IAENG International Journal of Computer Science

# Robust and Secure Video Authentication: A Hash-Based Watermarking Approach

Preeti Saini and Rakesh Ahuja

*Abstract*—**While video is a vital component of modern communication, it also presents opportunities for manipulation and unauthorized access. The preservation and authenticity of video footage have become increasingly important as digital video data expands. Video authentication techniques ensure video integrity, confidentiality, availability, data sharing, and privacy during the data lifecycle. A video authentication technique that can prevent the copying, tampering, or alteration of video footage without permission is digital watermarking. The suggested method is notable for its ease of use, efficacy, and efficiency in handling the video authentication problem. Whereas current methods could use intricate algorithms or need much processing power, the suggested approach stands out by directly utilizing hash-based watermarking that employs SHA256 hashing and authentication. It is appropriate for real-time applications as it offers strong security against tampering while preserving computing efficiency by embedding the hash value of each odd-numbered frame into the subsequent even-numbered frame. The proposed approach is a secure and economical option for video authentication, improving the security and authenticity of video content for use in various applications.**

*Index Terms*—**Digital watermarking, forgery detection, privacy, integrity, confidentiality, Video authentication, hashing, copyright protection, and secure hash algorithm.**

## I. Introduction

TODAY various forms of multimedia information are available. The content of digital video data is more informative than digital image data. A variety of electronic devices, including cell- phones, CCTV cameras, camcorders, and digital cameras, record videos are available. Because so much information on the internet, society takes everything as gospel without questioning its veracity [1][2]. Video data may be subject to a variety of attacks. A video sequence consists of multiple frames that collectively define its regional property. Hiding video data is a popular issue in research [3]. Security measures such as safeguarding digital privacy, managing digital rights, detecting and protecting against forgeries, tracking dishonest affiliates, distributing watermarked multimedia, and identifying copies are frequently used in integrating networks or services, sharing data on social media [4], and maintaining data privacy [5]. Authentication is typically achieved through passwords and biometric scans like fingerprints, with enhanced security often involving the use of multiple authentication methods.

The video has become an essential medium in today's communication landscape; however, its digital format makes it susceptible to alteration and unauthorized use, leading to the potential spread of misinformation or the misrepresentation of significant occurrences [6]. Therefore, reliable methods for video authentication are essential. Video hashing is a critical method used to counteract video tampering, and piracy, and to authenticate video content. For authentication purposes, the system generates new verification data using a cryptographic key and checks it against the video in question. In the realm of video surveillance, maintaining the integrity and confidentiality of footage is a significant hurdle. The approach presented in this article introduces a technique for determining the authenticity of video by incorporating hash values as watermarks. This technique entails calculating the hash value of every other frame and integrating it back into the video. This allows for confirmation of the video's authenticity when needed. The paper is structured into several parts, with Section 2 discussing extensive related studies and research. Section 3 explains the core concepts and theories behind the study, whereas Section 4 details the proposed approach. Section 5 explores the experimental framework and its findings, and Section 6 concludes the paper.

## II. Literature Review

A systematic review to research and analyze the available literature in the field of video authentication in multimedia security has been conducted. The literature study highlighted three key methods for video authentication: (a) digital signature-based methods (b) spread spectrum-based methods, and (c) hash-based methods. Digital signatures offer authenticity but are susceptible to replay and transcoding assaults. Spread-spectrum-based techniques are reliable but computationally demanding. Hash-based watermarking approaches have drawn interest due to their effectiveness and simplicity. Researchers have suggested improvements to increase the robustness of hash-based watermarking approaches against attacks.

As multimedia data has proliferated, the environment of digital content authentication and security has grown more intricate. Researchers and specialists in the field have worked hard to develop new methods and systems to solve the issues provided by deliberate tampering, forgeries, and illegal access. The major contributions made by different writers in the field of multimedia authentication. It has covered a wide range of methods, such as hashing, tamper detection algorithms, and watermarking of images and videos as summarized in Table I. The different hash-based video watermarking algorithms were reviewed in the paper. A comparative analysis of different parameters for video authentication, integrity, copyright protection, and imperceptibility from the papers on hash-based video watermarking has been compiled in Table II.

Manuscript received October 30, 2023; revised July 11, 2024.

Preeti Saini is an assistant professor at Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab (140401), India. (Corresponding author, phone: 917015899692; e-mail: (preeti.saini@chitkara.edu.in).

Rakesh Ahuja is a professor at the Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab(140401), India.(phone:917906045922, e-mail: rakesh.ahuja@chitkara.edu.in).

TABLE I: Major Contributions Made By Different Researchers

| Author and Ref no | Method/Technique used | Summary |
|---|---|---|
| Janu et al. [3] | Content-based Authentication using QR Code and Arnold Transform | Authentication signature generation based on content, utilizing QR code generation and the Arnold transform |
| Yu et al. [7] | Video Data Concealment using Error Rectification | Rectification of repeated errors for area code penalties, superior data concealment, privacy of surveillance video, simulation results |
| Pitopakis et al. [8] | Watermarking and ML for Image Authentication | Use of deep neural networks to attack watermark embedding techniques, prevention of watermark removal by classifiers, attacking decision trees, blocking model extraction attacks |
| Fang et al. [9] | Image Authentication using Singular Values and LSB | Use of singular values to generate images, enhancement of original image by adding LSB of original picture bit to randomly selected pixels |
| Hosler et al. [10] | Video Forensics Database (VACID) for Authentication | Introduction of VACID, a video forensics database, methodology for creation, utility in video authentication and camera identification |
| Sajjad et al. [11] | Image Hashing using Canny Operator and Dominating Coefficients | Combining the dominating coefficients of the sampled rich edge image blocks with the difference between their positions yielded a hash value. |
| Khelifi and Bouridane [12] | Video Hashing for Content Identification using DCT and DST | Signal calibration, perceptual video hashing method based on DCT and DST |
| Maung et al. [13] | MP4 File Format Authentication Scheme | Authentication of MP4 files without quality loss, detection of manipulation signs |
| Du et al. [14] | Overview of Image Hashing-based Tamper Detection Systems | Exploration of Structure and Classifications, suggestions, and best practices |
| Kumar et al. [15] | Analysis of Digital Watermarking Methods | Summarized analyses of different methods, features, and performance, highlighting the lack of a comprehensive solution |
| Kulkarni et al. [16] | Video Watermarking using DWT | The mechanism at the ground root of DWT for watermark embedding in video frames |
| Chen et al. [17] | Video Tamper Detection using CNN and Perceptual Hashing Learning | Tamper detection method based on convolutional neural network and perceptual hashing learning |
| Tang et al. [18] | Video Hashing with DCT and NMF | Video hashing approach using DCT and Non-Negative Matrix Factorization (NMF) |
| Hasso and Taha [19] | Tamper Detection Algorithm for Video using NMF and DCT | Tamper detection algorithm based on non-negative matrix decomposition and DCT |
| Hammami et al. [20] | Blind Semi-Fragile Watermarking for Video Authentication | Proposal of a blind semi-fragile watermarking scheme for video authentication |
| Birouk et al. [21] | Evaluation of Watermarking Quality for Video Security | Evaluation of watermarking quality and functionality for two watermarks from the same video using different formats |

Table continued on next page

TABLE I – continued from previous page

| **Author and Ref no** | **Method/Technique used** | **Summary** |
|---|---|---|
| Zainol et al. [22] | Hybrid SVD Picture Watermarking Systems | Study of hybrid SVD schemes, SVD security difficulties, classification of schemes, types of embedding tactics, and comparison of SVD schemes |
| Tang et al. [23] | Perceptual Hashing (P-Hash) | P-Hash highlighted for superior perceptual resilience and authentication property |
| Chen et al. [24] | Video Hashing with 3D DWT Coefficients in LL Sub-band | Video hashing method with secondary frames, invariant moments, and 3D DWT coefficients in the LL sub-band, faster but with shorter hash length |
| Mareen et al. [25] | Video Hashing Based on Low-Rank Frames | Video hashing method based on low-rank frames, demonstrating computational efficiency and classification performance |
| Jabbar et al. [26] | Perceptual Hashes for Accelerated Detection in Medical Image Authentication | Utilization of perceptual hashes for speeding up detection in medical Image Authentication |
| Sujatha et al. [27] | Integrity and Validity of Medical Pictures with Hash and Frequency Domain Watermarking | Combination of hash with frequency domain watermarking for integrity and validity of medical pictures |
| Ma and Xing [28] | Tamper Detection using Difference Hashing (D-Hash) | Employment of D-Hash for tamper detection, emphasis on computing efficiency and detection accuracy |
| Aradhana and S. M. Ghosh [29] | Robust Video Content with Perceptual Hashing and TIRI Frame Features | Perceptual hashing for maximizing robustness of video contents using TIRI frame features, good security and resilience but with longer completion times |
| Al-Hooti et al. [30] | Reduced difference expansion | using pixels with LSB technique for hiding text and visual data, outperforming other methods |
| Shang et al. [31] | 3D Boolean CNN algorithm with Arnold transform | Better image preprocessing and encryption accelerate progress in color image-level authentication and copyright, although this algorithm's usefulness is somewhat limited. |

TABLE II: Comparison of Different Watermarking Techniques

| Parameter | Authentication | Integrity | Copyright Protection | Imperceptibility |
|---|---|---|---|---|
| Hash-based | Yes | Yes | Yes | Moderate to High |
| Spread spectrum | Yes | Yes | Yes | Moderate to High |
| DCT-based | No | Yes | Yes | Low to Moderate |
| LSB-based | No | No | Yes | High |

The research gaps that exist include the need for improved methods for video authentication, tamper detection, copyright protection, and ownership proofing, as well as the inadequacy of current strategies to survive some attacks. Despite this, the extraction of watermark patterns from a host of video multimedia objects is less robust against some of the prominent attacks like ambiguity, collusion, and frame-specific attacks. Therefore, the suggested method makes it possible to identify certain changes made to the video frames, including additions, deletions, or modifications. Any alterations to the video data lead to a discrepancy between the embedded hash values and the recalculated hashes following extraction since hash values are embedded into neighboring frames

## III. Preliminaries

The work began with the exploration of two key concepts: hashing and watermarking. These core pieces are critical to preserving data security and integrity, with hashing incomprehensibly linked to important modules such as block ciphers. Watermarking, on the other hand, emerges as an effective method for protecting intellectual property and verifying digital content via discrete markings. This serves as a guiding light, leading readers through the fundamental ideas of hashing and watermarking.

### A. Watermarking

The technique of adding a secret message or signal to digital information, such as pictures or movies, is known as watermarking. Watermarking serves as a way to identify the source and owner of the material as well as a safeguard against unauthorized use and dissemination. Watermarks can be visible or invisible, and invisible watermarks are of three types: robust, public and private, and fragile and semi-fragile [32][33]. The following is the formula for a basic watermark embedding method for a digital video frame.

$$I_w(x, y) = I(x, y) + \mu \cdot W(x, y) \tag{1}$$

where
$I(x, y) =$ the original frame at pixel coordinates (x, y)
$W(x, y) =$ the watermark frame at the same pixel coordinates
$I_w(x, y) =$ The watermarked frame
$\mu =$ Embedding strength (between 0 and 1)
Equation (2) outlines the process of adding the watermark to the original image and creating the watermarked image.

*1) Video Watermarking :* Recent advancements in image watermarking techniques have led researchers to explore new and exciting research areas, such as digital video watermarking. The video watermarking approach applies image-watermarking techniques to either uncompressed or compressed videos [34]. However, not all solutions are effective in protecting video data.

Video watermarking incorporates a secret message or signal into video material. The majority of video watermarking methods include changing the video frames or including extra information in the video file. A copyright management system is used to protect video data, establish ownership, and detect theft. The attack on a video can take several forms, including those involving scenes, pixels, blocks, frames, and shots. There are several types of attacks, including spatial tempering, temporal tempering, and spatial-temporal tampering. Tampering can result in defamation, forensic investigations, and public places [35]. An active method for forgery detection and a passive method for video forensics are commonly used to detect video manipulation. The availability, secrecy, and integrity of video surveillance systems are all guaranteed by video hashing.

### B. Hashing

In computer science, hashes are used to map arbitrary-sized data to fixed-sized values that may be applied for several security functions, such as digital signature, credential retention, and data security checks or cryptographic applications [36]. The process of hashing involves taking an input and applying a mathematical function to it, which produces a fixed-size output. Inputs to hash functions are called messages and outputs are called digests. A hash function is defined as a function h such that

$$\mathbf{h} : \mathbf{X} - > \mathbf{Y} \tag{2}$$

Where

$$X = \{x \in \{0, 1\}^j : j \in \mathbb{N}\}$$

is the set of all bit sequences of arbitrary length ‘x’ and Y = 0, 1 k is the set of all bit sequences of a specific, generally short, length k. Graphically hashing is shown in Fig1. In this model, the hash function H takes an input message X of any length and produces a fixed-size output Y. The hash value Y is unique to the input message X, meaning that even a small change to X will result in a completely different hash value.

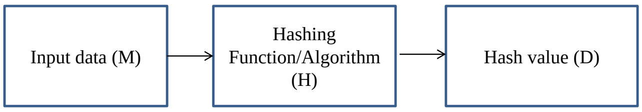


Fig. 1: Basic Process of Hashing

Hashing algorithms typically involve a combination of bitwise logic, modular arithmetic, and logical functions to transform the input message into the hash value. The specific details of each algorithm are complex and depend on the specific function being used. MD5(Message digest), SHA (secure hash algorithm) [36] [37], bcrypt (Blowfish crypt) [34], scrypt, and Argon2 are widely used in cryptography and computer science. Each technique has its strengths and weaknesses and is suited for different applications.

*1) Secure Hash Algorithm (SHA) :* SHA is a hash function that includes several variants [38] [39]. These algorithms produce hash values of varying sizes.SHA-256 is considered one of the best hashing algorithms available due to its high security and resistance to attack. It has a large 256-bit output size, which has been widely standardized, tested, and vetted. Additionally, it is relatively efficient in terms of computation time and memory usage and is widely supported by software and hardware. These factors make it a practical and popular choice for many cryptographic applications.

SHA-256 hashing algorithm is illustrated in Fig.2. It shows the main steps involved in generating the 256-bit hash value.

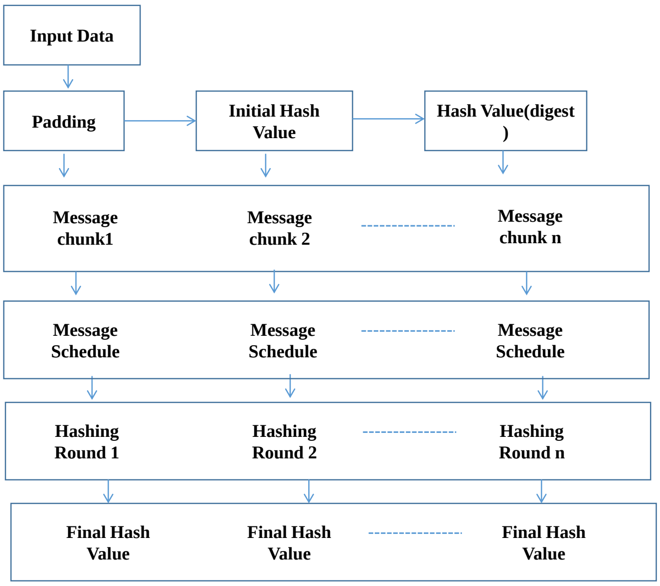
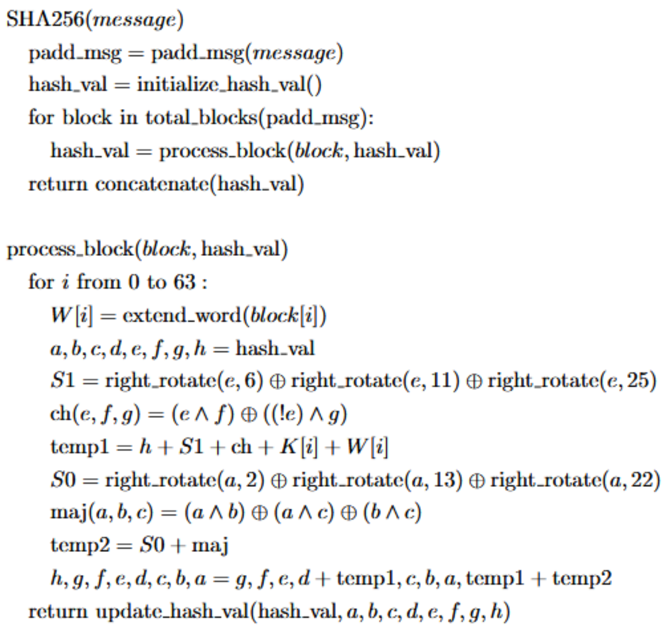


Fig. 2: Basic Working of the SHA-256 Algorithm

The input data is first padded to ensure that it is 512 bits in length. Then it is divided into 512-bit chunks. For each chunk, an initial hash value is combined with the chunk to produce an intermediate hash value. It then processes this intermediate value through 64 rounds of hashing to generate a new intermediate value for use in the next round. At the end of the 64 rounds, the final intermediate value is combined with the initial hash values to produce the final hash value, or ”digest.” This digest is a 256-bit binary value that represents the unique digital fingerprint of the input message. The pseudo-code in Fig.3 describes the fundamental workings of the SHA-256 algorithm [36] [37].



```
SHA256($message$)
    padd_msg = padd_msg($message$)
    hash_val = initialize_hash_val()
    for block in total_blocks(padd_msg):
        hash_val = process_block($block$, hash_val)
    return concatenate(hash_val)

process_block($block$, hash_val)
    for $i$ from 0 to 63 :
        $W[i]$ = extend_word($block[i]$)
        $a, b, c, d, e, f, g, h$ = hash_val
        $S1$ = right_rotate($e$, 6) $\oplus$ right_rotate($e$, 11) $\oplus$ right_rotate($e$, 25)
        ch($e, f, g$) = $(e \wedge f) \oplus ((!e) \wedge g)$
        temp1 = $h + S1 +$ ch $+ K[i] + W[i]$
        $S0$ = right_rotate($a$, 2) $\oplus$ right_rotate($a$, 13) $\oplus$ right_rotate($a$, 22)
        maj($a, b, c$) = $(a \wedge b) \oplus (a \wedge c) \oplus (b \wedge c)$
        temp2 = $S0$ + maj
        $h, g, f, e, d, c, b, a = g, f, e, d +$ temp1, $c, b, a,$ temp1 + temp2
    return update_hash_val(hash_val, $a, b, c, d, e, f, g, h$)
```

Fig. 3: Pseudocode Illustrating the Basic SHA-256 Algorithm

## IV. Proposed Methodology

Hash computation-based video authentication and digital watermarking techniques are prominent ways to confirm the veracity and integrity of video footage. This study addressed the fundamental problem of video authenticity by introducing a revolutionary video forensics approach. To authenticate and validate video integrity, the methodology employs hash computation algorithms as well as digital watermarking. Various video sources, including, have been investigated for input acquisition. The proposed technique comprises two stages: embedding and extraction. The embedding method selects odd-numbered frames, computes their hash values, and smoothly integrates them into the matching even-numbered frames. The extraction technique retrieves hash values for verification, proving the validity of each frame. This unique approach not only provides a dependable solution for video authentication, but it also has a cheap computing cost without sacrificing visual quality. The procedure is described successively below.

### A. *Sources of Videos as Input Acquisition*

Several techniques can be used to capture video footage, such as digital video cameras, smartphones, webcams, screen recording software, drones, and surveillance cameras. Digital video cameras save the footage in a digital format, while smartphones have excellent cameras. Webcams are integrated into laptops and computers, and screen-recording software captures computer screens. Some of the input films with different resolutions, frame rates, bitrates, frame widths, frame heights, and lengths used in the experiment are shown in Fig.4. A brief description of the frame properties has been given in Section 5.

### B. *Source of Watermarks*

The source of watermarks has been generated from the video frames itself. Video forensics relies on hash computation algorithms and digital watermarking methods to authenticate and verify the integrity of video data. Hash algorithms generate a unique fingerprint of a video file to detect any tampering, while digital watermarking methods embed information as a watermark in the video file to verify its authenticity and trace its source [40]. The procedures utilized in the suggested method for computing hashes and making watermarks are shown in Fig.5.

### C. *Generation of Hash values and Embedding Algorithm*

In the proposed method, the odd-numbered frames of the video have been selected, and their hash values have been computed using a secure hash function The SHA-256 hash function generates hash values for each frame of the video as illustrated in Fig.6. In the embedding stage, the hash values have been then embedded into the corresponding/adjacent even-numbered frames as a watermark. The procedure for embedding the watermark has been illustrated in Fig.7. The detailed procedural sequence is shown in Fig.8 and the corresponding pseudo-code is described in Alogrithm1. In this method, video frames are hashed and placed as watermarks on other frames, following several processes. Initially, SHA-256 hash functions were used to create a fixed-length hash value for each odd-numbered video frame. Each chosen odd-numbered frame receives a distinct hash value from the hash function. The relevant hashed frame has been included as a

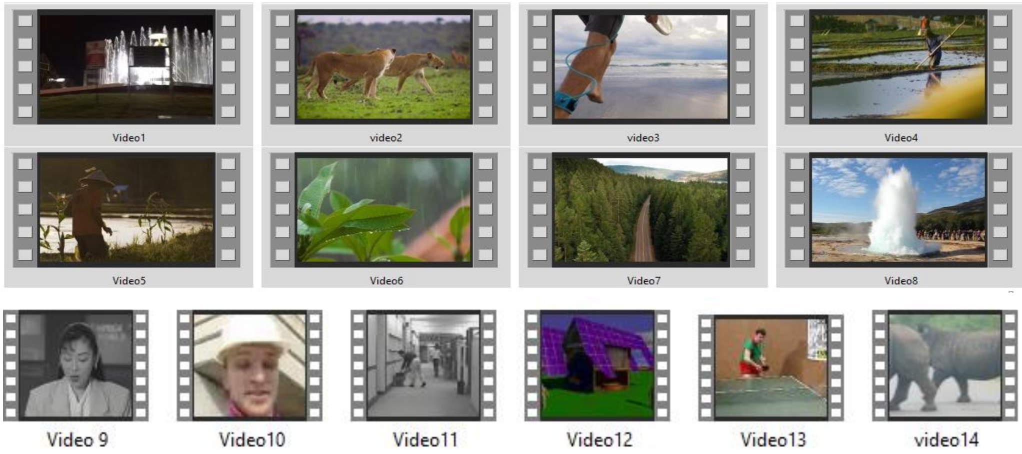


Fig. 4: Clips of Input videos

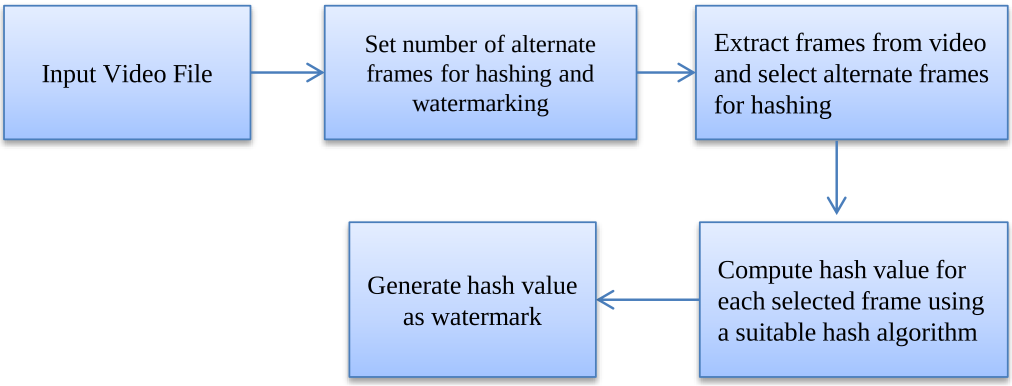

Input Video File
Set number of alternate frames for hashing and watermarking
Extract frames from video and select alternate frames for hashing
Compute hash value for each selected frame using a suitable hash algorithm
Generate hash value as watermark

Fig. 5: Calculating Hashes and Creating Watermarks

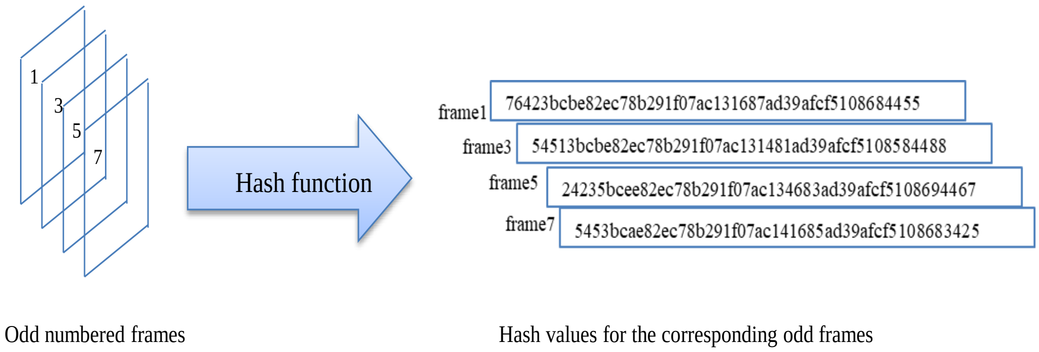

1
3
5
7
Hash function
frame1 76423bcbe82ec78b291f07ac131687ad39afcf5108684455
frame3 54513bcbe82ec78b291f07ac131481ad39afcf5108584488
frame5 24235bcee82ec78b291f07ac134683ad39afcf5108694467
frame7 5453bcae82ec78b291f07ac141685ad39afcf5108683425
Odd numbered frames
Hash values for the corresponding odd frames

Fig. 6: Generation of Hash Values Using SHA256 Hashing

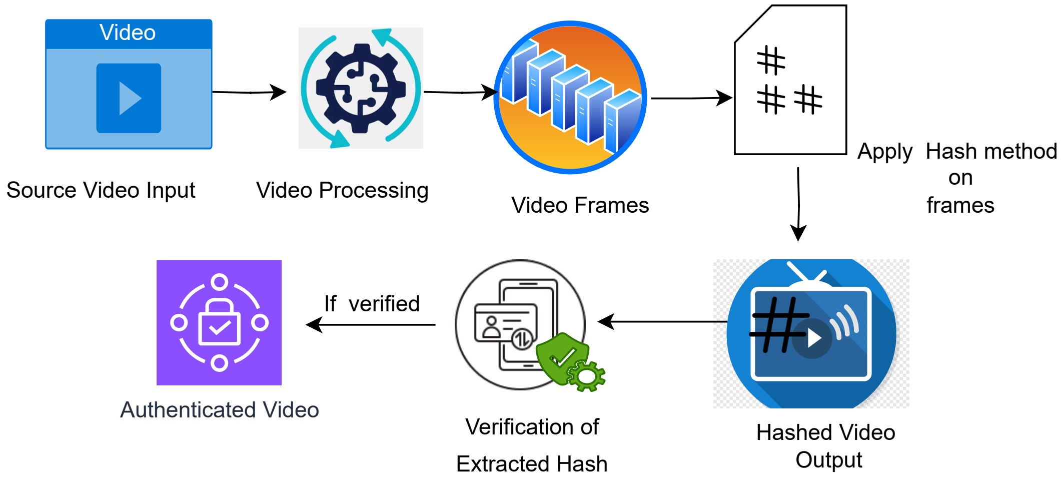


Fig. 7: Embedding Process

watermark to the adjacent even-numbered frame. Then the watermarked video was created as a new file and saved. The procedure was repeated for every frame of the video.

**Algorithm 1** Watermark Embedding Algorithm

**Algorithm Description:** This algorithm outlines the process of embedding a watermark into a video file using the SHA-256 hash function.

```
Start
Take video file as input and generate the frames
Select the odd-numbered frames from the generated frames
for each selected odd-numbered frame $F_i$ do
    Apply hash function (SHA-256) on the selected odd-numbered frame $F_i$
    A unique hash value of fixed length is generated: $H(F_i) = \text{SHA-256}(F_i)$
    Find the adjacent even-numbered frame $F_{i+1}$
    Embed the generated hash value $H(F_i)$ into the adjacent even-numbered frame $F_{i+1}$ as a watermark
end for
Stop
Save the watermarked video as the output file
```

**Algorithm 2** Apply hash function

```
Apply hash function (SHA-256) on the frame:
Let $F_i$ denote the $i$-th selected odd numbered frame.
Let $H(F_i)$ denote the hash value generated by applying the SHA-256 hash function on $F_i$.
The hash value $H(F_i)$ can be calculated as:
    $H(F_i) = \text{SHA-256}(F_i)$
Where:
  • $H(F_i)$ is the hash value of frame $F_i$.
  • SHA-256$(F_i)$ represents the application of the SHA-256 hash function on frame $F_i$.
```

The detailed visual representation of the proposed method is depicted in Fig. 9.

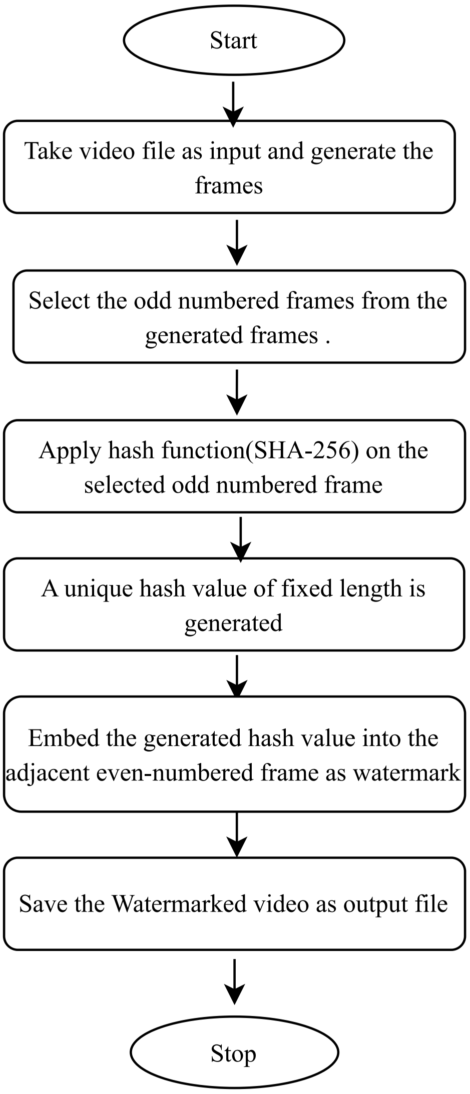

Start
Take video file as input and generate the frames
Select the odd numbered frames from the generated frames .
Apply hash function(SHA-256) on the selected odd numbered frame
A unique hash value of fixed length is generated
Embed the generated hash value into the adjacent even-numbered frame as watermark
Save the Watermarked video as output file
Stop

Fig. 8: Sequential Process for Embedding the Hashed Watermark

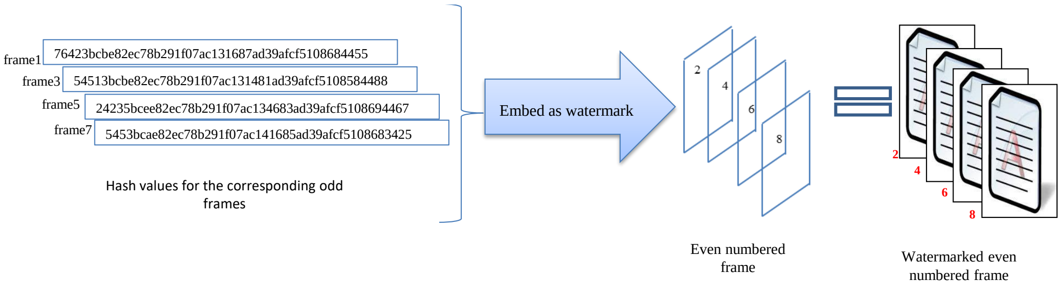

frame1
76423bcbe82ec78b291f07ac131687ad39afcf5108684455
frame3
54513bcbe82ec78b291f07ac131481ad39afcf5108584488
frame5
24235bcee82ec78b291f07ac134683ad39afcf5108694467
frame7
5453bcae82ec78b291f07ac141685ad39afcf5108683425
Hash values for the corresponding odd frames
Embed as watermark
2
4
6
8
Even numbered frame
2
4
6
8
Watermarked even numbered frame

Fig. 9: Using Hash Values as Watermarks on Even-Numbered Video Frames

**Algorithm 3** Embed the hash value

```
1: Embed the hash value into the adjacent even-numbered frame:
2: Let $F_{i+1}$ denote the adjacent even-numbered frame to $F_i$.
3: Let $W_{i+1}$ denote the watermarked frame after embedding the hash value.
4: The watermarked frame $W_{i+1}$ can be calculated as:
       $$W_{i+1} = \text{Embed}(F_{i+1}, H(F_i))$$
   Where:
     • $W_{i+1}$ is the watermarked frame.
     • $F_{i+1}$ is the adjacent even-numbered frame.
     • $H(F_i)$ is the hash value of the odd-numbered frame $F_i$.
     • $\text{Embed}(\cdot)$ is the function to embed the hash value into the adjacent even-numbered frame.
```

**Algorithm 4** Watermark Verification Algorithm

**Algorithm Description:** This algorithm verifies the authenticity of watermarked frames in a video file. Let $H_i$ denote the hash value extracted from the $i$-th even-numbered frame, and $H_i'$ denote the original hash value of the corresponding odd-numbered frame. The algorithm compares $H_i$ with $H_i'$ to determine the authenticity of the video frame.

```
1: Start
2: Extract the hash watermarked even-numbered frames from the watermarked video file
3: Extract the watermarked hash value $H_i$ from the $i$-th even-numbered frame
4: Compare $H_i$ with the original hash value $H_i'$ of the corresponding odd-numbered frame
5: if $H_i = H_i'$ then
6:     The video frame is authentic
7: else
8:     The video frame is tampered
9: end if
10: Stop
```

The outcome of the embedding process is considered as the watermarked video clips. There is no viable difference between the watermarked video and the original video as illustrated in Fig.10. Overall, by adding a distinctive hash value to the non-selected frames, this method enables the authentication of video frames and offers a tool to check the video's integrity.

### D. Extraction Algorithm

The watermarked video's odd-numbered frames were extracted during the extraction stage, and the same hash function was used to calculate their hash values. The detailed procedural sequence is shown in Fig.11.The corresponding extraction pseudo-code has been described in Algorithm4. These steps have been used to extract the hash value from the hashed watermarked video frame and verify a watermarked video's authenticity. First, the watermarked video's embedded hashed frames need to be removed. To achieve this, take the watermarked frames and subtract the original, non-hashed frames. Second, the recovered hashed frames have been utilized to construct a hash value using the same hash function used to embed the frames.

The original hash value for each frame can then be compared to the retrieved hash value. The frame was regarded as authentic and has not been tampered with if the original and extracted hash values match. This procedure maintained the watermarked video's legitimacy in that confirmed the frames hadn't been tampered with or altered. Because any modification to the video content results in a different hash value and, as a result, the authentication fails, the odd-even frame embedding approach offers a reliable solution to video authentication. Therefore, the proposed method has no impact on the video's visual quality.

## V. Result Discussion

Thirty videos with different resolutions, frame rates, bitrates, frame widths, frame heights, and lengths were used to evaluate the suggested technique. A machine with an Intel Core i7 processor and 16 GB of RAM was used to experiment. The OpenCV and NumPy libraries were used to implement the watermarking and extraction processes in Python. The watermark was included in the chosen frames of each video using the odd-even frame embedding method. The experiment employed the SHA-256 hash function. The odd-even frame extraction method was used to retrieve the watermark. Two measures, PSNR and Structural Similarity Index (SSIM), provided by the following equations, were used to assess the performance of the suggested technique.

$$\text{PSNR} = \frac{\sum_{i=1}^{\text{total frames}} P}{\text{Total frames}} \tag{3}$$

where $P = $ PSNR of one frame and given as

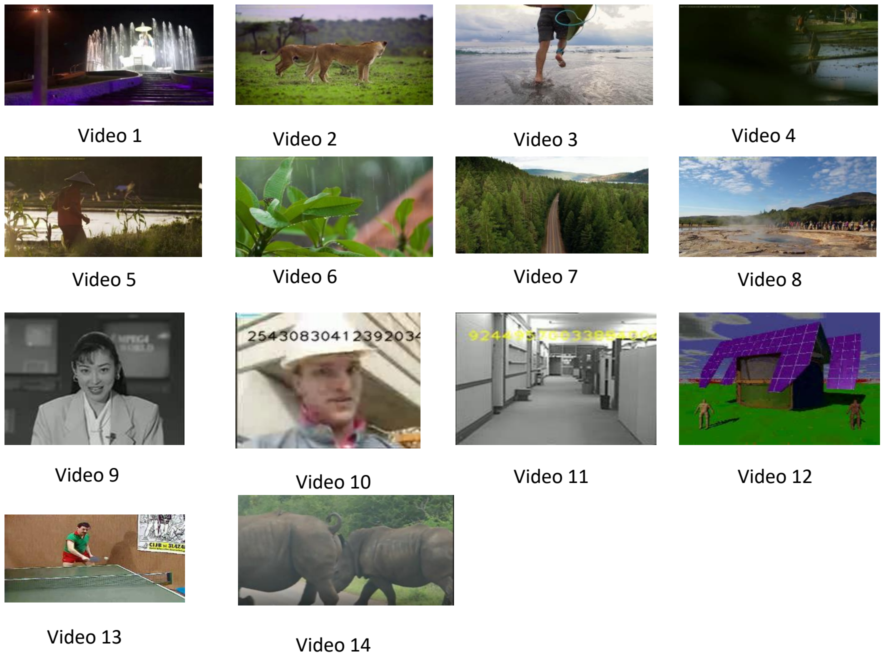


Fig. 10: Watermarked Video Clips

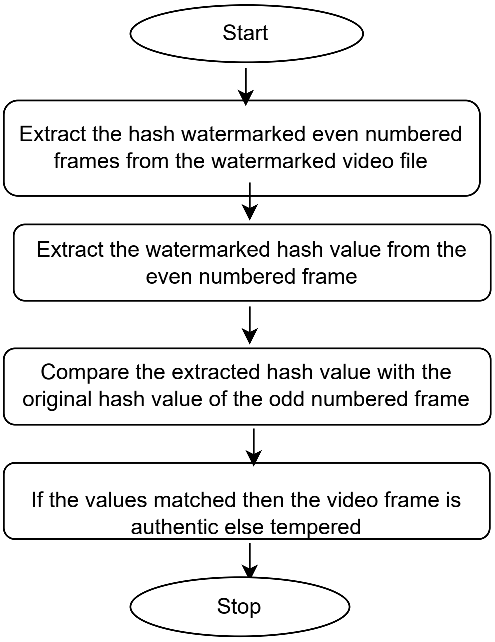

Start
Extract the hash watermarked even numbered frames from the watermarked video file
Extract the watermarked hash value from the even numbered frame
Compare the extracted hash value with the original hash value of the odd numbered frame
If the values matched then the video frame is authentic else tempered
Stop

Fig. 11: Extraction Process

$$P = 10 \cdot \log_{10}\left(\frac{255^2}{\text{MSE}}\right) \tag{4}$$

where $MSE$ = mean square error and given as

$$\text{MSE} = \frac{\sum_{i=1}^{X}\sum_{j=1}^{Y}(A_{ij} - B_{ij})^2}{X \cdot Y} \tag{5}$$

Here $x$ = frame width and $y$ = frame height. MSE should be positive and as small as possible.

$$\text{SSIM} = \frac{\sum_{x=0}^{M}\sum_{y=0}^{N} C(x,y) \cdot WC(x,y)}{\sum_{x=0}^{M}\sum_{y=0}^{N} \lceil C(x,y)^2 \rceil} \tag{6}$$

Here $C$ = Original frame and $WC$ = Watermarked frame at x and y coordinates.

The proposed method was analyzed using the evaluation metrics. Only 14 of the 30 inputs tested were presented graphically. Due to space limits, the additional 17 inputs were not visually demonstrated. The count of frames present in different videos is illustrated in Fig.12. The corresponding average PSNR and SSIM are calculated across diverse videos are listed in Figs.13 and 14.

Key metrics from the examination of 30 videos (with an average of approximately 504 frames per video) are included in the descriptive statistics shown in Table III. These statistics give a thorough summary of the number of frames, video quality, and structural similarity in the dataset. They provide valuable information on the range and distribution of these metrics. The average PSNR is 37.45, which reflects the videos' signal integrity. The SSIM scores vary between 0.00 and 0.99, with an average of 0.88. It indicates different levels of structural similarity between the original and watermarked videos with a standard deviation of 0.24.

TABLE III: Descriptive Statistics for Video Analysis

| **Statistic** | **No. of videos** | **No. of frames** | **Avg PSNR** | **SSIM** |
|---|---|---|---|---|
| Count | 30 | 30 | 30 | 30 |
| Mean | 15.5 | 504.4 | 37.45 | 0.88 |
| Std | 8.8 | 248.98 | 7.54 | 0.24 |
| Min | 1.0 | 25.00 | 0.00 | 0.00 |
| 25% | 8.25 | 333.25 | 38.27 | 0.92 |
| 50% | 15.5 | 500.00 | 39.25 | 0.95 |
| 75% | 22.75 | 737.50 | 39.86 | 0.97 |
| Max | 30.00 | 932 | 40.97 | 0.99 |

The average PSNR and SSIM values for the 14 watermarked videos were 36 dB to 41 dB and 0.997, respectively.

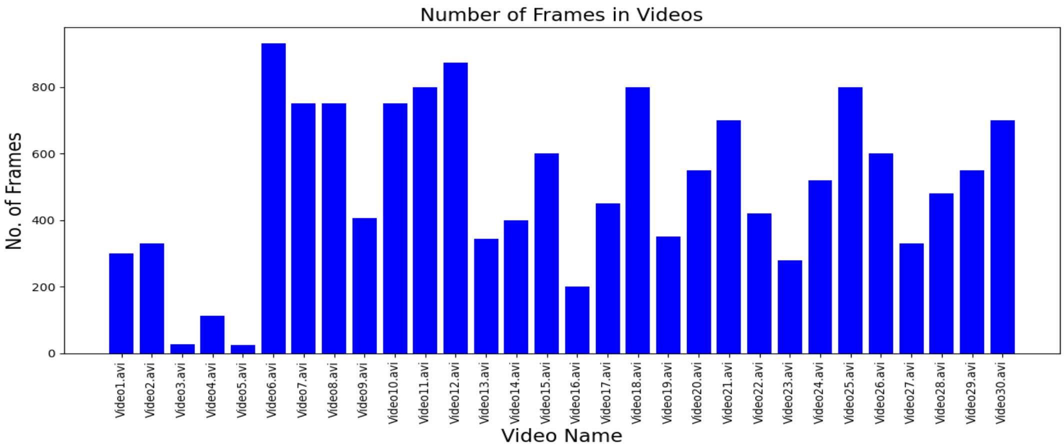


Fig. 12: Count of Frames Present In Different Videos

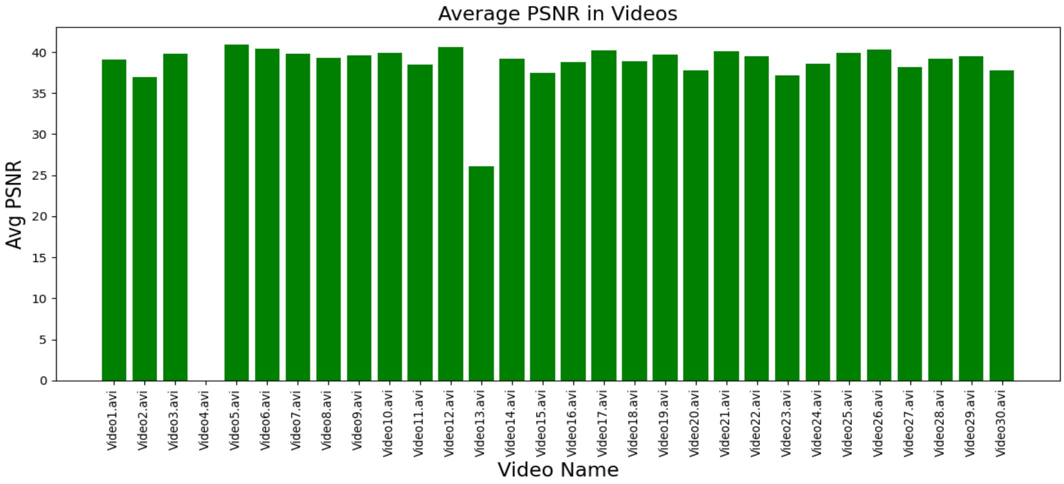


Fig. 13: Average PSNR Calculated Across Diverse Videos.

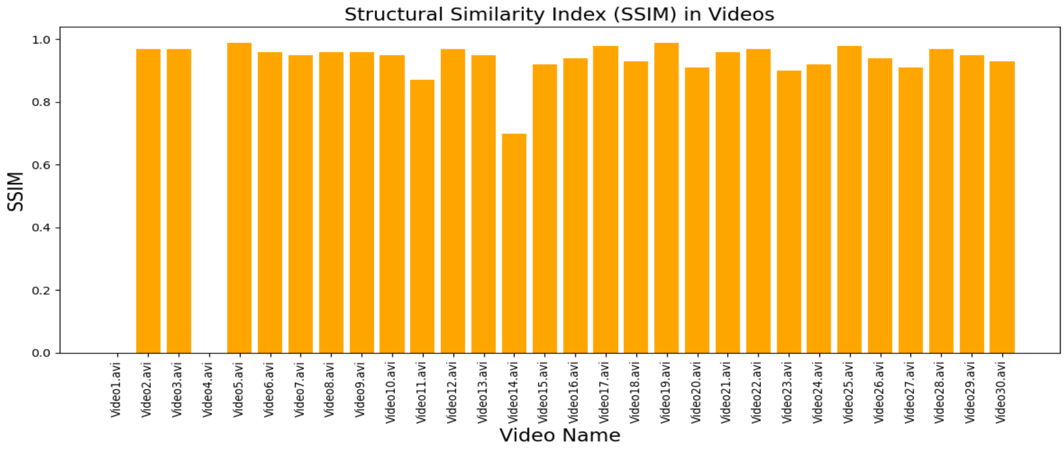
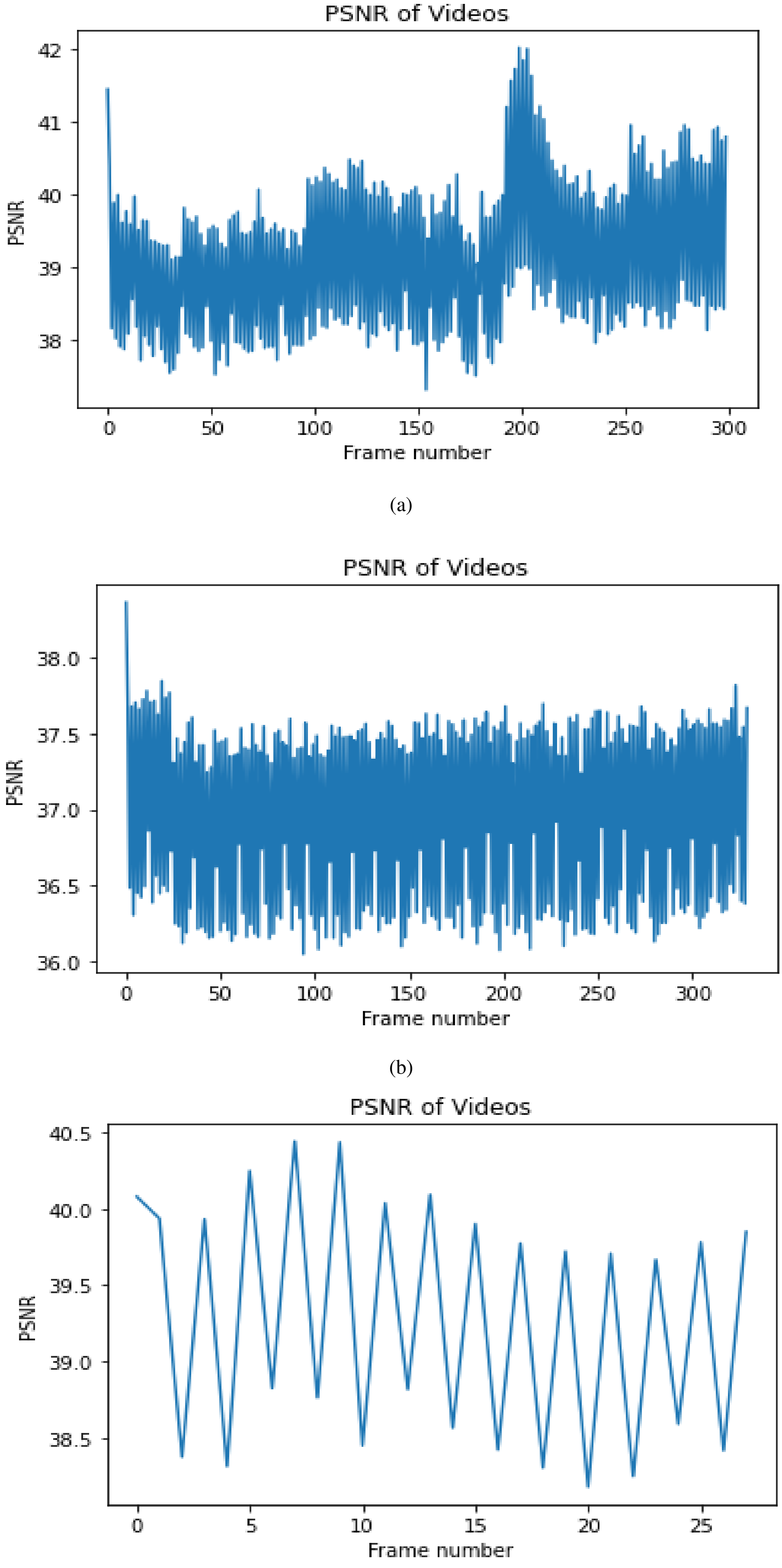


Fig. 14: SSIM Computed Across Various Videos.

These results indicated that the proposed method has successfully embedded the watermark in the video frames without significantly degrading the video quality. The mean PSNR and SSIM values of the original videos and the watermarked videos, as well as the frame-wise PSNR between the original and hashed watermarked videos, are described in Table IV. It provides some metrics for different videos regarding the number of frames, the average PSNR between the original and hashed watermarked video, and the SSIM. It provides some insights into the effectiveness of the hashing and digital watermarking methods used in these videos.

Further, the graphical representation for the PSNR between each original even-numbered frame and the hashed watermarked even-numbered frame is shown in Fig.15.

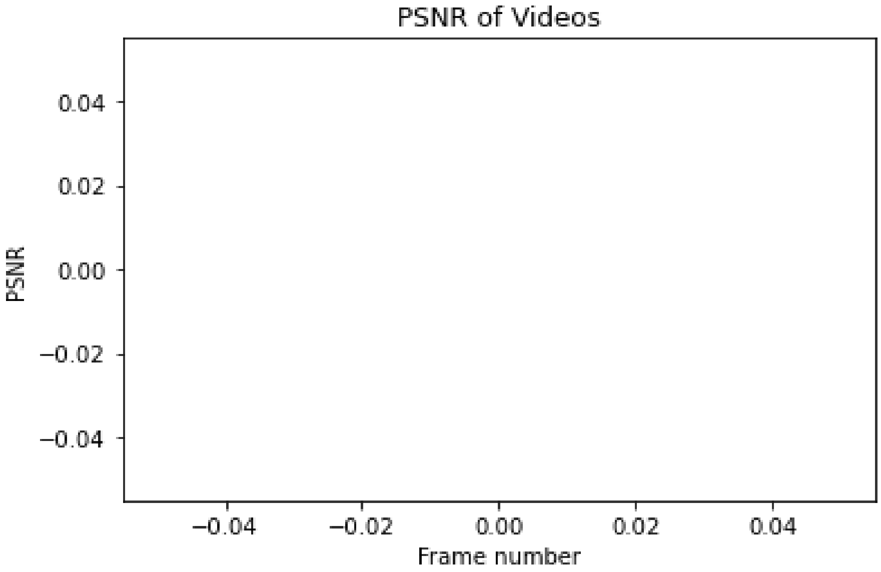
The authentication results for different tampering techniques that include frame deletion, modification, and transcoding have been illustrated in Fig.16 and Table V. The red line has points on it for each kind of tampering


PSNR of Videos
PSNR
42
41
40
39
38
0
50
100
150
200
250
300
Frame number
PSNR of Videos
PSNR
38.0
37.5
37.0
36.5
36.0
0
50
100
150
200
250
300
Frame number
PSNR of Videos
PSNR
40.5
40.0
39.5
39.0
38.5
0
5
10
15
20
25
Frame number

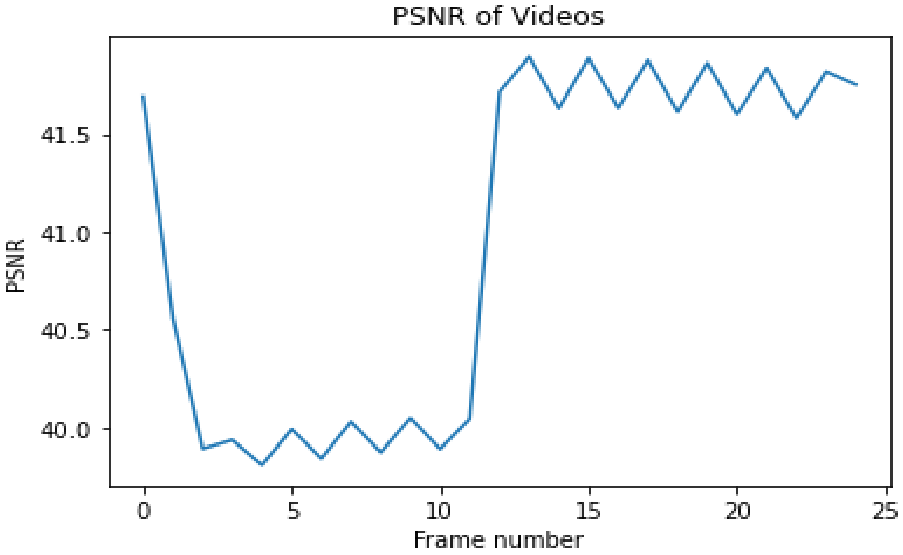
(a)

(b)

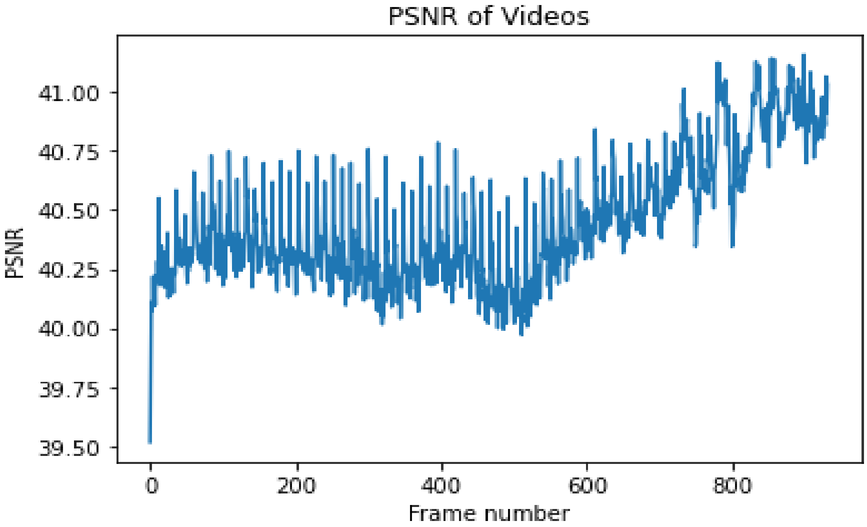
(c)


PSNR of Videos
PSNR
0.04
0.02
0.00
−0.02
−0.04
−0.04
−0.02
0.00
0.02
0.04
Frame number

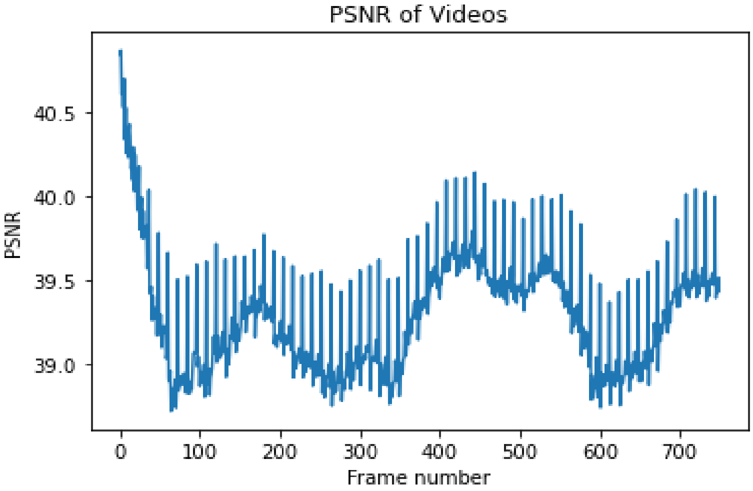
(d)


PSNR of Videos
PSNR
41.5
41.0
40.5
40.0
0
5
10
15
20
25
Frame number

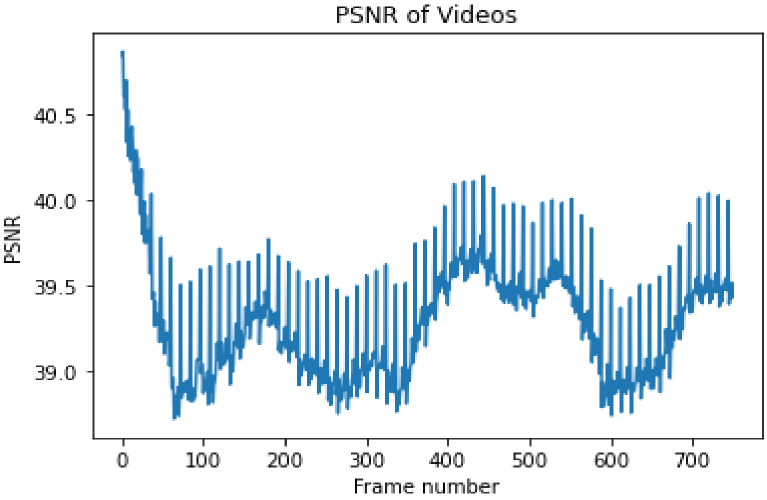
(e)


PSNR of Videos
PSNR
41.00
40.75
40.50
40.25
40.00
39.75
39.50
0
200
400
600
800
Frame number

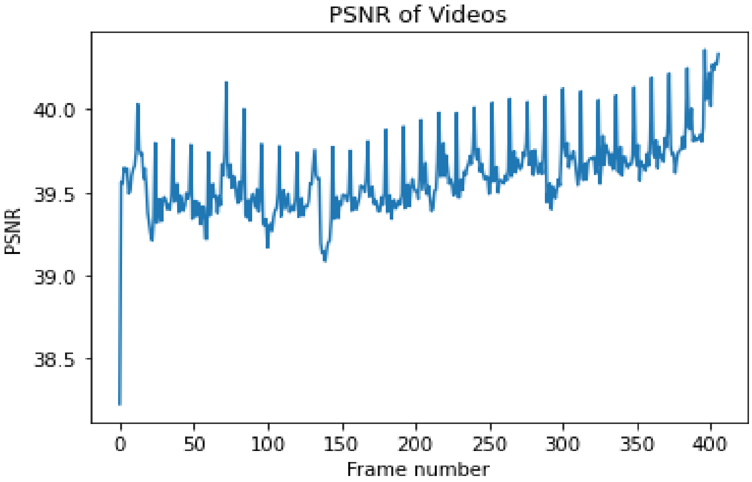
(f)


PSNR of Videos
PSNR
40.5
40.0
39.5
39.0
0
100
200
300
400
500
600
700
Frame number

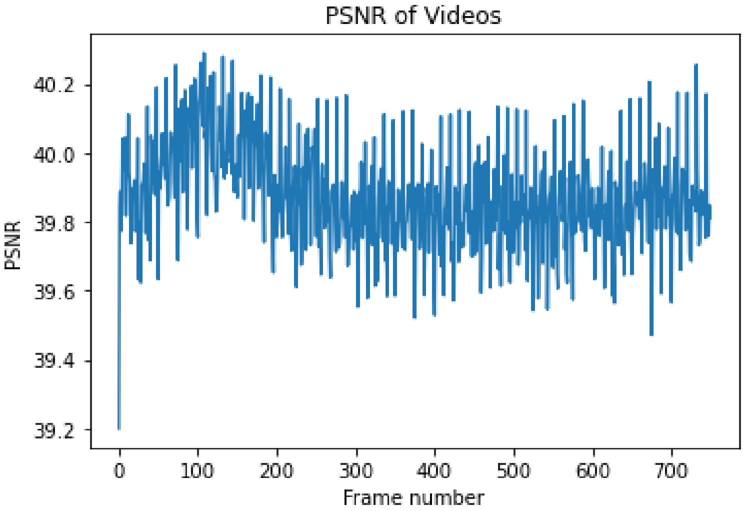
(g)


PSNR of Videos
PSNR
40.5
40.0
39.5
39.0
0
100
200
300
400
500
600
700
Frame number

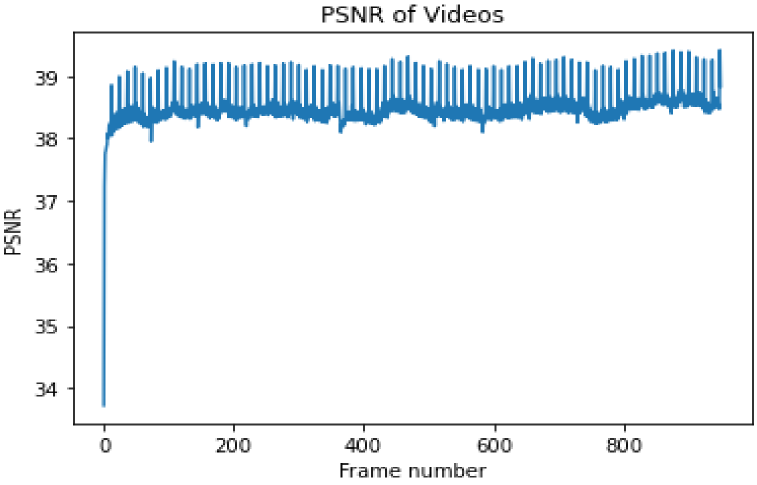
(h)


PSNR of Videos
PSNR
40.0
39.5
39.0
38.5
0
50
100
150
200
250
300
350
400
Frame number

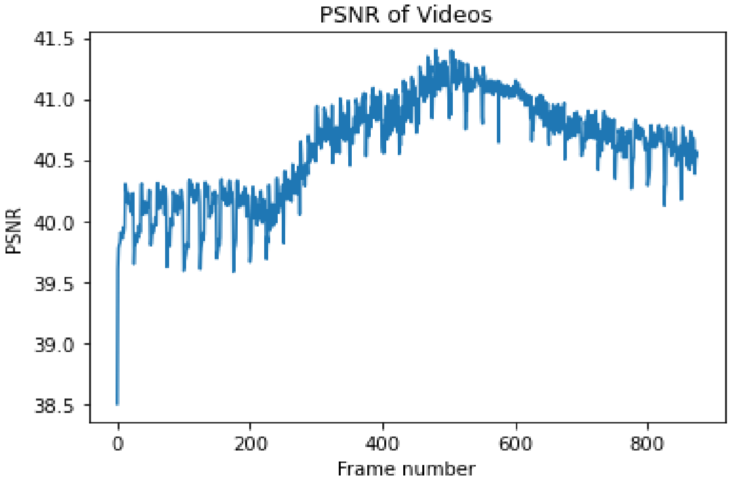
(i)


PSNR of Videos
PSNR
40.2
40.0
39.8
39.6
39.4
39.2
0
100
200
300
400
500
600
700
Frame number

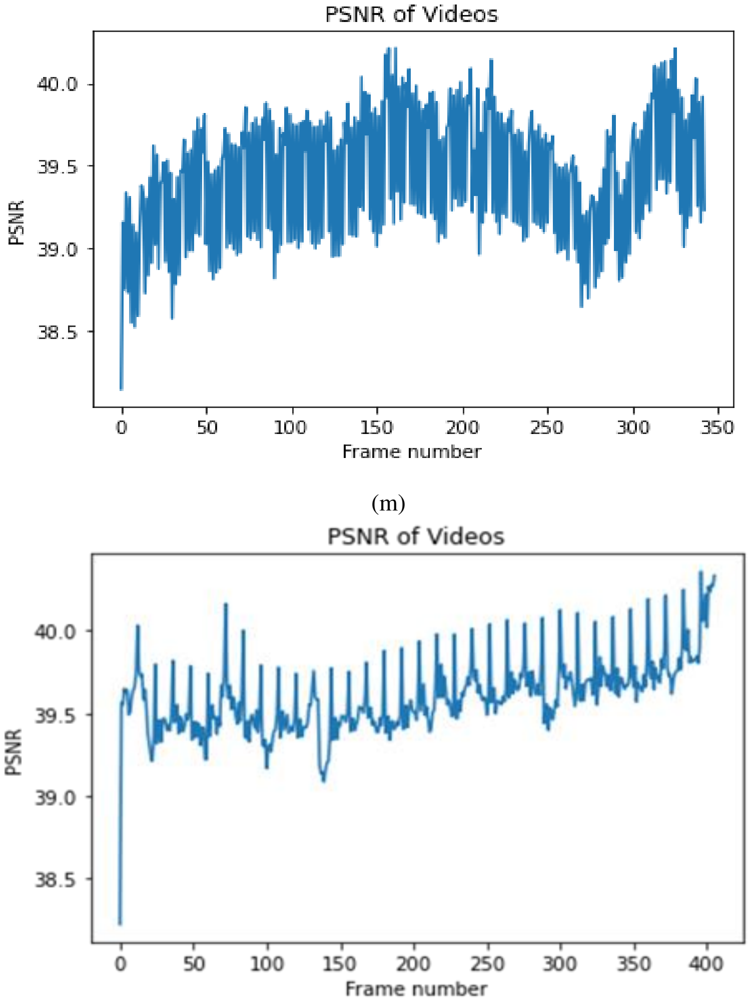
(j)


PSNR of Videos
PSNR
39
38
37
36
35
34
0
200
400
600
800
Frame number

(k)


PSNR of Videos
PSNR
41.5
41.0
40.5
40.0
39.5
39.0
38.5
0
200
400
600
800
Frame number

(l)



(m)

(n)

Fig. 15: Frame-wise PSNR between Original and Hashed Watermarked Video

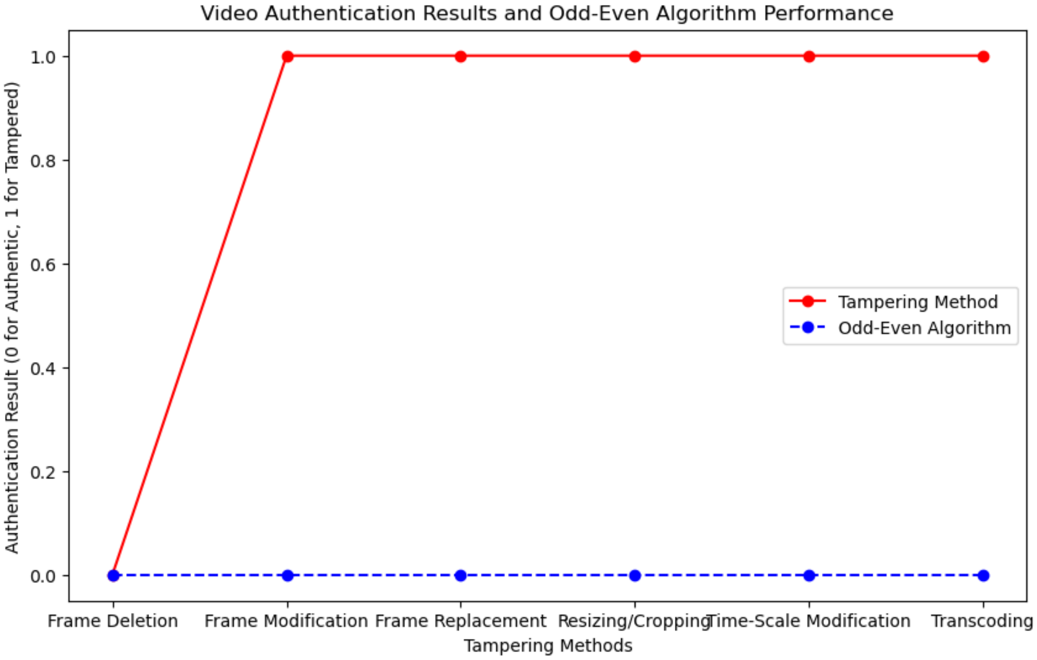


Fig. 16: Results of Video Authentication and the Performance of the Odd-Even Algorithm

TABLE IV: The Mean PSNR and SSIM of Different Videos

| Sr no | Input Video name | Avg PSNR | SSIM |
|---|---|---|---|
| 1. | Video1.avi | 39.13 | 0 |
| 2. | Video2.avi | 36.94 | 0.97 |
| 3. | Video3.mp4 | 39.84 | 0.97 |
| 4. | Video4.avi | 0.00 | 0 |
| 5. | Video5.mp4 | 40.97 | 0.99 |
| 6. | Video6.mp4 | 40.45 | 0.96 |
| 7. | Video7.mp4 | 39.82 | 0.95 |
| 8. | Video8.mp4 | 39.30 | 0.96 |
| 9. | Video9.mp4 | 39.60 | 0.96 |
| 10. | Video10.mp4 | 39.87 | 0.95 |
| 11. | Video11.mp4 | 38.49 | 0.87 |
| 12. | Video12.mp4 | 40.61 | 0.97 |
| 13. | Video13.mp4 | 26.07 | 0.95 |
| 14. | Video14.mp4 | 39.20 | 0.96 |

TABLE V: Comparison of Authentication-Results for Tampering Methods

| Sr no. | Tampering Method | Authentication Result |
|---|---|---|
| 1. | Frame deletion | Hash not found |
| 2. | Frame modification | Tampered |
| 3. | Frame replacement | Tampered |
| 4. | Resizing cropping | Tampered |
| 5. | Time scale modification | Tampered |
| 6. | Transcoding | Tampered |

technique. A legitimate video has a value of 0 on the y-axis, but a manipulated video has a value of 1. The blue dashed line shows the odd-even algorithm's outcomes simultaneously. This line gives a comparison view of the algorithm's performance across the identical tampering techniques. The points at which the blue line coincides with the red line or diverges from it provide information about how well the odd-even algorithm works in comparison to other tampering methods and the algorithm's performance in various tampering situations.

A couple of frames of the hashed watermarked video, used for testing the experiment in which the hash values associated with previous odd-numbered frames were embedded into the next even-numbered frames as a watermark, have been shown in Fig.17. The close-up depiction of the odd-numbered frames and the corresponding hash value for the watermarked even-numbered frames of the video in the experiment is shown in Fig.18. The extraction of the watermark, which is in the form of a hash value string, is depicted in Fig.19. The string was retrieved from the even-numbered frame. The previous odd-numbered frame's hash value for the original video was then compared. There was no change in the frames. By embedding the hash values of other frames as watermarks into the non-selected frames, the proposed approach can provide an extra layer of security to the video material. Any modifications to the watermarked frames will influence the hash values of the embedded frames, making it more difficult for an attacker to tamper with the video content without being discovered.

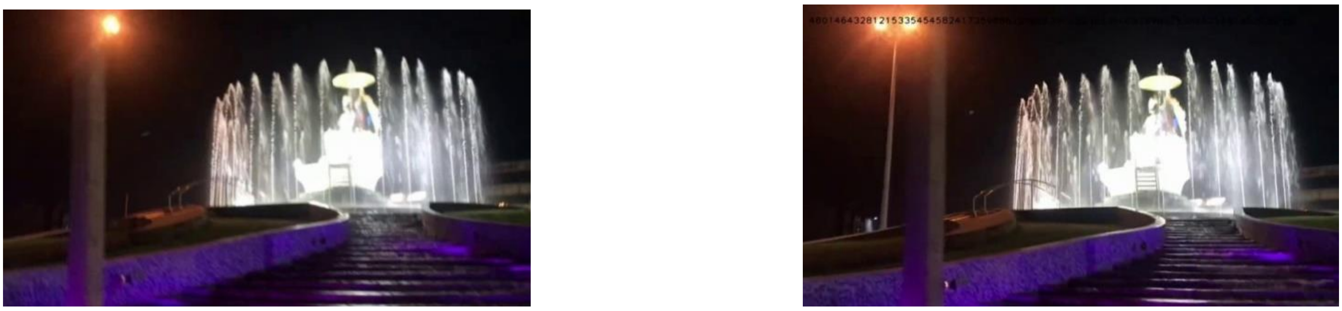
## VI. Performance Comparison with state-of-art methods

The suggested approach distinguishes itself from cutting-edge techniques [13] [17] [27] by being more straightforward, effective, and efficient in handling the problem of video authentication. The suggested approach provides a reasonable compromise between security and usability. The authentication results in Table V demonstrate the resilience of the system when it is exposed to various tampering techniques, including frame deletion, alteration, replacement, resizing/cropping, time scale manipulation, and transcoding. The hash is not discovered when a frame is purposefully deleted from the video, indicating that tampering has been successfully detected. In a similar vein, the authentication accurately detects tampering when frames are changed, replaced, or undergo scaling or cropping, offering a dependable way to guarantee the integrity of the video material. The detection of even more intricate kinds of manipulation, including transcoding and time scale change, highlights how successful the suggested approach is. Overall, the comparison demonstrates how resistant the approach is to different types of tampering, establishing it as a workable option for video authentication in a variety of applications.High degrees of security may be offered by some sophisticated techniques, however, these techniques frequently have a large computational cost or complicated implementation. On the other hand, the suggested method puts efficiency ahead of security without sacrificing either, making it suitable for a variety of uses.

## VII. Conclusion

The proposed method of video authentication using hash-based watermarking, where an odd-numbered frame hash value is embedded into an even-numbered frame of the video, is a robust and efficient way to prevent video tampering and ensure the authenticity of the video. This technique can be used in various applications, such as forensic analysis, surveillance, and content protection. By embedding the hash value of each odd-numbered frame into the even-numbered frame, the proposed method ensures that any alteration made to the video will be detected, even if it is made to a single frame. Moreover, the proposed technique is computationally efficient and does not require significant resources to implement. Therefore, it can be deployed in real-time applications without any significant impact on performance. Overall, the proposed method offers a reliable and cost-effective solution for video authentication, which can enhance the security and integrity of video data in various applications. Future work could focus on improving its robustness to more advanced attacks and evaluating its performance on a larger dataset.

Fig. 17: Watermarked Output Video Frames from the Proposed Video Used for Experiment



(a)

(b)

Fig. 18: (a) Odd-numbered frame (b)Even numbered frames with a hash of odd-numbered frame as watermark

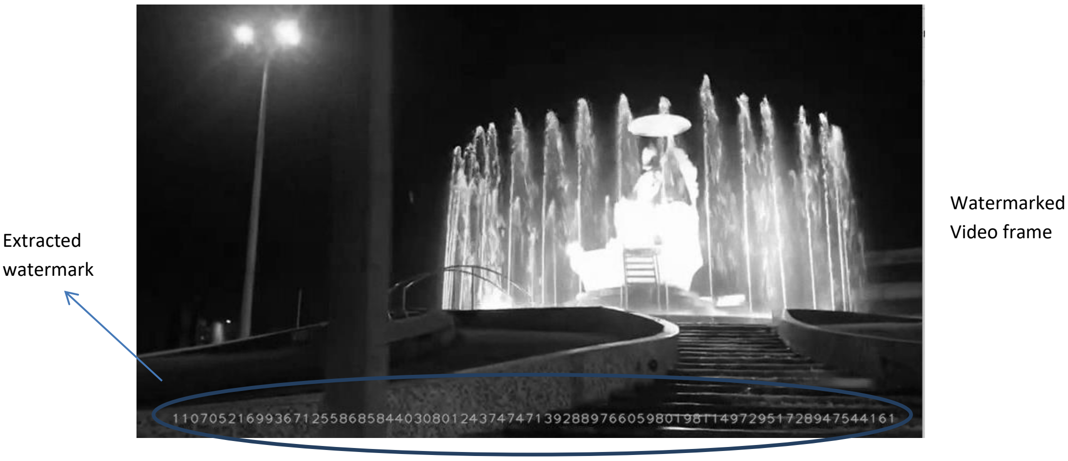


Fig. 19: Extraction of hash value from the following even-numbered frame

## References

[1] A. F. Qasim, F. Meziane, and R. Aspin, "A reversible and imperceptible watermarking scheme for mr images authentication," in *24th International Conference on Automation and Computing (ICAC)*. IEEE, pp. 1–6, 2018.

[2] P. Saini and R. Ahuja, "Watermarked hashing as a video content authentication technique," *ECS Transactions*, vol. 107, no. 1, pp. 5211–5218, 2022.

[3] N. Janu, A. Kumar, P. Dadheech, G. Sharma, A. Kumar, and L. Raja, "Multiple watermarking scheme for video & image for authentication & copyright protection," *IOP Conference Series: Materials Science and Engineering*, vol. 1131, no. 1, pp. 1–15, 2021.

[4] P. S. Rathore, J. M. Chatterjee, A. Kumar, and R. Sujatha, "Energy-efficient cluster head selection through relay approach for wsn," *The Journal of Supercomputing*, vol. 77, pp. 7649–7675, 2021.

[5] N. A. S. Al-maweri, R. Ali, W. A. W. Adnan, A. R. Ramli, and S. M. S. Ahmad, "State-of-the-art in techniques of text digital watermarking: Challenges and limitations." *J. Comput. Sci.*, vol. 12, no. 2, pp. 62–80, 2016.

[6] M. Vatsa, R. Singh, S. K. Singh, and S. Upadhyay, "Video authentication using relative correlation information and svm," *Computational intelligence in multimedia processing: recent advances*, vol. 96, pp. 511–529, 2008.

[7] X. Yu, C. Wang, and X. Zhou, "A hybrid transforms-based robust video zero-watermarking algorithm for resisting high efficiency video coding compression," *IEEE Access*, vol. 7, pp. 115 708–115 724, 2019.

[8] N. Pitropakis, E. Panaousis, T. Giannetsos, E. Anastasiadis, and G. Loukas, "A taxonomy and survey of attacks against machine learning," *Computer Science Review*, vol. 34, pp. 100 199–100 218, 2019.

[9] H. Fang, W. Zhang, Z. Ma, H. Zhou, S. Sun, H. Cui, and N. Yu, "A camera shooting resilient watermarking scheme for underpainting documents," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 11, pp. 4075–4089, 2019.

[10] B. C. Hosler, X. Zhao, O. Mayer, C. Chen, J. A. Shackleford, and M. C. Stamm, "The video authentication and camera identification database: A new database for video forensics," *IEEE Access*, vol. 7, pp. 76 937–76 948, 2019.

[11] M. Sajjad, I. U. Haq, J. Lloret, W. Ding, and K. Muhammad, "Robust image hashing based efficient authentication for smart industrial environment," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6541–6550, 2019.

[12] F. Khelifi and A. Bouridane, "Perceptual video hashing for content identification and authentication," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 1, pp. 50–67, 2017.

[13] A. P. M. Maung, Y. Tew, and K. Wong, "Authentication of mp4 file by perceptual hash and data hiding," *Malaysian Journal of Computer Science*, vol. 32, no. 4, pp. 304–314, 2019.

[14] L. Du, A. T. Ho, and R. Cong, "Perceptual hashing for image authentication: A survey," *Signal Processing: Image Communication*, vol. 81, pp. 115 713–115 778, 2020.

[15] R. Ajithkumar, K. S. Reddy, and G. G. Devadhas, "Watermarking schemes for high security with applications and attacks: Research challenges and open issues," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 4, pp. 2277–3878, 2019.

[16] P. Kulkarni and A. O. Mulani, "Robust invisible digital image watermarking using discrete wavelet transform," *International Journal of Engineering Research & Technology (IJERT)*, vol. 4, no. 01, pp. 139–141, 2015.

[17] H. Chen, Y. Wo, and G. Han, "Multi-granularity geometrically robust video hashing for tampering detection," *Multimedia Tools and Applications*, vol. 77, pp. 5303–5321, 2018.

[18] Z. Tang, L. Chen, H. Yao, X. Zhang, and C. Yu, "Video hashing with dct and nmf," *The Computer Journal*, vol. 63, no. 7, pp. 1017–1030, 2020.

[19] S. A Hasso and T. Basheer Taha, "A new tamper detection algorithm for video," *Journal of Engineering Science and Technology (JESTEC)*, vol. 15, no. 5, pp. 3375–3387, 2020.

[20] A. Hammami, A. Ben Hamida, and C. Ben Amar, "Blind semi-fragile watermarking scheme for video authentication in video surveillance context," *Multimedia Tools and Applications*, vol. 80, pp. 7479–7513, 2021.

[21] Z. Zainol, J. S. Teh, M. Alawida, A. Alabdulatif *et al.*, "Hybrid svd-based image watermarking schemes: a review," *IEEE Access*, vol. 9, pp. 32 931–32 968, 2021.

[22] W. Birouk, A. Lahoulou, A. Melit, and A. Bouridane, "Robust perceptual fingerprint image hashing: a comparative study," *International Journal of Biometrics*, vol. 15, no. 1, pp. 59–77, 2023.

[23] Z. Tang, S. Zhang, X. Zhang, Z. Li, Z. Chen, and C. Yu, "Video hashing with secondary frames and invariant moments," *Journal of Visual Communication and Image Representation*, vol. 79, pp. 103 209–103 217, 2021.

[24] Z. Chen, Z. Tang, X. Zhang, R. Sun, and X. Zhang, "Efficient video hashing based on low-rank frames," *IET Image Processing*, vol. 16, no. 2, pp. 344–355, 2022.

[25] H. Mareen, N. Van Kets, P. Lambert, and G. Van Wallendael, "Fast fallback watermark detection using perceptual hashes," *Electronics*, vol. 10, no. 10, pp. 1155–1172, 2021.

[26] A. K. Jabbar, A. T. Hashim, and Q. F. Al-Doori, "Secured medical image hashing based on frequency domain with chaotic map," *Engineering and Technology Journal*, vol. 39, no. 5A, pp. 711–722, 2021.

[27] G. Sujatha, D. Hemavathi, K. Sornalakshmi, and S. Sindhu, "Video tampering detection using difference-hashing algorithm," in *Journal of physics: conference series*, vol. 1804, no. 1, pp. 012 145–012 151, 2021.

[28] Q. Ma and L. Xing, "Perceptual hashing method for video content authentication with maximized robustness," *EURASIP Journal on Image and Video Processing*, vol. 2021, no. 1, pp. 36–52, 2021.

[29] S. Gupta, S. K. Yadav, A. P. Singh, and K. C. Maurya, "A comparative study of secure hash algorithms," in *Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems*, vol. 2, pp. 125–133, 2016.

[30] M. H. A. Al-Hooti, T. Ahmad, and S. Djanali, "Improving the capability of reduced difference expansion based digital image data hiding." *IAENG International Journal of Computer Science*, vol. 46, no. 4, pp. 677–690, 2019.

[31] C. Shang, Y. Xue, W. X. Liu, and Y. Liu, "Visual image digital watermarking embedding algorithm combining 3d boolean cnn and arnold technology." *IAENG International Journal of Computer Science*, vol. 50, no. 4, pp. 1221–1231, 2023.

[32] A. Christian and R. Sheth, "Digital video forgery detection and authentication technique-a review," *International Journal of Scientific Research in Science and Technology (IJSRST)*, vol. 2, no. 6, pp. 138–143, 2016.

[33] P. Saini, R. Ahuja, and A. Kaur, "A review on video authentication technique exploiting watermarking methods," in *IEEE 9th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)*, pp. 1–5, 2021.

[34] T. P. Batubara, S. Efendi, and E. B. Nababan, "Analysis performance bcrypt algorithm to improve password security from brute force," vol. 1811, no. 1, pp. 012 129–012 136, 2021.

[35] N. A. Shelke and S. S. Kasana, "A comprehensive survey on passive techniques for digital video forgery detection," *Multimedia Tools and Applications*, vol. 80, no. 4, pp. 6247–6310, 2021.

[36] N. Khairina, M. K. Harahap, and J. H. Lubis, "The authenticity of image using hash md5 and steganography least significant bit," *International Journal of Information System and Technology*, vol. 2, no. 1, pp. 1–6, 2018.

[37] D. Rachmawati, J. Tarigan, and A. Ginting, "A comparative study of message digest 5 (md5) and sha256 algorithm," in *Journal of Physics: Conference Series*, vol. 978, pp. 012 116–012 122, 2018.

[38] P. P. Pittalia, "A comparative study of hash algorithms in cryptography," *International Journal of Computer Science and Mobile Computing*, vol. 8, no. 6, pp. 147–152, 2019.

[39] B. Aditya, U. Avaneesh, K. Adithya, A. Murthy, R. Sandeep, and B. Kavyashree, "Invisible semi-fragile watermarking and steganography of digital videos for content authentication and data hiding," *International Journal of Image and Graphics*, vol. 19, no. 03, pp. 1 950 015–1 950 033, 2019.

[40] M. S. U. Islam, A. Kumar, and Y.-C. Hu, "Context-aware scheduling in fog computing: A survey, taxonomy, challenges and future directions," *Journal of Network and Computer Applications*, vol. 180, pp. 103 008–103 026, 2021.