LSB-BASED ROBUST IMAGE WATERMARKING USING TURTLE-SCHEME ON NEIGHBORHOOD PIXEL

A MASTER'S THESIS

SUBMITTED TO
THE SCHOOL OF ELECTRICAL ENGINEERING

By

LAILATUN ADZIMAH

2101222075

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF ENGINEERING

TELKOM UNIVERSITY
BANDUNG
2024

APPROVAL PAGE

A MASTER'S THESIS

THESIS TITLE

LSB-BASED ROBUST IMAGE WATERMARKING USING TURTLE-SCHEME ON NEIGHBORHOOD PIXEL

by

LAILATUN ADZIMAH
2101222075

Approved and authorized to fulfill one of the requirements of
Program of Master of Electrical-Telecommunication Engineering
School of Electrical Engineering
Telkom University
Bandung

Bandung, September 16, 2024
Menyetujui,

Supervisor	Co-Supervisor
Dr. Gelar Budiman, S.T., M.T.	**Sofia Saidah, S.T., M.T.**
NIP. 08780030	NIP. 20890021

SELF DECLARATION AGAINST PLAGIARISM

In this thesis report entitle:

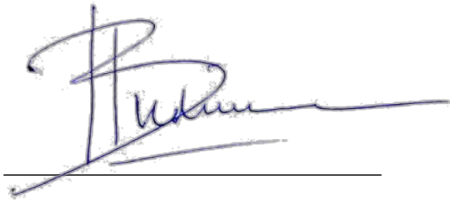
LSB-Based Robust Image Watermarking Using Turtle-Scheme on Neighborhood Pixel

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Bandung, September 16, 2024
Lailatun Adzimah

Signature: _____

Bandung, September 16, 2024
Supervisor: Dr. Gelar Budiman, S.T., M.T



Signature: _____

Bandung, September 16, 2024
Co-Supervisor: Sofia Saidah, S.T., M.T

Signature: _____

ABSTRACT

In the digital era, multimedia plays a crucial role in disseminating information across the globe. Multimedia content such as audio, video, and images can be easily accessed and shared, but they are also at high risk of misuse and copyright infringement. Image watermarking is a method used to protect multimedia content by embedding secret information into a carrier image, thereby safeguarding copyright. Although digital image watermarking is an effective method, it also faces challenges, particularly from attacks aimed at damaging or removing the watermark. Therefore, a robust watermarking algorithm scheme is required to protect multimedia content from attacks. This research proposes a modified turtle shell based watermarking scheme to improve the insertion capacity and robustness against attacks. In this scheme, Least Significant Bit (LSB) technique is used to embed the watermark image and modified turtle shell technique for watermark coordinate mapping. We evaluate imperceptibility using Peak Signal to Noise Ratio (PSNR) and Mean Opinion Score (MOS). The average PSNR result obtained in the proposed method is 51.24 dB and the MOS value is 4.48. We also evaluate the robustness of the watermarked image against attacks such as Gaussian Noise, Salt and Pepper, Compression, Low Pass Filter, Rescaling, Speckle Noise, and Median Filter using BER parameters. The BER before the attack is zero, and after the attack results in varying BER values. The results show that the scheme is robust against Gaussian Noise (BER: 0.08), Salt and Pepper (BER: 0.003), and Speckle Noise (BER: 0.003) attacks, but less robust against Compression (BER: 0.13), Low Pass Filter (BER: 0.45), Rescaling (BER: 0.43), and Median Filter (BER: 0.45) attacks.

Keywords: Image Watermarking, LSB, Robustness, Turtle Shell .

ACKNOWLEDGMENTS

Bismillaahirrahmaanirrahiim. Alhamdulillahirabbil'alamin.
Arrahmaanirrahiim. Maalikiyawmiddiin.

In the name of Allah, the Most Gracious, the Most Merciful. All praise is due to Allah, the Lord of the Worlds, the Most Compassionate, the Most Merciful. The Master of the Day of Judgment. O Allah, thank you for Your Grace and Bless, so the author was able to complete the study. Without Your Kindness, the author will not be able to finish this thesis and achieve the Master's degree.

The journey to earn a Master's degree ended up with the completion of this thesis. The author would never reach this point without the help, support, efforts, and prayers from family, relatives, friends, and lecturers. Therefore, the author would like to express the deepest gratitude and thanks to:

1. **To my beloved parents**, who have always accompanied me with prayers, unwavering love, and endless support, both morally and materially, throughout this journey. Thank you for always giving your best without ever demanding anything, even in the most difficult times. You are my source of strength and inspiration, constantly teaching me patience and sincerity without expecting anything in return. May Allah always bless you with happiness, health, and His grace.

2. **Dr. Gelar Budiman, S.T., M.T.,** as my supervisor, I would like to express my deepest gratitude for all the guidance, advice, and invaluable support throughout the process of completing this thesis, as well as for the new knowledge you have generously shared, enabling the completion of this thesis. I feel incredibly fortunate and honored to have been under your supervision, where you patiently and with great dedication devoted your time and valuable knowledge. Thank you for the trust and attention you have shown, allowing me to successfully complete this thesis. May all your kindness, dedication, and sincerity be recorded as continuous good deeds, and may Allah always bless you with health, prosperity, and a long life.

3. **Sofia Saidah, S.T., M.T.,** as my Co-Supervisor, I would like to extend my deepest gratitude for all the guidance, understanding, and invaluable support throughout the completion of this thesis. I am truly grateful for the attention,

trust, and patience you have shown in guiding me through this process. Your support has not only contributed to the development of this thesis but has also played a significant role in building my motivation and spirit. May all your kindness and dedication be recorded as multiplied good deeds by Allah. I feel incredibly fortunate to have crossed paths with you on this academic journey.

4. **To all the lecturers of the Master of Electrical Engineering program**, I extend my deepest gratitude for the knowledge, insights, perspectives, and invaluable experiences that have been shared throughout my studies in the Master of Electrical Engineering program. Every lesson and guidance provided has become a significant asset, not only in the academic realm but also in personal development. I feel incredibly fortunate to have learned from such remarkable lecturers, and this experience will always be cherished as I move forward in my career journey.

5. **Dr. Ida Wahidah, S.T., M.T.**, as the Head of the Master of Electrical Engineering Study Program, I extend my gratitude for your wisdom, policies, and efforts to continuously improve the quality of the program. I would also like to express my thanks to Mr. Rudie, who previously served as the administrative staff, as well as Mrs. Giashinta and Mr. Insan, who are currently serving as administrative staff for the Master of Electrical Engineering Study Program, for their assistance and support in academic administration. Your support has been invaluable in ensuring the smooth progress of my studies.

6. **Nurul Izzah Luthfiah Nur, S.T.,** who has always given unwavering support, always present through every joy and sorrow, and has been a loyal listener. Thank you for all the valuable advice you've provided in helping to find solutions, and for your extraordinary patience in dealing with me. I feel incredibly fortunate and grateful to have known you. Your presence has added color and meaning to this academic journey, making every step feel lighter and full of enthusiasm.

7. **May Refiyanti, S.T.,** who has provided support and been a companion in completing this thesis. Thank you for the companionship, guidance, and your time that was always available. I am truly grateful to have a friend who fought alongside me, making this process feel lighter and ensuring that I never felt alone. Your support and encouragement have been an important part of finishing this thesis.

8. Friends who have accompanied my days, provided entertainment, and spent

time playing together during my Master's studies: **Daffa, Dimas, Faqih, Luthfi, Amir, Fasha, Timo, Muti,** and **Trischa**. It has been a pleasure to know and learn alongside all of you. Our time together has not only been a welcome escape from the busyness but also one of the most cherished memories of my academic journey. Thank you for making this experience more colorful and memorable.

9. Friends who have always provided encouragement, support, and patiently listened to all my concerns: **Naqliya, Asha, Dara**, and **Alya**. Thank you for all the moral support you have given me. I am truly grateful and feel fortunate to have you as friends. Your presence not only helped me get through tough times but also gave me more confidence in facing every challenge throughout this process.

10. My deepest gratitude goes to **Lab INACOS**, which has been a shelter and a place to work throughout my studies. In this space, I not only completed my coursework and thesis but also experienced warm companionship with friends, sharing laughter and daily moments together. The sweet memories created there will forever be etched in my mind as a cherished part of this journey.

11. I would also like to extend my heartfelt thanks to **Lab AICOMS**, which has been a place to learn and deepen my knowledge of watermarking. Thank you for the guidance, support, and the opportunity to further explore this field during the thesis process. All the knowledge I have gained has been invaluable in helping me achieve my academic goals and broaden my horizons.

Perhaps that all is what the author can say. I do appologize if there is a misspelled for name and titles. For the parties who missed from the above gratitude, the author apologizes profusely for the negligence. May Allah the All-Knowing, whom nothing escapes His control, record your kindness and reward it with the best possible reward.

PREFACE

All praise is due to Allah SWT for His grace and blessings, enabling the author to complete this thesis titled **"LSB-Based Robust Image Watermarking Using Turtle-Scheme on Neighborhood Pixel."** May peace and blessings be upon the Prophet Muhammad SAW, his family, companions, and all his followers. Aamiin.

The rapid development of digital technology today demands robust *watermarking* methods to protect image data against various types of attacks. This thesis proposes an *LSB (Least Significant Bit)-based image watermarking* method with modifications to the *Turtle Shell Scheme*. This approach aims to enhance the resilience of watermarks against common attacks, such as noise and image compression.

The author hopes that this research can contribute to the development of stronger and more efficient watermarking techniques. However, the author acknowledges that this research is still at an early stage, leaving ample opportunity for further exploration and enhancement by future researchers.

The author is also aware that there may be unintended errors or shortcomings in this thesis. Therefore, constructive feedback and suggestions from readers are highly appreciated to improve the quality of this work.

In conclusion, it is the author's hope that this thesis will provide benefits to those interested in deepening their knowledge of *watermarking* techniques and the *Turtle Shell Scheme* modifications. It is also hoped that this research will serve as a useful reference for the advancement of knowledge in this field.

Bandung, September 16, 2024

Lailatun Adzimah

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

LSB	: Least Significant Bit
RGB	: Red, Green, Blue
PSNR	: Peak Signal to Noise Ratio
MOS	: Mean Opinion Score
BER	: Bit Error Rate
LPF	: Low Pass Filter

CHAPTER I
INTRODUCTION

This chapter provides a brief overview of the research. Consist of six sections; the explanation starts with background, problem identification and objective, scope of work, research methodology, and structure of this thesis. A more detailed explanation will be later in the next chapter.

1.1 Background

In digital era, multimedia plays an important role in disseminating information around the world. Multimedia content such as audio, video, and images can be easily accessed and shared with anyone around the world. However, behind this convenience lies a great risk of information misuse and copyright infringement. Manipulation, unauthorized distribution, and illegal copying are major challenges for multimedia content owners. Digital image tagging is an effective solution to protect the authenticity and copyright of information contained in multimedia content [1].

Generally, watermarking has two area classifications, it is spectrum domain and space domain. Spectrum domain is a domain where the watermark embedding process involves manipulating the coefficients of the original image. While space domain is processing watermark hiding which is carried out directly into the picture or other media in the form of image pixels. This hiding process uses low computational complexity, but the process cannot withstand digital signal processing .

Image watermarking is a method to protect multimedia content with secret information embedded into carrier images. The inserted secret information is called a watermark or label. Watermarking is used to protect copyright, so watermarking must be embedded and extracted by the owner with ease, thus need Exactly embedding process [2]. Along with the increasing uses of digital images, then the research area in watermarking is more extensive. Data authentication and copyright protection is an important application scope in the use of image watermarking. These applications include ownership identification, broadcast monitoring, usage control, forgery detection and authentication, copy control, medical applications, and copyright protection [3].

Although digital image watermarking is an effective method to protect multi-

media content, it also faces challenges, especially from attacks that aim to damage or remove the watermark. Geometric distortion, compression, low pass filter, and Gaussian noise are some type of attack at image watermarking [4]. Therefore, to protect multimedia content from attacks, a robust watermarking algorithm scheme is required. In [5], Chang et al. first proposed a turtle shell-based information hiding scheme. Then, in [6], Liu et al. redeveloped the research of Chang et al. [5] to improve the insertion capacity and image quality using the turtle shell scheme.

Over the past five years, research on turtle shell schemes has continued to develop. Lin et al. [7] proposed a real-time dual-image-based reversible data hiding scheme using turtle shells, achieving a PSNR of 49.38 dB, though it lacked testing against image attacks. Lin et al. [8] introduced a fragile watermarking scheme with the turtle-shell technique, improving embedding capacity and reducing distortion, with a PSNR of 46.8 dB, but also lacking attack testing. Chang and Liu [9] proposed two real-time turtle-shell-based data embedding mechanisms to reduce computational complexity and enhance visual quality, achieving a PSNR above 45 dB, yet without testing for attacks. Xieo et al. [10] presented a modified 2D histogram-based turtle shell scheme, expanding the embedding area and achieving a PSNR around 30 dB, but this study also did not include attack testing and reported lower PSNR values. Lastly, Li et al. [11] proposed a scheme for sharing secret images with easy authentication using a turtle shell structure, achieving a PSNR of 47.87 dB. In this study [11], the PSNR value is good and has explained the attack,but the variety of attacks is slight.

In this paper, we propose a secret information hiding scheme based on a turtle shell technique. This research not only examines imperceptibility and capacity but also addresses robustness, unlike previous studies [7] [8] [9] [10] that did not discuss robustness. The inclusion of attack testing is a significant advantage of this paper, demonstrating that the proposed scheme can not only effectively embed information but also protect it against various types of attacks.

This research is organized into five chapters to discuss this research comprehensively. Chapter 1 is the introduction, explaining the background, problem identification, objective, scope of work, and research methodology. Chapter 2 discusses the theoretical foundation, presenting theories related to the turtle shell scheme and data hiding techniques. Chapter 3 outlines the methodology, describing the system model, embedding process, extraction process, attacks, and performance parameters used for evaluation. Chapter 4 presents the results, presenting the experimental findings and analysis. Finally, Chapter 5 contains conclusions and suggestions for future research.

1.2 Problem Identification

In various previous studies [7] [8] [9] [10] [11], turtle shell-based watermarking methods have generally achieved PSNR values above 40 dB, which typically indicates that the watermark is not visible to the human eye. However, there is a study that reported a PSNR value as low as 30 dB [10]. A PSNR value below 40 dB indicates that watermarking may cause significant visual distortion, thereby reducing imperceptibility and potentially compromising the visual quality of the image [12]. The presence of such low PSNR values indicates a gap in the development of methods to improve imperceptibility.

In previous research, several studies have tested the robustness of watermarks against certain attacks, such as Gaussian noise, Salt and Pepper, Speckle noise [11]. However, there are still studies that have not discussed resilience against attacks [7] [8] [9] [10]. The turtle-shell technique itself is still limited in terms of testing variations against attacks, which generally include various types of attacks such as noise, filtering, and compression. Attack resistance is a crucial aspect in assessing the effectiveness of watermarking methods, so further research is needed that includes testing against a more diverse range of attacks to ensure the validity and strength of the proposed scheme. A wider variety of attacks is needed to test the robustness of the watermark more thoroughly, especially on the turtle-shell method, which is still little discussed. This opens up opportunities to develop further tests to improve watermark robustness against various types of attacks.

1.3 Objective

The objectives of this research are as follows:

1. To propose a watermarking scheme using the Turtle-Scheme on Neighborhood Pixel method that aims to enhance image quality, ensuring higher imperceptibility and achieving superior PSNR compared to previous approaches.

2. To evaluate the robustness of the proposed Turtle-Scheme on Neighborhood Pixel watermarking method against a range of attacks, including but not limited to Gaussian noise, Salt and Pepper noise, compression, low-pass filtering, rescaling, speckle noise, and median filter with the potential to include additional attacks as the research progresses.

1.4 Scope of Work

To maintain focus and prevent the experiment from becoming overly extensive, this thesis limits the scope of work as follows:

1. The watermarking method implemented in this research is the Turtle-Scheme on Neighborhood Pixel, a modification of the Turtle shell algorithm.
2. The images used in the experiments are grayscale images to standardize the testing conditions.
3. The watermark used in all experiments is a randomly generated binary image
4. The imperceptibility of the proposed method will be evaluated using the following parameters: Peak Signal to Noise Ratio (PSNR), Payload, and Mean Opinion Score (MOS).
5. The robustness of the proposed method against various attacks will be assessed using the Bit Error Rate (BER) as the primary performance parameter.

1.5 Research Methodology

This thesis is divided into 3 work packages (WP) to produce high quality results.

- WP 1: Study of Literature
  This thesis studies the basic concepts and theories related to Digital Image, Watermarking, Binary Representation, Least Significant Bit (LSB),and Turtle shell algorithm.
- WP 2: System design and simulation
  Make a system design for the image watermarking process, embedding process, and perform extraction using a modified turtle scheme. After that, simulate based on the previously designed embedding to extraction system model into the MATLAB program.
- WP 3: Testing and Analysis
  This final project tests the system and analyzes the data obtained from the testing process to determine the performance results produced by the system.

CHAPTER II
BASIC CONCEPTS

This chapter discusses the basic concepts of this thesis. These theories serve as an introduction to the design proposed in this thesis, such as Digital Image, Watermarking, Binary Representation, Turtle Shell-based Data Hiding Method, and Least Significant Bit (LSB).

2.1 Digital Image

The digital image is data overview-based digital which has pixel-pixel. Every pixel has a numeric value that determines the color and the intensity. Digital images have discrete value or infinity. Digital image can be represented as a matrix, with every element matrix, called a pixel. Pixel is the smallest unit of an image with a numeric value that can determine color or brightening at a specific location in the image. Using matrix to represents image enable computer processing that efficient includes the manipulation, analysis, and storage of images in various applications such as digital photography, medical image processing, and graphic design [13]. In digital image processing applications, digital images are divided into 3 namely color image Red, Green, and Blue (RGB), black and white image (grayscale), and binary image. Color images have specific color pixels in the range 0 - 255. Black and white image (grayscale) has color pixels graduating from white to black. The color range in black and white is very suitable for processing image files. Black and white has a pixel range value of 8 bits. Binary images consist of black or white colors only, the pixel value in binary images is only 0 and 1 at each one bit, if in 8 bits then the pixel value is 0 and 255 [14].

The main components of a digital image include resolution, color depth, and file format. Resolution indicates the number of pixels in a digital image; the higher the resolution, the sharper and more detailed the resulting image. Resolution is usually expressed in terms of pixels per inch (PPI) or dots per inch (DPI). Color depth describes the number of bits used to represent the color at each pixel. Examples of color depths include 8-bit, 16-bit, and 24-bit. File formats such as JPEG, PNG, BMP, and TIFF define how digital images are stored and compressed [15].

2.2 Watermarking

Digital watermarking is important aspect in image processing. It is an effective technique for protecting secret information stored in images. Multimedia content such as video, audio, and images can be used in the watermarking process. Watermarking involves the use of a host image and a watermark image, which are then processed through embedding and extraction procedures. The applications of watermarking include copyright protection, owner identification, authentication, and content protection [16].

Digital watermarking is the process of embedding information into digital media such as images, video, or audio, with the aim of protecting copyright, tracking distribution, and verifying authenticity. The basic characteristics of watermarking include robustness, which is the resistance of the watermark to various processing operations and attacks, so that the watermark remains even if the data is modified. Non-perceptibility, where the watermark cannot be seen by the human eye or heard by the human ear, but can only be detected through specialized processing. Verifiability, which allows the watermark to provide reliable proof of ownership and help identify authenticity and monitor data dissemination. And security, where watermarked information has a unique mark that can only be accessed by authorized users to detect, extract or modify it, in order to achieve the purpose of copyright protection [17].

View-based watermarking techniques can be divided into two main categories, namely visible watermarks and invisible watermarks. Visible watermarking is a technique similar to pasting a watermark on an official document or notice paper. These watermarks are clearly visible to the eye and serve as a visual marker or identification. In contrast, invisible watermarks are much more complex and are not visible to the naked eye. This technique is commonly used for identification and security purposes, where information is inserted into digital media without disturbing its visual appearance [18].

2.3 Binary Representation

Binary notation differs from decimal notation which uses base 10, as binary notation is based on the representation of real numbers using base 2. In binary notation, all real numbers can be represented by a sequence of numbers 0 and 1, similar to the way expansion is used in decimal notation. any real number can be

represented as a sequence of binary numbers, as shown below:

$$---1000111010.00101010110111---$$

When data is processed by an algorithm, both the input and output consist of a sequence of symbols that can be related to real numbers, which are then represented in the form of binary numbers 0 and 1. Each number has a binary expansion, similar to the expansion in decimal notation. Binary notation is an alternative way of describing real numbers, using the base 2, which is different from decimal notation which uses the base 10.

Each number has a binary expansion, similar to the number expansion in decimal notation which uses base 10. For example, let's consider the number 203. Its expansion in decimal notation is given by [19]:

$$203 = 200 + 0 + 3 = 2 \times 10^2 + 0 \times 10^1 + 3 \times 10^0$$

Meanwhile, the expansion in binary notation is as follows:

$$\begin{aligned} 203 &= 128 + 64 + 8 + 2 + 1 \\ &= 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \end{aligned} \quad (2.1)$$

Hence, the numeral 203 can be expressed in binary form as follows:

$$\rightarrow 11001011 : \text{Binary representation}$$

2.4 Least Significant Bit (LSB)

The least significant bit (LSB) is the least significant bit of a binary sequence. LSB substitution is a commonly used technique in image steganography, which inserts information into the least significant bits of image pixels and replaces the actual information. The least significant bit (LSB) is the bit that contains the least amount of information for a binary image, which means that LSB changes can be ignored as a whole in the image [20]. The 1-bit secret information replaces the LSB bit value of the carrier pixel. The LSB replacement algorithm is shown in Fig. 2.1. The binary value of 224 is "11100000", and its LSB is "0." The LSB replacement method inserts the secret message "1" into 224, which then changes the value of 224 (11100000) to 225 (11100001). The pixel value where the secret message is

embedded can be extracted directly from the LSB [21].

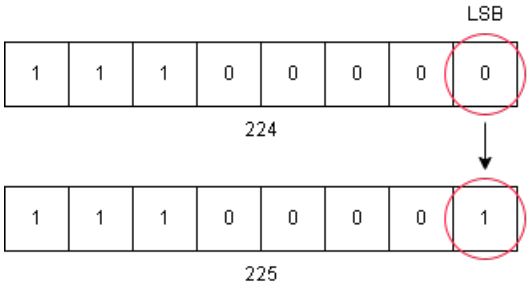


Fig 2.1 An example of LSB replacement method

The Least Significant Bit (LSB) method has advantages in terms of simplicity and ease of implementation. Data can be inserted into the least significant bits without causing noticeable visual changes [22], [23]. This makes the method effective for hiding secret information [22], [24], and it allows for the insertion of a relatively large amount of data depending on the size of the image used as a carrier [24].

However, this method has some weaknesses, such as being vulnerable to manipulations like cropping, resizing, or rotating the image, which can cause the hidden data to be lost [24], [23]. Additionally, it is easy to detect through advanced steganalysis techniques, as changes to the least significant bits can be recognized through forensic analysis [22], [24]. This method is also less resistant to noise-based attacks such as Gaussian and Poisson noise, which can degrade the watermark and reduce the overall quality of the image [23].

2.5 Turtle Shell-based Data Hiding Method

A data hiding method based on ”Turtle shell” was first proposed in 2014 [5], the research was then re-proposed in 2015 [6]. The ”Turtle Shell” method is a hexagon that represents the digits of a secret octal number from 0 to 7 (in binary form, from ’0000’ to ’1111’). Several turtle shells are utilized in the creation of the M reference matrix for embedding data. In this case, the difference between two adjacent elements in the same row of the reference matrix M is increased to ’1’, while the difference between two adjacent elements in the same column is alternately increased to ’2’ for even columns and ’3’ for odd columns.Figure 2.3 is the image of matrix M. In order to increase the concealment capacity, a Location table T is created which contains 16 possible situations of the tortoise shell in the reference matrix M. The sixteen situations in the Location table T can be grouped into four categories, as shown in Fig. 2.2. According to the characteristics of the reference matrix T,

the sets of element values matching Location 1 and Location 4 are always 1,3,5,6 and the elements matching Location 2 and 3 are always 0,2,4,6. Each Location in the Location table T can be represented by $T(S_j, S_j+1)$, where S_j denotes row and S_j+1 denotes column, and S_j and S_j+1 include 00,01,10,11 [6]. The followings are the steps of the turtle shell algorithm [20]:

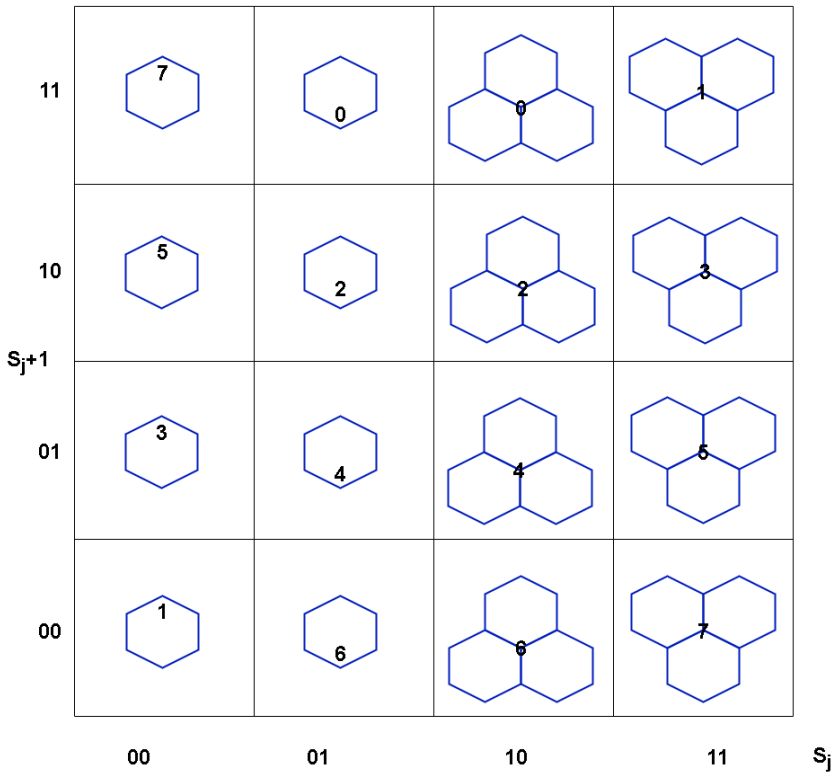


Fig 2.2 Turtle shell-based reference matrix T

1. Split the carrier image into 1×2 image blocks. If the size of the carrier image is $A \times B$, then the gray values of the carrier pixels are $t_1, t_2, \ldots, t_{A \times B}$. Each image block consists of a pair of carrier pixels (t_i, t_{i+1}), with $i \in 1, 3, \ldots, A \times B - 1$. The values t_i and t_{i+1} are used as the X-axis and Y-axis coordinates in matrix M, so that we obtain the point $M(t_i, t_{i+1})$ in matrix M.

2. Determined the set E based on the type of point $M(t_i, t_{i+1})$:

   - Case 1: If $M(t_i, t_{i+1})$ is inside the "Turtle Shell", it is considered an interior point. The set E contains all the points inside the "Turtle shell".
   - Case 2: If $M(t_i, t_{i+1})$ is on an edge of a "Turtle Shell" or an involves an edge number, it is referred to as an vertex point. E is the collection of all point numbers in all "turtle shells" that contain $M(t_i, t_{i+1})$.

- Case 3: If$M(t_i, t_{i+1})$ is outside all "Turtle Shell", it is a boundary point. E is the set of 3×3 matrices containing $M(t_i, t_{i+1})$. (Each 3×3 block in the reference matrix M contains the numbers 0-7.)

3. Step 3: Find the point $M(t_i, t_{i+1})$ corresponding to the 3-bit secret message value in the set E.

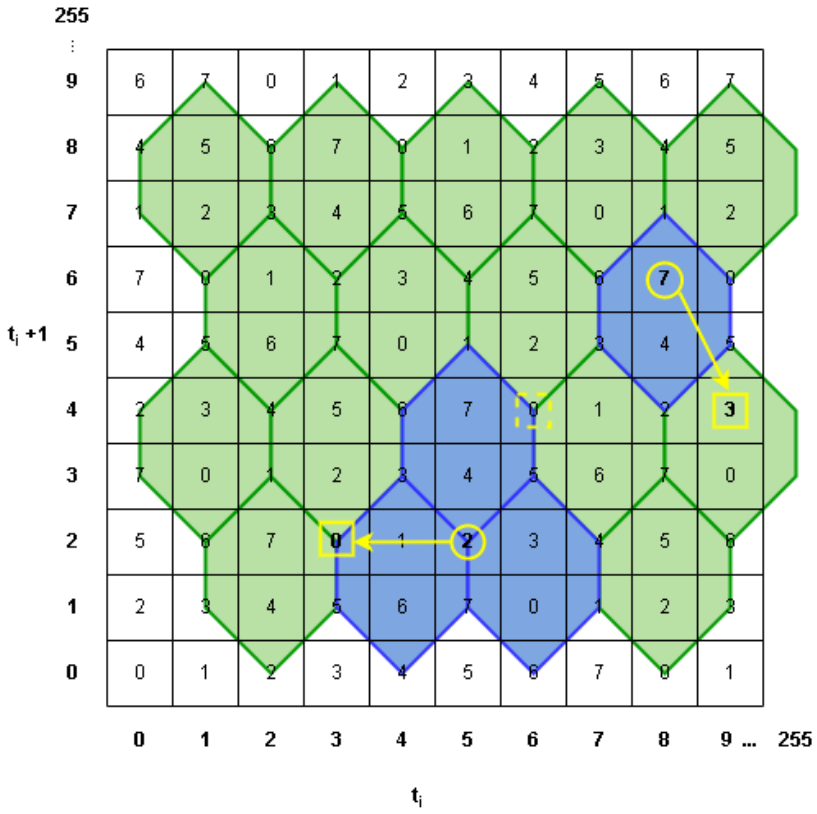


Fig 2.3 Turtle shell-based reference matrix M

The embedding process is demonstrated in Fig. 2.3. Let's consider the original binary secret message as $(10110001)_2$. The corresponding decimal values for the secret digits are 0 and 3. Suppose we have the masking pixel pairs $(5,5) = 2$ and $(8,6) = 7$. The following steps outline how each secret digit is embedded into its respective cover pixel pair and masking pixel pair.

- The secret message is (1011) and the original pixel pair is $(5,2)$. The point $T(5,2)$ lies within the intersection of three turtle shells. The set E includes all points in the three turtle shells containing M, namely $(6,4)$ and $(3,2)$. The closest point to $(5,2)$ is $(3,2)$, so the pixel pair is modified to $(3,2)$ to hide the message.

- If the original pixel pair is $(8,6)$ and the secret information is (0001), the corresponding point $T(8,6)$ is inside the tortoise shell. The set E, which forms a hexagonal-shaped tortoise shell, includes the points $(6,2)$, $(7,8)$, and $(9,4)$. Therefore, the pixel pair $(8,6)$ is replaced with $(9,4)$ to hide the message (0001). In decimal form, the secret information is 7.

The turtle shell method has several significant advantages in terms of embedding capacity, visual quality, resistance to attacks, and low computational complexity. One of its main strengths is its high embedding capacity. In the study [5], the method is introduced with a data hiding capacity of 1 bpp, which is later improved to 1.25 bpp in a follow-up study [25]. Additionally, this method preserves the visual quality of images with embedded data. For example, the study [7] shows that the method can achieve an average image quality of 49.38 dB on the primary shadow and 45.55 dB on the secondary shadow, meaning the images remain high-quality despite the data insertion. In terms of security, the turtle shell scheme also shows good resistance to certain attacks, such as pixel-value differencing (PVD) histogram attacks, which are important for protecting hidden data from detection [25]. Furthermore, this method excels in having low computational complexity, making it suitable for various real-time applications [25].

The turtle shell method has several limitations. One key drawback is its limited reversible capability, especially in earlier versions of the method, as discussed in [5], where the dual-image scheme faced challenges in achieving full reversibility. Additionally, most research on the turtle shell method focuses on grayscale images, as seen in [6], which could present challenges when applied to colored images or other multimedia data. The complexity of implementation is another issue, particularly in dual-image schemes. The process of constructing two shadows for data hiding requires intricate arrangements, especially regarding pixel pair orientation, which adds to the complexity, as described in [7]. Furthermore, while the method shows resilience against certain attacks, it remains vulnerable to noise interference, such as salt-and-pepper noise, which can significantly reduce image quality, especially in high-resolution image applications [25].

2.6 Literature Review

In recent years, various techniques have been proposed to improve data hiding and watermarking schemes, especially those using turtle-shell mechanisms. These techniques focus on aspects such as invisibility, fragility, and attack robustness, aimed at improving the security and quality of the hidden data. However, de-

spite these advancements, there are still gaps in previous tests, especially in terms of robustness against different image attacks. Table 2.1 summarizes the contributions of previous research, describing the methods used, Peak Signal-to-Noise Ratio (PSNR) values, and types of attacks tested.

Table 2.1 Literature Review Previous Research

Author	**Proposed**	**PSNR**	**Attack**	**Lack**
J.-Y. Lin et al. [7]	a real-time dual-image-based reversible data hiding scheme using turtle shells	49.38 dB	no attack	lacking attack testing
C.-C. Lin et al. [8]	introduced a fragile watermarking scheme with the turtle-shell technique	46.8 dB	no attack	lacking attack testing
Chang and Liu [9]	Proposed two real-time turtle-shell-based data embedding mechanisms to reduce computational complexity and enhance visual quality	above 45 dB	no attack	lacking attack testing
Xieo et al. [10]	presented a modified 2D histogram-based turtle scheme	around 30 dB	no attack	did not include attack testing and reported lower PSNR values
Li et al. [11]	proposed a scheme for sharing secret images with easy authentication using a turtle shell structure	47.87 dB	Gaussian Noise Attack, Speckle Noise Attack, Salt and Pepper, and Poison Attack	attacks are still few

Based on Table 2.1, various previous studies have examined watermarking methods using the turtle-shell technique. J.-Y. Lin et al. [7] proposed a real-time dual-image-based reversible data hiding scheme using the turtle-shell technique, which resulted in a PSNR of 49.38 dB. However, this research has not included tests against image attacks, so the security aspects still need to be explored further. C.-C. Lin et al. [8] introduced a fragile watermarking scheme that also utilizes the turtle-shell technique with a PSNR of 46.8 dB, but did not perform attack testing, showing its potential vulnerability. Furthermore, Chang and Liu [9] developed two

real-time turtle-shell-based data embedding mechanisms that aim to reduce computational complexity and improve visual quality, with PSNR above 45 dB. However, this study also did not include tests against attacks, which is a significant drawback. Xieo et al. [10] proposed a modified 2D histogram-based turtle-shell scheme with a PSNR of about 30 dB, but without attack testing and with a relatively low PSNR value, thus the PSNR value can still be improved. Finally, Li et al. [11] developed a scheme for secret image sharing with easy authentication using a turtle-shell structure, which resulted in a PSNR of 47.87 dB and was tested against several types of attacks such as Gaussian Noise, Speckle Noise, Salt and Pepper, and Poison Attack. However, the number of attacks tested is still limited, so further testing is required to ensure the robustness of the scheme.

CHAPTER III
SYSTEM MODEL AND THE PROPOSED DESIGN

This chapter discusses the system model and the construction of the proposed simple design. This research design covers various aspects, from system model, embedding process, extraction process, attacks, and performance parameter.

3.1 Watermarking Process Overview

The system process flow aims to provide a sequential description of the proposed research. Fig. 3.1 is the flowchart used in this study. In this thesis, the image watermarking that resists attacks uses LSB-based techniques with a turtle scheme in a neighborhood. The watermarking process is divided into four stages: the embedding process, the attack process, the extraction process, and performance parameters.

The embedding process involves embedding the watermark into the host image using the LSB modification technique in the neighborhood. The watermarked image is then tested by calculating the PSNR value to assess the similarity between the watermarked image and the original image, as well as the capacity and payload values.

Next, the watermarked image undergoes several attacks through image processing to test the robustness of the method used. The attacks include Gaussian Noise Attack, Salt and Pepper Noise Attack, Compression Attack, Low Pass Filter Attack, Rescaling Attack, Speckle Noise Attack, and Median Filter Attack.

The next process is extraction, which separates the host image from the watermark obtained during the extraction process, followed by measuring the watermark's resistance to attacks by calculating the BER value.

3.1.1 Embedding Process

The embedding process is inserting a watermark in the host image. In this research, the watermark is randomly generated, while the host image is a grayscale image of seven pieces measuring 256 x 256 pixels.

Based on Fig.3.1 below are the steps of watermark embedding using the image watermarking method based on neighborhood based pixel:

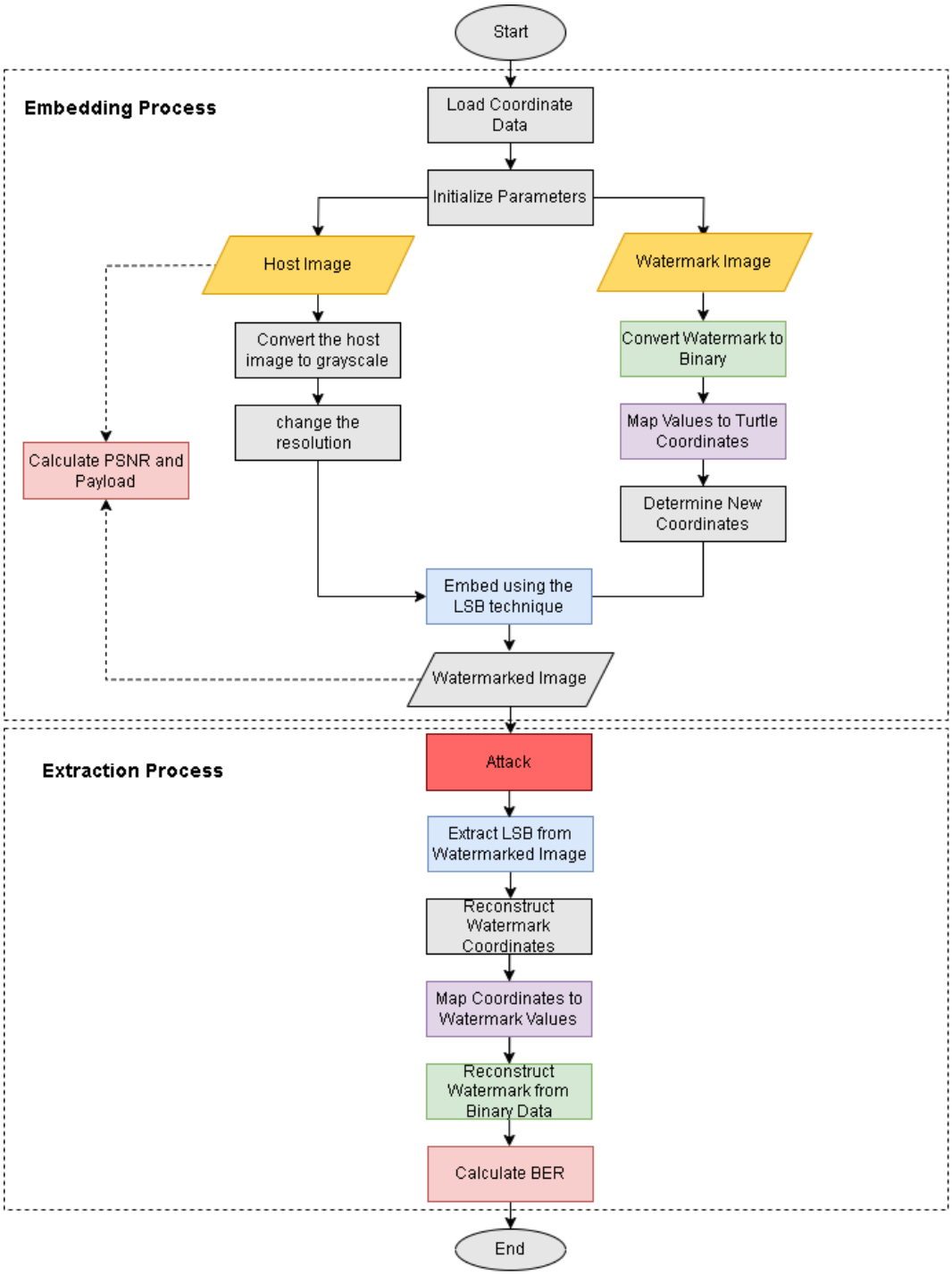


Fig 3.1 Flowchart of the Overall Watermarking Process

1. Read host image and watermark image. Select an RGB-formatted host image with a file format of *.jpg, *.bmp, or *.png and a resolution of 512 x 512 pixels.
2. Convert the host image to grayscale.
3. Change the resolution of the host image to 256 x 256 pixels.
4. Create a watermark image from a binary random generator with a size of 800 to 8000 bits.
5. Retrieve coordinate data from a table containing turtle shell coordinates.
6. Convert the watermark to a decimal value.
7. Map the decimal watermark value to the turtle shell coordinate table.
8. Determine the new coordinates based on the watermark value and the minimum distance from the original coordinates. The original coordinates are the coordinates that were previously mapped, while the new coordinates are the coordinates that have changed due to the position insertion process.
9. Save the original pixel coordinates and the new coordinates in the image.
10. Embed the watermark into the host image using the LSB technique.
11. Modify the pixel values in the host image based on the mapped coordinates. The least significant bit of the pixels in the host image is used to store the watermark.
12. Save the watermarked image.

3.1.2 Extraction Processes

The extraction process aims to separate the host image from the inserted watermark. Fig. 3.1 shows the flowchart of the watermarked image extraction process.

1. Load the image with watermark from storage.
2. Apply attacks on the watermarked image
3. Extract LSB of the watermarked image. To get the hidden watermark data, extract the smallest bit from each pixel of the watermarked image.
4. Convert the bits extracted from LSB into binary form.

5. Reconstruct the original coordinates and the new coordinates from the binary bits. The coordinates will be used to remap the watermark.

6. Map the extracted coordinates to the watermark value, this means reconnecting the stored coordinates with the corresponding watermark value.

7. Recombine the extracted binary bits and convert the decimal value into the original binary form of the watermark.

3.2 Attacks

Attacks are used to test the robustness of watermarks embedded in images against various disturbances or modifications. These attacks aim to assess how well the watermark remains recognizable or survives despite changes to the image. In this study, several types of attacks are used to evaluate the strength of the watermarking technique under various conditions. The following are the attacks used in this study:

1. Gaussian Noise

   Gaussian noise attack is a method that adds noise to a digital image by following a normal distribution. In the context of watermarking and image processing, Gaussian noise is used to test the robustness of the watermark, as it introduces random value changes according to a normal distribution pattern, which can obscure the embedded watermark [26]. Gaussian noise is widely used to test the robustness of image watermarking against attacks. This attack has been tested in several studies [11] [27] [28] [29] [30] [31].

   The characteristics of Gaussian noise affect the image pixels randomly based on a Gaussian probability distribution with zero mean and a certain variance. This noise can reduce the visual quality of the image, which is measured using PSNR (Peak Signal-to-Noise Ratio). The robustness of the watermark is tested to see how well it can remain recognizable even when the image is subjected to significant noise interference [31].

   The effects of Gaussian noise include a decrease in the visual quality of the image, as the noise introduces random specks that make the image appear rough or grainy. The higher the intensity of Gaussian noise, the worse the visual quality becomes. This results in blurring the details of the image and making the edges of objects unclear. Specifically, Gaussian noise can affect the imperceptibility of the embedded watermark, making it harder

to recognize or even causing it to disappear if the noise intensity is high enough [27] [31].

2. Salt & Pepper Noise

   Salt & Pepper Noise is a commonly used method for corrupting images. In the context of a watermarking attack, random pixels in a grayscale image are altered to black or white. In color images, pixels are randomly selected for modification, with the values of the color channels (R, G, B) changed to either 0 or 255 [20]. This alteration results in random black and white specks appearing across the image. This type of noise is frequently employed to test the robustness of the watermark against extreme distortions, as it can significantly degrade the visual quality of the watermarked image and potentially damage the embedded watermark [27]. Salt & Pepper Noise attacks are widely used in image watermarking tests to assess the robustness of watermarking techniques against such distortions. This method has been evaluated in several studies [11] [27] [28] [29] [30].

   The characteristics of Salt & Pepper Noise involve randomly altering pixels to either black or white (values 0 or 255). Salt & Pepper Noise can cause noticeable specks on the image and often occurs during digital transmission or image compression processes [8]. The effects of Salt & Pepper Noise include a significant reduction in image quality, especially when black and white specks begin to dominate large areas of the image. This noise makes the image appear rougher and reduces detail, particularly in areas with smooth gradients or colors. Additionally, Salt & Pepper Noise can affect the visibility of the embedded watermark, making it difficult to detect or damaging it due to the extreme noise [27] [8].

3. JPEG Compression

   JPEG compression is a widely used image compression technique that reduces the file size of images by sacrificing some visual details. This process involves several stages, including transforming the image from the spatial domain to the frequency domain using the Discrete Cosine Transform (DCT), followed by quantization and entropy coding. The result is a smaller image file with varying degrees of quality loss, depending on the level of compression applied [32]. JPEG compression is commonly used to test the robustness of image watermarking against attacks, and it has been evaluated in several studies [32] [29] [30] [31].

   Image compression reduces data size by removing visual information con-

sidered unnecessary, which can damage the embedded watermark. This issue frequently occurs during image storage and transmission processes, especially when using JPEG compression [31]. JPEG compression can lead to a significant loss of visual detail in the image, particularly in areas with smooth gradients or fine details. The higher the compression level, the worse the image quality becomes, causing the embedded watermark to become blurred or even disappear entirely. Compression can also introduce blocking artifacts, further degrading the visual quality of the image. This reduction in quality can affect the detection and robustness of the watermark against compression-based attacks [32] [31].

4. Rescaling

   Rescaling is the process of changing the size of an image, either by enlarging or reducing it, while maintaining its aspect ratio. In the context of image watermarking, rescaling is used as an attack method to evaluate how well the watermark can withstand changes in image size. The rescaling process can alter the pixel distribution within the image and potentially blur or distort the watermark, thereby testing the robustness of the proposed watermarking scheme [31]. Rescaling is commonly employed to assess the durability of watermarking techniques, and this method has been evaluated in various studies [30] [31].

   The characteristics of rescaling involve changing the image size, either by enlarging or reducing it, which can cause the watermark to become distorted or even disappear due to the change in image resolution [31]. The effects of rescaling include the potential distortion of the embedded watermark, especially when the image is significantly enlarged or reduced. These changes in size can affect the visibility of the watermark or even render it unrecognizable. Additionally, rescaling can damage the image details, which impacts the overall visual quality. If the image resolution is changed too drastically, the watermark may be completely removed or become impossible to extract correctly [31].

5. Low Pass Filter

   A low-pass filter is a filter that allows low-frequency components to pass through while effectively reducing or eliminating high-frequency components from the signal. In the context of image watermarking, a low-pass filter is used to blur fine details in an image and test the watermark's resilience against the loss of high-frequency information. The low-pass filter is employed as one

of the attacks to assess the watermark's robustness by focusing on the low-frequency components of the image, thereby reducing the impact of high-frequency elements that might affect the watermark [31]. Low-pass filters are frequently used in testing the robustness of image watermarks against various attacks, and this method has been evaluated in studies [31].

The characteristics of LPF involve the removal of high-frequency components from the image, resulting in a blurred or softened appearance. This attack specifically targets watermarks embedded in high-frequency components, reducing their visibility or making them more vulnerable to degradation [31]. The effects of an LPF attack include significantly blurring the image, reducing sharpness, and weakening fine details. This blurring effect can impact the embedded watermark, especially if the watermark is hidden within the high-frequency details of the image. As a result, the watermark may become more difficult to detect or even completely unrecognizable, depending on the intensity of the applied filter. This makes LPF attacks a common method for testing watermark robustness, particularly for techniques that rely heavily on frequency-domain methods [31].

6. Speckle Noise

   Speckle noise is a type of interference that affects the visual quality of digital images by introducing granular noise due to random interference when coherent waves interact with rough surfaces. In the context of image watermarking, speckle noise is used as an attack method to evaluate how well a watermark can withstand such distortions. The presence of speckle noise can introduce random specks or grains across the image, potentially obscuring or degrading the watermark, thereby assessing the robustness of the proposed watermarking scheme [27]. Speckle noise is commonly used to assess the durability of watermarking techniques, and this method has been evaluated in various studies [11] [27] [30].

   The characteristics of speckle noise are that it is a type of multiplicative noise, typically found in radar or ultrasound images. This noise introduces random variations across the entire image, which can affect the watermark embedded in both low and high-frequency domains. The effects of speckle noise on an image can include a significant reduction in visual quality, as this noise introduces random specks throughout the image, making the image appear rougher or less smooth. Speckle noise can blur fine details in the image and reduce the visibility of the embedded watermark. In some cases, the watermark may be

damaged or even lost if the intensity of the speckle noise is high enough. This makes speckle noise an effective attack method for testing the robustness of watermarks against disturbances in both low and high-frequency domains [8].

7. Median Filter

   The Median Filter attack is an image processing technique aimed at reducing the presence of noise in images, thereby improving image quality. The Median Filter works by sorting the pixels in the area of the image covered by the filter and then replacing the central pixel value with the median of the surrounding pixel intensities. In the context of watermark attacks, the Median Filter is used due to its ability to remove small signals that are considered noise, including the watermark signal present in the image [33]. The Median Filter is used to test the robustness of image watermarking against attacks, and this method has been evaluated in several studies [27] [28] [30] [32].

   The characteristic of the Median Filter is that it replaces each pixel value with the median of its neighboring pixels, which is effective at removing noise such as salt and pepper. However, this filter can degrade the watermark by smoothing the pixel values that contain the watermark. The effect of the Median Filter on the image is an improvement in visual quality by eliminating noise like salt and pepper, making the image appear smoother. However, since the Median Filter works by replacing pixel values based on the median of surrounding pixels, it can cause damage to the embedded watermark. Watermarks, especially those hidden within the finer details of the image, are often treated as noise by the filter. As a result, after applying the Median Filter, the watermark may become less visible or even completely unrecognizable [8].

3.3 Performance Parameters

In this study there are several parameters used to assess the performance of the system. The parameters used are as follows.

1. Mean Opinion Score (MOS)

   Mean Opinion Score (MOS) is a method used to measure the perceived quality of media, such as audio, video, images, or other audiovisual content. This score is obtained by taking the average of the scores given by a group of subjects in a subjective test. The MOS generally uses a 5-point scale, labeled from “very bad” to “very good”. This scale is designed to capture the general quality perception of the subjects. MOS is a subjective measurement,

which means that the results depend heavily on the perceptions of the individuals participating in the test. Factors such as the testing environment, the subject's level of expertise, and physical and mental state can affect the judgment given [34]. The Mean Opinion Score (MOS) scale can be seen in Table [35].

Table 3.1 MOS (Mean Opinion Score) Scale

MOS	Quality	Impairment
1	Bad	Very Annoying
2	Poor	Annoying
3	Fair	Slightly Annoying
4	Good	Perceptible but not Annoying
5	Excellent	Imperceptible

2. Bit Error Rate (BER)
   Bit Error Rate (BER) is a parameter used to calculate the bit error rate received by the system after the watermark extraction process. The BER value has a significant effect on image quality, where the closer the BER value is to 0, the greater the similarity between the extracted watermark and the original, which indicates the stronger it is. The following formula can determine the BER value [12]:

$$BER = \frac{Number\ of\ Wrong\ bits}{Number\ of\ bits} \times 100\% \tag{3.1}$$

3. Peak Signal to Noise Ratio (PSNR)
   Peak Signal-to-Noise Ratio (PSNR) is a parameter used to measure how good the image quality is after the watermark insertion process. PSNR evaluates the invisibility requirements by comparing the similarity of the original image file with the watermarked image file. The higher the PSNR value, the better the watermarking quality. This means that the watermarked image is very close to the original image. PSNR is measured in decibels (dB). The PSNR value can be determined by the following formula [12]:

$$PSNR = 10log_{10}\left(\frac{d^2}{MSE}\right) \tag{3.2}$$

   where:

   - d is the maximum pixel value of the image. For an 8-bit per channel

image, the d value is typically 255.

- MSE (Mean Squared Error) is the average of the squared differences between the original and distorted image pixels, calculated by the formula:

$$\text{MSE} = \frac{1}{M \times N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left(I_0(x,y) - I_w(x,y)\right)^2 \tag{3.3}$$

where:

- M and N are the dimensions of the image.
- $I_0(x,y)$ is the pixel value at position (x,y) in the original image.
- $I_w(x,y)$ is the pixel value at position (x,y) in the distorted image.

High PSNR values can be achieved with low MSE, which indicates minimal degradation. In general, PSNR above 40 dB indicates that the watermark is not visible, while values below 30 dB indicate significant distortion.

4. Payload

Payload is a measure that indicates how many bits are used to represent the watermark in each pixel of the original image. Usually, the payload is encoded as bits per pixel (bpp), which indicates the number of bits that can be inserted in each pixel of the image. The higher the payload value, the more data can be stored in the watermark. Payload value can be determined by the following formula [36].

$$\text{Payload} = \frac{\text{Number of embedded bits}}{\text{Total number of host image pixels}} \tag{3.4}$$

CHAPTER IV
PERFORMANCE EVALUATIONS

This study evaluates the performance of the proposed scheme using seven test images: 'Airplane', 'Baboon', 'Barbara', 'Boat', 'Peppers', 'Sailboat', and 'Zelda'. Each image has a resolution of 256x256 pixels. The secret image W, in binary form (0 or 1), is generated using a random number generator. These test images were obtained from https://www.hlevkin.com/hlevkin/06testimages.htm, which provides frequently used image datasets for MATLAB/C/Python/Shell programming and image/video processing and compression. The analysis is performed on a Windows Professional 11 64-bit operating system using MATLAB R2024a with an academic license. Fig. 4.1 shows the five grayscale images and the watermark image used in this research.

Fig 4.1 Seven 256 × 256 grey-scale images and their watermarked images

The testing process is divided into two stages: embedding testing and extraction

testing. Embedding testing aims to measure the Peak Signal-to-Noise Ratio (PSNR) and embedding capacity. PSNR evaluates the visual quality of the watermarked image, ensuring that the embedding process does not significantly degrade the image quality. Embedding capacity determines the amount of watermark data that can be inserted into the image without compromising its visual integrity.

In addition to technical testing using PSNR, to test the quality of embedding subjectively, an assessment is made using Mean Opinion Score (MOS). MOS is an evaluation method that involves a group of respondents to assess the visual quality of an image that has been embedded with a watermark. The respondents are asked to give an assessment based on their visual perception of the watermarked image, which is then collected and averaged into an MOS score.

The MOS assessment aims to measure how good the embedding quality is from the perspective of human vision. This method is very important because while the PSNR value can indicate the technical quality of an image, it does not always reflect how that quality is received by a human observer. By using MOS, this research can combine objective test results with subjective assessments to provide a more comprehensive picture of the watermark embedding quality. This helps to ensure that the embedded watermark is not only invisible to automated systems, but also to human observers, thus maintaining the visual quality of the image.

Extraction testing evaluates the reliability of the watermark after the embedding process. During this stage, various attacks are applied to the watermarked images for further analysis. The purpose of these attacks is to test the robustness of the watermark against different types of disturbances or manipulations that may occur in the images. After the attacks are applied, the extraction process assesses how well the watermark can be recovered and measures its resistance to the applied attacks. This comprehensive testing ensures the effectiveness and efficiency of the watermarking technique used.

4.1 Analysis of Watermark on Color Channel

In this research, the main focus is on grayscale images. However, in this subsection, a test is conducted by embedding a watermark into one of the color channels, specifically the red channel. The purpose of this test is to determine whether a watermark can be embedded into a color image through a specific channel. In this case, the red channel is selected to observe if watermarking can be applied without significantly altering the visual appearance. The test shows that the watermark is successfully embedded into the red channel, and the result is shown in Fig. 4.2. Al-

though the watermark is applied to only one color channel, the result demonstrates that this technique works effectively for color images.

(a)
Host Image

(b)
Watermarked image

Fig 4.2 Original Image and Watermarked Image Peppers in Red Channel

In this test, an image of peppers consisting of green, yellow, and red peppers is used. The red pepper has a higher color intensity, while the green and yellow peppers have lower red intensity. The red channel is one of the three channels in the RGB color model (Red, Green, Blue), which is responsible for capturing and displaying the red components in the image. In this image of peppers, the red channel dominates the red pepper areas while providing lower intensity for the green and yellow objects.

In Fig. 4.2 (a), a full-color image is shown, containing information from all three RGB channels. The parts of the image with red objects, such as the red pepper, appear brighter, while areas with green or yellow objects appear darker due to the lower red intensity. Fig. 4.2 (b) shows the result after the watermark is applied to the red channel. This image is displayed in grayscale to illustrate the changes that occur in the red channel after the watermark is embedded. Areas rich in red color appear brighter, while regions with less red appear darker, reflecting the differences in intensity within the red channel.

Fig. 4.2 (b) also demonstrates that the watermark has been embedded into the red channel, and the result shows subtle changes in color intensity in some parts of the image. The watermark is applied using the Least Significant Bit (LSB) method, modified with the turtle shell approach. In the LSB method, the watermark is embedded into the least significant bits of each pixel in the red channel, allowing the watermark to be inserted without drastically altering the visual appearance of the image. In the watermarked image, even though the watermark has been embedded, the image remains very similar to the host image. The red channel experiences slight changes in intensity, but these changes are not easily noticeable to the human

eye.

The test results show that the PSNR (Peak Signal-to-Noise Ratio) value obtained is 61.18 dB, with a watermark bit length of 800 bits. This indicates that the image quality remains very high after the watermark is embedded. The high PSNR value demonstrates that the watermark does not significantly degrade the visual quality of the image. Although areas with intense red color may be slightly affected, the difference is barely noticeable, and the watermark does not interfere with the overall visual perception of the image.

4.2 Image Embedding Quality Analysis

In this analysis, testing is conducted to evaluate the impact of bit quantity on PSNR and storage capacity on PSNR. Table 4.1 presents the results of the embedding process tests, with measured parameters including Bit, PSNR, and Payload. Peak Signal-to-Noise Ratio (PSNR) is a parameter used to measure the quality of an image after the watermark embedding process. PSNR evaluates the transparency level by comparing the similarity between the original image file and the image file that has been embedded with a watermark. The higher the PSNR value, the better the watermarking quality, indicating that the watermarked image closely resembles the original image. PSNR is measured in decibels (dB). The PSNR value can be calculated using a formula that involves comparing the maximum signal in the original image with the noise generated after the watermark is embedded. In the context of watermarking, PSNR helps ensure that the visual quality of the image remains intact even after the watermark is embedded.

However, in this embedding test, the BER value is not discussed as it consistently remains zero. This is due to the extraction process being based on predetermined coordinates, ensuring that the embedding values and coordinates are always aligned. The zero BER value occurs because the embedding process ensures that the embedded image and the extracted image do not undergo any changes, thanks to the turtle shell modification that maps the image to fixed coordinates. During extraction, the watermark image is remapped using the same coordinates. Therefore, this analysis will focus solely on discussing PSNR, payload, and watermark bit length.

The results of the image embedding process can be seen in Table 4.1 below.

Table 4.1 Testing results of the embedding process

Bit	Payload (bpp)	PSNR (dB)						
		Airplane	**Baboon**	**Barbara**	**Boat**	**Peppers**	**Sailboat**	**Zelda**
800	0.012	61.18	61.25	61.18	61.17	61.19	61.15	61.20
1600	0.024	58.20	58.25	58.29	58.24	58.26	58.27	58.27
2400	0.037	56.43	56.51	56.48	56.40	56.44	56.47	56.50
3200	0.049	55.24	55.21	55.25	55.18	55.23	55.23	55.22
4000	0.061	54.28	54.30	54.26	54.28	54.25	54.27	54.23
4800	0.073	53.45	53.45	53.48	53.50	53.50	53.46	53.49
5600	0.085	52.81	52.78	52.80	52.78	52.80	52.81	52.82
6400	0.098	52.24	52.20	52.21	52.21	52.22	52.22	52.23
7200	0.110	51.69	51.68	51.70	51.68	51.67	51.70	51.68
8000	0.122	51.24	51.26	51.25	51.24	51.24	51.27	51.23

4.2.1 Comparative Analysis of Watermark of Bit Length with PSNR

Table 4.1 shows the test results of the embedding process with various watermark bit lengths and the PSNR values for various test images. The watermark with bit length inserted into the image significantly affects the resulting PSNR value. From the data presented, it can be seen that there is a downward trend in PSNR value as the watermark bit length increases. This happens because the more bits that are inserted, the greater the changes that occur in the original image, resulting in a decrease in the visual quality of the image.

In the low bit lengths, such as 800 bits, the PSNR value ranges around 61 dB for all test images, which indicates that the image quality after embedding the watermark remains very high. However, as the bit length increases to 8000 bits, the PSNR value decreases significantly to about 51 dB.

This decrease in PSNR value indicates that the visual quality of the image decreases as the number of watermark bits inserted increases. This is due to the greater modification of the pixels of the original image as the inserted bits increase, resulting in increased distortion. Although the PSNR value decreases, values above 50 dB are still considered good and are generally not easily distinguishable by a casual observer.

The bit length of the watermark is inversely proportional to the quality of the resulting image. The more bits inserted, the lower the PSNR value, which reflects the decrease in imperceptibility of the watermark. Thus, in the image watermarking

process, it is necessary to consider the appropriate watermark bit length to maintain the visual quality of the image.

4.2.2 Comparative Analysis of Payload with PSNR

Table 4.1 shows the test results of the embedding process with various payloads and PSNR values for various test images. From the data presented, there is a clear relationship between payload and PSNR; as the payload increases, the PSNR value consistently decreases.

Payload is measured in bits per pixel (bpp), which indicates the total amount of secret data that can be inserted or potentially inserted into an image or other media. Payload reflects how much secret information can be hidden without significantly compromising the quality of the media. At low payloads, the PSNR value is relatively high, which indicates that the visual quality of the image is hardly affected by watermark embedding. However, as the payload increases, the PSNR value decreases significantly.

The decrease in PSNR as the payload increases indicates a trade-off between data embedding capacity and image quality. When more data is inserted into an image, the visual quality of the image tends to degrade due to greater modifications made to the image pixels. Nonetheless, PSNR values above 50 dB can still be considered good enough in some contexts, although distortions may start to become visible to some observers. Therefore, to maintain high visual quality, the payload should be kept low.

4.3 Analysis of Imperceptibility using Mean Opinion Score (MOS)

Mean Opinion Score (MOS) is a collection of subjective assessments obtained from a number of respondents to support the results of this research. The assessment is carried out through the process of observation by respondents, based on the Human Visual System (HVS), in comparing the original image (host) with the image that has been waitmarked. In this Mean Opinion Score (MOS), the assessment is carried out on a scale of 1 to 5, the smaller the number, the more visible the difference between the Host Image and the Watermarked Image.

Although PSNR is already used as an objective metric to measure image quality, MOS remains necessary because it provides an important subjective perspective. PSNR only measures the pixel intensity differences between the original and processed images mathematically, without considering how humans perceive and

evaluate the quality of the image. MOS, on the other hand, is based on human perception and allows us to understand how users perceive the visual quality of an image after a watermark has been embedded. This is important because the human eye may not detect small distortions that could affect the PSNR value. Therefore, the combination of PSNR and MOS provides a more comprehensive and thorough evaluation of the quality of the embedded watermark in the image.

Table 4.2 is an assessment scale considered by respondents on the assessment of both images, namely the host image and watermarked image.

Table 4.2 MOS Assessment Description Scale

Scale	**Rating Description**	**Remarks**
1	Different and Very Distracting	The difference is clear and very distracting
2	Different and Distracting	The difference is noticeable and somewhat distracting
3	Different and Slightly Distracting	The difference is noticeable and slightly distracting
4	Slightly Different	The difference is not distracting
5	Exactly the Same	No difference at all

The testing was conducted by distributing a questionnaire. From the distribution, 50 respondents provided their evaluations, comparing the original images with the watermarked images across several samples, including images of an Airplane, Baboon, Barbara, Boat, Pepper, Sailboat, and Zelda. The evaluations were made using a scale from 1 to 5, and the results can be seen in Table 4.3.

Table 4.3 Mean Opinion Score (MOS) for Different Image Types

Host Images	**MOS**
Airplane	4.52
Baboon	4.45
Barbara	4.52
Boats	4.43
Pepper	4.53
Sailboat	4.53
Zelda	4.37

Mean Opinion Score (MOS) is used as a subjective indicator to assess the visual quality of a watermarked image. MOS gives an idea of how visible the watermark embedded in the host image is based on the user's perception. In this study, the

MOS values for various host images ranged from 4.37 to 4.53, indicating that the watermarking on those images was generally considered unobtrusive or almost invisible to the human eye.

the average MOS value for the entire host image is 4.48. This value indicates that overall, the proposed watermarking method has high imperceptibility. In other words, the watermark embedded in the images is not significantly visible to the user, which indicates that the method is effective in preserving the visual quality of the watermarked images.

4.4 Analysis of Watermark Robustness to Attack

This testing aims to evaluate the robustness of the watermark on images against various types of interference and manipulation. A series of attacks are applied to watermarked images to measure how well the watermark can survive and remain identifiable after distortion. The main parameter used in this testing is Bit Error Rate (BER), which measures the percentage of bit errors during the watermark extraction process. The lower the BER, the better the watermark's robustness to distortion and attacks.

In the context of image watermarking, BER serves as an indicator of the accuracy of watermark extraction after the image undergoes an attack. A high BER indicates that many watermark bits were not extracted correctly, meaning the watermark has poor resistance to attacks. Conversely, a low BER signifies that the watermark was successfully extracted, even after the image has been distorted, indicating a high level of watermark robustness.

If the BER exceeds 0.4, the watermarking is considered not resistant to attacks. This is because more than 40% of the watermark bits are corrupted or lost, making the watermark information inaccurate or difficult to recognize. With this level of error, the watermark can no longer be identified or validated, which undermines its function as an identifier or authentication tool. A high BER is also often accompanied by a reduction in the visual quality of the image, showing that the watermarking method is not robust enough to protect the watermark from attacks or manipulation. Therefore, when the BER exceeds 0.4, the watermark is considered to have failed in maintaining its integrity.

The attacks used in this testing include Gaussian noise, Salt and Pepper noise, compression, Rescaling, Low-Pass Filter (LPF), Speckle noise, and Median Filter.

4.4.1 Watermark Robustness to Gaussian Noise attack

Gaussian noise is a type of disturbance that frequently occurs in image and signal processing. The effect of Gaussian noise makes the image appear blurry or grainy. At low levels, Gaussian noise is almost invisible to the eye, but if the intensity of the disturbance is high, the image experiences significant damage due to the noise effect. In Fig. 4.3, the effect of a Gaussian noise attack with a sigma value of 0.5 is shown.

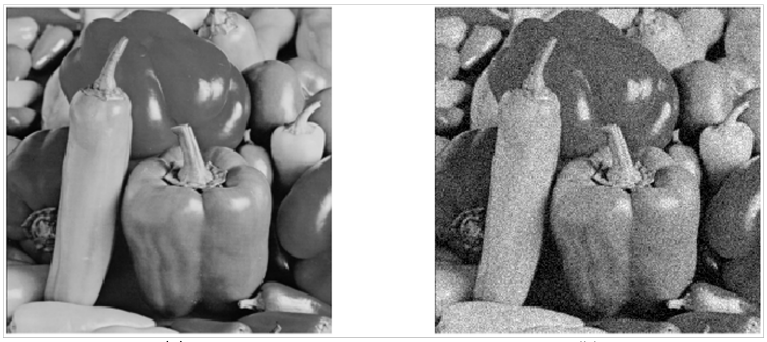


Fig 4.3 Effect of Gaussian Noise Attack (0.05)

Table 4.4 shows the results of watermark robustness testing against Gaussian noise attacks at various sigma values. In this context, sigma represents the standard deviation of the Gaussian distribution used to add noise to the image. The higher the sigma value, the greater the level of disturbance added to the image, which ultimately increases distortion in the original image.

Table 4.4 Watermark Robustness to Gaussian Noise Attack

Sigma	**BER**						
	Airplane	**Baboon**	**Barbara**	**Boats**	**Peppers**	**Sailboat**	**Zelda**
0.000001	0.10	0.08	0.09	0.09	0.08	0.10	0.09
0.0000015	0.21	0.20	0.21	0.19	0.22	0.19	0.20
0.000002	0.24	0.25	0.29	0.29	0.27	0.25	0.25
0.0000025	0.32	0.32	0.32	0.33	0.34	0.32	0.33
0.000003	0.36	0.37	0.33	0.36	0.34	0.35	0.36
0.0000035	0.38	0.36	0.38	0.38	0.37	0.39	0.39
0.000004	0.40	0.41	0.38	0.40	0.42	0.40	0.40

From Table 4.4, it is seen that as the sigma value increases, there is an increase in the Bit Error Rate (BER) for all test images. At the lowest sigma (0.000001),

BER ranges from 0.08 to 0.10, showing that the watermark is relatively unaffected by very low levels of noise. However, as sigma increases to 0.000004, BER rises to between 0.38 and 0.42, indicating greater distortion of the watermark.

Even so, the increase in BER is not too significant with each sigma increase, indicating that Gaussian noise still shows good robustness to moderate levels of noise. In other words, even as sigma increases, the resulting BER remains within acceptable limits, allowing the watermark's integrity to be maintained.

A graph illustrating the relationship between sigma and BER, as shown in Fig. 4.4, shows that although there is an increase in BER as sigma rises, this change remains within moderate bounds. This shows that the watermarking method used is quite resistant to Gaussian noise attacks, except at very high noise intensities.

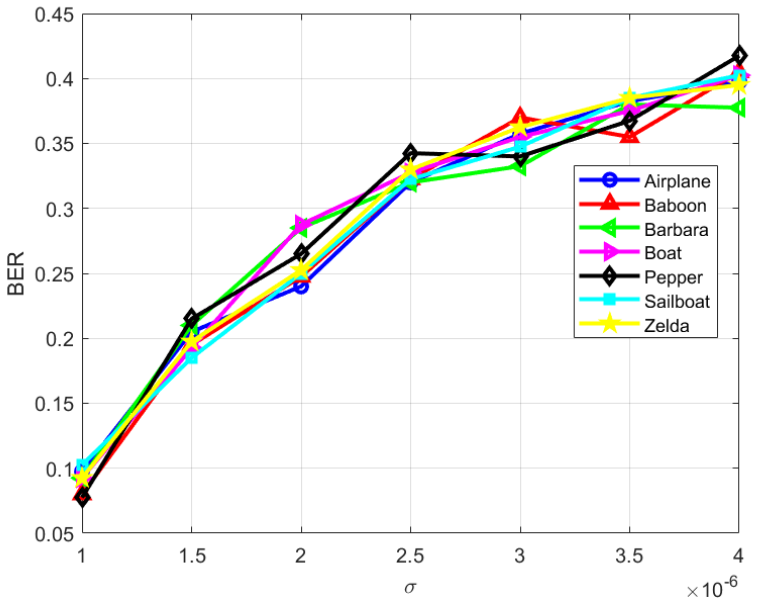


Fig 4.4 Watermark Robustness to Gaussian Noise Attack on seven host Images

In the Gaussian noise attack, this research demonstrates robustness against the noise due to the use of the LSB method and the modified Turtle Shell. In this method, the noise is evenly distributed across the entire image. This causes the image to appear with random disturbances spread throughout, so with low-intensity attacks, the watermark is relatively unaffected. However, with high-intensity attacks, the image and watermark experience more significant distortion.

The even distribution of noise in the image and the Turtle Shell method provides an advantage, as duplicate information remains in other areas of the image if an attack occurs. The Turtle Shell method works by utilizing specific coordinate patterns within the image, ensuring that when the image undergoes a Gaussian noise

attack, the watermark is still preserved. The coordinates and patterns in the Turtle Shell method ensure that, even with distortion, the watermark does not completely disappear and can still be extracted.

4.4.2 Watermark Robustness to Salt & Pepper attacks

Salt & Pepper noise is a type of distortion characterized by the random appearance of white and black pixels in an image. The name ”salt and pepper” comes from the scattered white and black dots that resemble grains of salt and pepper in the image. This noise is often used to simulate data corruption caused by imperfect communication channels or storage devices. The characteristics of Salt & Pepper noise involve randomly changing pixels in the image to black (0) or white (255), resulting in rough specks across the image. This leads to a significant reduction in visual quality, especially in areas with smooth gradients or soft colors. Fig. 4.5 shows the effect of Salt & Pepper noise with a sigma value of 0.04.

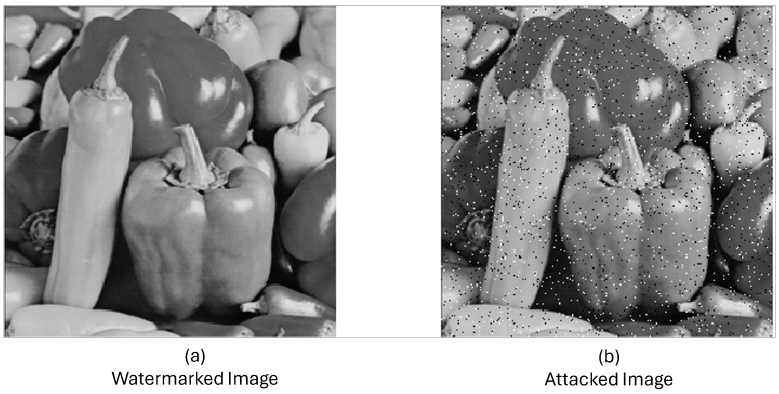


Fig 4.5 Effect of Salt & Pepper Noise Attack (0.04)

Table 4.5 shows the results of testing the robustness of the watermark against Salt and Pepper noise attacks at various sigma values. In the context of Salt & Pepper attacks, sigma represents the intensity or density of the noise added to the image. The higher the sigma value, the more pixels are randomly changed to black (0) or white (255), increasing the distortion in the original image.

Table 4.5 Watermark Robustness to Salt and Pepper Noise Attack

Sigma	BER						
	Airplane	**Baboon**	**Barbara**	**Boat**	**Peppers**	**Sailboat**	**Zelda**
0.02	0.003	0.02	0.02	0.01	0.01	0.01	0.02
0.04	0.03	0.03	0.03	0.03	0.02	0.03	0.03
0.06	0.06	0.06	0.04	0.07	0.04	0.04	0.06
0.08	0.07	0.06	0.07	0.07	0.08	0.07	0.07
0.1	0.10	0.09	0.08	0.10	0.08	0.10	0.08
0.2	0.17	0.19	0.19	0.18	0.16	0.18	0.19
0.4	0.30	0.31	0.30	0.31	0.33	0.31	0.31
0.6	0.38	0.39	0.39	0.37	0.40	0.39	0.36

From Table 4.5, it can be seen that although there is an increase in Bit Error Rate (BER) with the rise in sigma, the change is not always significant for all test images. At the lowest sigma (0.02), the BER ranges from 0.003 to 0.02, indicating that the watermark is relatively unaffected by low levels of noise. As sigma increases to 0.6, the BER rises to between 0.36 and 0.40, indicating greater distortion, but not enough to fully threaten the integrity of the watermark.

The increase in BER shows that the robustness of the watermark against Salt & Pepper noise decreases as sigma increases. However, at lower sigma values, the change in BER is relatively small, indicating that the watermark remains resistant to noise at those levels. Only at higher sigma values does the increase in BER become more significant, indicating greater distortion and a potential threat to the integrity of the watermark.

A graph illustrating the relationship between sigma and BER, as shown in Fig. 4.6, demonstrates that although there is an increase in BER with the rise in sigma, this change remains within moderate bounds at lower sigma levels. This suggests that the watermarking method used has good resistance to Salt & Pepper noise attacks, except at higher noise intensities.

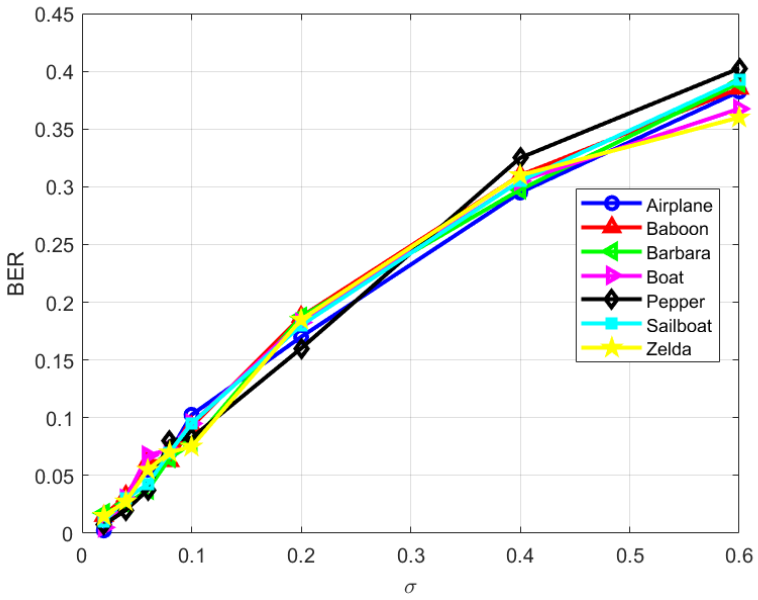


Fig 4.6 Watermark Robustness to Salt and Pepper Noise Attack on seven host Images

In the Salt and Pepper attack, this research shows robustness against the attack due to the use of the LSB method and a modified Turtle Shell. The Salt and Pepper attack introduces random black and white specks across the image, causing disruptions in pixel values. In the LSB method, although some of the pixels containing the watermark may be affected by these specks, the watermark is distributed across the least significant bits of many pixels. This redundancy allows the watermark to remain recoverable even if parts of it are damaged, as sufficient information remains intact in unaffected pixels.

Similarly, in the Turtle Shell method, the watermark is stored according to a specific coordinate system spread throughout the image. Even when certain areas are disturbed by the noise, the overall coordinate pattern used by the Turtle Shell remains preserved in other parts of the image. This ensures that the watermark is not entirely lost, as the structured redundancy helps to protect the watermark from being fully corrupted by the noise. This makes the method resistant to varying intensities of Salt and Pepper noise attacks.

4.4.3 Watermark Robustness to Compression Attack

Compression often aims to reduce file size without losing important information. However, compression can alter pixel values in the image, which may affect the embedded data. JPEG compression, for example, works by transforming the image from the spatial domain to the frequency domain, and during this process,

some information considered less important is discarded. In Fig. 4.7, the effect of a compression attack with a quality level of Quality 40 is shown. The result of this attack shows a blur effect on the image, caused by the removal of high-frequency components and aggressive quantization in JPEG compression. At this compression level, fine details and image edges are removed, making the image appear blurry. In addition, compression also produces blocky patterns that disrupt the transition between parts of the image and sacrifices color and texture details, further degrading the image's visual quality.

Fig 4.7 Effect of Compression Attack (40)

Table 4.6 shows the results of testing the robustness of the watermark against compression attacks at various compression quality levels (Q%). The Q% value represents the quality level of compression, where a higher value indicates lower compression, and a lower value indicates higher compression.

Table 4.6 Watermark Robustness to Gaussian Noise Attack

Q%	**BER**						
	Airplane	**Baboon**	**Barbara**	**Boat**	**Peppers**	**Sailboat**	**Zelda**
80	0.47	0.44	0.45	0.48	0.45	0.46	0.43
90	0.44	0.46	0.44	0.46	0.46	0.47	0.46
100	0.19	0.17	0.17	0.17	0.18	0.17	0.13

From Table 4.6, it can be seen that when the compression quality level is at 100%, the Bit Error Rate (BER) is relatively low, ranging from 0.13 to 0.19 across all test images. This indicates that at this compression level, the watermark can still be extracted with a low bit error rate. However, when the compression quality is reduced to 90% and 80%, the BER increases significantly. At Q% 90, the BER

ranges from 0.44 to 0.47, and at Q% 80, the BER ranges from 0.43 to 0.48. This increase shows that the watermark becomes less robust to compression, as the higher distortion caused by compression leads to a significant increase in the bit error rate during watermark extraction..

A graph illustrating the relationship between compression quality and BER, as shown in Fig. 4.8, demonstrates that as the Q% value (higher compression quality) increases, the BER decreases. Conversely, when Q decreases (higher compression), the BER increases, indicating that the watermark is more vulnerable to distortion and bit errors under more aggressive compression conditions.

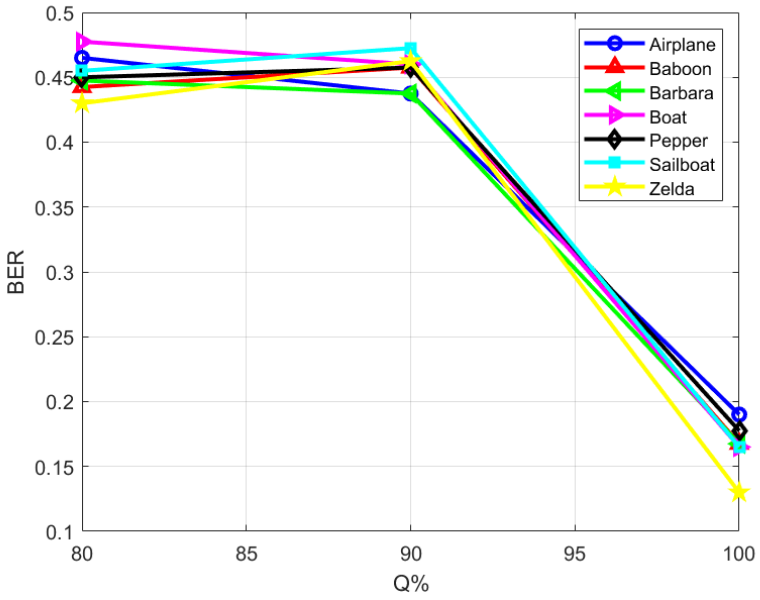


Fig 4.8 Watermark Robustness to Compression on seven host Images

In this compression attack, the watermark embedded in this research lacks robustness against compression attacks. This is due to the use of the LSB (Least Significant Bit) method and the modified Turtle Shell. Since LSB works in the spatial domain directly on the image pixels, during the compression process, pixel values are altered or removed to reduce the file size. As a result, the watermark embedded in the least significant bit of each pixel is lost or damaged because compression changes or removes these bits.

The Turtle Shell method used in this research is also affected by compression. This method relies on coordinates within the host image, where the coordinates of the length and width must match the original image. If there is a change in the image size due to compression, the coordinate information in the Turtle Shell is lost as well, because compression removes part of this information. As a result, both the LSB and Turtle Shell methods experience significant damage when subjected to

compression attacks.

4.4.4 Watermark Robustness to Low Pass Filter (LPF) attack

Low Pass Filter (LPF) is a method used to reduce or eliminate high frequencies from a signal or image. The characteristic of LPF is its ability to attenuate high-frequency components while retaining low frequencies, which often results in a smoother or blurred image. In Fig. 4.9, the effect of a Low Pass Filter attack with a filterSize of 11 is shown. The effect of this attack causes the image to become blurry because LPF removes high-frequency details.

Fig 4.9 Effect of Low Pass Filter Attack (11 x 11)

Table 4.7 shows the results of testing the watermark's robustness against Low Pass Filter (LPF) attacks at various filter sizes. Filtersize represents the size of the filter kernel used to attenuate high frequencies in the image, so that only low frequencies are retained. This process usually smooths the image and can cause the loss of important details, including the watermark.

Table 4.7 Watermark Robustness to Low Pass Filter Attacks

filtersize	**BER**						
	Airplane	**Baboon**	**Barbara**	**Boat**	**Peppers**	**Sailboat**	**Zelda**
3	0.46	0.45	0.47	0.46	0.46	0.45	0.46
5	0.47	0.47	0.47	0.45	0.46	0.46	0.46
7	0.48	0.47	0.45	0.46	0.46	0.45	0.45

From table 4.7, it can be seen that at all filter sizes tested (3, 5, and 7), the Bit Error Rate (BER) remains above 0.4 for all test images, with values ranging between 0.45 and 0.48. These high BER values indicate that the watermark experiences significant distortion when the image is filtered using LPF. This shows that

the watermarking method used is not robust to LPF attacks, as BER values above 0.4 indicate that the watermark undergoes drastic changes and is difficult to extract correctly.

The consistent increase in BER values across all filter sizes, as well as the high BER values, indicates that the watermark is not robust against Low Pass Filter (LPF) attacks. LPF effectively removes high-frequency components from the image, which most likely includes the inserted watermark. As a result, the quality of the watermark is significantly compromised, and the ability to extract the watermark correctly is drastically reduced. This suggests that in the context of using LPF, a watermarking method more resistant to high-frequency loss is needed so that the watermark can be preserved despite attacks such as LPF.

Fig. 4.10 shows the relationship between the filter size used in Low Pass Filter (LPF) attacks and Bit Error Rate (BER) for seven different test images. From this graph, it can be seen that BER values remain high across all filter sizes tested, indicating that the watermark embedded in the image cannot withstand LPF attacks.

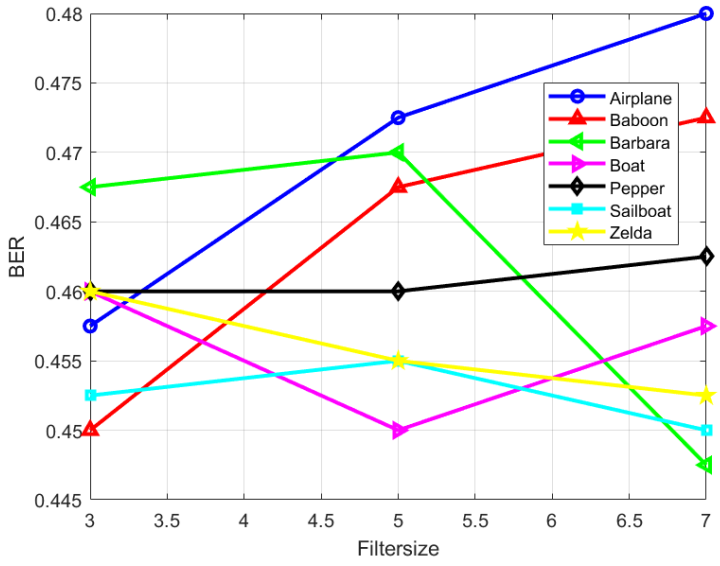


Fig 4.10 Watermark Robustness to Compression on seven host Images

In this LPF attack, the watermarking method used in this research does not demonstrate robustness against LPF attacks. This is due to the use of the LSB method and the modified Turtle Shell. LSB operates in the spatial domain, directly on the image's pixel values, while LPF removes high frequencies and smooths the image by altering pixel values. Since the LSB watermark is embedded in the least significant bits of the pixels, changes in pixel values due to LPF can damage or remove the watermark. LSB relies on small pixel changes, so when LPF modi-

fies pixels to remove fine details, the watermark becomes corrupted and difficult to extract. Therefore, LSB is not robust against attacks that occur in the frequency domain, such as LPF.

The Turtle Shell method describes the movement of coordinates within the image. In this technique, significant changes in size or frequency lead to large coordinate shifts, while LPF only retains low frequencies. As a result, important coordinate data from the Turtle Shell is not correctly read or is lost during the filtering process, causing the watermark to become difficult to extract.

4.4.5 Watermark Robustness to Rescaling attack

Rescaling is the process of changing the size of an image, either by enlarging or reducing it, while maintaining its aspect ratio. The purpose is to adjust the image dimensions as needed without losing key information. The effects of rescaling include changes in the image's dimensions and spatial resolution. This process can affect the integrity of the watermark, making it more or less visible, and impacting the overall quality and durability of the watermarked image. The characteristics of a rescaling attack involve resizing the image, either by enlarging or reducing it, which can cause the watermark to become distorted or even completely disappear. In Fig. 4.11, the effect of a rescaling attack with a scaling factor of 0.25 is shown. The effect of this attack results in blocky blurring, caused by significant scaling changes that lead to pixel interpolation or repositioning, making the image lose fine details and appear fragmented.

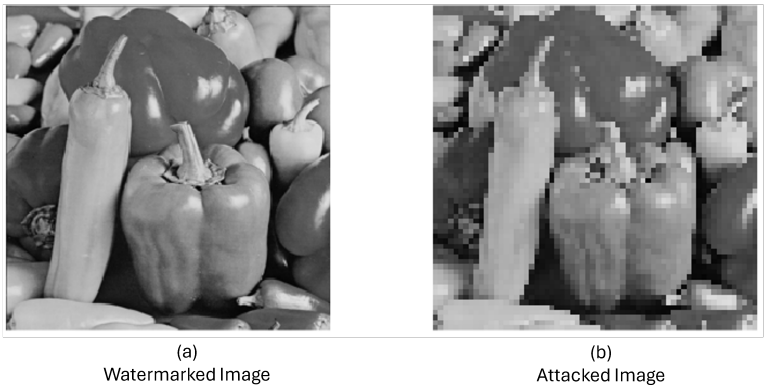


Fig 4.11 Effect of Rescaling Attack (0.25)

Table 4.8 shows the results of testing the watermark's robustness to rescaling attacks at various scaling values. In this context, the scaling value represents the degree of change in image size, either reducing or enlarging the image. This change

in image size can cause significant distortion to the watermark, which is evident from the change in Bit Error Rate (BER).

Table 4.8 Watermark Robustness to Rescaling Attacks

Scaling	**BER**						
	Airplane	**Baboon**	**Barbara**	**Boat**	**Peppers**	**Sailboat**	**Zelda**
0.125	0.45	0.45	0.46	0.48	0.43	0.47	0.44
0.25	0.46	0.45	0.44	0.46	0.45	0.48	0.47
0.5	0.45	0.46	0.44	0.46	0.43	0.44	0.44
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.43	0.42	0.46	0.45	0.46	0.46	0.46
4	0.46	0.45	0.46	0.46	0.44	0.46	0.46
8	0.44	0.46	0.47	0.44	0.47	0.44	0.44

From Table 4.8, it can be seen that at scales larger than 1 (2, 4, 8) and smaller than 1 (0.125, 0.25, 0.5), the BER values consistently remain high, ranging from 0.43 to 0.48 across all test images. However, at a scale of 1, the BER values for all tested images are 0. This happens because at a scale of 1, there is no change in the image size, so the watermark remains intact without distortion, and no attack occurs that could alter the BER.

As shown in Fig. 4.12, the BER values tend to stay above 0.4 for both small and large scale values, demonstrating consistency in the graph. However, when the scale reaches 1, the graph drops immediately to zero and then rises again as the scale increases. This happens because the rescaling process causes significant distortion to the image, both when the image is enlarged and reduced, making it difficult to extract the watermark correctly. These results indicate that watermarked images are not robust to rescaling attacks, as even small changes in scale significantly affect the integrity of the watermark and lead to substantial degradation in image quality. The high BER values across different scale levels suggest that current watermarking methods may not be effective enough to protect against distortions caused by rescaling.

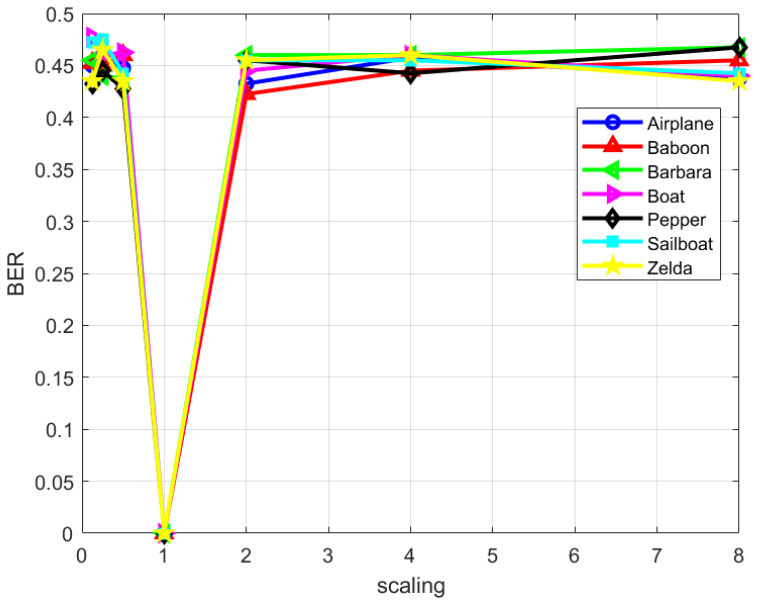


Fig 4.12 Watermark Robustness to Rescaling on seven host Images

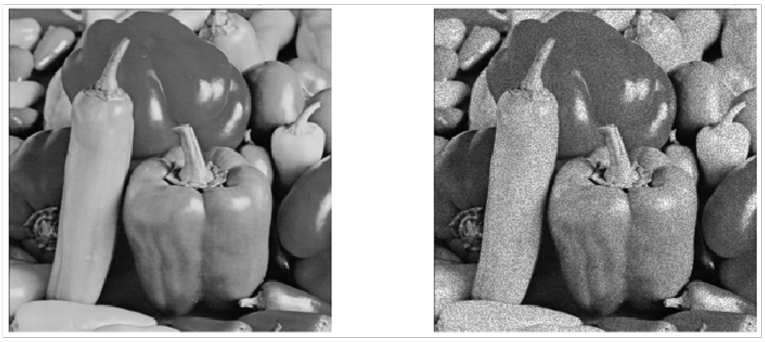
In this rescaling attack, the watermark embedded in this research is not robust to the attack. This is due to the use of the LSB method and Turtle Shell modification. In the LSB method, the watermark is embedded directly into the least significant bits of the image pixels. When the image scale is changed, the pixel distribution is altered, causing the watermark information hidden in those bits to become distorted or completely lost due to changes in the position and value of the pixels.

Meanwhile, in the Turtle Shell modification, the watermark is stored using specific coordinates within the image, which must match the host image's size. When the image scale changes, the size and position of the coordinates are also affected, so the information embedded in the Turtle Shell coordinates no longer aligns with the resized host image. As a result, the watermark data that relies on these coordinates is lost or becomes inaccessible, making the watermark difficult or even impossible to extract.

4.4.6 Watermark Robustness to Speckle noise

The speckle noise attack introduces random noise spots across the image. The characteristic of speckle noise is that it is multiplicative, typically found in radar or ultrasound images. The effect of speckle noise significantly reduces the visual quality of the image by adding random specks throughout the image, which can affect the watermark embedded within it. In Fig. 4.13, the effect of the speckle noise attack with a sigma value of 0.01 is shown. The effect is represented by white

spots resembling snowflakes.



(a) Watermarked Image (b) Attacked Image

Fig 4.13 Effect of Speckle Noise Attack (0.01)

Table 4.9 shows the test results of watermark robustness to Speckle noise attack at various sigma values. Speckle noise is a type of multiplicative noise commonly found in images generated by wave-based imaging systems, such as radar and ultrasound. The sigma value in this context indicates the intensity of the noise added to the image.

Table 4.9 Watermark Robustness to Speckle Attacks

Sigma	**BER**						
	Airplane	**Baboon**	**Barbara**	**Boat**	**Peppers**	**Sailboat**	**Zelda**
0.000005	0.003	0.01	0.01	0.01	0.01	0.01	0.02
0.000007	0.14	0.16	0.18	0.17	0.17	0.16	0.15
0.000009	0.30	0.28	0.30	0.31	0.28	0.29	0.30
0.000011	0.37	0.40	0.38	0.38	0.37	0.37	0.38
0.000013	0.40	0.40	0.39	0.42	0.43	0.42	0.39
0.000015	0.42	0.42	0.43	0.43	0.45	0.43	0.43

Table 4.9 shows the results of testing the watermark's robustness against speckle noise attacks at various sigma values. In this context, sigma represents the intensity of the noise added to the image. At a small sigma value (0.000005), the Bit Error Rate (BER) is low, ranging from 0.003 to 0.02, indicating that the watermark is well preserved at low noise intensity. However, as the sigma value increases, the BER also rises, although the increase is not significant. At a sigma value of 0.000015, the BER ranges from 0.42 to 0.45.

Fig. 4.14 shows the relationship between sigma values and BER. Although there is an increase in BER as sigma increases, the change is not significant, especially

at higher sigma ranges. This indicates that the watermarking method tested is fairly robust against speckle noise attacks since, despite the increase in noise intensity, the change in BER remains within acceptable limits.

Overall, the watermark is quite resistant to speckle noise attacks, with BER values remaining within acceptable ranges even as the noise intensity increases. This watermarking method shows that the distortion caused by speckle noise does not significantly impair the ability to extract the watermark.

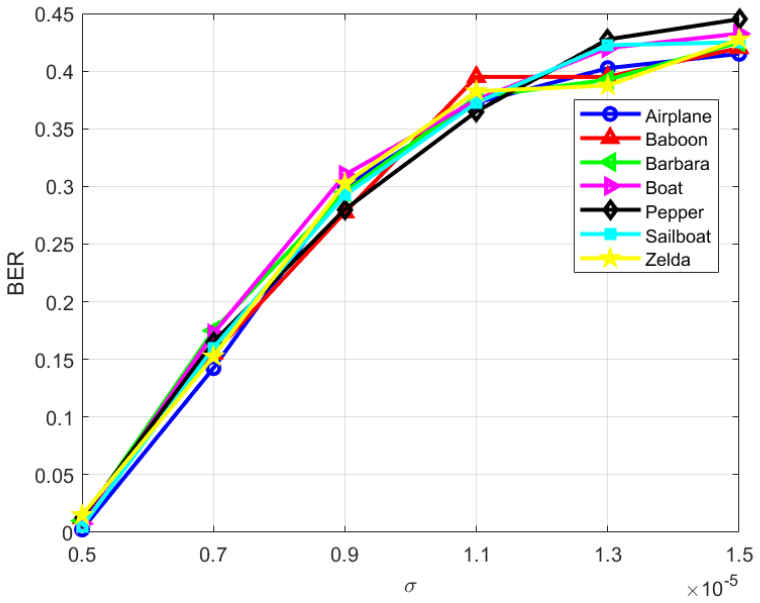


Fig 4.14 Watermark Robustness to Speckle on seven host Images

In this speckle noise attack, the research shows robustness to the attack. This is due to the use of the LSB method and Turtle Shell modification. In the LSB method, the speckle noise that is evenly spread across the image may damage some of the embedded watermark bits, but since the watermark is distributed in the least significant bits of many pixels, damage to a small portion of the pixels does not completely remove the watermark. The information still present in other pixels allows the watermark to remain extractable.

Meanwhile, in the Turtle Shell method, the watermark is stored based on specific coordinate patterns distributed throughout the image. Although the noise disturbs some points in the image, the coordinate patterns in the Turtle Shell remain largely unaffected due to the uniform noise distribution. This allows the watermark to be preserved, as other parts of the image still retain the watermark information properly.

4.4.7 Watermark Robustness to Median Filter

The median filter is an image processing technique aimed at reducing noise and improving image quality. Its main characteristic is the ability to replace each pixel value with the median of its neighboring pixels within a certain area. This filter is effective in removing noise but can distort the watermark embedded in the image. In Fig. 4.15, the effect of a median filter attack with a filtersize of 7 is shown, where the result of the attack makes the image appear smoother or blurred. This occurs because the filter replaces the pixel values containing the watermark with new values based on the median of the surrounding pixel intensities, making the watermark less visible or damaged.

Fig 4.15 Effect of Median Filter Attack (7)

Table 4.10 shows the results of testing the watermark's robustness against median filter attacks with various filter sizes. This filter replaces the pixel values with the median of the surrounding values, causing distortion to the watermark, especially if the watermark is hidden within small pixels or image details.

Table 4.10 Watermark Robustness to Median Filter Attacks

filtersize	**BER**						
	Airplane	**Baboon**	**Barbara**	**Boat**	**Peppers**	**Sailboat**	**Zelda**
3	0.46	0.43	0.46	0.47	0.45	0.47	0.46
5	0.47	0.46	0.45	0.47	0.46	0.47	0.45
7	0.47	0.46	0.45	0.47	0.46	0.47	0.45

From Table 4.10, it can be seen that the Bit Error Rate (BER) is quite high for all filter sizes tested, with values ranging from 0.43 to 0.47 for all test images. At a filter size of 3, BER ranges from 0.43 to 0.47. As the filter size increases to 5 and

7, the BER remains within the same range, indicating that increasing the filter size does not significantly reduce the BER.

The consistently high BER values for various filter sizes show that the watermarking method is not resistant to median filter attacks. Although this filter is designed to reduce noise, the results show that this technique also causes significant distortion to the watermark, making it difficult to extract correctly. This can be seen in Fig. 4.16, where the graph remains above 0.4.

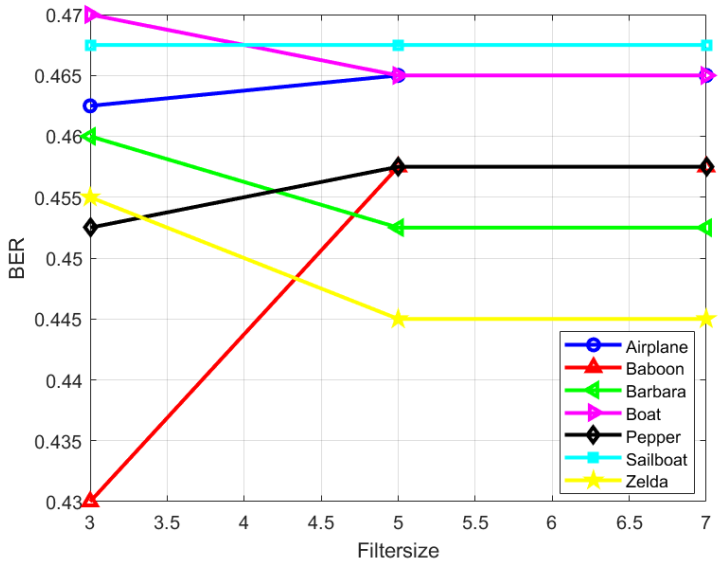


Fig 4.16 Watermark Robustness to Median Filter on Seven Host Image

In this median filter attack, the watermark embedded using the LSB and Turtle Shell methods is not robust against the attack. The LSB method works by embedding watermark information in the least significant bits of the image pixels. The median filter attack, which aims to remove noise, replaces pixel values with the median of neighboring pixels within a specific window. This process alters the pixel values containing the LSB watermark because the median filter replaces the pixels with new values based on the surrounding pixel intensities. As a result, the watermark hidden in the least significant bits is considered noise by the filter and is removed during the filtering process, causing the watermark to become damaged, less visible, or even disappear entirely.

In the Turtle Shell method, the median filter tends to disrupt the pattern or structure used to store the watermark. Turtle Shell relies on specific coordinates and patterns in the image. The filtering process replaces pixel values based on the median of surrounding values, which disrupts the structure and pattern of the image. As a result, the coordinates used by Turtle Shell are disturbed, making the water-

mark difficult to extract or completely lost.

4.5 Comparative Analysis of Watermark of Bit Length with BER

This test aims to observe the effect of watermark bit length on the BER value. The test uses two types of attacks, Gaussian noise and compression, with the Baboon image. Additionally, variations in watermark bit length provide a clearer understanding of the level of damage to the watermark in each type of attack.

4.5.1 Comparative Analysis of Watermark of Bit Length with BER in Gaussian Noise Attack

Fig. 4.17 shows the comparison between watermark bit length and Bit Error Rate (BER) under a Gaussian noise attack with a sigma value of 0.000001. The graph visualizes how variations in watermark bit length, ranging from 800 to 8000 bits, affect the BER.

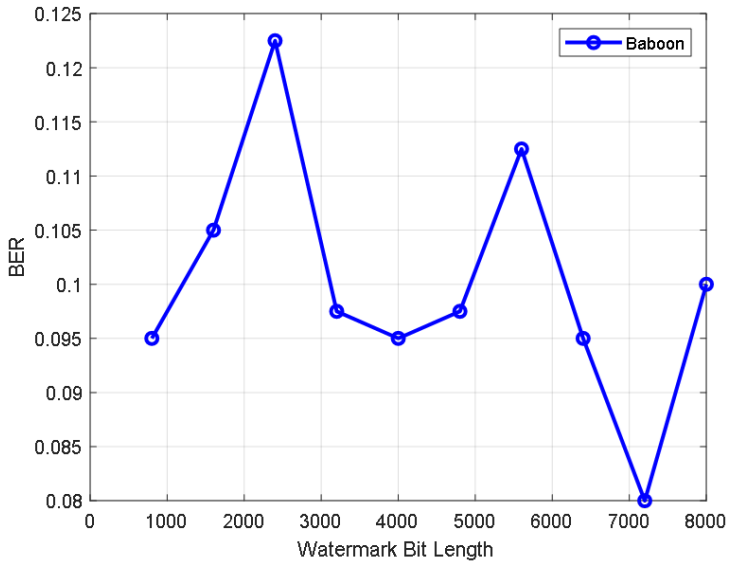


Fig 4.17 Comparison of Watermark Bit Length and BER in Gaussian Noise Attack

From the graph, it can be seen that as the watermark bit length increases, the BER generally tends to decrease, although there are fluctuations at certain points. With shorter watermark bit lengths, around 800 to 2000 bits, the BER is higher. However, as the watermark bit length increases, especially around 8000 bits, the

BER reaches its lowest point. Despite this, the graph shows some rises and falls in the BER at certain intervals, indicating that the decrease is not completely linear.

The decrease in BER with longer watermark bit lengths can be explained by the increasing number of pixels modified when embedding the watermark. The larger the number of bits embedded, the wider the distribution of the watermark across the image, which provides better resistance against the Gaussian noise attack. With a larger distribution of watermark bits, damage to a small portion of pixels due to Gaussian noise will not affect the extraction of the watermark as a whole.

4.5.2 Comparative Analysis of Watermark of Bit Length with BER in Compression Attack

Fig. 4.18 shows a comparison between watermark bit length and Bit Error Rate (BER) in a compression attack with a quality level of 100. This graph visualizes how the variation in watermark bit length, ranging from 800 to 8000 bits, affects BER when the image is subjected to compression attacks.

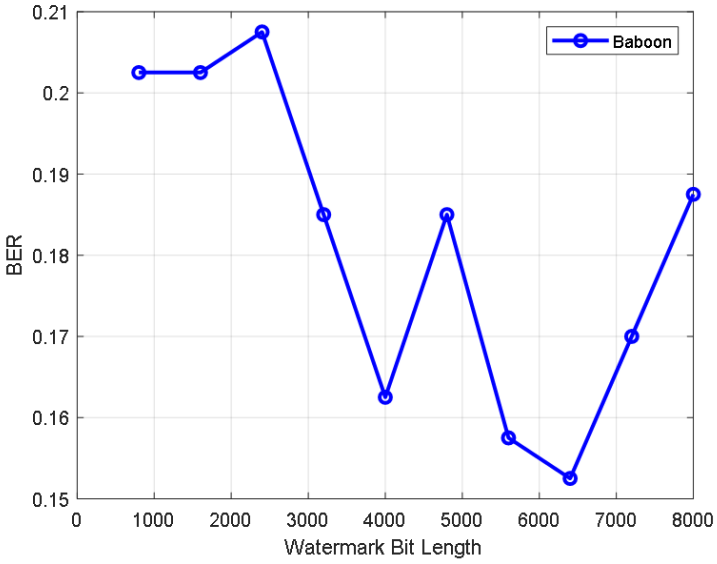


Fig 4.18 Comparison of Watermark Bit Length and BER in Compression Attack

From the graph, it can be observed that BER values tend to fluctuate significantly across different watermark bit lengths. For shorter watermark bit lengths, around 1000 to 2000 bits, the BER remains in the range of 0.20, indicating a relatively high error rate. However, as the watermark bit length approaches 4000 bits, the BER drops dramatically, reaching its lowest value around 0.16. Afterward, the BER rises again for longer watermark bit lengths, peaking at around 8000 bits with

a BER close to 0.20.

The fluctuation in BER indicates that while a longer watermark may provide broader distribution within the image, this method does not always ensure consistent resistance to compression attacks. At certain watermark bit lengths, such as around 4000 bits, compression may not significantly distort the watermark, resulting in a lower BER. However, at either longer or shorter watermark bit lengths, compression may cause greater distortion, leading to higher BER.

This graph suggests that the watermark bit length has a fluctuating impact on BER in compression attacks, with some watermark lengths showing better resilience while others are more prone to distortion caused by compression.

4.6 Comparison of Attacks in Previous Research with the Proposed Method

This section explains the comparison of attacks used in the proposed method with various types of attacks in previous research. The attacks include Gaussian Noise, Salt & Pepper Noise, compression, Low Pass Filter, Rescaling, Speckle Noise, Median Filter, and Poison. Table 4.11 summarizes the attacks tested by Li et al. [11] and the proposed method. A checkmark (✓) indicates that the method is tested against the corresponding attack, while a dash (-) indicates that the attack is not used in the testing.

Table 4.11 Types of Attacks Applied in Previous Research and the Proposed Method

Attacks	**Li et al [11]**	**Proposed**
Gaussian Noise	✓	✓
Salt & Pepper Noise	✓	✓
Compression	-	✓
Low Pass Filter	-	✓
Rescaling	-	✓
Speckle Noise	✓	✓
Median Filter	-	✓
Poison	✓	-

Table 4.11 shows the attacks used in previous research and the proposed research. In the previous research by Li et al. [11], the attacks used include Gaussian Noise, Salt & Pepper Noise, Speckle Noise, and Poisson. In the proposed research, the attacks used are Gaussian Noise, Salt & Pepper Noise, Compression, Low Pass Filter, Rescaling, Speckle Noise, and Median Filter.

In the previous research, the Bit Error Rate (BER) is not explained in the attack testing, so the information regarding watermarking robustness against attacks is incomplete. The previous research only explains the Detection Rate (DR), which relates more to shadow detection and does not directly relate to watermarking robustness. However, the previous research states that the method is robust against Gaussian Noise, Salt & Pepper Noise, Speckle Noise, and Poisson attacks. Meanwhile, the proposed research presents BER values, providing a clearer explanation of watermarking robustness against attacks. In the proposed research, watermarking proves to be resistant to Gaussian Noise, Salt & Pepper Noise, and Speckle Noise attacks, but less resistant to filter attacks like Low Pass Filter and Median Filter. Gaussian Noise, Salt & Pepper, and Speckle are more robust across various methods compared to filter attacks. Therefore, it is better to use BER for attack testing.

BER measures the percentage of bit errors that occur when the watermark is received compared to when the watermark is sent, thus providing an evaluation of how well the watermark withstands attacks. The use of BER provides a more accurate and objective assessment of watermarking robustness.

4.7 Robustness of Watermarking at Different Attack Methods Under Various Methods

Attacks on image watermarking aim to test the robustness of the watermarking methods against various types of attacks. Each research method uses specific attacks to evaluate the watermarking's robustness. Table 4.12 combines the results from different studies [11], [27], [28], [29], [30], [31], [32], and the proposed method. This summary focuses on the attacks relevant to the proposed method since different studies often use various attack types. This approach allows for a more comprehensive comparison between the proposed method and existing ones based on their robustness against the same attacks.

Table 4.12 BER for Different Attacks

Attacks	BER							
	[11]	**[27]**	**[28]**	**[29]**	**[30]**	**[31]**	**[32]**	**prop**
Gaussian	-	-	-	-	-	-	-	0.08
Salt & Pepper	-	0.051	-	-	-	-	-	0.003
Compression	-	-	-	-	-	0.22	0.1	0.13
Low Pass Filter	-	-	-	-	-	-	-	0.45
Rescaling	-	-	-	-	-	0.24	-	0.43
Speckle	-	-	-	-	-	-	-	0.003
Median Filter	-	0.034	-	-	-	0.17-	-	0.45
Poison	-	-	-	-	-	-	-	-

Table 4.12 presents BER data for different methods under various attacks. In previous studies [11], [27], [28], [29], [30], [31], [32], most research does not provide complete BER data for all types of attacks. Only a few studies include BER information, such as for Salt & Pepper Noise attacks, where BER is reported as 0.051 [27]; Compression, where BER is reported as 0.22 [30] and 0.1 [32]; Rescaling, where BER is reported as 0.24 [30]; or for Median Filter attacks, where BER is reported as 0.034 [27] and 0.17 [30]

In watermarking robustness testing, BER is an important parameter to objectively assess whether the watermarking can withstand attacks. Previous methods focus more on general image watermarking attacks rather than the turtle shell approach. Based on the data in the table, attacks like Gaussian Noise, Salt and Pepper Noise, Speckle Noise, and Compression are commonly used to test the robustness of image watermarking. For the proposed method, the BER for Gaussian Noise is 0.08, for Salt and Pepper Noise is 0.003,Compression is 0.13, Low Pass Filter is 0.45, Rescaling is 0.43, Speckle Noise is 0.003, and for Median filter is 0.45, indicating that the proposed method has varying levels of robustness against different types of attacks.

However, the proposed method shows weaknesses against certain attacks. Specifically, the Low Pass Filter, Rescaling, and Median Filter attacks result in high BER values of 0.45, 0.43, and 0.45, respectively, indicating that the applied watermark is more susceptible to damage or interference from these attacks. These results suggest that while the proposed method is robust against some types of attacks, there are still vulnerabilities that need to be addressed, particularly with filter-based attacks.

In the Median Filter attack, the proposed method was not robust, as indicated by the high BER values. In contrast, in studies [27] and [31], the BER values for the

Median Filter attack were relatively low. This was because both studies did not use the LSB method. Study [27] used the DWT and SVD methods, while study [31] employed the SVD and IWT methods. The LSB method was more vulnerable to attacks in the frequency domain, making it less resistant to attacks like the Median Filter

CHAPTER V
CONCLUSIONS AND FUTURE WORKS

This chapter provides the conclusion and notifies the future works of this thesis.

5.1 Conclusions

This research develops an information hiding technique using the modified turtle-shell method. The study aims to improve the quality of watermark imperceptibility, measured by Mean Opinion Score (MOS) and Peak Signal-to-Noise Ratio (PSNR), while also evaluating the robustness of the watermark against various types of attacks.

Imperceptibility is assessed by PSNR and MOS values. The average PSNR value of 51.24 dB indicates excellent visual quality, with the inserted watermark causing no visible disturbance to the image. The average MOS value of 4.48 reflects the subjective opinions of users, where most users perceive the watermark as sufficiently hidden, although some users can still detect it. These high PSNR and MOS values demonstrate the effectiveness of the watermarking method in maintaining imperceptibility.

Additionally, the research evaluates the robustness of the watermark against several types of attacks, including Gaussian noise, Salt and Pepper noise, compression, low-pass filter, rescaling, Speckle noise, and median filter. The system exhibits robustness against Gaussian noise, Salt and Pepper noise, and Speckle noise attacks. However, the robustness is lower when subjected to low-pass filter, rescaling, and median filter attacks. The more watermarks embedded in the image, the more duplication occurs within the host image, which ultimately enhances robustness against attacks. In particular, increasing the number of identical watermark duplications embedded in the image strengthens its robustness against attacks.

One of the main strengths of the proposed method is its high PSNR value, indicating excellent visual quality and imperceptibility. The system also demonstrates strong robustness against common noise-based attacks such as Gaussian noise, Salt and Pepper noise, and Speckle noise, making it suitable for maintaining image quality under certain attack conditions. However, despite its strong performance in some areas, the proposed method is less robust against frequency domain attacks, such as low-pass filtering, rescaling, and median filter attacks. This suggests that the

method struggles with attacks targeting high-frequency components, indicating the need for further improvements to enhance robustness against these types of attacks.

5.2 Future Works

In this watermarking research, there are several directions for further research, namely:

1. The use of other insertion methods besides LSB, such as Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), Singular Value Decomposition (SVD), or other methods, while still using the Turtle Shell algorithm, can be considered.

2. Programming improvements are made so that watermarking becomes more resistant to attacks. Although the programming still uses the Turtle Shell algorithm, the proposed algorithm is expected to be more robust and able to deal with various types of attacks more effectively.

REFERENCES

[1] W.-W. Hu, R.-G. Zhou, A. El-Rafei, and S.-X. Jiang, "Quantum image watermarking algorithm based on haar wavelet transform," *IEEE Access*, vol. 7, pp. 121 303–121 320, 2019.

[2] S. Sowmya, S. Karanth, and S. Kumar, "Protection of data using image watermarking technique," *Global Transitions Proceedings*, vol. 2, no. 2, pp. 386–391, 2021.

[3] R. Singh, M. Saraswat, A. Ashok, H. Mittal, A. Tripathi, A. C. Pandey, and R. Pal, "From classical to soft computing based watermarking techniques: A comprehensive review," *Future Generation Computer Systems*, vol. 141, pp. 738–754, 2023.

[4] M. W. Hatoum, J.-F. Couchot, R. Couturier, and R. Darazi, "Using deep learning for image watermarking attack," *Signal Processing: Image Communication*, vol. 90, p. 116019, 2021.

[5] C. C. Chang, Y. Liu, and T. S. Nguyen, "A novel turtle shell based scheme for data hiding," in *2014 tenth international conference on intelligent information hiding and multimedia signal processing*. IEEE, 2014, pp. 89–93.

[6] Y. Liu, C.-C. Chang, and T.-S. Nguyen, "High capacity turtle shell-based data hiding," *IET Image Processing*, vol. 10, no. 2, pp. 130–137, 2016.

[7] J.-Y. Lin, Y. Liu, and C.-C. Chang, "A real-time dual-image-based reversible data hiding scheme using turtle shells," *Journal of Real-Time Image Processing*, vol. 16, pp. 673–684, 2019.

[8] C.-C. Lin, S.-L. He, and C.-C. Chang, "Pixel-based fragile image watermarking based on absolute moment block truncation coding," *Multimedia Tools and Applications*, vol. 80, no. 19, pp. 29 497–29 518, 2021.

[9] C.-C. Chang and Y. Liu, "Fast turtle shell-based data embedding mechanisms with good visual quality," *Journal of Real-Time Image Processing*, vol. 16, pp. 589–599, 2019.

[10] X.-Z. Xie, C.-C. Chang, C.-C. Lin, and J.-L. Lin, "A turtle shell based rdh scheme with two-dimensional histogram shifting," *Multimedia Tools and Applications*, vol. 78, pp. 19 413–19 436, 2019.

[11] X.-S. Li, C.-C. Chang, M.-X. He, and C.-C. Lin, "A lightweight authenticable visual secret sharing scheme based on turtle shell structure matrix," *Multimedia Tools and Applications*, vol. 79, no. 1, pp. 453–476, 2020.

[12] S. Boujerfaoui, R. Riad, H. Douzi, F. Ros, and R. Harba, "Image watermarking between conventional and learning-based techniques: A literature review," *Electronics*, vol. 12, no. 1, 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/1/74

[13] S. Gani and B. Setiyono, "Teknik invisible watermarking digital menggunakan metode dwt (discrete wavelet tarnsform)," *Jurnal Sains dan Seni ITS*, vol. 7, no. 2, pp. 24–30, 2019.

[14] R. Kusumanto and A. N. Tompunu, "pengolahan citra digital untuk mendeteksi obyek menggunakan pengolahan warna model normalisasi rgb," in *Seminar Nasional Teknologi Informasi & Komunikasi Terapan*, vol. 2011, 2011, pp. 1–7.

[15] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed. Prentice Hall, 2018.

[16] A. Dixit and R. Dixit, "A review on digital image watermarking techniques," *International Journal of Image, Graphics and Signal Processing*, vol. 9, no. 4, p. 56, 2017.

[17] J. Xuehua, "Digital watermarking and its application in image copyright protection," in *2010 International Conference on Intelligent Computation Technology and Automation*, vol. 2, 2010, pp. 114–117.

[18] A. Dixit and R. Dixit, "A review on digital image watermarking techniques," *International Journal of Image, Graphics and Signal Processing*, vol. 9, no. 4, p. 56, 2017.

[19] B. E. Baaquie and L.-C. Kwek, "Quantum computers: Theory and algorithms," 2023.

[20] M.-X. Wang, H.-M. Yang, D.-H. Jiang, B. Yan, J.-S. Pan, and T. Liu, "A novel quantum color image steganography algorithm based on turtle shell and lsb," *Quantum Information Processing*, vol. 21, no. 4, p. 148, 2022.

[21] J.-L. Yao, H.-M. Yang, D.-H. Jiang, B. Yan, J.-S. Pan, and M.-X. Wang, "A novel quantum image steganography algorithm based on double-layer gray code," *International Journal of Theoretical Physics*, vol. 62, no. 3, p. 52, 2023.

[22] A. Purbaningrum, K. S. Amalia, and I. A. Saputro, "Penerapan metode least significant bit (lsb) dalam menyisipkan pesan rahasia pada citra digital: Sebuah pendekatan steganografi," in *Seminar Nasional AMIKOM Surakarta (SEMNASA)*. AMIKOM Surakarta, 2023, pp. 176–183.

[23] R. A. Mohammed, "An improvement of rgb color image watermarking technique using isb stream bit and hadamard matrix," Master's thesis, Universiti Teknologi Malaysia, 2014.

[24] D. Darisman, P. Sokibi, and M. Asfi, "Aplikasi steganografi untuk penyembunyian data ke dalam citra digital dengan kombinasi metode least significant bit (lsb) dan algoritma vigenere cipher," *Jurnal Digit*, vol. 4, no. 2, pp. 240–257, 2014.

[25] C.-C. Chang and Y. Liu, "Fast turtle shell-based data embedding mechanisms with good visual quality," *Journal of Real-Time Image Processing*, vol. 16, pp. 589–599, 2019.

[26] E. Nugraheni and N. H. Lestriandoko, "Uji ketahanan metode block-base watermarking pada domain wavelet terhadap serangan gaussian blur dan random noise," in *Proceedings of P2 Informatika - LIPI*, Jl. Cisitu, Sangkuriang, Bandung 40135, 2024.

[27] M. W. Hatoum, J.-F. Couchot, R. Couturier, and R. Darazi, "Using deep learning for image watermarking attack," *Signal Processing: Image Communication*, vol. 90, p. 116019, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0923596520301715

[28] R. Rajkumar and A. Vasuki, "Reversible and robust image watermarking based on histogram shifting," *Cluster Computing*, vol. 22, pp. 12 313–12 323, 2019.

[29] L. Lidyawati, A. R. Darlis, L. Jambola, and T. H. Yanuary, "Performance analysis image watermarking using discrete cosine transforms," in *2019 IEEE International Conference on Signals and Systems (ICSigSys)*, 2019, pp. 50–55.

[30] M. Nazari and M. Mehrabian, "A novel chaotic iwt-lsb blind watermarking approach with flexible capacity for secure transmission of authenticated med-

ical images," *Multimedia Tools and Applications*, vol. 80, pp. 10 615–10 655, 2021.

[31] T. Zhu, W. Qu, and W. Cao, "An optimized image watermarking algorithm based on svd and iwt," *The Journal of Supercomputing*, vol. 78, pp. 222–237, 2022.

[32] N. Tarhouni, M. Charfeddine, and C. Ben Amar, "Novel and robust image watermarking for copyright protection and integrity control," *Circuits, Systems, and Signal Processing*, vol. 39, pp. 5059–5103, 2020.

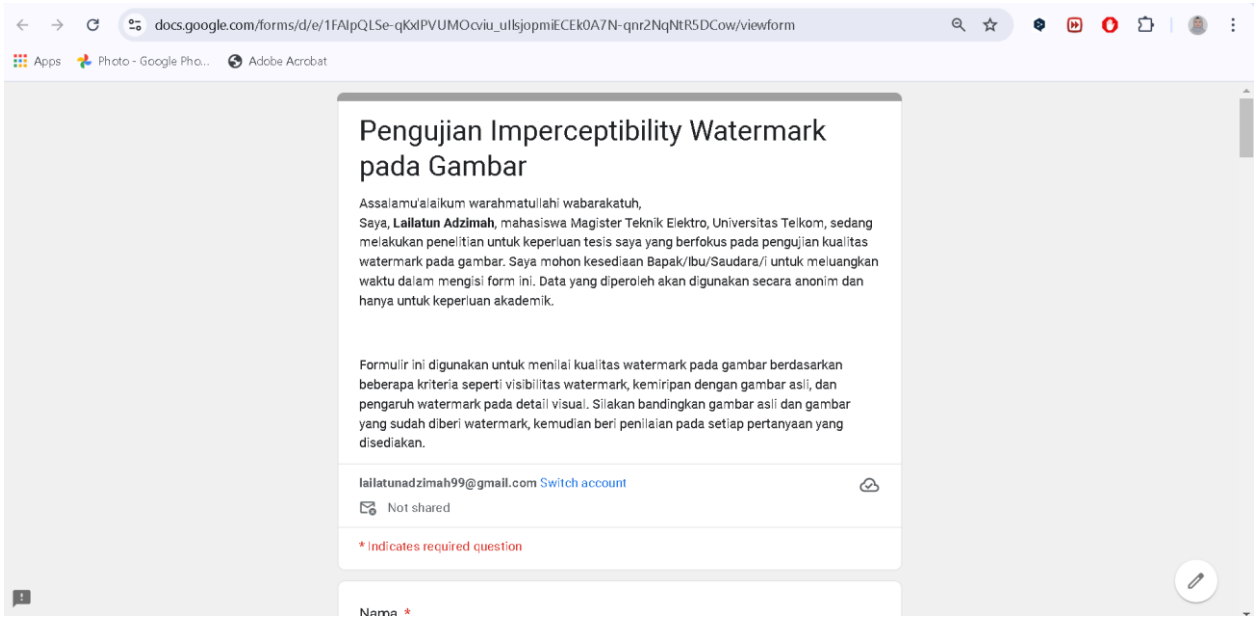
[33] P. Singh, A. Agarwal, and J. Gupta, "Image watermark attacks: Classification & implementation," *IJECT*, vol. 4, no. 2, pp. 1–, 2013.

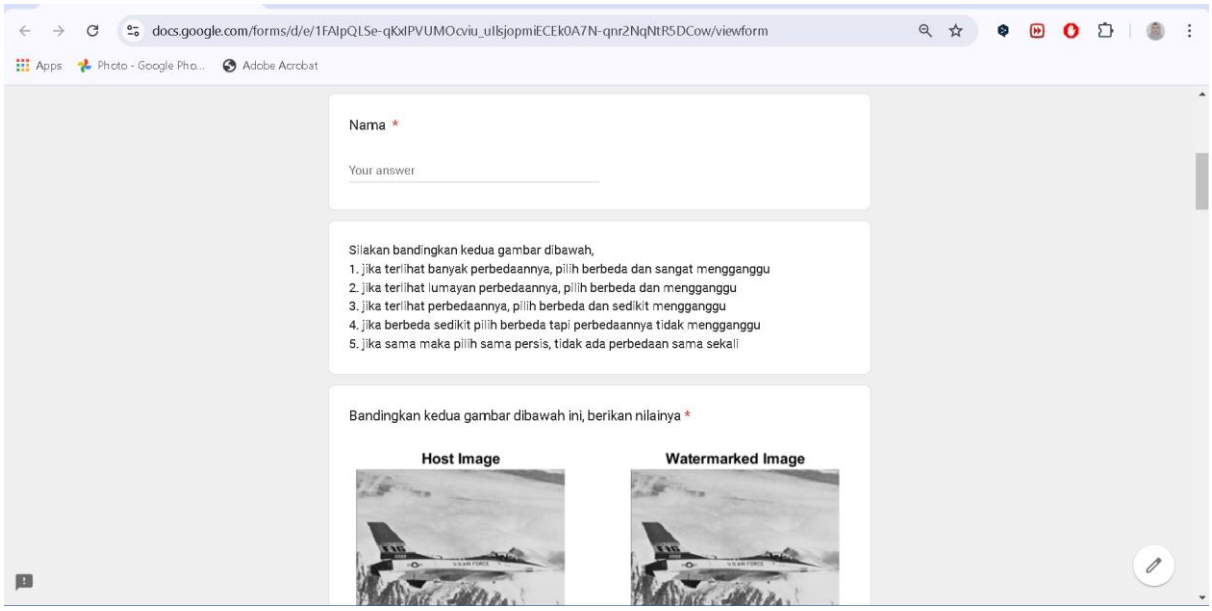
[34] R. Streijl, S. Winkler, and D. Hands, "Mean opinion score (mos) revisited: methods and applications, limitations and alternatives," *Multimedia Systems*, vol. 22, no. 2, pp. 213–227, 2016.

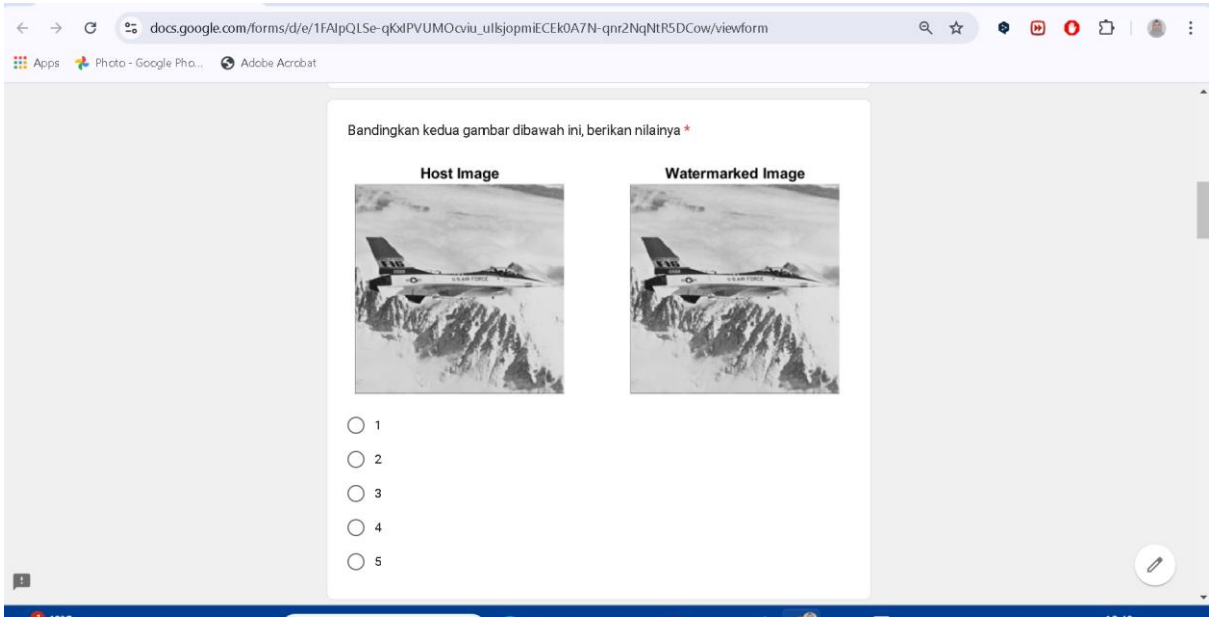
[35] H.-N. Huang, S.-T. Chen, M.-S. Lin, and W.-M. Kung, "Optimization-based embedding for wavelet-domain audio watermarking," *Journal of Signal Processing Systems*, vol. 80, 08 2013.

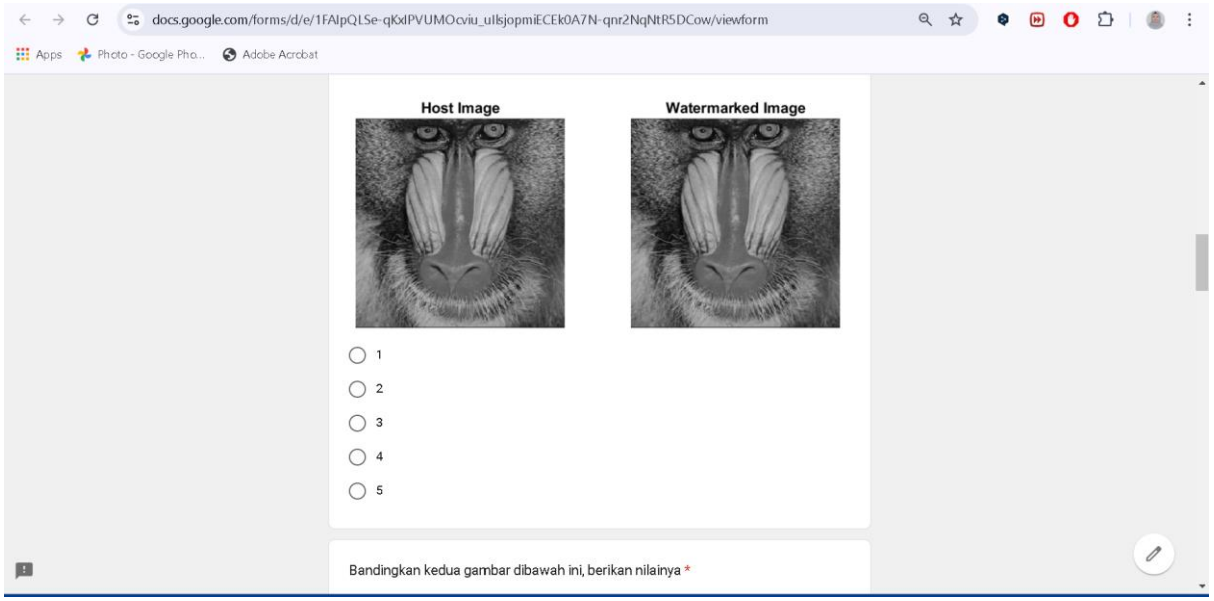
[36] M. Jia-Fa, Z. Ru, N. Xin-Xin *et al.*, "Research of spatial domain image digital watermarking payload," *EURASIP Journal on Information Security*, vol. 2011, p. 502748, 2011.

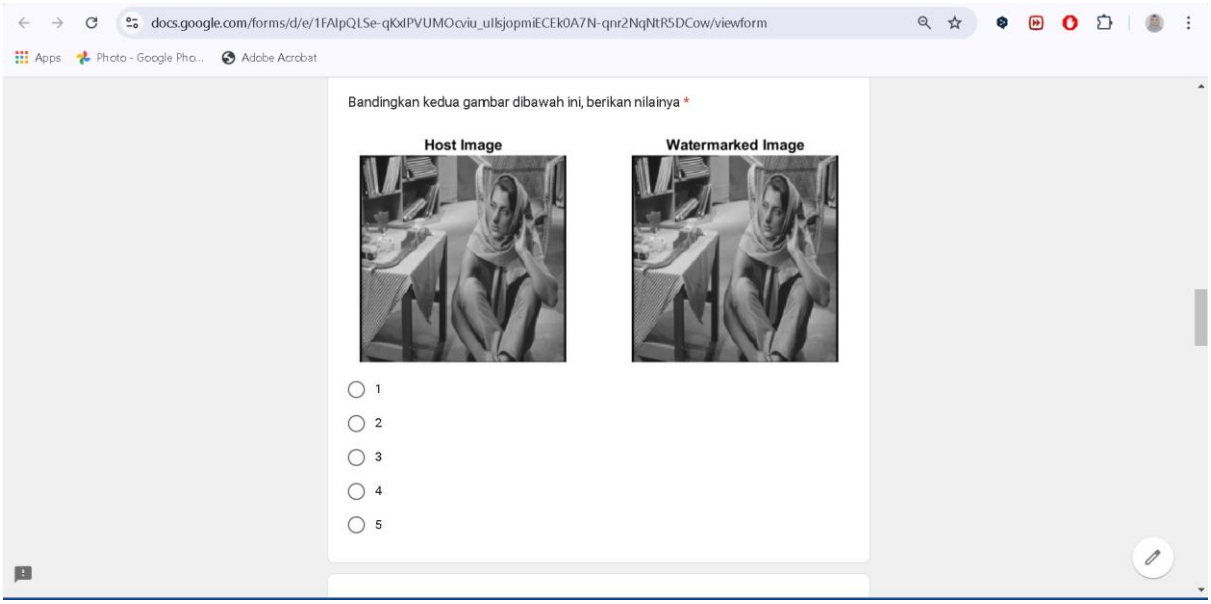
Appendices

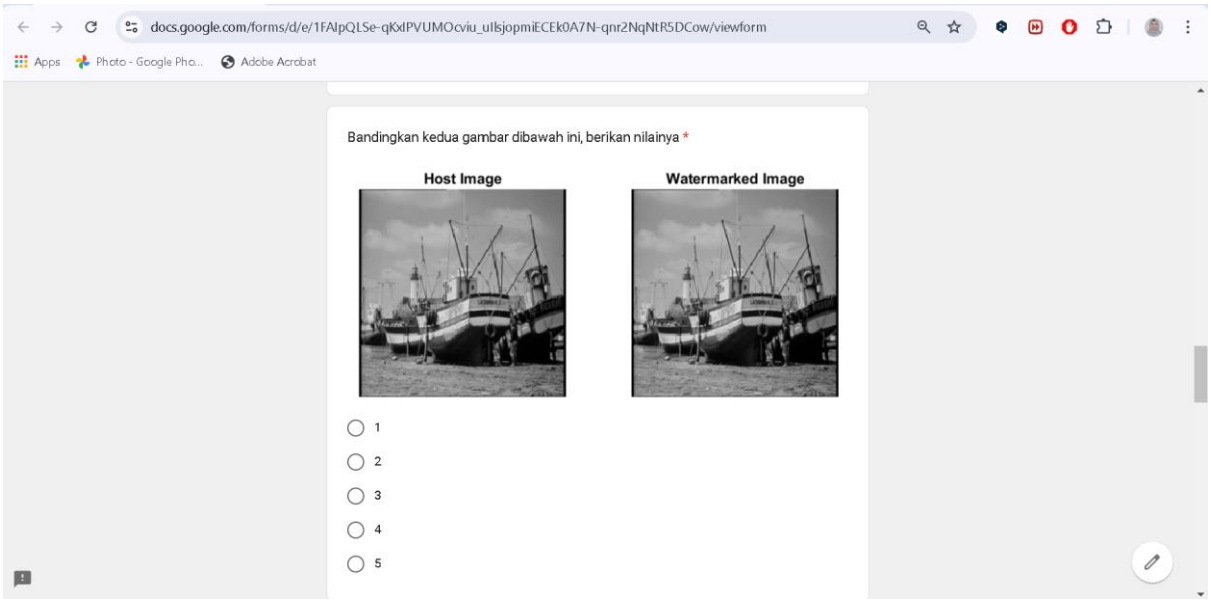

docs.google.com/forms/d/e/1FAIpQLSe-qKxlPVUMOcviu_uIlsjopmiECEk0A7N-qnr2NqNtR5DCow/viewform
Apps
Photo - Google Pho...
Adobe Acrobat
Pengujian Imperceptibility Watermark pada Gambar
Assalamu'alaikum warahmatullahi wabarakatuh,
Saya, Lailatun Adzimah, mahasiswa Magister Teknik Elektro, Universitas Telkom, sedang melakukan penelitian untuk keperluan tesis saya yang berfokus pada pengujian kualitas watermark pada gambar. Saya mohon kesediaan Bapak/Ibu/Saudara/i untuk meluangkan waktu dalam mengisi form ini. Data yang diperoleh akan digunakan secara anonim dan hanya untuk keperluan akademik.
Formulir ini digunakan untuk menilai kualitas watermark pada gambar berdasarkan beberapa kriteria seperti visibilitas watermark, kemiripan dengan gambar asli, dan pengaruh watermark pada detail visual. Silakan bandingkan gambar asli dan gambar yang sudah diberi watermark, kemudian beri penilaian pada setiap pertanyaan yang disediakan.
lailatunadzimah99@gmail.com Switch account
Not shared
* indicates required question
Nama *

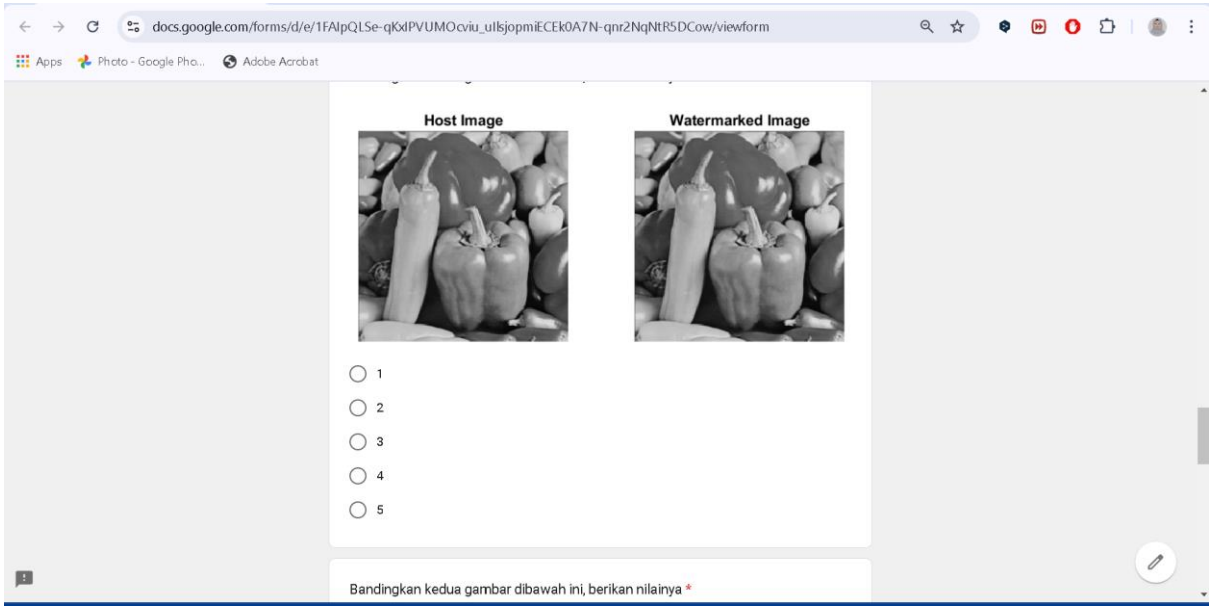

docs.google.com/forms/d/e/1FAIpQLSe-qKxlPVUMOcviu_uIlsjopmiECEk0A7N-qnr2NqNtR5DCow/viewform
Apps
Photo - Google Pho...
Adobe Acrobat
Nama *
Your answer
Silakan bandingkan kedua gambar dibawah,
1. jika terlihat banyak perbedaannya, pilih berbeda dan sangat mengganggu
2. jika terlihat lumayan perbedaannya, pilih berbeda dan mengganggu
3. jika terlihat perbedaannya, pilih berbeda dan sedikit mengganggu
4. jika berbeda sedikit pilih berbeda tapi perbedaannya tidak mengganggu
5. jika sama maka pilih sama persis, tidak ada perbedaan sama sekali
Bandingkan kedua gambar dibawah ini, berikan nilainya *
Host Image
Watermarked Image

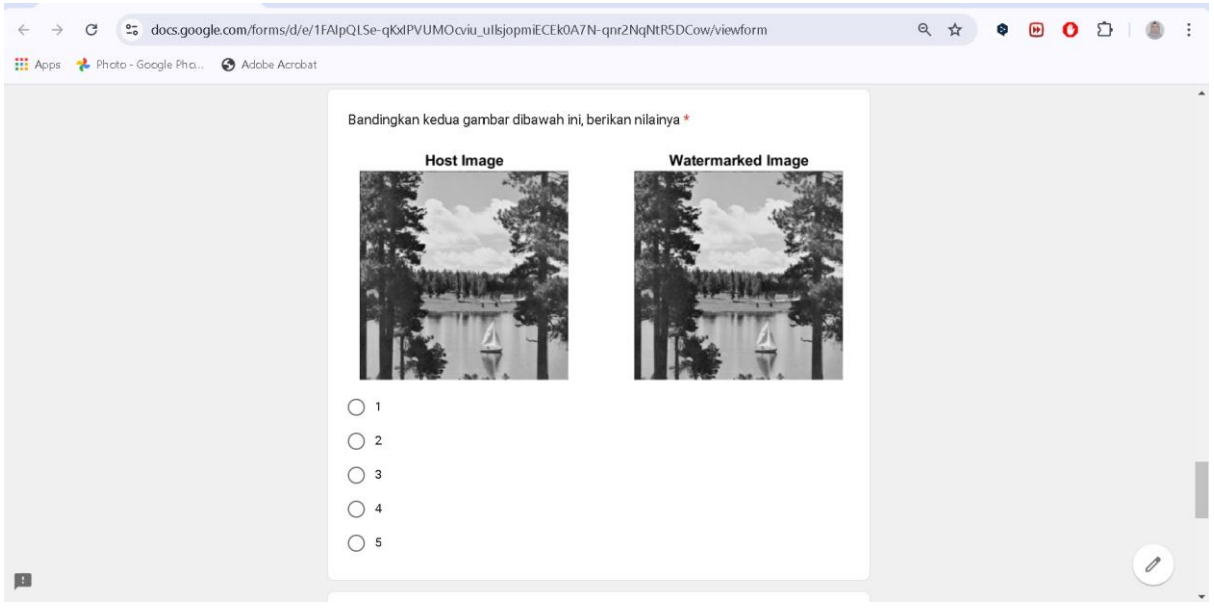

docs.google.com/forms/d/e/1FAIpQLSe-qKxlPVUMOcviu_uIlsjopmiECEk0A7N-qnr2NqNtR5DCow/viewform
Apps
Photo - Google Pho...
Adobe Acrobat
Bandingkan kedua gambar dibawah ini, berikan nilainya *
Host Image
Watermarked Image
1
2
3
4
5

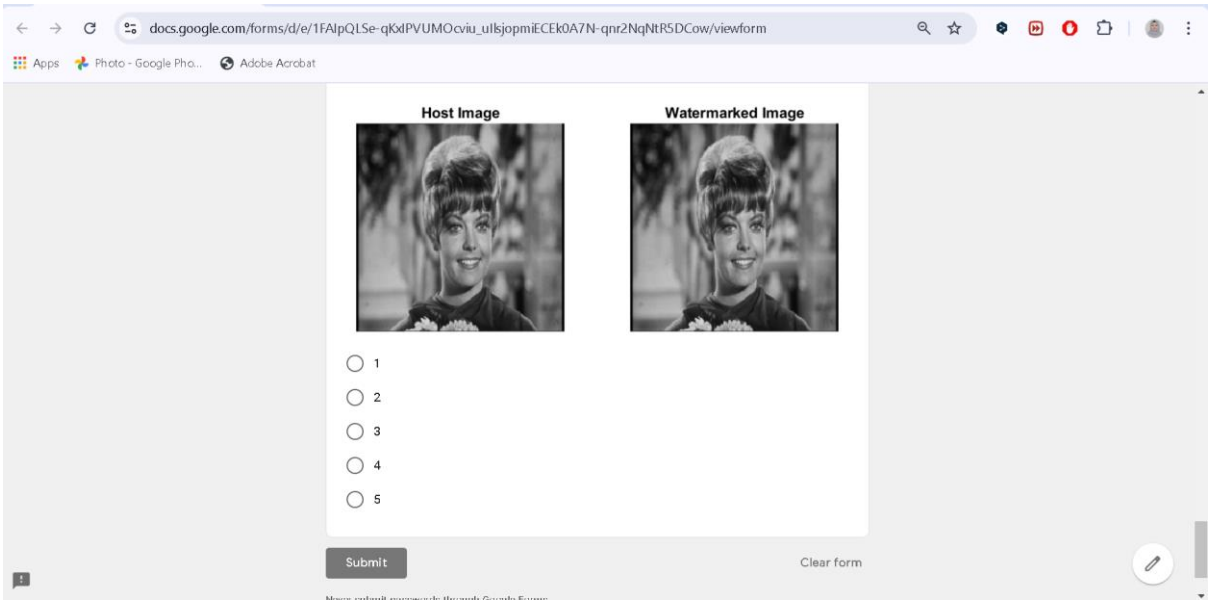

docs.google.com/forms/d/e/1FAIpQLSe-qKxlPVUMOcviu_uIlsjopmiECEk0A7N-qnr2NqNtR5DCow/viewform
Apps
Photo - Google Pho...
Adobe Acrobat
Host Image
Watermarked Image
1
2
3
4
5
Bandingkan kedua gambar dibawah ini, berikan nilainya *

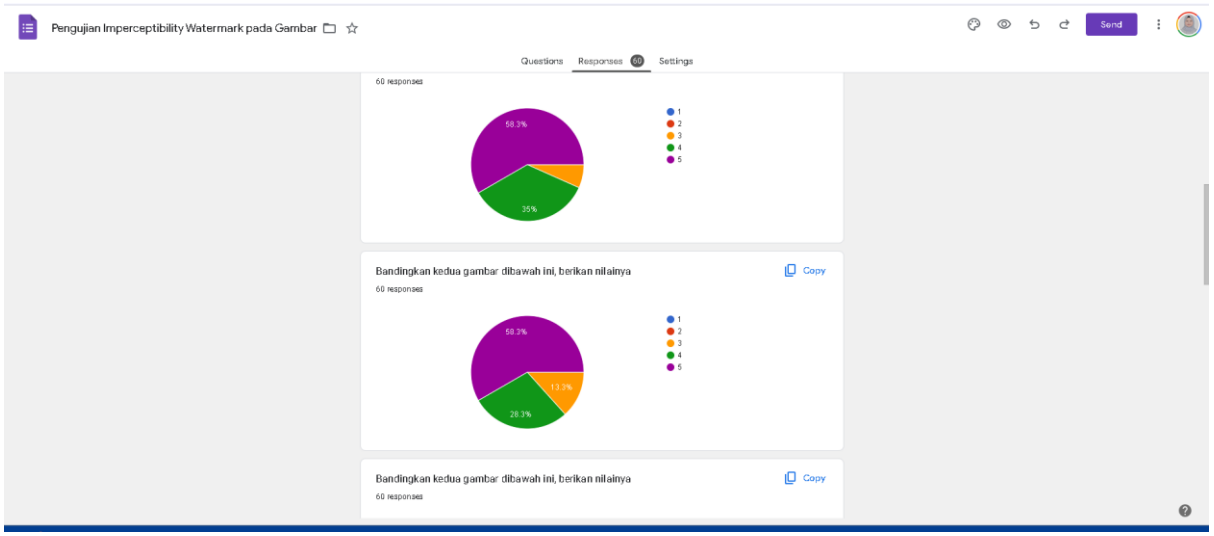

docs.google.com/forms/d/e/1FAIpQLSe-qKxlPVUMOcviu_uIlsjopmiECEk0A7N-qnr2NqNtR5DCow/viewform
Apps
Photo - Google Pho...
Adobe Acrobat
Bandingkan kedua gambar dibawah ini, berikan nilainya *
Host Image
Watermarked Image
1
2
3
4
5

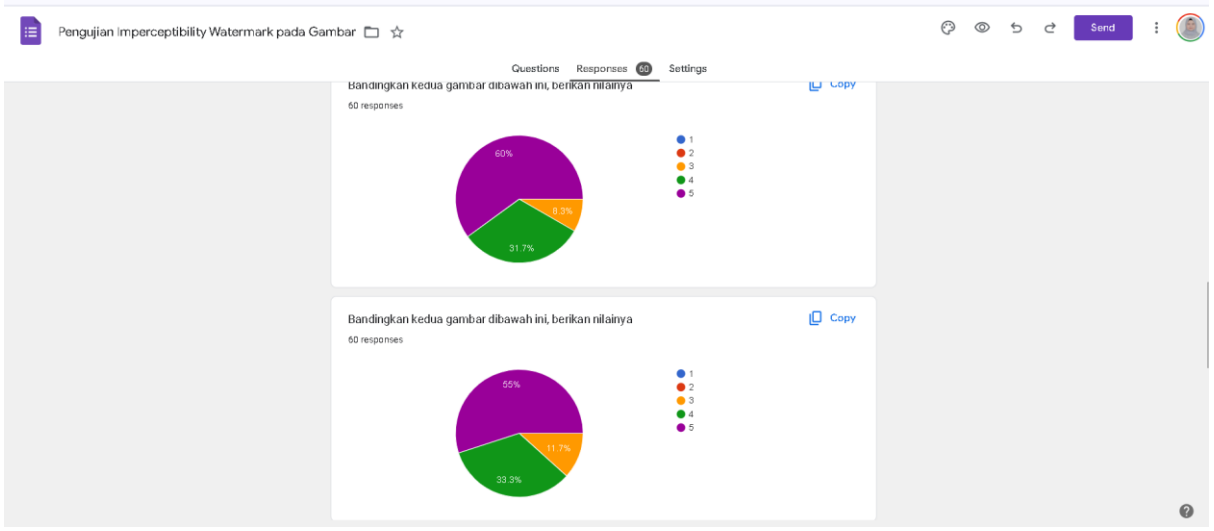

docs.google.com/forms/d/e/1FAIpQLSe-qKxlPVUMOcviu_uIlsjopmiECEk0A7N-qnr2NqNtR5DCow/viewform
Apps
Photo - Google Pho...
Adobe Acrobat
Bandingkan kedua gambar dibawah ini, berikan nilainya *
Host Image
Watermarked Image
1
2
3
4
5

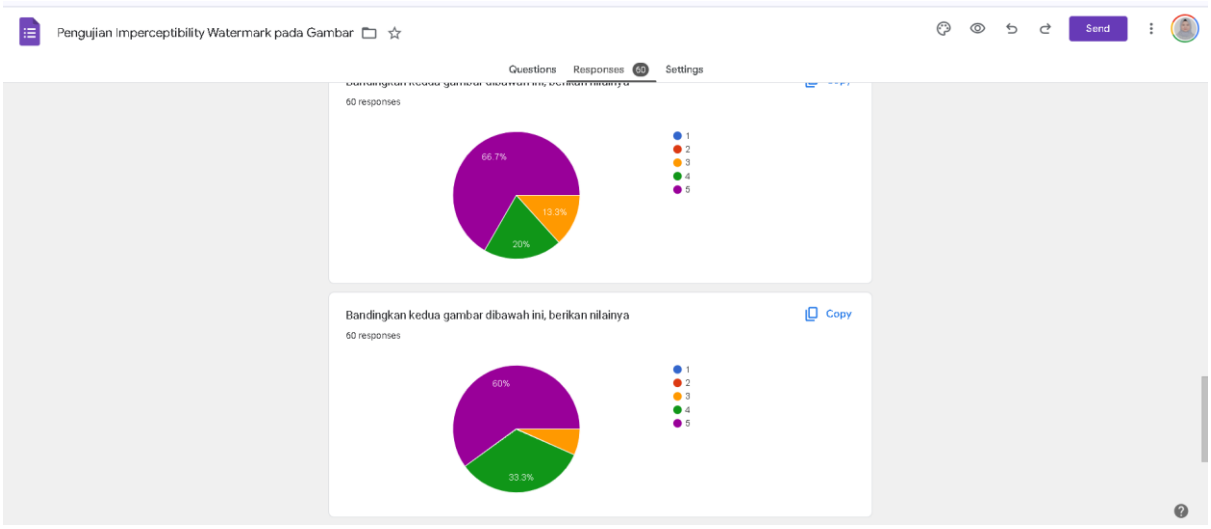

docs.google.com/forms/d/e/1FAIpQLSe-qKxlPVUMOcviu_uIlsjopmiECEk0A7N-qnr2NqNtR5DCow/viewform
Apps
Photo - Google Pho...
Adobe Acrobat
Host Image
Watermarked Image
1
2
3
4
5
Bandingkan kedua gambar dibawah ini, berikan nilainya *

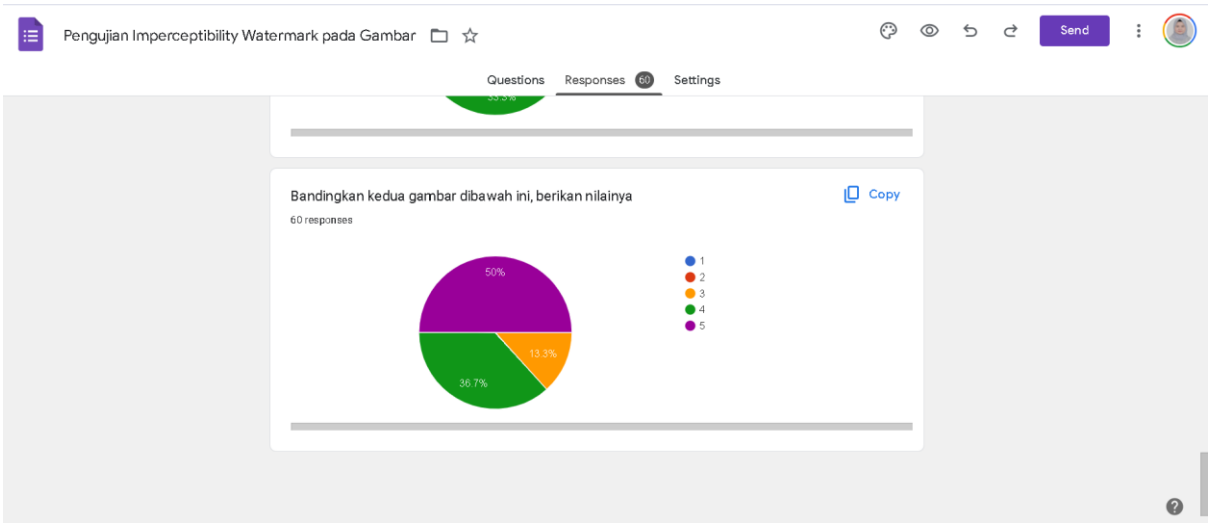

docs.google.com/forms/d/e/1FAIpQLSe-qKxlPVUMOcviu_uIlsjopmiECEk0A7N-qnr2NqNtR5DCow/viewform
Apps
Photo - Google Pho...
Adobe Acrobat
Bandingkan kedua gambar dibawah ini, berikan nilainya *
Host Image
Watermarked Image
1
2
3
4
5


docs.google.com/forms/d/e/1FAIpQLSe-qKxlPVUMOcviu_uIlsjopmiECEk0A7N-qnr2NqNtR5DCow/viewform
Apps
Photo - Google Pho...
Adobe Acrobat
Host Image
Watermarked Image
1
2
3
4
5
Submit
Clear form


Pengujian Imperceptibility Watermark pada Gambar
Send
Questions
Responses 60
Settings
60 responses
58.3%
35%
1
2
3
4
5
Bandingkan kedua gambar dibawah ini, berikan nilainya
Copy
60 responses
58.3%
13.3%
28.3%
1
2
3
4
5
Bandingkan kedua gambar dibawah ini, berikan nilainya
Copy
60 responses


Pengujian Imperceptibility Watermark pada Gambar
Send
Questions
Responses 60
Settings
Bandingkan kedua gambar dibawah ini, berikan nilainya
Copy
60 responses
60%
8.3%
31.7%
1
2
3
4
5
Bandingkan kedua gambar dibawah ini, berikan nilainya
Copy
60 responses
55%
11.7%
33.3%
1
2
3
4
5


Pengujian Imperceptibility Watermark pada Gambar
Send
Questions
Responses 60
Settings
60 responses
66.7%
13.3%
20%
1
2
3
4
5
Bandingkan kedua gambar dibawah ini, berikan nilainya
Copy
60 responses
60%
33.3%
1
2
3
4
5


Pengujian Imperceptibility Watermark pada Gambar
Send
Questions
Responses 60
Settings
Bandingkan kedua gambar dibawah ini, berikan nilainya
Copy
60 responses
50%
13.3%
36.7%
1
2
3
4
5