Multimedia Systems (2023) 29:361–375
https://doi.org/10.1007/s00530-022-01006-5

REGULAR PAPER

Fake COVID-19 videos detector based on frames and audio watermarking

Nesrine Tarhouni[1] · Salma Masmoudi[1] · Maha Charfeddine[1] · Chokri Ben Amar[1]

Received: 15 March 2022 / Accepted: 12 September 2022 / Published online: 27 September 2022

Abstract
With the innovation and development of advanced video editing technology and the widespread use of video information and services in our society, it is increasingly necessary to maintain the reliability of video information. As a result, sensitive video contents in various fields such as surveillance, medical, and others should be secured against attempts to alter them because malicious modifications could impact decisions based on these videos. In this paper, we present a fake video detector based on combining audio and frames watermarking. Furthermore experimental results demonstrate good robustness and integrity verification results for COVID-19 video news.

Keywords Audio watermarking · Image watermarking · Integrity verification · Fake video

1 Introduction

Social media is now one of the top platforms for distributing and consuming news. However, it faces the main problem, which is the fast and wide-spreading of false information. These social networks have a massive worldwide user base as they are very easy to access, which makes monitoring the content posted on them a very difficult thing to achieve. As a result, there is a lot of doubt about the credibility of news and information shared on these networks, and though measures are being taken to combat this problem, it remains on the rise. The majority of research in this area focuses on the use of machine learning to detect fake news, whereas very limited research has attempted to approach the issue using digital watermarking. This technology [1] presents an important research branch of multimedia data hiding embed some additional information as a watermark in the host audio and then extracts it when necessary. This watermark data can meet the requirements of certain applications such as authentication [8], copyright protection [2, 3], indexation, watermark tracing [4, 5] etc.

The motivation behind this paper is to fight the spread of fake news videos and false information using combined video and audio watermarking. Using combined watermarking, we can provide a way for individuals or companies to verify the integrity of their news video in a short time independently of the modification affecting the audio channels or images. In addition, in this paper, we prove the efficiency of the proposed method to detect fake COVID-19 news videos.

The innovation of the proposed method resides in using video watermarking for fake verification and copyright protection. The use of combined watermarking (frames and audio watermarking) allows for detecting manipulation in both channels. Moreover, the approach is semi-fragile. So, it allows protecting and detecting content changes simultaneously. In addition, watermarking contrarily the machine learning based-process provides a less time-consuming and rapid response.

Section 2 in this paper is reserved to present a short literature survey. In Sect. 3, the proposed combined watermarking is introduced. Test results are given in Sect. 4. The conclusion in Sect. 5 summarizes the results and the perspectives of this paper.

✉ Nesrine Tarhouni
nesrine.tarhouni@enis.tn

Salma Masmoudi
salma.masmoudi@gmail.com

Maha Charfeddine
maha.charfeddine@enis.tn

Chokri Ben Amar
chokri.benamar@ieee.org

[1] National Engineering School of Sfax, Ecole Nationale d'Ingenieurs de Sfax, Sfax, Tunisia

2 Related works

In this part, we aim to review several important existing watermarking schemes for video authenticity using semi-fragile methods. In [14], the authors proposed a semi-fragile video watermarking technique that can identify both frame attack and video tampering. The frame number serves as the watermark information in this paper, and the authentication code is based on the correlation between the nonzero discrete cosine transform (DCT) coefficients. The watermark is embedded in the 4×4 subblocks with sufficiently complex DCT nonzero coefficients. The ratio of these non-zero coefficients at the middle frequency was changed for watermarking. Empirical findings reveal that the embedded watermarked video's visual quality is almost untouched, and the method is stable. The approach can also correctly detect frame attacks and video tampering.

In [15], they presented a semi fragile watermarking scheme for video content authentication based on Singular Value Decomposition (SVD) and Discrete Wavelet Transform (DWT). Extracting strong characteristics from areas of interest is the beginning of creating an authentication signature based on content. The Arnold transform, as well as the QR code generating process, are utilised to increase the watermark's security. Using an additive embedding methodology, the latter is efficiently concealed in mid-frequency subband of the wavelet and retrieved using a blind extraction algorithm. The results reveal that the suggested technique has acceptable perceptual quality as well as a large watermark capacity. It also has the ability to discern between malicious assaults and unintended changes. Indeed, the suggested watermarking approach is particularly sensitive to malicious alteration while enabling legitimate processing.

In [16], they employed a watermarking technique, which involves hiding copyright information in the original video as a watermark. Using a video as a watermark allows users to hide a lot of information. This paper's watermarking approach is semi-fragile, which indicates that tampering with the video may be discovered reasonably simply. They implanted the watermark in the frequency domain, enhancing the watermark's robustness using DWT, DCT, and SVD. The singular values of the watermark with few embedding strength are joined to the singular values of the original video to generate a watermarked video, and the original and watermarked videos are progressively modified using the DWT and DCT. Using the calculated PSNR values, the impact of numerous attacks on the watermarked video were examined.

In [17], this article proposes a semi-fragile video watermarking technique based on chromatic DCT that can conserve copyright and detect tampering. First, experiments show that chrominance blocks have more stable prediction modes than luminance blocks, which can efficiently decrease the possibility of desynchronization due to prediction mode changes. In this case, more non-zero (NNZ) coefficients the block has, the lower the estimation mode changes. Consequently, to each macroblock, the algorithm sorts the four sub-macroblocks based on the NNZ chroma DCT coefficients and then selects the subblock with the highest residual NNZ coefficient. Using the secret key K and the medium frequency stability of Intra 4×4 coefficients, modify the relationship of three DCT coefficients near middle frequency. Second, the prediction mode's sensitivity to malicious attacks and recompression operations differs. As a consequence, the algorithm classifies the macroblock's prediction model and generates an authorization code based on the estimation model. Watermark embedding has little influence on video quality, and the change in video bitrate is basically constant, according to the findings of the experiments. Using the authentication code of the prediction model, the chromatic DCT method can identify and find tampering at the 4×4 sub-block level. In [18] a dual watermarking for video authentication based on moving objects is presented in this paper. The frame index is initially inserted as a watermark into the moving objects of the relevant frame using a reversible watermarking method, with the goal of detecting temporal tampering. The main content and details of the moving objects are then incorporated into the frame, together with the authentication code as the other watermark, allowing spatial tampering detection and recovery. In this article [19], the authors suggested a real-time video watermarking scheme for MPEG, in which the original video sequence is first exploited for fast scene segmentation, and then appropriate scenes are adaptively selected to be embedded. A visual model is also used to control the strength of the watermark. Watermarks are embedded by changing the level of run-level pairs to change the number of bits in the bitstreams. The results of the experiments reveal that there is little loss of video quality and that the system is extremely resistant to a variety of attacks. A semi-fragile authentication technique for high-efficiency video coding is proposed in this study [20], 4×4 intra luma transform blocks of I-frames are separated into two distinct subsets in this scheme. One subset is used to generate authentication codes, while another is used to embed the generated code. Blocks are chosen from these subsets based on quality and robustness thresholds. Using a low-complexity spatial analysis, these thresholds are determined at runtime. The number of positive and negative quantized discrete sine transformed coefficients in a block is used to construct the authentication code. During the encoding of the video sequence, the created authentication code is

embedded by changing the magnitudes of quantized discrete sine transformed coefficients. Experiments reveal that the system is resistant to dropping, re-compression, frame, and noise attacks, but it is vulnerable to malicious attacks. This research [21] proposed a spatial domain fragile watermarking technique for ensuring the integrity of video digital content. The watermark is a binary image that has been reproduced to be the same size as the video frame size. Before insertion, the watermark is encoded by XOR-ing it with a random image to enhance security. Arnold Cat Map is used to generate the random image. By altering the pixel values of video frames, the encrypted watermark is inserted. The watermarked video has been exposed to certain attacks. The technique can recognize modified regions of video frames, according to the findings of the experiments. A new method for video authentication is proposed in this paper [22]. The approach works by creating watermark images that act as a secondary carrier for the binary sequence. Each video frame's coefficients of the Discrete Wavelet Transform contain a unique watermark image. The analysis of video frames enables the detection of spatial attacks, and the sequence carried by the extracted images allows for the determination of the sort of temporal attack of tampered frames. The method is suitable for addressing authentication tasks, according to the findings of experiments.

Finally, in [23], a new semi-fragile watermarking method for MPEG4 AVC protection, is introduced in this article. The Intra prediction mode types provide the authentication information that allows the method to be fragile. This signature is embedded in an m-QIM technique's quantized error prediction of the DCT coefficients, ensuring the method's robustness. SPYART was tested within the context of a video surveillance application; the findings demonstrate fragility to content substitution (spatial and temporal accuracy of 1/81 frames and 3 seconds respectively) and resistance to transcoding (4x MPEG4 AVC compression).

In this paper, a blind and semi-fragile video watermarking scheme for tamper detection is proposed. The originality of this system resides in watermarking both frames and audio channels for dual applications: copyright protection and fake verification. The major innovation related to this novel approach covers the following main points:

1. Blind watermarking of frames and audio channels for fake detection in COVID-19 videos.
2. Using two fragile-content watermarks which are robust against not malicious attacks.
3. Preserving robustness to desynchronize geometric attacks such as cropping and rotation due to using SURF-based features only in the detection step.
4. Assuring high inaudibility and robustness of the audio watermarking due to a preliminary study on the adequate hiding regions.
5. Verifying integrity in both frames and audio of the video.
6. Watermark detecting directly in the video compressed domain without preprocessing stage.
7. Providing less computational complexity for real-time applications.
8. Using relevant features characterizing both audio channel (side information features of MP3 file) and frame channel (texture, and color data) to generate watermarks. These features allow the detection of content manipulation attacks.

3 The proposed methodology

In this paper part, we detail our proposed video watermarking. Figure 1, Algorithms 1 and 2 well define the proposed method. The suggested approach includes the embedding process in frames and audio channel, Stirmark attacks application, and watermark extraction. Figure 2 depicts the embedding process in frames and audio. Besides the Fig. 3 describes the watermark extraction and the integrity verification process.

The proposed digital watermarking scheme is applied for fact-checking and fake news video inquiry. Through this scheme, we use digital watermarking for a combination of applications including proof of ownership and content description. For a given input mp4 video, we apply simultaneously an audio watermarking for the audio channel (the audio is an MP3 format file) and an image watermarking for frames that are ppm format. Therefore we use two descriptive watermarks. These watermarks should include enough information to guarantee the detection of content manipulation in the two channels of video. For each channel, the video allocated to verification must go through a stage of watermark detection and integrity verification. The final decision is based on the result of the OR function applied to the two decisions resulting from the detection algorithm of the audio and frames channels.

Algorithm 1: Algorithm of video watermarking

Data: V: original video, wat: watermark vector
Result: V_W watermarked video

Read the original video V;
Demultiplex V to get the frames and the audio;
X=read(MP3_file);
X1=read(MP3_Recompressed_File);
[Big region1, index big region1]=detect region(X);
big region2,index big region2]=detect region(X1);
indice=calcul min (big region1,index big region1, big region2, index big region2, X, X1);
Y=X;
[FV1 FV2 FV3 FV4 FV5 FV6 FV7 FV8]=feature extraction (X);
signature=MD5([FV1 FV2 FV3 FV4 FV5 FV6 FV7 FV8]);
for $i \leftarrow 1$ *to numberOf(frames)* **do**
    Convert frames(i) from RGB to YCbCr;
    Seperate Y, Cb and Cr components;
    Divide Y and Cb into 8×8 blocks;
    Y_{DCT}=DCT(Y);
    Cb_{DCT}=DCT(Cb);
    Wat=dec2bin(wat) ; `/* Convert the watermark to binary */`
    Y_W = insertion_Watermark_FrameF(Y_{DCT},wat);
    Cb_W = insertion_Watermark_Frame(Cb_{DCT},wat);
    iwycbcr= concatenate (Y_W, Cb_W, Cr);
    $frame_W$=Convert iwycbcr to RGB;
    Save $frame_W$ as watemarked frame;
end
$Audio_W$= insertion_Watermark_Audio(X, signature);
Multiplex $frame_W$ and $Audio_W$ to get V_W

Algorithm 2: Algorithm of fake verification

Data: V_w watermarked video, watAudio Embedded watermark in audio, secret_key_audio
Result: $watAudio_{EX}$ extracted watermark

Read the watermarked V_w;
Demultiplex V_{wA} to get the watermarked attacked frames and audio;
X=read the MP3 watermarked file;
for $i \leftarrow 1$ *to numberOf(frames)* **do**
    Convert $frames_{wA}$(i) from RGB to YCbCr;
    Seperate Y, Cb and Cr components;
    $featFrames_{WA}$=featureExtraction(Y,Cb,Cr); `/* get texture and color features */`
    Divide Y and Cb into 8× blocks;
    Y_{DCT}=DCT(Y);
    Cb_{DCT}=DCT(Cb);
    $watFrames_{EX}$=watermark_extraction_frames(Y_{DCT})
    $watFrames_{EX}$=watermark_extraction_frames(Cb_{DCT})
end
for $i \leftarrow 1$ *to size(secret key)* **do**
    signature=X(secret key(i));
end
$watAudio_{EX}$=watermark treatment(signature, watAudio);
[FV1 FV2 FV3 FV4 FV5 FV6 FV7 FV8]= feature extraction(X);
$featAudio_{EX}$=MD5([FV1 FV2 FV3 FV4 FV5 FV6 FV7 FV8]);
if $watAudio_{EX} \neq featAudio_{WA}$ *or* $watFrames_{EX} \neq featFrames_{WA}$ **then**
    message(The video is fake!); `/* Fake verfication */`
end

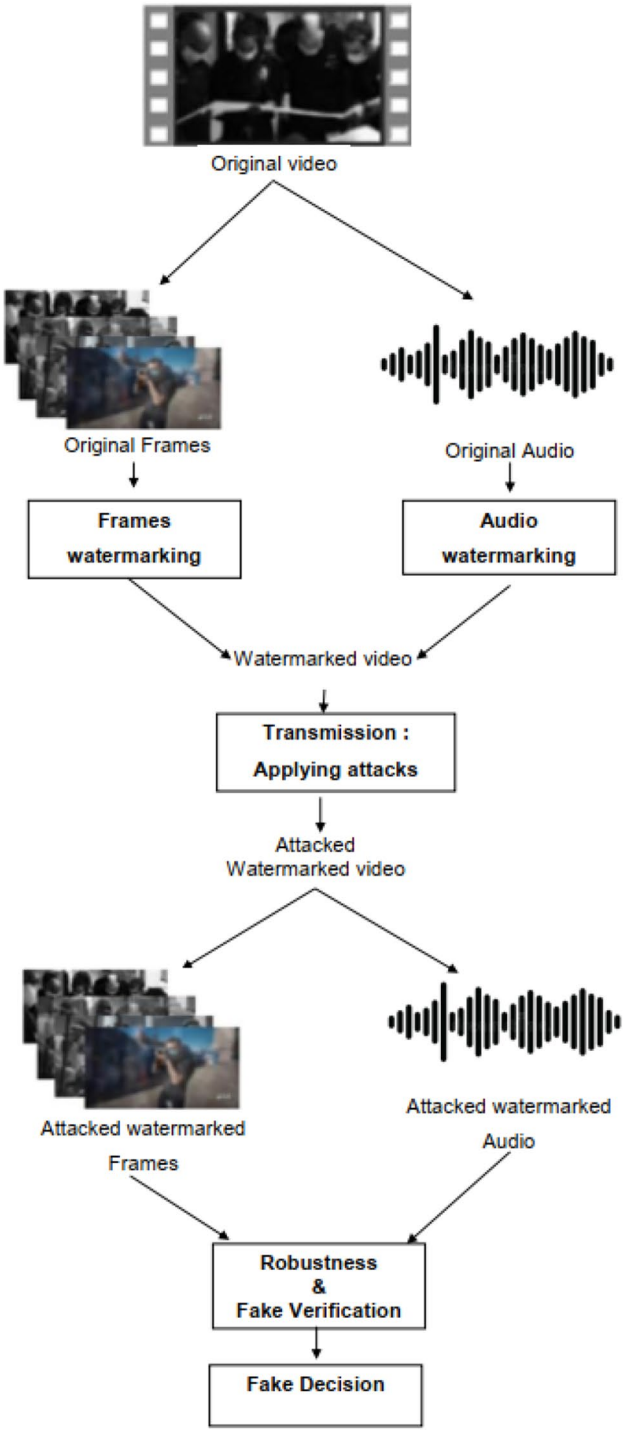


Fig. 1 General flowchart

3.1 Audio stream watermarking

This section describes the process of concealing, extracting, and deciding on the integrity of the suggested audio watermarking technique.

The suggested watermarking system takes an MP3 bitstream as input. It embeds watermark using Huffman data extracted directly from the compressed bitstream. This watermarking technique is based on the scheme described in the publication [7], however it has been improved to regulate the integrity of MP3 audio files. This scheme is based on the MP3 side information features and the recompression effects. The step of features extraction helps us to construct the watermark. The suggested watermarking system is divided into four sections: watermark generation, watermark embedding, watermark extraction and integrity verification.

3.1.1 Watermark construction

Extracted audio features represent the content-fragile watermark. This watermark is robust against allowed signal attacks, but it can also detect content manipulations. When we use MP3 file, we focus on the features of MPEG audio. To circumvent the time-consuming problem, we use features extracted directly from the MP3 bitstream without decoding [6]. Therefore, the embedding watermark is a set of side information calibrated features. To avoid the embedding capacity problem, we utilize a checksums function. Instead of inserting the features vector,we only insert their checksums vector. The checksums can be compared to the recalculated attacked and watermarked bitstream features checksums to detect the content modifications. The hash function MD5, [9], is computed as checksums of the features vector. Therefore, the used watermark in this work is a 128-bit binary sequence.

3.1.2 Watermark embedding

First, the host MP3 audio file undergoes a step of silence trimming [10]. The second step is the use of a partial MP3 decoder to extract the header, the scale factors, also the side information, and then the Huffman data of every frame. As a third step, we proceed to use the Huffman decoder to detect the big value region. This watermarking algorithm utilizes the Huffman data to boost the embedding capacity (More details in [7]). We use bits of Huffman data essentially candidate bits to be in the big values region (region2) of the

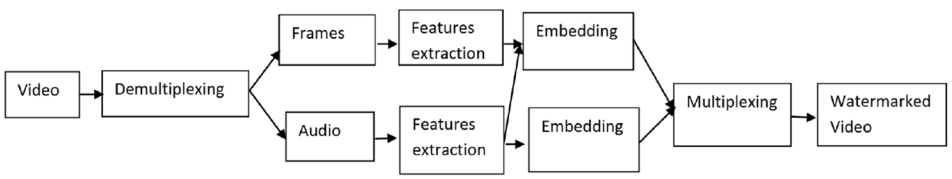


Fig. 2 Proposed watermarking scheme's embedding process

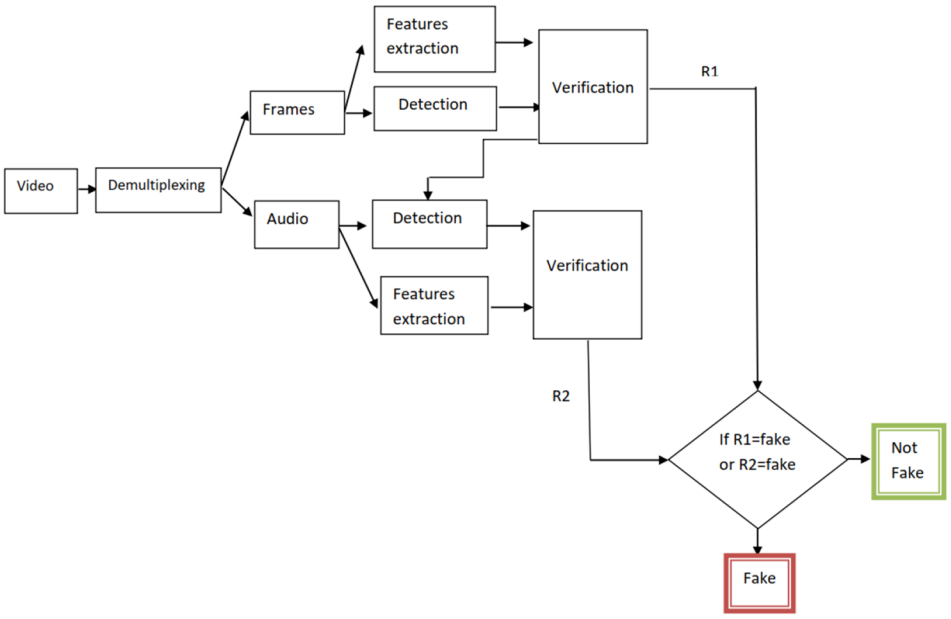


Fig. 3 Proposed watermarking scheme's detection and Integrity verification process

mp3 frame selected in the calibration step. This bit is picked out by the calibration of the MDCT distribution. The choice of the region2 from a big value part to hide the watermark is argued by the fact that region 2 holds spectral information in the high-frequency range (5–14 kHz at 44, 1 kHz sampling rate) [6] and most of the spectral energy is concentrated in region0 and region1 of the signal due to energy compaction properties of MDCT [6, 11], thus any modification in this region introduces lower noise in the host. The candidate bit should also verify that after embedding the index of the Huffman table does not change. The Embedding strategy is substitutive; we replace the bit by the existing watermark bit.

3.1.3 Watermark detection and fake verification

The extraction process is a blind extraction of an embedded watermark that does not require the carrier audio. The embedding process uses the insertion position found in the embedding process. These positions form the secret key of the proposed scheme. In addition, to enhance the detection of the right watermark we try to correct the watermark detected by recuperating the audio watermark (MD5 of Audio features) extracted from the watermarked frames stream and we apply an OR function. This process of detection is rapidly done due to the fact that we do not need to use a part of the decoder. During fake verification, we compare the vector of features extracted from watermarked file with the extracted watermark (original embedded features). If changes are detected, current content and embedded watermark are different, and the system sends a warning message (Fig. 4).

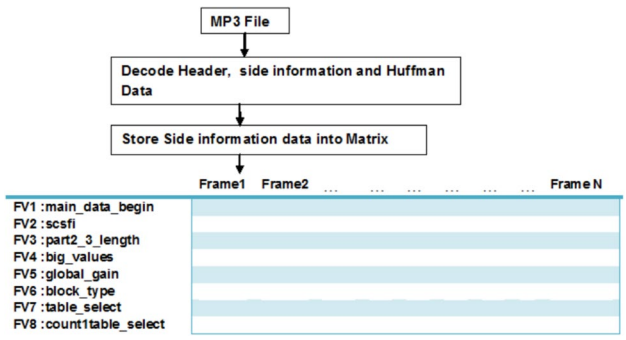


Fig. 4 Side information features extraction

3.2 Frames stream watermarking

In our method, both frames and audio channel are watermarked using different watermarks.

3.2.1 Watermark generation

The proposed scheme employs two watermarks. The first one is the result of combining the color and texture characteristics of the original frame. The second one is the audio features.

All along with watermark generation, the RGB frame is converted to YCbCr before dividing the Y and CbCr components by 64. DCT is applied to every block after components splitting to extract texture and color characteristics from Y, Cb and Cr.

Texture characteristics Texture characteristics are important low-level elements of an image. Color features alone are not enough to recognize an image because histograms from different images can look the same. So, texture

characteristics can be used to define visual elements in addition to color features.

The gray level co-occurrence matrix (GLCM) is an important indicator for describing texture. It is an $M \times M$ square matrix, where M represents the number of gray levels in the main image. Probability value of two pixels separated by direction θ and distance d, with x gray-scale intensity and y gray-scale intensity, is stored in that matrix. As a result, the probability value is denoted by $P(x, y, d, \theta)$. The distance d can be any number between 1 and 8, and the direction can be any number between 0 and 45, 90, 135, 180, 225, 270, and 315.

Several statistics are extracted from the GLCM to characterize the texture of the image. Energy, homogeneity, and contrast are the statistical measures used in this paper, with 0 representing the chosen direction and 1 representing distance.

Energy This is often referred to second angular momentum (ASM). In GLCM, energy is calculated as the sum of square elements.

$$\sum_{x,y=0}^{M-1} P^2(x, y, d, \theta) \tag{1}$$

Homogeneity It computes the closeness of the GLCM elements' distributions to the GLCM diagonal. The diagonal GLCM has a value of 1 and a range of [0,1]. Homogeneity is determined by:

$$\sum_{x,y} \frac{P(x, y)}{1 + |x - y|} \tag{2}$$

Contrast It computes the intensity contrast between a pixel and its neighbor across the entire image. In a smooth image, contrast is low, while in a coarse image, contrast is high. It is determined by:

$$\sum_{x,y=0}^{M-1} (x - y)^2 P(x, y, d, \theta) \tag{3}$$

Color features Using standard deviation and means, we extract important visual cues from Cb and Cr planes. The brightness of the image is revealed using the mean.

$$\text{mean} = \frac{1}{N} \sum_{i=0}^{N-1} X_i \tag{4}$$

The contrast of images is represented by the standard deviation.

$$\text{standard deviation} = \sigma = \sqrt{\frac{1}{L} \sum_{i=0}^{L} (X_i - \mu)^2} \tag{5}$$

where the number of elements in a component is denoted by X_i, while the chrominance components size is denoted by L.

Finally, we selected to utilise the average of the output vector as our watermark to decrease the amount of the retrieved texture and color characteristics.

3.2.2 Watermark embedding algorithm

This section describes the procedure for inserting a watermark. The procedure is shown in Fig. 3 and is based on our prior work in [24]. First, the SURF features of the original frame are extracted. These characteristics are stored as element of the watermarking key. The RGB frames are then transformed to YCbCr. Because its constituents have minimal correlations [25], YCbCr is employed. As long as, the YCbCr color space is adopted in the JPEG standard, which improves the robustness of the results. The Y and Cb components are used for watermark embedding. Cb is chosen because it is resistant to geometric attacks and Y is chosen as it's resistant to the well-known attack JPEG compression. The selected components are then divided into 8×8 blocks, totaling $(M \times M)/64$ blocks where $M \times M$ is the size of Y and Cb. The pixels in the blocks are all subtracted by 128 before being translated to the frequency domain with DCT. Following that, we quantized and zigzagged scanned all 64 DCT coefficients. A treatment algorithm determines the suitable blocks to ensure the best imperceptibility and robustness. This algorithm was thoroughly detailed in our prior publication [24]. After completing the watermark generation, we inserted the watermark in the LSB of the selected coefficients, exactly in the middle band frequency. For the following reasons, we chose frequency coefficients from the middle band:

The majority of an image's energy is concentrated at low frequencies. As a result, inserting there will have an effect on the beheld quality. So, the required level of imperceptibility will be missed. Furthermore, the high band frequency will be deleted after applying lossy compression, image noise and low-pass filter. As a result, using that band will not satisfy the robustness requirement. So, the middle band frequency is recommended for embedding the watermark as it assures imperceptibility and robustness improvement in various watermarking algorithms with different input like image and audio.

3.2.3 Watermark extraction and fake verification

Because we do not use the original frame, the detection scheme is blind. The extraction procedure is depicted as follows:

First, transform the watermarked frame to the YCbCr color space and detach the Y, Cb, and Cr components. Then

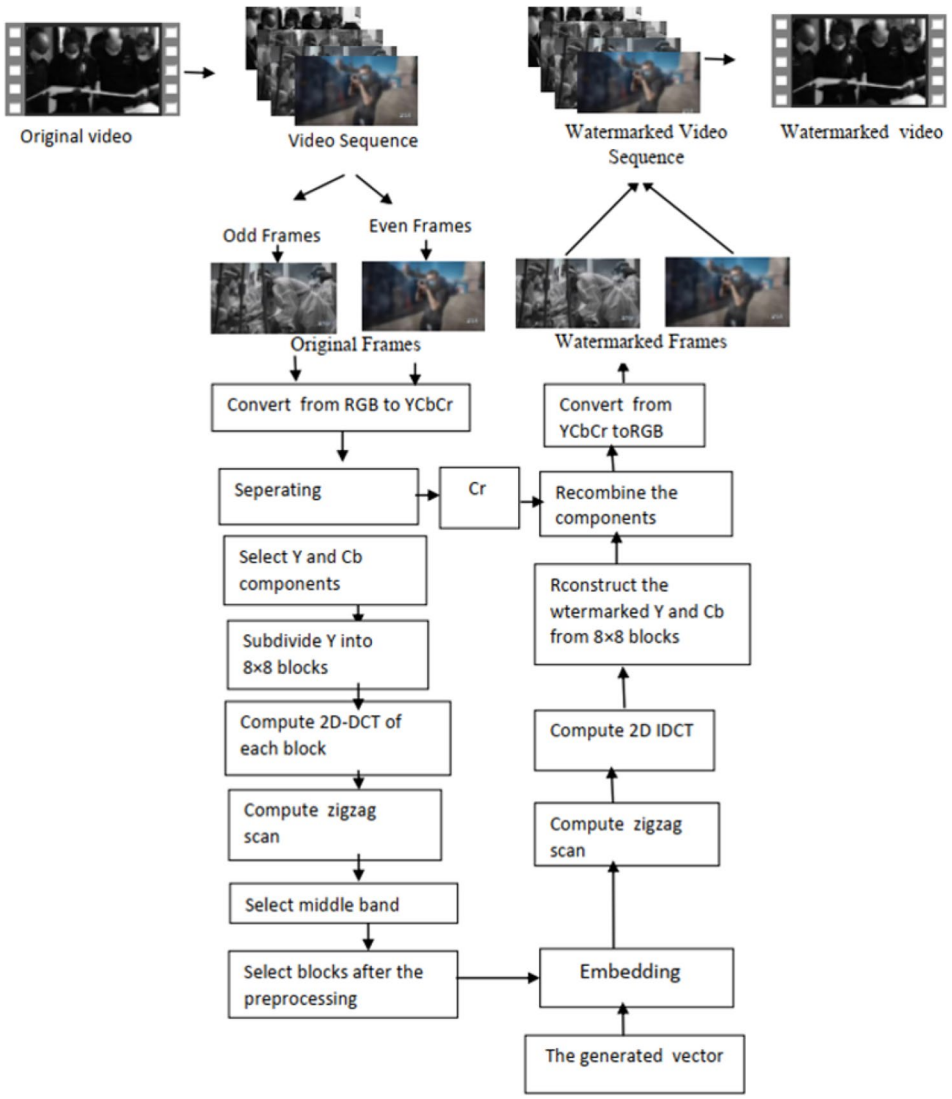


Fig. 5 The frames watermarking approach's flow chart

Table 1 Details of the videos

Video	Size in pixels	fps	Number of frames	Length in second
Vid1	1280 × 720	29.97	4230	141
Vid2	1280 × 720	29.97	5637	188
Vid3	1280 × 720	29.97	4326	144
Vid4	1280 × 720	29.97	4049	135
Vid5	1280 × 720	29.97	4200	140

split the Y and Cb components into blocks of 8x8 pixels. Every block is then increased by 128 and DCT, quantization, and zigzag scans are applied to the 64 DCT coefficients. Then we runs the detection mechanism with the use of a key consisting of the blocks index generated after the

Table 2 Frames watermarking and integrity verification time computation

	SURF elicitation process/frame	Block choice process/frame	Watermark insertion process	Watermark detection and integrity verification
Mean duration (s)	0.05	0.89	36.1	0.67

Table 3 MP3 audio watermarking time computation

	Feature elicitation process/frame	Big region choice process/frame	Watermark insertion process	Watermark detection and integrity verification
Average time (s)	0.6	0.4	1.5	2

pretreatment and the SURF characteristics, and then finally merge the extracted bits to get the watermark.

To see if the frame has been tampered with after the Stirmark attack, we tend to compare the vector derived from the attacked frame with the vector that is the outcome of the detection mechanism. Also, check the detection mechanism outcome to the original watermark to ensure that the copyright is preserved (Fig. 5).

4 Experimental results

The efficiency of our system is presented in this part. The experiments make use of COVID-19 news videos. These videos are MPEG 4 Format. The details of these test videos are described in Table 1 The experiments were carried out on a machine with a 2.00 GHz Intel Core i74510U processor and 8 GB of RAM, and MATLAB 15a software was used. Table 2 shows the mean duration in seconds for SURF elicitation, block choice, watermark insertion and detection for frames watermarking. Concerning audio watermarking, the mean duration of feature elicitation, big region choice, insertion and detection are presented in Table 3. Each process's calculation time is acceptable, confirming the effectiveness of the watermarking methodology in achieving video authenticity.

4.1 Video watermarking experimental results

4.1.1 Metrics

The Peak Signal to Noise Ratio (PSNR) is applied to assess our scheme's imperceptibility. The PSNR is employed to check the perceptual quality of the watermarked video after watermark insertion [27]. PSNR is measured in decibels (dB) and is calculated by:

$$\mathrm{PSNR} = 20 \times \log \frac{2^d - 1}{\sqrt{\frac{1}{H \times W} \sum_{x=1}^{H} \sum_{y=1}^{W} \sum_{j=1}^{c} (\mathrm{Fr}(x,y)\mathrm{Fr}'(x,y))^2}} \tag{6}$$

In which H, W are the dimensions of the frame, Fr and Fr′ are the original and watermarked frames respectively, d and c values are 8 and 3 respectively in case of RGB frame with 256 diverse gray levels.

Robustness measurement determines the persistence of the hidden signature. Thus, the normalized correlation (NC) is used as an evaluation metric. NC is used to calculate the correlation between the hidden mark and the retrieved one:

$$\mathrm{NC} = \frac{\sum_{i,j=1}^{n} \mathrm{wa}_{\mathrm{original}}(i,j)\mathrm{wa}_{\mathrm{watermarked}}(i,j)}{\sqrt{\sum_{i,j=1}^{n} \mathrm{wa}_{\mathrm{original}}(i,j)^2 \sum_{i,j}^{n} \mathrm{wa}_{\mathrm{watermarked}}(i,j)^2}} \tag{7}$$

Table 4 ODG values

Audio stream	Vid1.mp3	Vid2.mp3	Vid3.mp3	Vid4.mp3	Vid5.mp3
ODG	0.0402	0.2020	0.1619	0,19	0.04

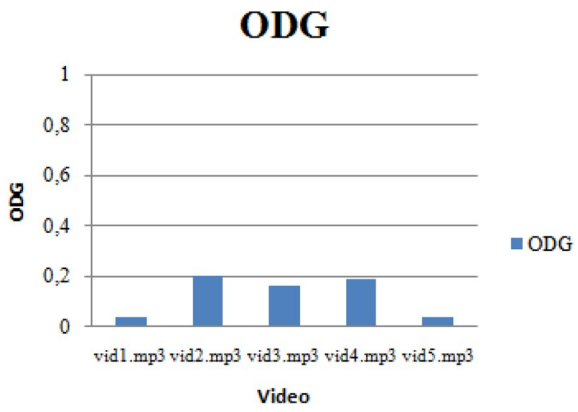


Fig. 6 ODG values

where $\mathrm{wa}_{\mathrm{original}}(i,j)$ and $\mathrm{wa}_{\mathrm{watermarked}}(i,j)$ are the hidden watermark and the extracted watermark, respectively. If NC ≥ 0.75, the hidden watermark and the retrieved watermark are considered similar [24].

To check the content integrity of the extracted audio stream and frames of videos, we used the bit error rate (BER).

BER is the proportion of the number of mistake bits to the full number of bits obtained. BER is designated as:

$$\mathrm{BER} = \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{\mathrm{wa}_{\mathrm{original}}(i,j) \bigoplus \mathrm{wa}_{\mathrm{watermarked}}(i,j)}{n \times m} \tag{8}$$

where $W(i,j)$ and $W'(i,j)$ are the original watermark and the extracted watermark, respectively, with dimensions of n and m and $\bigoplus$ means the xor operation.

Video PSNR, NC, and BER esteems are determined as the mean of the values for every video frame. For example, for a video consisting of N_{F} frames, the BER value is determinated as:

$$\mathrm{BER} = \frac{\sum_{i=0}^{N_{\mathrm{F}}} \mathrm{BER}_{\mathrm{F}_i}}{N_{\mathrm{F}}} \tag{9}$$

To evaluate the proposed video watermarking performance, we need to perform the tests for both frames and audio streams.

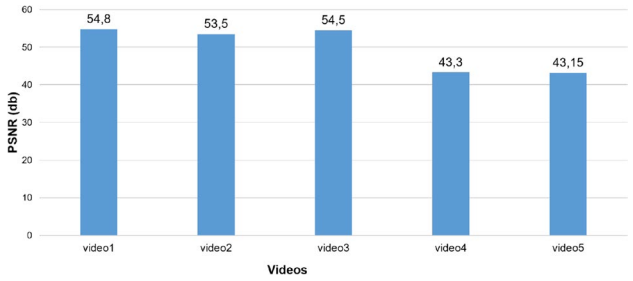


Fig. 7 PSNR values of several watermarked videos

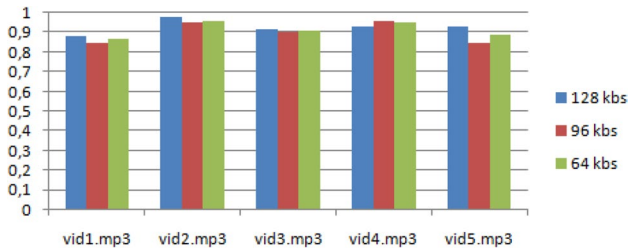


Fig. 8 NC values corresponding to hidden watermark and retrieved on after MP3 doubly compression attacks

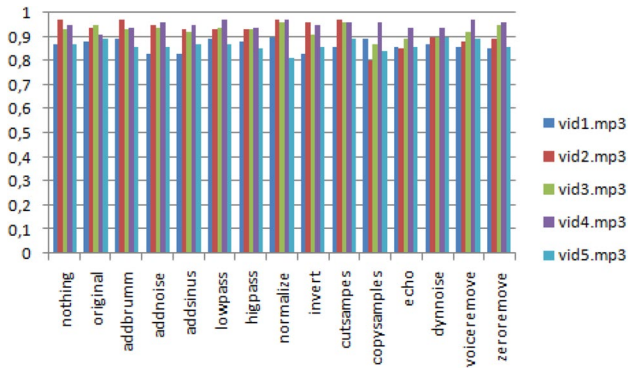


Fig. 9 NC values corresponding to hidden watermark and retrieved on after StirMark attacks

4.1.2 Audio stream imperceptibility results

Transparency performance aims to ensure that the watermarking does not degrade considerably the host Bitstream. Otherwise, the watermark embedding process should not introduce a noticeable noise in the host carrier. Thus, an objective metric, the objective difference grade (ODG) [12] is used. ODG can take a value between − 4 and 0. The closer the ODG value is to 0, the more the degradation is imperceptible. Reported results for some MP3 digital audio are presented in Fig. 6 and Table 4. The achieved ODG values display that the watermark transparency is confirmed by a positive ODG value which confirms that this elaborated algorithm fulfills the needed characteristics of an audio watermarking that is optimal in terms of inaudibility.

4.1.3 Frames stream imperceptibility results

Figure 5 presents the findings of applying the proposed watermarking scheme for various videos. A PSNR value above the threshold of 36 dB indicates a high-quality watermarked video. The good results of PSNR are due to the watermark insertion in the LSB of the frequency domain (Fig. 7).

4.1.4 Audio stream robustness results

Robustness against doubly MP3 compression Figure 8 shows that this technique of watermarking gives good results of robustness against MP3 doubly compression with different rate. In fact, the most value of NC are greater than 0.8.

Robustness against StirMark attacks In general, applying attacks to watermarked audio is done in the decompressed domain. As a result, the following actions are required to change MP3 watermarked audio. To begin, the watermarked MP3 audio will be decompressed and later some parts are altered using some StirMark attacks [13] such as adding noise (fftnoise, dynnoise, addsinus,addbrummm,echo), filters (highpass and lowpass), content transformation attacks

Table 5 Robustness results of COVID-19 videos against JPEG compression

Videos	Vid1		Vid2		Vid3		Vid4		Vid5	
	NC	BER	NC	BER	NC	BER	NC	BER	NC	BER
Quality factor										
15	1	0	0.98	0.02	0.98	0.02	0.95	0.06	0.8	0.09
20	1	0	0.98	0.02	0.99	0.01	0.98	0.02	0.99	0.009
25	1	0	0.98	0.02	1	0	0.97	0.03	1	0
30	0.99	0.009	0.99	0.01	0.98	0.02	0.97	0.03	1	0
35	0.99	0.009	1	0	0.98	0.02	0.96	0.04	1	0
50	1	0	0.99	0.009	1	0	0.97	0.03	0.99	0.009
70	1	0	1	0	0.99	0.01	0.98	0.02	1	0
80	1	0	1	0	1	0	0.98	0.02	1	0
90	1	0	1	0	1	0	0.98	0.02	0.96	0.04

(like copying, slicing and flipping samples). Finally, the attacked audio is recompressed and reconstituted as a new MP3 Bitstream. The Fig. 9 displays the NC values of the hidden and retrieved mark of decompressed and attacked watermarked Bitstream. Despite the fact that the test signal (MP3 watermarked and attacked audio) is doubly attacked (decompression + StirMark attack NC values are near to 1 in the most cases) confirming the proposed scheme's robustness to various attacks.

4.1.5 Frames stream robustness results

The Stirmark 3.1 benchmark attacks are employed to test the robustness of the schema. A strong watermarking technique should be able to withstand various attacks and signal distortions. Following the detection step, we compute the NC and BER values to assess the robustness against frame deterioration and handling.

Robustness against JPEG compression Table 5 displays the experimental results of Stirmark JPEG attacks on videos frames. Before being compressed with JPEG, the watermarked frames are in '.ppm' format. The proposed method, as demonstrated, is resistant to JPEG attack at rates ranging from 15 to 90. Actually, the NC values range from 0.98 to 1, and the BER values range from 0.01 to 0. We get excellent NC and BER results because the watermark bits are embedded in the *Y* component, which is used in the JPEG standard and is resistant to JPEG compression attacks.

Unlike other methods that decompress the frames before detection [26], we detect directly the watermarks from the attacked frames without decompressing them. This feature underscores the novelty of our technique.

Robustness to Stirmark's unique attacks We tested our method against Stirmark attacks in addition to JPEG compression. The original and watermarked frames are in ppm format, while the outcomes of Stirmark are in both ppm and jpeg. Stirmark's unique attacks are those with ppm format.

Figure 9 shows the NC values of the retrieved watermark from the attacked frames after applying geometric and combined distortions; the attacked frames pictures are in ppm format.

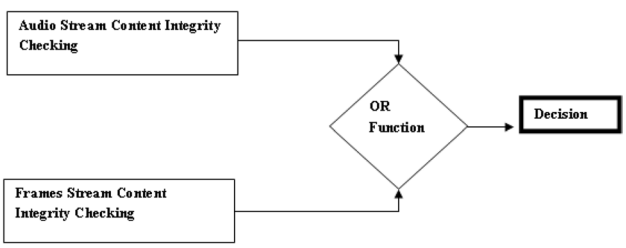


Fig. 10 Fake video detection process

Looking at the results in Fig. 9, it can be seen that, our method is tolerant to conventional signal processing such as Gaussian filters and median filters as well as various geometric distortions such as symmetric and asymmetric lines, scaling from 0.5 to 2, and column deletion, cropping, shearing. In all cases, the NC values exceed the predetermined threshold $T_{NC} = 0.75$ [27] and the watermark is extracted without improving the attacked frames [28].

Robustness to Stirmark's double attacks We analyze our approach against duplicate attacks in addition to unique attacks.

Stirmark Benchmark merges the attacks with the exceedingly destructive JPEG compression. This type of combination is known as dual attacks. Figure 9b shows performances against standard signal processing, including Gaussian filter and average, symmetrical lines and column removals, geometrical attacks such cutting and rotation

The proposed approach is resistant to double attacks, has BER values below $T_{NC} = 0.2$, and recovers the watermark from the JPEG attacked frame, eliminating the need to decompress it as in prior methods [28].

Because of the embedding of the watermark in the Cb component of the YCbCr, our method is resistant to a variety of geometric distortions and signal processing. In addition, using the SURF characteristics to resynchronize the attacked frame has improved the results after destructive geometric attacks like cropping and rotation.

4.2 Fake video detector

This section shows the results of fake video detection. In other words, the tests of content integrity verification of selected news videos. The Fig. 10 shows that the decision video fake or not is related to content integrity checking of both audio and frames stream.

4.2.1 Audio content integrity checking

To check whether the watermarked audio stream is changed or not after StirMark attacks and some content manipulation attacks as adding silence, removing and replacing parts, the idea of integrity checking is based on comparing the characteristics of the embedded content with the actual content of the attacked watermarked MP3 audio file.

If no attack occurs, the bit rate error (BER) is equal to zero. Table 8 displays findings for the feature vector after applying a StirMark benchmark audio attack and content manipulation attacks. The process's basic idea is to embed a feature vector and then apply a variety of audio manipulations of varying strengths to the marked file. The watermark is then detected, and the checksums of the retrieved and recalculated feature vectors are compared. The attacks that preserve audio content, such as "invert", "normalize"

Table 6 Comparison between the proposed approach and other existing video watermarking schemes using NC

Methods	Referenced [30]	Referenced [29]	Proposed method
Attacks			
Noise	0.75	0.87	0.89
Filter	0.75	0.52	0.84
Cropping	0.76	–	0.85
Blurring	0.81	–	1
Sharpening	–	0.94	1

– Data not available

and "amplify" produce the same error rates as the "nothing" attack used by StirMArk BenchMark or after an ideal exchange (BER = 1). An error rate equal or below the bit error of the no operation attack can be seen as a threshold for operations that preserve the audio content. Content manipulations like the addition of silence and some noises, removal of voice and removal of samples have higher error rates than the no operation attack. The results prove that some attacks like filters (lowpass filter and highpass filter) and voice remove attack may be considered to be content preserving attacks in some cases. The results also show that the error rates varies with the attack strength,i.e.lower noise value lead to lower error rates.

Table 7 Comparison between the proposed approach and other existing video watermarking schemes using BER

Methods	Referenced [31]	Proposed method
Attacks		
JPEG compression (Quality factor = 50)	0.52	0.04
sharpening (alpha = 0.1)	0.0046	0
blurring (Sigma = 0.5)	0.005	0
Gaussian noise (variance = 0.01)	0.01	0.07
Gamma correction (alpha = 0.5)	0.05	0.05
Frames dropping	0.056	0
Frames swapping (50%)	0.1	0

– Data not available

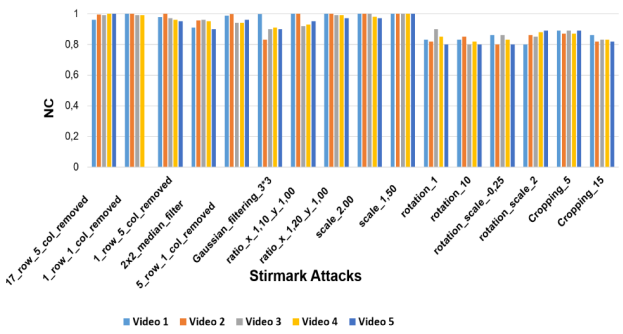
4.2.2 Frames stream content integrity checking

As previously stated, integrity verification is the process of comparing embedded content features to the actual content of the frames that have been attacked. The integrity is examined using JPEG compression, as well as double and unique attacks of stirmark.

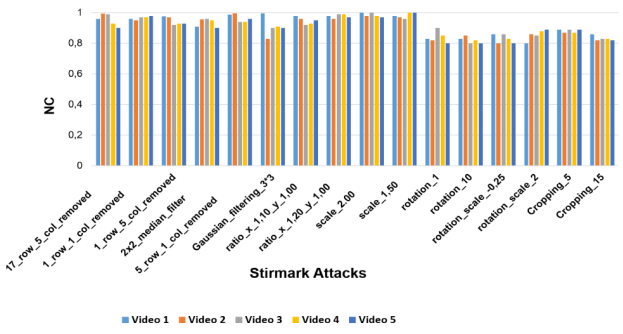
Tables 9 and 10 show the BER values. They can be used not just to compare the acquired features with the actually extracted features, but also to determine if the integrity is compromised. In case that the BER results are bordering to zero, the frame's content is unaffected; if they are greater than 0.2, the frame is altered, so the video is fake.

5 Comparative study

A comparison with previously established methods is performed to further validate the performance of the proposed algorithm. Analyzing the NC results in Table 6, it can be noted that our method outperforms [29] and [30] in noise, filter, cropping, blurring and sharpening attacks. The video watermarking technique presented in [29] is in DCT domain and based on meta-heuristic algorithm and the method in [30] is based on dual-tree complex wavelet transform and adaptive cuttlefish optimization algorithm. Also, based on the BER results in Table 7, it is possible to conclude that our



(a) The NC values of stream frames against unique attacks of Stirmark



(b) The NC values of stream frames against double attacks of Stirmark

Fig. 11 The NC values against Stirmark's unique and double attacks

Table 8 Audio stream content integrity verification

Videos	Vid1		Vid2		Vid3		Vid4		Vid5	
	BER	Decision	BER	Decision	BER	Decision	BER	Decision	BER	Decision
Stirmark attacks										
Nothing	0.32	NOT FAKE	0.39	NOT FAKE	0.45	NOT FAKE	0.49	NOT FAKE	0.37	NOT FAKE
Addbrumm_100	0.32	NOT FAKE	0.39	NOT FAKE	0.47	FAKE	0.38	NOT FAKE	0.37	NOT FAKE
Addnoise_100	0.32	NOT FAKE	0.45	FAKE	0.55	FAKE	0.66	FAKE	0.36	NOT FAKE
Copysample	0.33	FAKE	0.5	FAKE	0.52	FAKE	0.63	FAKE	0.4	FAKE
Cutsamples	0.33	FAKE	0.51	FAKE	0.47	FAKE	0.6	FAKE	0.62	FAKE
Echo	0.32	NOT FAKE	0.42	FAKE	0.48	FAKE	0.61	FAKE	0.5	FAKE
rc_highpass	0.32	NOT FAKE	0.39	NOT FAKE	0.47	FAKE	0.67	FAKE	0.33	NOT FAKE
rc_lowpass	0.34	FAKE	0.4	FAKE	0.47	FAKE	0.6	FAKE	0.47	FAKE
Invert	0.32	NOT FAKE	0.39	NOT FAKE	0.47	FAKE	0.61	FAKE	0.37	NOT FAKE
Normalize	0.32	NOT FAKE	0.39	NOT FAKE	0.525	FAKE	0.49	NOT FAKE	0.37	NOT FAKE
Voiceremove	0.34	FAKE	0.42	FAKE	0.49	FAKE	0.56	FAKE	0.35	NOT FAKE
Content manipulationattacks										
Silence	0.53	FAKE	0.47	FAKE	0.46	FAKE	0.44	NOT FAKE	0.48	FAKE
Removal parts	0.42	FAKE	0.45	FAKE	0.44	NOT FAKE	0.5	FAKE	0.47	FAKE
Replacement	0.49	FAKE	0.51	FAKE	0.5	FAKE	0.53	FAKE	0.48	FAKE

Table 9 Frames fake checking of COVID-19 videos after unique attacks of Stirmark using BER

Videos	Vid1		Vid2		Vid3		Vid4		Vid5	
	BER	Decision	BER	Decision	BER	Decision	BER	Decision	BER	Decision
Attacks										
1 row 1 col removed	0.07	NOT FAKE	0.15	NOT FAKE	0.07	NOT FAKE	0.14	NOT FAKE	0.1	NOT FAKE
1 row 5 col removed	0.1	NOT FAKE	0.14	NOT FAKE	0.1	NOT FAKE	0.13	NOT FAKE	0.11	NOT FAKE
5 row 17 col removed	0.17	NOT FAKE	0.18	NOT FAKE	0.15	NOT FAKE	0.17	NOT FAKE	0.12	NOT FAKE
2×2 median filter	0.2	FAKE	0.26	FAKE	0.27	FAKE	0.27	FAKE	0.3	FAKE
3×3 median filter	0.13	Not FAKE	0.1	NOT FAKE	0.13	NOT FAKE	0.17	FAKE	0.13	NOT FAKE
Gaussian filtering 3×3	0.25	FAKE	0.29	FAKE	0.33	FAKE	0.29	FAKE	0.35	FAKE
Cropping 1%	0.23	FAKE	0.25	FAKE	0.29	FAKE	0.3	FAKE	0.4	FAKE
Cropping 2%	0.26	FAKE	0.2	FAKE	0.25	FAKE	0.3	FAKE	0.2	FAKE
Cropping 10%	0.27	FAKE	0.3	FAKE	0.38	FAKE	0.4	FAKE	0.37	FAKE
Cropping 15%	0.27	FAKE	0.3	FAKE	0.38	FAKE	0.23	FAKE	0.3	FAKE
Cropping 25%	0.28	FAKE	0.3	FAKE	0.4	FAKE	0.29	FAKE	0.3	FAKE
Ratio *x* 0.80 *y* 1.00	0.13	NOT FAKE	0.14	NOT FAKE	0.07	NOT FAKE	0.02	NOT FAKE	0.1	NOT FAKE
Ratio *x* 0.90 *y* 1.00	0.14	NOT FAKE	0.12	NOT FAKE	0.07	NOT FAKE	0.13	NOT FAKE	0.03	NOT FAKE
Ratio *x* 1.20 *y* 1.00	0.13	NOT FAKE	0.11	NOT FAKE	0.06	NOT FAKE	0.07	NOT FAKE	0.12	FAKE
Rotation − 0.75	0.23	FAKE	0.3	FAKE	0.35	FAKE	0.34	FAKE	0.25	FAKE
Rotation 10	0.26	FAKE	0.36	FAKE	0.4	FAKE	0.29	FAKE	0.25	FAKE
Rotation 90	0.31	FAKE	0.34	FAKE	0.4	FAKE	0.27	FAKE	0.32	FAKE
Rotation scale − 0.25	0.3	FAKE	0.3	FAKE	0.3	FAKE	0.3	FAKE	0.35	FAKE
Rotation scale 5	0.33	FAKE	0.4	FAKE	0.4	FAKE	0.23	FAKE	0.3	FAKE
Rotation scale 90	0.3	FAKE	0.34	FAKE	0.4	FAKE	0.43	FAKE	0.36	FAKE
Scale 0.75	0.17	NOT FAKE	0.17	NOT FAKE	0.08	NOT FAKE	0.09	NOT FAKE	0.14	NOT FAKE
Scale 0.90	0.13	NOT FAKE	0.14	NOT FAKE	0.09	NOT FAKE	0.04	NOT FAKE	0.05	NOT FAKE
Scale 2.00	0.2	FAKE	0.12	NOT FAKE	0.04	NOT FAKE	0.05	NOT FAKE	0.1	NOT FAKE
Shearing *x* 1.00 *y* 0.00	0.2	FAKE	0.25	FAKE	0.2	FAKE	0.2	FAKE	0.27	FAKE
Shearing *x* 5.00 *y* 5.00	0.21	FAKE	0.34	FAKE	0.35	FAKE	0.34	FAKE	0.32	FAKE

Table 10 Frames fake checking of COVID-19 videos after double attacks of Stirmark using BER

Videos	Vid1		Vid2		Vid3		Vid4		Vid5	
	BER	Decision	BER	Decision	BER	Decision	BER	Decision	BER	Decision
Attacks										
1 row 1 col removed	0.09	NOT FAKE	0.1	NOT FAKE	0.07	NOT FAKE	0.13	NOT FAKE	0.12	NOT FAKE
1 row 5 col removed	0.14	NOT FAKE	0.15	NOT FAKE	0.12	NOT FAKE	0.17	NOT FAKE	0.19	NOT FAKE
5 row 17 col removed	0.18	NOT FAKE	0.19	NOT FAKE	0.14	NOT FAKE	0.11	NOT FAKE	0.1	NOT FAKE
2×2 median filter	0.3	FAKE	0.27	FAKE	0.28	FAKE	0.29	FAKE	0.34	FAKE
3×3 median filter	0.19	Not FAKE	0.14	NOT FAKE	0.18	NOT FAKE	0.2	FAKE	0.21	NOT FAKE
Gaussian filtering 3×3	0.34	FAKE	0.31	FAKE	0.33	FAKE	0.4	FAKE	0.41	FAKE
Cropping 1%	0.3	FAKE	0.35	FAKE	0.31	FAKE	0.32	FAKE	0.42	FAKE
Cropping 2%	0.29	FAKE	0.3	FAKE	0.35	FAKE	0.4	FAKE	0.29	FAKE
Cropping 10%	0.4	FAKE	0.35	FAKE	0.4	FAKE	0.43	FAKE	0.4	FAKE
Cropping 15%	0.3	FAKE	0.34	FAKE	0.4	FAKE	0.27	FAKE	0.36	FAKE
Cropping 25%	0.4	FAKE	0.41	FAKE	0.42	FAKE	0.43	FAKE	0.4	FAKE
ratio x 0.80 y 1.00	0.15	NOT FAKE	0.17	NOT FAKE	0.1	NOT FAKE	0.19	NOT FAKE	0.12	NOT FAKE
Ratio x 0.90 y 1.00	0.15	NOT FAKE	0.13	NOT FAKE	0.1	NOT FAKE	0.17	NOT FAKE	0.06	NOT FAKE
Ratio x 1.20 y 1.00	0.15	NOT FAKE	0.13	NOT FAKE	0.08	NOT FAKE	0.1	NOT FAKE	0.15	FAKE
Rotation − 0.75	0.29	FAKE	0.35	FAKE	0.38	FAKE	0.37	FAKE	0.3	FAKE
Rotation 10	0.29	FAKE	0.4	FAKE	0.42	FAKE	0.3	FAKE	0.27	FAKE
Rotation 90	0.33	FAKE	0.37	FAKE	0.42	FAKE	0.3	FAKE	0.36	FAKE
Rotation scale − 0.25	0.35	FAKE	0.37	FAKE	0.37	FAKE	0.38	FAKE	0.4	FAKE
Rotation scale 5	0.37	FAKE	0.42	FAKE	0.45	FAKE	0.3	FAKE	0.35	FAKE
Rotation scale 90	0.38	FAKE	0.4	FAKE	0.45	FAKE	0.5	FAKE	0.45	FAKE
Scale 0.75	0.19	NOT FAKE	0.19	NOT FAKE	0.1	NOT FAKE	0.12	NOT FAKE	0.16	NOT FAKE
Scale 0.90	0.15	NOT FAKE	0.16	NOT FAKE	0.1	NOT FAKE	0.09	NOT FAKE	0.1	NOT FAKE
Scale 2.00	0.25	FAKE	0.14	NOT FAKE	0.06	NOT FAKE	0.08	NOT FAKE	0.13	NOT FAKE
Shearing x 1.00 y 0.00	0.3	FAKE	0.35	FAKE	0.36	FAKE	0.38	FAKE	0.39	FAKE
Shearing x 5.00 y 5.00	0.3	FAKE	0.36	FAKE	0.4	FAKE	0.41	FAKE	0.43	FAKE

method is better than [31] for JPEG compression, sharpening, blurring, gamma correction, frames dropping and swapping attacks. In [31], the watermark is embedded in discrete wavelet transform and SIFT features are used to restore the attacked frames. [31] is better for gaussian noise attack.

6 Conclusion

This paper proposed a video watermarking-based solution to combat the spread of fake news videos and false information on social media mainly in the COVID-19 crisis. The first advantage of this scheme is the combination of frames and audio watermarking which makes possible the detection of modifications in the two video channels. In addition, the watermarking assures a fast detection of fake video. Experimental results show that our proposed method is highly imperceptible and robust against JPEG compression, Stirmark's attacks. Furthermore, the results show that the proposed technique performs significantly better than comparable existing techniques under a variety of attacks. At this stage of analysis, we would assert that though our findings can be regarded as worthwhile, our research remains a step that may be extended, taken further, and built upon as it offers further promising research directions. Indeed, in future works, we may concentrate on video recovery after fake detection and localization (Fig. 11).

Acknowledgements The research leading to these results has received funding from the Ministry of Higher Education and Scientific Research of Tunisia under the grant agreement number LR11ES48.

References

1. Cox, M., Miller, B.: Digital Watermarking. Academic Press, San Diego (2002)
2. El'Arbi, M., Charfeddine, M., Masmoudi, S., Koubaa, M., Ben Amar Chokri, C.: Video Watermarking algorithm with BCH error correcting codes hidden in audio channel. In: Proceeding of the Third IEEE Symposium Series in Computational Intelligence CICS, IEEE Symposium on Computational Intelligence in Cyber Security, Paris-France, pp. 164–170 (2011)

3. Masmoudi, S., Charfeddine, M., Ben Amar, C.: A robust audio watermarking technique based on the perceptual evaluation of audio quality algorithm in the multi-resolution domain. In: Proceedings of the 10th IEEE International Symposium on Signal Processing and Information Technology ISSPIT, Luxor-Egypt, pp. 326–331 (2011)
4. Chaabane, F., Charfeddine, M., Ben Amar, C.: A QR-code based audio watermarking technique for tracing traitors. In: 23rd European Signal Processing Conference (EUSIPCO 2015) Nice, pp. 51–55 (2015)
5. Chaabane, F., Charfeddine, M., Ben Amar, C.: A Survey on digital Tracing Traitors Schemes. In: Proceeding of the 9th IEEE Conference on Information Assurance and Security IAS, Gammarth-Tunisia, pp. 85-90 (2013)
6. Pan, D.: A Tutorial on Mpeg/audio compression. IEEE Multimedia, pp. 60–74 (1995)
7. Masmoudi, S., Charfeddine, M., Ben Amar, C.: A semi-fragile digital audio watermarking scheme for MP3-encoded signals using Huffman data. In: Circuits Systems and Signal Processing, pp. 1–16 (2019)
8. Steinebach, M., Dittmann, J.: Watermarking-based digital audio data authentication. In: EURASIP Journal on Applied Signal Processing, pp. 1001–1015 (2003)
9. Simitopoulos, D., Zissis, N., Georgiadis, P., Emmanouilidis, V., Strintzis, M.G.: Encryption and watermarking for the secure distribution of copyrighted MPEG video on DVD. Multimedia Syst. (2010). https://doi.org/10.1007/s10207-009-0100-7
10. Mezghani, E., Charfeddine, M., Ben Amar, C.: Audio Silence Deletion before and after MPEG video compression. In: 2013 International Conference on Computer Applications Technology (ICCAT), pp. 1–5 (2013). https://doi.org/10.1109/ICCAT.2013.6521969
11. Servetti, A., Testa, C., De Martin, J., e Informatica, D.: Frequency-selective partial encryption of compressed audio. Acoust. Speech Signal Process. (2003). https://doi.org/10.1109/ICASSP.2003.1200059
12. Union Internationale des Télécommunications (UIT) . Recommandation B.S. 1387 . Méthode de mesure objective de la qualité du son perçu (2001)
13. Lang, A., et al.: Audio watermark attacks: from single to profile attacks. In: Proceedings of the 7th Workshop on Multimedia and Security (ACM), New York, pp. 39–50 (2005)
14. Li, C., Yang, Y., Liu, K., Tian, L.: A Semi-Fragile Video Watermarking Algorithm Based on H. 264/AVC, Wireless Communications and Mobile Computing (2020)
15. Hammami, A., Hamida, A.B., Amar, C.B.: Blind semi-fragile watermarking scheme for video authentication in video surveillance context. Multimedia Tools Appl., 7479–7513 (2021)
16. Aditya, B.P., Avaneesh, U.G.K., Adithya, K., Murthy, A., Sandeep, R., Kavyashree, B.: Invisible semi-fragile watermarking and steganography of digital videos for content authentication and data hiding. Int. J. Image Gr. (2019). https://doi.org/10.1142/S0219467819500153
17. Tian, L., Dai, H., Li, C.: A semi-fragile video watermarking algorithm based on chromatic residual DCT. Multimedia Tools Appl., 1759–1779 (2020)
18. Shi, Y., Qi, M., Yi, Y., Zhang, M., Kong, J.: Object based dual watermarking for video authentication. Optik **124**(19), 3827–3834 (2013)
19. Liu, S., Bo-Wei Chen, D., Gong, L., Ji, W., Seo, S.: A real-time video watermarking algorithm for authentication of small-business wireless surveillance networks. In: International Journal of Distributed Sensor Networks (2015)
20. Kaur, G., Kasana, S.S., Sharma, M.K.: An efficient authentication scheme for high efficiency video coding/H. 265. Multimedia Tools Appl., 21245–21271 (2019)
21. Munir, R.: A secure fragile video watermarking algorithm for content authentication based on Arnold Cat Map. In: 4th International Conference on Information Technology (InCIT), pp. 32–37 (2019)
22. Vybornova, Y.: A new watermarking method for video authentication with tamper localization. In: International Conference on Computer Vision and Graphics, pp. 201–213 (2020)
23. Hasnaoui, M., Mitrea, M.: Semi-fragile watermarking for video surveillance applications. In: 2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO), pp. 782–1786 (2012)
24. Tarhouni, N., Charfeddine, M., Ben Amar, C.: A new robust and blind image watermarking scheme in frequency domain based on optimal blocks selection. In: International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, pp. 78–86 (2018)
25. Charfeddine, M., El'arbi, M., Amar, C. B.: A new DCT audio watermarking scheme based on preliminary MP3 study. Multimedia Tools Appl., 1521–1557 (2014)
26. Wang, C.-p., Wang, X.-y., Xia, Z.-q., Zhang, C., Chen, X.-j.: Geometrically resilient color image zero-watermarking algorithm based on quaternion exponent moments. J. Vis. Commun. Image Rep., 247–259 (2016)
27. Tarhouni N., Charfeddine M., Ben Amar C.: Novel and robust image watermarking for copyright protection and integrity control. Circuits Syst. Signal Process., 1–45 (2020)
28. Rashmi, S., Priyanka , Maheshkar, S.: Robust multiple composite watermarking using LSB technique. In: Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications, Advances in Intelligent Systems and Computing (2017). https://doi.org/10.1007/978-981-10-3153-3_21
29. Aman, B., Chirag, S., Khalid, M., Singh, Aman, N., Osman, A., Alwetaishi, M.: A robust video watermarking scheme with squirrel search algorithm. Comput. Mater. Continua, 3069–3089 (2022)
30. Dhevanandhini, G., Yamuna, G.: An effective and secure video watermarking using hybrid technique. Multimedia Syst., 953–967 (2021)
31. Manglem Singh, K.: A robust rotation resilient video watermarking scheme based on the SIFT. Multimedia Tools Appl., 16419–16444 (2018)