CUD Digital Repository

Title (Article)	A new watermarking scheme for digital videos using DCT
Author(s)	Al-Gindy, Ahmed; Omar, Aya Al-Chikh; Mashal, Omar; Shaker, Yomna; Alhogaraty, Eslam; and Moussa, Sherif
Journal Title	*Open Computer Science*
Citation	Al-Gindy, A., Omar, A. A. -., Mashal, O., Shaker, Y., Alhogaraty, E., & Moussa, S. (2022). A new watermarking scheme for digital videos using DCT. *Open Computer Science*, 12(1), 248-259. doi:10.1515/comp-2022-0238
Link to Publisher Website	https://doi.org/10.1515/comp-2022-0238
Link to CUD Digital Repository	http://hdl.handle.net/20.500.12519/693
Date added to CUD Digital Repository	August 23, 2022
Term of Use	Creative Commons Attribution 4.0 International (CC BY 4.0) License

Research Article

Ahmed Al-Gindy*, Aya Al-Chikh Omar, Omar Mashal, Yomna Shaker, Eslam Alhogaraty, and Sherif Moussa

A new watermarking scheme for digital videos using DCT

https://doi.org/10.1515/comp-2022-0238
received April 4, 2021; accepted March 15, 2022

Abstract: With the advent of high-speed broadband Internet access, the need to protect digital videos is highly recommended. The main objective of this study is to propose an adaptive algorithm for watermarked digital videos in the frequency domain based on discrete cosine transform (DCT). The watermark signature image is embedded into the whole frame of the video. The green channel of the RGB frame is selected for the embedding process using the DCT algorithm as it shows the recommended quality of the watermarked frames. The experiment results indicate that the proposed algorithm shows robustness and high quality of the watermarked videos by testing various strength values *Δ* for different videos. It offers resistance against different types of attacks.

Keywords: watermarking, DCT, digital videos, attacks, robustness

1 Introduction

In the quickly developing technological age, intellectual property has become an enormous concern as information can be distributed around the globe in a matter of seconds. Corporations and individuals alike constantly seek ways to protect their intellectual property without having the broad Internet reach dwindle a project's potential or idea. One solution that has dramatically grown in development over the last few decades is the tool of watermarking. The basis of watermarking involves adding a layer over media files (whether visible or invisible) that can be used to identify ownership of the work and deduce if it was exploited. The development of watermarking over the years has come with many challenges related to robustness, imperceptibility, and complexity, with watermarking developers are attempting to find the perfect balance to create a solid and efficient watermark as much as possible. This study delves into the more niche world of video watermarking. Content creators need a concrete way to protect their videos without diminishing quality or requiring massive amounts of storage/processing speed.

In software development, the biggest challenge comes not from building a program but the hackers attempting to find the holes in the weak secured program. When developing a watermark, people will always break it either for profit or free distribution online. Piracy plagues the internet, and media companies bleed income as internet pirates allow for the free consumption of their movies or TV programs. Video piracy involves stealing, copying, or otherwise acquiring a video without the owner's consent, infringing copyright law, and distributing it illegally. The Internet has made video piracy wildly easy and accessible to the point where it is near impossible to police and, in some cases, has become more viable to consumers than buying legitimate copies of media.

Video watermarking is a viable solution to video piracy as many layers of protection are involved in the process. Watermarking can be used to claim ownership, discover piracy, track pirates, block playback, or prove that a file has been tampered. Every day, there are advancements in the technological development of video watermarking and new solutions that intend to thwart hackers attempting to remove watermarks, though there are different challenges related to each of these methods [1].

* **Corresponding author: Ahmed Al-Gindy,** Department of Electrical Engineering, Canadian University Dubai, City Walk, Dubai, 117781, United Arab Emirates, e-mail: agindy@cud.ac.ae
Aya Al-Chikh Omar: Department of Electrical Engineering, University of Sharjah, Sharjah, UAE
Omar Mashal: Department of Electrical Engineering, Canadian University Dubai, City Walk, Dubai, 117781, United Arab Emirates
Yomna Shaker: Department of Electrical Engineering, University of Science and Technology of Fujairah, Fujairah, United Arab Emirates
Eslam Alhogaraty: Department of Electrical Engineering, Al Dar University College, Dubai, United Arab Emirates
Sherif Moussa: Department of Electrical Engineering, Canadian University Dubai, City Walk, Dubai, 117781, United Arab Emirates

Some of the popular video watermarking applications include:

Copyright protection, where the owner of a video embeds their copyright information into a video either visibly or invisibly (the most common type of watermarking that jumps to mind).

Fingerprinting is an encryption process where a unique code identifies the owner. This method is aimed mainly toward cinemas as, if there is a leak, the specific cinema can be caught and the person responsible can be punished for infringing copyright.

Video authentication is the method of embedding many loose and fragile watermarks to discover tampering. This method is less useful for protection in the short run and more useful for locating repeat offenders. Once a company identifies tampering, it can track the exact person who attempted to break it.

Content filtering is used to monetize stolen content essentially. Videos are embedded with a code that watermarking decoders can read to trigger a specific action, such as playing an advertisement before the video. It is a more temporary method used to earn small amounts of money on content rather than acquiring nothing from a pirated piece of media.

An online location is a severe form of watermarking, where before releasing a video and services on the Internet, it will scour the web hunting for watermarks. Though this is a robust watermarking method, it involves high-speed Internet, and a strong processor as thousands of websites have to be searched every second.

Broadcast monitoring is similar to an online location, though more targeted on live events. It also involves vital broadband and processing power, though it is more reasonable to handle as the searching algorithm does not have to run constantly. It must only be activated for the duration of the live stream.

Playback control is an older form of watermarking technology. This process involves embedding a watermark that DVD players can read. If the DVD player detects a watermark, it will block the playback of the video. This is a robust form of watermarking but outdated as Internet streaming has become the norm and physical DVDs have been completely phased out of our lives.

2 Literature review

Many different watermarking techniques have been developed with their advantages and disadvantages. They are implemented across many video standards, such as the MPEG-2, MPEG-4, H.264/AVC, H.265/HEVC, and more. Each comes with drawbacks and a multitude of techniques.

The digital watermarking embedding process must not include a perceptual degradation, and the watermark must be non-removable. The transform domain watermarking techniques embeds the watermark into the frequency coefficients of the video frames. Before embedding the watermark, the input frames are converted first to the frequency domain using one of the transformation techniques such as discrete cosine transform (DCT) [2], discrete wavelet transform (DWT), or discrete Fourier transform (DFT). The inverse of the frequency transform is performed to reform the watermark [3].

On the spatial domain, the watermark is embedded directly into the pixel data [4,5], which causes perceptual degradation of the original host.

Authors of ref. [3] proposed a video watermarking algorithm combining DCT and DWT techniques. First, the video frames are randomly selected, and then the DCT algorithm is applied to the selected video frames. After that, the first column of the selected video frames is scrambled using the Arnold algorithm. Furthermore, every column with four direct current (DC) coefficients is reshaped and transformed into four different sub-bands using the DWT technique. Next the watermark is embedded into the approximation (LL) sub-band.

Authors of refs [6,7] proposed a digital video watermarking scheme, which combines DWT and Singular Value Decomposition (SVD) in which watermarking is done in the high-frequency sub-band, and then various attacks have been applied.

Authors of refs [4,8] implemented a solution that involves Spread Spectrum Watermarking. Spread Spectrum Watermarking partially embeds a watermark in specific frames conforming to DCT values (i.e., pseudo-random) and overlays the watermark on the intended media. This MPEG-2 watermarking performs well against watermark estimation attacks as the frame rate is altered and spliced with frames just containing a watermark, containing a watermark and media, or containing media with no watermark at all. This frame rate alteration makes this method immensely powerful against any watermark decoding tool designed to scan specific frames and estimate the watermark to remove them. The drawback of this method is that it is open to geometric attacks. Though a geometric attack is less desirable for a pirate as it involves altering the geometry of the copyrighted media, it is still a commonly used method of piracy, and the copyrighted material is still stolen even though its integrity is compromised. While block-based DCT divides the input image/frame into 8 × 8 non-overlapping blocks, it includes two phases, the first one for embedding the watermark, and the second phase is to extract the watermark.

The algorithm in ref. [9] embeds the watermark in the image without focusing on the output image's quality as it randomly selected the blocks. In contrast, our proposed algorithm embeds the watermark based on the DCT coefficient values, which results in a more robust approach in digital video watermarking.

In this study, the video watermarking algorithm is performed using block-based DCT transform. Sinusoids of various magnitudes and frequencies represent each input video frame.

3 The proposed method

The proposed watermarking method works on embedding multi copies of a binary signature image in each frame in a video. The proposed algorithm is based on the Discrete Cosine algorithm, a mathematical transform used to disconnect image pixels and elements. It divides the image based on the visual quality. Also, DCT transforms the image from the spatial domain to the frequency domain, where the main difference between the images in both domains is that in the spatial domain, the pixels' values of the image do not change, which means the image stays as it is. On the other hand, we deal with the pixel change rate in the frequency domain, so this domain focuses on changing the features in the image. DCT enables the frames to be divided into their frequency bands, as shown in Figure 1.

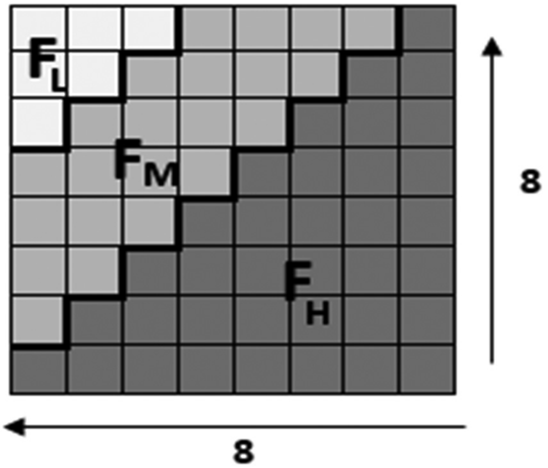


Figure 1: Low-medium-high frequency band in a DCT 8 × 8 block.

To calculate the DCT for a video frame, let us suppose that DCT is denoted by $F(s,t)$ and a video frame is denoted by $F(m,n)$.

$$F(s,t) = c(s)c(t)\,\frac{2}{\sqrt{MN}}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}F(m,n) \times \cos\left[\frac{\pi}{2M}(2m+1)s\right]\cos\left[\frac{\pi}{2N}(2n+1)t\right]. \tag{1}$$

where M is the width of the frame in pixels, N is the height of the frame in pixels, and (s,t) are the DCT coefficients. The values of $c(t)$ and $c(s)$ can be calculated as:

$$c(s) = \begin{cases} 1, & s = 1, 2 \ldots, M-1, \\ 1/\sqrt{2}, & s = 0, \end{cases} \qquad c(t) = \begin{cases} 1, & s = 1, 2 \ldots, M-1, \\ 1/\sqrt{2}, & s = 0. \end{cases} \tag{2}$$

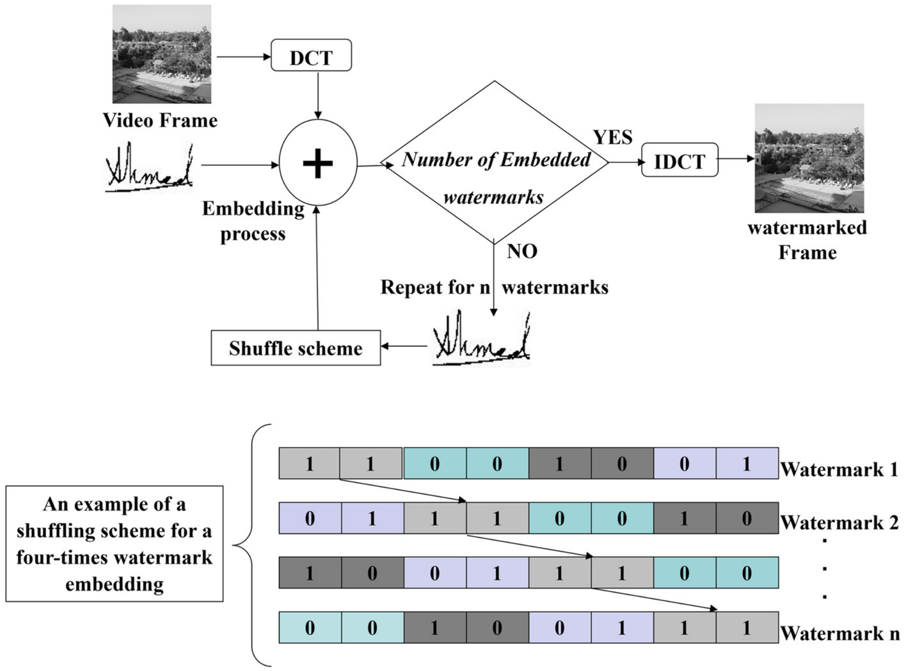
While the inverse DCT which is used in the extracting process is denoted by $F(m,n)$,

$$F(m,n) = \frac{2}{\sqrt{MN}}\sum_{s=0}^{M-1}\sum_{t=0}^{N-1}c(s)c(t)\,F(s,t) \times \text{coscos}\left[\frac{\pi}{2M}(2m+1)s\right]\cos\left[\frac{\pi}{2N}(2n+1)t\right]. \tag{3}$$

Before starting the embedding process, a shuffling is made for each frame's watermark copies. The shuffling process is performed using a secret key generated by the computer. Thus, the shuffling scheme differs in each iteration, which arranges different 8 × 8 sub-block of the image. According to ref. [10], the watermark is embedded into eight different frequency bands. Embedding the watermark eight times in each frame aids in protecting several attacks such as cropping, filtering, scaling, etc. The shuffling process will preserve the watermark from cropping attacks and provide a full extracted watermark. A graphical illustration of the shuffling process is shown in Figure 2.

3.1 RGB channels analysis

Some popular measurement metrics have been applied between the grayscale version of the colored image and each R, G, B color component individually to justify selecting the green channel for watermark embedding over the red and blue channels [11]. A library of approximately 35 color images was depicted using a digital camera, plus some standard photos and their associated gray-level versions were used. Each image is 24 bits/pixel RGB of size 512 × 512. The analytical measurements have


DCT
Video Frame
+
Embedding process
Number of Embedded watermarks
YES
NO
IDCT
watermarked Frame
Repeat for n watermarks
Shuffle scheme
An example of a shuffling scheme for a four-times watermark embedding

1	1	0	0	1	0	0	1	Watermark 1
0	1	1	1	0	0	1	0	Watermark 2
1	0	0	1	1	1	0	0	. . .
0	0	1	0	0	1	1	1	Watermark n

Figure 2: Shuffling process for *n* watermarks.

been carried out using the mean square error (MSE) and peak signal-to-noise ratio PSNR. PSNR penalizes noise visibility in an image, while MSE compares the original and modified versions' corresponding pixels. Thus, two precisely same images will produce an infinite PSNR value and a zero MSE value. In addition to the MSE and PSNR, the correlation between the DC values representing the red, green, and blue components and the gray-level version of the original image was carried out for various images. Different analytical measurements can be obtained by computing the following equations:

$$\mathrm{MSE} = \frac{1}{XY}\sum_{x,y}(p_{x,y} - \tilde{p}_{x,y})^2, \tag{4}$$

$$\mathrm{DC\_Value} = \frac{1}{\sqrt{XY}}\sum_{x,y}P_{x,y}, \tag{5}$$

$$\mathrm{DC\_Error} = \frac{1}{\sqrt{XY}}\left[\sum_{x,y}P_{x,y} - P'_{x,y}\right], \tag{6}$$

$$\mathrm{PSNR} = XY \max_{x,y} p_2^{x,y} \Big/ \sum_{x,y}(p_{x,y} - \tilde{p}_{x,y})^2. \tag{7}$$

where $P_{x,y}$ is the grayscale image, and $P'_{x,y}$ is the individual R, G, B components, respectively. For results shown in Figures 3–6, the red, green, and blue colors represent R, G, and B components, respectively, while the gray color in Figure 4 illustrates the grayscale results. It is evident that the green channel of the RGB color image produces the closest statistical performance to that of the gray equivalent of the color image. Embedding the watermark in the Y channel of the YIQ format of a color image produces excellent invisible watermarking results. It is logical to choose the G channel of the RGB format of the same color image to embed the watermark. Analytical measurements and experimental results in Figures 3–6 prove that embedding the watermark in the G channel of a color image could produce better invisibility properties.

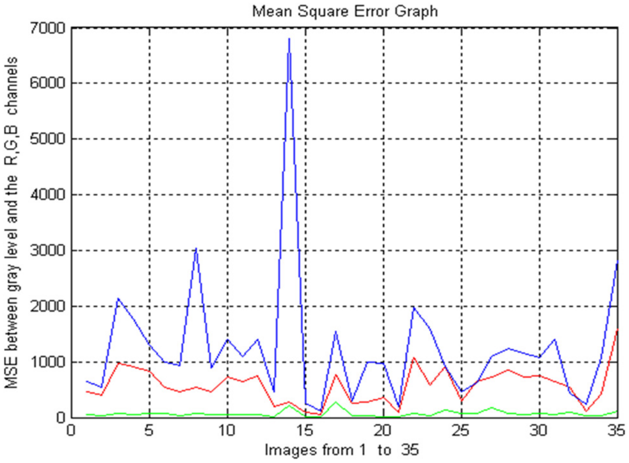


Figure 3: MSE measurements between the R, G, B components and the grayscale version for 35 images.

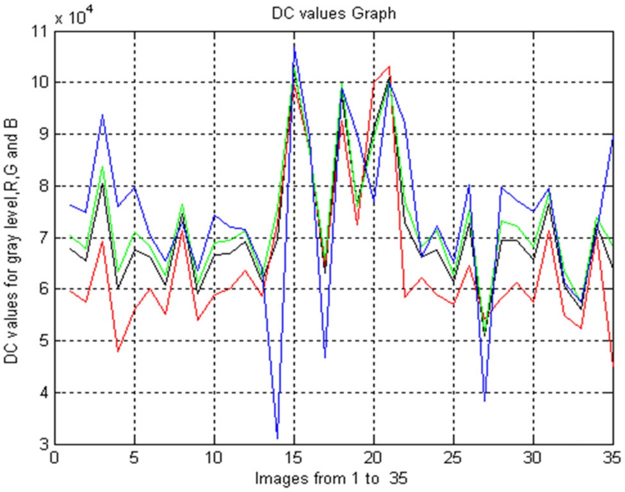


Figure 4: DC values correlation between R, G, B, and grayscale version.

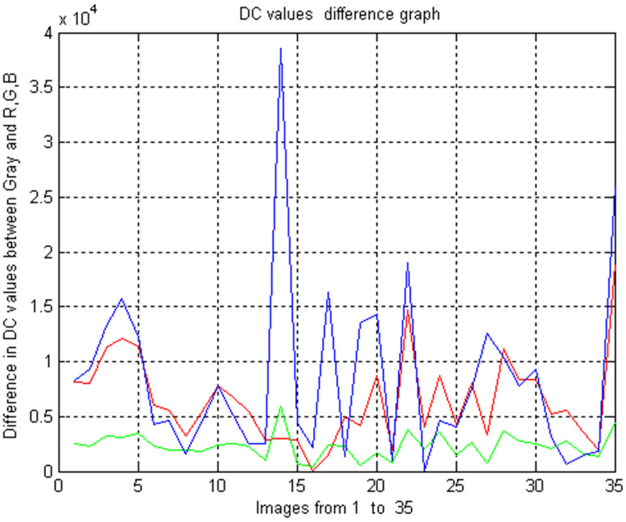


Figure 5: Difference in DC values between grayscale images and each of R, G, B components.

3.2 Watermark embedding algorithm

The embedding process of a watermark image of size 96 × 64 in a video can be described in several steps, as shown in Figure 7.

Step 1: The video is divided into frames.

$$\text{Number of frames} = \frac{\text{Frames}}{\text{Seconds}} \times \text{Duration of the video.} \tag{8}$$

Step 2: Each frame is resized to 512 pixels width and 512 pixels height.

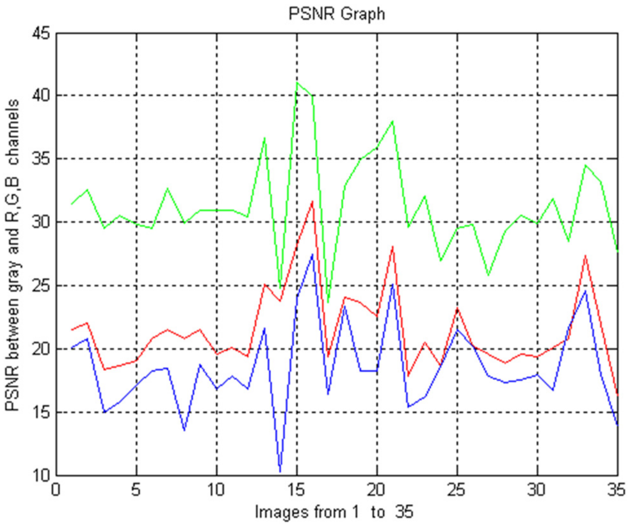


Figure 6: PSNR measurements between R, G, B components and grayscale version for 35 color images.

Step 3: Each frame is passed through DCT transformation and divided into 8 × 8 blocks denoted by $u(k)$, where each video frame is divided into N_{HB} $1 \le k \le N_{\mathrm{HB}}$ non-overlapped blocks of size 8 × 8.

$$F_k(u,v) = \mathrm{DCT}\{\,f_k(i,j)\}, \tag{9}$$
$$1 \le u, v \le 8, 1 \le k \le N_{\mathrm{HB}}.$$

For each 8 × 8 sub-block, DCT coefficients are defined and then the highest block coefficient value is selected to embed the watermark.

Step 4: The binary watermark image is converted into a vector of size 1 × N_{W} to be shuffled using a generated secret key. The secret key differs in each embedded watermark. The secret key is needed to re-order the shuffled watermark through the extraction process.

Step 5: The watermark is divided into N_{WB} 1 × 8 sub-blocks. Each sub-block of the watermark will be embedded into each frame of 8 × 8 sub-blocks. Let us assume that $w(i,\ j)$ is the binary watermark image of size N_{W} bits which are smaller than the frame size and $f(i,\ j)$ is the gray-level frame of size N_{H} pixels.

According to ref. [10], the bit embedding equation is defined as follows:

If $w\ (i,\ j) = 1$, then equation (10) is:

$$F_k(u,v) = \begin{cases} \Delta Q_e\left(\dfrac{F_k(u,v)}{\Delta}\right) u, v \in U_k, & 1 \le k \le N_{\mathrm{WB}}, \\ F_k(u,v)\ u, v \notin U_k, & 1 \le k \le N_{\mathrm{WB}}. \end{cases}$$

If $w(i,\ j) = 0$, then

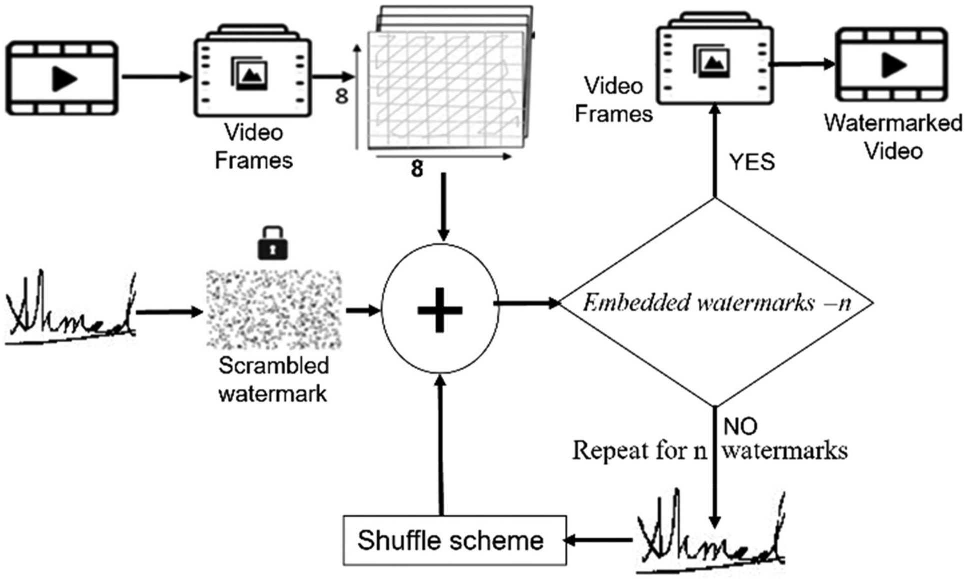


Figure 7: Embedding process for *n* watermarks.

$$F_k(u,v)=\begin{cases}\Delta Q_{\text{o}}\left(\dfrac{F_k(u,v)}{\Delta}\right)u,v\in U_k, & 1\le k\le N_{\text{WB}},\\ F_k(u,v)\,u,v\notin U_k, & 1\le k\le N_{\text{WB}},\end{cases}\tag{10}$$

where Δ is the scaling quantity, Q_{e} is the quantization to the nearest even number, and Q_{o} is the quantization to the nearest odd number.

Step 6: The embedding equation in step 4 is repeated N times to embed the n number of watermarks in the frame. According to ref. [12], before repeating step 4, a shuffle scheme is applied for each watermark copy before repeating the embedding process.

Step 7: The constructed watermarked frame is retrieved using the inverse DCT transform for all $F_k(u,v)$, $1\le k\le N_{\text{HB}}$

Step 8: The steps from 2 to 6 are repeated for the whole number of frames in the video.

3.3 Watermark extraction algorithm

The watermark can be extracted from the video's frames as follows:

Step 1: Divide the video into its whole frames.

Step 2: Perform DCT transform to divide each watermarked frame into 8 × 8 blocks.

Step 3: Indicate the block of the watermarked frame that contains the first copy of the watermark image.

Step 4: Indicate the same 8 × 8 sub-block coefficients with watermark bits.

Step 5: The extraction equations of the bits are:

$$\begin{aligned}&\text{If } Q\left(\frac{F_k\,(u,v)}{\Delta}\right)\text{odd},\quad w(i,j)=0,\\ &\text{If } Q\left(\frac{F_k\,(u,v)}{\Delta}\right)\text{even},\quad w(i,j)\;=\;1.\end{aligned}\tag{11}$$

The Q value is rounded to the nearest integer.

Step 6: Steps 4 and 5 are repeated for each sub-block to extract the other watermark copies in the frame.

Step 7: A reverse of the shuffling process is performed for each copy of the reconstructed watermark. Each watermark is reshuffled using the same previously generated secret key for each copy of the watermark used in the embedding process.

Step 8: Finally, an averaging is made for all the extracted bits by adding the output watermark copies, as illustrated in Figure 8. Then, the resultant watermark, which provides the minimum MSE, high NC, and High CC, is selected.

4 Simulations and results

The proposed algorithm is tested using different videos with different properties, as shown in Table 1. The used signature image is a grayscale image of 832 bytes and 96 × 64 pixels. The used watermarking technique is tested and evaluated with various embedding strengths Δ, as shown in Table 2.

Several ways of evaluating the quality of the watermarked frames such as MSE, PSNR, and structural similarity

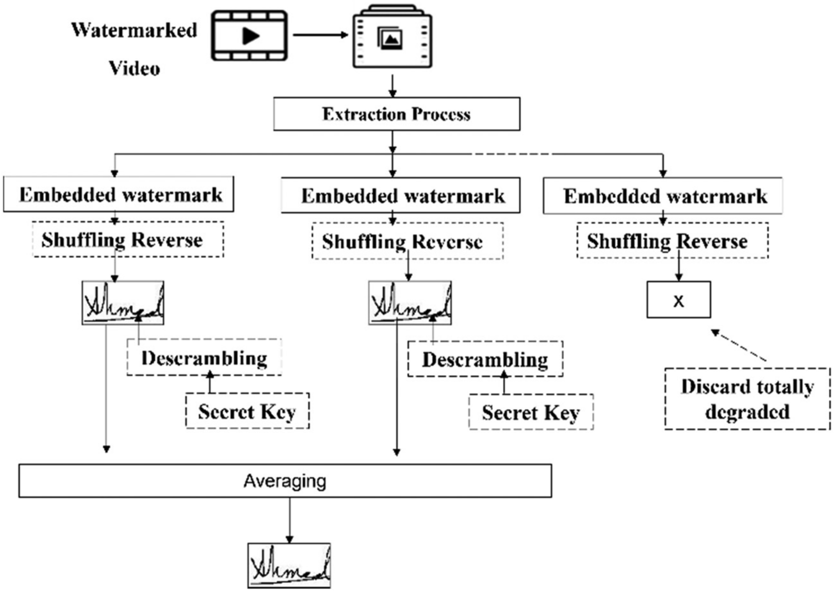


Figure 8: Extraction process for *n* watermarks.

Table 1: Properties of the host videos

	Size (MB)	Frame width (pixels)	Frame height (pixels)	Frame rate (FPS)	Data rate (kbps)	Total number of frames (Frames)
Video 1	25	320	240	15.00	27,648	113
Video 2	496	1,280	720	29.97	1,201	100
Video 3	256	640	640	25.00	250	250

Table 2: Embedding results at different watermarking strengths

Strength values	Watermarked frame	PSNR	SSIM	MSE
$\Delta = 8$		68.5207	0.9942	0.0091
$\Delta = 16$		62.9974	0.9838	0.0326
$\Delta = 24$		60.0851	0.9700	0.0638
$\Delta = 34$		56.6772	0.9505	0.1398

index measurement (SSIM) and extracted watermarks such as normalized correlation (NC) were used. MSE is calculated by averaging the squared intensity differences of the distorted and reference image pixels. PSNR is also used to evaluate the original video frames and the watermarked video after embedding the signature in each frame in the video. MSE and PSNR can be calculated using equations (12 and 13) [2,13]. NC in equation 14 is used to evaluate the quality of the extracted watermark. It measures the similarity between the original and extracted watermarks. The value of NC is between 0 and 1; the higher the NC value, the more is the similarity between the original and extracted watermarks. If NC = 1, the extracted watermark is the same as the original. Based on our experimental results, the NC value is greater than 0.9, which is accepted as good watermark extraction.

$$\mathrm{MSE} = \frac{1}{m{\star}n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2, \tag{12}$$

$$\mathrm{PSNR} = 10 \star \log_{10}\left(\frac{255^2}{\mathrm{MSE}}\right), \tag{13}$$

$$\mathrm{NC} = \frac{\sum_{i=0}^{N=1} W(i)x\ W'(i)}{\sqrt{\sum_{i=0}^{N-1}(W(i))^2}}. \qquad (14)$$

Another evaluation method is implemented using the SSIM between the original and watermarked videos. The higher the SSIM value, the higher the percentage similarity between the original and watermarked frames.

The proposed algorithm works on embedding the watermark that is invisible to humans and robust at the same time. Several values of embedding strengths are used for the evaluation to demonstrate the functional performance for most of the videos. The implementation proved that any increase in embedding strength value would achieve stronger robustness. Still, watermarked video frames' quality will be less than the original ones.

As shown in Figures 9–11, the PSNR values are higher for lower *Δ* values, for example, when *Δ* = 8, PSNR values are greater than 65, while on the other hand, the PSNR values are less than 60 for *Δ* = 34.

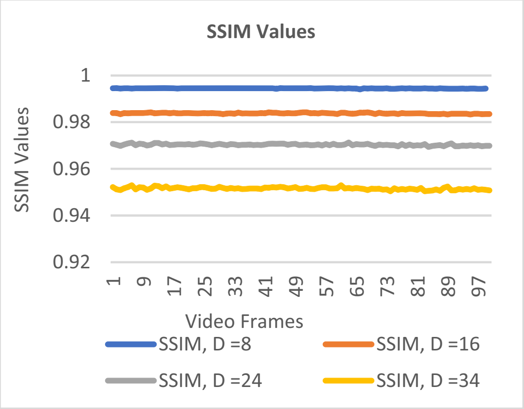


Figure 9: SSIM values of watermarked frames (*D* represents the strength value *Δ*).

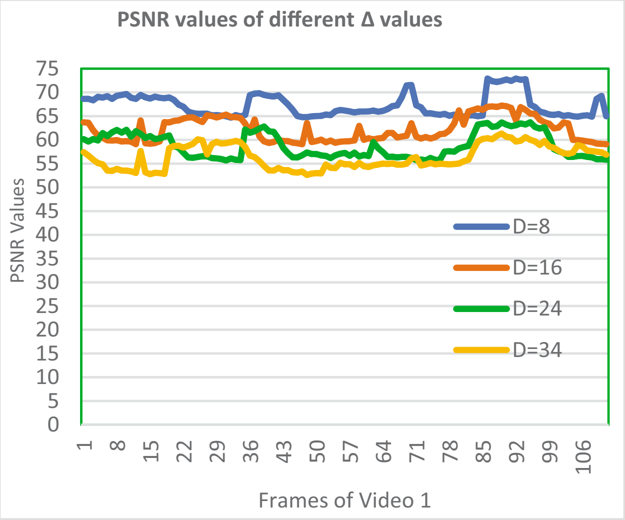


Figure 10: PSNR values for watermarked video 1.

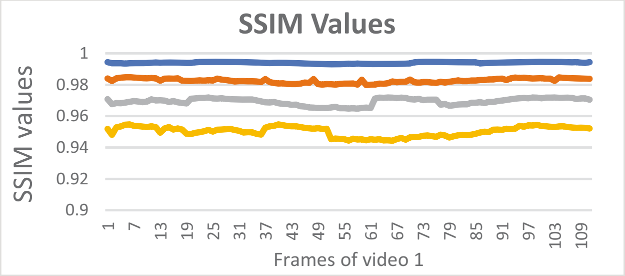


Figure 11: SSIM values of video 1.

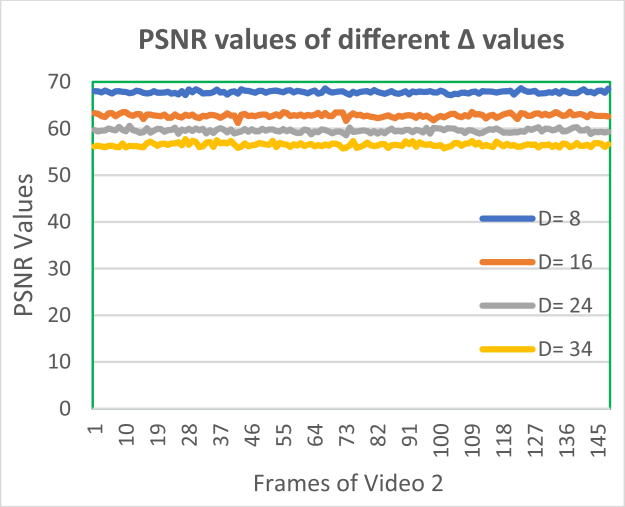


Figure 12: PSNR values for watermarked video 2.

The SSIM values differ according to strength values *Δ*, and the SSIM values are also inversely proportional with

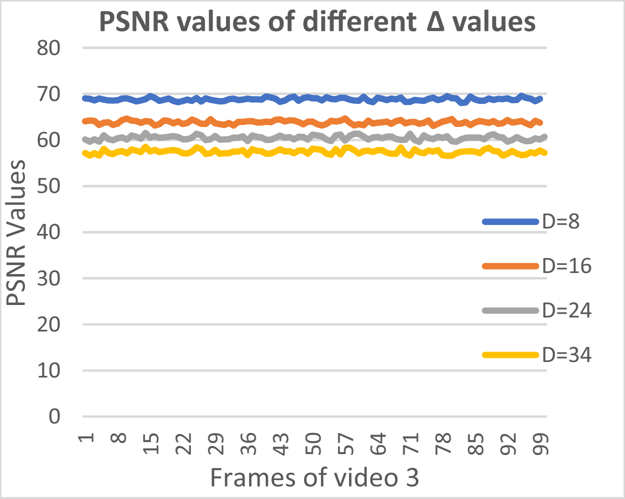


Figure 13: PSNR values for watermarked video 3.

the strength values. Thus, Figures 9 and 11 show the SSIM values of watermarked frames for different strength values and video, respectively.

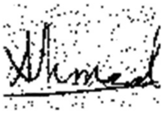
The SSIM values for all the frames are almost constant for all watermarked frames of the same strength value Δ. As the SSIM values indicate the quality of the watermarked frame, this means that the quality of the watermarked video is higher for lower strength values and greater SSIM values.

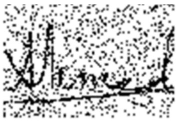
For the first video with 113 frames and a 27,648 kbps data rate, PSNR values fluctuate for each Δ value. High data rate might be the reason for the instability in the output values of SSIM and PSNR in video 1, as shown in Figure 10.

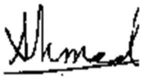
PSNR values for the second and third videos are almost stable through the number of frames. As illustrated in Figures 12 and 13, PSNR values are inversely proportional to strength values Δ.

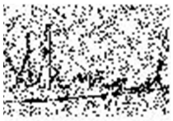
Table 3: Noise attacks at strength value Δ = 34

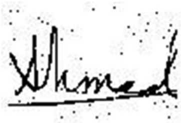
Noise attacks at Δ = 34			
Attacks	**Extracted watermarks**	**Attacked watermarked frames**	**Quality of the attacked frames**

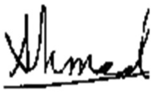
Salt and pepper noise, d = 0.01	
	NC = 0.9578
			PSNR = 41.4713
			SSIM = 0.7053

Salt and pepper noise, d = 0.02	
	NC = 0.8524
			PSNR = 37.8933
			SSIM = 0.5290

Gaussian noise, m = 0, V = 0.001	
	NC = 1
			PSNR = 45.4581
			SSIM = 0.5899

Gaussian noise, m = 0, V = 0.005	
	NC = 0.8294
			PSNR = 39.4007
			SSIM = 0.2905

Speckle noise, V = 0.005	
	NC = 0.9901
			PSNR = 42.4594
			SSIM = 0.5308

Speckle noise, V = 0.001	
	NC = 1
			PSNR = 52.8337
			SSIM = 0.9519

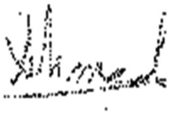
To test and verify the robustness of the proposed algorithm, several attacks were performed through the evaluation process with different strength values 8, 16, 24, and 34. The experimental results demonstrated that the performance achieved by the watermarking algorithm using DCT for the extracted watermark after testing several types of attacks is perceptually visible and robust at high strength values such as 24 and 34, which is illustrated in Table 3. Higher strength values provide strong robustness for the extracted watermark image. Still, at the same time, it reduces the quality of the watermarked video through reducing PSNR and SSIM values of the attacked frames and *vice versa*.

Table 4: Cropping attacks at strength value Δ = 34 and Δ = 8

Attacks	Extracted watermarks	Quality of the attacked frames
Cropping attacks, Δ = 34		

Cropping horizontal 50%	
	NC = 1
		PSNR = 52.8879
		SSIM = 0.4696

Cropping horizontal 25%	
	NC= 1
		PSNR = 52.7624
		SSIM = 0.7104

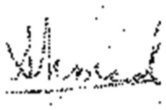
Cropping vertical 50%	
	NC = 0.9991
		PSNR = 52.6139
		SSIM = 0.4669

Cropping vertical 25%	
	NC = 1
		PSNR = 52.6367
		SSIM = 0.7094

Cropping attacks, Δ = 8		

Cropping horizontal 50%	
	NC = 1
		PSNR = 65.2733
		SSIM = 0.4908

Cropping horizontal 25%	
	NC = 1
		PSNR = 65.0210
		SSIM = 0.7425

Cropping vertical 50%	
	NC = 0.9987
		PSNR = 65.1315
		SSIM = 0.4885

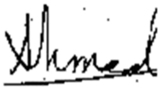
Cropping vertical 25%	
	NC = 0.9989
		PSNR = 65.2399
		SSIM = 0.7416

Table 5: Filtering attacks at strength value Δ = 34

Filtering attacks at Δ = 34		
Attacks	**Extracted watermark**	**Quality of the attacked frames**

Gaussian low pass filter	
	NC = 1
		PSNR = 52.8991
		SSIM = 0.9517

Unsharp high pass filter	
	NC = 1
		PSNR = 52.8731
		SSIM = 0.9523

Winner2 [3 × 3]	
	NC = 1
		PSNR = 42.3792
		SSIM = 0.9575

Median [3 × 3]	
	NC = 1
		PSNR = 45.1532
		SSIM = 0.9599

Sobel edge detection	
	NC = 1
		PSNR = 52.6813
		SSIM = 0.9518

Laplacian, alpha = 0.1	
	NC = 1
		PSNR = 52.9163
		SSIM = 0.9512

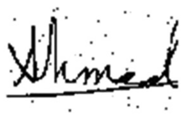
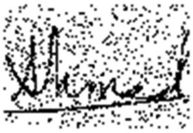
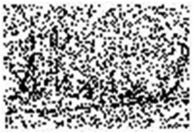
Color enhancement	
	NC= 0.9947
		PSNR = 36.7286
		SSIM = 0.9011

Table 6: Noise attacks at strength value *Δ* = 8

Noise attacks at *Δ* = 8		

Salt and pepper noise, *d* = 0.01
PSNR = 40.9417
SSIM = 0.7313
	
	NC = 0.9128	

Salt and pepper noise, *d* = 0.02
PSNR = 37.7662
SSIM = 0.5557
	
	NC = 0.7345	

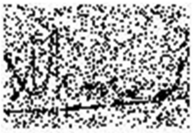
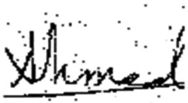
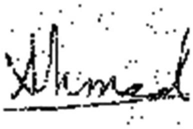
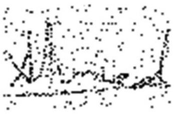
Speckle noise, *V* = 0.001
PSNR = 49.4215
SSIM = 0.7886
	
	NC = 0.7869	

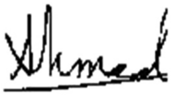
Table 7: Filtering attacks at strength value *Δ* = 8

Filtering attacks at *Δ* = 8		

Gaussian low pass filter	
	NC = 0.9923
		PSNR = 44.7038
		SSIM = 0.9868

Winner2 [3 × 3]	
	NC = 0.9929
		PSNR = 54.7806
		SSIM = 0.9859

Median [3 × 3]	
	NC = 0.9606
		PSNR = 45.6090
		SSIM = 0.9851

Laplacian, alpha = 0.1	
	NC = 1
		PSNR = 65.1799
		SSIM = 0.9944

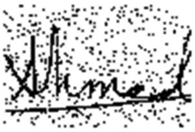
Color enhancement	
	NC= 0.9223
		PSNR = 36.7373
		SSIM = 0.9225

Table 3 illustrates the robustness of the algorithm by implementing different noise attacks. Comparing PSNR and SSIM values in Figures 7 and 8 with the values in Table 3, it is noticeable that PSNR and SSIM values are reduced.

Additionally, PSNR and SSIM values are affected by several factors such as the type of noise and the noise variable for each type, thus, increasing the noise variable or decreasing it affects the values of PSNR and SSIM.

Table 4 demonstrates the performance of the proposed algorithm against horizontal and vertical cropping attacks at strength values 8 and 34.

Overall, the quality and robustness of the extracted watermark are high, except for the vertical cropping 50%.

Tables 6 and 7 demonstrate the algorithm's performance at strength value = 8. Although in Table 2, the lower strength values such as 8 and 16 give the highest PSNR and SSIM values, resulting in high-quality watermarked frames, after running the attacks at *Δ* = 8, PSNR and SSIM values become lower, and the noise in some attacks distorts the watermark image.

The value of NC illustrates the similarity between the extracted and the original watermark. Based on the experimental results in Tables 3–7, the similarity between the original and extracted watermark increases when the NC value is equal or near to 1.

5 Recommendations for future work

Using the lower frequency components with the highest magnitudes is recommended to maintain the quality and robustness after embedding the watermark. In the proposed method, the used DCT blocks consisted of 8 × 8 coefficients. Thus, those frequencies can be screened to find the coefficient with the highest magnitude and register its location. The process will be repeated for all DCT blocks. The locations which are repeated more are recorded and used for embedding. The location will vary from one image to another according to the spatial frequency contents of the image. The selection process will help to increase the robustness.

A digital watermark can be implemented using various techniques and algorithms that hide information for multiple documents or digital media files, such as watermarking audio or PDF files using text or numbers.

In the future, different algorithms can be used to water digital videos such as Wavelet transform or other transform types. Moreover, watermarking digital videos can be performed using a mobile phone number which provides high protection and strong authentication.

6 Conclusion

This article proposes a new adaptive algorithm for digital video watermarking. The suggested algorithm uses the green channel for the embedding process. We conclude that the algorithm provides a watermarked video with high quality and strong robustness. Furthermore, the watermarked frames' robustness and quality can be adjusted and increased for high strength values.

Several strength values were used for the testing and evaluation processes. The result through the extraction process of the watermark was invisible for all strength values. Still, it is more apparent for lower strength values according to the described results in the study.

Through the evaluation process of the proposed algorithm against attacks, higher strength values are recommended. However, to protect the watermark against attacks and get high-quality watermarked frames simultaneously, Δ values between 20 and 24 are preferred to give the required quality and robustness.

Conflict of interest: The authors state no conflict of interest.

References

[1] Y. Li, H. Wang, and M. Barni. "A survey of deep neural network watermarking techniques," *Neurocomputing*, vol. 461, pp. 171–193, 2021.

[2] M. M. Salih, E. F. Ahmed, and M. A. Al-Abaji. "Digital video watermarking methods using DCT and SVD," *J. Educ. Sci.*, vol. 29, no. 1, pp. 266–278, 2020. (ISSN 1812-125X).

[3] Q. Liu, C-L. Song, and J. Sang. "Robust video watermarking using a hybrid DCT-DWT approach,"*J. Electron. Sci. Technol.*, vol. 18, p. 100052, 2020.

[4] S. P. Mohanty, P. Guturu, E. Kougianos, and N. Pati. "A novel invisible color image watermarking scheme using image adaptive watermark creation and robust insertion-extraction," *In Proceedings of the 8th IEEE International Symposium on Multimedia* (*ISM*), pp. 153–160, 2006

[5] G. Chareyron and A. Trémeau. *Watermarking of Color Images Based on a Multi-layer Process*, Université Jean Monnet, France, 2020.

[6] D. K. Thind and S. Jindal. "A semi blind DWT-SVD video watermarking," *in International Conference on Information and Communication Technologies, India*, 2015.

[7] Q. Liu, S. Yang, J. Liu, P. Xiong, and M. Zhou. "A discrete wavelet transform and singular value decomposition-based digital video watermark method," *Appl. Math. Model.*, vol. 85, pp. 273–293, 2020.

[8] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *Image processing, IEEE Trans. on*, vol. 6, no. 12. pp. 1673–87, 1997.

[9] C. Way, A. Gortan, W. G. Junior, and H. Fung. "A review study on image digital watermarking," *in ICN 2011: The Tenth International Conference on Networks*, 2011.

[10] A. N. Al-Gindy, H. A. Ahmad, A. Tawfik, and R. A. Qahwaji. "A new blind image watermarking of handwritten signatures," *IEEE International Conference on Signal Processing and Communications*, vol. 8, pp. 17–24, 2015.

[11] A. Al-Gindy, H. Al-Ahmad, R. Qahwaji, and A. Tawfik. "A novel blind image watermarking technique for colour RGB images in," *Mosharaka International Conference on Communications, Computers, and Applications*, 2008.

[12] A. Al-Gindy, H. Al-Ahmad, R. Qahwaji, and A. Tawfik, "Enhanced DCT based technique with shuffle," *In proceeding of ICCCP'07 International conference for communication, Computer, and power, Muscat, Oman*, 2007.

[13] M. Hashim, M. S. Mohd Rahim, and A. A. Alwan. "A review and open issues of multifarious image steganography techniques in spatial domain," *J. Theor. Appl. Inf. Technol.*, vol 4, pp. 956–977. 2015.

[14] G. Eason, B. Noble, and I. N. Sneddon., "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," *Phil. Trans. Roy. Soc*, vol. 247, no. 935, pp. 1934–1990, 1995.

[15] P. Porwal, T. Ghag, N. Poddar, and A. Tawde. "Digital video watermarking using modified LSB and DCT technique," *Int. J. Res. Eng. Technol.*, vol. 3, no. 4. pp. 630–634, 2014.