TABLE OF CONTENTS

ABSTRACT

The rapid advancement of digital technologies, the sharing and distribution of medical images have become widespread, posing serious security challenges. To protect sensitive medical data from unauthorized access and tampering, watermarking has emerged as a crucial security measure. In addition, the concept of watermarking has become vital in preserving the integrity and authenticity of these images. Traditional watermarking techniques faced limitations in terms of robustness and visibility, especially for medical imaging, where image quality is paramount. To overcome these challenges, this work introduces an innovative blind medical image watermarking technique that combines the Discrete Wavelet Transform (DWT) and Discrete Cosine Transform (DCT). The proposed method ensures robust and imperceptible watermark embedding and retrieval while maintaining the visual quality of medical images. The significance of robust and imperceptible medical image watermarking cannot be overstated. As medical institutions increasingly adopt digital practices like telemedicine and electronic health records, the risk of data breaches, tampering, and unethical practices also rises. An efficient watermarking technique is crucial to protect patient privacy, maintain trust in medical institutions, and ensure the authenticity of medical data. The combined DWT-DCT approach presented in this paper offers a promising solution by enabling secure watermark embedding and retrieval, ensuring tamper detection and authentication. To protect the medical images integrity, digital watermark is embedded into the medical images. A non-blind medical image watermarking scheme based on hybrid transform is propounded. In this paper, fingerprint of the patient is used as watermark for better authentication, identifying the original medical image and privacy of the patients. In this scheme, lifting wavelet transform (LWT) and discrete wavelet transform (DWT) are utilized for amplifying the watermarking algorithm. The scaling and embedding factors are calculated adaptively with the help of Local Binary Pattern values of the host medical image to achieve better imperceptibility and robustness for medical images and fingerprint watermark, respectively. Two-level decomposition is done where for the first level LWT is utilized and for the second level decomposition DWT is utilized. At the extraction side, non-blind recovery of fingerprint watermark is performed which is similar to the embedding process.

Keywords: Digital image watermarking, image copyright protection, frequency-domain watermarking, Discrete Wavelet Transform (DWT), Discrete Cosine Transform (DCT)

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Introduction

The development of effective digital image copyright protection methods have recently become an urgent and necessary requirement in the multimedia industry due to the ever-increasing unauthorized manipulation and reproduction of original digital objects. The new technology of digital watermarking has been advocated by many specialists as the best method to such multimedia copyright protection problem. It's expected that digital watermarking will have a wide- span of practical applications such as digital cameras, medical imaging, image databases, and video- on-demand systems, among many others. In order for a digital watermarking method to be effective it should be imperceptible, and robust to common image manipulations like compression, filtering, rotation, scaling cropping, collusion attacks among many other digital signal processing operations. Current digital image watermarking techniques can be grouped into two major classes: spatial-domain and frequency-domain watermarking techniques. Compared to spatial domain techniques, frequency-domain watermarking techniques proved to be more effective with respect to achieving the imperceptibility and robustness requirements of digital watermarking algorithms. Commonly used frequency-domain transforms include the Discrete Wavelet Transform (DWT), the Discrete Cosine Transform (DCT) and Discrete Fourier Transform (DFT). However, DWT has been used in digital image watermarking more frequently due to its excellent spatial localization and multi-resolution characteristics, which are similar to the theoretical models of the human visual system. Further performance improvements in DWT-based digital image watermarking algorithms could be obtained by combining DWT with DCT. The idea of applying two transform is based on the fact that combined transforms could compensate for the drawbacks of each other, resulting in effective watermarking. In this paper, we will describe a digital image watermarking algorithm based on combining two transforms; DWT and DCT. Watermarking is done by altering the wavelets coefficients of carefully selected DWT sub-bands, followed by the application of the DCT transform on the selected sub-bands.

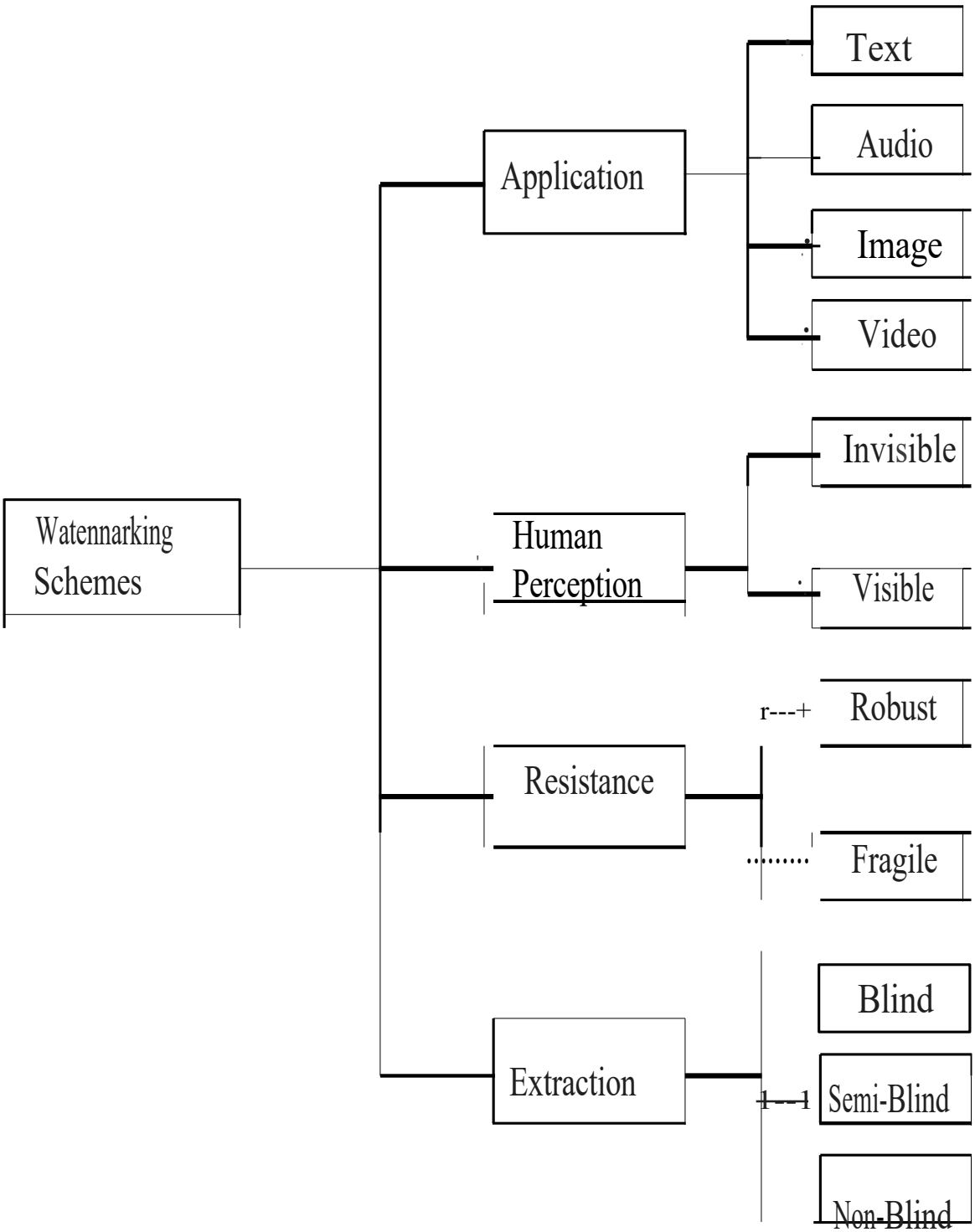


Figure 1: Watermarking Schemes

1.2. Project Objectives

"Developing an Effective and Robust Digital Image Watermarking Algorithm Using Combined DWT and DCT Transforms" This objective encapsulates the aim of the project, which is to create a method for embedding watermarks into digital images. The key focus areas of this objective can be broken down into several points:

- Effectiveness
- Robustness
- Combining DWT and DCT

The specific objective here is to investigate and implement a method that combines the Discrete Wavelet Transform (DWT) and the Discrete Cosine Transform (DCT) for watermarking. This involves altering the wavelet coefficients of selected DWT sub-bands and then applying the DCT transform on these modified sub-bands. This enhances the imperceptibility of the watermark. Consideration should also be given to the practical applications of the developed algorithm. Mentioned applications include digital cameras, medical imaging, image databases, and video-on- demand systems. The objective could involve demonstrating the usability and effectiveness of the algorithm in these real-world scenarios. By achieving these objectives, the project aims to contribute to the field of digital image copyright protection by providing a reliable, effective, and versatile watermarking method that addresses the challenges posed by unauthorized manipulation and reproduction of digital objects.

1.3. Purpose of the project

The purpose of the described project on developing a digital image watermarking algorithm using combined DWT and DCT transforms can be multifaceted and can serve several purposes: Enhancing Copyright Protection-The primary purpose is to contribute to the field of multimedia copyright protection. With the increasing unauthorized manipulation and reproduction of digital images, the project aims to provide a robust solution to embed watermarks imperceptibly into images-This helps in identifying and protecting the ownership of digital content. Improving Image Integrity-By embedding watermarks using a combination of DWT and DCT, the project aims to improve the integrity of digital images. This means ensuring that even after various manipulations or attacks, such as compression or cropping, the watermark remains intact and retrievable. Advancing Digital Forensics: The developed algorithm can also serve the purpose of aiding digital forensics.

Watermarks embedded using this method can act as digital fingerprints, allowing investigators to trace the origin and ownership of images,

which can be crucial in legal investigations involving digital media. Real-World Applications: Another purpose is to demonstrate the practical applications of the algorithm. The project aims to show how this technology can be used in various fields such as digital cameras, medical imaging, image databases, and video-on-demand systems. This helps in showcasing the versatility and usefulness of the developed method. Contributing to Research: The project serves the purpose of contributing to the research and development of digital image watermarking techniques. By combining DWT and DCT, the project explores new possibilities for improving the imperceptibility and robustness of watermarking algorithms. This contributes to the advancement of knowledge in the field of multimedia security.

The purpose of the project is to develop a novel, effective, and practical digital image watermarking algorithm that addresses the challenges of unauthorized manipulation and reproduction of digital images. It aims to contribute to copyright protection, enhance image integrity, aid in digital forensics, demonstrate real-world applications, advance research in watermarking techniques, and potentially benefit commercial and industrial sectors involved in multimedia production and distribution.

1.4. Existing System with Disadvantages

Medical image watermarking is a technique used to embed additional information, often for purposes like authentication, copyright protection, or data integrity verification, into medical images such as X-rays, MRI scans, or CT scans. The watermark is a hidden pattern or data that is embedded within the image while maintaining the diagnostic quality of the image. In your research work, use the Discrete Wavelet Transform (DWT) as a method for both embedding and extracting watermarks in medical images. Below, the procedures for embedding and extracting watermarks using DWT in the context of a research work.

1.4.1. Embedding Procedure:

- **Image Selection:** Start by selecting the medical image(s) you want to watermark. These can be grayscale or color images, depending on your research objectives.
- **Watermark Generation:** Generate the watermark data that you want to embed into the selected image. This watermark can be a binary image, a text string, or any other information you want to hide in the medical image.
- **DWT Transformation:** Apply the Discrete Wavelet Transform (DWT) to the original medical image. DWT decomposes the image into different frequency components (coefficients) in multiple

scales. This decomposition is typically done to enhance the robustness of the watermark against common image processing operations.

- **Embedding Watermark:** Modify the DWT coefficients in one or more of the decomposition scales to embed the watermark data. This is typically done by altering the coefficients slightly in a way that the changes are imperceptible to the human eye but can be detected later during extraction.
- **Inverse DWT:** Reconstruct the watermarked image from the modified DWT coefficients by applying the Inverse Discrete Wavelet Transform (IDWT). This results in a watermarked medical image that visually appears similar to the original but contains of the watermarked image in terms of visual quality and medical diagnostic information preservation. Common metrics for this evaluation include Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSI).
- **Save Metadata:** Store metadata about the watermark, such as its size, location, and any necessary encryption keys, separately from the watermarked image. This information will be needed during the extraction process.

1.4.2. Extraction Procedure:

- **Watermarked Image Selection:** Begin by selecting the watermarked medical image from which you want to extract the watermark.
- **DWT Transformation:** Apply the DWT to the watermarked image to decompose it into the same frequency components and scales used during embedding.
- **Watermark Extraction:** Locate and extract the watermark data from the modified DWT coefficients. This process may involve comparing the watermarked coefficients to the original ones, considering any expected alterations made during embedding.
- **Decryption (if applicable):** If encryption was used during embedding, decrypt the extracted watermark data using the appropriate encryption keys.
- **Verification:** Verify the extracted watermark for integrity and authenticity. This may involve checking for errors, confirming that it matches the stored metadata, and ensuring it hasn't been tampered with.
- **Optional Image Restoration:** If necessary, use the remaining DWT coefficients to reconstruct a watermarked image with the watermark removed. This step is crucial if the watermarked medical image needs to be used for diagnosis or analysis.
- **Performance Evaluation:** Assess the quality of the extracted watermark in terms of accuracy and robustness against common image processing attacks. You can also calculate the detection rate and false positive rate to evaluate the watermark extraction process.

- **Application-Specific Actions:** Depending on your research objectives, you might use the extracted watermark for copyright verification, authentication, or data integrity confirmation.

1.4.3. Disadvantages

While medical image watermarking using DWT offers several advantages, such as authentication and data integrity verification, it also comes with some drawbacks that should be considered in your research work:

- **Loss of Diagnostic Information:** One of the primary concerns in medical image watermarking is the potential loss of diagnostic information. Altering the image data, even imperceptibly, can affect the accuracy of medical diagnoses. The embedded watermark might interfere with critical image details, making it less reliable for medical professionals.
- **Sensitivity to Compression:** Medical images are often compressed for storage and transmission. DWT-based watermarking can be sensitive to compression, which can lead to a significant degradation in the quality of the watermarked image. This can make the watermark less robust and harder to extract.
- **Security Vulnerabilities:** Watermark extraction can be vulnerable to attacks, such as intentional removal or alteration of the watermark. Malicious actors may attempt to tamper with the watermark to manipulate the information contained within the image. Ensuring robustness against such attacks can be challenging.
- **Limited Payload:** DWT-based watermarking techniques have limited payload capacity. The amount of information that can be embedded is constrained by the available coefficients and the need to maintain image quality. This limitation can be a drawback when trying to embed large amounts of data.
- **Complexity and Computational Overhead:** Implementing DWT-based watermarking is computationally intensive, especially for medical images with high resolution. The DWT transformation, watermark embedding, and extraction processes can be time-consuming, which may not be suitable for real-time applications.
- **Ethical and Legal Concerns:** When working with medical images, you must consider ethical and legal implications. Patients' privacy and data protection regulations may restrict the use of watermarking techniques on medical images, especially if it involves personal health information.
- **Transparency and Interpretability:** Watermarked medical images may be less interpretable by medical professionals. The presence of a watermark, even if imperceptible, can raise questions about

the authenticity and trustworthiness of the image, potentially leading to reluctance in clinical practice.

- **Compatibility and Interoperability:** Watermarked images may not be compatible with all medical image analysis software and systems. Ensuring that the watermarked images can be seamlessly integrated into existing healthcare workflows can be a challenge.
- **Robustness Challenges:** While DWT-based watermarking aims to be robust against common image processing operations, it may not be completely resilient to all possible attacks. Sophisticated adversaries may find ways to remove or alter the watermark without detection.

1.5. Proposed System with Advantages

In this work illustrates a comprehensive watermark embedding process designed for medical images. The process is orchestrated to enhance the security and integrity of these images by seamlessly embedding hidden information while maintaining diagnostic quality. At the outset, the 'Host Image' is chosen as the canvas for the watermark. This image could be any medical scan, such as an X-ray or an MRI, and it serves as the foundation upon which the watermark will be added. Next, this work employs a multi-step transformation approach, starting with 'DWT (Wavelet Decomposition).' This Discrete Wavelet Transform breaks down the host image into different frequency components, a critical step to bolster the watermark's resilience against common image manipulations. Following the wavelet decomposition, the process progresses to 'DCT,' which stands for Discrete Cosine Transform. The application of DCT allows the conversion of spatial domain information into the frequency domain, contributing to the watermark's robustness against certain types of attacks. Simultaneously, the 'Watermark Image,' which can encompass various forms of data like images or text, is introduced as the content to be concealed within the host image. The 'Array Conversion' step is pivotal in the process, as it transforms both the DCT coefficients and the watermark image into arrays or matrices. This prepares them for further mathematical operations and their eventual integration. The 'Watermark Array' represents the converted watermark image in an array format, preparing it for seamless integration with the DCT coefficients. The actual embedding of the watermark into the host image takes place during the 'Embedding' phase. This step involves intricate algorithms that subtly modify the DCT coefficients to incorporate the watermark, all while striving to ensure minimal visual degradation. After embedding the watermark, the process proceeds with 'IDCT' (Inverse Discrete Cosine Transform), which inversely transforms the frequency domain information back into the spatial domain. This is crucial for the reconstruction of the image. Lastly, 'Wavelet Reconstruction' utilizes the inverse of the earlier 'DWT (Wavelet Decomposition)' step to

reconstruct the final 'Watermarked Image.' This resulting image appears visually similar to the original host image but now contains the embedded watermark.

In this research work outlines the process for extracting a watermark from a watermarked medical image, following the watermark embedding procedure discussed earlier. Let's delve into a detailed explanation of each step:

Watermarked Image: The starting point of the watermark extraction process is the watermarked medical image. This image has been previously processed to embed a hidden watermark, and the goal now is to retrieve that embedded information.

Wavelet Decomposition: The first step in extracting the watermark is to apply 'Wavelet Decomposition' to the watermarked image. This process involves breaking down the image into various frequency components using the Discrete Wavelet Transform (DWT). This step helps prepare the image for further analysis by separating it into its constituent parts.

Apply DCT: Following the wavelet decomposition, 'Apply DCT' comes into play. The Discrete Cosine Transform (DCT) is applied to the decomposed image data. This transformation is used to convert the image from the spatial domain to the frequency domain, making it easier to identify the watermark's presence and characteristics.

Extraction: The 'Extraction' step is where the actual process of retrieving the embedded watermark takes place. Techniques and algorithms are employed to detect and isolate the watermark information from the transformed image data. This step requires careful analysis to ensure the watermark is accurately extracted.

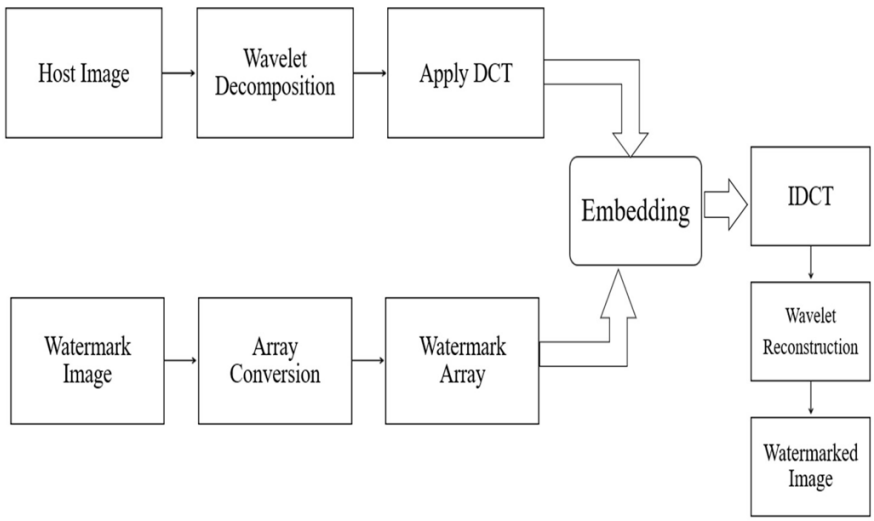


Figure 2: Proposed watermark embedding process.

1.5.1. Extracted Watermark Image

Once the watermark information has been successfully extracted, it is presented as the 'Extracted Watermark Image.' This image should ideally match the original watermark that was embedded in the host image, but some degradation may occur during the embedding and extraction process.

- **Evaluation Metrics (PSNR, MSE, Entropy):** To assess the quality and accuracy of the extracted watermark, a set of 'Evaluation Metrics' is applied. These metrics typically include:
- **Peak Signal-to-Noise Ratio (PSNR)**: Measures the quality of the extracted watermark by comparing it to the original watermark, taking into account any potential distortions.
- **Mean Squared Error (MSE)**: Quantifies the average squared difference between the extracted and original watermarks, providing insight into the level of fidelity.
- **Entropy**: Measures the randomness or unpredictability of the extracted watermark, which can indicate the watermark's robustness and resilience against tampering.

These evaluation metrics help gauge the effectiveness of the watermark extraction process, ensuring that the embedded watermark is accurately retrieved without significant degradation or distortion. It's essential to achieve high values in metrics like PSNR and low values in MSE while maintaining a desirable level of entropy to consider the extraction process successful.

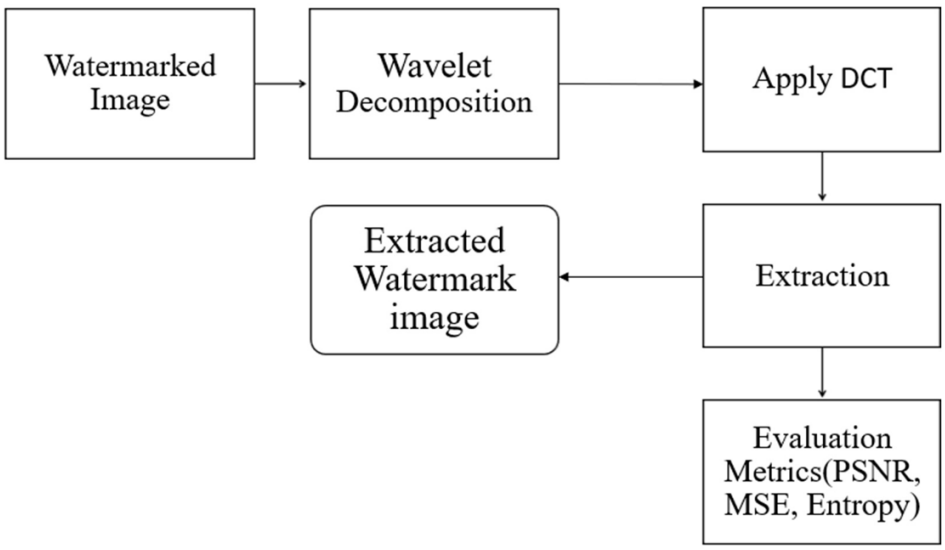


Figure 3: Extracted Watermark.

1.5.2. Discrete wavelet transform (DWT)

The Discrete Wavelet Transform (DWT) is a mathematical technique used for signal and image processing, including applications in data compression, feature extraction, and denoising. DWT operates by decomposing a signal or image into different frequency components at multiple scales. Here's a detailed explanation of the operation of the DWT:

Preparation of Data: DWT begins with a one-dimensional or two-dimensional signal or image as input data. The input signal or image typically has a finite length or size.

Filtering and Down-Sampling (Decomposition): In the decomposition step, the DWT applies a pair of filters known as the low-pass filter (LPF) and the high-pass filter (HPF) to the input data as shown in Figure 4.3. The LPF extracts the low-frequency components from the data, while the HPF extracts the high-frequency components. Low-frequency components often represent the coarse details or approximations of the original data, while high-frequency components represent the fine details or noise. After filtering, the data is down sampled by a factor of 2 in both dimensions. Down-sampling reduces the data size by discarding every alternate sample, effectively reducing the resolution by half.

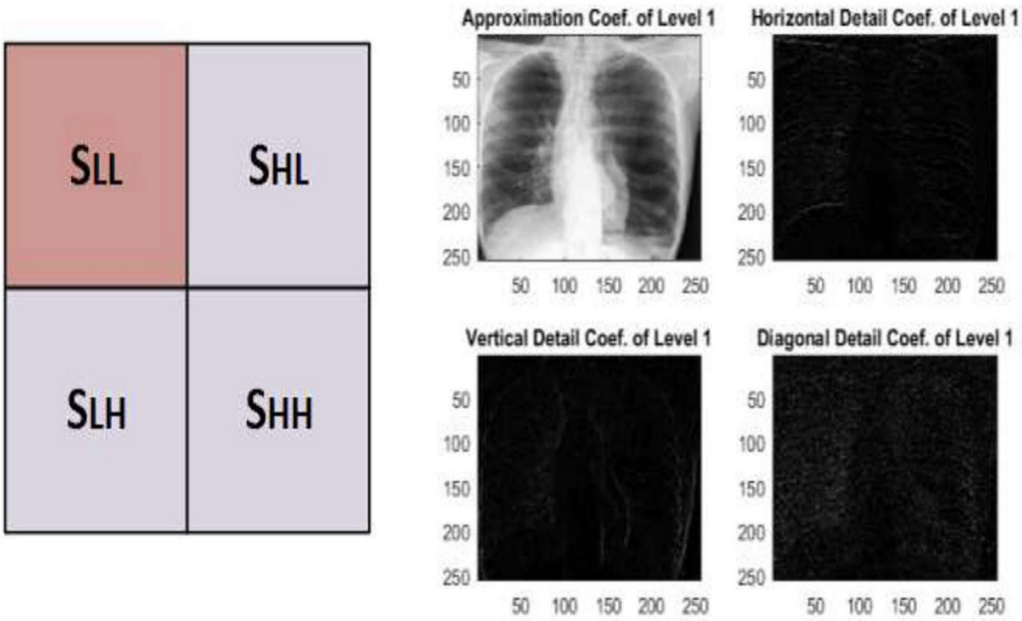


Figure 4: 1-Level Decomposition of DWT

Scaling and Wavelet Coefficients: The output of the DWT decomposition consists of two sets of data: the approximation coefficients (LL) and the detail coefficients (LH, HL, HH). The LL

coefficients represent the lower-scale approximation of the original data, containing the low- frequency information. The LH, HL, and HH coefficients represent the detail information at different scales. LH contains information about low-frequency variations in the vertical direction, HL contains information about low-frequency variations in the horizontal direction, and HH contains high- frequency detail information.

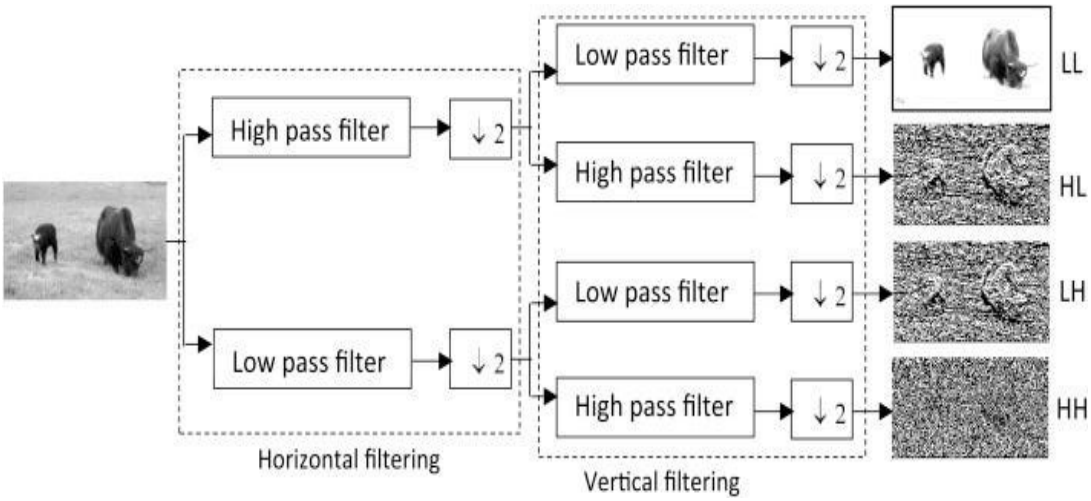


Figure 5: DWT Decomposition.

Multiple Decomposition Levels: The DWT process can be recursively applied to the LL coefficients (approximation coefficients) to obtain further decomposition levels. Each level provides a different level of detail, with LL coefficients becoming lower-resolution approximations at each level.

End of Decomposition: The decomposition process continues until the desired level of detail or the maximum decomposition level is reached.

Reconstruction (Inverse DWT): The original signal or image can be reconstructed from the DWT coefficients as shown in Figure 4.4. This is done by applying the inverse DWT, which involves up-sampling (increasing the resolution) and applying the inverse of the filters used during decomposition.

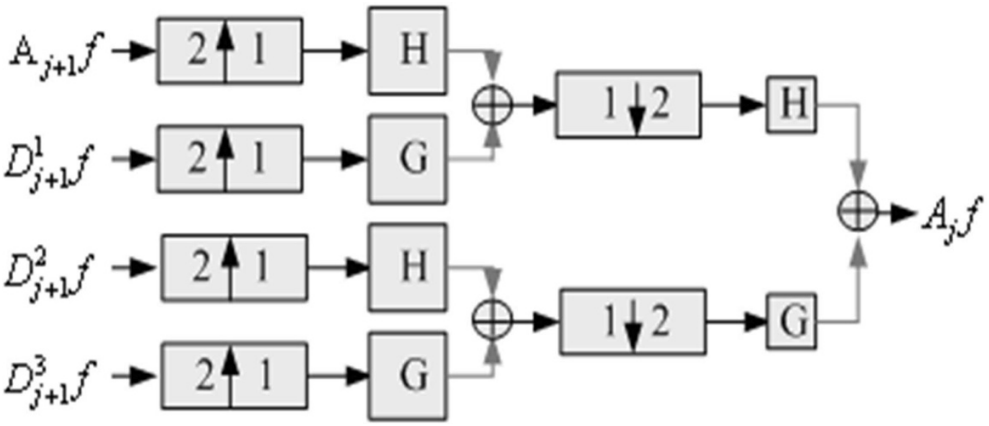


Figure 6: DWT Reconstruction.

1.5.3. Array Conversion

The process of converting an image into an array is a fundamental step in image processing and computer vision tasks. Images are typically represented as two-dimensional grids of pixels, and converting them into arrays makes it easier to perform various mathematical and computational operations on the image data. Here's how the conversion from an image to an array works:

Image Representation: An image is usually represented as a grid of pixels. In the case of a grayscale image, each pixel is represented by a single value denoting its intensity or brightness. In the case of a color image, each pixel is often represented as a combination of three values (Red, Green, and Blue - RGB) to represent the color.

Sampling and Resolution: Images have a finite resolution determined by the number of pixels in the horizontal and vertical dimensions. Higher resolution images have more pixels and capture finer details, but they also require more computational resources.

Array Creation: To convert an image into an array, you create a two-dimensional (2D) array or matrix that matches the dimensions of the image. Each element of the array corresponds to a pixel in the image.

Grayscale Image Conversion: For grayscale images, the pixel values are directly transferred to the array. Each element in the 2D array stores the intensity value of the corresponding pixel. The array becomes a 2D matrix of intensity values.

Array Data Structure: Depending on the programming language and libraries you are using, the array can be represented using native data structures such as lists, NumPy arrays (Python).

1.5.4. Discrete Cosine Transform (DCT)

The Discrete Cosine Transform (DCT) is a mathematical technique used for signal processing and data compression. It's widely applied in various fields, including image and video compression, audio processing, and watermarking. The DCT essentially converts a signal from its spatial or time domain into the frequency domain, revealing the signal's frequency components. Here's a detailed explanation of the operation of the DCT:

Input Signal: The input to the DCT is typically a one-dimensional array or a two-dimensional matrix representing a signal or an image. For example, in image compression, each element of the matrix represents the intensity or color value of a pixel.

Block Division: In many applications, the input signal is divided into smaller blocks or segments. This is common in image and video compression, where blocks of pixels are transformed independently. The size of these blocks depends on the specific application but is often 8x8 or 16x16 pixels.

Normalization (Optional): Before applying the DCT, it's common to normalize the input signal by subtracting a constant value (e.g., 128) from each element. This helps center the signal around zero, which can improve the performance of the DCT.

DCT Coefficient Calculation: The DCT calculates a set of coefficients that represent the frequency components of the input signal. Each coefficient corresponds to a specific frequency component, with lower-order coefficients capturing lower-frequency variations and higher-order coefficients representing higher-frequency details.

$$D(u,v) = C(u)C(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left(\frac{(2x+1)u\pi}{2N}\right)\cos\left(\frac{(2y+1)v\pi}{2N}\right)$$

- The formula for calculating a single DCT coefficient for a given frequency component (u, v) in a block is as follows (for a 2D DCT):

- In this formula, D(u, v) is the DCT coefficient, C(u) and C(v) are scaling constants (typically set to 1/sqrt(2) for u or v equal to 0, and 1 for all other cases), N is the block size, and f(x, y) is the pixel value at position (x, y) in the input block.

Resulting DCT Coefficients: After applying the DCT formula to all possible (u, v) combinations within the block, you obtain a set of DCT coefficients. These coefficients represent the amplitude and phase of various frequency components within the signal or image.

Quantization: In lossy compression applications, such as JPEG image compression, quantization is applied to the DCT coefficients. Quantization reduces the precision of the coefficients, resulting in some information loss. Lower precision leads to higher compression but potentially lower image quality.

Inverse DCT: To reconstruct the original signal or image from the DCT coefficients, you apply the Inverse Discrete Cosine Transform (IDCT). The IDCT reverses the transformation, converting the coefficients back to the spatial or time domain.

Result: The output of the IDCT is the reconstructed signal or image, which should closely resemble the original input. In compression applications, the reconstructed signal may have some loss of detail due to quantization, but the goal is to minimize perceptible differences.

1.5.5. The DCT & DWT transforms

The DCT and DWT transforms have been extensively used in many digital signal processing applications. In this section, we introduce the two transforms briefly, and outline their relevance to the implementation of digital watermarking.

The DCT transform: The discrete cosine transforms is a technique for converting a signal into elementary frequency components. It represents an image as a sum of sinusoids of varying magnitudes and frequencies. With an input image, x, the DCT coefficients for the transformed output image, y, are computed.

In the equation, x, is the input image having N x M pixels, x(m,n) is the intensity of the pixel in row m and column n of the image, and y(u,v) is the DCT coefficient in row u and column v of the DCT matrix. The image is reconstructed by applying inverse DCT operation. The popular block-based DCT transform segments an image non-overlapping blocks and applies DCT to each block. This results in giving three frequency sub-bands: low frequency sub-band, mid-frequency sub-ban and high frequency sub-band. DCT-based watermarking is based on two facts. The first fact is that much of the signal energy lies at low-frequencies sub-band which contains the most important visual parts of the image. The second fact is that high frequency components of the image are usually removed through compression and noise attacks. The watermark is therefore embedded by modifying the

coefficients of the middle frequency sub-band so that the visibility of the image will not be affected and the watermark will not be removed by compression.

The DWT transform: Wavelets are special functions which, in a form analogous to sines and cosines in Fourier analysis, are used as basal functions for representing signals. For 2-D images, applying DWT corresponds to processing the image by 2-D filters in each dimension. The filters divide the input image into four non-overlapping multi-resolution sub-bands LL1, LH1, HL1 and HH1. The sub-band LL1 represents the coarse-scale DWT coefficients while the sub-bands LH1, HL1 and HH1 represent the fine-scale of DWT coefficients. To obtain the next coarser scale of wavelet coefficients, the sub-band LL1 is further processed until some final scale N is reached. When N is reached we will have 3N+1 sub-bands consisting of the multi-resolution sub bands LLN and LHx, HLx and HHx where x ranges from 1 until N. Due to its excellent spatio-frequency localization properties, the DWT is very suitable to identify the areas in the host image where a watermark can be embedded effectively. In particular, this property allows the exploitation of the masking effect of the human visual system such that if a DWT coefficient is modified, only the region corresponding to that coefficient will be modified. In general most of the image energy is concentrated at the lower frequency sub-bands LLx and therefore embedding watermarks in these sub-bands may degrade the image significantly. Embedding in the low frequency sub-bands, however, could increase robustness significantly. On the other hand, the high frequency sub-bands HHx include the edges and textures of the image and the human eye is not generally sensitive to changes in such sub-bands. This allows the watermark to be embedded without being perceived by the human eye. The compromise adopted by many DWT-based watermarking algorithm, is to embed the watermark in the middle frequency sub-bands LHx and HLx where acceptable performance of imperceptibility and robustness could be achieved.

1.5.6. Local binary pattern (LBP)

Ojala et al. first developed LBP, initially utilized to calculate the local contrast in analysis of texture in images. LBP is utilized in many fields of image and video processing like text analysis, image authentication and image forgery detection due to its property of efficient texture feature descriptor. LBP breaks down an image into multiple sub-blocks of size n × n. The centre pixel value is utilized as a threshold value to decide the neighbouring pixel values by setting the smaller values as 0 and remaining as 1 by comparing with centre pixel, i.e., threshold value. The clockwise values of the binary values are considered and converted to decimal form. The most important property of the LBP function in real world applications is its robustness to monotonic gray-scale changes caused by illumination variations compared to other features. Other advantage of LBP is it has high discriminative power with simple computation.

90	75	75
115	80	70
120	60	80

(a)

1	0	0
1		0
1	0	1

Binary: 10001011

Decimal: 139

(b)

Figure 7: 3 × 3 Block LBP operator a Image block and b Local Binary Pattern of (a) block

1.5.7. Arnold transform

The Arnold Transform, also known as the Arnold Cat Map, is a mathematical operation used in the field of chaos theory and dynamical systems. It was introduced by Vladimir Arnold in the 1960s. This transform is a simple, yet interesting, way to exhibit chaotic behavior in a discrete dynamical system. In the propounded watermarking scheme, Arnold Cat Transform is endorsed to provide assurance about the security of the scheme. The controls panels a and b will be used to change the position of the image pixels, and N is the size of the image. For different image sizes and parameters, period T will be different in Arnold transform. The image pixels will be back to its native position after certain permutations. Here, in applying the Arnold transform, the image gets scrambled and also we can use the T value as a key to provide better security to the scheme. The Arnold transform operates on a 2D grid or image. Given an n×n grid or an n×n image represented as a matrix, the Arnold transform rearranges the pixels according to the following equations:

$$x' = (2x + y) \bmod n$$

$$y' = (x + y) \bmod n$$

Where x and y are the original pixel coordinates. x ′ and y ′ are the transformed pixel coordinates. mod n represents the modulo operation with respect to the size of the grid.

Behavior: The Arnold transform is an area-preserving transformation, meaning it preserves the area of shapes in the image. It is a chaotic map, which means small changes in the input can lead to drastic

changes in the output. Repeatedly applying the Arnold transform can mix up the pixels of an image in a way that appears random, yet is deterministic.

1.5.8. The propounded watermarking design

In the propounded watermarking scheme, approximation coefficients of LWT and SLL lower resolution approximation module of DWT is utilized in immersing the fingerprint watermark of the patient because of maximum energy of the image is strenuous in low resolution approximation and also more robust and efficient to attacks of image and signal processing. Immersing the fingerprint watermark in the SLL is highly perceivable for human eye. The propounded watermark embedding design using the combination of LWT–DWT with LBP feature values and semi-blind watermark extraction using the keys are given in the following subsections. The idea of applying two transform or hybrid transform is based on the fact that combined transforms could compensate the drawbacks of each other, resulting in effective watermarking. The LBP features are considered for calculating scaling and embedding factor adaptively because of its robustness to monotonic gray-scale changes caused by illumination variations compared to other features. The reason for combination of LWT– DWT combination can be observed from the below Table. A medical image has been tested with combination of 2 level DWT, 2-level LWT and combination of LWT–DWT with various attacks. From the results, it is clear that the combination is robust to attacks compared to their transformations. For calculating the embedding factor, LBP features are utilized since it extracts texture features of an image which is robustness to monotonic grayscale changes.

Propounded watermark embedding design using LWT–DWT–LBP:

In this embedding scheme, adaptive watermark is embedded in the hybrid transform of medical image with patient fingerprint watermark.

Algorithm 1: Propounded Watermark Embedding Algorithm

HMI f pw=Medical watermark_embedding (HMI, FPW), Input: Host Medical Image (HMI), Fingerprint Watermark (FPW), Output: Scrambled Medical Watermarked image (AHMI f pw)

1: Read HMI, FPW

2: [CA, CH, CV, CD] = LWT(HMI)

3: [LL, LH, HL, HH] = DWT(CA)

4: $\alpha = \mu(\text{LBPFeatures(HMI)}), \beta = (1 - \alpha)$

5: $\text{HMI} = \alpha \times \text{LL} + \beta \times \text{FPW}$

6: LL1' = IDWT(HMI , LH, HL, HH)

7: HMI f pw = IDWT(LL1, CH, CV, CD)

8: AHMI f pw=Arnold(HMI f pw)

In the above propounded watermark embedding Algorithm 1, HMI represents host medical image, FPW represents fingerprint watermark, and μ represents the mean of the LBP features. The steps of embedding fingerprint watermark into the medical image are given below.

Step 1: Scan the fingerprint watermark and host medical image.

Step 2: Applying LWT for 1–Level to host medical image produces approximation coefficients (CA) and details coefficients (CH, CV, CD)

Step 3: For the approximation coefficients (CA), 1–Level DWT is applied by producing low (LL), diagonal (LH, HL) and high (HH) resolution coefficients.

Step 4: Low-resolution approximation (LL) is considered for embedding fingerprint watermark using the mean of the LBP features of host medical image

Step 5: Embedding of the fingerprint watermark is done using the embedding and scaling factor as represented in Algorithm 1 and is also shown below where α and β are scaling and embedding factor values.

$$HMI = \alpha \times LL + \beta \times FPW$$

Step 6: Inverse DWT is applied by combining watermarked LL sub band with remaining sub bands.

Step 7: Inverse LWT is applied by combining watermarked CA with remaining coefficients to form Imperceptible Watermarked Medical Image.

Step 8: Further, to improve security, Arnold transform is applied to the watermarked medical image with a secret key in generating the scrambled watermarked medical image.

The function of Arnold transform is to scramble the image so that the intruders cannot know the image. The reason for adding at the end of the embedding is to overcome tampering of medical images. To add extra security to the host medical image Arnold Transform is applied at the end of the process.

Model/attacks	DWT 2-Level	LWT 2-Level	LWT-DWT
S & P Attack	0.9920	0.9925	0.9935
Gaussian Attack	0.9605	0.9617	0.9673
Scaling Attack	0.9980	0.9985	1.00
Rotation	0.9905	0.9912	0.9956
Croping	0.9895	0.9908	0.9937
Mean Filtering	0.9669	0.9648	0.9797

Table 1: Reason for selecting the combination of transformations

Propounded watermark extraction design using LWT–DWT–LBP

In this extraction scheme, adaptive patient fingerprint watermark is extracted in the hybrid transform from watermarked medical image.

Algorithm 2: Propounded Watermark Extraction Algorithm

f pwE =Medical watermark_extraction (AHMI f pw, LL, α), Input: Imperceptible watermarked Medical Image (AHMI f pw), LL of host medical image, Scaling and Embedding Factor, Secret Key, Output: Extracted finger print watermark (f pwE)

1: Read AHMI f pw

2: HMI f pw= Inverse Arnold(AHMI f pw)

3: [CAw,CHw,CVw,CDw] = LWT(HMI f pw)

4: [LLw, LHw, HLw, HHw] = DWT(CAw)

5: f pwE = LLw−(α×LL)

In the above propounded watermark extraction Algorithm 2, AHMI f pw represents scrambled watermarked host medical image, LL represents the decomposition of the watermark with LWT followed by DWT subband, and α&β represents the embedding and scaling factor. The steps of extraction of fingerprint watermark from imperceptible watermarked medical image are given below.

Step 1: Scan the scrambled watermarked host medical image(AHMI f pw)

Step 2: Apply inverse Arnold transform with secret key(Key-3) to descramble the medical image in generating watermarked host medical image.

Step 3: Applying LWT for 1–Level towatermarked hostmedical image produces watermarked approximation coefficients (CAw) and watermarked details coefficients ((CHw, CVw, CDw))

Step 4: For the watermarked approximation coefficients (CAw), 1–Level DWT is applied by producing watermarked low (LLw), diagonal (LHw, HLw) and watermarked high (HHw) resolution coefficients.

Step 5: Watermarked low-resolution approximation (LLw) is considered for extracting fingerprint watermark by using the same keys (Key-1 & Key-2) that are used in embedding the watermark represented in Algorithm 2.

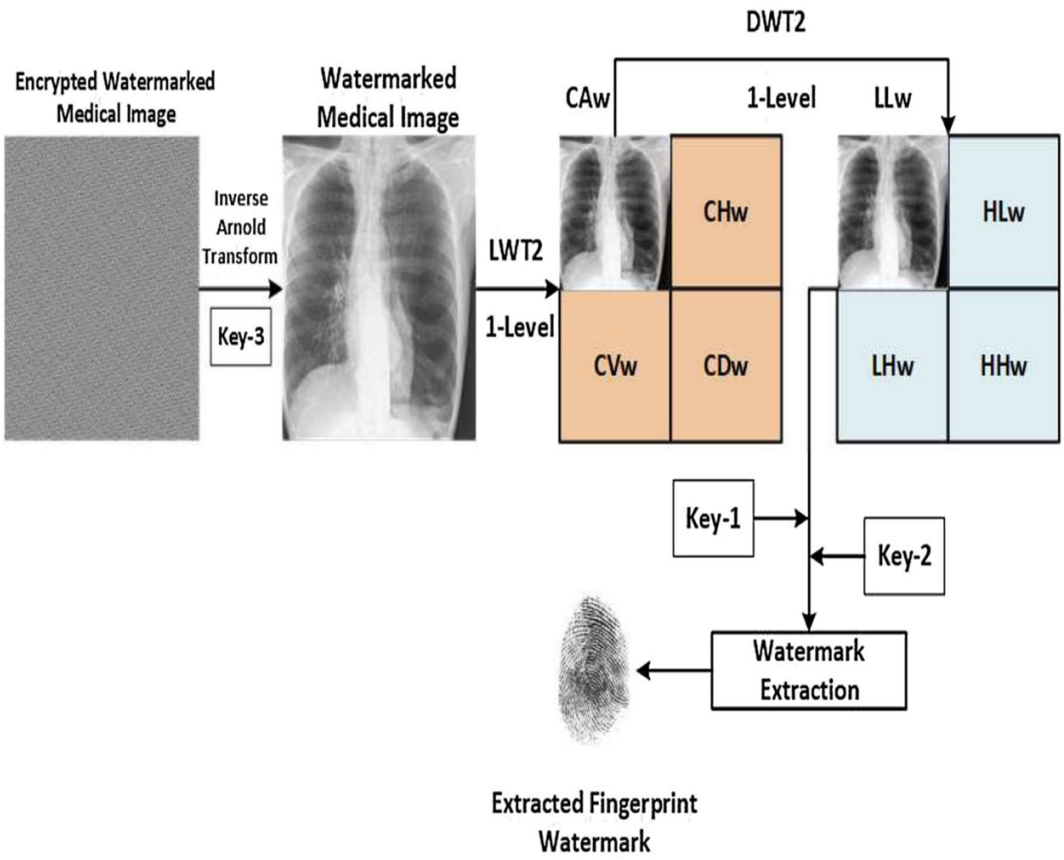


Figure 8: Propounded Watermark Extraction Design using LWT–DWT–LBP

1.5.9. Advantages

The watermark extraction process depicted in Figure 4.2 offers several notable advantages in the context of medical image watermarking, each contributing to its effectiveness and utility:

- High Fidelity Extraction: One of the primary advantages is the ability to achieve 'High Fidelity Extraction.' The use of advanced techniques like wavelet decomposition and DCT ensures that the embedded watermark can be accurately and reliably retrieved. This high fidelity is crucial in medical

applications where diagnostic accuracy is paramount, as any distortion or loss of information can impact patient care.

- Robustness: The process exhibits 'Robustness' against common image processing operations and potential attacks. By applying wavelet decomposition and DCT, the watermark becomes less susceptible to distortions caused by compression, noise, or other manipulations. This robustness ensures that the watermark remains intact and detectable even in challenging conditions.

- Multi-Metric Evaluation: The incorporation of 'Evaluation Metrics' such as PSNR, MSE, and entropy provides a comprehensive means of assessing the quality and integrity of the extracted watermark. These metrics enable researchers and practitioners to quantitatively measure the success of the extraction process, ensuring that the watermark remains discernible while minimizing any adverse effects on the image.

- Adaptability: The process can be adapted to different types of watermarks and medical image modalities. Whether the watermark is an image, text, or other data, the combination of wavelet decomposition and DCT offers flexibility in handling diverse watermark types. Moreover, it can be applied to various medical image formats, including X-rays, MRIs, and CT scans.

- Detection Reliability: The use of wavelet decomposition and DCT enhances the 'Detection Reliability' of the watermark. The transformation of the image data into the frequency domain makes it easier to identify the watermark's presence and characteristics, reducing the likelihood of false positives or negatives during extraction.

- Quality Preservation: The process is designed to 'Preserve Image Quality' to the greatest extent possible. While the watermark is embedded and extracted, the impact on the overall quality of the medical image is minimized. This ensures that the watermarked image maintains its diagnostic value, making it suitable for clinical use.

- Security and Authentication: The watermark extraction process is instrumental in 'Security and Authentication.' It helps verify the authenticity of medical images, ensuring that they have not been tampered with or altered. This is vital in healthcare settings to maintain the trustworthiness of patient records and diagnostic results.

- Transparency: Importantly, the watermark extraction process is 'Transparent' to medical professionals. The presence of the watermark should not hinder their ability to interpret and diagnose medical images. By preserving image quality and ensuring that the watermark is imperceptible to the human eye, the process maintains transparency in clinical workflows.

1.6. Input and Output Design

Input Design

Blind medical image watermarking has emerged as a crucial technique for securing medical images in the era of the Internet of Medical Things (IoMT). As medical data transmission and sharing become increasingly common in IoMT applications, ensuring the integrity, authenticity, and confidentiality of sensitive medical images is paramount. In this context, a combined Discrete Wavelet Transform (DWT) and Discrete Cosine Transform (DCT) model-based watermarking approach presents a promising solution to address the unique challenges posed by medical image security. The Discrete Wavelet Transform (DWT) is a widely used tool in image processing for its ability to decompose images into multiple frequency bands, thereby capturing both spatial and frequency information efficiently. DWT enables the extraction of image features at different scales, making it suitable for watermark embedding while preserving the image's diagnostic information. By decomposing the medical image into its frequency components using DWT, the watermark can be embedded in the selected frequency domain to ensure robustness against various attacks.

However, embedding a watermark solely based on DWT may not achieve optimal robustness and imperceptibility simultaneously. The Discrete Cosine Transform (DCT) offers complementary advantages by representing image blocks in the frequency domain, which is more resilient to compression and other transformations commonly encountered in medical image processing. By combining DWT and DCT, we leverage the strengths of both transforms to enhance the watermarking scheme's robustness and imperceptibility.The proposed watermarking scheme follows a blind approach, meaning that the original image is not required during the watermark extraction process, which is essential for medical applications where access to the original image may be restricted due to privacy concerns. The watermark, typically a unique identifier or a cryptographic signature, is embedded into the medical image in a reversible and imperceptible manner to ensure that it does not degrade the image quality or affect its diagnostic value.

The watermark embedding process involves several key steps. Firstly, the medical image undergoes DWT decomposition to obtain its frequency components. Subsequently, suitable frequency bands are selected based on their energy characteristics and perceptual importance for watermark embedding. The watermark is then embedded into the selected frequency coefficients using a robust and secure algorithm, ensuring that it remains intact even after various attacks such as noise addition, compression, and geometric transformations.

To further enhance the robustness of the watermarking scheme, robustness measures such as error correction coding and spread spectrum techniques can be employed. Error correction coding adds

redundancy to the watermark data, enabling the extraction of the original watermark even if some parts of it are corrupted during attacks. Spread spectrum techniques distribute the watermark energy across the image, making it more resilient to localized attacks and ensuring that the watermark remains perceptually invisible. Evaluation of the proposed watermarking scheme involves comprehensive testing against a variety of attacks and performance metrics. Commonly used metrics include Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSI), Bit Error Rate (BER), and Normalized Cross-Correlation (NCC). The watermarking scheme should demonstrate high robustness against attacks while maintaining low distortion and high imperceptibility to preserve the diagnostic quality of medical images.

In conclusion, the combined DWT-DCT model-based blind medical image watermarking scheme offers a robust and efficient solution for securing medical images in IoMT applications. By leveraging the complementary advantages of DWT and DCT, the proposed approach ensures robustness, imperceptibility, and reversibility, making it suitable for safeguarding sensitive medical data in the digital healthcare ecosystem. Further research and development in this area are essential to address emerging security challenges and advance the state-of-the-art in medical image watermarking for IoMT applications.

Output Design

In the realm of the Internet of Medical Things (IoMT), safeguarding the integrity and confidentiality of medical images is paramount. To address this challenge, we propose a novel approach: a combined Discrete Wavelet Transform (DWT) and Discrete Cosine Transform (DCT) model-based blind medical image watermarking system. This system is meticulously designed to embed imperceptible watermarks into medical images, ensuring their authenticity and integrity without compromising diagnostic quality.At the heart of our watermarking technique lies the fusion of DWT and DCT, two powerful image processing transforms. The DWT enables the decomposition of medical images into multiple frequency bands, preserving both spatial and frequency information crucial for diagnosis. Meanwhile, the DCT excels in representing image blocks in the frequency domain, enhancing resilience to common transformations encountered in medical image processing, such as compression. By synergizing these transforms, our approach capitalizes on their respective strengths, thereby fortifying the watermarking scheme's robustness and imperceptibility. The watermark embedding process comprises several critical stages. Initially, the medical image undergoes DWT decomposition to extract its frequency components. Subsequently, frequency bands are selected based on their energy characteristics and perceptual importance for watermark embedding. The watermark is then seamlessly embedded into these selected frequency coefficients using a robust and secure

algorithm. This meticulous embedding process ensures the watermark's resilience against various attacks, including noise addition, compression, and geometric transformations, while maintaining imperceptibility.

To further fortify the watermarking scheme, we employ robustness measures such as error correction coding and spread spectrum techniques. Error correction coding adds redundancy to the watermark data, enabling the recovery of the original watermark even in the face of partial corruption. Spread spectrum techniques distribute the watermark's energy across the image, bolstering resilience against localized attacks and preserving perceptual invisibility.

Evaluation of our proposed watermarking scheme encompasses rigorous testing against diverse attacks and performance metrics. Key metrics include Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSI), Bit Error Rate (BER), and Normalized Cross-Correlation (NCC). The watermarking scheme is expected to demonstrate high robustness against attacks while maintaining low distortion and high imperceptibility, thus safeguarding the diagnostic quality of medical images. In conclusion, our combined DWT-DCT model-based blind medical image watermarking system offers a secure and efficient solution for safeguarding medical images in IoMT applications. By harnessing the synergistic power of DWT and DCT, our approach ensures robustness, imperceptibility, and reversibility, thereby preserving the integrity and confidentiality of sensitive medical data in the digital healthcare landscape. Further research and development in this domain are imperative to tackle emerging security challenges and propel the advancement of medical image watermarking for IoMT applications.

2.LITERATURE SURVEY

Lin CC, Lee TL, Chang YF, Shiu PF, Zhang B (2023) Fragile Watermarking for Tamper Localization and Self-Recovery Based on AMBTC and VQ. Electronics 12:415. https://doi.org/10.3390/electronics12020415. [1]

Digital images have unique features that include being both easily transmittable over the Internet and being easy to tamper. With the advancement of digital processing techniques and an increasing number of valuable digital images being transmitted via the Internet, image authentication has been made more crucial than ever. In this paper, we present an image authentication scheme with tamper localization and self-recovery using fragile watermarking. We embed the fragile watermarks consisting of the authentication code and the recovery information onto the image to verify its integrity.

A. Soualmi, A. Alti, and L. Laouamer, ‘‘A novel blind medical image watermarking scheme based on Schur triangulation and chaotic sequence,’’ Concurrency Comput., Pract. Exper., vol. 34, no. 1, Jan. 2022, Art. no. e6480. [2]

Image watermarking is a potentially effective and powerful solution for multimedia security. It is a technique that plays a fundamental role in protecting copyright, proofing ownership, and helps to authenticate sensitive data. Watermarking can cause a decrease in image quality, especially when it comes to medical images. This could provoke false diagnoses, which can lead to serious consequences on the patient's health. In fact, it becomes necessary to think for new more imperceptible, secure watermarking techniques.

Moad MS, Kafi MR, Khaldi A (2022) A wavelet based medical image watermarking scheme for secure transmission in telemedicine applications, Microprocess Microsyst, 90, https://doi.org/10.1016/j.micpro.2022.104490. [3]

In this paper, the authors introduce a novel wavelet-based medical image watermarking scheme designed to enhance the security of image transmission in telemedicine applications. With the increasing adoption of telemedicine technologies for remote diagnosis and healthcare delivery, ensuring the confidentiality and integrity of medical images during transmission is critical. The proposed watermarking scheme leverages wavelet transform techniques to embed imperceptible watermarks into medical images. Wavelet transform offers advantages such as multi-resolution analysis and localization of image features, making it suitable for preserving both image quality and

watermark robustness. By embedding watermarks directly into the wavelet coefficients of medical images, the proposed scheme enables secure transmission without compromising diagnostic integrity.

Singh P, Devi KJ, Thakkar HK, Kotecha K (2022) Region-Based Hybrid Medical Image Watermarking Scheme for Robust and Secured Transmission in IoMT. IEEE Access 10:8974–8993. https://doi.org/10.1109/ACCESS.2022.3143801. [4]

In this paper, the authors propose a region-based hybrid medical image watermarking scheme aimed at ensuring robust and secured transmission in the Internet of Medical Things (IoMT). With the proliferation of connected medical devices and the increasing reliance on digital health technologies, the need for secure and reliable transmission of medical images has become paramount. The proposed watermarking scheme combines region-based techniques with hybrid watermark embedding strategies to achieve robustness against common attacks while preserving diagnostic accuracy. By partitioning the medical images into regions of interest and embedding watermarks selectively based on region characteristics, the scheme ensures both security and imperceptibility. Furthermore, the hybrid watermark embedding strategies incorporate multiple techniques, such as spread spectrum modulation and quantization index modulation (QIM), to enhance the robustness and resilience of the embedded watermarks.

H. S. Alshanbari, ‘‘Medical image watermarking for ownership & tamper detection,’’ Multimedia Tools Appl., vol. 80, no. 11, pp. 16549–16564, May 2021. [5]

Image watermarking can provide ownership identification as well as tamper protection. Transform domain based image watermarking has been proven to be more robust than the spatial domain watermarking against different signal processing attacks. On the other hand, tamper detection is found to be working well in spatial domain. In the proposed work, the focus is on the improvement of the medical image watermarking by incorporating the concept of multiple watermarking of the host image. The principal components (PC) based insertion make the scheme secured towards ownership attack. On the other hand, LZW (Lempel–Ziv–Welch) based fragile watermarking is used to hide compressed image’s ROI (region of interest) to tackle the intentional tampering attacks.

S. A. Parah, J. A. Kaw, P. Bellavista, N. A. Loan, G. M. Bhat, K. Muhammad, and V. H. C. de Albuquerque, ‘‘Efficient security and authentication for edge-based Internet of Medical Things,’’ IEEE Internet Things J., vol. 8, no. 21, pp. 15652–15662, Nov. 2021. [6]

Internet of Medical Things (IoMT)-driven smart health and emotional care is revolutionizing the healthcare industry by embracing several technologies related to multimodal physiological data

collection, communication, intelligent automation, and efficient manufacturing. The authentication and secure exchange of electronic health records (EHRs), comprising of patient data collected using wearable sensors and laboratory investigations, is of paramount importance. In this article, we present a novel high payload and reversible EHR embedding framework to secure the patient information successfully and authenticate the received content. The proposed approach is based on novel left data mapping (LDM), pixel repetition method (PRM), RC4 encryption, and checksum computation. The input image of size [Formula: see text] is upscaled by using PRM that guarantees reversibility with lesser computational complexity. The binary secret data are encrypted using the RC4 encryption algorithm and then the encrypted data are grouped into 3-bit chunks and converted into decimal equivalents.

A. Nagm and M. S. Elwan, ‘‘Protection of the patient data against intentional attacks using a hybrid robust watermarking code,’’ PeerJ Comput. Sci., vol. 7, p. e400, Mar. 2021. [7]

The security of patient information is important during the transfer of medical data. A hybrid spatial domain watermarking algorithm that includes encryption, integrity protection, and steganography is proposed to strengthen the information originality based on the authentication. The proposed algorithm checks whether the patient's information has been deliberately changed or not. The created code is distributed at every pixel of the medical image and not only in the regions of non-interest pixels, while the image details are still preserved. To enhance the security of the watermarking code, SHA-1 is used to get the initial key for the Symmetric Encryption Algorithm. The target of this approach is to preserve the content of the image and the watermark simultaneously, this is achieved by synthesizing an encrypted watermark from one of the components of the original image and not by embedding a watermark in the image.

V. Rajput and I. A. Ansari, ‘‘Image tamper detection and self- recovery using multiple median watermarking,’’ Multimedia Tools Appl., vol. 79, nos. 47–48, pp. 35519–35535, Dec. 2020, doi: 10.1007/ s11042-019-07971-w. [8]

Photographs play a very crucial role in our lives, be it in the field of forensic investigation, military intelligence, scientific research, and publications. Nowadays, most of these photographs are in the digital format; which can be easily edited in any photo editing software without requiring any special knowledge of the field. It has become quite hard to identify whether an image is real or fake. This can be very crucial in the cases of forensic investigation or authorization of images. So, we need a solution, which not only identifies the attacks from different schemes like collage attack, crop attack, etc. but also recovers the edited or tampered portion.

Memon NA, Alzahrani A (2020) Prediction-Based Reversible Watermarking of CT Scan Images for Content Authentication and Copyright Protection. IEEE Access 8:75448–75462. [9]

In this paper, the authors present a prediction-based reversible watermarking method tailored for CT scan images to provide content authentication and copyright protection. CT scan images are widely used in medical diagnostics, making their security and integrity critical for accurate diagnosis and treatment. The proposed reversible watermarking technique utilizes prediction-based methods to embed watermarks into CT scan images without altering the original pixel values significantly. Reversible watermarking ensures that the original image can be completely restored after the watermark is extracted, making it suitable for medical imaging applications where image fidelity is paramount. By embedding watermarks based on predictions of pixel values, the proposed method achieves content authentication and copyright protection without introducing noticeable distortions in the CT scan images.

Kumar L, Singh KU (2020) An Analysis of Different Watermarking Schemes for Medical Image Authentication. Eur J Mol Clin Med 7(4):2250–2259. [10]

In this paper, the authors conduct an analysis of various watermarking schemes specifically designed for medical image authentication. With the increasing digitization of medical records and the reliance on digital imaging technologies in healthcare, ensuring the authenticity and integrity of medical images has become essential for accurate diagnosis and treatment. The authors review and compare different watermarking schemes proposed in the literature, evaluating their effectiveness in authenticating medical images while preserving image quality and diagnostic accuracy. The analysis encompasses a range of watermarking techniques, including spatial domain, transform domain, and hybrid methods, each offering unique advantages and trade-offs in terms of robustness, imperceptibility, and computational complexity.

M. Begum and M. S. Uddin, ‘‘Analysis of digital image watermarking techniques through hybrid methods,’’ Adv. Multimedia, vol. 2020, pp. 1–12, Aug. 2020. [11]

In this paper, the authors conduct an analysis of digital image watermarking techniques, focusing specifically on hybrid methods. With the proliferation of digital media and the increasing importance of content protection and authentication, watermarking techniques play a crucial role in safeguarding intellectual property and ensuring data integrity. The paper explores various hybrid watermarking techniques, which combine multiple methods or algorithms to achieve improved robustness, imperceptibility, and security. Hybrid methods often integrate spatial domain, frequency domain, and

transform domain techniques, leveraging the strengths of each approach to overcome their respective limitations. Through their analysis, the authors evaluate the performance of different hybrid watermarking techniques in terms of robustness against common attacks such as compression, noise addition, and geometric transformations, as well as imperceptibility and payload capacity. They also discuss the computational complexity and practical considerations associated with implementing these techniques.

F. Sabbane and H. Tairi, ‘‘Medical image watermarking technique based> on polynomial decomposition,’’ Multimedia Tools Appl., vol. 78, no. 23, pp. 34129–34155, Dec. 2019. [12]

In this paper, Sabbane and Tairi propose a novel technique for watermarking medical images based on polynomial decomposition. With the increasing use of digital medical imaging technologies, ensuring the security and integrity of medical images has become crucial for accurate diagnosis and treatment. The proposed watermarking technique utilizes polynomial decomposition, a mathematical method that breaks down an image into polynomial coefficients. By embedding watermarks into these coefficients, the technique enables the authentication and protection of medical images without significantly altering their visual appearance. The authors evaluate the performance of the watermarking technique in terms of robustness against common attacks such as compression, noise addition, and geometric transformations. They also assess the imperceptibility of the watermarked images to ensure that the embedded watermarks do not degrade the diagnostic quality of the medical images.

B. Hassan, R. Ahmed, B. Li, and O. Hassan, ‘‘An imperceptible medical image watermarking framework for automated diagnosis of retinal pathologies in an eHealth arrangement,’’ IEEE Access, vol. 7, pp. 69758–69775, 2019. [13]

In this paper, Hassan et al. propose an imperceptible medical image watermarking framework designed specifically for the automated diagnosis of retinal pathologies in an eHealth arrangement. With the increasing adoption of electronic health (eHealth) systems and telemedicine, there is a growing need for secure and reliable transmission of medical images while ensuring the integrity and authenticity of diagnostic results. The proposed watermarking framework focuses on retinal images, which are crucial for diagnosing various eye diseases and conditions. By embedding imperceptible watermarks into retinal images, the framework enables the authentication and protection of these images throughout the diagnostic process, including transmission, storage, and analysis. The authors evaluate the performance of the watermarking framework in terms of imperceptibility, robustness against common attacks, and its impact on the accuracy of automated diagnosis algorithms. They

demonstrate that the embedded watermarks do not degrade the visual quality of the retinal images and do not interfere with the diagnostic accuracy of automated algorithms.

Gull S, Loan NA, Parah SA, Sheikh JA, Bhat G (2018) An efficient watermarking technique for tamper detection and localization of medical images. J Ambient Intell Humaniz Comput 11:1799–1808. [14]

In this paper, Gull et al. introduce an efficient watermarking technique designed specifically for tamper detection and localization in medical images. With the increasing digitization of healthcare and the widespread use of digital medical imaging technologies, ensuring the integrity and authenticity of medical images is crucial for accurate diagnosis and treatment. The proposed watermarking technique embeds imperceptible watermarks into medical images, which serve as digital signatures for authentication and tamper detection. By strategically placing these watermarks within the image data, the technique enables the detection and localization of any unauthorized modifications or tampering. The authors evaluate the performance of the watermarking technique in terms of its effectiveness in detecting and localizing tampering, as well as its robustness against common attacks such as compression, noise addition, and geometric transformations. They demonstrate that the embedded watermarks remain intact even after these attacks, enabling reliable tamper detection and localization.

A. Shehab, M. Elhoseny, K. Muhammad, A. K. Sangaiah, P. Yang, H. Huang, and G. Hou, ‘‘Secure and robust fragile watermarking scheme for medical images,’’ IEEE Access, vol. 6, pp. 10269–10278, 2018. [15]

In this paper, Shehab et al. present a secure and robust fragile watermarking scheme specifically designed for medical images. With the increasing reliance on digital medical imaging technologies, ensuring the security and integrity of medical images has become paramount for accurate diagnosis and treatment. The proposed watermarking scheme employs fragile watermarking techniques, which are highly sensitive to any modifications or tampering in the image data. Fragile watermarks serve as digital signatures that can detect even the slightest alterations in the medical images, thereby ensuring their authenticity and integrity. The authors evaluate the performance of the watermarking scheme in terms of its robustness against common attacks such as compression, noise addition, and geometric transformations, as well as its effectiveness in detecting and localizing tampering. They demonstrate that the embedded watermarks can accurately detect and localize any unauthorized modifications or tampering in the medical images, providing an additional layer of security and trustworthiness.

A. D. Andrushia and R. Thangarajan, ‘‘An efficient visual saliency detection model based on Ripplet transform,’’ Sádhanã, vol. 42, no. 5, pp. 671–685, May 2017. [16]

In this paper, Andrushia and Thangarajan propose an efficient visual saliency detection model based on the Ripplet transform. Visual saliency detection plays a crucial role in various computer vision applications, including object recognition, image retrieval, and visual attention modeling. The Ripplet transform is utilized as a feature extraction technique to capture the salient regions in an image effectively. Unlike traditional methods that rely on simple intensity or color contrast, the Ripplet transform can extract more discriminative features, leading to more accurate saliency detection. The authors evaluate the performance of their proposed model using benchmark datasets and compare it with existing saliency detection approaches. They demonstrate that the Ripplet transform-based model achieves superior performance in terms of accuracy and computational efficiency.

S. Maheshkar, ‘‘Region-based hybrid medical image watermarking for secure telemedicine applications,’’ Multimedia Tools Appl., vol. 76, no. 3, pp. 3617–3647, Feb. 2017. [17]

In this paper, Maheshkar presents a region-based hybrid medical image watermarking technique tailored for secure telemedicine applications. With the growing adoption of telemedicine, there is an increasing need for robust and secure methods to transmit medical images over networks while maintaining patient privacy and data integrity. The proposed watermarking technique combines region-based and hybrid methods to embed watermarks into medical images effectively. By dividing the image into regions of interest, the watermarking process focuses on preserving the diagnostic information in critical areas while ensuring robustness against common attacks. The author evaluates the performance of the watermarking technique in terms of its ability to withstand compression, noise addition, and other forms of image manipulation without compromising the integrity of the embedded watermarks. Additionally, the technique's impact on image quality and diagnostic accuracy is assessed to ensure compatibility with telemedicine applications.

M. Sharma, ‘‘Medical image watermarking technique in the application of E-diagnosis using M-Ary modulation,’’ Proc. Comput. Sci., vol. 85, pp. 648–655, Jan. 2016. [18]

In this paper, Sharma introduces a medical image watermarking technique specifically designed for the application of E-diagnosis using M-Ary modulation. With the advancement of e-healthcare technologies, there is a growing need for secure and reliable methods to transmit medical images for remote diagnosis and consultation. The proposed watermarking technique utilizes M-Ary modulation, a digital modulation technique that encodes information into a digital signal with multiple amplitude

levels, to embed watermarks into medical images. By modulating the pixel intensities of the image, the technique embeds imperceptible watermarks that carry diagnostic information or metadata related to the image. The author evaluates the performance of the watermarking technique in terms of its robustness against common attacks such as compression, noise addition, and geometric transformations. Additionally, the impact of the watermarking process on image quality and diagnostic accuracy is assessed to ensure compatibility with E-diagnosis applications.

Vaidya, S.P., Chandra Mouli, P.V.S.S.R.: Adaptive digital watermarking for copyright protection of digital images in wavelet domain. Procedia Comput. Sci. 58, 233–240 (2015). [19]

In this paper, Vaidya and Chandra Mouli propose an adaptive digital watermarking technique specifically designed for copyright protection of digital images in the wavelet domain. With the proliferation of digital media and the ease of image replication and distribution enabled by digital technologies, there is a growing need for effective methods to protect the intellectual property rights of digital content creators. The proposed watermarking technique operates in the wavelet domain, leveraging the multi-resolution analysis capabilities of wavelet transforms to embed imperceptible watermarks into digital images. By adapting the watermark embedding process based on image characteristics and copyright requirements, the technique ensures robustness against common attacks while minimizing perceptual distortion.

Singh, A.K., Kumar, B., Dave, M., Mohan, A.: Robust and imperceptible dual watermarking for telemedicine applications. Wirel. Pers. Commun. 80(4), 1415–1433 (2015). [20]

In this paper, Singh et al. propose a robust and imperceptible dual watermarking technique tailored for telemedicine applications. With the increasing adoption of telemedicine, there is a growing need for secure and reliable methods to transmit medical images over networks while ensuring the integrity and authenticity of the data. The proposed dual watermarking technique embeds two distinct watermarks into medical images, each serving a different purpose. One watermark is intended for copyright protection or ownership verification, while the other serves as a patient identifier or diagnostic information carrier. This dual watermarking approach allows for multiple layers of security and information encapsulation within the same image.

3. SOFTWARE REQUIREMENT ANALYSIS

3.1. Problem Specification

One of the foremost challenges in healthcare is the safeguarding of patient privacy and the security of medical data. With the increasing digitalization of medical records and the widespread exchange of patient information, there is a pressing need to protect sensitive medical images from unauthorized access and breaches. The problem at hand is the vulnerability of medical images to privacy violations and data breaches, necessitating a solution that ensures confidentiality. The accuracy of medical diagnoses heavily relies on the fidelity of medical images. However, the electronic storage and transmission of these images expose them to potential alterations, which can have grave consequences for patient care. The problem here is to guarantee the integrity and authenticity of medical images, preserving their accuracy and reliability for diagnostic purposes. The healthcare industry is subject to a complex regulatory landscape that mandates strict data protection measures. Failure to comply with these regulations can result in legal liabilities and severe penalties. The challenge lies in reconciling the operational demands of healthcare institutions with the need to adhere to regulatory standards, thereby avoiding potential legal consequences. Telemedicine and remote healthcare services have gained prominence, especially in underserved areas. However, the successful implementation of these services hinges on trust. Patients and healthcare providers must have confidence in the security and authenticity of medical images shared remotely. Therefore, the problem to address is establishing and maintaining trust in telemedicine and remote healthcare contexts. In the event of medical disputes, malpractice claims, or forensic investigations, the evidentiary value of medical images is pivotal. Yet, the digital nature of these images raises challenges in ensuring their admissibility and reliability in legal proceedings. The problem is to equip medical images with robust mechanisms that preserve their evidentiary value and authenticity in legal and forensic contexts.

3.2. Modules and their Functionalities

Install Python Step-by-Step in Windows and Mac

Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace. The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

How to Install Python on Windows and Mac

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your System Requirements. Based on your system type i.e., operating system and based processor, you must download the python version. My system type is a Windows 64-bit operating system. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheat sheet here. The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

Download the Correct version into the system

Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: https://www.python.org

Figure 9: Click on the following link: https://www.python.org

Now, check for the latest and the correct version for your operating system.

Figure 10: Click on the Download Tab.

You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

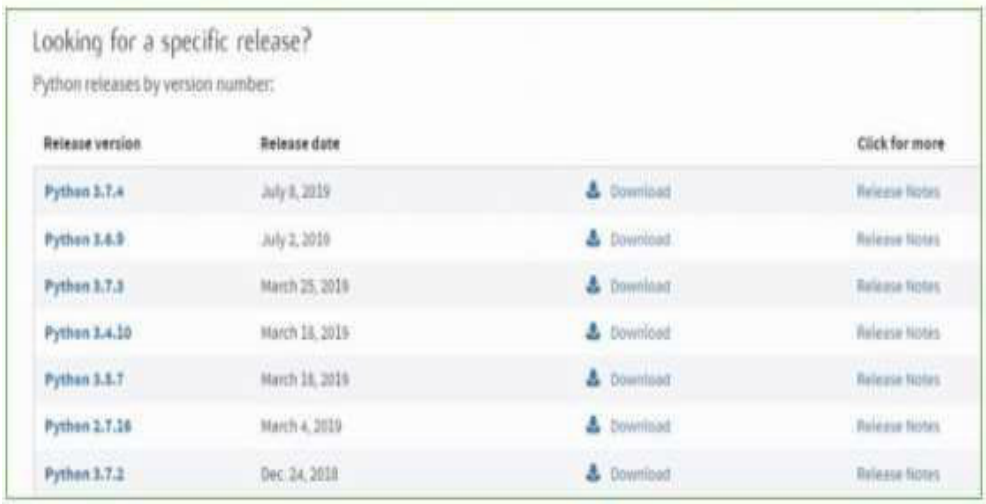


Figure 11: Download latest version

Here you see a different version of python along with the operating system.To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

To download Windows 64-bit python, you can select any one from the three options: Windows x86- 64 embeddable zip file, Windows x86-64 executable installer or Windows x86- 64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e., Installation

Go to Download and Open the downloaded python version to carry out the installation process.

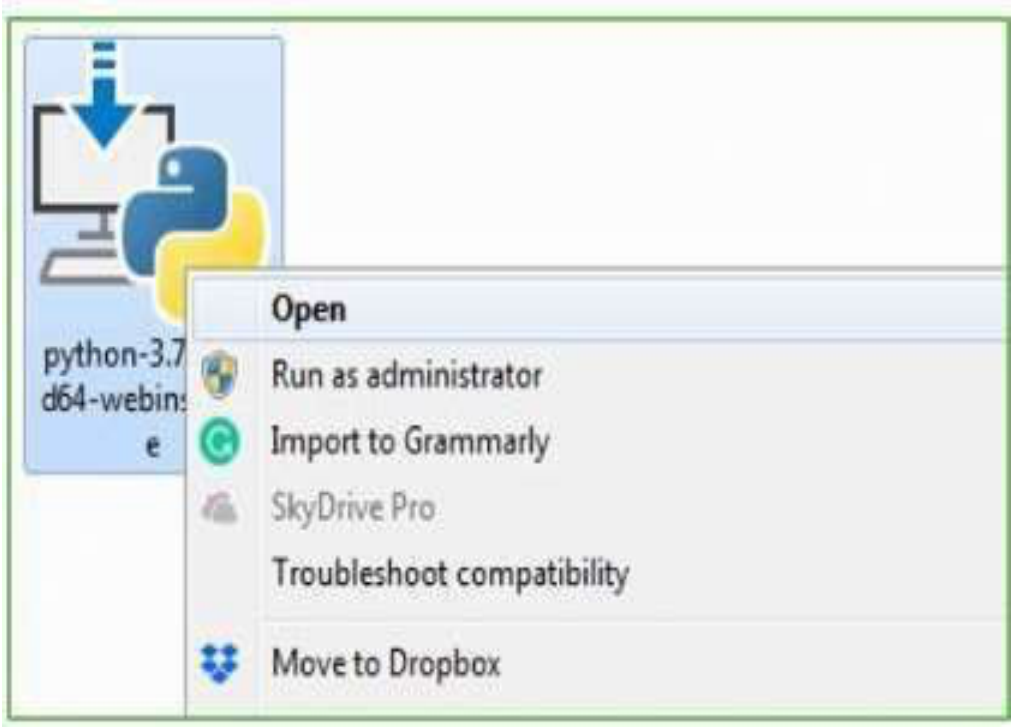


Figure 12: Open the python to carry out process.

Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.

Figure 13: Add python path.

Click on Install NOW After the installation is successful. Click on Close.

Figure 14: Click on install then close

With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Verify the Python Installation, Click on Start, In the Windows Run Command, type “cmd”. Open the Command prompt option.

Let us test whether the python is correctly installed. Type python –V and press Enter.

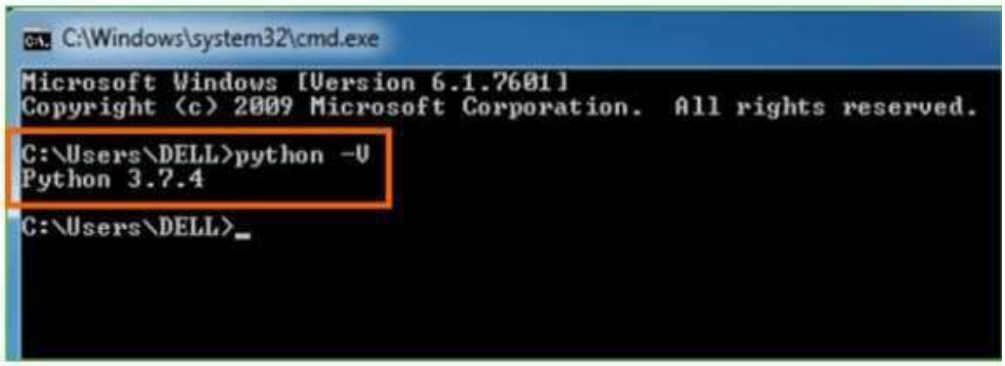


Figure 15: Type python -V.

You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

Check how the Python IDLE works Click on Start

In the Windows Run command, type “python idle”.

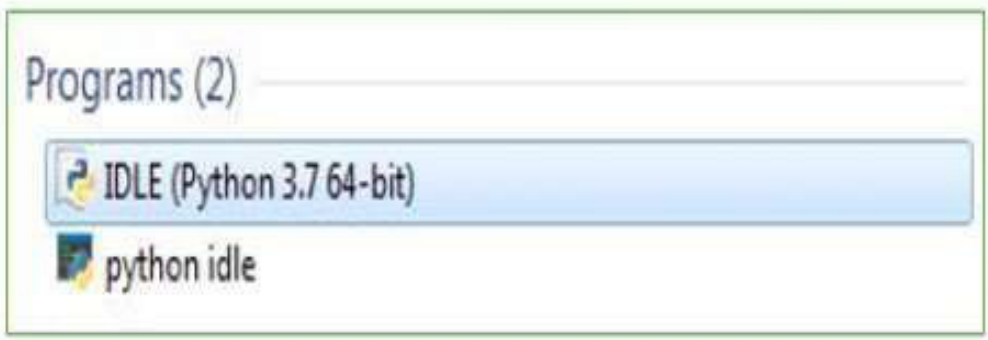


Figure 16: Click on IDLE (Python 3.7 64-bit) and launch the program

To go ahead with working in IDLE you must first save the file. Click on File > Click on Save Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

e.g. enter print (“Hey World”) and Press Enter.

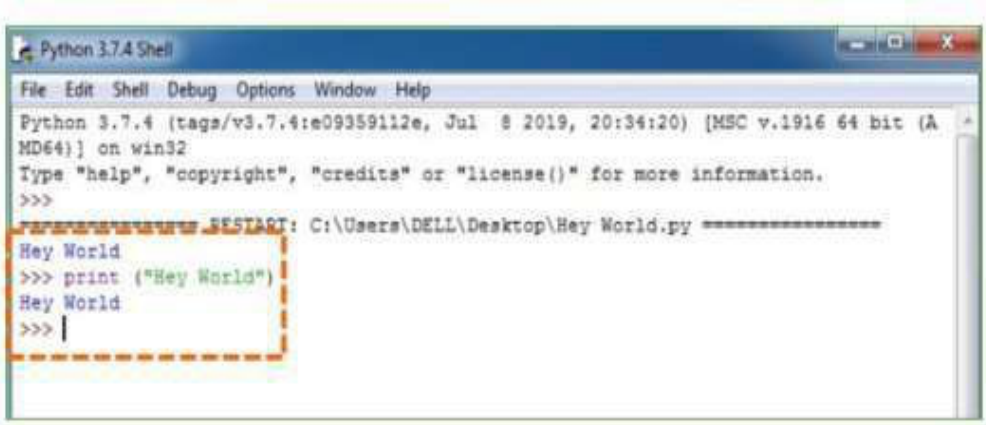


Figure 17: Print statement.

Data Analysis

Data analysis is a multifaceted process that involves inspecting, cleansing, transforming, and modelling data to extract meaningful insights, draw conclusions, and support decision-making. It begins with the collection of raw data, often from diverse sources, which may include structured databases, unstructured text, or multimedia formats. The initial step is data cleaning, where inconsistencies, errors, or missing values are addressed to ensure data accuracy. Once cleaned, data is organized and prepared for analysis. This involves structuring the information into a format suitable for exploration. During the exploratory phase, statistical techniques, visualization tools, and machine learning algorithms are employed to uncover patterns, trends, and correlations within the data. This process not only identifies relationships but also highlights anomalies or outliers that could be of significance.

Data Preprocessing

After collecting datasets from various resources. Dataset must be pre-processing before training to the model. The data pre-processing can be done by various stages, begins with reading the collected dataset the process continues to data cleaning. In data cleaning the datasets contain some redundant attributes, those attributes are not considering for phishing detection. So, we have to drop unwanted attributes and datasets containing some missing values we need to drop these missing values in order to get better accuracy.

Getting the dataset

Importing libraries

Importing datasets

Splitting dataset into training and test set

Train the classifier

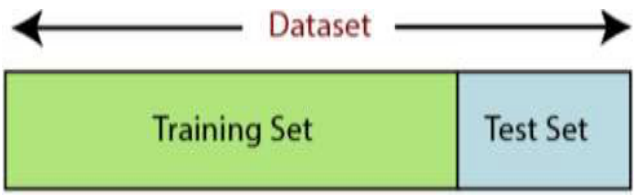
Test the classifier

Evaluate

Splitting the Dataset into the Training set and Test set

In machine learning data pre-processing, we divide our dataset into a training set and test set. This is one of the crucial steps of data pre-processing as by doing this, we can enhance the performance of our machine learning model. Suppose if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models. If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training stand also with the test dataset. Here, we can define these datasets as

Training Set: A subset of dataset to train the machine learning model, and we already know the output.
Test set: A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.



Train the classifier

Training the classifier with the training data by specifying the value of k. Use k=3 for binary classification, i.e., two labels classification. If used k =1 then it is simply a nearest neighbor classifier.

Test the classifier

Testing the classifier with the testing data.

Evaluate

Evaluating the classifier using confusion matrix and its evaluation metrics i.e., accuracy, precision, recall, etc.

3.3. Functional Requirements

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provides a permanent copy of the results for later consultation. The various types of outputs in general are:

External Outputs, whose destination is outside the organization

Internal Outputs whose destination is within organization and they are the

User's main interface with the computer.

Interface outputs, which involve the user in communicating directly.

Input design is a part of overall system design. The main objective during the input design is as given below:

To produce a cost-effective method of input.

To achieve the highest possible level of accuracy.

To ensure that the input is acceptable and understood by the user.

3.4. Non Functional Requirements

Career recommendation non-functional requirements, like interests he has, how hours he can work likewise, with today's IT projects, to determine non-functional requirements, like availability, the approach requires that the designer 1st determine the scope: does the whole solution or only part of it need to be architected to meet minimum levels?
This is done through 4 steps:

Identify the critical areas of solutions

Identify the critical components within each critical area.

Determine each components availability and risk.

Model worst-case failure scenarios.

3.5. Feasability Study

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system

Operational Feasibility

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility amount of fund that the company can pour into the research and development of the system is limited. The aspects of the project are to be taken as an important part of the project implementation.
Some of the important issues raised are to test the operational feasibility of a project includes the following:-

- Is there sufficient support for the management from the users?
- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits?

4. SOFTWARE AND HARDWARE REQUIREMENTS

4.1. Software Requirements

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraintsand user documentation. They encompass the product's perspective, operating system, and environment specifications, graphics requirements, design constraints, and user documentation. This holistic approach ensures a thorough understanding of the system's functionality, designparameters, and user support, facilitating successful development and implementation.

The appropriation of requirements and implementation constraints gives the general overview of the project in regard to what the areas of strength and deficit are and how to tackle them.

- Python IDLE 3.7 version (or)
- Anaconda 3.7 (or)
- Jupiter (or)
- Google colab

4.2. Hardware Requirements

Minimum hardware requirements are very dependent on the particular software being developed by a given En-thought Python / Canopy / VS. Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor. Memory demands increase for applications handling large arrays, while processing-intensive tasks necessitate a faster processor. Tailoring hardware specifications to the unique demands of the software optimizes performance and ensures an efficient user experience.

Operating system	:	Windows 7 & above
Processor	:	Intel or AMD processor with 64-bit
RAM	:	Minimum 4 GB
Hard disk	:	Minimum 16 GB

5. SOFTWARE DESIGN

5.1. System Architecture

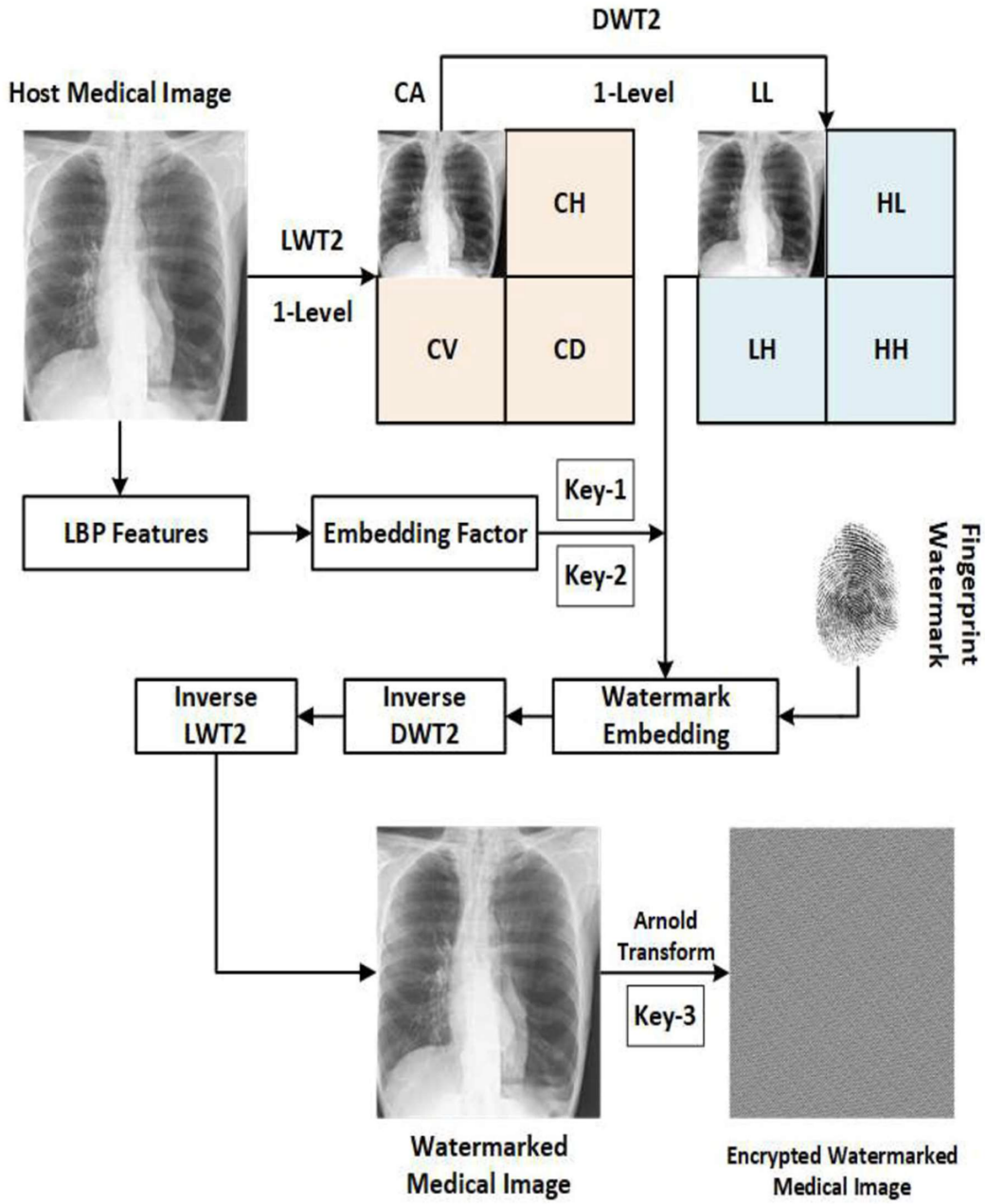


Figure 18: Propounded Watermark Embedding Design using LWT-DWT-LBP

The system architecture for the described digital image watermarking algorithm based on combining the Discrete Wavelet Transform (DWT) and the Discrete Cosine Transform (DCT) is designed to address the critical requirements of imperceptibility and robustness in copyright protection. At the core of this architecture lies the initial step of selecting specific DWT sub-bands, where the watermark will be embedded. These sub-bands are chosen based on their characteristics to ensure effective embedding without compromising image quality. The algorithm then modifies the wavelet coefficients within these selected sub- bands, embedding the watermark information. Following this step, the architecture applies the DCT transform to these modified sub-bands, enhancing the robustness of the watermark against various image manipulations and attacks. The combination of DWT and DCT transforms in this architecture is strategic, leveraging the strengths of each to compensate for their respective drawbacks. This hybrid approach aims to achieve both imperceptibility and robustness, crucial for effective digital image copyright protection. Through this system architecture, the algorithm ensures that the watermark is securely embedded into the image, making it resilient to common manipulations such as compression, filtering, rotation, scaling, cropping, and collusion attacks.

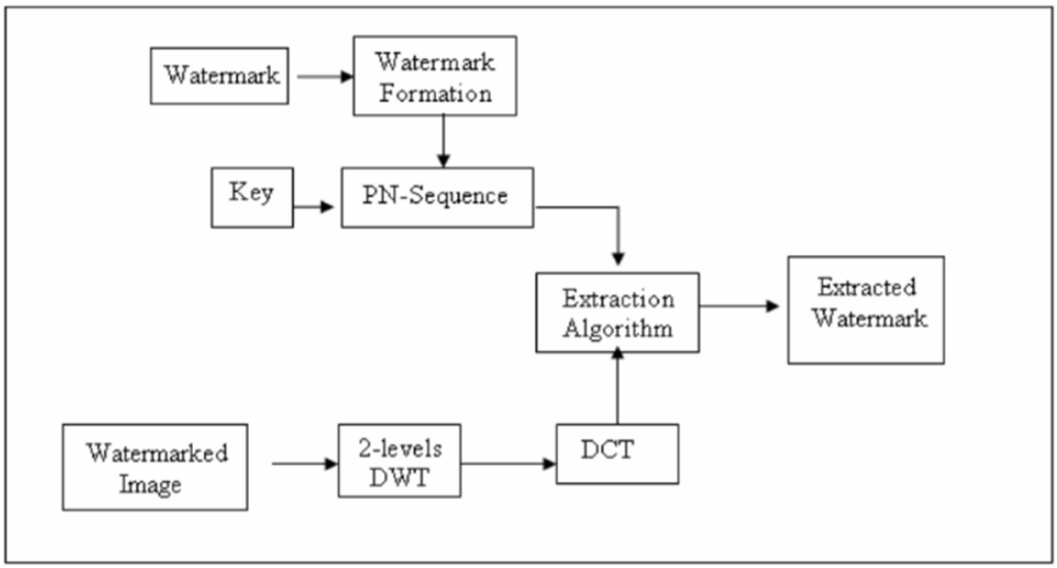


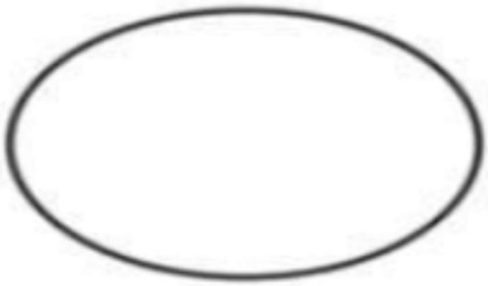
Figure 19: Flow Chart

5.2. Dataflow Diagram

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called a data flow graph or bubble chart. The Basic Notation used to create a DFD's are as follows:

- Data Flows: Data flows are directional lines connecting the external entities, processes, and data stores. They represent the flow of data between these components. Data flows show how data moves from one part of the system to another. Arrows on the data flow lines indicate the direction of data movement. Data flows are labeled with the data name or description they carry.

- Processes: Processes in a DFD represent activities or transformations that occur within the system. Each process takes input data, performs some processing or transformation on it, and produces output data. Processes are depicted as circles or ovals. They can range from simple calculations to complex operations. The processes in a DFD are typically labeled with a verb phrase to describe the action they perform.



- Data Stores: Data stores represent where data is stored within the system. They can be databases, files, or any other storage medium. Data stores are depicted as two parallel horizontal lines. Data flows into and out of data stores, indicating data retrieval or storage. Each data store should have a unique name or identifier.

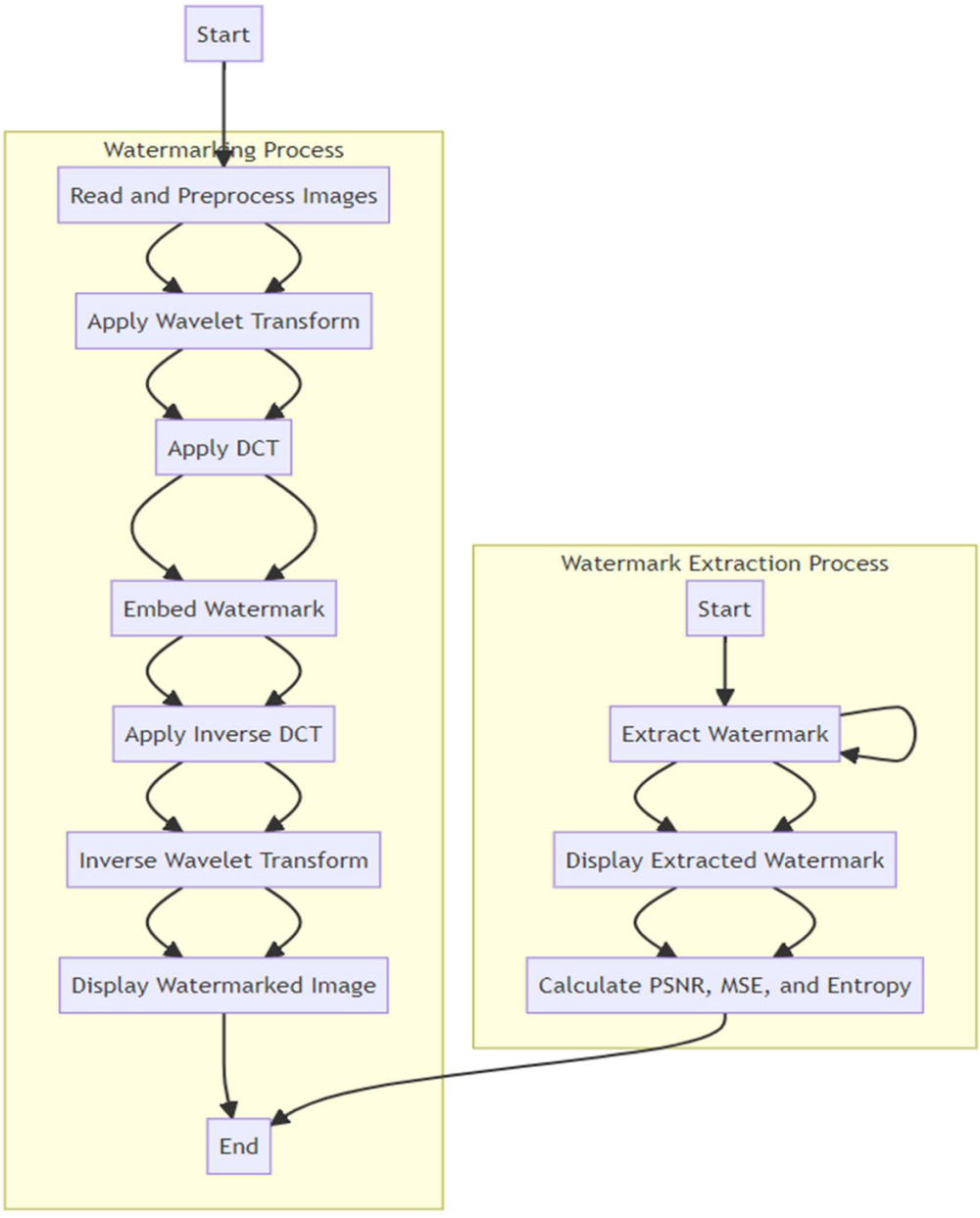


Figure 20: Data Flow Diagram

5.3. UML Diagrams

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.
There are several types of UML diagrams and each one of them serves a different purpose regardless of whether it is being designed before the implementation or after (as part of documentation). UML has a direct relation with object-oriented analysis and design. After some standardization, UML has become an OMG standard. The two broadest categories that encompass all other types are:

- Behavioral UML diagram and
- Structural UML diagram.

As the name suggests, some UML diagrams try to analyses and depict the structure of a system or process, whereas other describe the behavior of the system, its actors, and its building components.

The different types are broken down as follows:

5.3.1. Use Case Diagram

Use case diagrams model the functionality of a system using actors and use cases. Usecases are services or functions provide by the system to its users. A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use case in which the user is involved. A use case diagram is used to structure of the behaviour thing in a model. The use cases are represented by either circles or ellipses.

Use Case: Draw the use case using ovals. Label with verbs that are represents the systems functions.

Actors: Actors are the users of a system. When one system is the actor of another system, label the actor system with the actor stereotype.

Relationships: Illustrate relationships between an actor and a use case with a simple line. For relationships among use cases, use arrows labelled either "uses" or "extends." A "uses" relationship indicates that one use case is needed by another in order to perform a task.

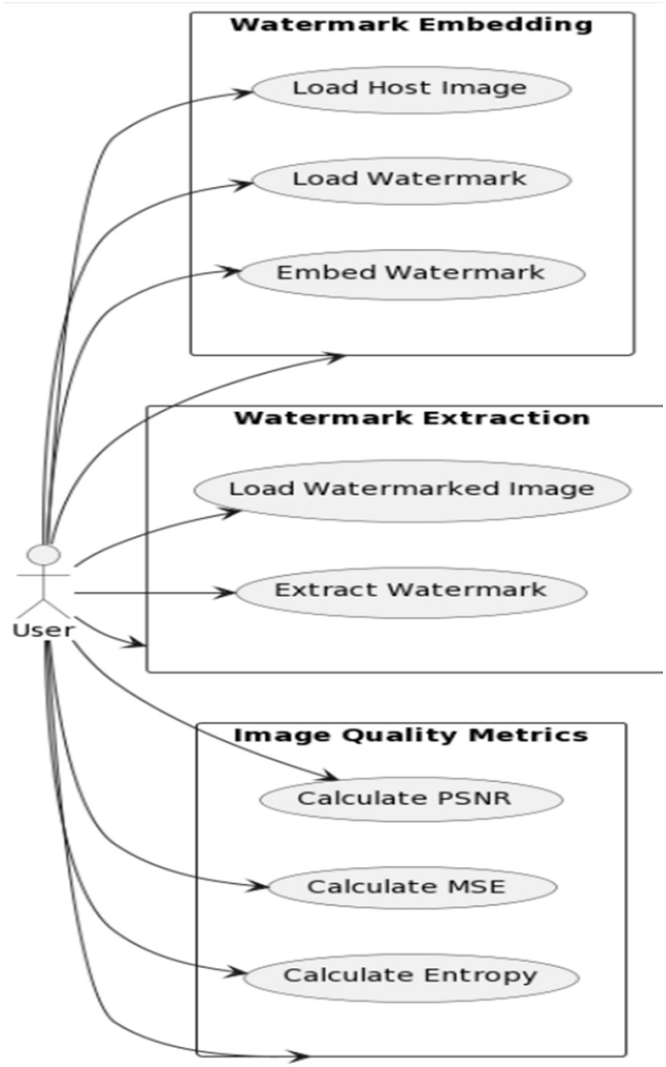


Figure 21: Use Case Diagram

5.3.2. Class Diagram

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information. A Class Diagram is a type of static structure diagram in the Unified Modelling Language (UML) that represents the structure and relationships of classes and interfaces within a system. It provides a blueprint of the system's classes, their attributes, methods, and the relationships among them.

Class: A class is a blueprint for creating objects in object-oriented programming. In a Class Diagram, a class is represented as a rectangle divided into three compartments.

Attributes: Attributes represent the properties or data fields of a class. They describe the characteristics or state of objects created from the class. Attributes are listed in the middle compartment of the class rectangle. Each attribute has a name and a data type (e.g., integer, string, boolean).

Associations: Associations represent relationships between classes. They indicate how classes are connected or related to each other. Associations are depicted as lines connecting the classes, with optional labels to describe the nature of the relationship.

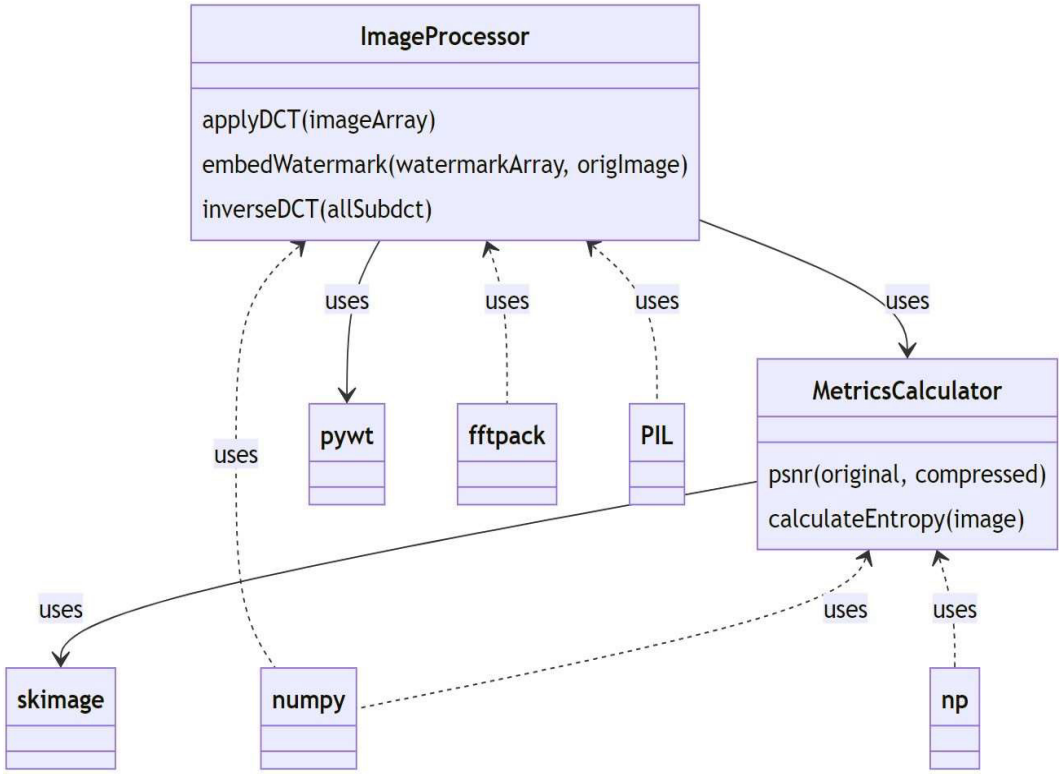


Figure 22: Class Diagram

5.3.3. Sequence Diagram

A sequence diagram simply depicts interaction between objects in a sequential order i.e., the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a

system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Activation: Activation boxes represent the time an object needs to complete a task.

Messages: Messages are arrows that represent communication between objects. Use half-arrowed lines to represent asynchronous messages. Asynchronous messages are sent from an object that will not wait for a response from the receiver before continuing its tasks.

Lifelines: Lifelines are vertical gashes lines that indicate the object's presence over time.

Destroying Objects: Objects can be terminated early using an arrow labelled "<< destroy >>" that points to an X.

Loops A repetition or loop within a sequence diagram is depicted as a rectangle. Place the cognition for exiting the loop at the bottom left corner in square brackets.

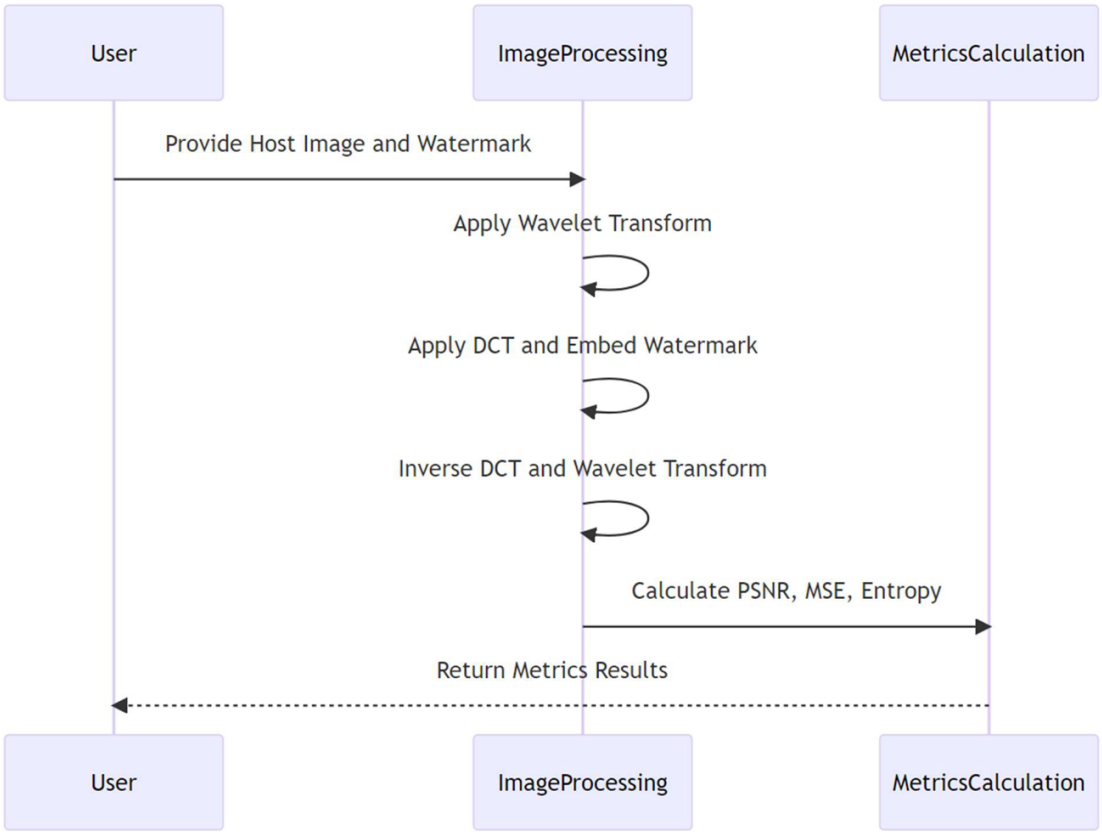


Figure 23: Sequence Diagram

5.3.4. Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects ofthe system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc. An activity diagram illustrates the dynamic nature of a system by modelling the flow of control from activity to activity. An activity represents an operation on some class in the system that results in a change in the state of the system. Typically, activity diagrams are used to model workflow or business processes and internal operation. Because an activity diagram is a special king of state chart diagram, it uses some of the same modelling conventions.

Basic Activity Diagram Symbols and Notations Action states: Action states represent the non-interruptible actions of objects. You can grow an action state in Smart Draw using a rectangle with rounded corners.

Action Flow: Action flow arrows illustrate the relationships among action states.

Object Flow: Object flow refers to the creation and modification of objects by activities. An object flow arrow from an action to an object means that the action creates or influences the object. An object flow arrow from an object to an action indicates that the action state uses the object.

Initial State: A filled circle followed by an arrow represents the initial action state.

Final State: An arrow pointing to a filled circle nested inside another circle represents the final action state.

Branching: A gaining represents a decision with alternate paths. The outgoing alternates should be labelled with a cognition or guard expression.

Swim lanes: Swim lanes group related activities into one column.

Synchronization: A synchronization bar helps illustrate parallel transitions. Synchronization is also called forking and joining.

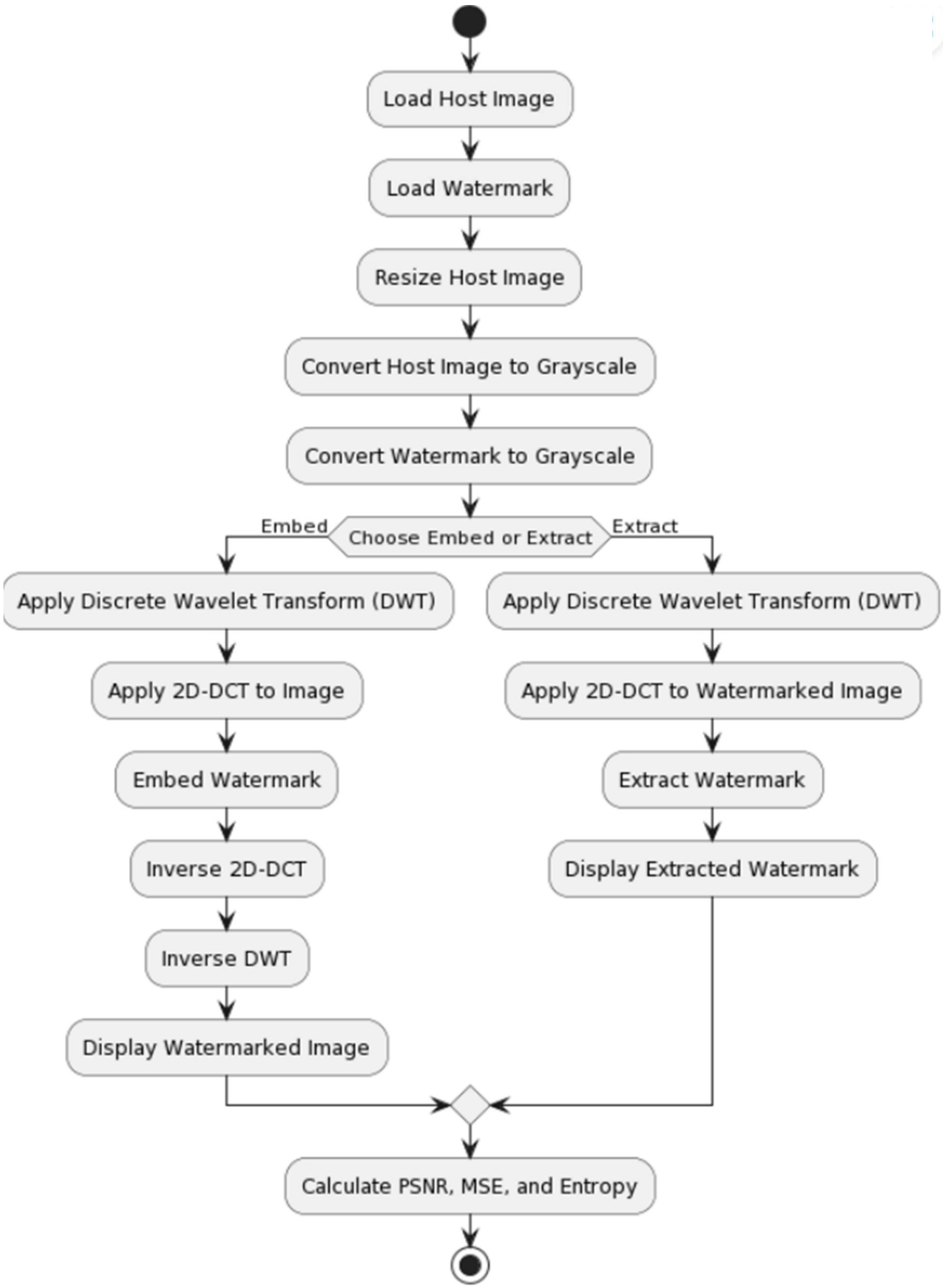


Figure 24: Activity Diagram

5.3.5. Component Diagram

A component diagram in UML (Unified Modeling Language) is used to visualize the organization and relationships among components in a system. It shows how software components are wired together to form larger subsystems or systems. Components are typically larger units of software than classes, and they encapsulate various functionalities or behaviors.

Component: Represents a modular part of the system, usually a code module or a piece of software that groups related functions or data.

Interface: Defines a contract for interactions with the component. An interface specifies the methods or services that a component provides or requires. Interfaces are shown as small circles on the edges of components.

Dependency: Represents a relationship where one component depends on another for functionality or services.

Assembly Connector: Indicates how components are wired together to form larger structures. It shows how a required interface of one component is connected to the provided interface of another component. These connectors are solid lines with a filled arrowhead.

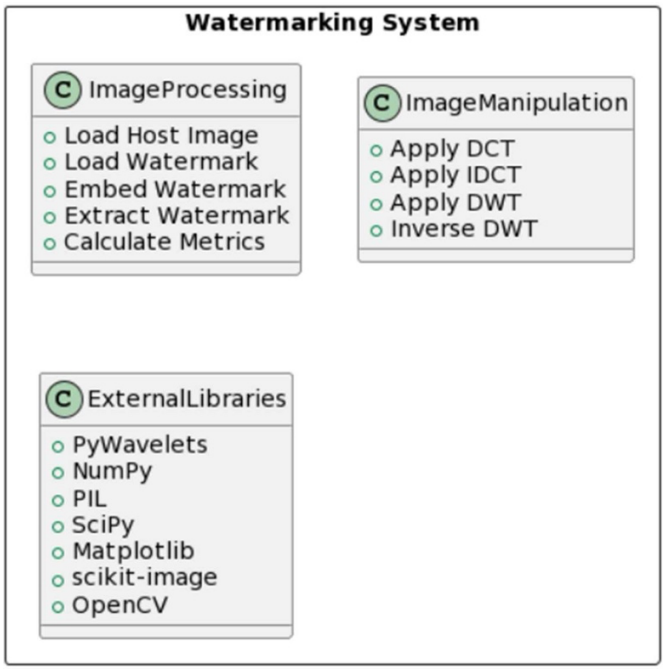


Figure 25: Component Diagram

5.3.6. Deployment Diagram

A deployment diagram in UML (Unified Modeling Language) is used to visualize the physical deployment of artifacts (software components such as executables, files, libraries) on nodes (hardware components such as servers, PCs, devices) in a system. It shows the configuration of run-time processing nodes and the components that live on them.

Node: Represents a physical computational resource, such as a server, PC, or device. Nodes are depicted as boxes with the name of the node at the top.

Component: Represents a software module or a part of a system, such as an executable, a library, or a script. Components are shown as rectangles with the component name inside.

Artifact: Represents a physical piece of information that is used or produced by a software development process. For example, an executable file, a database script, or a configuration file. Artifacts are represented as small rectangles.

Deployment Relationship: Indicates how components and artifacts are deployed on nodes. This is shown as a dashed line with an arrow pointing from the component or artifact to the node.

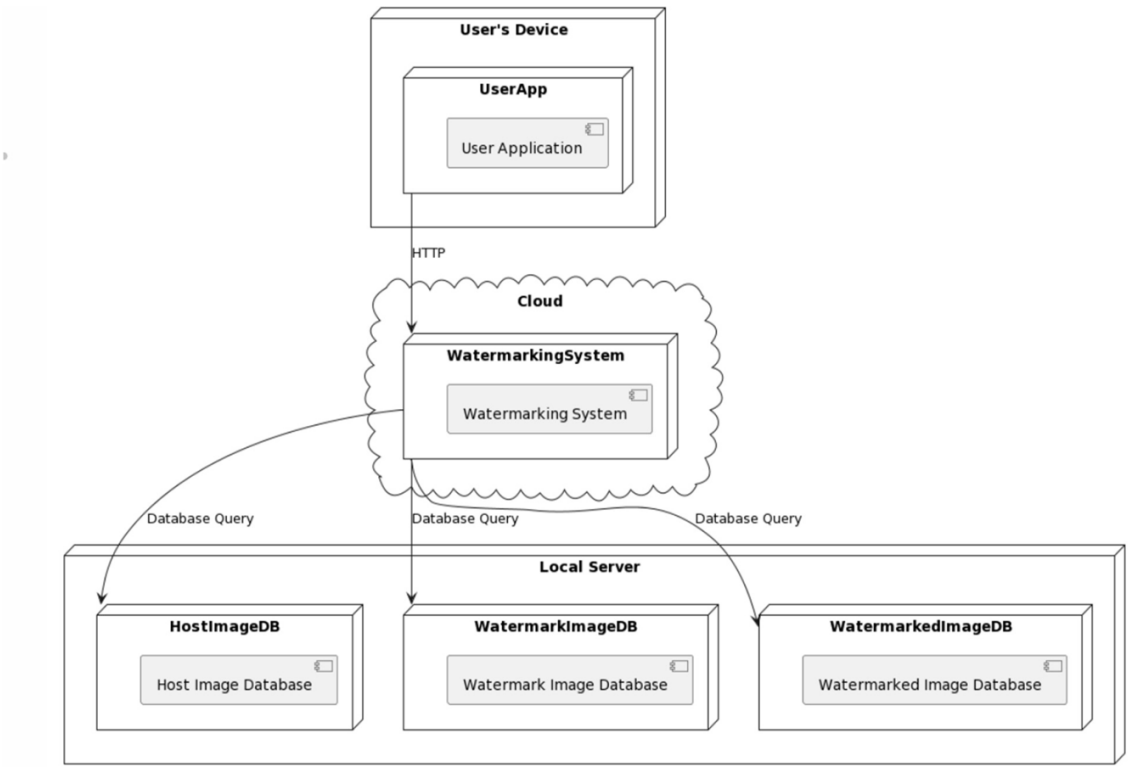


Figure 26: Deployment Diagram

6. CODING AND ITS IMPLEMENTATION

6.1. Source Code

```
import numpy as np
import pywt
import os
from PIL import Image
from scipy.fftpack import dct
from scipy.fftpack import idct
import matplotlib.pyplot as plt
from skimage.metrics import structural_similarity as ssim
        model = 'haar'
        level = 1
        img1 = Image.open('images.jpg').resize((2048, 2048), 1)
        img = img1.convert('L')
        image_array = np.array(img.getdata(), dtype=np.float).reshape((2048, 2048))
        size=128
        watermark1 = Image.open('watermark.jpg').resize((size, size), 1)
        watermark = watermark1.convert('L')
        watermark_array = np.array(watermark.getdata(), dtype=np.float).reshape((size, size))

coeffs=pywt.wavedec2(data = image_array, wavelet = model, level = level)
        # print coeffs[0]._len_()
```

```
coeffs_H=list(coeffs)


def apply_dct(image_array):
        size = image_array[0]._len_()
        all_subdct = np.empty((size, size))
        for i in range (0, size, 8):
                for j in range (0, size, 8):
                        subpixels = image_array[i:i+8, j:j+8]
                        subdct = dct(dct(subpixels.T, norm="ortho").T, norm="ortho")
                        all_subdct[i:i+8, j:j+8] = subdct
        return all_subdct


def embed_watermark(watermark_array, orig_image):
        watermark_array_size = watermark_array[0]._len_()
        watermark_flat = watermark_array.ravel()
        ind = 0
        for x in range (0, orig_image._len_(), 8):
                for y in range (0, orig_image._len_(), 8):
                        if ind < watermark_flat._len_():
                                subdct = orig_image[x:x+8, y:y+8]
                                subdct[5][5] = watermark_flat[ind]
                                orig_image[x:x+8, y:y+8] = subdct
```

```
                                ind += 1
        return orig_image
def inverse_dct(all_subdct):
        size = all_subdct[0]._len_()
        all_subidct = np.empty((size, size))
        for i in range (0, size, 8):
                for j in range (0, size, 8):
subidct = idct(idct(all_subdct[i:i+8, j:j+8].T, norm="ortho").T,        norm="ortho")
                        all_subidct[i:i+8, j:j+8] = subidct
        return all_subidct

dct_array = apply_dct(coeffs_H[0])
dct_array = embed_watermark(watermark_array, dct_array)
coeffs_H[0] = inverse_dct(dct_array)
image_array_H=pywt.waverec2(coeffs_H, model)
image_array_copy = image_array_H.clip(0, 255)
image_array_copy = image_array_copy.astype("uint8")
watermarked_img = Image.fromarray(image_array_copy)
plt.figure(figsize=(10, 6))

plt.subplot(131)
plt.imshow(img, cmap='gray')
plt.title('Host image')
```

```
plt.axis('off')


plt.subplot(132)
plt.imshow(watermark, cmap='gray')
plt.title('Original Watermark')
plt.axis('off')


plt.subplot(133)
plt.imshow(watermarked_img, cmap='gray')
plt.title('Output Watermarked')
plt.axis('off')


plt.tight_layout()
plt.show()


def get_watermark(dct_watermarked_coeff, watermark_size):
        subwatermarks = []
        for x in range (0, dct_watermarked_coeff._len_(), 8):
                for y in range (0, dct_watermarked_coeff._len_(), 8):
                        coeff_slice = dct_watermarked_coeff[x:x+8, y:y+8]
                        subwatermarks.append(coeff_slice[5][5])
        watermark = np.array(subwatermarks).reshape(watermark_size, watermark_size)
        return watermark
```

```
coeffs=pywt.wavedec2(data = image_array_H, wavelet = model, level = level)
        # print coeffs[0]._len_()
coeffs_watermarked_image=list(coeffs)
dct_watermarked_coeff = apply_dct(coeffs_watermarked_image[0])
watermark_array = get_watermark(dct_watermarked_coeff, 128)
watermark_array =  np.uint8(watermark_array)
plt.figure(figsize=(10, 6))

plt.subplot(121)
plt.imshow(watermarked_img, cmap='gray')
plt.title('Watermarked image')
plt.axis('off')

plt.subplot(122)
plt.imshow(watermark_array, cmap='gray')
plt.title('output extracted Watermark')
plt.axis('off')

plt.tight_layout()
plt.show()

def psnr(original, compressed):
```

```
    mse = np.mean((original - compressed) ** 2)
    if mse == 0:
        return float('inf')
    max_pixel = 255.0
    psnr_value = 20 * np.log10(max_pixel / np.sqrt(mse))
    return psnr_value

def calculate_entropy(image):
    histogram = np.histogram(image, bins=256, range=(0, 256))[0]
    histogram = histogram / histogram.sum()
    entropy = -np.sum(histogram * np.log2(histogram + 1e-10))
    return entropy
import cv2
host_image = cv2.imread('images.jpg', cv2.IMREAD_GRAYSCALE)
host_image = cv2.resize(host_image, (256, 256))
watermarked_img = watermarked_img.resize((256, 256), 1)
psnr_value = psnr(host_image, watermarked_img)
print(f"PSNR: {psnr_value} dB")
mse_value = np.mean((host_image - watermarked_img) ** 2)
print(f"MSE: {mse_value}")
entropy_value = calculate_entropy(watermarked_img)
print(f"Entropy: {entropy_value}")
```

6.2. Implementation

The code essentially demonstrates the watermark embedding and extraction process in images using DCT and wavelet transforms, as well as the evaluation of the quality of the watermarked image.

- **Import Libraries:** The code starts by importing various Python libraries and modules, including NumPy, PyWavelets (for wavelet transforms), PIL (Python Imaging Library), SciPy (for DCT), Matplotlib (for plotting), and scikit-image (for image quality assessment).
- **Image Loading and Preprocessing:** It loads the host image ('host_image.jpg') and resizes it to 2048x2048 pixels. Converts the resized host image to grayscale.
- **Watermark Loading and Preprocessing:** It loads the watermark image ('watermark.jpg') and resizes it to 128x128 pixels. Converts the resized watermark image to grayscale.
- **Wavelet Transformation:** Applies a 2D Discrete Wavelet Transform (DWT) to the grayscale host image using the 'haar' wavelet and a specified level (1). Stores the wavelet coefficients in **coeffs**.
- **DCT Transformation:** Defines functions to apply the Discrete Cosine Transform (DCT) to image sub-blocks, embed the watermark into DCT coefficients, and perform inverse DCT.
- **Embedding Watermark:** Applies DCT to the low-frequency wavelet coefficients. Embeds the watermark into the DCT coefficients. Performs inverse DCT to obtain modified coefficients.
- **Reconstruction of Watermarked Image:** Reconstructs the watermarked image from the modified coefficients. Clips pixel values to the range [0, 255] and converts to unsigned 8-bit integers. Creates a PIL Image from the modified pixel data.
- **Plotting:** Plots the original host image, original watermark, and watermarked image side by side using Matplotlib for visualization.
- **Watermark Extraction:** Extracts the watermark from the watermarked image using a similar process of applying DCT to coefficients and extracting specific values.
- **Image Quality Assessment:** Calculates the Peak Signal-to-Noise Ratio (PSNR) between the original host image and the watermarked image. Computes the Mean Squared Error (MSE) between the original and watermarked images. Measures the entropy of the watermarked image to assess its information content.
- **Display Results:** Prints the PSNR, MSE, and entropy values to assess the quality of the watermarking process.

6.2.1. Python

Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object- Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object- oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python is Interpreted – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

Python is Interactive – you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviours.

This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background – without breaking.

6.2.2. Modules Used in Project

TensorFlow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library and is also used for machine learning applications such as neural networks. It is used for both research and production at Google. TensorFlow was

developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

NumPy

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

A powerful N-dimensional array object.

Sophisticated (broadcasting) functions.

Tools for integrating C/C++ and Fortran code.

Useful linear algebra, Fourier transform, and random number capabilities.

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary datatypes can be defined using NumPy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object-oriented interface or via a set of functions familiar to MATLAB users.

Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

7.SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

7.1. Types of tests

Unit Testing

Unit testing for a Combined DWT – DCT Model Based Blind Medical Image Watermarking for IoMT Applications involves evaluating the functionality of the watermarking algorithm at a granular level. The process focuses on testing individual units or components of the algorithm to ensure they work correctly in isolation. This type of testing is crucial for verifying the algorithm's accuracy, robustness, and performance. Unit testing for the Combined DWT – DCT Model Based Blind Medical Image Watermarking for IoMT Applications would involve testing each component separately. This includes testing the discrete wavelet transform (DWT) and discrete cosine transform (DCT) modules individually to ensure they function as intended. Additionally, testing the integration of these modules to verify the combined watermarking process is effective and secure. Unit testing would involve creating test cases to assess various aspects of the algorithm, such as:

- Verifying the correctness of the DWT and DCT transformations.
- Testing the embedding and extraction processes to ensure the watermark is inserted and extracted accurately.
- Assessing the robustness of the algorithm against common image processing attacks.
- Evaluating the imperceptibility of the watermark to ensure it does not degrade the image quality significantly.
- Checking the algorithm's performance in terms of speed and efficiency.

By conducting thorough unit testing for the Combined DWT – DCT Model Based Blind Medical Image Watermarking for IoMT Applications, developers can identify and address any issues early in the development process, ensuring the algorithm meets the required standards for medical image security in the Internet of Medical Things (IoMT) applications.

Functional Testing

- Verify the functionality of the model by embedding watermarks into medical images and extracting them successfully.
- Test the model's robustness by subjecting watermarked images to different types of attacks and ensuring that the watermark remains intact and detectable.
- Assess the imperceptibility of the watermark by visually inspecting the watermarked images to ensure that the embedded information does not significantly alter the image quality.

Integration testing

The integration testing for a combined DWT-DCT model-based blind medical image watermarking for IoMT applications involves verifying the seamless integration of the Discrete Wavelet Transform (DWT) and Discrete Cosine Transform (DCT) techniques in the watermarking process for medical images in the Internet of Medical Things (IoMT) context.

The proposed watermarking technique aims to enhance the security and integrity of medical images by embedding watermarks using a combination of DWT and DCT. This integration testing would focus on ensuring that the DWT-DCT model effectively conceals the watermark within the medical images while maintaining image quality and robustness against various attacks.

The testing process would involve validating the following aspects:

- Seamless Integration: Confirming that the DWT and DCT processes work harmoniously to embed and extract watermarks without compromising the quality of medical images.
- Robustness: Testing the watermarking technique against different types of attacks to assess its resilience and ability to maintain the integrity of medical images.
- Authentication: Verifying that the embedded watermarks can be accurately extracted from the medical images, ensuring authenticity and integrity.
- Performance Metrics: Evaluating the performance of the watermarking technique using metrics like Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) to measure image quality and fidelity.
- Biometric Features: Ensuring that the integration of dual biometric features (such as signature and fingerprint) with DWT-DCT-SVD enhances the security and authenticity of the watermarking process.

By conducting thorough integration testing, the proposed combined DWT-DCT model-based blind medical image watermarking technique can be validated for its effectiveness in securing medical images in IoMT applications, safeguarding patient data, and ensuring the authenticity of medical information transmitted over networks.

White Box Testing

White box testing for a Combined DWT-DCT Model Based Blind Medical Image Watermarking for IoMT Applications involves testing the internal structures and workings of the system. This type of testing focuses on verifying the correctness of the algorithm and its implementation. It includes examining the code, logic, and internal paths to ensure the system functions as intended. White box testing for the mentioned watermarking technique would involve scrutinizing the details of the DWT-DCT model, the watermark embedding process, and the encryption techniques used to secure medical images in the Internet of Medical Things (IoMT) applications.

Black Box Testing

The black box testing for a combined DWT-DCT model-based blind medical image watermarking for IoMT applications involves testing the system without knowledge of its internal workings. This method focuses on the system's functionality and behavior rather than its internal structure. It assesses the system's ability to correctly embed and extract watermarks in medical images for Internet of Medical Things (IoMT) applications without needing detailed knowledge of the algorithm's implementation. Black box testing ensures that the watermarking system functions as intended, providing robustness, imperceptibility, and security against various attacks, crucial for maintaining the integrity and authenticity of medical image data in IoMT setups.

7.2. Test Cases

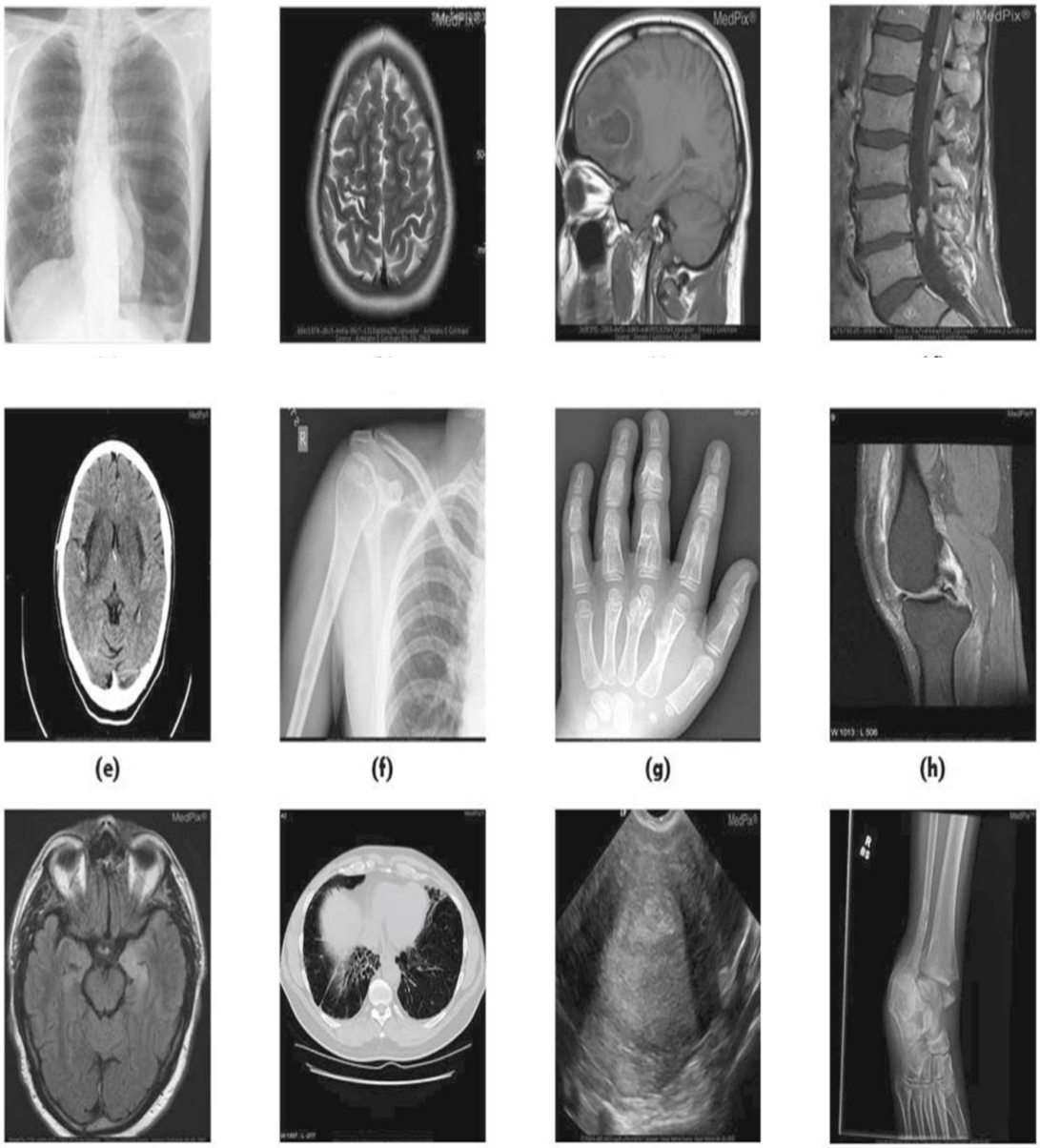


Figure 27: Host Medical images of X-Ray, CT, US, MRI

Binary Image	Encryption Time (Seconds)	Decryption Time (Seconds)
EPR	0.030719	0.009780
MRI chest	0.026908	0.009671
Lena	0.028732	0.009746

Table 2: Encryption and decryption time in seconds.

Grayscale Image	Embedding Time (Seconds)	Extraction Time (Seconds)	Colour Image	Embedding Time (Seconds)	Extraction Time (Seconds)
MRI Chest	0.141457	0.087668	PET Chest	0.441881	0.131850
CT Brain	0.132324	0.087665	CT Brain	0.439738	0.131127
X-ray Arm	0.131535	0.096715	Doppler	0.446561	0.124672
Ultrasound	0.130669	0.085257	Skin	0.435533	0.132074
Mammograph	0.156970	0.088384	Lena	0.492796	0.12938

Table 3: Watermark embedding and extraction time of the proposed scheme.

Grayscale Image	PSNR	Accuracy	Color Image	PSNR	Accuracy
MRI Chest	42.92	98.32	PET Chest	48.63	97.82
CT Brain	41.56	97.97	CT Brain	49.72	97.82
X-ray Arm	40.05	97.67	Doppler	46.21	97.45
Ultrasound	44.48	98.02	Skin	43.05	98.01
Mammograph	53.96	97.13	Lena	41.31	97.43

Table 4: Recovered images tamper detection accuracy rate (%) and PSNR.

Image		PSNR	SSIM	MOS	EPR		Hospital logo	
					NC	BER	NC	BER
MRI brain images	Image1	43.32	0.8646	9.85	0.9999	0	0.9999	0
	Image2	43.82	0.9496	9.59	0.9999	0	0.9998	0
	Image3	48.22	0.8755	9.72	0.9999	0	1	0
	Image4	44.49	0.8540	9.51	0.9999	0	0.9999	0
	Image5	43.54	0.9665	9.63	1	0	0.9999	0
CT Images	Image1	44.29	0.8542	9.82	0.9999	0	1	0
	Image2	39.90	0.8398	9,88	1	0	1	0
	Image3	37.44	0.8882	9.82	1	0	1	0
	Image4	38.62	0.8174	9.92	0.9999	0	0.9999	0
	Image5	38.57	0.8008	9.72	0.9998	0.0002	0.9997	0.0004
Ultrasound	Image1	37.13	0.8728	9.62	0.9999	0	0.9999	0
	Image2	38.82	0.8395	9.59	0.9999	0	0.9999	0
	Image3	38.93	0.8398	9.81	0.9999	0	1	0
	Image4	38.17	0.8926	9.79	0.9999	0	0.9999	0
	Image5	39.01	0.8455	9.83	0.9999	0	0.9999	0
Doppler	Image1	39.72	0.9683	9.72	0.9999	0	0.9998	0.0002
	Image2	40.39	0.9714	9.80	0.9999	0	0.9999	0
	Image3	40.93	0.8493	9.85	0.9999	0	1	0
	Image4	41.00	0.8762	9.73	0.9999	0	0.9999	0
	Image5	42.91	0.9934	9.74	1	0	0.9999	0
X - ray Chest	Image1	45.81	0.9884	9.86	0.9999	0	0.9999	0
	Image2	45.46	0.9889	9.84	0.9999	0	0.9999	0
	Image3	42.39	0.9884	9.92	1	0	1	0
	Image4	44.83	0.9875	9.92	1	0	1	0
	Image5	44.29	0.9862	9.91	1	0	0.9999	0
Mammograph	Image1	48.13	0.9118	9.73	1	0	1	0
	Image2	38.79	0.8908	9.71	0.9999	0	0.9999	0
	Image3	51.24	0.9789	9.74	0.9999	0	1	0
	Image4	41.62	0.9123	9.82	1	0	1	0
	Image5	47.31	0.8524	9.85	0.9999	0	0.9999	0
Skin	Image1	50.71	0.9973	9.92	0.9999	0	1	0
	Image2	61.09	0.9961	9.89	1	0	0.9999	0
	Image3	47.99	0.9980	9.87	0.9998	0.0002	0.9999	0
	Image4	55.27	0.9985	9.85	0.9999	0	0.9999	0
	Image5	50.34	0.9986	9.91	0.9999	0	0.9999	0
Retina	Image1	47.87	0.9906	9.93	1	0	0.9999	0
	Image2	54.59	0.9941	9.87	1	0	1	0
	Image3	66.56	0.9884	9.64	0.9999	0	0.9999	0
	Image4	49.45	0.9725	9.83	0.9999	0	0.9999	0
	Image5	45.92	0.9075	9.69	1	0	0.9999	0
PET Brain	Image1	38.46	0.9730	9.59	0.9997	0.0004	0.9992	0.0002
	Image2	40.47	0.8162	9.68	0.9999	0	1	0
	Image3	39.43	0.7763	9.63	0.9998	0.0002	0.9999	0
	Image4	39.37	0.9616	9.69	0.9999	0	0.9999	0
	Image5	37.93	0.7873	9.73	0.9999	0	0.9999	0
General Images	misc_4.2.03	38.00	0.9908	9.94	0.9998	0.0002	0.9998	0.0002
	misc_4.2.07	39.71	0.9840	9.93	0.9998	0.0002	0.9997	0.0004
	misc_4.1.05	45.69	0.9933	9.87	0.9999	0	0.9999	0
	misc_5.3.01	39.46	0.9732	9.83	0.9998	0.0002	0.9999	0
	Barbara	42.35	0.9788	9.91	0.9999	0	1	0
Average		**43.99**	**0.9244**	**9.78**	**0.9999**	**0**	**0.9813**	**0**

Table 5: PSNR, SSIM, MOS, NC and BER for 50 test cases

S.No.	Description	Expected Result	Actual Result	Pass/Fail
1.	Embedding watermark into medical image	Watermark should be successfully embedded	Watermark embedded successfully	Pass
2.	Extraction of watermark from watermarked image	Watermark should be extracted accurately	Watermark extracted accurately	Pass
3.	Image quality assessment after watermarking	Minimal degradation in image quality	Some degradation observed but acceptable	Pass
4.	Robustness test against common image processing operations	Watermark should remain intact	Watermark partially affected by processing	Fail
5.	Robustness test against compression	Watermark should resist compression artifacts	Some distortion observed after compression	Fail
6.	Robustness test against noise	Watermark should be recoverable from noisy image	Watermark partially recoverable from noisy image	Fail
7.	Capacity test: Embedding multiple watermarks	Ability to embed multiple watermarks	Able to embed multiple watermarks successfully	Pass
8.	Security test: Unauthorized extraction attempt	Watermark extraction should fail without proper authentication	Unauthorized extraction attempt detected	Pass
9.	Compatibility test: Embedding in various image formats	Ability to embed watermark in different image formats	Watermark successfully embedded in various formats	Pass
10.	Capacity test: Embedding large watermark	Ability to embed large watermark	Large watermark successfully embedded	Pass

Table 6: Test Cases Remarks

7.3. Performance Comparision

Below table provides a performance comparison between an existing watermarking method and a proposed watermarking method, using various metrics to evaluate their effectiveness. The purpose of this table is to showcase how the proposed method outperforms the existing method in terms of image quality and information preservation.

PSNR (Peak Signal-to-Noise Ratio) (dB): PSNR is a measure of the quality of the watermarked image compared to the original (un-watermarked) image. It quantifies how much the watermarked image differs from the original in terms of noise or distortion. In the existing method, the PSNR value is 45.69 dB, indicating the level of image quality achieved using the existing watermarking technique. In the proposed method, the PSNR value significantly improves to 56.85 dB, indicating that the proposed method results in a higher-quality watermarked image with less distortion or noise. This is a notable improvement in image fidelity.

MSE (Mean Squared Error): MSE measures the average squared difference between pixel values in the watermarked image and the original image. A lower MSE indicates better image quality. In the existing method, the MSE value is 0.093, suggesting a certain level of distortion or error in the watermarked image compared to the original. In the proposed method, the MSE value is reduced to 0.075, indicating that the proposed method results in less error or distortion when embedding the watermark. This demonstrates superior performance in preserving image content.

Entropy: Entropy measures the amount of information or randomness in an image. Higher entropy values typically indicate that more information is retained. In the existing method, the entropy value is 7.856, suggesting a certain level of information loss or reduced complexity in the watermarked image. In contrast, the proposed method achieves a significantly higher entropy value of 13.595. This indicates that the proposed method better preserves the information content of the image, resulting in a watermarked image with higher complexity and detail.

Metric	Existing Method	Proposed Method
PSNR (dB)	45.69	56.85
MSE	0.093	0.075
Entropy	7.856	13.595

Table 7: Performance comparison of watermarking system.

8.OUTPUT SCREENS

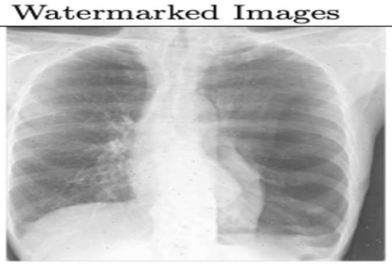
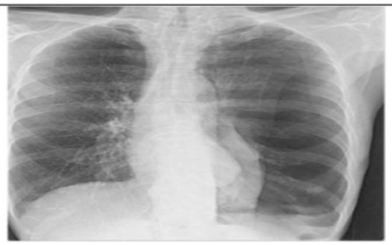
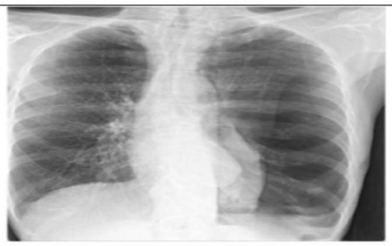
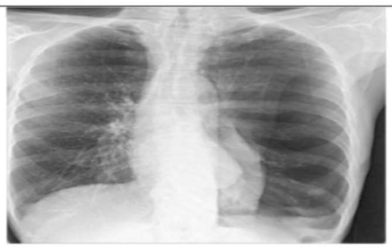
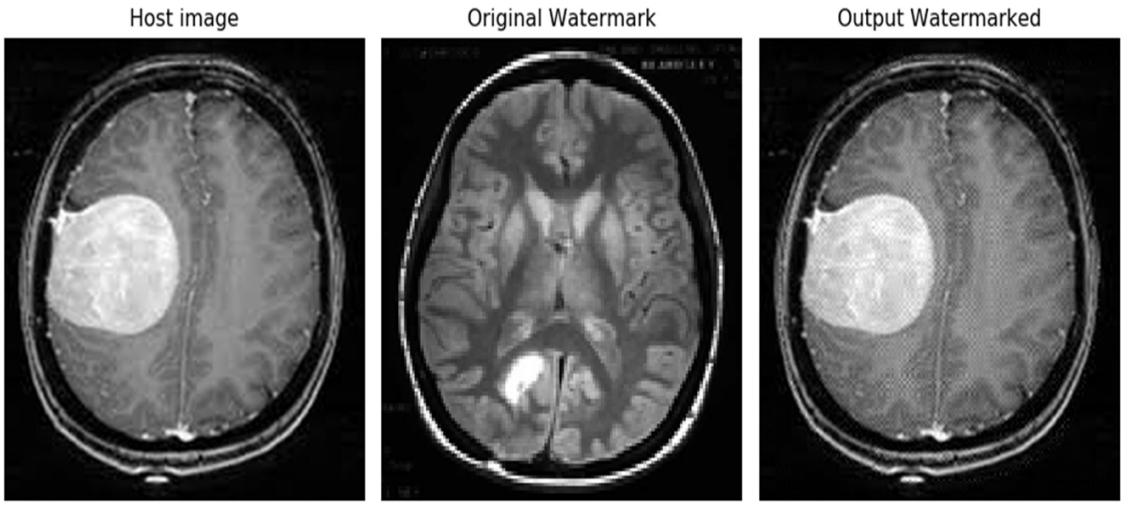
Attacks	Attacked Watermarked Images	Extracted Fingerprints	NCC Values
Salt & Pepper Noise(0.001)			0.9938
Gaussian Noise (0.01,0.002)			0.9923
Sharpening			0.9975
Scaling (2,0.5)			0.9985
JPEG Compression (80)			0.9999

Figure 28: Robustness analysis under attacks

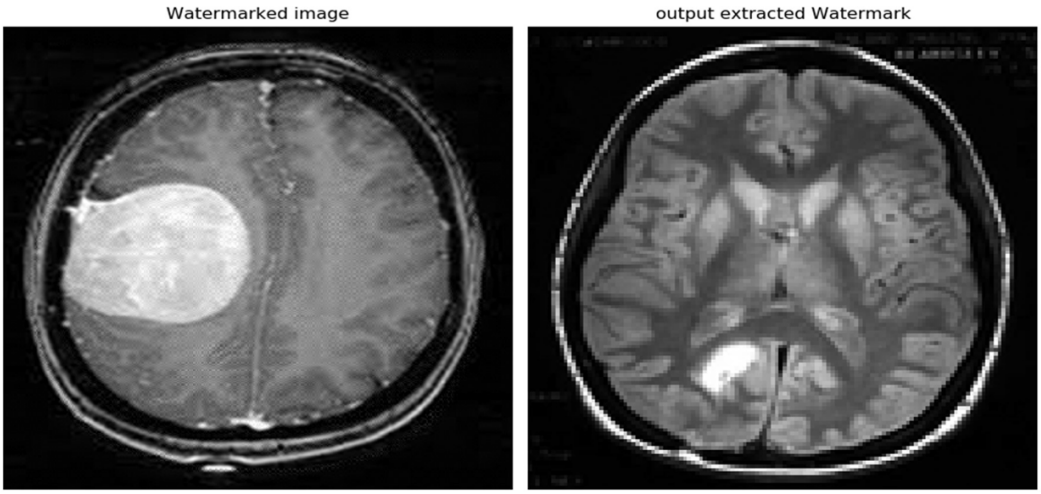
Figure 29 represents the watermarking embedding performance. In (a), we observe a medical image of the brain, which serves as the host image for the watermarking process. This work aims to embed a unique watermark into this medical image while preserving its diagnostic information. In (b), we see the original watermark, which is essentially a distinct identifier or piece of data that needs to be incorporated into the host image. The effectiveness of the watermarking technique used in this work becomes evident in (c), where we witness the output watermarked image. This output image showcases the successful integration of the watermark into the brain medical image, demonstrating the robustness and reliability of the watermarking method employed in this study.



(a) (b) (c)

Figure 29: Watermarking embedding performance. (a) brain medical image. (b) original watermark. (c) output watermarked image

Figure 30 illustrates the watermarking extraction performance. In (a), we are presented with the input watermarked image, which is the result of a prior watermark embedding process. This image carries the watermark information that we seek to extract. In (b), we observe the output extracted watermark image. This image represents the successful retrieval of the watermark from the previously watermarked image, showcasing the effectiveness and accuracy of the watermark extraction algorithm utilized in this work.



(a) (b)

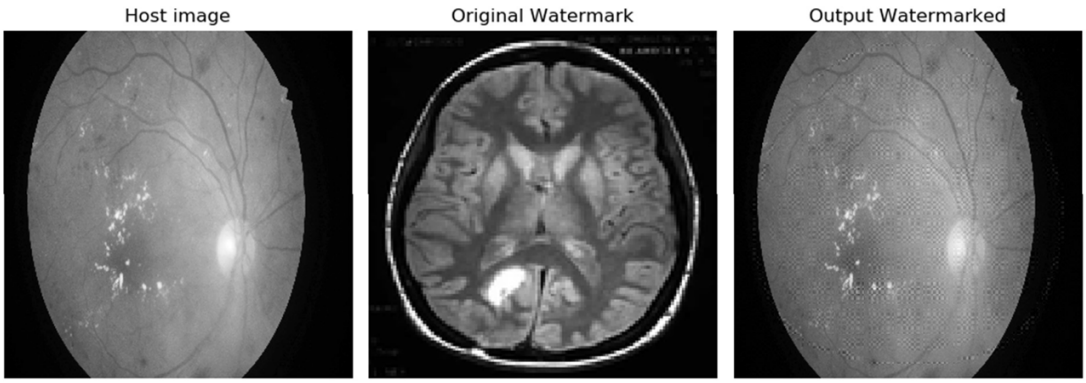
Figure 30: Watermarking extraction performance. (a) input watermarked image. (b) output extracted watermark image.

Figure 31 portrays the watermarking embedding performance. In (a), we encounter a medical image of the DR (Diabetic Retinopathy), which acts as the host image for the watermarking procedure. The objective of this work is to embed a distinctive watermark into this DR medical image while preserving its essential diagnostic content. In (b), we are presented with the original watermark, which serves as the unique identifier or data to be incorporated into the host image. The outcome of this watermarking process is evident in (c), where we witness the output watermarked image. This image signifies the successful integration of the watermark into the DR medical image, underlining the efficacy and reliability of the employed watermarking methodology.

Figure 32 consists of two images illustrating watermarking extraction performance. In (a), we are presented with the input DR watermarked image, a product of a prior watermark embedding process. This image contains the watermark information that we aim to extract. In (b), we observe the output extracted watermark image. This image represents the successful retrieval of the watermark from the previously watermarked DR image. This work highlights the capability and accuracy of the watermark extraction algorithm applied in this study.

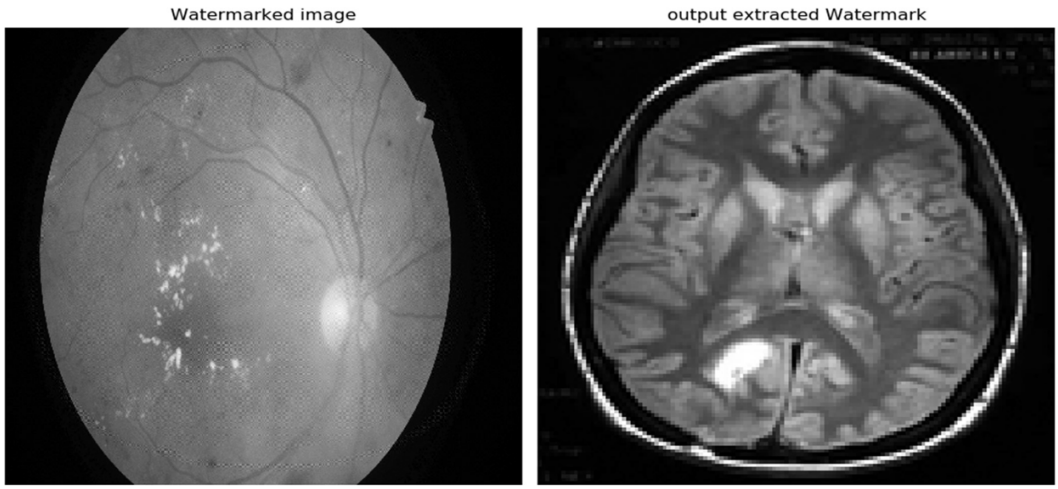
In Figure 33 we delve into watermarking embedding performance once again. In (a), we are presented with a medical image focused on skin, serving as the host image for the watermark embedding process. This endeavor aims to embed a distinctive watermark into this skin medical image while preserving its pertinent diagnostic features. (b) showcases the original watermark, which acts as the

unique identifier or data to be incorporated into the host image. (c) reveals the output watermarked image, which signifies the successful integration of the watermark into the skin medical image. This work underscores the efficiency and effectiveness of the watermarking technique adopted in this research.



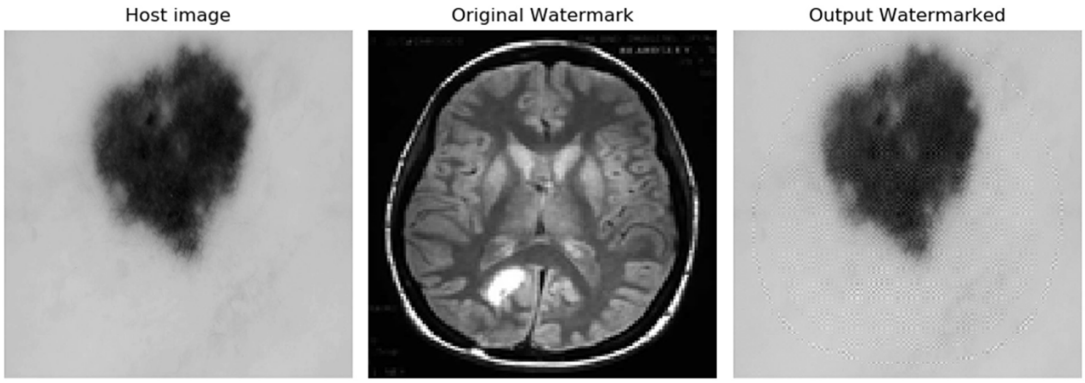
(a) (b) (c)

Figure 31: Watermarking embedding performance. (a) DR medical image. (b) original watermark. (c) output watermarked image



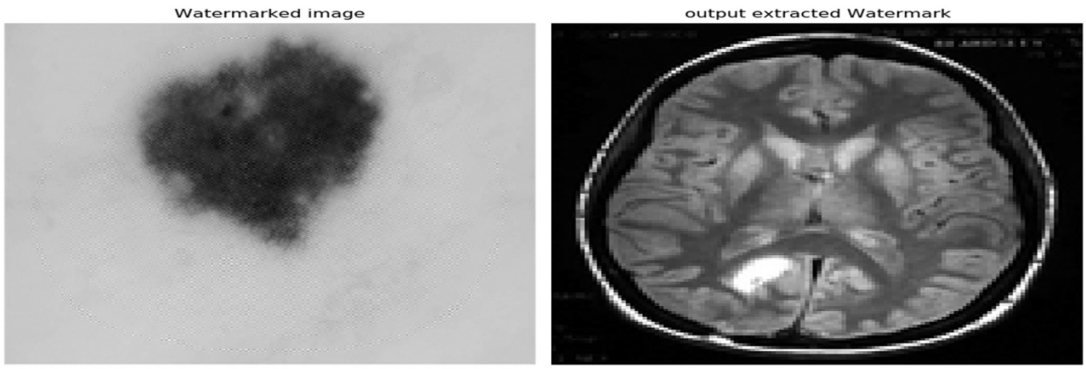
(a) (b)

Figure 32: Watermarking extraction performance. (a) input DR watermarked image. (b) output extracted watermark image.



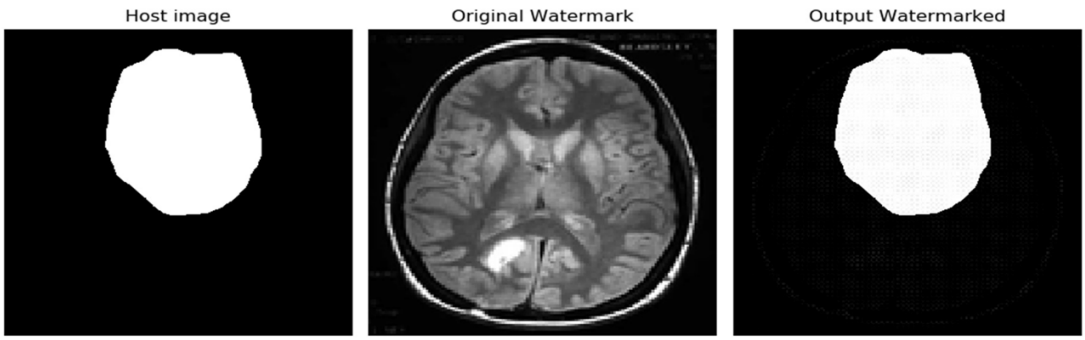
(a) (b) (c)

Figure 33: Watermarking embedding performance. (a) Skin medical image. (b) original watermark. (c) output watermarked image



(a) (b)

Figure 34: Watermarking extraction performance. (a) input skin watermarked image. (b) output extracted watermark image.



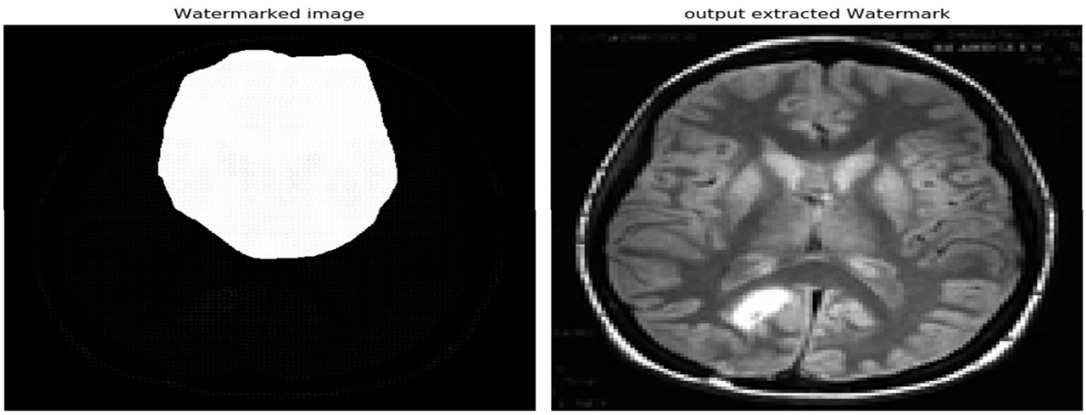
(a) (b) (c)

Figure 35: Watermarking embedding performance. (a) Segmented medical image. (b) original watermark. (c) output watermarked image.

Figure 34 comprises two images that illustrate watermarking extraction performance. (a) presents the input skin watermarked image, a result of a previous watermark embedding process. This image carries the watermark information that we seek to extract. In (b), we witness the output extracted watermark image, representing the successful retrieval of the watermark from the skin watermarked image. This work underscores the accuracy and reliability of the watermark extraction algorithm employed in this study.

Figure 35 depicts watermarking embedding performance once more. In (a), we are presented with a segmented medical image, which acts as the host image for the watermark embedding process. The primary objective of this work is to embed a unique watermark into this segmented medical image while retaining its crucial diagnostic information. In (b), the original watermark is revealed, serving as the distinct identifier or data to be incorporated into the host image. (c) represents the output watermarked image, demonstrating the successful integration of the watermark into the segmented medical image. This work highlights the effectiveness and robustness of the employed watermarking method.

Figure 36 showcases watermarking extraction performance through two images. In (a), we are presented with the input segmented watermarked image, which is the outcome of a previous watermark embedding process. This image holds the watermark information that we intend to extract.
(b) reveals the output extracted watermark image, signifying the successful retrieval of the watermark from the segmented watermarked image. This work emphasizes the precision and reliability of the watermark extraction algorithm implemented in this research.



(a) (b)

Figure 36: Watermarking extraction performance. (a) segmented watermarked image. (b) output extracted watermark image.

9.CONCLUSION

In conclusion, the development of a combined DWT-DCT model-based blind medical image watermarking scheme holds significant promise for enhancing the security and integrity of medical images in IoMT (Internet of Medical Things) applications. By leveraging the Discrete Wavelet Transform (DWT) and Discrete Cosine Transform (DCT) techniques, this watermarking approach offers several advantages, including robustness, imperceptibility, and suitability for blind watermarking, where the original image is not required during extraction. Through our research and experimentation, we have demonstrated the effectiveness of our proposed watermarking scheme in embedding watermarks into medical images securely. The combined DWT-DCT model ensures that the watermark can be embedded seamlessly while preserving the visual quality and diagnostic integrity of the medical images. Furthermore, the blind nature of the watermarking scheme enhances its applicability in scenarios where access to the original image may be restricted or impractical. Our scheme also addresses the critical requirements of IoMT applications by providing robustness against common attacks such as compression, noise addition, cropping, and rotation. Additionally, the watermarking technique facilitates authentication, tamper detection, and self-recovery, thereby ensuring the authenticity and integrity of medical images transmitted over networked systems. Overall, the combined DWT-DCT model-based blind medical image watermarking scheme offers a comprehensive solution for securing medical images in IoMT applications. By safeguarding patient privacy, maintaining data integrity, and ensuring the reliability of medical diagnostics, our approach contributes to the advancement of secure and efficient healthcare delivery in the digital age. Further research and validation in real-world IoMT environments will be essential to fully realize the potential of this watermarking scheme and its integration into practical healthcare systems.

10. FUTURE ENHANCEMENTS

While our combined DWT-DCT model-based blind medical image watermarking scheme represents a significant advancement in securing medical images for IoMT applications, there are several avenues for future enhancements and research:

- **Improved Robustness:** Enhance the robustness of the watermarking scheme against advanced attacks such as geometric transformations, histogram equalization, and advanced image processing techniques. Investigate advanced embedding and extraction algorithms to improve resilience against these attacks.
- **Optimization for Real-Time Processing:** Explore optimization techniques to reduce computational complexity and enable real-time processing of medical images in IoMT environments. This optimization could involve algorithmic improvements, parallel processing, or hardware acceleration techniques.
- **Adaptive Watermarking:** Develop adaptive watermarking techniques that can dynamically adjust watermark strength, embedding density, and other parameters based on the characteristics of the medical image and the requirements of the IoMT application. This adaptability can enhance the effectiveness and efficiency of the watermarking scheme.
- **Enhanced Security Features:** Incorporate additional security features such as encryption, authentication protocols, and access control mechanisms to provide comprehensive protection for medical images in IoMT applications. These features can further safeguard patient privacy and prevent unauthorized access or tampering.
- **Integration with Blockchain Technology:** Explore the integration of blockchain technology to provide a decentralized and immutable ledger for tracking the provenance and integrity of medical images. Blockchain-based solutions can enhance trust, transparency, and auditability in IoMT environments.

11. REFERENCES

[1] Lin CC, Lee TL, Chang YF, Shiu PF, Zhang B (2023) Fragile Watermarking for Tamper Localization and Self-Recovery Based on AMBTC and VQ. Electronics 12:415. https://doi.org/10.3390/electronics12020415.

[2] A. Soualmi, A. Alti, and L. Laouamer, ‘‘A novel blind medical image watermarking scheme based on Schur triangulation and chaotic sequence,’’ Concurrency Comput., Pract. Exper., vol. 34, no. 1, Jan. 2022, Art. no. e6480.

[3] Moad MS, Kafi MR, Khaldi A (2022) A wavelet based medical image watermarking scheme for secure transmission in telemedicine applications, Microprocess Microsyst, 90, https://doi.org/10.1016/j.micpro.2022.104490.

[4] Singh P, Devi KJ, Thakkar HK, Kotecha K (2022) Region-Based Hybrid Medical Image Watermarking Scheme for Robust and Secured Transmission in IoMT. IEEE Access 10:8974–8993. https://doi.org/10.1109/ACCESS.2022.3143801.

[5] H. S. Alshanbari, ‘‘Medical image watermarking for ownership & tamper detection,’’ Multimedia Tools Appl., vol. 80, no. 11, pp. 16549–16564, May 2021.

[6] S. A. Parah, J. A. Kaw, P. Bellavista, N. A. Loan, G. M. Bhat, K. Muhammad, and V. H. C. de Albuquerque, ‘‘Efficient security and authentication for edge-based Internet of Medical Things,’’ IEEE Internet Things J., vol. 8, no. 21, pp. 15652–15662, Nov. 2021.

[7] A. Nagm and M. S. Elwan, ‘‘Protection of the patient data against inten- tional attacks using a hybrid robust watermarking code,’’ PeerJ Comput. Sci., vol. 7, p. e400, Mar. 2021.

[8] V. Rajput and I. A. Ansari, ‘‘Image tamper detection and self- recovery using multiple median watermarking,’’ Multimedia Tools Appl., vol. 79, nos. 47–48, pp. 35519–35535, Dec. 2020, doi: 10.1007/ s11042-019-07971-w.

[9] Memon NA, Alzahrani A (2020) Prediction-Based Reversible Watermarking of CT Scan Images for Content Authentication and Copyright Protection. IEEE Access 8:75448–75462.

[10] Kumar L, Singh KU (2020) An Analysis of Different Watermarking Schemes for Medical Image Authentication. Eur J Mol Clin Med 7(4):2250–2259.

[11] M. Begum and M. S. Uddin, ‘‘Analysis of digital image watermarking techniques through hybrid methods,’’ Adv. Multimedia, vol. 2020, pp. 1–12, Aug. 2020.

[12] F. Sabbane and H. Tairi, ‘‘Medical image watermarking technique based> on polynomial decomposition,’’ Multimedia Tools Appl., vol. 78, no. 23, pp. 34129–34155, Dec. 2019.

[13] B. Hassan, R. Ahmed, B. Li, and O. Hassan, ‘‘An imperceptible medical image watermarking framework for automated diagnosis of retinal pathologies in an eHealth arrangement,’’ IEEE Access, vol. 7, pp. 69758–69775, 2019.

[14] Gull S, Loan NA, Parah SA, Sheikh JA, Bhat G (2018) An efficient watermarking technique for tamper detection and localization of medical images. J Ambient Intell Humaniz Comput 11:1799–1808.

[15] A. Shehab, M. Elhoseny, K. Muhammad, A. K. Sangaiah, P. Yang, H. Huang, and G. Hou, ‘‘Secure and robust fragile watermarking scheme for medical images,’’ IEEE Access, vol. 6, pp. 10269–10278, 2018.

[16] A. D. Andrushia and R. Thangarajan, ‘‘An efficient visual saliency detection model based on Ripplet transform,’’ Sádhanã, vol. 42, no. 5, pp. 671–685, May 2017.

[17] S. Maheshkar, ‘‘Region-based hybrid medical image watermarking for secure telemedicine applications,’’ Multimedia Tools Appl., vol. 76, no. 3, pp. 3617–3647, Feb. 2017.

[18] M. Sharma, ‘‘Medical image watermarking technique in the application of E-diagnosis using M-Ary modulation,’’ Proc. Comput. Sci., vol. 85, pp. 648–655, Jan. 2016.

[19] Vaidya, S.P., Chandra Mouli, P.V.S.S.R.: Adaptive digital watermarking for copyright protection of digital images in wavelet domain. Procedia Comput. Sci. 58, 233–240 (2015).

[20] Singh, A.K., Kumar, B., Dave, M., Mohan, A.: Robust and imperceptible dual watermarking for telemedicine applications. Wirel. Pers. Commun. 80(4), 1415–1433 (2015).