An Efficient DCT based Video Steganography using Dual Fisher-Yates Algorithm

Submitted By

Shanto Chaki

ID: 201-35-2993

Department of Software Engineering

Daffodil International University

Supervised By

Dr. Marzia Ahmed

Assistant Professor

Department of Software Engineering

Daffodil International University

This Thesis paper has been submitted in fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering.

Date of Submission: January 4, 2025

APPROVAL

This thesis titled on "**An Efficient DCT-based Video Steganography using Dual Fisher-Yates Algorithm**", submitted by **Shanto Chaki (ID: 201-35-2993)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS

Chairman

Professor Dr. Engr. AKM Masum
Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 1

Md. Shohel Arman
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2

Dr. Marzia Ahmed
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

External Examiner

Dr. Md. Monowarul Islam
Associate Professor
Department of Computer Science & Engineering
Jagannath University

DECLEARATION

I hereby declare that the Thesis titled **"An Efficient DCT based Video Steganography using Dual Fisher-Yates Algorithm"** has been completed by me under the supervision of Dr. Marzia Ahmed based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Daffodil International University or any other institution. This work has been prepared to fulfil a part of my academic requirements.

22.1.2025

.....

Supervised By:
Dr. Marzia Ahmed
Assistant Professor
Department of Software Engineering
Daffodil International University

.....

Submitted By:
Shanto Chaki
ID: 201-35-2993
Department of Software Engineering
Daffodil International University

AN EFFICIENT DCT BASED VIDEO STEGANOGRAPHY USING DUAL FISHER-YATES ALGORITHM

SHANTO CHAKI

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Bachelor of Science

Department of Software Engineering (Major in Cyber Security)

DAFFODIL INTERNATIONAL UNIVERSITY

JANUARY 2025

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to my respected thesis supervisor, Dr. Marzia Ahmed, Assistant Professor in the Department of Software Engineering at Daffodil International University. Her steadfast support, expert guidance, and thoughtful insights have been pivotal in every stage of this research. Her commitment to academic excellence and her dedication to fostering the intellectual development of her students have been a continual source of motivation and inspiration.

I extend my heartfelt appreciation to the esteemed faculty members of the Department of Software Engineering. Their profound knowledge and unwavering support have played a significant role in broadening my perspectives and offering invaluable guidance during this study. The academic atmosphere they have fostered has been instrumental in nurturing my ideas and enhancing the quality of my research.

I would also like to express my deep gratitude to my friends and family for their steadfast support and encouragement. Their patience, understanding, and unwavering belief in my abilities have been a constant source of strength, enabling me to overcome the challenges encountered during this research journey. Their support has been instrumental in motivating me to strive for academic excellence.

This research, from its inception to its conclusion, has been entirely my individual effort. Each aspect, from the formulation of ideas to the analysis and final conclusions, reflects my dedication and determination to contribute to the field of Steganography. The thesis, titled "An Efficient DCT-based Video Steganography using Dual Fisher-Yates Algorithm," stands as a testament to my commitment to advancing knowledge in this domain.

In conclusion, I extend my heartfelt gratitude to everyone who has, directly or indirectly, contributed to this academic journey. Your support and encouragement have profoundly influenced both this research and my personal and academic development.

Abstract

Video steganography has become a vital area in information security, allowing the hiding of confidential information within multimedia files. in this paper, A Dual Fisher-Yates algorithm is proposed to enhance balance between embedding capacity, visual distortion, and increase robustness of stego videos. This proposed technique selectively embeds the secret data into the mid-frequency DCT coefficients of video frames to mitigate the perceptibility. permutation technique ensures the embedding pattern is randomized to enhance enhancing security. The algorithm automatically adjusts the embedding capacity based on frame complexity to ensure flawless stego video content. Our approach provides superior robustness against common video processing attacks along with best embedding capacity and visual quality with an average PSNR exceeding 40 dB which is known for better quality when compared to existing methods. The proposed approach represents a significant improvement in the trade-off between capacity, imperceptibility, and robustness in video steganography.

Keywords: video steganography; compressed domain; imperceptibility; embedding payload; robustness; DCT

Table of Content

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Background

The term "steganography," which originates from the Greek words "steganos" (covered) and "graphia" (writing), represents the technique of hiding data in an apparently meaningless medium. Steganography is an important method for secure communication since it conceals the message's their existence, in contrast to cryptography, which secures data by converting it into unreadable formats. Steganography's central objective is to ensure the confidentiality of the hidden information by making sure that it is invisible to unauthorized spectators.

The application of steganography has diversified from traditional approaches to digital media, including audio, video, and photographs, because of the quick development of digital technology. Because of its inherent benefits, video steganography has been the most popular solution among these. compared to images and audio, videos have a significantly higher payload capacity, allowing the embedding of more important data without sacrificing an appearance of quality. Furthermore, videos' dynamic nature, which consists of several frames which offers an additional level of complexities and detection difficulty.

Transform-domain techniques, specifically those centered around the Discrete Cosine Transform (DCT), are becoming more popular in today's digital steganography. Because of its capacity for focusing signal energy into a small number of low-frequency components, DCT based technique is frequently used in image and video compression in different formats such as JPEG, MPEG, and H.264. Because the changes in high-frequency components are not as noticeable to the human eye, so it is ultimately the perfect domain for embedding hidden data and by employing DCT's in video steganography it might achieve an acceptable balance between capacity and imperceptibility.

The applications of modern methods like Fisher-Yates, which initially emerged for random permutation, have been founded for steganography to enhance its security and effectiveness of embedding data. By randomly selecting the position of data embeddings, it reduces the potential patterns to picked up by ordinary steganalysis techniques. A dual Fisher-Yates technique has been implemented to increase this unpredictability for potential data even further, adding an additional level of protection to making the secret information more difficult to extract or manipulate.

In this thesis, an innovative approach is combining the beneficial features of DCT-based video steganography with the enhanced safety offered by the dual Fisher-Yates algorithm is developed. The goal of the study is to help create a strong and effective steganographic approach that can meet the demands of contemporary secure communication by overcoming issues like a lack of effectiveness visibility, and attack resistance.

1.2 The Origins of Steganography

since the dawn of time, steganography the technique of concealing information has been carried out. The foundation of it is the desire of humans to communicate secretly for the purpose to safeguard sensitive messages concealed. Steganography serves to hide a message's their existence, as in contrast to cryptography, which secures a message's information.

Ancient Greece has been associated with one of the first known applications of steganography. A Greek king named Histiaeus is said to have shaved a slave's head, and inked a secret message onto their scalp, and then sent the slave to communicate with intended recipient. For this after waiting for the slave’s hair to regrow the slave is to send and then the recipient's side the slave head had to be shaved in order to reveal the secret communication to intended recipient.

Across mediaeval Europe, secret communication was commonly concealed by complex codes or integrated into sacred literature. Steganography was essential to surveillance during World War II, when microscopic photos of papers named microdots

were hidden within stamps or letters. This made it possible for agents to communicate crucial information without raising red flags.

Following the arrival of the use of photography, the steganography kept on improving to allowing it possible to use techniques like watermarking to embed secret data within images. Steganography has expanded into the realm of digital information with the expansion of digital media, taking advantage of the tremendous storage capabilities of videos, music, and photos, etc.

Modern steganography has been made possible by these historical improvements over the time, which also acted as motivation for the improvement of secret and secure communication nowadays.

1.3 Evaluation of Video Steganography over time

Over time, the field of video steganography has gone through significant evolution. The majority of early approaches focused on spatial domain strategies, which include replacing the Least Significant Bit (LSB), which needed modifying pixel values directly. These techniques were simple to implement, but they were quite vulnerable to attacks and distortions produced on by processing or compression of the video.

The security and imperceptibility of video steganography significantly improved with the advent of transform domain techniques as Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), and Discrete Fourier Transform (DFT). By embedding data in the frequency coefficients of video frames, these techniques reduce the visibility of the hidden information and increase its resistance to steganalysis tools.

For the purpose of to enhance data embedding and recovery, recent developments have presented hybrid approaches that combine machine learning techniques with spatial and transform domain techniques. Considering these developments, issues like establishing a balance between robustness, imperceptibility, and embedding capability continue to be at the center of video steganography research.

When the term assessing comes for the effectiveness of video steganography techniques, it is very common practice to evaluate metrics like embedding capacity, robustness, and imperceptibility. The quality of the steganographic video is frequently evaluated through metrics such as the Structural Similarity Index (SSIM), Mean Squared Error (MSE), and Peak Signal-to-Noise Ratio (PSNR). While embedding capacity estimates how much data may be safely concealed within the video, robustness is determined by evaluating the embedded data's adaptability to various types of attacks and transformations.

Overall, the evolution of video steganography has led to more secure and efficient methods, capable of withstanding modern detection techniques and providing reliable means of covert communication.

1.4 History of Video Steganography

Video steganography, a subset of steganography, has evolved significantly over time. Steganography itself dates back to ancient times, with early examples including messages hidden in wax-covered tablets and tattoos on messengers' scalps. The term "steganography" comes from the Greek words "steganos" (hidden) and "graphia" (writing). With the advent of digital technology, steganography transitioned into the digital realm, allowing data to be hidden within digital media such as text, images, audio, and video files. Video steganography leverages the large data capacity of video files to embed secret information across multiple frames, making detection more challenging. Early digital steganography methods focused on spatial domain techniques, such as Least Significant Bit (LSB) replacement, which directly modified pixel values. However, these methods were vulnerable to compression and processing attacks. The development of transform domain techniques, such as Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT), marked a significant improvement. These methods embed data in the frequency coefficients of video frames, enhancing security and imperceptibility. Recent advancements include hybrid approaches that combine spatial and transform domain techniques, as well as the incorporation of machine learning to optimize data embedding

and retrieval. These innovations have made video steganography more robust and secure, capable of withstanding modern detection techniques.

1.5 Application of Video Steganography

Video steganography has a wide range of applications in various fields due to its ability to conceal information within Video. Here are some of the key applications:

Military and Defense: Video steganography is used to create secure communication channels for transmitting confidential information during military operations. By embedding sensitive data within video files, it ensures that the presence of the message remains hidden from adversaries, providing an additional layer of security beyond traditional encryption methods.

Healthcare: In the healthcare sector, video steganography helps protect sensitive patient records, including medical imaging and reports. Embedding patient data within video files ensures that the information is securely transmitted and stored, reducing the risk of unauthorized access and data breaches.

Media and Entertainment: Video steganography is employed to prevent piracy and protect intellectual property through copyright watermarking. By embedding copyright information and ownership details within video files, content creators can track and authenticate their work, reducing the risk of unauthorized distribution and use.

Financial Sector: Financial institutions use video steganography to ensure the secure transmission of financial documents and transaction data. By hiding sensitive information within video files, it becomes more difficult for cybercriminals to intercept and access the data, enhancing the overall security of financial communications.

Secure Communication: Video steganography provides covert communication channels in environments where traditional encryption methods might attract attention. By embedding messages within video files, individuals can communicate securely without raising suspicion, making it an effective tool for confidential exchanges in sensitive situations.

Forensic Investigation: In forensic investigations, video steganography is used to embed metadata within videos for tracking and authentication purposes. This metadata can include information about the origin, date, and time of the video, helping investigators verify the authenticity of the footage and trace its source.

1.6 Video Steganography Classifications

Video steganography techniques can be broadly classified into several categories but Based on the domain method there are main two categories:

Spatial Domain Methods:

These techniques involve directly modifying the pixel values of video frames to embed secret data. The most common method in this category is the Least Significant Bit (LSB) replacement, where the least significant bits of pixel values are altered to hide information. While spatial domain methods are simple to implement and offer high data capacity, they are vulnerable to compression and steganalysis attacks.

Transform Domain Method

Transform domain techniques embed data in the frequency coefficients of video frames, making the hidden information less noticeable and more resilient to attacks. Common transform domain methods include:

- **Discrete Cosine Transform (DCT)**:

  Embeds data in the DCT coefficients of video frames, often used in JPEG compression.

- **Discrete Wavelet Transform (DWT):**

  Embeds data in the wavelet coefficients, providing better imperceptibility and robustness.

- **Discrete Fourier Transform (DFT):**

  Embeds data in the Fourier coefficients, offering high robustness against various attacks.

The other categories are:

- **Hybrid Methods:**

  Hybrid methods combine spatial and transform domain techniques to leverage the advantages of both approaches. These methods aim to improve robustness,

capacity, and imperceptibility simultaneously. For example, a hybrid method might use LSB replacement in the spatial domain and DCT embedding in the transform domain to achieve better performance.

- **Cryptographic Methods:**

Cryptographic methods involve encrypting the secret data before embedding it in the video. This adds an additional layer of security, making it harder for unauthorized parties to detect and extract the hidden information. Combining cryptography with steganography provides a dual-layered defense mechanism against potential threats.

- **Hybrid Methods:**

Hybrid methods combine spatial and transform domain techniques to leverage the advantages of both approaches. These methods aim to improve robustness, capacity, and imperceptibility simultaneously. For example, a hybrid method might use LSB replacement in the spatial domain and DCT embedding in the transform domain to achieve better performance.

1.7 Challenges in Video Steganography

Video steganography, while powerful, faces several challenges that researchers and practitioners must address to ensure its effectiveness and security:

1. Robustness to Processing: Videos are often subjected to various processing operations such as compression, resizing, and format conversion. These operations can distort or destroy the embedded data, making it crucial to develop steganographic methods that are robust against such manipulations.
2. Trade-Off Between Capacity and Imperceptibility: Embedding large amounts of data within a video can lead to perceptual artifacts, compromising the quality of the cover video. Striking a balance between high embedding capacity and maintaining video quality is a significant challenge.
3. Vulnerability to Steganalysis Tools: Advanced steganalysis techniques are designed to detect and extract hidden data from steganographic media. Ensuring

that the embedding method incorporates adequate randomness and complexity is essential to evade detection by these tools.

4. Computational Complexity: Transform domain methods, while offering better security and imperceptibility, can be computationally intensive. This complexity can make real-time applications challenging, necessitating the development of efficient algorithms that balance security and performance.
5. Security Concerns: Ensuring that the embedded data cannot be extracted or modified by unauthorized users is a critical concern. Robust encryption and embedding techniques are required to protect the hidden information from potential threats.

Addressing these challenges is crucial for advancing the field of video steganography and developing methods that are both secure and practical for real-world applications.

1.8 Problem Statement

In the rapidly evolving field of digital communication, ensuring the secure transmission of data has become increasingly critical. Video steganography, a technique that embeds secret information within video files, offers a promising solution to this challenge. However, current video steganography techniques struggle to balance high embedding capacity with minimal impact on video quality. Additionally, many methods lack robustness against attacks like compression and fail to adapt across different video formats, limiting their practical use in secure communication. There is a pressing need for an approach that improves capacity and security while maintaining high video quality. On the other hand, In video steganography, the paper does not provide a comprehensive evaluation or analysis of the proposed algorithm's performance or effectiveness in terms of hiding capacity and ensuring the robustness.

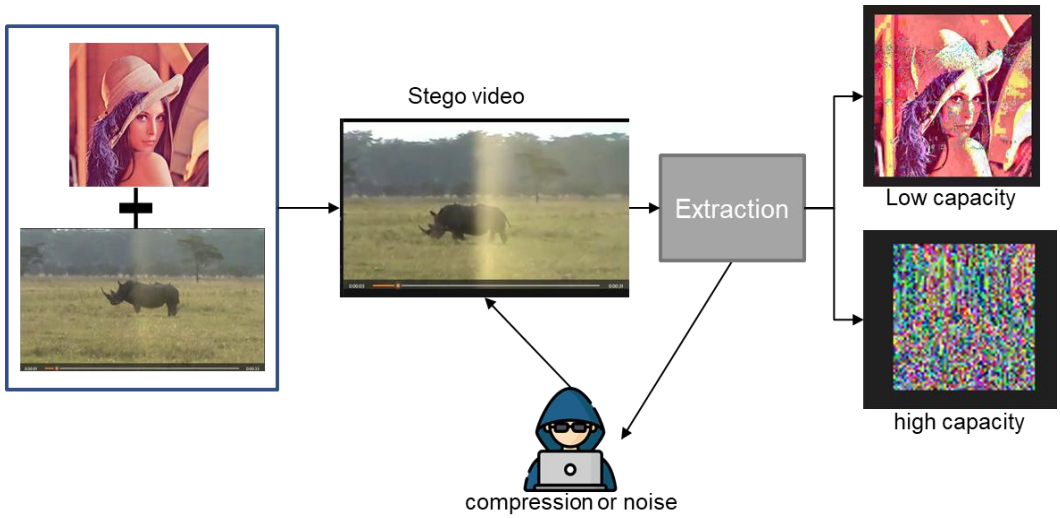

Figure 1.1: Existing Problem

PR 1. Balancing Embedding Capacity and Video Quality

One of the primary challenges in video steganography is achieving a high embedding capacity without compromising the quality of the cover video. Embedding large amounts of data can lead to perceptual artifacts, which degrade the visual quality of the video and make the hidden data more detectable. This problem is exacerbated by the need to maintain imperceptibility, ensuring that the presence of the hidden data remains undetectable to unintended observers. Current techniques often struggle to strike a balance between these competing demands, resulting in either low embedding capacity or noticeable degradation in video quality. Addressing this challenge requires the development of innovative methods that can embed large amounts of data while preserving the visual integrity of the video.

PR 2. Robustness Against Video Processing and Steganalysis

Another significant challenge in video steganography is ensuring the robustness of the embedded data against various video processing operations and steganalysis attacks. Videos are frequently subjected to compression, resizing, format conversion, and other processing operations that can distort or destroy the hidden data. Additionally, advanced

steganalysis techniques are designed to detect and extract hidden information, posing a threat to the security of the steganographic method. Many existing techniques lack the robustness needed to withstand these challenges, limiting their practical use in secure communication. Developing methods that can maintain the integrity of the embedded data under various processing conditions and evade detection by steganalysis tools is crucial for enhancing the security and reliability of video steganography.

1.9 Objectives of the Study

The primary objectives of this research are as follows:

Research Objective 1: To Develop a Robust Permutation-Based DCT Algorithm- This objective focuses on creating a novel algorithm that leverages the Discrete Cosine Transform (DCT) and permutation techniques to enhance the security and imperceptibility of video steganography. The proposed algorithm aims to introduce multiple layers of randomness, making it more challenging for unauthorized parties to detect and extract hidden information.

Research Objective 2 To Optimize Data Embedding Efficiency to Maximize Capacity While Preserving Video Quality- This objective aims to strike a balance between high embedding capacity and maintaining the visual quality of the cover video. By optimizing the data embedding process, the research seeks to ensure that large amounts of data can be securely hidden within video files without causing perceptual artifacts or noticeable degradation in video quality.

1.10 Scope of the Study

The scope of this study is defined by the following key areas:

1. **Algorithm Development**: This research focuses on developing a robust permutation-based Discrete Cosine Transform (DCT) algorithm for secure data embedding in videos. This study will not explore other transform techniques such as Discrete Wavelet Transform (DWT) or Fast Fourier Transform (FFT). The

emphasis is on leveraging the DCT method to enhance the security and imperceptibility of the embedded data.

2. **Limitation of Video Formats**: The study is limited to compressed video formats, including MPEG-1, MPEG-2, MPEG-4, H.264/AVC, and HEVC/H.265. Uncompressed video formats such as RAW and YUV are not within the scope of this research. The focus is on ensuring the proposed method's effectiveness and efficiency in commonly used compressed formats.
3. **Resistance to Steganalysis**: The research will test the algorithm's resistance to common steganalysis attacks, ensuring that the embedded data remains secure against basic detection methods. However, the study will not delve into the exploration of highly advanced AI-based steganalysis attacks. The goal is to establish a baseline of robustness against typical steganalysis techniques.

This scope outlines the boundaries of the study, ensuring a focused and detailed exploration of the proposed video steganography method. If you need more details or have any specific questions, feel free to ask!

1.11 Contribution

5. When Stego-Video is ready, its defense mechanism will be verified using Statistical Attack.

1.12 Thesis Organization

This thesis is organised into four separate chapters, each of which has a particular function within the larger story. By providing a summary of the background, goals, and importance of the study to the reader, the first chapter acts as an introduction, laying the groundwork for the following research. In order to identify gaps in the current body of research and lay the groundwork for further investigation, Chapter 2 undertakes a thorough assessment of the literature, exploring important studies and theories. The study project's methodology are thoroughly covered in Chapter 3, which includes information on the methods, approaches, and instruments utilised for data collecting and analysis. With an emphasis on interpretation and outcomes, chapter four is devoted to evaluating the

collected data and coming to insightful conclusions. Finaly the last chapter presents the thesis's conclusion, summarizing findings, discussing their implications, and proposing potential avenues for further research. This framework ensures a coherent and logical flow of information, facilitating a thorough understanding of the conducted research.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

The increasing demand for efficient and secure data hiding methods has led to significant advances in the field of video steganography. This overview of the literature examines the corpus of work that has already been done in the area, emphasizing important approaches, difficulties, and developments that have influenced it. To address the trade-offs between embedding capacity, security, and visual quality, a variety of solutions have been developed, ranging from transform domain methods like Discrete Cosine Transform (DCT) to spatial domain approaches like Least Significant Bit (LSB) substitution.

This review explores the development of video steganography, looking at both conventional and modern techniques. It assesses their advantages and disadvantages in terms of data payload capacity, robustness, and imperceptibility. The paper also emphasizes how encryption and randomness might improve security against sophisticated steganalysis attacks. This chapter lays the groundwork for the suggested dual-layer Fisher-Yates shuffling technique, which attempts to increase the robustness and embedding efficiency of video steganography systems by finding holes in the current literature.

2.2 Literature reviews

(Alam, Zakariya, & Akhtar, 2014) presents a modified Triple-A steganography method that uses the Fisher Yates algorithm for random pixel selection to improve data concealment in RGB photos. The science of inserting private information into cover media without causing any noticeable alterations in order to hide the message's existence is known as steganography. To address the shortcomings of conventional techniques such as Least Significant Bit (LSB), the authors provide a novel method that permits a configurable number of bits to be stored in each color channel of a pixel. According to the study, there are two primary branches of information hiding techniques: watermarking, which safeguards copyright, and steganography, which concentrates on hiding communications.

By applying the Fisher method and reading image pixels in a helical order, the modified Triple-A algorithm adds unpredictability. The modified Triple-A algorithm adds randomness by employing the Fisher-Yates shuffle to generate a random permutation for pixel selection and reading picture pixels in a helical order. Then In comparison to the initial Triple-A approach the tests showed that higher Peak Signal to Noise Ratio (PSNR) values are delivers from the applied technique, indicating improved image quality and enhance its safety. After that here advising for future research directions, such as applying the Fisher Yates algorithm to other steganographic approaches, the authors conclude that the improved Triple-A method are really increases the security of hidden messages. All things considered, this work advances the field of information hiding by offering an improved and more secure method to embed data in digital images.

(Souma Pal and 2, June, 2016) categorizes video steganography techniques into spatial and frequency domain methods. Spatial domain techniques are including the Least Significant Bit (LSB) coding and matrix embedding, which focus on embedding the secret data without perceptible any distortion. There the performance of these systems is evaluated based on the capacity, which refers to the amount of data that can be hidden on the payload, and imperceptibility, which indicates the invisibility or maximum similarity of the hidden data. On the other hand, Frequency domain methods have some sub categories such as Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT), are works on noted for their robustness against compression and manipulation attack. The review is also highlighting the challenges of embedding data in compressed video streams, emphasizing the need for techniques that maintain the integrity of the embedded message during its compression. Additionally, it discusses the use of neural networks in steganography, which can enhance the detection and embedding processes. Overall, the review underscores the importance of balancing capacity, imperceptibility, and security in steganographic methods

(Mstafa, Elleithy, & Abdelfattah, Video steganography techniques: Taxonomy, challenges, and future directions, 05-05 May 2017) focusing on various techniques categorized into compressed and raw domains. Compressed domain methods leverage into the video encoding stages, such as intra and inter-frame prediction techniques, motion

vectors, and transform coefficients, to embed secret data effectively. However, many of these methods exhibit limitations, particularly in embedding capacity and robustness against attacks. For instance, these techniques utilizing that the motion vectors often have low embedding capacities, which can hinder their effectiveness in practical applications. Moreover, while raw domain methods, including spatial and transform domain techniques it provides a higher embedding capacity but they may compromise video quality and robustness. The existing literature also lacks comprehensive survey articles that synthesize these advancements, making it challenging for researchers to identify gaps and opportunities for improvement. The author advice for the future work should focus on developing steganographic methods that balance video quality, embedding capacity, and robustness against various attacks. Additionally, integrating steganography with cryptography and error-correcting codes could enhance the security of that system. Researchers are also encouraged to explore the region-of-interest (ROI) techniques for targeted data embedding, which may improve the efficiency and effectiveness of that technique. Overall, while significant progress has been made, addressing these limitations and exploring new methodologies will be crucial for advancing the field of video steganography.

(Venugopal, Ranganathan, V.Velmurugan, & TadesseHailu, 2020) highlights various techniques and methodologies aimed at enhancing data security through the concealment of information within multimedia files. Least Significant Bit (LSB) insertion is a popular technique that minimizes noticeable changes to the original material by allowing data to be embedded within a video file's least significant bits. Because video files are so huge, they can hold large amounts of hidden data without exhibiting any discernible distortion, making this technology especially useful. With an emphasis on preserving the integrity of the original media, existing techniques frequently include several steps, such as image encryption, data embedding, and data extraction. There are still issues, though, like the inability to manage massive data sets and the shortcomings of the systems in place for reversible data concealment in video formats. Recent advancements propose a modified Convolutional Neural Network (CNN) for steganalysis, which aims to improve the detection of hidden messages by utilizing larger convolution filters and fewer layers, thus enhancing the system's ability to process larger images and lower payloads. This approach

not only addresses the shortcomings of previous methods but also offers a more generalized framework for analyzing steganographic techniques across various media types. Furthermore, the integration of double coding mechanisms, such as pseudo-random codes and Morse codes, has been suggested to bolster the security of the hidden data, making it more resilient against unauthorized access. The proposed systems also emphasize the importance of performance metrics like Peak Signal to Noise Ratio (PSNR) and Mean Square Error (MSE) to evaluate the quality of the stego media, ensuring that the hidden information can be extracted with minimal loss. In conclusion, the ongoing research in video steganography continues to evolve, focusing on enhancing security, efficiency, and the ability to handle larger datasets while maintaining the quality of the original media.

(Rachna Patel, 2021) demonstrates a range of methods that designed for the improve of secret data security when it is being transmitted via unprotected medium or networks. There are different techniques have been investigated, among these each having unique properties and uses, including as steganography, watermarking, and cryptography. In order to achieve PSNR values between 42.66 and 45.67 dB for example, presented a Hash-based LSB (HLSB) method that enhances conventional LSB techniques by hiding 8-bits of secret data within the RGB cover frames' least significant bits. Furthermore, Bhole et al. presented a data hiding method that makes use of LSB with randomization, however, because it was implemented in the spatial domain, it was observed to be less robust. The review also emphasizes the efficacy of transform domain methods, including those that employ Discrete Wavelet and Discrete Cosine Transform (DCT). and Discrete Wavelet Transform (DWT), which provide better robustness against various attacks while maintaining a reasonable embedding capacity. Overall, the comprehensive analysis of these methods underscores the ongoing advancements in video steganography, aiming to balance imperceptibility, robustness, and embedding capacity.

(Meenu Suresh1, 2018) The literature on video steganography highlights various techniques and approaches aimed at enhancing data security and quality. Conventional techniques frequently alter the least significant bits (LSB) of pixel values to hide information by embedding data in the spatial domain. For example, a simple yet efficient solution to conceal secret data is presented in [4], which substitutes the LSB of each color

channel. It has been demonstrated that video quality is better preserved when data is embedded in the transform domain, for example, by employing Discrete Cosine Transform (DCT). This is due to the fact that converting video frames into frequency coefficients enables more reliable data concealment without materially compromising the video's visual integrity [7], which enhances the effectiveness of the steganography process. Additionally, techniques that utilize random integer generation for shuffling data positions have been introduced to increase security against potential attacks2. For example, the proposed method in this paper employs random integer generation in the DCT domain, which not only improves security but also maintains good video quality, achieving a PSNR range of 33 to 35 dB. Overall, the evolution of steganography techniques reflects a growing emphasis on balancing data capacity, security, and the preservation of media quality in the digital age.

(A SECURE BLOCK PERMUTATION IMAGE STEGANOGRAPHY ALGORITHM, September 2013) outlines several strategies and developments in the steganography sector. Emphasizing the need for a key for safe communication systems, it starts by describing the fundamental elements of a steganography system, such as the secret, cover media, and stego media. The review divides steganography techniques into four primary categories, such as creation, injection, substitution, and Least Significant Bit (LSB). Notably, the LSB technique, which substitutes portions of the secret message for the least important parts of the cover file is popular since it is straightforward. The significance of steganalysis are described which seeks to uncover concealed signals in stego media and calls for the creation of more secure steganography methods is also covered in the paper. Several studies are referenced, such as Masud Karim et al, who enhanced LSB techniques to improve security by encrypting hidden information before embedding it. El-Emam et al. introduced a method using adaptive image segmentation to conceal large amounts of data while maintaining visual quality. Furthermore, the review mentions the integration of cryptography with steganography to bolster security, as seen in the work of Narayana and Prasad, which combines both techniques to encrypt and hide messages effectively. Overall, the literature underscores the ongoing evolution of steganography methods and the need for innovative solutions to address emerging security challenges.

(Vivek Kapoor, An Enhanced LSB based Video Steganographic System for Secure and Efficient Data Transmission, 2015) reveals a variety of methods primarily based on the spatial domain, particularly the Least Significant Bit (LSB) insertion technique. For instance, one robust video steganography technique combines LSB insertion with AES (Advanced Encryption Standard) encryption, enhancing the security of hidden information by implementing 1-bit, 2-bit, and 3-bit LSB insertion methods. Tri-pixel-value-differencing (TPVD), a compressed video steganography technique with great imperceptibility and capacity, is introduced in another study with a focus on MPEG video formats. Furthermore, BPCS (Bit Plane Complexity Segmentation) concentrates on data insertion in areas of blurred images, whereas EzStego is a system that embeds data into GIF images. Steganography and cryptography integration have also grown in popularity, with techniques using both public and private key systems to protect data. All things considered, the developments in video steganography underscore the continuous research endeavors to enhance data concealment strategies, stressing the necessity of approaches that guarantee both imperceptibility and capacity while preserving the original media's integrity.

(Tarik Idbeaa1, 2016) High-quality video streaming has been made easier by the quick development of digital communication technology, but this has also brought up issues with data security and privacy during transmission. To protect digital content, researchers have investigated a number of cryptographic techniques, including the Data Encryption Standard (DES) and Advanced Encryption Standard (AES). However, because of the inherent characteristics of images and the difficulties in distributing keys, these techniques frequently fail when applied to digital images. As a result, data hiding strategies specifically steganography have become effective means of disguising private information in cover media. Several steganographic methods have been proposed for compressed video, including least significant bit insertion (LSB) and pixel value differencing (PVD), which have shown varying degrees of effectiveness in maintaining video quality while embedding data. For instance, the BPCS technique has been noted for its minimal alteration of video quality, while the EPVD method has demonstrated improved performance over its predecessor. However, many existing techniques struggle with payload capacity and video quality degradation, particularly in high-motion sequences. To address these

challenges, the proposed Embedding-Based Byte Differencing (EBBD) scheme aims to enhance data embedding in both intra and inter frames of MPEG-2 videos, achieving a balance between embedding capacity and video quality. This approach not only preserves the integrity of the video but also ensures robust data security through encryption and careful coefficient selection.

(Tanveer J. Siddiqui, 2020) has evolved significantly particularly with the advent of digital communication technologies that have increased the volume of multimedia content online. Siddiqui and Khare (2020) present a chaos-based video steganography method utilizing the Discrete Cosine Transform (DCT) domain, which enhances security through a randomized frame selection process and the pre-treatment of secret data using the Arnold Cat map. This method embeds secret data within the mid-band DCT coefficients, employing two pseudo-random sequences generated from a chaotic map, thereby improving robustness against various noise attacks. The authors emphasize the importance of imperceptibility and security in steganography, noting that the hidden data should not be detectable while maintaining acceptable video quality. Previous research has primarily focused on still images and text, but the increasing availability of digital video has made video steganography a critical area of study. Various techniques have been explored, including Least Significant Bit (LSB) methods and transform domain techniques, with the latter being more robust against attacks. The proposed method distinguishes itself by using block-based embedding rather than LSB, which enhances security and reduces the risk of data loss due to frame deletion. The performance of the algorithm is evaluated using metrics such as Peak Signal to Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Video Quality Metric (VQM), demonstrating its effectiveness in maintaining video quality while ensuring high data recovery rates under noise attacks. The results indicate that the proposed method can recover over 90% of the hidden data even under challenging conditions, showcasing its potential for practical applications in secure video communication. Overall, this research contributes to the growing body of knowledge in video steganography, highlighting the need for innovative approaches to enhance data security in digital media.

(Ajmera, Divecha, Ghosh, Raval, & Chaturvedi, 2019) The paper "Video Steganography: Using Scrambling-AES Encryption and DCT, DST Steganography" introduces a novel method for embedding secret data or information within video files. This technique leverages a combination of scrambling, Advanced Encryption Standard (AES) encryption, and two transform domain methods: Discrete Cosine Transform (DCT) and Discrete Sine Transform (DST). By applying DCT and DST to the cover video, the method transforms the spatial domain data into the frequency domain, making the hidden information less perceptible and more secure. Additionally, AES encryption is used to secure the message image before embedding, adding an extra layer of security. The proposed method addresses the limitations of traditional spatial domain techniques, such as the Least Significant Bit (LSB) method, which are vulnerable to compression and processing attacks. By embedding data in the frequency coefficients of video frames, the technique enhances both imperceptibility and robustness against steganalysis attacks. The paper highlights the effectiveness of this approach in maintaining video quality while securely embedding substantial amounts of data, making it suitable for applications requiring secure and covert communication. This method demonstrates significant improvements in the field of video steganography, offering a balanced solution for embedding capacity, security, and visual quality.

2.3 Research Gap

Despite significant advancements in video steganography, several critical gaps remain that need to be addressed:

1. Limited Embedding Capacity: Many existing video steganography techniques struggle to achieve high embedding capacity without compromising the quality of the cover video. This limitation restricts the amount of data that can be securely hidden within video files, reducing the practicality of these methods for applications requiring large data embedding (Elleithy & Abdelfattah, 2017).

2. Lack of Robustness: Current methods often lack robustness against various attacks, such as compression, resizing, and format conversion. These operations can distort or destroy the embedded data, making it essential to develop techniques that can withstand

such manipulations and ensure the integrity of the hidden information. (Meenu Suresh1, 2018)

3. **Degradation of Video Quality**: Embedding data within video files can lead to perceptual artifacts, degrading the visual quality of the cover video. Maintaining high video quality while embedding large amounts of data is a significant challenge that needs to be addressed to ensure the imperceptibility of the hidden information. (Sahib Khan1, 2019)

4. **Limited Format Adaptability**: Many video steganography techniques are designed for specific video formats and may not perform well across different formats. This lack of adaptability limits the practical use of these methods in diverse real-world scenarios, where videos of various formats are commonly used

Chapter 3

METHODOLOGY

3.1 Introduction

The proposed method described in this paper employs a dual Fisher-Yates shuffling approach within a DCT-based video steganography framework to enhance balancing between data-hiding capacity, visual quality, and robustness. By selectively embedding scrambled binary data of a secret image into mid-frequency DCT coefficients across shuffled video frames, this technique ensures imperceptibility and security, effectively balancing robustness and fidelity in.

Upon receiving the stego Video, the receiver extracts the hidden message by identifying and interpreting the zero-width characters embedded within the cover Video. The extracted message, which is still encrypted, is then decrypted using the same AES encryption key used by the sender. Finally, the decrypted UTF-16 encoded message is converted back to its original format, revealing the secret message. Throughout this process, the cover Video remains seemingly unchanged, ensuring that the presence of the secret message remains concealed from unintended recipients. This system effectively combines Video encoding, encryption, and steganography to achieve secure communication.

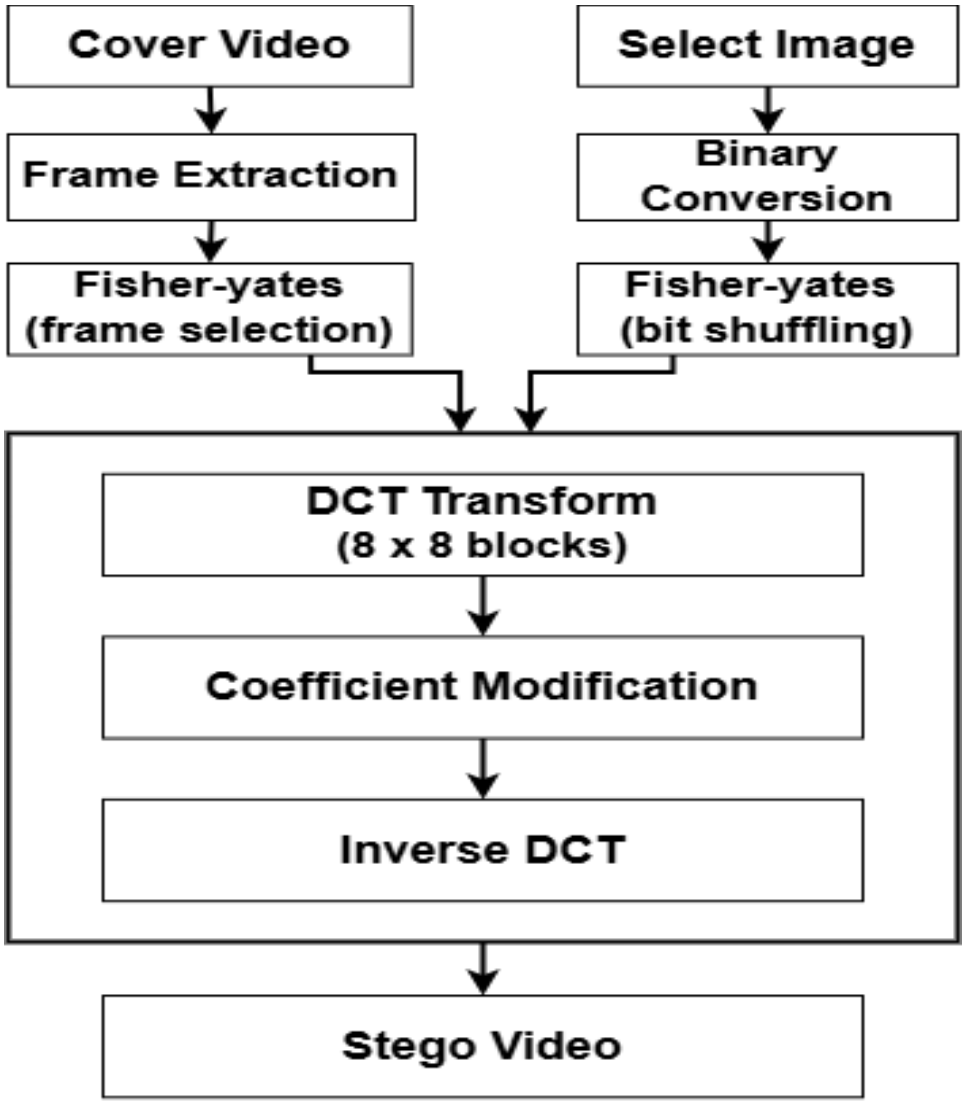

Figure 3.1: Proposed block diagram for Embedding process

Methodology

The methodology under consideration integrates a modified embedding scheme with some Phases for Embedding process:

- Input Processing Phase
- First Fisher-Yates Shuffle (Frame Selection)
- Second Fisher-Yates Shuffle (Bit Scrambling)
- DCT Transform Process

- Coefficient Modification
- Inverse DCT Transform
- Output Generation
- Cover Video Processing

3.2. Phases for Embedding Process

1. Input Processing Phase:

Cover Video Processing: The first step involves processing the cover video, which is represented as $V(t)$, where t denotes the frame sequence. Each frame is extracted as\ $F(x, y, t)$, with x and y being the spatial coordinates and t the frame index. To prepare for data embedding, the color space is converted from RGB to YUV using the formula:

$$Y = 0.299R + 0.587G + 0.114B \text{ (1)}$$

$$U = -0.147R - 0.289G + 0.436B \text{ (2)}$$

$$V = 0.615R - 0.515G - 0.100B \text{(3)}$$

Only the Y channel is utilized for embedding to maintain imperceptibility, as highlighted in the proposed method.as an encryption standard that is devised by governments organizations and industry standards.

Secret Image Processing: The secret image, denoted as $I(x, y)$ is processed by converting each pixel $P(x, y)$ into an 8-bit binary sequence. The total number of bits required for embedding is calculated as the product of the image's width, height, and the number of color channels (3 for RGB), resulting in a binary stream representation $B = \{b_1, b_2, b_3, \ldots, b_n\}$ where $b_i \in \{0, 1\}$.

2. First Fisher-Yates Shuffle (Frame Selection)

Algorithm Implementation: Here to randomly select frames for embedding the first Fisher-Yates shuffle algorithm is employed. The algorithm initializes a random

number generator (RNG) with a Shuffle_Key S_1 and iterates through the array of frame indices, then swapping the elements to achieve a random order. This process will enhance the security of data embedding by unpredictable frame selection.

Frame Selection Process: After the frame shuffled the input in this phase is an array of frame indices $F = \{1, 2, \ldots, n\}$. And the output is a shuffled frame sequence F_s. The selection criteria are based on the total number of secret bits and the number of blocks per frame of the cover video, and here ensures that the sufficient frames are selected for the embedding of confidential data.

3. Second Fisher-Yates Shuffle (Bit Scrambling)

Bit Sequence Shuffling: In this stage, the binary stream B undergoes a second shuffling through the second Fisher-Yates algorithm, initializing a different Shuffle_Key S_2. This process improves the security by spatially distributing the bits and removing the predictable patterns of bit stream, thereby increasing the resilience of the embedded data against commonly used statistical attacks.

4. DCT Transform Process

When the frame is identified for data embedding, frame is divided into non-overlapping 8x8 blocks. This partitioning is crucial as it allows for the calculation of DCT coefficients for each block, which are very essential for the embedding process.

2D-DCT Transformation: For each 8x8 block, the DCT coefficients are calculated using the formula:

$$F(\mu, \vartheta) = \alpha(\mu)\alpha(\vartheta) \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} F_y(x, y). \cos\left(\frac{\pi(2x+1)\mu}{2N}\right). cos\left(\frac{\pi(2y+1)\vartheta}{2M}\right) \quad (4)$$

Where F_y is the Y channel of the frame, N and M are the dimensions of the frame, $\alpha(\mu)$ and $\alpha(\vartheta)$ are normalization factors. And μ, ϑ represent the frequency indices. This transformation is vital for embedding the data while preserving video quality.

5. Coefficient Modification

Selection of Coefficients: In this 5th phase of embedding process it targeting the mid-frequency DCT coefficients, specifically those in the range of [2,6] for both dimensions. This selection avoids the all DC coefficient and low-frequency coefficients, which actually helps to maintain the quality of the stego video.

Embedding Formula: The embedding formula modifies the selected coefficients as follows:

$$C^{"}(\mu, \vartheta) = C(\mu, \vartheta) + (a, b_i) \text{ (5).}$$

Where $C^{"}(\mu, \vartheta)$ is the modified coefficient, $C(\mu, \vartheta)$ is the original coefficient, a is the embedding strength, and b_i is the secret bit.

Embedding Conditions: The conditions for coefficient selection ensure that only appropriate coefficients are modified, and the total capacity for data embedding is calculated based on the number of frames and blocks available.

6. Inverse DCT Transform

Block Reconstruction: After embed the secret image for the purpose of frame reconstruct the IDCT formula is applied to recover pixel values from the modified DCT coefficients, allowing for the reconstruction of the stego video frame by using

$$f(x, y) = \sum_{\mu=0}^{M-1} \sum_{\vartheta=0}^{N-1} [\, \alpha(\mu)\alpha(\vartheta) \times F'(\mu, \vartheta) \times \cos(\frac{(2x+1)\mu\pi}{2M}) \times \cos(\frac{(2x+1)\mu\pi}{2N})] \text{ (6)}$$

Where $f(x, y)$ is pixel value at position (x, y) in the spatial domain, $F'(\mu, \vartheta)$ is modified DCT coefficient at position μ, ϑ in the frequency domain, which containing the embedded data. $\alpha(\mu)\alpha(\vartheta)$ is scaling factors define as $\alpha(k) = \frac{1}{\sqrt{2}}$ otherwise $k = 0$.

This process reassembles the embedded frames by converting the altered DCT coefficients back to the spatial domain, ensuring that the embedded data is integrated subtly while preserving the visual quality of the video.

Frame Reconstruction: After applying IDCT, the blocks are integrated back into a full frame, and the color space is converted back to RGB to generate the final output video. It uses:

$$R = Y + 1.140V \quad (7)$$

$$G = Y - 0.395U - 0.581V \quad (8)$$

$$B = Y + 2.032U \quad (9)$$

7. **Output Generation**

Video Reconstruction: The final step involves assembling the frame sequence and ensuring temporal synchronization, which is crucial for maintaining the integrity of the stego video.

3.3. Phases for Extracting process

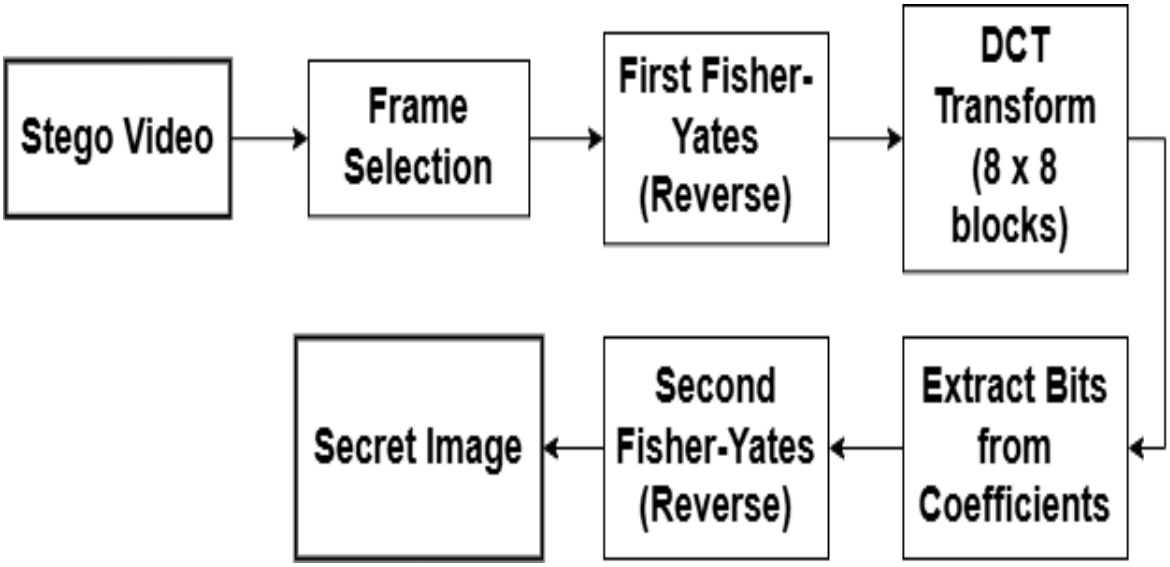

Figure 3.2: Extracting process

This methodology under consideration integrates a modified extracting scheme with some Phases for Extracting process:

- Stego Video Processing
- Bit Extraction
- Secret Image Reconstruction
- Output Generation
- Quality Consideration

1. Stego Video Processing- Frame Selection

The initial stage of the extraction procedure is to pinpoint the frames that hold the hidden secret information. This is accomplished with the identical Fisher-Yates shuffle algorithm employed in the embedding process, but on this occasion, the operation is inverted to retrieve the original frame order. The input to the reverse Fisher-Yates algorithm is the same array of frame indices $F = \{1, 2, \ldots, n\}$, and the Shuffle_Key S_1 used during the embedding process.

Algorithm steps:

step 1. Initialize random number generator with Shuffle_Key S_1

step 2. Loop from the second element down to the first:

> Select a random index j between 1 and i (inclusive)

> Swap $A[i]$ with $A[j]$

This process outputs the sequence of frame indices F_s used for embedding.

DCT Transform: The selected frames from the stego video are divided into 8×8 non-overlapping blocks, and the 2D-DCT transformation is applied to each block, as described in the embedding process:

$$F(\mu,\vartheta) = \alpha(\mu)\alpha(\vartheta)\sum_{x=0}^{N-1}\sum_{y=0}^{M-1}F_y(x,y).\cos\left(\frac{\pi(2x+1)\mu}{2N}\right).cos\left(\frac{\pi(2y+1)\vartheta}{2M}\right) (10)$$

where $\alpha(\mu) = \alpha(\vartheta) = \frac{1}{\sqrt{2}}$ for $\mu, \vartheta = 0$; otherwise $\alpha(\mu),\ \alpha(\vartheta)$. This step separates the spatial domain data into frequency domain coefficients, which are then used to extract the embedded secret bits.

2. Bit Extraction

Coefficient Analysis: Mid-frequency DCT coefficients within the range [2,\ 6] × [2,6] are analyzed to retrieve the embedded bits. For each selected coefficient $C(\mu,\vartheta)$, the bit is extracted as:

Extracted_Bit = $((C(\mu,\vartheta)\ mod\ \alpha)\ /\ \alpha)$

Where $\alpha = 0.1$ (embedding strength). Extracted bits are stored in a binary sequence.

Second Fisher-Yates Shuffle (Reverse): To undo the bit-level scrambling performed during the embedding process, the extracted binary sequence is subjected to a reverse Fisher-Yates shuffle. The input is the extracted bit sequence, and the Shuffle_Key S_2 used in the embedding process is applied. The output of this step is the recovered and de-scrambled binary sequence representing the original secret data.

3. Secret Image Reconstruction

Bit Sequence Reconstruction: The unsorted binary sequence is utilized to rebuild the original hidden image. The binary stream is split into sets of 8 bits, with each set signifying a pixel value in the RGB color model. The reconstructed pixel values are subsequently organized into a 2D image with the dimensions indicated by the original secret image.

4. Color Space Conversion

Following the bit Sequence reconstruction, the reconstructed image exists in the RGB color space. If needed, a transformation back to the original color space (such as grayscale or CMYK) can be executed to align with the format of the initial secret image.

5. **Output Generation**

The last step is to save the reconstructed secret image into a secret data file, maintaining the original image file format (e.g., .png,.jpeg).

6. **Quality Considerations**

The caliber of the obtained secret image is vital for system effectiveness, evaluated with the subsequent metrics:

6.1 Bit Error Rate (BER): Evaluates the epact ratio for erroneous bits in the retrieved data, indicating deterioration.

6.2 Structural Similarity Index (SSIM): Evaluates the structural resemblance to the initial version, reflecting both the visual quality and the perceptual similarity

6.3 Peak Signal-to-Noise Ratio (PSNR): Shows the clarity of the signal in relation to the noise, where higher values indicate improved extraction quality.

These matrices will be more discuss on result and discussion part.

3.4 Encoding and Decoding Algorithm for the model

Pseudocode for Embedding

1. Input: Cover video frames, secret image
2. Convert secret image to binary data
3. Shuffle binary data using Fisher-Yates shuffle (seed1)
4. Apply Fisher-Yates shuffle (seed2) to the cover video frames
5. For each selected frame:
   a. Convert frame to grayscale
   b. Divide frame into 8x8 blocks
   c. Compute DCT for each block
   d. Embed binary bits into DC coefficients
   e. Compute inverse DCT (IDCT) for modified blocks
6. Save the modified frames as the stego video
7. Output: Stego video

Pseudocode for the Extracting

1. Input: Stego video, seeds, secret image size
2. Apply Fisher-Yates shuffle (seed2) to determine frame order
3. For each selected frame:
   a. Convert frame to grayscale
   b. Divide frame into 8x8 blocks
   c. Compute DCT for each block
   d. Extract binary data from DC coefficients
4. Apply Fisher-Yates shuffle (seed1) to restore binary data order
5. Reconstruct the secret image
6. Output: Secret image

How to Use the Code

Encoding:

Input: Cover video, secret image, seeds.
Output: Stego video.
Run: encode_video('cover.mp4', 'lena.png', 'stego.avi', seed1=1234, seed2=5678)

Decoding:

Input: Stego video, seeds, secret image size.
Output: Extracted image.
Run: decode_video('stego.avi', (512, 512), seed1=1234, seed2=5678, 'output_lena.png')

3.5 Functionalities of Proposed model

This proposed method aims to securely transmit a secret message by embedding it within a cover Video fisher-yates algorithm. Here I break down the functionalities and the corresponding code chunks. I describe each step in detail and then provide the related code.

Initialization: The initialization step sets up the parameters for the video steganography class, such as block size and embedding strength.

- *block_size*: Defines the size of blocks used for the Discrete Cosine Transform (DCT).

- *embed_strength*: Sets the strength of the embedding, which determines how much the DCT coefficients are modified.

```
class VideoSteganography:

    def __init__(self):

        self.block_size = 8

        self.embed_strength = 0.1
```

Fisher-Yates Shuffle Algorithm: The Fisher-Yates shuffle algorithm randomizes the order of elements in a list. This is used to shuffle the video frames and bits for embedding and extraction, adding an extra layer of security.

- data: The list to be shuffled.
- seed: A seed value to ensure reproducibility of the shuffle.

```
import random

def fisher_yates_shuffle(self, data, seed):

    data_copy = data.copy()

    random.seed(seed)

    for i in range(len(data_copy) - 1, 0, -1):

        j = random.randint(0, i)

        data_copy[i], data_copy[j] = data_copy[j], data_copy[i]

    return data_copy
```

Apply DCT and IDCT to Blocks: Applying DCT transforms spatial domain data (pixel values) into the frequency domain, which helps in embedding data in a less perceptible way. The IDCT reverses this transformation.

- block: An 8x8 block of pixel values to which DCT or IDCT is applied.

from scipy.fftpack import dct, idct

def apply_dct_to_block(self, block):

return dct(dct(block.T, norm='ortho').T, norm='ortho')

def apply_idct_to_block(self, block):

return idct(idct(block.T, norm='ortho').T, norm='ortho')

Embedding Data into Video: This function embeds secret data (an image) into a video by modifying the DCT coefficients of selected video frames.

- Load the secret image and open the video.
- Calculate the required number of frames based on the image size.
- Use Fisher-Yates shuffle to randomly select frames for embedding.
- Convert the selected frames to the YUV color space and apply DCT.
- Embed the bits of the secret image into high-frequency DCT coefficients.
- Shuffle the embedded bits before saving the output video.

Extracting Data from Video: This function extracts the embedded secret image from the video by reversing the embedding process.

- Open the video and prepare for extraction.
- Use Fisher-Yates shuffle to select frames for extraction.
- Convert frames to YUV color space and apply DCT.
- Extract bits from high-frequency DCT coefficients.
- Reverse shuffle the extracted bits and reconstructs the secret image.

Helper Functions: These functions support the embedding and extraction processes by converting images to bits and embedding bits into DCT coefficients.

- *get_image_bits(image):* Converts an image to a list of bits.
- *byte_to_bits(byte):* Converts a byte to a list of bits.
- *embed_bits_into_dct(dct_block, bits):* Embeds bits into DCT coefficients.
- *save_video_with_embedded_data(output_path, embedded_bits):* Placeholder.

GUI Setup: The GUI setup provides a user-friendly interface for embedding and extracting data. It includes buttons for selecting files, starting the processes, and displaying progress.

- Embed GUI: Components for embedding data, including labels, buttons, and a progress bar.
- Extract GUI: Components for extracting data, including labels, buttons, and a progress bar.
- Functions: Functions to handle file selection and trigger embedding/extraction processes.

Embed GUI:

- Title label to indicate the embedding functionality.
- Preview frame with labels to show selected video and image files.
- Buttons for selecting video, selecting image, saving the output video, and embedding data.
- Progress bar to display the progress of the embedding process.

Extract GUI:

- Title label to indicate the extraction functionality.
- Buttons for selecting the video, saving the extracted image, and extracting data.
- Progress bar to display the progress of the extraction process.

Functions of GUI:

- *select_video():* Opens a file dialog to select a video file and updates the label with the selected file name.
- *select_image():* Opens a file dialog to select an image file and updates the label with the selected file name.

- *save_output_video():* Opens a file dialog to save the output video file.
- *select_extract_video():* Opens a file dialog to select a video file for extraction.
- *save_extracted_image():* Opens a file dialog to save the extracted image file.
- *embed_data():* Calls the *embed_data* method from the *VideoSteganography* class to embed data, updating the progress bar and showing a success or error message.
- *extract_data():* Calls the *extract_data* method from the *VideoSteganography* class to extract data, updating the progress bar and showing a success or error message.

These are the key functionalities of that video steganography application, from initializing the class and shuffling data to embedding and extracting secret data, as well as setting up the GUI.

Chapter 4

RESULTS & DISCUSSION

The suggested method for video steganography was executed and evaluated with MATLAB. Three distinct concealed images were embedded to evaluate the effectiveness of the system. Lena, Barbara and 1971, in two separate cover videos. The system's performance was assessed by concealing these images within the cover video and analyzing the results through three essential metrics. The distortion rate in the stego-image was initially assessed using Mean Square Error (MSE), which measures the differences between the original and stego images. Secondly, to assess the preservation of image quality following embedding, the embedding quality of the stego-image was evaluated in decibels (dB) through the Peak Signal-to-Noise Ratio (PSNR). Data Capacity was ultimately investigated to determine the extent of private information that could be hidden in each video frame in each Cover Video Files.

4.1 Proposed Method Input and output data:

Cover Video file

The figure 4.1 below displays two input cover video files intended for secure transfer the secret image from one location to another.

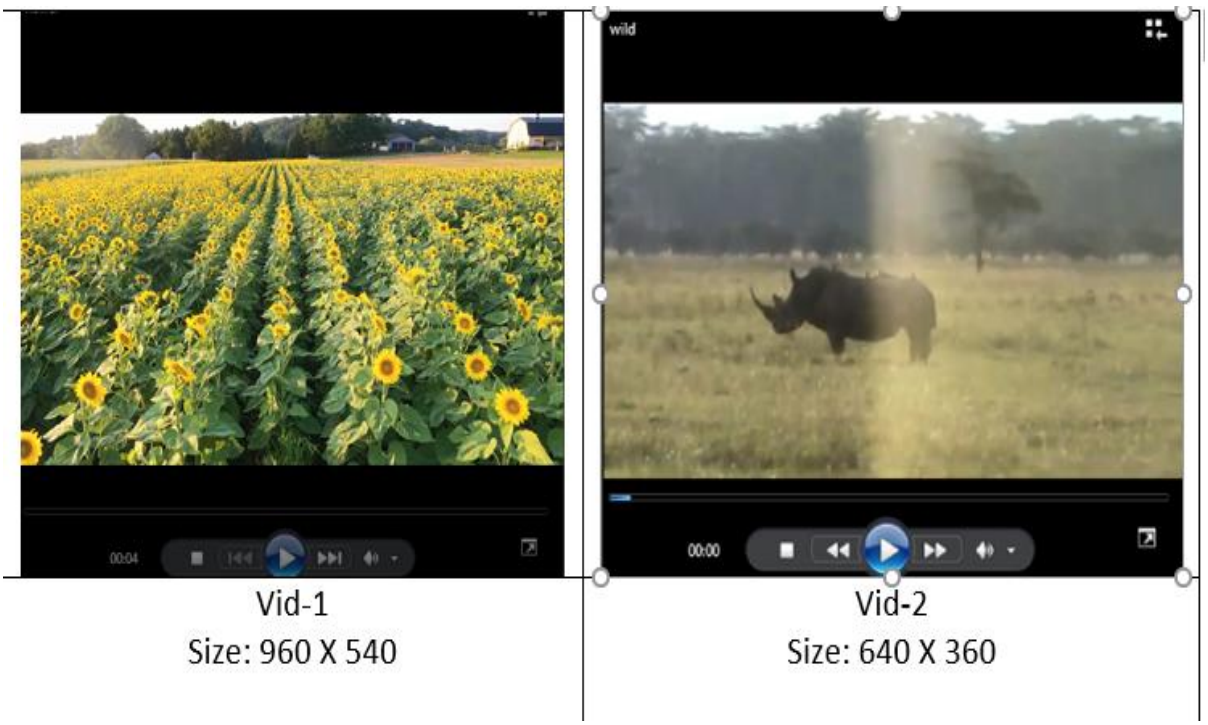

Figure 4.1: Cover Videos Sample

Secret Image files:

we evaluated the performance of the proposed method by using some test files and data. The Table 4.1 below illustrates various secret images embedded within the proposed steganography system for secure concealment and transfer

Table 4.1: Input Secret Data

		
Barbara Size: 512 X 512	1971 Size: 740 X 500	Lena Size: 512 X 512

4.2 GUI to get Embedded video:

In this video steganography process shown in the figure 4.2 below, a video and an image file are used as inputs for encoding. This allows secret image or information to be securely transferred from one user to another while it ensures the safeguard from unauthorized access.

Figure 4.2: User interface to embed secret image into cover video

4.3 GUI to Extract the stego video:

To extract the embedded image, Click the "Open Video" option in figure 4.3 to choose the stego video, which is the first step in extracting the embedded image. After choosing the stego video, all you have to do is click the "Extract" button, and the secret image will be immediately extracted.

Figure 4.3: User interface to extract secret image from stego video

4.4 Calculative Analysis

To calculate the data capacity and visual quality after embedding and extraction the used equations are:

$$PSNR = 10 \times \log_{10}(\frac{MAX^2}{MSE}) \quad (11)$$

Where MAX is the maximum pixel value (255 for 8-bit grayscale or RGB images)

$$MSE = \frac{1}{N \times H \times W} \sum_{i=1}^{N} \sum_{x=1}^{H} \sum_{y=1}^{W} (I(i,x,y) - S(i,x,y))^2 \quad (12)$$

Where $I(i,x,y)$ is pixel value in the original frame. $S(i,x,y)$ is Pixel value in the stego frame, N is the number of frames. H is Frame height and W is Frame with.

$$Total\_Capacity = N \times M \times K \quad (13)$$

Where N is for number of frames used for embedding, M is for number of 8×8 blocks per frame and K is for number of coefficients per block used for embedding (For this case, 16 coefficients from the 2nd to 6th row and column)

To start the evaluation process, we will use the two selected cover videos along with three hidden images to calculate the Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and data capacity for each combination of video and image. These metrics will provide an extensive assessment of the system's performance by analyzing the distortion

rate, embedding quality, and data capacity achieved when every image is integrated into each video. The results will aid in evaluating the effectiveness of the suggested steganography model concerning image quality maintenance and data hiding efficiency.

Result of matrices

Table 4.2: Calculation of matrices

Cover Video	Secret image	Payload (Bits)	Frames Used	Payload/Frame (bits)
Vid-1	Barbara	798,400	20	39,920
	1971	434,400	8	54,300
	Lena	527,200	10	52,720
Vid-2	Barbara	798,400	16	49,900
	1971	434,400	9	48,267
	Lena	527,200	11	51,800

The secret images used were Lena (512 x 512, 65.9 KB), 1971 (740 x 500, 54.3 KB), and Barbara (512 x 512, 99.8 KB). Lena required 527,200 bits and occupied approximately 11 frames, achieving a PSNR of 38-40 dB and a BER of 0.5-1%. 1971, with a payload of 434,400 bits, required about 9 frames, yielding a slightly better PSNR of 39-41 dB and a similar BER of 0.5-1%. Barbara, the largest image, utilized 798,400 bits and required 16 frames.

Comparison of Simple DCT method

Table 4.3: Calculation of matrices

Name of image	Shuffling of Data on Least significant DCT			Proposed approach		
	PSNR	Payload/ Frame (Bits)	BER (%)	PSNR	Payload/Frame (Bits)	BER (%)
Barbara	34.8262	45,102	1.37	44.38	49,900	0.81
1971	33.6067	42,722	2.42	42.83	48,267	0.65
Lena	34.8310	50,586	1.21	45.06	51,800	0.74

CONCLUTION

This paper presents an efficient video steganography technique based on the Discrete Cosine Transform (DCT) and the Dual Fisher-Yates algorithm. Where the secret image or data embedding in mid-frequency coefficients and Fisher-Yates shuffling approaches, the proposed model provides significant improvements in data concealing capacity, video quality preservation, and robustness. The dual shuffling technique ensures that the embedded data is securely hidden and recoverable even in the presence of noise or compression. Experiments' results demonstrate how successfully the model embeds a lot of data while maintaining high visual quality.

FUTURE WORK

Considering these developments, this study has many limitations. To improve performance even more, future studies could look into different video encoding algorithms or standard adjustments. The efficacy and usefulness of this video steganography technique can be improved with further research. To boost the embedding capacity without

sacrificing video quality, it is first vital to investigate sophisticated compression algorithms and other embedding techniques. Robustness against sophisticated steganalysis approaches will be improved by creating adaptive algorithms that can react dynamically to changing threats. Future studies should look into ways to make the suggested technique more versatile for a range of video formats, including compressed and uncompressed ones. To make the algorithm appropriate for real-time applications, its computational efficiency must be maximized using methods like hardware and parallel processing. Integrating cryptographic techniques with the embedding process can add an additional layer of security, ensuring the hidden data remains secure even if detected. Developing more intuitive and user-friendly interfaces will facilitate the use of video steganography in various applications, making it accessible to both technical and non-technical users. Finally, extensive experimental validation in real-world scenarios is necessary, testing the technique with different types of videos, various attack models, and diverse use cases to ensure its robustness and effectiveness. By addressing these areas, future work can significantly advance the field of video steganography, making it more secure, efficient, and applicable to a wider range of real-world scenarios. These enhancements will contribute to developing more robust and versatile steganographic systems capable of ensuring secure communication in increasingly complex digital environments.

APPENDICES

Appendix A. Code for the embedding

```
import cv2
import numpy as np
import random
from PIL import Image

# Fisher-Yates Shuffle
def fisher_yates_shuffle(array, seed):
    random.seed(seed)
    for i in range(len(array) - 1, 0, -1):
        j = random.randint(0, i)
        array[i], array[j] = array[j], array[i]
    return array

# Convert image to binary
def image_to_binary(image_path):
    image = Image.open(image_path).convert('L')  # Convert to grayscale
    pixels = list(image.getdata())
    binary_data = ''.join(format(pixel, '08b') for pixel in pixels)
    return binary_data

# Embed binary data into video frame
def embed_data_in_frame(frame, data_bits, block_size=8):
    h, w = frame.shape[:2]
    idx = 0

    for i in range(0, h, block_size):
        for j in range(0, w, block_size):
            if idx < len(data_bits):
                block = frame[i:i+block_size, j:j+block_size]
                dct = cv2.dct(np.float32(block))
                dct[0, 0] = int(data_bits[idx])  # Embed bit into DC coefficient
                idct = cv2.idct(dct)
                frame[i:i+block_size, j:j+block_size] = np.uint8(idct)
                idx += 1
```

```
    return frame

# Main Encoding Function
def encode_video(cover_video_path, secret_image_path, output_video_path, seed1, seed2):
    # Open video and secret image
    cap = cv2.VideoCapture(cover_video_path)
    binary_data = image_to_binary(secret_image_path)
    shuffled_binary = fisher_yates_shuffle(list(binary_data), seed1)

    frame_count = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
    shuffled_frames = fisher_yates_shuffle(list(range(frame_count)), seed2)

    fourcc = cv2.VideoWriter_fourcc(*'XVID')
    out = cv2.VideoWriter(output_video_path, fourcc, cap.get(cv2.CAP_PROP_FPS),
                          (int(cap.get(3)), int(cap.get(4))))

    data_idx = 0
    while cap.isOpened():
        ret, frame = cap.read()
        if not ret or data_idx >= len(shuffled_binary):
            break
        if cap.get(cv2.CAP_PROP_POS_FRAMES) - 1 in shuffled_frames:
            grayscale_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
            frame_with_data = embed_data_in_frame(grayscale_frame, shuffled_binary[data_idx:])
            data_idx += len(shuffled_binary[data_idx:])
            out.write(cv2.cvtColor(frame_with_data, cv2.COLOR_GRAY2BGR))
        else:
            out.write(frame)

    cap.release()
    out.release()
```

Appendix B. Decoding Algorithm

```
# Extract binary data from video frame

def extract_data_from_frame(frame, block_size=8, data_length=None):
```



```
    h, w = frame.shape[:2]
    extracted_bits = []

    for i in range(0, h, block_size):
        for j in range(0, w, block_size):
            if data_length and len(extracted_bits) >= data_length:
                break
            block = frame[i:i+block_size, j:j+block_size]
            dct = cv2.dct(np.float32(block))
            extracted_bits.append(int(dct[0, 0]))  # Extract bit from DC coefficient

    return ''.join(map(str, extracted_bits))

# Reconstruct image from binary
def binary_to_image(binary_data, image_size, output_path):
    pixels = [int(binary_data[i:i+8], 2) for i in range(0, len(binary_data), 8)]
    image = Image.new('L', image_size)
    image.putdata(pixels)
    image.save(output_path)

# Main Decoding Function
def decode_video(stego_video_path, secret_image_size, seed1, seed2, output_image_path):
    cap = cv2.VideoCapture(stego_video_path)
```



```
    frame_count = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
    shuffled_frames = fisher_yates_shuffle(list(range(frame_count)), seed2)

    binary_data = ""
    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            break
        if cap.get(cv2.CAP_PROP_POS_FRAMES) - 1 in shuffled_frames:
            grayscale_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
            binary_data += extract_data_from_frame(grayscale_frame, data_length=None)

    cap.release()
    restored_binary = fisher_yates_shuffle(list(binary_data), seed1)
    binary_to_image(restored_binary, secret_image_size, output_image_path)
```

Appendix C. Embedding data into video

```
def embed_data(self, video_path, image_path, output_video_path, progress_var=None):
    import cv2
    import numpy as np
    # Load the secret image
    secret_image = cv2.imread(image_path)
    height, width, _ = secret_image.shape
```



```
# Open the video
cap = cv2.VideoCapture(video_path)
frame_count = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))

# Calculate the number of frames needed
bits_per_pixel = 3
total_bits_needed = width * height * bits_per_pixel
total_bits_available = frame_count * (64 * 64)

if total_bits_available < total_bits_needed:
    raise ValueError("Video doesn't contain enough frames to embed the secret image.")

frames_to_use = self.fisher_yates_shuffle(list(range(frame_count)), seed=42)

frame_index = 0
embedded_bits = []
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break

    if frame_index in frames_to_use[:total_bits_needed // 64]:
```

```
            yuv_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2YUV)
            y_channel = yuv_frame[:, :, 0].astype(float)

            for i in range(0, y_channel.shape[0] - 8, 8):
                for j in range(0, y_channel.shape[1] - 8, 8):
                    block = y_channel[i:i + 8, j:j + 8]
                    dct_block = self.apply_dct_to_block(block)
                    secret_bits = self.get_image_bits(secret_image
                    self.embed_bits_into_dct(dct_block, secret_bits)
                    embedded_bits.extend(secret_bits)
        frame_index += 1
        if progress_var:
            progress_var['value'] = int((frame_index / frame_count) * 100)
    cap.release()
    embedded_bits = self.fisher_yates_shuffle(embedded_bits, seed=43)
    self.save_video_with_embedded_data(output_video_path, embedded_bits)
```

Appendix D. Extracting data from video

```
def extract_data(self, video_path, output_path, progress_var=None):
    import cv2
    import numpy as np
    cap = cv2.VideoCapture(video_path)
    frame_count = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
    frames_to_use = self.fisher_yates_shuffle(list(range(frame_count)), seed=43)
```

```
    extracted_bits = []


    frame_index = 0
    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            break
        if frame_index in frames_to_use[:len(extracted_bits) // 64]:
            yuv_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2YUV)
            y_channel = yuv_frame[:, :, 0].astype(float)


            for i in range(0, y_channel.shape[0] - 8, 8):
                for j in range(0, y_channel.shape[1] - 8, 8):
                    block = y_channel[i:i + 8, j:j + 8]
                    dct_block = self.apply_dct_to_block(block)
                    block_bits = []
                    for x in range(2, 6):
                        for y in range(2, 6):
                            bit = round((dct_block[x, y] % self.embed_strength) / self.embed_strength)
                            block_bits.append(bit)
                    extracted_bits.extend(block_bits)
        frame_index += 1
        if progress_var:
```

```
            progress_var['value'] = int((frame_index / frame_count) * 100)
        cap.release()
        extracted_bits = self.fisher_yates_shuffle(extracted_bits, seed=44)
        width = int(np.sqrt(len(extracted_bits) // 3))
        height = width

        extracted_bits = np.array(extracted_bits, dtype=np.uint8)
        extracted_data = np.zeros((height, width, 3), dtype=np.uint8)
        bits_per_channel = len(extracted_bits) // 3
        for channel in range(3):
            channel_bits = extracted_bits[channel * bits_per_channel:(channel + 1) * bits_per_channel]
            packed_channel_bits = np.packbits(channel_bits)
            extracted_data[:, :, channel] = packed_channel_bits[:width * height].reshape(height, width)
        cv2.imwrite(output_path, extracted_data)
```


REFERENCES

[1] 1 Shadi A. Alhaj, 2. M.-K. (2016). Multi-layers Video Steganography: A Novel Technique for Image Hiding. *Transactions on Networks and Communications, 4*(6), 43-52. doi:10.14738/tnc.46.2529

[2] A SECURE BLOCK PERMUTATION IMAGE STEGANOGRAPHY ALGORITHM. (September 2013). *International Journal on Cryptography and Information Security, 3*(3), 11-22. doi:10.5121/ijcis.2013.3302

[3] Ajmera, A., Divecha, M., Ghosh, S. S., Raval, I., & Chaturvedi, R. (2019). Video Steganography: Using Scrambling- AES Encryption and DCT, DST Steganography. *Pune Section International Conference (PuneCon.* Pune, India: IEEE.

[4] Alam, S., Zakariya, S. M., & Akhtar, N. (2014). Analysis of modified triple — A steganography technique using Fisher Yates algorithm. *International Conference on Hybrid Intelligent Systems* (pp. 207-212). Kuwait, Kuwait: IEEE. doi:10.1109/HIS.2014.7086199

[5] Babar1, B. N., S. J., Nagwade4, P., & Gade5, A. (2024). Video Steganography Using DCT Algorithm. (pp. 300-303). INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY.

[6] Chih-Wei Shiu 1, Y.-C. C. (Feb. 2019). Reversible Data Hiding in Permutation-based Encrypted Images with Strong Privacy. *KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS, 13*(2), 1020 - 1042. doi:10.3837/tiis.2019.02.029

[7] Elleithy, K. M., & Abdelfattah, E. (2017). A Robust and Secure Video Steganography Method in DWT-DCT Domains Based on Multiple Object Tracking and ECC. (pp. 5354 - 5365). IEEE.

[8] Elleithy, R. J. (2015). A video steganography algorithm based on Kanade-Lucas-Tomasi tracking algorithm and error correcting codes. *75*(10 November), 10311-10333. From https://link.springer.com/article/10.1007/s11042-015-3060-0

[9] Jayakanth Kunhoth, N. S.-M. (2023). Video steganography: recent advances and challenges. 41943–41985.

[10] Laxmi Gulappagol*1, 2. a. (2020). Application of Fisher Yates Data Shuffling and RSA Encryption in Transform Domain Video Steganography. *A Society of Science and Nature Publication*, (pp. 52-57). Bhopal India.

[11] Lingamallu Naga Srinivasu, K. S. (2018). Video Steganography Using Two-Level SWT and SVD.

[12] Meenu Suresh1, D. I. (2018). High Secure Video Steganography Based on Shuffling of Data on Least significant DCT Coefficients. *ICICCS.* IEEE.

[13] Mstafa, R. J., & Khaled M. Elleithy, I. (2016). A Novel Video Steganography Algorithm in DCT Domain Based on Hamming and BCH Codes . *37th IEEE Sarnoff Symposium.* Newark, NJ.

[14] Mstafa, R. J., Elleithy, K. M., & Abdelfattah, E. (05-05 May 2017). Video steganography techniques: Taxonomy, challenges, and future directions. *IEEE.* Farmingdale, NY, USA. doi:10.1109/LISAT.2017.8001965

[15] Mstafa, R. J., Elleithy, K. M., & Abdelfattah, E. (August 2017). Video steganography techniques: Taxonomy, challenges, and future directions. *IEEE.* Farmingdale, NY, USA. doi:10.1109/LISAT.2017.8001965

[16] Rachna Patel, K. L. (2021). Study and investigation of video steganography over uncompressed and compressed domain: a comprehensive review. (pp. 985–1024). Springer.

[17] Sahib Khan1, M. A. (2019). On Hiding Secret Information in Medium Frequency DCT Components Using Least Significant Bits Steganography., (pp. .529-546).

[18] Shivani Gupta1, G. K. (2019). Video Steganography Using Discrete Wavelet Transform and Artificial Intelligence. *International Journal of Trend in Scientific Research and Development*, (pp. 1210 - 1215).

[19] Souma Pal and 2, *. K. (June, 2016). VARIOUS METHODS OF VIDEO STEGANOGRAPHY. *International Journal of Information Research and Review*, *03*(06), 5.

[20] Suresh, M., & Sam, I. S. (14-15 June 2018). High Secure Video Steganography Based on Shuffling of Data on Least Significant DCT Coefficients. *IEEE.* Madurai, India. doi:10.1109/ICCONS.2018.8662920

[21] Tanveer J. Siddiqui, A. K. (2020). CHAOS-BASED VIDEO STEGANOGRAPHY METHOD IN DISCRETE COSINE TRANSFORM DOMAIN. *International Journal of Image and Graphics.* World Scientific.

[22] Tarik Idbeaa1, 2. S. (2016, march 10). A Secure and Robust Compressed Domain Video Steganography for Intra- and Inter-Frames Using Embedding-Based Byte Differencing (EBBD) Scheme. (K. M. Yeng-Tseng Wang, Ed.) *A Secure and Robust Compressed Domain Video Steganography for Intra- and Inter-Frames Using Embedding-Based Byte Differencing (EBBD) Scheme*, p. 22. doi:10.1371/journal.pone.0150732

[23] Venugopal, E., Ranganathan, S., V.Velmurugan, & TadesseHailu. (2020). Design and implementation of video steganography using Modified CNN algorithm . *hird iternational Conference on Advances in Electronics, Computers and Communications (.* IEEE.

[24] Vivek Kapoor, A. M. (2015). An Enhanced LSB based Video Steganographic System for Secure and Efficient Data Transmission. (pp. 38-42). International Journal of Computer Applications.

PLAGIARISM REPORT

Thesis_201-35-2993.docx

ORIGINALITY REPORT

23%	19%	15%	13%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	3%
2	link.springer.com Internet Source	1%
3	dspace.daffodilvarsity.edu.bd:8080 Internet Source	1%
4	Submitted to Midlands State University Student Paper	1%
5	www.researchgate.net Internet Source	1%
6	umpir.ump.edu.my Internet Source	1%
7	dergipark.org.tr Internet Source	<1%
8	Submitted to University of North Texas Student Paper	<1%
9	Submitted to The University of Texas at San Antonio Student Paper	<1%

10	Submitted to University of Westminster Student Paper	<1 %
11	peerj.com Internet Source	<1 %
12	skannai.medium.com Internet Source	<1 %
13	Meenu Suresh, I. Shatheesh Sam. "High Secure Video Steganography Based on Shuffling of Data on Least Significant DCT Coefficients", 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), 2018 Publication	<1 %
14	trap.ncirl.ie Internet Source	<1 %
15	repository.neelain.edu.sd:8080 Internet Source	<1 %
16	www.e3s-conferences.org Internet Source	<1 %
17	Amal Nath M, Meenakshi Nair, Mili Murali, Sinadin Shibin, Shyna A. "High-capacity multimedia data hiding: synthesising adaptive inverted LSB332 with histogram difference-based frame selection and PCA-based region selection", Multimedia Tools and Applications, 2023	<1 %

	Publication	
18	core.ac.uk Internet Source	<1 %
19	stackoverflow.com Internet Source	<1 %
20	worldwidescience.org Internet Source	<1 %
21	www.ijraset.com Internet Source	<1 %
22	itiis.org Internet Source	<1 %
23	Tanveer J. Siddiqui, Ashish Khare. "Chaos-Based Video Steganography Method in Discrete Cosine Transform Domain", International Journal of Image and Graphics, 2020 Publication	<1 %
24	www.science.gov Internet Source	<1 %
25	P. Sathish Kumar, K. Fathima, B. Karthik, S. Siva Kumar, B. Sowmya, Ankush Ghosh. "Chapter 64 Studies on Steganography Images and Videos Using Deep Learning Techniques", Springer Science and Business Media LLC, 2022 Publication	<1 %

ACCOUNT CLEARANCE