Open Access Library Journal
2024, Volume 11, e11489
ISSN Online: 2333-9721
ISSN Print: 2333-9705

Enhancing Video Steganography Techniques Using Hybrid Algorithms

Mohammad Anwar Hossain, Noshin Un Noor, Ahsan Ullah, Md. Sabbir Hossain Noman, Shubra Das Pranta, Lubna Mostafa Bristy, Md. Mainul Bashar

Department of Computer Science & Engineering, World University of Bangladesh, Dhaka, Bangladesh
Email: hossainanwar1616@gmail.com, neha68219@gmail.com, ahsan.ullah@cse.wub.edu.bd, sabbirhossainnoman@gmail.com, sd.pranta1552@gmail.com, lubnabristy1234@gmail.com, sopon6227@gmail.com

How to cite this paper: Hossain, M.A., Noor, N.U., Ullah, A., Noman, Md.S.H., Pranta, S.D., Bristy, L.M. and Bashar, Md.M. (2024) Enhancing Video Steganography Techniques Using Hybrid Algorithms. *Open Access Library Journal*, **11**: e11489.
https://doi.org/10.4236/oalib.1111489

Received: March 26, 2024
Accepted: September 27, 2024
Published: September 30, 2024

Abstract

The proposed work addresses the imperative of safeguarding sensitive information through a dual-security approach of encryption and steganography. Employing the Advanced Encryption Standard (AES) for the first level, and bitmap graphics as video frames with steganography techniques in the second, it achieves heightened data protection and imperceptibility, validated through superior statistical quality assessments. It advocates combining AES encryption with Fisher-Yates-based randomization for heightened data security. The paper introduces automated processes for recovery, decryption, concealment, and encryption, utilizing XOR, 1 bit LSB, and spiral pixel selection to optimize payload capacity and quality metrics in films. The research suggests implementing the Advanced Encryption Standard (AES) algorithm for secure video steganography. The proposed method involves AES encryption, Fisher-Yates frame shuffle, Prewitt edge detection, and triple XOR embedding, enhancing data security and imperceptibility. The algorithm employs a dynamic frame selection technique and pixel filtering using Prewitt edge detection, contributing to a robust video steganography system. The embedding and retrieval processes ensure secure communication, demonstrating the methodology's effectiveness in concealing and extracting information within video frames.

Subject Areas

Information and Communication Theory and Algorithms

Keywords

AES, Video-Steganography, XOR, LSB, Spiral-Pixel-Selection

1. Introduction

Steganography is the practice of hiding data in plaintext files or conversations so

that it cannot be discovered. It is frequently used in conjunction with encryption to increase security. Digital content such as text, images, movies, and audio can all be concealed; to improve concealment, buried text is usually encrypted or processed. With the help of video steganography, which conceals information within videos, covert communication is made possible to avoid prospective attackers in a number of industries, including the military, intelligence, healthcare, and multimedia. To protect national security, steganography techniques are essential for encrypting communications. Although there are trade-offs between them, resilience, security, imperceptibility, and concealing ability are necessary for each steganography technique. For picture, audio, and video steganography, the Least Significant Bits (LSB) substitution method is frequently utilised. It provides quick and effective hiding, however overloading carrier pictures can jeopardise security. While compressed domain techniques improve security, they may also result in the loss of unnecessary data. Finding a balance between these attributes is essential when choosing a suitable steganography method.

In order to improve resilience, imperceptibility, and payload capacity with higher video quality, as well as to quickly and error-free conceal secret communications, the primary objective of this research article is to develop an automated system that combines dependable encryption and steganographic technology. The following is the paper's contribution:

Integrating the Fisher-Yates principle-based randomization process with the AES algorithm to increase data security. N, or the number of frames depending on the message bit length, is selected from a range of permutations. This will encode the secret data using a 128-bit key AES technique, resulting in a more secure video steganography solution. The secret data's automatic encryption and hiding, as well as its recovery and decryption, employ spiral pixel selection in conjunction with XOR and 1 bit LSB techniques to get an excessive payload capacity to PSNR ratio and other quality assessment metrics in movies.

Literature Review

The steganography articles that are relevant to this subject are included in this section. Along with cryptography, random video frame selection, pixel selection methodology, and XOR with LSB approach, these papers also cover video steganography. The majority of these studies discussed the LSB approach, a common steganographic technology technique.

The authors of [1] introduce a video steganography method utilizing random frame selection and LSB to conceal sensitive data, including one-time passwords, within a carrier video file. The technique involves embedding metadata in the first frame and employing a second frame with a high PSNR. However, a lack of pixel selection strategy limits data hiding effectiveness. While improving carrier and stego file quality, the method's security is limited by using a single security level.

The authors of [2] propose a technique combining AES encryption with LSB algorithm for securely embedding secret data into video frames. Plain text is

encrypted using AES before embedding into the cover video frames using LSB with a 1-1-1 scheme. However, details regarding frame selection and observational methods are lacking. While quality metrics show promising results, enhancing security through improved frame and pixel selection methods is recommended.

The authors of [3] suggested a technique that has a good peak signal noise ratio and integrates neural networks, fuzzy logic, secure base LSB approach, and video steganography (the cover video is in AVI format, which is ideal for the research). Higher levels of security are, however, challenging to get due to a few shortcomings in the model, such as pixel and frame selection, imprecise descriptions of the video and frame format information, and so forth.

In [4], two metamorphic methods for text obfuscation were introduced by the author: "XOR of information with LSBs" and "XOR of message with symmetric key". The first method, "XOR with LSB", is the most structurally similar to the original cover carrier in terms of measurement metrics; nevertheless, "XOR of text with symmetric cryptography" offers higher data hiding security. The main drawback of the suggested approach is that data gradually becomes less visible with each frame.

The authors of [5] propose a strategy to address weaknesses in spatial domain processes by utilizing steganography in the transform domain. They employ DCT and DST techniques alongside Scrambling-AES encryption on the cover video. However, the lack of mention of frame and pixel selection procedures limits the method's effectiveness. Despite this, the outcome is deemed acceptable.

The authors of [6] put forth a model that could improve security by hiding sensitive information in audio cover carriers. One potential drawback of this approach is that AES-256 takes longer than AES-128 to encrypt the secret data, even though the AES technique with the smallest key size of 128 bits is unbreakable.

A good result was obtained by the authors in [7] when they combined the 2-bit LSB and DES algorithms to create a model for hiding secret data inside picture cover carriers. However, the cover carrier lacks pixel selection, and DES is less secure and faster than AES. Consequently, it can be this paradigm's drawback.

In [8], the authors propose a technique to embed an audio track within a carrier video clip. They utilize CryptGenRandom to select random frames for embedding secret bits, which are encrypted and XORed with the original LSBs of the carrier video frames to ensure security and improve PSNR. However, employing a more appropriate pixel selection technique could have enhanced security.

In [9], the authors discuss the challenges of maintaining data security and privacy in cloud computing and introduce a study that proposes an enhanced method for securing data through image steganography. The authors developed Multi Level Encryption Algorithm (MLEA) and Two-Level Encryption Algorithm (TLEA) to conceal data within images using the Least Significant Bit (LSB) technique. Through comparison with existing methods, the study shows that their approach offers better security performance, averaging 19.7814% improvement.

In [10], the authors discuss the rise of cloud computing as a paradigm shift from traditional computing models, emphasizing its reliance on shared resources rather than local servers. It highlights the increasing adoption of cloud technology and predicts even greater integration into daily work processes, particularly through mobile devices, over the next decade. While cloud computing offers benefits such as improved performance, scalability, and resource utilization, it also presents security challenges, especially regarding data storage. The proposed algorithm aims to address these security concerns by enhancing data protection in cloud environments.

2. Fundamental Principles

2.1. Methodology

This research paper presents a comprehensive approach to designing and implementing encryption and embedding techniques to ensure secure and imperceptible data hiding. The authors outlined the procedure using six distinct phases. The different phases involved in this process include planning, requirement analysis, proposed algorithm, implementation, testing, and maintenance.

2.2. Description of Methodology

We can see that Figure 1 comprised the development process with the following phases:

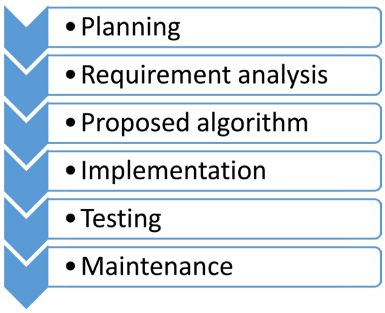

Figure 1. Proposed methodology.

2.2.1. Planning

Planning well is the first step to a fruitful study. So, when the authors started their research, they had a well-thought-out plan. The strategy includes both the operational technique and the study topic. First, the authors looked over a lot of research on connected topics. The writers then used their comprehension of those pieces to determine the title. After looking over those research, the authors found certain flaws in each one. In order to overcome those limitations and preserve the novelty of their discoveries, the researchers planned their investigation.

2.2.2. Requirement Analysis

Every task has specific requirements based on needs. So, some specialised resources are needed for the work we have recommended.

1. System Requirements—Two categories of system requirements can be distinguished:

- Software prerequisite
- Hardware prerequisite

❖ Software Requirements

1) Language—C#

2) Environment—.NET Framework 4.8

3) Operating System—Windows

4) Local Server

5) External Algorithm—AES, XOR, Fisher yaets

❖ Hardware Requirements

1) Laptop or Desktop with processor

2) USB cable

2. User Requirements: What the user expects from the system is one of the user requirements. The user needs data security that includes payload capacity, resilience, and imperceptibility for this. The following describes the user requirements for LSB XOR with Prewitt edge detection in video steganography:

Key Functionalities:

1) Seamless Embedding:

-Permit users to insert confidential information into videos without any discernible alterations.

2) Secure Encryption:

-Use strong algorithms to encrypt data so that it remains confidential even if it is intercepted.

3) Reliable Extraction:

-Reliable and integrity-preserving extraction of hidden data from Stego films.

Interface Friendly to Users:

1) Intuitive Design:

-Easily guide users with explicit directions through the embedding and extraction processes.

2) Adaptive Input/Output:

-Permit users to choose from a variety of video files and output formats.

3) Secure Key Management:

-To preserve confidentiality, provide safe methods for handling encryption keys.

Options for Customisation:

1) Payload Size:

-Permit consumers to modify data embedding according to their own requirements and quality preferences.

2) Edge Detection Sensitivity:

-Permit sensitivity modification to maximise visual impact and embedding efficiency.

3) Encryption Algorithm:

-Offer choices for choosing encryption methods that adhere to security spe-

cifications.

2.2.3. Proposed Algorithm

The AES algorithm, which converts plaintext data into cipher text using a 128-bit key and 10 cycles, is recommended by the author for use in the study. AES ensures security with 25 active S-Boxes every four rounds by using specialised encryption and decryption techniques. In order to improve and safeguard data in cover films, each round uses a distinct 128-bit round key that is produced from the primary AES key.

2.2.4. Implementation

The user must utilise our suggested tool to provide the secret data, which is automatically encoded after it is received. The widely used and safe symmetric encryption technique AES is used to protect it. AES encrypts data blocks using symmetric keys that have progressively more bits—128, 192, or 256. Sensitive data is also decoded using the same encryption key. But in this experiment, a 128-bit key length was used to encrypt the confidential data, which produces faster results and uses less RAM. Although it is currently unbreakable, a computerised function automatically allocates a 128-bit key size. As a result, 128-bit AES was employed to ensure data security whether or not it was taken from video frames. A function built in the C# programming language handles the entire AES process. Using the video splitter function built into the C# language, the cover video carrier is extracted to frames using BMP picture format once the secret message has been encrypted.

1. Choosing the Hidden Message and Video:

-Select a video file to serve as your message's carrier.

-Get the hidden message ready to be hidden in the video.

2. Extracting Frames:

-Break up the video into separate frames, just like you would with a book. For data embedding, every frame functions as a still image.

3. Pixel Selection Using Prewitt Edge Detection:

-Use Prewitt edge detection to find areas of each frame that have a high contrast. These borders are the best for embedding since they make little pixel changes less noticeable.

4. Secret Message Encryption and Conversion:

-To maintain confidentiality, encrypt the secret message using AES 128-bit encryption.

-To prepare it for embedding, convert the encrypted message to binary.

5. Integrating Confidential Information:

-Repeat through a subset of each frame's edge pixels.

-Find the least significant bit (LSB) for each RGB value of a pixel, then perform XOR with a message bit.

6. Merging Stego Frames to Make Stego Video:

-Reassemble the altered frames to create a sequence of videos.

-The generated video should be saved as the stego video with the secret message.

Key Points:

-To reduce visual impact, embedding is prioritised for edge areas.

-Message secrecy is protected by AES encryption.

-LSB XOR embedding efficiently strikes a balance between visual quality and concealing.

2.2.5. Testing

The authors' implementation yielded a result that satisfies the requirements. They then found that the method fits every necessity for enhancing encryption security and embedding techniques to provide secure and undetectable data concealing from other algorithms after comparing it to other comparable current algorithms.

2.2.6. Maintenance

Secrecy in steganography software secures data concealment by embedding messages covertly within various cover objects, safeguarding against detection. Key Principles for Maintaining Information Hiding.

Key Principles for Maintaining Information Hiding: Enhancing the inconspicuousness of stego objects, embedding efficiency preserves cover object quality. Robustness guarantees that the message will withstand image processing. Security requires elaborate embedding and encryption to prevent steganalysis.

Maintenance Strategies: For efficient steganography and data concealment, update algorithms frequently, keep an eye on performance, evaluate security, train users, and stay informed.

By adhering to these principles and implementing effective maintenance strategies, developers of steganography software can ensure that their tools continue to provide a secure and reliable means of concealing sensitive information.

3. Research Design and Analysis

3.1. Proposed Algorithms

3.1.1. Encrypting the Secret Message

The AES algorithm was recommended by the author for the investigation (**Figure 2**). The proposed method converts a block of data containing up to 256 bits of plain text into a cypher text. This algorithm includes many specialised methods for both encryption and decoding. For both encryption and decryption, the key has a length of 128 bits, and there are 10 cycles in a repeating cycle. Every round, the data is encrypted and decrypted using the same key. The AES branch number protects against disparity and linear cryptanalysis by guaranteeing that there are at least 25 active S-Boxes in every constant four rounds of the AES. Every round uses a different 128-bit round key that is produced from the main AES key. The recommended approach is to enhance and protect the data in cover videos.

In **Figure 2**, we can see that A 128-bit key with 10 repeating cycles is used in the AES encryption process. Each cycle consists of the following four steps: AddRoundKey, MixColumns, SubBytes, and ShiftRows. SubBytes uses an S-box table

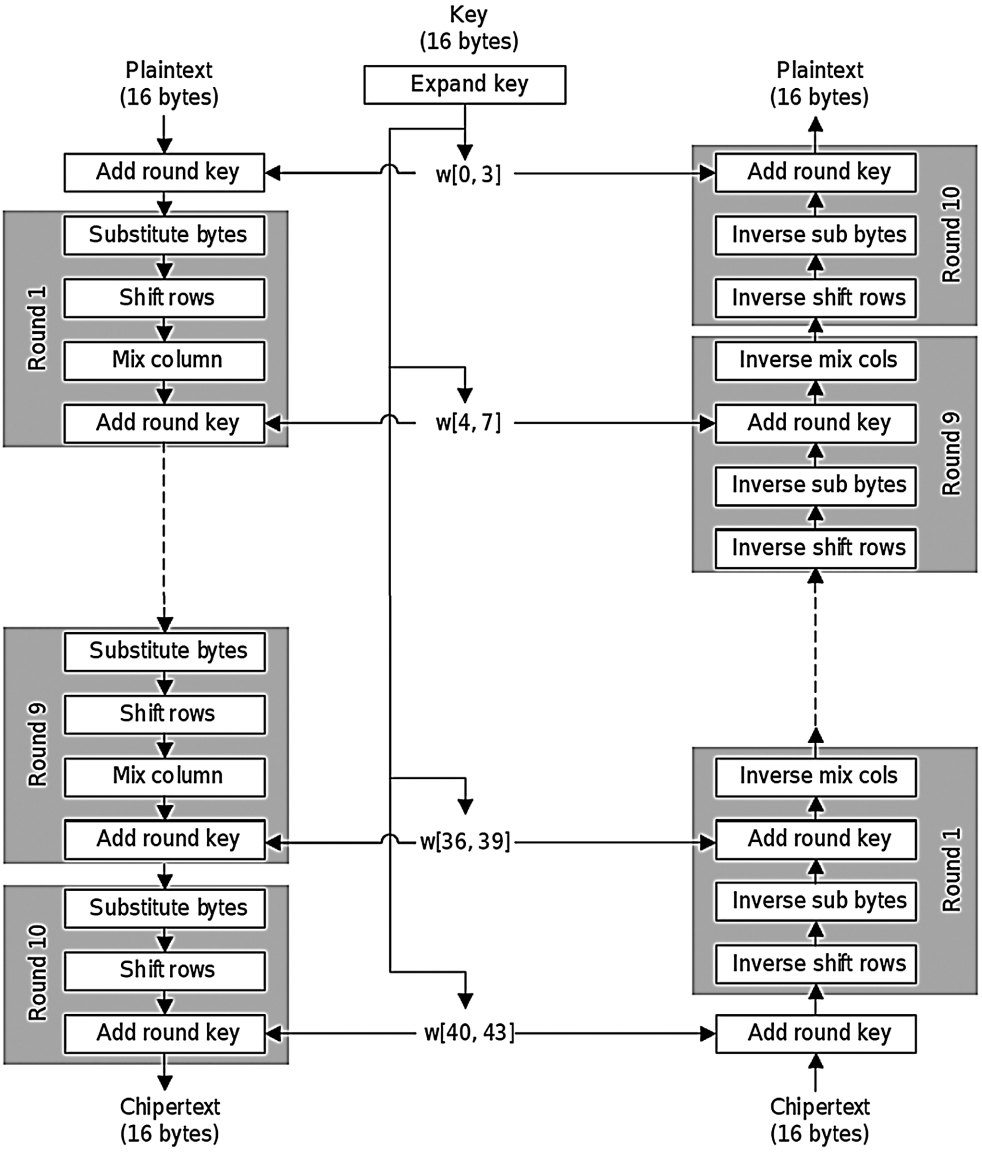

Figure 2. AES Encryption and Decryption process.
https://www.researchgate.net/figure/AES-encryption-and-decryption-process-block-diagram-4-Near-Field-Communication-NFC_fig1_318528672.

and a special method to replace each byte. ShiftRows modifies the internal state of the cypher in each row. The branch number of AES provides protection against linear cryptanalysis and discrepancy. A unique 128-bit round key that is obtained from the primary AES key is used for each round. The AES procedure is controlled by a C# function, and the encrypted cover video is then divided into frames using the BMP format.

3.1.2. Randomization Frame Selection Method

The number of frames (Nf) required for embedding is determined by the length of the secret encrypted message (Sm L) and the dimensions (Dv) of the cover video, which are calculated using Equations (1) and (2). Based on Fisher Yeats

principles, a random sampling process is used to choose the retrieved frames (in BMP format).

$$\text{MEMB} = \text{MaxDv of Dv} * 4\text{CD} * 3 \text{ bit} \quad (1)$$

$$\text{Nf} = \text{CB/MEMB [Where, (Nf + 1) > 1]} \quad (2)$$

In this case, Maximum Embed Message is being utilized. MEMB stands for "bit per frame", CB for "total length of Cypher text in bits", and MaxDv for "maximum dimension of the inputted video", which can be either "width" or "height". Eight directions are represented by the Complete Direction (CD) that is displayed. The 3-bit embedding capacity of a single pixel is represented by 3 bit, and MaxDv multiplied by 4CD yields the maximum number of pixels that can be embedded in a single frame.

Figure 3 depicts the framework, a crucial tactic that provides us with a permutation based on the input key and the quantity of frames. The frames we choose will be the first (Nf + 1) > 1 frame from the permutation. Specifically, the meticulously gathered information allows for the same key to be used for reverse shuffling, which will simplify the process of extracting data from the frame. The length of the secret data, the shuffle key, and other meta data are always retained in the first frame, and the actual encrypted messages are progressively incorporated in the subsequent frames.

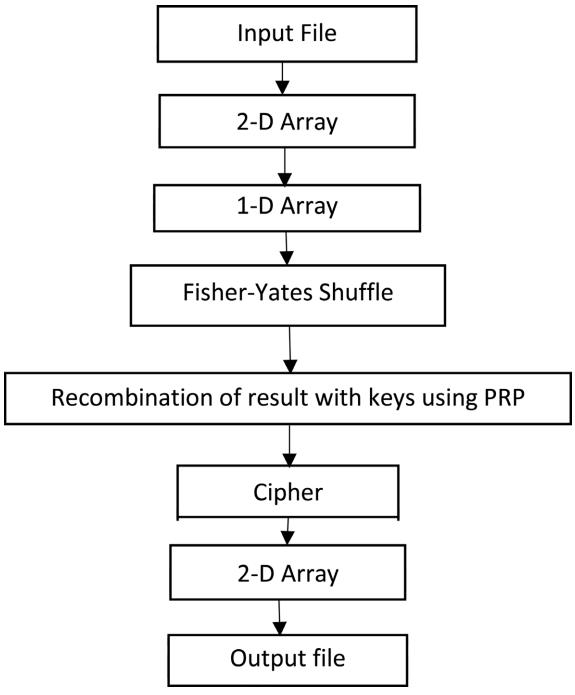

Figure 3. Frame selection based on fisher yeats.

3.1.3. Pixel Filtering Algorithm

Prewitt edge detection is a method that recognises boundaries inside an image, much like a skilled mapper traversing through uncharted territory. It uses numerical grids, known as kernels, that are responsive to variations in brightness both

horizontally and vertically, giving them the appearance of boats navigating pixelated seas. Prewitt filters define edges similarly to a cartographer charting ridgelines by identifying abrupt transition points by computing gradients based on pixel intensity levels. A threshold separates important borders from small deviations, producing an edge-detected image made up of important edges. Prewitt is straightforward and effective, yet its simplicity can cause noise to be misinterpreted. Still, its directness and speed make it useful for finding objects and gathering important data. On the other hand, Canny edge detection provides a dynamic method of edge detection that improves accuracy and versatility by utilising gradient magnitude and direction equations.

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \tag{3}$$

$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} \Big/ \frac{\partial f}{\partial x}\right) \tag{4}$$

Using Equation (2) and a user-specified method, an even or odd pair is determined from the first and second MSB locations. Where FXB denotes the binary conversion function and FXr denotes the red value retrieval function mentioned in Figure 4. We can see a sample example of sampling of prewitt edge detection in Figure 5.

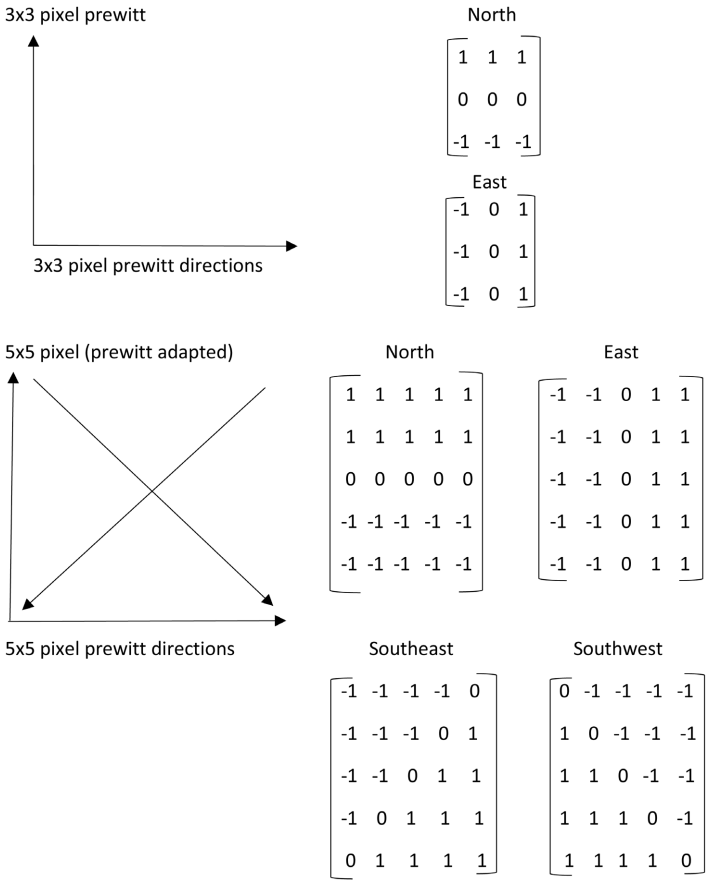

Figure 4. Approach of Prewitt edge detection.

3.1.4. Steganographic Process

The Embedding method and the retrieval technique are the two pathways of the steganographic process. The suggested system's embedding procedure is depicted in Figure 6. As soon as the stoner provides the system with secret plain text, the transmission is essentially encrypted using 128-bit AES. The photos are then taken out of the cover carrier, and the pixel selection system works with the systems mentioned earlier to choose fewer than 1% of the total pixels in a single

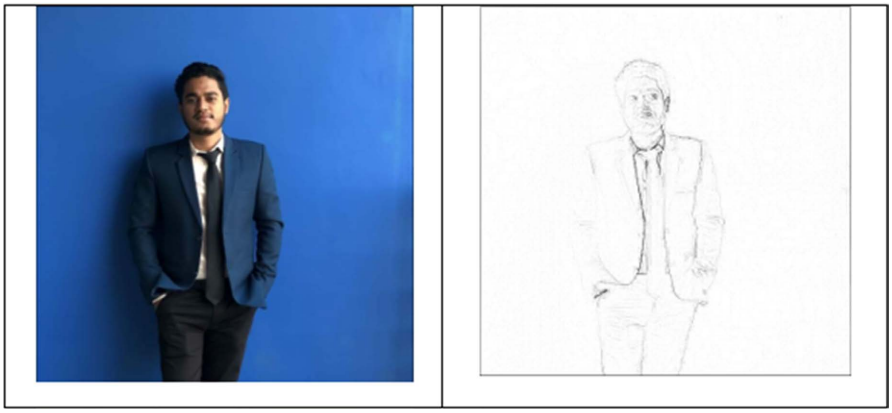

Figure 5. Sampling of Prewitt edge detection.

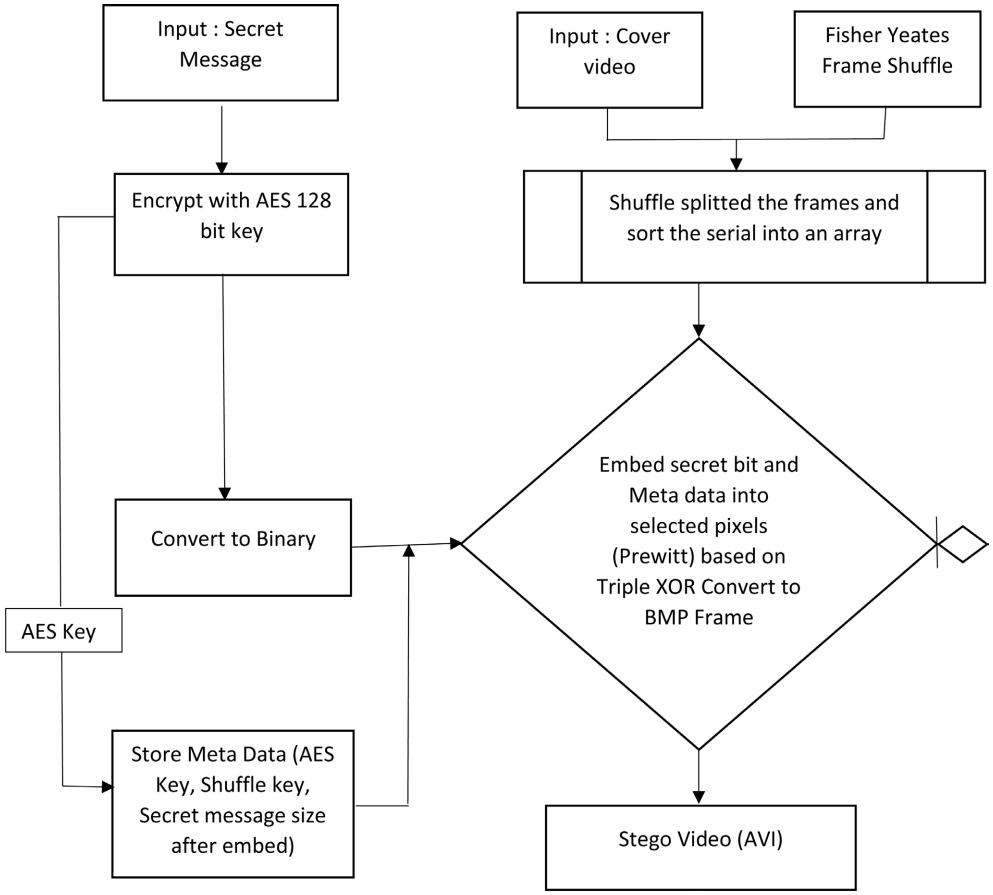

Figure 6. Embedding process.

frame. In the third phase, a triple XOR operation will be used to decode the secret message into 8-bit binary data, which will then be inserted with filtered pixels at the 1 bit LSB location. Here, the final indexed to RGB blocks will be replaced by the secret message bit, sixth indexed bit, and triple XOR. The process of retrieval is shown in Figure 7. Understanding the information—such as the message size, secret message embedding key, and pixel filtering technique that uses Equation to store the fixed four pixels—is the first step towards extracting secret data.

$$1^{st}\text{-pixel position} \quad (X_1, Y_1) = (H - 1/2 - 3, 1) \tag{5}$$

$$2^{nd}\text{-pixel position} \quad (X_2, Y_2) = (H, W - 1/2 - 3) \tag{6}$$

$$3^{rd}\text{-pixel position} \quad (X_3, Y_3) = (H - 1/2 + 3, W) \tag{7}$$

$$4^{th}\text{-pixel position} \quad (X_4, Y_4) = (1, H - 1/2 + 3) \tag{8}$$

Data that can be utilised to extract the AES key, message size, and filtering method is stored in these pixels. Once we know the filtering technique, our system will be able to obtain filtering pixels where the hidden message bit is kept by applying the embedding strategy. The secret message bit will then be obtained by triple XORing the RGB blocks' sixth and seventh indexed bits. Depending on the size of the message, we can use the AES key to decode the secret message once we have the bits and obtain the required plain text that was hidden.

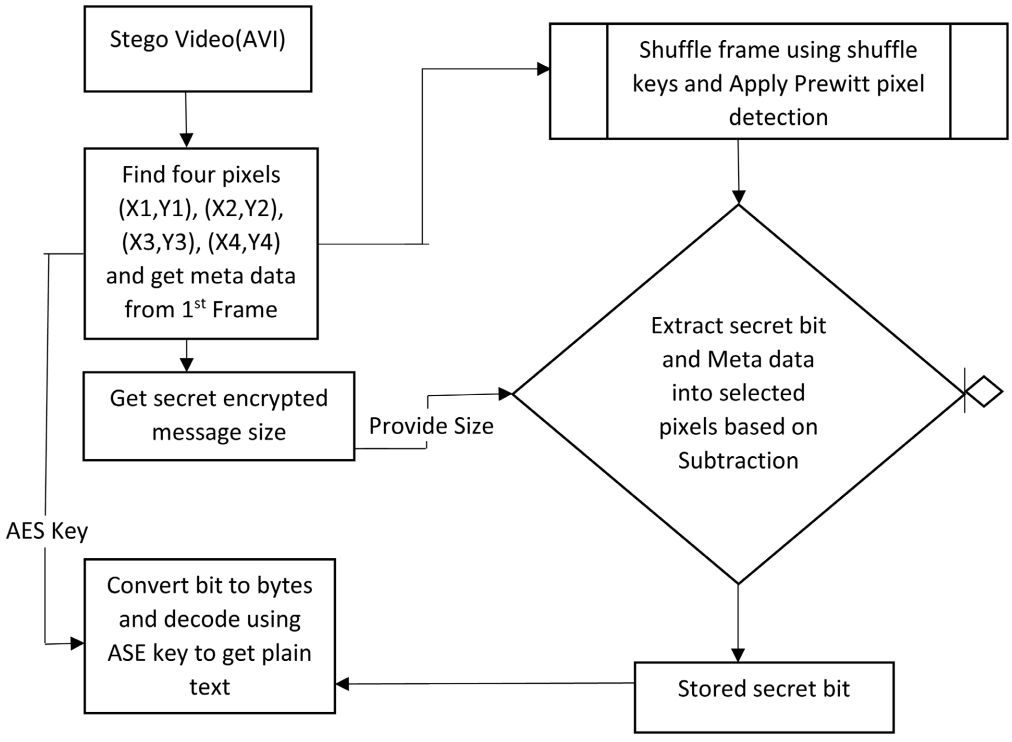

Figure 7. Retrieving process.

3.1.5. Algorithm for Embedding and Retrieving

The methods for embedding and retrieving data make it easier to hide and extract

sensitive information from cover videos. Users enter the cover video (CV) and secret message (Sm) into the embedding process. The FisherYeatsShuffle function creates a permutation for frame selection, whereas the AES function encrypts the secret data. XOR is used to embed metadata in the first frame, and triple XOR is used to embed the remaining secret data in the following frames. To recover encryption and shuffle keys, metadata is extracted in the retrieval algorithm. To extract the complete secret message, shuffle keys are used to produce random permutations that are used to choose frames.

Embedding Algorithm:

Result: Stego Video

```
Sm ← input
Cv ← input
Ct = 128-bit AES (Sm, key);
FE [] = Extract_Frames (Cv);
FLp [] = FisherYeatsShuffle (FE [], key);
SF = CB / MEMB;
MD = binary (shuffle key + salt + encryption key + salt + length of Sm)
embedMetaData (SF [0], MD);
(x1, y1), (x2, y2), (x3, y3), (x4, y4) ← length of MD;
SMB = binary (Sm);
v = 0;
For → SF [1 to N-1] & SMB [0 to N-1]
        If MEMB < SM embedSecretDataXOR (SF [v], SMB [v to n])
        else
            W = Frame Width;
            H = Frame Height;
            (Cx, Cy) = (H/2, W/2);
embed ((Cx, Cy));
            BL= Length of M;
Ppn= PrewittEdgeDetction(I);
             SMBL    length (SMB)
           (x1, y1), (x2, y2), (x3, y3), (x4, y4)    SMBL
           a = 0;
           while a ≤ Ppn do
                        Ds = (Cx±a, Cy±a)
embedXOR (Ds);
a++;
           function embed (position):
           RGB← position
UpdateRGB← message
stegoFrame.Add (SF)
stegoVideo←videoAssembler (stegoFrame [])
```

Retrieving Algorithm

Result: Secret Message

S_V ← input

F_E [] = Extract_Frames (S_v);

M_D = Retrive (F_E [0])

ShuffleKey = M_D.ShuffleKey

EncryptionKey = M_D.EncryptKey

TotalSecretMessageLength = M_D.MLength

FL_p [] = FisherYeatsShuffle (F_E [], ShuffleKey);

S_F = selectedFrame (FL_p [], M_D.MLength)

v = 0;

For→ S_F [1 to N-1]

If M_{EMB} <TotalSecretMessageLength

S_{MBL}← (x1, y1), (x2, y2), (x3, y3), (x4, y4)

S_D []=retrieveSecretDataTripleXOR (S_F [v], S_{MBL})

else

W = Frame Width;

H = Frame Height;

(C_x, C_y) = (H/2, W/2);

retrieve ((C_x, C_y));

BL= Length of M;

P_{pn} = PrewittEdgeDetction(I)

S_{MBL}← (x1, y1), (x2, y2), (x3, y3), (x4, y4)

a = 0;

while a ≤ P_{pn} do

D_s = ($C_{x\pm a}$, $C_{y\pm a}$)

S_D []= retrieveTripleXOR (D_s, S_{MBL});

a++;

function retrieve (position):

secretData.Add (S_D)

encryptData ← bitToBytes (secretData)

PlainSecretData ←Dycrypt(encryptData, EncryptionKey)

4. Result and Discussion

This section compares the outcomes of the suggested steganographic methodology with alternative approaches and provides a visual explanation of the differences. For statistical analysis, six quality evaluation indicators are used: MAE, SSIM, MSE, RMSE, PSNR, and SNR. Furthermore, the Embedding Time (TCEP) is used to evaluate the effectiveness of the suggested solution.

The two films that were chosen for the experimental test are Dhaka and Naymar (Figure 8). Downloadable versions of these two 512 by 512 AVI movies are offered. They have a five-second duration at the same frame rate. The suggested

method conceals a lengthy text within a video clip. The technique is written in C#, and the test results are generated using the .NET Framework version 4.7.2.

Figure 8. Cover videos.

Equations (11) to (15) provide a scientific explanation of the six necessary frame quality assessment matrices: PSNR, SSIM, MAE, SNR, RMSE, and MSE. These matrices are used to evaluate the efficacy and security of the steganographic process.

The Mathematical explanation for PSNR is

$$\mathrm{PSNR} = 10\log_{10}\frac{255^2}{\mathrm{MSE}} \tag{9}$$

In this case, PSNR is dependent on MSE and expressed in dB. Several investigations show that it is acceptable if the PSNR is greater than 40 dB between the stego frame and cover.

The definition of SSIM in science is

$$\mathrm{SSIM}(x,y) = \frac{\left(2\mu_x\mu_y + C_1\right) + \left(2\sigma_{xy} + C_2\right)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \tag{10}$$

In this case, x and y represent the image dimension. With a weak denominator, x and y represent the average of x and y, while K_1 and K_2 have default values of 0.01 and 0.03, respectively. Furthermore, L is set as the dynamic range of the pixel values. To make the partition easier, there are two variable quantities: $C_1 = (k_1L)^2$ and $C_2 = (k_2L)^2$.

The explanation for MAE in mathematics is

$$\mathrm{MAE} = \frac{1}{3MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[C(x,y) - S(x,y)\right]1 \tag{11}$$

In this case, the pixel position and the picture dimension denoted by M &N pertain to (x, y). The cover frame is denoted by C, the stego frame by S, and the city-block standard by [1].

The explanation of SNR mathematically is

$$\mathrm{SNR} = \frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\hat{f}(x,y)^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[f(x,y) - \hat{f}(x,y)\right]^2} \tag{12}$$

The Digital Image Processing formula is used. In this case, the original frame is indicated by f and x, the noisy frame by $\hat{f}$, and y denotes the location of a pixel.

The squared root base of the mean square error (RMSE) according to science

$$\text{RMSE} = \sqrt{\text{MSE}} \tag{13}$$

The Scientific definition for MSE is

$$\text{MSE} = \left(1 \times M \times N\right) \sum_{i=1}^{M} \sum_{j-1}^{N} \left(a_{ij} - b_{ij}\right)^2 \tag{14}$$

In this formula, the pixel value of the location i and j of the cover frame refers as a_{ij} where b_{ij} refers to the pixel value of the location i and j of stego frame.

The suggested method's results are constrained to a payload of 15 kilobases on four chosen video frames shown in Figure 9. The PSNR quality measurement matrices findings for the specified frames are shown in Table 1.

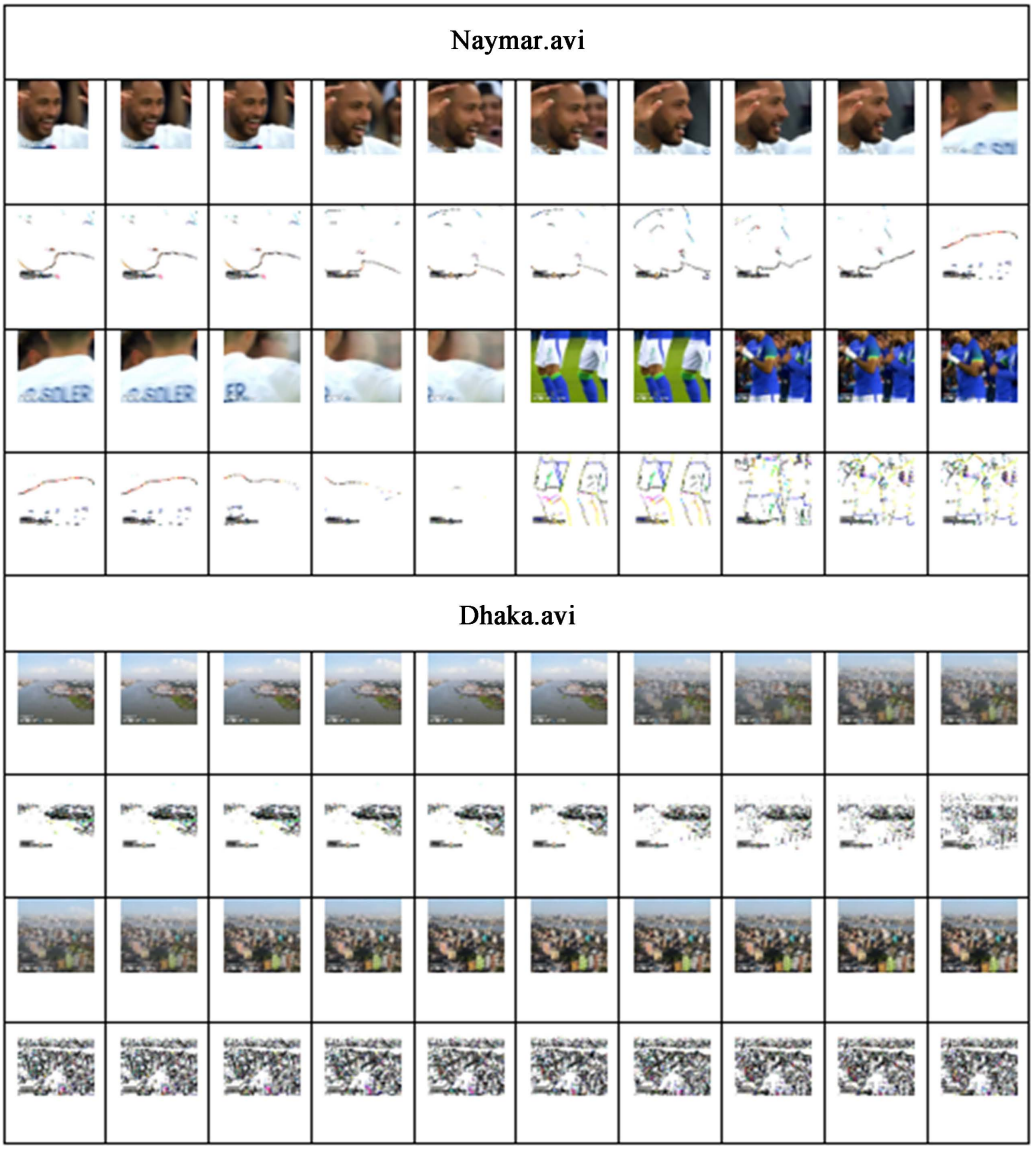

Figure 9. Encrypted and decrypted video images.

Table 1. PSNR for selected videos.

Frame No	Naymar (PSNR)	Dhaka (PSNR)
01 (M_D)	89.2390	88.6434
02 (S_{MB})	72.7453	72.5564
03 (S_{MB})	72.4536	72.3543
04 (S_{MB})	72.7766	72.9854
05 (S_{MB})	72.7867	72.6534
06 (S_{MB})	72.4545	72.3212
07 (S_{MB})	72.7865	72.8645
08 (S_{MB})	72.7678	72.2131
09 (S_{MB})	72.7867	72.8778
10 (S_{MB})	72.6383	72.4534
11 (S_{MB})	72.7877	72.5345
12 (S_{MB})	72.7896	72.4344
13 (S_{MB})	72.7456	72.1234
14 (S_{MB})	72.9867	72.2344
15 (S_{MB})	72.7896	72.5343
16 (S_{MB})	72.5344	72.9777
17 (S_{MB})	72.6678	72.1223
18 (S_{MB})	72.4563	72.4241
19 (S_{MB})	72.4755	72.1342
20 (S_{MB})	72.6355	72.3453
21 (S_{MB})	75.9447	75.8978

Our algorithm used 21 frames, each of which gradually disguised a maximum of 700+ bytes, to embed all secret data in this table. The PSNR values of the frames from Naymar, Dhaka, were marginally higher than those of the other chosen video frames. For Naymar, Dhaka, 512 × 512 sized frames carrying a payload of 15 Kilobytes, or 15,000 bytes, were used in this table.

Table 2 presents the results of six quality assessment matrix values for different payloads, 512, 256, and 128 bytes for a certain frame, as well as the embedding duration of the stego frame. Here, Naymar is represented as NA and Dhaka as DA.

The analysis in this table, which yields greater PSNR values for Naymar's film, makes use of 512 × 512 video frames for NA and DA, with different payload sizes of 512 bytes, 256 bytes, and 128 bytes, respectively.

The results of a comparison between two contemporary steganographic algorithms with 512 Bytes of payload and 512 × 512 sized frames are shown in Table 3. In this table, the XOR substitution model is designated as Model1, the 8-directional based model as Model2, our suggested model as P-Model, and the

Table 2. Quality measurement metrics of the projected method using different standard sized payload.

Frame	Dimension	Payload	PSNR	SSIM	MAE	SNR	RMSE	MSE	TCEP
NA	512 × 512	512 Bytes	74.01545	0.999992913	0.0026	68.641512	0.0509	0.0026	9.47s
		256 Bytes	77.542112	0.999997723	0.0012	71.6545221	0.0352	0.0012	7.45s
		128 Bytes	80.5435132421	0.999999726	0.0006	74.84534	0.0248	0.0006	5.52s
DA	512 × 512	512 Bytes	74.215454	0.999995854	0.0026	68.32131	0.0509	0.0026	10.23s
		256 Bytes	77.45341112	0.999999646	0.0012	70.945335	0.0351	0.0012	8.93s
		128 Bytes	80.5645341	0.999999867	0.0006	74.537865435	0.0249	0.0006	5.45s

Table 3. Comparison among 2 recent steganographic techniques.

Techniques	Frame	PSNR	SSIM	MAE	SNR	RMSE	MSE	TCEP
Model1	DV	70.7483	0.999984985	0.0029	65.4785	0.0539	0.0029	8.4545s
Model2		73.13254	0.999991956	0.0027	68.8686	0.0520	0.0027	5.9646s
P-Model		74.534131	0.999992913	0.0026	69.5991	0.0509	0.0026	9.4754s
Model1	SM	71.0966	0.999989566	0.0029	66.1299	0.0539	0.0029	8.2331s
Model2		73.878954	0.999991023	0.0026	68.2094	0.0508	0.0026	5.5413s
P-Model		74.5242	0.999995854	0.0026	69.3356	0.0509	0.0026	10.2311s

chosen frames, such as Naymar as NA and Dhaka as DA, are designated as Models.

The performance of the P-Model (Proposed Model) is superior to that of the Models 1 and 2, whose values for the NA frame were 74.534131 and 74.5242, respectively, as indicated by the first column of this table. The suggested model outperforms the current models without the TCEP result in each column. Because our suggested approach offers greater security than the current model, it requires a little bit more time to complete.

The histogram for the 512 × 512 sized cover and stego frames for the four chosen video frames above can be found in **Figure 10** and **Figure 11**.

According to the histogram's findings, there is not much of a difference between the two frames, and the change is unpredictable. This experiment, which uses the data hiding technique, demonstrates that the projected procedure performs better than similar algorithms as well.

5. Conclusion

5.1. Conclusion

This study provides an automated two-level secure data concealing solution for video steganography, which conceals secret information using robust 128-bit AES encryption in the cover movie. The aforementioned justification and logical result analysis show that the suggested steganography data concealing approach offers more security and less imperceptibility when compared to certain other current data concealing techniques. However, in a 512 by 512 cover film that runs at 25

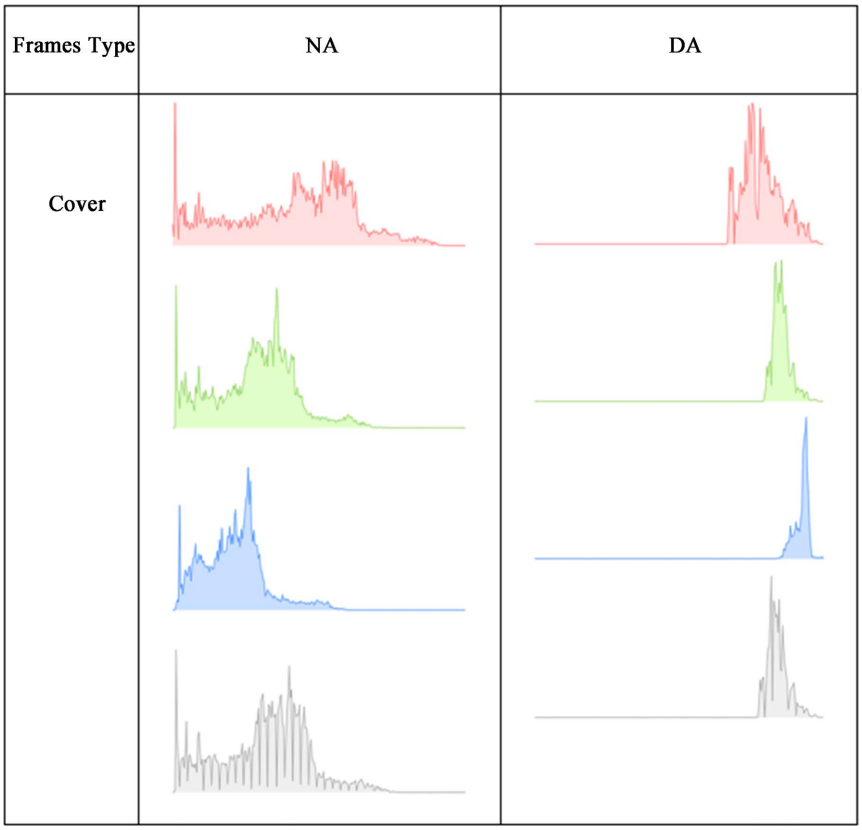

Figure 10. Comparative histogram for cover frames.

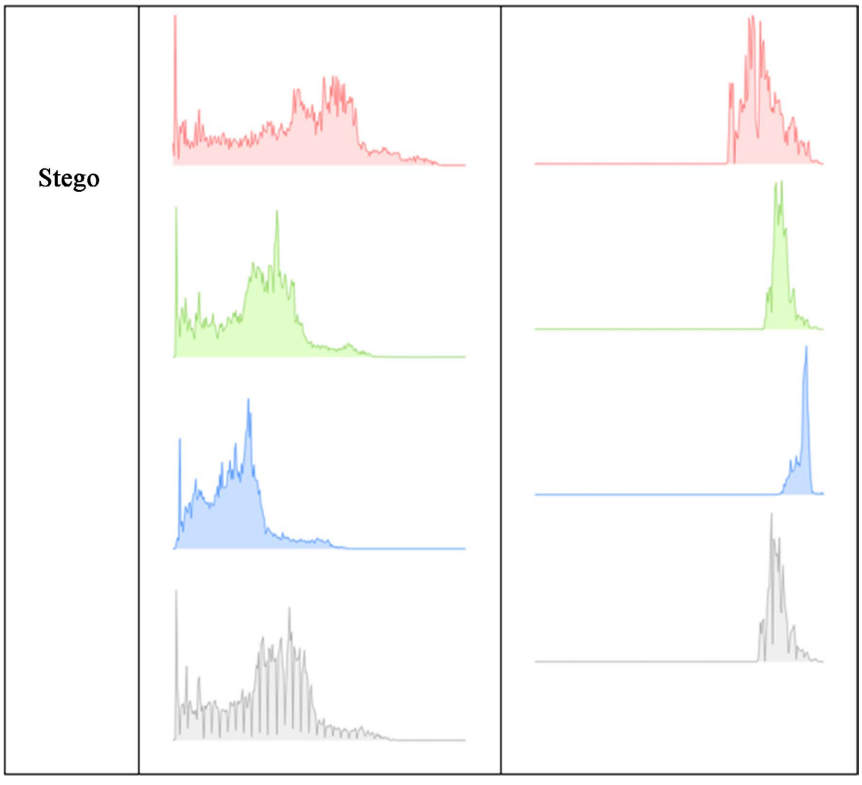

Figure 11. Comparative histogram for stego frames.

frames per second for five seconds, this model can hide up to 100KB of secret data. We will remove these restrictions in our subsequent work.

5.2. Limitations

- Proposed model is able to manipulate only.AVI video files.
- Maximum processing video's length up to 11.1111 hours long.
- Maximum processing messages' length is 714.28 Megabytes.
- Pixel selection technique depends on static mathematical calculations.

5.3. Future Works

Future steganography research should concentrate on expanding the XOR LSB-based algorithm's single-frame embedding capacity beyond its present 765 bytes limit. This might include investigating other algorithms or refining the current strategy to use pixel values more effectively without sacrificing picture quality. Using a multi-frame steganographic technique is also necessary to provide greater data payloads while preserving imperceptibility. To make the algorithm more resilient to different types of assaults, it is imperative that its security be strengthened using error correction, encryption, and anti-steganalysis techniques. In order to allay privacy worries and stop any abuse, it's crucial to address ethical issues and create rules for appropriate steganography usage.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Karthikeyan, B., Raj, M.M.A., Yuvaraj, D. and Joseph Abraham Sundar, K. (2020) A Hybrid Approach for Video Steganography by Stretching the Secret Data. In: Ranganathan, G., Chen, J. and Rocha, Á., Eds., *Inventive Communication and Computational Technologies*, Springer, 1081-1087. https://doi.org/10.1007/978-981-15-0146-3_104

[2] Patil, A., Keshkamat, S.M., Desai, V.V. and Arlimatti, T. (2018) Embedding of Advanced Encryption Standards Encoded Data in Video Using Least Significant Bit Algorithm. 2018 *International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering* (*ICRIEECE*), Bhubaneswar, India, 27-28 July 2018, 617-621. https://doi.org/10.1109/icrieece44171.2018.9009202

[3] Manohar, N. and Kumar, P.V. (2020). Data Encryption & Decryption Using Steganography. 2020 4*th International Conference on Intelligent Computing and Control Systems* (*ICICCS*), Madurai, India, 13-15 May 2020, 697-702. https://doi.org/10.1109/iciccs48265.2020.9120935

[4] Singh, N. (2019) XOR Encryption Techniques of Video Steganography: A Comparative Analysis. In: Abraham, A., Cherukuri, A.K., Melin, P. and Gandhi, N., Eds., *Intelligent Systems Design and Applications*, Springer, 203-214. https://doi.org/10.1007/978-3-030-16657-1_19

[5] Ajmera, A., Divecha, M., Ghosh, S.S., Raval, I. and Chaturvedi, R. (2019) Video Steganography: Using Scrambling-AES Encryption and DCT, DST Steganography. 2019

IEEE Pune Section International Conference (*PuneCon*), Pune, India, 18-20 December 2019, 1-7. https://doi.org/10.1109/punecon46936.2019.9105666

[6] Hashim, J., Hameed, A., Abbas, M.J., Awais, M., Qazi, H.A. and Abbas, S. (2018) LSB Modification Based Audio Steganography Using Advanced Encryption Standard (AES-256) Technique. 2018 12*th International Conference on Mathematics*, *Actuarial Science*, *Computer Science and Statistics* (*MACS*), Karachi, Pakistan, 24-25 November 2018, 1-6. https://doi.org/10.1109/macs.2018.8628458

[7] Karthikeyan, B., Deepak, A., Subalakshmi, K.S., Anishin Raj, M.M. and Vaithiyanathan, V. (2017) A Combined Approach of Steganography with LSB Encoding Technique and DES Algorithm. 2017 *Third International Conference on Advances in Electrical*, *Electronics, Information*, *Communication and Bio-Informatics* (*AEEICB*), Chennai, India, 27-28 February 2017. https://doi.org/10.1109/aeeicb.2017.7972388

[8] Qasim Ahmed Alyousuf, F., Din, R. and Qasim, A.J. (2020) Analysis Review on Spatial and Transform Domain Technique in Digital Steganography. *Bulletin of Electrical Engineering and Informatics*, **9**, 573-581. https://doi.org/10.11591/eei.v9i2.2068

[9] Alam, K., Nushrat, S., Patwary, A.H., Ullah, A. and Robin, K.H. (2023) An Improved Approach of Image Steganography Based on Least Significant Bit Technique for Secure Communication in Cloud. In: Woungang, I., Dhurandher, S.K., Pattanaik, K.K., Verma, A. and Verma, P., Eds., *Advanced Network Technologies and Intelligent Computing,* Springer, 215-233. https://doi.org/10.1007/978-3-031-28180-8_15

[10] Hossain, M.A., Ullah, A., Khan, N.I. and Alam, M.F. (2019) Design and Development of a Novel Symmetric Algorithm for Enhancing Data Security in Cloud Computing. *Journal of Information Security*, **10**, 199-236. https://doi.org/10.4236/jis.2019.104012