Singular Value Decomposition based Image Authentication Systems and its Applications in Telemedicine

RAMAKRISHNA MISSION RESIDENTIAL COLLEGE(AUTONOMOUS),
NARENDRAPUR, KOLKATA - 700103

B.Sc. Sixth Semester Examination, 2023

Subject: Computer Science

Paper: DSE04 (Project)

Submitted by:

Swarnadeep Das (Registration No: A03-1112-0175-20, Roll no: 6R23CMSA2003)

Ayantanu Laha (Registration No: A03-1112-0184-20, Roll no: 6R23CMSA2011)

Rajarshi Saha (Registration No: A03-1112-0176-20, Roll no: 6R23CMSA2004)

Supervised by:

Sri Bibek Ranjan Ghosh

Submitted on: May 27, 2023

Acknowledgment

It is our great pleasure to express our profound sense of gratitude to our esteemed Supervisor Sri Bibek Ranjan Ghosh, for providing his constructive academic advice and guidance, constant encouragement, and valuable suggestions at crucial junctures and all other support throughout this project work, and for helping us to prepare the project report successfully. We really benefited from his excellent supervision. We would extend our sincere thanks to our respected Head of the Department. Dr. Siddhartha Banerjee, for allowing us to use the facilities available. I would like to thank our lab assistant Shyamaprosad Chakravorty & teachers of our department for extending this wonderful opportunity of working on a project as our DSE-4 in our curriculum.

Certificate

I hereby certify that the project report titled "Singular Value Decomposition-based image authentication systems and its applications in Telemedicine." which is submitted by Swarnadeep Das (Registration No.: A03-1112-0175-20. College roll no: CSUG/060/20), Ayantanu Laha (Registration No: A03-1112-0184-20. College roll no: CSUG/151/20) and Rajarshi Saha (Registration No: A03-1112-0176-20. College roll no: CSUG/066/20) to the faculty of the Computer Science Department of Ramakrishna Mission Residential College in Complete fulfillment of the requirements for the Degree of Bachelor of Science (Honours) in Computer Science, is a record of the project work carried out by the students under my supervision in the academic session of the final semester (semester VI) of 2022- 2023.

(Signature of Head of the Department) (Signature of Supervisor)

Index

Problem Description

Digital watermarking is a crucial technique for embedding and extracting hidden information in digital media, including medical images. Medical image authentication plays a critical role in ensuring the integrity and authenticity of digital medical images, which are essential for accurate diagnosis, treatment planning, and research. The project focuses on the development of a robust watermarking algorithm for medical image authentication using 3 different methods, including techniques like DWT, SVD, Hamming code. The methods aim to embed an imperceptible watermark into medical images, which can later be extracted to verify the authenticity and integrity of the images.

Chapter 1: Introduction

1.1 Steganography:

Steganography is the technique of hiding secret data within ordinary, non-secret files or messages to avoid detection; the secret data is then extracted at its destination. The use of steganography can be combined with encryption as an extra step for hiding or protecting data [1].

After the data is protected in the cover image, the secret data is extracted from the stego image. Perfectly reconstructing the secret data along with the cover image is called reversible steganography [2].

The word steganography is derived from the Greek word ‘steganos ‘(hidden or covered) and ‘graph’(write) [3].

In digital steganographic systems, the fundamental requirement is that the stego-image be perceptually indistinguishable to the degree that it does not raise suspicion [4]. In other words, the hidden informationally briefly modifies to the cover object. Most passive wardens detect the stego-images by analysing their statistical features.

In general, steganalytic systems can be categorized into two classes: spatial domain steganalytic systems (SDSSs) and frequency domain steganalytic systems (FDSSs).

SDSSs are adopted for checking lossless compressed images by analysing the statistical features of the spatial domain. For lossy compressed images, such as JPEG files, FDSSs are used to analyse the statistical features of the frequency domain.

1.2 Watermarking:

Watermarking is a technique with similarities to steganography. It has been around for centuries and is commonly used in money and stamps to assist in identifying counterfeiting. The idea behind watermarking is to create a translucent image on the paper to provide authenticity. Since mailing letters was far more expensive centuries back, it was common for people to use counterfeit stamps on their mail. For example, a translucent elephant watermark was used on stamps in India to deter counterfeiting.

Digital watermarking is used to maintain ownership and authenticity of digital media such as music and videos [4].

It is important to note that although watermarking has many similarities to steganography in terms of embedding data, but the intent of watermarking is not to make it difficult to detect that embedded data, but rather make it difficult to remove the embedded data so as to prevent the unauthorized reuse of the file [5].

Watermarking is of two types; visible watermarking and invisible watermarking.

Visible Watermarking, that refers to the information visible on the image or video or picture. Visible watermarks are typically logos or text. For example, in a TV broadcast, the logo of the broadcaster is visible at the right side of the screen[6]. And Invisible Watermarking, that refers to adding information in a video or picture or audio as digital data. It is not visible or perceivable, but it can be detected by different means. It may also be a form or type of steganography and is used for widespread use. It can be retrieved easily[6].

1.3 Basic model of Steganography

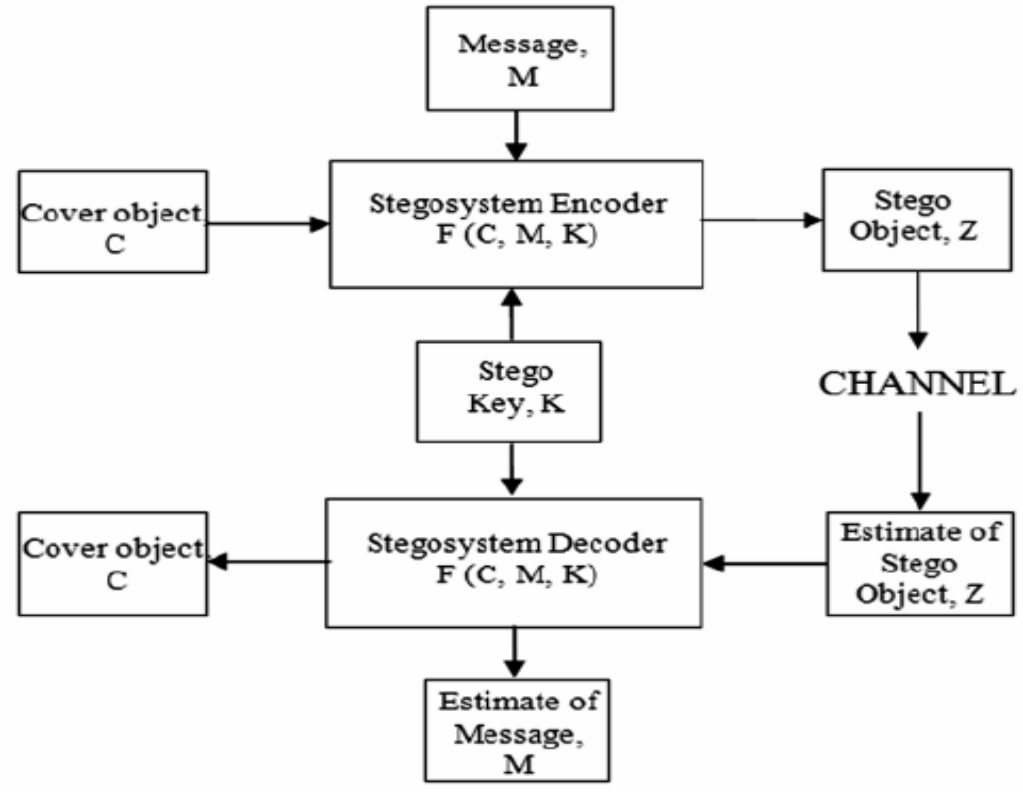


Figure 1: Basic model of steganography

1.4 Characteristics of Steganography:

In steganography, the message to be hidden inside the cover–media must consider the following features[1].

1.4.1 Hiding Capacity: This feature deals with the size of information that can be hidden inside the cover file. A larger hiding capacity allows the use of a small cover and thus reduces the bandwidth required to transmit the stego–media. For example, if we have an RGB image with a size of 200 x 200 pixels, that means that we have 120,000 colour values to be used as cover values for the secret message (200:width x 200:height x 3:R,G,B), then if we use only one bit per colour channel for hiding the message we have a hiding capacity of 120,000 bits or 15,000 bytes, if we use 2 bits per colour channel for hiding the message we have 30,000 bytes, but if we use only one colour channel and one bit per pixel, the hiding capacity will be 40000 bits or 5000 bytes[10].

1.4.2 Perceptual Transparency: Perceptual transparency is an important feature of steganography. Each cover-media has certain information hiding capacity. If more information or data is hidden inside the cover, then it results in degradation of the cover–media. As a result, the stego–media and cover–media will appear to be different. If the attacker notices this distortion, then our steganographic technique fails and there is the possibility that our original message can be extracted or damaged by the attacker[9][10].

1.4.3 Robustness: Robustness is the ability of the hidden message to remain undamaged even if the stego–media undergoes transformation, sharpening, linear and non-linear filtering, scaling and blurring, cropping and various other techniques[7][8].

1.4.4 Tamper–resistance: Of all the features, this feature is very important. This is because, if the attacker is successful in destroying the steganographic technique then the tamper–resistance property makes it difficult for the attacker or pirates to alter or damage the original data.

1.5 Image Steganographic Techniques:

steganography techniques can be divided into following domains [6].

1.5.1 Spatial Domain Methods: There are many versions of spatial steganography, all directly change some bits in the image pixel values in hiding data. Least significant bit (LSB)-based steganography is one of the simplest techniques that hides a secret message in the LSBs of

pixel values without introducing many perceptible distortions. Changes in the value of the LSB are imperceptible for human eyes. Spatial domain techniques are broadly classified into:

1. Least significant bit (LSB)
2. Pixel value differencing (PVD)
3. Edges based data embedding method (EBE)
4. Random pixel embedding method (RPE)
5. Mapping pixel to hidden data method
6. Labelling or connectivity method
7. Pixel intensity-based method
8. Texture based method
9. Histogram shifting methods

Advantages of spatial domain LSB technique are:

1. There is less chance for degradation of the original image.
2. More information can be stored in an image.

Disadvantages of LSB technique are:

1. Less robust, the hidden data can be lost with image manipulation.
2. Hidden data can be easily destroyed by simple attacks.

1.5.2 Transform Domain Technique: This is a more complex way of hiding information in an image. Various algorithms and transformations are used on the image to hide information in it. Transform domain embedding can be termed as a domain of embedding techniques for which a number of algorithms have been suggested. The process of embedding data in the frequency domain of a signal is much stronger than embedding principles that operate in the time domain. Most of the strong steganographic systems today operate within the transform domain. Transform domain techniques have an advantage over spatial domain techniques as they hide information in areas of the image that are less exposed to compression, cropping, and image processing. Some transform domain techniques do not seem dependent on the image format

and they may outrun lossless and lossy format conversions. Transform domain techniques are broadly classified into:

1. Discrete Fourier transformation technique (DFT).

2. Discrete cosine transformation technique (DCT).

3. Discrete Wavelet transformation technique (DWT).

4. Lossless or reversible method (DCT)

5. Embedding in coefficient bits

1.6 Performance metrics for image watermarking:

Various methods are used to evaluate the quality of image watermarking. Each of these methods assesses a different aspect of the result obtained after watermarking. Some of the well-known methods are Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), Structured Similarity Index Measure (SSIM), Payload Capacity[12][13].

1.6.1 Payload capacity:

Payload capacity refers to the measure of the volume of information present within the cover image. This measure is important in a steganographic system as the communication overhead depends on the maximum payload capacity. It is measured in Bits Per Pixel (BPP).
BPP = NUMBER OF SECRET BITS EMBEDDED/ TOTAL NUMBER OF PIXELS

1.6.2 Mean Square Error (MSE):

Mean Square Error is the averaged value of the square of the pixel-by-pixel difference between the original image and stego-image. It gives us a measure of the error produced in the cover image due to the data embedding process.

$$\mathrm{MSE}\ (q1, q2) = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (q_1(i,j) - q_2(i,j))^2$$

Equation 1: Mean Square Error

M, N = Dimensions of the image

I = Original Image

K = stego-image

Where q_1 (i, j) and q_2 (i, j) indicate the original and extracted images, respectively.

Note:

Lower value of MSE indicates good quality of embedding.

1.6.3 Peak Signal to Noise Ratio (PSNR):

PSNR is another popular way to measure the degree of distortion in the cover image due to embedding. It is the ratio between the maximum possible value of a signal and the power of distortion noise (MSE). It is measured in dbs.

$PSNR = 10x\log (Max^2/MSE)$

Max = 255 for an 8-bit grayscale image

Note:

A higher value of PSNR indicates a better-quality embedding.

1.6.4 Structured Similarity Index Measurement (SSIM):

SSIM is a metric of comparison to check the similarity between the cover image and stego-image. It measures the perceptual difference between the two images.

$SSIM= (2\mu_x\mu_y + c_1)(2\sigma_{xy} +c_2)/((\mu_x)^2+(\mu_y)^2 +c_1)((\sigma_x)^2 +(\sigma_y)^2 + c_2)$

c_1 and c_2 are the two stabilizing parameters,

$c_1 = (k_1L)^2$

$c_2 = (k_2L)^2$

L is the dynamic range of pixel values $(2^{\#bits\ per\ pixel} - 1)$

Let the contents, $k_1=0.01$ and $k_2=0.03$.

μ_x and μ_y are the mean intensity values of images x and y.

$(\sigma_x)^2$ is the variance of x,

$(\sigma_y)^2$ is the variance of y

$(\sigma_{xy})^2$ is the covariance of x and y.

Note:

SSIM value close to 1 indicates good quality.

1.6.5 Normalized Cross Correlation (NCC):

It involves computing the similarity between two images by sliding a window over the images and comparing the pixel values within the window.

The similarity is computed using the cross-correlation formula, which involves multiplying the corresponding pixel values in the two images and summing the results. The result is then normalized to obtain a value between -1 and 1, which indicates the degree of similarity between the two images.

$$NCC = \frac{\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}(I_c(i,j)\ I_s(i,j))}{\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}(I_c^2\ (i,j))}$$

Equation 2: Equation of NCC

Here, M is row , N is column, I_c is cover image and I_s is stego image.

Target of performance as a whole:

A low MSE (close to 0), high PSNR(>30dB) and high SSIM value (nearly 1) is desired as a result.

1.7 Discrete Wavelet transformation technique (DWT)

DWT stands for Discrete Wavelet Transform, which is a mathematical transformation that analyzes signals and data in terms of their frequency components and time localization. It breaks down a signal into its constituent wavelets, which are small wave-like functions that are scaled and translated to capture different frequencies and time intervals of the original signal.

The DWT is widely used in signal processing, data compression, image and audio processing, and other fields where the efficient representation of signals in a compact form is essential. It has numerous applications in areas such as image and video compression, denoising, feature extraction, and pattern recognition.

The DWT has several advantages over other signal processing techniques, such as the Fourier transform, because it can capture both time and frequency information simultaneously, and it can handle non-stationary signals with varying frequency content [11].

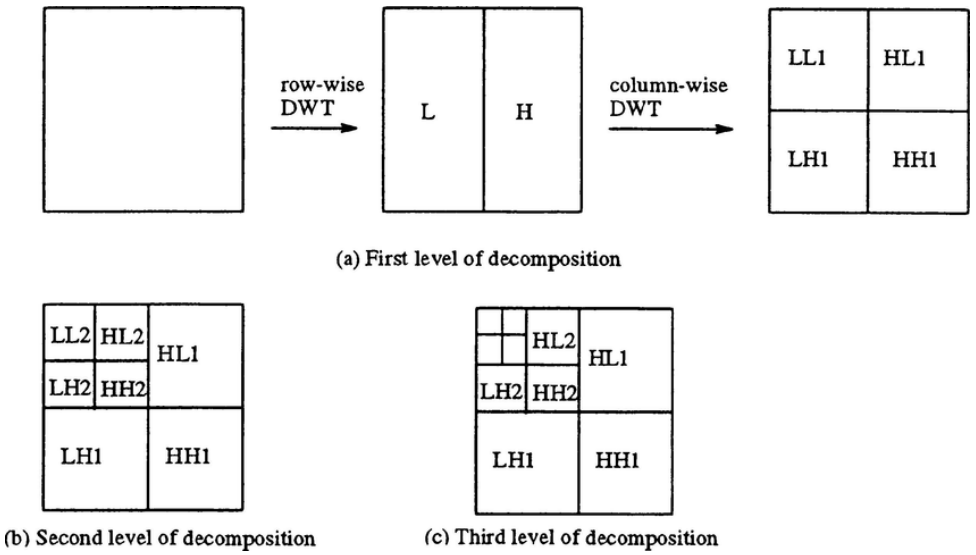


Figure 2: 2D DWT Example

1.8 Singular value decomposition (SVD):

SVD stands for Singular Value Decomposition, which is a fundamental matrix factorization technique used in linear algebra, signal processing, data analysis, and machine learning. It decomposes a matrix into three separate matrices: a left singular matrix, a diagonal matrix, and a right singular matrix[14].

In more detail, given an m x n matrix A, SVD factorizes it into the product of three matrices:

$$A = U * S * V^T$$

where U is an m x m orthogonal matrix, S is an m x n diagonal matrix with non-negative entries called singular values, and V is an n x n orthogonal matrix.

SVD has many applications in data analysis and machine learning. For example, it can be used for data compression, low-rank matrix approximation, image processing, and recommender systems. In particular, SVD can be used to perform principal component analysis (PCA), which is a popular technique for dimensionality reduction and feature extraction.

SVD has several properties, such as it always exists for any matrix, and it provides the optimal low-rank approximation for a given matrix. It also has connections to other matrix factorization techniques, such as eigenvalue decomposition and QR decomposition.

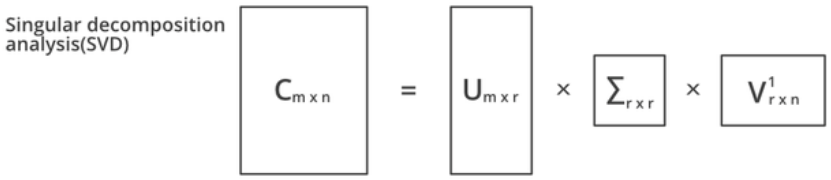


Figure 3: Singular value Decomposition

1.9 Hamming Code:

Hamming code is an error-correcting code used in digital communications to detect and correct errors that can occur during data transmission. It was invented by Richard Hamming in 1950 and is named after him.

Hamming code works by adding extra parity bits to a data message to enable error detection and correction. The parity bits are added in such a way that the resulting code has a specific Hamming distance, which is the minimum number of bit changes required to convert one valid code word into another. This distance is used to detect and correct errors that may occur during transmission [16].

Hamming code is widely used in digital communications, such as in satellite transmissions, data storage devices, and computer networks, where reliable data transmission is critical.

1.10 Location Map Generation:

This process [17] takes input from a seed value S which is the size of the 2D matrix taken as input.

Step-1: Initialize a vector V as {0, 1, 2, …., (S-1)}.

Step-2: Find an M such that M∈V, M>2 and (M, S) are co-prime.

Step-3: Initialize location map LM=φ

Step-4: Generate a random integer F from set V and append F to LM and set Next=F

Step-5: Repeatedly compute Next = (Next + M) Mod S and append Next to LM until Next and F are equal.

1.11 Security Analysis:

1.11.1 Passive attack:

Image steganalysis is a binary pattern classification process whose objective is to correctly distinguish a stego image from a clean one. Broadly, it can be based on the knowledge of the steganographic algorithm behind or without it. The former is specific steganalysis and the latter is universal or blind steganalysis.[18] It is the process of detecting steganography. Basically, there are two methods of detecting modified files. One is called visual analysis, which involves comparing a suspected file with the original copy. It intends to reveal the presence of secret communication through inspection, either by eyes or with the help of a computer system, typically decomposing the image into its bit planes. Although this method is very simple, it is not very effective; most of the time, the original copy is unavailable. [5]

1.11.2 Active attack:

1.11.2.1 Gaussian Filter Attack:

The idea of blurring is to decrease the magnitude of high-frequency components. A low-pass filter in the frequency domain is equivalent to a mountain shape in the spatial domain. Therefore, a smoothing filter in the spatial domain should have all positive coefficients, with the largest in the centre. The simplest low-pass filter would be a mask with all coefficients having a value of 1. [5] A sample 3 × 3 Gaussian filter is: $\frac{1}{12}\begin{bmatrix}1 & 1 & 1\\ 1 & 4 & 1\\ 1 & 1 & 1\end{bmatrix}$

1.11.2.2 Median Filter Attack:

In image enhancement, a median filter is often used to achieve noise reduction rather than blurring. The method is rather simple: we calculate the median of the gray values surrounding the pixel's neighbourhood and assign it to the pixel.[5] The resistance of a watermarked algorithm against mean and median filter depends largely on where the watermark information is embedded. High frequency edge embedding will likely suffer from mean and median filters while low frequency intensity embedding will remain relatively resistant to such filter attacks. [19]

1.11.2.3 JPEG Attack:

JPEG 2000, which is a standard image compression technique created by the JPEG committee, uses the state-of-the-art wavelet-based technology. This creation aims at overcoming the

blocky appearance problem occurred in the discrete cosine transform–based JPEG standard. Its architecture has various applications ranging from prepublishing, digital cameras, medical imaging, and other industrial imaging. In general, JPEG 2000 is able to achieve a high compression ratio and simultaneously avoid blocky and blurry artifacts. [5]

1.11.2.4 Salt & Pepper Attack:

Salt and pepper noise refers to a wide variety of processes that result in the same basic image degradation: only a few pixels are noisy, but they are very noisy. The effect is similar to sprinkling white and black dots—salt and pepper—on the image. By randomizing which pixels are changed, the noise is scattered throughout the image. The combination of these randomizations creates the "salt and pepper" effect throughout the image.

1.11.2.5 Crop Attack:

A crop attack on watermarking refers to a specific type of digital image manipulation aimed at removing or altering watermarks applied to images for copyright or ownership protection. Watermarking is a technique used to embed a visible or invisible marker into digital media, such as images, to identify the copyright holder or the owner of the content. The goal of a crop attack on watermarking is usually to evade detection or claim ownership of the image without proper authorization. It is commonly employed by individuals or entities seeking to use copyrighted images without permission or to misrepresent ownership.

1.11.2.6 Histogram Equalization Attack:

Histogram Equalization is a computer image processing technique used to improve contrast in images. It accomplishes this by effectively spreading out the most frequent intensity values, i.e., stretching out the intensity range of the image. This method usually increases the global contrast of images when its usable data is represented by close contrast values. This allows for areas of lower local contrast to gain a higher contrast.

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) \quad k = 0,1,2,\dots\dots\dots,L-1$$

Equation 3: Histogram Equalization

L is the number of possible intensity levels in an image. Thus a output image is obtained by using the above Eq. to map each pixel in the input image with intensity r_k into a corresponding pixel with level s_k in the output image, is called histogram equalization.

1.12 Gray Level Co-occurrence Matrix (GLCM):

GLCM in texture analysis method in 1973 for satellite image analysis. If M is a gray level image then, co-occurrence matrix C of M, calculates the number of times a pixel pair with an offset $(\Delta x, \Delta y)$ occurs in the image. Here Δx is the difference in vertical pixel position and Δy is difference in horizontal pixel positions of the pair .So an offset of (0,1) means two horizontally adjacent pixel pair whereas offset (1,0) means two vertically adjacent pixel pair. If the number of gray levels in the image M is G, then the size of C is G x G. So for an 8 bit grayscale image the GLCM has size 256x256. If the image size is mxn then, for a pair of intensity levels (i,j), GLCM of M at offset $(\Delta x, \Delta y)$ is defined as.

$$C_{\Delta x,\Delta y}(i,j) = \sum_{x=1}^{m} \sum_{y=1}^{n} \begin{cases} 1, & \text{if } M(x,y) = i \text{ and } M(x+\Delta x, y+\Delta y) = j \\ 0, & \text{otherwise} \end{cases}$$

Equation 4: GLCM(Gray Level Co-occurance matrix)

Offset $(\Delta x, \Delta y)$ can also be represented in terms of (d, θ) where d= relative distance (number of pixels) between pixel pairs and θ = Relative angle (e.g., 0, 45, 90, 135 etc.) between them. [17]

1.13 Principal Component Analysis (PCA) feature of average GLCM:

The average GLCM has 256x256=65536 dimensions, each one is a legitimate feature to feed to the neural network. To reduce this huge input dimension, we have applied PCA to use low dimensional features which retain most of variance/information in data. We have applied singular value decomposition (SVD) technique to do that. The N average GLCMs corresponding to N input images are first flattened to form a combined matrix CGLCM of dimension Nx65536 where each row represents a flattened average GLCM. Now, the SVD of CGLCM is performed to decompose it in three matrices U, S and V i.e CGLCM=USV^T

If original matrix GLCM is rectangular matrix of Nx65536 size then U is an orthogonal matrix of size NxN ($U^T U = I$, where I is identity matrix) and the columns of U are orthonormal

eigenvectors of CGLCM*CGLCM^T. Similarly, V is orthogonal matrix of size 65536x65536 ($V^T V = I$) and rows of V^T are eigenvectors of CGLCM^T * CGLCM. The matrix S is diagonal and it has size Nx65536. The diagonal values of S, called singular values, $\sigma 1, \sigma 2, \ldots \sigma p$ are sorted in descending order of values i.e. $\sigma 1 > \sigma 2 > \cdots > \sigma p$, are the square roots of the non-zero eigenvalues $\lambda 1, \lambda 2, \ldots, \lambda p$ of both CGLCM*CGLCM^T and CGLCM^T * CGLCM .Here, p represent the rank of the matrix CGLCM. Now we have projected the CGLCM in first K eigenvector/principal component of U to transform CGLCM into PGLCM of size NxK in low dimensional Eigen space. [17]

PGLCM=CGLCM. $U^T(K)$

Chapter 2: Literature Survey

R.Liu. et al [20] (2002) presented an SVD-based method. The cover is decomposed into singular matrices U, V, and diagonal matrix S. Watermark W is combined with S with a strength factor i.e., S+aW, and SVD is performed on it to get three matrices. Again, constituents are combined to get the watermarked image. Random matrices are chosen as a watermark. PSNR reported is 53.83dB for watermark size 8192 bits for capacity 0.0315 bpb.

In this article R. A. Ghazy et al. [21] (2007) author proposed a special domain-based algorithm that depends on embedding the watermark into the SVs of the original image after dividing it into blocks. They also experimented with some attacks like Gaussian noise, cropping, and JPEG compression to check the fidelity and robustness of the method. First of all, they have divided the matrix into blocks. Then the watermark is embedded in each block's diagonal matrix (S matrix), giving new matrices. An SVD is performed on each of these new matrices to get the SV matrices of the watermarked image blocks. Then these SV matrices are used to build the watermarked image blocks. By combining these blocks again into one matrix of the original image dimensions, the watermarked image F_w is built in the spatial domain. Some attacks are applied after embedding and then to detect the watermark, they divided the watermarked image (F_{*w} attacked matrix) into blocks having the same size used in the embedding process. Then SVD is performed on each watermarked block (B^*_{wi} matrix) to obtain the SVs of each one (S^*_{wi} matrix), then they extracted the watermark image using U_{wi}, V_{wi}, S_{*wi}, matrices. In this way, they have extracted the possibly corrupted watermark (W_* matrix) from the matrices. The correlation coefficient of the watermark before embedding was 0.9975, after the extraction it was 0.8308.

In this paper, the author N. M. Makbol et al. [22] (2013) proposed an IWT robust image watermarking technique. The property of IWT which maps integer to integer offers a high robustness and good imperceptibility. The good stability and the descending order of singular values of S of SVD transform also contributed significantly to robustness and imperceptibility. They scrambled the watermark by using Arnold Transform to enhance the security aspect. There

are several geometrical and non-geometrical attacks and this method achieved good imperceptibility and high resistance against this. At first, the watermark image is exposed to the Arnold transform to scrabble it before embedding. First one level IWT is applied to decompose the watermark image into 4 sub-band. Now each sub-band is half of its original size. Apply SVD to each of the sub-band. Embedded the scrambled watermark image into the singular value of each subband with a gain factor alpha and again apply SVD to it. Here they use a gain factor of 0.05. Now perform the new modified IWT coefficient. Apply inverse IWT to obtain the watermarked image. In this method, the cover image size is 512 X 512 and the watermark image size is 256 X 256. They conduct the experiment using MATLAB. In this paper, non-geometrical attacks (image processing attacks) such as noise addition (e.g., Pepper & salt noise, Gaussian noise, and Speckle noise), filtering, gamma correction, and JPEG compression attacks are applied, while cutting, shearing, scaling, rotation and translating attacks are selected as the geometrical attacks. The PSNR value they achieved in this paper is 43.8738 dB. The scheme is blind and it shows more security due to scrambling it before embedding.

The main objective of the medical image authentication technique is to protect medical images from tampering and verifying their integrity. In this paper, the author N. Divecha et al. [23] (2013) proposed a DC coefficient-based watermarking technique. Arnold Transform is used to get good robustness and imperceptibility. The author tested this method on various attacks and conclude that the result is good for the LL sub-band as compared to other bands. If watermark removal is difficult for various attacks like rotation, scaling, compression, and noise then watermarking scheme is robust. The proposed 3 different techniques DWT, DCT, and SVD for watermarking. This is also a block-based watermarking technique. They apply 2nd level DWT on the host image and decompose it into four $\frac{N}{4} X \frac{N}{4}$ sub-bands LL_LL, LL_HL, LL_LH, and LL_HH. They select the LL_HH sub-band, divide it into 4 X 4 non-overlapping sub-blocks, and apply DCT. Select the first DCT value and get the DCT coefficient matrix B and apply SVD to B. The watermark image is taken and directly apply SVD to the watermark image. Modify the diagonal matrix of the SVD of the host image with a gain factor and diagonal matrix of the watermark image. The new matrix is modified with inverse SVD. After that apply inverse DCT. Finally, apply the 2nd level inverse DWT to get the watermarked image. The robustness is checked based on different evaluation metrics. In this method, the cover image size is 512 X 512 and the watermark image size is 32 X 32 and 256 X 256. The proposed method is

implemented using MATLAB 7.8.0 software. The highest PSNR value for different types of attacks is 50.8039 dB for the Lena image.

F. N. Thakkar et al. [24] (2016) proposed a blind image watermark scheme based on DWT and block-SVD. They divided the original image into two parts ROI and RONI. They took the ROI for watermark embedding. First, they did the Discrete Wavelet Transform of the ROI of the original image and take the LL matrix to do the next operation. They broke the LL sub-band in to non-overlapping 4x4 parts. They took SVD of each of those 4x4 parts. The left singular matrix U is used for bit embedding. In the watermark image part, they have converted the image into binary strings and also converted the EPR to binary strings. For error correction they used hamming code on EPR. If the watermark image is 32x32 in size then maximum text of 367 characters can be embedded. They embedded the watermark string into the U(2,1) and U(3,1) position of the U matrix in t=such a manner that the maximum of them suggests the value of the bit either 0 or 1. As the embedding key, they took the average of the differences between U(2,1) and U(3,1). After the embedding they calculated the inverse SVD and reconstruct the LL sub-band with all the 4x4 parts. And then did the IDWT to construct the stego-image. In the recovery process, they did the DWT, 4x4 blocks on LL and then SVD to the attacked image. Then they constructed the watermark bit string by simply taking the maximum of U(2,1) and U(3,1). And thus they recover the watermark image and information(EPR). They achieved a PSNR of 44.0333, WPSNR of 50.8285, SSIM of 0.9673. They achieved NCC score of 0.994 with 0.005 salt & peeper noise density.

Chandrasekaran et al. proposed a novel method of reversible Data Hiding primarily focused on medical imagery, i.e., aiming at higher payload capacity, least distortion, and complete reversibility. [25] Chandrasekaran V. et al. (2017) introduced a reversible method of data embedding with medical images with better payload capacity, and least distortion. So, both spatial and transform domains are used for embedding. A discrete wavelet transforms (DWT) on the ‘Haar’ wavelet which generates integer coefficients is used for secret data embedding, and additional reversibility information in the form of a location map is stored in the spatial domain with histogram modification indicating the location of the coefficients positions which are modified. A binary tree is used to hold the multiple peaks and zero points in different histograms. The high and middle frequencies of DWT components are scanned in zigzag format and then the secret data is embedded. During extraction, histogram recovery is applied

to the stego image to find the auxiliary data and then integer to integer Haar transform is used to get the image in the transform domain. The proposed method is tested using R2010 sample images in MATLAB. Experimental results reveal that PSNRs are 62.56, 52.85, 50.92, and 45.85 dB at 0.1, 0.3, 0.6, and 0.8 bpb respectively for “hepatitis” image.

In this paper, V. Malik et al. [26] (2017) have proposed a DWT-SVD-based algorithm to embed watermark. First, they have taken the 1-level 2D DWT of the original image. Then they applied SVD to the LL sub-band of the image. They did the same with the watermark image. Then they applied their watermark embedding algorithm to the two images to generate the watermarked image. In the extraction process, they have taken two images, i.e., the original image (cover image) and watermarked image. Then they applied DWT transform on both images. SVD on the LL part of both images has been applied. Then they applied the extraction algorithm on the two images to generate the watermark image. In the experiments, they have taken 3 images, Baboon, Lena, and Peppers from the USC SIPI dataset. In the embedding, they got PSNR of 44.0242, 35.6922, and 35.8694 respectively, and in extraction, they got PSNR of 20.46, 21.0853, and 14.3917 respectively.

In the past decades, usage of the digital media is the transfer of data itself over the internet. Digital image watermarking plays a vital role in this transfer of data securely. This technique is mainly employed for ownership identification, authorization, and copyright protection. There are two types of image watermarking techniques. The watermarking technique which is applied to the entire image is called non-block-based watermarking and those applied to each block of the image are known as the Blocked based watermarking technique which will decrease the rate of distortion of the watermark on several types of attacks. D. K. Shaw et al. [27] (2017) proposed a blocked-based DWT -SVD-based watermarking technique. In this paper the author divides into 8 X 8 block size and then apply SVD on each block size for watermarking. They apply 2nd level DWT on the LL sub-band and produce LL2 LH2 HL2 and HH2 sub-bands. As they are using a color image instead of a Gray level image, there is a red, green, and blue component of the image. Apply SVD to each of the components. Take the watermark image and repeat the same process. Embed the SVs of each Red, Green, and Blue component of each block of the watermark block into corresponding blocks of the original image with a factor a where a is the gain factor. By using a new value the RGB component of the image is recreated

using inverse SVD. Finally, perform 2 level inverses DWT with modified subband and found out the watermarked image. The proposed method is implemented using MATLAB (R2015a) software. In this technique of the watermark embedding process, they don't use any encryption technique and embedding in random order in the host image. The performance evaluation metrics like PSNR, Correlation coefficient are used. To check the performance of the proposed method the author used different attacks and gave a comparative study between them. This paper is well resistant to the impact of Gaussian noise, salt, and pepper noise.

In this paper, R.H. Laskar et al. [28] (2018) proposed a neural network-based robust image watermarking technique in the lifting wavelet transform domain. The neural network is incorporated into the watermark extraction process to achieve improved robustness against different attacks. The first 3-level LWT of the cover image is taken. Then the HL3 sub-band coefficients are randomized using secret key 1. Then the randomized coefficients are divided into 2X2 blocks and then all blocks are again randomized using secret key 2. The watermark image is divided into 2 1-D vectors SW and RW which are again randomized using secret key 3 and then concatenated. Then the binary single-dimensional concatenated watermark is embedded with those 2X2 blocks. After taking the inverse LWT the stego-image is produced. During the extraction process, Artificial Neural Network is used. 10 statistical features are extracted from the image and then they are trained in the ANN. Then using the secret key 1, 2, and 3 the watermark is extracted. This technique is tested on 200 gray images of size (512X512). The proposed scheme provides an average imperceptibility of 43.88 dB on a large image database with an embedding capacity of 512 bits. It has been observed that the algorithm provides robustness against different intentional and non-intentional attacks. Experimental results suggest that the proposed technique gives a satisfactory performance on different types of images under various signal-processing attacks.

S. Ojha et al. [29] (2018) proposed a method that converts RGB cover images of size 512x512 to the YCbCr coefficient. Then crop the central 256x256 part ROI from Y as X. Apply DWT to X and perform SVD on the HL component. Now resize the watermark to a 256x256 grayscale image and perform DWT on it and further SVD. Now embed the SVD component of the cover and watermark with a strength factor. Then get the Stego ROI (Region of Interest) image using inverse SVD followed by inverse DWT. Now paste the ROI into its position of Y

and inverse transform to RGB space to get the final Stego image. PSNR of 58.48 is reported for Lena as a cover image with a 256x256 watermark.

In this paper, author A. K. Singh [30] (2019) proposed an LWT-DCT-based method. The host image of size 512 X 512, the watermark image of size 64 X 64, and a text watermark of size 80 characters are considered as 'patient report'. At first, the host image is decomposed using 3rd-level LWT. The LH sub-band is taken and apply DCT on it. Transform the signature watermark by DCT. After that encrypt the transformed DCT watermark image using the MD5 hash algorithm. Embedded the watermark into the host image using a gain factor. Embedded the encoded 'patient report' in the HL sub-band. Finally, the watermarked image is obtained using inverse DCT and inverse LWT. The best PSNR value is 34.72 dB for '60' and '80' characters at a gain of 0.05. However, the NC value is 0.8572 (for 60 characters) and 0.8524 (for 80 characters) at the same gain factor. The best NC value is 0.9813 for 25 characters at a gain factor of 0.2.

A. Anand et al. [16] (2020) proposed an improved non-blind, robust, imperceptible, and secure watermarking technique for medical image watermarking. In this paper, the author proposed an improved DWT-SVD domain-based watermarking technique. They also used hamming code to the text watermark in order to reduce the channel noise. They use a text watermark and an image watermark both. Before embedding the watermark, the image is divided into two parts. These two watermarks are embedded into the cover image. In the first part, consider an MRI image and 2nd level DWT is performed on the LL subband. In 1st level DWT the image is decomposed into four sub-bands: LL1, HL1, LH1, and HH1. LL is the approximate coefficient of the image. Again, apply DWT on the LL subband and decomposed it into four subbands: LL2, HL2, LH2, and HH2. After that, they perform SVD (Singular value decomposition) on HL1 and LH1 subbands and embedded the two divided image watermarks into this two-sub band using a gain factor-alpha. The text watermark is first converted into binary form and applied hamming error code to obtain the encoded message. The text watermark is embedded into the HH2 subband. The watermarked image is obtained using inverse DWT. Finally, the watermarked image is encrypted using a Chaotic or Hyperchaotic encryption algorithm. For compression of the watermarked image, the author used Huffman compression or LZW compression, or a hybrid of both compressing techniques. In this method, the cover image size is 512 X 512 and the watermark image size is 256 X 256 and the text size is up to 12 characters. They conduct the experiment using MATLAB R2017a. The proposed method is simulated on

different values of gain factor (α) as 0.05 and 0.10. Without any noise attack, the highest PSNR = 36.1007 dB is obtained using chaotic encryption with α = 0.05 with the highest NC = 0.9911 at α = 0.10 and BER=0. The highest PSNR value of 44.1944 dB is observed for Cell images using the Hyperchaotic-LZW technique. The author applies different types of attacks to check the robustness of the Hyperchaotic-LZW technique. We see that for the cropping attack the NC value is small i.e., 0.5082.

Today digital image watermarking plays a vital role in copywriting protection, and security in today's digital transmission medium. In this paper, F. Yasmeen et al. [31] (2021) proposed an embedding and extraction process carried out through multilevel operations of DWT, and SVD. The intangibility is calculated by PSNR value and robustness is calculated by correlation among the original watermark and recovered watermark image. The author applies up to the 4^{th} level DWT on the host image and up to the 3^{rd} level DWT on the cover image. In LL there are low-frequency components and our human visual system is more sensible in the low-frequency component so the watermark is ideally put into the other 3 sub bands. But in this method, the author embedded the watermark in the LL band. The author applies the 4^{th} level DWT on the cover image and produces LL4, LH4, HL4, HH4, and 3^{rd} level DWT on the cover image and produces LL3, LH3, HL3, and HH3. After that, the author applies the 1^{st} and 2^{nd} level SVD on the HL4 and LH4 sub-band. Fit the watermark into the host image using singular values with a gain factor-alpha. Finally, apply Inverse SVD and after that 4^{th}, 3^{rd}, 2^{nd}, and 1^{st} level inverse DWT to compute the final watermarked image. Here in this method, the author doesn't apply any encryption algorithm to secure the robustness and imperceptibility of the watermarked image. In this method, the cover image size is 512 X 512 and the watermark image size is 256 X 256. They conduct the experiment using MATLAB R2016a software. The Performance evaluation metrics like PSNR, NC, SSIM, etc. are used. The PSNR values of Lena (grayscale) and Lena (color) are found here as 43.8362 and 34.7266 dB. In SVD watermarking technique the main problem is a false positive error. The error happens during the extraction of the watermark image.

In today's E-era R. Thanki et al. [32] (2021) proposed a watermarking technique in telemedicine applications. They proposed a watermarking scheme combination of finite ridgelet transform (FRT), SVD, and Arnold scrambling-based encryption. If an image is of size

M X N then, after applying ridgelet transform the size becomes 2M X 2N which improves the payload capacity of the watermarking scheme. The reason behind using Arnold scrambling in the proposed scheme is that it is easy to implement and has a fast computational time compared to other encryption algorithms such as DES, AES, etc. In this method, the size of the host image was 128 X 128 and the size of the watermark image is 256 X 256. Sample patient information is also taken as the watermark. The cover medical image and watermark image are taken. The first level forward FRT is applied to the cover medical image to get its ridgelet to transform coefficients after that apply SVD to the coefficients. SVD is applied to the watermark image. The singular value of the watermark image is embedded into the hybrid transform coefficients of the cover medical image. Inverse SVD is applied to the modified singular value. First-level inverse FRT is applied to get the watermarked image. Finally, forward Arnold scrambling is applied to the watermarked medical image using a secret key. PSNR is calculated between the original cover medical image and watermarked medical image and NC is calculated between the original watermark image and extracted watermark image. The best PSNR value and NC values such as 52.8761 and 1.000 for the proposed scheme without any attack. The NC value after various attacks is always greater than 0.75.

In this paper, C. C. Adaobi et al. [33] (2021) proposed a novel MIW technique based on the redundant discrete wavelet transform (RDWT) with singular value decomposition (SVD) to increase its performance in embedding a watermark image that may be as large as the cover picture. According to them, firstly the RDWT-SVD is a reliable solution as compared to the conventional DWT and secondly modifying the wavelet domain coefficient ensures that integer values in the spatial domain change and that the watermarking process is reversible. As a result, it decreases the amount of original picture change and improves imperceptibility as compared to the conventional approaches. Their used method is as follows: The biological picture is first pre-processed in the model using an enhanced successive mean quantization transform called RDWT, which also employs SVD decomposition. RDWT and IRDWT are then used to insert the watermark in the picture. Finally, using the inverse operation of the embedding technique, the watermark is recovered from the biological picture. The two key components of the process are the high-speed permutation process and adaptive diffusion. They have also used the chi-square test to analyze the regularity of the histogram. Their used formula for this new approach for embedding is $S_{1*} = S_1 + x * S_{w;}$ where S_1denotes the obtained SVD coefficients of LL (sub-part of cover image), while S_w is obtained SVD coefficients of LL_1(sub-part of watermark). S_1

indicates the SVD coefficients of LL and S_W means the SVD coefficients of LL_1. The formula used for extraction of the watermark is $S_E = (S_W - S_1)/x$: where S_E is the SVD coefficient of the watermarked image. As the final result, using this approach the result of PSNR becomes 65.456 along with SSIM and NCC, 0.983 and 0.9738 respectively. Also, they have proved changing singular values of the cover picture gives great robustness with common attacks.

In this paper, N. Zermi et al. [34] (2021) author proposed a hybrid approach based on a discrete wavelet transform (DWT) and singular value decomposition (SVD). Dividing the image into sub-frequency bands allows the development of psycho-visual marks easily. First, they concatenated the watermark image parts by applying the MD5 algorithm and made a 128-bit hash. Then they converted the watermark into binary strings. They applied DWT in the cover image and then applied SVD to the LL sub-band. They have taken the S matrix i.e., the Singular value part of SVD to embed the watermark bits. They have calculated the embedding process in 2 variants. In the first variant, they subtracted two consecutive coefficients of the S matrix and calculated the remainder by dividing it by 2. According to the result, they have modified the S values. In the second variant, 3 successive coefficients are used to integrate two watermark bits in the same fashion as stated before. After the embedding, they merged the U, S, and V matrices to reconstruct the LL subband and then did the inverse DWT to get the watermarked image. In the extraction process again, they divided the image into 4 sub-bands by taking its DWT and then calculated the SVD of the LL sub-band. Then they have taken 2 consecutive singular values for the first variant and 3 successive singular values for the 2nd variant and subtracted them to get the bits of the watermark image. For the first variant, they have got a PSNR of 56.12, MSE of 0.10, NPCR of 4.2%, SSIM of 0.9998, and BPP of 0.10. For the 2nd variant, they have got a PSNR of 57.41, MSE of 0.13, NPCR of 4.0%, SSIM of 0.9998, and BPP of 0.13.

To secure the heartbeat sound exchange the author M. S. Moad et al. [35] (2021) proposed a blind watermarking approach based on combining a Non-Subsampled Shearlet Transform and Singular Value Decomposition. Combined patient Photography, audio file acquisition data, and patient information are used as watermark. Arnold transform is performed in patient photography. The host audio files used in their experiments are heartbeat sounds from Stanford University. The host audio file is converted into a 2d signal. To better present the low and high-

frequency information, a Non-Subsampled Shearlet Transform is then applied to the host signal. Using an NSLP the audio file is in this manner decomposed into two high-pass and low-pass bands. A Non-Sampled Laplacian pyramid transforms (NSLP) is carefully applied to the low-pass bands and shift-invariant fshearing filters are applied to the high-pass bands. After that SVD is applied and modified the singular value to get the watermarked audio file. Without any attack, the SNR value is 32.64, and with attacks the NC value for NSST based approach is 0.9875.

The author D. Awasthi et al. [36] (2022) proposed a DWT SVD embedding scheme using JAYA and the Particle Swarm Optimization technique. The scale factor is highly important in watermark embedding since it regulates the robustness and imperceptibility of the watermark image. To discover an acceptable scale factor JAYA and Particle swarm optimization technique is used. This paper [7] proposed a comparison of two schemes DWT-DCT-SVD and LWT-DCT-SVD. The host image is taken and decomposed into 4 subbands using DWT/LWT technique and an LL subband is produced. The LL subband is again decomposed using DWT/LWT technique and the LL2 subband is produced. The LL2 sub-band is further decomposed using DWT/LWT technique and four smaller sub-bands: LL3, HL3, LH3, and HH3is produced. DCT is applied on the LL3 subband. Apply SVD on it and we get two unitary matrices and one dominant matrix. The watermark image is taken and apply 2^{nd} level DWT/LWT on it. Apply DCT on the LL subband. Apply SVD on it. The singular component of the host and the watermark picture are added using an embedding function and alpha is the scaling factor calculated by JAYA and PSO. Rebuild the sub-bands using inverse SVD, then apply inverse DCT to get the third-level sub-band. Again, apply 3-level inverse DWT/LWT on sub-bands to get the watermarked image. In this method, the cover image size is 512 X 512 and the watermark image size is 256 X 256. The peak signal-to-noise ratio in all the cases is greater than 30 dB and the peak signal-to-noise ratio in the case of Lena image is the highest i.e., 44.9488 using JAYA optimization. DCT-SVD scheme with particle swarm and JAYA optimization is shown, the PSNR value in the case of PSO without attack is 42.5765 and in JAYA the value is 44.9488. When we apply different types of attacks the value of PSNR varies. The lowest value of PSNR for a dithering attack with PSO is 16.7423. When we apply JAYA optimization PSNR values are greater in comparison with particle swarm optimization.

In this paper A. A. Mohammed et al. [37] (2022) proposed a FDCT and radon transform (RT) based watermarking algorithm. Randon Transform is used to increase the robustness of the watermarking scheme against geometric attacks. Frequency wrapping-based FDCT is applied up to scale 4. At first, the binary logo watermark is taken, and transform the watermark into RT to get the sinogram. Read the host medical image and apply the frequency wrapping base FDCT. The low-frequency and high-frequency curvelet coefficients are obtained. Perform SVD on the low-frequency and high-frequency coefficients separately and embedded the first watermark and second watermark into a diagonal matrix obtained from SVD respectively. Apply inverse SVD on both separately. Apply inverse frequency wrapping-based FDCT and get the final watermarked image. In this method, the cover image size is 1024 X 1024 and the watermark image size is 64 X 64. The work is conducted under a PC with Intel 2.5 GHz core i7cpu with 4 GB of ram under the MATLAB environment. Before the attack, the PSNR value of Original medical images and their watermarked versions vary from 55 dB to 59 dB.

In this paper, G. P. Dubey et al. [38] (2022) proposed a DWT SVD watermarking technique for the YCbCr color model. This format is frequently used for watermarking because this model allows no correlation between components and perceptually optimum color space. The actual RGB image is taken and converted into YCbCr. The number of DWT level are selected automatically based on the size of the cover image and logo. Haar wavelet filter is used in DWT. After that SVD is applied. YCbCr watermark and a grayscale signature image were taken and embedded into the host image using a scaling factor. Now inverse SVD and DWT were performed and finally convert YCbCr to RGB. In this method, the cover image size is 512 X 512, and the variable watermark size is 256x256, 128x128, and 64x64. The PSNR value after extraction without any attack is 30.95 dB.

Image watermarking is a vital process. Recently a large number of works were found to be to enhance the performance of digital watermarking. In this paper [39] the author Ms. R. R. Kumari et al. [39] (2023) proposed a DWT SVD scheme with an enhanced tunicate swarm optimization algorithm. It is a block-based watermarking technique. The block is selected randomly based on low variance by TSA optimization. Before watermarking the watermark is scrambled by Arnold scrambling and Tent Map chaotic encryption for double-layer security to the watermark image. The AT scrambling and chaotic Tent map is applied in the watermark to improve the security of

the watermark image. The host image is taken and DWT is computed on it. The LL subband is taken and divided into 4X4 nonoverlapping subblocks. The random blocks based on low variance are selected using TSA optimization. The SVD function is used extensively to gain strength and imperceptibility to the embedding process and divides the image into three matrices. The watermark image is taken and compute DWT on it. After that computer SVD on LL sub-band. The secret key image is scrambled by Arnold's transformation. Invaders may have to struggle to recognize encrypted watermarks without a secret key. Insert a scrambled watermark image into the cover image and the watermark embedding process is achieved. In this method, the cover image size is 512 X 512 and the watermark image size is 64 X 64. The proposed method is implemented using MATLAB software. When a watermarked image is passed over the network, the attacker may attempt to delete or recover a secret image. Therefore, it is necessary to create a robust watermarking algorithm in these types of situations. The proposed method achieves better results in terms of average PSNR = 82.56db, NC = 0.999, and SSIM = 0.999.

In this paper S. Sattarpour et al. [40] (2023) combines graph-based and discrete wavelet transforms with the whale optimization algorithm to embed a watermark into an image in an optimal and secure manner. The whale optimization algorithm is used to find the best value for embedding parameters. The host image is considered as an RGB image & the watermark image is a binary image. The blue channel is taken for watermarking and divided into 8 X 8 blocks. For each selected block a 2D DWT using the Haar wavelet transform is applied to the block. Choose the LL band and apply Graph-based transform (GBT) and SVD to it. The embedding process is completed using the diagonal matrix S. S_W is the biggest singular value of the watermark image and S_h is the biggest singular value of the host image. If the watermark bit is 1 then $S_W=S_h+\alpha$, else $SW=S_h-\alpha$. The proposed method is semi-blind. Because we need S_h values, i.e., a part of the host image for the extraction. Now we can use Sw, the inverse of SVD, and the inverse of GBT to make the LL component again. Then we apply inverse DWT to make the watermarked image. The performance of the proposed method is evaluated through various experiments, including a comparison with existing watermarking techniques. The results demonstrate that the proposed approach achieves improved robustness against common attacks such as noise addition, compression, and filtering. Additionally, it exhibits better imperceptibility, ensuring that the watermark does not significantly affect the visual quality of the watermarked image. In the grayscale image, the average PSNR value between watermarked image and the host image

without attack is 36 dB. But in the whole image with all three-color channels, the PSNR value has reached more than 70 dB.

In this paper, P. Garg et al. [41] (2023) focused on the development of a secure and reliable method for biometric image authentication. The authors address the need for protecting biometric images from unauthorized modifications or tampering, which could compromise the integrity and authenticity of the biometric data. The LL band is considered for watermark embedding because of holding more information. The medical image comprises 4X4 Gray blocks and entropy is estimated. The block which has high entropy is considered further. Using haar wavelet transform decomposed the image into 2 levels after that perform SVD. The watermark image is taken and perform chaotic encryption. Perform DWT and SVD on the encrypted watermark. Embedded the singular value of the watermark into the host image using a gain factor-alpha. Perform SVD using the new singular value. Apply inverse DWT to generate the watermarked image. The average PSNR values for medical images are 50.43 and 52.97 dB. The average PSNR values for non-medical images are 51.86 and 51.06 Db.

Chapter 3: Proposed Method

Watermarking in medical images is a pretty difficult and very sensitive work. The information of patient should be kept accurately without any error. So, it should be robust against various attacks. For watermark embedding in medical images 3 different methods are proposed. All have been experimented with SIPI, BOSSbase and medical image dataset.

3.1. Robust DWT based Image authentication using S matrix of Block based SVD (RDWTSVDSVAUTH)

The first method consists of DWT and block-based SVD. DWT is taken for better imperceptibility. S matrix of SVD is chosen for embedding as experimental results show that some change in the values of S matrix, affect very little to the actual data. A location map generation algorithm for block selection is also applied for better security.

3.1.1 Embedding Process

After dividing the cover image by taking its 1-level Discrete Wavelet Transform (DWT), the LL sub-band is divided into non-overlapping blocks of size 2x2. Each block is then decomposed by Singular Value Decomposition (SVD). The S matrix is taken for embedding. The watermark image is then normalized between value 0 and 1 by dividing it by maximum pixel value, e.g., 255 for 8-bit image.

$$w'_{ij} = \frac{w_{ij}}{\text{Maximum pixel value } (\alpha)}$$

Equation 5: Dividing by maximum pixel

A random block selection algorithm is selected for selecting blocks for embedding which will increase some security. The normalized watermark bits are then added with the top-left element of S matrix of every block.

$$s'_{ij}[0][0] = s_{ij}[0][0] + w'_{ij}$$

Equation 6: Added watermark bit

The HH sub-band is divided into 2x2 non-overlapping blocks. SVD is taken for each of the blocks and the S(0,0) element of the old S matrix is stored in the (0,0) position of the S matrix of each block for future use.

Algorithm1: Embedding (COVER, WI)

Step1: Take the 1-level 2D DWT of COVER image.

Step2: Divide the LL sub-band into non-overlapping blocks of size 2x2.

Step3: Generate a sequence for selection of blocks and take the SVD of each block according to the sequence.

Step4: Store the (0,0) value of S matrix in the HH sub-band for future use.

Step5: Add normalized watermark pixels with the (0,0) element of S matrix of each block.

Step6: Calculate inverse SVD to reconstruct each block.

Step7: Join the blocks to create the new LL.

Step8: Take IDWT to get the watermarked image.

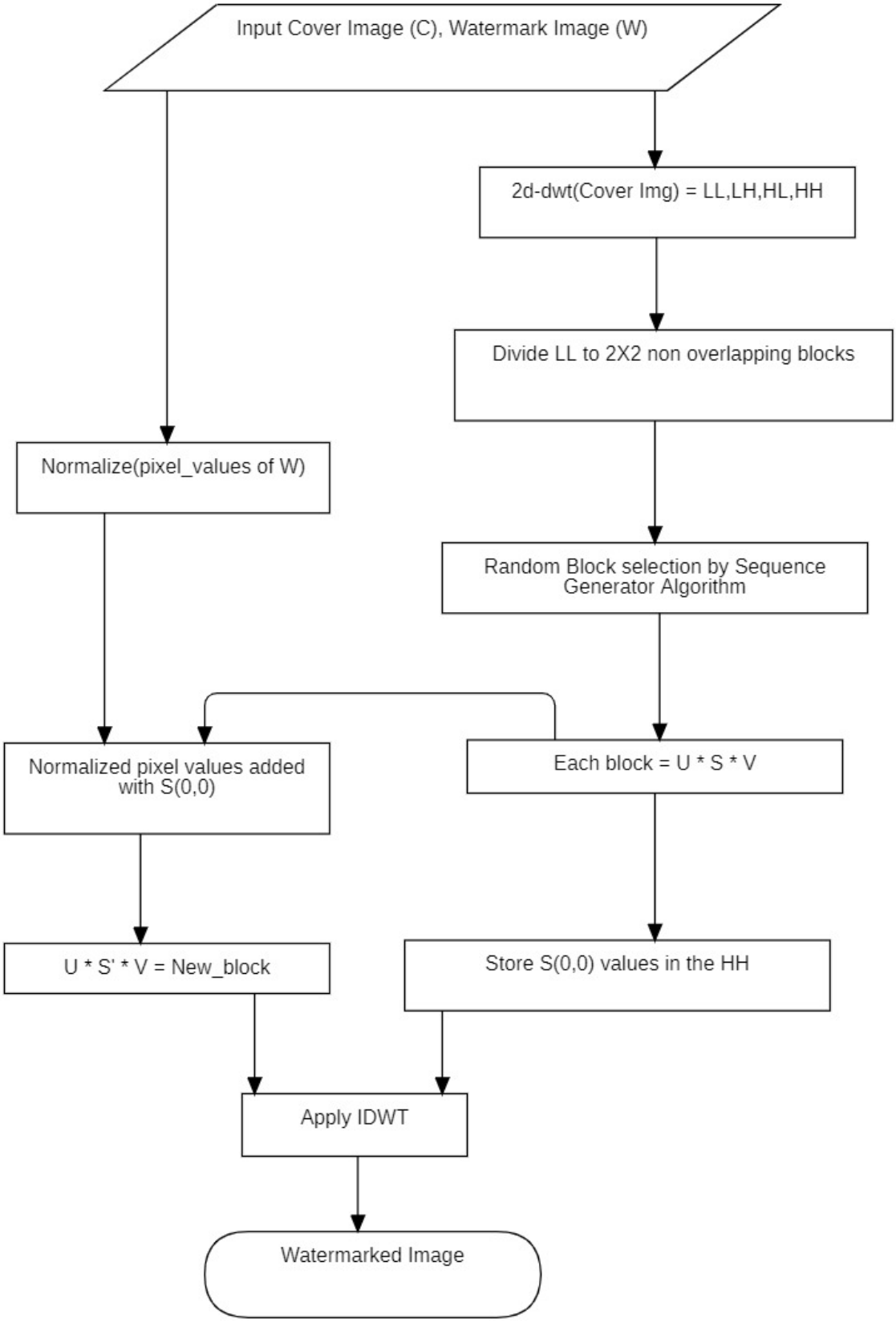


Fig: 4 Flowchart for embedding process of RDWTSVDSVAUTH

3.1.2 Extraction Process

For the extraction of the watermark, first 1-level 2D DWT of the watermarked image is taken. HH sub-band is then divided into 2x2 non-overlapping blocks and the first element of each S matrix is taken as the watermarking key by taking SVD of each block. LL sub-band is then divided into 2x2 non-overlapping blocks. The block selection algorithm is again used to generate the block sequence. S matrix is taken after decomposing every block using SVD. The watermark image pixels are then calculated using the equation

$$W_{ij} = \left(S_{ij}^{ll} - S_{ij}^{hh}\right) * \alpha$$

Equation 6: Extraction process

Algorithm2: Extraction (STEGO, WSIZE, α)

Step1: First, divide the given watermarked image into 4 sub-bands LL, HL, LH & HH using 1-level 2D DWT.

Step2: Take the HH part and divide it into 2x2 non-overlapping blocks.

Step3: Take SVD of each block and extract the value of position (0,0) of S matrix of each block, say it S'.

Step4: Take the LL part and divide it into 2x2 non-overlapping blocks.

Step5: Again, use the block selection algorithm to generate block sequence.

Step6: Calculate SVD of each block.

Step7: Calculate the watermark image pixels by subtracting the value of (0,0) index of S matrix with corresponding value of S' and multiply it with α.

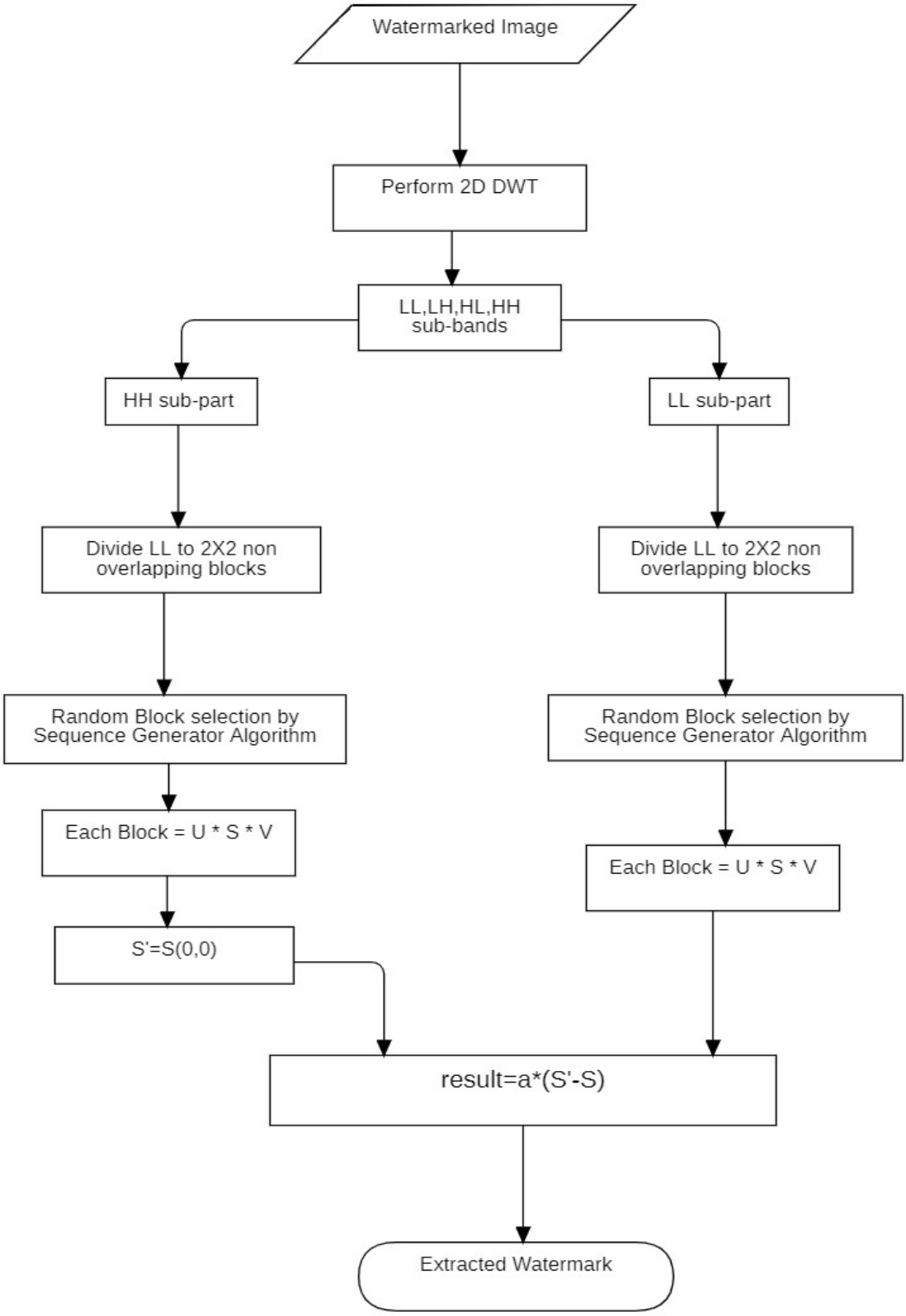


Fig: 5 Flowchart for extraction process of RDWTSVDSVAUTH

3.2 Robust DWT based Image authentication using U matrix of Block based SVD (RDWTSVDUMAUTH)

The first method gives good PSNR value and very good robustness under a certain number of attacks. But fails to give satisfactory result under some attacks like JPEG compression, Histogram Equalization etc. So, another method is thought of to gain robustness. This method is also based on DWT and block-based SVD. But here the U matrix of SVD is selected for embedding. Without embedding actual watermark pixels into the cover image, the difference of 2 values of U matrix is taken as a threshold parameter to determine a single binary bit of watermark.

3.2.1 Embedding Process

Watermark image is converted into 8-bit binary string. LL sub-band is taken after taking 2D DWT of cover image. The LL sub-band is then divided into 2x2 non-overlapping blocks and SVD of each block is taken. Sequence generation algorithm is applied for block selection. Two values of position (0,0) and (1,0) of U matrix of each block is checked for one single bit embedding. If the watermark bit is 0 then U(0,0) should be greater than U(1,0) and if the watermark bit is 1 then the reverse is true. A threshold value (K) is taken for this. K is the average difference of the U(0,0) and U(1,0) of all blocks. Experimental result shows that the difference of U(0,0) and U(1,0) is the minimum and for that reason less error is expected.

Algorithm3: Watermark bit embedding

```
If watermark bit it 0 :
    If U(0,0) < U(1,0) :
        Interchange U(0,0) and U(1,0)
        If U(0,0) – U(1,0) < K :
            U(0,0) = U(0,0) + K/2
            U(1,0) = U(1,0) – K/2
```

```
If watermark bit is 1 :
    If U(0,0) > U(1,0) :
        Interchange U(0,0) and U(1,0)
        If U(1,0) – U(0,0) < K :
            U(0,0) = U(0,0) - K/2
            U(1,0) = U(1,0) + K/2
```

The new U matrix is then multiplied with S and V to reconstruct the blocks. All the blocks are then joined to generate newLL. Finally, IDWT is done to get the watermarked image.

Algorithm4 : Embedding (COVER, WB)

Step1: Take the 1-level 2D DWT of the COVER image.

Step2: Divide LL sub-band into 2x2 non-overlapping blocks.

Step3: Apply the sequence generator algorithm to generate block sequence.

Step4: Apply SVD for each block according to sequence.

Step5: Apply bit embedding algorithm to change the U matrix.

Step6: Do the inverse SVD to reconstruct each block.

Step7: Re-join the blocks to construct newLL.

Step8: Take IDWT to get the watermarked image.

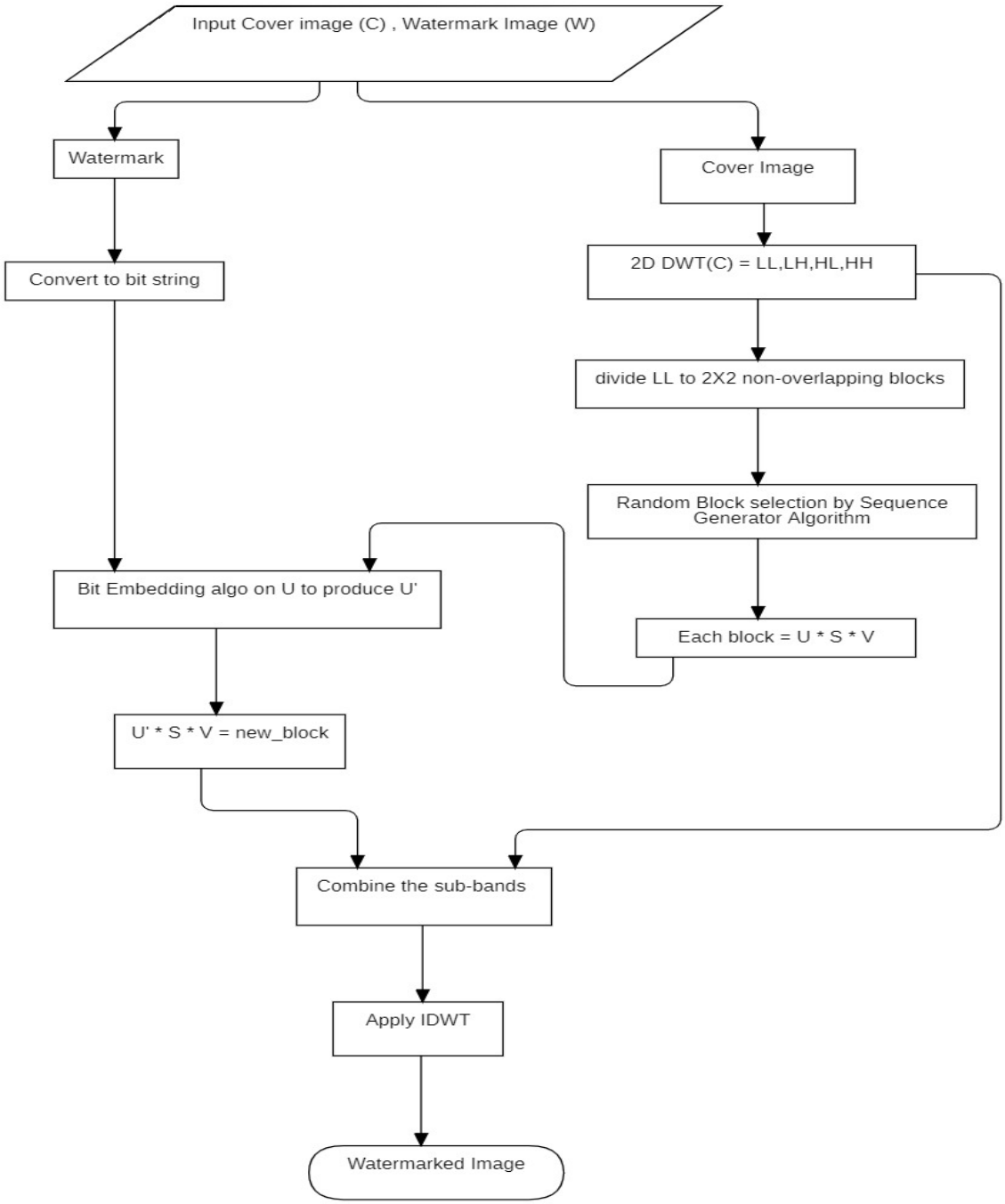


Fig: 6 Flowchart for embedding process of RDWTSVDUMAUTH

3.2.2 Extraction Process

LL sub-band is taken for extraction after taking the DWT of noisy watermarked image. The LL is then divided into 2x2 non-overlapping blocks. Sequence generation algorithm is used to generate block selection sequence. According to the sequence, each block is decomposed using

SVD. The watermark bit extraction algorithm is applied to extract the bit from U matrix of each block.

Algorithm 5: Watermark bit extraction

If U(0,0) < U(1,0):

    Watermark bit is 1

Else:

    Watermark bit is 0

After generating all the watermark bits, they are added to generate the binary bit string. Bit string is then converted into the watermark format.

Algorithm 6: Extraction (SI, WSIZE)

Step1: Take DWT of the watermarked image.

Step2: Take the LL sub-band and divide it into 2x2 non-overlapping blocks.

Step3: Apply the sequence generator algorithm to generate the block selection sequence.

Step4: According to the sequence decompose each block using SVD.

Step5: Take the U matrix and apply the watermark bit extraction algorithm to extract watermark bits.

Step6: Merge all bits to generate watermark bit string.

Step7. Convert the binary string into watermark format.

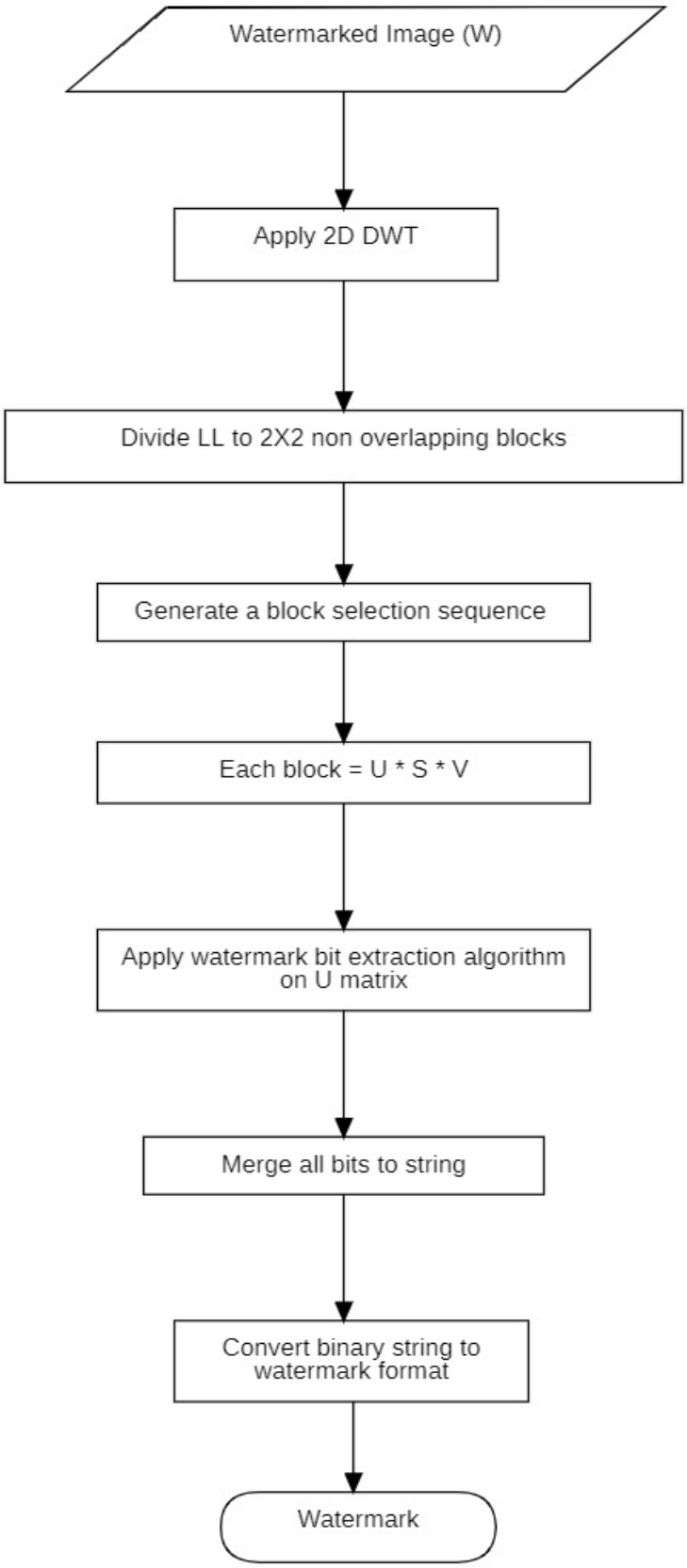


Fig: 7 Flowchart for extraction process of RDWTSVDUMAUTH

3.3 Robust Error correction-based Image authentication using U matrix of Block based SVD (RESVDUMAUTH)

The first method gives good PSNR value and very good robustness under a certain number of attacks. But fails to give satisfactory result under some attacks like JPEG compression, Histogram Equalization etc. And the second method gives very good NCC score under various kind of attack but fails to give satisfactory PSNR. So, another method is thought of to gain robustness and imperceptibility. This method is based on block-based SVD in spatial domain. Here the U matrix of SVD is selected for embedding. Without embedding actual watermark pixels into the cover image, the difference of 2 values of U matrix is taken as a threshold parameter to determine a single binary bit of watermark. Hamming code is used as Error Correcting Code for more accuracy.

3.3.1 Embedding Process

Watermark image is converted into 8-bit binary string. Hamming code is applied for every 8 bits of the watermark. The cover image is first divided into 2x2 non-overlapping blocks and SVD of each block is taken. Sequence generation algorithm is applied for block selection. Two values of position (0,0) and (1,0) of U matrix of each block is checked for one single bit embedding. If the watermark bit is 0 then U(0,0) should be greater than U(1,0) and if the watermark bit is 1 then the reverse is true. A threshold value (K) is taken for this. K is the average difference of the U(0,0) and U(1,0) of all blocks. Experimental result shows that the difference of U(0,0) and U(1,0) is the minimum and for that reason less error is expected.

Algorithm 7: Watermark bit embedding

```
If watermark bit it 0 :
    If U(0,0) < U(1,0) :
        Interchange U(0,0) and U(1,0)
        If U(0,0) – U(1,0) < K :
            U(0,0) = U(0,0) + K/2
            U(1,0) = U(1,0) – K/2
```

```
If watermark bit is 1 :
    If U(0,0) > U(1,0) :
        Interchange U(0,0) and U(1,0)
        If U(1,0) – U(0,0) < K :
            U(0,0) = U(0,0) - K/2
            U(1,0) = U(1,0) + K/2
```

The new U matrix is then multiplied with S and V to reconstruct the blocks. All the blocks are then joined to generate the watermarked image.

Algorithm 8: Embedding (COVER, WB)

Step1: Apply hamming code for every 8 bits of watermark.

Step2: Divide the cover image in spatial domain into 2x2 non-overlapping blocks.

Step3: Apply the sequence generator algorithm to generate block sequence.

Step4: Apply SVD for each block according to sequence.

Step5: Apply bit embedding algorithm to change the U matrix.

Step6: Do the inverse SVD to reconstruct each block.

Step7: Re-join the blocks to reconstruct the watermarked image.

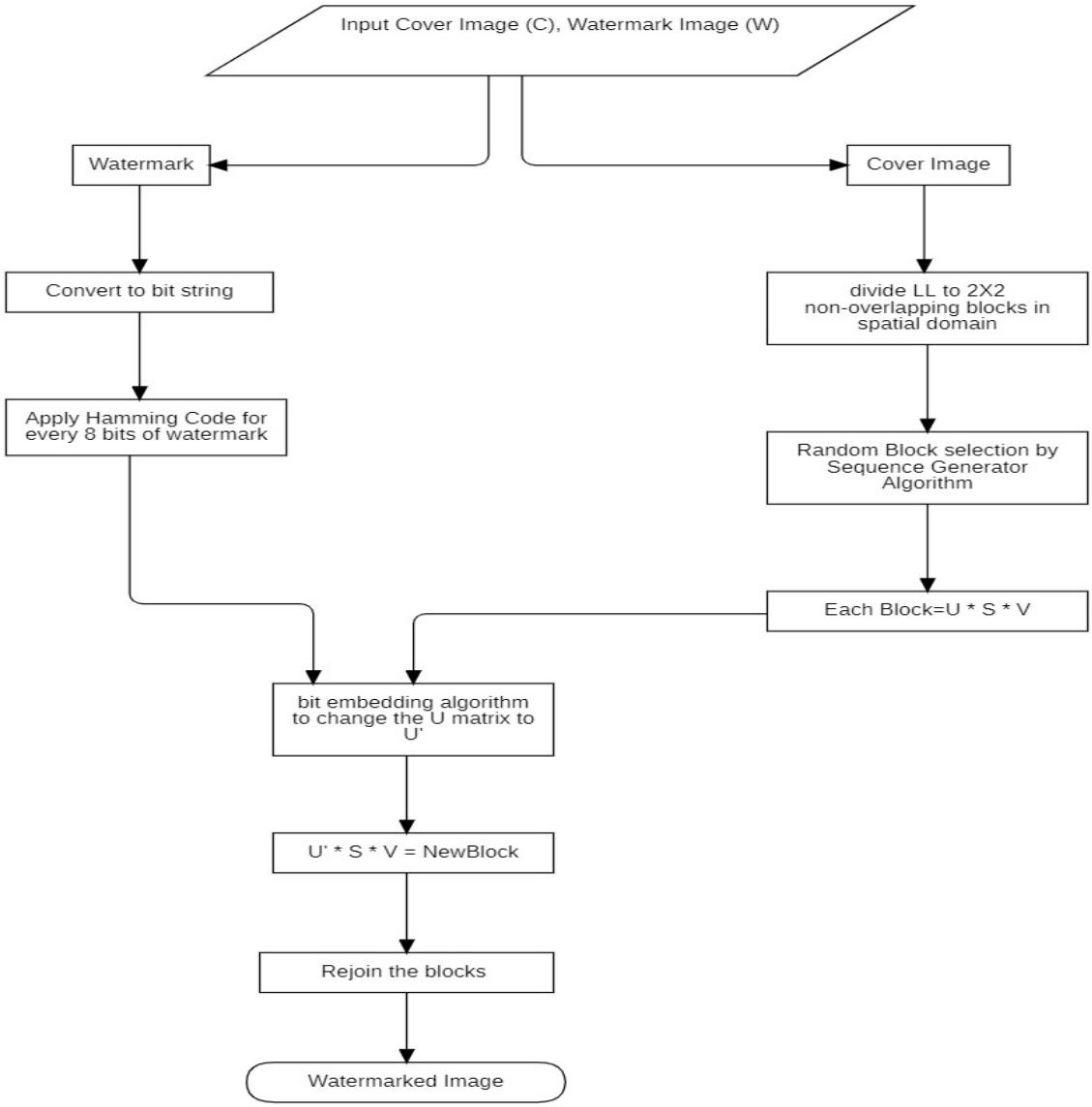


Fig: 8 Flowchart for embedding process of RESVDUMAUTH

3.3.2 Extraction Process

Noisy watermarked image is taken in spatial domain for extraction. The image is then divided into 2x2 non-overlapping blocks. Sequence generation algorithm is used to generate block selection sequence. According to the sequence, each block is decomposed using SVD. The watermark bit extraction algorithm is applied to extract the bit from U matrix of each block.

Algorithm 9: Watermark bit extraction

If U (0,0) < U (1,0):
    Watermark bit is 1
Else:
    Watermark bit is 0

After generating all the watermark bits, they are added to generate the binary bit string. After that Hamming Code error correction is done and corrected bit string is generated. Bit string is then converted into the watermark format.

Algorithm 10: Extraction (SI, WSIZE)

Step1: Take the watermarked image in spatial domain and divide it into 2x2 non-overlapping blocks.

Step2: Apply the sequence generator algorithm to generate the block selection sequence.

Step3: According to the sequence decompose each block using SVD.

Step4: Take the U matrix and apply the watermark bit extraction algorithm to extract watermark bits.

Step5: Merge all bits to generate watermark bit string.

Step6: Apply hamming code error correction to correct any single bit error.

Step7. Convert the corrected binary string into watermark format.

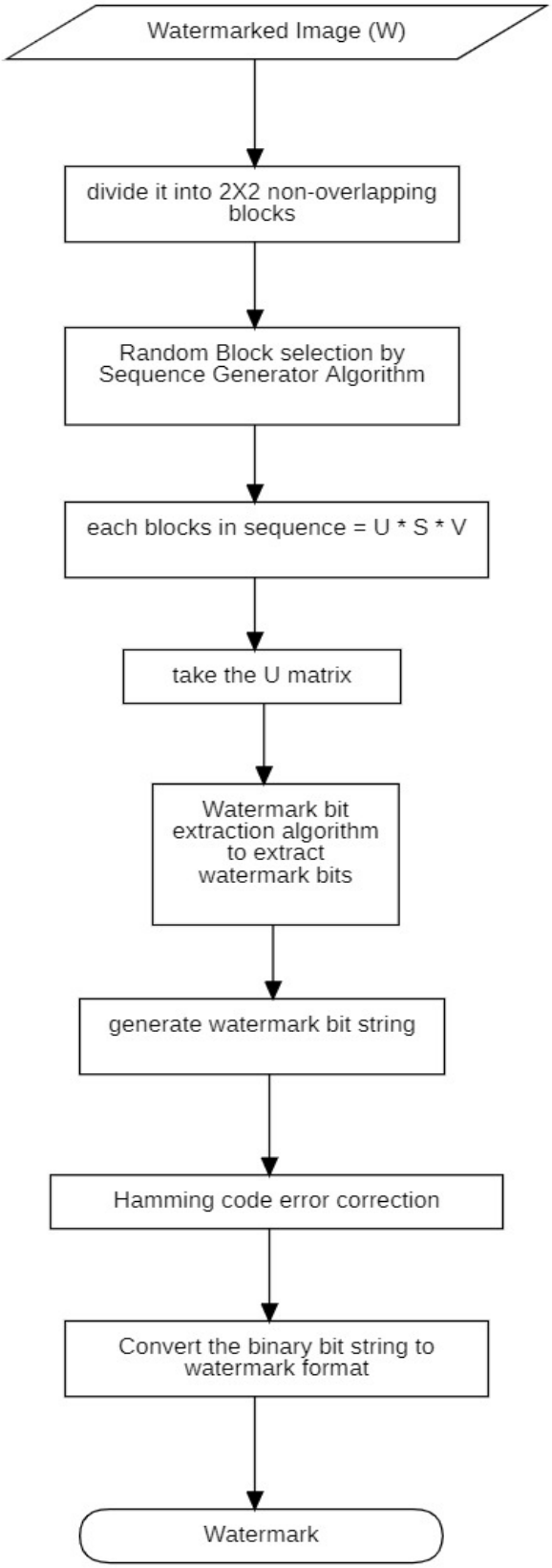


Fig: 9 Flowchart for the extraction process of RESVDUMAUTH

Chapter 4: Experimental Results and Analysis

4.1 Dataset:

The 'USC SIPI Image Dataset', 'Bossbase Dataset', "Medical Image dataset" is taken into consideration in this work. In the RESVDUMAUTH method we only use medical image. All images in the USC SIPI database are currently stored in TIFF format.[42] The BOSSBase 1.01 dataset contains 10,000 grayscale images from BOSS competition.[43] The medical image is taken from Kaggle. They consist of the middle slice of all CT images taken where valid age, modality, and contrast tags could be found. This results in 475 series from 69 different patients.[44]

4.1.1 USC SIPI Image dataset:

Lena Image

Airplane Image

Barbara Image

Fig: 10 USC SIPI Image dataset

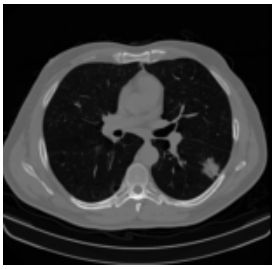
4.1.2 BOSSbase Image dataset:

(a)

(b)

Fig: 11 BOSSbase Image dataset

4.1.3 Medical Image dataset:



(a)

(b)

Fig: 12 Medical Image dataset

4.2 Experimental results of Robust DWT based Image authentication using S matrix of Block based SVD (RDWTSVDSVAUTH)

This method is experimented using BOSSbase dataset and SIPI dataset, cover image of 512x512 pixels, payload of 128x128 bytes. Without any kind of active attack on image, watermark is successfully recovered without any error. Average PSNR of 41.45 dB is achieved. An average NCC of 0.92 is achieved after applying salt & pepper noise of 0.001 density. Very good results are achieved after applying cropping, gaussian noise and salt & pepper noise. But the method fails to give good robustness against jpeg compression, histogram equalization and median filter.

Table 1: NCC & PSNR value of extracted watermark image under salt & pepper noise of 0.001 density

Cover Image	Watermark Image	NCC	PSNR
1063.pgm	Goldhill.tif	0.951125	54.29117
1082.pgm	house.tiff	0.906952	45.13591
1058.pgm	lena.tiff	0.939768	48.76279
1062.pgm	baboon.tiff	0.911922	48.19379

Table 2: NCC & PSNR value of extracted watermark image under gaussian noise with variance 0.001

Cover Image	Watermark Image	NCC	PSNR
1063.pgm	Goldhill.tif	0.819854	54.29117
1082.pgm	house.tiff	0.823249	45.13591

1058.pgm	lena.tiff	0.868613	48.76279
1062.pgm	baboon.tiff	0.82369	48.19379

Table 3: NCC & PSNR value of extracted watermark image under JPEG compression

Cover Image	Watermark Image	NCC	PSNR
1011.pgm	Goldhill.tif	0.033205	36.24174
1013.pgm	house.tiff	0.073048	41.66139
10.pgm	lena.tiff	0.024879	38.40452
1007.pgm	baboon.tiff	0.013044	33.30988

(a) (b) (c) (d)

Fig: 13 Watermarking results after applying Gaussian Noise, (a) Cover Image, (b) Watermark image, (c) Watermarked image after attack, (d) Retrieved Watermark

(a) (b) (c) (d)

Fig: 14 Watermarking results after JPEG Compression, (a) Cover Image, (b) Watermark image, (c) Watermarked image after attack, (d) Retrieved Watermark

Additional data:

Table 4: NCC (Between watermark & retrieve watermark) & PSNR(Cover image & watermarked image) of extracted watermark image under different attacks of RDWTSVDSVAUTH

		Histogram		Salt & pepper		JPEG	
Image	Watermark	NCC	PSNR	NCC	PSNR	NCC	PSNR
1.pgm	airplane.tiff	0.015882	40.17271	0.891907	40.17271	0.063776	40.17271
10.pgm	airplane.tiff	0.016827	38.39715	0.885594	38.39715	0.018809	38.39715
100.pgm	airplane.tiff	0.013157	41.04342	0.885594	41.04342	0.031847	41.04342
1000.pgm	airplane.tiff	0.028317	38.44203	0.88828	38.44203	0.04012	38.44203
10000.pgm	airplane.tiff	0.027006	41.00375	0.886001	41.00375	-0.01378	41.00375
1001.pgm	airplane.tiff	0.044592	46.24129	0.885594	46.24129	0.007992	46.24129
1002.pgm	airplane.tiff	-0.01162	35.32616	0.885594	35.32616	0.000254	35.32616
1003.pgm	airplane.tiff	0.011614	39.5588	0.885797	39.5588	0.010158	39.5588
1004.pgm	airplane.tiff	-0.00055	42.81206	0.887355	42.81206	-0.01687	42.81206
1005.pgm	airplane.tiff	0.006161	37.94674	0.887519	37.94674	0.011895	37.94674
1006.pgm	airplane.tiff	0.009019	29.91696	0.887767	29.91696	-0.01782	29.91696
1007.pgm	airplane.tiff	0.020567	33.3077	0.885594	33.3077	0.032924	33.3077
1008.pgm	airplane.tiff	0.049291	42.28457	0.887122	42.28457	0.007194	42.28457
1009.pgm	airplane.tiff	0.024914	38.68567	0.885594	38.68567	-0.00552	38.68567
101.pgm	airplane.tiff	0.028281	34.2795	0.885594	34.2795	-0.01487	34.2795
1010.pgm	airplane.tiff	0.061385	45.24824	0.885594	45.24824	0.047915	45.24824
1011.pgm	airplane.tiff	0.023912	36.23655	0.885594	36.23655	-0.02017	36.23655
1012.pgm	airplane.tiff	0.022263	38.21045	0.885594	38.21045	0.042672	38.21045
1013.pgm	airplane.tiff	0.022042	41.65585	0.885594	41.65585	0.02226	41.65585
1014.pgm	airplane.tiff	0.029117	42.40153	0.885594	42.40153	0.016141	42.40153
1.pgm	baboon.tiff	0.010981	40.18334	0.885594	42.21758	0.038249	40.18334
10.pgm	baboon.tiff	0.023609	38.40421	0.885594	51.28375	0.014796	38.40421
100.pgm	baboon.tiff	-0.01496	41.05641	0.885594	50.40162	0.014602	41.05641
1000.pgm	baboon.tiff	0.024422	38.44917	0.887235	49.61587	0.010084	38.44917
10000.pgm	baboon.tiff	-0.00037	41.01662	0.885594	38.62503	-0.01697	41.01662
1001.pgm	baboon.tiff	0.046862	46.28444	0.885594	38.42538	-0.00291	46.28444
1002.pgm	baboon.tiff	-0.00952	35.32964	0.885594	37.11187	0.029333	35.32964
1003.pgm	baboon.tiff	0.019913	39.56803	0.885594	35.68438	0.018601	39.56803
1004.pgm	baboon.tiff	-0.00838	42.8316	0.885594	39.67773	-0.02651	42.8316
1005.pgm	baboon.tiff	-0.0032	37.95311	0.885594	42.56705	-0.00836	37.95311
1006.pgm	baboon.tiff	0.007876	29.91796	0.885594	48.05044	-0.01585	29.91796
1007.pgm	baboon.tiff	0.011465	33.30988	0.885594	47.70736	0.013044	33.30988
1008.pgm	baboon.tiff	0.038014	42.30187	0.885594	37.07155	0.003255	42.30187
1009.pgm	baboon.tiff	-0.01024	38.69321	0.887963	42.36868	-0.0128	38.69321
101.pgm	baboon.tiff	0.005112	34.28224	0.885594	41.00064	-0.00821	34.28224
1010.pgm	baboon.tiff	0.066392	45.28254	0.888288	41.79528	0.046655	45.28254
1011.pgm	baboon.tiff	0.027142	36.24084	0.885594	41.61656	-0.01504	36.24084
1012.pgm	baboon.tiff	0.013175	38.21722	0.885594	42.01208	0.036015	38.21722
1013.pgm	baboon.tiff	0.010438	41.67082	0.885594	48.54092	-0.01003	41.67082

1014.pgm	baboon.tiff	0.031649	42.4193	0.88706	30.81539	0.011249	42.4193
1.pgm	lena.tiff	0.006713	40.18381	0.887039	38.19264	-0.03504	40.18381
10.pgm	lena.tiff	0.037995	38.40452	0.889341	36.53875	0.024879	38.40452
100.pgm	lena.tiff	-0.01839	41.05698	0.886627	44.96929	-0.03113	41.05698
1000.pgm	lena.tiff	0.001189	38.44948	0.885594	39.72127	-0.04619	38.44948
10000.pgm	lena.tiff	0.014742	41.01719	0.885594	51.46431	-0.00421	41.01719
1001.pgm	lena.tiff	-0.02671	46.28634	0.885594	51.7479	0.01346	46.28634
1002.pgm	lena.tiff	0.014398	35.32979	0.8861	38.68186	-0.00655	35.32979
1003.pgm	lena.tiff	-0.02412	39.56843	0.885594	39.44365	-0.03124	39.56843
1004.pgm	lena.tiff	0.018694	42.83246	0.885594	36.83819	0.06102	42.83246
1005.pgm	lena.tiff	0.004882	37.95339	0.885594	30.70614	-0.01511	37.95339
1006.pgm	lena.tiff	0.006558	29.918	0.885594	50.35939	-0.00078	29.918
1007.pgm	lena.tiff	-0.0392	33.30998	0.885594	39.69904	-0.03802	33.30998
1008.pgm	lena.tiff	0.036659	42.30263	0.885594	47.34149	-0.00428	42.30263
1009.pgm	lena.tiff	0.058562	38.69354	0.885594	47.89839	7.27E-05	38.69354
101.pgm	lena.tiff	0.059679	34.28236	0.885557	46.00191	0.008202	34.28236
1010.pgm	lena.tiff	0.015836	45.28405	0.885594	41.66785	-0.00148	45.28405
1011.pgm	lena.tiff	0.007941	36.24103	0.885325	41.60537	0.009855	36.24103
1012.pgm	lena.tiff	-0.01585	38.21751	0.885594	47.63057	-0.00102	38.21751
1013.pgm	lena.tiff	0.005873	41.67148	0.886006	42.88607	-0.02295	41.67148
1014.pgm	lena.tiff	0.06011	42.42008	0.885594	34.32523	-0.02248	42.42008
1.pgm	Goldhill.tif	-0.00495	40.18558	0.885594	45.63266	-0.21273	40.18558

		Crop20		Crop40		Gaussian Noise		Median Filter	
Image	Watermark	NCC	PSNR	NCC	PSNR	NCC	PSNR	NCC	PSNR
1.pgm	airplane.tiff	0.996386	40.17271	0.956939	40.17271	0.846136	40.17271	0.05841	40.17271
10.pgm	airplane.tiff	0.996386	38.39715	0.959361	38.39715	0.850576	38.39715	0.019804	38.39715
100.pgm	airplane.tiff	0.996386	41.04342	0.958236	41.04342	0.862751	41.04342	0.043258	41.04342
1000.pgm	airplane.tiff	0.996386	38.44203	0.962269	38.44203	0.848391	38.44203	0.041895	38.44203
10000.pgm	airplane.tiff	0.996386	41.00375	0.957371	41.00375	0.8527	41.00375	-0.00676	41.00375
1001.pgm	airplane.tiff	0.996386	46.24129	0.958684	46.24129	0.850074	46.24129	0.027297	46.24129
1002.pgm	airplane.tiff	0.996386	35.32616	0.957703	35.32616	0.873666	35.32616	0.007909	35.32616
1003.pgm	airplane.tiff	0.996386	39.5588	0.957506	39.5588	0.846528	39.5588	0.016608	39.5588
1004.pgm	airplane.tiff	0.996386	42.81206	0.956422	42.81206	0.848063	42.81206	-0.01685	42.81206
1005.pgm	airplane.tiff	0.996386	37.94674	0.960036	37.94674	0.847829	37.94674	0.015801	37.94674
1006.pgm	airplane.tiff	0.996814	29.91696	0.960836	29.91696	0.850581	29.91696	-0.00875	29.91696
1007.pgm	airplane.tiff	0.996386	33.3077	0.955794	33.3077	0.853242	33.3077	0.042404	33.3077
1008.pgm	airplane.tiff	0.996386	42.28457	0.959182	42.28457	0.85113	42.28457	0.013744	42.28457
1009.pgm	airplane.tiff	0.996386	38.68567	0.957285	38.68567	0.847732	38.68567	-0.0154	38.68567
101.pgm	airplane.tiff	0.996386	34.2795	0.958524	34.2795	0.860189	34.2795	-0.0175	34.2795
1010.pgm	airplane.tiff	0.996386	45.24824	0.96083	45.24824	0.866792	45.24824	0.057293	45.24824
1011.pgm	airplane.tiff	0.996386	36.23655	0.958486	36.23655	0.849163	36.23655	-0.00331	36.23655
1012.pgm	airplane.tiff	0.996386	38.21045	0.956276	38.21045	0.866632	38.21045	0.053805	38.21045
1013.pgm	airplane.tiff	0.996386	41.65585	0.959177	41.65585	0.871189	41.65585	0.023359	41.65585
1014.pgm	airplane.tiff	0.996386	42.40153	0.958928	42.40153	0.861832	42.40153	0.010506	42.40153

4.3 Experimental results of Robust DWT based Image authentication using U matrix of Block based SVD (RDWTSVDUMAUTH)

This method is experimented using BOSSbase dataset and SIPI dataset, cover image of 512x512 pixels, payload of 8,192 bits. Without any kind of active attack on image, watermark is successfully recovered without any error. Average PSNR of 27.23 dB is achieved, which is low comparing to the previous method. An average NCC of 0.97 is achieved after applying JPEG compression, which is a great success comparing to the previous method. Very good results are achieved after applying cropping, gaussian noise, salt & pepper noise, jpeg compression, histogram equalization and median filter.

Table 5: NCC value of extracted watermark image under JPEG compression

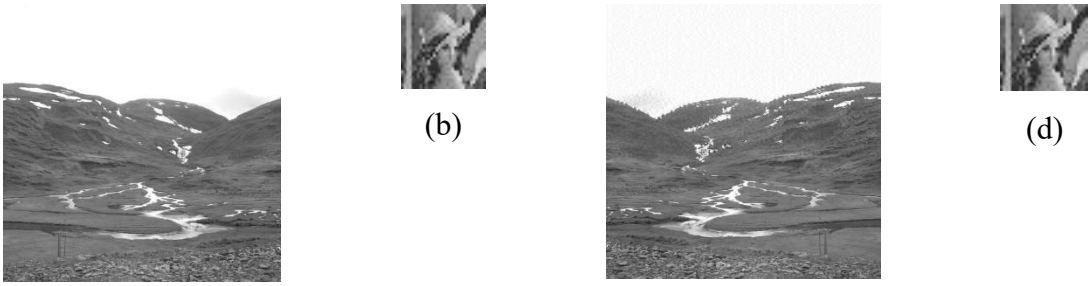
Cover Image	Watermark Image	NCC	PSNR
10.pgm	lena.tiff	0.972377	23.40228
1005.pgm	baboon.tiff	1	27.86258
1001.pgm	airplane.tiff	0.858322	30.18773
1.pgm	airplane.tiff	1	30.61746

Table 6: NCC value of extracted watermark image under histogram equalization

Cover Image	Watermark Image	NCC	PSNR
10.pgm	lena.tiff	0.972377	23.40228
1005.pgm	baboon.tiff	1	27.86258
1001.pgm	airplane.tiff	0.858322	30.18773
1.pgm	airplane.tiff	1	30.61746

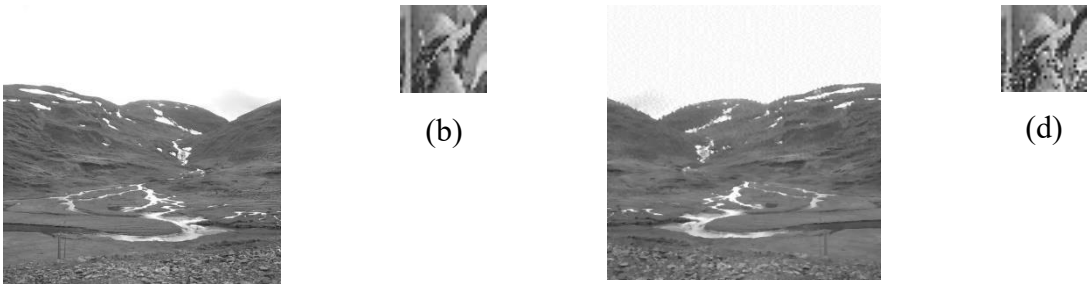
Table 7: NCC value of extracted watermark image under median filter

Cover Image	Watermark Image	NCC	PSNR
10.pgm	lena.tiff	0.635886	23.40228
1005.pgm	baboon.tiff	0.761992	27.86258
1001.pgm	airplane.tiff	0.649695	30.18773
1.pgm	airplane.tiff	0.751055	30.61746



(a) (b) (c) (d)

Fig: 15 Watermarking results after JPEG Compression, (a) Cover Image, (b) Watermark image, (c) Watermarked image after attack, (d) Retrieved Watermark



(a) (b) (c) (d)

Fig: 16 Watermarking results after applying median filter, (a) Cover Image, (b) Watermark image, (c) Watermarked image after attack, (d) Retrieved Watermark

Additional data:

Table 8: NCC (Between watermark & retrieve watermark) & PSNR (Cover image & watermarked image) of extracted watermark image under different attacks of RDWTSVDUMAUTH

		Histogram Equalization		Median Filter		Jpeg	
Image	Watermark	NCC	PSNR	NCC	PSNR	NCC	PSNR
1.pgm	airplane.tiff	0.83037076	30.6174624	0.751054936	30.61746	1	30.6174624
10.pgm	airplane.tiff	0.81311964	23.6195499	0.628869512	23.61955	0.982846	23.6195499
100.pgm	airplane.tiff	0.99976027	33.9582981	0.798521447	33.9583	1	33.9582981
1000.pgm	airplane.tiff	0.63093555	26.8189568	0.771263993	26.81896	1	26.8189568
10000.pgm	airplane.tiff	0.88116681	24.1557477	0.79871876	24.15575	0.923389	24.1557477
1001.pgm	airplane.tiff	0.90965576	30.1877301	0.649694675	30.18773	0.858322	30.1877301
1002.pgm	airplane.tiff	0.93028456	25.2519639	0.82381415	25.25196	0.984636	25.2519639
1003.pgm	airplane.tiff	0.66579946	23.2788744	0.608538049	23.27887	0.949752	23.2788744
1004.pgm	airplane.tiff	0.75551592	26.5960781	0.662143979	26.59608	0.9944	26.5960781
1005.pgm	airplane.tiff	0.91526179	27.9463834	0.682785753	27.94638	0.999996	27.9463834
1.pgm	baboon.tiff	0.83829324	30.2686897	0.803282845	30.26869	1	30.2686897

10.pgm	baboon.tiff	0.87102046	23.4586753	0.65301401	23.45868	0.98633	23.4586753
100.pgm	baboon.tiff	0.99965762	33.8697212	0.811387961	33.86972	1	33.8697212
1000.pgm	baboon.tiff	0.64354193	26.6543502	0.750993096	26.65435	1	26.6543502
10000.pgm	baboon.tiff	0.88374427	24.1944003	0.788678896	24.1944	0.943321	24.1944003
1001.pgm	baboon.tiff	0.93746521	30.456689	0.701494902	30.45669	0.890806	30.456689
1002.pgm	baboon.tiff	0.93751302	24.9977841	0.830317378	24.99778	0.996743	24.9977841
1003.pgm	baboon.tiff	0.7071146	23.3104232	0.652200721	23.31042	0.961959	23.3104232
1004.pgm	baboon.tiff	0.77550187	26.8104887	0.714463185	26.81049	0.992111	26.8104887
1005.pgm	baboon.tiff	0.9280676	27.8625809	0.761992492	27.86258	1	27.8625809
1.pgm	lena.tiff	0.85356882	30.606547	0.836870305	30.60655	1	30.606547
10.pgm	lena.tiff	0.87474657	23.4022821	0.635886446	23.40228	0.972377	23.4022821
100.pgm	lena.tiff	0.99977906	33.9455704	0.873768402	33.94557	1	33.9455704
1000.pgm	lena.tiff	0.6691457	26.801442	0.828916905	26.80144	1	26.801442
10000.pgm	lena.tiff	0.90609391	24.1943762	0.812046224	24.19438	0.950716	24.1943762
1001.pgm	lena.tiff	0.93243129	30.2678639	0.735633348	30.26786	0.875932	30.2678639
1002.pgm	lena.tiff	0.93733675	25.3973615	0.860711337	25.39736	0.986403	25.3973615
1003.pgm	lena.tiff	0.76238571	23.267389	0.686598436	23.26739	0.964701	23.267389
1004.pgm	lena.tiff	0.84130006	26.9985861	0.781758264	26.99859	0.990675	26.9985861
1005.pgm	lena.tiff	0.94765029	27.8406263	0.782319304	27.84063	0.999997	27.8406263

4.4 Experimental results of Robust Error correction-based Image authentication using U matrix of Block based SVD (RESVDUMAUTH)

This method is experimented using SIPI dataset, cover image of 512x512 pixels, payload of 12,288 bits. Without any kind of active attack on image, watermark is successfully recovered without any error. Average PSNR of 36.32 dB is achieved, which is low comparing to the previous method. And also, a PSNR of 43.82 dB is achieved after applying on medical image. An average NCC of 0.99 is achieved after applying JPEG compression, which is a great success comparing to the previous 2 methods. Very good results are achieved after applying cropping, gaussian noise, salt & pepper noise, jpeg compression, histogram equalization and median filter.

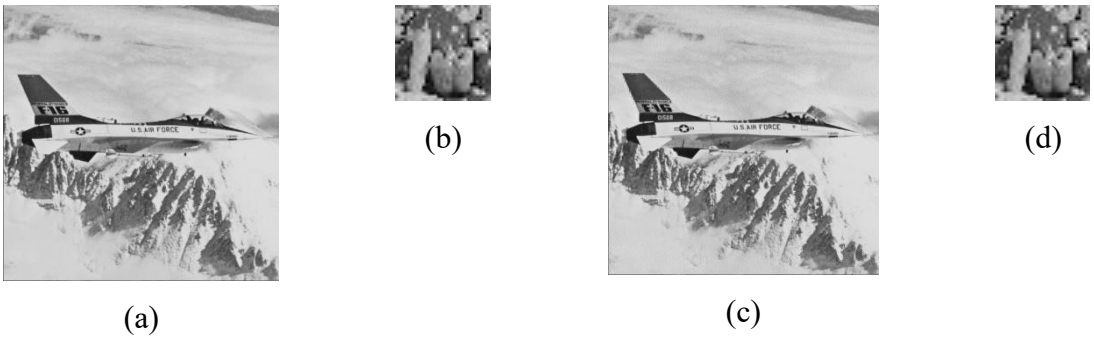
Table 9: NCC value of extracted watermark image under JPEG compression

Cover Image	Watermark Image	NCC	PSNR
airplane.tiff	peeper.tiff	1	37.52934
elaine.tiff	splash.tiff	1	36.84927
lena.tiff	peeper.tiff	1	36.56643

tank.tiff	splash.tiff	1	37.23136

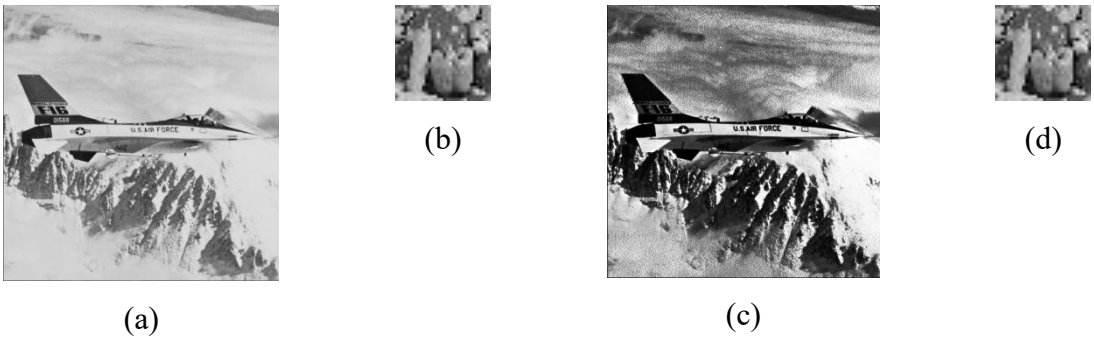
Table 10: NCC value of extracted watermark image under gaussian noise

Cover Image	Watermark Image	NCC	PSNR
airplane.tiff	peeper.tiff	1	35.31552
elaine.tiff	splash.tiff	1	35.12001
lena.tiff	peeper.tiff	1	33.41158
tank.tiff	splash.tiff	1	37.3193



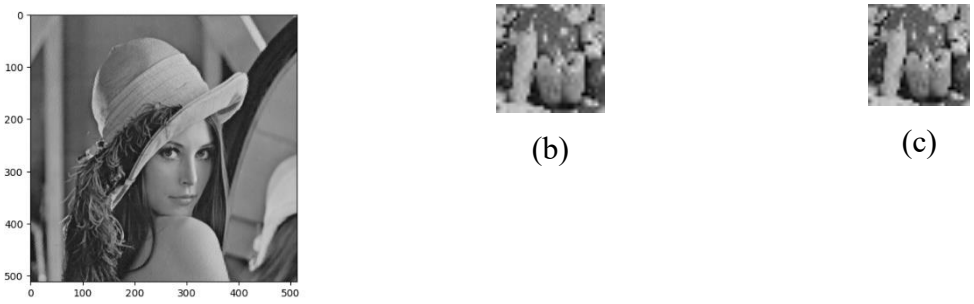
(a) (b) (c) (d)

Fig: 17 Watermarking results after JPEG Compression, (a) Cover Image, (b) Watermark image, (c) Watermarked image after attack, (d) Retrieved Watermark



(a) (b) (c) (d)

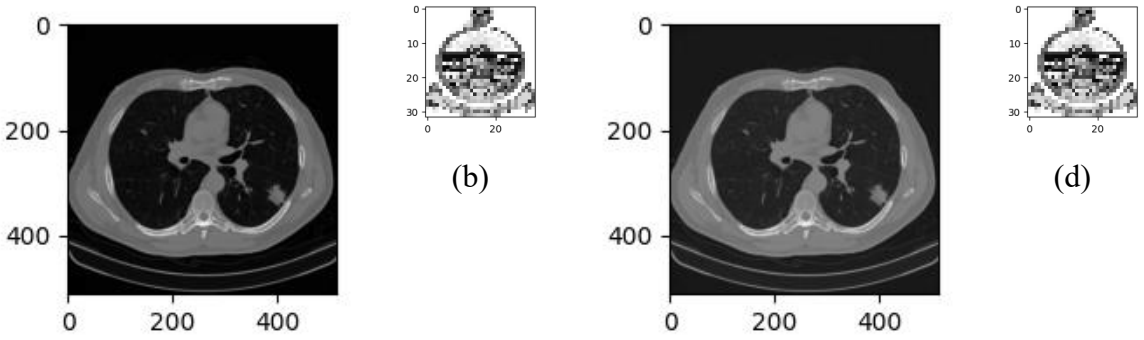
Fig: 18 Watermarking results after applying histogram equalization, (a) Cover Image, (b) Watermark image, (c) Watermarked image after attack, (d) Retrieved Watermark



(a) (b) (c)

Fig: 19 Watermarking results without performing attack on lena.tiff image (a) Cover Image, (b)Watermark image, (c) Retrieved Watermark

This method is also applied on medical images and very good results are obtained.



(a) (b) (c) (d)

Fig: 20 Watermarking results on medical image, (a) Cover Image, (b) Watermark image, (c) Watermarked image after watermarking, (d) Retrieved Watermark

EPR Data:

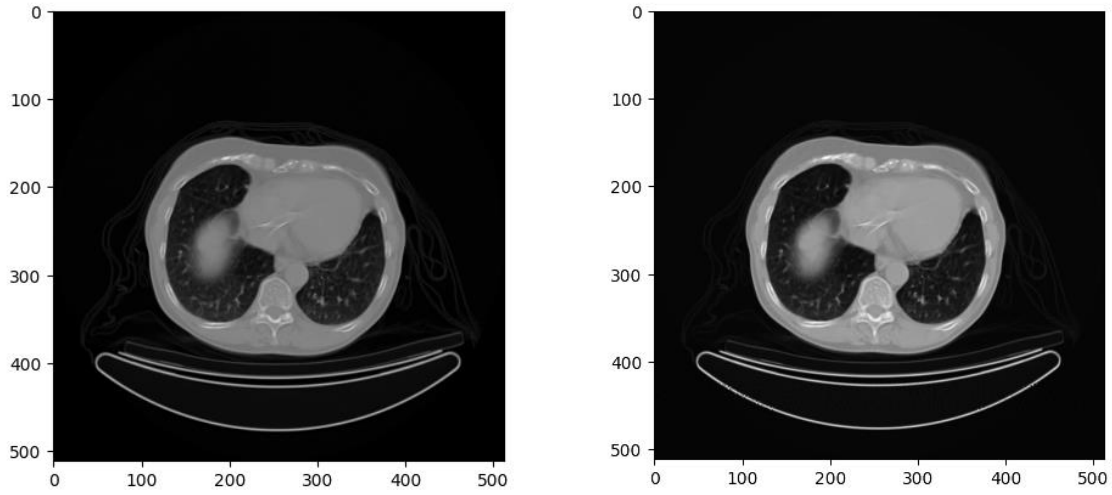
Doctor: Jojo

Hospital Code: ABC_HOSPITAL_64

Image: Brain Cancer

Patient Name: Temp Roy

Copyright: Swarnadeep



(a) (b)

Fig: 21 Watermarking on medical image with EPR data, 49.82 dB PSNR is achieved (a)Cover Image, (b) Watermarked Image

Retrieved EPR Data:

Doctor: Jojo

Hospital Code: ABC_HOSPITAL_64

Image: Brain Cancer

Patient Name: Temp Roy

Copyright: Swarnadeep

Additional data:

Table 10: NCC (Between watermark & retrieve watermark) & PSNR (Cover image & watermarked image) of extracted watermark image under different attacks of RESVDUMAUTH

	Crop50		Crop100	
Image	PSNR	NCC	PSNR	NCC
watermarkairplane_peeper.tiff	37.99056	1	21.43568	1
watermarkairplane_splash.tiff	37.97614	1	21.44146	1
watermarkbaboon_peeper.tiff	35.40472	1	20.3968	1
watermarkbaboon_splash.tiff	35.4035	1	20.40264	1
watermarkBarbara_peeper.tiff	36.16829	1	17.7207	1
watermarkBarbara_splash.tiff	36.18242	1	17.72722	1

watermarkelaine_peeper.tiff	37.55276	1	19.39858	1
watermarkelaine_splash.tiff	37.56024	1	19.41067	1
watermarkfishingboat_peeper.tiff	36.42763	1	20.12748	1
watermarkfishingboat_splash.tiff	36.44493	1	20.13557	1
watermarkGoldhill_peeper.tiff	36.96812	1	17.79599	1
watermarkGoldhill_splash.tiff	36.86313	1	17.79504	1
watermarkhouse_peeper.tiff	35.8008	1	19.5978	1
watermarkhouse_splash.tiff	35.78346	1	19.58375	1
watermarklena_peeper.tiff	37.04838	1	19.68738	1
watermarklena_splash.tiff	37.06979	1	19.67863	1
watermarksailboat_peeper.tiff	35.23629	1	18.87648	1
watermarksailboat_splash.tiff	35.24478	1	18.88092	1
watermarktank_peeper.tiff	38.02642	1	17.79049	1
watermarktank_splash.tiff	37.91963	1	17.78984	1

	Gaussian		Jpeg	
Image	PSNR	NCC	PSNR	NCC
watermarkairplane_peeper.tiff	35.31552	1	37.52934	1
watermarkairplane_splash.tiff	35.45912	1	37.54439	1
watermarkbaboon_peeper.tiff	28.05285	1	35.11527	0.999787
watermarkbaboon_splash.tiff	28.08701	1	35.10521	0.999094
watermarkBarbara_peeper.tiff	31.23567	1	35.88418	1
watermarkBarbara_splash.tiff	31.38434	1	35.89974	1
watermarkelaine_peeper.tiff	34.69104	1	36.8496	1
watermarkelaine_splash.tiff	35.12001	1	36.84927	1
watermarkfishingboat_peeper.tiff	32.25995	1	35.92619	1
watermarkfishingboat_splash.tiff	32.39411	1	35.96658	1
watermarkGoldhill_peeper.tiff	35.64388	1	36.39102	1
watermarkGoldhill_splash.tiff	35.58662	1	36.31273	1
watermarkhouse_peeper.tiff	31.1987	1	35.5023	1
watermarkhouse_splash.tiff	30.99993	1	35.50763	1
watermarklena_peeper.tiff	33.41158	1	36.56643	1
watermarklena_splash.tiff	33.20987	1	36.60792	1
watermarksailboat_peeper.tiff	30.58886	1	35.10067	1
watermarksailboat_splash.tiff	30.65511	1	35.1189	1
watermarktank_peeper.tiff	37.37799	1	37.29446	1
watermarktank_splash.tiff	37.3193	1	37.23136	1

	Median Filter		Salt & Pepper (0.005)		Salt & Pepper (0.001)	
Image	PSNR	NCC	PSNR	NCC	PSNR	NCC
watermarkairplane_peeper.tiff	32.62	0.464004	27.09761	0.984752971	32.06609	1
watermarkairplane_splash.tiff	32.60651	0.405842	27.1194	0.989206366	32.13214	1
watermarkbaboon_peeper.tiff	22.83962	0.265403	25.33639	0.993604154	27.33872	1
watermarkbaboon_splash.tiff	22.83362	0.271059	25.35351	0.996211086	27.36912	1
watermarkBarbara_peeper.tiff	24.33615	0.266189	26.56304	0.987957427	29.8341	1
watermarkBarbara_splash.tiff	24.35122	0.302912	26.61147	0.996036843	29.94038	1
watermarkelaine_peeper.tiff	32.45518	0.22805	27.62302	0.987605139	32.22771	1
watermarkelaine_splash.tiff	32.50296	0.197506	27.70332	0.989209046	32.46687	1
watermarkfishingboat_peeper.tiff	28.96733	0.330257	26.97577	0.987200587	30.50723	1
watermarkfishingboat_splash.tiff	28.98698	0.397886	27.01247	0.989206674	30.59645	1
watermarkGoldhill_peeper.tiff	30.78777	0.45971	27.72725	0.990458929	32.48406	1
watermarkGoldhill_splash.tiff	30.77895	0.316985	27.71815	0.989215348	32.45611	1
watermarkhouse_peeper.tiff	29.11287	0.387726	26.40249	0.987615822	29.70536	1
watermarkhouse_splash.tiff	28.96615	0.362355	26.34024	0.989206436	29.56398	1
watermarklena_peeper.tiff	32.96806	0.294822	27.24356	0.987609117	31.36335	1
watermarklena_splash.tiff	32.92331	0.26476	27.19462	0.98920936	31.23741	1
watermarksailboat_peeper.tiff	28.91278	0.315131	26.13934	0.99045857	29.16642	1
watermarksailboat_splash.tiff	28.88706	0.304669	26.1626	0.98921445	29.21355	1
watermarktank_peeper.tiff	31.79799	0.284178	28.41311	0.987612542	33.5939	1
watermarktank_splash.tiff	31.79991	0.307169	28.40487	0.989209046	33.56949	1

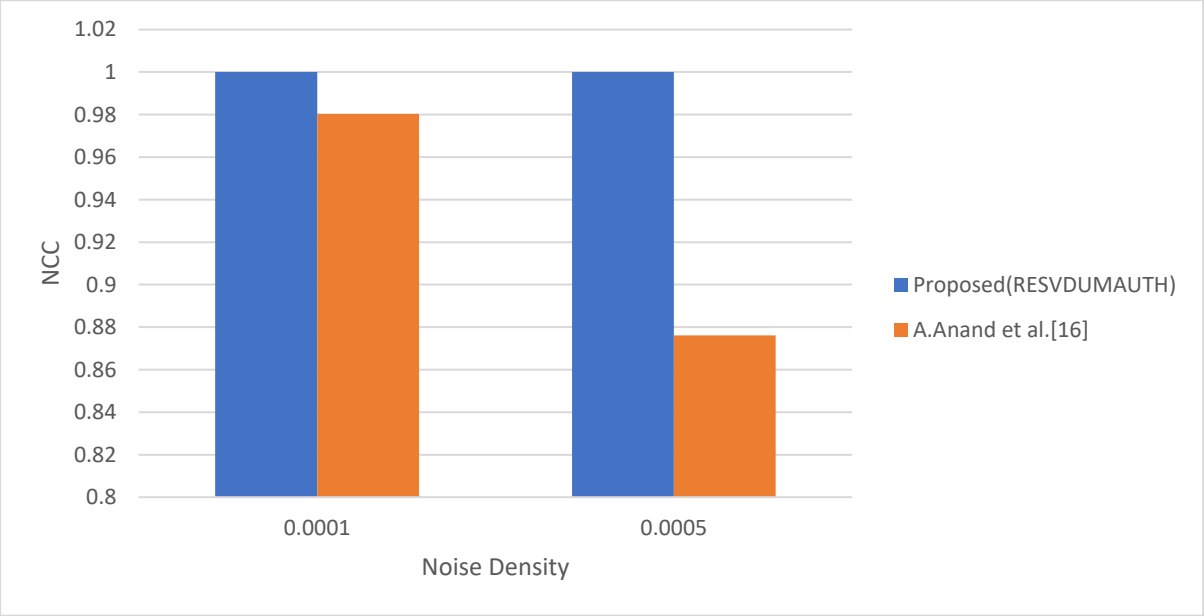


Fig: 22 Average NCC value comparison between proposed and existing scheme [16] for watermark image using medical image under gaussian noise attack

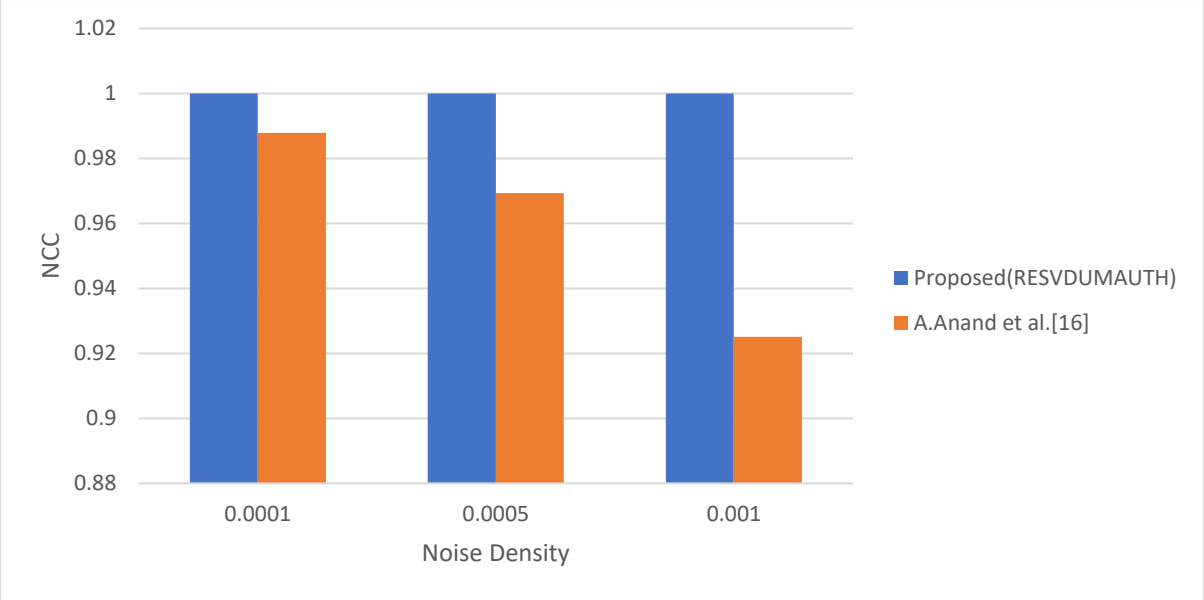


Fig: 23 Average NCC value comparison between proposed and existing scheme [16] for watermark image using medical image under salt and pepper noise attack

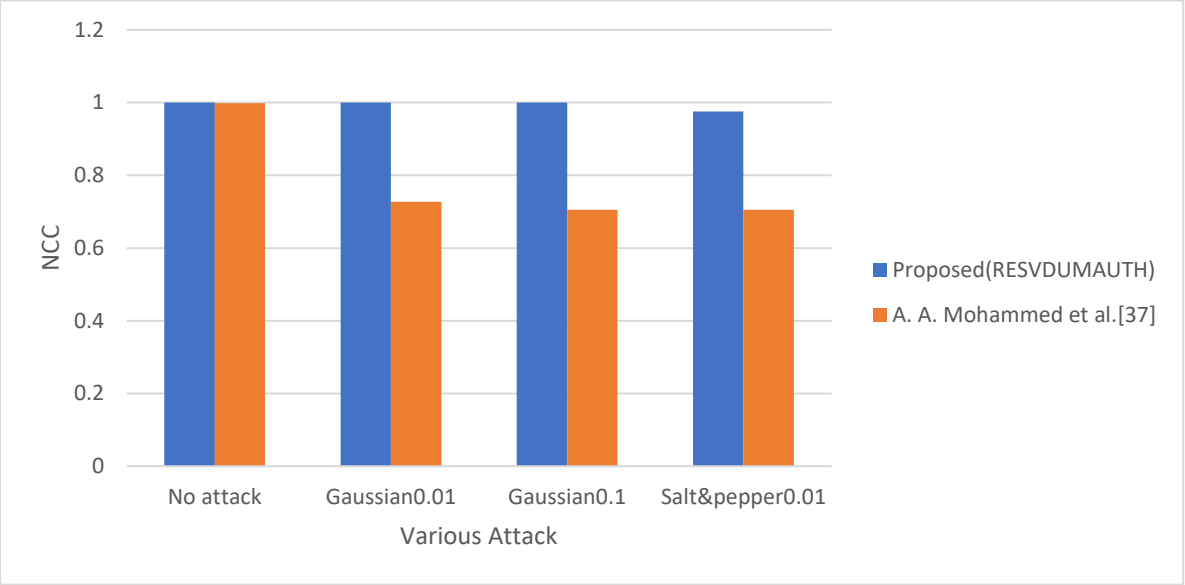


Fig: 24 NCC value comparison between proposed and existing scheme [35] for watermark image using Brain CT medical image under various active attack

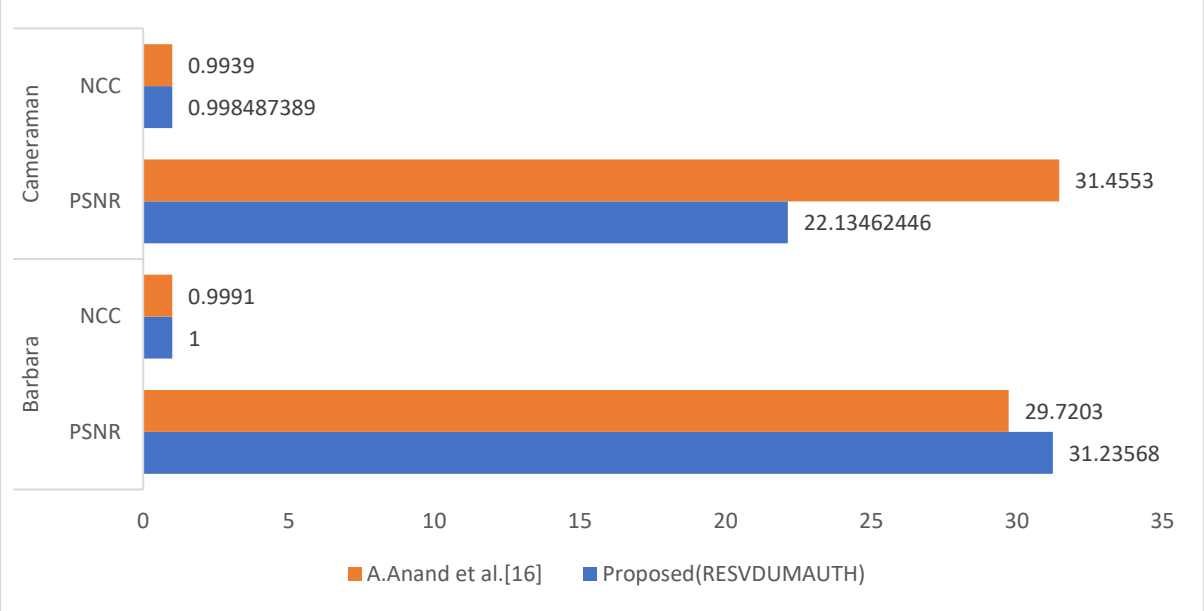


Fig: 25 NCC and PSNR value comparison between proposed and existing scheme [16] for watermark image using USC SIPI image with no attack

4.4.1 Steganalysis

To measure the effectiveness of steganography steganalysis is done using grey-level co-occurrence matrix and neural network [17]. First 2,000 images have been generated using the method. Then grey level co-occurrence matrix (GLCM) has been calculated for each image. PCA is applied on every GLCM and 200 features have been extracted. A neural network is created which takes input of the PCA features and gives output as 0 or 1. 0 indicates non-stego image and 1 indicates stego-image. The model has been trained on those PCA features of 2,000 stego images and 2,000 non-stego images. Then the model has been tested using our test data. In the steganalysis model we've got the FPR of 0.25% of detecting stego image. This clarifies that our steganography method is powerful.

	Original Stego	Original Non-stego
Predicted Stego	404	396
Predicted Non-stego	0	0

Table: 11 Confusion Matrix of steganalysis model

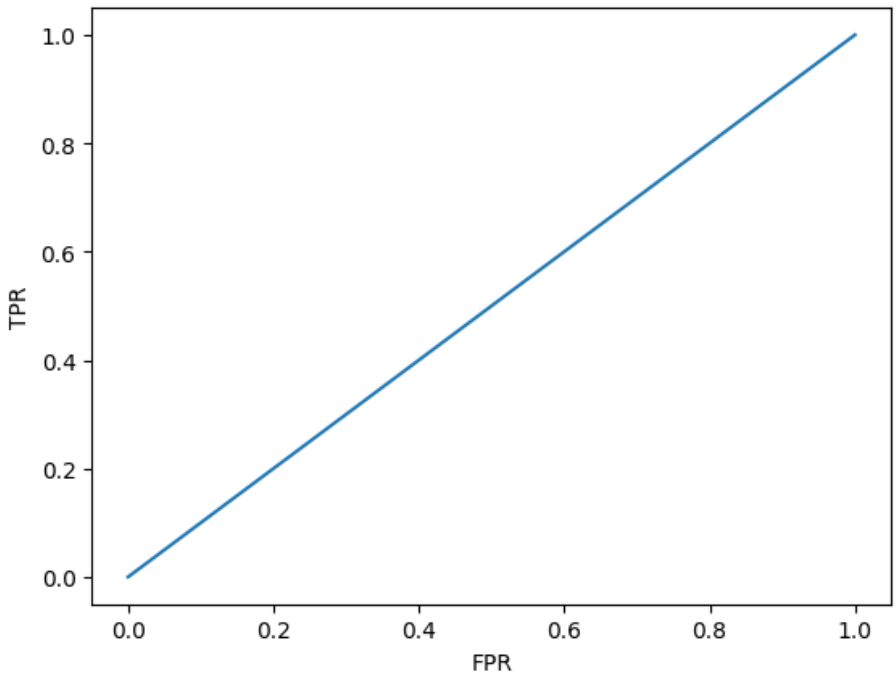


Fig: 26 ROC Curve of Steganalysis network

Table 12 Comparison of proposed method with other algorithms' performance

Sl. No.	Article	Techniques	Encryption	Domain	Dataset	PSNR (dB) and NCC	Security Analysis
1	A. Anand et al. [16]	DWT, SVD, Hamming code	Chaotic or hyperchaotic	Transform	512x512 medical images, USC SIPI	36.1007 dB NCC = 0.991	Cropping attack (NCC=0.5082) Salt and Pepper (Density=0.001, NCC=0.7553)
2	P. Garg et al. [41]	Block based DWT-SVD	Chaotic (3d Logistic)	Transform	CASIA-v5, USC SIPI	45.36 dB	Salt & Pepper, Gaussian Noise, Crop, Median filter (NCC=0.93)
3	F. N. Thakkar et al. [24]	Block based DWT-SVD	NA	Transform	USC SIPI, Osirix DICOM image	42 dB payload of 8,559 bits	Salt & Pepper, Gaussian Noise, Crop, JEPG compression (NCC=0.9484)
4	V. Malik et al. [26]	DWT-SVD	NA	Transform	USC SIPI	35.69 dB	Salt & Pepper, Gaussian Noise, Crop, Rotation (MSE=402.929)
5	**RDWTSVDSVAUTH**	Block based DWT-SVD with S matrix	Block sequence	Transform	USC SIPI, BOSSbase	41.45 dB payload of 131,072 bits	Salt & Pepper, Gaussian Noise (NCC=0.9733)
6	**RDWTSVDUMAUTH**	Block based DWT-SVD with U matrix	Block sequence generation	Transform	USC SIPI, BOSSbase	27.23 dB payload of 8,192 bits	JPEG compression, Histogram equalization, Salt & Pepper (NCC=0.97)
7	**RESVDUMAUTH**	Block based SVD with U matrix, Hamming code	Block sequence generation	Spatial	USC SIPI, BOSSbase, 512x512 Medical images.	43.82 dB payload of 12,288 bits	JPEG compression, Histogram equalization, Salt & Pepper (NCC=1) GLCM based steganalysis (Accuracy=0.25%)

Conclusion:

In conclusion, this project focused on developing various robust image watermarking techniques using Singular Value Decomposition (SVD) in the context of telemedicine. The rapid advancement of telemedicine has brought about the need for secure transmission and protection of medical images, ensuring patient privacy and data integrity.

Through extensive research and experimentation, we developed different watermarking algorithms based on DWT, SVD, Hamming code, which proved to be effective in embedding imperceptible watermarks into medical images. These techniques not only provide robustness against common attacks but also maintain the visual quality and diagnostic information of the images.

The SVD-based watermarking algorithms demonstrated the ability to withstand various attacks such as compression, noise addition and cropping. This robustness is essential in telemedicine applications where medical images may be transmitted over unreliable networks or stored in vulnerable environments.

Moreover, the project explored the trade-off between imperceptibility and robustness, considering factors such as embedding capacity, computational complexity, and visual quality degradation. The evaluation of these techniques was conducted using metrics such as Peak Signal-to-Noise Ratio (PSNR), Normalized Correlation Coefficient (NCC), and perceptual quality assessment.

Overall, this project contributes to the understanding and advancement of robust image watermarking techniques using SVD, providing valuable insights for securing medical images in telemedicine. By safeguarding the integrity and confidentiality of these images, we can foster trust, enable accurate diagnosis, and ultimately improve healthcare services in the telemedicine domain.

References:

1. B. Li, J. He, J. Huang, Y. Q. Shi, “A Survey on Image Steganography and Steganalysis”, Journal of Information Hiding and Multimedia Signal Processing., Vol. 2, Issue 2, pp.142-172. (2011).
2. T. Lu and T. N. Vo, "Digital Media Steganography, Principles, Algorithms, and Advances”, Academic Press, Egypt, pp.189-213, ISBN. 9780128194386. (2020)
3. J.R. Krenn, “Steganography and Steganalysis”. (2004)
4. A. Cheddad, J. Condell, K. Curran, P. M. Kevitt, “Digital image steganography: Survey and analysis of current methods”, Volume 90, Issue 3, Pages 727-752. (2010)
5. F.Y. Shih, Digital Watermarking and Steganography: Fundamentals and Techniques, CRC Press. (2017)
6. I. J. Kadhim, P. Prashan, P. J. Vial, B. Halloran, “Comprehensive survey of image steganography: Techniques, Evaluations, and trends in future research”, Neurocomputing, Volume 335, Issue C, pp 299–326, https://doi.org/10.1016/j.neucom.2018.06.075. (2019)
7. H. Wang, S. Wang, “Cyber warfare: steganography vs. steganalysis” Communications of the ACM, Volume 47, Issue 10, pp 76–82. (2004)
8. H. Mathkour, B. Al-Sadoon, and A. Touir, "A New Image Steganography Technique," 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing, Dalian, China, pp. 1-4, doi: 10.1109/WiCom.2008.2918. (2008)
9. A. A. J. Altaay, S. B. Sahib and M. Zamani, "An Introduction to Image Steganography Techniques," 2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT), Kuala Lumpur, Malaysia, 2012, pp. 122-126, doi: 10.1109/ACSAT.2012.25. (2012)
10. VenkatramanS, Ajith Abraham and M. Paprzycki, "Significance of steganography on data security," International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004., Las Vegas, NV, USA, pp. 347-351 Vol.2, doi: 10.1109/ITCC.2004.1286660. (2004)
11. H. Daren, L. Jiufen, H. Jiwu, and L. Hongmei, "A DWT-based image watermarking algorithm," IEEE International Conference on Multimedia and Expo, ICME 2001., Tokyo, Japan, 2001, pp. 313-316, doi: 10.1109/ICME.2001.1237719. (2001)
12. A. Pradhan, A. K. Sahu, G. Swain, and K. R. Sekhar, "Performance evaluation parameters of image steganography techniques," 2016 International Conference on Research Advances

in Integrated Navigation Systems (RAINS), Bangalore, India, pp. 1-8, doi: 10.1109/RAINS.2016.7764399. (2016)

13. S. Kalman, D. Zheng, J. Zhao, W. J. Tam, and F. Speranza, "An Image Quality Evaluation Method Based on Digital Watermarking," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no. 1, pp. 98-105, doi: 10.1109/TCSVT.2006.887086. (2007)
14. D. Kalman, “A Singularly Valuable Decomposition: The SVD of a Matrix”, 2018
15. A. K. Singh, "Error detection and correction by hamming code," 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC), Jalgaon, India, pp. 35-37, doi: 10.1109/ICGTSPICC.2016.7955265. (2016)
16. A. Anand, A. K. Singh, “An improved DWT-SVD domain watermarking for medical information security”, Computer Communications Vol.152 pp.72–80, ISSN 0140-3664, (2020)
17. B. R. Ghosh , “Comparison of Two Watermark based Image Authentication Systems under LSB Substitution” , E-ISSN 2348-1269, P- ISSN 2349-5138, IJRAR November 2021, Volume 8, Issue 4(2021)
18. B. R. Ghosh, S. Banerjee, A. Chakraborty, S. Saha and J. K. Mandal, "A Deep Learning Based Image Steganalysis Using Gray Level Co-Occurrence Matrix," 2022 Second International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), Bhilai, India, 2022, pp. 1-8, doi: 10.1109/ICAECT54875.2022.9808013.
19. J. C. Lee, “Analysis of attacks on common watermarking technique.” IEEE Electrical and Computer Engineering Department University of British Columbia 2Anand6 Main Mall, Vancouver, BC Canada V6T 1Z4.
20. R. Liu and T. Tan, "An SVD-based watermarking scheme for protecting rightful ownership," in IEEE Transactions on Multimedia, vol. 4, no. 1, pp. 121-128, doi: 10.1109/6046.985560. (2002)
21. R. A. Ghazy, N. A. El-Fishawy, M. M. Hadhoud, M. I. Dessouky and F. E. A. El-Samie, "An Efficient Block-by-Block SVD-Based Image Watermarking Scheme," 2007 National Radio Science Conference, Cairo, Egypt, pp. 1-9, doi: 10.1109/NRSC.2007.371376. (2007)
22. N. M. Makbol, B. E. Khoo, “A Hybrid Robust Image Watermarking Scheme Using Integer Wavelet Transform, Singular Value Decomposition, and Arnold Transform.” In: Zaman,

H.B., Robinson, P., Olivier, P., Shih, T.K., Velastin, S. (eds) Advances in Visual Informatics. IVIC 2013. Lecture Notes in Computer Science, vol 8237. Springer, Cham. https://doi.org/10.1007/978-3-319-02958-0_4. 92013)

23. N. Divecha and N. N. Jani, “Implementation and performance analysis of DCT-DWT-SVD based watermarking algorithms for color images.” 2013 International Conference on Intelligent Systems and Signal Processing (ISSP), Vallabh Vidyanagar, India, pp. 204-208, doi: 10.1109/ISSP.2013.6526903. (2013)

24. F. N. Thakkar, V. K. Srivastava, “A blind medical image watermarking: DWT-SVD based robust and secure approach for telemedicine applications.” Multimed Tools Appl 76, 3669–3697 (2017). https://doi.org/10.1007/s11042-016-3928-7

25. V. Chandrasekaran, P. Sevugan, "Applying Reversible Data Hiding for Medical Images in Hybrid Domain Using Haar and Modified Histogram.", International Journal of Intelligent Engineering & Systems, Vol.10, Issue.4. (2017)
26. Sunesh, V. Malik, N. Sangwan, S. Sangwan “Digital Watermarking using DWT-SVD Algorithm” Advances in Computational Sciences and Technology ISSN 0973-6107 Volume 10, Number 7, pp. 2161-2171 © Research India Publications. (2017)

27. R. K. Singh, D. K. Shaw & J. Sahoo “A secure and robust block-based DWT-SVD image watermarking approach.” Journal of Information and Optimization Sciences, pp. 911-925, DOI: 10.1080/02522667.2017.1372137

28. M. Islam, A. Roy, R. Laskar, “Neural Network Based Robust Image Watermarking Technique in LWT Domain.” 1 Jan. 2018: 1691 – 1700. (2018)

29. S. Ojha, A. Sharma, R. Chaturvedi, “Centric-Oriented Novel Image Watermarking Technique Based on DWT and SVD.” In: Pant, M., Ray, K., Sharma, T., Rawat, S., Bandyopadhyay, A. (eds) Soft Computing: Theories and Applications. Advances in Intelligent Systems and Computing, vol 583. Springer, Singapore. https://doi.org/10.1007/978-981-10-5687-1_20. (2017)

30. A. K. Singh, “Robust and distortion control dual watermarking in LWT domain using DCT and error correction code for color medical image.” Multimed Tools Appl 78, 30523–30533 (2019). https://doi.org/10.1007/s11042-018-7115-x

31. Yasmeen, F., Uddin, M.S., “An Efficient Watermarking Approach Based on LL and HH Edges of DWT–SVD.” SN COMPUT. SCI. 2, 82 (2021). https://doi.org/10.1007/s42979-021-00478-y

32. R. Thanki, A. Kothari, “Multi-level security of medical images based on encryption and watermarking for telemedicine applications.” Multimed Tools Appl 80, 4307–4325. https://doi.org/10.1007/s11042-020-09941-z. (2021)
33. A. Miracle A, C. C. Adaobi, E. S. Eneji, I. W. Eyong, A. G. Ozioma, A. M. Udie “Implementation of medical image watermarking using RDWT and SVD for secure medical data transmission in healthcare systems.” https://www.stujournal.org [Original Article]. Vol 6 Issue 1, pp. 78-83. (2021)
34. N. Zermia, A. Khaldib, R. Kafib, F. Kahlessenaneb, S. Euschi “A DWT-SVD based robust digital watermarking for medical image security.”, Forensic Science International, Volume 320, 110691, ISSN 0379-0738. (2021)
35. M. S. Moad, M. R. Kafi, A. Khaldi, “A non-subsampled Shearlet transform based approach for heartbeat sound watermarking”, Biomedical Signal Processing and Control, Volume 71, Part A, ISSN 1746-8094, https://doi.org/10.1016/j.bspc.2021.103114. (2021)

36. D. Awasthi, Srivastava, V.K., “LWT-DCT-SVD and DWT-DCT-SVD based watermarking schemes with their performance enhancement using Jaya and Particle swarm optimization and comparison of results under various attacks.” Multimed Tools Appl 81, 25075–25099 (2022). https://doi.org/10.1007/s11042-022-12456-4

37. A. A. Mohammed, M. A M Abdullah, S. R. Awad “A Novel FDCT-SVD Based Watermarking with Radon Transform for Telemedicine Applications” International Journal of Intelligent Engineering and Systems, Vol.15, No.1 (2022)
38. A. K. Appu, G. P. Dubey and N. Gupta, "Designing of a Hybrid DWT-SVD Watermarking Technique of Colour Images with Digital Signature." IEEE International Conference on Current Development in Engineering and Technology (CCET), Bhopal, India, pp. 1-6, doi: 10.1109/CCET56606.2022.10080560. (2022)

39. Kumari, M.R.R., Kumar, V.V. & Naidu, K.R., “Digital image watermarking using DWT-SVD with enhanced tunicate swarm optimization algorithm.” Multimed Tools Appl (2023). https://doi.org/10.1007/s11042-023-14618-4

40. S. Sattarpour, “Robust optimal image watermarking using graph-based and discrete wavelet transforms, and whale optimization algorithm.” Multimed Tools Appl 82, 6667–6685 (2023). https://doi.org/10.1007/s11042-022-13639-9

41. P. Garg, A. Jain, “A robust technique for biometric image authentication using invisible watermarking.” Multimed Tools Appl 82, 2237–2253 (2023). https://doi.org/10.1007/s11042-022-13314-z

42. A. G. Weber, “The USC-SIPI Image Database: Version 6”, Ming Hsieh Department of Electrical Engineering Signal and Image Processing Institute. (2018)

43. https://dde.binghamton.edu/download/

44. https://www.kaggle.com/datasets/kmader/siim-medical-images