People’s Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
KASDI MERBAH UNIVERSITY - OUARGLA
Faculty of New Technologies of Information and Telecommunication
Department of Computer Science and Information Technology

MASTER

Domain : Computer Science
Speciality : Fondemental Computer Science
Field : Image Processing and Information Security
Submitted by : **Ben Habirech Mohammed Lamine** and **Kerichi Rokia**

Thesis :

Advancements in Secure Watermarking Techniques for Medical Image Authentication

Evaluation Date : 24/06/2024

Before the Jury :
Fares Kahlesnane President UKM Ouargla
Djediai Ahmida Examiner UKM Ouargla
Salah Euschi Supervisor UKM Ouargla

Academic year: 2023/2024

Acknowledgments

We would like to express our sincere gratitude to all those who have contributed to the completion of this study. First and foremost, we extend our deepest appreciation to [Salah Euischi], our supervisor, for their invaluable guidance, unwavering support, and insightful feedback throughout the research process. Their expertise and mentorship have been instrumental in shaping the direction and scope of this work.

Also, we extend our heartfelt thanks to our families, friends, and loved ones for their unwavering support, encouragement, and understanding throughout this journey. Their patience, encouragement, and belief in our abilities have been a constant source of strength and motivation.

Dedication

This work is dedicated to my family and friends, whose unwavering support and encouragement have been the cornerstone of my academic and personal achievements. To my parents, who have always believed in me and provided the foundation for my aspirations; to my friends, who have been a source of constant motivation and inspiration.

A special acknowledgment goes to my mentors and colleagues, whose guidance and wisdom have been invaluable throughout this journey. Their insights and support have greatly enriched this work.

Lastly, to all the researchers and pioneers in the field of medical image processing and watermarking, whose groundbreaking work has paved the way for this study. Thank you for your dedication and contributions to advancing science and technology.

Abstract

Medical image watermarking is essential for ensuring the integrity, authenticity, and confidentiality of digital medical images used in healthcare for diagnosis, treatment planning, and research purposes. In this study, we propose a 3 watermarking techniques based on the Least Significant Bit (LSB), the Discrete Cosine Transform (DCT) and the Discrete Wavelet Transform (DWT) for embedding and extracting watermarks in medical images.

The objective of our research is to develop a robust watermarking algorithm capable of withstanding common attacks while preserving the diagnostic quality of medical images. We evaluate the performance of our proposed method using matrices (MSE, PSNR, SSIM and NC), using a diverse dataset of medical images. Our findings demonstrate that the proposed watermarking technique achieves acceptable levels of robustness and imperceptibility, making it suitable for secure transmission and storage of medical images.

The significance of this research lies in its contribution to advancing the field of medical image security and facilitating the development of improved watermarking algorithms tailored to the specific requirements of medical imaging applications.

Keywords: Medical Image Watermarking, Discrete Cosine Transform (DCT), Least Significant Bit (LSB), Discrete Wavelet Transform (DWT), Robustness, Imperceptibility.

Résumé

Le tatouage d'images médicales est essentiel pour garantir l'intégrité, l'authenticité et la confidentialité des images médicales numériques utilisées dans les soins de santé à des fins de diagnostic, de planification de traitement et de recherche. Dans cette étude, nous proposons 3 techniques de filigrane basées sur le bit le moins significatif (LSB), la transformation cosinus discrète (DCT) et la transformation en ondelettes discrètes (DWT) pour intégrer et extraire des filigranes dans des images médicales.

L'objectif de nos recherches est de développer un algorithme de tatouage robuste capable de résister aux attaques courantes tout en préservant la qualité diagnostique des images médicales. Nous évaluons les performances de notre méthode proposée à l'aide de matrices (MSE, PSNR, SSIM et NC), en utilisant un ensemble de données diversifié d'images médicales. Nos résultats démontrent que la technique de tatouage proposée atteint des niveaux acceptables de robustesse et d'imperceptibilité, ce qui la rend adaptée à la transmission et au stockage sécurisés d'images médicales.

L'importance de cette recherche réside dans sa contribution à l'avancement du domaine de la sécurité des images médicales et à la facilitation du développement d'algorithmes de tatouage améliorés adaptés aux exigences spécifiques des applications d'imagerie médicale.

Mots clés: Filigrane d'images médicales, Transformation cosinus discrète (DCT), Bit de poids faible (LSB), Transformation en ondelettes discrètes (DWT), Robustesse, Imperceptibilité.

ملخص

تعد العلامة المائية للصور الطبية ضرورية لضمان سلامة وأصالة وسرية الصور الطبية الرقمية المستخدمة في الرعاية الصحية لأغراض التشخيص وتخطيط العلاج وأغراض البحث. في هذه الدراسة، نقترح ثلاث تقنيات للعلامة المائية تعتمد على البت الأقل أهمية (LSB)، وتحويل جيب التمام المنفصل (DCT) وتحويل المويجات المنفصلة (DWT) لتضمين واستخراج العلامات المائية في الصور الطبية.

الهدف من بحثنا هو تطوير خوارزمية قوية للعلامة المائية قادرة على تحمل الهجمات الشائعة مع الحفاظ على الجودة التشخيصية للصور الطبية. نقوم بتقييم أداء طريقتنا المقترحة باستخدام المصفوفات (MSE, PSNR, SSIM, NC)، وذلك باستخدام مجموعة بيانات متنوعة من الصور الطبية. توضح النتائج التي توصلنا إليها أن تقنية العلامة المائية المقترحة تحقق مستويات مقبولة من المتانة وعدم الإدراك، مما يجعلها مناسبة لنقل الصور الطبية وتخزينها بشكل آمن.

تكمن أهمية هذا البحث في مساهمته في تطوير مجال أمن الصور الطبية وتسهيل تطوير خوارزميات العلامات المائية المحسنة المصممة خصيصًا للمتطلبات المحددة لتطبيقات التصوير الطبي.

الكلمات المفتاحية: العلامة المائية للصور الطبية، تحويل جيب التمام المنفصل، البت الأقل أهمية، تحويل المويجات المنفصلة، المتانة، عدم القدرة على الإدراك.

Table of Contents

List of Figures

List of Tables

Acronyms

BER Bit Error Rate.

DCT Discrete Cosine Transform.

DFT Discrete Fourier Transform.

DHT Discrete Hartley Transform.

DST Discrete Sine Transform.

DWT Discrete Wavelet Transform.

HH High-High Frequency.

HL High-Low Frequency.

HVS Human Visual System.

ICT Information and Communication Technology.

JPEG Joint Photographic Experts Group (Image Compression Standard).

LH Low-High Frequency.

LL Low-Low Frequency.

LSB Least Significant Bit.

MAE Mean Absolute Error.

MIW Medical Images Watermarking.

MSE Mean Squared Error.

NC Normalized Cross-Correlation.

NPCR	Number of Pixel Change Rate.
PNG	Portable Network Graphics.
PSNR	Peak Signal-to-Noise Ratio.
RGB	Red Green Blue (Color Model).
SNR	Signal-to-Noise Ratio.
SSIM	Structural Similarity Index.
UACI	Unified Average Changing Intensity.

Chapter 1

General Introduction

In recent years, the digitization of medical imaging has revolutionized healthcare by enabling efficient storage, transmission, and analysis of medical images such as X-rays, CT scans, MRIs, and ultrasound images [38]. However, the widespread adoption of digital medical imaging also raises concerns about the security these images, particularly regarding issues such as unauthorized access, tampering, and data breaches. Medical image watermarking emerges as a promising solution to address these challenges by embedding imperceptible yet robust digital signatures or identifiers into medical images, thereby ensuring their authenticity and confidentiality [28][57][54].

The primary objective of medical image watermarking is to enhance the security and reliability of digital medical imaging systems while preserving the diagnostic quality and clinical utility of medical images. By embedding watermarks directly into medical images, healthcare providers, researchers, and patients can safeguard the integrity of sensitive medical data, prevent unauthorized modifications or tampering, and trace the origin of digital images throughout their lifecycle.

The goal of this work is to create and assess sophisticated watermarking methods that may safely and effectively incorporate data into medical images without sacrificing image quality. We seek to improve the robustness, imperceptibility, and resilience of watermarks in medical imaging by investigating techniques including Discrete Wavelet Transform (DWT), Discrete Cosine Transform (DCT), and Least Significant Bit (LSB) embedding. In this study, we focus on investigating and developing advanced watermarking techniques tailored specifically to the unique characteristics and requirements of medical imaging applications.

Through comprehensive experimentation and evaluation, we aim to assess the performance, efficacy, and practical feasibility of our proposed watermarking technique, as well as the potential contributions of LSB, DCT and DWT techniques in the context of medical image security. By addressing key challenges and limitations in existing watermarking methods, our research endeavors to advance the state-of-the-art in medical image watermarking and facilitate the development of more secure and reliable digital medical imaging systems.

The rest of this thesis is structured as follows:

1. **Requirements for MIW:** This chapter outlines the fundamental requirements for effective and secure medical image watermarking (MIW). It delves into the essential criteria that watermarking techniques must meet to ensure robust protection of medical images. Key requirements include imperceptibility, robustness against various attacks, capacity for sufficient data embedding, computational efficiency, and reversibility to restore the original image when needed. By defining these requirements, the chapter sets the foundation for evaluating and comparing different MIW techniques, guiding the development of advanced methods tailored to the unique demands of medical imaging.

2. **Methods and Performance of MIW:** This chapter presents a comprehensive examination of the various methods utilized in medical image watermarking (MIW) and evaluates their performance. It covers traditional and contemporary watermarking techniques, including spatial and transform domain approaches such as Least Significant Bit (LSB), Discrete Cosine Transform (DCT), and Discrete Wavelet Transform (DWT). The chapter also discusses the performance metrics used to assess these methods, such as imperceptibility, robustness, capacity, and computational complexity. Through detailed analysis and comparison, this chapter aims to highlight the strengths and weaknesses of different MIW techniques, providing insights into their practical applications and guiding future research in the field.

3. **Experiment and Results discussion:** This chapter details the experimental setup and procedures used to evaluate the performance of various medical image watermarking techniques. It presents the results of these experiments, including quantitative metrics such as Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Normalized Cross-Correlation (NC). The discussion interprets these results, comparing the effectiveness and efficiency of different watermarking methods under various conditions. By analyzing the outcomes, this chapter provides a critical assessment of the techniques' strengths and weaknesses, offering insights into their practical implications and potential areas for improvement.

Chapter 2

Requirements for MIW

2.1 Introduction

One important method for addressing the confidentiality, integrity, and validity of digital medical images is medical image watermarking (MIW). Ensuring the security and dependability of digital imaging technologies is crucial as the healthcare sector uses them more and more for patient records, treatment planning, and diagnostics. MIW provides a way to incorporate undetectable data into medical images, which can then be retrieved to confirm the authenticity and integrity of the image. The fundamental needs for a successful MIW are discussed in this chapter, including reversibility, capacity, security, robustness, and imperceptibility. Comprehending these prerequisites is essential in order to create watermarking algorithms that satisfy the demanding specifications of medical imaging applications. fig 2.1 has an illustration of Watermarking medical images requirements.

2.2 Image Imperceptibility

It is a factor that is utilized to determine how comparable the original medical image and medical images with watermark [17][72][77]. It is necessary to insert the watermark. in the host medical image so that the watermark is invisible to humans Visual System (HVS) and preserves the host image's quality. This suggests that HVS should be impossible to differentiate between the original and the watermarked medical image.

medical illustration. One of the most important characteristics of a watermarking system should be image quality. have, particularly for systems that watermark medical images, where even the smallest change can lead to an incorrect diagnosis. Consequently, watermarking systems for medical images should Try to maintain the highest level of perceptivity feasible.[22]

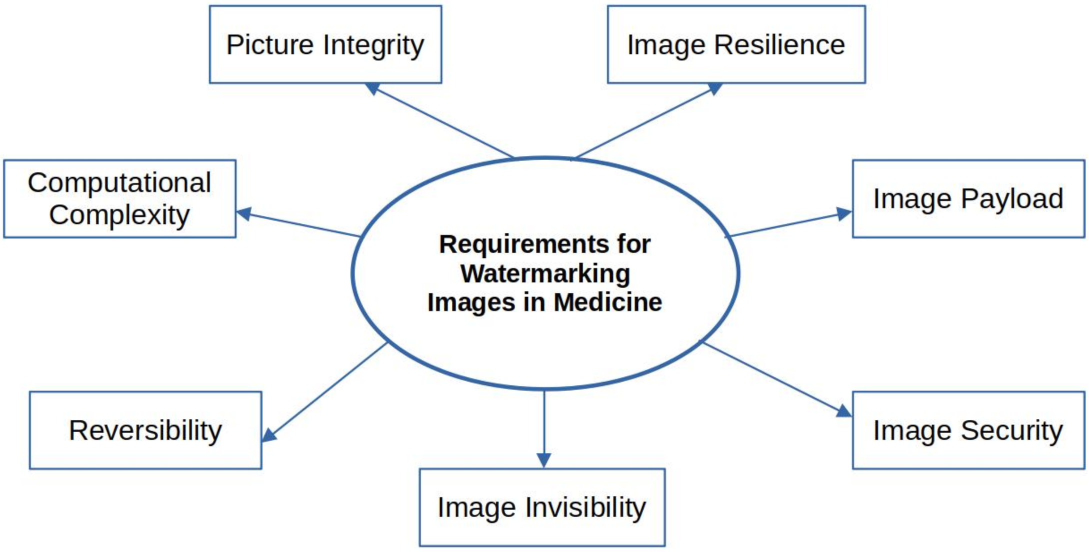


Figure 2.1: Requirements for Watermarking Images in Medicine.

2.3 Image Robustness

The medical image watermarking system's robustness is measured by its resistance to several signal processing and geometric assaults [73]. It is necessary to verify how resilient the watermark is against these kinds of assaults because these images are vulnerable to both deliberate and inadvertent ones. Certain medical image watermarking methods may be delicate, even if they don't all have to be. Watermarks that are not resistant to deliberate or inadvertent assaults are considered fragile.

Ensuring the resilience of medical images to various forms of degradation and manipulation is essential for maintaining their integrity, authenticity, and diagnostic utility [49]. Medical images are often subject to a range of distortions during acquisition, transmission, storage, and processing, including compression artifacts, noise, geometric transformations, and malicious attacks [16]. Therefore, it is crucial to develop watermarking techniques that can withstand these challenges and reliably recover embedded watermarks under adverse conditions.

One approach to enhancing image resilience is through the use of robust watermarking algorithms that exploit the redundancy and perceptual properties of medical images to embed watermark data in a manner that is resistant to common distortions and attacks [16]. These algorithms typically employ error correction coding, spread spectrum modulation, and perceptual modeling techniques to embed redundant and robust watermark data into the image while minimizing the impact on its visual quality and diagnostic information [16][21].

Furthermore, the adoption of reversible watermarking techniques, such as reversible data hiding (RDH), allows for the embedding of watermark data in a manner that preserves the original image content and enables lossless extraction of the watermark when needed [21]. Reversible watermarking algorithms achieve this by exploiting the spatial and frequency domains of medical images to embed watermark data without introducing irreversible modifications, thus ensuring the integrity and authenticity of the original image [21].

In addition to algorithmic approaches, the resilience of medical images can be enhanced through the integration of authentication and verification mechanisms, such as digital signatures and cryptographic techniques [33]. These mechanisms enable the verification of image authenticity and integrity through the validation of embedded watermarks and the detection of any unauthorized modifications or tampering attempts [33].

By combining robust watermarking algorithms with authentication mechanisms and error correction coding, medical imaging systems can ensure the resilience of images against a wide range of distortions and attacks, thereby enhancing their reliability for clinical diagnosis, telemedicine, and secure data exchange.

2.4 Image Payload

The payload capacity of medical image watermarking systems refers to the maximum amount of data that can be embedded into an image while maintaining its diagnostic quality and visual integrity [21]. The payload capacity is a critical consideration in watermarking design, as it determines the amount of auxiliary information that can be reliably conveyed within the image for various applications, including patient identification, copyright protection, and data authentication.

The payload capacity of a watermarking system is influenced by several factors, including the size and complexity of the medical image, the embedding algorithm, the desired level of robustness, and the perceptual constraints imposed by human visual perception. In general, larger and more complex images can accommodate a greater payload capacity due to their increased spatial redundancy and spectral diversity [33]. However, the payload capacity must be carefully balanced with the imperceptibility and robustness requirements of the watermarking system to ensure that the embedded data remains undetectable and resistant to common distortions and attacks [49].

One approach to increasing the payload capacity of medical image watermarking systems is through the use of data compression techniques, such as wavelet and JPEG compression [25]. These techniques exploit the spatial and spectral redundancies present in medical images to reduce their size while preserving their diagnostic quality, thereby creating additional space for embedding watermark data without perceptible degradation.

Furthermore, recent advances in reversible watermarking techniques have enabled the embedding of high-capacity payloads into medical images without introducing irreversible modifications . [10] Reversible watermarking algorithms achieve this by exploiting the spatial and spectral redundancies of medical images to embed watermark data in a reversible manner, allowing for lossless extraction of the embedded data when needed.

By optimizing the payload capacity of medical image watermarking systems, researchers can maximize the utility and effectiveness of watermarking techniques for various applications in healthcare, including image authentication, data integrity verification, and patient information management.

2.5 Image Security

Image security refers to the protection of images from unauthorized access, tampering, or theft. In the context of digital images, ensuring security is essential to safeguard sensitive information and maintain privacy [2]. Various techniques are employed to enhance image security, including encryption, watermarking, and authentication.

2.5.1 Encryption

Encryption is a fundamental technique used to secure digital images by converting them into an unreadable format using cryptographic algorithms. Only authorized users with the appropriate decryption key can access the original image. Advanced encryption standards such as AES (Advanced Encryption Standard) and RSA (Rivest-Shamir-Adleman) are commonly utilized to ensure robust security 7.

2.5.2 Watermarking

Watermarking is another effective method for image security, wherein imperceptible digital signatures or marks are embedded into the image data. These watermarks can be visible or invisible and serve various purposes, including copyright protection, content authentication, and ownership verification. Techniques like digital watermarking and steganography are widely employed for this purpose [18].

2.5.3 Authentication

Authentication mechanisms are crucial for verifying the integrity and authenticity of digital images. Digital signatures, hash functions, and digital certificates are commonly used for image authentication. These techniques enable users to validate the origin and integrity of images, ensuring that they have not been tampered with or altered maliciously [74], 3].

2.6 Image Invisibility

Image invisibility refers to the ability to conceal or hide information within digital images without altering their visual appearance to the human eye. This technique is commonly employed in applications where maintaining the visual integrity of the image is paramount, such as covert communication, data embedding, and copyright protection.

2.6.1 Steganography

Steganography is a prevalent method for achieving image invisibility, where secret data is embedded within the image pixels in a manner that is imperceptible to human observers. Various steganographic techniques exist, including LSB (Least Significant Bit) embedding, frequency domain techniques, and spatial domain methods. These techniques ensure that the hidden information remains undetectable, even upon close inspection of the image [26].

2.6.2 Visual Transparency

Invisible watermarking is another approach to achieve image invisibility, wherein a digital watermark is embedded into the image data without visibly altering its appearance. The watermark is imperceptible to the human eye but can be extracted and decoded using specialized algorithms. This method is commonly used for copyright protection, content authentication, and digital rights management [13].

2.7 Reversibility

Reversibility in image watermarking refers to the ability to recover the original image or data accurately after the watermark has been embedded and extracted. In many applications, it is essential for the watermarking process to be reversible to ensure that the original image quality is not significantly degraded and that the embedded information can be reliably retrieved.

2.7.1 Lossless Embedding

Lossless embedding techniques ensure that the watermark is inserted into the image without causing any irreversible changes to the original pixel values. This approach guarantees that the original image can be reconstructed exactly as it was before the watermarking process, allowing for perfect reversibility. Lossless embedding methods often involve modifying specific image components, such as the least significant bits (LSBs) or the discrete cosine transform (DCT) coefficients, to embed the watermark while minimizing distortion [18][26].

2.7.2 Watermark Extraction

During watermark extraction, the embedded watermark is retrieved from the watermarked image without altering the image content. Reversible extraction algorithms recover the watermark with high fidelity, enabling the original data to be reconstructed accurately. These algorithms typically exploit the redundancy or statistical properties of the watermark to achieve reliable extraction while preserving image quality [13].

2.8 Computational Complexity

Computational complexity is a critical consideration in image watermarking systems, as it directly impacts the efficiency and performance of the embedding and extraction processes. Various factors contribute to the computational complexity of watermarking algorithms, including the size of the image, the complexity of the embedding technique, and the level of security required.

2.8.1 Embedding Complexity

The computational complexity of embedding algorithms depends on the method used to insert the watermark into the image. For example, frequency domain techniques such as the discrete cosine transform (DCT) or discrete wavelet transform (DWT) may involve complex mathematical operations, such as matrix multiplications or convolution, resulting in higher computational overhead [18][13]. On the other hand, spatial domain methods like least significant bit (LSB) substitution may have lower computational complexity but may lack robustness and security [26].

2.8.2 Extraction Complexity

The complexity of watermark extraction algorithms also varies based on the chosen technique. Reversible extraction methods, which aim to recover the watermark without altering the original image, often require sophisticated processing to accurately estimate and remove the embedded watermark. In contrast, non-reversible techniques may involve simpler computations but may sacrifice image quality or require additional post-processing steps [46].

2.8.3 Trade-offs

In practice, watermarking algorithms must strike a balance between computational complexity, robustness, and security. Highly complex algorithms may offer stronger protection against attacks but may require significant computational resources and time to execute. Conversely, simpler techniques may be faster but may be more vulnerable to attacks or may produce lower-quality watermarked images.

2.9 Conclusion

In the context of digital healthcare, ensuring the confidentiality and integrity of medical images requires compliance with watermarking rules. Robustness assures that the watermark will remain visible even after numerous image changes, while imperceptibility ensures that it won't affect medical diagnostics. The quantity of data that can be safely inserted without sacrificing image quality is determined by capacity. Reversibility enables the full restoration of the original image when needed, while security guards against unwanted discovery and modification of the watermark. Medical photographs can be safely protected using MIW procedures, guaranteeing that they will always be a reliable part of healthcare information systems. The essential needs that must be taken into account while developing and putting into practice MIW systems have been delineated in this chapter, laying the groundwork for additional study and advancement in this crucial field.

Chapter 3

Methods and Performance of MIW

3.1 Introduction

When it comes to safeguarding Electronic Patient Records (EPRs) that include sensitive patient information during the transfer of medical images, watermarking is seen to be the best option [5][12][15][17][30][29]. Researchers are increasingly concerned about the possibility of biological image theft resulting from transmission across unsecure networks. Usually, there is a chance that this data will be altered, whether on purpose or accidentally. Furthermore, the newly established medical image communication system may lose the trustworthiness of the data due to a variety of harmful assaults. When sending across insecure networks, watermarking has been thought to be the best option for protecting and authenticating medical images and EPRs that include sensitive patient data.

3.2 Image-Based Watermark Embedding

Image-based watermark embedding techniques involve the direct insertion of watermark data into the pixel values of an image. These methods aim to embed imperceptible watermarks while ensuring minimal distortion to the original image content. Image-based watermarking approaches can be categorized into spatial domain techniques, which operate directly on pixel values, and transform domain techniques, which exploit mathematical transformations to embed watermarks.

3.2.1 Spatial Domain Techniques

Spatial domain watermarking methods manipulate the pixel values of an image to embed the watermark. The Least Significant Bit (LSB) substitution is one of the simplest and most widely used spatial domain techniques. In LSB substitution, the least significant bits of selected pixels are replaced with the bits of the watermark data, causing minimal visual distortion while embedding the watermark [14]. Other spatial domain techniques include spread spectrum watermarking, where the water-

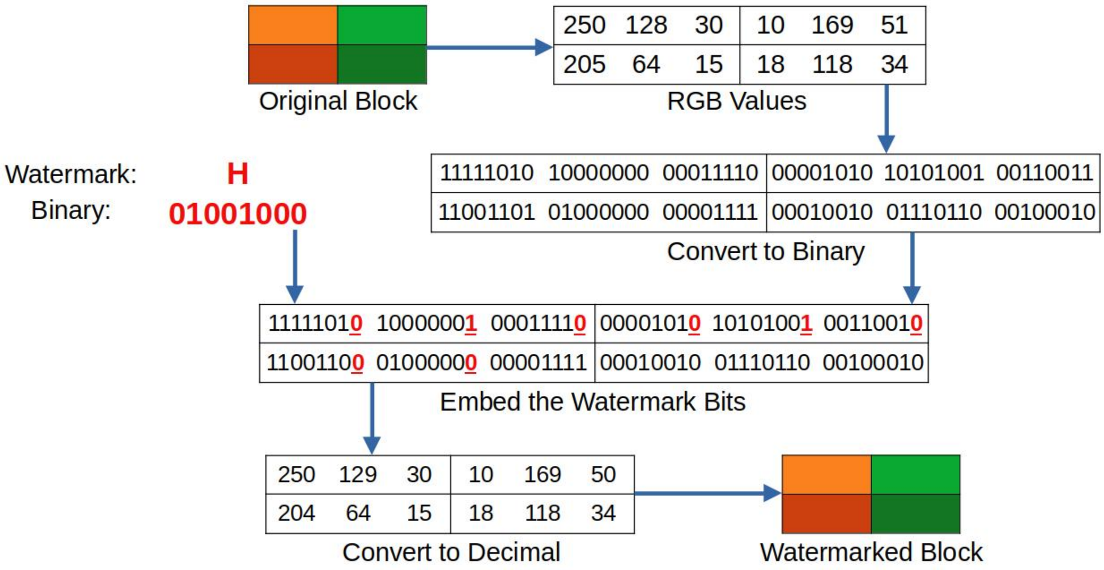


Figure 3.1: LSB Algorithm.

mark signal is spread across the image using pseudo-random sequences to enhance robustness against attacks [47].

3.2.1.1 Least Significant Bit (LSB) Embedding

It is among the most traditional and straightforward techniques for watermarking in the spatial domain. LSB embedding involves replacing the least significant bits of image pixels with the watermark data [60][20]. This technique is simple and effective, especially for lossless compression formats like PNG. However, it is vulnerable to various attacks and may introduce visible artifacts in the image.

The watermark is embedded in the LSB to do this. Encoding the watermark comes before embedding. The encoded bits are embedded by first converting the pixel values to binary form and then replacing the rightmost bit of each pixel with the encoded watermark bits. The binary value image pixel is transformed back to the decimal value image pixel following the replacement of the LSB. Fig. 3.1 provides an illustration of this, Fig. 3.2 displays the flow graph for LSB substitution.

Calculating the embedding space for LSB (Least Significant Bit) watermarking involves determining the total number of available LSBs in the image pixels that can be used for embedding the watermark. Here's how we can calculate the embedding space for LSB watermarking:

1. Image Size: Determine the dimensions (width and height) of the image in pixels. Let's denote the width as W and the height as H.
2. Color Depth: Determine the color depth of the image, which specifies the number of bits used to represent each pixel. Common color depths include 8 bits per channel for grayscale images and 24 bits (8 bits per channel) for RGB color images.

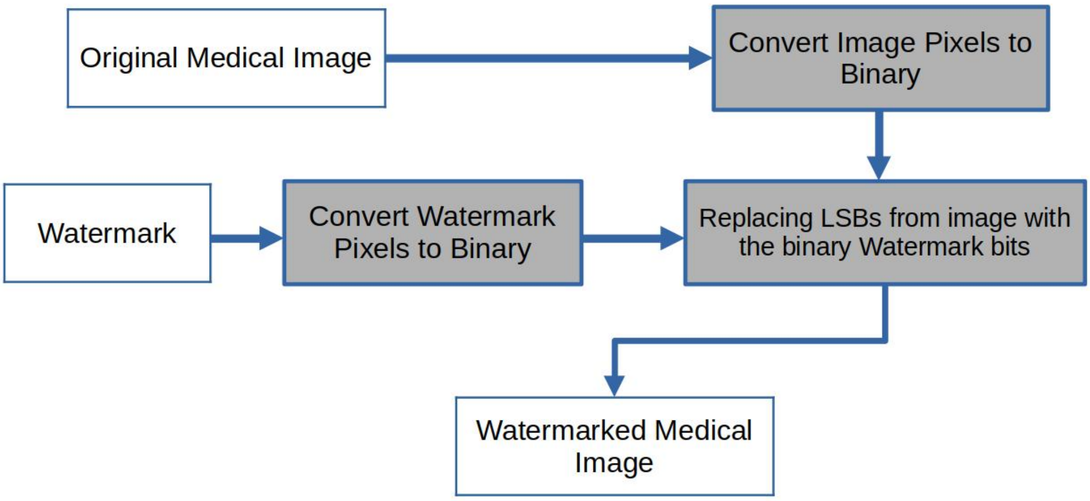

Original Medical Image
Convert Image Pixels to Binary
Watermark
Convert Watermark Pixels to Binary
Replacing LSBs from image with the binary Watermark bits
Watermarked Medical Image

Figure 3.2: Flow chart for replacing LSBs.

3. Number of Pixels: Calculate the total number of pixels in the image, which is the product of the width and height:

$$N = W \times H \tag{3.1}$$

4. Number of LSBs per Pixel: For LSB watermarking, each pixel typically provides one LSB per color channel (e.g., red, green, blue for RGB images). Therefore, for an image with C color channels, the number of available LSBs per pixel is C.

5. Total Embedding Space: Multiply the number of pixels by the number of LSBs per pixel to obtain the total embedding space. This represents the maximum number of bits that can be embedded in the image:

$$\text{Embedding Space} = N \times C \tag{3.2}$$

The embedding space calculated using this method gives we the total number of bits that can be used to embed the watermark. Keep in mind that the actual payload size may be lower due to constraints such as the need to maintain image quality, avoid perceptual distortion, and ensure robustness against attacks and image processing operations.

After determining the embedding space, we can use it to calculate the maximum payload size for your watermarking application, taking into account any additional factors or constraints specific to your implementation.

3.2.1.2 Spatial Domain Filtering

Spatial domain filtering modifies the pixel values in the image to embed the watermark. Examples include techniques based on adding noise or altering pixel intensities to encode the watermark information. Spatial filtering methods offer flexibility and robustness but may degrade image quality [42].

3.2.1.3 Modification of Image Histogram

Another spatial domain method that has been applied to data concealment in medical images is histogram modification [58]. Peak bins are used in this approach to incorporate data in a histogram. Although this approach is simple to use, it can only integrate data into a limited number of maximum or peak points that are accessible.

Histogram modification techniques are widely used in image processing for various purposes, including contrast enhancement, brightness adjustment, and equalization. These techniques aim to redistribute the pixel intensities in an image to achieve desired visual effects or improve image quality.

Contrast Enhancement One common application of histogram modification is contrast enhancement, which aims to improve the visual appearance of an image by increasing the difference in intensity between different regions. Techniques such as histogram stretching and histogram equalization are often used for this purpose.

Brightness Adjustment Histogram modification can also be used for brightness adjustment, allowing users to control the overall brightness of an image. This can be achieved by shifting the histogram along the intensity axis or by scaling its values [62].

Histogram Equalization Histogram equalization is a popular technique used to enhance the contrast of an image by redistributing the pixel intensities to cover the entire intensity range more uniformly. This can be particularly useful for improving the visibility of details in images with low contrast.

3.2.1.4 Local Binary Patterns (LBP)

Local Binary Patterns (LBP) is a texture descriptor used in computer vision for texture classification. In the context of watermarking, LBP has been employed as a feature extraction technique to capture the texture characteristics of the image. The watermark embedding process using LBP involves encoding the watermark information into the texture features extracted by the LBP algorithm. The image is divided into non-overlapping pieces, and then the differences are computed. After that, these pixels are embedded using the guidelines provided in [75].

3.2.1.5 Edge-based Techniques

Edge-based watermarking techniques focus on embedding watermarks in image edges or regions of interest. By exploiting edge information, these methods aim to enhance robustness against attacks while minimizing visual distortion. Edge-based techniques often involve edge detection algorithms and selective embedding in edge regions [39].

3.2.2 Transform Domain Techniques

Transform domain watermarking techniques utilize mathematical transformations, such as the Discrete Cosine Transform (DCT) or Discrete Wavelet Transform (DWT), to embed watermarks into images. These techniques exploit the frequency or spatial-frequency characteristics of images to embed imperceptible watermarks while minimizing distortion. For example, DCT-based watermarking modifies the frequency coefficients of an image to embed the watermark, leveraging the frequency localization property of the DCT to ensure invisibility [4]. Similarly, DWT-based watermarking decomposes the image into different frequency bands and embeds the watermark in selected sub-bands to achieve robustness and invisibility [32], those are some of the transform domain techniques:

3.2.2.1 Discrete Cosine Transform (DCT)

The Discrete Cosine Transform (DCT) is a widely used transform technique that decomposes an image into a set of frequency components. In DCT-based watermarking, the cover image is divided into small blocks, and the DCT is applied to each block. The watermark information is then embedded into selected DCT coefficients. Since the DCT concentrates most of the image energy in a few low-frequency coefficients, it is more robust against compression and common image processing operations.

DCT offers an appealing and effective image transformation that converts an n-dimensional vector to n number of coefficients in a linear fashion. Low-frequency component (LFC), middle-frequency component (MFC), and high-frequency component (HFC) are the three distinct frequency components into which it splits the image. Compressed LFC has the maximum energy [61]. While DCT is more resilient to JPEG compression, it is less resilient to geometric attacks such as scaling, rotation, and cropping. The following formulas display the DCT and its inverse.

$$C_T(u,v) = \frac{2}{\sqrt{pq}}\beta(u)\beta(v)\sum_{x=0}^{p-1}\sum_{y=0}^{q-1} f_t(x,y) \times Cos\left(\frac{(2x+1)u\pi}{2p}\right) * Cos\left(\frac{(2y+1)v\pi}{2q}\right) \tag{3.3}$$

and

$$f_T(x,y) = \frac{2}{\sqrt{pq}}\sum_{u=0}^{p-1}\sum_{v=0}^{q-1} \beta(u)\beta(v) f_T(x,y) \times Cos\left(\frac{(2x+1)u\pi}{2p}\right) * Cos\left(\frac{(2y+1)v\pi}{2q}\right) \tag{3.4}$$

In this case, the block sizes are p and q, the original image pixel is represented by $f_T(x,y)$, the transform domain coefficient is $C_T(u,v)$, and the values of $\beta(u)$ and $\beta(v)$ are computed as

$$\beta(u),\beta(v) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{if } u,v = 1 \\ 1, & \text{else} \end{cases} \tag{3.5}$$

Several studies have proposed DCT-based watermarking techniques for various applications, including medical image watermarking, copyright protection, and authentication. These techniques typically focus on optimizing the embedding process to maximize watermark robustness while minimizing perceptual distortion.

The embedding space for DCT (Discrete Cosine Transform) watermarking refers to the number of coefficients in the DCT domain that can be used to embed the watermark. In DCT-based watermarking, not all DCT coefficients are suitable for embedding the watermark. Typically, only a subset of coefficients, often referred to as "watermarkable coefficients," are used for embedding to minimize the impact on image quality.

To calculate the embedding space for DCT watermarking, we need to consider the following factors:

1. **Image Size:** Determine the dimensions (width and height) of the image.
2. **Block Size:** Determine the size of the DCT blocks used for processing. Common block sizes include 8x8, 16x16, or 32x32.
3. **Compression Ratio:** If the image undergoes compression (e.g., JPEG compression), consider the compression ratio, as it affects the number of non-zero DCT coefficients.
4. **Watermarking Method:** Depending on the watermarking method (e.g., spread spectrum, LSB modification), identify the specific DCT coefficients suitable for embedding the watermark.

Once we have these factors, we can calculate the embedding space using the following formula:

$$\text{Embedding Space} = NB_B \times NB_{WC} \tag{3.6}$$

where NB_B is the number of blocks, and NB_{WC} is the number of watermarking per blocks

The number of blocks is determined by dividing the image dimensions by the block size. For example, for an image of size $W \times H$ and a block size of $B \times B$, the number of blocks would be:

$$\text{Number of Blocks} = \left\lceil \frac{W}{B} \right\rceil \times \left\lceil \frac{H}{B} \right\rceil \tag{3.7}$$

The number of watermark-able coefficients per block depends on the watermarking method and the selected coefficients for embedding.
Keep in mind that the actual embedding space may be further constrained by factors such as the robustness requirements, perceptual quality constraints, and compatibility with image compression standards.
Once we have calculated the embedding space, we can use it to determine the maximum payload size or the embedding capacity for the watermarking process.

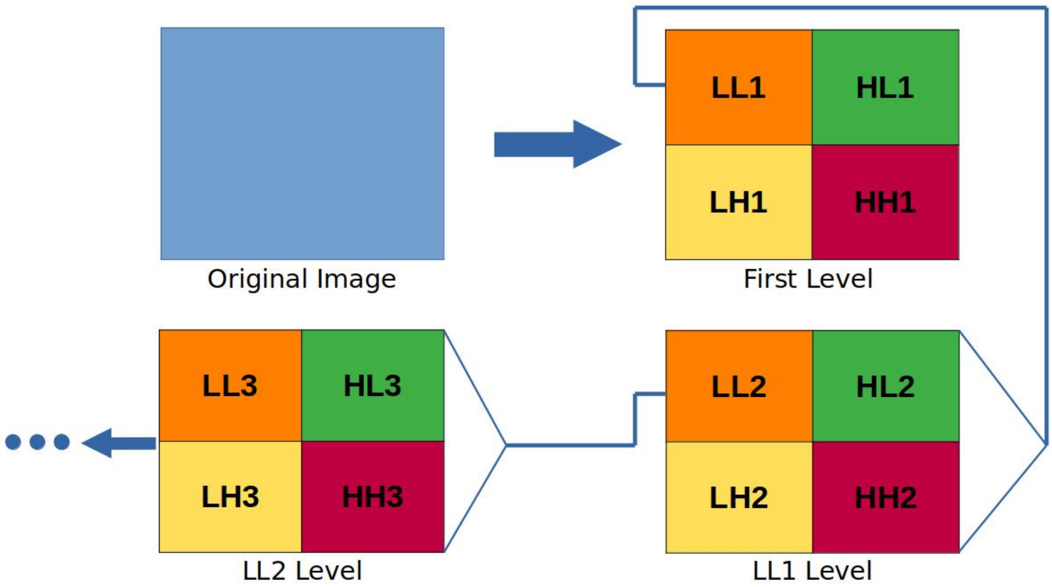


Figure 3.3: Breakdown of the sub-bands within DWT.

3.2.2.2 Discrete Wavelet Transform (DWT)

The Discrete Wavelet Transform (DWT) is another widely used transform technique that decomposes an image into different frequency sub-bands. Without a doubt, wavelet transformations have become quite well-liked in the image watermarking community [17].
Unlike DCT, DWT provides both frequency and spatial localization, making it suitable for watermarking applications requiring both robustness and localization. The most effective and widely applied transform domain approach is DWT [22].

The wavelet transform is usually based on a principle that is similar to that of the Fourier transform [27], with the exception that a centered, windowed function with a null integral is used to weight the complex sinusoid.
This function can take many various forms, but it always has the ability to adjust to the signal's frequency after compression or expansion [24]. In fact, a wavelet's temporal compression causes the specific frequency to rise noticeably, whereas a wavelet's temporal expansion usually causes it to fall. Generally speaking, a multi-resolution decomposition employing adaptive filters is employed to perform wavelet decomposition on used images [55]. It is accomplished by subsampling after a series of directed low-pass and high-pass filtering.
The number of times that these filtering and subsampling are done is usually n, which corresponds to the appropriate level of the functional decomposition. To obtain each local sub-band's complex coefficients, Using the Haar filter, we can compute the following values with accuracy:

$$LL(x,y) = \frac{p(x,y) + p(x,y+1) + p(x+1,y) + p(x+1,y+1)}{2} \tag{3.8}$$

$$LH(x,y) = \frac{p(x,y) + p(x,y+1) + p(x+1,y) - p(x+1,y+1)}{2} \tag{3.9}$$

$$HL(x,y) = \frac{p(x,y) - p(x,y+1) + p(x+1,y) - p(x+1,y+1)}{2} \tag{3.10}$$

$$HH(x,y) = \frac{p(x,y) - p(x,y+1) - p(x+1,y) - p(x+1,y+1)}{2} \tag{3.11}$$

where $p(x,y)$ are the position in the original image.

In DWT-based watermarking, the cover image is decomposed into multiple wavelet subbands, and the watermark information is embedded into selected coefficients in these subbands. DWT offers multi-resolution analysis, allowing for efficient embedding in different frequency bands based on the desired trade-off between robustness and invisibility.

Accurate spatial localization is provided by the multi-resolution properties. As seen in 3.3. it divides the image into four sub-bands: Low Low (LL), Low High (LH), High Low (HL), and High High (HH). While other sub-bands provide the detail that the LL sub-band misses, the LL sub-band contains the most important information regarding the image details. Moreover, DWT provides a hierarchical breakdown of the LL sub-band [7][69]. The following formulas are used to determine the energy in the DWT scenario [22].

$$E_N = \frac{1}{P_N Q_N} \sum_k \sum_l |l_C(k,l)| \tag{3.12}$$

where $p(x,y)$ are the position in the original image.

3.2.2.3 Discrete Fourier Transform (DFT)

The Discrete Fourier Transform (DFT) technique used to convert a finite sequence of equally spaced samples of a function into a same-length sequence of complex numbers representing the frequency domain of the original function. It is widely used in signal processing, image processing, and various other fields, the following formulas display the DFT:

$$X[K] = \sum_{n=0}^{N-1} x[n] \times e^{-j\frac{2\pi}{N}kn}, K = 0, 1, 2, ..., N-1 \tag{3.13}$$

where:

- $X[k]$ is the k-th frequency component.

- $x[n]$ is the n-th sample of the input sequence.
- j is the imaginary unit ($j = -1$).
- N is the total number of samples.

The inverse DFT (IDFT), which converts the frequency domain representation back to the time domain, is given by:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \times e^{j\frac{2\pi}{N}kn}, n = 0, 1, 2, ..., N-1 \tag{3.14}$$

3.2.2.4 Discrete Sine Transform (DST)

The Discrete Sine Transform (DST) is similar to the DFT, but it uses only sine functions. It is particularly useful for certain types of boundary conditions in solving partial differential equations and for signal processing applications where odd symmetry is assumed, there are several types of DST, but the most commonly used one is DST Type-II, the following formulas display it:

$$X[k] = \sum_{n=0}^{N-1} x[n] \times sin\left(\frac{\pi(n+1)(k+1)}{N+1}\right), k = 0, 1, 2, ..., N-1 \tag{3.15}$$

where:

- $X[k]$ is the k-th frequency component.
- $x[n]$ is the n-th sample of the input sequence.
- N is the total number of samples.

the following equation displays The inverse DST (IDST) Type-II:

$$x[n] = \frac{2}{N-1} \sum_{k=0}^{N-1} X[k] \times sin\left(\frac{\pi(n+1)(k+1)}{N+1}\right), n = 0, 1, 2, ..., N-1 \tag{3.16}$$

3.2.2.5 Discrete Hartley Transform (DHT)

The Discrete Hartley Transform (DHT) is a real-valued transform similar to the DFT. It is particularly useful because it avoids the use of complex numbers, making it simpler to implement in certain applications, the following formulas display the DHT:

$$H[k] = \sum_{n=0}^{N-1} x[n] \times \left(cos\left(\frac{2\pi}{N}kn\right) + sin\left(\frac{2\pi}{N}kn\right)\right), k = 0, 1, 2, ..., N-1 \tag{3.17}$$

where:

- $H[k]$ is the k-th transform component.
- $x[n]$ is the n-th sample of the input sequence.
- N is the total number of samples.

The inverse DHT (IDHT) is identical to the forward DHT, which means applying the DHT twice will yield the original sequence (scaled by NN):

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} H[k] \times \left(cos\left(\frac{2\pi}{N}kn\right) + sin\left(\frac{2\pi}{N}kn\right)\right), n = 0, 1, 2, ..., N-1 \quad (3.18)$$

3.3 Self-Generated Watermark Embedding

Self-generated watermark embedding techniques involve creating a watermark signal directly from the content of the image itself. Unlike traditional watermarking methods that rely on external watermark signals, self-generated watermarking extracts features or properties from the image and uses them to generate the watermark. This approach offers several advantages, including simplicity, robustness, and resistance to attacks that target external watermark signals.

3.3.1 Feature-Based Watermarking

Feature-based watermarking methods extract distinctive features from the image, such as texture patterns, color distributions, or geometric structures, and use them to generate the watermark. Common techniques include using local binary patterns (LBP) to extract texture features [43], color histograms for color-based features [68], and edge detection algorithms for shape-based features [9]. These extracted features are then encoded into a watermark signal that is embedded into the image.

3.3.2 Content-Based Watermarking

Content-based watermarking techniques analyze the semantic content of the image, such as objects, scenes, or regions of interest, and derive a watermark signal based on this analysis. For example, content-based watermarking may identify key objects or regions within the image and encode information about them into the watermark. This approach enables the watermark to be closely tied to the content of the image, making it more robust to modifications or transformations that preserve the image's semantic meaning [19].

3.4 Metrics of Performance

Numerous metrics, including resilience, temporal complexity, encrypted and decrypted image quality, and computing speed, can be used to evaluate the suggested algorithms. Fig. 11 presents an overview of these metrics.

3.4.1 Measurement of Image Quality

One of the crucial performance indicators that needs to be addressed. Data dependability may be impacted by certain image encryption algorithms that cause some distortion in the ciphered images. Accurate decrypted images are necessary for some military and medical uses. Misdiagnosis is the result of influencing ROI. The quality of ciphered images is represented by this metric, which is applied by analyzing the correlation between source and decrypted images. Peak signal-to-noise ratio (PSNR), bit correct ratio (BCR), structural similarity index measure (SSIM), signal-to-noise ratio (SNR), mean absolute error (MAE), mean squared error (MSE), and standard dynamic range (SDR) are the assessments of the wide-spread image aspect. Metrics are frequently contested notwithstanding signal fidelity because they ignore the characteristics of the image signal. As an indicator of image quality, they are still commonly used [76].

3.4.1.1 PSNR

The suggested algorithm's high imperceptibility is determined using PSNR criteria, which take into account the degree of similarity between the original and watermarked images. A high PSNR score indicates that the two photos are highly comparable [38]. It is depicted as

$$\mathrm{PSNR} = 10_{log}\frac{(255)^2}{\mathrm{MSE}} \tag{3.19}$$

the optimum value for PSNR is Hight as possible.

3.4.1.2 MSE

MSE stands for Mean Squared Error. It's a widely used metric in image processing and other fields to measure the average squared differences between the original values and the predicted values. In the context of image processing, MSE quantifies the difference between the original image and a modified or reconstructed version of that image. Mathematically, MSE is calculated as follows:

$$\mathrm{MSE} = \frac{1}{X \times Y}\sum_{i=1}^{X}\sum_{j=1}^{Y}(I_{ij} - W_{ij})^2 \tag{3.20}$$

the value range for MSE is between 0 and 1, the optimum value is 0.

3.4.1.3 SSIM

SSIM stands for Structural Similarity Index Measure. It is a metric commonly used to evaluate the similarity between two images, taking into account both their luminance and structural information. SSIM is widely used in image processing and computer vision tasks to assess the quality of image compression, denoising, enhancement, and other image manipulation techniques [74][6].

The SSIM index compares local patterns of pixel intensities in the reference image (usually the original image) and the distorted image (usually the modified or reconstructed image). It computes three components: luminance similarity, contrast similarity, and structural similarity.

Mathematically, SSIM is calculated as follows:

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{3.21}$$

Where:

- x and y are the reference and distorted images, respectively.
- μ_x and μ_y are the means of x and y.
- σ_x^2 and σ_y^2 are the variances of x and y.
- σ_{xy} is the covariance of x and y.
- c_1 and c_2 are small constants to stabilize the division, typically $c_1 = (k_1L)^2$ and $c_2 = (k_2L)^2$, where L is the dynamic range of the pixel values (e.g., 255 for 8-bit images), and k_1 and k_2 are constants to control the impact of the luminance and contrast terms.

The SSIM index ranges between -1 and 1, where 1 indicates perfect similarity between the two images, and -1 indicates complete dissimilarity. Higher SSIM values correspond to greater similarity between the images.

3.4.1.4 SNR

SNR stands for Signal-to-Noise Ratio. It is a measure used to quantify the ratio of the strength of a signal to the strength of background noise that may affect the signal. In various fields such as telecommunications, electronics, audio engineering, and image processing, SNR is an essential metric for assessing the quality of a signal.

In the context of images, SNR is often used to evaluate the quality of an image by comparing the strength of the desired image signal (i.e., the useful information) to the strength of the background noise present in the image. A higher SNR indicates that the signal is stronger relative to the noise, which generally corresponds to a higher quality image.

Mathematically, SNR is calculated as the ratio of the power of the signal (P_{signal}) to the power of the noise (P_{noise}):

$$\mathrm{SNR} = 10 \cdot \log_{10}\left(\frac{P_{\text{signal}}}{P_{\text{noise}}}\right) \tag{3.22}$$

Where:

- P_{signal} is the power of the signal.
- P_{noise} is the power of the noise.

SNR is typically expressed in decibels (dB). Higher SNR values indicate better signal quality, while lower SNR values indicate a higher level of noise relative to the signal.

3.4.1.5 MAE

MAE stands for Mean Absolute Error. It is a metric used to evaluate the performance of a predictive model or the accuracy of an estimation technique.

In the context of image processing or machine learning, MAE measures the average magnitude of errors between predicted or estimated values and the actual values. It provides a simple and intuitive measure of the average deviation of predictions from the ground truth, without considering the direction of the deviations.

Mathematically, Mean Absolute Error is calculated as the average of the absolute differences between the predicted values ($\hat{y}_i$) and the actual values (y_i):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{3.23}$$

Where:

- n is the total number of samples or data points.
- $\hat{y}_i$ is the predicted value for the ith sample.
- y_i is the actual (true) value for the ith sample.

MAE provides a measure of the average magnitude of errors, with higher values indicating larger average errors and lower values indicating smaller average errors. It is commonly used alongside other evaluation metrics such as MSE (Mean Squared Error) and RMSE (Root Mean Squared Error) to assess the performance of models or algorithms.

3.4.1.6 NPCR

NPCR (Normalized Pixel Change Rate) is a metric used to evaluate the robustness of image encryption or watermarking algorithms. It measures the percentage of pixel value changes between two versions of the same image when a watermark or encryption operation is applied [74].

The NPCR value is calculated by comparing corresponding pixels in the original image and the watermarked/encrypted image. A high NPCR value indicates that a small change in the original image leads to a large change in the watermarked/encrypted image, which is desirable for robustness. Mathematically, NPCR is calculated as follows:

$$\text{NPCR} = N(C^1, C^2) = \sum_{I,J}\frac{D(i,j)}{T} \tag{3.24}$$

the value range for NPCR is between 0 and 100, the optimum value is 100.

3.4.1.7 UACI

UACI (Unified Average Changing Intensity) is another metric commonly used to evaluate the effectiveness of image encryption or watermarking algorithms. Similar to NPCR, UACI measures the average intensity change between corresponding pixels in the original and watermarked/encrypted images. Mathematically [44], UACI is calculated as follows:

$$\mathrm{NPCR} = U(C^1, C^2) = \sum_{I,J} \frac{|C^1(i,j) \times C^2(i,j)|}{F.T} \tag{3.25}$$

the value range for UACI is between 0 and 100, the optimum value is 100.

3.4.2 Security Analysis

The evaluation of watermarking techniques for medical images involves assessing their security aspects in addition to their performance metrics. Security analysis aims to determine the robustness of the watermarking scheme against various attacks and the potential vulnerabilities that may compromise the integrity and authenticity of medical images. Several performance metrics are commonly used to evaluate the security of watermarking techniques in medical imaging applications [13].

3.4.2.1 NC

The similarity between the extracted and original watermarks is calculated using NC, and the coefficient values range from 0 to 1. Mathematically, it may be expressed as

$$\mathrm{NC} = \frac{\sum_{I=1}^{X} \sum_{J=1}^{Y} (W_{orgij} \times W_{recij})}{\sum_{I=1}^{X} \sum_{J=1}^{Y} (W_{org-ij^2})} \tag{3.26}$$

the value range for NC is between 0 and 100, Ideally, NC=1 but 0.7 is acceptable

3.4.2.2 BER

BER stands for Bit Error Rate. It is a metric used to quantify the number of erroneous bits transmitted or received in a communication system compared to the total number of bits transmitted or received. BER is commonly used in digital communication systems to assess the quality of the transmission channel or the performance of the communication system itself [59].

Mathematically, BER is defined as the ratio of the number of bits received in error to the total number of bits transmitted:

$$\mathrm{BER} = \frac{\text{Number of bits received in error}}{\text{Total number of bits transmitted}} \tag{3.27}$$

A lower BER indicates better performance, as it means fewer errors occurred during transmission. BER is typically expressed as a decimal fraction or as a percentage.

In practical applications, BER is often measured experimentally by comparing the transmitted and received bit streams and counting the number of discrepancies. It is an important metric in digital communication systems, especially in systems where data integrity and reliability are critical, such as wireless communication, optical communication, and digital data storage systems.

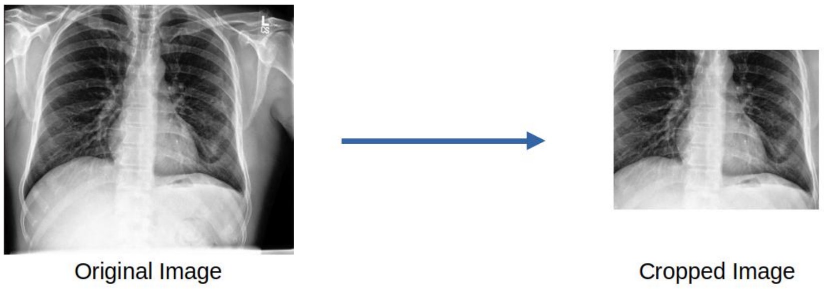


Figure 3.4: Example of Cropping attack.

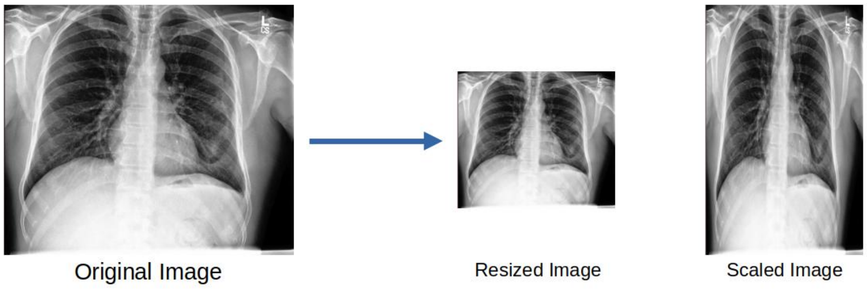


Figure 3.5: Example of Resizing and Scaling attack.

3.5 Attacks on Watermarked Images

In the context of image watermarking, attacks refer to various techniques or processes aimed at tampering with or removing watermarks from images. These attacks can be categorized into different types based on their objectives and methods. Here's a section discussing attacks in the context of image watermarking:

3.5.1 Removal Attacks

Removal attacks aim to completely eliminate or obscure the watermark from the image. These attacks often involve modifying the image in such a way that the watermark becomes undetectable or irrelevant [35]. Examples of removal attacks include:

3.5.1.1 Image Cropping

It is the process of removing undesirable regions from a image [35], an example was providing in the fig 3.4.

3.5.1.2 Image Resizing and Scaling

Resizing or Rescaling the image to change the watermark's size or aspect ratio, an example was providing in the fig 3.5.

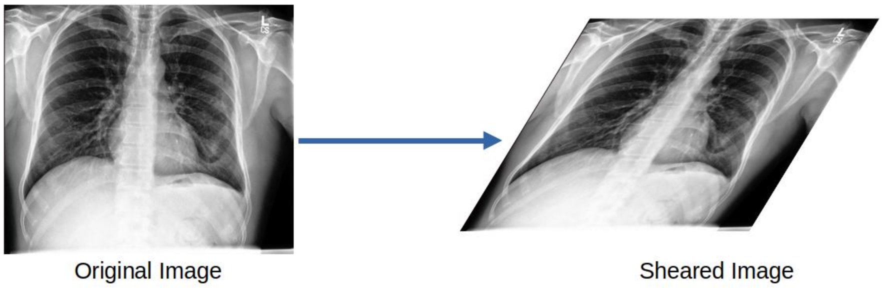


Figure 3.6: Example of Shearing attack.

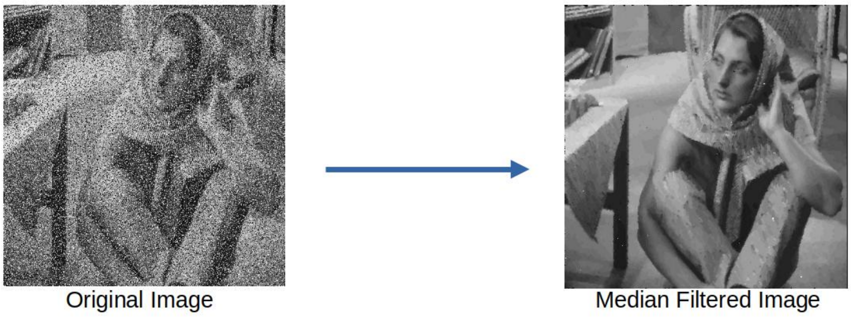


Figure 3.7: Example of Median Filter attack.

3.5.1.3 Shearing

A shearing attack is a form of geometric distortion applied to digital images. It involves transforming the image by displacing pixels along one axis, typically either horizontally or vertically, while keeping the other axis fixed. This displacement creates a sheared or skewed appearance in the image [35], an example was providing in the fig 3.6.

3.5.1.4 Image Filtering

Applying filters or image processing techniques to blur or remove the watermark[11], an example was providing in the fig 3.7.

3.5.1.5 Salt and Pepper

Salt and pepper noise, also known as impulse noise, is a type of image corruption where random pixels in the image are either set to the maximum intensity value (salt) or the minimum intensity value (pepper). This type of noise can occur due to errors in image acquisition or transmission, and it can significantly degrade the quality of the image [51].

The ”salt” pixels typically appear as bright white spots, while the ”pepper” pixels appear as dark black spots, an example was providing in the fig 3.8.

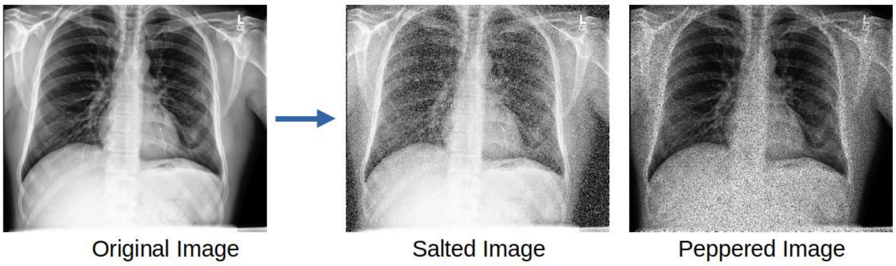


Figure 3.8: Example of Salt and Pepper attack.

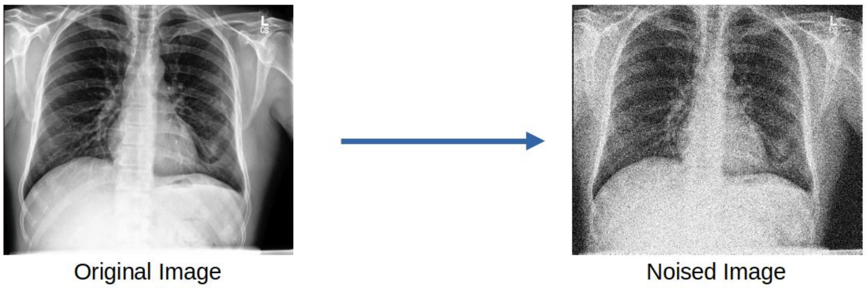


Figure 3.9: Example of Gaussian Noise attack.

3.5.1.6 Gaussian Noise

Gaussian noise, also known as additive white Gaussian noise (AWGN), is a type of statistical noise that is characterized by its Gaussian (bell-shaped) probability distribution. In the context of images, Gaussian noise appears as random variations in pixel intensity that follow a Gaussian distribution.

Gaussian noise is often added to images as an attack in the context of image processing or computer vision to simulate noise that can occur during image acquisition or transmission. It can model various sources of noise, including electronic sensor noise, thermal noise, and environmental factors [51].

Mathematically, Gaussian noise is generated by sampling from a Gaussian distribution with a specified mean (usually 0) and standard deviation (σ), where the mean represents the average intensity shift and the standard deviation represents the magnitude of the noise, an example was providing in the fig 3.9.

3.5.2 Geometric Attacks

Geometric attacks involve transforming or distorting the watermarked image to degrade the watermark's integrity. These attacks can include:

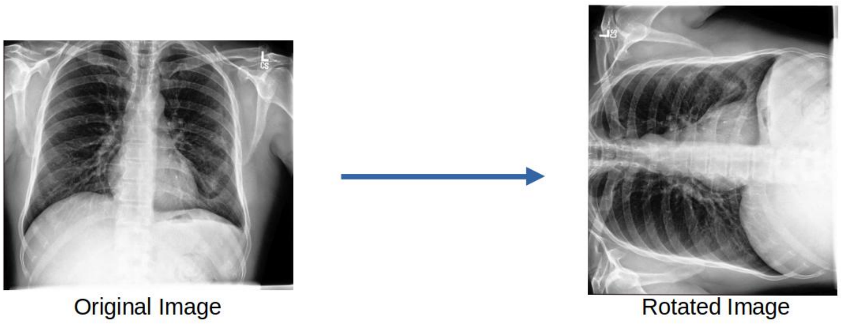


Figure 3.10: Example of Rotating attack.

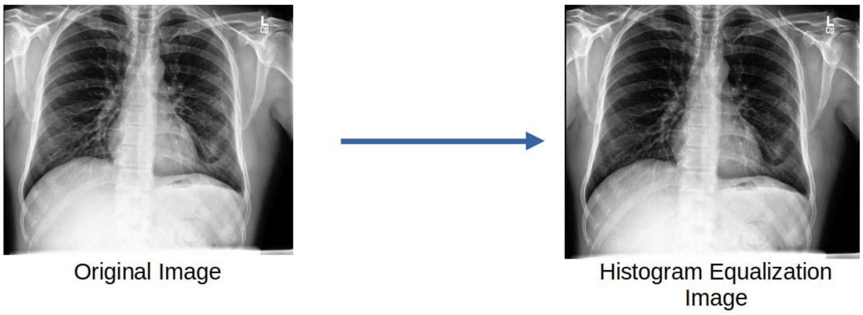


Figure 3.11: Example of Histogram Equalization attack.

3.5.2.1 Rotation

Rotating the image to alter the watermark's orientation [35], an example was providing in the fig 3.10.

3.5.2.2 Histogram Equalization

Histogram equalization is not typically referred to as an "attack"; instead, it's a method used in image processing to improve the contrast and overall appearance of an image. It's a technique used to adjust the contrast of an image by redistributing pixel intensities, an example was providing in the fig 3.11.

3.5.3 Content Attacks

Content attacks focus on modifying the image content while preserving the appearance of the original image. These attacks attempt to embed misleading information or artifacts into the image to undermine the watermark. Examples include:

3.5.3.1 Copy-Paste

Copying a region of the image containing the watermark and pasting it onto another part of the image, an example was providing in the fig 3.12.

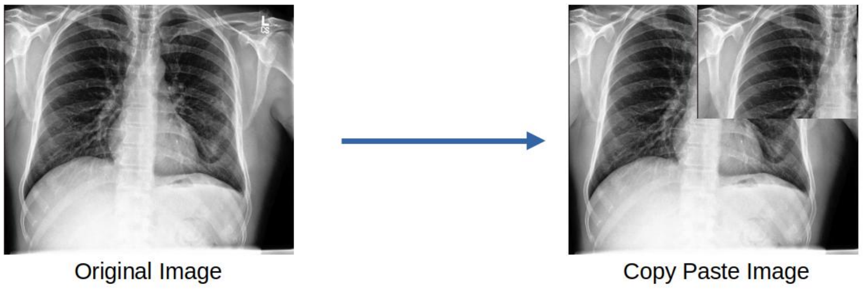


Figure 3.12: Example of Copy Paste attack.

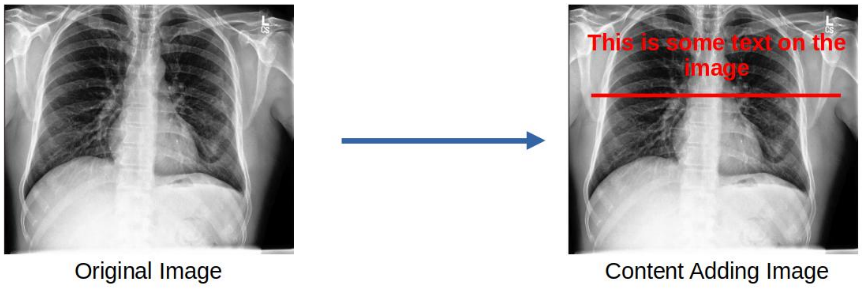


Figure 3.13: Example of Content Adding attack.

3.5.3.2 Content Addition

Adding new elements or content to the image to distract from or obscure the watermark, an example was providing in the fig 3.13.

3.5.4 Compression Attacks

Compression attacks exploit image compression algorithms to degrade the watermark's quality during compression and decompression processes. Common compression attacks include:

3.5.4.1 Lossy Compression

Using lossy compression algorithms that discard some image data, potentially affecting the watermark's visibility.
By deleting certain information that is not as perceptually significant, the lossy compression technique can shrink the file size of images. The JPEG compression algorithm is a popular approach for lossy compression.

JPEG scans every area of an image, identifying and eliminating everything that is difficult for your eyes to see.
Since human eyes are not flawless, JPEG takes advantage of these differences to eliminate information that our eyes struggle to perceive. For instance, the human eye contains two distinct types of light-receptive cells. cones and rods Your eyes are therefore far more sensitive to an image's brightness and darkness, or luminance,

and far less sensitive to its colors, or chrominance. Rods are not color sensitive and are essential for seeing in low light, whereas cones, with their color receptors for red, green, and blue, are color sensitive. Additionally, each eye contains 100 million rod cells compared to only six million cone cells.

Color Space Conversion: First, the input image is converted from the RGB (Red, Green, Blue) color space to the YCbCr color space. The image is divided into its chrominance (Cb and Cr) and brightness (Y) components using YCbCr. The human eye is more sensitive to changes in brightness than in color, which is why this distinction works [1].

$$\text{Y} = (0.299 \times R) + (0.587 \times G) + (0.114 \times B) \tag{3.28}$$

$$\text{Cb} = (-0.1687 \times R) + (-0.3313 \times G) + (0.5 \times B) + 128 \tag{3.29}$$

$$\text{Cr} = (0.5 \times R) + (-0.4187 \times G) + (-0.0813 \times B) + 128 \tag{3.30}$$

Where R: is Red value, G: is Green value and B: is Blue value. Each values is limited between 0 to 255.
There is no data loss during the conversion process, and the process is reversible.

Pheromones Down-Sampling: Down sampling involves taking the red and blue chrominance component images, dividing them into two by two blocks of pixels, calculating the average value for each block, eliminating repetitive information, and shrinking the image so that each average value of a four-pixel block occupies a single pixel. This reduces the information that the red and blue prominence component images, which our eyes are not very good at perceiving, to a quarter of their original size while maintaining the same luminance [1].
With that the image is half the size it was originally. It should be noted that the red and blue prominence images are rescaled to match the size of the luminance component when the image is reassembled. The rgb values are recalculated from the luminance blue chrominance and red chrominants, and since the luminance varies from pixel to pixel, so too can the rgb values.

Block Division: The image will be divided into 8 by 8 blocks, or blocks, in this step. Each block will have 64 pixels representing the luminance at each pixel, ranging from 0 to 255. Next, we will shift each value by subtracting 128 from each pixel, making the range negative 128 to 127, where negative 128 is black and 127 is white.

Discrete Cosine Transform (DCT): A two-dimensional DCT is used on each block in order to convert the spatial domain pixel values into frequency domain coefficients.
By combining these 64 base images, also known as the DCT Base Image (fig 3.14), we can rebuild any block of 64 pixels. Each base image is multiplied by a value or constant that indicates how much of it is used; as a result, the 64-pixel block that contains 64 values is changed into 64 values or constants that indicate how much of each base image is used [8].
Nothing in DCT truly shrinks or compresses the image; quantization, the following stage, achieves that.

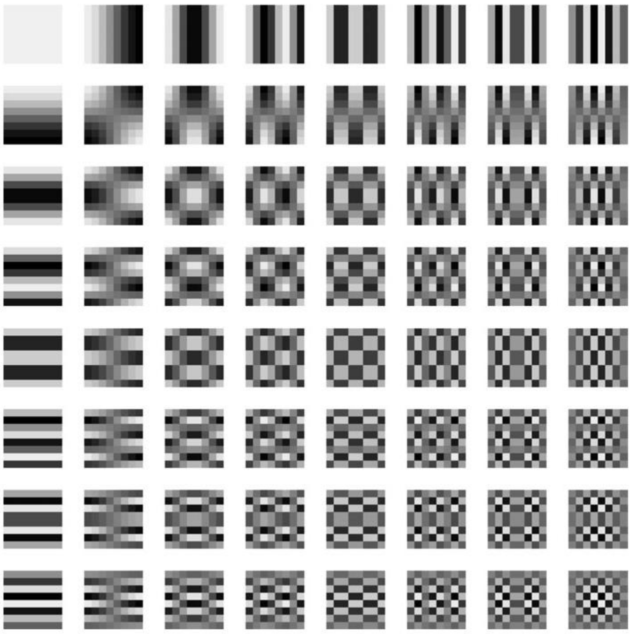
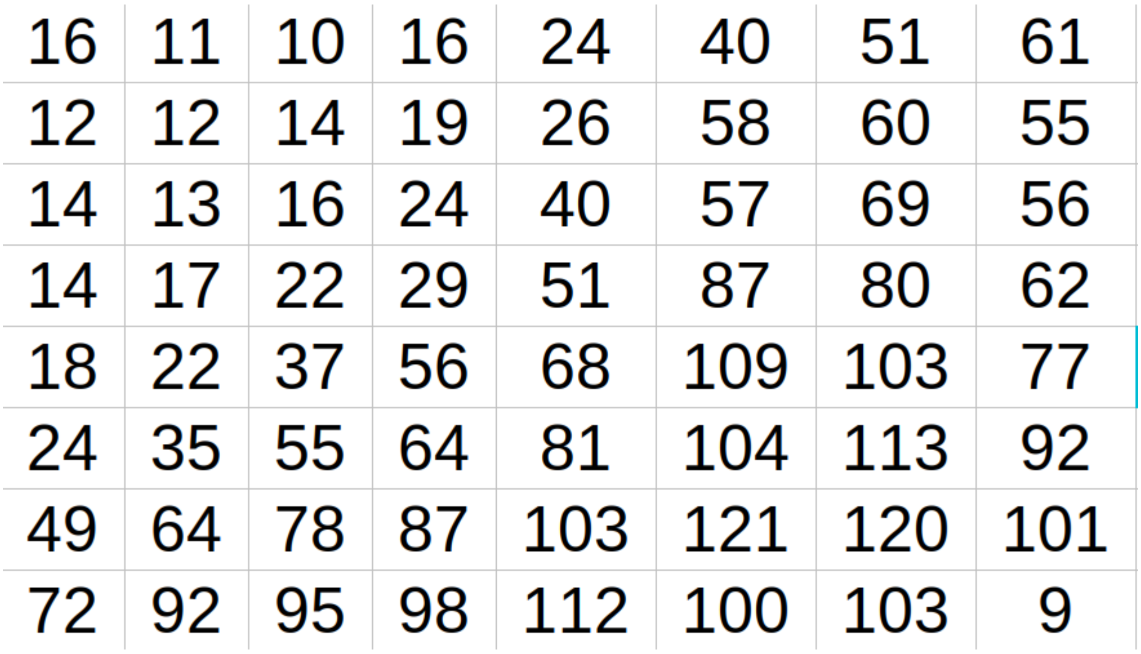
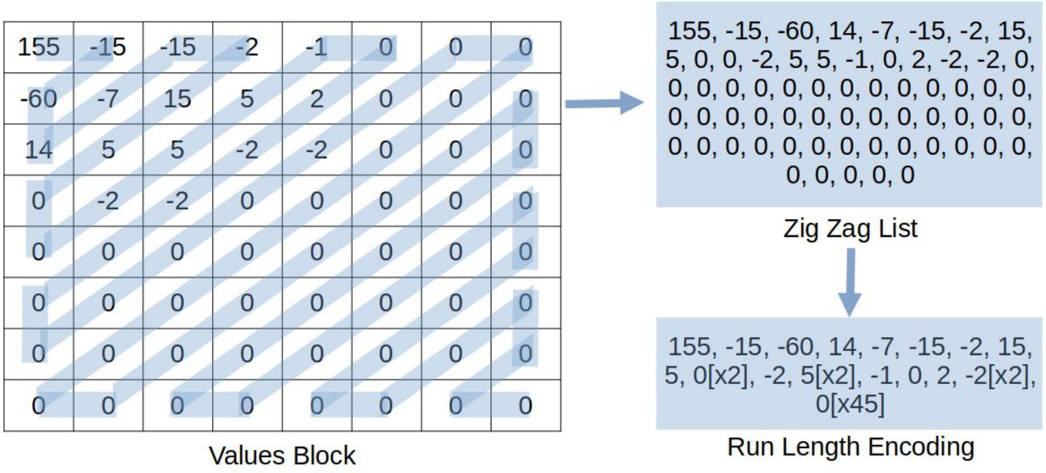


Figure 3.14: DCT 64 Base Block.



16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	9

Figure 3.15: Standard JPEG Quantisation Matrix.



155	-15	-15	-2	-1	0	0	0
-60	-7	15	5	2	0	0	0
14	5	5	-2	-2	0	0	0
0	-2	-2	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Values Block

155, -15, -60, 14, -7, -15, -2, 15, 5, 0, 0, -2, 5, 5, -1, 0, 2, -2, -2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

Zig Zag List

155, -15, -60, 14, -7, -15, -2, 15, 5, 0[x2], -2, 5[x2], -1, 0, 2, -2[x2], 0[x45]

Run Length Encoding

Figure 3.16: Example of Zig Zag List and Run Length Encoding.

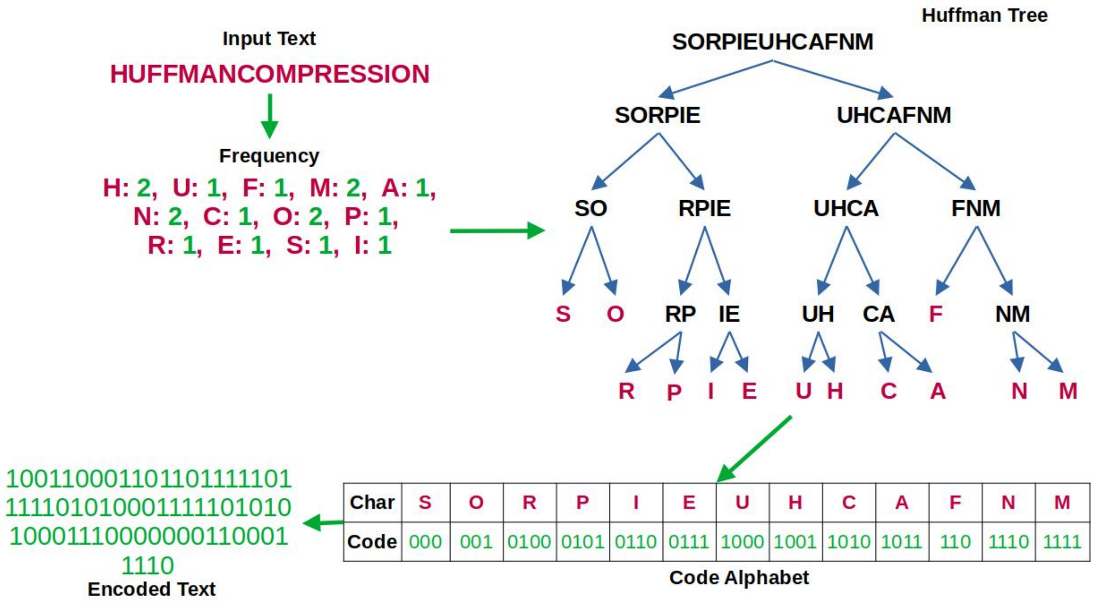
Quantization: To minimize the amount of bits needed to express the DCT coefficients, they are quantized. Each coefficient is quantized by dividing it by the matching value from a quantization matrix (fig. 3.15) and round it to integer. Compression is the result of information loss introduced by the quantization process, especially in higher-frequency components [1].

Run Length and Huffman Encoding: The values for each block in the luminance and prominence images are listed in this step using a zigzag pattern, as it is more likely that the non-zero numbers will be found up here. Next, we use a run length encoding algorithm, where the numbers are listed, and instead of listing all the zeros, we just state how many there are (fig 3.16).
As we can notice, this list of only a few dozen numbers is much more compressed than 64 pixels, which are each represented by a number ranging from zero to 255.

Finally, we utilize the Huffman encoding scheme, a well-liked method for lossless data compression that is widely applied in text and image compression methods like ZIP and JPEG. It achieves effective compression by allocating variable-length codes to distinct symbols according to their likelihood of occurring, with shorter codes denoting more often occurring symbols. (fig 3.17).

3.5.4.2 Quantization

Another method for compressing images is quantization, which limits the range of pixel values to a more manageable set of representative values.
In quantization Attacks we applying aggressive quantization to reduce the number of colors or levels in the image, which can degrade the watermark.



Char	S	O	R	P	I	E	U	H	C	A	F	N	M
Code	000	001	0100	0101	0110	0111	1000	1001	1010	1011	110	1110	1111

Figure 3.17: Example of Huffman Encoding.

3.6 Related Works

We covered a number of state-of-the-art image-based water-marking techniques in this section, including DCT and DWT, that have been applied to color, grayscale, medical, and other types of images. These techniques have been evaluated using a variety of performance matrices, including Bit Error Rate (BER) and Peak-Signal-to-Noise-Ratio (PSNR), among others. Numerous attacks, including image processing and geometric attacks, noise, filtering, manipulation, etc., have also been carried out to assess the resilience and imperceptibility of these methods for a range of applications, including copyright protection, owner identification and verification [31].

- A revolutionary blind color image watermarking approach was proposed by Su et al. [65]. In this case, the color image watermark is embedded into the color host image using QR decomposition techniques.

- A spatial domain blind color image watermarking system was presented by Su et al. [67] to protect color images' copyright. Here, the watermark is embedded using the largest Eigenvalue of the Schur decomposition achieved using algebraic operations.

- Su et al.'s [66] innovative watermarking method uses the color host image's DC and AC coefficients to safeguard the color image's copyright. The findings of the experiment demonstrate that the suggested method is more resilient to many attacks and exhibits improved imperceptibility.

- A spatial domain watermarking approach for the copyright protection of the color image has been presented and studied by Su et al. [64]. Here, a resilient watermarking strategy is devised based on the characteristics of the

DC coefficient of DFT blind.

- Hessenberg decomposition is the foundation of a revolutionary blind watermarking that Su [63] has described. Comparing the provided approaches to alternative methods based on QR decomposition or singular value decomposition, the latter two demonstrate higher computing complexity.

- A fully original self-implanting fragile watermarking method for photos has been proposed by Qin et al. [50]. This system, which was created for tampering recovery, is based on the adaption option of embedding mode and the reference-data interlocking method. The strategy is more effective than the rumored schemes, according to the experimental results.

- A strategy to fend off geometric and image modification attacks was put out by Roy et al. [52]. The cover image has no bearing on this technique. Its foundation is the key alone. The outcomes of the experiment show that this process is effective in straightforward situations. It is strong. Additionally, the cover's authenticity is preserved. Nevertheless, this approach fails when dealing with intricate attacks (such as cropping, rotation, and scaling). A comparative examination reveals that the projected technique outperforms a number of competing watermarking techniques in terms of both time demand and hardiness. As composite attacks that combine scaling, cropping, and rotating attacks, future study might focus on improving the anticipated technique for including improved resistance to compression attacks.

- Block-based DCT constant modification was assisted by the powerful blind watermarking technique developed by Parah et al. [45]. According to the experimental results, the predicted scheme's common PSNR price is 41.25 dB, which is superior to other state-of-the-art methods. Furthermore, the values of Normalized Correction (NC) are higher than [91, 99].

- A wholly original strong blind color image watermarking technique was planned by Huynh et al. [23]. In order to incorporate a grayscale watermark into a color host image, a quantization technique within the ripple domain is used to convert the grayscale image to binary photos from LSB to MSB. The testing results demonstrate that the planned methodology outperforms the remaining state-of-the-art in terms of the physical quality of embedded host photos and the robustness of extracted watermarks.

- Four blind watermarking techniques were planned and contrasted by Aggarwal et al. [3]. For a thorough examination of the pro- exhibit watermarking algorithms S1, S2, S3, and S4, the author has conducted eight trials. According to the experimental result, S3 and S4 have greater PSNRs than [36][37], but S1 and S2 have lower PSNRs.

- A robust blind watermarking technique was planned by Vaidya & PVSSR [48] using the Bhat-Tacharyya distance and a mathematical function. When the experimental results are compared to the state-of-the-art, the outcome is superior.

- For photos taken using a camera, Thongkor et al. [70] proposed a digital watermarking method. In this approach, a watermark bit was embedded in each pixel of the host image to embed a binary image of the same size. The suggested scheme has an average wPSNR of 35dB and an average SSIM value of 0.93.

- Lai [34] designed a technique that was mostly based on the Tiny-Genetic algorithm and single value decomposition (SVD). Here, the watermark is embedded via an adaptation to the cover image's single values. The findings of the simulation demonstrate that the embedded watermark is resilient enough to withstand attacks or image processing processes, and that this planned strategy outperforms other comparable approaches in terms of hardiness.

- A contourlet transformation & quantization index modulation largely based watermarking technology was developed by Najih et al. [41]. Additionally, the Lagrange approach was applied for optimization. Experiments demonstrate greater physical property utility, transparency, and smart capacity in addition to offering superior robustness compared to alternative strategies for various attacks.

- Image security was addressed in a proposal made by Sarreshtedari & Akhaee [56]. The source channel was to be encoded. Its foundations were set partitioning in hierarchical transforms (SPIHT) and Reed-Solomon (RS). The encoder bits, which are utilized for content recovery, are found in the first segment. Parity bits are found in the second part, and check bits are found in the last zone. The experiment's outcome demonstrates the suggested method's effectiveness and superiority over alternative approaches.

Table 3.1 presents a comparative examination of the state-of-the-art in image watermarking, while Tables [3.2, 3.3, 3.4, 3.5] is a summary of the state-of-the-art in this field [31].

Table 3.1: Comparative analysis of watermarking images.

Ref	Domain	Robust	Inperceptiblity	Capacity	Blind	Watermark Type	Objective
[65]	Transform	Yes	Yes	Low	Yes	Color Image	—
[67]	Special	Yes	Yes	Low	Yes	Color Image	Copyright Protection
[66]	Transform	Yes	Yes	Low	No	Color Image	Copyright Protection
[64]	Special	Yes	Yes	—	Yes	Color Image	Copyright Protection
[63]	Transform	Yes	Yes	High	Yes	Color Image	Copyright Protection
[50]	Special	No	Yes	High	No	Random Sequence	Tamper Detection
[52]	Special	Yes	Yes	—	Yes	Binary Image	Models geometric attck on watermarked images
[45]	Transform	Yes	Yes	High	Yes	Binary Image	—
[23]	Transform	Yes	Yes	Low	Yes	Grayscale Image	Robust blind image color watermarking technique
[3]	Transform	Yes	Yes	—	Yes	Grayscale Image	Owner identification
[48]	Dual	Yes	Yes	—	Yes	Binary Image	Copyright Protection
[70]	Spatial	Yes	Yes	Dependent	No	Binary Image	Copyright Protection
[34]	Transform	Yes	Yes	Dependent	No	Grayscale Image	To find optimal scale factor
[41]	Transform	Yes	Yes	High	No	Random Sequence	Content authentication
[56]	Spatial	No	Yes	High	No	—	Detection

Table 3.2: An overview of the recent state-of-the-art image watermarking methods

Ref	Purpose	Techniques Used	Input	Performance Metrics	Attack Performed	Remark
[65]	Create a scheme for watermarking a blind color image.	QR Decomposition	Cover color image: 512x521, Color watermark image: 32x32	NC, PSNR & SSIM	JPEG 2000 Compression, Geometric attacks & common image processing attacks.	resilient against attacks from geometry and traditional image processing.
[67]	Preserve color images' copyright.	Schur decomposition based spaital domain blind color image watermarking, two-level DCT	Cover color image: 512x512, color watermark image: 32x32	PSNR, SSIM & NC	JPEG Compression, Geometric attacks & common image processing attacks	Gain from robustness and performance in real-time.
[66]	Robust color image watermarking	Two-level DCT	Cover color image: 512x512, Color watermark image: 64x64	PSNR, SSIM & NC	JPEG compression, JPEG 2000 compression, cropping, adding noise, scaling, low-pass filtering, median filtering, rotation, blurring	The suggested method's capability is somewhat greater than the state-of-the-art.
[64]	Preserve color images' copyright.	Features of the DC coefficient of 2D-DFT	Cover color image: 512x512, Color watermark image: 32x32	PSNR, SSIM & NC	Standard benchmark optimal software	The suggested approach is highly robust and has a low running time.
[63]	Preserve color images' copyright.	Hessenberg Decomposition	Cover color image: 512x512 Color watermark image: 32x32	PSNR, SSIM & NC	JPEG 2000 Compression, Geometic attacks & common image processing attacks, Tampering	Compared to existing methods, the proposed method has a lower computational complexity.

Table 3.3: Continued

Ref	Purpose	Techniques Used	Input	Performance Metrics	Attack Performed	Remark
[50]	Tampering recovery	Self-embeding fragile watermarking based on reference-data interleaving adaptive selection of embedding mode	Image size: 512x512	PSNR, SSIM	Tampering	To demonstrate the efficacy of their plan, content recovery for intentional medding and unintentional meddling of block missing within the wireless attenuation channels is simulated.
[52]	Stand against Grometric	Blind image technique	Cover color image: 631x833, Watermark image: 100x91	BER, PSNR & SSIM	Geometric attacks & common image processing attacks.	High visual quality of the cover with watermark. requires increased defense against compression attacks.
[45]	Robust and Blind	DCT, inter-block Coefficient Difference	Color & Gray scale image 512x512 Binary watermark image 64x64	PSNR, BER & NC	Geometric attack, singular and hybrid attacks	robust against both single and multiple hybrid attacks. ability to produce superb watermarked images
[23]	Robust Blind color image watermarking	Selective bit embedding scheme	Eight: 512x512 color images. Four: 64x64 grayscale watermark image	PSNR, SSIm, NC, CPSNR & NCC	Average & Median Filtering. Motion Blurring, Rotation & Cropping, Salt & Pepper Noise & Gaussian Noise JPEG Compression (lossy)	Strong and performs well in terms of embedded host image imperceptibility

Table 3.4: Continued

Ref	Purpose	Techniques Used	Input	Performance Metrics	Attack Performed	Remark
[3]	Owner identification/verfication technology	RDWT & DWT	8-bit gray scale image 512x512 8-bit gray scale face image as watermark image 64x64	PSNR, NC	Cropping from center, Gaussian filtering, salt and pepper noise rotation, JPEG compression Resize	The effectiveness of watermarking techniques has significantly improved thanks to weighted binary coding.
[48]	Ownership identification and copyright protection	Bhattacharyya distance and bit manipulation, DWT	Host image: 512x512 Watermark image 64x64	PSNR, NCC	Salt & Pepper Noise, Mean filtering, Gaussian noise, Median filtering, Cropping, Scaling, Speakle Noise, Rotation, Blurring Translate & JPEG compression	withstand many attacks using signal and image processing; note the little differences between NCC and PSNR
[70]	Protect Camera-Captured Images	Wiener filtering and low noticeable distortion	Image by -DSLR camera - Compact camera - Camera phone	RMSE, PSNR, NC and SSIM	-	Robust against a variety of threats, accomplish both trustworthy watermark extraction from a printed
[34]	Improvements of the Scaling Factors that are used ot control the strength of the embedded watermark	SVD & tiny GA	Cover iamge: 256x256, Watermark image: 64x64 gray-level	NC	Image processing attack	provides a means of enhancing the Watermark scaling factor.

Table 3.5: Continued

Ref	Purpose	Techniques Used	Input	Performance Metrics	Attack Performed	Remark
[41]	Authentication over an unknown channel and imperceptibility	Angle Quantizaion in Discrete Contourlet Transform	Cover & Watermark image: 512x512 pix, 262,144 samples	NCC & PSNR etc.	Gaussian noise, Salt & Pepper noise, Poison noise, Speckle noise, Gaussian motion , Median & Weiner Histogram equalization, Cropping, Rotation, Resizing (512-256-512). JPEG (Q=1) Gray scale inversion, Gamma corrector 1.6 Gaussian blur.	Improved resilience, enhanced effectiveness, increased transparency, and strong capacity.
[56]	Detecting the Tampered are of the Received image and Recovering the Lost information in the Tampered Zones.	Source-channel coding, SPIHT encoding, RS code	Image 8-bit grayscale: 512x512	TTR & PSNR	Tampering	Compared to previous schemes, effective in the event of channel parity bit tampering

3.7 Conclusion

In conclusion, research on medical image watermarking (MIW) techniques and efficacy emphasizes how crucial it is to protect the confidentiality, accuracy, and integrity of medical images. Each method has its own advantages and disadvantages, including the Least Significant Bit (LSB), Discrete Cosine Transform (DCT), and Discrete Wavelet Transform (DWT). While DCT and DWT offer more imperceptibility and resilience at the expense of increased computational complexity, LSB is less resilient despite being simpler and more efficient.

Metrics like PSNR, SSIM, NC, and BER are useful for assessing the efficacy of watermarking techniques and for striking a balance between computing needs and robustness. The ongoing advancement of sophisticated approaches that combine several strategies or cryptographic components holds potential for improving MIW's security and resilience.

All things considered, continuous improvements in MIW methods seek to safeguard medical images without sacrificing their diagnostic quality. Subsequent investigations ought to concentrate on enhancing these methods to augment their resilience, safety, and effectiveness, therefore endorsing the safe management of medical images in healthcare.

Chapter 4

Experiment and Results discussion

4.1 Introduction

In this chapter, we outline the methods employed in our study of watermarking techniques for medical images along with the resources and results. Our objective is to provide a comprehensive overview of the approaches and tools used in our investigation.

4.2 Hardware Setup

The experiments and analyses conducted in this research were performed on a workstation with the following specifications:

4.2.1 Operating System

- OS: Linux Mint 21.3 x86_64.
- Kernel: 5.15.0-107-generic.

4.2.2 Host and Environment

- Host: Lenovo ThinkPad T590 (20N5S8L200).
- Resolution: 1920x1080.
- Desktop Environment: Cinnamon 6.0.4.
- Window Manager: Mutter (Muffin).
- Window Manager Theme: Adapta-Nokto (Mint-Y).
- Theme: Adapta-Nokto [GTK2/3].
- Icons: Breeze-Dark [GTK2/3].

4.2.3 Terminal and Shell

- Terminal: Gnome Terminal.
- Shell: Bash 5.1.16.

4.2.4 Hardware Specifications

- CPU: Intel i5-8365U (4 cores, 8 threads) @ 4.100GHz.
- GPU: Intel WhiskeyLake-U GT2 [UHD Graphics 620].
- Memory: 16GB.

4.3 Programming Language

Python has emerged as a popular programming language in the field of image processing research due to its versatility, ease of use, and extensive libraries tailored for scientific computing [40]. In our study of watermarking techniques for medical images, we have leveraged Python for various aspects of our research, including algorithm implementation, data manipulation, visualization, and evaluation [71].

Python boasts an extensive ecosystem of libraries specifically designed for image processing and computer vision tasks. Libraries such as OpenCV, scikit-image, and Pillow provide comprehensive functionality for reading, writing, manipulating, and analyzing images. These libraries offer a rich set of functions for tasks such as image filtering, transformation, feature extraction, and visualization, allowing researchers to efficiently implement and experiment with different watermarking techniques.

4.4 Python Packages

Our research on watermarking techniques for medical images relies on a variety of Python packages tailored for image processing, data manipulation, visualization, and evaluation. In this section, we provide an overview of the key Python packages we utilized and their roles in our research.

4.4.1 OpenCV (Open Source Computer Vision Library)

OpenCV is a widely-used open-source library for computer vision and image processing tasks. It provides a comprehensive set of functions and algorithms for reading, writing, manipulating, and analyzing images and videos. In our research, we leveraged OpenCV for tasks such as image loading, color space conversion, filtering, feature extraction, and geometric transformations, OpenCV Documentation.

4.4.2 Pillow (Python Imaging Library)

Pillow is a fork of the Python Imaging Library (PIL), offering support for opening, manipulating, and saving many image file formats. We utilized Pillow for image file I/O operations, format conversion, resizing, cropping, and other basic image processing tasks in our research pipeline, Pillow Documentation.

4.4.3 NumPy (Numerical Python)

NumPy is a fundamental package for scientific computing in Python, providing support for multidimensional arrays, linear algebra, mathematical functions, and random number generation. We utilized NumPy extensively for representing images as arrays, performing array operations, and implementing mathematical operations required for various watermarking algorithms, NumPy Documentation.

4.4.4 Matplotlib

Matplotlib is a plotting library for creating static, interactive, and animated visualizations in Python. It offers a wide range of plotting functions and customization options for generating publication-quality figures and plots. In our research, we employed Matplotlib for visualizing images, histograms, signal waveforms, and performance metrics to analyze and interpret experimental results, Matplotlib Documentation.

4.4.5 Tkinter

Tkinter is the standard GUI (Graphical User Interface) toolkit for Python. It is included with most Python installations, so there's no need to install it separately, Tkinter Documentation.
Tkinter provides a fast and easy way to create GUI applications in Python, allowing developers to create windows, dialogs, buttons, menus, and more, with minimal effort. Its simplicity and ease of use make it a popular choice for building desktop applications with Python.

4.5 Dataset Discription

The success of any research in medical image processing heavily relies on the quality and diversity of the dataset used for experimentation and evaluation. In our study on watermarking techniques for medical images, we utilized a database of chest X-ray images for COVID-19 positive cases, along with normal and viral pneumonia images, has been created by a team of researchers from Qatar University, Doha, Qatar, and the University of Dhaka, Bangladesh, along with their collaborators from Pakistan and Malaysia in collaboration with medical professionals.

It is composed of 33,920 chest X-ray (CXR) images. For the full dataset, 10,701 Normal Ground-truth lung segmentation masks are provided, along with 11,956 COVID-19 and 11,263 Non-COVID infections (Viral or Bacterial Pneumonia), COVID-19 Radiography Database.
In our study, we included 200 images from each group, for a total of 800 medical images used in the testing phase.

4.6 Algorithm's used in the study

In this section we provide an overview of the main techniques we used in our research to watermark medical images. These algorithms are essential for both embedding and extracting watermarks and for assessing how effectively watermarking methods work.

4.6.1 Least Significant Bit (LSB)

LSB substitution is a quick and efficient way to include data into digital images. This method replaces the least significant pixel values in the image with portions of the watermark message. We employ LSB substitution as one of the watermarking techniques in our analysis.
In our study, we used 3 types of Least Significant Bit (LSB) techniques, each with a version to embed the watermark as a message and as an image.

4.6.1.1 LSB Embedding with RGB Pixels

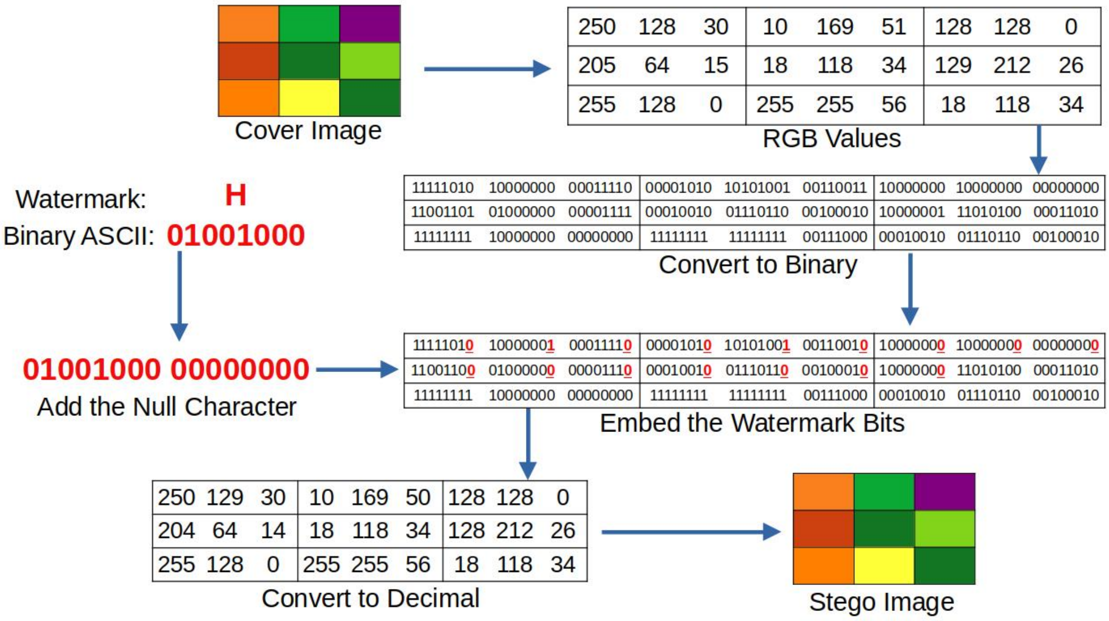
Three color channels—Red, Green, and Blue—represent each pixel in an image according to the RGB color paradigm. With eight bits in each color channel, there are 256 different levels of intensity for every color. In LSB embedding, a portion of the watermark message is substituted for each color channel's least significant bit (LSB).
In this technique each Pixel from the Cover Image will provide 3 bits of space to embed the watermark on it, the total space could be calculated with this formula:

$$\text{LSB RGB Embedding Space} = W \times H \times 3 \tag{4.1}$$

with W: represent the width of the Cover image, and H: represent the height of the Cover image.

Text Watermark: To embed a Text Watermark using this technique, first we extract the color channels of each pixel from the cover image, then we convert the Text Watermark into the ASCII binary format. After that we loop throw each color channel from each pixel in the cover image and loop throw each bit from the Binary Text Watermark with change the LSB from each color channel with the bit from the watermark, after embedding all the characters we embed the NULL character to stop the extracting process later, the null character represented as 8 Zeroes. the process steps are shown in the (fig 4.1).
Each character from the Text Watermark will take a space of 8 bits in the ASCII binary format, so each character will need approximately 3 pixels from the Cover image to be embedded.

Image Watermark: To embed an Image Watermark using this technique, first we extract the color channels of each pixel from the cover image, then we extract the color channels of each pixel from the watermark image, then we resize the Watermark Image to certain size to be able to extracted later, after that we convert



RGB Values

250	128	30	10	169	51	128	128	0
205	64	15	18	118	34	129	212	26
255	128	0	255	255	56	18	118	34

Convert to Binary

11111010	10000000	00011110	00001010	10101001	00110011	10000000	10000000	00000000
11001101	01000000	00001111	00010010	01110110	00100010	10000001	11010100	00011010
11111111	10000000	00000000	11111111	11111111	00111000	00010010	01110110	00100010

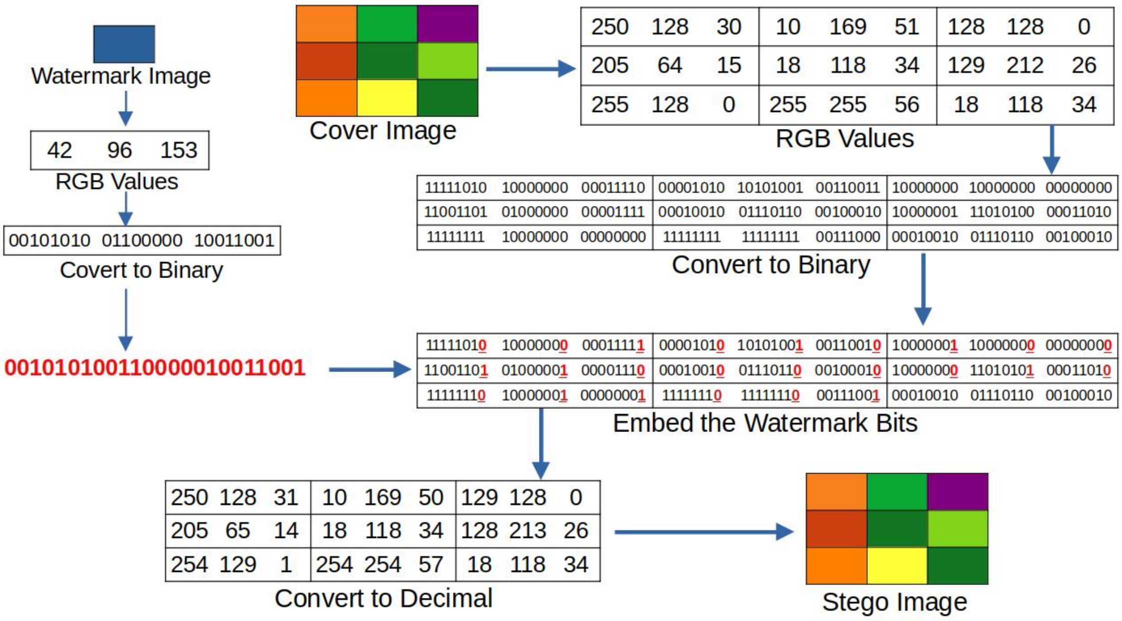
Embed the Watermark Bits

11111010	10000001	00011110	00001010	10101001	00110010	10000000	10000000	00000000
11001100	01000000	00001110	00010010	01110110	00100010	10000000	11010100	00011010
11111111	10000000	00000000	11111111	11111111	00111000	00010010	01110110	00100010

Convert to Decimal

250	129	30	10	169	50	128	128	0
204	64	14	18	118	34	128	212	26
255	128	0	255	255	56	18	118	34

Figure 4.1: LSB RGB Text Algorithm



Watermark RGB Values

42	96	153
00101010	01100000	10011001

RGB Values

250	128	30	10	169	51	128	128	0
205	64	15	18	118	34	129	212	26
255	128	0	255	255	56	18	118	34

Convert to Binary

11111010	10000000	00011110	00001010	10101001	00110011	10000000	10000000	00000000
11001101	01000000	00001111	00010010	01110110	00100010	10000001	11010100	00011010
11111111	10000000	00000000	11111111	11111111	00111000	00010010	01110110	00100010

Embed the Watermark Bits

11111010	10000000	00011111	00001010	10101001	00110010	10000001	10000000	00000000
11001101	01000001	00001110	00010010	01110110	00100010	10000000	11010101	00011010
11111110	10000001	00000001	11111110	11111110	00111001	00010010	01110110	00100010

Convert to Decimal

250	128	31	10	169	50	129	128	0
205	65	14	18	118	34	128	213	26
254	129	1	254	254	57	18	118	34

Figure 4.2: LSB RGB Image Algorithm

each color channel from each pixel in the resized watermark into the binary format and concatenate them into one long string of binary values. After that we loop throw each color channel from each pixel in the cover image and loop throw each bit from the Binary Watermark Image with changing the lsb of the binary cover image pixel with the corresponding bit form the binary watermark, the process steps are shown in the (fig 4.2).
Each pixel from the Watermark Image will be represented as 3 color channels, and each color channel will take a space of 8 bits in the binary format, so each Pixel form the Watermark Image will need 8 pixels from the Cover image to be embedded.

4.6.1.2 LSB Gray Scale Embedding

The procedure for LSB embedding using grayscale pixels is the same as for RGB LSB embedding, but it uses a single intensity channel rather than three different color channels. In grayscale images, a single intensity value between 0 (black) and 255 (white) represents each pixel. In order to encode the watermark data in grayscale photos, LSB embedding entails adjusting the least significant bit of each pixel's intensity value. this formula shows how to convert and RGB pixel into a Gray Scale pixel:

$$\text{Gray Scale Value} = (0.299 \times \mathit{Red}) + (0.587 \times \mathit{Green}) + (0.114 \times \mathit{Blue}) \tag{4.2}$$

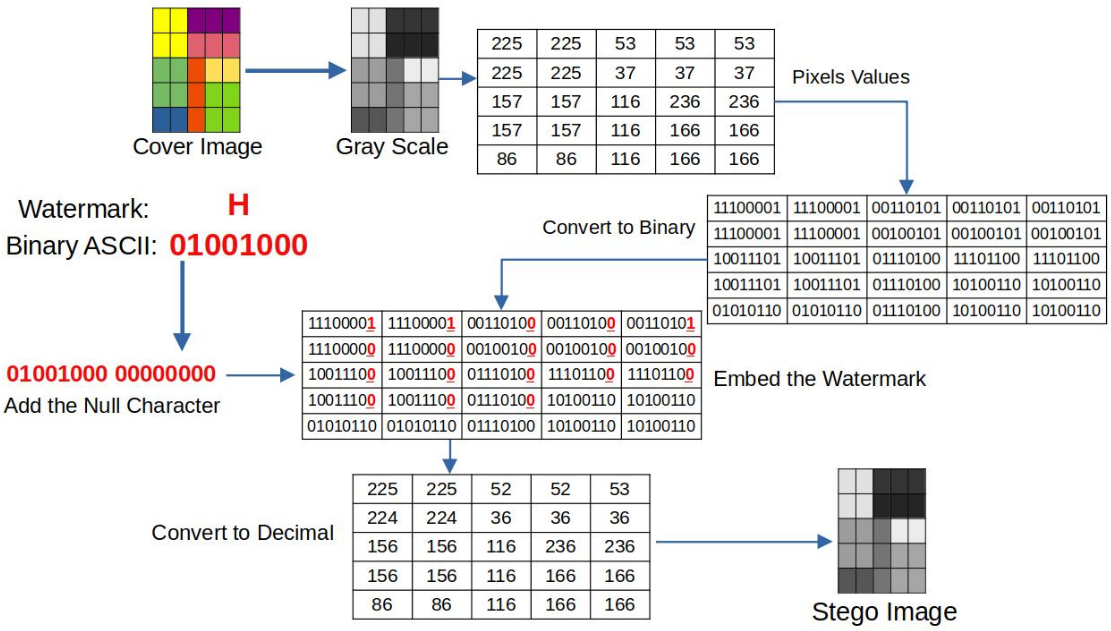
Since each pixel in a grayscale image only has one intensity channel, the embedding space is only as big as the image resolution. So In this technique each Pixel from the Cover Image will provide 1 bits of space to embed the watermark on it, the total space could be calculated with this formula:

$$\text{LSB Gray Scale Embedding Space} = W \times H \tag{4.3}$$

with W: represent the width of the Cover image, and H: represent the height of the Cover image.

Text Watermark: To embed a Text Watermark using this technique, first we load the cover image as gray scale format, then extract the pixel values, then we convert the Text Watermark into the ASCII binary format. After that we loop throw each pixel value in the cover image and loop throw each bit from the Binary Text Watermark with changing the LSB from each pixel value with the bit from the watermark, after embedding all the characters we embed the NULL character to stop the extracting process later, the null character represented as 8 Zeroes. the process steps are shown in the (fig 4.3).
Each character from the Text Watermark will take a space of 8 bits in the ASCII binary format, so each character will need approximately 8 pixels from the Cover image to be embedded.

Image Watermark: To embed an Image Watermark using this technique, first we load the Cover image as gray scale format, and also we do the same for the



Pixels Values

225	225	53	53	53
225	225	37	37	37
157	157	116	236	236
157	157	116	166	166
86	86	116	166	166

Convert to Binary

11100001	11100001	00110101	00110101	00110101
11100001	11100001	00100101	00100101	00100101
10011101	10011101	01110100	11101100	11101100
10011101	10011101	01110100	10100110	10100110
01010110	01010110	01110100	10100110	10100110

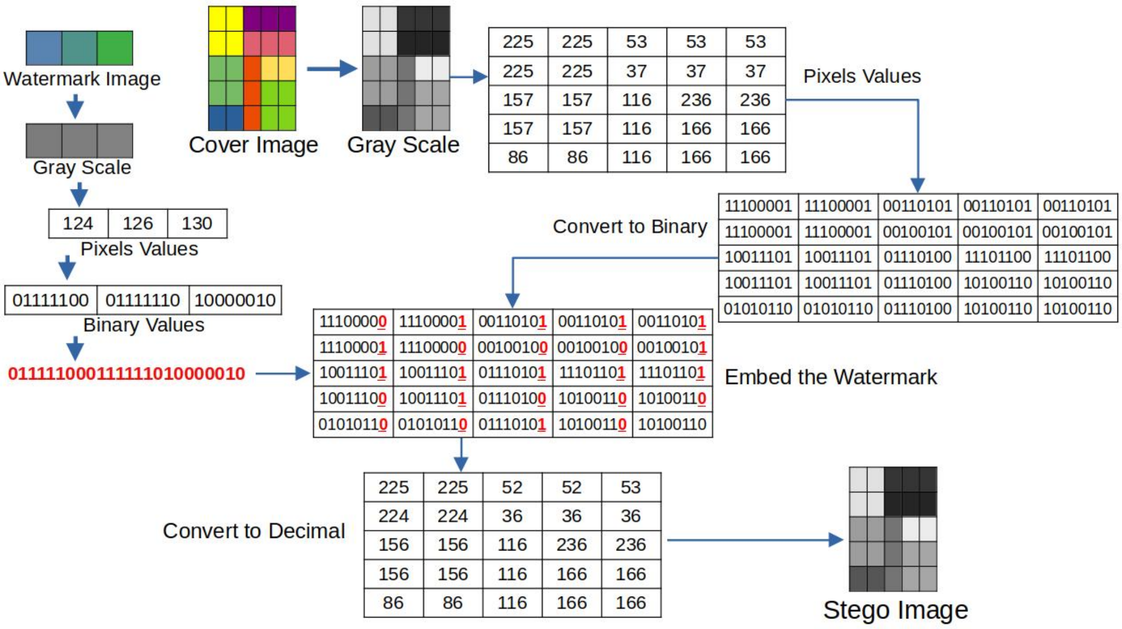
Embed the Watermark

11100001	11100001	00110100	00110100	00110101
11100000	11100000	00100100	00100100	00100100
10011100	10011100	01110100	11101100	11101100
10011100	10011100	01110100	10100110	10100110
01010110	01010110	01110100	10100110	10100110

Convert to Decimal

225	225	52	52	53
224	224	36	36	36
156	156	116	236	236
156	156	116	166	166
86	86	116	166	166

Figure 4.3: LSB Gray Scale Text Algorithm.



Pixels Values

124	126	130

Binary Values

01111100	01111110	10000010

Pixels Values

225	225	53	53	53
225	225	37	37	37
157	157	116	236	236
157	157	116	166	166
86	86	116	166	166

Convert to Binary

11100001	11100001	00110101	00110101	00110101
11100001	11100001	00100101	00100101	00100101
10011101	10011101	01110100	11101100	11101100
10011101	10011101	01110100	10100110	10100110
01010110	01010110	01110100	10100110	10100110

Embed the Watermark

11100000	11100001	00110101	00110101	00110101
11100001	11100000	00100100	00100100	00100101
10011101	10011101	01110101	11101101	11101101
10011100	10011101	01110100	10100110	10100110
01010110	01010110	01110101	10100110	10100110

Convert to Decimal

225	225	52	52	53
224	224	36	36	36
156	156	116	236	236
156	156	116	166	166
86	86	116	166	166

Figure 4.4: LSB Gray Scale Image Algorithm.

Table 4.1: Sample Values Redistribution Condition.

Sample Values Reorganization	Watermark Bits
Min, Avr, Max	0, 0
Max, Avr, Min	1, 1
Min, Max, Avr	0, 1
Max, Min, Avr	1, 0

Watermark image, then we resize the Watermark Image to certain size to be able to extracted later, after that we convert each pixel from the resized watermark into the binary format and concatenate them into one long string of binary values. After that we loop throw each pixel in the binary cover image and loop throw each bit in the Binary Watermark Image with changing the lsb of the binary cover image pixel with the corresponding bit form the binary watermark, the process steps are shown in the (fig 4.4).

Each pixel from the Watermark Image will be represented as one color channels with value between 0 and 255, and the color channel will take a space of 8 bits in the binary format, so each Pixel form the Watermark Image will need 8 pixels from the Cover image to be embedded.

4.6.1.3 LSB (Min Avr Max) Method

The LSB (Min Avr Max) method involves comparing three successive sample values; the results of this comparison will yield an average, a maximum, and a minimum element. The watermark's components will be divided into two groups, resulting in four possible combinations: 00, 01, 10, and 11. Then, as indicated in Table 4.1, the values (minimum, maximum, and average) will be re-distributed in accordance with the watermark bits that will be introduced. Two bits are hidden in the suggested watermarking approach using three sample values. Let Min, Max, and Avr represent the lowest, highest, and average values of three consecutive sample values, respectively [53].

In case of getting 2 or all of the samples equal we change the value of some of them by adding or subtracting 1 or 2, for example if we get (0,0,0) we change it with (0,1,2).

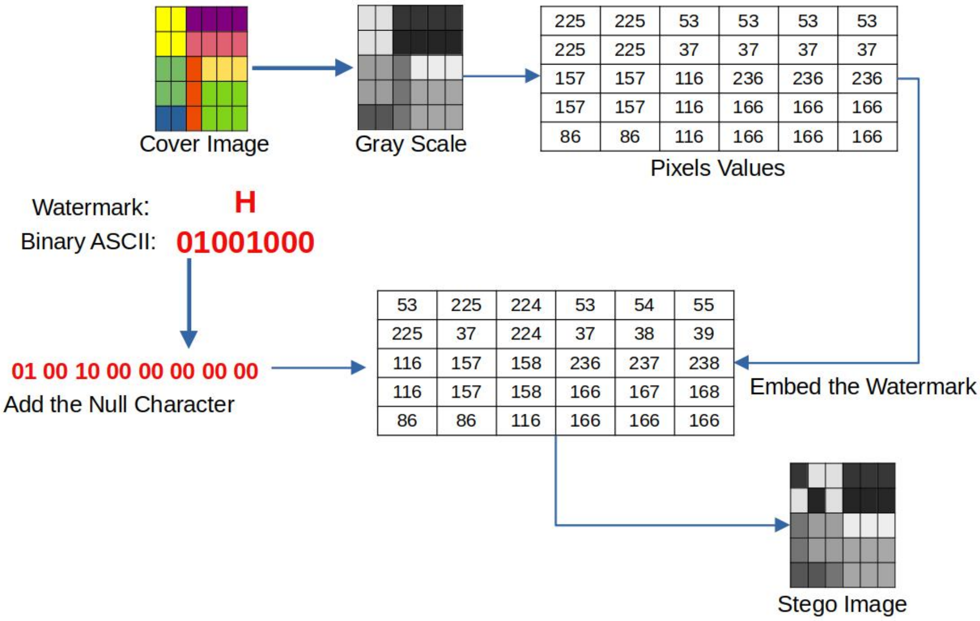
The embedding space of this technique could be calculated with this formula:

$$\text{LSB (Min Avr Max) Embedding Space} = \frac{(W \times H) \times 2}{3} \tag{4.4}$$

with W: represent the width of the Cover image, and H: represent the height of the Cover image.

The watermarked file's sample values are extracted by first grouping them into three groups and comparing them to find the lowest, maximum, and average values. In accordance with the guidelines in Table 4.1, the received order permits the extraction of two bits [53].

Text Watermark: To embed a Text Watermark using this technique, first we load the cover image as gray scale format, then we extract the pixel values, then we



225	225	53	53	53	53
225	225	37	37	37	37
157	157	116	236	236	236
157	157	116	166	166	166
86	86	116	166	166	166

53	225	224	53	54	55
225	37	224	37	38	39
116	157	158	236	237	238
116	157	158	166	167	168
86	86	116	166	166	166

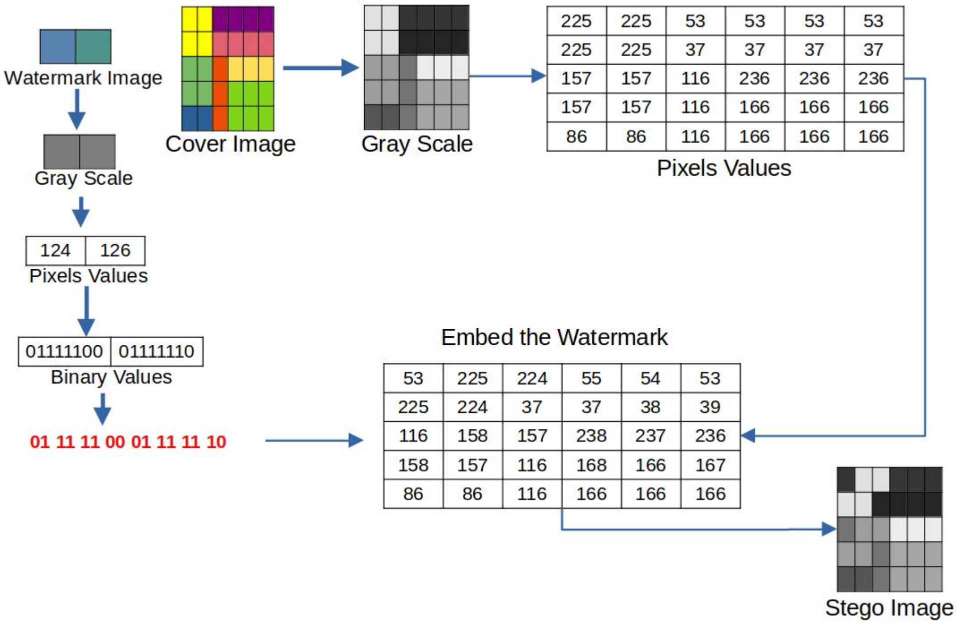
Figure 4.5: LSB (Min Avr Max) Text Algorithm.

convert the Text Watermark into the ASCII binary format. After that we loop throw the Cover image 3 pixels by 3 pixels and loop throw the Binary Text Watermark two bits by two bits with changing the order of the 3 pixels according to the table 4.1, after embedding all the characters we embed the NULL character to stop the extracting process later, the null character represented as 8 Zeroes. the process steps are shown in the (fig 4.5).
Each character from the Text Watermark will take a space of 8 bits in the ASCII binary format, so each character will need 12 pixels from the Cover image to be embedded.

Image Watermark: To embed an Image Watermark using this technique, first we load the Cover image as gray scale format, and also we do the same for the Watermark image, then we resize the Watermark Image to certain size to be able to extracted later, after that we convert each pixel from the resized watermark into the binary format and concatenate them into one long string of binary values. After that we loop throw the Cover image 3 pixels by 3 pixels and loop throw the Binary Image Watermark two bits by two bits with changing the order of the 3 pixels according to the table 4.1, the process steps are shown in the (fig 4.6).
Each pixel from the Watermark Image will be represented as one color channels with value between 0 and 255, and the color channel will take a space of 8 bits in the binary format, so each Pixel form the Watermark Image will need 12 pixels from the Cover image to be embedded.

4.6.2 Discrete Cosine Transform (DCT)

One method that is frequently used in image processing and watermarking is the Discrete Cosine Transform (DCT). An image is converted from the spatial domain to the frequency domain, where its frequency components serve as its representation.



124	126

01111100	01111110

01 11 11 00 01 11 11 10

225	225	53	53	53	53
225	225	37	37	37	37
157	157	116	236	236	236
157	157	116	166	166	166
86	86	116	166	166	166

53	225	224	55	54	53
225	224	37	37	38	39
116	158	157	238	237	236
158	157	116	168	166	167
86	86	116	166	166	166

Figure 4.6: LSB (Min Avr Max) Image Algorithm.

In this work, we effectively incorporate watermarks into medical images using the DCT method.

In this technique the watermark is loaded as gray scale image, then each pixel will be converted into binary format of 8-bits.
In case of Test watermark, the text will be converted into a binary image as shown in the (fig 4.7) and (fig 4.8).

4.6.2.1 Divide the Image into Blocks

The image is divided into small non-overlapping blocks. Typically, square blocks of fixed size, such as 8x8 or 16x16 pixels, are used.

4.6.2.2 Apply DCT to Each Block

Every block is subjected to DCT individually. In the frequency domain, each coefficient denotes the magnitude of a distinct frequency component, converting the spatial domain representation of the image into the frequency domain.

4.6.2.3 Select Embedding Positions

Establish the locations of the watermark's embedding within the DCT coefficients. These places are typically chosen based on how robustly they adhere to standard image processing methods or how important they are perceptually.
It's usually advisable to embed the watermark bit in the DCT block's lower-right corner for a number of reasons:

Energy Concentration: DC coefficients, or low-frequency components, are often represented by DCT coefficients in the lower-right corner. These coefficients provide significant perceptual information about the block. Since changes in low-frequency

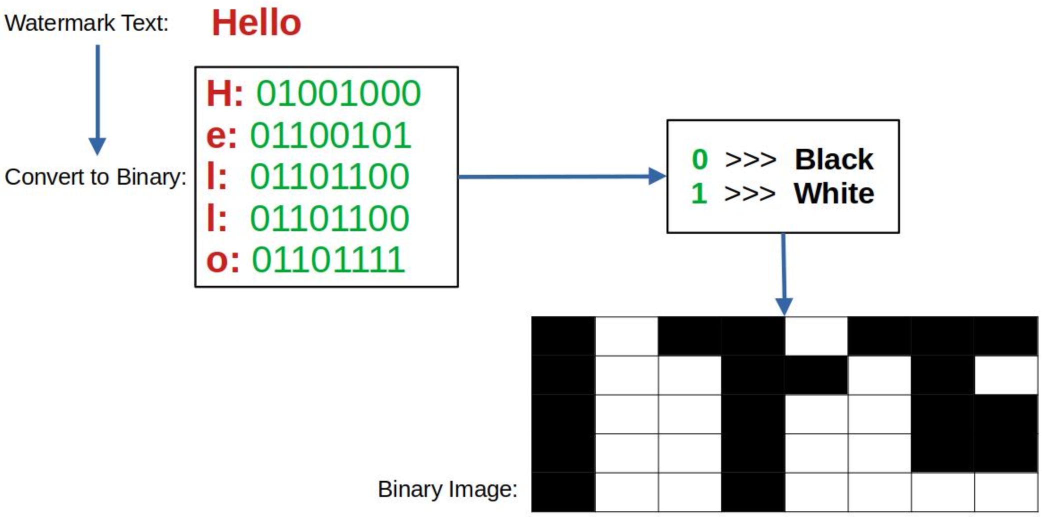
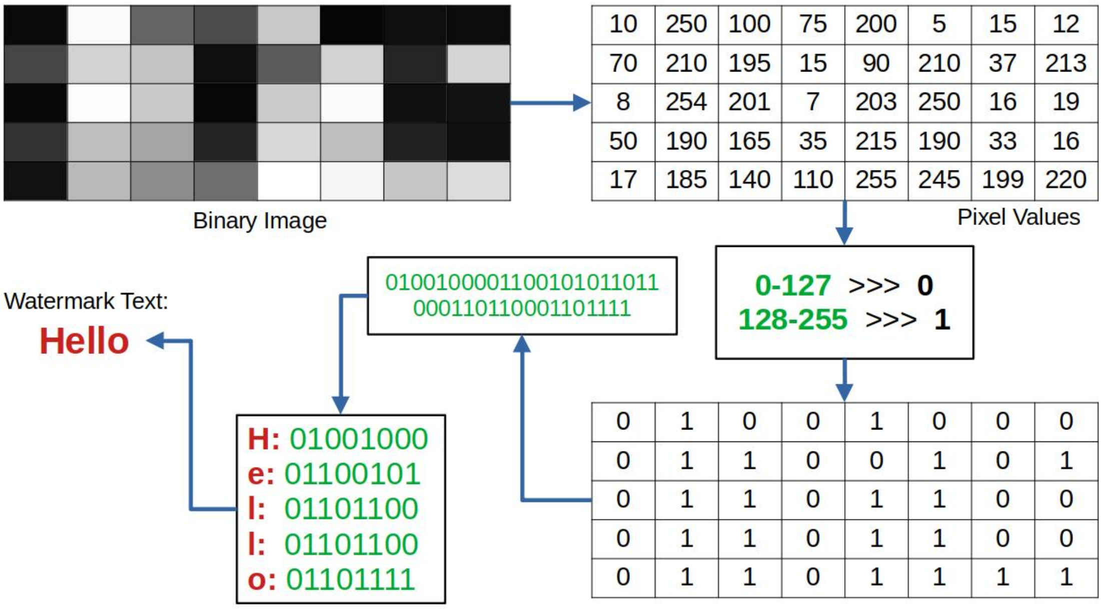


Figure 4.7: Convert Text to Binary Image.



10	250	100	75	200	5	15	12
70	210	195	15	90	210	37	213
8	254	201	7	203	250	16	19
50	190	165	35	215	190	33	16
17	185	140	110	255	245	199	220

0	1	0	0	1	0	0	0
0	1	1	0	0	1	0	1
0	1	1	0	1	1	0	0
0	1	1	0	1	1	0	0
0	1	1	0	1	1	1	1

Figure 4.8: Convert Binary Image to Text.

components are less evident to the human eye than changes in high-frequency components, embedding the watermark in this area ensures that it is less perceptually detectable.

Robustness to Compression: The quantization process tends to retain low-frequency components while eliminating high-frequency components in many image compression algorithms, including JPEG compression. The watermark has a better chance of surviving compression without suffering from severe deterioration if it is embedded in the low-frequency area.

Robustness to Cropping and Resizing: Compared to high-frequency components, low-frequency components usually stay intact or are less damaged when an image is cropped or resized. As a result, adding a watermark to the lower-right corner of the image improves its resistance to typical image alteration techniques.

Diminished Visibility: Viewers will see the watermark as less noticeable or undetectable due to changes in the lower-right corner of the DCT block that are less noticeable to the human eye. This guarantees that the appearance and visual quality of the watermarked image are preserved.

4.6.2.4 Change DCT Coefficients

To embed the watermark information, change or swap out a few DCT coefficients. A minor number can be added or subtracted from the coefficient, or it can be immediately replaced with a new value that is obtained from the watermark, depending on the adjustment.

4.6.2.5 Inverse DCT

To return the modified coefficients to the spatial domain, apply the inverse DCT (IDCT) to each modified block.

4.6.2.6 Combine Blocks

To create the watermarked image, put the altered blocks back together.

4.6.2.7 Embedding Space

Using this technique we can embed one bit from the binary watermark into each DCT Block, the embedding space could be calculated with this formula:

$$\text{DCT Embedding Space} = \frac{W \times H}{B} \tag{4.5}$$

Where W: is the Width of the Cover image, H: is the Height of the Cover image and B: is the DCT Block Size.

4.6.3 Discrete Wavelet Transform (DWT)

A thorough explanation of the Discrete Wavelet Transform (DWT) watermark embedding procedure is given in this section.

4.6.3.1 Decomposition

Input Image: The DWT is used to break the input image up into several tiers of wavelet subbands.

Wavelet Filters At each decomposition level, the image is convolved using high-pass and low-pass filters, producing approximate (low-frequency) and detailed (high-frequency) coefficients.

4.6.3.2 Selection of Embedding Location

Choice of Subbands Certain wavelet subbands are chosen for the watermark's embedding based on the intended trade-off between resilience and invisibility.

Typical Selection Due to their stability, the low-frequency approximation subband (LL) and their sensitivity to changes, the high-frequency detail subbands (HL, LH, HH) are popular choices.

Differences when Embedding into LL vs HL vs LH vs HH: The choice of subbands (LL, HL, LH, HH) when employing the Discrete Wavelet Transform (DWT) to embed a watermark dictates where the watermark is embedded in the frequency domain and how it affects the image. Here's a quick rundown of the variations:

LL (Low-Low) Subband:

- Comprised of the image's low-frequency elements.
- LL subband is typically utilized for watermark embedding when robustness against standard image processing procedures, such as resizing and compression, is desired.
- The watermark can be distributed throughout the entire image by embedding it in the LL subband, which reduces the likelihood of distortion.

HL (High-Low) Subband:

- Has high-frequency horizontal features.
- Since it alters the image's horizontal boundaries and details, including the watermark in the HL subband may result in the watermark becoming invisible.
- In contrast to LL subband, HL subband could not be as resistant to specific image alterations.

LH (Low-High) Subband:

- Has high-frequency vertical features.
- Embedding in the LH subband, like in the HL subband, can accomplish watermark invisibility without compromising the image's vertical edges and details.
- Depending on the type of image and watermark, the robustness properties of the LH subband may differ from those of the HL subband.

HH (High-High) Subband:

- Has high-frequency diagonal features.
- Embedding in the HH subband may cause the watermark to become embedded in the image's tiny features or texture.
- The HH subband can be resilient to some assaults and transformations, but if the watermark is not well-designed, it can potentially lead to more pronounced artifacts.

4.6.3.3 Watermark Embedding

Scaling of Watermark: In order to align with the dynamic range of the chosen wavelet subbands, the watermark signal is scaled.

Addition to Coefficients: The chosen coefficients in the wavelet subbands receive an addition of the scaled watermark signal, in our case we selected each 11th bit to write the information on it.

4.6.3.4 Reconstruction

Inverse DWT (IDWT): The inverse DWT is used to rebuild the changed wavelet coefficients into the spatial domain.

Combination of Subbands: The altered coefficients from the chosen subbands are combined with the unaltered coefficients from the other subbands to create the reconstructed image.

4.7 Results

4.7.1 Imperceptibility test

It is essential that medical image watermarking preserve patient information while maintaining high image quality. Many distortions may arise throughout the integrating process [27].

Table 4.2: Imperceptibility Test Result for LSB RGB.

Attacks	Text Watermark			Image Watermark		
	MSE	PSNR	SSIM	MSE	PSNR	SSIM
No Attacks	0.0006	80.29	0.9999	0.201	55.0968	0.9988
Salt	10.155	38.077	0.1923	1.1727	47.4387	0.7978
Pepper	10.124	38.0914	0.65	1.2615	47.1207	0.7543
Gaussian Noise	94.5038	28.3791	0.1822	94.7291	28.3659	0.1876
Histogram Equalization	95.7986	28.3656	0.8459	113.82	27.5682	0.9619
Content	6.7725	39.8478	0.8752	2.6751	43.8572	0.9681
Compress	27.7843	33.7588	0.8565	33.2874	32.908	0.7981

Table 4.3: Imperceptibility Test Result for LSB Gray.

Attacks	Text Watermark			Image Watermark		
	MSE	PSNR	SSIM	MSE	PSNR	SSIM
No Attacks	0.0021	74.8068	0.9999	0.2031	55.0535	0.9989
Salt	10.1549	38.0773	0.1922	1.2127	47.2929	0.7917
Pepper	10.1359	38.086	0.228	1.2701	47.0921	0.755
Gaussian Noise	94.5105	28.3789	0.1821	94.6947	28.3675	0.1877
Histogram Equalization	95.7783	28.3668	0.8461	111.34	27.6642	0.9619
Content	10.8218	37.8481	0.8917	2.2399	44.6283	0.9645
Compress	27.7846	33.7588	0.8565	33.3141	32.9045	0.7581

Table 4.4: Imperceptibility Test Result for LSB Min Avr Max.

Attacks	Text Watermark			Image Watermark		
	MSE	PSNR	SSIM	MSE	PSNR	SSIM
No Attacks	0.0831	59.6989	0.9998	12.9569	37.0057	0.909
Salt	10.2282	38.046	0.1923	13.8136	36.7277	0.7249
Pepper	10.2025	38.0576	0.2281	13.8969	36.7016	0.6882
Gaussian Noise	94.5063	28.379	0.1821	95.0351	28.3519	0.1704
Histogram Equalization	95.7499	28.3676	0.8462	109.4924	27.7369	0.8722
Content	10.9028	37.8142	0.8916	15.0736	36.3486	0.8824
Compress	27.793	33.7575	0.8565	34.3584	32.7704	0.7853

Table 4.5: Imperceptibility Test Result for DCT.

Attacks	Text Watermark			Image Watermark		
	MSE	PSNR	SSIM	MSE	PSNR	SSIM
No Attacks	3.3142	42.9362	0.9756	15.5411	36.2159	0.8963
Salt	4.2965	41.8033	0.7436	16.4049	35.981	0.72
Pepper	4.2922	41.8077	0.74	16.4401	35.9717	0.6781
Gaussian Noise	94.5881	28.3752	0.1814	63.89	30.1449	0.5659
Histogram Equalization	95.833	28.3591	0.8149	97.5688	28.2376	0.8656
Content	14.1339	36.655	0.8674	17.806	35.6251	0.8679
Compress	81.0615	29.0594	0.6772	86.7302	28.749	0.5821

Table 4.6: Imperceptibility Test Result for DWT.

Attacks	Text Watermark			Image Watermark		
	MSE	PSNR	SSIM	MSE	PSNR	SSIM
No Attacks	26.4997	33.8984	33.8984	34.7349	32.31	0.8176
Salt	27.2491	33.7773	0.6164	35.3966	32.6411	0.6631
Pepper	27.2457	33.7778	0.6016	35.4334	32.6366	0.6286
Gaussian Noise	95.0683	28.3527	0.1734	66.8416	29.8803	0.5376
Histogram Equalization	96.0476	28.3451	0.6002	100.3276	28.1165	0.7825
Content	31.5567	33.1409	0.6919	36.5171	32.5058	0.7929
Compress	81.1797	29.0528	0.5564	86.7736	28.7469	0.5375

4.7.2 Robustness test

There are numerous ways to alter the image. It is crucial to remember that these changes are not always attempts to manipulate the image for unauthorized use or to fake its authenticity. These may be adjustments meant to modify the image for one's own usage. However, the watermark needs to remain the same in every situation. The most popular assaults are applied to the watermarked image in order to assess the resilience of our approach. The extracted watermark is then compared to the original mark by computing the normalized correlation between the two. According to Hashim et al. (2018), an NC > 0.85 indicates a strong similarity between the extracted watermark and the original watermark.

Table 4.7: Robustness Test Result for LSB RGB.

Attacks	Text Watermark	Image Watermark
	NC	NC
No Attacks	0.9999	0.9988
Salt	0.01	0.7846
Pepper	0	0.8341
Gaussian Noise	0.1058	0.02
Histogram Equalization	0.1198	0.0796
Content	0.9999	0.8007
Compress	0.0526	0.011

Table 4.8: Robustness Test Result for LSB Gray.

Attacks	Text Watermark	Image Watermark
	NC	NC
No Attacks	0.9999	0.9785
Salt	0.7265	0.9769
Pepper	0.7384	0.9705
Gaussian Noise	0.2365	0.005
Histogram Equalization	0.3234	0.06
Content	0.9999	0.9525
Compress	0.0506	0.0033

Table 4.9: Robustness Test Result for LSB Min Avr Max.

Attacks	Text Watermark	Image Watermark
	NC	NC
No Attacks	0.9999	0.9785
Salt	0.4111	0.085
Pepper	0.4121	0.0024
Gaussian Noise	0.1821	0.0053
Histogram Equalization	0.4883	0.0145
Content	0.9999	0.3746
Compress	0.1197	0.0078

Table 4.10: Robustness Test Result for DCT.

Attacks	Text Watermark	Image Watermark
	NC	NC
No Attacks	0.9999	0.875
Salt	0.9756	0.5859
Pepper	0.9765	0.8539
Gaussian Noise	0.5111	0.4158
Histogram Equalization	0.9637	0.8654
Content	0.9999	0.8272
Compress	0.4824	0.3052

Table 4.11: Robustness Test Result for DWT.

Attacks	Text Watermark	Image Watermark
	NC	NC
No Attacks	0.9313	0.8586
Salt	0.8483	0.7611
Pepper	0.8314	0.7506
Gaussian Noise	0.4586	0.6314
Histogram Equalization	0.7211	0.7054
Content	0.8859	0.8371
Compress	0.5546	0.3908

4.8 Conclusion

The studies carried out for this work offer important new perspectives on the efficiency and performance of several medical image watermarking methods. We evaluated their effects on the computational efficiency, imperceptibility, and robustness of watermarking in medical images using techniques including LSB, DCT, and DWT. While each method had its own advantages, DCT and DWT proved to be more resilient and imperceptible than LSB, albeit requiring more computing power.

A thorough comparison of the watermarking methods was made possible by the assessment metrics, which included MSE, PSNR, SSIM, and NC. This comparison highlighted the trade-offs between resilience, imperceptibility, and computing complexity. The outcomes emphasized how crucial it is to choose the right watermarking techniques for medical imaging applications based on particular needs and limitations.

To sum up, the outcomes of the experiment highlight how important it is to use strong and effective watermarking methods to protect medical photos. The results highlight the necessity of continued study to improve and develop these techniques in order to guarantee the safe and dependable handling of medical images in the healthcare sector.

Chapter 5

General Conclusion

In conclusion, the utilization of LSB, DWT, and DCT watermarking techniques in medical image security presents promising avenues for safeguarding sensitive patient data while ensuring the integrity and authenticity of medical images. Through our investigation, we have demonstrated the effectiveness of these methods in embedding and extracting watermarks with minimal distortion to the original image quality. The LSB method, with its simplicity and ease of implementation, offers a straightforward approach to watermarking, albeit with limitations in terms of robustness and capacity. The DWT technique provides superior robustness and security through its multi-resolution analysis, making it suitable for applications where high levels of security are required. Additionally, the DCT technique offers a viable alternative, with its ability to efficiently represent image content in the frequency domain.

Furthermore, our results underscore the importance of considering the specific requirements and constraints of medical image watermarking, such as imperceptibility, capacity, and robustness to various attacks. By tailoring the watermarking approach to meet these requirements, we can ensure the successful integration of watermarking techniques into clinical workflows without compromising diagnostic accuracy or patient privacy. Additionally, our study highlights the need for further research and development in this field to address emerging challenges, such as the mitigation of potential vulnerabilities to sophisticated attacks.

In essence, the integration of LSB, DWT, and DCT watermarking techniques holds great promise for enhancing the security and integrity of medical image data, thereby contributing to improved patient care and clinical outcomes. As the healthcare industry continues to embrace digital technologies, it is imperative that robust and efficient watermarking solutions be developed and deployed to safeguard the confidentiality, integrity, and authenticity of medical images in the digital era.

Possibly the future word improvement could be as following:

Advanced Watermarking Techniques

- **Hybrid Methods:** Combining multiple watermarking techniques (e.g., LSB with DWT or DCT with DWT) to leverage the strengths of each method and achieve better robustness and imperceptibility.

- **Machine Learning Integration:** Using machine learning algorithms to dynamically adjust watermark embedding parameters based on the characteristics of the medical images and the type of watermark.

- **Deep Learning:** Implementing deep learning-based watermarking approaches that can learn optimal embedding and extraction strategies from large datasets.

Enhanced Security Measures

- **Cryptographic Watermarking:** Integrating cryptographic techniques to ensure the security and integrity of the watermark, making it more resistant to intentional attacks.

Comprehensive Evaluation:

- **Robustness Testing:** Conducting extensive robustness testing against a wider range of attacks, including geometric distortions, compression, noise addition, and combined attacks.

- **Comparative Studies:** Performing comparative studies with other state-of-the-art watermarking techniques to benchmark performance in terms of robustness, imperceptibility, and computational efficiency.

Bibliography

[1] How JPEG works. https://cgjennings.ca/articles/jpeg-compression/, 2017. [Online; accessed 14-May-2024].

[2] Mohammad Abdullatif, Akram M Zeki, Jalel Chebil, and Teddy Surya Gunawan. Properties of digital image watermarking. In *2013 IEEE 9th international colloquium on signal processing and its applications*, pages 235–240. IEEE, 2013.

[3] Himanshu Agarwal, Balasubramanian Raman, and Ibrahim Venkat. Blind reliable invisible watermarking method in wavelet domain for face image watermark. *Multimedia Tools and Applications*, 74:6897–6935, 2015.

[4] Adnan M Alattar. Reversible watermark using the difference expansion of a generalized integer transform. *IEEE transactions on image processing*, 13(8):1147–1156, 2004.

[5] Hanan S Alshanbari. Medical image watermarking for ownership & tamper detection. *Multimedia tools and applications*, 80(11):16549–16564, 2021.

[6] Alireza Avanaki. Exact histogram specification optimized for structural similarity. *arXiv preprint arXiv:0901.0065*, 2008.

[7] K Balasamy and S Suganyadevi. A fuzzy based roi selection for encryption and watermarking in medical image using dwt and svd. *Multimedia tools and applications*, 80(5):7167–7186, 2021.

[8] William J Buchanan. Dct (discrete cosine transform)). https://asecuritysite.com/comms/dct2, 2024. Accessed: May 14, 2024.

[9] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[10] Xiaochun Cao, Ling Du, Xingxing Wei, Dan Meng, and Xiaojie Guo. High capacity reversible data hiding in encrypted images by patch-level sparse representation. *IEEE transactions on cybernetics*, 46(5):1132–1143, 2015.

[11] Ruchika Chandel and Gaurav Gupta. Image filtering algorithms and techniques: A review. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(10), 2013.

[12] Digvijay Singh Chauhan, Amit Kumar Singh, Abhinav Adarsh, Basant Kumar,

and Jai Prakash Saini. Combining mexican hat wavelet and spread spectrum for adaptive watermarking and its statistical detection using medical images. *Multimedia Tools and Applications*, 78:12647–12661, 2019.

[13] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. *Digital watermarking and steganography.* Morgan kaufmann, 2007.

[14] Ingemar J Cox, Joe Kilian, F Thomson Leighton, and Talal Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE transactions on image processing*, 6(12):1673–1687, 1997.

[15] Anuja Dixit and Rahul Dixit. A review on digital image watermarking techniques. *International Journal of Image, Graphics and Signal Processing*, 9(4):56, 2017.

[16] LP EX and NOTIVA INNO. International journal of engineering and advanced technology...

[17] Tzuo-Yau Fan, Her-Chang Chao, and Bin-Chang Chieu. Lossless medical image watermarking method based on significant difference of cellular automata transform coefficient. *Signal Processing: Image Communication*, 70:174–183, 2019.

[18] Jessica Fridrich. *Steganography in digital media: principles, algorithms, and applications.* Cambridge University Press, 2009.

[19] Jessica Fridrich, Miroslav Goljan, and Rui Du. Detecting lsb steganography in color, and gray-scale images. *IEEE multimedia*, 8(4):22–28, 2001.

[20] Jessica Fridrich, Miroslav Goljan, and Rui Du. Reliable detection of lsb steganography in color and grayscale images. In *Proceedings of the 2001 workshop on Multimedia and security: new challenges*, pages 27–30, 2001.

[21] Jessica Fridrich, Miroslav Goljan, and Rui Du. Lossless data embedding—new paradigm in digital watermarking. *EURASIP Journal on Advances in Signal Processing*, 2002:1–12, 2002.

[22] Solihah Gull and Shabir A Parah. Advances in medical image watermarking: a state of the art review. *Multimedia Tools and Applications*, 83(1):1407–1447, 2024.

[23] Thien Huynh-The, Cam-Hao Hua, Nguyen Anh Tu, Taeho Hur, Jaehun Bang, Dohyeong Kim, Muhammad Bilal Amin, Byeong Ho Kang, Hyonwoo Seung, and Sungyoung Lee. Selective bit embedding scheme for robust blind color image watermarking. *Information Sciences*, 426:1–18, 2018.

[24] Milad Jafari Barani, Peyman Ayubi, Milad Yousefi Valandar, and Behzad Yosefnezhad Irani. A blind video watermarking algorithm robust to lossy video compression attacks based on generalized newton complex map and contourlet transform. *Multimedia Tools and Applications*, 79(3):2127–2159, 2020.

[25] Yu-Ze Jheng, Chien-Yuan Chen, and Chien-Feng Huang. Reversible data hiding

based on histogram modification over ternary computers. *J. Inf. Hiding Multim. Signal Process.*, 6(5):938–955, 2015.

[26] Neil F Johnson and Stefan Katzenbeisser. A survey of steganographic techniques. In *Information hiding*, pages 43–78. Artech House Norwood, MA, 2000.

[27] Fares Kahlessenane, Amine Khaldi, Redouane Kafi, and Salah Euschi. A dwt based watermarking approach for medical image protection. *Journal of Ambient Intelligence and Humanized Computing*, 12(2):2931–2938, 2021.

[28] Jidagam Venkata Karthik and B Venkateswara Reddy. Authentication of secret information in image stenography. *International Journal of Computer Science and Network Security (IJCSNS)*, 14(6):58, 2014.

[29] P Kishore, M Rao, Ch Prasad, and D Kumar. Medical image watermarking: run through review. *ARPN J Eng Appl Sci*, 11(5):2882–2899, 2016.

[30] R Rama Kishore et al. A novel and efficient blind image watermarking in transform domain. *Procedia Computer Science*, 167:1505–1514, 2020.

[31] Sanjay Kumar, Binod Kumar Singh, and Mohit Yadav. A recent survey on multimedia and database watermarking. *Multimedia Tools and Applications*, 79(27):20149–20197, 2020.

[32] Deepa Kundur and Dimitrios Hatzinakos. Digital watermarking using multiresolution wavelet decomposition. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, volume 5, pages 2969–2972. IEEE, 1998.

[33] Deepa Kundur and Dimitrios Hatzinakos. Digital watermarking for telltale tamper proofing and authentication. *Proceedings of the IEEE*, 87(7):1167–1180, 1999.

[34] Chih-Chin Lai. A digital watermarking scheme based on singular value decomposition and tiny genetic algorithm. *Digital Signal Processing*, 21(4):522–527, 2011.

[35] Vinicius Licks and Ramiro Jordan. Geometric attacks on image watermarking systems. *IEEE multimedia*, 12(3):68–78, 2005.

[36] Wei-Hung Lin, Shi-Jinn Horng, Tzong-Wann Kao, Pingzhi Fan, Cheng-Ling Lee, and Yi Pan. An efficient watermarking method based on significant difference of wavelet coefficient quantization. *IEEE Transactions on Multimedia*, 10(5):746–757, 2008.

[37] Bin Ma, Yunhong Wang, Chunlei Li, Zhaoxiang Zhang, and Di Huang. Secure multimodal biometric authentication with wavelet quantization based fingerprint watermarking. *Multimedia tools and applications*, 72:637–666, 2014.

[38] Mahmoud Magdy, Khalid M Hosny, Neveen I Ghali, and Said Ghoniemy. Security of medical images for telemedicine: a systematic review. *Multimedia Tools and Applications*, 81(18):25101–25145, 2022.

[39] Andrea Manno-Kovacs. Direction selective vector field convolution for contour detection. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 4722–4726. IEEE, 2014.

[40] Wes McKinney. *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython.* " O'Reilly Media, Inc.", 2012.

[41] Abdulmawla Najih, SAR Al-Haddad, SJ Hashim, Mohammad Ali Nematollahi, et al. Digital image watermarking based on angle quantization in discrete contourlet transform. *Journal of King Saud University-Computer and Information Sciences*, 29(3):288–294, 2017.

[42] Nikos Nikolaidis and Ioannis Pitas. Robust image watermarking in the spatial domain. *Signal processing*, 66(3):385–403, 1998.

[43] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002.

[44] Fatih Özkaynak. Role of npcr and uaci tests in security problems of chaos based image encryption algorithms and possible solution proposals. In *2017 International conference on computer science and engineering (UBMK)*, pages 621–624. IEEE, 2017.

[45] Shabir A Parah, Javaid A Sheikh, Nazir A Loan, and Ghulam M Bhat. Robust and blind watermarking technique in dct domain using inter-block coefficient differencing. *Digital Signal Processing*, 53:11–24, 2016.

[46] Christine I Podilchuk and Edward J Delp. Digital watermarking: algorithms and applications. *IEEE signal processing Magazine*, 18(4):33–46, 2001.

[47] Christine I Podilchuk and Wenjun Zeng. Image-adaptive watermarking using visual models. *IEEE Journal on selected areas in communications*, 16(4):525–539, 1998.

[48] Chandra Mouli PVSSR et al. Adaptive, robust and blind digital watermarking using bhattacharyya distance and bit manipulation. *Multimedia Tools and Applications*, 77(5):5609–5635, 2018.

[49] Asaad F Qasim, Farid Meziane, and Rob Aspin. Digital watermarking: Applicability for developing trust in medical imaging workflows state of the art review. *Computer Science Review*, 27:45–60, 2018.

[50] Chuan Qin, Huili Wang, Xinpeng Zhang, and Xingming Sun. Self-embedding fragile watermarking based on reference-data interleaving and adaptive selection of embedding mode. *Information sciences*, 373:233–250, 2016.

[51] P Ramana Reddy, Munaga VNK Prasad, and D Sreenivasa Rao. Robust digital watermarking of color images under noise attacks. *International Journal of Recent Trends in Engineering*, 1(1):334, 2009.

[52] Ratnakirti Roy, Tauheed Ahmed, and Suvamoy Changder. Watermarking

through image geometry change tracking. *Visual Informatics*, 2(2):125–135, 2018.

[53] Euschi Salah, Khaldi Amine, Kafi Med Redouane, and Kahlessenane Fares. Spatial and frequency approaches for audio file protection. *Journal of Circuits, Systems and Computers*, 30(12):2150210, 2021.

[54] Jamal N Bani Salameh. A new approach for securing medical images and patient's information by using a hybrid system. *Int J of Computer Sci Netw Secur*, 19(4):28–39, 2019.

[55] K Sakthidasan Sankaran, H Abhi Rayna, Vaishnavi Mangu, VR Prakash, and N Vasudevan. Image water marking using dwt to encapsulate data in medical image. In *2019 International Conference on Communication and Signal Processing (ICCSP)*, pages 0568–0571. IEEE, 2019.

[56] Saeed Sarreshtedari and Mohammad Ali Akhaee. A source-channel coding approach to digital image protection and self-recovery. *IEEE transactions on image processing*, 24(7):2266–2277, 2015.

[57] Dhawal Seth, L Ramanathan, and Abhishek Pandey. Security enhancement: combining cryptography and steganography. *International Journal of Computer Applications*, 9(11):3–6, 2010.

[58] Shifa Showkat, Shabir A Parah, and Solihah Gull. Embedding in medical images with contrast enhancement and tamper detection capability. *Multimedia Tools and Applications*, 80:2009–2030, 2021.

[59] Amit Kumar Singh, Basant Kumar, Mayank Dave, and Anand Mohan. Robust and imperceptible dual watermarking for telemedicine applications. *Wireless Personal Communications*, 80:1415–1433, 2015.

[60] Priyanka Singh, K Jyothsna Devi, Hiren Kumar Thakkar, and Ketan Kotecha. Region-based hybrid medical image watermarking scheme for robust and secured transmission in iomt. *IEEE Access*, 10:8974–8993, 2022.

[61] Siddharth Singh, Rajiv Singh, Amit Kumar Singh, and Tanveer J Siddiqui. Svd-dct based medical image watermarking in nsct domain. *Quantum computing: an environment for intelligent large scale real application*, pages 467–488, 2018.

[62] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis and machine vision.* Springer, 2013.

[63] Qingtang Su. Novel blind colour image watermarking technique using hessenberg decomposition. *IET image processing*, 10(11):817–829, 2016.

[64] Qingtang Su, Decheng Liu, Zihan Yuan, Gang Wang, Xiaofeng Zhang, Beijing Chen, and Tao Yao. New rapid and robust color image watermarking technique in spatial domain. *IEEE Access*, 7:30398–30409, 2019.

[65] Qingtang Su, Yugang Niu, Gang Wang, Shaoli Jia, and Jun Yue. Color image blind watermarking scheme based on qr decomposition. *Signal Processing*,

94:219–235, 2014.

[66] Qingtang Su, Gang Wang, Shaoli Jia, Xiaofeng Zhang, Qiming Liu, and Xianxi Liu. Embedding color image watermark in color image based on two-level dct. *Signal, Image and Video Processing*, 9:991–1007, 2015.

[67] Qingtang Su, Zihan Yuan, and Decheng Liu. An approximate schur decomposition-based spatial domain color image watermarking method. *IEEE Access*, 7:4358–4370, 2018.

[68] Michael J Swain and Dana H Ballard. Color indexing. *International journal of computer vision*, 7(1):11–32, 1991.

[69] Rohit Thanki, Surekha Borra, Vedvyas Dwivedi, and Komal Borisagar. An efficient medical image watermarking scheme based on fdcut–dct. *Engineering science and technology, an international journal*, 20(4):1366–1379, 2017.

[70] Kharittha Thongkor, Thumrongrat Amornraksa, and Edward J Delp. Digital watermarking for camera-captured images based on just noticeable distortion and wiener filtering. *Journal of Visual Communication and Image Representation*, 53:146–160, 2018.

[71] Jake VanderPlas. *Python data science handbook: Essential tools for working with data.* " O'Reilly Media, Inc.", 2016.

[72] Usha Verma and Neelam Sharma. Hybrid mode of medical image watermarking to enhance robustness and imperceptibility. *Int J Innov Technol Explor Eng*, 9(1):351–359, 2019.

[73] Sviatoslav Voloshynovskiy, Shelby Pereira, Victor Iquise, and Thierry Pun. Attack modelling: towards a second generation watermarking benchmark. *Signal processing*, 81(6):1177–1214, 2001.

[74] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[75] Zhang Wenyin and Frank Y Shih. Semi-fragile spatial watermarking based on local binary pattern operators. *Optics Communications*, 284(16-17):3904–3912, 2011.

[76] Shaoping Xu, Shunliang Jiang, and Weidong Min. No-reference/blind image quality assessment: a survey. *IETE Technical Review*, 34(3):223–245, 2017.

[77] Xiaoyan Yu, Chengyou Wang, and Xiao Zhou. Review on semi-fragile watermarking algorithms for content authentication of digital images. *Future Internet*, 9(4):56, 2017.