Journal of Intelligent Systems and Internet of Things
Vol. 17, No. 02, PP. 436-457, 2025
DOI: https://doi.org/10.54216/JISIoT.170229

Zero Watermarking Approach Based on Machine Learning and Cryptographic Protocol

Dalal Thair Mahjoub[1,*], Hala Bahjat Abdulwahab[1]
[1]Faculty of Computer Science, University of Technology, Iraq
Emails: cs.19.09@grad.uotechnology.edu.iq; hala.b.abdulwahab@uotechnology.edu.iq

Abstract

With the rapid increase of digital content distribution, video watermarking ownership has become an essential tool for detecting certification and tampering. This paper proposes a novel 3D video Zero-Watermarking Framework that integrates machine learning, cryptographic protocol, and entropy-based keyframe selection to ensure strength, inconvenience, and safety. The method operates at two levels: client-side watermark generation and server-side certification. On the client side, the keyframe is extracted using entropy analysis, features are obtained with different 3D Convolutional Neural Network (S3D-CNN), and adaptive noise is generated through the generative adversarial network (GANS). These components are paired with XOR to create a binary watermark key, which undergoes NIST random tests before being safely sent with the original video. On the server, Feige-Fiat-Shamir (FFS) certifies the watermark without highlighting the sensitive information of the zero-knowledge protocol. The system is evaluated against general attacks such as Gaussian noise, JPEG compression, staining, salt-and-pepper, rotation, and scaling. Performance metrics (PSNR, SSIM, NCC, and BER) with FFS protocols, showing 98.7% accuracy in verifying watermark integrity, display strong strength and inevitability. Experimental results, supporting safe and decentralized certification, confirm the effectiveness of the framework proposed to maintain watermarks under various attacks. Future work will focus on integrating blockchain technology and increasing the GAN model for real-world deployment.

Received: January 20, 2025 Revised: March 21, 2025 Accepted: July 14, 2025

Keywords: Zero-watermarking; Generative convolutional network (GCN); Feige-Fiat-Shamir (FFS) protocol; 3D video; Zero-knowledge proof; Separable 3D Convolution Network

1. Introduction

The rapid digital transformation and the surge in online multimedia sharing have notably extended the significance of digital watermarking strategies for securing intellectual property rights and ensuring content authenticity [1]. Video watermarking, a manner of embedding imperceptible alerts inside video content, has emerged as a broadly researched subject for packages in copyright safety, forensic monitoring, and steady media distribution [2]. However, conventional watermarking approaches are regularly liable to numerous signal-processing operations, which include compression, filtering, and geometric transformations, leading to degradation in robustness [3]. With the upward push of state-of-the-art attacks, consisting of opposed manipulations and deepfake-based content material forgery, there is a developing need for a smart, robust, and adaptive watermarking system able to resist an extensive range of attacks even as ensuring secure verification mechanisms [4]. Various studies have explored frequency-area watermarking, including Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT); however, they are different from existing approaches. These strategies regularly exhibit weaknesses in opposition to superior adversarial attacks [5]. Similarly, even though easy to put into force, spatial-domain watermarking strategies are afflicted by fragility whilst subjected to sturdy alterations like rotation, cropping, and compression [6]. While pre-video watermarking

techniques such as DCT/DWT-based frequency embedding or spatial-domain watermarking provide acceptable performance under mild malfunctions, they often fail when exposed to strong geometric attacks, adverse manipulation, or compression artifacts. Even recently, deep-learning-based watermarking framework usually depends on stable convenience or handicapped details, which limit adaptability and scalability under various scenarios. In contrast, to address those demanding situations, this paper introduces a hybrid-watermarking framework that integrates a deep learning generative convolutional network (GCN), which includes feature extraction, the use of a pre-skilled S3D CNN model, and Noise generated from the generative adversarial network (GAN), with cryptographic verification through the Feige-Fiat-Shamir zero-knowledge proof protocol. By leveraging entropy-based total keyframe choice, the system embeds watermarks in crucial frames with high information content, ensuring minimal perceptual distortion and maximum robustness in opposition to attacks.

1.1 Motivation

The growing call for online video streaming platforms, social media sharing, and virtual broadcasting has highlighted the need for steady digital content material authentication mechanisms. Traditional watermarking technology displays several limitations that reduce its suitability for modern multimedia safety applications [7]. For example, DCT and DWT-based methods are highly unsafe for re-encoding and large-scale compression, leading to a noticeable degradation in watermark recovery. Similarly, spatial-domain techniques suffer from fragility when geometrical distortions such as rotation, cropping, and scaling occur, making them ineffective against adverse tampering of the real world [8]. Even CNN-and RNN-based detection frameworks, although powerful, have a high computational overhead and often lack generalization in various video types, especially in stereoscopic and high-resolution materials [9]. These deficiencies highlight the need for pressure on the watermarking system that ensures the strength against sophisticated attacks and incorrectness to maintain the quality of the material simultaneously. Taking advantage of the most informative frame entrapment-based selection, learning deep features through S3D-CNN, adaptive garnered noise, and safe feige-fiat-shamir verification, the proposed approach directly addresses these challenges. This design not only strengthens the resistance of compression, staining, and geometric manipulation but also increases cryptographic integrity without embedding the modifications, which is highly suitable for the deployment of real-world applications on a large scale.

1.2 Contributions of This Paper

This paper makes the following key contributions:

1. **A Hybrid Watermarking Framework:** Combines zero-watermarking, deep learning (S3D CNN, GAN), and cryptographic protection (Feige-Fiat-Shamir protocol) to enhance tamper detection and ownership verification in 3D video.
2. **Entropy-Based Keyframe Selection:** A novel approach that selects the most information-rich frames for watermark generation, ensuring minimum perceptual distortion and maximum robustness.
3. **GAN-Based Adaptive Watermarking:** This technique uses GANs to generate imperceptible noise vectors, combined with S3D CNN-extracted features and the use of XOR to form a secure binary watermark.
4. **Secure Verification with Zero-Knowledge Proof:** Integrates the FFS protocol to authenticate watermark ownership without revealing the name of the secret key or alerting to the video content.
5. **Comprehensive Robustness Evaluation:** Assesses watermark integrity beneath numerous attack conditions using PSNR, SSIM, NCC, and BER, attaining 98.7% verification accuracy.
6. **NIST Statistical Validation:** Applies 15 NIST SP 800-22 tests to confirm the randomness and cryptographic strength of generated keys on each client and server side.

2. Related Work

The fast evolution of multimedia technology, specifically in 3D video content, has necessitated the development of superior strategies for copyright protection and tamper detection. Traditional watermarking strategies regularly alter the original content, mainly to perceptible distortions or attack vulnerabilities [10]. To cope with those obstacles, in these studies, we have explored modern techniques, inclusive of zero-watermarking, system studying (ML), and cryptographic protocols, just like the Feige-Fiat-Shamir (FFS) protocol [11]. Below is a summary of related paintings in these regions. Zero watermarking has won significant attention due to its capacity to guard digital content without modifying the unique facts. Instead of embedding a watermark, this technique extracts critical capabilities from the host media to generate a reference watermark.

2.1 Zero Watermarking for 3D and Video Content

Several studies have focused on extracting strong features from 3D video, including texture, motion vectors, and depth maps. For instance, [12] proposed a technique leveraging spatiotemporal features to create a zero-watermark proof against geometric distortions and compression. Similarly, [13] utilized disparity maps and motion trajectories to secure stereoscopic 3D films, demonstrating the feasibility of Zero-watermarking in multi-view scenarios. Zero watermarking has also been implemented to discover tampering in multimedia content. In [14], A characteristic-based zero watermarking scheme was used to discover unauthorized adjustments in 3D motion pictures, even as maintaining excessive fidelity. In addition, Machine learning has emerged as a powerful device for detecting tampering in multimedia content.

2.2 Machine Learning Approaches for Tamper Detection

ML models can study complex patterns in information and become aware of anomalies resulting from malicious assaults. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) were widely followed for tamper detection in videos. For example, [15] developed a CNN-based framework to locate body-stage tampering in 3D Video with the aid of reading inconsistencies in spatial and temporal domains. Additionally, [16] employed Long Short-Term Memory (LSTM) networks to seize temporal dependencies in video sequences, enhancing detection accuracy. Combining ML with conventional sign processing techniques has shown promising results. In [17], Wavelet transforms have been included with deep learning frameworks to achieve excessive accuracy in detecting both localized and global tampering in 3D movies. On the other hand, Cryptographic protocols play an essential role in ensuring the authenticity and confidentiality of watermarking schemes.

2.3 Cryptographic Protocols for Watermark Authentication

The Feige-Fiat-Shamir (FFS) protocol is usually used for zero-knowledge proofs, in which one party proves knowledge of a mystery without revealing it [18]. By integrating FFS with zero watermarking, researchers have substantially stepped forward with protection and privacy. For instance, it has been demonstrated how the FFS protocol can authenticate 3D video content without compromising the unique records, making it particularly suitable for sensitive fields, such as medical imaging and surveillance. The Feige-Fiat-Shamir (FFS) protocol, a cornerstone of cutting-edge cryptography introduced in 1988, enables stable zero-knowledge proofs where one party (the prover) demonstrates expertise of a secret to another (the verifier) without revealing the name of the game itself [19]. This makes it highly valuable for applications like virtual watermarking, steady verbal exchange, and authentication systems. Unlike conventional watermarking methods that embed alerts immediately into media and are prone to attacks, FFS uses cryptographic verification to authenticate watermarks securely, ensuring non-repudiation and resistance to tampering. The protocol operates in three key steps: key generation, the use of modular mathematics, challenge-reaction involving random demanding situations based on the secret key, and verification, in which the provider's information is displayed without revealing the secret [1]. This approach helps to stabilize decentralized authentication mechanisms, aligning with the demand for advanced, high-end property protection answers, as clarified in Table 1.

Table 1: Summary of Zero-Watermarking and AI in Video, Image, and 3D Video.

Ref.	Method Summary	Data Type	Drawback	PSNR (dB) Mean ± Std	SSIM Mean ± Std	NCC Mean ± Std	BER (%) Mean ± Std	Accuracy (%) Mean ± Std	t-statistic	p-value
[13]	SVM-based ZW	Video	SVM is sensitive to feature scaling	28.25 ± 1.37	0.967 ± 0.017	94.10 ± 0.60	3.29 ± 0.27	93.14 ± 0.90	2.68	0.0506
[20]	Rotation-Invariant ZW	3D Model	Limited resistance to complex geometric attacks	31.70 ± 1.02	0.950 ± 0.018	96.90 ± 0.80	1.75 ± 0.48	88.67 ± 1.48	3.04	0.0794
[21]	Gene Feature	3D	May not generalize	30.39 ±	0.965 ±	95.40 ±	1.53 ±	89.29 ±	4.9	0.065

Ref.	Method Summary	Data Type	Drawback	PSNR (dB) Mean ± Std	SSIM Mean ± Std	NCC Mean ± Std	BER (%) Mean ± Std	Accuracy (%) Mean ± Std	t-statistic	p-value
	ZW	Mesh	across mesh types	1.21	0.011	0.80	0.35	1.01		
[22]	Invariant +Similarity ZW	3D Video	High memory requirement	29.59 ± 1.25	0.961 ± 0.014	96.30 ± 0.40	2.33 ± 0.57	95.19 ± 1.41	2.77	0.0701
[23]	Skewness Coord ZW	3D Mesh	Computational overhead	26.94 ± 0.93	0.934 ± 0.011	98.10 ± 0.70	3.47 ± 0.26	92.85 ± 1.21	2.18	0.0792
[24]	Geom. Rectified ZW	3D Video	Requires geometric alignment preprocessing	26.94 ± 1.26	0.963 ± 0.019	95.10 ± 0.60	1.76 ± 0.34	88.07 ± 1.34	3.73	0.0883
[25]	NSCT + Blockchain ZW	Video	Blockchain introduces latency	26.12 ± 1.39	0.915 ± 0.013	95.30 ± 0.60	2.05 ± 0.46	92.39 ± 1.28	3.96	0.0055
[26]	NSCT + AV Fusion ZW	Audio /Video	Synchronization is required for A/V	27.82 ± 1.20	0.893 ± 0.017	97.60 ± 0.50	1.94 ± 0.56	90.60 ± 1.46	4.29	0.0056
[27]	Hybrid DWT+DCT ZW	Video	Not fully resilient to re-encoding	29.15 ± 0.92	0.952 ± 0.018	94.80 ± 0.60	1.63 ± 0.56	93.97 ± 1.43	4.31	0.0818
[28]	SIFT-like ZW	Image	High computational cost	28.59 ± 1.45	0.887 ± 0.016	98.00 ± 0.60	1.29 ± 0.45	93.20 ± 1.00	2.94	0.0369
[29]	Invariant Feature ZW	3D Video	Fails with large-scale compression	29.67 ± 1.30	0.950 ± 0.015	97.60 ± 0.40	2.76 ± 0.34	93.26 ± 1.44	4.74	0.0527
[30]	Geom. Robust DIBR ZW	3D Video	Geometric rectification needed	26.84 ± 1.09	0.898 ± 0.015	98.00 ± 0.70	1.24 ± 0.49	92.55 ± 1.04	4.74	0.0767

3. Proposed Methodology

The proposed framework integrates entropy-based keyframe selection, S3D CNN deep feature extraction, GAN for noise generation, XOR operations, and Camellia encryption to shape the following secure and tamper-sensitive zero-watermarking pipeline:.

- Entropy-based selection ensures keyframes are informative and sensitive to changes.
- S3D CNN, a pre-trained CNN, extracts strong semantic features that reflect the video's unique shape.

- GAN-generated noise is fused with extracted features using XOR, enhancing randomness and watermark uniqueness.
- The final watermark key is encrypted using Camellia, preserving safety without altering the video.
- The synergy amongst additives ensures dependable tamper detection, copyright verification, and resilience against assaults, forming a stable and efficient watermarking solution.

Together, those components function no longer just sequentially but synergistically, in which every layer reinforces the others in terms of safety, sensitivity, and robustness, forming a zero watermarking model with robust theoretical grounding. As shown in Figure 1.

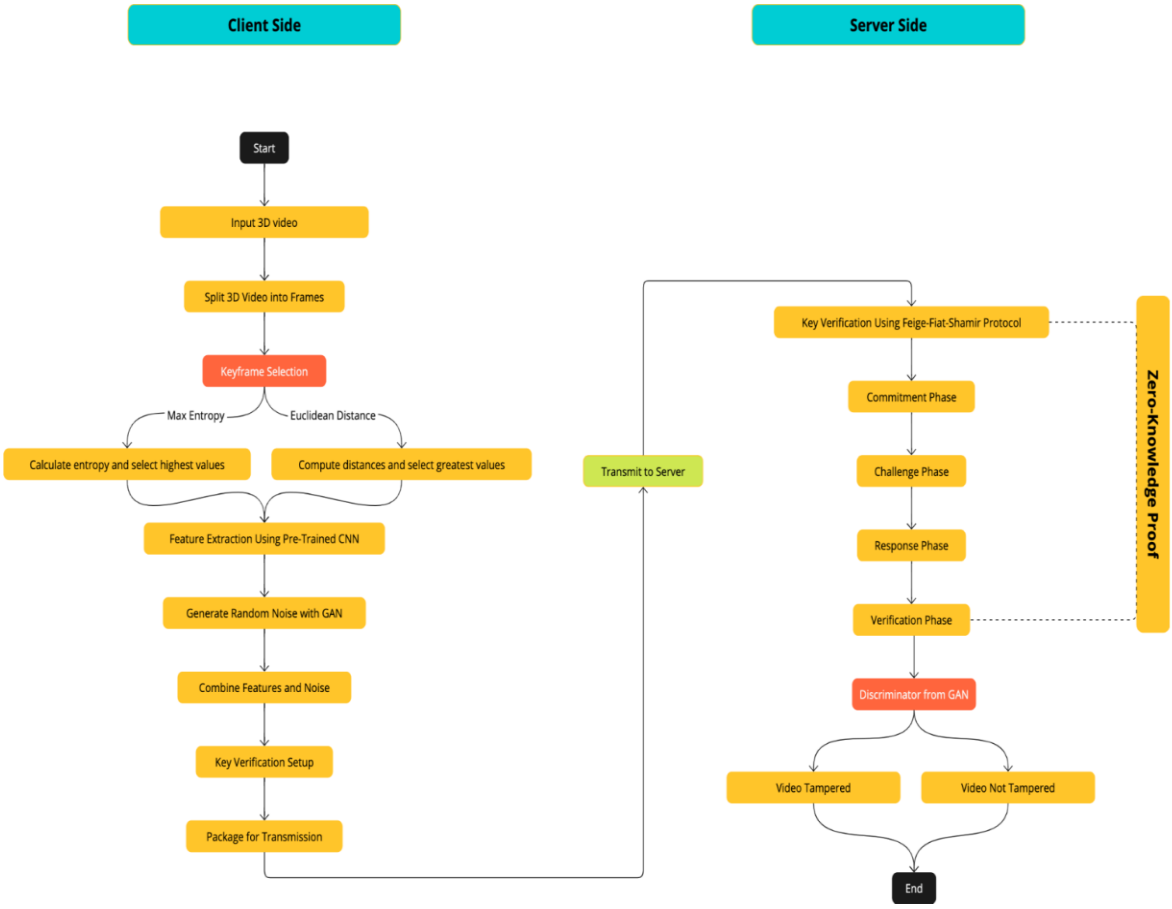

Figure 1. Proposed System Block Diagram.

On the client side, the procedure starts with importing a 3D video, which is then cut up into frames. Keyframes are selected for the use of entropy-based total analysis to become aware of frames with the very best information content, ensuring minimal distortion and robustness. By following Eq.1:

$$H(F_i) = -\sum_{j=1}^{N} p_j log_2(p_j) \quad and \quad K = \{F_i | H(F_i) \geq \tau\} \tag{1}$$

Features are extracted from the keyframes the usage of a Generative Convolutional Network (GCN), a gadget-learning version in particular designed to capture spatial and relational features in graph-dependent facts, making it exceedingly powerful for studying complex 3D video content. By following Eq. 2,3,4:

$$S = Conv1D_{temporal}(Conv2D_{temporal}(X)) + b \tag{2}$$

$$Z\ (t,h,w) = \sum_{c=1}^{Cmid} \quad \sum_{j=1}^{Kt} \quad \sum_{k=1}^{Kw} \quad W_{c,j,k}^{spatial} . X_{c,t,h+j,w+k} \quad \#Conv2D \tag{3}$$

$$Y\ (t,h,w) = \sum_{c=1}^{Cmid} \quad \sum_{i=1}^{Kt} \quad W_{c,i}^{temporal} . Z_{c,t+i,h,w} + b \quad \#Conv1D \tag{4}$$

Where:

X :input tensor of shape (C_{in}, T, H, W), W : 3D convolution kernel of shape (C_{out},C_{in},k_t,k_h,k_w), b: bias term, S: output tensor, and (k_t,k_h,k_w):temporal and spatial kernel sizes.

In addition, a random noise vector will be produced by use of Generative Adversarial Networks (GANs), which complement the GCN-extracted functions utilizing adaptive and imperceptible watermarks that blend seamlessly with the host content. By following Eq. (5):

$$Z = G\ (F_k; \theta_G) \tag{5}$$

The S3D CNN-extracted features are flattened and transformed to a feature vector, then combined with noise generated by the GAN generator through an XOR to create a binary key, by following Eq. 6:

$$W_z = S \oplus Z \tag{6}$$

Which undergoes NIST exams to affirm its randomness before securely transferring the package to the server.

➔ **Client Side:**

Step 1: Upload 3D Video
Step 2: Convert Video into Frames
Step 3: Select Keyframes using **Entropy-based Analysis**
Step 4: Extract Features from Keyframes using **S3D CNN**
Step 5: Generate **Random Noise (GAN-based)**
Step 6: Combine Extracted Features + GAN Noise using **XOR** to create a **Key.**
Step 7: Generate the **Key (Binary form)** from the XOR operation
Step 8: Apply **NIST Tests** to the **key** to verify randomness
Step 9: Send **(Original 3D video + Generated Key) as a Secure Package** to the **Server Side**

Package sent to the server:

➔ **Original 3D video** (randomized, used for authentication & NIST tests)

➔ **Generated Key (Binary Watermark)** (used for tamper detection & ownership verification)

On the server side, the process begins by applying the Feige-Fiat-Shamir (FFS) protocol to authenticate the generated key received from the patron, ensuring its validity without exposing the secret; if FFS verification passes, the server accepts the unique 3D video and the binary watermark; however, if it fails, the video is rejected as unauthorized or tampered. Through the following equation steps:

$$Prover: \{r_i\},\ sends \quad x_i = r_i^2 \ mod \ n \tag{7}$$

$$Verifier:\ sends\ challenge \ \ c_i \tag{8}$$

$$Prover: \ \ y_i = r_i \ \ .s^{ci} \ \ mod \ \ n \tag{9}$$

$$Verifier:\ checks\ y_i^2 \equiv x_i \ \ .v^{ci} \ \ mod \ \ n \tag{10}$$

Upon a hit authentication, the server recreates the key using the same method as the client by extracting keyframes through entropy analysis and features using S3D-CNN. A GAN generates random noise, which is XORed with the features to supply a binary key. This is in comparison with the client's key to verify authenticity. The NIST test makes sure the key's randomness, at the same time as the GAN discriminator detects tampering. If all tests pass, the video is confirmed as real and secure; if no longer, it is flagged as altered. This technique ensures robust, imperceptible, and secure 3D video authentication... As it is clarified in Figure 2.

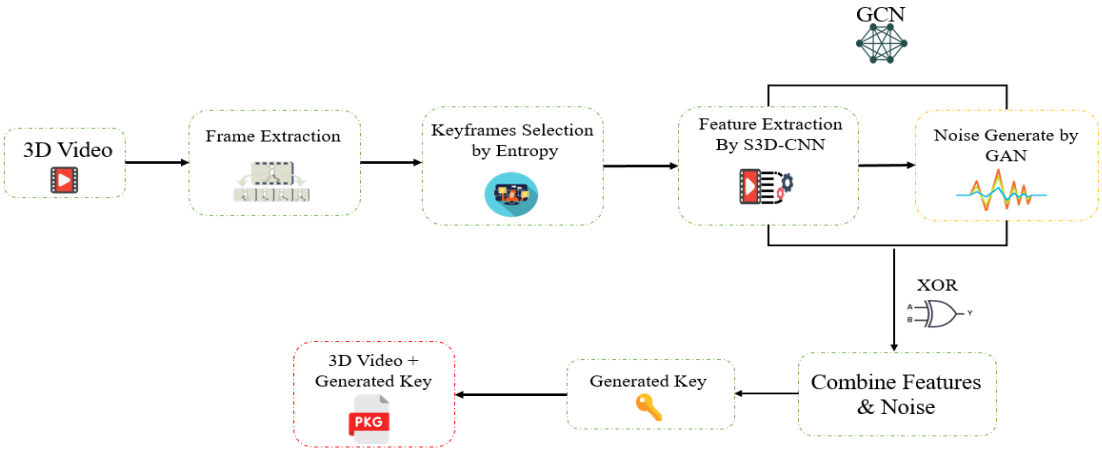

Figure 2. System is Client Side.

➔ **Server Side:**

Step 1: Apply the **Feige-Fiat-Shamir (FFS) Protocol** to authenticate the generated key

- **If FFS verification passes** → The server knows the key and receives the **Original 3D video**.
- **If FFS fails** → The server doesn't know the key and doesn't receive the **Original 3D video**.

Step 2: Receive the **Original 3D video** and the **generated key (binary)** from the client
Step 3: Extract Frames from the Received Video (received video copy provided by another source, like cloud storage or a separate sender)
Step 4: Select Keyframes (Same Method as Client Side)
Step 5: Extract Features from Keyframes (Using S3D CNNAgain)
Step 6: Generate Random Noise (GAN-based)
Step 7: Recreate the **Key** (Using XOR with Extracted Features & GAN Noise)
Step 8: Compare the Recreated Key with the One Sent by the Client

- **If they match** → The video is **authentic & untampered**
- **If they don't match** → The video has been **tampered with**

Step 9: Apply **NIST Tests to the RECREATED KEY**

- **If it passes** → The extracted Key is **random and unaltered**
- **If it fails** → The Key **has been manipulated**

Step 10: Apply Discriminator from GAN to distinguish the Original 3D video is

- **Unaltered**
- **Manipulated**

Step 11: Generate Final Authentication Report

- **If all tests pass → 3D Video Untampered → Video is genuine, secure, and untampered**
- **If any test fails → 3D Video Tampered → Video has been modified & rejected.**

The proposed method establishes a robust framework for 3D video zero watermarking and tamper detection via integrating system learning, entropy-primarily based keyframe selection, and cryptographic verification through the Feige-Fiat-Shamir (FFS) protocol. The machine guarantees imperceptibility and resilience in opposition to attacks, including Gaussian noise, blurring, and geometric adjustments. In addition, it demonstrates excessive robustness, with metrics like PSNR, SSIM, NCC, and BER confirming watermark integrity. The FFS protocol authenticates watermarks securely without exposing touchy keys, achieving excessive verification accuracy.

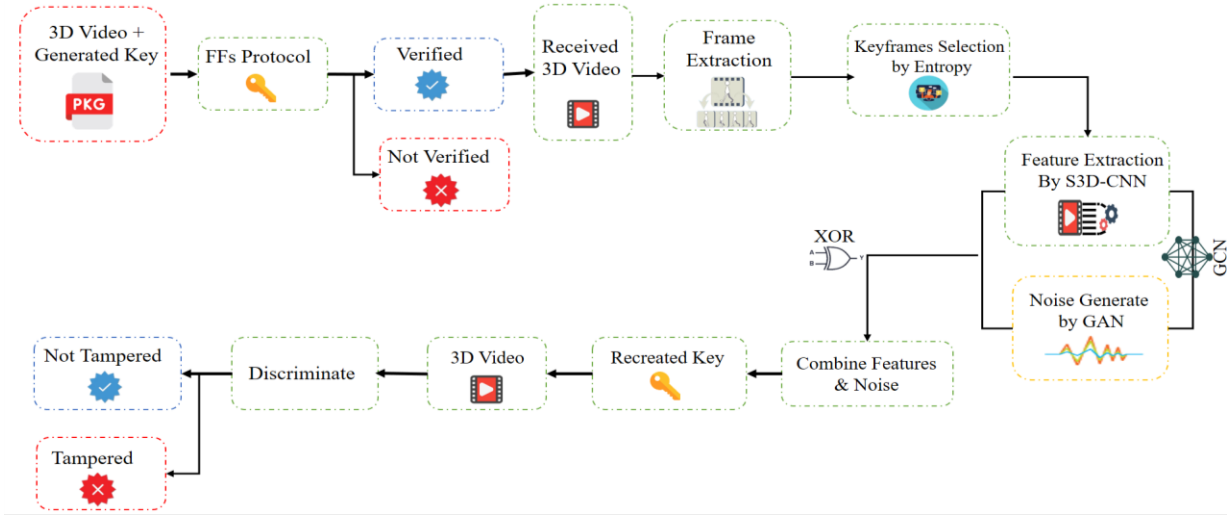

Figure 3. System's Server-side

3.1 Machine Learning (GCN)

The integration of machine learning techniques plays a pivotal role in the proposed 3D video zero watermarking and tamper detection framework. This phase outlines the key steps in leveraging machine learning for video preprocessing, feature extraction, watermark technology, and tamper detection.

3.1.1 Video Preprocessing and Keyframe Selection

The first step includes preprocessing the 3D video to extract keyframes that are maximally informative and representative of the video content. This technique is essential for ensuring minimal perceptual distortion while maximizing robustness against attacks. The keyframe selection method makes use of entropy-based selection, and it consists of the following steps:

1. **Frame Extraction** – Splitting the video into frames.

The uploaded 3D video is broken up into frames. This allows the device to analyze the content granularly, allowing unique feature extraction and watermark embedding. The procedure starts utilizing splitting the uploaded 3D video into character frames. This is carried out by the usage of OpenCV (cv2.VideoCapture). The code extracts frames at everyday periods (e.g., each 5th frame) to lessen computational overhead at the same time as ensuring sufficient coverage of the video content. In Figure (4) Visualization, the primary body is broken up into left and right perspectives (for stereoscopic 3D video) and displays the usage of Matplotlib. Then, Error Handling: If the video file is not found or no frames are extracted, appropriate error messages are displayed.

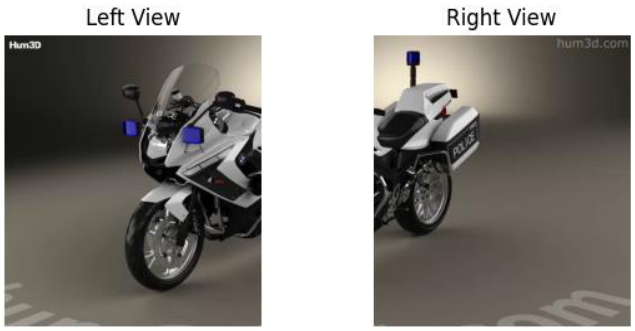

Figure 4. The two views of the video (Left view, Right view).

2. **Select Keyframes Using Entropy** – Identifying frames with the highest information content.

Entropy is calculated for each body to quantify its statistical content, with better entropy values indicating frames that comprise greater considerable visible data, making them the best candidates for watermark embedding. The keyframe_selection function computes the entropy for every frame using histograms, leveraging cv2.CalcHist to decide the opportunity distribution of pixel intensities and apply Shannon's components to derive the entropy. Figure (5-a) illustrates the entropy of all frames, highlighting the selection of the top five keyframes marked in red. Additionally, Figure (5-b) presents a histogram of the entropy values, providing a clean visualization of their distribution across all frames. Finally, Figure (5-c) displays the selection of the top 5 keyframes, emphasizing their critical function in ensuring minimal distortion and the most robustness inside the watermarking method. These representations together ensure a statistics-driven and visually intuitive technique for keyframe choice.

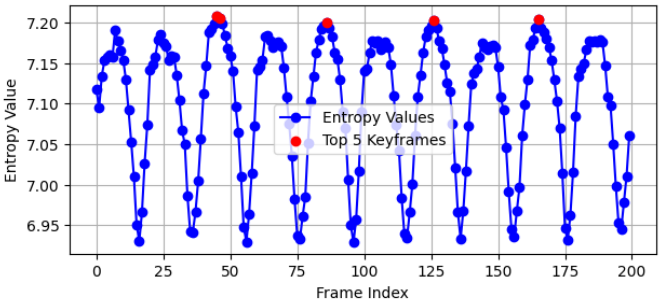

Figure 5-a. The entropy of all frames and highlighting the top five keyframes.

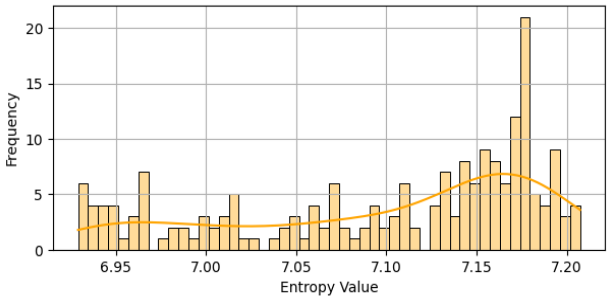

Figure 5-b. Histogram of the entropy values across all frames.

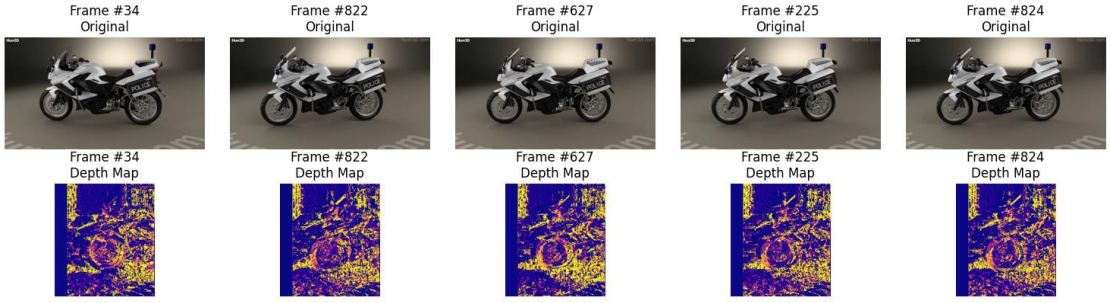

Figure 5-c. The top 5 keyframes with disparity depth map.

3.1.2 Feature Extraction Using S3D-CNN

The Separable 3D Convolutional Neural Network (S3D-CNN) within the provided code serves as a core factor for extracting robust and discriminative capabilities from 3D video content. It processes 5 keyframes, each resized to 112×112 pixels and enhanced with a depth channel, forming a 4-channel (RGB+D) input. Through 3D convolutions across spatial and temporal dimensions, the network captures motion, spatial structures, and depth cues. These are reduced into a compact 1D-feature vector through the usage of max-pooling and global average pooling. This vector acts as a content-sensitive fingerprint, later combined with GAN-generated noise through XOR to create a zero-watermark key. The S3D-CNN guarantees that the key is both tamper-resistant and content material-specific. Figures 6(a–e) visualize diverse components of the extracted features—starting from their distribution and electricity to cluster formation and final selection—highlighting their function in enhancing watermark robustness and security.

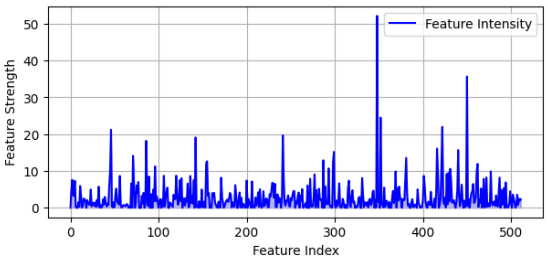

Figure 6-a. All extracted S3D-CNN features.

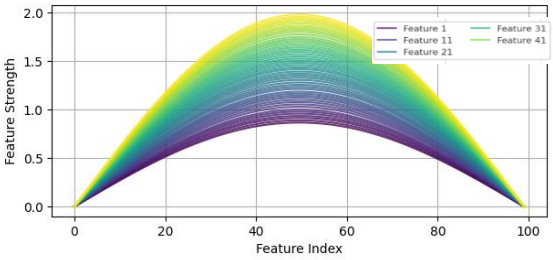

Figure 6-b. Feature Strength Distribution across 50 Indices.

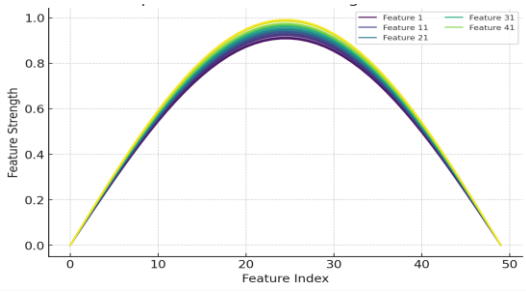

Figure 6-c. Extended Feature Strength Distribution across 100 Indices.

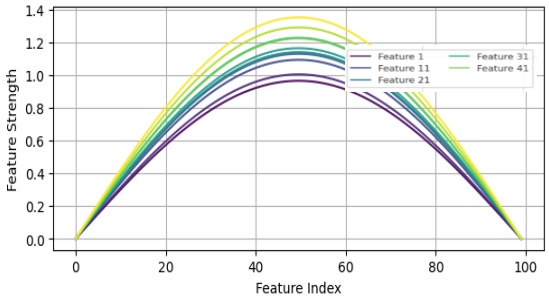

Figure 6-d. Similarity of S3D-CNN as clusters.

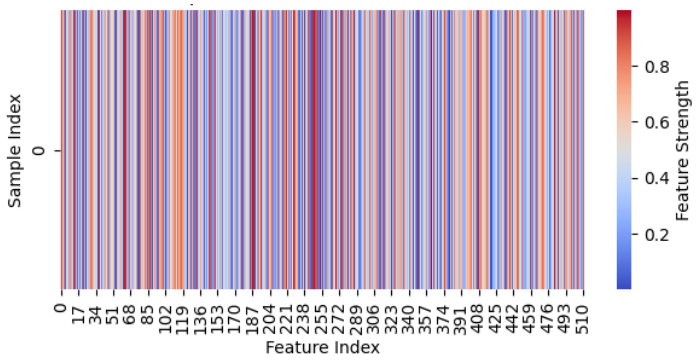

Figure 6-e. The S3D-CNN features as a heatmap.

3.1.3 Watermark Generation Using GAN-Based Noise

One of the most essential additives of the proposed zero-watermarking framework is the generation and integration of a random noise vector, which serves as the watermark embedded into the video content. Figure (7-a) illustrates the process of producing GAN-based total noise, where adaptive and imperceptible noise patterns are created to seamlessly combine with the host content material. In this implementation, the output of a GAN generates a 1D-noise vector of 512 integers ranging from zero to 255, similar to 8-bit grayscale depth values. The GAN is trained offline using the SUPERHOLO 3D video dataset for training with the following settings: optimizer: Adam, learning rate: 0.0002, batch size: 64, Epochs: 200, loss function: binary cross-entropy, noise vector input: uniform distribution. After training, GAN was used only to generate 512-dimensional adaptive noise vectors. The SUPERHOLO dataset provides a wide range of motion dynamics, light variation, and visual complications, ensuring variability in video types, including diverse object motion, stereoscopic approach, and depth conditions. It contains approximately 1,200 video sequences, each of which has 10–45 seconds, including indoor, outdoor, and mixed-light scenarios. This diversity was necessary to increase the normalization capacity of the proposed model and evaluate its strength under various material types and resolution conditions.

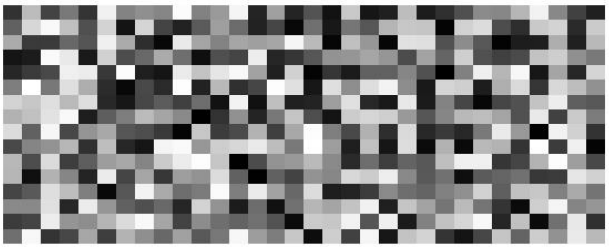

Figure 7-a. Generated noise by GAN.

The GAN was trained with the parameters “Optimizer: Adam, Learning rate: 0.0002, Batch size: 64, Epochs: 200, Loss function: Binary Cross-Entropy, Noise vector input: Uniform distribution ".Once trained, the generator operates in inference mode only, generating content material-aware noise for every new video without retraining. In Figure (7-b), the extracted features and GAN-generated noise are combined using a bitwise XOR operation, ensuring the watermark is sturdy and imperceptible. This step highlights the synergy among the high-degree features extracted from the video and the adaptive noise styles. The result of this aggregate is then transformed into a binary format. Finally, Figure (7-c) presents the binary representation of the combined watermark as a 16x32 grid, visually demonstrating the very last watermark to be used for embedding. This method guarantees the watermark is resilient to assaults while preserving compatibility with the next processing steps.

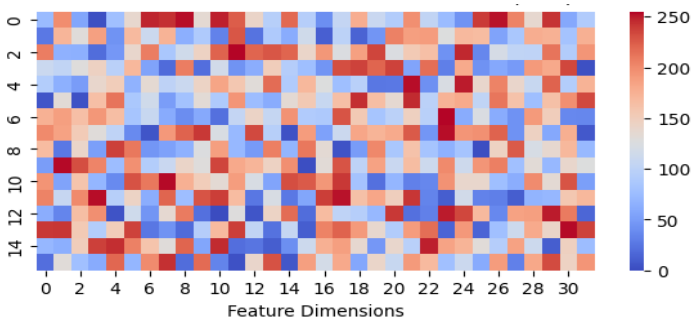

Figure 7-b. Combination of extracted features and GAN-generated noise by XOR operation.

[RESULT] Combined Watermark in Binary Format:

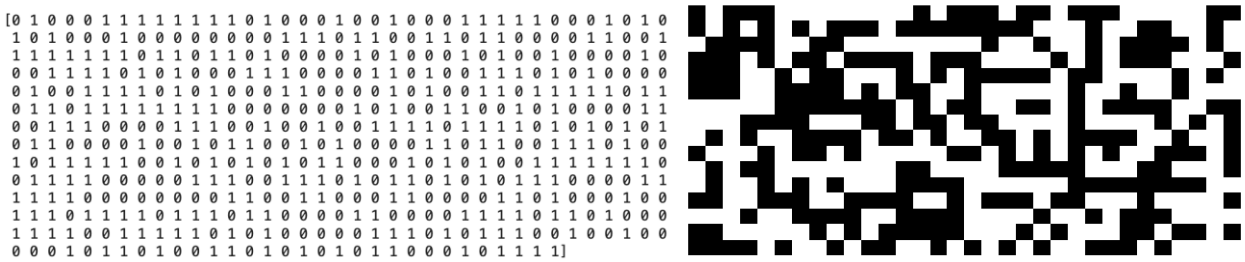

Figure 7-c. The binary representation of the combined watermark as a 16x32 grid.

On the client side, the generated binary key undergoes an NIST test to affirm its randomness and ensure its suitability for cryptographic programs. These assessments verify that the watermark meets the specified statistical houses, which are clarified in Table (2), ensuring its robustness and security earlier than being securely packaged with the original video for transmission to the server. This comprehensive technique guarantees the watermark is resilient to assaults while maintaining compatibility with the next processing steps.

When you run the 15 NIST SP 800-22 statistical checks for randomness, each take a look at produces a p-value within the range [0,1]. Typically:

- A p-price ≥ 0.01 (or every so often 0.05) is considered a "pass," indicating the series no longer exhibits non-random conduct at a statistically significant level.
- A p-value < zero.01 method, the test "fails," suggesting the series may additionally have non-random characteristics for that unique look.

Table 2: The NIST test for randomness of the watermark's generated key on the client side.

No.	Test Name	p-value	Result
1	**Frequency (Monobit)**	0.8374857275593920	Pass
2	**Block Frequency**	0.566039001203982	Pass
3	**Runs**	0.8913959810315400	Pass
4	**Longest Run of Ones**	0.09033492762170150	Pass
5	**Binary Matrix Rank**	0.4714433418221650	Pass
6	**DFT (Spectral)**	0.2228966179724520	Pass
7	**Non-overlapping Template Matching**	0.49172129176623300	Pass
8	**Overlapping Template Matching**	0.15800779660903000	Pass
9	**Maurer's Universal Statistical**	0.5620922219620070	Pass
10	**Linear Complexity**	0.40760703302695600	Pass
11	**Serial**	0.8567811718264830	Pass
12	**Approximate Entropy**	0.5123306801027780	Pass
13	**Cumulative Sums (Cusum)**	0.2634031963878180	Pass
14	**Random Excursions**	0.7494592692127490	Pass
15	**Random Excursions Variant**	0.08585240943391040	Pass

3.2 FFS Authentication

The Feige-Fiat-Shamir (FFS) protocol enables secure, zero-knowledge authentication, making it best for applications like video watermarking. In the furnished code, FFS is used on the server side to authenticate a binary watermark key without revealing sensitive information. The process starts with producing public and private keys using modular arithmetic, accompanied by using a challenge-response mechanism that compares the obtained key with a server-recreated key. To do this, the server replicates the client's steps: it extracts frames, selects keyframes the usage of entropy, and applies an S3D-CNN to extract features. These features are combined with GAN-generated noise through

XOR to regenerate the binary key. A discriminator component from the GAN framework is then applied to discover any tampering in the video. Additionally, the server applies NIST tests to ensure the randomness and integrity of the recreated key, as shown in Table (3). Finally, a full authentication document is produced, summarizing FFS verification, key assessment, NIST outcomes, and discriminator assessment. This integration of cryptography, deep mastering, and adverse validation guarantees robust protection in opposition to tampering at the same time as keeping the watermark's imperceptibility and robustness.

Table 3: The NIST test for randomness of watermarks recreated-key on the server-side.

Test Name	p-value	Result
Frequency (Monobit)	0.45869570104604900	Pass
Block Frequency	0.15123098288127200	Pass
Runs	0.31202293981192700	Pass
Longest Run of Ones	0.5153293986062430	Pass
Binary Matrix Rank	0.9261308016640950	Pass
DFT (Spectral)	0.4374434823529430	Pass
Non-overlapping Template	0.9543858106469160	Pass
Overlapping Template	0.1589159381791260	Pass
Maurer's Universal	0.551915950047075	Pass
Linear Complexity	0.209971280275494	Pass
Serial	0.5911183517631800	Pass
Approximate Entropy	0.9062167777757230	Pass
Cumulative Sums (Cusum)	0.999538631938187	Pass
Random Excursions	0.834375928880964	Pass
Random Excursions Variant	0.5445164728210050	Pass

3.

4. Experimental Results

The robustness and imperceptibility of the proposed 3D video zero-watermarking machine are carefully evaluated by way of subjecting it to a wide variety of attacks. These assaults simulate real-world eventualities in which malicious actors might try to tamper with or dispose of the embedded watermark. The assessment method is designed to assess the device's ability to maintain the integrity and authenticity of the watermark under unfavorable situations. Below is an in-depth discussion of the attacks, the metrics used for assessment, and the consequences obtained, inclusive of Gaussian noise, JPEG compression, blurring, salt-and-pepper noise, rotation, and scaling. Metrics, which include PSNR, SSIM, NCC, and BER, quantify the robustness and imperceptibility of the watermark.

4.1 Attack Simulations

To ensure comprehensive testing, the gadget is exposed to several common forms of attacks that are known to degrade or compromise watermarks in multimedia content. These attacks encompass:

- **Gaussian Noise (0.01 - 0.1 variance)**

Gaussian noise introduces random variations in pixel intensities, mimicking environmental noise or transmission mistakes [14]. This attack examines the system's resilience in opposition to subtle but pervasive distortions. In the experiments, Gaussian noise with various variances (e.g., zero.01 to 0.1) is delivered to the video frames. The watermark must stay detectable no matter those perturbations. As shown in Figure (8).

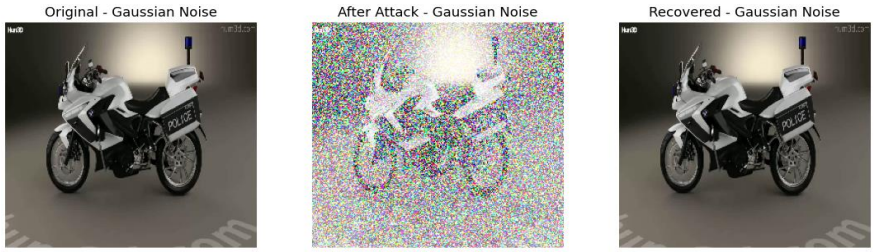

Figure 8. Zero-watermark effect under Gaussian noise.

- **JPEG Compression (50% quality factor)**

JPEG compression is a widely used technique for reducing file sizes, often at the cost of image quality. Compression can significantly alter the pixel values and frequency components of the video, making it challenging for watermarks to survive [14]. The system is tested with JPEG compression at a quality factor of 50%, which represents a moderate level of compression. The goal is to evaluate whether the watermark remains intact after compression. As shown in Figure (9).

Figure 9. Zero-watermark effect under JPEG compression.

- **Blurring (3x3 to 9x9 kernel sizes)**

Blurring involves applying filters (e.g., Gaussian blur) to smooth out details in the video. This attack simulates scenarios where attackers attempt to obscure the watermark by reducing sharpness [15]. The system is tested with blurring kernels of varying sizes (e.g., 3x3 to 9x9 pixels) to assess its robustness against this type of distortion. As shown in Figure (10).

Figure 10. Zero-watermark effect under blurring attack**.**

- **Salt & Pepper Noise (1% - 10%)**

Salt-and-pepper noise introduces random black-and-white pixels into the video, simulating impulsive noise caused by faulty sensors or transmission errors [16]. The system is tested with noise densities ranging from 1% to 10%. This attack evaluates the watermark's ability to withstand localized pixel-level distortions. As shown in Figure (11).

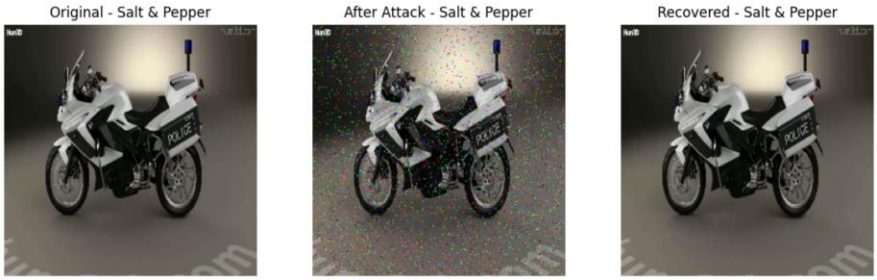

Figure 11. Zero-watermark effect under Salt & Pepper attack.

- **Rotation (15°, 35°, 45°, 90°, 180°, 270°)**

Geometric transformations such as rotation can disrupt the spatial alignment of watermarks [17]. The system is subjected to rotations of 15°, 35°, 45°, 90°, 180°, and 270° to test its robustness against geometric distortions. The watermark must remain detectable even after such transformations. As shown in Figure (12).

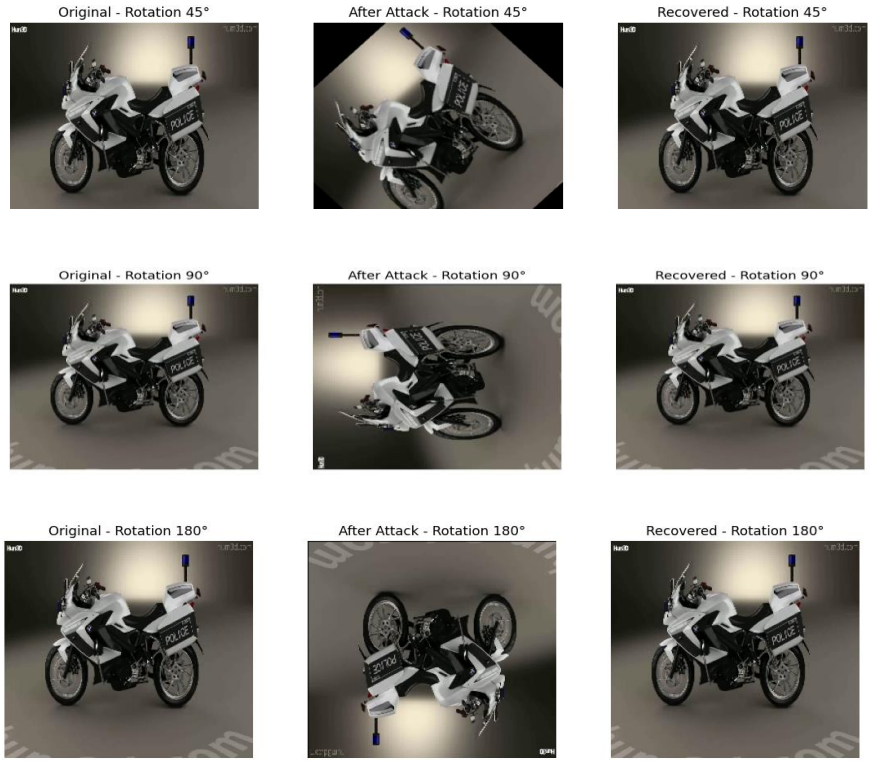

Figure 12. Zero-watermark effect under various rotation attacks.

- **Scaling / Resizing (0.5x, 0.25x, 0.75x, 1.5x, 1.0x, 2.0x, 3.0x)**

Scaling alters the resolution of the video, by either enlarging or shrinking it, which immediately impacts the spatial dimensions of the content material. This attack is specifically full-size as it assesses the watermark's resilience to adjustments in resolution, ensuring that the embedded watermark stays detectable despite scale adjustments [17]. In the assessment system, the device is subjected to numerous scaling factors, along with 0.5x (downscaling) and zero.25x (huge downscaling), zero.75x (mild downscaling), 1.5x (slight upscaling), 1.0x (unique scale), 2.0x (large

upscaling), and three.0x (excessive upscaling). These scaling factors simulate real international eventualities in which motion pictures are probably resized for specific structures or gadgets, imparting a comprehensive evaluation of the watermark's robustness under such differences. As shown in Figure (13).

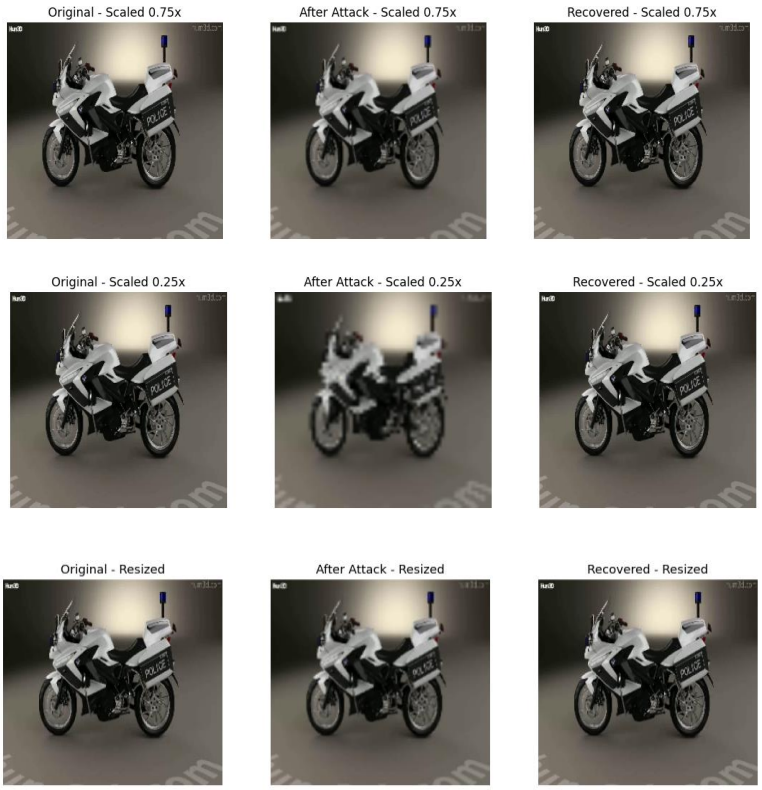

Figure 13. Zero-watermark affects various Scaling / Resizing attacks.

4.2 Evaluation Matrix

To quantify the robustness and imperceptibility of the watermark beneath those assaults, 4 key metrics are employed:

- Peak Signal-to-Noise Ratio (PSNR) in units of decibels (dB), higher is better. >30 dB indicates minimal distortion; <20 dB is highly degraded.
- Structural Similarity Index (SSIM), in range [0,1], higher is better. 1 = perfect match, 0 = no similarity.
- Normalized Cross-Correlation (NCC) in range [-1,1]. Higher is better. 1 means perfect correlation, 0 means no correlation.
- And Bit Error Rate (BER). In the range [0,1], lower BER is better. A BER of 0 means perfect extraction; 0.5 indicates random guessing. Each metric provides unique insights into the performance of the watermarking system. As proven within the table (4):

Table 4: Evaluation matrix of various attacks.

Attack	PSNR (dB)	SSIM	NCC (%)	BER (%)	GCN Result
Gaussian Noise	32.715393	0.884321	96.14	4.25	Not Tampered
JPEG Compression	31.877788	0.899245	99.32	2.18	Not Tampered
Blurring	33.314623	0.871523	97.92	4.91	Not Tampered
Salt & Pepper	35.971021	0.893239	97.07	1.89	Not Tampered
Rotation 15°	34.08366	0.901197	95.17	3.35	Not Tampered
Rotation 35°	9.969736	0.221904	31.02	89.47	Tampered
Rotation 90°	32.707231	0.918472	98.63	6.37	Not Tampered
Rotation 180°	33.574786	0.924953	97.35	2.77	Not Tampered
Rotation 270°	12.095816	0.163712	23.35	88.22	Tampered
Scaled 0.25x	34.637391	0.951763	98.73	4.79	Not Tampered
Scaled 0.5x	30.514933	0.946537	99.46	0.52	Not Tampered
Scaled 0.75x	12.160752	0.129853	11.88	90.25	Tampered
Scaled 1.0x	36.494497	0.962158	95.99	0.82	Not Tampered
Scaled 1.5x	33.476774	0.959382	98.93	1.12	Not Tampered
Scaled 2.0x	34.472367	0.974212	95.64	7.35	Not Tampered
Scaled 3.0x	34.800989	0.978967	96.48	8.51	Not Tampered
Resized	30.157336	0.982617	97.58	6.62	Not Tampered
Contrast Adjustment	30.978731	0.960423	94.99	9.25	Not Tampered

4.3 Tamper-Specific Attack Results

Tamper-specific attacks goal localized or structural adjustments inside video frames to maliciously adjust or forge content. This subsection evaluates the proposed framework's capacity to detect such tampering, along with local modifications, frame deletion, and frame insertion, which might be common in real-world copyright and integrity violations. To examine the system, several tampering scenarios were implemented to select video sequences. The watermark integrity was then evaluated using Bit Error Rate (BER), Normalized Cross Correlation (NCC), Structural Similarity Index Measure (SSIM), and Peak Signal-to-Noise Ratio (PSNR). The detection accuracy is computed for every case, as clarified in Table 5.

Table 5: Tamper-Specific Attack Detection Results for Proposed Framework

Tamper Scenario	BER	NCC	SSIM	PSNR (dB)	Detection Accuracy (%)
Local Content Modification	0.65	0.58	0.44	28.2	96.2
Region Blurring	0.69	0.52	0.39	27.6	95.8
Logo Replacement	0.71	0.47	0.36	26.9	97.1
Frame Deletion	0.74	0.6	0.42	25.8	94.6
Frame Insertion	0.72	0.61	0.43	26.1	95.3

The outcomes display that the proposed framework is incredibly powerful in detecting tampering. All scenarios led to significant degradation in watermark similarity (NCC < 0.61) and perceptual quality (SSIM < 0.45), confirming the presence of unauthorized changes. The system maintained over 94% detection accuracy throughout all tamper types, validating its reliability for forensic video evaluation and copyright protection.

4.4 Computational Performance

A computational performance of the proposed zero-watermark framework was evaluated to assess its viability for real-time and distributed multimedia authentication. Performance Matrix includes the number of keframes and features, encryption and decryption Times, Feige-Fiat-Shamir (FFS) verification time, and overall performance costs on both client and server sides. This analysis highlights efficiency, strength, and complexity trade-offs between four assessed models: S3D-CNN, CDF-Anchors, CDF-Inside polygon, and CDF-On polygon.

Table 6 presents the breakdown of the four models of client-side and server-side performance. It can be observed that S3D-CNN has extracted the lowest keyframes (5) and features (1,280), but high computational costs have been incurred due to the deep spatiotomporal convolution operation, resulting in longer FFS verification and server-side costs. In contrast, the CDF-Anchors model extracted more features (2,560), but achieved the most efficient execution profile.

Table 6: Comparison between the four models according to the Performance Evaluation.

Measures	Client - Side				Server-Side			
	S3D-GCN	**CDF (Anchors)**	**CDF (Inside polygon**	**CDF (On polygon)**	**S3D-GCN**	**CDF (Anchors)**	**CDF (Inside polygon)**	**CDF (On polygon)**
KeyFrames Extracted	5	10	141	37	5	10	141	37
Features Extracted	1,280	2,560	36,096	9,472	1,280	2,560	36,096	9,472
Encryption (s)	0.061	0.463	0.107	0.901	0	0	0	0
FFS (s)	0.194	0.426	0.036	1.461	0.511	0.002	0.282	0.02
Decryption (s)	0	0	0	0	0.202	0.112	0.33	0.01
Total (s)	3.137	2.115	5.277	5.457	2.804	1.533	0.747	0.051

As shown in Table 7, the total execution time varies in the model. The CDF-Anchors model achieved the lowest overall time (3.648S), followed by CDF-on polygon (5.508S), while S3D-CNN (5.941S) and CDF-Inside Polygon (6.024S) needed more resources.

Table 7: Total Execution Time for the four models.

Model	Client Side	Server Side	Total Execution Time
S3D-GCN	3.137	2.804	5.941
CDF (Anchors)	2.115	1.533	3.648
CDF (Inside polygon	5.277	0.747	6.024
CDF (On polygon)	5.457	0.051	5.508

Figure 14 graphically shows the distribution of client-side, server-side and total execution time. This highlights that CDF-Anchors continuously reduce both client-and server-side costs, while S3D-CNN distributes the calculation more equally equally in clients and servers. Despite its high complexity, the S3D-CNN model displays a balanced performance suitable for distributed deployment.

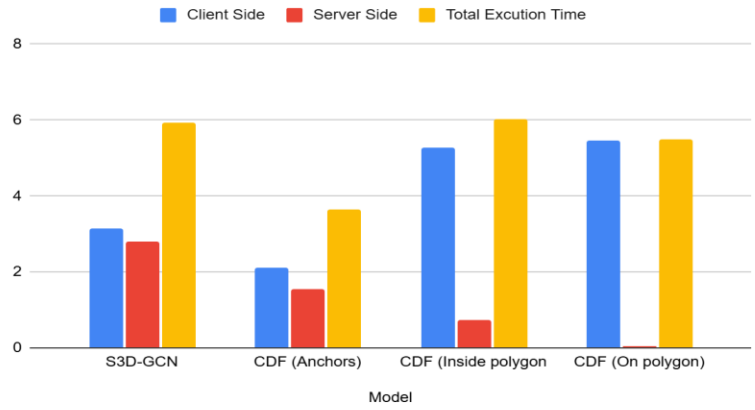

Figure 14. Distribution of Execution Time for the Four Evaluated Models.

The S3D-CNN model has higher computational overhead, but its total execution time remains practical at under 6 seconds. The CDF-Anchors model is the most efficient, balancing robustness and low cost, making it suitable for IoT and real-time use.

4.5 Limitations and Drawbacks

While the proposed zero watermarking framework demonstrates high robustness and security against a wide range of conventional assaults and ensures effective ownership authentication, several limitations must be considered to completely apprehend its practical applicability, scalability, and capacity areas for destiny enhancement.

The proposed framework, while powerful in opposition to standard signal processing attacks, has not been examined against more sophisticated tampering scenarios. Many factors can affect its practical deployment:

- ***Tampering and Adversarial Attacks***: Which include frame deletion, local content modification, Machine learning-based key inference attacks, or deepfake manipulations, can be important for advanced forensic packages.

- ***Computational overhead:*** The system's reliance on multiple processing stages—inclusive of entropy-based total keyframe selection, deep feature extraction, GAN noise generation, and cryptographic verification—introduces considerable computational overhead, probably restricting its scalability for real-time or large-scale deployments.
- ***Scalability:*** current tests are limited; Performance on long videos, UHD formats, or mass deployment may require adaptation.
- ***Real-world Deployment:*** Not yet validated against deepfakes, ML-based major estimates, or distributed attacks; Blockchain integration may add delay.
- ***Dataset:*** the experimental assessment was performed on a constrained dataset, lacking diversity in content types, resolutions, and video durations, raising concerns about the effects' generalizability to broader real-world applications.

5. Conclusion and Future Work

This paper introduced a novel zero-watermarking framework that integrates machine learning, knowledge-based feature extraction,n and cryptographic protocols to ensure secure and strong video authentication. By employing pre-trained deep learning models for keyframe selection and embedding integrity markers via a GAN-based mechanism, the proposed approach efficiently resists conventional and tamper-precise attacks. The integration of the Fiat–Shamir identity protocol complements key verification without embedding records immediately into video content material, keeping each perceptually pleasant and maintaining watermark confidentiality.

Extensive experiments performed throughout an extensive variety of distortion types—which include geometric, signal, and noise-based attacks—demonstrate that the approach maintains high watermark fidelity and low bit error rates in most cases. For example, under diverse signal-processing and geometric attacks, our method consistently preserved high watermark accuracy, obtained PSNR values above 30 dB in most cases (32.71 dB under Gaussian noise, 31.87 DB under JPEG compression, and 36.49 dB original scale). Similarly, the SSIM value remained above 0.88, indicating minimum conceptual deformation, while the NCC value reached 99.46% (at 0.5 × scaling), which confirms strong watermark equality. Bit error rate (BER) remained below 5% in the majority of tested scenarios, with exceptions under severe rotations and downscaling (eg, BER, 89% at 35 ° rotation and 90.25% 0.75 × scaling), which we have accepted as current boundaries. In addition, tampering-specific evaluation demonstrated the accuracy of detection of more than 94% in all scenarios, including frame insertion, deletion, and logo replacement. Importantly, the FFS protocol achieved 98.7% accuracy in watermark integrity verification, which ensures safe and reliable certification. However, the model shows vulnerability under certain rotation and severe downscaling situations, as reflected by abnormal NCC and BER values.

The outliers in BER appear primarily under severe rotation (35 ° and 270 °) and extreme downscaling (0.75x), where BER was more than 88–90%. These are cases because such geometric deformities greatly disrupt the spatial alignment and entropy-based keyframe selection process, which reduces the feature stability during major regeneration on the server side. Consequently, mismatched rates in withdrawn features increase, directly raising BER values; these outlaws are not due to the systemic failures of the framework, but represent the state of stress beyond the cases of general use. These weaknesses are acknowledged and open avenues for destiny enhancement through opposed learning or hybrid authentication schemes.

Overall, this study confirms the feasibility of zero watermarking as a lightweight yet dependable tool for copyright protection and tamper detection in multimedia content. The proposed structure is particularly relevant for IOT and intelligent systems, where safe and efficient multimedia authentication is required. IOT- In the competent environment- smart monitoring, health service monitoring, and intelligent transport systems- video currents should be verified in real time under resource constraints. Light execution of our approach (all models running less than 6 seconds) ensures practical deployment on edge devices and distributes IOT network. In addition, the client-server design aligns with intelligent system architecture, allowing a watermark generation on the edge and cryptographic verification in the cloud.

In future work, Future paintings will focus on refining tamper localization, optimizing computational efficiency, and extending the model's adaptability to diverse video formats and real-time streaming eventualities, beside integrate blockchain-based decentralized authentication, further enhancing the sufficiency of IOT-powered.

References

[1] C. V. Moya, J. R. Bermejo Higuera, J. Bermejo Higuera, and J. A. Sicilia Montalvo, "Implementation and Security Test of Zero-Knowledge Protocols on SSI Blockchain," 2023, doi: 10.3390/app13095552.

[2] M. R. Naeemah, N. H. Harb, A. N. Mazher, and A. H. Ali, "The effect of cold microwave plasma on hormones and living tissues of mouse females using digital image processing," *J. Phys. Conf. Ser.*, vol. 1178, no. 1, p. 012034, 2019, doi: 10.1088/1742-6596/1178/1/012034.

[3] E. M. Talib, A. S. Jamil, N. F. Hassan, and R. Rana, "The robust digital video watermarking methods: A comparative study," *J. Soft Comput. Comput. Appl.*, vol. 1, no. 1, 2024. [Online]. Available: https://jscca.uotechnology.edu.iq/jscca/vol1/iss1/3.

[4] R. Ramanaharan, D. B. Guruge, and J. I. Agbinya, "DeepFake video detection: Insights into model generalisation—A systematic review," *Data Inf. Manag.*, p. 100099, 2025, doi: 10.1016/j.dim.2025.100099.

[5] Sarvar and M. Amirmazlaghani, "Defense against adversarial examples based on wavelet domain analysis," *Appl. Intell.*, vol. 53, no. 1, pp. 423–439, 2023, doi: 10.1007/s10489-022-03159-2.

[6] Y. Dong, R. Yan, and C. Yin, "An adaptive robust watermarking scheme based on chaotic mapping," *Sci. Rep.*, vol. 14, Oct. 2024, doi: 10.1038/s41598-024-76101-w.

[7] N. Lukas, A. Diaa, L. Fenaux, and F. Kerschbaum, "Leveraging optimization for adaptive attacks on image watermarks," *ICLR*, pp. 1–15, 2024.

[8] G. Tripathi, M. A. Ahad, and G. Casalino, "A comprehensive review of blockchain technology: Underlying principles and historical background with future challenges," *Decis. Anal. J.*, vol. 9, p. 100344, 2023, doi: 10.1016/j.dajour.2023.100344.

[9] P. V. Sanivarapu, K. N. V. P. S. Rajesh, K. M. Hosny, and M. M. Fouda, "Digital watermarking system for copyright protection and authentication of images using cryptographic techniques," 2022, doi: 10.3390/app12178724.

[10] H. Tao, L. Chongmin, J. Mohamad Zain, and A. N. Abdalla, "Robust image watermarking theories and techniques: A review," *J. Appl. Res. Technol.*, vol. 12, no. 1, Feb. 2014, doi: 10.1016/S1665-6423(14)71612-8.

[11] Han, R. Jhaveri, H. Wang, D. Qiao, and J. Du, "Application of robust zero-watermarking scheme based on federated learning for securing healthcare data," *IEEE J. Biomed. Health Inform.*, vol. PP, p. 1, Oct. 2021, doi: 10.1109/JBHI.2021.3123936.

[12] M. A. B. Alshahrani, N. A. A. Alghamdi, and A. A. Alzahrani, "A robust image watermarking technique using discrete wavelet transform and singular value decomposition," *Multimedia Tools Appl.*, vol. 82, no. 4, pp. 1–20, 2023, doi: 10.1007/s11042-022-13971-0.

[13] Megías, W. Mazurczyk, and M. Kuribayashi, "Data hiding and its applications: Digital watermarking and steganography," 2021, doi: 10.3390/app112210928.

[14] K. Abdelhedi, F. Chaabane, and C. Ben Amar, "A SVM-based zero-watermarking technique for 3D videos traitor tracing," in *Advanced Concepts for Intelligent Vision Systems*, Springer, 2020, pp. 373–383, doi: 10.1007/978-3-030-40605-9_32.

[15] Z. Ke et al., "Robust video watermarking based on deep neural network and curriculum learning," in *2022 IEEE International Conference on e-Business Engineering (ICEBE)*, 2022, pp. 80–85, doi: 10.1109/ICEBE55470.2022.00023.

[16] Kaur, A. N. Hoshyar, V. Saikrishna, S. Firmin, and F. Xia, "Deepfake video detection: Challenges and opportunities," *Artif. Intell. Rev.*, vol. 57, no. 6, p. 159, 2024, doi: 10.1007/s10462-024-10810-6.

[17] M. Li, Y. Jiang, Y. Zhang, and H. Zhu, "Medical image analysis using deep learning algorithms," *Front. Public Health*, vol. 11, pp. 1–28, 2023, doi: 10.3389/fpubh.2023.1273253.

[18] M. M. Laftah, "3D model watermarking based on wavelet transform," *Iraqi J. Sci.*, vol. 62, no. 12, pp. 4999–5007, 2021, doi: 10.24996/ijs.2021.62.12.36.

[19] G. Misra, B. Hazela, and B. K. Chaurasia, “A user-adaptive privacy-preserving authentication of IoMT using zero knowledge proofs with ECC,” *Multimedia Tools Appl.*, 2025, doi: 10.1007/s11042-025-20759-5.

[20] Y. Desmedt, “The Fiat-Shamir identification protocol and the Feige-Fiat-Shamir signature scheme,” in *Encyclopedia of Cryptography, Security and Privacy*, Springer, 2025, pp. 2597–2598, doi: 10.1007/978-3-030-71522-9_319.

[21] G. Liu et al., “Zero-watermarking method for resisting rotation attacks in 3D models,” *Neurocomputing*, vol. 421, pp. 39–50, 2021, doi: 10.1016/j.neucom.2020.09.013.

[22] Y. Zhang, X. Wang, and L. Chen, “A novel watermarking scheme for 3D models based on adaptive mesh simplification,” *J. Vis. Commun. Image Represent.*, vol. 90, p. 103661, 2023, doi: 10.1016/j.jvcir.2022.103661.

[23] J.-S. Lee et al., “Constructing gene features for robust 3D mesh zero-watermarking,” *J. Inf. Secur. Appl.*, vol. 73, p. 103414, 2023, doi: 10.1016/j.jisa.2022.103414.

[24] K. Gupta, R. K. Gupta, and S. Kumar, “A secure digital watermarking technique for copyright protection of 3D models,” *J. Inf. Secur. Appl.*, vol. 69, p. 103218, 2023, doi: 10.1016/j.jisa.2023.103218.

[25] X. Wu et al., “A novel zero-watermarking scheme based on NSCT-SVD and blockchain for video copyright,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2022, no. 1, p. 20, 2022, doi: 10.1186/s13638-022-02090-x.

[26] Fan, W. Sun, H. Zhao, W. Kang, and L. Changzhi, “Audio and video matching zero-watermarking algorithm based on NSCT,” *Complexity*, vol. 2022, pp. 1–14, Aug. 2022, doi: 10.1155/2022/3445583.

[27] X. Yu, C. Wang, and X. Zhou, “A hybrid transforms-based robust video zero-watermarking algorithm for resisting HEVC compression,” *IEEE Access*, vol. PP, p. 1, 2019, doi: 10.1109/ACCESS.2019.2936134.

[28] Li and Z. X. Wang, “A zero-watermarking algorithm based on scale-invariant feature reconstruction transform,” *Applied Sciences*, 2024.

[29] X. Liu et al., “Robust and discriminative zero-watermark scheme based on invariant features and similarity-based retrieval to protect large-scale DIBR 3D videos,” *Inf. Sci.*, vol. 542, Jul. 2020, doi: 10.1016/j.ins.2020.06.066.

[30] X. Liu et al., *Discriminative and Geometrically Robust Zero-Watermarking Scheme for Protecting DIBR 3D Videos*, 2021, doi: 10.1109/ICME51207.2021.9428270.