



ARTICLE

## New Fragile Watermarking Technique to Identify Inserted Video Objects Using H.264 and Color Features

Raheem Ogla<sup>1,\*</sup>, Eman Shakar Mahmood<sup>1</sup>, Rasha I. Ahmed<sup>1</sup> and Abdul Monem S. Rahma<sup>2</sup>

<sup>1</sup>Computer Science Department, University of Technology, Baghdad, 11030, Iraq

<sup>2</sup>Computer Science Department, Al\_Maarif University College, Anbar, 31001, Iraq

\*Corresponding Author: Raheem Ogla. Email: Raheem.a.ogla@uotechology.edu.iq

Received: 18 February 2023 Accepted: 13 June 2023 Published: 08 October 2023

### ABSTRACT

The transmission of video content over a network raises various issues relating to copyright authenticity, ethics, legality, and privacy. The protection of copyrighted video content is a significant issue in the video industry, and it is essential to find effective solutions to prevent tampering and modification of digital video content during its transmission through digital media. However, there are still many unresolved challenges. This paper aims to address those challenges by proposing a new technique for detecting moving objects in digital videos, which can help prove the credibility of video content by detecting any fake objects inserted by hackers. The proposed technique involves using two methods, the H.264 and the extraction color features methods, to embed and extract watermarks in video frames. The study tested the performance of the system against various attacks and found it to be robust. The evaluation was done using different metrics such as Peak-Signal-to-Noise Ratio (PSNR), Mean Squared Error (MSE), Structural Similarity Index Measure (SSIM), Bit Correction Ratio (BCR), and Normalized Correlation. The accuracy of identifying moving objects was high, ranging from 96.3% to 98.7%. The system was also able to embed a fragile watermark with a success rate of over 93.65% and had an average capacity of hiding of 78.67. The reconstructed video frames had high quality with a PSNR of at least 65.45 dB and SSIM of over 0.97, making them imperceptible to the human eye. The system also had an acceptable average time difference ( $T = 1.227/s$ ) compared with other state-of-the-art methods.

### KEYWORDS

Video watermarking; fragile digital watermark; copyright protection; moving objects; color image features; H.264

## 1 Introduction

With the rapid advancement of modern technology, the interest in copyright security and the verification of computerized media content (music, images, and video) are becoming increasingly important [1,2]. Information transmitted over interconnected networks, especially sensitive video content on the internet, is subjected to various malicious attacks. These attacks cast many doubts about the security, authenticity, and integrity of the video content on the receiving end [3]. Watermarking technology has become popular for ensuring the authenticity of digital video content [4]. One of the main challenges of image processing is the processing of digital video, which is key for the identification



of interesting objects within images [5,6]. Digital watermarking is a technique that embeds hidden keys known as watermarks into video data; these hidden keys serve many purposes [7–9]. For example, watermark files are added to images, audio, and videos for authentication [9]. There are two types of watermarks: blind and non-blind [10]. Non-blind watermarks require a dataset for retrieval, while blind watermarks do not require the original video contents to be retrieved [11]. In cases where the authenticity of a video's contents is in question, a secure validation framework can be used to show that no tampering has occurred and to detect and identify tiny changes in watermarked interactive media information. To achieve this, sophisticated watermarks are designed [12]. The main goal of developing a watermark framework is to ensure the integrity of live video by detecting and verifying that no modifications have been made [13–15]. However, traditional watermarking techniques are susceptible to tamper attacks and require careful solutions to address challenges such as the ability to detect changes without the original video and sensory transparency [16,17]. Many watermarking algorithms have been proposed to protect copyright and certify standards [17], including recent ones that focus on the H.264/AVC (Advanced Video Coding) standard encoding [18,19]. However, it is not possible to directly apply many of the previously proposed video watermarking algorithms; therefore, it is necessary to develop new algorithms to meet these standards.

This paper proposes a hybrid watermarking technique that is both imperceptible and robust. Two hybrid methods, H.264 and color feature extraction, are used to embed digital watermarks into videos. This technique ensures that contents or moving objects are not inserted into the original video contents, thus providing proof of ownership and addressing unresolved challenges by bridging the digital gap to authenticate video content against hackers. The H.264/AVC video authentication method focuses on the compressed domain, and the embedding process takes place within the compressed bit stream provided by H.264/AVC [20].

## 2 Related Work

The system introduced in this paper includes watermarks that are embedded into the spatial domain. In this section, in this paper, many review research conducted on the proposed methodology.

The authors of [21] presented a new approach for fragile watermarking that has a high capacity for retrieval. Their method involved embedding authentication bits in the adaptive Least Significant Bit (LSB) layers of each block's central pixel, based on the block's complexity for the mean value of bits. The researchers used three methods to retrieve adjacent blocks, and despite a large tampered area, they claimed that their proposal achieved high quality compared to other methods.

The authors of [22] introduced the independent embedding domain (ED) by adopting two RRW (robust reversible watermarking) stages. They recommended transforming the cover image into two independent EDs and embedding robust and reversible watermarks into each domain separately. The authors confirmed that the embedding performance of the original RRW was greatly improved through their proposed method.

The authors of [23] studied a set of features using Log-Polar, discrete wavelet transform (DWT), and singular value decomposition (SVD) techniques to embed and retrieve watermarks for the purpose of safeguarding copyright and creating a robust and undetectable watermark system. They employed the strategy of discarding frames and achieving specific features to enhance their method. The researchers also used scramble and deep learning-based approaches to create a secret sharing image of the watermark, which improved comparison speed compared to the traditional table-based approaches of Log-Polar, DWT, and SVD.

The authors of [24] proposed a technique for authenticating surveillance videos using semi-fragile watermarking in the frequency domain. The process involved generating a binary watermark and identifying regions of interest to use as holders for the watermark during the embedding process. These regions were decomposed into different frequency sub-bands using SVD and DWT. Then, the watermark was embedded in the selected bands in an additive manner. Blind detection was used to retrieve the hidden signature from the watermarked video.

The authors of [25] proposed a low-cost algorithm for detecting video tampering by using the correlation coefficients between video frames and merging them as encoded data in the first frame of the video stream. To embed the encoded data, the correlation values were calculated and encrypted using the Advanced Encryption Standard (AES) algorithm. Finally, the encryption result was hidden using bit-substitution technology, which randomly selects the two least significant bits from the first frame.

The authors of [26] proposed a modern watermarking technique that involved embedding multiple messages and a photograph into a single image for protection, as well as repeating N-frames in the film. They used multiple video sequences and evaluated their approach using three types of content: MP4, AVI, and MPEG. They also proposed a method for combining two watermarks in a specific video format.

The authors of [27] introduced a video watermarking technique that utilized a moving object detection algorithm to embed a bit stream watermark. The goal of the watermark was to increase the moving object's robustness by identifying large blocks that belonged to it in each frame of type P. An animated object detection algorithm was used to embed the watermark in the blocks to ensure the continuity of the bit stream.

The method proposed in this paper offers a precise authentication technique that maintains the original bitrate and sensory quality while offering high fragility and accuracy. The method can detect tampered frames in cases of color manipulation and spatiotemporal tampering.

The proposed method creatively connects the content of video sequences, and it is designed to be sensitive to malicious attacks by hackers. The method is robust and has low computational complexity, and can be applied to different video formats. Additionally, it is sensitive to detecting and verifying that no modifications occur in live video by using a fragile watermark embedded in video content to achieve video copyright protection. The distinctive features and novelty of the proposed method can be summarized as follows:

1. The accuracy of determining the location of the movement of the objects in the video frame, and considering it a fragile region to insert the watermark.
2. Investing in the real-time movement of the moving object in the video frames to insert the largest capacity of the watermark to secure the copyright of the video.
3. The method has less computational complexity. Despite, there being many measures and criteria used to verify.
4. The method is powerful in protecting online video regardless of video resolution or format.

But, when various practical experiments for various video films were conducted, proposed method faced some limitations and challenges. It was found that low-resolution video films have a very small amount of watermark insertion compared to high-resolution video films. In addition to, the cartoon videos prepared for children, the movement of objects in them is far apart (the moving objects) in the spatial space, which leads to difficulty in identifying the fragile area used to insert the watermark. This leads to poor security of this type of video, which leads to the ease of identifying the watermark by hackers.

In recent years, data transmission via multimedia has become an important issue that cannot be abandoned. At the same time, deliberate attacks to sabotage and falsify this data have escalated, so ensuring the credibility and integrity of digital media is a major issue that requires the development of basic mechanisms and solutions. The issue of multimedia security is a fundamental issue, as videos, audio, images, and text files lose their credibility, as tamperers use multiple tools to distort or manipulate their contents by deleting or entering unwanted information, so all kinds of information and data are transmitted over the network with many problems related such as illegal distribution, copying, manipulation, and forgery. Therefore, the motive behind this paper came to contribute to solving the above-mentioned issues related to securing data transmitted via multimedia in the proposed way.

The remainder of the paper is structured as follows, [Section 3](#) introduces the proposed work, [Section 4](#) details the advanced features of video encoding, [Section 5](#) describes the technique for a fragile watermark and its structure, as well as embedded and extracted color features, [Section 6](#) presents the experimental results and corresponding discussion, finally, [Section 7](#) concludes the paper and suggests areas for future work.

### **3 Methodology**

This section describes the methodology used in this research. To watermark video content, two hybrid methods are used: H.264 and color feature extraction. The aim is to identify the most vulnerable regions caused by the rapid movement of objects in video frames, by determining the background of moving objects in digital video. This enables the insertion of watermarks in fragile areas to secure real live videos and protect their copyright. The video watermark scheme is designed to detect various attacks by hackers that tamper with moving objects and their backgrounds in the video, such as inserting unwanted information or video piracy.

#### ***3.1 H.264 Method***

The proposed method for video watermarking is based on the H.264 standard developed by the Joint Video Team (JVT). This method uses novel coding approaches to improve the Rate-distortion (RD) performance compared to the previous H.263 standard. A fragile video watermarking technique is introduced where the watermark data is contained in a movement-mapped, extremely fragile matrix. The H.264/AVC video watermarking approach that is used during encoding embeds the watermark in the motion object matrix, taking advantage of its fragility based on statistical analysis to determine the best RD cost using the H.264/AVC RD. This approach also detects the location of moving objects in a video sequence by analyzing the movement vectors using the H.264/AVC RD.

#### ***3.2 Color Feature Extraction Method***

The second method that is used to protect the ownership of video copyrights involves the extraction of the color features from the key frames of the video. This involves comparing frames

with lower level features such as color and structure to extract an accurate key frame that covers the entire video sequence using a two-stage method. First, an alternate sequence is created by identifying color feature differences between adjacent frames in the original sequence. Then, by analyzing the structural feature differences between adjacent frames in the alternate sequence, the final key frame sequence is obtained. An optimization step is included to ensure efficient key frame extraction based on the desired number of final key frames.

The measure of structural similarity is determined by calculating the covariance. This involves measuring the similarity of the blocks within a frame at different points, denoted as  $P(x_i, y_i)$ . The covariance of  $x$  and  $y$ , which is used to determine the correlation coefficient, is then calculated using Eq. (1).

$$\sigma_{xy} = \frac{1}{N - 1} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y) \tag{1}$$

where  $\mu_i$  indicates the value of an average and  $N$  is the number of blocks within the frame.

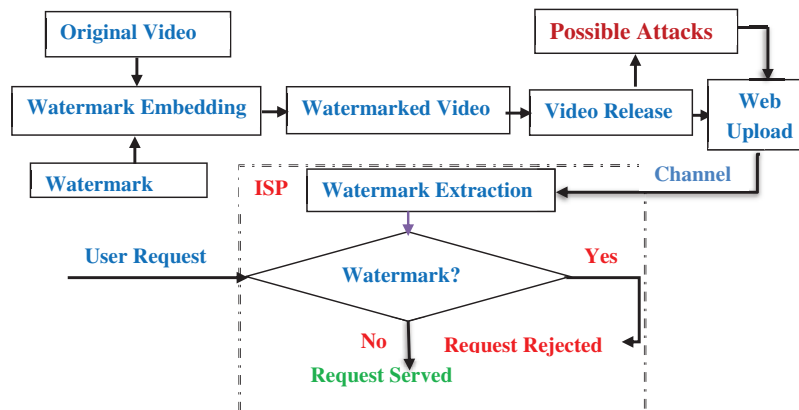
The structure similarity component between two frame blocks is calculated as  $s(x, y)$ , based on the two blocks of the corresponding frame at the same position  $x$  (in the original image) and  $y$  (in the test image) as shown in Eq. (2).

$$s(x, y) = \frac{\sigma_{xy} + I}{\sigma_x \sigma_y + I} \tag{2}$$

where  $I = ((M \times N)^2 / 2)$ ,  $M \in [0, 255]$ ,  $N \ll 1$ , and the variance of  $x$  and  $y$  are  $\sigma_x$  and  $\sigma_y$ , respectively.

#### 4 Proposed Method

The proposed method aims to detect different types of attacks done by hackers who tamper with moving objects and their backgrounds in videos by inserting unwanted information or engaging in video piracy. The detection of such tampering is achieved by embedding fragile watermarks in video content for copyright verification. Two techniques, H.264/AVE and color feature extraction, are used to authenticate and verify the watermark. The suggested method involves two stages, embedding and extracting watermarks. A general block diagram of the proposed digital video watermark for protecting transmitted video content is shown in Fig. 1.



**Figure 1:** General block diagram of the proposed digital video watermark for protecting transmitted video contents

The proposed embedding phase has been described in Algorithm 1, involves multiple steps.

---

**Algorithm 1:** Embedded Watermark Phase

---

**Input:** Video Sequence, Message Watermark

**Output:** Watermarked Video

**Begin**

Step 1: Deconstruction of the video content into a sequence of frames (2 to  $N - 1$ ).

Step 2: Dividing each frame into equal size blocks ( $n \times n$ ).

Step 3: For each frame in the video, determine object motion and consider as fragile region.

Step 4: If object motion occurs in the current frame, then mark the motion position of the object's block as 1, else mark as 0.

Step 5: Construct a mapped matrix of 0s and 1s, it will be used in the extraction phase.

Step 6: Immediately exploit the background of motion blocks marked by 1 to embed a message watermark as a secret key in 2LSB.

Step 7: Update the background subtraction conducted for the frames.

Step 8: For each frame in the video repeat Steps 3–7.

Step 9: **End.**

---

The suggested approach verifies if the video content has been tampered with or not. This method takes into account certain environmental parameters, such as accurate background frames, quality, and ability to handle changes in illumination. Additionally, techniques for updating the background are needed to detect moving objects. The proposed extraction phase has been described in Algorithm 2, involves several steps. Those steps including designing a process for decomposing digital videos, identifying the positions where fragile watermarks should be inserted, and creating a matrix consisting of 0s and 1s that maps out where the watermarks should be embedded as secret keys. To prevent sudden background movements caused by external factors like water ripples, cloud movements, leaf movements, snow, and rain, a technique for extracting color features is used. These processes are illustrated in [Fig. 2](#).

---

**Algorithm 2:** Extraction Watermark Phase (Verifying Phase)

---

**Input:** Watermarked Video

**Output:** Original Video, Message Watermark

**Begin**

Step 1: Deconstruction of received watermarked video into frames.

Step 2: Divide each frame into equal size blocks ( $n \times n$ ).

Step 3: According to the received map matrix:

Step 3.1: For each frame, test for blocks marked by 1, then extract the embedded watermark from 2LSB of the block.

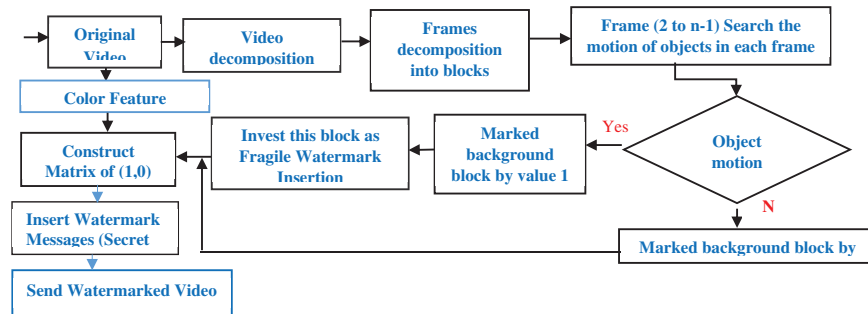
Step 3.2: Else omit this frame.

Step 3.3: Test if the number of extracted 1s = number of 1s in the map matrix, then video contents untampered with, else the video contents have been tampered with.

**End**

---





**Figure 2:** Overview of separation of foreground objects from image background

### 4.1 Watermark-Based Color Feature Extraction Method

The proposed method is based on the idea that in live videos, the background color of objects usually differs from the color of the foreground. Thus, in addition to intensity and color information, there are other factors that help differentiate the background and foreground of objects in frames. Each pixel  $P(X_c, Y_c)$  in a frame consists of three color components (R, G, and B). However, adding color information can increase the length of the binary bits, leading to an increase in pattern dimensions and a reduction in algorithm efficiency. To overcome this, the Color of Spatial of Binary Patterns local binary patterns (CS\_LBP) technique can be used to reduce patterns by utilizing central symmetry [11], choosing a small number  $K$ , and dropping one of the three color bits, which does not impact the intensity of the brightness since the three bits are highly chromatic.

The CS\_LBP utilized in this work is defined in Eq. (3), and the video features are described in Algorithm 3.

$$SCBP_{2N,R}(X_c, Y_c) = CS\_LBP_{2N,R}(X_c, Y_c) + 2^{N+1f}(R_c, G_c | \gamma) + 2^{N+2f}(G_c, B_c | \gamma) \quad (3)$$

where SCBP is a method that uses the average color of pixels in a region to represent its feature, CS\_LBP is Color of Spatial of Binary Patterns local binary patterns, and  $(X_c, Y_c)$  is the center of the pixel block. The total number of SC\_LBP patterns is 64 if the  $K$  value is set to 4. The histogram of SC\_BP is computed based on the radius of the circular region surrounding a pixel and is used to represent the pixel's feature vector. A background model is created using these feature vectors.

---

#### Algorithm 3: Extract Unique Video Features

---

**Input:** Video Sequence

**Output:** Feature Mapping to Matrix of Motion Object Background

**Begin**

Step 1: Video pre-processing.

A: Extract all frames from the video, except the first and last frames.

B: Convert frames to blocks.

Step 2: Apply color features Using Eq. (3).

Step 3: Detect object motion and determine the fragile region in the number of pixels of blocks in consecutive frames.

Step 4: Aggregate the values from Step 3 as a feature matrix.

Step 5: Embed the key in the fragile background of the moving object.

**End**

---

### 4.2 Background Design

To hide the secret watermark key in the frame background, the method computes the average movement rating of objects in the frames. This is done by using Eq. (4) to model the background.

$$B_i(x, y) = B_{i-1}(X, Y) + \frac{1}{t} (I_i(X, Y) - B_{i-1}(X, Y)) \tag{4}$$

where  $B_{i-1}(x, y)$  is the background of the previous frame,  $I_i(x, y)$  is the upcoming video frame, and  $t$  is the number of frames in the video sequence.

### 4.3 Background Subtraction

The detection model's output, as shown in Fig. 3, is a binary image that displays the background pixels' values. The threshold parameter  $T_p(x, y)$  is defined as the value of primitive  $T_p$ . Whenever the process of updating the background model and updating the threshold of the background is carried out, it is updated as shown in Eq. (5)

$$T_p(x, y) = (1 - \alpha) T_p(x, y) + \alpha (f(x, y) - 0.05) \tag{5}$$

where  $f(x, y)$  represents the similarity between the background histogram and the feature vector,  $T_p$  is a threshold and the learning rate  $\alpha$  is close to 1.

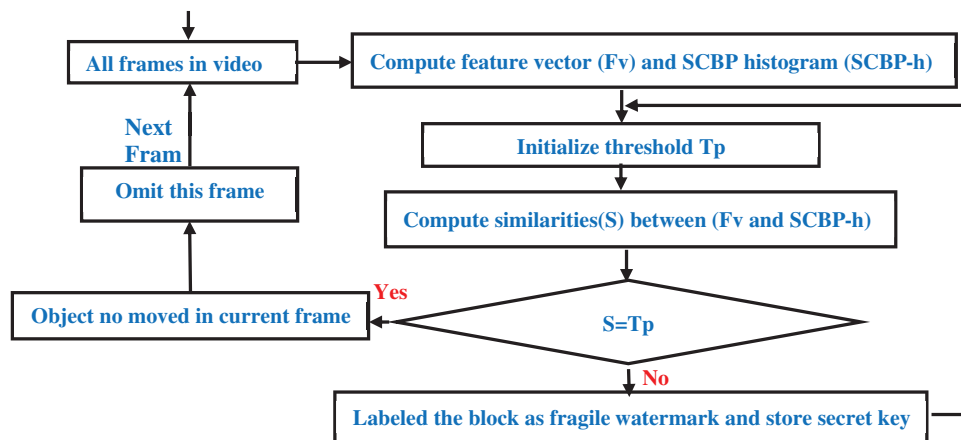
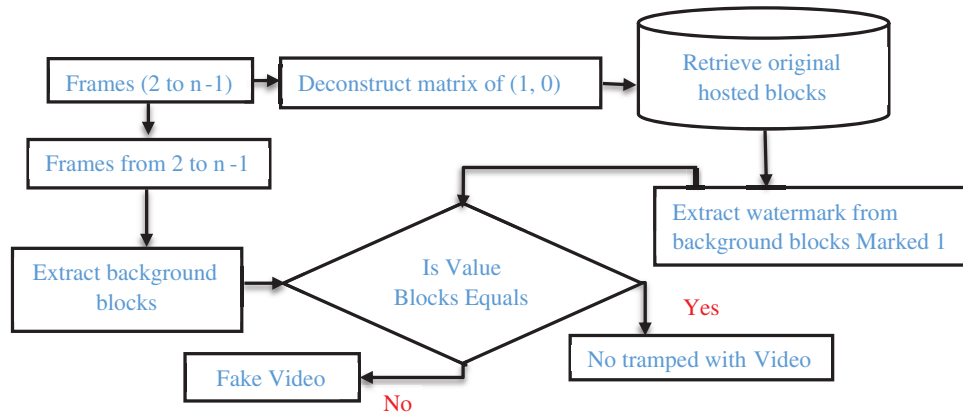


Figure 3: Background subtraction process

### 4.4 Digital Video Verification Process Design

This step of the proposed system involves checking the received video for tampering. The received video is loaded into the graphical user interface to undergo two types of operations: feature extraction and message extraction. Feature extraction is used to obtain unique features of the received video, while message extraction is used to obtain the user's secret key embedded in the video frames. Using the secret key, the original video features can be accessed. The extracted features from the received video are then compared with the original features to detect any tampering that may have occurred. The integrity of the submitted video is demonstrated in Fig. 4.





**Figure 4:** Proposed method to ensure the integrity of the submitted

#### 4.5 Validation Process

The proposed algorithm for video validation involves comparing the pixel values of original blocks with the decoding blocks of successive frames. To add a watermark to a video frame, it is first divided into blocks, and the watermark is added to each sub-block. During watermark retrieval, each sub-block is validated, and if any sub-block does not contain the correct watermark, the entire block is considered tampered with. This ensures that the entire video is secure and that any tampering can be detected.

### 5 Watermark Based H.264 Method

The proposed method utilizes the block size motion compensation feature of H.264/AVC codec, which allows for an accurate representation of motion in a macro block and also provides opportunities for watermark embedding. In traditional video codecs, motion compensation is done using a fixed block size, while H.264/AVC uses variable block sizes. The codec compresses temporal redundancy between frames by estimating motion vectors and using them in inter-frame coding. The watermark is embedded by adding it to the original motion vectors, computing the new prediction error, and encoding the new motion vectors along with updated prediction errors into compressed bit streams. This allows for watermark embedding without the need to decompress the video. The proposed method is based on this H.264/AVC technique and will be implemented in the following sections.

#### 5.1 Feature Extraction Based on H.264

The precise identification of the location of a fragile watermark can be used to indicate where tampering has occurred. In order to achieve effective and efficient tampering detection, it is important to have a well-designed localization feature, as shown in Fig. 5. In Fig. 5a, the video frame is divided into multiple rectangular groups, with 9 groups per frame in Quarter Common Intermediate Format (QCIF) format ( $176 \times 144$  pixels). These groups are further divided into three subgroups, each containing different macro-blocks. Since the dividing process of frames into blocks is interested in video frames that are typically located in the central area of the frame, where each three macroblocks are grouped together to form one subgroup when searching in the central area, and 4 macroblocks together when searching in the left or right subgroup. Additionally, each H.264 macroblock contains 16 pieces, as specified in the Discrete Cosine Transform (DCT) used in the H.264/AVC standard, as

shown in Fig. 5b. Using a private key, can able to select a coefficient at random from each DCT piece, resulting in 16 coefficient values for each macroblock. These 16 coefficients are then subjected to an Exclusive-Or (XOR) operation, producing a single coefficient value. The watermark for each subgroup coefficient is generated by performing XOR on the previously XORed coefficients of the macroblocks within the subgroup. The system then searches for object movement and detects fragile regions based on the number of pixels of blocks in consecutive frames. The resulting values are aggregated into a feature matrix and the key is then extracted from the fragile background of the moving object.

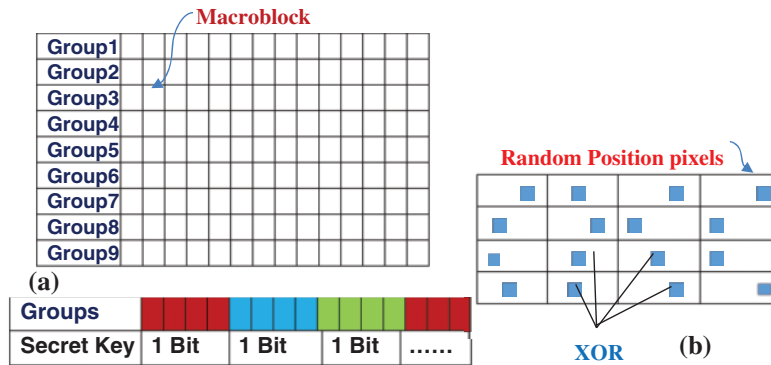


Figure 5: (a) Block division and (b) generation of authentication watermark

### 5.2 Watermark Embedding Based on H.264

Watermark embedding has been suggested that the watermark is carried using motion vectors. Previous approaches to watermark embedding using motion vectors have relied on intuition rather than statistical analysis to select embedding locations. In contrast, our proposed method utilizes statistics, such as the Bjontegaard Delta PSNR (BD-PSNR) used by the JVT group to evaluate the PSNR differences between the RD curves of two calculations' bit rates. In H.264, the encoder conveys not only the motion vector but also the contrast value of the vector (Delta MV) between the motion vector (MV) and its expected value based on surrounding blocks. Where the MVs are categorizing into seven classes based on its length and perform watermarking embedding independently on each class to assess PSNR degradation and bit rate increase. The resulting BD-PSNR values are computed and plotted in Fig. 6, showing that certain MV classes, such as |MV| between 2 and 22, are better for embedding due to less BD-PSNR degradation. Therefore, the final embedding location are selected from MVs in the 1 to 10 length range using a secret key and frame numbers.

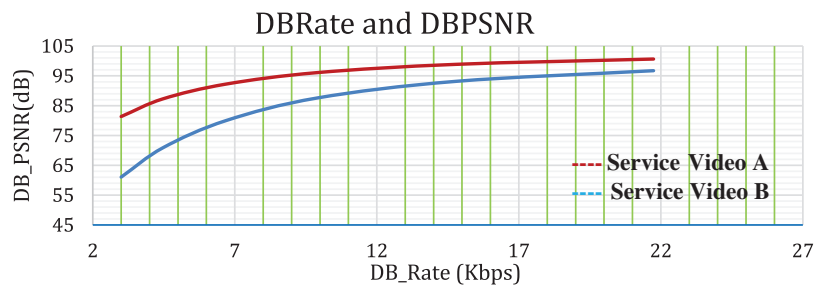


Figure 6: Selecting embedding position by using BD-Rate and BD-PSNR

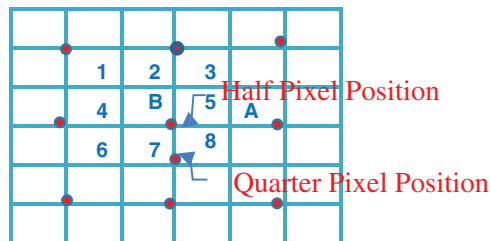
Watermarking often involves modifying and adjusting the value of the  $MV$  or motion differential vector during the embedding process. To achieve this, to determining the typical motion vector offset a criterion must be established. This is accomplished by minimizing the cost function of the Lagrangian RD and H.264/AVC, and the resulting value can be computed using Eq. (6).

$$J = \min \{D(S_i, MV) + \lambda R(MVD)\} \tag{6}$$

where  $D$  and  $R$  represent distortion and rate, respectively, and are used to calculate the corresponding Lagrange multiplier. The value of  $J = l$  is used by H.264 to select the optimum mode and MV for that mode. To embed the watermark, the value of MV must obtain firstly, and then calculate its length. Then, Eq. (7) is used to insert and embed the watermark into the MV.

$$\begin{aligned} & \text{if } ((MV_x \% 2) \text{ XOR } (MV_y \% 2)) \neq WM \text{ } (MV'_x, MV'_y) = (MV_x, MV_y); \\ & \text{or } (MV'_x, MV'_y) = (MV_x + 1, M V_y); \\ & \text{or } (MV'_x, MV'_y) = (MV_x, MV_y + 1); \\ & \text{or } (MV'_x, MV'_y) = (MV_x, MV_y - 1); \text{ else} \\ & MV'_x = MV_x; MV'_y = MV_y \end{aligned} \tag{7}$$

where  $MV'_x$  and  $MV'_y$  represent the watermarked  $MV_x$  and  $MV_y$ , respectively, and  $WM$  represents the two-bit watermark to be embedded. If the condition is satisfied, then the MV remains unchanged after embedding the watermark. Otherwise, it is need to modify the MV in an RD sense by adjusting either the horizontal component  $MV_x$  or the vertical component  $MV_y$ . In Fig. 7 the optimal quarter-pixel MV is computed in area 7 using (1). However, it is need to capture the half-pixel motion vector during the motion search (in area B). If (2) is true, then can embed the watermark in area 7 without modifying the MV. Otherwise, it is need to need to modify the MV by replacing it with  $MV'$ , which is determined by finding the point with the lowest value of (1) among B, 1, 3, 6, or 8.



**Figure 7:** Fractional modification position

### 5.3 Watermark Detection Based on H.264 Method

To detect the watermark using the suggested fragile method, a specific video is needed for extracting the watermark, which can also result in difficult detection. The watermark detection process involves several steps:

- Identify watermarking areas that utilize the same calculation as the embedding preparation.
- Determine the watermark information using Eq. (8).

$$WM = (MV'_x \% 2) \text{ XOR } (MV'_y \% 2) \tag{8}$$

### 5.4 Watermark Verification

The previous frame can be verified by comparing the watermarks that have been detected in the current frame with the extracted features, as illustrated in Fig. 8.

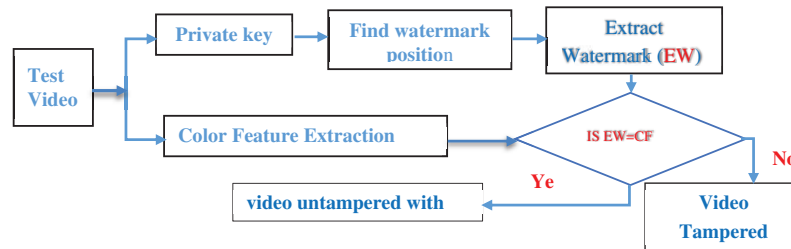


Figure 8: Watermark verification

## 6 Experimental Results and Analysis

To evaluate the proposed watermarking approach, where the H.264/AVC JM9.2 computer program [9] and other assistance programs as references, it is need to tested the approach on several video sequences, including Foreman (QCIF), News (QCIF), Silent (QCIF), Container (QCIF), and Mobile (CIF), among others. To comply with the H.264 baseline profile, it is need to use the IPP H.264 Encoder to encode the video frames with QP values of 24, 28, 32, 36, and 40. And the results, shown in Table 1 and Fig. 9, demonstrate the proposed method outperforms three other motion-based watermarking strategies proposed by Hammami [24] in terms of RD performance metrics such as Bitrate/Kbps, PSNR/dB, and RD.

Table 1: PSNR and bit rate twisting

Sequences		Foreman		News		Mobile		Container		Silent	
(QP)		29	35	25	33	26	31	28	36	28	36
Non-WM	PSNR	35.97	30.69	36.70	30.65	33.82	35.94	35.94	30.39	35.74	30.24
	BR	81.80	28.62	47.73	17.85	736.43	27.68	27.68	8.42	63.62	21.50
Sang et al. [8]	PSNR	35.74	30.53	36.62	30.58	33.74	35.86	35.86	30.30	35.60	30.17
	BR	88.90	29.84	54.58	19.09	876.56	36.88	36.88	9.65	70.58	22.26
Zhang et al. [28]	PSNR	35.76	30.58	36.65	30.60	33.75	35.84	35.84	30.30	35.61	30.15
	BR	87.34	29.56	52.13	18.82	866.84	36.36	36.36	9.65	70.17	22.26
Qin et al. [21]	PSNR	35.75	30.57	36.62	30.61	33.74	35.86	35.86	30.33	35.62	30.17
	BR	86.93	29.49	51.95	18.68	851.74	35.15	35.15	9.70	69.80	22.19
Proposed WM	PSNR	35.76	30.60	36.68	30.61	33.75	35.86	35.86	30.39	35.64	30.19
	BR	85.46	29.21	49.68	18.33	824.80	33.01	33.01	9.32	68.39	22.15

Fig. 9 also illustrates the proposed watermarking approach achieves the highest quality of RD compared to other methods. This is because the proposed method has been carefully designed to

embed the watermark in a specific and optimal location, based on statistical analysis. Additionally, the proposed method is more effective since all potential directional offsets of the motion vector for watermark insertion has been used.

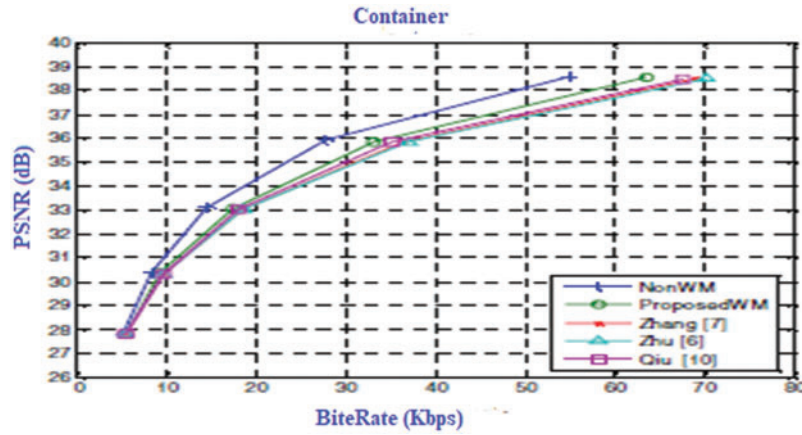


Figure 9: Rate distortion curves

### 6.1 Execution Time Estimation

The implemented process of inserting and extracting the watermark using programs written in the C# programming language. The operations were carried out on a computer with Windows 10 (64 bits) and an Intel(R) Core i5-4200 m CPU@2.50 Ghz 2.49 processor. The execution times for different video sizes are presented in Table 2.

Table 2: Embedding time and extraction processes for watermark

Frames #	Videos	Frame size	Insertion time(s)/average	Extraction time(s)/average
1-60	Foreman	256 × 256	0.1611	0.1732
1-60	News	256 × 256	0.2965	0.1372
1-60	Mobile	512 × 512	0.2231	0.1413
1-60	Container	512 × 512	0.2342	0.1397
1-60	Silent	512 × 512	0.2510	0.1547
1-60	Paris2	1024 × 1024	0.4989	0.2564
1-60	All average/s		<b>0.2774</b>	<b>0.1670</b>

In terms of testing and evaluating the elapsed time, the experimental tests were carried out using two binary messages that were inserted as watermarks during the embedding and extraction process, and the results were calculated by applying the watermark to the selected frames of the video. In order to examine the effect of time, experimental time complexity was carried out. It is taken that the embedded time value is the same for both the first watermark and the second watermark. and show that the embedded time value is proportional directly to selected frame numbers. A total of 6 frames

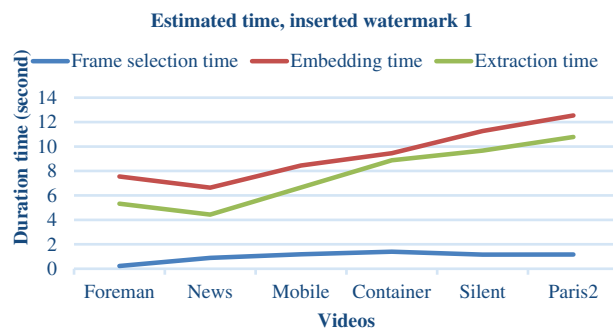
were specified from pure storage video; Thus, the total embed time is the highest for the same video. Tables 3 and 4 collect the processing time (in seconds) required to perform frame selection, embedding, and watermark extraction for a given set of video frames. The results proved that the time depends entirely on processor specifications and the selection of frames from the video. Figs. 10 and 11 present the estimation time for embedding and Extracting of deferent videos when two watermarks are inserted.

**Table 3:** Elapsed time (in second) of 6 videos based watermark message one

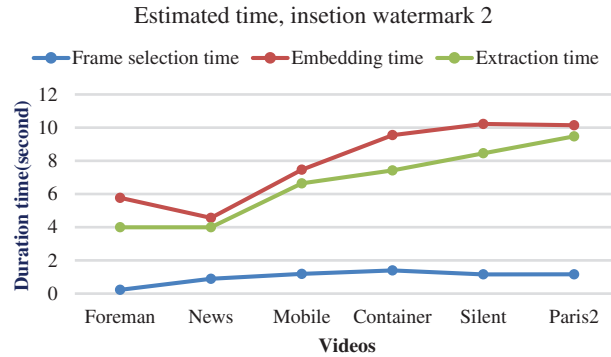
Video	Frame selection time	Embedding time	Extraction time
Foreman	0.22735	7.5461	5.321
News	0.89223	6.6366	4.432
Mobile	1.18576	8.4525	6.654
Container	1.39662	9.4586	8.873
Silent	1.15627	11.2723	9.672
Paris2	1.16236	12.5432	10.784

**Table 4:** Elapsed time (in second) of 6 videos of 6 videos based watermark message two

Video	Frame selection time	Embedding time	Extraction time
Foreman	0.22735	5.7689	3.995
News	0.89223	4.5636	3.998
Mobile	1.18576	7.4657	6.642
Container	1.39662	9.5486	7.423
Silent	1.15627	10.2234	8.454
Paris2	1.16236	10.1422	9.476



**Figure 10:** Estimation time for embedding and extracting of deferent videos: Watermark 1



**Figure 11:** Estimation time for embedding and extracting of deferent videos: Watermark 2

## 6.2 Performance Metrics

To assess the effectiveness of the proposed method, several standard performance metrics, including PSNR, SSIM, and bit correction ratio (BCR) are used. These metrics were used to evaluate the imperceptibility, robustness, and quality of watermarked video frames. These metrics are commonly used in watermarking and are defined by Eqs. (9)–(13). Specifically, the PSNR metric used to measure the quality of the original video frame and the frame on which the watermark was inserted, as described in Eq. (9).

$$PSNR = 10 \log_{10} \frac{Max^2}{MSE} \quad (9)$$

where  $Max$  is the maximum pixel value of the original sequence frames in the original video, and  $MSE$  (Mean Square Error) is the error between two  $h \times w$  frames (that is, the original frames and the watermark) as specified in Eq. (10).

$$MSE = \frac{1}{m^2} \sum_{i=1}^m \sum_{j=1}^m [I_1(i, j) - I_2(i, j)]^2 \quad (10)$$

To extract the watermark from a watermarked video frame, the value of PSNR determined by  $\geq 65$  dB indicates good quality frame reconstruction, as described in Eq. (9). Additionally, the metric SSIM metric has been used, as defined in Eq. (11), to assess the perceptual quality degradation produced by watermark insertion. SSIM quantifies the perceived difference between two similar frames and generates a quality reference by comparing the original and modified frames.

$$SSIM = \frac{(2\mu_1\mu_2 + K_1)(2\sigma(F_1, F_2) + K_2)}{(\mu_1^2 + \mu_2^2 + K_1)(\sigma(F_1)^2 + \sigma(F_2)^2 + K_2)} \quad (11)$$

where  $K_1$  and  $K_2$  are constants used to assure stability when the denominator equals 0,  $\mu_1$ , and  $\mu_2$  represent mean values, and  $\sigma$  is the values of variance frames  $F_1$  and  $F_2$ .

To evaluate the accuracy of the extracted watermark bits, the BCR (Benefit Cost Ratio) metric are used, the metric compares the two sequences of digital binary frames: the inserted data and the extracted watermarks ( $B$ ,  $B'$ , respectively). BCR is defined as the number of bits that were correctly extracted as a percentage of the total number of bits that were embedded in the host frame, as described in Eq. (12).



$$BCR = \frac{1}{n} \sum_{l=0}^{n-1} (B(l) \oplus B'(l)) \times 100 \quad (12)$$

where  $n$  is the bit's sequence length and  $\oplus$  represents the XOR operator. The BCR value is 100%. According to Eq. (12), if the extracted watermark contains no errors, the BCR accuracy value will be 100%.

### 6.3 Effective Gain Index

When performing image watermarking, there is a trade-off between embedding and robustness. To strike a balance between these qualities, an appropriate gain index factor (BCR) value should be chosen for embedding the watermark. In experimented results three different gain factor values has been tested ( $\lambda = 0.1, 0.2,$  and  $0.3$ ) to measure the invisibility and robustness of the video frame watermarking, as shown in Table 5. The table displays the different gain factor values used and their impact on the watermark's visibility and robustness.

**Table 5:** Fidelity-based criteria for PSNR (dB), SSIM, and BCR (%)

Gain Factor	$\lambda = 0.1$			$\lambda = 0.2$			$\lambda = 0.3$		
	PSNR (dB)	SSIM	BCR	PSNR	SSIM	BCR	PSNR	SSIM	BCR
Foreman	45.9543	0.989	99.99	42.176	0.974	99.99	38.699	0.916	99.99
News	45.8942	0.976	99.99	42.393	0.981	99.99	38.583	0.924	99.99
Mobile	44.9687	0.996	99.99	41.164	0.974	99.99	38.438	0.962	99.99
Container	41.8565	0.985	99.99	36.458	0.895	99.99	32.561	0.773	99.99
Silent	47.8562	0.9867	99.998	41.8721	0.9633	99.998	36.4462	0.9528	99.998
Paris2	45.6794	0.9899	99.998	42.2128	0.9861	99.998	38.6801	0.9367	99.998
Average	45.368	1.1517	99.999	41.0471	0.9921	99.998	43.5626	0.9829	99.998

### 6.4 Robustness and Attack Analysis

To compare the extracted watermarks with the original frames, the normalized correlation (NC) metric is used, which produces values between 0.0 and 1.0. Eq. (13) describes the calculation of this metric.

$$NC = \frac{\sum_{i=1}^M \sum_{j=1}^N W(i,j) W'(i,j)}{\sum_{i=1}^M \sum_{j=1}^N W(i,j)^2} \quad (13)$$

When the NC value approaches 1.0, it means that the extracted watermark is very similar to the original one. Table 6 compares the PSNR and NC values of the proposed watermark extraction technique with other conventional algorithms. It is clear that the proposed technique outperforms the conventional algorithms in terms of watermark quality. To test the robustness of the proposed technique, various attacks were applied to the watermarked frames, including gamma correction, contrast adjustment, Gaussian noise, rotation, cropping, MPEG4 compression, resizing, JPEG compression, noise addition, filtering, frame averaging, histogram equalization, and frame dropping. Table 6 shows the results of most of these attacks.

**Table 6:** Comparison of the proposed technique with other algorithms (Attacks)

Attack	Attack frame/PSNR (dB)	Extracted watermark/ (NC)	Attack	Attack frame/PSNR	Extracted watermark/ (NC)
Gamma correction (0.5)	14.399	0.982	Rotate 15°	17.243	0.812
Gamma correction (1.5)	15.817	0.974	Cropping	6.2982	0.864
Gamma correction (2)	11.304	0.895	Gaussian noise	29.647	0.867
Resizing	41.217	1.0	Rotate 15°	17.243	0.812
Proposed algorithms				65.45	0.998

### 6.5 Security Analysis

In order to analyse the security of the watermarked video it must be subjected to different video processing such as Gaussian, Poisson, and Speckle noise, and Salt & Pepper. Its effect is similar to what hackers do to remove the watermark embedded in video content. Therefore, the proposed watermarked method has been evaluated and tested with various attacks having numerous parameters until the quality of the watermarked videos is acceptable according to the known quality metrics. (explained in [Section 4.5](#)). Those attacks have been implemented by inserting specific noise ratios on watermarked video frames, the selected attacks are Gaussian, Poisson, Salt & Pepper, and Speckle noise. Where Gaussian noise and Speckle noise attacks have been implemented with variances (0.001 to 0.002 and 0.001 to 0.003), respectively, the watermarked video frames were subject to a Poisson noise attack to evaluate them, and then the original video is effectively extracted by the proposed watermarking method, Salt and pepper noise attack implemented with noise density 0.001 and 0.002 on the watermarked video frames. The outcomes reflect that the efficiency, performance, and strength of the proposed method for video frames are promising, where original video frames are effectively extracted by the proposed watermarking method. The effectiveness of attacks is shown in [Tables 7–10](#), respectively.

**Table 7:** Extracting watermarks using Gaussian noise attack with variance values (0.001 and 0.002)

Variance	NC/Frame	Q/Frame	PSNR/Frame	Watermark extracted (%)
0.001	1.8170	0.987621	36.2421	100
0.002	1.7124	0.986532	35.6173	100

**Table 8:** Applying Poisson noise attack to extract watermarks from watermarked frames

NC/Frame	Q/Frame	PSNR/Frame	Watermark extracted (%)
0.89986	0.89977	45.3231	100

**Table 9:** Extracting watermarks using Salt and pepper noise attack with variance values (0.001 and 0.002)

Noise Density	NC/Frame	Q/Frame	PSNR/Frame	Watermark extracted (%)
0.001	0.88732	0.98732	28.21321	100
0.002	0.88213	0.987423	26.60606	100

**Table 10:** Extracting watermarks using Speckle noise attack with variance values (0.001–0.003)

Variance	NC/Frame	Q/Frame	PSNR/Frame	Watermark extracted (%)
0.001	0.897434	0.899972	40.32111	100
0.002	0.88675	0.996750	37.23476	100
0.003	0.89765	0.989865	37.49998	100

## 6.6 Discussion

To validate the effectiveness of the proposed approach, the Bit Error Rate (BER) was used as another performance metric in addition to the BCR measure discussed in Eq. (12) and Table 4. Table 5 presents the BER values obtained by the Mostafa, Wand, Nisreen methods [29], and the proposed method. The results demonstrate that the proposed method outperforms the other methods, as shown in Table 11.

**Table 11:** Comparison of bit error rates for different attacks

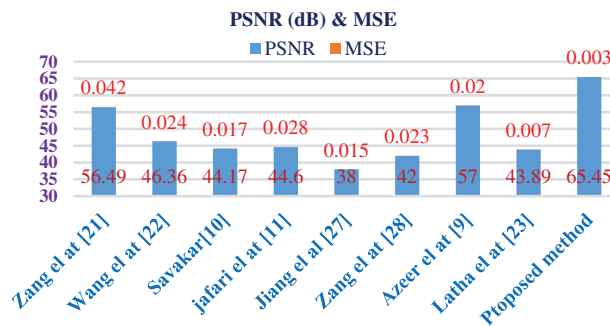
Attack	Wand [21]	Mostafa [21]	Nisreen [21]	Proposed method
Gaussian noise (0.05)	5%	42%	40.6%	41%
Smoothing	25%	3%	22%	6%
Salt & pepper noise (0.05)	5%	46%	37.7%	47%
Sharpening	17%	2%	1.5%	3%
Rotate 0.4°	25%	20%	49%	25%

## 6.7 Comparing Suggested Method to Other Methods

A comparison was performed between the suggested work's results employing fragile watermarks to identify tampered with and those of previously used approaches. The major two measurements (PSNR and MSE) have been tested and it becomes clear the work resulted has been given better quality and a higher PSNR value than previous methods. Table 12 and Fig. 12 show the result comparisons.

**Table 12:** Comparing proposed method with other methods

References	Techniques used	PSNR (dB)	MSE
Qin et al. [21]	Overlapping embedding strategy	56.49	0.042
Wang et al. [22]	Robust reversible watermarking	46.36	0.024
Savakar et al. [10]	Hybrid scheme	44.17	0.017
Jafari Barani et al. [11]	Newton complex map	44.6	0.028
Sun et al. [27]	Bit stream domain	38	0.015
Zhang et al. [28]	Discrete Fourier transform	42	0.023
Azeez et al. [9]	Watermark technique	57	0.020
Latha et al. [23]	Neural networks	43.89	0.007
Proposed method	H.264 and color extraction	65.45	0.003



**Figure 12:** PSNR and MSE contrasting the proposed method with other methods

## 7 Conclusion

A new technique was developed to create a watermarking scheme that is both robust and imperceptible. This technique uses a fragile watermark to conceal the embedded message by leveraging the instantaneous movement of objects between consecutive video frames. The purpose of this approach is to prevent any attempts to tamper with or modify digital video content while also ensuring its authenticity. The movement of objects is tracked using two algorithms, H.264 and color feature extraction. The results of the experiments indicate that the proposed system is highly accurate and effective in identifying various moving objects in video frames. The accuracy of object identification ranged between 96.3% and 98.7% even when subjected to different attacks. The rate of hiding the fragile watermark to insert the secret key was more than 93.65%, particularly in backgrounds with a fixed color level such as water or sky. The average capacity of hiding was 40.67%, the average quality of retrieved videos was high (PSNR = 65.45), and the average time difference was very acceptable (T = 0.1670/s).

## 8 Contribution and Future Work

The proposed scheme can be applied to help combat piracy by identifying malicious users that illegally distribute videos, even when they attempt to lower the video quality significantly. The research can provide a means for investigating and verifying copyright integrity and security of various digital materials, including digital content, music, photos, and videos, for various institutions.

The experiments were only conducted and verified on a small number of video types and limited video formats, and whether the proposed method is suitable for a limited number of videos, so, it is still up for debate. There are many untested watermarking methods and each method has its scope of application. Whether there is a better-watermarked approach to copyright protection for the contents of videos transmitted via media is worth exploring. There are many trends that can be continued to study in the future.

**Acknowledgement:** The authors would like to acknowledge the Department of Computer Science, University of Technology, Baghdad, Iraq, for providing moral support for this work. The publishing fees have been paid by the authors.

**Funding Statement:** The authors did not receive any specific funding for this study. The study was funded from the authors' own funds.

**Author Contributions:** Conceptualization and design: Raheem Ogla, Abdul Monem S. Rahma, Eman Shakar Mahmood; software: Raheem Ogla, Rasha I. Ahmed; validation: Rasha I. Ahmed, Raheem Ogla; formal analysis: Raheem Ogla, Eman Shakar Mahmood; data curation: Rasha I. Ahmed, Abdul Monem S. Rahma; writing—original draft preparation: Raheem Ogla, Eman Shakar Mahmood, Rasha I. Ahmed; writing—review and editing: Raheem Ogla, Rasha I. Ahmed; visualization: Raheem Ogla, Abdul Monem S. Rahma; analysis and interpretation of results: Raheem Ogla, Eman Shakar Mahmood, Rasha I. Ahmed; supervision: Raheem Ogla, Abdul Monem S. Rahma; project administration: Raheem Ogla, Eman Shakar Mahmood, Rasha I. Ahmed; funding acquisition: All authors; draft manuscript preparation: Abdul Monem S. Rahma; All authors have reviewed the results and approved the final version of the manuscript.

**Availability of Data and Materials:** Not applicable.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] K. J. Giri, Z. Jeelani, J. I. Bhat and R. Bashir, "Survey on reversible watermarking techniques for medical images," In K. J. Giri, S. A. Parah, R. Bashir and K. Muhammad (Eds.), *BT—Multimedia Security: Algorithm Development, Analysis and Applications*, pp. 177–198, Singapore: Springer Singapore, 2021. [https://doi.org/10.1007/978-981-15-8711-5\\_9](https://doi.org/10.1007/978-981-15-8711-5_9)
- [2] M. Asikuzzaman and M. R. Pickering, "An overview of digital video watermarking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2131–2153, 2017.
- [3] M. Tanha, S. D. S. Torshizi, M. T. Abdullah and F. Hashim, "An overview of attacks against digital watermarking and their respective countermeasures," in *Proc. Title: 2012 Int. Conf. on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec)*, Kuala Lumpur, Malaysia, pp. 265–270, 2012. <https://doi.org/10.1109/CyberSec.2012.6246095>

- [4] X. Yu, C. Wang and X. Zhou, "A survey on robust video watermarking algorithms for copyright protection," *Application Science* 2018, vol. 8, no. 10, pp. 1891, 2018.
- [5] C. Sharma, A. Bagga, B. K. Singh and M. Shabaz, "A novel optimized graph-based transform watermarking technique to address security issues in real-time application," *Mathematical Problems in Engineering*, vol. 2021, no. 4, pp. 1–27, 2021.
- [6] L. Sharma, A. Anand, N. K. Trivedi, M. Sharma and J. Singh, "Digital video watermarking: Features, techniques, and challenges," *Annals of the Romanian Society for Cell Biology*, vol. 25, no. 2, pp. 3376–3385, 2021.
- [7] N. Hasan, M. S. Islam, W. Chen, M. A. Kabir and S. Al-Ahmadi, "Encryption based image watermarking algorithm in 2DWT-DCT domains," *Sensors*, vol. 21, no. 16, pp. 5540, 2021.
- [8] J. Sang, Q. Liu and C. L. Song, "Robust video watermarking using a hybrid DCT-DWT approach," *Journal of Electronic Science and Technology*, vol. 18, no. 2, pp. 100052, 2020.
- [9] R. A. Azeez, M. K. Abdul-Hussein, M. S. Mahdi and H. T. H. S. ALRikabi, "Design a system for an approved video copyright over cloud based on biometric iris and random walk generator using watermark technique," *Periodicals of Engineering and Natural Sciences*, vol. 10, no. 1, pp. 178–187, 2021.
- [10] D. G. Savakar and A. Ghuli, "Robust invisible digital image watermarking using hybrid scheme," *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3995–4008, 2019.
- [11] M. Jafari Barani, P. Ayubi, M. Yousefi Valandar and B. Yosefnezhad Irani, "A blind video watermarking algorithm robust to lossy video compression attacks based on generalized Newton complex map and contourlet transform," *Multimedia Tools and Applications*, vol. 79, no. 3, pp. 2127–2159, 2020.
- [12] P. Kadian, S. M. Arora and N. Arora, "Robust digital watermarking techniques for copyright protection of digital data: A survey," *Wireless Personal Communications*, vol. 118, no. 4, pp. 3225–3249, 2021.
- [13] M. K. Hasan, S. Kamil, M. Shafiq, S. Yuvaraj, E. S. Kumar *et al.*, "An improved watermarking algorithm for robustness and imperceptibility of data protection in the perception layer of internet of things," *Pattern Recognition Letters*, vol. 152, pp. 283–294, 2021. <https://doi.org/10.1016/j.patrec.2021.10.032>
- [14] P. Ayubi, M. Jafari Barani, M. Yousefi Valandar, B. Yosefnezhad Irani and R. Sedagheh Maskan Sadigh, "A new chaotic complex map for robust video watermarking," *Artificial Intelligence Review*, vol. 54, no. 2, pp. 1237–1280, 2021.
- [15] Q. Wang, Y. Bai, W. D. Chen, X. H. Pan, S. K. Wang *et al.*, "An overview on digital content watermarking," in *BT—Signal and Information Processing, Networking and Computers*, pp. 1311–1318, Singapore: Springer, 2023.
- [16] C. Kant and S. Chaudhary, "A watermarking based approach for protection of templates in multimodal biometric system," *Procedia Computer Science*, vol. 167, pp. 932–941, 2020.
- [17] K. U. Singh, H. S. Abu-Hamatta, A. Kumar, A. Singhal, M. Rashid *et al.*, "Secure watermarking scheme for color DICOM images in telemedicine applications," *Computers, Materials & Continua*, vol. 70, no. 2, pp. 2525–2542, 2021.
- [18] K. Ait Sadi, A. Guessoum, A. Bouridane and F. Khelifi, "Content fragile watermarking for H.264/AVC video authentication," *Electron International Journal of Electronics*, vol. 104, no. 4, pp. 673–691, 2017.
- [19] M. E. Farfoura, S. J. Horng, J. M. Guo and A. Al-Haj, "Low complexity semi-fragile watermarking scheme for H. 264/AVC authentication," *Multimedia Tools and Applications*, vol. 75, no. 13, pp. 7465–7493, 2016.
- [20] Y. Li and H. X. Wang, "Robust H. 264/AVC video watermarking without intra distortion drift," *Multimedia Tools and Applications*, vol. 78, no. 7, pp. 8535–8557, 2019.
- [21] C. Qin, P. Ji, X. Zhang, J. Dong and J. Wang, "Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy," *Signal Processing*, vol. 138, pp. 280–293, 2017.
- [22] X. Wang, X. Li and Q. Pei, "Independent embedding domain based two-stage robust reversible watermarking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 8, pp. 2406–2417, 2019.
- [23] S. B. Latha, D. V. Reddy and A. Damodaram, "Video watermarking using neural networks," *International Journal of Information and Computer Security*, vol. 14, no. 1, pp. 40–59, 2021.

- [24] 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, no. 5, pp. 7479–7513, 2021.
- [25] S. A. Hasso and T. B. Taha, "A new tamper detection algorithm for video," *Journal of Engineering Science and Technology*, vol. 15, no. 5, pp. 3375–3387, 2020.
- [26] N. Janu, A. Kumar, P. Dadheech, G. Sharma, A. Kumar *et al.*, "Multiple watermarking scheme for video & image for authentication & copyright protection," *IOP Conference Series: Materials Science and Engineering*, vol. 1131, no. 1, pp. 1–12, 2020.
- [27] J. Sun, X. Jiang, J. Liu, F. Zhang and C. Li, "An anti-recompression video watermarking algorithm in bitstream domain," *Tsinghua Science and Technology*, vol. 26, no. 2, pp. 154–162, 2020.
- [28] X. Zhang, Q. Su, Z. Yuan and D. Liu, "An efficient blind color image watermarking algorithm in spatial domain combining discrete Fourier transform," *Optik*, vol. 219, pp. 165272, 2020.
- [29] N. I. Yassin, N. M. Salem and M. I. El Adawy, "Block based video watermarking scheme using wavelet transform and principle component analysis," *International Journal of Computer Science Issues*, vol. 9, no. 1, pp. 296, 2012.