Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
College of Information Technology
Software Department

Video Steganography Based on Object's Appearance Times and Area

A Thesis

Submitted to the Council of the College of Information Technology for Postgraduate Studies of University of Babylon in Partial Fulfillment of the Requirements for the Degree of Master in Information Technology - Software

By

Qusai Munir Diab AL-Durrah

Supervised by

Prof. Dr. Tawfiq A. AL-Assadi

2023 A.D. **1445 A.H.**

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

(قَالَ رَبِّ اشْرَحْ لِي صَدْرِي * وَيَسِّرْ لِي أَمْرِي *
وَاحْلُلْ عُقْدَةً مِنْ لِسَانِي * يَفْقَهُوا قَوْلِي)

صَدَقَ اللَّهُ العَظِيمُ

سورة طه (25- 28)

Declaration

I hereby declare that this dissertation entitled "**Video Steganography Based on Object's Appearance Times and Area**", submitted to University of Babylon in partial fulfilment of requirements for the degree of Master in Information Technology \ Software, has not been submitted as anexercise for a similar degree at any other University. I also certify that this work described here is entirely my own except for experts and summaries whose source are appropriately cited in the references.

Signature:

Name: Qusai Munir Diab AL-Durrah

Date: / / 2023

Supervisor Certification

I certify that the thesis entitled "**Video Steganography Based on Object's Appearance Times and Area**" was prepared under my supervision at the department of Software/ College ofInformation Technology/the University of Babylon as partial fulfillment of the requirements of the degree of Master in Information Technology - Software.

Signature:
Supervisor Name: Prof. Dr. Tawfiq A. AL-Assadi
Date: / / 2023

The Head of the Department Certification

In view of the available recommendations, I forward the thesis entitled "**Video Steganography Based on Object's Appearance Times and Area**" for debate by the examination committee.

Signature:
Name: Prof. Dr. Ahmed Saleem Abbas
Head of Software Department
Date: / / 2023

Certification of the Examination Committee

We, the undersigned, certify that (**Qusai Munir Diab AL-Durrah**) candidate for the degree of Master in Information Technology - Software, has presented his thesis of the following title (**Video Steganography Based on Object's Appearance Times and Area**) as it appears on the title page and front cover of the thesis that the said thesis is acceptable in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on: -------------- ---, 2023.

Signature:
Name: Dr. Wesam S. Bhaya
Title: Professor
Date: / / 2023
(Chairman)

Signature:
Name: Dr. Nashwan Jasim Hussein
Title: Assistant Professor
Date: / / 2023
(Member)

Signature:
Name: Dr.Saeed Mohammed Hashim
Title: Assistant Professor
Date: / / 2023
(Member)

Signature:
Name: Dr.Tawfiq A. AL-Assadi
Title: Professor
Date: / / 2023
(Member & Supervisor)

Signature:
Name: Dr. Wesam S. Bhaya
Title: Professor
Date: / / 2023
(**Dean of Collage of Information Technology**)

Dedication

This work is dedicated to…

The martyrs of Iraq of all sects and nationalities.

The one who guides me to the way of God And God gave me

guidance on his hands,

my Shafi'i on the Day of Deen,

the Beloved Muhammad.

My father,

I will always be your son who is proud of you, and I will not disappointyou, my beloved father.

My mother,

I ask Allah to protect you from all evil and I will not disappoint you.

My beloved brother Qaisar,

Who stands by me when things look very difficult.

Acknowledgements

Foremost, I am highly grateful to Allah (God) for His unlimited blessings that continue to flow into my life, and because of You, I made this through against all odds.

To my supervisor, **Prof. Dr. Tawfiq A. AL-Assadi**: I feel highly indebted to you. I am deeply grateful for your suggestions on this topic, and without your support, comments, and guidance, it would be difficult to finish this work.

To my wonderful Parents: Words cannot express my appreciation to you, the best father and mother. I wish to give you all thanks and love for your guidance, advice, invitations, and endless support.

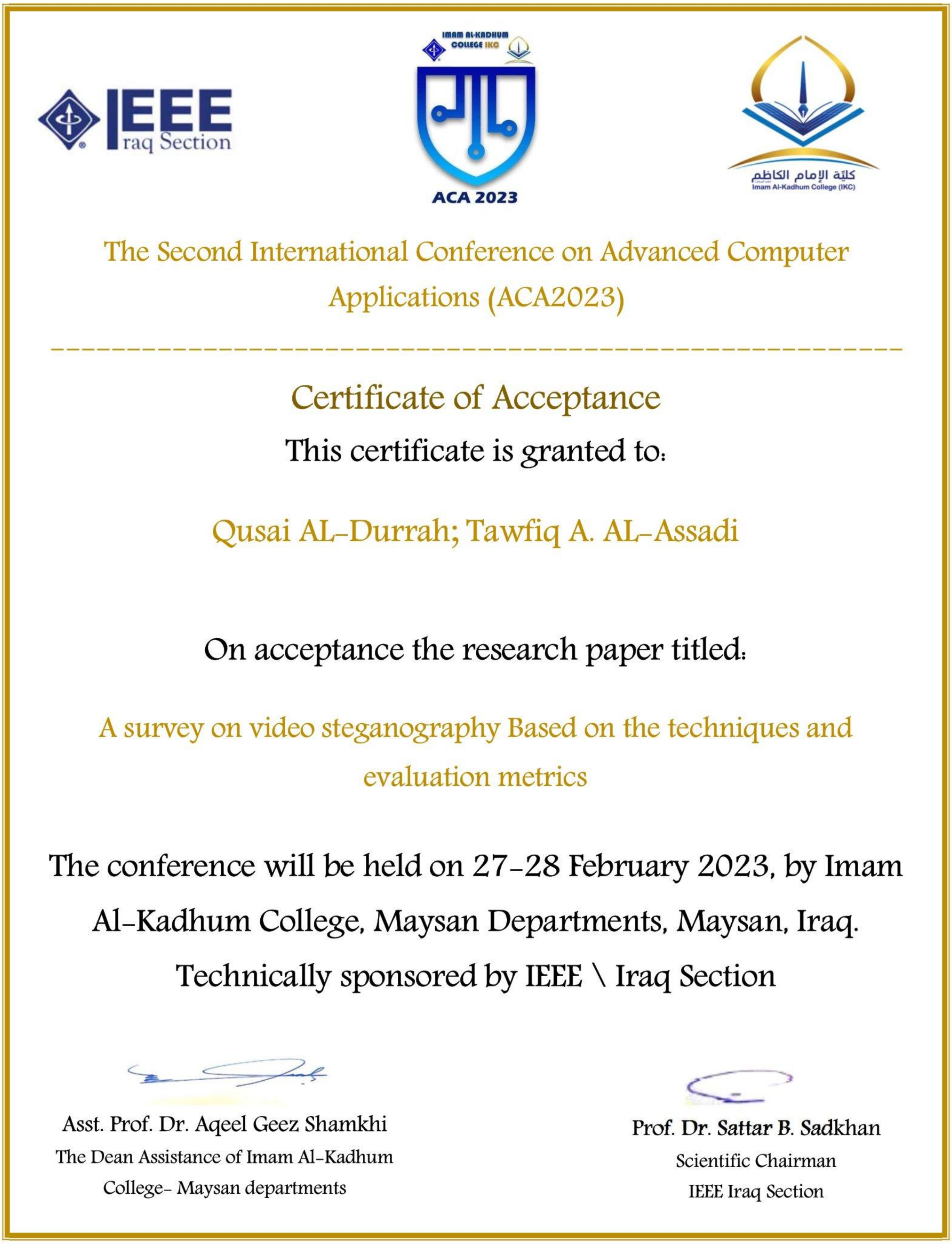
Thank you to my brother for the endless support. My mind cannot write that I love you, and I find in my heart nothing but gratitude for what you have given me throughout my study.

To my family, and my friends: I give you all thanks for your belief in me, yourconstant support, encouragement, and cooperation at all times

Last but not least, I would like to thank all the kind, helpful and lovely people who helped me directly or indirectly to complete this work and apologize to them for not being able to mention them by name here, but they are in my heart.

Qusai Munir Diab AL-Durrah

Thesis related publications



IEEE Iraq Section

IMAM AL-KADHUM COLLEGE IKC

ACA 2023

كلية الإمام الكاظم
Imam Al-Kadhum College (IKC)

The Second International Conference on Advanced Computer Applications (ACA2023)

Certificate of Acceptance

This certificate is granted to:

Qusai AL-Durrah; Tawfiq A. AL-Assadi

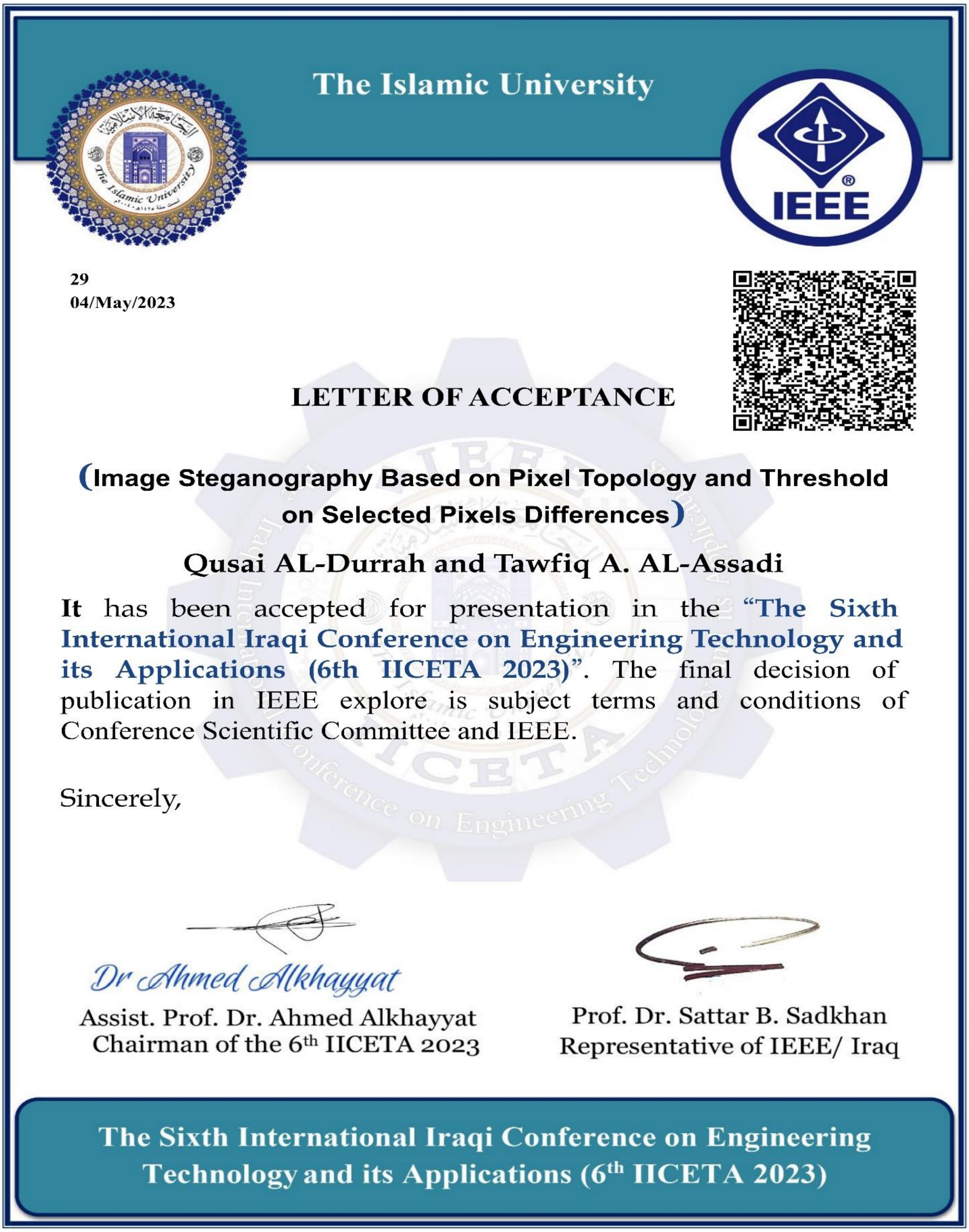
On acceptance the research paper titled:

A survey on video steganography Based on the techniques and evaluation metrics

The conference will be held on 27-28 February 2023, by Imam Al-Kadhum College, Maysan Departments, Maysan, Iraq. Technically sponsored by IEEE \ Iraq Section

Asst. Prof. Dr. Aqeel Geez Shamkhi
The Dean Assistance of Imam Al-Kadhum College- Maysan departments

Prof. Dr. Sattar B. Sadkhan
Scientific Chairman
IEEE Iraq Section



The Islamic University

29
04/May/2023

LETTER OF ACCEPTANCE

(Image Steganography Based on Pixel Topology and Threshold on Selected Pixels Differences)

Qusai AL-Durrah and Tawfiq A. AL-Assadi

It has been accepted for presentation in the **"The Sixth International Iraqi Conference on Engineering Technology and its Applications (6th IICETA 2023)"**. The final decision of publication in IEEE explore is subject terms and conditions of Conference Scientific Committee and IEEE.

Sincerely,

Dr Ahmed Alkhayyat

Assist. Prof. Dr. Ahmed Alkhayyat
Chairman of the 6th IICETA 2023

Prof. Dr. Sattar B. Sadkhan
Representative of IEEE/ Iraq

The Sixth International Iraqi Conference on Engineering Technology and its Applications (6th IICETA 2023)

Abstract

Video Steganography is the process of concealing data such as text, images, or videos within a cover Video. There are various ways to conceal messages in videos while making the changes to the video undetectable to the human eye. However, there hasn't been enough methods done to produce videos that are identical to unaltered videos while extracting the secret embedded message without any loss of data.

In this thesis, the author suggests a steganography technique that exploits certain objects, where the objects are detected and tracked throughout the selected video, and also exploits pixels topology with a pre-determined embedding threshold. The object will be selected based on its area and appearance times then the secret message will be embedded using the least significant bit (LSB) technique based on some pixels topology and the threshold.

The method used in embedding and extraction of the secret message is new and will be discussed thoroughly in the thesis and the targeted secret message is of image type only.

The quality matric used was the Peak Signal to Noise Ratio (PSNR) to measure the difference in the cover video before and after embedding the secret message. Experimental results denote impressive PSNR dB values where the stego-video in terms of visual is indistinguishable from the original video, and the secret image extracted from the stego-video is an identical copy of the original embedded secret image with 100 dB PSNR.

Table of Contents

Dedication i

Acknowledgements ii

Thesis related publications iii

Abstract v

Table of Contents vi

List of Tables viii

List of Figures ix

List of Abbreviations x

List of Algorithms xi

CHAPTER ONE INTRODUCTION 1

1.1 Introduction 1

1.2 Research Problem 1

1.3 Thesis Objectives 2

1.4 Research Significance and Contributions 2

1.5 Research Challenges 3

1.6 Related Work 3

1.7 Thesis Organization 7

CHAPTER TWO THEORETICAL BACKGROUND 9

2.1 Overview 9

2.2 Ancient History 9

2.3 Steganography Fundamental 10

2.4 Steganography versus Watermarking and Cryptography 11

2.5 Steganography Architecture 12

2.6 Steganography in Depth 13

2.7 Steganography Classifications 15

2.7.1 Cover Type-Based Classification 16

2.7.2 Hiding Method-Based Classification 16

2.7.2.1 Insertion-based technique 16

2.7.2.2 Generation-based technique 17

2.7.2.3 Substitution-based technique 17

2.8 Least Significant Bit Substitution 17

2.9 Video-Based Steganography and LSB Substitution 19
2.9.1 Spatial Domain LSB Substitution 20
2.10 Video tracking 20
2.11 Steganalysis Principles 23
2.11.1 Techniques of Steganalysis 24
2.11.1.1 Targeted Steganalysis 25
2.11.1.2 Blind Attacks 25
2.12 Video Quality Metrics MSE and PSNR 26
CHAPTER THREE THE PROPOSED SYSTEM 28
3.1 Overview 28
3.2 The Proposed System 28
3.2.1 The Message Embedding Part 28
3.2.1.1 Preprocessing Stage 28
3.2.1.2 Objects tracking 29
3.2.1.3 Object Selection preparing 31
3.2.1.4 Secret image Embedding 32
3.2.2 The Message Extraction Part 39
3.2.2.1 Video selection and Objects tracking 40
3.2.2.2 Object Selection preparing 40
3.2.2.3 Secret Image Extraction 40
CHAPTER FOUR RESULTS AND DISCUSSION 45
4.1 Overview 45
4.2 Software and Hardware 45
4.3 Experiments and Results Discussion 45
CHAPTER FIVE CONCLUSION AND FUTURE WORKS 55
5.1 Overview 55
5.2 Conclusion 55
5.3 Recommendations 56
5.4 Future Work 56
REFERENCES 57

List of Tables

Table 2. 1: Comparison between cryptography, watermarking, and steganography techniques [40] 12

Table 2. 2: Sequence of cover bytes before and after Embedding by Sequential LSB 18

Table 4. 1: Test results of various videos and secret images from the project .. 54

List of Figures

Figure 2. 1: steganography method General diagram..13
Figure 2. 2: MOT algorithm output...21

Figure 3. 1: Block diagram of the embedding part 28
Figure 3. 2: The percentage of information in each bit of one byte of data.......34
Figure 3. 3: Block diagram of the extraction part...40

Figure 4. 1: Objects detection and tracking 46
Figure 4. 2: Results of embedding five copies of 50x50 secret image in 720p video 47
Figure 4. 3: Results of extracting the five copies of 50x50 secret image in 720p video 47
Figure 4. 4: Results of embedding five copies of 50x50 secret image in 720p video 48
Figure 4. 5: Results of extracting the five copies of 50x50 secret image in 720p video 48
Figure 4. 6: Results of embedding five copies of 50x50 secret image in 1080p video 49
Figure 4. 7: Results of extracting the five copies of 50x50 secret image in 1080p video 49
Figure 4. 8: Results of embedding five copies of 100x100 secret image in 1080p video 50
Figure 4. 9: Results of extracting the five copies of 100x100 secret image in 1080p video 50
Figure 4. 10: Results of embedding five copies of 100x100 secret image in 1440p video 51
Figure 4. 11: Results of extracting the five copies of 100x100 secret image in 1440p video 51
Figure 4. 12: Results of embedding five copies of 108x101 secret image in 1080p video 52
Figure 4. 13: Results of extracting the five copies of 108x101 secret image in 1080p video 52
Figure 4. 14: Results of embedding five copies of 100x100 secret image in 1440p video 53
Figure 4. 15: Results of extracting the five copies of 108x101 secret image in 1440p video 53

List of Abbreviations

Abbreviation	Description
AES	Advanced Encryption Standard
ARP	Address Resolution Protocol
AVI	Audio Video Interleave
BMP	Bitmap image
dB	deciBels
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
EOF	End Of File
FFmpeg	Fast Forward Motion Picture Experts Group
HFYU	Huff YUV
ICMP	Internet Control Message Protocol
ID	Identification
JPG	Joint Photographic Group
JPEG	Joint Photographic Experts Group
LSB	Least Significant Bit
MP4	Moving Picture-4
MSB	Most Significant Bit
MSE	Mean Square Error
PNG	Portable Network Graphics
PSNR	Peak Signal-to-Noise Ratio
RCNN	Region-Based Convolutional Neural Network
RGB	Red, Green and Blue
ROI	Region Of Interest
SOAP	Simple Object Access Protocol
SSD	Single-Shot Detector
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
YOLO	You Only Look Once

List of Algorithms

Algorithm 3. 1: Objects tracking..30
Algorithm 3. 2: Object selection preparing...31
Algorithm 3. 3: Embedding data..35
Algorithm 3. 4: Secret image information embedding 38
Algorithm 3. 5: Extracting data..42

Chapter One
General Introduction

1.1 Introduction

Concerns regarding the surge in unauthorized use of such data have grown in recent years. However, due to the quick development of encryption, watermarking, and steganography techniques, the issues are dissipating. The use of steganography to validate the source of digital information is arguably an essential requirement given the growing use of digital media and the widespread usage of digital data, including text voice, video, and image by practically all institutions and organizations. In terms of the significance to this matter, this thesis primarily addressed video steganography.

Steganography is the art of hiding a file inside of a different file. Videos are ideal for embedding secret data because of their huge size and high level of redundancy. Video embedding techniques can be divided into two categories: embedding into uncompressed or compressed area. The former places more emphasis on encryption capacity whereas the latter prioritizes encryption security and speed. In uncompressed techniques, the video must be fully decompressed into a series of frames. Then, using encryption methods, secret data is inserted into each frame. Even though the techniques are slow and don't use frame dependence, they have a great capacity.

The research aims to develop and put into practice a novel algorithm for embedding data into video files by examining widely used Steganography methods based on the idea of embedding into uncompressed space. An algorithm that uses motion objects and verifies data changes in each object to intelligently split and embed the secret data in each object.

1.2 Research Problem

Due to the significant increase in video data, video processing researchers are becoming more interested in video steganography. A significant number of video steganography methods have been reported in recent literatures. However, there is an absence in current video steganography algorithms with preparatory phases for both cover videos and secret messages. Additionally, recent steganography methods have significant flaws in a number of areas, including imperceptibility, embedding capacity, robustness against attacks, and the ability to retrieve the original secret message without losing any data.

1.3 Thesis Objectives

Secure communication is one of the most crucial challenges for both individuals and organizations. Most governments keep tabs on how people, organizations, and even other governments communicate. Attackers put in excessive effort in order to eavesdrop on some parties. Therefore, communications could not be secure from watching or being attacked. This research looks into a few cutting-edge strategies to enhance video steganography techniques. The main objective of this thesis is to validate and develop a novel method that can retrieve secret data and outperform the existing video steganography methods.

1.4 Research Significance and Contributions

Due to the rapid growth of digital media and the usage of digital data, including image, text, and video and voice messages the use of Steganography to verify the sender's identity is indisputable.

Because of the rapid development of cryptography, watermarking, and steganography, concerns about the potential misuse of data have diminished. In recent years, concerns in this area have increased due to the unauthorized use of these data. The following are the primary contributions of this thesis:

1. Retrieving the embedded secret image from the stego-video with 100% PSNR (without any loss of data).
2. Increasing the embedding capacity of the stego video without

sacrificing quality of the original video.

3. Resistance to Steganalysis algorithms making the secret data hard to detect or even extracting it if it were to be detected by hackers.

1.5 Research Challenges

Many barriers were faced during different stages of this research as summarized below:

1. The first challenge was finding a video steganography technique that achieves a decent trade between visual performance, strong robustness, and large embedding capacity to resist a variety of unanticipated attacks.
2. The second challenge was the video steganography method, which can be used in conjunction with other approaches like deep learning. The secret message protection and visual quality of video steganography will be improved by the application of deep learning, which allows for the embedding of messages into regions of interest in the video, such as moving cars, human behaviors, and so on.
3. How to track objects after detecting them in each frame was also a challenge.
4. The final challenge was Retrieving the secret image after embedding with 100 PSNR in other words without any data loss.

1.6 Related Work

Several video steganography algorithms have been proposed over the past ten years using various spatial and transform domain techniques. The use of object detection in video steganography is a novel technique that needs further study. This subsection describes the related work of video steganography using different embedding techniques.

Video steganography can be done by employing transform and spatial domain techniques. Due to their quick and simple implementation, spatial domain-based techniques were frequently used by researchers;

nevertheless, researchers occasionally used hybrid spatial domain-based approaches for embedding, such as LSB with DWT and other transform domain techniques.

In 2014, Ramalingam et al. [1] proposed an integer wavelet transform technique using the LSB and Haar wavelet in order to embed secret information in a video frame. The RGB components of the high-frequency sub-bands HH, HL, and LH coefficients were used for the secret data embedding. Without reduction in the quality or size of the cover video, the exact data could be extracted using the suggested method.

In 2015, Abbas et al. [2] proposed a method for video steganography using the algorithm Cuckoo search. In this method, the secret data was broken down into individual bytes and then 5 distinct types were used to display each byte's bits. By comparing the similarity between the pixels and various byte types, the Euclidian distance was then used to choose a better pixel. After the secret message was transferred at random from one pixel to another using the Levy flight random walk, the secret message was then embedded inside the video frame using the LSB method. The average PSNR is 51.19 dB.

In 2015, Mstafa et al. [3] suggested a video steganography algorithm based on the (KLT) object tracking algorithm with BCH and DWT algorithms. The suggested method made use of Viola-Jones algorithm for facial recognition and the KLT tracking algorithm for video tracking. For the purpose of embedding secret data, the approach used face as ROI. To increase security, the secret data was encrypted using the error-correcting code BCH before being embedded. High and Middle-frequency of 2D-DWT sub-band coefficients of are used for embedding in the facial region of the frames. The testing findings showed that the suggested approach, with an average concealment ratio of 4.4%, reached high embedding capacity and efficiency. The average PSNR is 46.26 dB.

In 2016, Sudeepa et al. [4] presented a technique for video steganography. This technique encrypts the secret message using a feedback shift register (FSR) and a key, which avoids repetition by selecting the frame at random. The LSB technique was then used to embed the secret data inside a video.

In 2016, Sethi and Kapoor [5] proposed a technique for video steganography that makes use of the genetic algorithm and the AES cryptographic algorithm. This approach involved compressing the secret message to make it smaller. The encoded data was then embedded in the image using the LSB method and a genetic algorithm, where the genetic algorithm is used to choose the pixel used for embedding the message by using the LSB method. Next, the compressed message was transferred using the AES algorithm into the cipher text.

In 2016, Solichin and Painem [6] suggested the less significant frame (LSF) approach for video steganography. Using the properties of an optical stream, the frame's movement determined which frame was chosen to contain the secret information in this method. The average PSNR is 39.51 dB.

In 2017, Mumthas and Lijiya [7] proposed a novel technique for video steganography, encrypting the secret message with RSA and random DNA before compressing it with the Huffman encoding. The secret message is then embedded using the 2D DCT to further strengthen system security. The average PSNR is 37.29 dB.

In 2017, Mstafa et al. [8] moving objects in video frames were tracked using several object tracking methods as ROI for inserting secret data. When secret information were already encoded using BCH and Hamming codes, the suggested strategy used coefficients from two techniques, DWT and DCT, to embed the secret data using LSB. The test results showed that, in comparison to other relevant studies, the technique was able to attain high embedding capacity and PSNR. The average PSNR for DWT is 49.01 dB and 48.67 dB for the DCT.

In 2018, Kumar and Singh [9] proposed an algorithm for video steganography that inserts 3-D and 2-D confidential images into the human skin area of cover video frames. The red color channels of the frames' third-level DWT decomposition components were used for the embedding. The experimental findings showed that the suggested approach was resilient to MPEG compression and could insert secret images of various sizes with great imperceptibility. The average PSNR is 62.47 dB.

In 2019, Dalal et al. [10] introduced a video steganography technique for various SD and HD videos, focusing on robustness and imperceptibility. Before embedding, the suggested method transformed the frames to YUV components. It then applied second-level 2D-DWT to the Y components of the frame by iteratively applying it to all of the level one 2D-sub-bands. DWT's The results showed that the suggested method was effective at achieving imperceptibility with good PSNR values for all of the videos, and that it was also reliable with adequate embedding capacity. The average PSNR is 58.48 dB.

In 2020, Kumar & Soundrapandiyan [11] presented a strong video steganography method that defends against geometrical and signal processing attacks utilizing a combined keypoints detection algorithm. A set of predefined geometrical and signal processing assaults are used to extract keypoints first from each video frame utilizing a speeded-up robust features (SURF) and scale-invariant feature transform (SIFT) descriptor. After that The SURF and SIFT keypoint descriptors of the original frame and the attacked versions of the frame are compared to produce ROI keypoints. Then the embedding capacity of each ROI keypoint is calculated by grouping them into four LSB groups. A symmetric key-based shift cipher is then used to encrypt the secret data, adding another degree of protection for secret communication. Finally, using the LSB substitution method based on the values of the four LSB groups, the encrypted secret data have been inserted into the ROI keypoints. The average PSNR is 51.59 dB.

In 2021, Dalal & juneja [12] proposed a scheme that utilizes H.264/AVC video format for steganography. The method, which was focused on monitoring several moving objects, used DWT on the Region of Interest (ROI). Each object gets injected with a separate secret image after being tracked by several others in order to increase capacity. The average PSNR is 50.31 dB.

In 2022, Roselinkiruba et al. [13] presented a new pattern-based reversible information hiding technique for video steganography based on moving regions detection, where Every frame of the video is compressed using the DCT algorithm. Following the detection of moving objects in the frames, the image is divided into RGB channel 2x3 pixel blocks. By taking into account the weights between the pixels and applying LSB data hiding, the suggested Reversible Data Hiding Interpolation with Pixel Value Differencing method conceals the information into the up-scaled image. The average PSNR is 42.65 dB.

In 2022, Dalal & juneja [14] proposed a novel video steganography method with an excellent compromise between imperceptibility and robustness using 2D-DWT depending upon object tracking and detection that includes hiding the secret bits in the middle frequency sub-bands after using 2D-DWT to object detection for the video frames with moving objects. The average PSNR is 46.31 dB.

1.7 Thesis Organization

This thesis is divided into five chapters. Each chapter begins with a short overview that offers a general impression on the chapter. The key contents of other chapters are as follows:

- **Chapter Two**: entitled **"Theoretical Background"**. This Chapter provides an overview about steganography in general and every aspect of video steganography, video tracking, video quality metrics, and video Steganalysis in detail.

- **Chapter Three**: entitled **"The Proposed System"**. It covers the proposed system and its algorithms.
- **Chapter Four**: entitled **"Results and Discussion"**. This Chapter demonstrates the results of the proposed system and the research experiments. It also discusses the evaluation of the system's performance.
- **Chapter Five**: entitled **"Conclusion and Future Works"**. This Chapter presents the research conclusion and the possible future research directions to improve this work.

Chapter Two

THEORETICAL BACKGROUND

2.1 Overview

Steganography has been used for data transmission over a very long time ago. It can now be used to discreetly send secret data via digital communication channels thanks to the advancement of digital communication technologies. This Chapter gives an introduction and identifies the core principles and concepts of Steganography. Provides a thorough explanation of steganography and all of its facets. Also gives a brief introduction to video tracking and its techniques. Furthermore, discusses the video quality metrics that well be used to assess the results. And finally cover Steganalysis and its techniques briefly.

2.2 Ancient History

Greek word Steganos, which means "covered," and graphia, which means "writing," are the roots of the word steganography [15, 16]. Hence, "Covered Writing," which is the mixture of them, is steganography [17, 18]. It has been utilized for decades in a numerous ways [19]. Histaiacus shaved a slave's head in the 5th century BC and tattooed a message on his skull to conceal it following the slave's hair growth. Then he sent the slave with the message on his way [20, 21]. A Chinese secret writing technique that relies on utilizing a paper mask with holes was revived 500 years ago by the Italian mathematician Jerome Cardan. Cardan Grille was used as the name for the approach. During World War II, the Nazis developed a number of steganographic techniques and repeatedly used null ciphers and invisible ink [19]. Steganography is now widely employed in digital domains due to the rapid expansion of the information technology sector, and its techniques are constantly improving. Steganography can be used for multiple purposes, like copyright protection and identification, data authentication, watermarking.

2.3 Steganography Fundamental

The process of encrypting data to enable covert communication is known as social steganography. When sending confidential information to a distant recipient over a public network like the Internet, where anyone can access it, and the information should not be made available to unauthorized individuals, it is necessary to send the information covertly, so that no one is aware that it is being sent. Steganography is used mostly for this type of covert communication because it prevents people from suspecting that stego files contain a secret message. Steganography thus seeks to conceal communication by disguising the existence of information transmission. The benefit of this is that nobody is aware of the information, which allows to protect it from even attempted attacks [22].

In order to send the produced item holding the information to the intended individuals through the communication channel, the data must be embedded within a medium, also known as the cover media. The data-hiding object is referred to as the cover-file, and the final product is referred to as the stegogramme or stego-file. There are a variety of possible cover media. It could be a text, html, audio, video, image, or any other type of object. After the data has been concealed, a high-quality stegogramme must be produced in a certain way so that no one can tell that it includes secret data. As closely as possible, it should match the cover-file. Stego-files that differ from the cover-files could raise suspicion and catch the notice of an attacker. Hence, stego-files should have statistical and visual characteristics that are as similar to those of the cover-file as much as possible [22].

Stego-files may be attacked while they are delivered to the intended recipient to determine if they contain concealed data or not. Steganalysis is a field of study that focuses on identifying stegogrammes and steganographic messages [23]. Several steganographic methods have been created to generate high-quality stego-files, making it as challenging as possible to discover them using steganalysis tools. Hence, identifying stegogrammes is the main goal of steganalysis. So steganography is more than just information hiding in files; it also allows for secret communication. Additionally, it makes use of network protocols like SOAP and TCP to conceal the secret information within them or their headers because network headers typically have a large number of optional or unused fields for delivery [24, 25]. These protocols are known as carrier-protocols, along with others like ARP, TCP, UDP, or ICMP [24].

2.4 Steganography versus Watermarking and Cryptography

Data protection and confidentiality are the main goals of both steganography and cryptography. Although the cryptography establishes an overt communication channel, steganography establishes a hidden communication channel between authorized participants [26]. In cryptography, the secret data is detectable, but its substance is rendered incomprehensible to unauthorized personnel. Steganography and cryptography can both be used in a single system to offer more layers of protection [27].

To prevent unauthorized users from accessing the copyright information, digital watermarking systems use a preservation mechanism. This is done by blending the watermark data with the visible carrier data [28].Similar to steganography, watermarking has a wide range of uses, including copyright protection, broadcast monitoring, content authentication, digital fingerprints, and intellectual property protection [28-32]. There are various watermarking methods described in the literatures [33-39]. The general similarities and differences between cryptography, watermarking, and steganography methods are displayed in Table 2.1 [40].

Table 2. 1: Comparison between cryptography, watermarking, and steganography techniques [40]

Description	**Cryptography**	**Watermarking**	**Steganography**
carrier file	Image, Text	Image, video	Audio, text, video, image
Secret data	Text	watermark	all data types
Secret key	Must exist	Might exist	Might exist
Extraction stage	During the decoding phase, carrier data is not essential.	Availability of carrier data depends on the program	No need for carrier data.
Output	Cryptogram	Watermarked file	stegogramme
Transparency of information	Visible	The application determines transparency	Not visible
Level of Robustness	Robust Against deciphering	Fragile watermarking, semi-fragile watermarking, and robust watermarking	Robust Against detection
level of Security	Subject to the secret keys	Subject to the watermarking algorithm	Subject to the embedding algorithm
Requirements	Robustness	Fragile and semi-fragile watermarking do not require robustness while Robust watermarking needs robustness	Embedding capacity, embedding efficiency, indetectability
attacks	Cryptanalysis	Signal processing operations	Steganalysis
achieved Goal	Covert Data on communication channels	Copyright protection	Covert communication channels
failed Goal	Plain-text is extracted	Watermark is exchanged or erased	Communication is detected

2.5 Steganography Architecture

Two algorithms make up a steganographic system: one for message concealing and the other for message extraction. The goal of the hiding procedure is to embed information in the cover media, creating a stegogramme. This technique should be carefully constructed to make sure that the stegogramme is as similar to the cover medium as feasible because the message must be transmitted undetected. Consequently, the essential building blocks of the embedding procedure system are a cover file and a secret data as inputs, a steganography algorithm as the means of concealment, and an output stego-file [25, 41, 42]. The retrieval procedure, on the other hand, focuses on taking information out of the stego-file. This technique is merely the opposite of the hiding procedure. Stegogramme and the secret key are inputs for the retrieving procedure, which outputs the secret data [25, 41, 42]. A steganographic system's architecture is depicted in Figure 2.1.

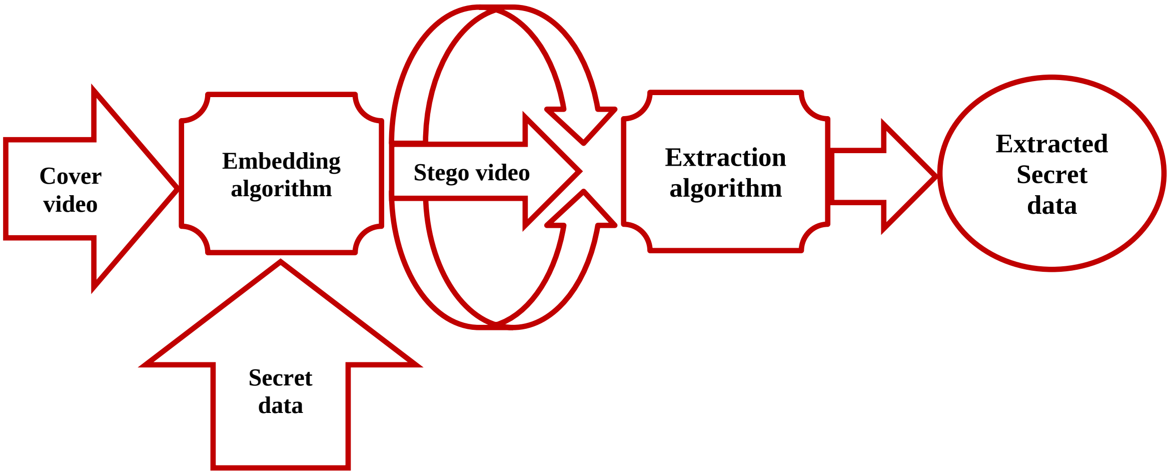


Figure 2. 1: steganography method General diagram

When the secret information is embedded inside some cover media, the stego system should be carefully constructed to embed the information and create a stego-file that is a perfect copy of the cover media, or at least as near to it as feasible. This will ensure that the attacker views the stegogramme and the communication currently happening as an ordinary one. Once the stegogramme has been obtained, it is often transferred to a distant receiver along with the secret key to reveal the embedded data [25, 41, 42].

2.6 Steganography in Depth

As steganography has been utilized, it has progressed, and numerous new techniques and algorithms have been created to enhance the concealing process and boost the security of the hidden data. To make cover mediums contain the data and produce stegogrammes, the embedding procedure involves changing their contents and setting their values. We are unable to change the data in all places of the cover file, though. The cover file may be destroyed if certain values are changed, or they may cause some obvious and visible distortion. The likelihood of discovering the stegogramme would therefore be quite high if the distortion was discernible. Hence, the probability of becoming undetectable increases with decreasing distortion. Steganographic systems must therefore basically recognize the redundant information of the cover-file [22].

Redundant information is minor information that, when adjusted, has no discernible effect on the cover file's overall perceptibility. As a result, the change in the information isn’t readily apparent [25, 43]. So, any adjustment to these superfluous bits shouldn't compromise the cover medium's integrity, retaining the quality that in turn would increase the stegogrammes' imperceptibility and indetectability. Therefore, even if the hiding method is known to the public, the existence of hidden data cannot be detected if the stego-file shows no suspicious changes or indications. As a result, In order to avoid raising any suspicions and make it challenging for steganalysts to detect steganography in stego-files similar to original files, steganographic systems should produce stego-files that are as similar to the cover-file as much as possible [20].

By increasing the robustness or the imperceptibility, secret data security is improved. Moreover, some algorithms expand the amount of hidden data that may be stored on covert mediums. Thus, imperceptibility, robustness, and capacity are the main steganographic system characteristics that must be taken into account [20, 44, 45]. Hence, the following can be taken into account as efficiency criteria for steganographic systems:

1. Robustness: The degree of a steganographic system's resistance to Steganalysis and other attacks, as well as how challenging it is to ascertain whether or not it contains hidden data [20, 44, 46]. Being robust means being able to endure the Steganalysis methods used for hidden data discovery, extraction, and destruction.
2. Imperceptibility: The degree to which there is no perceptibly observable change or distortion in the stego file. the quality of the resultant stego file is therefore crucial. This is accomplished by not altering the resulting stego medium significantly [20, 44, 47]. The stego-medium must also not be statistically perceptive, therefore its statistics must be the same as those of the cover medium [25].
3. Capacity: The capacity of a cover media is the amount of secret information that can be covertly put within it. Data hiding within cover files may occasionally be performed with significant amounts, but the resulting stego files would be so obvious as to contain secret data. So, enhancing an algorithm's capacity must be completed while preserving the cover files' quality and causing the least amount of disruption to its properties as feasible [23, 25].

Yet, there is a trade-off between capacity and imperceptibility, where adding more data produces more artifacts into cover materials, which then makes hidden data more perceptible [25]. Data embedding should be as minimal as possible as a result, as the more data that is embedded, the more the cover-file is affected, and the easier it is for steganalysts to find the stego-file [20, 44, 47]. Thus, it is challenging to simultaneously increase capacity and retain imperceptibility [25].

2.7 Steganography Classifications

There were several methods established for categorizing a steganographic system, however there are two primary methods. The first is dependent on the kind of cover file, and the second is based on the technique employed for hiding [25].

2.7.1 Cover Type-Based Classification

Since there are many types of cover mediums available, steganography can be categorized as follows depending on the type of cover medium used to conceal the data:

1) Network steganography.
2) HTML steganography.
3) Text steganography.
4) Audio steganography.
5) Image steganography.
6) Video steganography.

However, because the characteristics of different types of cover files vary, so do the hiding procedures themselves.

2.7.2 Hiding Method-Based Classification

Steganography can be categorized based on how data is concealed. As a result, there are three ways to hide data using steganography [25]:

1. Insertion-based technique.
2. Generation-based technique.
3. Substitution-based technique.

2.7.2.1 Insertion-based technique

The insertion-based approach finds locations in cover medium that is disregarded by programs that read this cover medium, and hide the secret information there. This technique has both an advantage and a drawback. The drawback is that because this technique embeds the data inside the cover medium, the size of the file containing the created stego-file would be larger than the cover file size. Since it is uncommon to have access to the original cover file for comparison, this method might be successful if the stego file size is sufficient. The advantage is that the quality of the cover file is maintained and that the stegogramme will not noticeably change because the content of the cover file isn't changed.

For example, numerous files contain the End-Of-File (EOF) marker flag. This flag helps applications find the end of a file so they can finish processing it. If you add the data after the cover file's EOF marker, the application won't take into account the embedded information when reading the produced stegogramme. This method's drawback is that the size of the final stego-file equals the sum of the sizes of both the cover file and the embedded information, which could create questions if the cover file's size is too large to adequately account for. [19, 48]

2.7.2.2 Generation-based technique

In order to hide the data, the generation-based method first creates a cover file. Hence, a cover file that already exists is not necessary. The primary advantage of this technique is that the stego-file doesn't have distortion or is larger than the original cover file. But since the generated files have random content, people can find them to be unrealistic. So, perhaps random-looking images are appropriate for this type of information concealment. [48]

2.7.2.3 Substitution-based technique

The substitution-based approach uses cover files to find unnecessary information and replace it with secret data by finding and replacing the appropriate values. This method's main advantage over the insertion-based method is that the sizes the stego-file of and the cover-file are the same because no new data was inserted. On the other hand, it's possible that the change diminished the quality of the cover file. Additionally, the quantity of secret information that can be inserted is limited by the size of the trivial data that may be deleted or altered. [48]

2.8 Least Significant Bit Substitution

One of the first and most used steganography techniques is LSB substitution [21, 49]. In computer science, the smallest (right-most) bit of a binary sequence is referred to as the (LSB) [20]. The act of putting a secret data bit into a cover binary sequence's Least Significant Bit is known as "LSB substitution", No matter how the order in which the sequences of the cover binary used to contain the secret information were used or whether the order of embedding was random or sequential, by replacing the value of the Least Significant Bit cover binary sequence with the secret bit value. Hence, if you need to conceal the secret bit using the LSB substitution approach and you have a cover binary sequence of 11100111 and a secret bit with a value of 0, all you need is swapping the secret bit with the LSB of the cover binary sequence, making the cover binary sequence 11100110. From the start to the end of the cover file, the Hide & Seek algorithm successively inserts secret bits into the LSBs of the cover binary sequences, is the simplest algorithm that conceals data through LSB substitution. When you wish to hide a byte of secret information, all you have to do is take the 8 bits that make up the secret byte, replace the least bit in a series of 8 binary sequences of the cover information with these bits, and so forth. Your secret bits are simply substituted for the least bit in each cover binary sequence. As a result, to embed each secret byte, eight cover binary sequences are needed [43].

For example, assume that the cover data's binary sequence unit is one byte, and the secret byte is 10001011 that is needed to be embedded using the LSB substitution technique into a string of cover bytes. The procedure would be carried out as illustrated in Table 2.2:

Table 2. 2: Sequence of cover bytes before and after Embedding by Sequential LSB

Cover byte Index	**The series of cover bytes before the embedding process**	**The resulting stego bytes after the embedding process**
0	**10111110**	**10111111**
1	**10011001**	**10011001**
2	**10100001**	**10100000**
3	**01111100**	**01111101**
4	**01110110**	**01110110**

5	**11110000**	**11110000**
6	**11011111**	**11011110**
7	**11010011**	**11010011**
The new bit value differs from its original **The new bit value is identical to its original**		

The LSB substitution method has a number of advantages. First off, because it doesn't change the number of cover data bytes or their size, it has no impact on the size of the cover medium. It merely swaps out part of the cover bits for the secret bits. Next, it does not significantly alter the cover data, which is supported by two factors. Initially, the change happens in the bit with the least weight (rightmost) out of all the bits in a byte. Simply put, if this bit's value changes from 1 to 0 or from 0 to 1, as indicated in Table 2.2 at bytes 3 and 2 respectively, the value of the byte is modified simply by 1 progressively or decreasingly in each case. The human visual systems can't pick up on this level of change. Second, some of the bits are changed to have the same value. For instance, at byte 1 the byte is unchanged. Therefore, on average the maximum cover size used to modify is half of the cover bits [43, 46, 50]. Hence, even if the second least significant bit is used for concealing, it is better to employ two bit planes than one, if we hide any information inside [51]. From a capacity perspective, LSB uses a modest amount of cover bytes to encode a specific amount of secret data. Whereas it requires 8 bytes of the cover for each secret byte to be placed. So, all that is required to conceal the data is the employment of an appropriately sized cover-file.

2.9 Video-Based Steganography and LSB Substitution

A video is a collection of numbers that make up various frames that are images with varying light intensities, and pixels which are the numerical values form a grid of points [43]. The goal of video steganography is to develop and enhance algorithms and techniques for concealing data inside of videos. Video steganography is The most widely used type of steganography, because capacity and robustness to attacks [25]. The two possible domains for information hiding in videos are the transform domain and the spatial domain [43, 44, 46, 52]. Transform domain techniques use lossy videos such as mp4, mkv, etc., and spatial domain techniques use uncompressed videos as avi.

2.9.1 Spatial Domain LSB Substitution

The video frames' spatial domain consists of the images' planes; the pixels that make up an image [23]. Data embedding in spatial domain methods is accomplished by directly embedding the secret information bits into the values of the image pixels [53, 54]. As LSB substitution technique embeds the secret bits into the cover-file directly, it is the approach most frequently utilized in the spatial domain. Therefore, in video steganography, the secret information is directly embedded into the LSBs of the pixels values that make up the cover video images. Spatial domain techniques are applicable to videos that are uncompressed as avi [53].

2.10 Video tracking

Video Steganography utilizes Multiple Object Tracking (MOT), which is a computer vision application that seeks to examine videos to recognize and track objects that belong to one or multiple classifications, like animals, humans, cars, and non-living things, without having any background knowledge about the targets shape and number. MOT techniques associate each box with a target ID (referred to as a detection) to differentiate between intra-class objects, opposite to object detection algorithms, which only provide a group of rectangular bounding boxes identifiable by their width, height, and coordinates. Figure 2.2 shows an illustration of a MOT algorithm's output. [14]

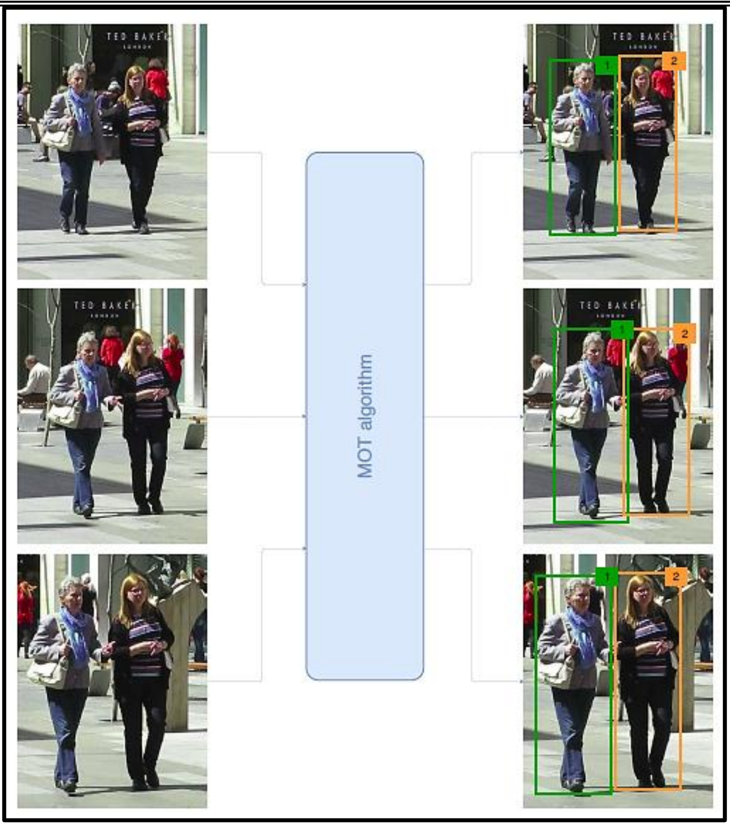


Figure 2. 2: MOT algorithm output

There are several ways to detect and track objects, and the one that was utilized in this project is YoloV5 for detection with a novel tracking algorithm plus OpenCV for video processing. OpenCV is a large library that is open-source for image processing, machine learning, and computer vision that is written in optimized C/C++ to take the benefit of multi-core processing [55]. It currently has a huge role in real-time applications, which is essential in modern systems. With it, one may analyze videos and images to identify human handwriting, faces, and objects of any kind. In order to identify the visual pattern and all of its various features, OpenCV uses vector space and applies mathematical operations on these features [56].

A more sophisticated form of image categorization called object detection uses a neural network to recognize items in an image and to highlight them with bounding boxes [57]. Therefore, the term "object detection" refers to the location and identification of objects within an image that fall into a predefined set of classes. The wide application of tasks like recognition, detection, and localization in practical situations makes object detection a key area in computer vision [57]. Generally, there are two methods for detecting objects, which are:

- **Two-stage object detection**
- The phrase "two-stage object detection" describes the application of methods that separate the object detection problem statement into the 2 stages listed below:
  a) Detecting potential object regions.
  b) Classifying the image into object classes in those regions.

Common two-step algorithms such as Faster-RCNN and Fast-RCNN often employ a Region Proposal Network, which suggests potential object-containing regions of interest. [58]

- **One-stage object detection**

One-stage detectors anticipate all of the bounding boxes in a single neural network pass. It is significantly quicker and more suited for portable devices. The most popular one-stage object detectors include DetectNet, SSD, SqueezeDet, and YOLO. [59]

The highest accuracy rates are achieved by two-stage object detection models, which are often slower. In contrast, one-stage object detection models are substantially faster than two-stage object detectors but achieve lower accuracy rates. Even while two-stage object detection models produce accurate findings with a high mean Average Precision, they require a lot of iterations within the same image, which prevents real-time detection and slows down the algorithm's detection speed [59].

You only look once (Yolo) is an algorithm that (in real-time) detects and identifies various elements in images. The object identification process in YOLO, which is carried out as a regression problem, offers the class probabilities of the identified images. Convolutional neural networks

(CNN) are used by the YOLO method to recognize objects in real-time [60]. The method just needs one forward propagation through a neural network to detect objects, as suggested by the name. This shows that prediction is carried out throughout the entire image using a single algorithm run. The CNN is used to forecast multiple bounding boxes and class probabilities at once. The YOLO algorithm comes in a wide variety. The three most popular ones are YOLOv3, YOLOv4, and YOLOv5. Ultralytics, the same company that created the Pytorch version of YOLOv3, released YOLOv5 in June 2020 [60]. YOLOv5 is available in four models, which are X, L, M, and S, each one of them offering different detection accuracy and performance and the L model was used in this research.

2.11 Steganalysis Principles

Steganalysis is the art of recognizing and detecting stego-files that hold embedded secret information [20]. Hence, while the primary goal of Steganalysis is to identify stego files, it also includes data destruction and data extraction [25]. While extracting the message should be regarded as extremely unlikely, doing so would need the steganalyst to figure out the mechanism used to conceal the message, and the secret data is typically encrypted before being inserted. Therefore, Steganalysis becomes successful and the steganography system is compromised if an attacker detects the existence of the embedded secret information. Steganalysts are individuals who use Steganalysis with the goal of detecting and intercepting stego-files and the concealed information in the communication channel [23]. The properties of a cover file typically change when some parts are modified to embed secret data, and this can be a sign that concealed data is there [25]. A comparison between a cover-file and its associated stego-file may therefore be used to determine whether a hidden message is there or not. Thus, In order to prevent this kind of comparison, cover-files used to hide information shouldn’t be made available online for everyone, and they might even need to be deleted after finishing [25]. The resulting stegogramme is often conveyed to a remote receiver across a communication channel after the secret data has been embedded within a cover medium. It might undergo Steganalysis along the way. An attacker may choose to target the communication in several ways after using Steganalysis tools to identify a stego-medium. According to the Steganalyst's role, there are three different types of steganography attacks: passive attack, active attack, and malicious attack [20, 23, 25]:

- A Passive Attack is the situation in which the attacker merely keeps track of the communication, allows or rejects the message delivery.
- An "active attack" is when the attacker interferes with the communication by tampering with the detected stegogrammes and causing distortion to them.
- A malicious attack occurs when the attacker substitutes a false message for the secret message and attempts to deceive one of the communication parties by pretending to be that party. Nevertheless, this form of attack is too difficult to execute, because the attacker must be completely aware of the hiding procedure, including the method employed and the secret key, if one exists. Additionally, this kind of attack is rather simple for receivers to spot because the hidden messages are illogical.

2.11.1 Techniques of Steganalysis

Steganalysis, as previously stated, is the science involved with uncovering secret communications by looking for concealed information in stego media that are being transmitted between parties. Because the secret message is embedded there, steganographic technologies leave certain traces in stego files. The stego files can be recognized in some way thanks to these traces. Steganalysis therefore focuses on using these traces to find the stegogrammes.

2.11.1.1 Targeted Steganalysis

Targeted Steganalysis approaches are developed in direct contract with a certain embedding methodology in order to locate stego files by evaluating known side effects of specific steganographic systems, so it is necessary to have in-depth knowledge of the steganographic algorithm that the attack is directed at [20].

- **Visual Attacks**

Visual attacks involve inspecting the target file or specific parts of it visually while using a software to spot any evident discrepancies [20]. Naturally, stego files with quality degradation due to steganographic alteration appear suspicious and can be seen with the naked eyes. Therefore, a steganographic algorithm should maintain the quality of cover files as a means of concealing data inside as the first guideline to avoid visual attacks. When steganalysts conduct visual attacks, they focus solely on the possible embedding locations inside stego files to look for manipulation indicators.

- **Statistical Attacks**

These attacks rely on identifying changes that have been made to the statistical characteristics of cover files [25]. Statistical Attacks can show that a file has been modified, but they cannot show what method was employed. Because they may be automated, statistical attacks are frequently favored [20]. Many statistical characteristics of images, including the kurtosis, differential values, standard deviation, median, and skew can be analyzed [25]. The RS Analysis, Chi-square Test, Sample Pair Analysis, and Histogram Analysis are all examples of statistical attacks.

2.11.1.2 Blind Attacks

These attacks seek to assess the likelihood of embedding based just on the data in the allegedly suspected video, despite the fact that it is unclear how the data may have been inserted. The core principle is that neither the algorithm nor the cover video are known [20]. Some of the most common blind attacks are Farid's Wavelet Based Attack, and Wavelet Moment Analysis (WAM) [53].

2.12 Video Quality Metrics MSE and PSNR

The most popular and extensively used metrics for evaluating video quality are mean square error (MSE) and peak signal-to-noise ratio (PSNR) [25]. Whereas MSE assesses the difference between two videos (how dissimilar they are from one another), PSNR assesses the similarity between two videos (how similar they are) [25]. As a result, better video quality is achieved with higher PSNR values and lower MSE values. Since there is little difference between the original video and the reassembled video, the highest video quality is obtained when the MSE value is very low or close to zero [25]. For PSNR, the greater the PSNR number, the better the level of imperceptibility [25]. Nonetheless, PSNR values that are between 25 and 40 can be considered as normal values [48]. For example, if the PSNR value surpasses 30 dB, it is impossible to distinguish between a grey-scale cover video and its stego video [25, 61, 62]. PSNR and MSE equations are as following below:

$$\mathrm{PSNR} = 10\log_{10}\frac{\mathrm{max}^2}{\mathrm{MSE}} \tag{2.1}$$

Eq. 2.1 shows how to calculate the PSNR, Where max indicates the highest estimate of pixels in a frame, and (MSE) is used to assess the distortion between the stego-image and the cover-image [63], where it is determined using Eq. 2.2:

$$\mathrm{MSE} = \sum_{\mathrm{i}=1}^{\mathrm{M}*\mathrm{N}}(\mathrm{p_i} - \mathrm{p_i'})^2/(\mathrm{N} * \mathrm{M}) \tag{2.2}$$

Where pi and pi′ indicate the estimation of the pixel after and before the data embedding into the cover image, and N*M represents the size of the image (M=width, N=height) [64].

However, the MSE for colored images is defined in the following Eq. 2.3 [25]:

$$MSE_{Avg} = \frac{MSE_R+MSE_G+MSE_B}{3} \tag{2.3}$$

Where MSE_B, MSE_G, and $MSE_R$are the MSE of blue, green, and red respectively.

Chapter Three
THE PROPOSED SYSTEM

3.1 Overview

This chapter presents a novel video steganography scheme using objects appearance times and area with embedding threshold. The basic flow of the proposed approach is shown in block diagrams. Also, the object detection, embedding and extraction process are illustrated in detail.

3.2 The Proposed System

The proposed system consists of two parts, first is the embedding of the secret message and the second is the extraction of the secret image.

3.2.1 The Message Embedding Part

The embedding of the secret message part is shown in Figure 3.1 which consists of a set of stages that will be covered in the next subsections.

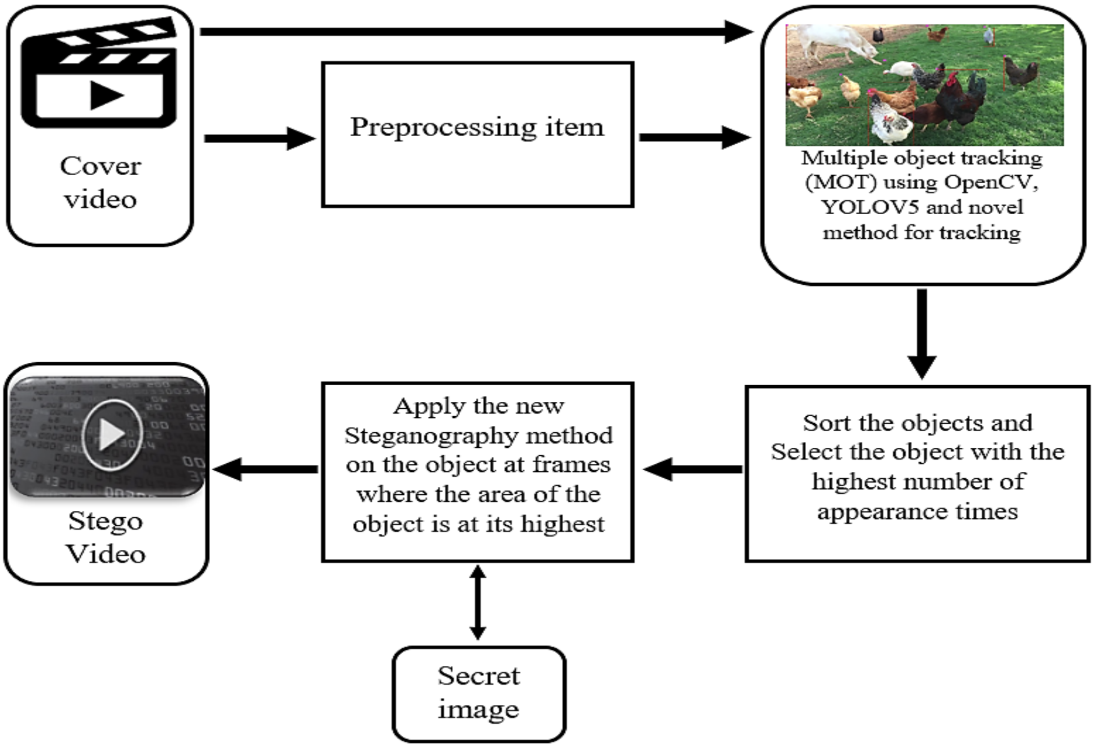

Cover video
Preprocessing item
Multiple object tracking (MOT) using OpenCV, YOLOV5 and novel method for tracking
Sort the objects and Select the object with the highest number of appearance times
Apply the new Steganography method on the object at frames where the area of the object is at its highest
Stego Video
Secret image

Figure 3. 1: Block diagram of the embedding part

3.2.1.1 Preprocessing Stage

The first Stage is to select a video that will be the cover of the secret message then Preprocess the selected video by applying Huffyuv (Hfyu) lossless compression and (avi) video format so that the RGB values don’t change during the embedding video rewriting phase and the extraction phase using OpenCV with ffmpeg (you can skip this step if the video is preprocessed before).

3.2.1.2 Objects tracking

In this Stage, the cover video will be read and put in a bufferlike array called deque of (Mats) to speed the process of detecting of objects and tracking them. Each frame will be taken from the deque and YoloV5 will detect the objects (YoloV5 was modified to ignore objects with area less than 1000 to decrease tracking confusion of unnecessary objects) in that frame and there coordinates points will be saved (X,Y) then these points for each will be sent to the tracker to give them an identifier and each object will have a number of information that will be saved as will (id, appearance times, area, number of current frame, box point number).

The new tracking method will have memory about the detected objects of (n) frames at a time, in the proposed system (6) frames were used at a time because too high memory will cause objects to be given IDs of another objects that were in the same points and too low memory will make the tracker give new IDs often. The principle of the new tracking method is that it will keep these information about the objects of the last 6 frames and it will compare them with the objects of the current frame if the coordinates are not too far off then the object will retain the same id that it was given before and if the object is new it will be given a new id. Yolo objects detection and tracking will be repeated for all the remaining frames in the deque.

Algorithm 3.1 illustrates the tracking algorithm that was used to track the objects.

Algorithm 3. 1: Objects tracking

Input: obj_info, current frame objects.

Variables definition:

obj_info = array of objects – an array that has the details about all the detected objects.

dis_vec = array of integers – incudes all the distance vectors for each object in the current frame.

prev_bbox_vec = array of objects - this array saves all objects that have appeared in the last 6 frames.

cur_index = integer – saves the index of the current object.

cur_dis = integer – the distance of the current object in contrast to its previously seen location.

max_object_count = Global – integer – how many objects have been tracked to this point.

Output: obj_info which has the information of all past, currently add and updated objects.

Begin

```
1. dis_vec ← calculate the distance vector for each object in the
   current frame
2. for i ← 0 to prev_bbox_vec length Do
3.     begin
4.         cur_index ← -1
5.         if the current object has the same ID of the current I object then
6.             begin
7.                 cur_dis ← calculate the current distance of the object by
                   using i object location with the location of the current
                   object as following: sqrt( (cur.x - prev.x) * (cur.x - prev.x)
                   + (cur.y - prev.y) * (cur.y - prev.y)).
8.                 if the cur_dis is lesser than 100 and dis_vec of the current
                   object is greater than the cur_dis then
9.                     begin
10.                        dis_vec of the current object ← cur_dis
11.                        cur_index ←  current object index
12.                    End if
13.            End if
14.        if cur_index greater or equals 0 and track ID is absent then
15.            current object track ID ← i track ID
16.    End for i
17. for i ← 0 to number of objects in the current frame Do
```

```
  begin
      If i object track ID equals 0 then
          i object track ID ← track ID + 1
  End for i
for i ← 0 to objects in the current frame Do
  begin
      if I object track ID greater than max_object_count then
          add the new object to obj_info with all its details and
          increment max_object_count by 1 with the object seen count
          to 1
      else
          update the current object with its new coordinates and
          details and increment the objects seen count by 1
  End for i
Return obj_info
End Algorithm.
```

3.2.1.3 Object Selection preparing

All the tracked objects will be sorted by the number of times they appeared in descending form and if the number of any object appearances is less than 70 it will be erased, after that each object as you know have a lot of frames that it appears in so each object will be sorted by its area and this in turn will lead us to the next most important step which is the embedding the secret message itself.

Algorithm 3.2 demonstrates the deletion, sorting by appearance times and then sorting by area for each object in the frames for the tracked objects.

Algorithm 3. 2: Object selection preparing

Input: obj_info, min_seen_threshold.

Variables definition:
obj_info = array of objects – an array that has the details about all the detected objects.
min_seen_threshold = integers – a threshold for the minimum number of appearances of the tracked object in the frames to accept it for embedding.

Output: obj_info after deletion and sortation.

```
Begin
for i ← 0 to obj_info length Do
    begin
        if the i object's seen count is lesser than min_seen_threshold
        then
            Erease the object from obj_info array
    End for i
Sort the tracked objects in obj_info in descending order by there seen
count
Sort each object in the frames in obj_info by there area in descending
order
Return obj_info
End Algorithm.
```

3.2.1.4 Secret image Embedding

In this final stage a secret image will be selected and the embedding process will start and (n) number of copies from the secret image will be embedded in the object selected, and as for the object selected it will be the object with the highest appearance times throughout the video and the embedding will be done in the frames were the area of the object is at its highest one after another. The entire new secret image embedding method is stated as follows:

a. Determine if the selected object has bytes enough to imbed the secret image into by checking how much bytes the secret image need and this in turn will lead to pixel selection. The pixel selection is an essential part of the image steganography. Its goal is to select a candidate pixel in the cover image in a specific order and embed the part of the secret image in that pixel value. In the proposed method the technique is as follows:

   - For example, start from the top left seventh pixel (X=6, Y=0) because the first six pixels is reserved for information about the secret image that will be discussed thoroughly in step (c).
   - The pixel selected for embedding the secret data in will be the middle pixel between each two pixels. For example, if X=6 now

then X+2=8 and by this the secret data will be embedded in pixel X=7 after that X=8 and X+2=10 so the secret data will be in X=9 and so on.

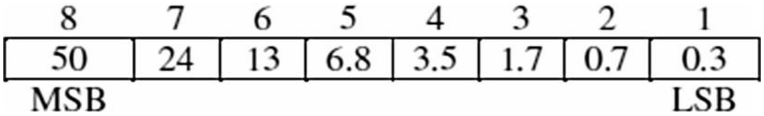
- If there is no middle pixel (X=width) then the pixel is skipped.

Now to determine how much bytes the secret image is going to need the method to do so will be as follows and it will be done for each color band(R, G, and B):

- Define the threshold and let it be for example 5, By default in the method proposed the number of bits that will be embedded in each selected pixel of the cover image will be (2), as long as the difference between the two pixels left and right(X=6, X=8) of the selected middle pixel(X=7) is less or equal to the threshold starting from the secret image Most significant bits (MSB) that will replace the cover image selected pixel`s first two Least significant bits (LSB) and then taking the next two bits from the secret data and so on, else if it is higher than the threshold then the pixel will be skipped.
- The above method will take number of bytes as much as: bytesneeded = secret image Width * secret image Height * 4, As well as the number of information bytes needed about the secret image to be reserved in the stego image first pixels that one pixel for each two numbers of the bytesneeded and two pixels one for the width and the other is for the height. To decrease the number of bytes needed the below method was used.
- now if the difference between the two pixels left and right(X=6, X=8) of the selected middle pixel(X=7) is zero then four bits will be taken from the secret data and embedded in the selected pixel`s four first least significant bits and decrease the bytes needed by (1).

After checking if the selected object has bytes less than what is needed to embed the secret image then return to step 1 and choose another cover that is larger with different objects.

b. If everything checks out from the previous step (a) then embedding the secret data will begin in the selected object as stated above by declaring a threshold and selecting the middle pixels between two pixels for embedding and embedding in the LSB of the cover image. What made the author decide to use (4-LSB) at most and not more is because these LSB bits have lower amount of information than the rest 4-MSB. The amount of data that is saved in each bit of a data byte is depicted in Figure 3.2 as a percentage.



8	7	6	5	4	3	2	1
50	24	13	6.8	3.5	1.7	0.7	0.3
MSB							LSB

Figure 3. 2: The percentage of information in each bit of one byte of data

Less than 1% of the information saved in a byte is held by the 2-LSB bits, which pertain to a single byte of data [65]. Therefore, altering the values of the image's 2-LSB bits will render the change invisible to human sight [66], and if there a slight difference in colors values between the left and right pixel of the middle pixel then 4-LSB bits will be used. Every pixel in color images is identified by 3 values, one each for green, blue and red color components. In the suggested technique, we embed 2 or 4 bits per color pixel (max is 12 bit per pixel). This large embedding rate will enable us to enhance the color image's payload capacity without compromising imperceptibility [67]. This process is repeated for each color band(R, G, and B) until all the secret information is embedded in the cover image.

c. Unless the receiver of the stego video knows the precise dimensions (height and width) of the secret image and the bytes needed for embedding the secret data for each color band (B, G, and R), the embedded secret data will not be correctly extracted by the receiver. A new header information system is created and put into place to solve this issue. The receiver will be able to correctly obtain the embedded secret data with the help of this header information. In the suggested method, it is the sender's responsibility to generate n bytes of header information from for example the object start at (X=0, Y=0 to X=?, Y=0) where:

- The first pixel will embed information about the number of information bytes reserved in the R band and how many numbers the bytesneeded have in the G band.
- The (second pixel to information bytes reserved – 2) will have information about the bytes needed of the secret data for each band and each band in these pixels will have its corresponding bytes needed. Because the number is too large and the largest value for colors is 255. A new encoding method was used to store and restore the values on the receiver`s side by embedding two numbers at a time from the bytes needed in each pixel.
- The rest two pixels is for the dimensions of the secret image where a similar to the above method will be used but will need only two bands in each pixel (R, G). So to decrease the height to 255 or lower to put the value in the color it will be divided by 255 and compute a plus number to return the precise value of the height in the receiver side, but in the receiver's side you will multiply the value by 255 then add the plus value. The same will be done for the width.

Algorithm 3.3 illustrates steps (a) and (b) above and algorithm 3.4 shows the embedding of the necessary information about the secret image in step (c) for extracting later.

Algorithm 3. 3: Embedding data
Input: messege_bmp, original_bmp, th, bytesneeded, info_bytes.

Variables definition:
messege_bmp = Bitmap – the secret image.
original_bmp = Bitmap – the cover image.
th = integer - this is the threshold for deciding to embed 2 bits or 4 bits in the pixel.
bytesneeded = Global – integer – number of bytes needed to embed the secret picture in the cover-picture for the color band.
info_bytes = integer – number of bytes needed to embed the information about the secret image.
yStego = integer – current pixel height pointer for the color band initial value is 0.
R_xStego = integer – current pixel width pointer for the color band initial value
count = integer – counter for the number of bits of the secret picture.
original_bmp = Bitmap – the stego picture.
C = Color – template variable to put colors values in temporarily to use and template variable for value of the color of the pixel left to the selected middle pixel.
bin = 1D array of character – store the values of the color band in binary format
y = integer – current pixel height pointer initial value is 0.
x = integer – current pixel width pointer initial value is 0.
C1 = Color – template variable for value of the color of the selected middle pixel.
C2 = Color – template variable for value of the color of the pixel right to the selected middle pixel.
k = integer – counter for the color band secret image bits.
within_th = Boolean – a flag that indicates whether the difference between the value of left and right pixel to the selected pixel is equal or lower than the threshold or not.

Output: The stego image without information about the secret image in the first 6-bytes.

Begin

```
Function Embedding( messege_bmp, original_bmp, th, bytesneeded,
info_bytes)
    begin
        yStego ← 0
        xStego ←  info_bytes
        count ← 0
        stego_bmp ← original_bmp
        Define C
        R_bin ← new char[(messege_bmp Width * messege_bmp
        Height) * 8]
    for y ← 0 to messege_bmp Height Do
        begin
            for x ← 0 to messege_bmp Width Do
                begin
                    C ← messege_bmp GetPixel(x, y)
                    msgdectobin(C, bin, count) // turn values from decimal to binary
                    Increase Count by 8
                End for x
        End for y
    Define C1,C2
    k ← 0
    while k not equal to bin Length Do
        begin
            if  bin Length not equal to k then
                begin
                    within_th ← false
                    while within_th equals false Do
                        begin
                            if  xStego + 2 greater or equal to stego_bmp Width then
                                Increase yStego by 1
                                xStego ← 0
                            Else
                                within_th ← thcheck(stego_bmp GetPixel(xStego, yStego), stego_bmp GetPixel(xStego + 2, yStego), th, ref xStego)
                        End while
                    C = stego_bmp GetPixel(xStego, yStego);
                    C1 = stego_bmp GetPixel(xStego + 1, yStego)
                    C2 = stego_bmp GetPixel(xStego + 2, yStego)
                    stego_bmp SetPixel(xStego + 1, yStego, Color FromArgb(C1.A, msgembed(ref k, C.R, C2.R, C1.R, ref bytesneeded, bin), C1.G, C1.B))
                    Increase xStego by 2
                End if
        End while
    return stego_bmp
End embedding Function

Function thcheck(int color,int color2 , int th, ref int xStego)
begin
    if  Math.Abs(color - color2) lesser r equal to th then
        return true
    Increase xStego by 2
```

```
        return false
    End  thcheck Function

Function msgembed(ref int k, int C,int C2, int C1, ref int
bytesneeded, char[] bin)
    begin
        StringBuilder sb ← new StringBuilder(dectobin(C1))
            if  C - C2 equals 0 and k less than bin.Length – 2 then
                sb[4,5,6,7] ← bin[k,k+1,k+2,k+3]
                Increase k by 4
                Decrease bytesneeded by 1
            Else
                sb[6,7] ← bin[k,k+1]
                Increase k by 2
        return (Convert.ToInt32(sb.ToString(), 2))
    End  msgembed Function
End Algorithm.
```

Algorithm 3. 4: Secret image information embedding

Input: messege_bmp, stego_image_bmp.

Variables definition:
message_bmp = Bitmap – the secret image that was embedded
stego_image_bmp = Bitmap – the stego image that has the secret message embedded in it
mWidth255 = integer – the width of the secret image after it is divided by 255.
mWidthplus = integer – the plus value to return the precis number of the width before it was divided.
mHeight255 = integer – the height of the secret image after it is divided by 255.
mHeightplus = integer – the plus value to return the precis number of the height before it was divided.
info_bytes = integer – how many pixels to reserve for the secret image information embedding
obj_pixel_index = integer – the stego image selected object pixel index pointer
index_of_message_bytes = integer – index that points to the secret message bytes numbers

Output: The stego image with the necessary information about the secret image.

```
Begin
1. mWidth255 ← message_bmp Width / 255
2. mWidthplus ← message_bmp Width - (255 * mWidth255)
3. mHeight255 ← message_bmp Height / 255
4. mHeightplus ← message_bmp Height - (255 * mHeight255)
5. info_bytes ← Floor(Log10(message_bmp Width * message_bmp
   Height * 4) + 1)
6. info_bytes ← Ceiling(info_bytes / 2) + 3
7. Embed in the first pixel of stego_image_bmp selected object the
   info_bytes in the R band and the full secret message how many
   bytes needed number to extract in G band
8. obj_pixel_index ← 1
9. index_of_message_bytes ← 0
10. while obj_pixel_index is lesser than info_bytes – 2 Do
11.   begin
12.       embed two numbers for each color band of the bytes needed in
          the stego_image_bmp, each in its corresponding band(2
          numbers from the bytes needed to extract the message R
          components in the R band and so on for the G and B but each
          one is also embedded in its corresponding band)
13.       increment obj_pixel_index by 1
14.       increment index_of_message_bytes by 2
15.       if index_of_message_bytes equals message bytes needed length
          -1 then
16.          Embed the last number for each color band of the bytes
             needed in the stego_image_bmp, each in its corresponding
             band.
17.   End while
18. Embed mHeight255 in the R band and mHeightplus in the G band
    of stego_image_bmp objects (X + info_bytes - 2).
19. Embed mWidth255 in the R band and mWidthplus in the G band
    of stego_image_bmp objects (X + info_bytes - 1).
End Algorithm
```

3.2.2 The Message Extraction Part

The extraction of the secret message from the stego-video is shown in Figure 3.3 which consists of a set of stages that will be covered in the next subsections.

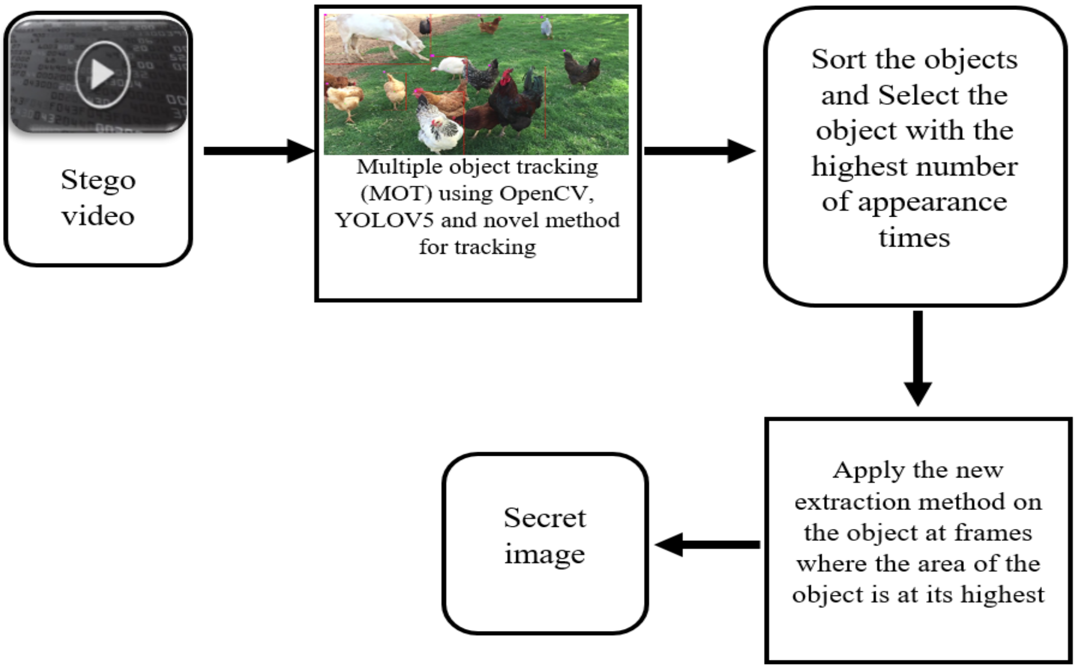

Stego video
Multiple object tracking (MOT) using OpenCV, YOLOV5 and novel method for tracking
Sort the objects and Select the object with the highest number of appearance times
Apply the new extraction method on the object at frames where the area of the object is at its highest
Secret image

Figure 3. 3: Block diagram of the extraction part

3.2.2.1 Video selection and Objects tracking

Choose the video that has the secret message embedded in it (stego-video). Then, apply YoloV5 object detection and new tracking method as in the embedding process to identify the objects.

3.2.2.2 Object Selection preparing

Again as the embedding process sort the detected objects by appearance times in descending form and discard those with 70 or less appearance times and then sort each object appeared in the frames by its area to begin the next step which is the extraction.

3.2.2.3 Secret Image Extraction

Select the object with the highest appearance times and the frames of that object with the highest areas sorted one after another and they as done before have the (n) copies of the secret message embedded in them, the extraction can now commence as following:

a. Retrieve the information of the secret image (bytes needed for each color band R G and B, width, and height) embedded in the first n-bytes of the stego video object and suppose the selected object starts from (X=0,Y=0) then:

- First retrieve the (n = number of information bytes reserved) number embedded in the first (0,0) pixel in the R color band and then retrieve the (L = length of the number of bytes needed) embedded in the G band.
- Extract the R band bytes needed value embedded in the R band at (X=1,Y=0 to X=n-2,Y=0) and like the embedding process each two numbers embedded in the pixels will be extracted and saved in a variable . The same technique will be used for the rest G and B bands but the values retrieved will be in the G for the G color and B for the B color.
- To retrieve the height of the secret image the information was embedded in (n-2,0) in the R and G bands where the value of the R band will be multiplied by 255 and the result will be added to the plus value embedded in the G band. The same will be done for the width but the information will be at (n-1,0) R and G bands.

b. The extraction process will be the same as the embedding process were the agreed upon threshold (5) will be declared and the selection of the pixels will begin from (X=n,Y=0) if for example the object started at(X=0,Y=0) and so on but in the extraction process the middle pixels will be selected and based on the difference between the left and right pixels to the middle pixel the result was greater than (5) then the pixel will be skipped and if it was equal or lower than (5) then if the difference was zero then 4 bits will be retrieved from the middle pixel else 2 bits will be retrieved and so on for the rest of the selected middle pixels. Each 8 bits extracted will be changed to decimal value and put in its color band , so the process above will be done for the R, G and B bands to retrieve the secret image.

Algorithm 3.5 illustrates the extraction process that is stated in step (b) above in detail.

Algorithm 3. 5: Extracting data

Input: stego_bmp, Extracted_msg_bmp, th, bytesneeded, info_bytes.

Variables definition:

stego_bmp = Bitmap – the stego image.

Extracted_msg_bmp = Bitmap – the extracted secret image.

th = integer – this is the threshold for deciding to embed 2 bits or 4 bits in the pixel.

bytesneeded = Global – integer – number of bytes needed to embed the secret picture in the cover-picture for the color band.

info_bytes = integer – number of bytes needed to embed the information about the secret image.

yStego = integer – current pixel height pointer for the stego image color band initial value is 0.

xStego = integer – current pixel width pointer for the stego image color band initial value is (info_bytes).

Ymsg = Global – integer – current pixel height pointer for the secret message color band initial value is 0.

Xmsg = Global – integer – current pixel width pointer for the secret message color band initial value is 0.

NumberOf2BitsRestored = integer – counter for how many times two or four bits have been restored form the stego image or the color band.

s = String – temporary container for the color band secret image retrieved bits.

count = integer – counter for the number of bits of the secret image.

original_bmp = Bitmap – the stego image.

C = Color – template variable for value of the color of the pixel left to the selected middle pixel.

C1 = Color – template variable for value of the color of the selected middle pixel.

C2 = Color – template variable for value of the color of the pixel right to the selected middle pixel.

completed = Boolean – a flag that indicates that the color band secret data has all be extracted from the stego image.

within_th = Boolean – a flag that indicates whether the difference between the value of left and right pixel to the selected pixel is equal or lower than the threshold or not.

Output: the secret image (Extracted_msg_bmp).

Begin

```
1. Function Extracting( stego_bmp, Extracted_msg_bmp, th,
   bytesneeded, info_bytes)
```

```
begin
    yStego ← 0, xStego ← info_bytes
    Ymsg ← 0, Xmsg ← 0
    NumberOf2BitsRestored ← 0
    s ← null
    Define C, C1, C2
    completed ← false
    while completed equals false DO
        begin
            within_th ← false
            while within_th equals false Do
                begin
                    if  xStego + 2 greater or equal to stego_bmp.Width
                    then
                        Increase yStego by 1
                        xStego ← 0
                    Else
                        within_th ← Thcheck(
                        stego_bmp.GetPixel(xStego, yStego),
                        stego_bmp.GetPixel(R_xStego + 2, R_yStego), th,
                        ref R_xStego)
                End while
            C ← stego_bmp.GetPixel(R_xStego, R_yStego)
            C1 ← stego_bmp.GetPixel(R_xStego + 1, R_yStego)
            C2 ← stego_bmp.GetPixel(R_xStego + 2, R_yStego)
            s ← msgextract(ref NumberOf2BitsRestored, C, C2, C1,
            bytesneeded, s, ref xStego)
            Extract_message(Extracted_msg_bmp, ref s)
            if  Ymsg greater or equals Extracted_msg_bmp.Height OR
            NumberOf2BitsRestored equals bytesneeded then
                completed ← true
        End while
    return Extracted_msg_bmp
End Extracting Function

Function thcheck(int color,int color2 , int th, ref int xStego)
begin
    if  Math.Abs(color - color2) lesser r equal to th then
        return true
    Increase xStego by 2
    return false
End thcheck Function

Function Extract_message(Bitmap Extracted_msg_bmp,ref String
s)
```

```
begin
    if  s.Length greater or equals 8 then
        begin
            Color C ← Extracted_msg_bmp.GetPixel(Xmsg, Ymsg)
            if  s.Length greater or equals 9 then
                Extracted_msg_bmp.SetPixel(R_Xmsg, R_Ymsg,
                Color.FromArgb(255, Convert.ToInt32(s.Remove(8,
                2), 2), C.G, C.B))
                s ← s.Substring(8, 2)
            Else
                Extracted_msg_bmp.SetPixel(Xmsg, Ymsg,
                Color.FromArgb(255, Convert.ToInt32(s, 2), C.G,
                C.B))
                s ← null
            Increase Xmsg by 1
            if  Xmsg equals Extracted_msg_bmp.Width then
                begin
                    Increase Ymsg by 1
                    Xmsg ← 0
                End if
End Extract_message Function

Function msgextract(ref int NumberOf2BitsRestored, int C, int C2,
  int C1, int bytesneeded, String s,ref int xStego)
begin
    StringBuilder sb ← new StringBuilder(dectobin(C1))
    If  C - C2 equals 0 and NumberOf2BitsRestored less than
    bytesneeded then
        if  NumberOf2BitsRestored  equals  bytesneeded  -  1  And
        s.Length equals 6 then
            s ← s + sbb[6] + sbb[7]
            Increase NumberOf2BitsRestored by 1
        Else
            s ← s + sbb[4] + sbb[5] + sbb[6] + sbb[7]
            Increase NumberOf2BitsRestored by 1
    Else
        s ← s + sbb[6] + sbb[7]
        Increase NumberOf2BitsRestored by 1
    Increase xStego by 2
    return s
End msgextract Function
End Algorithm.
```

Chapter Four
RESULTS AND DISCUSSION

4.1 Overview

To determine which method is more effective, steganography offers an evaluation mechanism for steganographic systems. No test or measurement is regarded as being standard at this time. However, there are guidelines and common procedures that can be applied for assessment [25]. This project's main goal is to secretly store an immense amount of data while maintaining the video's quality. This Chapter begins with the software and hardware requirements in implementing the proposed system. Afterwards, the proposed system is assessed by utilizing a number of components (payload, quality), and videos with various sizes as a dataset for evaluation.

4.2 Software and Hardware

The proposed system was done on a Lenovo Legion Y520 laptop with a windows 10 pro Version (21H2) and the following specifications:

- Intel Core i7 7700HQ CPU @ 2.8GHz.
- 16GB Ram.
- Samsung 128GB ssd.
- Nvidia GeForce GTX 1050 4GB

The program was coded on visual studio 2019 and two programing languages were used (C++, C#) using Common Language Runtime (CLR) also OpenCV, YoloV5, cuDNN, Eigen and ffmpeg libraries were used throughout the project.

4.3 Experiments and Results Discussion

The experiments that were conducted and discussion of the results that were obtained is presented throughout this section. Our algorithm has been evaluated on a dataset of cover videos to assess its efficacy. Our dataset consists of 4 random MP4 videos and 5 random PNG secret images, which are gathered from the internet.

PSNR is used to evaluate the video's quality upon embedding. The hidden information is invisible to human vision if the PSNR result is equal to or greater than 30 dB [2, 7]. The distortion between the stego video and the cover video is assessed using MSE. The following figures illustrates the result from the program itself where:

- Figure 4.1 shows the Yolo Objects detection and tracking of each object in the video by giving them unique IDs.

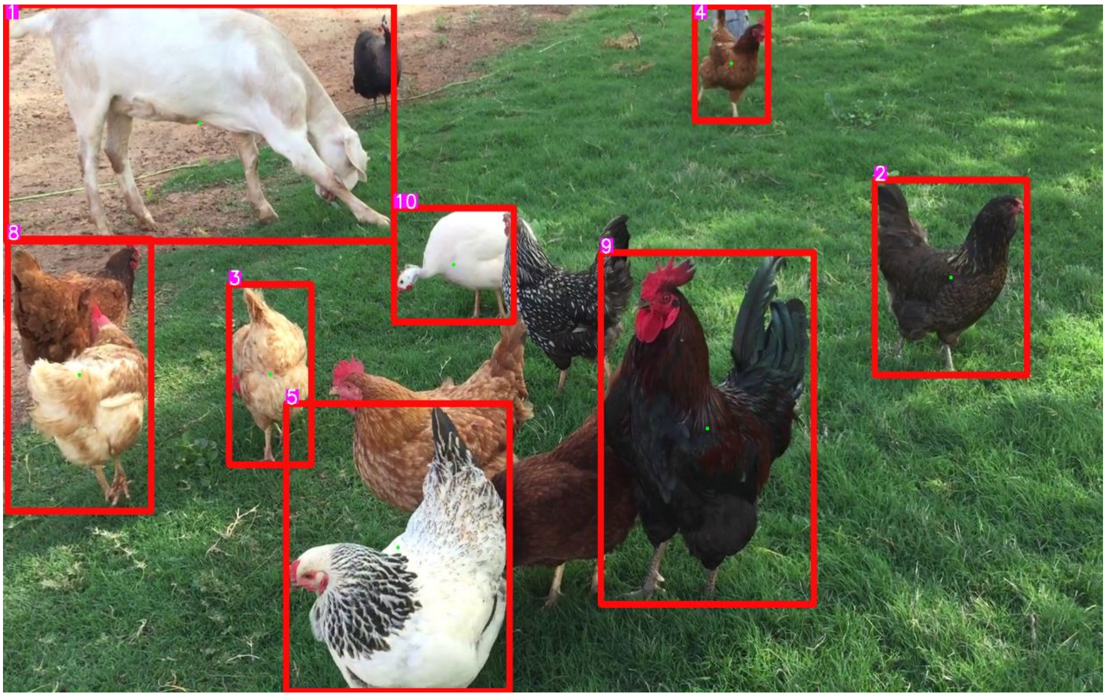
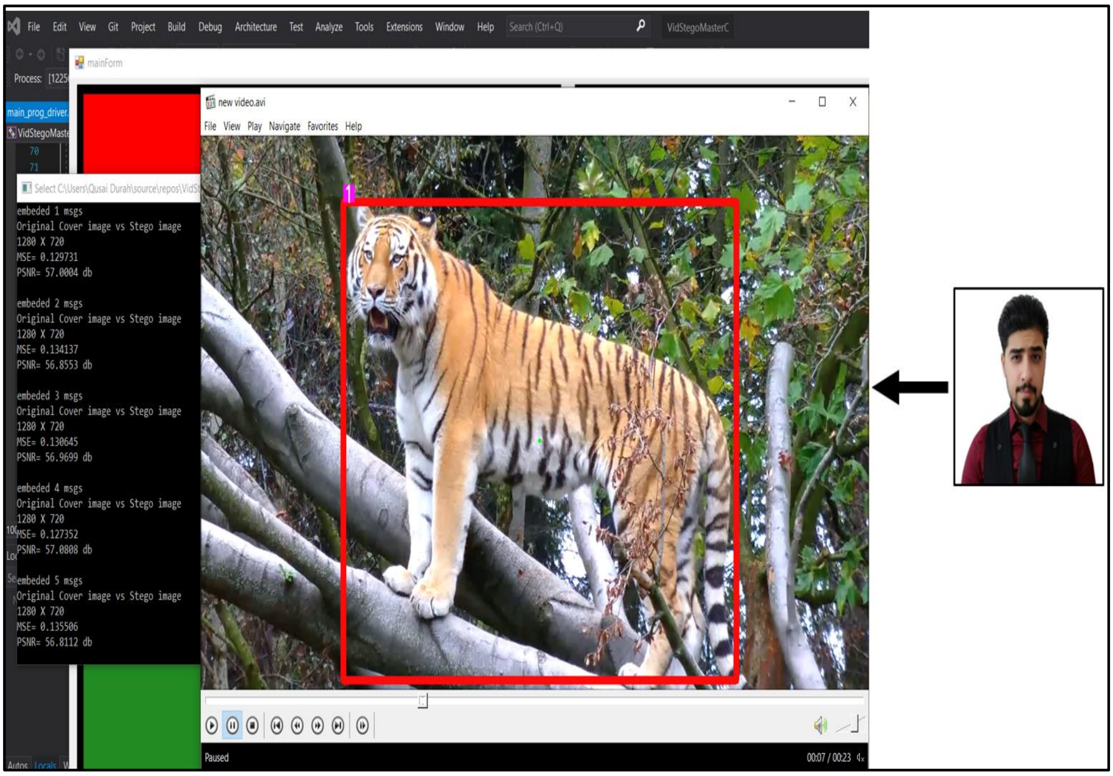


Figure 4. 1: Objects detection and tracking

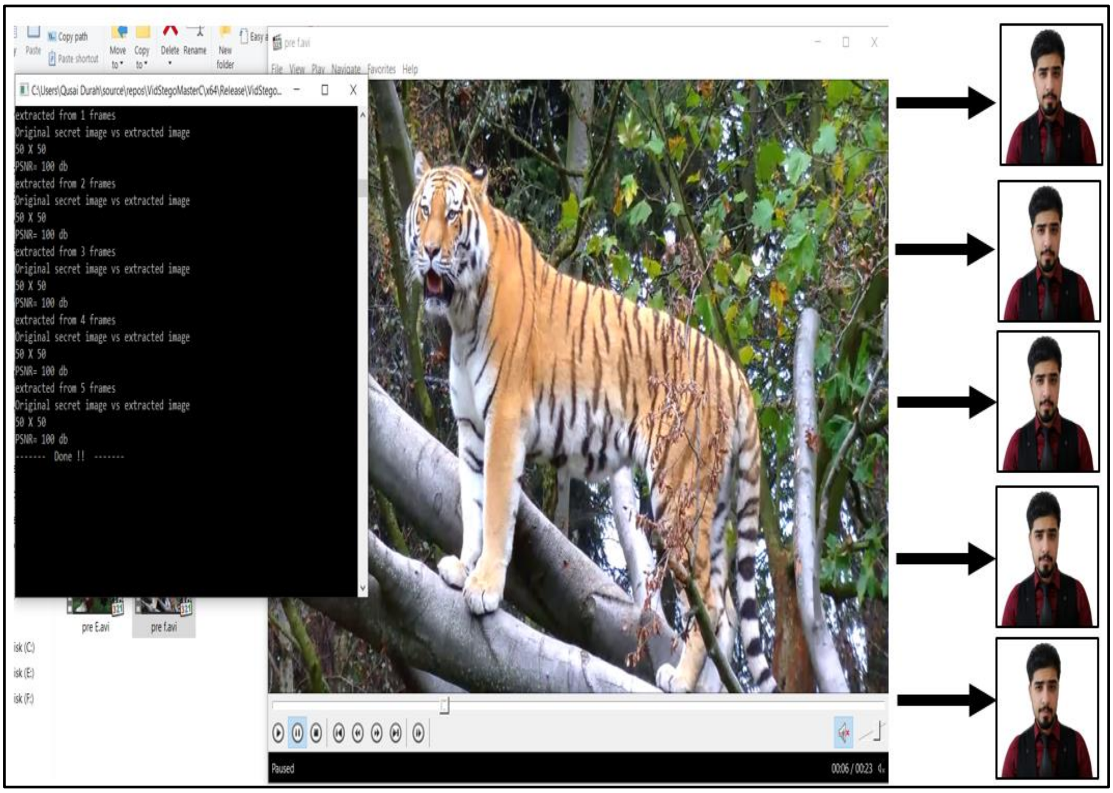
- Figure 4.2 illustrates the results of embedding five PNG secret image of size 50x50 in a 23 seconds 720p 30fps avi video in the lion which is the highest appeared object through the video, and figure 4.3 shows the extraction results of the same video and object after embedding.



```
embeded 1 msgs
Original Cover image vs Stego image
1280 X 720
MSE= 0.129731
PSNR= 57.0004 db

embeded 2 msgs
Original Cover image vs Stego image
1280 X 720
MSE= 0.134137
PSNR= 56.8553 db

embeded 3 msgs
Original Cover image vs Stego image
1280 X 720
MSE= 0.130645
PSNR= 56.9699 db

embeded 4 msgs
Original Cover image vs Stego image
1280 X 720
MSE= 0.127352
PSNR= 57.0808 db

embeded 5 msgs
Original Cover image vs Stego image
1280 X 720
MSE= 0.135506
PSNR= 56.8112 db
```

File Edit View Git Project Build Debug Architecture Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
VidStegoMasterC
mainForm
Process: [1225
main_prog_driver.
VidStegoMaste
Select C:\Users\Qusai Durah\source\repos\VidS
new video.avi
File View Play Navigate Favorites Help
Paused
00:07 / 00:23

Figure 4. 2: Results of embedding five copies of 50x50 secret image in 720p video

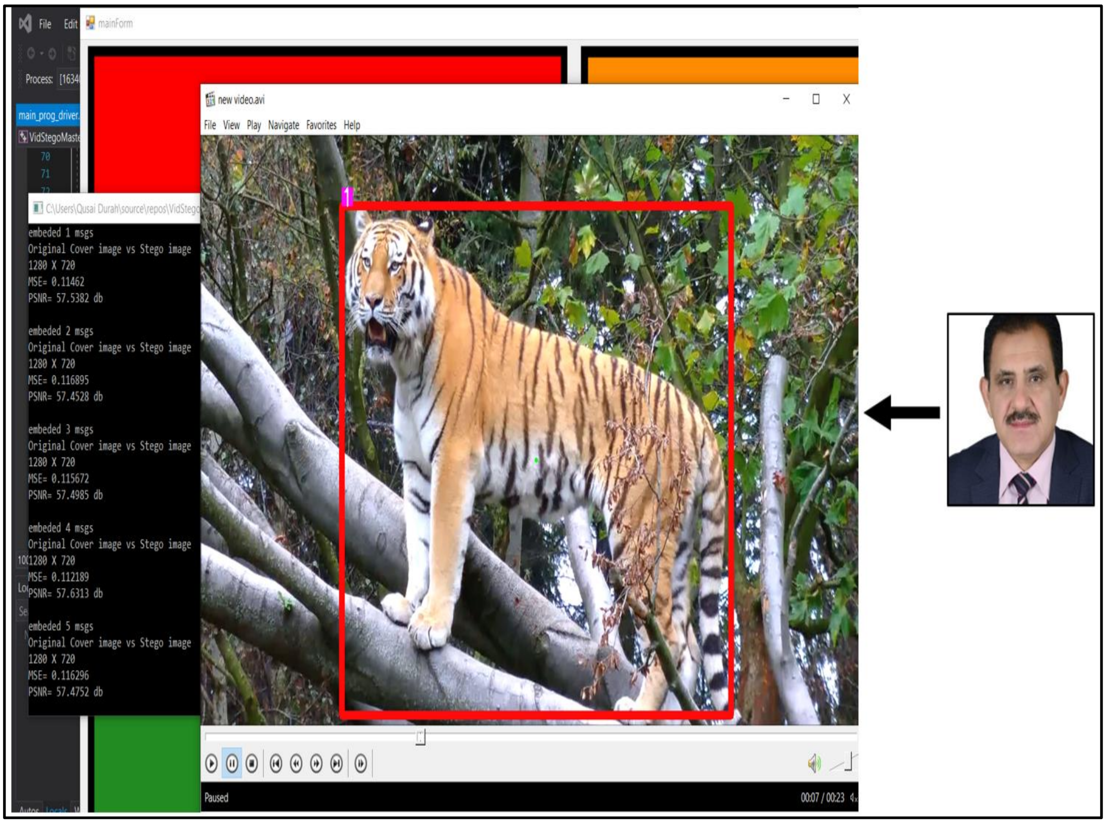


```
extracted from 1 frames
Original secret image vs extracted image
50 X 50
PSNR= 100 db
extracted from 2 frames
Original secret image vs extracted image
50 X 50
PSNR= 100 db
extracted from 3 frames
Original secret image vs extracted image
50 X 50
PSNR= 100 db
extracted from 4 frames
Original secret image vs extracted image
50 X 50
PSNR= 100 db
extracted from 5 frames
Original secret image vs extracted image
50 X 50
PSNR= 100 db
-------  Done !!  -------
```

Figure 4. 3: Results of extracting the five copies of 50x50 secret image in 720p video

- Figure 4.4 illustrates the results of embedding five PNG secret image of size 50x50 in the same 23 seconds 720p 30fps avi video in the lion which is the highest appeared object through the video, and figure 4.5

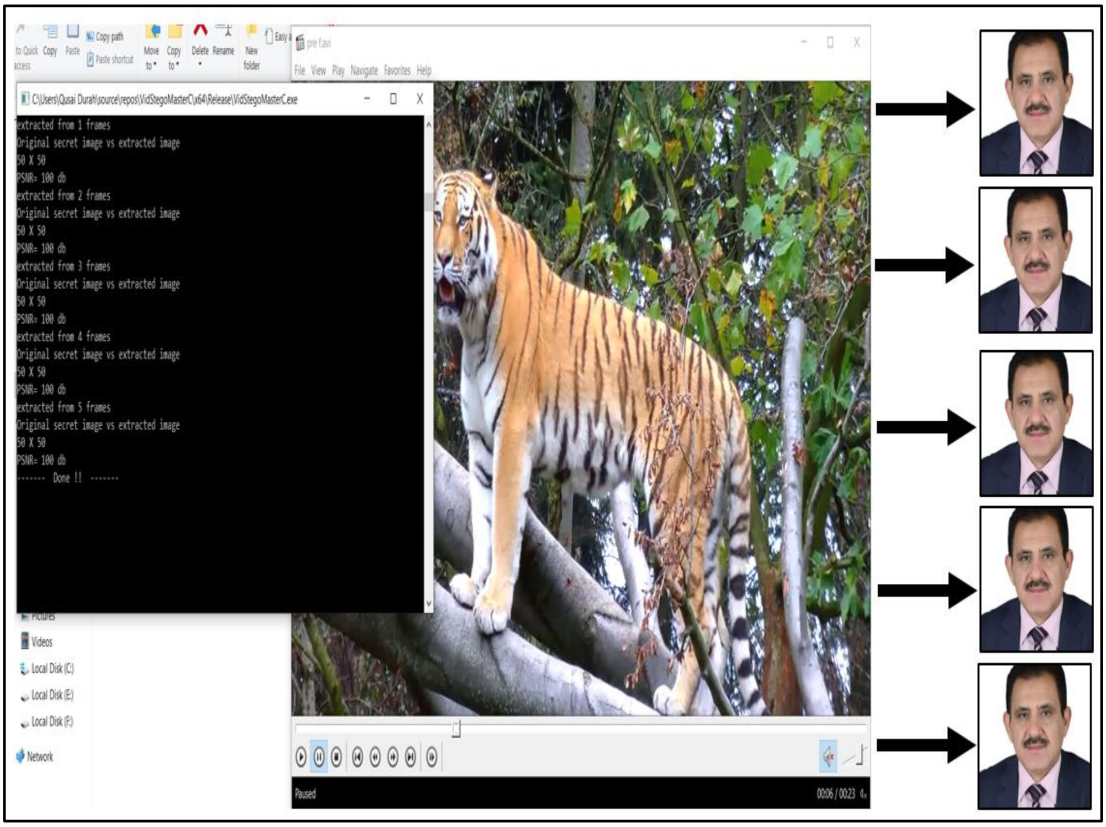
shows the extraction results of the same video and object after embedding.



C:\Users\Qusai Durah\source\repos\VidStego

```
embeded 1 msgs
Original Cover image vs Stego image
1280 X 720
MSE= 0.11462
PSNR= 57.5382 db

embeded 2 msgs
Original Cover image vs Stego image
1280 X 720
MSE= 0.116895
PSNR= 57.4528 db

embeded 3 msgs
Original Cover image vs Stego image
1280 X 720
MSE= 0.115672
PSNR= 57.4985 db

embeded 4 msgs
Original Cover image vs Stego image
1280 X 720
MSE= 0.112189
PSNR= 57.6313 db

embeded 5 msgs
Original Cover image vs Stego image
1280 X 720
MSE= 0.116296
PSNR= 57.4752 db
```

Figure 4. 4: Results of embedding five copies of 50x50 secret image in 720p video

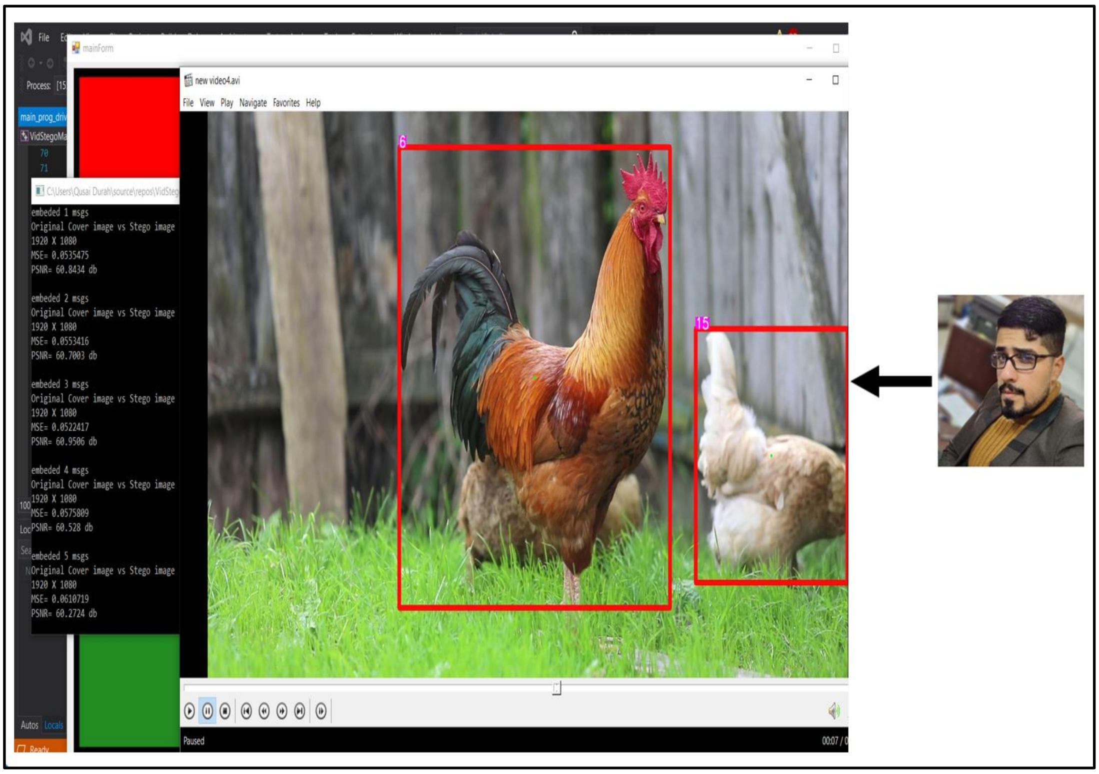


C:\Users\Qusai Durah\source\repos\VidStegoMasterC\x64\Release\VidStegoMasterC.exe

```
extracted from 1 frames
Original secret image vs extracted image
50 X 50
PSNR= 100 db
extracted from 2 frames
Original secret image vs extracted image
50 X 50
PSNR= 100 db
extracted from 3 frames
Original secret image vs extracted image
50 X 50
PSNR= 100 db
extracted from 4 frames
Original secret image vs extracted image
50 X 50
PSNR= 100 db
extracted from 5 frames
Original secret image vs extracted image
50 X 50
PSNR= 100 db
------- Done !! -------
```

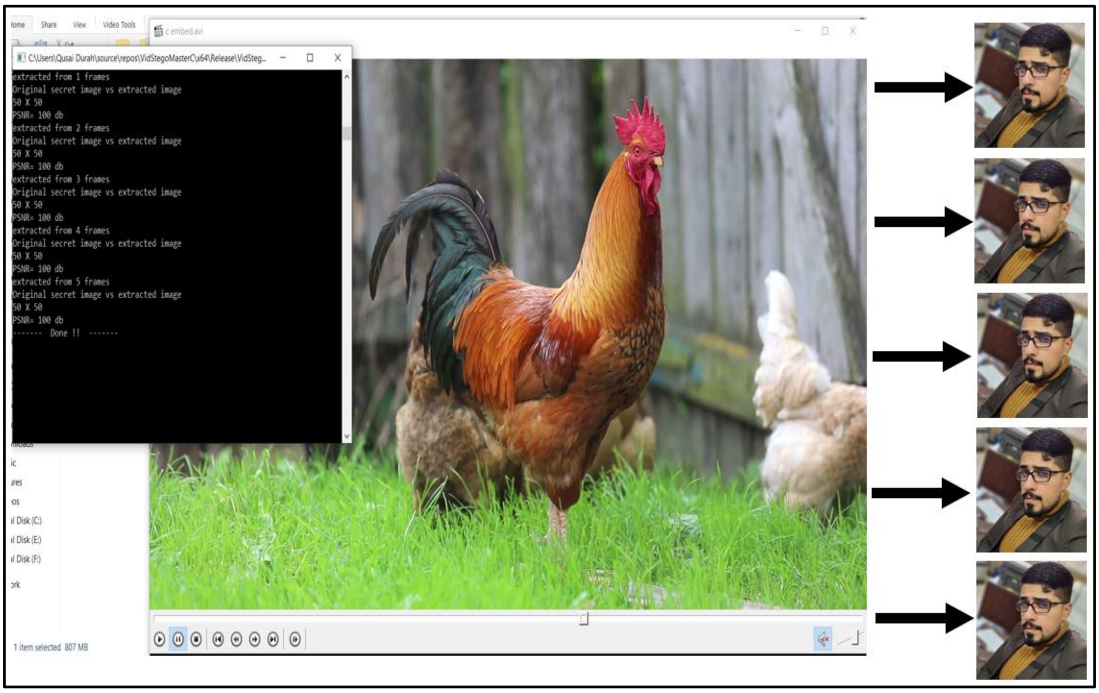
Figure 4. 5: Results of extracting the five copies of 50x50 secret image in 720p video

- Figure 4.6 illustrates the results of embedding five PNG secret image of size 50x50 in a 13 seconds 1080p 30fps avi video in the rooster which is the highest appeared object through the video, and figure 4.7 shows the extraction results of the same video and object after embedding.



```
C:\Users\Qusai Durah\source\repos\VidSteg
embeded 1 msgs
Original Cover image vs Stego image
1920 X 1080
MSE= 0.0535475
PSNR= 60.8434 db

embeded 2 msgs
Original Cover image vs Stego image
1920 X 1080
MSE= 0.0553416
PSNR= 60.7003 db

embeded 3 msgs
Original Cover image vs Stego image
1920 X 1080
MSE= 0.0522417
PSNR= 60.9506 db

embeded 4 msgs
Original Cover image vs Stego image
1920 X 1080
MSE= 0.0575809
PSNR= 60.528 db

embeded 5 msgs
Original Cover image vs Stego image
1920 X 1080
MSE= 0.0610719
PSNR= 60.2724 db
```

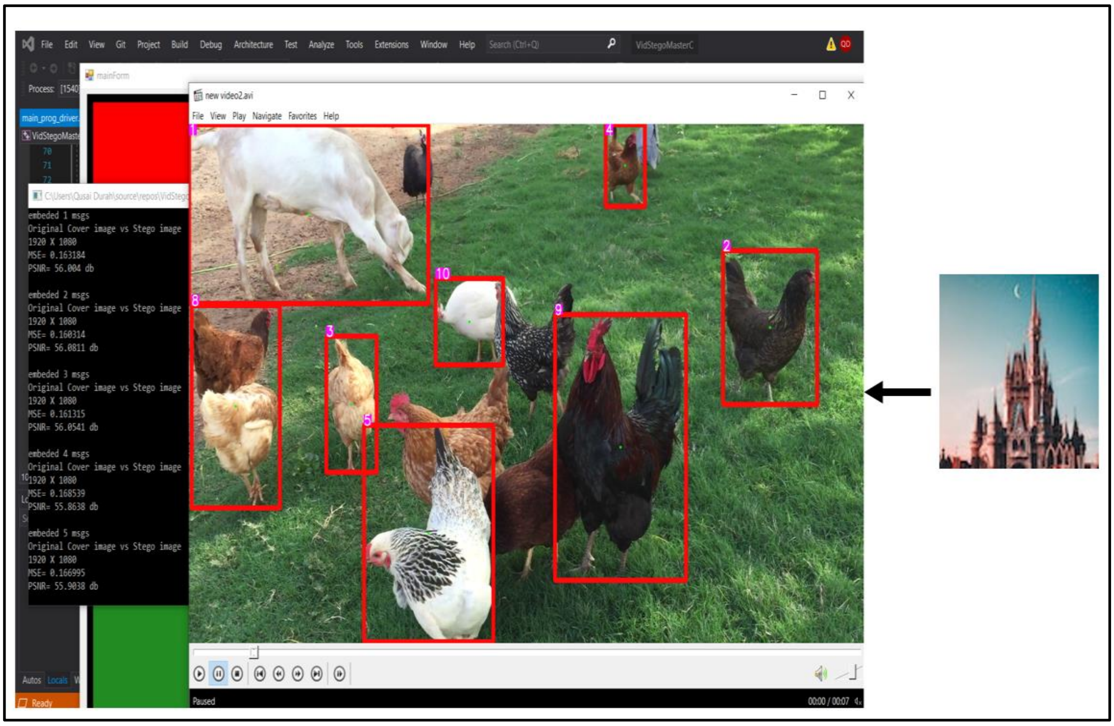
Figure 4. 6: Results of embedding five copies of 50x50 secret image in 1080p video



```
C:\Users\Qusai Durah\source\repos\VidStegoMasterC\x64\Release\VidSteg...
extracted from 1 frames
Original secret image vs extracted image
50 X 50
PSNR= 100 db
extracted from 2 frames
Original secret image vs extracted image
50 X 50
PSNR= 100 db
extracted from 3 frames
Original secret image vs extracted image
50 X 50
PSNR= 100 db
extracted from 4 frames
Original secret image vs extracted image
50 X 50
PSNR= 100 db
extracted from 5 frames
Original secret image vs extracted image
50 X 50
PSNR= 100 db
-------  Done !!  -------
```

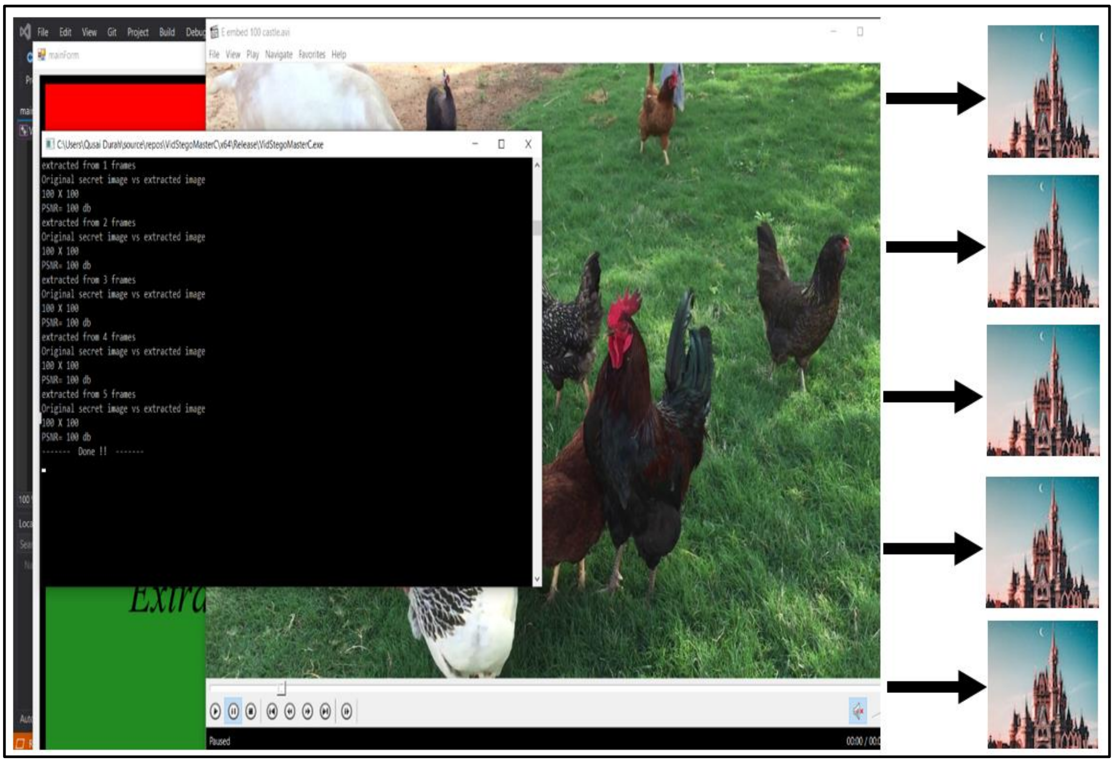
Figure 4. 7: Results of extracting the five copies of 50x50 secret image in 1080p video

- Figure 4.8 illustrates the results of embedding five PNG secret image of size 100x100 in a 7 seconds 1080p 60fps avi video in the dog which is the highest appeared object through the video, and figure 4.9 shows the extraction results of the same video and object after embedding.



```
C:\Users\Qusai Durah\source\repos\VidSteg
embeded 1 msgs
Original Cover image vs Stego image
1920 X 1080
MSE= 0.163184
PSNR= 56.004 db

embeded 2 msgs
Original Cover image vs Stego image
1920 X 1080
MSE= 0.160314
PSNR= 56.0811 db

embeded 3 msgs
Original Cover image vs Stego image
1920 X 1080
MSE= 0.161315
PSNR= 56.0541 db

embeded 4 msgs
Original Cover image vs Stego image
1920 X 1080
MSE= 0.168539
PSNR= 55.8638 db

embeded 5 msgs
Original Cover image vs Stego image
1920 X 1080
MSE= 0.166995
PSNR= 55.9038 db
```

Figure 4. 8: Results of embedding five copies of 100x100 secret image in 1080p video

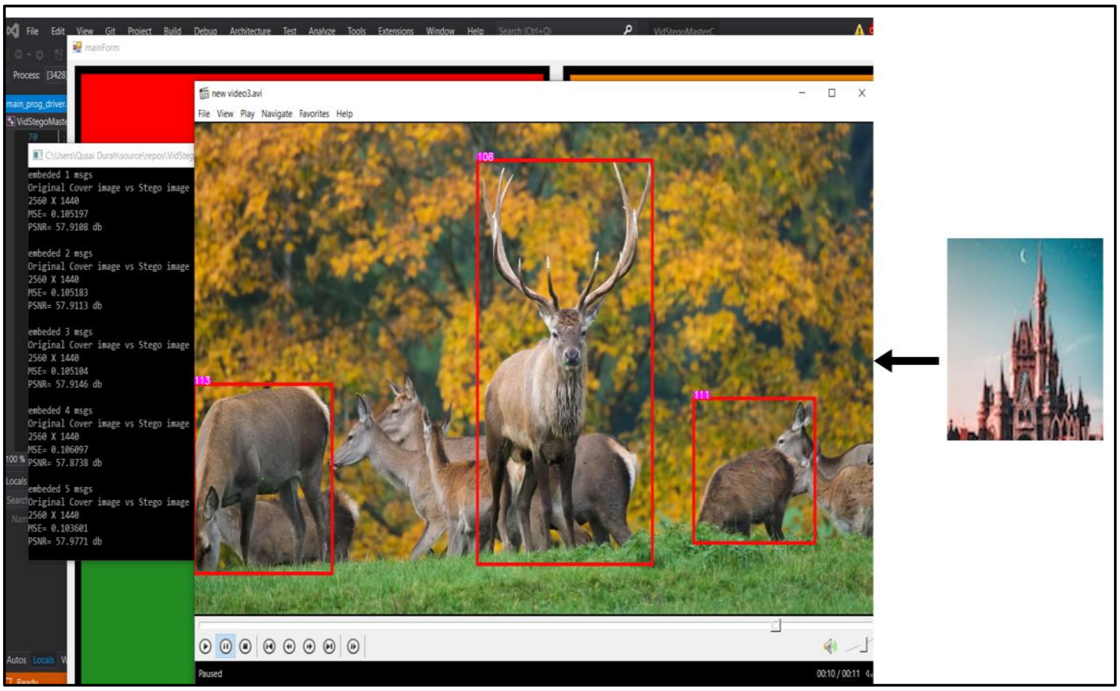


```
C:\Users\Qusai Durah\source\repos\VidStegoMasterC\x64\Release\VidStegoMasterC.exe
extracted from 1 frames
Original secret image vs extracted image
100 X 100
PSNR= 100 db
extracted from 2 frames
Original secret image vs extracted image
100 X 100
PSNR= 100 db
extracted from 3 frames
Original secret image vs extracted image
100 X 100
PSNR= 100 db
extracted from 4 frames
Original secret image vs extracted image
100 X 100
PSNR= 100 db
extracted from 5 frames
Original secret image vs extracted image
100 X 100
PSNR= 100 db
-------  Done !!  -------
```

File Edit View Git Project Build Debug
mainForm
E embed 100 castle.avi
File View Play Navigate Favorites Help
Extra
Paused

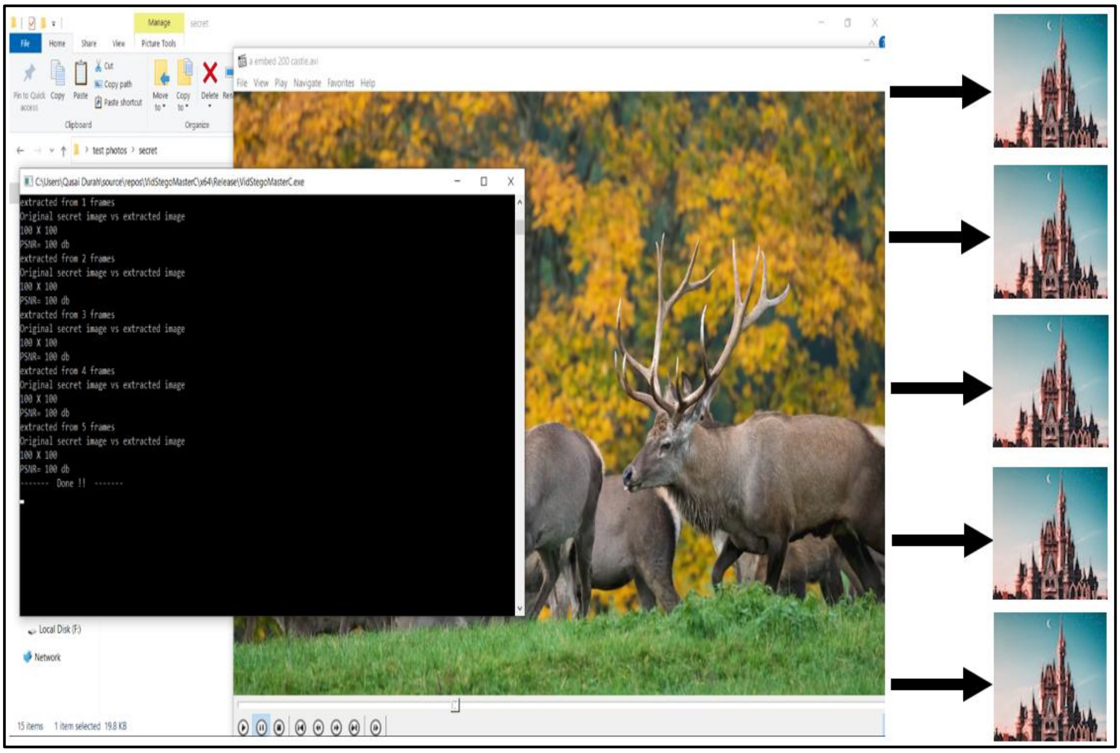
Figure 4. 9: Results of extracting the five copies of 100x100 secret image in 1080p video

- Figure 4.10 illustrates the results of embedding the same five PNG secret image of size 100x100 but this time in a 11 seconds 1440p 30fps avi video in the elk which is the highest appeared object through the video, and figure 4.11 also shows the extraction results of the same video and object after embedding.


mainForm
new video3.avi
File View Play Navigate Favorites Help
C:\Users\Qusai Durah\source\repos\VidSteg
108
113
111
Paused
00:10 / 00:11

```
embeded 1 msgs
Original Cover image vs Stego image
2560 X 1440
MSE= 0.105197
PSNR= 57.9108 db

embeded 2 msgs
Original Cover image vs Stego image
2560 X 1440
MSE= 0.105183
PSNR= 57.9113 db

embeded 3 msgs
Original Cover image vs Stego image
2560 X 1440
MSE= 0.105104
PSNR= 57.9146 db

embeded 4 msgs
Original Cover image vs Stego image
2560 X 1440
MSE= 0.106097
PSNR= 57.8738 db

embeded 5 msgs
Original Cover image vs Stego image
2560 X 1440
MSE= 0.103601
PSNR= 57.9771 db
```

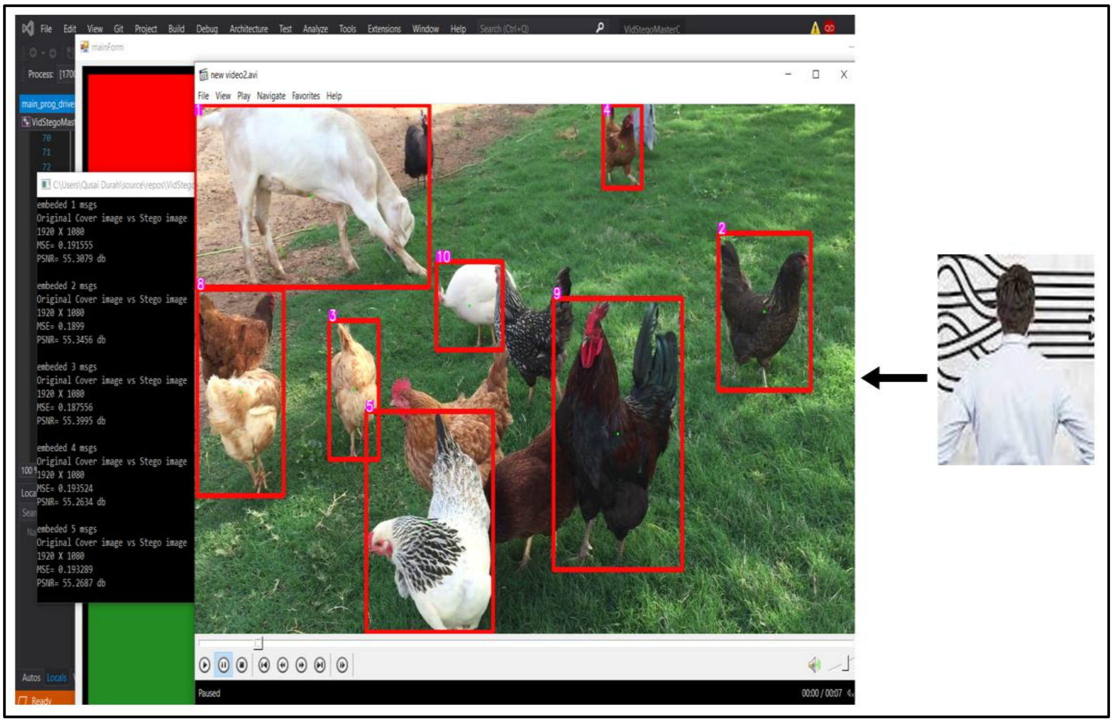
Figure 4. 10: Results of embedding five copies of 100x100 secret image in 1440p video


Manage
secret
File Home Share View Picture Tools
Pin to Quick access
Copy
Paste
Cut
Copy path
Paste shortcut
Clipboard
Move to
Copy to
Delete
Organize
test photos > secret
3 embed 200 castle.avi
File View Play Navigate Favorites Help
C:\Users\Qusai Durah\source\repos\VidStegoMasterC\x64\Release\VidStegoMasterC.exe
Local Disk (F:)
Network
15 items 1 item selected 19.8 KB

```
extracted from 1 frames
Original secret image vs extracted image
100 X 100
PSNR= 100 db
extracted from 2 frames
Original secret image vs extracted image
100 X 100
PSNR= 100 db
extracted from 3 frames
Original secret image vs extracted image
100 X 100
PSNR= 100 db
extracted from 4 frames
Original secret image vs extracted image
100 X 100
PSNR= 100 db
extracted from 5 frames
Original secret image vs extracted image
100 X 100
PSNR= 100 db
------- Done !! -------
```

Figure 4. 11: Results of extracting the five copies of 100x100 secret image in 1440p video

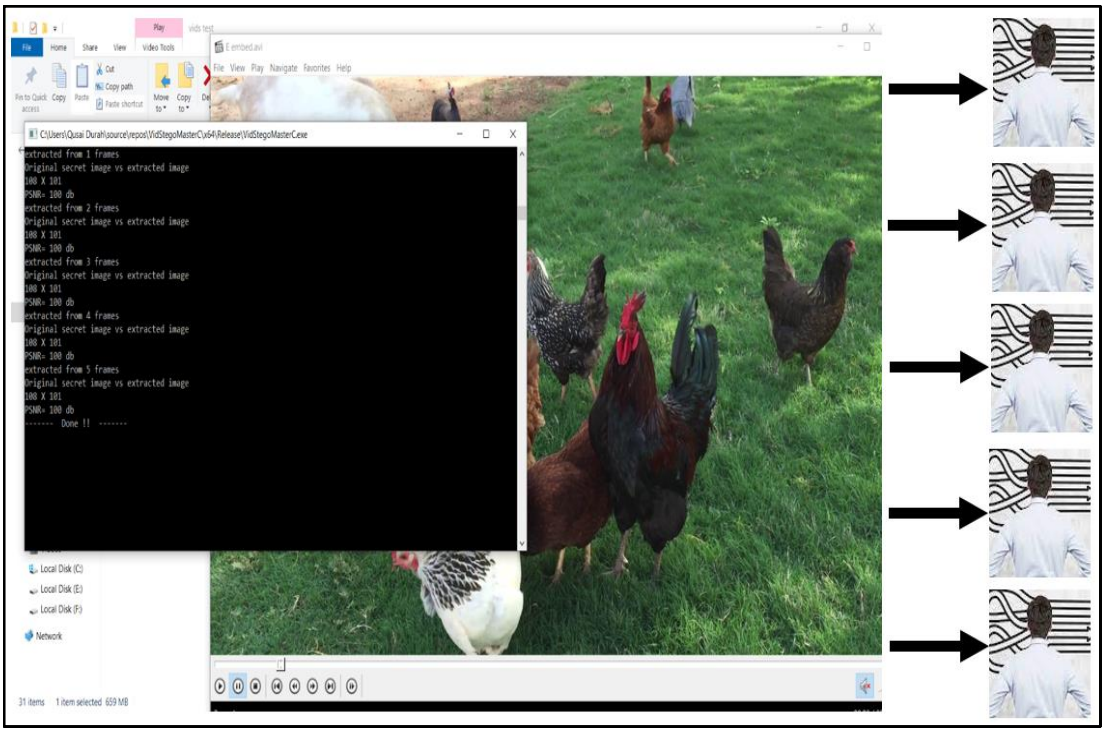
- Figure 4.12 illustrates the results of embedding five PNG secret image of size 108x101 in a 7 seconds 1080p 60fps avi video in the dog which is the highest appeared object through the video, and figure 4.13 shows the extraction results of the same video and object after embedding.



```
embeded 1 msgs
Original Cover image vs Stego image
1920 X 1080
MSE= 0.191555
PSNR= 55.3079 db

embeded 2 msgs
Original Cover image vs Stego image
1920 X 1080
MSE= 0.1899
PSNR= 55.3456 db

embeded 3 msgs
Original Cover image vs Stego image
1920 X 1080
MSE= 0.187556
PSNR= 55.3995 db

embeded 4 msgs
Original Cover image vs Stego image
1920 X 1080
MSE= 0.193524
PSNR= 55.2634 db

embeded 5 msgs
Original Cover image vs Stego image
1920 X 1080
MSE= 0.193289
PSNR= 55.2687 db
```

C:\Users\Qusai Durah\source\repos\VidStego
new video2.avi
File View Play Navigate Favorites Help
1
2
3
4
5
8
9
10
Paused
00:00 / 00:07

Figure 4. 12: Results of embedding five copies of 108x101 secret image in 1080p video

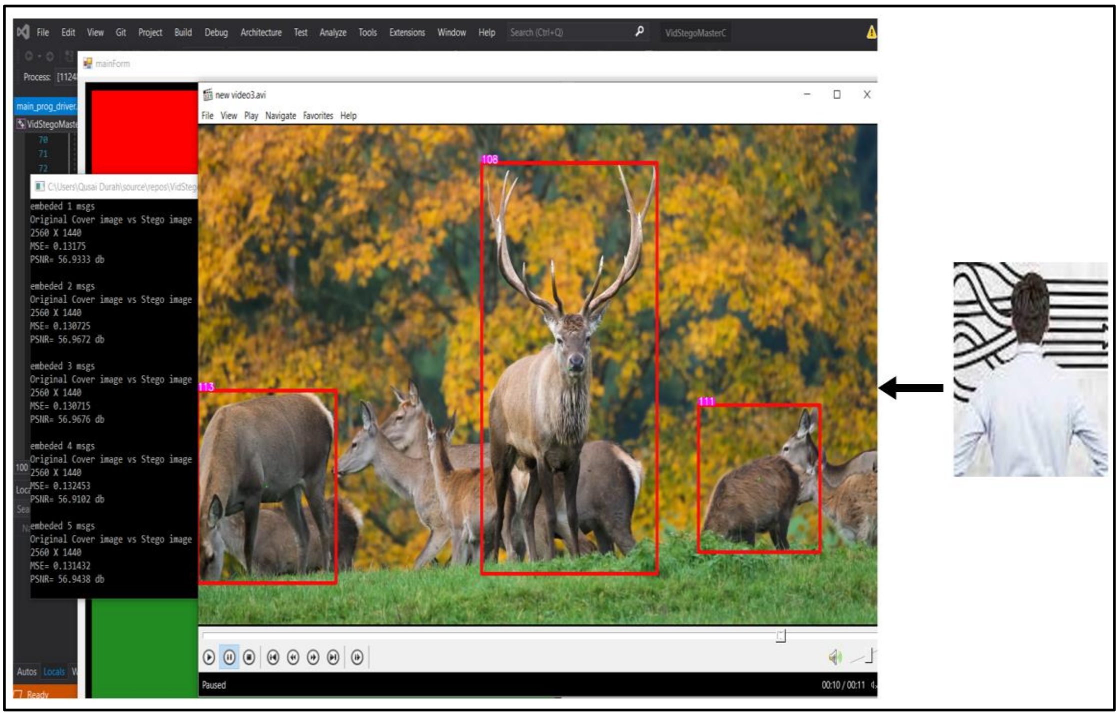


```
extracted from 1 frames
Original secret image vs extracted image
108 X 101
PSNR= 100 db
extracted from 2 frames
Original secret image vs extracted image
108 X 101
PSNR= 100 db
extracted from 3 frames
Original secret image vs extracted image
108 X 101
PSNR= 100 db
extracted from 4 frames
Original secret image vs extracted image
108 X 101
PSNR= 100 db
extracted from 5 frames
Original secret image vs extracted image
108 X 101
PSNR= 100 db
------- Done !! -------
```

C:\Users\Qusai Durah\source\repos\VidStegoMasterC\x64\Release\VidStegoMasterC.exe
E embed.avi
File View Play Navigate Favorites Help
Play
vids test
File
Home
Share
View
Video Tools
Local Disk (C:)
Local Disk (E:)
Local Disk (F:)
Network
31 items 1 item selected 659 MB

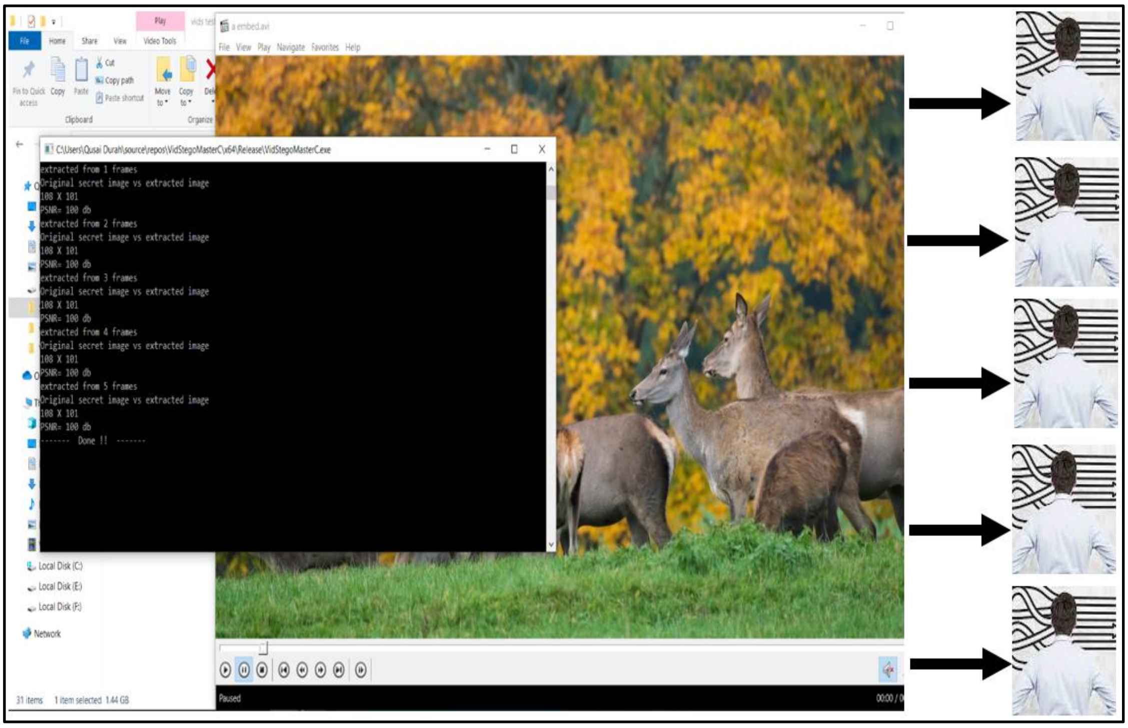
Figure 4. 13: Results of extracting the five copies of 108x101 secret image in 1080p video

- Figure 4.14 illustrates the results of embedding the same five PNG secret image of size 108x101 but this time in a 11 seconds 1440p 30fps avi video in the elk which is the highest appeared object through the video, and figure 4.15 also shows the extraction results of the same video and object after embedding



```
embeded 1 msgs
Original Cover image vs Stego image
2560 X 1440
MSE= 0.13175
PSNR= 56.9333 db

embeded 2 msgs
Original Cover image vs Stego image
2560 X 1440
MSE= 0.130725
PSNR= 56.9672 db

embeded 3 msgs
Original Cover image vs Stego image
2560 X 1440
MSE= 0.130715
PSNR= 56.9676 db

embeded 4 msgs
Original Cover image vs Stego image
2560 X 1440
MSE= 0.132453
PSNR= 56.9102 db

embeded 5 msgs
Original Cover image vs Stego image
2560 X 1440
MSE= 0.131432
PSNR= 56.9438 db
```

Figure 4. 14: Results of embedding five copies of 108x101 secret image in 1440p video



C:\Users\Qusai Durah\source\repos\VidStegoMasterC\x64\Release\VidStegoMasterC.exe

```
extracted from 1 frames
Original secret image vs extracted image
108 X 101
PSNR= 100 db
extracted from 2 frames
Original secret image vs extracted image
108 X 101
PSNR= 100 db
extracted from 3 frames
Original secret image vs extracted image
108 X 101
PSNR= 100 db
extracted from 4 frames
Original secret image vs extracted image
108 X 101
PSNR= 100 db
extracted from 5 frames
Original secret image vs extracted image
108 X 101
PSNR= 100 db
-------  Done !!  -------
```

Figure 4. 15: Results of extracting the five copies of 108x101 secret image in 1440p video

Table 4. 1: Test results of various videos and secret images from the project

Num. of fig.	Num. of Secret image copies	Cover video dimensions	Secret image dimensions	payload	Stego-Video MSE	Stego-Video PSNR	Extracted secret image PSNR
2,3	(1)	1280x720	50x50	20,000	0.129731	57.0004 dB	100.00 dB
	(2)				0.134137	56.8553 dB	100.00 dB
	(3)				0.130645	56.9699 dB	100.00 dB
	(4)				0.127352	57.0808 dB	100.00 dB
	(5)				0.135506	56.8112 dB	100.00 dB
4,5	(1)	1280x720	50x50	20,000	0.11462	57.5382 dB	100.00 dB
	(2)				0.116895	57.4528 dB	100.00 dB
	(3)				0.115672	57.4985 dB	100.00 dB
	(4)				0.112189	57.6313 dB	100.00 dB
	(5)				0.116296	57.4752 dB	100.00 dB
6,7	(1)	1920x1080	50x50	20,000	0.053547	60.8434 dB	100.00 dB
	(2)				0.055341	60.7003 dB	100.00 dB
	(3)				0.052241	60.9506 dB	100.00 dB
	(4)				0.057580	60.528 dB	100.00 dB
	(5)				0.061071	60.2724 dB	100.00 dB
8,9	(1)	1920x1080	100x100	80,000	0.163184	56.004 dB	100.00 dB
	(2)				0.160314	56.0811 dB	100.00 dB
	(3)				0.161315	56.0541 dB	100.00 dB
	(4)				0.168539	55.8638 dB	100.00 dB
	(5)				0.166995	55.9038 dB	100.00 dB
10,11	(1)	2560x1440	100x100	80,000	0.105197	57.9108 dB	100.00 dB
	(2)				0.105183	57.9113 dB	100.00 dB
	(3)				0.105104	57.9146 dB	100.00 dB
	(4)				0.106097	57.8738 dB	100.00 dB
	(5)				0.103601	57.9771 dB	100.00 dB
12,13	(1)	1920x1080	108x101	87,264	0.191555	55.3079 dB	100.00 dB
	(2)				0.1899	55.3456 dB	100.00 dB
	(3)				0.187556	55.3995 dB	100.00 dB
	(4)				0.193524	55.2634 dB	100.00 dB
	(5)				0.193289	55.2687 dB	100.00 dB
14,15	(1)	2560x1440	108x101	87,264	0.13175	56.9333 dB	100.00 dB
	(2)				0.130725	56.9672 dB	100.00 dB
	(3)				0.130715	56.9676 dB	100.00 dB
	(4)				0.132453	56.9102 dB	100.00 dB
	(5)				0.131432	56.9438 dB	100.00 dB

As you can see from the table 4.1 above the results show good PSNR values and a PSNR of 100dB that shows that the secret message embedded is the exact same without loss of information whatsoever before embedding and after extraction from the stego-video.

Chapter Five

CONCLUSION AND FUTURE WORKS

5.1 Overview

This chapter provides the conclusion. We also offer some suggestions for utilizing the proposed system most effectively and minimizing failure risk. Finally, several future works efforts are proposed.

5.2 Conclusion

The capacity, imperceptibility, and robustness of the data hiding process are improved with the use of steganography techniques and algorithms. This thesis presents a novel algorithm that addresses and enhances imperceptibility and robustness while maintaining high capacity. The technique is based on object appearances times, areas, and LSB substitution. The primary contribution of our study is a novel approach to data embedding that employs threshold-based secret data concealing and lossless secret data extraction.

Imperceptibility of the steganography system is evaluated by measuring the PSNR of the resulting stego images selected from the video.

Robustness is enhanced by embedding secret data in the object with the highest appearance times and area throughout the video frames and the data is embedded in the pixels the objects depending on a threshold to make it even harder to detect and retrieve the secret information.

Capacity is further enhanced by sometimes embedding secret information into the first 4 LSB bits instead of just 2. Data embedding into 4 LSB bits is decided upon based on an indicator. The technique often results in a 30% improvement in capacity over embedding into just 2 LSB bits.

5.3 Recommendations

Since steganographic systems cannot guarantee complete security, caution must be used while hiding secret information. First, the cover video needs to be the right size to fit the secret data inside of it, with the data typically taking up no more than 25% of the cover video frame. Second, a 1080p or higher resolution cover video is preferred. Third, when intending to insert information, the cover video should be new and unavailable. Fourth, the selected cover video should have large and clear objects as much as possible.

5.4 Future Work

Many directions can be highlighted based on the outcomes of this present thesis. These include, but are not limited to:

- Making the proposed method work for text data.
- Encrypting the secret data before embedding it in the cover video.
- The proposed system opens the door for further research on object tracking by enhancing the new tracking algorithm for better accuracy.

REFERENCES

[1] M. Ramalingam and N. A. Mat Isa, "Video Steganography based on Integer Haar Wavelet Transforms for Secured Data Transfer," *Indian Journal of Science and Technology*, vol. 7, pp. 897-904, 07/01 2014, doi: 10.17485/ijst/2014/v7i7.4.

[2] S. A. Abbas, T. I. B. E. Arif, F. F. M. Ghaleb, and S. M. Khamis, "Optimized video steganography using Cuckoo Search algorithm," *in 2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems* (ICICIS), 12-14 Dec. 2015 2015, pp. 572-577, doi: 10.1109/IntelCIS.2015.7397279.

[3] R. J. Mstafa and K. M. Elleithy, "A novel video steganography algorithm in the wavelet domain based on the KLT tracking algorithm and BCH codes," *in 2015 Long Island Systems, Applications and Technology*, 1-1 May 2015 2015, pp. 1-7, doi: 10.1109/LISAT.2015.7160192.

[4] K. B. Sudeepa, K. Raju, H. S. Ranjan Kumar, and G. Aithal, "A New Approach for Video Steganography Based on Randomization and Parallelization," *Procedia Computer Science*, vol. 78, pp. 483-490, 2016/01/01/ 2016, doi: https://doi.org/10.1016/j.procs.2016.02.092.

[5] P. Sethi and V. Kapoor, "A Proposed Novel Architecture for Information Hiding in Image Steganography by Using Genetic Algorithm and Cryptography," *Procedia Computer Science*, vol. 87, pp. 61-66, 2016/01/01/ 2016, doi: https://doi.org/10.1016/j.procs.2016.05.127.

[6] A. Solichin and Painem, "Motion-based less significant frame for improving LSB-based video steganography," *in 2016 International Seminar on Application for Technology of Information and Communication (ISemantic)*, 5-6 Aug. 2016 2016, pp. 179-183, doi: 10.1109/ISEMANTIC.2016.7873834.

[7] S. Mumthas and A. Lijiya, "Transform Domain Video Steganography Using RSA, Random DNA Encryption and Huffman Encoding," *Procedia Computer Science*, vol. 115, pp. 660-666, 2017/01/01/ 2017, doi: https://doi.org/10.1016/j.procs.2017.09.152.

[8] R. J. Mstafa, K. M. Elleithy, and E. Abdelfattah, "A Robust and Secure Video Steganography Method in DWT-DCT Domains Based on Multiple Object Tracking and ECC," *IEEE Access*, vol. 5,

pp. 5354-5365, 2017, doi: 10.1109/ACCESS.2017.2691581.

[9] P. Kumar and K. Singh, "An improved data-hiding approach using skin-tone detection for video steganography," *Multimedia Tools and Applications*, vol. 77, no. 18, pp. 24247-24268, 2018/09/01 2018, doi: 10.1007/s11042-018-5709-y.

[10] M. Dalal and M. Juneja, "A robust and imperceptible steganography technique for SD and HD videos," *Multimedia Tools and Applications*, vol. 78, no. 5, pp. 5769-5789, 2019/03/01 2019, doi: 10.1007/s11042-018-6093-3.

[11] S. Kumar and R. Soundrapandiyan, "Robust approach of video steganography using combined keypoints detection algorithm against geometrical and signal processing attacks," *Journal of Electronic Imaging*, vol. 29, no. 4, pp. 043007-043007, 2020, doi: https://doi.org/10.1117/1.JEI.29.4.043007.

[12] M. Dalal and M. Juneja, "A secure and robust video steganography scheme for covert communication in H.264/AVC," *Multimedia Tools and Applications*, vol. 80, no. 9, pp. 14383-14407, 2021/04/01 2021, doi: 10.1007/s11042-020-10364-z.

[13] R. Roselinkiruba, T. S. Sharmila, and J. J. Julina, "A novel pattern-based reversible data hiding technique for video steganography," *Research Square 2022*, doi: https://doi.org/10.21203/rs.3.rs-1619375/v1.

[14] M. Dalal and M. Juneja, "A secure video steganography scheme using DWT based on object tracking," Information Security Journal: *A Global Perspective*, vol. 31, no. 2, pp. 196-213, 2022/03/04 2022, doi: 10.1080/19393555.2021.1896055.

[15] S. N. Gowda and S. Sulakhe, "Block based least significant bit algorithm for image steganography," *in Proceedings of the Annual International Conference on Intelligent Computing, Computer Science & Information Systems*, Pattaya, 2016, pp. 16-19.

[16] V. Holub, "Content Adaptive Steganography: Design and Detection," *State University of New York at Binghamton, Thomas J. Watson School of ..., 2014*. [Online]. Available: http://dde.binghamton.edu/vholub/pdf/Holub_PhD_Dissertation_2014.pdf

[17] A. Sharif, M. Mollaeefar, and M. Nazari, "A novel method for digital image steganography based on a new three-dimensional chaotic map," *Multimedia Tools and Applications*, vol. 76, no. 6,

pp. 7849-7867, 2017/03/01 2017, doi: 10.1007/s11042-016-3398-y.

[18] J. Watkins, "Steganography-Messages Hidden in Bits," *Multimedia Systems Coursework, Dept of Electronics and CS, University of Southampton*, SO17 1BJ, UK, 2001.

[19] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "Digital image steganography: Survey and analysis of current methods," *Signal Processing*, vol. 90, no. 3, pp. 727-752, 2010/03/01/ 2010, doi: https://doi.org/10.1016/j.sigpro.2009.08.010.

[20] Y. JinaChanu, K. Singh, and T. Tuithung, "Image Steganography and Steganalysis: A Survey," *International Journal of Computer Applications*, vol. 52, pp. 1-11, 08/01 2012, doi: 10.5120/8171-1484.

[21] W. C. Easttom II, Computer security fundamentals, 5 ed. Pearson IT Certification, 2016.

[22] W. M. Saqer, "Steganography within LSB and Second LSB with Randomness Depending on Indicators Using Secret Key," *Master of Science, Environmental Sciences, Islamic University, Palestine (Gaza Strip)*, 2017. [Online]. Available: https://search.emarefa.net/detail/BIM-905920

[23] G. Kipper, Investigator's guide to steganography, 1st ed. Auerbach Publications, 2003.

[24] J. Lubacz, W. Mazurczyk, and K. Szczypiorski, "Principles and overview of network steganography," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 225-229, 2014, doi: 10.1109/MCOM.2014.6815916.

[25] A. Al-Mohammad, "Steganography-based secret and reliable communications: Improving steganographic capacity and imperceptibility," *Brunel University, School of Information Systems, Computing and Mathematics* …, 2010. [Online]. Available: http://bura.brunel.ac.uk/handle/2438/4634

[26] W. Abu-Marie, A. A.-A. Gutub, and H. Abu-Mansour, "Image Based Steganography Using Truth Table Based and Determinate Array on RGB Indicator," *International Journal of Signal & Image Processing*, vol. 1, no. 3, pp. 196-204, 2010. [Online]. Available: https://drive.uqu.edu.sa/_/aagutub/files/_/publications/J_p_paper.pdf.

[27] R. Das and T. Tuithung, "A novel steganography method for image based on Huffman Encoding," *in 2012 3rd National Conference on Emerging Trends and Applications in Computer Science*, 30-31 March 2012 2012, pp. 14-18, doi: 10.1109/NCETACS.2012.6203290.

[28] Kamran, A. Khan, and S. A. Malik, "A high capacity reversible watermarking approach for authenticating images: Exploiting down-sampling, histogram processing, and block selection," *Information Sciences*, vol. 256, pp. 162-183, 2014/01/20/ 2014, doi: https://doi.org/10.1016/j.ins.2013.07.035.

[29] D. Rosiyadi, S. J. Horng, P. Fan, X. Wang, M. K. Khan, and Y. Pan, "Copyright Protection for E-Government Document Images," *IEEE MultiMedia*, vol. 19, no. 3, pp. 62-73, 2012, doi: 10.1109/MMUL.2011.41.

[30] W.-H. Lin et al., "Image copyright protection with forward error correction," *Expert Systems with Applications*, vol. 36, no. 9, pp. 11888-11894, 2009/11/01/ 2009, doi: https://doi.org/10.1016/j.eswa.2009.04.026.

[31] S.-J. Horng, D. Rosiyadi, T. Li, T. Takao, M. Guo, and M. K. Khan, "A blind image copyright protection scheme for e-government," *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 1099-1105, 2013/10/01/ 2013, doi: https://doi.org/10.1016/j.jvcir.2013.07.008.

[32] S.-J. Horng, D. Rosiyadi, P. Fan, X. Wang, and M. K. Khan, "An adaptive watermarking scheme for e-government document images," *Multimedia Tools and Applications*, vol. 72, no. 3, pp. 3085-3103, 2014/10/01 2014, doi: 10.1007/s11042-013-1579-5.

[33] D. Rosiyadi, S.-J. Horng, N. Suryana, and N. Masthurah, "A Comparison between the Hybrid Using Genetic Algorithm and the Pure Hybrid Watermarking Scheme," *International Journal of Computer Theory and Engineering*, pp. 329-331, 01/01 2012, doi: 10.7763/IJCTE.2012.V4.476.

[34] W. H. Lin, S. J. Horng, T. W. Kao, P. Fan, C. L. Lee, and Y. Pan, "An Efficient Watermarking Method Based on Significant Difference of Wavelet Coefficient Quantization," *IEEE Transactions on Multimedia*, vol. 10, no. 5, pp. 746-757, 2008, doi: 10.1109/TMM.2008.922795.

[35] W.-H. Lin, Y.-R. Wang, S.-J. Horng, T.-W. Kao, and Y. Pan, "A

blind watermarking method using maximum wavelet coefficient quantization," *Expert Systems with Applications*, vol. 36, no. 9, pp. 11509-11516, 2009/11/01/ 2009, doi: https://doi.org/10.1016/j.eswa.2009.03.060.

[36] W.-H. Lin, Y.-R. Wang, and S.-J. Horng, "A wavelet-tree-based watermarking method using distance vector of binary cluster," *Expert Systems with Applications*, vol. 36, no. 6, pp. 9869-9878, 2009/08/01/ 2009, doi: https://doi.org/10.1016/j.eswa.2009.02.036.

[37] A. Khan et al., "Intelligent reversible watermarking and authentication: Hiding depth map information for 3D cameras," *Information Sciences*, vol. 216, pp. 155-175, 2012/12/20/ 2012, doi: https://doi.org/10.1016/j.ins.2012.06.014.

[38] H.-C. Huang, S.-C. Chu, J.-S. Pan, C.-Y. Huang, and B.-Y. Liao, "Tabu search based multi-watermarks embedding algorithm with multiple description coding," *Information Sciences*, vol. 181, no. 16, pp. 3379-3396, 2011/08/15/ 2011, doi: https://doi.org/10.1016/j.ins.2011.04.007.

[39] M. Arsalan, S. A. Malik, and A. Khan, "Intelligent reversible watermarking in integer wavelet domain for medical images," *Journal of Systems and Software*, vol. 85, no. 4, pp. 883-894, 2012/04/01/ 2012, doi: https://doi.org/10.1016/j.jss.2011.11.005.

[40] R. Mstafa and K. Elleithy, Efficient and Robust Video Steganography Algorithms for Secure Data Communication. 2017.

[41] P. Mathur and S. Adhikari, DATA HIDING IN DIGITAL IMAGES USING STAGNOGRAPHY PARADIGM: STATE OF THE ART. *Indian Institute of Technology–Kharagpur*, 2017.

[42] R. ak, A. Zaidan, B. Bahaa, and H. Alanazi, "Overview: Main Fundamentals for Steganography," *JOURNAL OF COMPUTING*, vol. 2, no. 3, pp. 158-165, 03/22 2010, doi: https://doi.org/10.48550/arXiv.1003.4086.

[43] T. Morkel, J. Eloff, and M. Olivier, An overview of image steganography. *Information Security for South Africa*, 2005, pp. 1-11.

[44] C.P.Sumathi, T.Santanam, and G.Umamaheswari, "A study of various steganographic techniques used for information hiding," *International Journal of Computer Science & Engineering Survey (IJCSES)*, vol. 4, no. 6, 2014, doi: 10.5121/ijcses.2013.4602.

[45] M. Saidi, H. Hermassi, R. Rhouma, and S. Belghith, "A new adaptive image steganography scheme based on DCT and chaotic map," *Multimedia Tools and Applications*, vol. 76, no. 11, pp. 13493-13510, 2017/06/01 2017, doi: 10.1007/s11042-016-3722-6.

[46] H. Wang and S. Wang, "Cyber warfare: steganography vs. steganalysis," *Commun. ACM*, vol. 47, no. 10, pp. 76–82, 2004, doi: 10.1145/1022594.1022597.

[47] R. Chandramouli, M. Kharrazi, and N. Memon, "Image Steganography and Steganalysis: Concepts and Practice," *in Digital Watermarking, Berlin, Heidelberg, T. Kalker, I. Cox, and Y. M. Ro*, Eds., 2004// 2004: *Springer Berlin Heidelberg*, pp. 35-49.

[48] E. Cole and R. D. Krutz, Hiding in Plain Sight: Steganography and the Art of Covert Communication. *John Wiley & Sons, Inc.*, 2003.

[49] M. Juneja and P. Sandhu, "Data Hiding with Enhanced LSB Steganography and Cryptography for RGB Color Images," *Indian Journal of Applied Research*, vol. 3, pp. 118-120, 10/01 2011, doi: 10.15373/2249555X/MAY2013/35.

[50] N. F. Johnson and S. Jajodia, "Exploring steganography: Seeing the unseen," *Computer*, vol. 31, no. 2, pp. 26-34, 1998, doi: 10.1109/MC.1998.4655281.

[51] A. D. Ker, "Steganalysis of Embedding in Two Least-Significant Bits," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 1, pp. 46-54, 2007, doi: 10.1109/TIFS.2006.890519.

[52] A. Podder, P. Roy, and S. Roy, "Steganography Techniques - An Overview," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp. 323-327, 11/20 2022, doi: 10.32628/CSEIT228642.

[53] P. Goel, "Data hiding in digital images: a Steganographic paradigm," *Master, Master's thesis in Computer Science and Engineering, Indian Institute of Technology Kharagpur*, 2008. [Online]. Available: http://cse.iitkgp.ac.in/~abhij/facad/03UG/Report/03CS3003_Piyush_Goel.pdf

[54] R. Jain and J. Boaddh, "Advances in digital image steganography," *in 2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH)*, 3-5 Feb. 2016 2016, pp. 163-171, doi: 10.1109/ICICCS.2016.7542298.

[55] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek, "A brief introduction to OpenCV," *in 2012 Proceedings of the 35th International Convention MIPRO*, 21-25 May 2012 2012, pp. 1725-1730.

[56] S. Gollapudi, Learn computer vision using OpenCV, 1st ed. Apress Berkeley, CA, 2019, p. 151.

[57] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object Detection in 20 Years: A Survey," *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257-276, 2023, doi: 10.1109/JPROC.2023.3238524.

[58] L. Du, R. Zhang, and X. Wang, "Overview of two-stage object detection algorithms," *Journal of Physics: Conference Series*, vol. 1544, no. 1, p. 012033, 2020/05/01 2020, doi: 10.1088/1742-6596/1544/1/012033.

[59] P. Rajeshwari, P. Abhishek, P. Srikanth, and T. Vinod, "Object detection: an overview," *International Journal of Trend in Scientific Research and Development (ijtsrd)*, vol. 3, no. 3, pp. 1663-1665, 2019. [Online]. Available: https://www.ijtsrd.com/papers/ijtsrd23422.pdf.

[60] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A Review of Yolo Algorithm Developments," *Procedia Computer Science*, vol. 199, pp. 1066-1073, 2022/01/01/ 2022, doi: https://doi.org/10.1016/j.procs.2022.01.135.

[61] K.-H. Jung and Y. Kee-Young, "Improved Exploiting Modification Direction Method by Modulus Operation," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 2, 03/01 2009.

[62] M. Backes and C. Cachin, "Public-Key Steganography with Active Attacks," *in Theory of Cryptography, Berlin, Heidelberg, J. Kilian, Ed.*, 2005: Springer Berlin Heidelberg, pp. 210-226.

[63] A. Horé and D. Ziou, "Image Quality Metrics: PSNR vs. SSIM," *in 2010 20th International Conference on Pattern Recognition*, 23-26 Aug. 2010 2010, pp. 2366-2369, doi: 10.1109/ICPR.2010.579.

[64] L. Gong, K. Qiu, C. Deng, and N. Zhou, "An optical image compression and encryption scheme based on compressive sensing and RSA algorithm," *Optics and Lasers in Engineering*, vol. 121, pp. 169-180, 2019/10/01/ 2019, doi: https://doi.org/10.1016/j.optlaseng.2019.03.006.

[65] N. Akhtar, "An LSB Substitution with Bit Inversion Steganography Method," *in Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics, New Delhi, A. Nagar, D. P. Mohapatra, and N. Chaki, Eds.*, 2016: Springer India, pp. 515-521.

[66] D. Neeta, K. Snehal, and D. Jacobs, "Implementation of LSB Steganography and Its Evaluation for Various Bits," *in 2006 1st International Conference on Digital Information Management*, 6-6 Dec. 2006 2007, pp. 173-178, doi: 10.1109/ICDIM.2007.369349.

[67] K. Thangadurai and G. S. Devi, "An analysis of LSB based image steganography techniques," *in 2014 International Conference on Computer Communication and Informatics*, 3-5 Jan. 2014 2014, pp. 1-4, doi: 10.1109/ICCCI.2014.6921751.

الخلاصة

إخفاء المعلومات في فيديو هو عملية إخفاء البيانات مثل النص أو الصور أو مقاطع الفيديو داخل فيديو حاوي. هناك طرق مختلفة لإخفاء الرسائل في مقاطع الفيديو مع جعل التغييرات في الفيديو لا يمكن اكتشافها بالعين البشرية. ومع ذلك، ليس هنالك طرق فعالة عديدة كافية لإنتاج مقاطع فيديو مطابقة لمقاطع الفيديو التي لم يتم تعديلها أثناء استخراج الرسالة السرية المضمنة دون أي فقدان للبيانات.

في هذه الأطروحة، يقترح المؤلف تقنية إخفاء معلومات تستغل كائنات معينة، حيث يتم اكتشاف هذه الكائنات وتتبعها خلال الفيديو المحدد، وأيضا تستغل طوبولوجيا البكسل مع عتبة تضمين محددة مسبقًا. يتم تحديد الكائن بناء على مساحته وعدد مرات ظهوره ثم يتم تضمين الرسالة السرية باستخدام تقنية البت الأقل أهمية (LSB) استنادًا إلى بعض طوبولوجيا البكسلات والعتبة.

الطريقة المستخدمة في تضمين واستخراج الرسالة السرية هي طريقة جديدة وستتم مناقشتها بالتفصيل في هذه الأطروحة وستكون الرسالة السرية المستهدفة من نوع صورة فقط.

كمقياس للجودة تم اعتماد ذروة الإشارة إلى نسبة الضوضاء (PSNR) لقياس الاختلاف في الفيديو الحاوي قبل وبعد تضمين الرسالة السرية. التجارب العملية لذروة الإشارة إلى نسبة الضوضاء تظهر نتائج مثيرة للإعجاب حيث لا يمكن تمييز الفيديو الحاوي للمعلومات السرية من حيث الصورة المرئية عن الفيديو الأصلي، والصورة السرية المستخرجة من الفيديو الحاوي للمعلومات السرية هي نسخة متطابقة من الصورة السرية الأصلية المضمنة مع 100 ديسيبل (PSNR).

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعـــة بابـــل
كلية تكنولوجيا المعلومات
قسم البرمجيات

إخفاء المعلومات في الفيديو بناء على عدد مرات ظهور كائن و مساحته

رسالة مقدمة الى
مجلس كلية تكنولوجيا المعلومات – جامعة بابل كجزء من متطلبات نيل
درجة الماجستير في تكنولوجيا المعلومات / البرمجيات

من قبل

قصي منير دياب محمد

بإشراف

أ.د. توفيق عبد الخالق الأسدي

١٤٤٥هـ ٢٠٢٣م

جامعة بابل
كلية تكنولوجيا المعلومات
قسم البرمجيات

محضر لجنة استشهاد بحث علمي

استنادا للصلاحيات المخولة لنا حسب الامر الاداري المرقم (4064) في 25/ 9/ 2022 ، تم الاطلاع وتدقيق البحث المقدم من قبل طالب الدراسات العليا **قصي منير دياب محمد الدره** (☑ ماجستير ، □ دكتوراه) وكما في ادناه:

ت	اسم البحث	اسم المجلة	حالة المجلة ضمن او خارج مستوعبات (كلاريفيت ، سكوباس)	Impact Factore or CiteScore	حالة البحث مقبول للنشر/ منشور
1	A survey on video steganography Based on the techniques and evaluation metrics	the second international conference on advanced computer applications (ACA 2023)	سكوباس في حالة النشر وظهوره في بروفايل الباحث	البحث مقبول للنشر في مؤتمر IEEE وبانتظار Cite Score	مقبول للنشر
2	Image Steganography Based on Pixel Topology and Threshold on Selected Pixels Differences	6th International Iraqi Conference on Engineering Technology and its Applications – 2023 (6th-IICETA 2023)	سكوباس في حالة النشر وظهوره في بروفايل الباحث	البحث مقبول للنشر في مؤتمر IEEE وبانتظار Cite Score	مقبول للنشر

قرار اللجنة:

ت	اسم البحث	القرار (مقبول)	القرار (مرفوض)
1	A survey on video steganography Based on the techniques and evaluation metrics	**مقبول**	□ مجلة كلاريفيت (لا تحتوي على معامل تأثير). □ مجلة سكوبس (لا تحتوي على سايت سكور). □ المجلة (مختطفة، مفترسة). □ دار النشر للمجلة (مفترس).
2	Image Steganography Based on Pixel Topology and Threshold on Selected Pixels Differences	**مقبول**	□ مجلة كلاريفيت (لا تحتوي على معامل تأثير). □ مجلة سكوبس (لا تحتوي على سايت سكور). □ المجلة (مختطفة، مفترسة). □ دار النشر للمجلة (مفترس).

م . حوراء شريف	**م. د. سرى جاسم**	**م.د. مازن كاظم**	**أ.م.د . نشوان جاسم**	**أ.د .أحمد سليم**
اسم وتوقيع عضو اللجنة	**اسم وتوقيع عضو اللجنة**	**اسم وتوقيع عضو اللجنة**	**اسم وتوقيع عضو اللجنة**	**اسم وتوقيع رئيس اللجنة**

مصادقة رئيس القسم/