Adaptive GAN-Based Watermarking for Robust Deepfake Detection

Saradra Chanda
Institute of Engineering & Management, Kolkata
University of Engineering and Management, Kolkata,
Kolkata, India
saradrachandauembca2025@gmail.com

Manojit Das
Institute of Engineering & Management, Kolkata
University of Engineering and Management, Kolkata,
Kolkata, India
manojitbcauem2025@gmail.com

Sudipta Maity
Institute of Engineering & Management, Kolkata
University of Engineering and Management, Kolkata,
Kolkata, India
sudipta67uembca2025@gmail.com

Bipul Naskar
Institute of Engineering & Management, Kolkata
University of Engineering and Management, Kolkata,
Kolkata, India
bipul5139@gmail.com

Aritra Ghosh
Institute of Engineering & Management, Kolkata
University of Engineering and Management, Kolkata,
Kolkata, India
aritraghosh2025uemkbca@gmail.com

Ankur Biswas
Institute of Engineering & Management, Kolkata
University of Engineering and Management, Kolkata,
Kolkata, India
ankur.biswas@uem.edu.in

Tuhin Subhra Panda
Belda College,
Belda, India
tuhinpanda@beldacollege.ac.in

Anirban Das
Institute of Engineering & Management, Kolkata
University of Engineering and Management, Kolkata,
Kolkata, India
anirban.das@uem.edu.in

2025 International Conference on Engineering Innovations and Technologies (ICoEIT) | 979-8-3315-2595-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICOEIT63558.2025.11211773

***Abstract*—Progressions in Generative Adversarial Networks (GANs) have amplified the threat of deepfakes, challenging digital trust and authenticity. This paper proposes a novel GAN-based visible watermarking framework that uses a generator-discriminator layer in a looping fashion to embed indistinguishable, anti-tamper patterns into video frames. Spatial and temporal features are extracted through CNN and RNN architectures, while attention mechanisms specify the manipulated facial regions. Evaluated on FaceForensics++, Celeb-DF, and DFDC datasets using six frameworks. Unlike existing approaches, it effectively detects both watermarked and non-watermarked deepfakes and resists compression and motion-related tampering. This work presents a scalable, interpretable detection pipeline combining proactive watermarking with passive feature-based analysis.**

Keywords*—*Deepfake Detection, Generative Adversarial Networks, Digital Watermarking, Computer Vision, Deep Learning, Media Authentication

I. Introduction

Deepfake technology, which was first created for artistic purposes [1] but is now commonly abused for disinformation, privacy violations, and security risks, is made possible by Generative Adversarial Networks (GANs) [15]. Conventional detection techniques, such as CNN-based models and machine learning, have trouble being robust and generalizing. By adding recognizable markers to modified media, GAN-based visible watermarking provides a proactive way to address this issue while improving detection resistance and accuracy [2]. In order to increase generalizability, robustness, and real-time ap- plicability, this project intends to develop and assess Deepfake detection methods with a particular emphasis on GAN-based watermarking. It improves automatic and human identification attempts by incorporating visible indicators, reducing the risk of adversarial attacks and the increasingly dangerous hyper- realistic Deepfakes. As Deepfakes becoming more complex, this research is vital for improving media ethics, security, and integrity while offering scalable defenses against abuse and false information.

II. Literature Review

Advancements in Generative Adversarial Networks (GANs) have made possible the generation or making of hyper- realistic DeepFakes, raising significant security issues and privacy concerns. While deep learning methods surpass tra- ditional machine learning in detection [13], still the cur- rent techniques struggle against adversarial manipulations and dataset variations. Convolutional Neural Networks (CNNs) often lose their accuracy for real-world problems caused by them, leaving no choice but needing proactive solutions. GAN- based visual watermarking improves robustness by embedding

979-8-3315-2595-8/25/$31.00 © 2025 IEEE
DOI: 10.1109/ICoEIT.2025.99

distinct markers for both automated and human detection [10]. Additionally, explainability in DeepFake detection has become crucial, particularly in high-stakes applications. The Two-branch Autoencoder Network (TAENet) enhances interpretability by visually distinguishing real and fake content, balancing detection accuracy with transparency [6]. This focus on explainable AI strengthens trust and usability in combating DeepFake threat by analyzing spatial and temporal features of adulterated media [8].

III. Problem Statement

Digital security, privacy, and trust are seriously threatened by deepfakes, which are extremely realistic synthetic media produced by AI and GANs. They enable political manipulation, identity theft, and disinformation on a number of sites. The goal of this project is to improve digital authenticity by creating an automated Deepfake detection system that analyzes visual and aural anomalies using machine learning. Media integrity and individual privacy are impacted when Deepfake technology develops and makes it harder to discern between authentic and fraudulent content. Adversarial attacks, hyper- realistic changes, breaches of privacy, disinformation, and the end of trust in digital communication are some of the major issues. To combat these problems and safeguard digital truth, robust legal measures, public awareness campaigns, and effective AI detection are needed.

IV. Solution

A. Approach

Resilient watermarks and non-watermarked, both visible and. Be equipped to deal with manipulations like permanent It should be. These attacks will be overcome by utilizing noise, cropping and compression. Deepfakes [5]. Durability and authenticity are guaranteed. use of applying watermarks or cryptography in frequency domains. While self-destructive watermarks guard. Heavy-duty decryption makes the process fool-proof [7]. automate watermark identification and verification. Real-time monitoring is made possible by integration with social media platforms, and ongoing testing guarantees scalability and efficacy, improving accountability and transparency in the fight against Deepfakes [3].

B. Mathematical Implications

*1) **Input Data**:* The input video frames are represented as:

$$I -> \{I_1, I_2, \ldots, I_N\},\ I_i \in \mathbb{R}^{H \times W \times C} \tag{1}$$

I: The set of video frames where I_i: The i-th frame in the video. and H, W, C: The height, width, and color channels (e.g., RGB) of the video frames. [9] [4]

*2) **Watermark Embedding**:* A generator G embeds a visible watermark W into each frame I_i, resulting in a watermarked frame I_i^w:

$$I_i^w -> G(I_i, W; \theta_G) \tag{2}$$

[12] Where: G is the generator model, W is the visible watermark, and θ_G: The parameters of the generator G. To maintain visual integrity, the objective is to reduce the difference between the original frame I_i and the watermarked frame I_i^w:

$$\min_{\theta_G} \|I_i - I_i^w\|_2 \tag{3}$$

*3) **Feature Extraction**:* Analects spatial and temporal features from the watermarked frames I_i^w:

$$\Phi_{\text{spatial}} -> \text{CNN}(I_i^w; \theta_{\text{CNN}}) \tag{4}$$

$$\Phi_{\text{temporal}} -> \text{RNN}(\{I_1^w, \ldots, I_N^w\}; \theta_{\text{RNN}}) \tag{5}$$

Where, Φ_{spatial} are the features extracted from individual frames like contours, color, gradient, hue, etc using a convolutional neural network (CNN). Φ_{temporal} features representing temporal dependencies which mainly include changes or inconsistencies across the sequence of frames during movements, are extracted using a recurrent neural network (RNN) [8] [10].Where the symbol θ_{CNN} is denoted as the set of parameters of the CNN. θ_{RNN} is denoted as the set of parameters of the RNN and last but not the least $\{I_1^w, \ldots, I_N^w\}$ which signifies the sequence of video frames that are watermarked. [14]

*4) **Discriminator**:* A discriminator D is trained to distinguish and classify frames as real or fake, focusing on watermark integrity. The loss function for the discriminator is defined as:

$$L_D -> -\mathbb{E}[\log D(I_i^w)] - \mathbb{E}[\log(1 - D(G(I_i, W)))] \tag{6}$$

Where, D is the discriminator model, I_i^w denotes the watermarked frame. G represents The generator model embedding the watermark. I_i is the original frame. W is the visible watermark. $\mathbb{E}$ is the expectation operator. L_D denotes the loss function of the discriminator. The goal of the discriminator is to correctly classify real frames (watermarked frames) and fake frames (frames generated by G). [6] [16]

*5) **Adversarial Loss**:* The function for adversarial loss for the generator is defined as:

$$L_G -> -\mathbb{E}[\log D(G(I_i, W))] \tag{7}$$

Where Generator Loss (L_G) represents the loss function for the generator G. The Negative Sign ($-$) indicates that the generator's objective is to boost the discriminator's belief that the generated frames are real. The Expectation ($\mathbb{E}[\cdot]$) operator denotes the average taken over all input frames I_i. Generator Output ($G(I_i, W)$) where the watermarked frame is generated by embedding the visible watermark W into the original frame I_i. The Discriminator Output $D(\cdot)$ representing the chances or the probability that its input frame is a real frame. Lastly the Log Probability ($\log D(G(I_i, W))$) is the log probability that the discriminator considers the generator's output $G(I_i, W)$ as real. [16]

*6) **Detection Model Training** :* The detection model M is trained to classify videos as Deepfake or authentic:

$$y = M(\Phi_{\text{spatial}}, \Phi_{\text{temporal}}; \theta_M) \quad (8)$$

Where Detection Model (M) is the model responsible for classifying videos based on extracted spatial and temporal features and (y) is the Output, where 0 denotes an original video and 1 represents a Deepfake video. Model Parameters (θ_M) are the learnable parameters or characteristics of the detection model M that are being extracted during training. which includes features like Φ_{spatial} and Φ_{temporal}

*7) **Performance Metrics**:* To evaluate the model's performance, the following metrics are used:

- Accuracy (Acc): Accuracy evaluates the overall correctness or closeness between the predicted output of the model and the actual output:

$$\text{Acc} = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

Where TP , TN , FP , and FN represent the predictions as true positives, true negatives, false positives, and false negatives, respectively.

- Precision (Prec): Precision calculates the proportion of true positive predictions among all positive predictions in a confusion matrix:

$$\text{Prec} = \frac{TP}{TP + FP} \quad (10)$$

- Recall (Rec): Recall assesses the model's ability to identify all true positives in a confusion matrix:

$$\text{Rec} = \frac{TP}{TP + FN} \quad (11)$$

- F1-Score (F1): The F1-score calculates or measures the performance of the model by calculating the average of precision and recall, maintaining a balance between both metrics

$$\text{F1} = \frac{2 \cdot \text{Prec} \cdot \text{Rec}}{\text{Prec} + \text{Rec}} \quad (12)$$

C. *Workflow*

Here is the step-by-step workflow:

- Gather authentic and manipulated videos using datasets like FaceForensics++, Celeb-DF, and DFDC from kaggle and through other websites on the web.
- For extracting the important facial traits like contours, hue, and inconsistencies among the frames using libraries like OpenCV to extract frames from a video and facial chracteristics using YOLOv8 a recent state-of-the-model, we cleaned and augmented the each video by just only focusing on the facial region. By following these procedures, a clean train-test-validation dataset was produced.

*1) **For detecting watermarked deepfake:**:*

- To ensure uniformity in the frames, the process of pre-processing involves removing frames from the combination of frames, showing the face, cropping the face, down- scaling, normalizing of the pixels values (generally to size 256 x 256). These steps make the data ready for further processing by ensuring every frame is consistent and the same. [6]
- Watermark embedding is performed using a generator G, which embeds a visible or invisible watermark W into frames I_i, resulting in watermarked frames I_i^w. This step introduces a known pattern into the video for easier detection of manipulation [11] [15]. The embedding is mathematically defined as :

$$I_i^w = G(I_i, W; \theta_G) \quad (2)$$

where G is the symbol for generator, W is the symbol for denoting the watermark, and θ_G are the generator's parameters.

- To keep consistency and integrity in the visual media, the generator is trained to reduce perceptual differences or contrasts between the original frame I_i and the watermarked frame I_i^w. The reconstruction loss function ensures that the watermark is nearly invisible:

$$\min_{\theta_G} \| I_i - I_i^w \|_2 \quad (3)$$

This enhancement ensures that the watermarked frames closely resemble the original frames.

- The generator-discriminator loop is trained using adversarial learning. The discriminator D is trained to identify frames as watermarked or not, while the generator is optimized to implant watermarked frames that are hard to classify. The formula for adversarial loss for the generator is:

$$L_G = -\mathbb{E}[\log D(G(I_i, W))] \quad (7)$$

The loss function for the discriminator is defined as:

$$L_D = -\mathbb{E}[\log D(I_i^w)] - \mathbb{E}[\log(1 - D(G(I_i, W)))] \quad (6)$$

This iterative process reduces the error and improves both the generator's and discriminator's performance.

- Attention mechanisms in deepfake detection keep a check on key facial anatomy , such as the eyes and mouth(especially lip region) and facial structure to improve accuracy and reduce computational load. They compute importance weights (a_i) for each region and calculate attention as a weighted sum of features:

$$\text{Attention} = \sum_i a_i \cdot \text{Feature}_i \quad (13)$$

This ensures the model dynamically prioritizes regions most indicative of tampering.

- CNNs decipher a watermarked image's texture, artifact, and pattern caused by watermarks, and help extract spatial features. This process is expressed as:

$$\Phi_{\text{spatial}} = \text{CNN}(I^w; \theta_{\text{CNN}}) \quad (4)$$

In the above equation, $\Phi_{spatial}$ is the spatial features and I_i^w is the image having watermark or tampering artifact and the symbol θ_{CNN} denoted as the parameters of the CNN. Through this, the model learns essential spatial features needed to recognize tampering.

- Temporal features consist of frame sequences that help to preserve the consistency of motion. RNNs study the time-related connections between two succeeding frames to detect anomalies due to tampering. The computation of temporal features is expressed as:

$$\Phi_{temporal} - > RNN(\{I^w_1, \ldots, I^w_N\}; \theta_{RNN}) \quad (5)$$

In this case, the representation $\Phi_{temporal}$ stands for the temporal features that are extracted. The representation $\{I^w_1, \ldots, I^w_N\}$ is a sequence of input frames containing the watermark or tampering artifacts. The representation θ_{RNN} denotes the parameters of RNN. This enables the model to identify tampering by detecting motion inconsistencies.

*2) **For detecting non-watermarked deepfake:**:*

- For non-watermarked deepfakes there is no concrete way to detect it other than just relying on comparing the spatial and temporal feature of each frames in a video and processing it with the help of CNN and RNN models. which is mathematically expressed as :

$$\Phi_{spatial} - > CNN(I^w_i; \theta_{CNN}) \quad (4)$$

$$\Phi_{temporal} - > RNN(\{I^w_1, \ldots, I^w_N\}; \theta_{RNN}) \quad (5)$$

- Temporal coherence refers to the comparison of retrieved temporal properties across frames to check the consistency of motion across successive frames. The Essential Step In The Detecting Process Is Temporal Analysis Because Watermarked Deepfakes Usually Have Tampered Motion Patterns.
- A validation dataset is used to assess the model's performance using measures like the F1-score (12), recall (11), accuracy (9), and precision (10). The metrics determine the model's robustness and performance to identify watermarked Deepfakes in various datasets and situations.

D. *Design and Architecture*

We have provided a comprehensive flowchart design of the working of the model.1

V. Experimental Setup and Result Analysis

A. *Framework Comparision*

We have the table for three different frameworks out of six and had a comparative study between them I. Hence we declare the best framework for detecting deepfakes that maintains accuracy, scalability, and model interaction is **TensorFlow-GAN**. It handles both watermarked and non-watermarked deepfakes and supports TPU-based inference like Google Cloud TPU v5e and allowing it to train for 50-55 number of epochs for batch sizes of 32-64, making it a perfect choice for large-scale production.

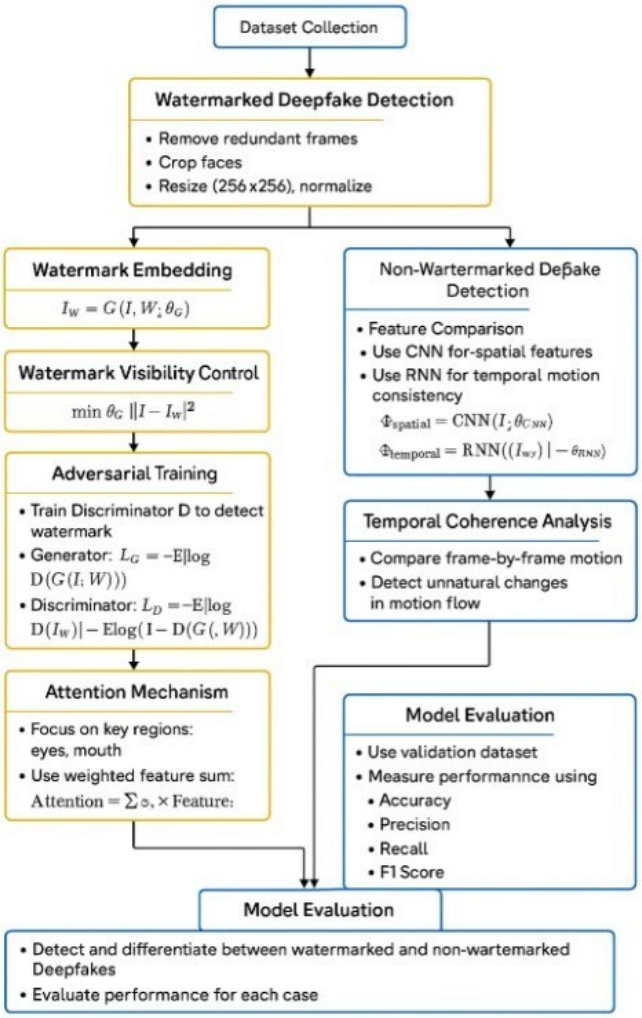

Fig. 1. Design and Architecture

B. *Results*

Accuracy, precision, recall, and F1-Score are some of the most important evaluation measures used to compare the performance of deepfake detection frameworks. FaceForensics++, Celeb-DF, and DFDC are three most popular and widely used datasets from which these metrics are computed for each framework.

*1) **Accuracy**:* We have calculated the accuracy using formula (9). Out of all the six **TensorFlow-GAN showed the most accuracy with FaceForencics++ which was 93%** and for Celeb-DF it showed 0.90 and for DFDC it showed 87% accuracy while hugging face diffusers came closer to Tensorflow-GAN rest were in range of about from 0 to 90% whose figures are given below 2

*2) **Precision**:* We have calculated the precision using formula (10). Out of all the six frameworks. **TensorFlow-GAN showed the most precision with FaceForencics++ which was 93%** and for Celeb-DF it showed 87% and for DFDC it showed 86% precision while hugging face diffusers came

TABLE I
COMPARISON OF GAN LIBRARIES

Libraries	Processing Power	Time Consumption	Hardware Support
TensorFlow-GAN	High processing power; efficient for large datasets	Fast with TPU acceleration	Supports GPU, TPU, and CPU for optimized performance
PyTorch-GAN	Medium computational power; slower for large-scale tasks	Slow for training and testing	Supports GPU and CPU; lacks TPU support
Hugging Face Diffusers	Optimized for static frame processing; lacks efficient video processing	Fast for images; slower for deepfake videos	Efficient on GPU and CPU; lacks native TPU integration

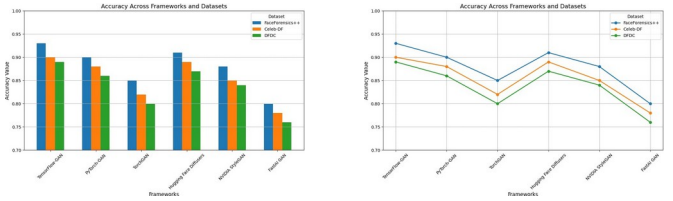

Fig. 2. Line and Bar plot for Accuracy across frameworks for datasets

closer to Tensorflow-GAN rest were in the range of about 0% to 90%. TensorFlow-GAN is the best-performing framework on the FaceForensics++ dataset, which is the most demanding. Its 92% precision further increases its appropriateness for high-quality deepfake detection applications. Like in the figure below3

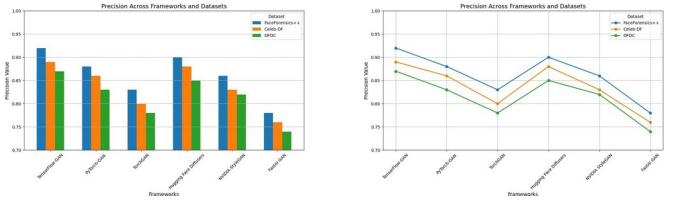

Fig. 3. Line and Bar plot for Precision across frameworks for datasets

*3) **Recall**:* With 90% on FaceForensics++, 88% on Celeb-DF, and 86% on DFDC, TensorFlow-GAN is once again the best on all datasets. PyTorch-GAN performs admirably, coming in second with 87% on FaceForensics++ and 85% on Celeb-DF. Hugging Face comes closer to it rest are all behind in terms of performance. With TensorFlow-GAN attaining 90%, FaceForensics++ exhibits the highest recall among the datasets, demonstrating its supremacy in identifying deepfake material. Which is demonstrated in figure 4

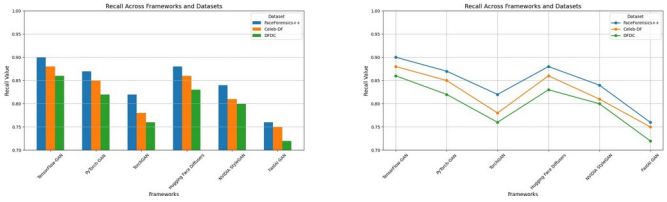

Fig. 4. Line and Bar plot for Recall across frameworks for datasets

*4) **F1-Score**:* TensorFlow-GAN continues to lead the F1 score with remarkable results of 88% on Celeb-DF, 86% on DFDC, and 91% on FaceForensics++. With F1 scores of 88% on FaceForensics++ and 85% on Celeb-DF, PyTorch-GAN also exhibits strong performance, showing a good overall balance between recall and precision. Competitive F1 scores are also displayed by Hugging Face Diffusers, especially on FaceForensics++ (89%). **With F1 score of 91%, TensorFlow- GAN demonstrates its balanced performance in both precision and recall on FaceForensics++**. 5

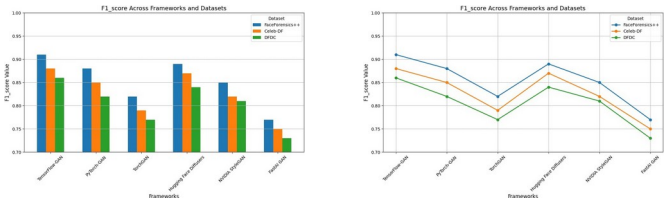

Fig. 5. Line and Bar plot for F1-Score across frameworks for datasets

C. *Key Findings*

We have found that:

- The TensorFlow-GAN with FaceForensics++ dataset gives the optimum results for all the parameters which includes metrics like Accuracy, Precision, Recall and F1-score.
- · The model works well with video footage that has been watermarked, but it has trouble with non-watermarked content.
- When the face is inside a given threshold, the model for facial deepfakes is quick and accurate; when the face is outside the threshold, the model fails
- While it can successfully identify deepfakes produced by Gen AI, it has trouble identifying manually modified deepfakes made with facial editing software.
- Even if the face is real, the model could produce inaccurate results for substantially manipulated videos, like those with VFX or backdrop alterations. One potential solution could be during pre-processing, if we remove the background and keep the designated figure or person intact in the video, and move forward with the rest of the process to get the desired output.

D. *Ethical Implications*

Due to our model's strong performance, there are very few ethical implications that can be very easily countered if certain conditions are kept in check. Certain concerns are:

- Using widely used datasets like FaceForencics++, DFDC. A balanced model has been trained due to that, there is a chance of increased bias which can be countered using a dataset with much varied data or is constantly updated.
- There can be a prevention or limitation on the freedom of speech and expression of a person under the pretext of fake content filtering which can be countered through transperency and government policies.

VI. Conclusion

This study uses both watermarked and non-watermarked techniques to provide a strong framework for identifying deepfakes. By optimizing reconstruction loss, the addition of visible or invisible watermarks improves the model's detection of alterations while preserving the film's visual quality. The suggested approach successfully captures discrepancies in texture, motion, and physiological behavior by extracting spatial and temporal information, offering a complete deepfake detection solution. An extra line of protection against artificial video manipulations is provided by the adversarial training configuration between the discriminator and generator, which guarantees that the watermarking process stays undetectable but subtle. The model's robustness in real-world situations is further demonstrated by its resistance to post-processing methods like blurring and compression. All things considered, this study provides the groundwork for creating sophisticated, trustworthy deepfake detection systems by highlighting the significance of fusing cutting-edge watermarking strategies with conventional detection techniques. Future research can concentrate on improving scalability, dealing with new manip- ulation methods, and including real-time detection features.

References

[1] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. "mesonet: a compact facial video forgery detection network". 2018. 10.1109/WIFS.2018.8630761.

[2] Joseph Bamidele Awotunde, Rasheed Gbenga Jimoh, Agbotiname Lucky Imoize, Akeem Tayo Abdulrazaq, Chun-Ta Li, and Cheng-Chi Lee. "an enhanced deep learning-based deepfake video detection and classification system". *Electronics*, 12(1):87, 2022. https://doi.org/10.3390/electronics12010087.

[3] Fatma Ben Aissa, Monia Hamdi, Mourad Zaied, and Mahmoud Mejdoub. "an overview of gan-deepfakes detection: proposal, improvement, and evaluation". *Multimedia Tools and Applications*, 83(11):32343–32365, 2024. https://doi.org/10.1007/s11042-023-16761-4.

[4] Akash Chintha, Bao Thai, Saniat Javid Sohrawardi, Kartavya Bhatt, Andrea Hickerson, Matthew Wright, and Raymond Ptucha. "recurrent convolutional structures for audio spoof and video deepfake detection". *IEEE Journal of Selected Topics in Signal Processing*, 14(5):1024–1037, 2020. https://doi.org/10.1109/JSTSP.2020.2999185.

[5] Liwei Deng, Hongfei Suo, and Dongjie Li. "deepfake video detection based on efficientnet-v2 network". *Computational Intelligence and Neuroscience*, 2022(1):3441549, 2022. https://doi.org/10.1155/2022/3441549.

[6] Fuqiang Du, Min Yu, Boquan Li, Kam Pui Chow, Jianguo Jiang, Yixin Zhang, Yachao Liang, Min Li, and Weiqing Huang. "taenet: Two-branch autoencoder network for interpretable deepfake detection". *Forensic Science International: Digital Investigation*, 50:301808, 2024. https://doi.org/10.1016/j.fsidi.2024.301808.

[7] Aya Ismail, Marwa Elpeltagy, Mervat Zaki, and Kamal A ElDahshan. "deepfake video detection: Yolo-face convolution recurrent approach". *PeerJ Computer Science*, 7:e730, 2021. https://doi.org/10.7717/peerj-cs.730.

[8] Rozi Khan, Muhammad Sohail, Imran Usman, Moid Sandhu, Mohsin Raza, Muhammad Azfar Yaqub, and Antonio Liotta. "comparative study of deep learning techniques for deepfake video detection". *ICT Express*, 10(6), 2024. https://doi.org/10.1016/j.icte.2024.09.018.

[9] Pavel Korshunov and Se´bastien Marcel. "vulnerability assessment and detection of deepfake videos". 2019. https://doi.org/10.1109/ICB45273.2019.8987375.

[10] Aakash Varma Nadimpalli and Ajita Rattani. "facial forgery-based deepfake detection using fine-grained features". pages 2174–2181, 2023. https://doi.org/10.1145/3625547.

[11] Guilin Pang, Baopeng Zhang, Zhu Teng, Zige Qi, and Jianping Fan. "mre-net: Multi-rate excitation network for deepfake video detection". *IEEE Transactions on Circuits and Systems for Video Technology*, 33(8):3663–3676, 2023. https://doi.org/10.1109/TCSVT.2023.3239607.

[12] Jiameng Pu, Neal Mangaokar, Lauren Kelly, Parantapa Bhattacharya, Kavya Sundaram, Mobin Javed, Bolun Wang, and Bimal Viswanath. "deepfake videos in the wild: Analysis and detection". 2021. https://doi.org/10.48550/arXiv.2103.04263.

[13] Md Shohel Rana, Mohammad Nur Nobi, Beddhu Murali, and Andrew H Sung. "deepfake detection: A systematic literature review". *IEEE access*, 10:25494–25513, 2022. 24-02-2022.

[14] Tal Reiss, Bar Cavia, and Yedid Hoshen. "detecting deepfakes without seeing any". *arXiv preprint arXiv:2311.01458*, 2023. https://doi.org/10.48550/arXiv.2311.01458.

[15] A. K. Sharma. "classification of indian classical music with time-series matching deep learning approach". *IEEE access*, 9:102041–102052, 2021. https://doi.org/10.1109/ACCESS.2021.3093911.

[16] Ziming Yang, Jian Liang, Yuting Xu, Xiao-Yu Zhang, and Ran He. "masked relation learning for deepfake detection". *IEEE Transactions on Information Forensics and Security*, 18:1696–1708, 2023. https://doi.org/10.1109/TIFS.2023.3249566.