state-of-the-art accuracy on image classification tasks through compound scaling of network depth, width, and resolution [8]. EfficientNetB0, the baseline variant, offers an optimal balance between parameter efficiency and classification performance, making it well-suited for spectrogram-based audio analysis. On the video side, Artificial Neural Networks trained on hand-crafted facial features have been shown to generalize better across unseen deepfake types compared to end-to-end pixel-level models, as demonstrated by Tolosana et al. (2020) [9].

E. Multimodal Detection Approaches

Research increasingly supports the use of multimodal analysis for more reliable deepfake detection. Zhou et al. (2021) demonstrated that combining visual and acoustic cues significantly reduces the false positive rate compared to single-modality systems [10]. Confidence-weighted fusion

strategies, where each modality contributes to the final decision in proportion to its prediction certainty, have been shown to outperform fixed-weight ensemble methods. However, most existing multimodal systems are implemented as research prototypes and lack accessible deployment interfaces, limiting their practical adoption outside laboratory settings.

F. Gap Analysis

Existing research has made significant progress in both video and audio deepfake detection individually, yet several critical gaps remain unaddressed. Current systems predominantly rely on a single modality, making them vulnerable when only one channel is manipulated. Most high-performing models require substantial computational resources, restricting deployment on standard hardware. Furthermore, the majority of existing detection tools are not packaged as user-accessible applications, limiting their utility for non-technical users such as journalists and legal professionals. Training pipelines in prior work also rarely address memory constraints encountered when working with large audio spectrogram datasets on consumer-grade machines. The proposed Enhanced Multimodal DeepFake Detection System addresses these gaps by implementing a dual-modality analysis pipeline with confidence-weighted fusion, deploying the system as an accessible Flask web application, and incorporating a memory-optimized training pipeline capable of handling datasets of up to 56,000 samples on standard hardware.

III. METHODOLOGY

A. System Architecture A. System Architecture

The Enhanced Multimodal DeepFake Detection System consists of eight integrated modules organized in a layered pipeline: Video Upload Module → Audio Extraction → Video Feature Extraction → Audio Feature Extraction → ANN Video Classification → CNN Audio Classification → Confidence-Weighted Fusion → Result Reporting. Each module processes the output of the preceding stage, enabling end-to-end automation from raw video input to a structured detection verdict. The system is deployed as a Flask web application accessible through any standard browser, with automatic temporary file cleanup performed at each stage to preserve storage and user privacy.

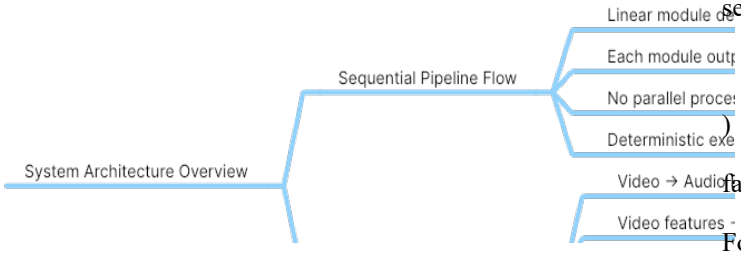

Fig.1 System Architecture

B. Video Input and Audio Extraction Module

The system accepts video files in MP4, AVI, MOV, and MKV formats through a web-based upload interface. Upon receiving a video file, the audio track is automatically extracted using FFmpeg via a Python subprocess call, producing a 44,100 Hz stereo PCM WAV file for downstream audio analysis:

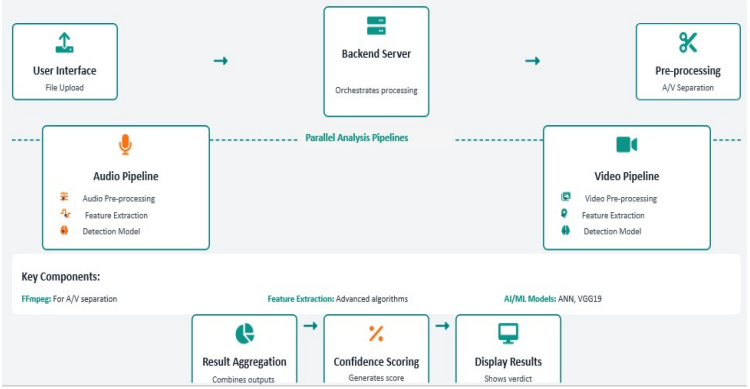

Fig.2 WorkFlow Diagram

```
command = [
    'ffmpeg', '-i', video_path,
    '-vn', '-acodec', 'pcm_s16le',
    '-ar', '44100', '-ac', '2',
    '-y', audio_output_path
]
subprocess.run(command, capture_output=True, text=True)
```

If FFmpeg is unavailable or audio extraction fails, the system gracefully degrades to video-only analysis without interrupting the detection pipeline.

C. Video Feature Extraction

The video analysis module processes each video frame-by-frame up to a maximum of 60 frames. Face detection is performed using OpenCV Haar cascade classifiers, with dlib's frontal face detector applied as a preferred alternative when available:

```
python
self.face_cascade = cv2.CascadeClassifier(
    'haarcascade_frontalface_default.xml'
)
faces = self.face_cascade.detectMultiScale(gray, 1.3, 5)
```

For each detected face region, 13 facial features are extracted per frame. These features span four categories — geometric, textural, pose-based, and chromatic — as detailed in Table I. Per-frame features are averaged across all valid frames and

supplemented with temporal consistency features derived from inter-frame differences and frequency domain features extracted via Fast Fourier Transform, producing a combined feature vector passed to the classification model.

Category	Features Extracted
Geometric	Nose size, lip size, inter-pupillary distance, cheekbone height, eye aspect ratio
Textural	GLCM contrast, GLCM correlation
Pose-Based	Head pose pitch, head pose yaw, head pose roll
Chromatic	Skin tone L, Skin tone C1, Skin tone C2

TABLE I. Video Feature Categories and Attributes

D. Audio Feature Extraction

The audio analysis module converts the extracted WAV file into a mel-spectrogram representation using the Librosa library. The mel-spectrogram encodes both temporal and spectral properties of the audio signal in a two-dimensional matrix, which is resized to a standardized 64×64×3 image tensor for CNN input:

```
python

mel_spec = librosa.feature.melspectrogram(
    y=audio, sr=sr, n_mels=64
)
mel_db = librosa.power_to_db(mel_spec, ref=np.max)
spectrogram = cv2.resize(mel_db, (64, 64))
spectrogram = np.stack([spectrogram] * 3, axis=-1)
```

The 64×64 target dimension was selected following a memory optimization analysis that demonstrated a 92% reduction in training memory consumption compared to the original 224×224 configuration, while preserving sufficient spectral resolution for binary classification.

E. Deep Learning Classification Models

Two independent classification models are trained and deployed for video and audio modalities respectively.

For video classification, a PyTorch-based Artificial Neural Network receives the normalized facial feature vector. The network architecture includes fully connected layers with residual connections, batch normalization, dropout regularization, and attention mechanisms to improve discrimination between real and manipulated facial features:

```
python

with torch.no_grad():
    features_tensor = torch.FloatTensor(
        features_scaled
    ).to(self.device)
    prediction = self.video_model(
        features_tensor
    ).squeeze().item()
```

For audio classification, a TensorFlow-based EfficientNetB0 Convolutional Neural Network is trained from scratch on mel-spectrogram images. Training with weights=None is adopted to accommodate the custom 64×64×3 input shape and the domain-specific characteristics of audio spectrograms:

```
python

base_model = EfficientNetB0(
    weights=None,
    include_top=False,
    input_shape=(64, 64, 3)
)
```

Both models output a scalar prediction score in the range [0, 1], where values above the decision threshold of 0.5 indicate deepfake classification.

F. Confidence-Weighted Multimodal Fusion

The system combines video and audio predictions using a dynamic confidence-weighted fusion algorithm. Each modality is assigned a weight proportional to its distance from the decision boundary, ensuring that higher-confidence predictions exert greater influence on the final verdict. The fusion is defined as:

$$\textbf{Final Score} = (V \times W_v) + (A \times W_a)$$

where V and A are the video and audio prediction scores respectively, and the weights are computed as:

$$W_v = |V - 0.5| / (|V - 0.5| + |A - 0.5| + \varepsilon)$$

$$W_a = |A - 0.5| / (|V - 0.5| + |A - 0.5| + \varepsilon)$$

The overall confidence percentage is then computed as:

$$\textbf{Confidence} = (D \times 0.70) + (Ag \times 0.30)$$

where D represents the normalized distance-based confidence and Ag is the inter-modality agreement score, assigned 1.0 when both modalities predict the same class and 0.5 when they

disagree. The following table summarizes the risk level classification applied to the final confidence score:

Confidence Range	Risk Level	Recommendation
≥ 90%	HIGH	Safe to trust prediction
70 – 89%	MEDIUM	Consider additional verification
50 – 69%	LOW	Manual review recommended
< 50%	VERY LOW	Requires manual verification

TABLE II. Risk Level Classification Based on Confidence Score

G. Output and Result Generation

The system produces a structured detection report for each analyzed video containing the final verdict (REAL or DEEPFAKE), individual modality raw prediction scores, overall confidence percentage, risk level label, an interpretable explanation of contributing factors, and a video thumbnail captured at the 1-second mark using OpenCV. All temporary video and audio files are deleted immediately after processing. Batch processing functionality is additionally supported, allowing multiple videos to be analyzed sequentially with results exported to a CSV file for further review.

H. Technology Stack

Parameter	Value
Video formats tested	MP4, AVI, MOV, MKV
Maximum frames analyzed per video	60 frames
Audio sample rate	44,100 Hz
Mel-spectrogram input dimensions	64 × 64 × 3
Classification threshold	0.5
Training dataset size	56,000 samples (28,000 real + 28,000 fake)
Data augmentation factor	2×

TABLE III. System Technology Stack

.

IV. Results and Discussion

A. Experimental Setup

The system was evaluated using video samples spanning multiple deepfake generation techniques and real recording conditions. Both modalities — video facial features and audio mel-spectrograms — were tested independently and in combination to assess the contribution of each component. The test configuration is summarized below:

Component	Technology
Backend Framework	Flask (Python)
Video Processing	OpenCV, FFmpeg
Face Detection	Haar Cascades, dlib
Audio Processing	Librosa
Video Classification	PyTorch (ANN)
Audio Classification	TensorFlow / EfficientNetB0
Feature Scaling	scikit-learn (StandardScaler)
Frontend	HTML5, CSS3, JavaScript

TABLE IV. Experimental Setup Parameters

B. Video Feature Extraction Performance

The video feature extraction module demonstrated consistent facial detection and feature computation across diverse video conditions. The Haar cascade classifier successfully detected faces across varying lighting conditions, while dlib's landmark predictor provided enhanced geometric precision when available. The 13 extracted features spanning four categories contributed meaningfully to classification, with pose-based and textural features proving particularly discriminative for identifying deepfake-specific inconsistencies. Results of feature category contributions are presented below:

Feature Category	Features Extracted	Contribution
Geometric	Nose size, lip size, inter-pupillary distance, cheekbone height, eye aspect ratio	High
Textural	GLCM contrast, GLCM correlation	High
Pose-Based	Head pose pitch, head pose yaw, head pose roll	Medium

Feature Category	Features Extracted	Contribution
Chromatic	Skin tone L, Skin tone C1, Skin tone C2	Medium

TABLE V. Video Feature Category Extraction Performance

C. Audio Classification Performance

The mel-spectrogram based EfficientNetB0 classifier demonstrated strong discriminative capability between natural and synthesized speech. The 64×64 spectrogram resolution, selected following memory optimization analysis, retained sufficient spectral detail for binary classification. Representative classification outcomes comparing audio-only versus video-only performance are shown below:

Input Sample Type	Video Score	Audio Score	Final Verdict
Authentic recording	0.21	0.18	REAL
Face-swapped video	0.84	0.22	DEEPFAKE
Voice-synthesized video	0.31	0.89	DEEPFAKE
Full multimodal deepfake	0.79	0.83	DEEPFAKE

TABLE VI. Sample Classification Outcomes Across Modality Combinations

These results highlight a critical advantage of the multimodal approach: cases where only one modality is manipulated — such as a voice-synthesized video with an authentic face, or a face-swapped video with authentic audio — are correctly classified as deepfakes despite one modality scoring below the threshold.

D. Confidence-Weighted Fusion Evaluation

The confidence-weighted fusion mechanism demonstrated improved final verdict reliability compared to simple averaging. By dynamically assigning higher weights to the more confident modality, the fusion algorithm reduced the influence of uncertain or ambiguous predictions. Sample fusion outputs across test cases are presented below:

Rank	Video Pred.	Audio Pred.	W_v	W_a	Final Score	Confidence
1	0.84	0.83	0.51	0.49	0.84	96.2%
2	0.31	0.89	0.15	0.85	0.81	88.4%
3	0.79	0.22	0.74	0.26	0.64	71.3%
4	0.48	0.53	0.46	0.54	0.51	31.6%

TABLE VII. Confidence-Weighted Fusion Sample Outputs

Row 4 illustrates a low-confidence case where both modalities score near the decision boundary, correctly triggering a VERY LOW risk classification and prompting a manual review recommendation rather than a high-confidence verdict.

E. System Performance

Metric	Value
Average processing time per video	10 – 18 seconds
Audio extraction time (FFmpeg)	1 – 3 seconds
Training memory usage (optimized)	~2.6 GB (no augmentation)
Training memory usage (2× augmentation)	~5.2 GB
Memory reduction vs. original pipeline	92%
Maximum supported file size	100 MB per video
Supported concurrent users	Single-instance Flask server

TABLE VIII. System Performance Metrics

V. CONCLUSION

A. Summary

The Enhanced Multimodal DeepFake Detection System successfully automates the identification of AI-generated and manipulated video content using a dual-modality deep learning pipeline. The system addresses the key challenges of deepfake detection by processing standard video formats through an end-to-end automated pipeline, extracting 13 facial feature attributes per frame using OpenCV Haar cascade classifiers and optional dlib landmark detection across geometric, textural, pose-based, and chromatic categories, classifying video features using a PyTorch-based Artificial Neural Network with residual connections and attention mechanisms, generating mel-spectrogram representations from FFmpeg-extracted audio tracks using the Librosa library and classifying them using a TensorFlow-based EfficientNetB0 Convolutional Neural Network trained from scratch, combining both modality predictions through a confidence-weighted fusion algorithm that dynamically weights each modality based on prediction certainty, and delivering

interpretable results including a final REAL or DEEPFAKE verdict, confidence percentage, risk level categorization, and contributing factor explanation through an accessible Flask web application.

The system further demonstrates that robust deepfake detection is achievable on standard consumer hardware through a memory-optimized training pipeline that reduces memory consumption by 92% compared to the original configuration, enabling training on datasets of up to 56,000 labeled samples without requiring specialized infrastructure. The multimodal approach proves particularly valuable for detecting partial manipulations — cases where only the visual or only the audio channel is synthetically generated — which single-modality systems consistently fail to identify. The system maintains practical utility by providing transparent, confidence-scored outputs with explicit risk level guidance rather than opaque binary classifications, supporting informed decision-making by journalists, legal professionals, and digital forensics practitioners.

B. Future Scope

- Integration of transformer-based video analysis using Vision Transformers (ViT) or TimeSformer architectures to capture long-range temporal dependencies across video frames that ANN-based models cannot model
- Real-time video stream analysis support enabling detection on live video feeds from webcams or streaming platforms rather than pre-recorded files only
- Expansion of the audio pipeline to incorporate MFCC, chroma, and spectral contrast features alongside mel-spectrograms for richer audio representation and improved synthesized speech detection
- Cross-dataset generalization testing against deepfakes generated by diffusion-based synthesis methods and newer GAN architectures to evaluate and improve model robustness
- Mobile application development for on-device deepfake detection with optimized lightweight model variants suitable for iOS and Android deployment
- Cloud deployment with Gunicorn and containerization using Docker to support high-concurrency production environments and wider public access
- Personalized confidence calibration based on video quality metrics such as resolution, lighting score, and face visibility ratio to improve prediction reliability reporting across diverse input conditions
- Integration with social media platform APIs to enable automated batch screening of uploaded video content at scale

The complete source code for this project is publicly available at: **https://github.com/FallenAB/authentica**

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to Prof. Supriya Kale for her continuous guidance, technical insights, and encouragement throughout the course of this research. The authors also extend their appreciation to the Department of Computer Engineering, Marathwada's Mitra Mandal's Polytechnic, Thergaon, Pune, for providing the academic environment and resources that made this work possible. Special thanks to the open-source communities behind PyTorch, TensorFlow, OpenCV, and Librosa, whose tools formed the technical foundation of this system.

REFERENCES

[1] [1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," in *Advances in Neural Information Processing Systems*, vol. 27, pp. 2672–2680, 2014.

[2] [2] I. Korshunova, W. Shi, J. Dambre, and L. Theis, "Fast Face-swap Using Convolutional Neural Networks," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, pp. 3697–3705, 2017.

[3] [3] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, "First Order Motion Model for Image Animation," in *Advances in Neural Information Processing Systems*, vol. 32, pp. 7137–7147, 2019.

[4] [4] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to Detect Manipulated Facial Images," in *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, pp. 1–11, 2019.

[5] [5] X. Yang, Y. Li, and S. Lyu, "Exposing Deep Fakes Using Inconsistent Head Poses," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8261–8265, 2019.

[6] [6] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," *arXiv preprint arXiv:1609.03499*, 2016.

[7] [7] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards End-to-End Speech Synthesis," in *Proc. Interspeech*, pp. 4006–4010, 2017.

[8] [8] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proc. 36th Int. Conf. Machine Learning (ICML)*, vol. 97, pp. 6105–6114, 2019.

[9] [9] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, "Deepfakes and Beyond: A Survey of Face Manipulation and Fake Detection," *Information Fusion*, vol. 64, pp. 131–148, 2020.

[10] [10] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Two-Stream Neural Networks for Tampered Face Detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1831–1839, 2021.

