Data hiding in the optimal keyframes using circular shifting and mutation operations for improvement in imperceptibility

Sahil Gupta*

Department of Electronics and Communication Engineering,
Maharaja Ranjit Singh Punjab Technical University,
Bathinda, Punjab, India
Email: sahil.gupta311@gmail.com
*Corresponding author

Naresh Kumar Garg

Department of Computer Science and Engineering,
Maharaja Ranjit Singh Punjab Technical University,
Bathinda, Punjab, India
Email: naresh2834@rediffmail.com

Abstract: Video steganography hides the information in the cover video and helps in a secure communication over the unsecured network. The existing steganography methods are lacking in terms of efficient keyframe selection algorithm and also show less imperceptibility. The proposed method used the Kullback Leibler divergence (KLD) and edge features to extract the keyframes from any type of video dataset whereas circular shifting and mutation operations are used to improve the imperceptibility. The circular shifting operation helps in searching for the optimal direction of secret data, whereas the mutation operation was deployed to adjust the pixel values to reduce the variability more. To validate the proposed algorithm, the computations were performed on the standard videos and image datasets. The computational results validate that the proposed algorithm has achieved better imperceptibility in a fewer number of iterations. Furthermore, the keyframes extraction technique helps to increase the security of the proposed system for any type of video dataset.

Keywords: imperceptibility; video steganography; mean square error; MSE; mutation; optimal data hiding; circular shifting; peak signal to noise ratio; PSNR; Kullback Leibler divergence; KLD; edge.

Reference to this paper should be made as follows: Gupta, S. and Garg, N.K. (2023) 'Data hiding in the optimal keyframes using circular shifting and mutation operations for improvement in imperceptibility', *Int. J. Information and Computer Security*, Vol. 20, Nos. 1/2, pp.158–175.

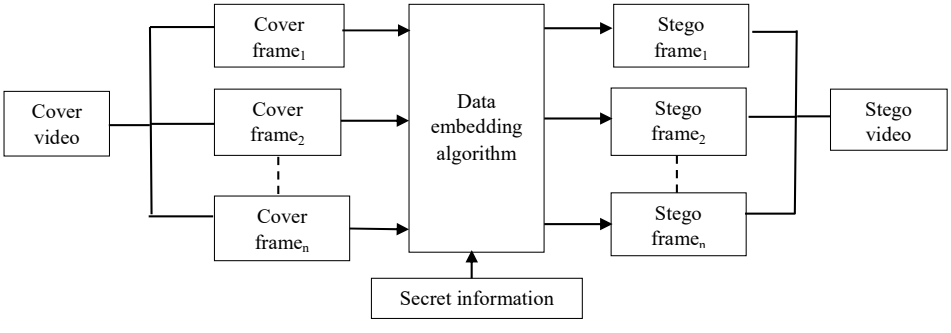
Biographical notes: Sahil Gupta received his BTech in Electronics and Communication Engineering in 2011, MTech degree in 2015. Presently, he is perusing PhD in Video Summarisation and Steganography under the supervision of Professor Naresh Kumar Garg from the Maharaja Ranjit Singh Punjab Technical University, Bathinda, Punjab, India. His research interests include image and video processing, data hiding, video summarisation, optimisation techniques.

Naresh Kumar Garg received his BTech in Computer Science and Engineering in 1997, MTech degree in 2007 and PhD degree in 2014 from the Punjabi University, Patiala, Punjab, India. His research interests include optical character recognition, image processing and pattern recognition. He has more than 40 publications in international journals. He is working as a Professor in GZSCCET, MRSPTU Bathinda.

1 Introduction

Due to advancements in digital technology, the sharing of multimedia content over unsecured networks is massively increased. However, it causes a serious concern of unauthorised access to information if any security mechanism is not adopted (Bai et al., 2017). To resolve this problem, steganography is widely used in the literature. It is a method of communicating the information inside some cover media without giving intuition to anyone (Johnson and Mason, 1998). It gains popularity for communicating information securely in many applications, i.e., banking, medical, military, etc. To increase the hiding capacity, video is preferred as the cover media over the text, image, and audio (Sadek et al., 2015). The block diagram of the basic steganography process is shown in Figure 1. In the beginning, the cover video is read and its frames were extracted. Then the secret information is embedded in the cover frames using the data embedding algorithm and stego frames are produced at the output. In the last, the stego frames are concatenated to make the stego video.

Figure 1 Block diagram of video steganography



The spatial and transform domain techniques are used to hide the secret information in the cover frames (Liu et al., 2018). In spatial domain techniques, the secret information is directly embedded in the cover pixels (Ramalingam and Isa, 2016), whereas in the transform domain, the information is embedded in the transformed frequency coefficients. However, data embedding is not possible in all frequency coefficients. Thus, embedding capacity is small while using the transform domain. In both domains, the least significant bit (LSB) is the most preferential data embedding technique. In the LSB technique, each cover frame pixel can hide only one-bit data. Therefore, the cover frame can hide a maximum of one-eighth of the data. To improve the embedding capacity, the k-bits LSB technique is used in which k-bits of the cover frame pixel are replaced with

information bits (Mstafa and Elleithy, 2016). However, it increases the variability in the same proportion and negatively impacts the visual quality of the cover frames. Imperceptibility is the most important parameter to be considered in steganography. It indicates the similarity between the original cover frame and the frame obtained after data hiding. More is the imperceptibility, lesser is the distortion in the stego frame. Some of the researchers have used various conventional techniques, i.e., LSB, flipping technique, pixel value differencing, etc. whereas others have applied the optimisation algorithms, i.e., genetic algorithm (GA), particle swarm optimisation (PSO), and grasshopper optimisation (GO) to improve the imperceptibility. However, their results are not optimal due to lesser imperceptibility and easy detection of secret data bits from the stego frame. In addition, metaheuristic algorithms are also taking a large number of iterations to reduce variability which overheads the time complexity of the system. As a result, there is a need to design an algorithm that can overcome these research challenges. The key contribution of the proposed work is as follows:

- To increase security, cover video is pre-processed to extract the keyframes. The keyframes were extracted based upon Kullback Leibler divergence (KLD) and edge features of frames. The extracted keyframes show high divergence among the other frames. The data hiding in these keyframes reflects the minimum changes which makes it difficult for attackers to identify the frames where data hiding is done.
- To reduce the variability, circular shifting and mutation operations are used. The circular shifting operation helps to find the starting position of secret data whereas, the mutation operation helps to lessen the variability more between the original cover and output stego frame. The computation results confirm the superiority of the proposed method in terms of imperceptibility when compared with other state of art methods.
- The GA, PSO, and GO algorithms are taking near or greater than 100 iterations to get the optimal results, but the proposed method has used 50 iterations of circular shifting operation and only one iteration of mutation operation to get the optimal results, which shows that proposed algorithm is taking less number of iterations to get the optimal result when compared with others techniques.

The rest of the paper is as follows. Section 2 describes the literature work-related to steganography. The preliminaries required for the proposed algorithm is discussed in Section 3. Section 4 illustrates the methodology of the proposed algorithm. The performance metrics and computational results are given in Section 5. In the last, the conclusion and future work is drawn in Section 6.

2 Review of literature

This section presents the review of various steganography algorithms which are using conventional and optimisation techniques for improvement in imperceptibility.

Mstafa and Elleithy (2016) proposed Kanade Lucas Tomasi (KLT) based video steganography. Here, the facial region is extracted from the video using the KLT algorithm and then data is hidden at the facial region using the K-bit LSB method. However, their technique offers less data hiding capacity due to the presence of smaller facial regions in the video. Paul et al. (2013) detected the keyframe based upon the scene

change detection technique. Here, the histogram difference method is used to detect the scene change in the video. The data is hidden using the 3-3-2 bit approach. The payload capacity of this algorithm is less due to the presence of fewer abrupt scene change frames. Sahu et al. (2018) used the flipping technique to reduce the variability in the stego media. This technique provides a little bit lesser variability as compared to the LSB technique; however, it is required to communicate the reference location map information with the receiver to decrypt the secret data. Almazaydeh (2020) proposed edge-based image steganography. The canny edge operator is used to find the edges of the image and then secret data is hidden at the 4th LSB position. However, data hiding at the 4th LSB position effects the pixels value most and distort the histograms. Another edge-based data hiding technique is presented by Prasad and Pal (2020). To increase the imperceptibility, data hiding is done using the modulus function, however, the results are not satisfactory as the value of PSNR is small, i.e., 36 dB. Shah and Bichkar (2018) used GA to find the appropriate pixel to hide the data; however, their algorithm shows small PSNR value and large distortion in the histogram of stego frame. Hemanth et al. (2016) applied the GA and PSO algorithm for the optimal selection of ridgelet-based transform coefficients. The experimental result confirms that the PSNR obtained by the PSO algorithm is better than the GA approach, where both algorithms are taking 100 iterations. Zhu et al. (2021) presented singular value decomposition (SVD) and integer wavelet transform (IWT) based watermarking scheme. Initially, the cover image is decomposed using the IWT, and SVD, and then GA is used to find the optimal value of the weight factor. Zear and Singh (2021) presented LWT-DCT based image steganography. Here, 3rd level DWT is used to extract the different subbands of the cover image and then DCT and SVD are applied to the LH bands. The secret data is hidden in the LH bands by taking any random value of the scaling factor which fails to trade-off between imperceptibility and robustness. Sharma et al. (2021) used the GO algorithm for optimal selection of scaling factor. Initially, discrete wavelet transform (DWT) and discrete cosine transform (DCT) are used to transform the cover image into the frequency domain, and then GO is used to find the optimal scaling factors. The computational results show that the PSNR value is in the acceptable range but has not improved significantly.

In spite of a lot of work has been done by researchers in the field of steganography using various conventional and optimisation algorithms, still their algorithms lack to achieve higher imperceptibility without affecting the computational complexity. The proposed work presents an optimal data hiding approach to improve the imperceptibility in a fewer number of iterations.

3 Preliminaries

The proposed work is based upon keyframe extraction using KLD and edge features of frames, whereas optimal data hiding is achieved by using circular shifting and mutation operations. Their basic concept and implementation in the steganography field is explained below:

3.1 Features used for keyframe extraction

- *KLD:* it measures the divergence between histograms of two frames. It is determined using equation (1).

$$d_{KL}(p, q) = \sum_i p_i \log \frac{p_i}{q_i} \tag{1}$$

where d_{KL} denotes the KL divergence, i subscripted variable, and p, q denotes the normalised distribution of the histogram.

- *Edges:* an edge defines the area with a significant local change or discontinuity in the image intensity. There are many edge detector operators are available in the literature, i.e., canny, Sobel, robert, and prewitt edge operator. Among them, the canny edge operator is widely used due to higher accuracy, higher precision, and a single response to an edge (Bai et al., 2017).

3.1.1 Implementation of keyframe extraction algorithm

The proposed keyframe extraction algorithm is a two-stage process. In the first stage, the candidate's frames are determined based on the KLD, whereas in the second stage, the candidate frames are processed based on the edge features to determine the keyframes. The flowchart of the keyframe extraction algorithm is shown in Figure 2 and its detailed description is given as:

Step 1 Read the cover video and extracted the frames.

Step 2 Compare the consecutive frames based on the KLD parameter.

Step 3 Find the threshold value (Th_1) using equation (2).

$$Th_1 = \propto \overline{KLD} \tag{2}$$

where $\propto$ is the threshold tuning parameter and its value is chosen 1.5 based on the experimental purposes to work under all scenarios. $\overline{KLD}$ is the mean of the KLD.

Step 4 Extract the candidate frames (C_F) by comparing the KLD feature value with the Th_1. The frames which show a higher KLD value than the Th_1 are selected as the candidate frames.

$$C_F = F_i \quad \text{If} \quad KLD_i > Th_1 \text{ for } 1 \le i \le n-1 \tag{3}$$

where F are the cover video frames, i is the subscripted variable and n is the total frames in the cover video.

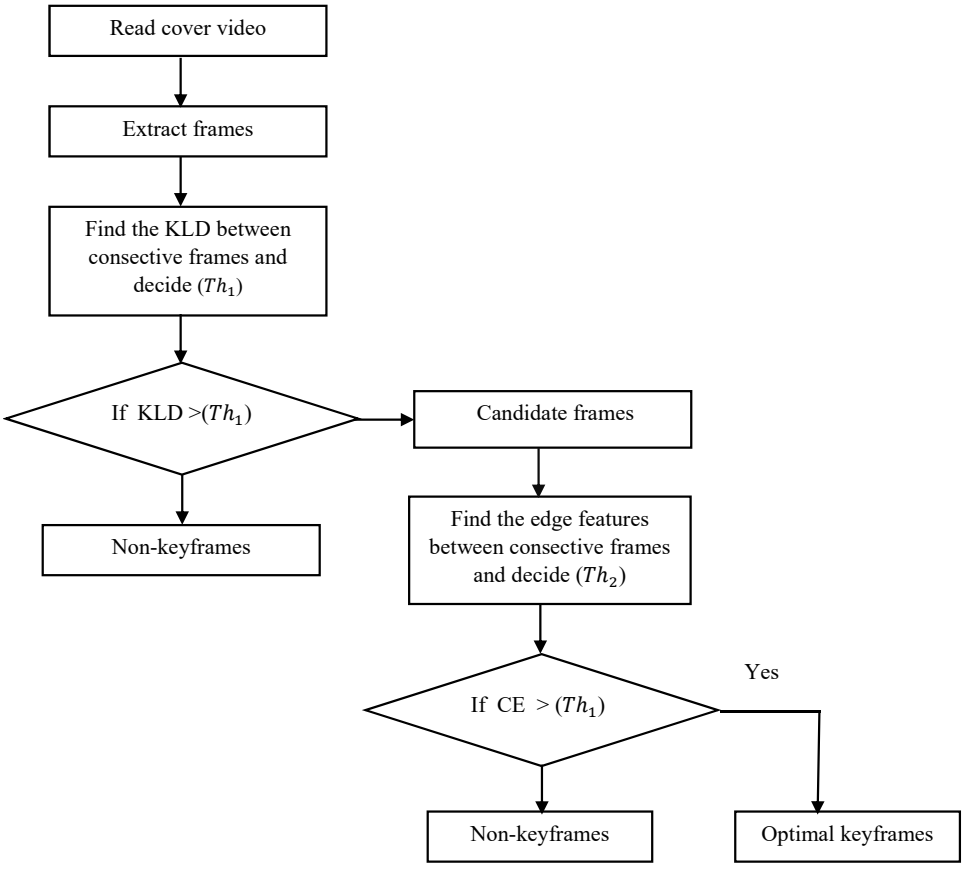
Step 5 Apply the canny edge detection algorithms on the candidate frame (C_F) and find the edges.

Step 6 Find the threshold value (Th_2) using equation (4).

$$Th_2 = \propto \overline{CE} \tag{4}$$

where $\propto$ is the tuning parameter of the threshold and its value chosen 1 based on the experimental purposes to work under all scenarios, $\overline{CE}$ is the mean of the canny edge parameter.

Figure 2 Flowchart of keyframe extraction algorithm



Step 7 Extract the optimal keyframes by comparing the edge feature value with the threshold value (Th_2) using equation (5). The frames with higher edges than the Th_2 are selected as the optimal keyframes.

$$O_{KL} = C_{F(i)} \quad \text{If } CE_{(i)} > Th_2 \text{ for } 1 \le i \le n-1 \tag{5}$$

where O_{KL} are optimal keyframes, i is the subscripted variable, n is the total frames in the cover video.

Thus the proposed algorithm uses the KLD and canny edge operator to extract the optimal keyframes of video.

3.2 Operations used for optimal data hiding

The overview of circular shifting and mutation operations is presented here:

- Circular shifting operation

  The circular shifting operation shifts the entire data in circular order. The circular shifting by +1 shifts the elements of the matrix down by 1, whereas circular shifting by –1 shifts the elements of the matrix up by 1. Figure 3 shows the original data matrix and Figures 4 and 5 show the down shifted data matrices by +1 and +2.

Figure 3 Original data matrix

2	3	4	6
1	2	3	4
5	6	7	8
4	5	6	7

Figure 4 Circular shifting by +1

4	5	6	7
2	3	4	6
1	2	3	4
5	6	7	8

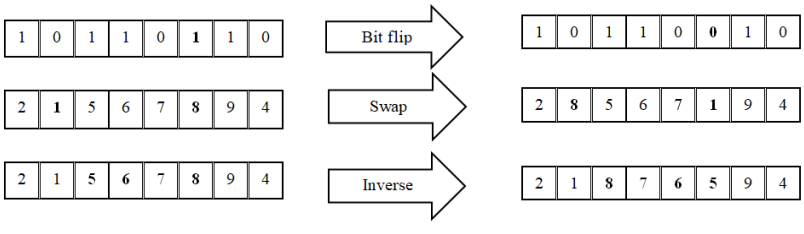
Figure 5 Circular shifting by +2

5	6	7	8
4	5	6	7
2	3	4	6
1	2	3	4

- Mutation operation

  The mutation operation generates a new population by changing some part of the population. It tries to maintain diversity in the solution and avoid the solution to stuck in the local optima. Bit flip, swap, and inverse are some of the most commonly used mutation operators (Katoch et al., 2021). Figure 6 shows the input and output of different mutation operators.

Figure 6 Output of different mutations operators



In the bit flip mutation, the selected data bit is flipped from 1 to 0 and vice versa. The swap mutation swap the content of two selected bits as shown in Figure 6, i.e., data at 3rd position is swapped with 7th position data, whereas in the inverse mutation, the order of writing the data is reversed.

3.2.1 Deployment of circular shifting and mutation operation for optimal data hiding

The following steps illustrate the use of circular shifting and mutation operations for optimal data hiding.

Step 1 Read the secret data and circulate shift it by random numbers.

Step 2 Hide the shifted secret data matrices in the keyframe (K_f) using the 2-bit LSB technique.

Step 3 Find the candidate stego frame (S_{cf}) which shows the least value of mean square error (MSE).

Step 4 Apply mutation operation at the 3rd bit position of the candidate stego frame.

Step 5 Find the optimal stego frame by computing the absolute difference between the keyframe (K_f), candidate stego frame (S_{cf}) and mutated stego frame (S_{mf}).

4 Proposed methodology

The proposed algorithm is divided into three stages. The first stage carried out reading of cover video and secret information and extraction of optimal keyframes. In the second and third stages, circular shifting and mutation operations are applied to reduce the variability and optimal stego frames are produced. At the receiver side, the data extraction algorithm is used to extract the secret information.

4.1 Data embedding

The flowchart of the proposed data embedding algorithm is shown in Figure 7 and its detailed description is given below:

Step 1 Read the cover video and extract the frames.

Step 2 Extract the keyframes (K_f) considering the KLD and edge features of frames as explained in Section 3.1.1.

Step 3 Select the blue plane for data hiding purposes based upon the human visual system (HVS) characteristics.

$$keyframe\_blue = K_f(:,:,3) \tag{6}$$

Step 4 Read the secret image (I) and divide it into 2 bits chunks. For this, use variable (a = 3, b12, c = 48 and 192) and apply AND operation as shown in equation (7).

$$\begin{cases} part\ 1(i,\ j) = bitand(I(i,\ j),\ a(1,1)) \\ part\ 2(i,\ j) = bitand(I(i,\ j),\ b(1,1)) \\ part\ 3(i,\ j) = bitand\ (I(i,\ j),\ c(1,1)) \\ part\ 4(i,\ j) = bitand\ (I(i,\ j),\ d(1,1)) \end{cases} \tag{7}$$

where *part* (1) shows the secret data bits at the first two LSB positions. Similarly *part* (2, 3 and 4) shows the secret data bits at 3rd–4th, 5th–6th, and 7th–8th positions.

Step 5 Apply bit shift operation to move the secret data bits at the first two LSB positions as shown in equation (8). The bit shifting by (–2) shift the data bits to the right side by two times. Similarly bit shifting by (–4 and –6) shift the data bits to the right side by four and six times.

$$\begin{cases} part\ 2(i,\ j) = bitshift\ (part\ 2(i,\ j), -2) \\ part\ 3(i,\ j) = bitshift\ (part\ 3(i,\ j), -4) \\ part\ 4(i,\ j) = bitshift\ (part\ 4(i,\ j), -6) \end{cases} \tag{8}$$

Step 6 Apply 50 times circular shift operations on the secret data bits and generate the 50 different secret data matrices.

Step 7 Hide the secret data matrices in the blue plane of the selected keyframe using the 2 bit LSB replacement technique. The logical operations are used for the LSB data hiding. After data hiding, the stego frames (S_f) are produced at the output.

$$\begin{cases} keyframe_{blue(i,\ j)} = bitand\left(keyframe_{blue(i,\ j)}, 252\right) \\ S_f(i,\ j) = bitor\left(keyframe_{blue(i,\ j)}, part\ 1(i,\ j), part\ 2(i,\ j), part\ 3(i,\ j), part\ 4(i,\ j)\right) \end{cases} \tag{9}$$

Step 8 Find the candidate stego frame (S_{cf}) by computing the MSE value between the (K_f) and (S_f). The stego frame with the least value of MSE is selected as the candidate stego frame.

$$S_{cf} = minimum\ of\ MSE\left(K_f, S_f\right) \tag{10}$$

Step 9 To reduce variability, apply the mutation operation at the 3rd-bit position of the (S_{cf}).

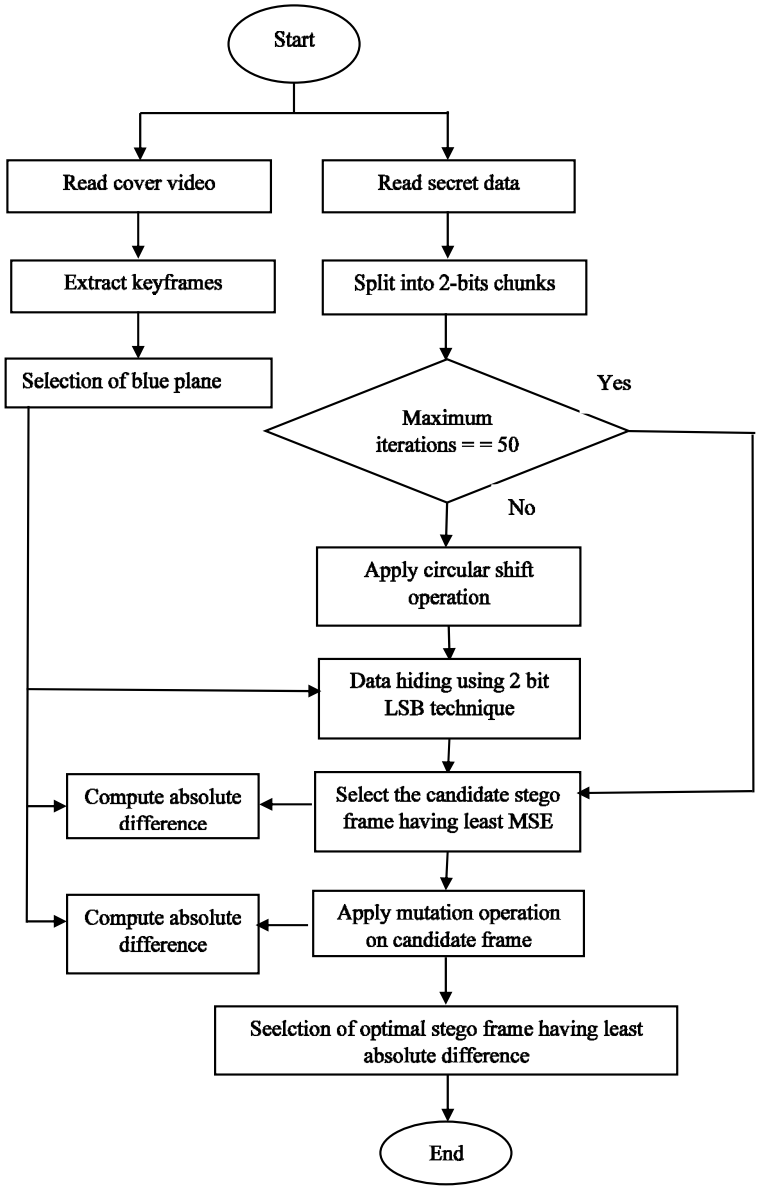
Step 10 Compute the absolute difference (d_1, d_2) between the (K_f), (S_{cf}) and mutated stego frame (S_{mf}).

$$\begin{cases} d_1 = \left|k_f - S_{cf}\right| \\ d_2 = \left|k_f - S_{mf}\right| \end{cases} \tag{11}$$

Step 11 Find the d_{f1} and d_{f2} by taking the sum of all differences of d_1 and d_2.

$$\begin{cases} d_{f1} = \overline{d_1} \\ d_{f2} = \overline{d_2} \end{cases} \tag{12}$$

Figure 7 Flowchart of the proposed algorithm



Step 12 Compare d_{f1} with d_{f2} to find the optimal stego frame as per the given equation (13). If d_{f1} is greater than d_{f2}, then the candidate stego frame is stored as the optimal stego frame, whereas if, If d_{f1} is less than d_{f2} then mutated stego frame is considered as the optimal stego frame.

$$\begin{cases} O_f = S_{mf} & if\ d_{f1} < d_{f2} \\ O_f = S_{cf} & if\ d_{f1} > d_{f2} \end{cases} \tag{13}$$

Table 1 shows the variation in keyframe pixel value after data hiding by taking an example. The original keyframe has pixels value of 90, 11, 204 and 15; whereas the circular shifted secret data bits are considered as 10110000. After data hiding using the 2 bit LSB replacement technique, the pixels value change to 88, 8, 207 and 14. Next, the mutation operation changes the pixels value to 92, 12, 202 and 12. The absolute difference d_1 between keyframe pixels and candidate stego frame pixels (|90 – 88|, |11 – 8|, |204 – 207|, |15 – 14|) are 2 bit, 3 bit, 3 bit, and 1 bit, whereas the absolute difference d_2 between keyframe pixels and mutated pixels (|90 – 92|, |11 – 12|, |204 – 203|, |15 – 12|) are 2 bit, 1 bit, 1 bit, and 3 bit. The value of d_{f1} and d_{f2} becomes the 9 and 7. The value of d_{f1} is greater than d_{f2}, thus the mutated frame is selected as the optimal stego frame. Thus the proposed algorithm helps in reducing the variability and correspondingly the PSNR improves significantly.

Table 1 Variation in pixel value after data hiding and mutation operation

Circular shifted secret data bits	10110000			
Keyframe pixels value	90	11	204	15
Keyframe pixels in bits	01011010	00001011	11001100	00001111
Candidate stego frame pixels	01011000	00001000	11001111	00001110
Candidate stego frame pixels value	88	8	207	14
Output after Mutation operation	01011100	00001100	11001011	00001010
Pixel of mutation frame	92	12	203	12
Original difference d_1	2 bit	3 bit	3 bit	1 bit
Difference after mutation d_2	2 bit	1 bit	1 bit	3 bit
d_{f1}	9			
d_{f2}	7			

4.2 Data extraction

Figure 8 shows the flowchart of extraction of secret information from the stego video.

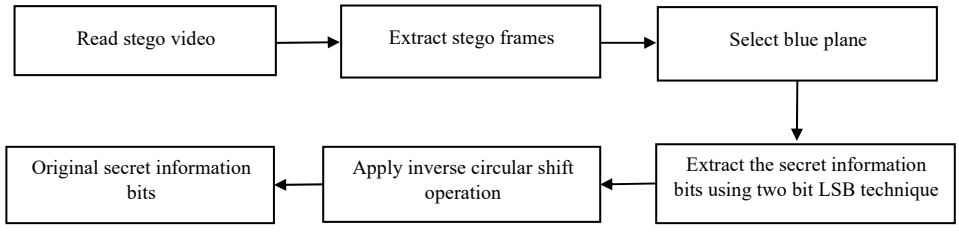
Step 1 Read the stego video and extract the stego frames.

Step 2 Select the blue plane of stego frames as it contains the secret data bits.

Step 3 Extract the secret information bits using the 2 bit LSB technique.

Step 4 Apply the inverse circular shift operation on the secret data bits to retrieve the secret image.

Figure 8 Flowchart of data extraction



5 Computational results and discussion

The proposed algorithm is implemented on the standard videos and images. The cover videos namely carphone, salesman, and Suzie have the dimension of 176*144 with the frame rate of 30 *fps* whereas the foreman video has the dimension of 352*288 with 25 *fps*. The mandrill image with dimension 512*512 is used as the secret image. One of the keyframes of the cover video and the secret image is shown in Figure 9. As the data hiding is done using the 2 bit LSB technique, the secret image is resized to 1/4th of the size of the keyframe. The computation results are measured in terms of imperceptibility and histogram analysis. The MSE, peak signal to noise ratio (PSNR), and structural similarity index (SSIM) are computed to support the imperceptibility analysis, whereas histograms of keyframe and stego frame are compared to evaluate the histogram analysis. The PSNR measures the perceptual quality of the keyframe after data hiding, whereas MSE calculates the square error in the keyframe due to data hiding (Ramalingam and Isa, 2016). The SSIM measured the structural similarity between the keyframe and stego frame. If $KF(i, j)$, $OF(i, j)$ denotes the keyframe and the optimal stego frame pixels, then MSE, PSNR, and SSIM can be determined using equations (14), (15) and (16).

$$MSE = \frac{1}{AB}\sum_{i=1}^{A}\sum_{j=1}^{B}(KF(i,\ j) - OF(i,\ j))^2 \tag{14}$$

$$PSNR = 10\log_{10}\frac{255^2}{MSE} \tag{15}$$

$$SSIM_{(x,\ y)} = \frac{\left(2\mu_{kf}\,\mu_{Of} + c_1\right)\left(2\sigma_{kfOf} + c_2\right)}{\left(\mu_{kf^2}\,\mu_{Of^2} + c_1\right)\left(\sigma_{kf^2}\sigma_{Of^2} + c_2\right)} \tag{16}$$

where μ_{kf}, μ_{of}, and σ_{kf}, σ_{of} represent the mean and standard deviation of the keyframe and stego frame respectively. $C_1 = (K_1L)^2$, and $C_2 = (K_2L)^2$ are constant. K_1 and K_2 have a default value of 0.01 and 0.03 respectively and L has a maximum value of 255 (Sahu and Swain, 2019).

5.1 Impercepbility analysis

A steganography algorithm with high value of PSNR and a low value of MSE indicates the higher imperceptibility. The computation results of the proposed algorithm are presented in Figures 10, 11, and 12. The results in Figure 10 show that the minimum value of MSE is 0.212 obtained for the Foreman video and the maximum MSE is 0.249 for the salesman video. The average value of MSE, i.e., 0.227 is very small, which is required in the steganography algorithm. The results in Figure 11 show that the maximum PSNR of 54.85 *dB* is achieved for the Foreman video and the minimum PSNR of 54.16 *dB* for the salesman video. The following results show that the PSNR value for all the standard video sequences is above 54.5 *dB* which is very high from the threshold value of 30 *dB*.

Figure 12 shows that the value of SSIM is 0.999 for all the video sequences. The value of SSIM is near to unity, which is required in the steganography algorithm.

Figure 9 Keyframe of different videos, (a) carphone, (b) salesman, (c) Suzie, (d) foreman, (e) mandrill as secret image (see online version for colours)

Figure 10 MSE for different video (see online version for colours)

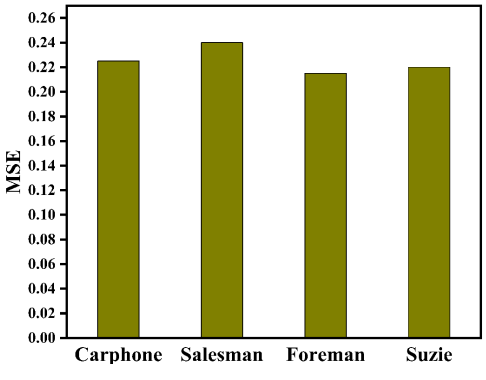


Figure 11 PSNR for different video (see online version for colours)

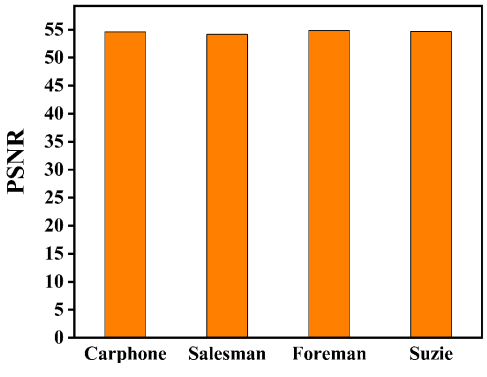


Figure 12 SSIM for different video (see online version for colours)

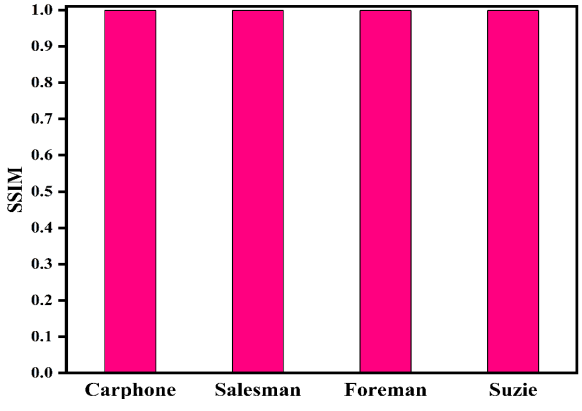
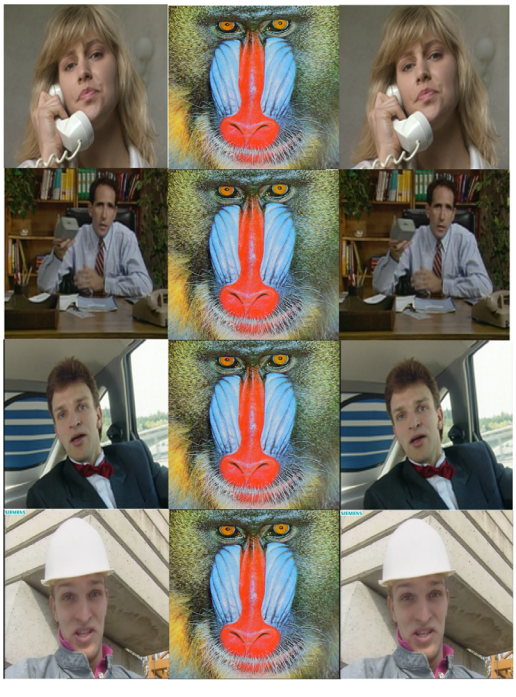


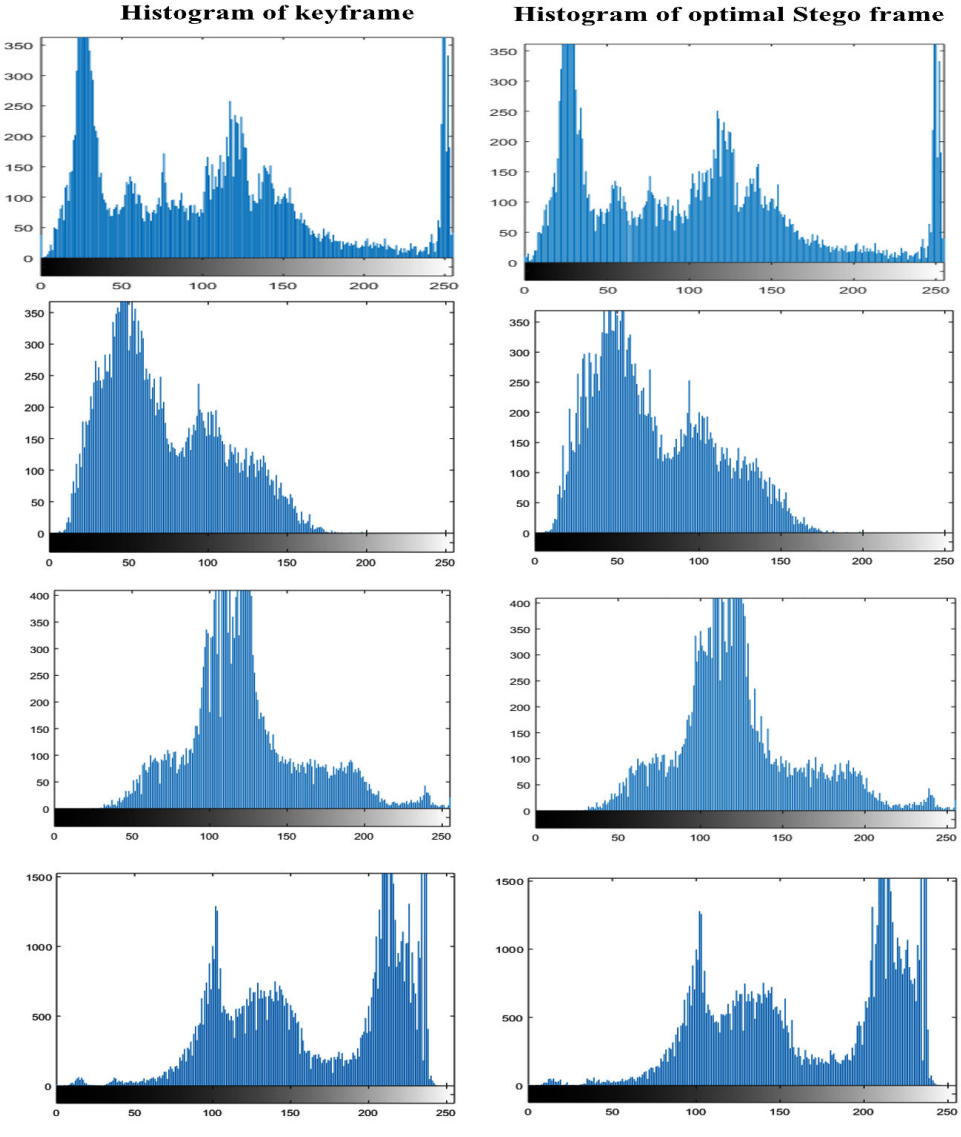
Table 2 shows the effect of circular shifting and mutation operation on PSNR value. The results show that the average value of PSNR obtains using the LSB-based data hiding algorithm is 50.65 *dB*, whereas the PSNR achieved using circular shifting with LSB data hiding is 53.84 *dB*. The average value of PSNR offered by the proposed algorithm is 54.57 *dB*, which shows the 7.18% and 1.33% improvement in the PSNR value when compared with the LSB and circular shifting-based LSB data hiding technique.

Figure 13 (a) Keyframe, (b) Secret image and (c) Optimal stego frame (see online version for colours)



(a) (b) (c)

Figure 14 Histogram of keyframe and optimal stego frame (see online version for colours)



To evaluate the visual quality, the keyframe and corresponding optimal stego frame are compared and the results are shown in Figure 13. The following results show that the optimal stego frames of all the standard video sequences look similar to keyframes, which supports that no one can know about the existence of secret data in the optimal keyframes.

5.2 Histogram analysis

A histogram is the graphical representation of the occurrence of pixel's value. Data embedding in the keyframe effects the pixel value and changes the histogram. In an ideal

case, the histogram of the keyframe and stego frame should be identical. The histogram of the different video sequences and corresponding optimal stego frame are shown in Figure 14.

It is observed that there is a very small distortion in the histograms of the optimal stego frame which makes the proposed technique to be used for secure data communication.

Table 2 Variation in the PSNR value with circular shifting and mutation operations

Cover video	*PSNR (dB)*		
	LSB data hiding	*Circular shifting + LSB data hiding*	*Circular shifting + LSB data hiding +mutation*
Carphone	50.56	53.82	54.61
Salesman	50.65	53.30	54.16
Foreman	50.66	54.11	54.85
Suzie	50.76	53.13	54.67
Average	50.65	53.84	54.57

Table 3 Comparative analysis with existing techniques based on PSNR in (db)

Images	*Zhu et al. (2021)*	*Shah and Bichkar (2018)*	*Sharma et al. (2021)*	*Sahu et al. (2018)*	*Zear and Singh (2021)*	*Proposed algorithm*
Lena	37.23	52.33	49.89	42.98	31.13	54.19
House	---	52.64	---	42.63	---	54.26
Pepper		---	---	43.23		54.29
Sailboat	---	---	---	41.75	---	55.01
Baboon	38.12	54.43	42.22	38.83	---	53.28
Barbara	---	53.80	---	39.03	---	54.54
Couple	---	---	---	41.53	---	54.45
Female	---	---	---	---	---	53.60
Average value	*37.88*	*53.30*	*46.05*	*41.42*	*30.93*	*54.20*

5.3 Comparison with existing techniques

The performance of the proposed algorithm is compared with the reference algorithms in terms of PSNR. The GA, GO, flipping, and DCT-SVD based algorithms are considered for the comparative analysis. The results of these reference algorithms are presented by considering the image as the cover media. The cover image and video frame are equivalent to each other. Thus, the comparative analysis is done by considering standard images as the cover images and the results are shown in Table 3. The computational results show that the average value of PSNR for the proposed algorithm is 54.20 dB, which shows the 30.11% and 1.66% improvement in PSNR value when compared with Zhu et al. (2021) and Shah and Bichkar (2018) which are using GA as an optimisation algorithm. Further, the proposed algorithm results improved by 15.0%, 23.5% and 75.2% when compared with Sharma et al. (2021), Sahu et al. (2018) and Zear and Singh (2021) which are using GO, flipping and DCT-SVD techniques. The following result indicates

that the proposed algorithm achieved higher imperceptibility when compared with other state of art methods.

6 Conclusions and future scope

An optimal data hiding technique using circular shifting and mutation operation has been proposed for the video steganography. The circular shifting operation is applied to search for the optimal direction for data hiding, whereas the mutation operation is used to reduce the variability more between the keyframe and stego frame. The computational results show that the proposed algorithm provides better visual quality and imperceptibility in a fewer number of iterations. In the last, comparative analysis shows significant improvement in PSNR value when compared with other state of art methods. The future research direction is to use transform domain techniques and error correction codes to provide robustness against attacks.

References

Almazaydeh, L. (2020) 'Secure RGB image steganography based on modified LSB substitution Laiali Almazaydeh', *International Journal of Embedded System*, Vol. 12, No. 4, pp.453–457.

Bai, J. et al. (2017) 'A high payload steganographic algorithm based on edge detection', *Displays*, December, Vol. 46, pp.42–51, DOI: 10.1016/j.displa.2016.12.004.

Hemanth, D.J. et al. (2016) 'Application of genetic algorithm and particle swarm optimization techniques for improved image steganography systems', *Open Physics*, Vol. 14, No. 1, pp.452–462, DOI: 10.1515/phys-2016-0052.

Johnson, N.F. and Mason, G. (1998) 'See the unseeing 1998.pdf', *IEEE Transactions on Image Processing*, Vol. 31, No. 2, pp.26–34.

Katoch, S., Chauhan, S.S. and Kumar, V. (2021) 'A review on genetic algorithm: past, present, and future', *Multimedia Tools and Applications*, DOI: 10.1007/s11042-020-10139-6.

Liu, Y. et al. (2018) 'Video steganography: a review', *Neurocomputing*, DOI: 10.1016/j.neucom.2018.09.091.

Mstafa, R.J. and Elleithy, K.M. (2016) 'A video steganography algorithm based on Kanade-Lucas-Tomasi tracking algorithm and error correcting codes', *Multimedia Tools and Applications*, Vol. 75, No. 17, pp.10311–10333, DOI: 10.1007/s11042-015-3060-0.

Paul, R. et al. (2013) 'Hiding large amount of data using a new approach of video steganography', *IET Conference Publications*, 647 CP, pp.337–343, DOI: 10.1049/cp.2013.2338.

Prasad, S. and Pal, K.A. (2020) 'Stego-key-based image steganography scheme using edge detector and modulus function', *International Journal of Computational Vision and Robotics*, Vol. 10, No. 3, pp.223–241.

Ramalingam, M. and Isa, N.A.M. (2016) 'A data-hiding technique using scene-change detection for video steganography', *Computers and Electrical Engineering*, Vol. 54, pp.423–434, DOI: 10.1016/j.compeleceng.2015.10.005.

Sadek, M.M., Khalifa, A.S. and Mostafa, M.G.M. (2015) 'Video steganography: a comprehensive review', *Multimedia Tools and Applications*, Vol. 74, No. 17, pp.7063–7094, DOI: 10.1007/s11042-014-1952-z.

Sahu, A.K. and Swain, G. (2019) 'An optimal information hiding approach based on pixel value differencing and modulus function', *Wireless Personal Communications*, Vol. 108, No. 1, pp.159–174, DOI: 10.1007/s11277-019-06393-z.

Sahu, A.K., Swain, G. and Suresh Babu, E. (2018) 'Digital image steganography using bit flipping', *Cybernetics and Information Technologies*, Vol. 18, No. 1, pp.69–80, DOI: 10.2478/cait-2018-0006.

Shah, P.D. and Bichkar, R.S. (2018) 'A secure spatial domain image steganography using genetic algorithm and linear congruential generator', *Advances in Intelligent Systems and Computing*, Vol. 632, pp.119–129, DOI: 10.1007/978-981-10-5520-1_12.

Sharma, S. et al. (2021) 'Digital watermarking using grasshopper optimization algorithm', *Open Computer Science*, Vol. 11, No. 1, pp.330–336, DOI: 10.1515/comp-2019-0023.

Zear, A. and Singh, P.K. (2021) 'Secure and robust color image dual watermarking based on LWT-DCT-SVD', *Multimedia Tools and Applications*, DOI: 10.1007/s11042-020-10472-w.

Zhu, T., Qu, W. and Cao, W. (2021) 'An optimized image watermarking algorithm based on SVD and IWT', *Journal of Supercomputing*, DOI: 10.1007/s11227-021-03886-2.