المؤتمر العلمي الدولي الثالث للعلوم و الهندسة
The 3RD Scientific International Conference in Science & Engineering

http://bwu.edu.ly/icse2024
Received 26/07/2024 Revised 15/08/2024 Published 10/09/2024
icse@bwu.edu.ly *

Lossy Compression of Biometric Images Implemented Using Walsh-Hadamard Transform (WHT)

Azuwam.A.Alhady [a] ,Muaz Adel[b] , Sofyan Alshoby[c]
[a]Department of Communication Engineering, College of Electronic Technology, Bani-Walid, Libya
[b]Department of Communication Engineering, College of Electronic Technology, Bani-Walid, Libya
[c]Department of Communication Engineering, College of Electronic Technology, Bani-Walid, Libya
Azuwamalhady@gmail.com

Abstract: Digital image compression plays a vital role in reducing storage requirements and facilitating efficient transmission of image data.This paper focuses on the application of the Walsh- Hadamard Transform (WHT) for biometric image compression. The objective is to develop an efficient compression algorithm that can effectively compress biometric images such as face, and finger images while maintaining acceptable image quality.The proposed approach involves inserting the biometric images into MATLAB, implementing the WHT algorithm, and experimenting with different truncation coefficients. The methodology begins by importing biometric images, consisting of face, and finger images, into MATLAB. Then, a custom WHT algorithm is developed to compress these images. The algorithm performs the forward WHT to transform the image coefficients. The transformed coefficients are then truncated using different truncation level such as 32,64,128,and 256 . After truncation, the inverse WHT is applied to assess the quality of the compression using well-established metrics such as Peak Signal-to-Noise Ratio (PSNR), Mean Square Error (MSE), and Structural Similarity Index (SSIM).The experimental results obtained through the evaluation metrics provide insights into the performance of the proposed biometric image compression approach. The PSNR metric measures the fidelity of the compressed images by comparing them to the original images. The MSE metric quantifies the average squared differences between the compressed and original images. Additionally, the SSIM metric evaluates the structural similarity between the compressed and original images. The outcomes of this research contribute to the field of biometric image compression by exploring the effectiveness of the Walsh-Hadamard Transform with truncation. The findings will provide valuable insights into the trade-off between compression ratios and image quality for biometric images. The developed algorithm and evaluation metrics can serve as a foundation for further research in optimizing biometric image compression techniques.

Keywords: *(PSNR, WHT, SSIM , Compression Ratio, MSE)*

Introduction

Image compression involves the efficient coding of digital images to minimize the number of bits needed to represent an image. With recent advancements in digital technology, visual information has become a crucial component in communication media. Some of the emerging applications include high definition TV, videoconferencing, video telephony, medical imaging, virtual reality, video wireless transmission and video server. The raw biometric digital image and video signal usually contain huge amount of information and therefore require a large channel or storage capacity if the number of users is large. In spite of advances in communications channel and storage capacity, the implementation cost often put constraint on capacity. Generally, the transmission or storage cost increase with increase in bandwidth requirement. To fulfill the requirements of channel or storage capacity, it becomes essential to utilize compression techniques that can effectively reduce the data rate while preserving the subjective quality of the decoded image or video signal. Image compression techniques achieve compression by exploiting statistical redundancies in the data and eliminating or reducing data to which the human eye is less sensitive. Two types of redundancies occur in images which are spatial and spectral redundancy. Spatial redundancy happens due to the correlation between neighboring pixels meanwhile spectral redundancy is due to correlation between different color planes [1]. Subband coding or transform coding, such as the Walsh Hadamard Transform (WHT), are employed in compression theory to eliminate spatial and spectral redundancies. However, there exists another form of redundancy known as temporal redundancy, which arises from the correlation between different frames in a sequence of images, such as those used in video conferencing or broadcast applications. To address this temporal redundancy, an inter-frame coding technique called Motion Compensated Predictive Coding is utilized [2]. Various compression algorithms are available for compressing videos and images, some of which have become compression standards like JPEG, H.261, and H.263 [3]. These algorithms typically rely on block-based compression and can be computationally intensive when executed on standard processors. The most computationally demanding components in these algorithms are the DCT, DFT, and WHT, which are commonly used in image processing applications. These linear image transforms are favored for their flexibility, energy compaction, and robustness. They effectively capture edges and offer energy compaction in modern approaches. Among these transforms, the WHT stands out due to its simplicity and computational efficiency

1. Biometric Image Compression

Effective transmission and storage of images rely heavily on image compression. The demand for storing, managing, and transferring digitized biometric images, particularly 2D biometric data and video signals, has experienced rapid growth. The size of these stored images can be enormous, consuming a significant amount of memory on storage devices. For instance, a 512x512 grayscale image consists of over 50,000 components, while an ideal color image at 640 x 480 pixels comprises nearly a million elements. Transferring these records from biometric sensors to internet servers for classification processes can be time-consuming [3]. Biometric images involve the measurement and statistical analysis of individuals' physical characteristics. This technology is primarily utilized for identification and access control purposes, as well as for surveillance-related individual identification. The underlying principle of biometric authentication is that each person is unique and can be identified by their inherent physical traits. For instance, if a government intends to implement a program for collecting biometric data, specifically iris, fingerprint, and facial patterns of its residents, managing the resulting storage demands, including database transfer over the internet or designing portable sensor devices, poses significant challenges. This motivates research into compression algorithms for various biometric images based on WHT (Walsh-Hadamard Transform) compression techniques. Furthermore, it is crucial to evaluate the compression performance of compressed biometric images using established measurement standards. As images typically consume a substantial portion of communication bandwidth, the development of efficient image compression techniques has become highly necessary [4]. The fundamental objective of image compression is to eliminate redundancy and discard irrelevant information. Redundancy removal focuses on reducing redundancy from the signal source, while irrelevancy elimination involves discarding pixel values that are not perceptible to the human eye.

2. Walsh Hadamard Transform (WHT)

The Walsh-Hadamard transforms (WHTs) are a class of non-sinusoidal transformations that enable the decomposition of a function into a linear combination of rectangular basis functions known as Walsh functions. These Walsh functions take on values of +1 and -1. The specific type of transform being computed is determined by the arrangement of the basis functions within the Walsh-Hadamard transformation matrix. When using Hadamard ordering, which is also referred to as natural ordering, the transformation matrix is derived by substituting the inverse transformation kernel [1].

$$s(x,u) = \frac{1}{\sqrt{N}}(-1)^{\sum_{i=0}^{n-1} b_i(x)b_i(u)} \quad (1)$$

where the summation in the exponent of Eq. (4.1) is performed in modulo 2 arithmetic, $N = 2^n$, and $b_k(z)$is the kth bit in the binary rep resentation of z. For example, if $n = 3$ and $z = 6$ (110 in binary), $b_0(z) = 0, b_1(z) = 1$ and $b_2(z) = 1$ If $N = 2$, the resulting Hadamard-ordered transformation matrix is

$$A_W = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (2)$$

where the matrix on the right (without the scalar multiplier) is called a Hadamard matrix of order 2. Letting H_N denote the Hadamard matrix of order N, a simple recursive relationship for generating Hadamard-ordered transfomation matrices is

$$A_W = \frac{1}{\sqrt{N}} H_N \quad (3)$$

Where

$$H_{2N} = \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix} \quad (4)$$

And

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (5)$$

Thus, Eq. (2) follows from Eqs. (3) and (5). In the same way,

$$H_4 = \begin{bmatrix} H_2 & H_2 \\ H_2 & -H_2 \end{bmatrix} \quad (6)$$

$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

And

$$H_8 = \begin{bmatrix} H_4 & H_4 \\ H_4 & -H_4 \end{bmatrix}$$

$$H_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \quad (7)$$

The Hadamard-ordered transformation matrices are derived by substituting specific values into Equation (3). The sequency of a row in a Hadamard matrix refers to the number of sign changes along that row. Similar to frequency, sequency measures the rate of change of a function, and like the sinusoidal basis functions in the Fourier transform, each Walsh function has a unique sequency. Since the elements of a Hadamard matrix are obtained from inverse kernel values, the concept of sequency applies to basis functions for the transformation as well. For example, the sequences of the basis vectors in Equation

(6) are 0, 3, 1, 2, while the sequences of the basis vectors in Equation (4.7) are 0, 7, 3, 4, 1, 6, 2, and 5. This ordering of sequences is a defining characteristic of the Hadamard-ordered Walsh-Hadamard transform. In signal and image processing applications, it is often desirable and common to arrange the basis vectors of a Hadamard matrix so that the sequence increases as a function of . The transformation matrix of the resulting sequency-ordered Walsh-Hadamard transform is obtained by substituting the inverse transformation kernel.

$$s_{(x,u)}=\frac{1}{\sqrt{N}}(-1)^{\sum_{i=0}^{n-1} b_i(x)p_i(u)} \quad (8)$$

where

$$p_0(u) = b_{n-1}(u)$$

$$p_1(u) = b_{n-1}(u) + b_{n-2}(u)$$

$$p_2(u) = b_{n-2}(u) + b_{n-3}(u) \quad (9)$$

$$b_{n-1}(u) = b_1(u) + b_0(u)$$

As before, the summations in Eqs. (8) and (9) are performed in modulo 2 arithmetic. Thus, for example,

$$H'_8=\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} (10)$$

The apostrophe (H') notation is used to indicate sequency ordering instead of Hadamard ordering. It is worth noting that the sequencies of the rows in H' match their respective row numbers, such as 0, 1, 2, 3, 4, 5, 6, and 7. Another approach to generating H' is by rearranging the rows of the Hadamard-ordered matrix H, where each row in H' corresponds to the row in H that represents the bit-reversed gray code of that row index. This relationship holds true for the n-bit gray code corresponding to $(s_{n-1} \dots s_2 s_1 s_0)_2$ can be computed as

$$g_i = s_i \oplus s_{i+1} \qquad 0 \le i \le n-2$$

$$g_{n-1} = s_{n-1} \;\text{ for }\; i = n-1 \qquad (11)$$

Where ⊕ denotes the exclusive OR operation, row s of H'_8 is the same as row $(g_0 g_1 g_2 \dots g_{n-1})_2$ of H_8. For example, row 4 or $(100)_2$ of H'_8, whose gray code is (110)2, comes from row $(011)_2$ or 3 of H_8. Note row 4 of H'_8 in Eq. (10) is indeed identical to row 3 of H_8 in Eq. (7). Figures (10).(a) and (b) depict graphically and numerically the sequency-ordered WHT transformation matrix for the case of N = 8. Note the sequency of the discrete

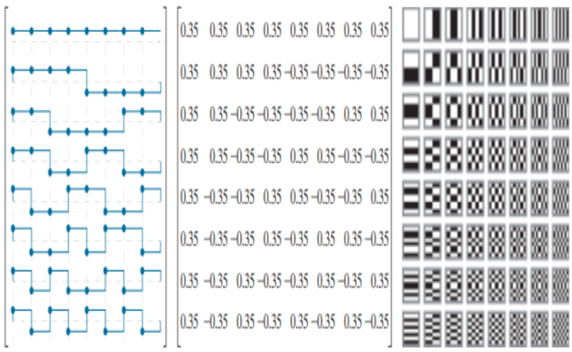


Fig. 1 : The transformation matrix and basis images for the sequency-ordered Walsh-Hadamard transform with N = 8 .

The graphical depiction of the orthogonal transformation matrix is shown in Figure 1(a). It is used in communication and imaging systems to enhance image quality with a minimal number of basis functions. The values in the matrix are rounded to two decimal places.b) The rounded version of the transformation matrix is illustrated in Figure (b). It is a real and symmetric matrix derived from Equations (11) and (3). The basis images corresponding to the transformation matrix are depicted in the figure as well

$$.\boldsymbol{A}_{W'} = \frac{1}{\sqrt{N}} \boldsymbol{H}'_8 \quad (12)$$

The task of demonstrating the orthogonality of the given item and verifying that it equals another item is assigned to the reader. Furthermore, it is worth noting the resemblance between the sequency-ordered basis images depicted in Figure 4.1(c). These basis images are derived from the separable 2-D inverse transformation kernel [6].

$$s = \frac{1}{N}(-1)^{\sum_{i=0}^{n-1}[b_i(x)p_i(u)+b_i(y)p_i(v)]} \quad (13)$$

Properties of WHT:-

- The Walsh transformation kernel possesses the properties of symmetry and orthogonality
- Each row index or column index in the kernel corresponds to the sequency of the respective row or column.
- Notably, the determinant of the smallest kernel is -2.
- The determinant of any other higher kernel of size $2^n \times 2^n \; is \; (2^n)^{2^{n-1}} \; for \; n > 1$

3. Truncation

Truncation is a technique used in various image compression algorithms to minimize the data required for image representation without compromising the visual quality beyond an acceptable level. It is commonly employed in transform-based compression methods, such as the one demonstrated in the provided code using the Hadamard transform.The Hadamard transform is a mathematical operation that converts an image from its spatial domain to its frequency domain. It decomposes the image into a set of coefficients that represent different frequency components. These coefficients capture the variations in pixel values across different spatial frequencies, ranging from low to high.When applying truncation in the context of the Hadamard transform, the coefficients are sorted based on their importance in representing the image. The most significant coefficients, which correspond to lower frequencies associated with essential image features, are retained, while the less significant coefficients, representing higher frequencies and fine details, are discarded. The choice of the truncation point, or the number of coefficients to keep, is a critical factor in balancing compression efficiency and image quality. A higher truncation point preserves more coefficients, resulting in better image fidelity but less compression. Conversely, a lower truncation point leads to higher compression but introduces more visual artifacts due to the loss of high-frequency information. It is worth noting that the decision on the truncation point can be influenced by several factors, including the specific compression algorithm employed, the desired compression ratio, and the visual quality requirements of the application. Different algorithms may have varying sensitivities to the amount of discarded high-frequency information. Additionally, certain types of images, such as smooth or textured images, may exhibit different compression characteristics and thus require different truncation strategies.The truncation is performed by setting a portion of the Hadamard transform coefficients to zero. The specific range of coefficients to be truncated can be adjusted by modifying the truncation coefficients in the code. By experimenting with different truncation points, one can explore the trade-off between compression efficiency and image quality, ultimately finding the optimal balance for a given compression scenario.Truncation is a key technique in image compression that selectively discards high-frequency information from transform coefficients to reduce data size. The choice of truncation point influences the compression efficiency and image quality trade-off. By understanding the characteristics of the image, the compression algorithm, and the requirements of the application, one can make informed decisions regarding truncation and achieve efficient image compression with acceptable visual quality.

4. Image Quality Assessment Methods

Objectively, four important similarity metrics are used to compare the reconstructed image after the compression algorithm process with the actual image.

4.1 Mean Square Error

MSE is the square of the maximum error between the compressed image and the original image. This metric is applied as:

$$MSE = \frac{1}{NM}\sum_{i=1}^{N}.\sum_{j=1}^{M}\left(I_o(i,j) - I_{rec}(i,j)\right)^2 \qquad (14)$$

$I_{original}$ is the original image before the compression process and $I_{reconstructed}$ is the image after applying DCT algorithms and MxN is size of image , A lower value of Mean Squared Error (MSE) indicates a reduced amount of error in the reconstructed image.

4.2 Peak Signal to Noise Ratio

PSNR is a commonly used metric for assessing the quality of a compressed image in comparison to the original image. It serves as a standard method to quantify the accuracy or fidelity of an image. The PSNR can be evaluated as:

$$PSNR = 20log_{10}\frac{255}{RMSE} \qquad (15)$$

4.3 Compression measures

The primary measure of the achievement of a compression algorithm is the compression ratio *(Cr)* and is determined by the formula.

$$\mathrm{Cr} = \frac{\text{original data size}}{\text{reconstrucated data size}} \qquad (16)$$

Higher compression values will make the image look worse, and vice versa

4.4 SSIM

SSIN is a modern method utilized for evaluating the quality of an image. It is assumed that the natural visual environment is statistical characteristic of have an important part in the evolution, development and adaptation of human visual system.

$$\mathrm{SSIM}(x,y) = \frac{(2\times\bar{x}\,\bar{y} + C1)(2\times\sigma_{xy} + C2)}{(\sigma_x^2 + \sigma_y^2 + C2)\times((\bar{x})^2 + (\bar{y})^2 + C1)} \qquad (17)$$

II .Flow Chart of the proposed WHT compression Algorithm

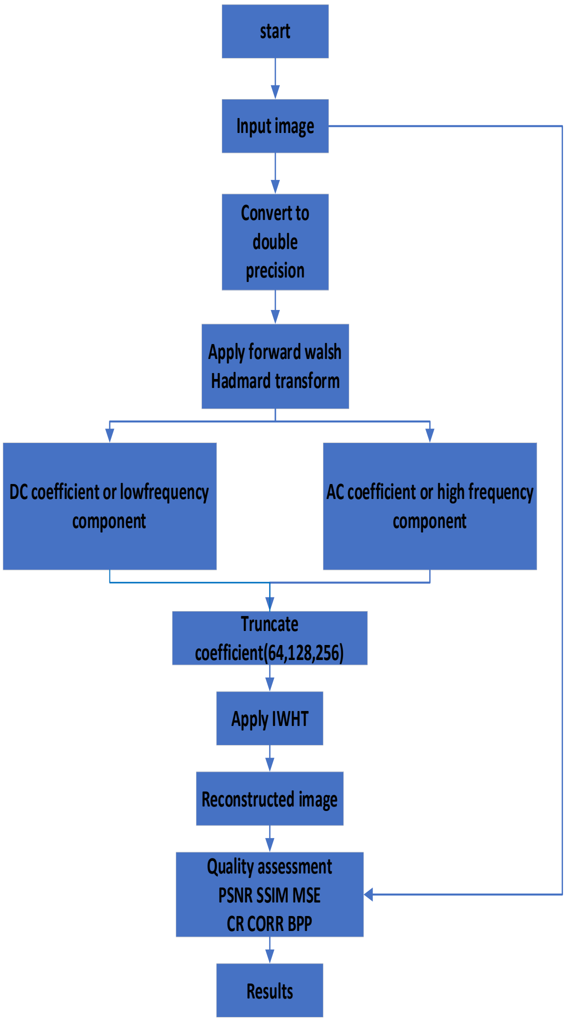


Fig. 2: Flow Chart of the proposed WHT compression Algorithm

1. WHT Algorithm implementationin MATLAB Environment

The Walsh Hadmard Transform WHT algorithm is modeled and verified using Matlab programming language . The provided algorithm written in MATLAB, a high-level programming language widely used for numerical computation and image processing. MATLAB provides built-in functions and tools for various image processing operations, making it suitable for implementing image compression algorithms.

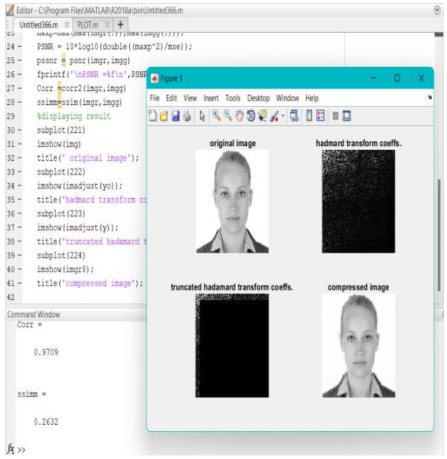


Fig . 3: Algorithm implementation in MATLAB

2. The proposed WHT compression algorithm steps

These procedure explain how WHT algorithm compress image using

1. File Selection:

- The code prompts the user to select a grayscale image file.

2. Image Processing:

- The selected image is read and converted to a double precision grayscale image.
- The size of the image is obtained.

3. The forward Hadamard Transform is applied column-wise and then row-wise to the image using the fwht function.

- The resulting Hadamard coefficients are stored in y.
- A backup of the coefficients is created in yo.

4. Truncation:

- The code truncates the Hadamard coefficients by setting all entries beyond the size of the original image to zero. This is done to remove high-frequency components and achieve compression.

- The truncated coefficients are stored in y.

5. Inverse Transform and Image Recovery:

- The inverse Hadamard Transform is applied column-wise and then row-wise to the truncated coefficients using the ifwht function.
- The recovered image is converted back to the uint8 data type.

6. PSNR and MSE Calculation:

- The mean square error (MSE) between the recovered image (imgr) and the original image (imgg) is calculated.
- The maximum pixel value in both images is determined, and the peak signal-to-noise ratio (PSNR) is calculated using the MSE.
- The PSNR value is printed.

7. Correlation and Structural Similarity Index (SSIM):

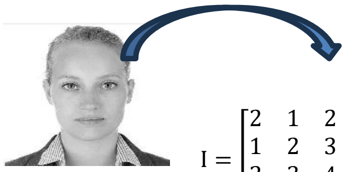
- The correlation coefficient between the recovered image and the original image is calculated using the corr2 function.
- The structural similarity index (SSIM) between the recovered image and the original image is calculated using the ssim function.

8. Result Display:

The original image, Hadamard transform coefficients, truncated coefficients, and compressed image are displayed in a

3. WHT Calculations

Step 1: Input Image Matrix:



$$I = \begin{bmatrix} 2 & 1 & 2 & 1 \\ 1 & 2 & 3 & 2 \\ 2 & 3 & 4 & 3 \\ 1 & 2 & 3 & 2 \end{bmatrix}$$

Let's go through the calculation steps in more detail:

$$I = \begin{bmatrix} 2 & 1 & 2 & 1 \\ 1 & 2 & 3 & 2 \\ 2 & 3 & 4 & 3 \\ 1 & 2 & 3 & 2 \end{bmatrix}$$

Step 2: Double Precision Hadamard Transform (WHT):

$$I = \begin{bmatrix} 2.000 & 1.000 & 2.000 & 1.000 \\ 1.000 & 2.000 & 3.000 & 2.000 \\ 2.000 & 3.000 & 4.000 & 3.000 \\ 1.000 & 2.000 & 3.000 & 2.000 \end{bmatrix}$$

The Hadamard matrix H4 is defined as follows:

$$H_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

To calculate the transformed matrix T, we perform the following multiplication:

$$T = H_4 \times I \times H_4 \qquad (18)$$

This involves multiplying the Hadamard matrix H4 with the input image matrix I, followed by another multiplication with H4.

$$T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \times \begin{bmatrix} 2.000 & 1.000 & 2.000 & 1.000 \\ 1.000 & 2.000 & 3.000 & 2.000 \\ 2.000 & 3.000 & 4.000 & 3.000 \\ 1.000 & 2.000 & 3.000 & 2.000 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} 6 & 8 & 12 & 8 \\ 2 & 0 & 0 & 0 \\ 0 & -2 & -2 & -2 \\ 0 & -2 & -2 & -2 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

After performing these calculations, we obtain the transformed matrix T:

$$T = \begin{bmatrix} 34 & 2 & -6 & -6 \\ 2 & 2 & 2 & 2 \\ -6 & 2 & 2 & 2 \\ -6 & 2 & 2 & 2 \end{bmatrix}$$

The transformed matrix T consists of the DC coefficient (top-left element), which represents the average intensity of the image, and the AC coefficients (non-DC elements), which represent the variations or differences from the DC coefficient.

Step 3: Truncate Coefficients:

In this step, we decide to truncate all but the DC coefficient (first element) in the transformed matrix T. Truncation involves discarding or setting to zero the less significant coefficients.

For example, if we truncate all the AC coefficients (non-DC elements) and keep only the DC coefficient, the truncated matrix T' becomes:

$$T' = \begin{bmatrix} 34 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

By truncating the AC coefficients, we reduce the amount of data required to represent the image.

Step 4: Inverse WHT (IWHT) for Reconstruction: To reconstruct the input image, we apply the inverse WHT using the truncated matrix T'. The inverse WHT is calculated as follows: Performing the inverse WHT calculation, we obtain the reconstructed image matrix I':

$$I' = \frac{1}{N \times N} \times (H_4 \times T' \times H_4) \qquad (19)$$

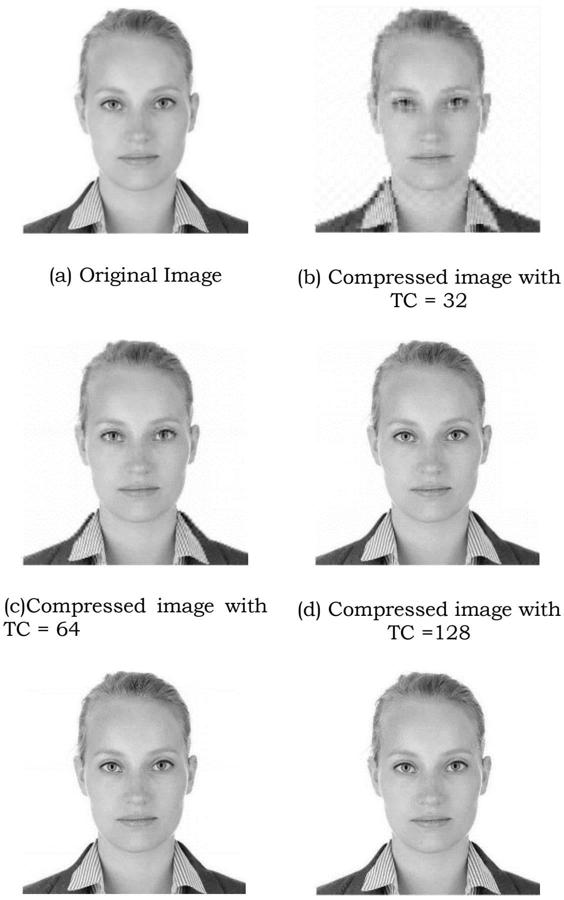
$$I' = \frac{1}{4\times4} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \times \begin{bmatrix} 34 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

$$I' = \begin{bmatrix} 2.125 & 2.125 & 2.125 & 2.125 \\ 2.125 & 2.125 & 2.125 & 2.125 \\ 2.125 & 2.125 & 2.125 & 2.125 \\ 2.125 & 2.125 & 2.125 & 2.125 \end{bmatrix}$$

The reconstructed image matrix I' contains the retained DC coefficient (average intensity) and zeros for the discarded AC coefficients. By truncating and using the inverse WHT, we have compressed the image by reducing the number of coefficients while sacrificing some detail information. The level and pattern of truncation can be adjusted to balance compression efficiency and image quality based on specific requirements.

III. Results and Discussion

In this analysis, we examine the performance of a proposed compression algorithm on various biometric images, employing different truncation thresholds. The threshold scores used range from 32 to 128. To evaluate the disparity between the original and reconstructed images, we employ metrics such as PSNR (Peak Signal-to-Noise Ratio), MSE (Mean Squared Error), CR (Compression Ratio), and SSIM (Structural Similarity Index). In this paper, we focus on the compression quality of biometric images, specifically investigating four different truncation thresholds. These thresholds serve as indicators of the relationship between achieving perfect reconstructed images and employing higher truncation thresholds. However, it is important to note that lower truncation levels result in inferior image quality due to reduced color depth and the removal of details within the image sections.Figure 4 and 5 visually illustrate the differences between the original and reconstructed images for both face and fingerprint images, respectively, representing various quality levels. Additionally, Table 1 and 2 present the error measurements for the reconstructed images, highlighting the impact of different truncation levels on image quality. It is evident from the tables that higher truncation levels yield excellent error measurements, while lower truncation levels lead to poorer image quality. The quality of the reconstructed images is best reflected in the PSNR values, with a quality level of 256 exhibiting no perceptual difference from the original image. Conversely, as indicated by Table 1 and 2, reducing the truncation level progressively worsens the error measurements, eventually reaching a point where perceptual differences from the original image become noticeable. Furthermore, Table 1 and 2 demonstrate an inverse relationship between PSNR and MSE. Higher PSNR values indicate a superior ratio of signal to noise, with the signal representing the original image and the noise representing the error in the compressed image. Thus, a compression algorithm with lower MSE and higher PSNR is considered superior. Additionally, the tables reveal an inverse relationship between CR and PSNR. A lower compression ratio implies the potential loss of more data. Consequently, a higher compression ratio preserves more subtle color gradations, resulting in more accurate images that closely resemble the original ones. Conversely, there exists a proportional relationship between compression ratio and MSE.



(a) Original Image

(b) Compressed image with TC = 32

(c)Compressed image with TC = 64

(d) Compressed image with TC =128

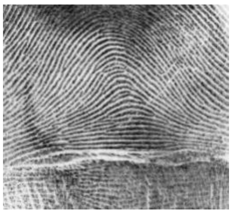
(e) Compressed image with TC =256

(f) Compressed image with TC = 512

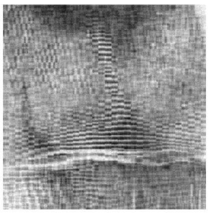
Fig. 4: Original & Reconstructed Face images with different truncation levels

Table 1: The Proposed Algorithm compression of Face Image

Face Image				
Q	**PSNR**	**MSE**	**CR**	**SSIM**
256	41.47	5.35	2	0.993
128	34.49	31.36	4	0.989
64	30.64	81.76	8	0.820
32	27.54	163.6	16	0.775

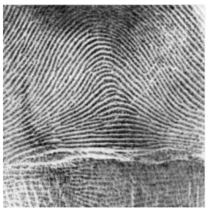


(a) Original Image

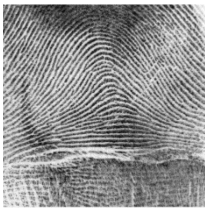


(b) Compressed image with TC = 32

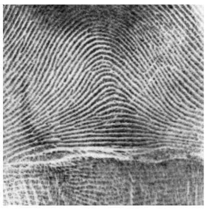
(c)Compressed image with TC = 64



(d) Compressed image with TC =128



(e) Compressed image with TC =256



(f) Compressed image with TC = 512

Fig. 5 : Original & Reconstructed finger print images with different truncation levels

Table 2: The Proposed Algorithm compression of finger print Image

Finger print Images				
Q	**PSNR**	**MSE**	**CR**	**SSIM**
256	39.13	7.94	2	0.9934
128	29.27	82.6	4	0.9383
64	22.05	122	8	0.7323
32	18.07	170	16	0.4133

IV. Conclusion

Based on the proposed method of applying the Walsh-Hadamard transform (WHT) and inverse transform (IWHT) for image compression, along with different truncation coefficients (32, 64, 128, and 256), the following conclusions can be drawn Increasing the truncation coefficient leads to higher compression ratios. As more coefficients are truncated, the resulting compressed data size decreases in comparison to the original image size. This indicates that the proposed method effectively achieves data reduction and compression. On other hand ,the image quality after compression and reconstruction is influenced by the truncation coefficient. Higher truncation coefficients result in more coefficients being discarded, leading to a higher level of compression. However, this can also cause a loss of important image details and a decrease in image quality. Therefore, as the truncation coefficient increases, it is expected that the image quality metrics may indicate a decline.

References

[1] Saini, P., & Singh, G. (2019). Image Compression using Modified Walsh-Hadamard Transform and Huffman Coding. International Journal of Advanced Computer Science and Applications, 10(5), 312-317.

[2] Nanda, A., & Mallick, P. (2017). Image Compression using Walsh Hadamard Transform and Run Length Encoding. International Journal of Engineering and Computer Science, 6(2), 20336-20346.

[3] Li, X., & Zhang, Q. (2020). Image Compression Based on Walsh-Hadamard Transform and Improved SPIHT Algorithm. Journal of Physics: Conference Series, 1069, 012043

[4] Li, C., & Li, H. (2017). Walsh-Hadamard Transform. In Digital Watermarking and Steganography (2nd ed., pp. 59-78). Morgan Kaufmann.

[5] Chen, C. H., & Tu, Y. H. (2017). Fast Image Compression Using Walsh-Hadamard

Transform and Huffman Coding. Journal of Applied Mathematics, 2015, 1-9.

[6] Chou, W. C., & Chang, P. C. (2016). Image Compression Based on Walsh-Hadamard Transform and Singular Value Decomposition. Journal of Information Hiding and Multimedia Signal Processing, 4(3), 191-203.

[7] Rao, K. R., & Yip, P. C. (2014). Discrete Cosine Transform: Algorithms, Advantages, Applications (2nd ed.). Academic Press.

[8] Sayood, K. (2017). Introduction to Data Compression (5th ed.). Morgan Kaufmann.

[9] Gonzalez, R. C., & Woods, R. E. (2017). Digital Image Processing (4th ed.). Pearson.

[10] Wallace, G. K. (1992). The JPEG Still Picture Compression Standard. Communications of the ACM, 34(4), 30-44.