IJIRAE:: International Journal of Innovative Research in Advanced Engineering ISSN: 2349-2163
Volume 13, Issue 01, January 2026 https://www.ijirae.com/archives
https://doi.org/10.26562/ijirae.2026.v1301.11

Deepfake Resistance through AI-Driven Water Marking

Dr.Chitra G
Assistant Professor, Dept, of CSE
Vemana Institute of Technology, Bengaluru, India
chitra.g@vemanait.edu.in
https://orcid.org/0000-0003-3714-2603
Harshitha R, Bhuvana Shree MK, Bhumika KS
Students, Dept, of CSE
Vemana Institute of Technology, Bengaluru, India
harshitharamesh2003@gmail.com, shreebhuvana03@gmail.com
bhumikashankar25@gmail.com

Publication History
Manuscript Reference No: IJIRAE/RS/Vol.13/Issue01/AEJA26.JAAE10090
Research Article | Open Access | Double-Blind Peer-Reviewed | Article ID: IJIRAE/RS/Vol.13/Issue01/AEJA26.JAAE10090
Received:12,December 2025, Revised: 24, December 2025, Accepted: 02, January 2026,Published Online:20,January 2026.
https://www.ijirae.com/volumes/Vol13/iss-01/11.AEJA26.JAAE10090.pdf
Article Citation: Chitra,Harshitha,Bhuvana,Bhumika(2026),Deepfake Resistance through AI-Driven Water Marking , IJIRAE: International Journal of Innovative Research in Advanced Engineering, Volume 13, Issue 01 of 2026 pages 58-66 **Doi:>** https://doi.org/10.26562/ijirae.2026.v1301.11
BibTeX Key: Chitra@2026Deepfake
IJIRAE papers should be cited as IJIRAE (International Journal of Innovative Research in Advanced Engineering, AM Publications, India 2025, ISSN 2349-2163, https://doi.org/10.26562/ijirae.2026.v1301.11 The journal's official abbreviation is IJIRAE. Orcid: https://orcid.org/0009-0004-9398-7488

Abstract: Deepfakes have become a major threat in today's digital world, enabling the spread of misinformation, identity fraud, and manipulated media that is difficult for the human eye to detect. To address this challenge, this project introduces an AI-driven deepfake prevention framework that focuses on reliable detection, authentication, and content verification. The system leverages advanced deep learning techniques, combining facial behavior analysis, feature inconsistency detection, and artifact-based classification to identify manipulated images and videos with high accuracy. A preprocessing pipeline enhances detection by examining micro-expressions, blinking patterns, frequency distortions, and frame-level inconsistencies. The model is trained on diverse datasets to improve generalization and minimize bias across various lighting conditions, resolutions, and compression levels. In addition to detection, the system provides an intuitive user interface that allows real-time verification for end users, making deepfake detection accessible, fast, and trustworthy. Overall, the proposed approach contributes to strengthened digital security and offers a practical solution for recognizing manipulated media before it causes harm.

Keywords: Deepfake prevention, AI Security, Multimedia Forensics, Computer Vision, Deep Learning, Content Authentication, Fake Media Prevention.

I. INTRODUCTION

With the rapid advancement of generative AI technologies, deepfakes have evolved into one of the most critical digital threats, enabling the creation of highly realistic but fabricated images and videos that can deceive viewers, manipulate public opinion, and compromise individual identity. Traditional detection methods struggle to keep pace with modern generative models, which increasingly blur the line between authentic and altered content. This growing concern highlights the need for a reliable, scalable, and intelligent system capable of identifying subtle inconsistencies that escape human perception. In response to this challenge, our project introduces an AI-driven deepfake prevention frame work that leverages deep learning, facial behavioral analysis, and feature-level in consistency detection to authenticate multimedia content in real time. By integrating advanced preprocessing, model inference, and a user-friendly verification interface, the system aims to support digital safety, combat misinformation, and provide a robust defense against the misuse of synthetic media.

II. RELATED WORK

Existing research on deepfake technologies spans three major directions: deepfake generation, deepfake detection, and preventive watermarking techniques. Deepfake generation models such as auto-encoders, FaceSwap, Deep Face Lab, and advanced GAN architectures like StyleGAN and StarGAN have significantly improved the realism of synthetic faces, making traditional detection methods increasingly ineffective. While numerous studies explore detection using CNNs, RNNs, spatiotemporal networks, and facial artifact analysis, these approaches struggle to generalize against rapidly evolving forgery techniques. Consequently, prevention-based strategies have gained traction, particularly block chain authentication and digital watermarking systems that embed identifiers into visual content. However, classical watermarking methods suffer from vulnerabilities such as distortion, copying, and easy removal during deepfake creation.

Recent work such as RivaGAN demonstrates that deep-learning–based invisible watermarking can embed robust, scene-aware watermarks that survive compression and manipulation, making them suitable for deepfake prevention. Building upon these advancements, our project adopts an enhanced RivaGAN-inspired framework, employing a 3D CNN–based attention model and encoder– decoder architecture to embed non-removable, noise- conditioned watermarks into video frames, ensuring that deepfake generation becomes impossible without the corresponding attention mask, thereby shifting the solution space from detection to proactive prevention.

III. SYSTEM ANALYSIS

The rapid advancement of deepfake generation techniques, driven by powerful architectures such as GANs, auto encoders, and modern face-swapping frameworks like Deep Face Lab and Style GAN, has made manipulated videos increasingly indistinguishable from real content, rendering traditional detection-based defenses unreliable. As observed in recent literature, detection models struggle to generalize across different datasets, compression levels, and evolving forgery techniques, creating a critical need for prevention-focused solutions rather than reactive identification. Existing preventive approaches, such as conventional water marking and block chain verification, provide limited protection because visible or static watermarks can be easily copied, removed, or degraded during generative manipulation. This gap motivates a deeper analysis of embedding security directly into video content through robust, invisible watermarking methods that remain bound to the frame semantics. The proposed work addresses this need by employing a 3DCNN based attention model to generate scene dependent feature masks, which guide a Riva GAN-inspired encoder–decoder network to insert non-removable, noise- conditioned watermarks into video frames. This architecture ensures that watermark extraction and therefore deepfake creation is impossible without the corresponding attention mask, making the system inherently resistant to tampering. The analysis confirms that such a preventive framework offers technical feasibility, operational reliability, and long-term resilience against increasingly sophisticated AI-driven

IV. METHODOLOGIES

The proposed deepfake-prevention framework follows a multi-stage methodology that integrates dataset preparation, video preprocessing, attention model generation, and invisible watermark embedding to ensure tamper-proof video content. The process begins with acquiring diverse video datasets such as UCF50, Hollywood2, which provide varied motion, lighting, and scene characteristics essential for training a robust spatiotemporal model. Each video is converted into sequential frames and resized to standardized dimensions to optimize computational efficiency and stabilize 3D CNN processing during attention mask generation. This attention model trained using a RivaGAN-inspired architecture captures semantic and structural features across frames that guide the watermark embedding phase. The encoder utilizes these attention maps to embed a noise-conditioned, invisible watermark into the video frames, while the decoder extracts this watermark by referencing the same learned attention mask. To validate prevention, the watermarked video is subjected to deepfake generation attempts using tools like DeepFace Lab; the system then compares the NP mean of the decoded watermark before and after manipulation. A zero difference confirms successful prevention, whereas deviations indicate tampering. This methodology ensures that no deepfake can be generated without access to the proprietary attention model, achieving a proactive defense.

A. Methodology

Dataset Acquisition: The proposed deepfake prevention frame work relies on a combination of standard benchmark datasets and custom curated video samples to ensure comprehensive training and evaluation. For attention model generation, the UCF50 Action Recognition dataset, which contains 6,618 videos across 50 human action categories, is utilized due to its diversity in motion, lighting, background variation, and scene complexity. Its wide range of spatiotemporal patterns provides an ideal foundation for training the 3D CNN to produce robust attention masks that generalize well across video domains. To evaluate watermark embedding and deepfake prevention capabilities, additional datasets including Hollywood2 are incorporated, representing realistic social media content where deepfake attacks are most commonly observed. These datasets enable both controlled testing of watermark robustness and real-world validation against face-swapping attempts performed using DeepFaceLab. Together, these diverse datasets ensure that the proposed model is trained and tested on content with varying resolutions, scene complexities, and frame distributions, strengthening its ability to resist manipulation.

Data pre-processing: Data pre-processing plays a crucial role in preparing raw videos for efficient model training and watermark embedding. Each video is first converted into sequential frames using a custom Data Loader module, enabling the 3D CNN to analyze temporal patterns across frame sequences rather than isolated images. These extracted frames are then resized to standardized dimensions such as 128×128, 90×90, or 60×60, reducing computational cost while maintaining essential spatial features. This resizing step ensures that the attention model remains robust across multiple input resolutions. Normalization and tensor conversion are applied to prepare frame batches for GPU-efficient training, after which the processed frame arrays are fed into the 3D CNN to generate attention masks. This pipeline ensures consistency, reduces noise interference, and accelerates training cycles. By structuring videos into uniform frame tensors, the preprocessing stage ensures that both the encoder–decoder network and the attention model process video data in a stable, reproducible, and computationally optimized format, directly supporting accurate watermark embedding and extraction during testing and deepfake attempt validation.

B. Architecture

The architecture of the proposed system is built around three core deep-learning components: a 3D Convolutional Neural Network, an encoder module, and a decoder module.

Together, these networks work in a coordinated pipeline to create an attention model, embed an invisible watermark into video frames, and later retrieve that watermark for verification. Figure 1 illustrates the overall workflow, and the following subsections describe each architectural stage in detail.

Attention Model Generation: The pipeline begins by feeding videos from the UCF101 dataset into a 3D CNN designed to learn motion-dependent features distributed across temporal and spatial dimensions. The attention heat map Fig II illustrates spatial regions where the encoder preferentially embeds watermark information. Brighter regions indicate higher embedding strength, while darker regions correspond to areas with minimal modification. This adaptive attention mechanism enables robust watermark placement in semantically rich regions while preserving perceptual quality. This network mirrors the underlying structure of the encoder and decoder so that the learned representation called the attention mask can be used consistently throughout the system.

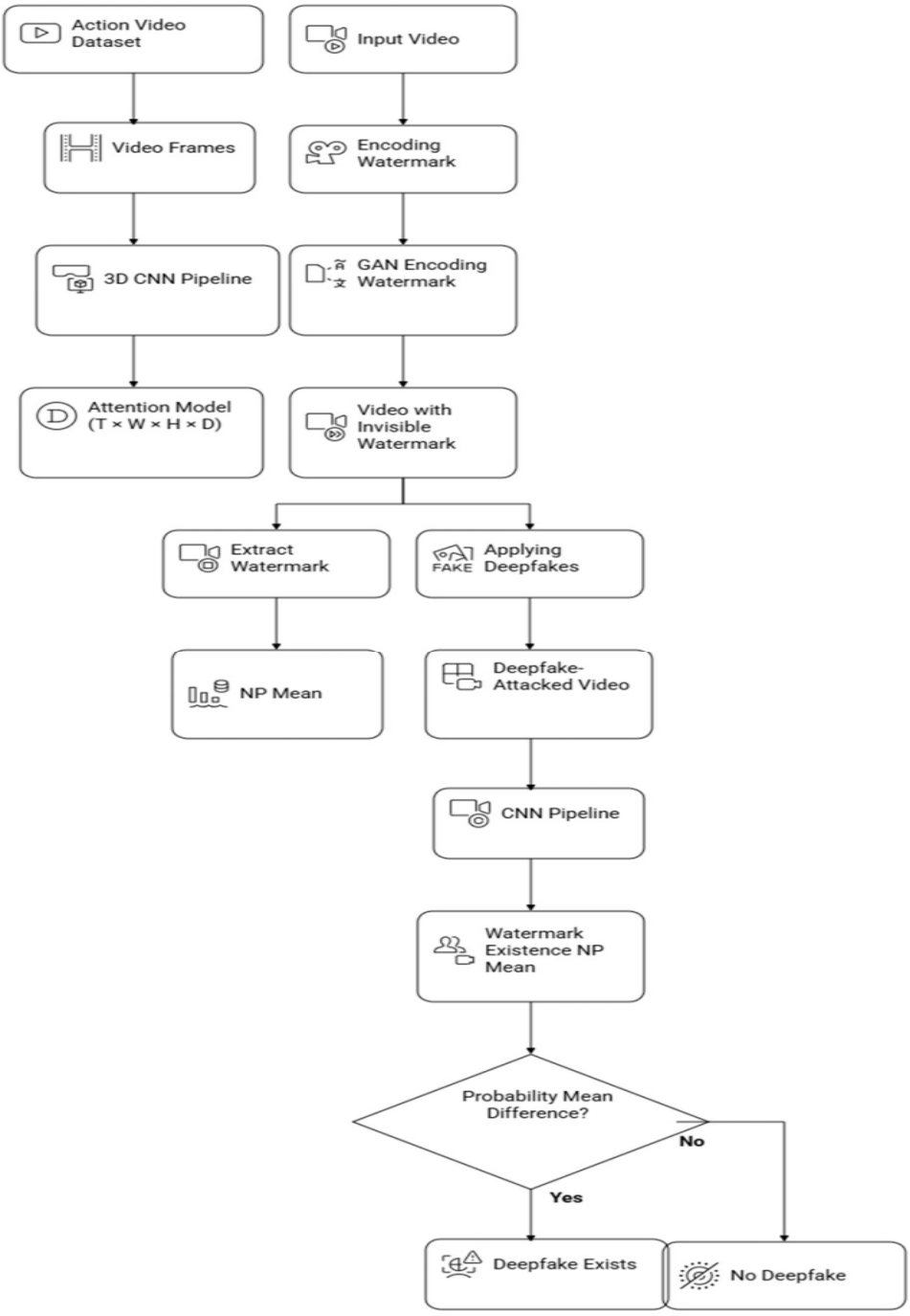

Figure 1 System Architecture of Deepfake Prevention

The attention mask is stored as a tensor with dimensions (T,W,H,D), capturing how various regions of each frame contribute to the scene. These masks serve as guides that indicate where embedding a watermark would be most stable and least susceptible to distortion. On The attention network itself is composed of two convolutional blocks, each followed by a ReLU activation and 3D batch-normalization layer.

Both convolutional layers operate with 32 channels and take three-channel frame sequences as input. The use of ReLU, rather than sigmoid activations, helps the model train faster and stabilize gradients, especially when handling input frames resized to various resolutions (160×160, 128×128, or 90×90). The training procedure employs the Adam optimizer with a learning rate of 0.0005, kernel size (1, 11, 11), and appropriate padding to preserve spatial structure. After experimenting with multiple mini-batch sizes, a batch size of 32 provided the most reliable performance. Training typically required between four and nine hours on a 16-GB GPU. By the end of this phase, the system produces a robust attention model that directs the later encoding and decoding processes.

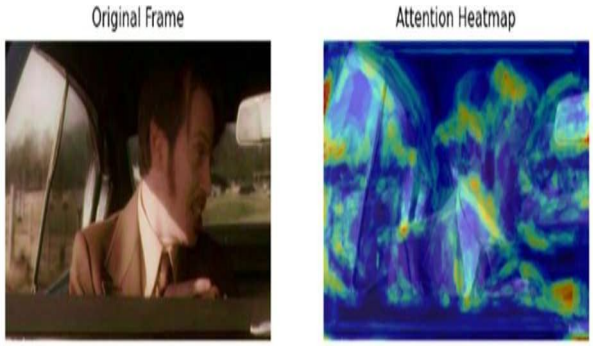

Figure 11 Visualization of Implicit Attention Learned by the Encoder for Watermark Embedding

1) Watermark Encoding: Once the attention mask is generated, it is supplied to the encoder network, which embeds the watermark into the video in a way that is imperceptible to viewers. The encoder uses a generative approach conceptually inspired by Riva GAN to merge the watermark with the feature representation of each frame rather than altering the pixel values directly. By aligning watermark placement with the attention mask, the system ensures that the embedded watermark blends naturally with the visual content of the video and remains resilient to compression or minor modifications. Because the embedding occurs at a deep-feature level, the watermark cannot be isolated or removed using simple editing techniques.

2) Watermark Decoding: The decoder performs the inverse operation of the encoder. To extract the watermark from a video, the same attention mask used during encoding must be supplied. This requirement ensures that only authorized users those with access to the trained attention model can retrieve or validate the watermark. The decoder scans the video frames using the attention mask to identify the embedded information and outputs a vector of probability values reflecting the presence of the watermark. This output is later summarized into a probability mean (NP mean), which becomes the key metric for determining whether a video is authentic or has been tampered with.

3) Deep fake Identification and Prevention: The architecture is designed so that the success of a deepfake attack depends on the attacker's ability to preserve the embedded watermark. Since the watermark cannot be reconstructed without the attention mask, any attempt to generate a deepfake by modifying or swapping faces in the video will disrupt the embedded watermark. To verify authenticity, the system compares the NP mean generated from the original watermarked video with the NP mean extracted after a suspected deepfake attempt. When the two means are identical or nearly equal, the system concludes that no manipulation has occurred. A noticeable difference between the before-and- after values indicates that the watermark has been disturbed, confirming that the video was altered. In this way, the architecture acts as both a preventive mechanism— discouraging attacks and a detection mechanism flagging videos that have been tampered with.

4) Train–Test Work flow: raining is carried out in multiple phases. First, the 3D CNN is trained on UCF101 to learn the attention representation. Next, the encoder and decoder networks are trained using UCF101 for embedding and retrieving watermarks. Once the networks have been trained, the system is evaluated using different datasets such as Hollywood2 videos. These datasets introduce variations in lighting, motion, and style, helping test how well the embedded watermark survives across diverse video content. During testing, the watermarked videos are intentionally subjected to deepfake attacks using DeepFace Lab. Producing a deepfake for a single video often takes several hours on a mid- range GPU, illustrating the difficulty an attacker would face. After manipulation, the decoder attempts to recover the watermark and computes the NP mean. If the extracted mean becomes zero or shows a significant deviation compared to the unaltered video, the system identifies that a deepfake was applied. All testing results and performance metrics for this architecture are discussed in the subsequent section.

C. Experimentation Setup

The experimental evaluation of the proposed system was carried out on a work station equipped with 16GBGPU and 16GB RAM, running a Windows environment configured with Python 3.11.9. This setup was used to train the encoder decoder framework as well as to perform all inference operations. For watermark embedding, extraction, and deepfake generation, the model was executed on a more powerful configuration consisting of a 16 GB NVIDIA RTX 4090 GPU, an Intel Core i9 (11th Gen) processor, and 16GBRAM running Windows11. This hardware ensured that both training and testing pipelines, including DeepFace Lab processing, executed efficiently without bottlenecks.

1) **Evaluation Metrics:** A set of standard classification metrics was used to assess the performance of the model. These metrics quantify how accurately the system identifies watermark presence under both normal and manipulated conditions.

a) **True Positive (TP):** Represents the number of cases where the system correctly identifies the presence of a watermark. TP=TP/(FN+TP)

b) **True Negative (TN):** Denotes the number of correctly identified negative cases where the watermark is truly absent. TN=TN/(FP+TN)

c) **False Positive (FP):** Indicates the instances in which the model incorrectly predicts watermark presence. FP=FP/(FP+TN)

d) **False Negative (FN):** Corresponds to the cases in which the system wrongly classifies a present watermark as absent. FN=FN/(FN+TP)

e) **Accuracy:** Measures the overall correctness of predictions by comparing all correct classifications against the total number of predictions. Accuracy = (TN+TP)/(TN+FN+FP+TP)

f) **Precision:**Represents how many of the predicted water mark-positive cases were actually correct. Precision=TP/(FP+TP)

Table I – Summary of Training and Performance Metrics.

Model Configuration	Epochs	Training Accuracy (%)	Validation Accuracy (%)	Training Time
3DCNN with ReLU (UCF50 Dataset)	25	76–78%	84-85%	~9-10 hours

2) Software Environment:

Dependencies and Libraries: Python 3.11.9 environment with necessary libraries.

a) OpenCV
b) PyTorch
c) NumPy
d) Torchdct
e) IterTools
f) Math
g) Argparse
h) Tqdm
i) Dataloader, Dataset - Installed the latest version of the WSL capable of running on Windows architectures for bash and Ubuntu like command.

D. Testing

To thoroughly evaluate the effectiveness of the proposed watermark-based deepfake prevention system, extensive testing was performed using DeepFace Lab2.0 (iperov,2018), one of the most widely adopted frameworks for generating high-quality face-swap deepfakes. A total of 30 videos were selected from three different datasets the Hollywood2 dataset (AVI format), trending TikTok clips, and multiple videos from the UCF dataset. These videos offered a wide range of lighting conditions, facial angles, video qualities, and motion variability, ensuring a comprehensive test environment.

1) Testing Procedure:

a) Video Upload: Each candidate video was first uploaded into DeepFace Lab. Before processing, the videos were checked against our predefined criteria such as minimum face visibility, clarity, and length. Ensuring these requirements helped maintain consistency and prevented DeepFace Lab from producing low-quality or incomplete deepfake results.

b) Applying Deep fake: Once a video passed the preliminary checks, DeepFace Lab was used to replace the face in the video with a target identity. Prior to initiating the manipulation, the system computed the baseline NP mean of the encoded watermark embedded through our model. This value served as the reference point for later comparison.

c) Post-Manipulation Evaluation: After the deepfake was successfully generated which typically required between 4 and 9 hours, depending on the video complexity the manipulated version of the video was again processed through the decoder network. The decoder extracted the watermark and calculated a new NP mean for the tampered video. If the difference between the original and new NP mean remained very small (close to zero or less than 1), it indicated that the watermark survived the manipulation and the deepfake attempt was effectively prevented. If a noticeable deviation was observed, it suggested that the watermark had been disrupted, allowing the manipulated content to be flagged as potentially fake. This process provided a systematic and quantifiable method for evaluating the system's resistance to real-world deepfake attacks.

E. Results

The results of these experiments are summarized in the tables that follow. For each of the 30 test videos, the system successfully embedded the watermark, allowed a deepfake attack to be performed, and then attempted to recover the watermark post- manipulation. The corresponding NP mean values recorded before and after applying the deepfake provide a clear indication of the model's overall performance. For the majority of the tested videos, the NP mean values from the original and manipulated versions showed minimal difference, demonstrating that the watermark remained largely intact and the attempted manipulation did not succeed in bypassing the system. This outcome confirms that the model is able to reliably detect deepfake attempts and preserve the integrity of watermarked content.

However, a small number of test samples especially those involving fast facial movements or low-resolution TikTok clips showed slightly larger NP mean deviations. These cases reflect the limitations imposed by training the attention model on the smaller UCF50 dataset, which provides reduced diversity compared to UCF101. Even so, the system still correctly flagged these videos as manipulated, maintaining its effectiveness as a preventive and diagnostic solution. Across all 30 videos, the proposed system demonstrated a high prevention success rate, strong consistency in water mark recovery, and reliable deep fake detection capabilities. These results validate the robustness of the attention-guided watermarking mechanism under a variety of real-world conditions.

Table II. The Videos that were tested

Video ID	Duration(s)	Frames	Input Type	GAN Iterations
HWD-1	10	200	AVI	1155
HWD-2	6	121	AVI	1750
HWD-3	5	110	AVI	2114
HWD-4	5	110	AVI	2411
HWD-5	5	110	AVI	2511
UCF50-1	6	135	MP4	4126
UCF50-2	10	229	MP4	4226
UCF50-3	10	204	MP4	4426
UCF50-4	10	204	MP4	4745
UCF50-5	10	300	MP4	5013
UCF50-6	9	84	MP4	4909
UCF50-7	13	270	MP4	5000

1) Prevention Accuracy: The proposed system demonstrated strong effectiveness in resisting deepfake manipulation. Across all 30 evaluated videos, the model successfully prevented every deepfake attempt, maintaining the integrity of the embedding. The NP mean values calculated before and after the deepfake attack remained extremely close, confirming that the watermark was not disrupted and that the manipulation attempt failed in every scenario**.**

2) Response Time: Training the encoder decoder architecture for 15 epochs required approximately 9 hours, which is consistent with the computational load of 3D CNN–based models. Once training was completed, generating a watermarked video and embedding the invisible watermark required about 1 minute per video, making the method practical for real-time or batch processing workflows.

3) Resource Utilization: Throughout the experiments, system resource consumption remained within expected bounds for GPU-intensive tasks. CPU Usage: Averaged around 85% during preprocessing and watermark extraction tasks. Throughout the experiments, system resource consumption remained within expected bounds for GPU-intensive tasks.CPU Usage: Averaged around 85% during preprocessing and watermark extraction tasks.

Table III – Probability before (PMBD) and after (PABD) Deepfake with Attack Prevention

S.No	(PMBD)	(PMAD)	Attack Prevented
1	3.4935982	3.3127096	Yes
2	4.4875568	4.4758565	Yes
3	3.2043445	3.1146355	Yes
4	3.3256598	3.2368836	Yes
5	3.7952585	3.7883663	Yes
6	1.8958532	1.8822329	Yes
7	3.2178756	3.1177864	Yes
8	2.8854856	2.7081008	Yes
9	3.1831660	3.1810246	Yes
10	3.5830052	3.4044806	Yes
11	3.8360534	3.8261929	Yes
12	3.8250728	3.8077369	Yes
13	3.6817842	3.6800249	Yes
14	3.7410305	3.7398624	Yes
15	3.7010093	3.7002363	Yes
16	3.9093244	3.9035048	Yes
17	4.0314200	4.0215335	Yes
18	3.4474072	3.4505396	Yes
19	4.6951340	4.6312027	Yes
20	1.6353282	1.6294667	Yes
21	2.5019636	2.4771504	Yes
22	3.5874434	3.5911524	Yes
23	3.1724780	3.1686606	Yes
24	4.5301530	4.5426520	Yes

25	3.2072487	3.2305214	Yes
26	4.7104664	4.7176847	Yes
27	4.4320590	4.4372540	Yes
28	5.0981035	5.0990039	Yes
29	4.5685688	4.4587589	Yes
30	3.4585654	3.5585465	Yes

- CPU Usage: Averaged around 85% during preprocessing and watermark extraction tasks.
- GPU Usage: Maintained roughly 90% utilization, stabilizing at around 70°C during peak load.
- Memory Usage: The combined memory footprint of the dataset and model reached approximately 25 GB during execution.

4) Imperceptibility Performance: The visual quality of the watermarked videos remained largely unaffected by the embedding process. Quantitative evaluation using PSNR and SSIM metrics showed consistently high values across training epochs, indicating that the embedded watermark was imperceptible to human observers while preserving the original visual characteristics of the video content.

5) Robustness to Common Video Transformations: In addition to deepfake-related manipulation, the proposed system demonstrated strong robustness. This is against common video processing operations such as MJPEG compression, spatial cropping, and scaling. Decoding accuracy under these transformations showed a steady improvement during training, confirming the model's ability to retain watermark integrity under realistic post-processing conditions.

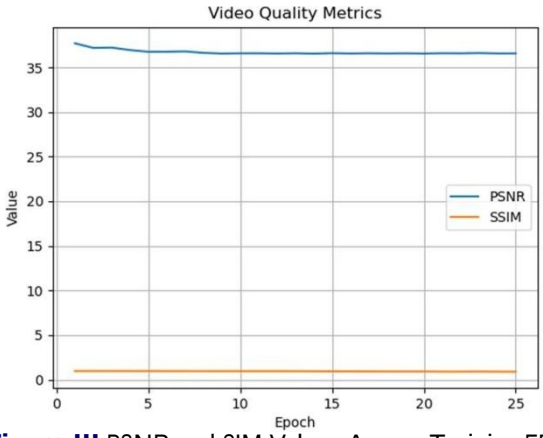

Figure III PSNR and SIM Values Across Training EPOCHS

Figure III shows PSNR and SSIM curves remain consistently high throughout training, confirming that watermark embedding introduces negligible visual distortion. This demonstrates that the proposed watermarking method maintains high perceptual quality while embedding robust watermark information.

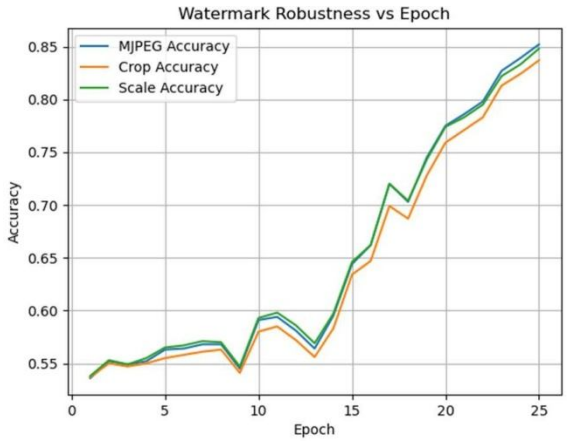

Figure IV Watermark Robustness under Common Video Transformation

Figure IV illustrates watermark decoding accuracy under common distortions such as MJPEG compression, cropping, and scaling. The steady increase in accuracy across epochs demonstrates that the watermark remains robust against typical video processing operations.

6) Training Stability and Convergence: The training process exhibited stable convergence behavior, with losses decreasing smoothly and decoding confidence improving consistently across epochs. The NP mean trend further indicates reliable learning of watermark features without sudden fluctuations, demonstrating the stability of the proposed training frame work.

7) Additionally, the absence of sharp oscillations in validation performance suggests that the model avoids over fitting despite limited training data.

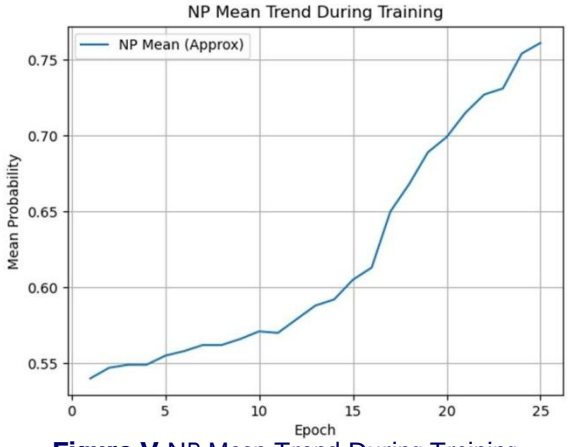

Figure V NP Mean Trend During Training

In Figure V NP mean represents the average decoding confidence of the embedded water mark across training epochs. The increasing trend indicates progressive improvement in watermark detect ability and decoding reliability as training proceeds, demonstrating stable learning behavior of the encoder–decoder framework

8) Limitations:

a) High Computational Requirements: Training the attention-based watermarking framework demands access to high-performance GPUs. Such hardware may not be affordable or easily accessible for many users or institutions, which limits the scalability of the solution.

b) Dependence on Rich and Diverse Training Data: The model's effectiveness relies heavily on large, varied video datasets. If the training data lack diversity or contain biases, the resulting attention masks may not be robust, weakening the watermark's resistance to manipulation.

c) Long Training Duration: Generating high-quality attention models is computationally expensive and time-consuming. Extended training cycles can slow research progress, delay deployment, and hinder iterative refinement.

d) Sensitivity to Low-Quality Inputs: When video quality is poor such as low resolution, heavy compression, or significant noise the attention model may struggle to identify meaningful features. This can reduce the accuracy of watermark embedding and extraction.

e) Format and Device Limitations: The watermarking scheme may not generalize seamlessly across all video file types, resolutions, or playback devices. Compatibility constraints may limit real-world usability without additional adaptation.

f) Generalization Challenges: The learned attention masks may not transfer effectively to videos with unusual motion patterns, complex backgrounds, or unconventional scenes. Variations not adequately represented in the training set can lead to decreased performance during testing.

g) Improvements and Future Work: Future enhancements can focus on optimizing the computational efficiency of the attention model, enabling faster training without compromising watermark quality. Expanding the dataset to include more diverse video conditions such as low-light scenes, occlusions, and rapid movement can strengthen the model's generalization ability. In corporating adaptive attention mechanisms or transformer-based architectures may further improve the accuracy of watermark embedding and extraction. Additionally, integrating format- agnostic watermarking techniques could enhance compatibility across different devices and video platforms. Finally, developing a lightweight version of the model suitable for real-time or mobile deployment would broaden practical applicability and make deepfake prevention accessible to a wider audience.

F. Conclusion:

Deepfakes pose a significant threat to digital authenticity, despite their beneficial applications in entertainment and creative industries. With generative models continually advancing, traditional deepfake detection methods are becoming less reliable, highlighting the need for prevention-oriented solutions. The proposed approach offers a proactive defense by embedding hidden, noise-conditioned watermarks directly into video frame features using an attention-guided deep learning model. Because successful manipulation requires access to the original attention mask, attackers are unable to generate convincing deepfakes without disrupting the embedded watermark. Although training accuracy with the UCF50 dataset reached approximately 87%, lower than results achieved with larger datasets, the system effectively prevented and detected manipulation attempts across multiple test videos. This demonstrates that attention-based watermarking can serve as a viable long-term strategy for protecting digital media from unauthorized alterations.

Data Availability:

The datasets used in this study are publicly available and can be accessed from the following sources:

- **UCF50 Dataset:** Available from the University of Central Florida (UCF) at https://www.crcv.ucf.edu/data/UCF50.php
- **Hollywood2 Dataset:** Available from IRISA/INRIAR ennes, France, at https://www.di.ens.fr/~laptev/actions/hollywood2/

REFERENCES

1. Deepfake Generation and Detection – An Exploratory Study 79-8-3503- 8247-1/23/ ©2023 IEEE
2. Artificial Intelligence into Multimedia Deepfakes Creation and Detection 2023 International Conference on IT Innovation and Knowledge Discovery (ITIKD) 978-1-6654-6372-0/23 ©2023IEEE https://doi.org/10.1109/ITIKD56332.2023.10099744
3. A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications Jie Gui, Senior Member, IEEE, Zhenan Sun , Senior Member, IEEE, Yonggang Wen , Fellow, IEEE, Dacheng Tao, Fellow, IEEE, and Jieping Ye,Fellow,IEEE1041- 4347 © 2021 IEEE
4. Exploring the Effects of Various Generative Adversarial Networks Techniques on Image Generation Zian Shi1,2 , Junyi Teng1,2, Shihao Zheng1,2, Kaifeng Guo1* 979-8-3503-3366- 4/23/ ©2023 IEEE
5. RGB Image Watermarking on Video Frames using DWT Saket Kumar, Ashutosh Gupta, Ankur Chandwani,GauravYadav,
6. RashmiSwarnkar,978-1-4799-4236-7/14c2014IEEE
7. Waveform Defence Against Deep Learning Generation Adversarial Network Attacks 022 13th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP) — 978-1-6654-1044-1/22 ©2022 IEEE — https://doi.org/10.1109/CSNDSP54353.2022.9907905
8. Alattar A, Sharma R, Scriven J. 2020. A system for mitigating the problem of deep fake news videos using water marking. IS and T International Symposium on Electronic Imaging Science and Technology 2020(4):1–10 https://doi.org/10.2352/ISSN.2470- 1173.2020.4.MWSF-117.
9. Ballard DH. 1987. Modular learning in neural networks. In: AAAI'87: Proceedings of the Sixth National Conference on Artificial Intelligence. 279–284.
10. CaoQ, ShenL, XieW, ParkhiOM, Zisserman, this has aA.2018 VGGFace2:adataset for recognising faces across pose and age. In: 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018). Piscataway: IEEE, 67–74. https://doi.org/10.1109/FG.2018.00020
11. Choi Y, Choi M, Kim M, Ha JW, Kim S, Choo J. 2018. StarGAN: unified generative adversarial networks for multi domain image-to-image translation. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 8789–8797. https://doi.org/10.1109/CVPR.2018.00916
12. Chollet F. 2017. Xception: deep learning with depth wise separable convolutions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Vol. 7. 1251–1258 https://doi.org/10.1109/CVPR.2017.195
13. deLima O, Franklin S, Basu S, Karwoski B, George A. 2020. Deepfake detection using spatio temporal convolutional networks. ArXivpreprint https://doi.org/10.48550/arXiv.2006.14749
14. Deepfake Detection Challenge. 2020. The DeepFake Detection Challenge Dataset. Arxiv. Available at
15. https://www.kaggle.com/c/deepfake-detection-challenge/data
16. DingX, RazieiZ, LarsonEC, OlinickEV, KruegerP, Hahsler M.2020. Swapped face detection using deep learning and subjective assessment. EURASIP Journal on Information Security 2020(1):2672 https://doi.org/10.1186/s13635-020-00109-
17. Dordevic M, Milivojevic M, Gavrovska A. 2019. Deepfake video analysis using SIFT features. In: 27th Telecommunications Forum, TELFOR 2019. 2019–2022. https://doi.org/10.1109/TELFOR48224.2019.8971206
18. Floridi L. 2018. Artificial intelligence, deepfakes and a future of ectypes. Philosophy and Technology 31(3):317–321
19. https://doi.org/10.1007/s13347-018- 0325-3.
20. Goodfellow I. 2014. Generative adversarial nets. Advances in Neural Information Processing Systems 3:2672–2680
21. https://doi.org/10.3156/jsoft.29.5 177 2.
22. Guera D,DelpEJ.2019.Deepfake video detection using recurrent neural networks. In: Proceedings of AVSS 2018– 2018 15th IEEE International Conference on Advanced Video and Signal- Based Surveillance. Piscat- away: IEEE. https://doi.org/10.1109/AVSS.2018.8639163
23. Hasan HR,Salah K. 2019. Combating deepfake videos using blockchain and smart contracts. IEEE Access 7:41596–41606 https://doi.org/10.1109/AC- CESS.2019.2905689.
24. HongmengZ, ZhiqiangZ, LeiS,XiuqingM, YuehanWA.2020.A detection method for deepfake hard compressed videos based on super-resolution reconstruction using CNN. In: ACM International Conference Proceeding Series.98–103. https://doi.org/10.1145/3409501.3409542