ISSN: 0976-9102 (ONLINE)
DOI: 10.21917/ijivp.2019.0293

A SPATIAL DOMAIN BASED SECURE AND ROBUST VIDEO WATERMARKING TECHNIQUE USING MODIFIED LSB AND SECRET IMAGE SHARING

Vivek Sharma, Mahesh Gangarde and Shruti Oza
Department of Electronics and Telecommunication Engineering, Pune Institute of Computer Technology, India

Abstract

This paper proposes a spatial domain based video watermarking scheme to improve security of the video data using a combination of modified LSB watermarking technique of spatial domain and a (n, n) secret image sharing scheme. In this scheme the secret image sharing scheme is applied during transmission right after the embedding of the watermark in the video frame to overcome the drawback of the single LSB technique. This combination ensures high security and robustness as the watermarked image is being distributed into 3 meaningless shares during transmission thus making it imperceptible to the attacker. This dual scheme improves the extraction capability of the secret message and enhances the information embedding capacity. The average PSNR obtained was 55dB which proves that the quality of reconstruction is high and higher than most of the existing watermarking techniques in similar domain.

Keywords:

LSB, Secret Image Sharing, Video Watermark, Robustness

1. INTRODUCTION

In the last few years, necessity of faster data transmission and modernization of technology has led to most people using the internet as the fastest means to transfer data. As the internet has gained increasing popularity, it has become necessary for the data transmission to be extremely secure. The internet has made the transmission secure, easy and prompt. Nowadays because of modernization of technology, security of digital multimedia is very much important and as they are digital in nature they can be morphed or replicated easily. The faster distribution of data over the network via images, audio, and video become a common resource and thus it enabling very easy way to transfer the data. Because of data portability, piracy and duplicity has reached its peak. The author or the producer of the data file is unaware of his work being spread on the internet for free and even when he gets aware of this breach, he cannot do anything.

Therefore, it is important to secure the information integrity in this mode of transmission and for this various techniques is being employed like digital watermarking, steganography, cryptography, encryption and decryption. Watermarking is basically a technique of hiding or embedding information (watermark) inside the content to be transmitted (cover) and then extracting it at receiver end. Watermarking ensures copyright protection, data integrity and authenticity. A good watermarking technique should be highly robust and imperceptible. Robustness ensures resilience to common signal processing and geometric attacks preserves the integrity of the information being transmitted. Imperceptibility is necessary for the security of the hidden content. Secret image sharing is the art and science about the protection of important images by distributed storages. The basic idea is to transform an image into multiple nondescript shadow images in such a way that a qualified subset of the shadow images can reconstruct the original image, but no secret information can be revealed by a forbidden subset of the shadow images. A secret sharing scheme can be evaluated by its security provided, reconstruction precision, computation complexity and storage requirement. A combination of digital watermarking technique (LSB) and steganography (image sharing) have been employed in this research to develop a robust technique.

2. LITERATURE REVIEW

Piracy has reached its peak mainly because of the modernization of internet and storage technology tied up with the vulnerability of internet media. Therefore, there is a dire need of research in content protection mechanisms and following this digital watermarking has gained interest from researchers for designing an effective algorithm for successful implementation. It consists of embedding secret data called as watermarks inside the video data which creates a facility for protecting copyright, data integrity and authentication.

Liu et al. [1] proposed a two level security and authentication technique. They employed secret image sharing as the first stage and embedded data into a cover image using matrix encoding in the second stage. Secret image sharing is a technique which provides high security in terms of authentication and keeps the data safe by employing multiple receivers. The shares generated by the image sharing stage are embedded in multiple cover images by matrix encoding by replacing pixel values of the cover images. At the receiver end, Huffman decoding is incorporated to extract the data from the cover images and subjected to further reconstruction of the shares into the secret image.

The watermarking techniques in the spatial domain use the pixel locations to embed the secret information while the techniques in the transformation domain use popular transformation like DCT, DWT, DFT etc. which take advantage of the spectral coefficients of the human visual system and these have been briefly explained by Sowmya and Chennamma [2] in their study for video authentication [2].

Singh et al. [3] performed experimental analysis on the popular LSB Watermarking technique to explain the impact of different types of noise. They also stated that watermarking is also a great technique employed for authentication, image protection and establish rights on a particular data.

A dual image technique was employed by Wang et al. [4] for developing a reversible hiding technique using LSB. They copied an image into two same images and through that they expected to embed and extract data with good quality. They used two pixels as a pair and chose two same images to embed and then chose both pixels to continue the algorithm. However, they concluded

the research that their proposed method provides better embedding capacity but decreases the quality of the image.

Giri and Bashir [5] stated digital watermarking as a potential solution for multimedia authentication. They conducted an exhaustive survey and explained the positives of digital watermarking with regard to protection of the owner copyrights and ensuring data integrity. Their survey is useful to understand the basic categories of watermarking.

Su and Chen [6] proposed a blind watermarking technique in which they embedded the binary watermark into the blue component of a RGB image in the spatial domain to resolve the problem of protecting copyright. Since the technique is based on the DC coefficients in the spatial domain, it exhibits simple and quick performance as well as high robustness which is a feature of transform domain.

A robust video watermarking technique has been proposed by Arab et al. [7] for tamper detection of surveillance systems. They determined that spatial domain techniques are better than transform domain techniques and devised two techniques namely VW8F and VW16E in which 8 bits and 16 bits are embedded respectively. They claimed to achieve better imperceptibility and better tamper detecting abilities through their techniques.

Jana et al. [8] proposed a dual image based reversible data hiding scheme. In their technique they divide the secret message into sub stream of n bits and embedded n-1 bits using pixel value differencing and 1 bit is embedded using difference expansion. They successfully recovered the secret message and the original image without any distortion and displayed good performance of their technique in terms of embedding capacity.

3. PROPOSED MODEL

Our proposed technique as stated above is a combination of a modified LSB watermarking and (*n*,*n*) secret image sharing scheme. These two techniques combined eliminate the drawbacks of each other and complement the advantages and hence provide a secure and imperceptible data security. The proposed model is majorly divided into three stages comprising:

- **Embedding** of secret image into the video frame using LSB technique.
- **Share generation** of the watermarked image and its reconstruction using (*n*,*n*) secret image sharing scheme.
- **Extraction** of the hidden image from the received watermarked image.

The basic overall algorithm incorporated here is the steps taken in the following order. Firstly, the video is obtained and its frames are extracted and preprocessed according to certain defined requirements. Similarly, the image to hide is obtained, preprocessed and a watermark is generated according to certain conditions and it is put up for embedding in the video frame using proposed modified LSB algorithm. Next after the frame is watermarked and the image to hide is hidden, the watermarked video frame is obtained and its shares are generated employing (*n*, *n*) secret sharing scheme and the three shares are transmitted through the media. These shares are obtained at the receiver side and the shares are reconstructed to form the watermarked video frame and are put up for the extraction of the hidden image from the watermarked frame. The extraction of the hidden image from the video frame is the last step of the proposed watermarking technique. These steps are explained in detail in the following sections of this paper.

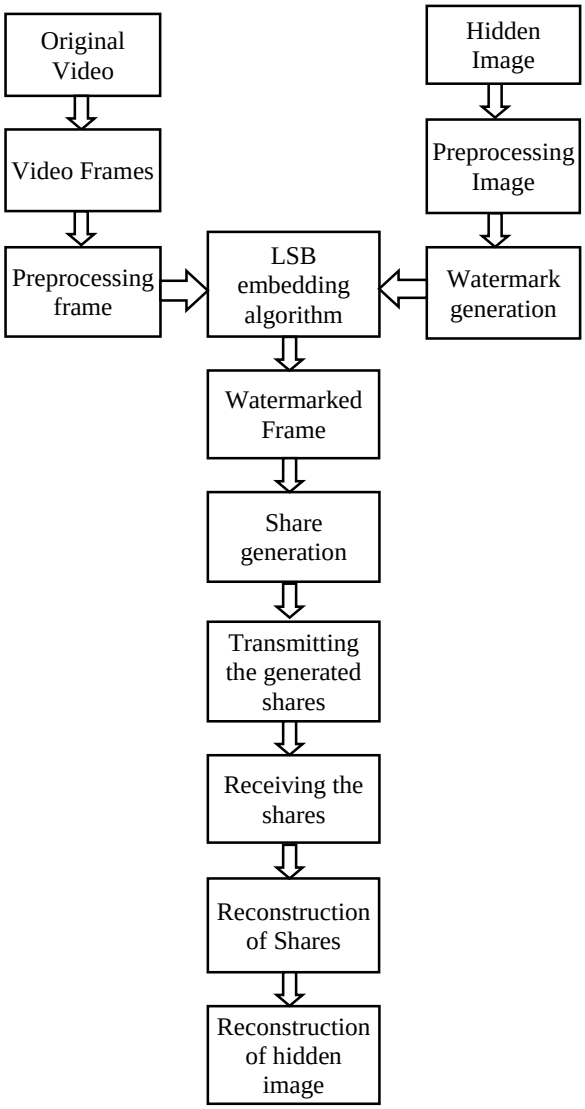


Fig.1. System Model

3.1 EMBEDDING

The embedding stage uses the LSB technique to embed the secret image into the bits of the video frame. The embedding algorithm is as follows:

Step 1: Load the video into the system.

Step 2: Extract the frames of the video to process.

Step 3: Preprocess the video frame which comprises of resizing the frame into the desired size and converting the RGB frames into gray.

Step 4: Load the secret image into the system.

Step 5: Preprocess the image as mentioned in Step 3 above.

Step 6: Generate watermark of the same size as that of the video frame.

Step 7: Embed every bit of the secret image in the desired bit of the frame preferably LSB. This leads to the generation of 8 watermarks as the secret image is an 8 bit image.

After completing all the above steps on the frame with the secret image 8 watermarked frames are obtained. The above embedding displays the results as shown in Fig.2 below.

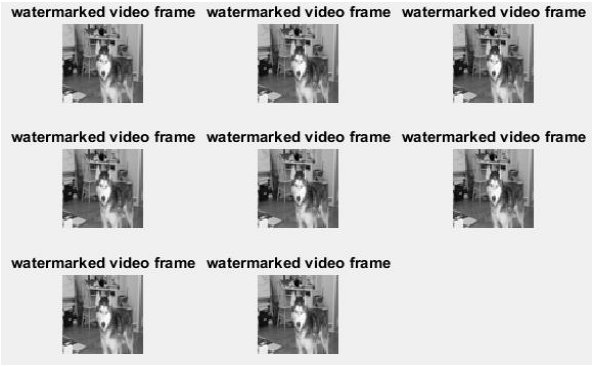


Fig.2. Eight Watermarked frames

3.2 SHARING (GENERATION OR RECONSTRUCTION)

The (*n*,*n*) threshold secret sharing scheme has proved to be the most precise in terms of reconstruction and produces no pixel expansion. It promises equal security as given by the other two schemes as it requires all the generated shares for reconstruction and in turn gives perfect reconstruction thus ensuring highest quality. The (*n*,*n*) scheme is best explained below through an algorithm provided by Dong and Ku [11] which was referred by us in our scheme. The algorithm is as follows:

Input: Image A with size $h \times w$

Output: Shadow image S_i, $i \in \{1,2,\ldots,n\}$

Share construction:

Step 1: Get permuted image PA by using a key to generate a permutation sequence to permute the pixels of A.

Step 2: Generate n-1 random matrices R_1, R_2,…,R_{n-1}, each of which has size $h \times h$ and element be $\{0\ldots\ldots255\}$.

Step 3: Compute $R_n = (I\text{-}R_1\text{-},\ldots,\text{--}R_{n-1}) \bmod 256$, where I is unit matrix with size $h \times h$.

Step 4: Compute $S_i = (R_i * PA) \bmod 256$, where '*' means matrix multiplication.

Revealing:

Step 1: $PA^o = (S_1 + \ldots.. + S_n) \bmod 256$

Step 2: Apply inverse-permutation operation to PA^o to get the reconstructed image A^o.

Example: A(3,3) secret image sharing scheme.

Input: grayscale image $A = \begin{bmatrix} 133 & 167 \\ 134 & 208 \end{bmatrix}$

Share construction:

Step *: $PA = \begin{bmatrix} 134 & 133 \\ 208 & 167 \end{bmatrix}$

Step 1: $R_1 = \begin{bmatrix} 171 & 251 \\ 101 & 254 \end{bmatrix}$, $R_2 = \begin{bmatrix} 172 & 136 \\ 52 & 192 \end{bmatrix}$

Step 2: $R_3 = I\text{-}R_1\text{-}R_2$

$$= \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 171 & 251 \\ 101 & 254 \end{bmatrix} - \begin{bmatrix} 172 & 136 \\ 52 & 192 \end{bmatrix} \right) \bmod 256$$

$$= \begin{bmatrix} 170 & 125 \\ 103 & 67 \end{bmatrix}$$

Step 3: get three shares by computing

$S_1 = (R_1 * PA) \bmod 256$

$$= \left(\begin{bmatrix} 171 & 251 \\ 101 & 254 \end{bmatrix} * \begin{bmatrix} 134 & 133 \\ 208 & 167 \end{bmatrix} \right) \bmod 256$$

$$= \begin{bmatrix} 114 & 148 \\ 60 & 43 \end{bmatrix}$$

$S_2 = (R_2 * PA) \bmod 256$

$$= \left(\begin{bmatrix} 172 & 136 \\ 52 & 192 \end{bmatrix} * \begin{bmatrix} 134 & 133 \\ 208 & 167 \end{bmatrix} \right) \bmod 256$$

$$= \begin{bmatrix} 136 & 20 \\ 56 & 68 \end{bmatrix}$$

$S_3 = (R_3 * PA) \bmod 256$

$$= \left(\begin{bmatrix} 170 & 125 \\ 103 & 67 \end{bmatrix} * \begin{bmatrix} 134 & 133 \\ 208 & 167 \end{bmatrix} \right) \bmod 256$$

$$= \begin{bmatrix} 140 & 221 \\ 90 & 56 \end{bmatrix}$$

Revealing: The reconstructed secret image is

Step 1: $PA^o = (S_1 + S_2 + S_3) \bmod 256$

$$\left(\begin{bmatrix} 114 & 148 \\ 60 & 43 \end{bmatrix} + \begin{bmatrix} 136 & 20 \\ 56 & 68 \end{bmatrix} + \begin{bmatrix} 140 & 221 \\ 90 & 56 \end{bmatrix} \right) \bmod 256$$

$$= \begin{bmatrix} 134 & 133 \\ 208 & 167 \end{bmatrix} = PA$$

Step *: $A^o = \begin{bmatrix} 133 & 167 \\ 134 & 208 \end{bmatrix} = A$

The example appropriately explains the (*n*,*n*) secret sharing scheme and displays that the image is perfectly reconstructed.

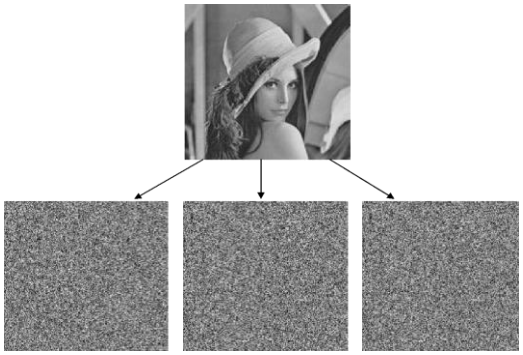


Fig.3. Share generation

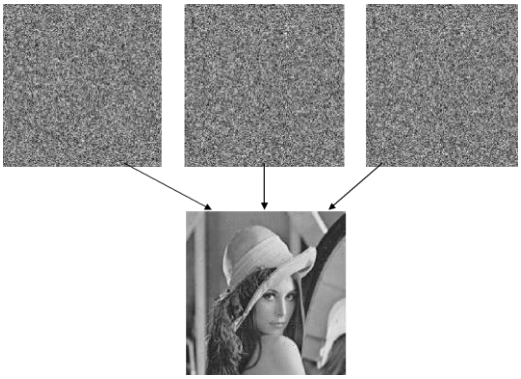


Fig.4. Share reconstruction

The Fig.3 and Fig.4 show the results displayed after shares were created at the transmitter side and reconstructed at the receiver side.

3.3 EXTRACTION

The extraction algorithm basically acquires the bits of the hidden image embedded in the video frame from the watermarked image after obtaining the reconstructed image from the shares. The following steps are taken to extract the hidden image:

Step 1: Obtain the reconstructed watermarked image after share reconstruction algorithm is applied.

Step 2: Extract one bit from each of the 8 watermarked images which have the 1 bit each embedded in their LSBs.

Step 3: Obtain the matrix for each bit and further shift to their bit locations and add all the 8 matrices to obtain a combined matrix for the recovered hidden image.

Step 4: Display the final combined matrix as the recovered hidden image and check the PSNR.

The above stated 4 steps describe the recovery or extraction of the hidden image from the video frame and the extraction result for hidden Lena image have been shown in below figures.

Thus the extraction of the hidden image is complete which culminates the final stage of the proposed technique. This ends this section on the methodology used in the paper and explains the basic concept and idea behind the proposed technique.

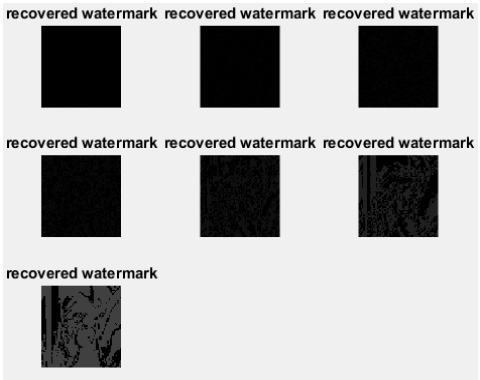


Fig.5. Eight Recovered Watermarks

Fig.6. Reconstructed watermark image

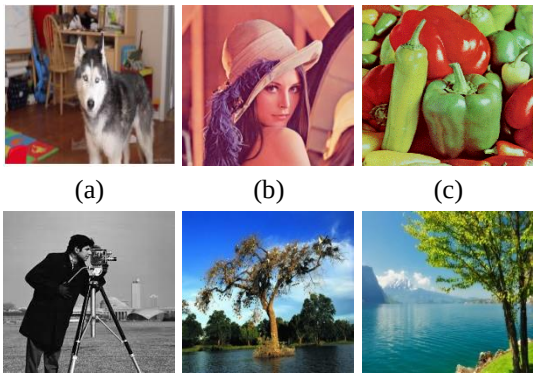
The Fig.5 and Fig.6 above display the eight recovered watermarks after extraction from the video frames and the final reconstructed watermark image which was hidden, respectively.

4. SIMULATION RESULTS

The simulations for the technique were done with considerations of checking the proposed technique with various image formats, various size of watermark images and various video formats. The detailed results are discussed in following subsections along with each being tabulated at the end.

4.1 RESULTS WITH VARIOUS IMAGE FORMATS

JPG Implementation



(a) (b) (c)

(d) (e) (f)

Fig.7. .jpg implementation

The Fig.7 shows the .jpg implementation of our technique where (a) the video 'Video_Dog.mp4' of frame size 1280×720 which was resized to 64×64 (can be varied as desired), (b) the image 'Lena64.jpg' of size 64×64 which was embedded completely inside the cover frame as the secret image and the resized frame were of same size and gave the PSNR of 54.14dB, (c) the image 'pepper.jpg' of size 64×64 which was embedded completely inside the cover frame as the secret image and the resized frame were of same size and gave the PSNR of 55.30dB, (d) the image 'cameraman.jpg' of size 204×204 which was embedded inside the cover frame after resizing the secret image as the secret image was bigger than the resized frame and gave the PSNR of 55.67dB, (e) the image 'Egrets.jpg' of size 467×308 which was embedded inside the cover frame after resizing the secret image as the secret image was larger than the resized frame and gave the PSNR of 59.40dB, (f) the image 'River_tree.jpg' of size 1920×720 which was embedded inside the cover frame after resizing the secret image as the secret image and the resized frame were of different size and gave the PSNR of 60.16dB.

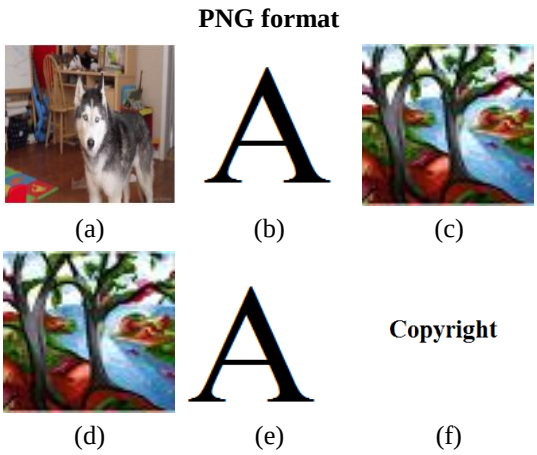


Fig.8. .png implementation

The Fig.8 shows the .png implementation of our technique where (a) the video 'Video_Dog.mp4' of frame size 1280×720 which was resized to 64×64 (can be varied as desired), (b) the image 'A32.png' of size 32×32 which was embedded completely inside after tiling it according to the cover frame as the secret image and the resized frame were of different size which gave the PSNR of 53.07dB, (c) the image 'trees32.png' of size 32×32 which was embedded completely inside after tiling it according to the cover frame as the secret image and the resized frame were of different size which gave the PSNR of 57.34dB, (d) the image 'trees64.png' of size 64×64 which was embedded completely inside the cover frame as the secret image and the resized frame were of same size which gave the PSNR of 57.38dB, (e) the image 'A128.png' of size 128×128 which was embedded completely inside the cover frame after resizing the secret image as the secret image and the resized frame were of different size which gave the PSNR of 52.86dB, (f) the image 'CopyrightII.png' of size 150×50 which was embedded completely inside the cover frame after resizing the secret image as the secret image and the resized frame were of different size which gave the PSNR of 52.96dB.

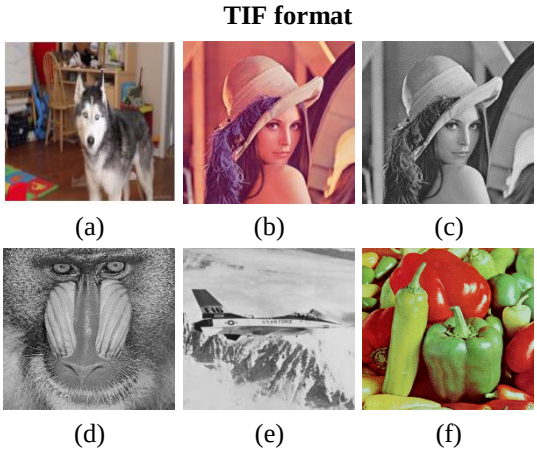


Fig.9: .tif implementation

The Fig.9 displays the .tif implementation where (a) the video 'Video_Dog.mp4' of frame size 1280×720 which was resized to 64×64(can be varied as desired), (b) the image 'lena_color.tif' of size 256×256 which was embedded completely inside the cover frame after resizing the secret image as the secret image and the resized frame were of different size which gave the PSNR of 54.13dB, (c) the image 'lena_gray.tif' of size 256×256 which was embedded completely inside the cover frame after resizing the secret image as the secret image and the resized frame were of different size which gave the PSNR of 56.15dB, (d) the image 'mandril_gray.tif' of size 512×512 which was embedded completely inside the cover frame after resizing the secret image as the secret image and the resized frame were of different size which gave the PSNR of 56.12dB, (e) the image 'jetplane.tif' of size 512×512 which was embedded completely inside the cover frame after resizing the secret image as the secret image and the resized frame were of different size which gave the PSNR of 53.96dB, (f) the image taken is 'pepper_color.tif' of size 512×512 which was embedded completely inside the cover frame after resizing the secret image as the secret image and the resized frame were of different size which gave the PSNR of 56.05dB.

The Table.1 consolidates the above results for various image formats, clearly depicting that the proposed system model is a general model and does not depend on any image format. It gives comparable results in terms of PSNR for all image formats which proves that this technique is applicable to all types of images and shows that the efficiency of reconstruction is retained.

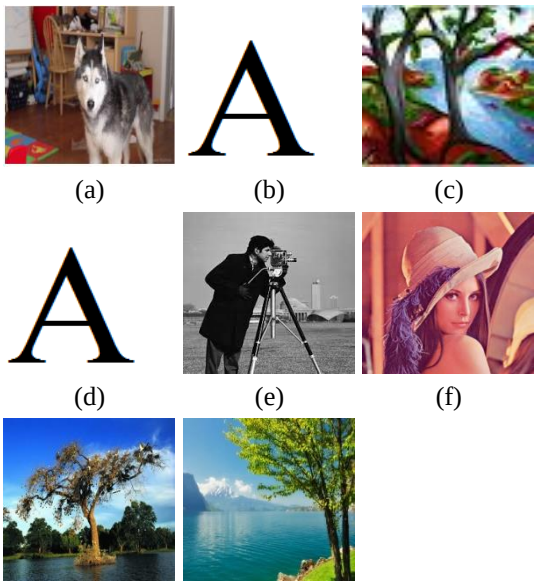
Table.1. Image format results of frame size 1280×720 and 64×64 resized frame size for a .mp4 video format

Cover Video	Hidden Image	Image Format	Image Size	PSNR
Video-Dog	Lena64	.jpg	64×64	54.14
Video-Dog	Pepper	.jpg	64×64	55.30
Video-Dog	Cameraman	.jpg	204×204	55.67
Video-Dog	Egrets	.jpg	467×308	59.40
Video-Dog	River_tree	.jpg	1920×1080	60.16
Video-Dog	A32	.png	32×32	53.07
Video-Dog	Trees32	.png	32×32	57.34
Video-Dog	Trees64	.png	64×64	57.38
Video-Dog	A128	.png	128×128	52.86
Video-Dog	Copyright-tII	.png	150×50	52.96
Video-Dog	Lena-color	.tif	256×256	54.13
Video-Dog	Lena-gray	.tif	256×256	56.15
Video-Dog	Mandril_gray	.tif	512×512	56.10
Video-Dog	Jetplane	.tif	512×512	53.96
Video-Dog	Pepper-color	.tif	512×512	56.05

4.2 RESULTS WITH VARIOUS SIZES

The Fig.10 shows the 32×32 implementation of our technique where (a) the video 'Video_Dog.mp4' of frame size 1280×720 which was resized to 64×64(can be varied as desired), (b) the image 'A32.png' of size 32×32 which was embedded inside after tiling it according to the cover frame as the secret image was smaller than the resized frame and gave the PSNR of 53.07dB, (c) the image 'trees64.png' of size 64×64 which was embedded completely inside the cover frame as the secret image and the resized frame were of same size which gave the PSNR of

57.38dB, (d) thee image 'A128.png' of size 128×128 which was embedded inside the cover frame after resizing the secret image as the secret image and the resized frame were of different size and gave the PSNR of 52.86dB, (e) the image 'cameraman.jpg' of size 204×204 which was embedded inside the cover frame after resizing the secret image as the secret image was larger than the resized frame and gave the PSNR of 55.67dB, (f) the image 'lena_color.tif' of size 256×256 which was embedded inside the cover frame after resizing the secret image as the secret image was larger than the resized frame and gave the PSNR of 54.13dB, (g) the image 'Egrets.jpg' of size 467×308 which was embedded completely inside the cover frame after resizing the secret image as the secret image and the resized frame were of different size and gave the PSNR of 59.40dB, (h) the image 'River_tree.jpg' of size 1920×720 which was embedded inside the cover frame after resizing the secret image as the secret image and the resized frame were of different size and gave the PSNR of 60.16dB.



(a) (b) (c) (d) (e) (f) (g) (h)

Fig.10. Image size implementation

The Table.2 consolidates the above results for various image sizes clearly depicting that the proposed system model is a general model and does not depend on any image size. The results show that the PSNR does not vary much with the size of the secret image, hence proving the robustness and the efficiency of extraction of the technique to be very high.

Table.2. Image size results of frame size 1280×720 and 64×64 resized frame size for a .mp4 video format

Cover video	Hidden image	Image format	Image size	PSNR
Video-Dog	A32	.png	32×32	53.07
Video-Dog	Trees64	.png	64×64	57.38
Video-Dog	A128	.png	128×128	52.86
Video-Dog	Camera-man	.jpg	204×204	55.67
Video-Dog	Lena-color	.tif	256×256	54.13
Video-Dog	Egrets	.jpg	467×308	59.40
Video-Dog	River-tree	.jpg	1920×1080	60.16

4.3 RESULTS WITH VARIOUS VIDEO FORMATS

The Fig.11 shows the .mp4 implementation of our technique where (a) the video 'Video_Dog.mp4' of frame size 1280×720 which was resized to 64×64 (can be varied as desired), (b) the image 'A32.png' of size 32×32 which was embedded completely inside after tiling it according to the cover frame as the secret image was smaller than the resized frame and gave the PSNR of 53.07dB, (c) the image 'trees64.png' of size 64×64 which was embedded completely inside the cover frame as the secret image and the resized frame were of same size and gave the PSNR of 57.38dB, (d) the image 'A128.png' of size 128×128 which was embedded completely inside the cover frame after resizing the secret image as the secret image is larger than the resized frame and gave the PSNR of 52.86dB, (e) the image 'lena_color.tif' of size 256×256 which was embedded inside the cover frame after resizing the secret image as the secret image and the resized frame were of different size and gave the PSNR of 54.13dB, (f) the image 'pepper_color.tif' of size 512×512 which was embedded completely inside the cover frame after resizing the secret image as the secret image is larger than the resized frame and gave the PSNR of 56.05dB.

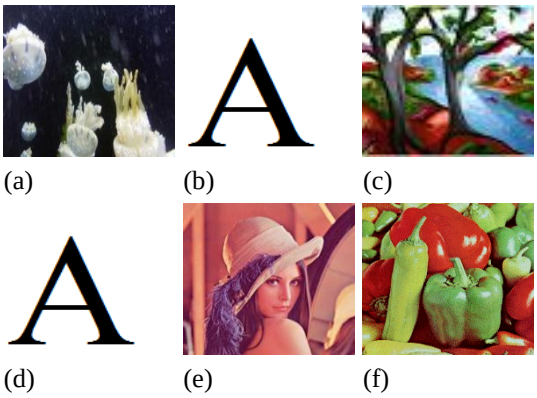
MP4 format

(a) (b) (c) (d) (e) (f)

Fig.11. .mp4 implementation

The Fig.12 shows the .3gp implementation of our technique where (a) the video 'Jellyfish.3gp' of frame size 1280×720 which was resized to 64×64(can be varied as desired), (b) the image 'A32.png' of size 32×32 which was embedded completely inside after tiling it according to the cover frame as the secret image was smaller than the resized frame and gave the PSNR of 53.07dB, (c) the image 'trees64.png' of size 64×64 which was embedded completely inside the cover frame as the secret image and the resized frame were of same size and gave the PSNR of 57.38dB, (d) the image 'A128.png' of size 128×128 which was embedded completely inside the cover frame after resizing the secret image as the secret image is larger than the resized frame and gave the PSNR of 52.86dB, (e) the image 'lena_color.tif' of size 256×256 which was embedded inside the cover frame after resizing the secret image as the secret image and the resized frame were of different size and gave the PSNR of 54.13dB, (f) the image 'pepper_color.tif' of size 512×512 which was embedded completely inside the cover frame after resizing the secret image as the secret image is larger than the resized frame and gave the PSNR of 56.05dB.

3GP format



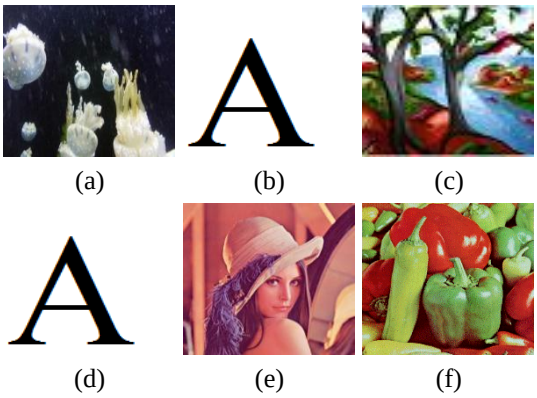
(a) (b) (c)

(d) (e) (f)

Fig.12: .3gp implementation

The Fig.13 shows the .avi implementation of our technique where (a) the video 'Jellyfish.avi' of frame size 1280×720 which was resized to 64×64(can be varied as desired), (b) the image 'A32.png' of size 32×32 which was embedded completely inside after tiling it according to the cover frame as the secret image was smaller than the resized frame and gave the PSNR of 53.07dB, (c) the image 'trees64.png' of size 64×64 which was embedded completely inside the cover frame as the secret image and the resized frame were of same size and gave the PSNR of 57.38dB, (d) the image 'A128.png' of size 128×128 which was embedded completely inside the cover frame after resizing the secret image as the secret image is larger than the resized frame and gave the PSNR of 52.86dB, (e) the image 'lena_color.tif' of size 256×256 which was embedded inside the cover frame after resizing the secret image as the secret image and the resized frame were of different size and gave the PSNR of 54.13dB, (f) the image 'pepper_color.tif' of size 512×512 which was embedded completely inside the cover frame after resizing the secret image as the secret image is larger than the resized frame and gave the PSNR of 56.05dB.

AVI format



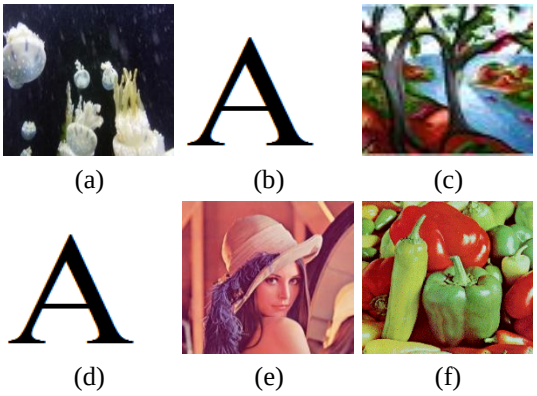
(a) (b) (c)

(d) (e) (f)

Fig.13: .avi implementation

The Fig.14 shows the .wmv implementation of our technique where (a) the video 'Jellyfish.wmv' of frame size 1280×720 which was resized to 64×64 (can be varied as desired), (b) the image 'A32.png' of size 32×32 which was embedded completely inside after tiling it according to the cover frame as the secret image was smaller than the resized frame and gave the PSNR of 53.07dB, (c) the image 'trees64.png' of size 64×64 which was embedded completely inside the cover frame as the secret image and the resized frame were of same size and gave the PSNR of 57.38dB, (d) the image 'A128.png' of size 128×128 which was embedded completely inside the cover frame after resizing the secret image as the secret image is larger than the resized frame and gave the PSNR of 52.86dB, (e) the image 'lena_color.tif' of size 256×256 which was embedded inside the cover frame after resizing the secret image as the secret image and the resized frame were of different size and gave the PSNR of 54.13dB, (f) the image 'pepper_color.tif' of size 512×512 which was embedded completely inside the cover frame after resizing the secret image as the secret image is larger than the resized frame and gave the PSNR of 56.05dB.

WMV format

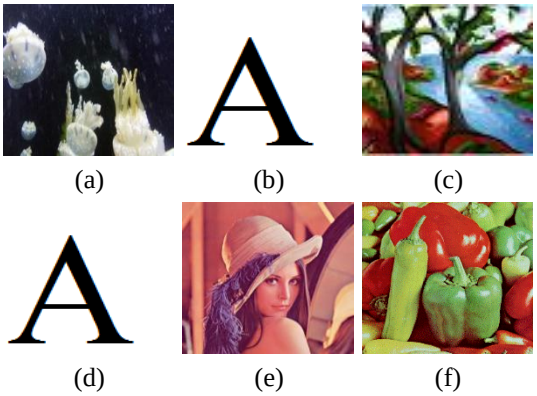


(a) (b) (c)

(d) (e) (f)

Fig.14. .wmv Implementation

MOV format



(a) (b) (c)

(d) (e) (f)

Fig.15. .mov Implementation

The Fig.15 shows the .mov implementation of our technique where (a) the video 'Jellyfish.mov' of frame size 1280×720 which was resized to 64×64(can be varied as desired), (b) the image 'A32.png' of size 32×32 which was embedded completely inside after tiling it according to the cover frame as the secret image was smaller than the resized frame and gave the PSNR of 53.07dB, (c) the image 'trees64.png' of size 64×64 which was embedded completely inside the cover frame as the secret image and the resized frame were of same size and gave the PSNR of 57.38dB, (d) the image 'A128.png' of size 128×128 which was embedded completely inside the cover frame after resizing the secret image as the secret image is larger than the resized frame and gave the PSNR of 52.86dB, (e) the image 'lena_color.tif' of size 256×256 which was embedded inside the cover frame after

resizing the secret image as the secret image and the resized frame were of different size and gave the PSNR of 54.13dB, (f) the image 'pepper_color.tif' of size 512×512 which was embedded completely inside the cover frame after resizing the secret image as the secret image is larger than the resized frame and gave the PSNR of 56.05dB.

The results for various video formats have been tabulated in the Table.3 which clearly depicts that the proposed system model is a general model and does not depend on any particular video format for desirable results. The proposed model gives comparable results which prove that this technique is applicable to most types of video formats and shows that if the video frames can be extracted, embedding is efficient which in turn makes the quality of reconstruction efficient.

Table.3. Video formats results of frame size 1280×720 and 64×64 resized frame size

Cover video	Video format	Hidden image	Image format	Image size	PSNR
Video_dog	.mp4	A32	.png	32×32	53.07
Video_dog	.mp4	Trees64	.png	64×64	57.38
Video_dog	.mp4	A128	.png	128×128	52.86
Video_dog	.mp4	Lena	.tif	256×256	54.13
Video_dog	.mp4	Pepper	.tif	512×512	56.05
Jellyfish	.3gp	A32	.png	32×32	53.07
Jellyfish	.3gp	Trees64	.png	64×64	57.38
Jellyfish	.3gp	A128	.png	128×128	52.86
Jellyfish	.3gp	Lena	.tif	256×256	54.13
Jellyfish	.3gp	Pepper	.tif	512×512	56.05
Jellyfish	.avi	A32	.png	32×32	53.07
Jellyfish	.avi	Trees64	.png	64×64	57.38
Jellyfish	.avi	A128	.png	128×128	52.86
Jellyfish	.avi	Lena	.tif	256×256	54.13
Jellyfish	.avi	Pepper	.tif	512×512	56.05
Jellyfish	.wmv	A32	.png	32×32	53.07
Jellyfish	.wmv	Trees64	.png	64×64	57.38
Jellyfish	.wmv	A128	.png	128×128	52.86
Jellyfish	.wmv	Lena	.tif	256×256	54.13
Jellyfish	.wmv	Pepper	.tif	512×512	56.05
Jellyfish	.mov	A32	.png	32×32	53.07
Jellyfish	.mov	Trees64	.png	64×64	57.38
Jellyfish	.mov	A128	.png	128×128	52.86
Jellyfish	.mov	Lena	.tif	256×256	54.13
Jellyfish	.mov	Pepper	.tif	512×512	56.05

5. COMPARISON WITH EXISTING TECHNIQUES

The success of a technique is achieved only if the proposed technique gives better results than other existing techniques using similar domain or similar techniques. In our case we have used the Peak signal to noise ratio (PSNR) and Embedding capacity as the evaluating parameter which will determine the success of the research work conducted.

Peak signal to noise ratio (PSNR) is one of the most commonly used parameter for the assessment of visual quality of watermarking system. Higher the value of PSNR better the quality of reconstruction of watermark while less PSNR shows the watermark is more perceptible. The extracted image and original image are same when the PSNR is infinity. To calculate the value of the PSNR, Mean square error (MSE) is calculated between the original watermark and the extracted watermark as follows:

The PSNR is defined as,

$$PSNR = 10\log\left(\frac{MAX^2}{MSE}\right) \quad (1)$$

Since the proposed technique has been implemented in the spatial domain and image sharing has also been incorporated to ensure higher security, it has been compared with three other techniques implemented in the same domain and have conducted similar research. The technique proposed in this report has been compared with the techniques proposed by Li Liu et al. [1], Wang et al. [4] and Chang et al. [12] for PSNR and with Wang et al. [4], Su et al. [6] and B. Jana et al. [8] for the embedding capacity. Their comparison has been represented below with analysis.

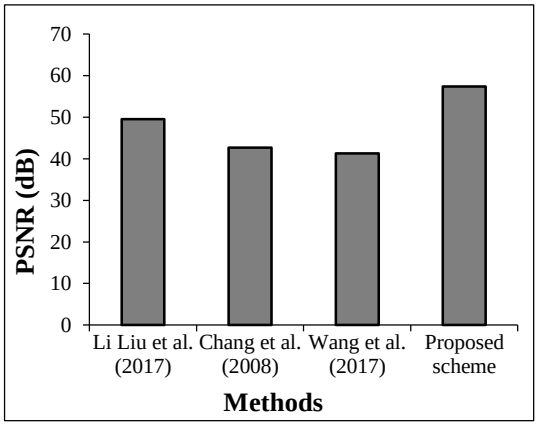


Fig.16. Comparison of PSNR

The Fig.16 shows the comparison of PSNR values for different existing techniques with the proposed method. The PSNR calculated by Li Liu et al. is 49.51dB [1], Chang et al. is 42.70dB [18] and by Wang et al. is 41.30dB [4]. Our proposed technique gives a PSNR of 57.39dB.

Embedding capacity describes how many information bits of the secret message can be embedded in the cover frame. Higher embedding capacity is usually obtained at the expense of either robustness strength or imperceptibility or both. The proposed technique has been compared in terms of embedding capacity with Wang et al. [4], Su et al. [6] and Jana et al. [8] for one 256×256 and one 512×512 image as the proposed technique's

embedding capacity depends on the size of the hidden image and the comparative results are shown in Fig.17 and Fig.18.

The Fig.17 and Fig.18 clearly depict that the embedding capacity provided by the proposed technique is much higher than the other three referred techniques. Wang et al. provided 316399 bits embedded in a 256×256 image and 632797 bits in a 512×512 image, B. Jana et al. offered 163592 bits in a 256×256 image and 327184 bits in a 512×512 image, Su et al. gave 262144 bits in a 256×256 image and 1048576 bits in a 512×512 image and the proposed scheme offers 458752 bits in a 256×256 image and 1835008 bits in a 512×512 image.

This proves the success of our proposed technique and proves that the technique is more robust and secure than other technique breaks the myth that the embedding capacity needs to be low to keep the robustness of the technique high. The image sharing concept provides better authentication capability and thus ensures security.

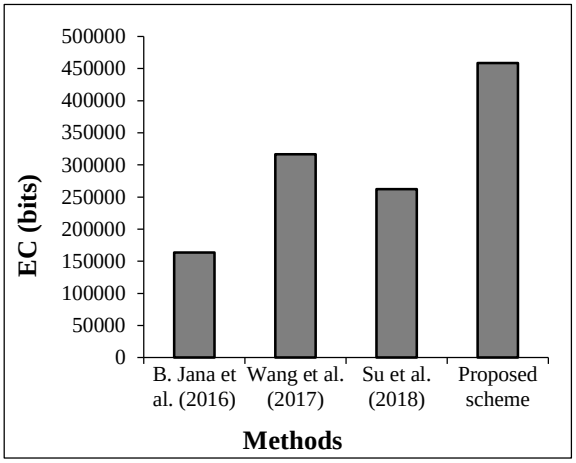


Fig.17. Comparison of embedding capacity (EC) (256×256)

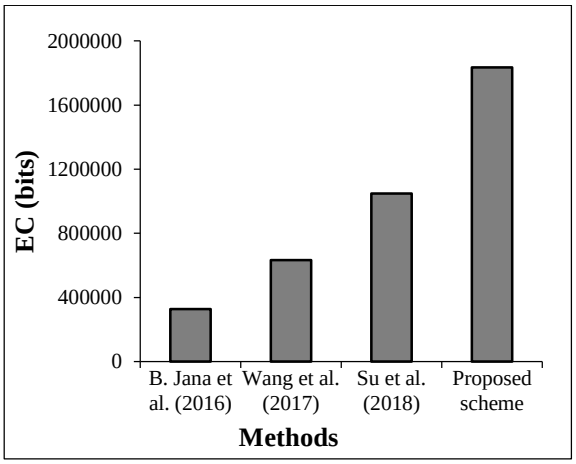


Fig.18. Comparison of embedding capacity (EC) (512×512)

6. CONCLUSIONS

The modified LSB technique employed in this research provided for high embedding capacity as all the 8 bits were embedded inside the MSB of the cover frame and also made it extremely easy to extract all the bits to get the best possible extracted quality image. The disadvantages of LSB technique like having less robustness and high perceptibility making it prone to attacks have been eliminated by employing secret image sharing scheme. This technique provided secure transmission in the form of meaningless shares which made it less prone to attacks and the (*n*,*n*) sharing scheme ensured perfect reconstruction of the watermarked image which ultimately enhanced the quality of extraction of the hidden image. All the results clearly determine the robustness and efficient extraction of the proposed technique. The results show that the technique can deal with most of the variations possible till now and provides stable results. However there is a threshold of employing all the shares generated at the time of reconstruction as perfect reconstruction of the watermarked video frame is of utmost importance for extracting the hidden image efficiently. If any numbers of shares less than the number of shares generated are employed, perfect reconstruction will not be possible which in turn will affect the extraction of the embedded bits thus degrading the value of PSNR.

REFERENCES

[1] Li Liu, Anhong Wang, Chin-Chen Chang and Zhihong Li, "A Secret Image Sharing with Deep-Steganography and two-Stage Authentication based on Matrix Encoding", *International Journal of Network Security*, Vol. 19, No. 3, pp. 327-334, 2017.

[2] K.N. Sowmya and H.R. Chennamma, "Video authentication using Watermark and Digital Signature-A Study", *Proceedings of 1st International Conference on Computational Intelligence and Informatics*, pp. 53-64, 2017.

[3] Ranjeet Kumar Singh, Dilip Kumar Shaw and M. Javed Alam, "Experimental Studies of LSB Watermarking with Different Noise", *Proceedings of 11th International Multi-Conference on Information Processing*, Vol. 54, pp. 612-620, 2015.

[4] Yu Lun Wang, Jau-Ji Shen and Min-Shiang Hwang, "An Improved Dual Image based Reversible Hiding Technique using LSB Matching", *International Journal of Network Security*, Vol. 19, No. 5, pp. 858-862, 2017.

[5] Kaiser J. Giri and Rumaan Bashir, "Digital Watermarking: A Potential Solution for Multimedia Authentication", *Proceedings of International Conference on Intelligent Techniques in Signal Processing for Multimedia Security*, Vol. 660, pp. 93-112, 2017.

[6] Qingtang Su and Beijing Chen, "Robust Color Image Watermarking Technique in the Spatial Domain", *International Journal on Soft Computing, A Fusion of Foundation, Methodology and Application*, Vol. 22, No. 1, pp. 91-106, 2018.

[7] Farnaz Arab, Shahidan M. Abdullah, Mazdak Zamani, Siti Zaiton Mohd Hashim, Azizah Abdul Manaf and Mazdak Zamani, "A Robust Video Watermarking Technique for the Tamper Detection of Surveillance Systems", *Multimedia Tools and Applications*, Vol. 75, No. 18, pp. 10855-10885, 2016.

[8] Biswapati Jana, Debasis Giri and Shyamal Kumar Mondal, "Dual-Image based Reversible Data Hiding Scheme using

Pixel Value Difference Expansion", *International Journal of Network Security*, Vol. 18, No. 4, pp. 633-643, 2016.

[9] Adi Shamir, "How to Share a Secret", *Communications of the ACM*, Vol. 22, No. 11, pp. 612-613, 1979.

[10] G.R. Blakley, "Safeguarding Cryptographic Keys", *Proceedings of International Conference on Computer History*, pp. 313-317, 1979.

[11] Lin Dong and Min Ku, "Novel (n, n) Secret Image Sharing Scheme based on Addition", *Proceedings of 6th International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 583-586, 2010.

[12] Chin-Chen Chang, Yi-Pei Hsieh and Chia-Hsuan Lin, "Sharing Secrets in Stego Images with Authentication", *Pattern Recognition*, Vol. 41, pp. 3130-3137, 2008.