DeepfakeGuard: An Open-Source Multi-Modal Toolkit for Detecting AI-Generated Video and Audio

Aryan Biswas, Kevin Hu, Eitan Zur, Muhammad Hamza Khan, Emmanuel Davidson
Queen's University, School of Computing
Kingston, Ontario, Canada
21ab43@queensu.ca, 20ch35@queensu.ca, 22TLG3@queensu.ca, 23qj44@queensu.ca, 20eobd@queensu.ca

***Abstract*—AI-generated videos and audio are now advanced enough to be used for large-scale fraud, harassment, and political manipulation; however, few tools exist to detect such uses of AI. We introduce DeepfakeGuard, a Python-based detection tool that uses three complementary detection methods that can be deployed as a single toolset. The first method utilizes a parameter-efficient DINOv3 Vision Transformer [15] to identify visual artifacts; the second method detects audio-visual lip-sync inconsistencies using a modified version of LipFD [17]; and the third method identifies AI-generated videos based on suppressed second-order motion dynamics using a training-free D3 model [18]. An ensemble of the output from each of these detection modules uses trust-weighted scoring and applicability gating to produce a single confidence score. An additional Vision-Language Model module receives structured context about each detector's known strengths and failure modes, and uses this to produce an informed forensic judgment that compensates for cases where the ensemble's quantitative scoring is insufficiently reliable. Each individual component was tested separately on existing benchmark datasets. Our DINOv3 detector achieved an Area Under the Receiver Operating Characteristic Curve (AUROC) of 0.88 on the Celeb-DF v2 dataset when testing strictly across all available datasets, and LipFD achieved an accuracy rate of 91.2% when detecting lip-synced manipulations in FakeAVCeleb. In addition to assessing the effectiveness of each individual component, we have also explored the ethical implications of releasing open-source forensic tooling, including potential dual-use risks, the potential for demographic bias within training data, and the necessity of incorporating human-in-the-loop oversight. Finally, we assert that open, multi-modal detection infrastructure is a necessary and currently underdeveloped counterbalance to the rapid democratization of generative AI.**

I. Introduction

Deepfakes are causing real-world harm. Today, humans can rarely tell the difference between real and AI-generated videos, and hundreds of thousands of these fakes are already spreading online [1]. The damage is clear. Scammers recently used synthetic video to pull off a $25 million impersonation fraud [3]. Others use it to mass-produce non-consensual sexual images [2] or spread election misinformation [1]. Tools like DeepFaceLab [5] and voice-cloning APIs [4] make it incredibly easy to create manipulated media. But our defences have not kept up.

The main reason is how we build these detectors. Most researchers train and test their models on just one specific type of fake or modality. This leads to great benchmark scores in the lab, but the models often fail in real-world use cases [28]. For example, a face-swap detector will miss a lip-sync fake, while a purely visual detector will miss temporal glitches or physics-based anomalies. Furthermore, if a model relies heavily on labelled data, it has to be retrained every time a new video generator comes out. Even laws like the EU AI Act [26] require transparency for AI content without explaining exactly how to detect it. Right now, there is no easy-to-use or accessible toolkit available for developers and the general public to check for all these different threats at once.

We propose DeepfakeGuard to help bridge this gap. It is an open-source, pip-installable Python library that combines three different detection methods into a single tool. We do not claim it is a perfect, final solution. Instead, it is a starting point that researchers can easily expand upon as AI-generated content and digital impersonation continue to evolve.

A. Contributions

To make DeepfakeGuard work, we had to solve several problems across both the models and the system architecture. Our main contribution is the toolkit itself, a modular framework that allows researchers to plug in new detection models and techniques as the field advances. At its core, it puts visual, audio-visual, and physics-based temporal detection behind a simple two-function API (`detect_video` and `ensemble_detect_video`), turning fragmented research code into a tool anyone can run in three lines of Python.

Under the hood, we built a highly efficient DINOv3 visual detector. By tuning just 8.3% of a ViT-B/16 backbone, we reached an AUROC of 0.88 on the Celeb-DF v2 dataset, easily beating previous techniques such as Xception under the same testing conditions (Section IV). For our audio-visual and temporal checks, we integrated LipFD and D3. We rebuilt the LipFD pipeline from scratch to resolve hidden preprocessing bugs in the original research code (Section V), and we implemented a full end-to-end version of the D3 temporal detector, which requires no AI-generated training data whatsoever (Section VI).

In our multimodal framework, each detector captures a different aspect of manipulation, which is why we designed an ensemble fusion layer to combine their results intelligently. Our ensemble logic ensures that a confused detector — for example, LipFD flagging a video with no visible lips — does not compromise the final verdict. We also integrated a Vision–Language Model (VLM) layer that samples six keyframes and prompts the model with each detector's confidence scores,

generating a human-readable explanation for cases where the ensemble alone is inconclusive (Sections VII and VIII). Finally, we discuss the ethical implications of releasing deepfake detectors, including the potential for demographic bias in training data and why human oversight must remain central to any deployment (Section IX).

II. Background and Other Work

A. Evolution of Deepfake Generation Techniques

Techniques for generating deepfakes have been evolving rapidly in both visual and audio spaces. Examples of face-swap based methods are DeepFaceLab [5] where one person's image is transplanted onto another's image, and face-reenactment based methods like Face2Face [6] that maps a person's facial expressions onto another video while their identity remains intact. Even more recent are diffusion-based video generators like OpenAI's Sora [27] that allow anyone who uses them to create photorealistic videos from a text prompt, gradually increasing the exposure of AI videos to the public. Audio generators such as Wav2Lip [23] will re-animate a speaker's lips to exactly match a target audio signal, allowing for highly realistic impersonations, without affecting the rest of the face. On the other hand, text-to-speech systems such as VALL-E [4] can clone an individual's voice from just a few seconds of reference audio, thereby breaking through speaker verification systems. Because there are many different ways to generate and manipulate media (face-swap, face-reenactment, lip-sync, full synthesis, voice-cloning), it is unrealistic to expect a single detection method to detect all potential threats. Therefore, DeepfakeGuard was designed to be multi-modal to protect against multiple forms of manipulation.

B. Methods for Detecting Visual Deepfakes

There are primarily four categories of visual detection techniques: artifact-based methods [7], temporal methods [8], biological signal methods [9], and representation learning methods that train discriminative features end-to-end. Among representation-learning methods, two well-established methods include Xception [10] and EfficientNet [11] that achieved high AUROCs on the FaceForensics++ dataset. However, they experienced a decrease in AUROC of 0.05–0.20 when these methods were tested on unseen datasets [12], [28]. These results suggest that large pre-trained representations may contain more generalizable forensic signals than task-specific architectures. Self-supervised Vision Transformers such as DINOv2 [13] tend to focus on low-frequency structural characteristics instead of artifacts produced by specific generators [16]. In addition, the use of register tokens [14] improved the quality of features. Our DINOv3 detector builds on this direction, combining parameter-efficient LayerNorm tuning of a DINOv3 ViT-B/16 [15] backbone with angular-margin metric learning to maximize cross-dataset transfer.

C. Audio-Visual Lip-Sync and Temporal Detection

A complementary detection technique relies on the strong correlation of speech with lip movement. If deepfakes have been applied to the face or if the lips have been poorly re-animated, then there will be small but measurable audio-visual mismatches that are invisible to the human eye but present in feature space. LipFD [17] formalized this by taking a frozen CLIP ViT-L/14 encoder [24] as a global audio-visual feature extractor and a Region-Awareness ResNet-50 backbone [25] that is focused on multi-scale lip crops to capture the temporal inconsistencies of such an approach that other purely visual approaches miss.

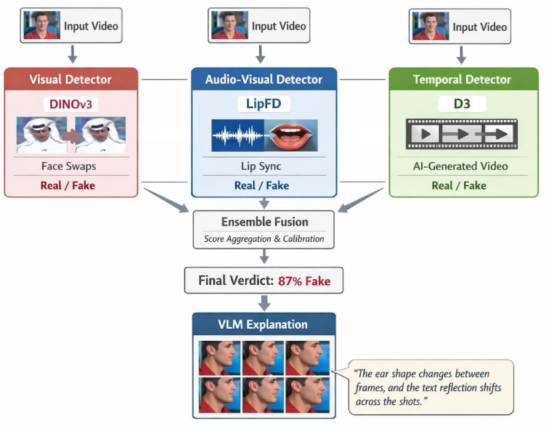

Fig. 1. DeepfakeGuard system architecture. Three complementary detectors process each input video in parallel and feed into a domain-aware ensemble fusion module, producing a final verdict alongside an optional VLM forensic explanation.

However, supervised detectors, including LipFD, need labelled data for each new generator family. In contrast, the D3 (Detection by Difference of Differences) method [18] detects the differences of differences between real and artificial video; that is, real video has a large amount of second-order temporal variance, which represents natural Newtonian motion. Conversely, artificially generated video has a much lower second order temporal variance. Therefore, because D3 does not use fine-tuned encoders, but instead uses pre-trained encoders to measure these properties, it can easily generalize to unseen generators and does not require any training data at all.

III. System Overview

The primary intent of this project was to develop a library that could be utilized by both developers and researchers. The library is centered on a single `DeepfakeGuard` orchestrator class that manages the loading of the detectors, their weights, and the combination of results from each detector. Each detector uses a simple, shared interface that returns a score between 0 and 1 (where 1 represents a high likelihood that the video is fake), a binary label, and a diagnostics dictionary. The public API exposes two functions, `detect_video(path)` for running a single detector on a video file, and `ensemble_detect_video(guards, path)` for running all three detectors and producing a fused verdict, with an optional VLM explainability pass.

IV. DINOv3 Visual Detector

A. Motivation and Pipeline

CNN-based forensic detectors tend to over-rely on cues that are specific to the dataset or generator they were trained on, which leads to weaker performance on unseen datasets. To improve cross-dataset robustness, we use self-supervised DINOv3 Vision Transformer features [15], which focus on generalizable structural patterns rather than generator-specific artifacts. The pipeline works in three stages. First, faces are extracted using MTCNN [19] with 30% extra padding and a vertical shift to emphasize the lower-face and jawline boundaries, capturing critical blending artifacts. The cropped frames are then encoded through a DINOv3 ViT-B/16 backbone that is partially fine-tuned using LayerNorm tuning. Finally, a lightweight classification head maps the embeddings to real and fake scores using a combination of focal loss and metric learning.

B. DINOv3 Encoder with LayerNorm Tuning

The backbone is a DINOv3 ViT-B/16 encoder that produces 768-dimensional embeddings. Rather than fine-tuning all 86M parameters, which risks overfitting to training-set artifacts, we fine-tune the model in three stages. We first freeze all encoder parameters, then unfreeze only the LayerNorm layers across all transformer blocks (roughly 40K parameters), and finally unfreeze the last transformer block, bringing the total to 7.1M trainable parameters (8.3%). This lets the model adapt its internal normalization and final representations to the forensic task while preserving the bulk of its pretrained visual knowledge.

C. Classification Head and Metric Learning

A single linear layer maps the encoder's 768-dimensional output to real and fake class scores. To prevent the model from relying on background shortcuts, we use paired sampling (real and fake faces from the same source video in each batch). The model is trained with Focal Loss ($\gamma = 2.0$) to prioritize hard-to-classify examples, plus an angular-margin term ((1)) that pushes the learned representations of real and fake faces further apart:

$$\mathcal{L} = \mathcal{L}_{FL} + \lambda \cdot \frac{1}{|P|} \sum_{(i,j)\in P} \max\big(0,\, \cos(e_i, e_j) + m\big) \qquad (1)$$

where P is the set of real–fake pairs in a batch, $\cos(e_i, e_j)$ measures embedding similarity, $m = 0.6$ is a margin enforcing separation, and $\lambda = 0.5$. Training also applies stochastic JPEG compression, Gaussian blur, and standard color and geometric augmentations to simulate social-media degradation and prevent overfitting to pristine training data. Together, these encourage a robust decision boundary that transfers to unseen datasets.

TABLE I
Cross-dataset detection on Celeb-DF v2 (trained on FF++).

Method	AUROC
MesoInception4 [21]	0.536
Xception-c23 [10]	0.653
Face X-ray [20]	0.741
DINOv3 (Ours)	**0.880**

D. Experiments

Models are trained purely on FaceForensics++ [10] and evaluated on Celeb-DF v2 [12] without any fine-tuning. Table I shows that our DINOv3 detector achieves an AUROC of 0.88, outperforming Xception-c23 by 23 points and Face X-ray by 14 points under the same cross-dataset protocol. This confirms that self-supervised ViT features generalize substantially better than CNN-based baselines when evaluated on unseen datasets.

V. LipFD Audio-Visual Detector

A. Motivation and Architecture

Lip-sync manipulations target the relationship between speech and lip movements. In this type of deepfake, the face may appear visually plausible and the spatial stream on individual frames alone offers few clues. To close this gap, we integrate the LipFD detector [17], which exploits temporal inconsistency between audio signals and visual lip dynamics through a dual-pathway architecture that fuses global audio-visual context with fine-grained spatial attention over the lip region. In the global audio-visual stream, a composite image is formed by vertically stacking a mel-spectrogram slice above five consecutive video frames, resized and downsampled before being passed through a frozen CLIP ViT-L/14 encoder [24] to produce a 768-dimensional global feature vector. In the region-awareness stream, a ResNet-50 backbone [25] independently encodes three progressively zoomed centre crops of each frame, and their features are fused with the global vector via a learned attention mechanism that emphasizes the most informative scale, typically the lip region.

A linear classifier maps this fused representation to a binary logit. Training uses cross-entropy loss augmented by a Region-Awareness (RA) loss:

$$\mathcal{L}_{\text{RA}} = \frac{1}{B} \sum_{b=1}^{B} \sum_{i=1}^{F} \frac{10}{\exp\big(\alpha_{\max}^{(b,i)} - \alpha_0^{(b,i)}\big)} \qquad (2)$$

where $\alpha_{\max}^{(b,i)}$ is the maximum attention weight across scales for frame i of sample b, and $\alpha_0^{(b,i)}$ is the weight for the full-frame crop. This term penalizes diffuse attention distributions and encourages the model to concentrate on the most diagnostically useful regions.

B. Integration and Implementation

The original LipFD codebase is structured as an offline preprocessing and training pipeline mainly for research purposes, which is unsuitable for a general-purpose inference tool over arbitrary videos. We therefore preserve the model architecture while re-engineering the surrounding infrastructure

into a runtime video-to-tensor pipeline. Ten groups of five consecutive frames are uniformly sampled from the video via OpenCV using frame-level seeking, while the audio track is extracted to a temporary WAV file and converted to a mel-spectrogram using librosa, with a grey placeholder substituted when no audio is present. Each frame group is assembled into a composite image with the spectrogram stacked above the frames, converted to BGR to match the training distribution, and transformed into multi-scale crops that are resized to 224×224 for the ResNet backbone. Standard CLIP normalization was not used, since the original training pipeline effectively overwrote it and the model was therefore trained on raw BGR pixel values in $[0, 255]$. The original crop extraction pattern was also preserved, since the original code used small horizontal shifts rather than 500-pixel jumps to separate the five frames, and replacing this pattern led to a noticeable drop in accuracy. We tested both details on three held-out reference clips and confirmed that our implementation produced the same raw model outputs as the original pretrained LipFD model, up to floating-point tolerance.

TABLE II
LIPFD DETECTION ON FAKEAVCELEB V1.2 (250 REAL + 250 FAKE PER ROW).

Manipulation	Acc.	AUROC	Prec.	Rec.	F1
FakeVid + FakeAud	**91.2%**	**0.962**	88.4%	94.8%	0.915
FakeVid + RealAud	75.2%	0.821	83.5%	62.8%	0.717
RealVid + FakeAud	49.6%	0.513	48.3%	11.6%	0.187

C. Experiments

We evaluate on FakeAVCeleb v1.2 [22], a multimodal benchmark containing real and fake videos across three manipulation types. Table II shows that LipFD achieves 91.2% accuracy and 0.962 AUROC on its target domain (lip-synced video with synthesized audio). Performance degrades on face-swap videos (75.2%) and drops to chance on audio-only fakes (49.6%), confirming that its discriminative power is primarily visual, though guided by audio-visual context. LipFD performs best precisely where DINOv3 and D3 show uncertainty, which supports the need for the ensemble approach described in Section VII.

VI. D3 Training-Free Temporal Detector

A. Motivation and Feature Extraction

In order to train supervised detectors that can detect synthetic videos, large amounts of labeled synthetic video need to be generated. Therefore, since it is impractical to label all possible synthetic videos generated by all possible generators, the D3 approach [18] utilizes an invariant derived from physical principles. Real videos contain random motions at the level of second-order accelerations due to the chaotic nature of Newtonian mechanics, while the motion of synthetic videos has artificially smoothed out second-order accelerations.

In our implementation, we approximate these dynamics using scalar distances between frame embeddings: second-order differences are computed by first calculating the scalar distance between consecutive frame embeddings in a pre-trained xCLIP-B/16 embedding space ($d_t = \|f_t - f_{t-1}\|$), and then taking the difference of consecutive distances:

$$f_t'' = d_{t+1} - d_t \tag{3}$$

Since the standard deviation of $\{f_t''\}$ should be much larger for natural video than for synthetic video, it is used as a measure of "volatility" which is then compared against a threshold τ that was set using only real video data, independent of any use of or access to synthetic video labels.

TABLE III
D3 DETECTION PERFORMANCE BY GENERATOR.

Generator	Avg. Volatility σ	AP
Pika	1.42	89.76%
Sora	6.81	$\approx 65\%$

B. Classification and Implementation

There is no training loop required. All processing is done in Python. First, each input video is reduced down to 32 frames. Then, each frame is embedded into a high-dimensional semantic space using the xCLIP-B/16 encoder. Finally, the second-order difference statistic for each frame is calculated, and the standard deviation of this statistic across the entire 32-frame video is returned as a single number representing the "volatility" of the original video.

The D3 detector achieved 89.76% AP on Pika-generated videos and approximately 65% on Sora-generated videos. Sora's higher measured volatility (6.81 vs. 1.42 for Pika) suggests its motion more closely resembles real-world physical complexity, making second-order volatility cues less discriminative.

VII. Ensemble Detection System

The three detectors are excellent for their respective types of forgeries; however, they are significantly degraded outside of their areas of expertise. Therefore, a simple majority vote among the detectors does not suffice. Two or more detectors which are not applicable to the detection task can potentially outvote the single applicable detector. In order to address this limitation, a weighted fusion method was developed using domain-aware weights. Before performing the weighted fusion operation, each detector is given an applicability factor $a_i \in [0, 1]$ based on input-specific heuristics developed from the detector's own diagnostic output. For example, the DINOv3 detector receives full weight when there are at least eight frames containing faces. The D3 detector down-weights clips having minimal motion based on the ratio of the clip's measured volatility and the decision threshold. The LipFD detector assigns high weight to video segments containing a large number of extractable audio-visual lip-sync samples. If a video segment contains many lip regions and has valid audio, a higher weight is assigned. Conversely, if the video segment is missing lip regions or valid audio, the applicability factor is set to zero. Additionally, if one detector produces a score greater than 0.80 while the remaining detectors have a score less than 0.35, the

unique detector is identified as a potential domain mismatch and the weight used during the fusion operation is reduced to 0.05. This prevents the unique detector from dominating the final decision.

The contribution of each detector to the final score is governed by

$$w_i = \pi_i \cdot \big(0.1 + 0.9 \cdot c_i\big) \cdot a_i, \tag{4}$$

where π_i is a fixed trust prior representing the overall reliability of detector i (DINOv3: 1.0, LipFD: 0.9, D3: 0.65), $c_i = 2|s_i - 0.5|$ is the confidence level of detector i's score, and a_i is the applicability factor. The raw fused score is then calculated as the weighted average

$$s_{\text{raw}} = \frac{\sum_i s_i \cdot w_i}{\sum_i w_i}, \tag{5}$$

and finally sharpened using a logistic function centred at 0.5,

$$s_{\text{ensemble}} = \frac{1}{1 + \exp\big(-k\,(s_{\text{raw}} - 0.5)\big)}, \quad k = 4. \tag{6}$$

VIII. VLM Reasoning Module

DeepfakeGuard contains an optional VLM reasoning module that extends beyond score-based fusion by leveraging each detector's known strengths and failure modes to detect cases in which simple arithmetic cannot provide resolution. The VLM also provides a human-readable description of forensic evidence for all instances where DeepfakeGuard declares a FAKE verdict.

Six uniformly sampled frames are taken from the video and assembled into a labelled 2×3 grid image. The grid, along with the ensemble score, is fed into a VLM with a structured prompt instructing the model to act as a digital forensics expert and identify anatomical errors, physics violations, temporal inconsistencies, or characteristic AI-generated artifacts. We support three interchangeable backends (GPT-4o mini, Claude, and a local Qwen2-VL-7B-Instruct option) that use identical prompts and output schemas.

The primary disadvantage of score-based ensemble fusion is that trust weights and applicability gates can reduce the weight of an unreliable detector but cannot explain why it produced a low confidence score or what forensic signals were missed. The VLM resolves this by receiving natural-language context for each detector's operating domain, known failure modes, and sample-by-sample diagnostic output, allowing it to reason jointly over signals that arithmetic scoring cannot combine. This positions the VLM not as a cosmetic explanation layer but as a reasoning component that mitigates the limitations of individual detectors. If its dependencies or API keys are unavailable, the module degrades gracefully and has no impact on the final detection decision.

IX. Ethical Considerations and Limitations

The deployment of publicly available open-source deepfake detectors has dual-use implications: while making the detection tools openly accessible empowers defenders to utilize them as well, the same access provides malicious actors with the ability to attempt to find weaknesses in the models and use those weaknesses to enhance their own forgery pipelines. In order to strike this balance, we have refrained from including adversarial robustness guarantees for our model and instead positioned DeepfakeGuard as a diagnostic tool and not as a definitive decision-making tool. Moreover, representation bias in the training datasets used by other researchers such as FaceForensics++ [10] and FakeAVCeleb [22] are likely to create demographic disparities in terms of the rates of false positives. To some degree, DeepfakeGuard addresses this issue through the utilization of self-supervised features (DINOv3) and training-free features (D3) that have lower dependencies on demographic-specific artifacts; however, this does not preclude the necessity for careful testing and analysis of DeepfakeGuard across all relevant demographics. Therefore, users must be cautioned against relying too heavily upon automated detection tools as automated determinations (without human-in-the-loop oversight) could result in the unjust flagging of genuine media or the failure to identify highly sophisticated and novel forms of forgery. Lastly, since generative models will continually evolve, the detection models will also require updates and re-calibration in a timely manner; thus, DeepfakeGuard should be considered as a component of a larger socio-technical response to deepfakes including policy, education, and regulation of platforms.

X. Conclusion

DeepfakeGuard shows that combining multiple detection methods, visual, audio-visual, and temporal, leads to stronger and more reliable deepfake detection than any single approach alone. While it is not a perfect or finalized solution, it is a meaningful step in the right direction. The goal was to build something open and flexible that other researchers can extend and improve upon over time. As deepfake technology continues to advance, tools like DeepfakeGuard will need to evolve alongside it, and technical solutions alone will not be enough. Addressing the broader threat of synthetic media will also require policy, education, and platform accountability.

References

[1] Y. Bengio *et al.*, "International AI Safety Report 2026," International Scientific Report on the Safety of Advanced AI, Feb. 2026. [Online]. Available: https://internationalaisafetyreport.org/sites/default/files/2026-02/international-ai-safety-report-2026.pdf (accessed Mar. 2026).

[2] Sensity AI, "The State of Deepfakes 2024: Landscape, Threats, and Detection," Sensity AI, Amsterdam, The Netherlands, Tech. Rep., 2024. [Online]. Available: https://5865987.fs1.hubspotusercontent-na1.net/hubfs/5865987/SODF%202024.pdf (accessed Mar. 2026).

[3] H. E. Chen and K. Magramo, "Finance worker pays out $25 million after video call with deepfake 'chief financial officer'," *CNN*, Feb. 4, 2024. [Online]. Available: https://www.cnn.com/2024/02/04/asia/deepfake-cfo-scam-hong-kong-intl-hnk (accessed Mar. 2026).

[4] C. Wang *et al.*, "Neural codec language models are zero-shot text to speech synthesizers," *arXiv preprint arXiv:2301.02111*, 2023.

[5] I. Perov *et al.*, "DeepFaceLab: Integrated, flexible and extensible face-swapping framework," *arXiv preprint arXiv:2005.05535*, 2020.

[6] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2Face: Real-time face capture and reenactment of RGB videos," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit. (CVPR)*, 2016, pp. 2387–2395.

[7] Y. Li and S. Lyu, "Exposing DeepFake videos by detecting face warping artifacts," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit. Workshops (CVPRW)*, 2019.

[8] D. Güera and E. J. Delp, "Deepfake video detection using recurrent neural networks," in *Proc. IEEE Int. Conf. Adv. Video Signal-Based Surveillance (AVSS)*, 2018, pp. 1–6.

[9] U. A. Ciftci, I. Demir, and L. Yin, "FakeCatcher: Detection of synthetic portrait videos using biological signals," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 11, pp. 2726–2740, 2020.

[10] A. Rössler *et al.*, "FaceForensics++: Learning to detect manipulated facial images," in *Proc. IEEE/CVF Int. Conf. Comput. Vision (ICCV)*, 2019, pp. 1–11.

[11] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 6105–6114.

[12] 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. Vision Pattern Recognit. (CVPR)*, 2020, pp. 3207–3216.

[13] M. Oquab *et al.*, "DINOv2: Learning robust visual features without supervision," *Trans. Mach. Learn. Res. (TMLR)*, 2024.

[14] T. Darcet, M. Oquab, J. Mairal, and P. Bojanowski, "Vision Transformers need registers," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2024.

[15] Meta AI Research, "DINOv3," *arXiv preprint arXiv:2508.10104*, Aug. 2025.

[16] X. Tan *et al.*, "Rethinking the up-sampling operations in CNN-based generative network for generalizable deepfake detection," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit. (CVPR)*, 2024.

[17] W. Liu *et al.*, "Lips are lying: Spotting the temporal inconsistency between audio and visual in lip-syncing deepfakes," in *Advances in Neural Inf. Process. Syst. (NeurIPS)*, 2024.

[18] C. Zheng *et al.*, "D3: Training-free AI-generated video detection using second-order features," in *Proc. IEEE/CVF Int. Conf. Comput. Vision (ICCV)*, 2025.

[19] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, 2016.

[20] L. Li *et al.*, "Face X-Ray for more general face forgery detection," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit. (CVPR)*, 2020, pp. 5001–5010.

[21] 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.

[22] H. Khalid, S. Kim, S. Tariq, and S. S. Woo, "FakeAVCeleb: A novel audio-video multimodal deepfake dataset," in *Advances in Neural Inf. Process. Syst. (NeurIPS) Datasets Benchmarks Track*, 2021.

[23] K. R. Prajwal, R. Mukhopadhyay, V. P. Namboodiri, and C. V. Jawahar, "A lip sync expert is all you need for speech to lip generation in the wild," in *Proc. ACM Int. Conf. Multimedia (MM)*, 2020, pp. 484–492.

[24] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 139, 2021, pp. 8748–8763.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit. (CVPR)*, 2016, pp. 770–778.

[26] European Parliament and Council of the European Union, "Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (AI Act)," *Official Journal of the European Union*, vol. L, 2024/1689, Jul. 2024. [Online]. Available: http://data.europa.eu/eli/reg/2024/1689/oj (accessed Mar. 2026).

[27] OpenAI, "Video generation models as world simulators," Tech. Rep., Feb. 2024. [Online]. Available: https://openai.com/index/video-generation-models-as-world-simulators (accessed Mar. 2026).

[28] Z. Yan, Y. Zhang, X. Yuan, S. Lyu, and B. Wu, "DeepfakeBench: A comprehensive benchmark of deepfake detection," in *Advances in Neural Inf. Process. Syst. (NeurIPS)*, 2023.