Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

# Non-blind Image Watermarking Algorithm based on Non-Separable Haar Wavelet Transform against Image Processing and Geometric Attacks

Muhammad Khairi A Razak[1], Kamilah Abdullah[1], Suhaila Abd Halim[1,*]

[1] School of Mathematical Sciences, College of Computing, Informatics and Media, Universiti Teknologi MARA, 40450 Shah Alam. Selangor Malaysia

**ARTICLE INFO**

***Article history:***
Received 30 September 2022
Received in revised form 30 October 2022
Accepted 20 December 2022
Available online 18 January 2023

***Keywords:***
Crypto-watermarking; Digital image watermarking; Discrete wavelet transform; Non-separable Haar wavelet transform; Singular value decomposition

**ABSTRACT**

The protection of digital media become crucial and needs to be developed as technology continues to become more advanced. This is to cater to the problems in copyright protection and data security. This paper presents a new non-blind image watermarking algorithm applying modified non-separable Haar wavelet transform (NSHWT), singular value decomposition (SVD), Arnold's cat map and Rabin-p cryptosystem. The traditional transform domain watermarking (DWT) is resource-consuming, especially when performed on large image data. In order to improve on this issue, the proposed algorithm applied the modified NSHWT, a much more efficient alternative to the DWT. Another concern of digital watermarking algorithms is it is often publicly known, so encryptions are important to keep the image secure. The embedded watermark image is protected by scrambling the image using Arnold's cat map, and the scrambling parameters are encrypted with the Rabin-p cryptosystem to keep the algorithm secure. In general, the application of modified NSHWT and SVD makes the watermark highly robust against image processing and geometric processing attacks, and its security is ensured with the protection by Arnold's cat map and Rabin-p cryptosystem. There are already many algorithms with high imperceptibility and robustness, but in finding the perfect balance, authors often forsake other metrics such as efficiency, security, and flexibility. These are also essential when the algorithm is considered in real applications. Thus, the proposed algorithm is evaluated in terms of imperceptibility, robustness, and efficiency. The algorithm achieved imperceptibility results of 51.5157 average PSNR and 0.9991 average SSIM, and robustness results from 0.9710 to 0.9966 NC values. The algorithm is also around 40% faster to execute with modified NSHWT compared to the traditional DWT. Finally, it has a high embedding capacity of 6 bits per pixel. Overall, results show that the algorithm is highly imperceptible and robust against image processing and geometric attacks, while additionally being secure and flexible.

## 1. Introduction

Nowadays, people around the world can easily communicate online and access the World Wide Web, data security is a great concern. Not only people, but even electronic devices communicate

[*] *Corresponding author.*
*E-mail address: suhaila889@uitm.edu.my*

https://doi.org/10.37934/araset.29.2.251267

with other devices using the Internet of Things (IoT), which is rapidly advancing and being adopted more and more around the world [1]. In this context, digital watermarking is a relevant field that is constantly developing to further find better ways to protect important media. Advancements in the image watermarking field also indirectly helps in other matters related to images, such as image enhancement and detection [2, 3].

The requirements of a watermark depend on its intended application. The watermark may have to be either visible or invisible, or either fragile or robust. Some examples of digital image watermarking application are intellectual property protection, monitoring, tamper detection, description, fingerprinting, and channel protection at both sender and receiver [4]. For applications in copyright protection and secure communications, the watermark must be imperceptible and robust.

Watermarks embedded directly onto the lack of the robustness and imperceptibility to keep the media protected. To improve on this, watermarks should be embedded into the transform domain instead of the spatial domain, which is the image usually seen. Images can be transformed from the spatial domain to the transform domain by using techniques such as discrete wavelet transform (DWT), discrete cosine transform (DCT), singular value decomposition (SVD), and others. Watermark embedded in the transform domain is much more robust and imperceptible. The disadvantages of transform domain watermarking compared to spatial domain watermarking are less data capacity, more resource-consuming and more complex [5]. Despite the shortcomings of the transform domain watermarking, it is worth to apply and improve upon, simply due to the huge boost in robustness that it offers compared to spatial domain watermarking.

On top of the transform domain techniques, encryption is also required to secure the watermark from being directly removed by an attacker. Image encryption is often done by chaotic scrambling, where each pixel is moved to a different position, resulting in a meaningless image. Another method of encryption is by incorporating established cryptosystems into algorithms. This may prove to be a challenge as images can be large, so directly encrypting each pixel of an image will be resource-consuming. If a tried and tested cryptosystem can be applied efficiently in an image watermarking algorithm, it will be much more secure against individuals with malicious intent [6].

In this paper, a non-blind digital image watermarking algorithm is introduced, which applies modified non-separable Haar wavelet transform (NSHWT), SVD, Arnold’s cat map, and Rabin-p cryptosystem. In addition, the algorithm is evaluated in term of accuracy using peak signal to noise ratio (PSNR), structural similarity index measure (SSIM) and normalized correlation (NC). The algorithm is robust and imperceptible with high efficiency and embedding capacity [7].

Many algorithms have been created and have a good balance between imperceptibility and robustness, but in the search for this balance, algorithms have become more complex, which means performance is sacrificed and the watermarking process consumes more resources [8]. On top of maintaining that balance, keeping the algorithm efficiency is also an important factor to be considered.

Most authors create algorithms with image quality and image robustness in mind. In addition to the transforming of images, achieving the best parameters to balance the quality and robustness lead to the use of optimization techniques, which can be highly time-consuming. As such, they are not suitable for time-sensitive real applications [9]. A time-efficient algorithm would greatly widen the scope of application, so speed should be taken into consideration as well when designing a new watermarking algorithm. It is found that the lack of performance in algorithms is in the security and performance, due to being complex and time-consuming [10]. Therefore, how can we create a highly robust and imperceptible watermarking algorithm, without sacrificing security or efficiency?

Image watermarking with the 2D Haar DWT has been implemented in hardware, and it has shown to have high performance in terms of frequency, throughput, and low resource usage [11]. Therefore, this technique or its variations will make a good choice in creating an algorithm with high efficiency.

This paper is divided into five sections. First, the Introduction section elaborates on the general introduction on image watermarking. Second, the Theoretical Background section briefly details each of the techniques used in the proposed algorithm. Third, the Methodology section explains the method of evaluation, the embedding process, and the extraction process. Fourth, the Results and Discussion section presents the results of the evaluation and analysis of the results. Lastly, the Conclusion section concludes the findings of the algorithm evaluation.

### *1.1 Main Contributions*

A non-blind, imperceptible, robust, efficient, and secure image watermarking algorithm to embed colour watermark images is introduced in this paper. The techniques used are modified NSHWT, SVD, Arnold's cat map, and Rabin-*p* cryptosystem.

The DWT is a very popular and commonly used whether on its own or in hybrid with one or more different transform domain techniques. Transform domain techniques offer high robustness against attacks, but often have high calculation complexity. Therefore, a more efficient alternative is chosen and introduced for the transform domain technique, which is the modified NSHWT. In addition to the benefits provided by DWT, it consumes much less resources such as memory and time, to perform the transform or inverse transform on an image.

It is also important to bring the importance of security for image watermarking algorithms. Watermarking algorithms are often publicly known as so simply encrypting by scrambling is not secure enough. Therefore, in the algorithm proposed in this paper, Rabin-*p* cryptosystem is used to encrypt the important side data, which is required to descramble the scrambled image with Arnold's cat map. The method helps to further secure the watermark, as well as barely increasing the computational complexity.

### *1.2 Related Works*

A number of related works are discussed in this subsection to show the methods of recent algorithms. Many were able to obtain a good balance between imperceptibility and robustness.

Singh *et al.,* [12] created a block-based DWT-SVD image watermarking with quick response (QR) codes as watermark images. Both cover and watermark images are transformed by two-level DWT and separated into blocks of mxn size. SVD is applied on each channel of the RGB cover image. The tests show that it is robust against Gaussian noise and salt and pepper noise.

Alzahrani [13] proposed a DWT-SVD based image watermarking designed to find the relationship between invisibility and robustness. The grayscale cover image is decomposed by one level DWT, then both the LL sub-band and the grayscale watermark image is decomposed by SVD. The singular is used to compute one scaling factor, and thus SVD is used to generate the watermarked LL sub-band. The algorithm is robust but lacks any sort of encryption and only applies on grayscale cover and watermark images.

Bajracharya and Koju [14] proposed an image watermarking algorithm which applies DWT-SVD in the YCbCr colour space. The red-green-blue (RGB) cover image is converted into the YCbCr colour space, then a channel is decomposed by 4-level DWT. SVD is then applied on the HH4 sub-band. For the RGB watermark image, the R channel is scrambled by Arnold transform before being decomposed by 3-level DWT, followed by SVD on the HH3 sub-band. The S singular value of the watermark is embedded in the S singular value of the cover image, and finally reconverted into RGB colour space. It is shown to be robust against geometric attacks, but the algorithm itself is rather complex and can be difficult to apply on larger data.

Roy *et al.,* [15] also proposed an image watermarking algorithm based on DWT-SVD in the YCbCr colour space. The Y component of the converted RGB cover image is decomposed into blocks of 32x32, then 3-level DWT is applied on each block. The watermark is decomposed into 4x4 blocks before being decomposed by SVD along with the LL sub-band of the cover image, then the S singular value of the watermark is embedded in the S singular value of the cover image. The algorithm is claimed to be robust against various attacks, as well as hybrid attacks, but lacks encryption and is also complex with many DWT transformations to be performed.

Ahmadi *et al.,* [16] proposed a blind dual watermarking scheme by embedding both a robust watermark and a fragile watermark. The cover image is resized into 256x256 then divided into 4x4 blocks. The visual and edge entropy of the blocks are calculated, then the 256x256 cover image is decomposed by one-level DWT. The blue channel of the LL sub-band is chosen is divided into 4x4 blocks, and embedding blocks are chosen based on the previously calculated visual and edge entropy. All the blocks are transformed by SVD to embed the watermark using two scaling factors obtained by particle swarm optimization (PSO). The image is then restored by inverse SVD and DWT. On top of being resistant against image processing, geometric, and hybrid attacks, the dual watermarking and blind watermarking property allows for great potential for application on a wide range of applications. Optimizations are also calculated to choose the best embedding spots and scaling factors. But these optimizations will come with overhead that consume a lot of resources and make the algorithm even more resource-consuming.

Naffouti *et al.,* [17] proposed an advanced image watermarking system using DWT and SVD. DWT decomposes both the grayscale cover image and watermark image, followed by eigendecomposition on the HH sub-band. Then, two unitary and one diagonal matrices obtained from SVD decomposition, are combined to form the watermarked image by restoring with inverse DWT. The algorithm achieved good balance between imperceptibility and robustness and has a suitable computational complexity. However, there is still a lack of encryption techniques in the algorithm.

## 2. Theoretical Background

This section discusses the techniques used in the proposed algorithm. Four techniques are used, which are modified NSHWT, SVD, Arnold's cat map, and Rabin-p cryptosystem. The use of the two transform domain techniques that are modified NSHWT and SVD aim to greatly improve the imperceptibility and robustness. The modified NSHWT also makes the algorithm more efficient as compared to using the traditional DWT. On the other hand, Arnold's cat map and Rabin-*p* cryptosystem encrypts the image and important parameters to make the watermark more secure.

### *2.1 Modified Non-Separable Haar Wavelet Transform*

The NSHWT [18] is an alternative to the traditional DWT for 2-dimensional (2D) signals. DWT on 2D signals is usually done in two steps, by doing the 1-dimensional transform row-by-row, followed by column-by-column to obtain four sub-bands, LL, LH, HL, HH. The NSHWT calculates all sub-bands in one step, by going through the signal by 2 × 2 blocks, hence the "non-separable" in its name. This method has the advantage of not requiring transposition and frame memory during calculation [18].

A small change is made to the NSHWT equations to make them more suitable for digital image watermarking, as shown in Figure 1. Eq. (1) presents the modified NSHWT denoted as *T* or *T-1* from the spatial domain to the transform domain or vice versa, and the equations for the forward modified NSHWT, *T* is shown in Eq. (2).

$$\begin{pmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{pmatrix} \underset{T^{-1}}{\overset{T}{\rightleftarrows}} \begin{pmatrix} LL & LH \\ HL & HH \end{pmatrix} \tag{1}$$

$$LL = \frac{x_{11}+x_{21}+x_{12}+x_{22}}{4} \quad LH = \frac{x_{11}-x_{21}+x_{12}-x_{22}}{2}$$
$$LH = \frac{x_{11}+x_{21}-x_{12}-x_{22}}{2} \quad LH = \frac{x_{11}-x_{21}-x_{12}+x_{22}}{2} \tag{2}$$

where
$x_{mn}$ = pixel value at coordinate $(m, n)$,
*LL*, *LH*, *HL*, *HH* = corresponding pixel of each sub-band

The reverse transform of the modified NSHWT, $T^{-1}$, is as shown in Eq. (5) to Eq. (8).

$$x_{11} = \frac{2LL+LH+HL+HH}{2} \quad x_{12} = \frac{2LL-LH+HL-HH}{2}$$
$$x_{21} = \frac{2LL+LH-HL-HH}{2} \quad x_{22} = \frac{2LL-LH-HL+HH}{2} \tag{3}$$

The denominator of the LL sub-band is changed to 4, and the reverse transform is accordingly changed by adding the coefficient 2 to the values from the LL sub-band. The transformation is done in 2 × 2 blocks of the image matrix, and the location of the resulting *LL*, *LH*, *HL*, *HH* corresponds to the location of the 2 × 2 blocks being processed. So, the results will be the four sub-bands, each with half the dimension of the image being transformed. (is robust and better). The watermark will be embedded in the LL sub-band where it can be more robust, compared to embedding in any of the other sub-bands [19].

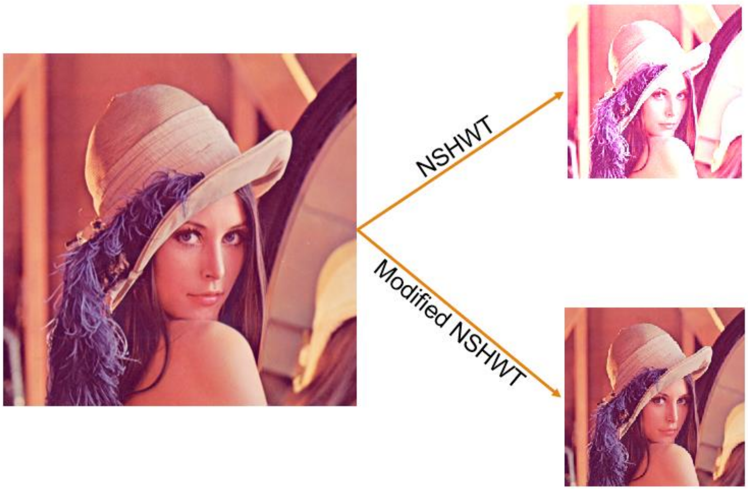

**Fig. 1.** Comparison between NSHWT and modified NSHWT

### *2.2 Singular Value Decomposition*

The SVD transforms a rectangular matrix into three singular values commonly referred to as *U*, *S*, and *V*. Eq. (4) shows the transformation of a rectangular matrix *A* of size *m* × *n*.

$$A = USV^T \tag{4}$$

where,
*U* = size $m \times m$ orthogonal matrix
*S* = size $m \times n$ diagonal matrix
*V* = size $m \times m$ orthogonal matrix

The *U* is an orthogonal matrix of size *m* × *m*, *S* is a diagonal matrix of size m × n, and $V^T$ is the transpose of an orthogonal matrix *V* of size n × n. The singular value *S* is chosen for watermark embedding as it has highly stable values which, when altered, does not affect the reconstructed image much, so the watermark will be imperceptible [20]. The SVD is also applied alongside modified NSHWT in hybrid to further improve robustness against more attacks and has low computational complexity [21].

### *2.3 Arnold's Cat Map*

The Arnold's cat map is a chaotic scrambling technique. The image is encrypted by scrambling each pixel to a different position. The scrambling is exactly reversible when provided with the correct parameters. Arnold's cat map is done using Eq. (5) [22].

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & a \\ b & ab+1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{5}$$

where,
$(x, y) =$ pixel coordinates
$(x', y') =$ pixel coordinates after scrambling
$a, b =$ positive integerscrambling parameters

The (*x*, *y*) is the pixel position before scrambling, (*x'*, *y'*) is the pixel position after scrambling, *a* and *b* are parameters for the scrambling, kept as secret keys. The process is done for every pixel in the matrix, and every pixel will have a unique position after scrambling. The inverse of the scrambling or the descrambling equation is shown in Eq. (6).

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} ab+1 & -a \\ -b & 1 \end{pmatrix} \begin{pmatrix} x' \\ y' \end{pmatrix} \tag{6}$$

Each pixel returns to their original positions after the descrambling is finished, and the image is perfectly reconstructed, assuming the scrambled matrix is not modified in any way and the scrambling parameters are correct. Encryption by scrambling is often used in image watermarking algorithms, but it is not secure enough [21], so an additional encryption is applied without sacrificing much efficiency, introduced in the following subsection.

### *2.4 Rabin-Cryptosystem*

The Rabin-*p* cryptosystem [23] is a variation of the Rabin cryptosystem. The target of encryption of this cryptosystem is the two parameters *a* and *b* used in the Arnold's cat map scrambling. The

Rabin-*p* cryptosystem has three processes: key generation, encryption, and decryption. The steps for each process are shown in Algorithm 1, Algorithm 2, and Algorithm 3.

```
Algorithm 1: Rabin-p Key Generation Algorithm
Input: The size $k$ of the security parameter
Output: The public key $N = p^2q$ and the private key $p$
    1) Choose two random and distinct primes p and q such that $2^k < p, q < 2^{k+1}$ satisfy $p, q \equiv 3 \ (mod\ 4)$.
    2) Compute $N = p^2q$
    3) Return the public key $N$ and the private key $p$.
```

```
Algorithm 2: Rabin-p Encryption Algorithm
Input: The plaintext $m$ and the public key $N$
Output: A ciphertext $c$
    1) Choose plaintext $0 < m < 2^{2k-1}$ such that $\gcd(m, N) = 1$
    2) Compute $c \equiv m^2 (mod\ N)$
    3) Return the ciphertext c.
```

```
Algorithm 3: Rabin-p Decryption Algorithm
Input: A ciphertext $c$ and the private key $p$
Output: The plaintext $m$
    1) Compute $w \equiv c \ (mod\ p)$
    2) Compute $m_p \equiv w^{\frac{p+1}{4}} \ (mod\ p)$
    3) Compute $i = \frac{c - {m_p}^2}{p}$
    4) Compute $j \equiv \frac{i}{2m_p} \ (mod\ p)$
    5) Compute $m_1 = m_p + jp$
    6) If $m_1 < 2^{2k-1}$, then return $m = m_1$. Else, return $m = p^2 - m_1$
```

In order to decrypt the ciphertext c, only the private key p is required. Hence the name Rabin-*p* cryptosystem. The correct output based on the condition at the end of Algorithm 3 that depends on the value obtained in the calculation. Since the Arnold's cat map scrambling generates important and sensitive side information, the scrambling parameters, it needs to be kept secure as well [21].

## 3. Methodology

This section discusses the methodology of the proposed algorithm and its analysis. The two transform domain techniques: modified NSHWT and SVD are used to transform the cover image into the transform domain, while the Arnold's cat map and Rabin-*p* cryptosystem encrypt the secret image before embedding. The same techniques are used in extracting the watermark and decrypting it. Figure 2 shows the flowchart of the whole process.

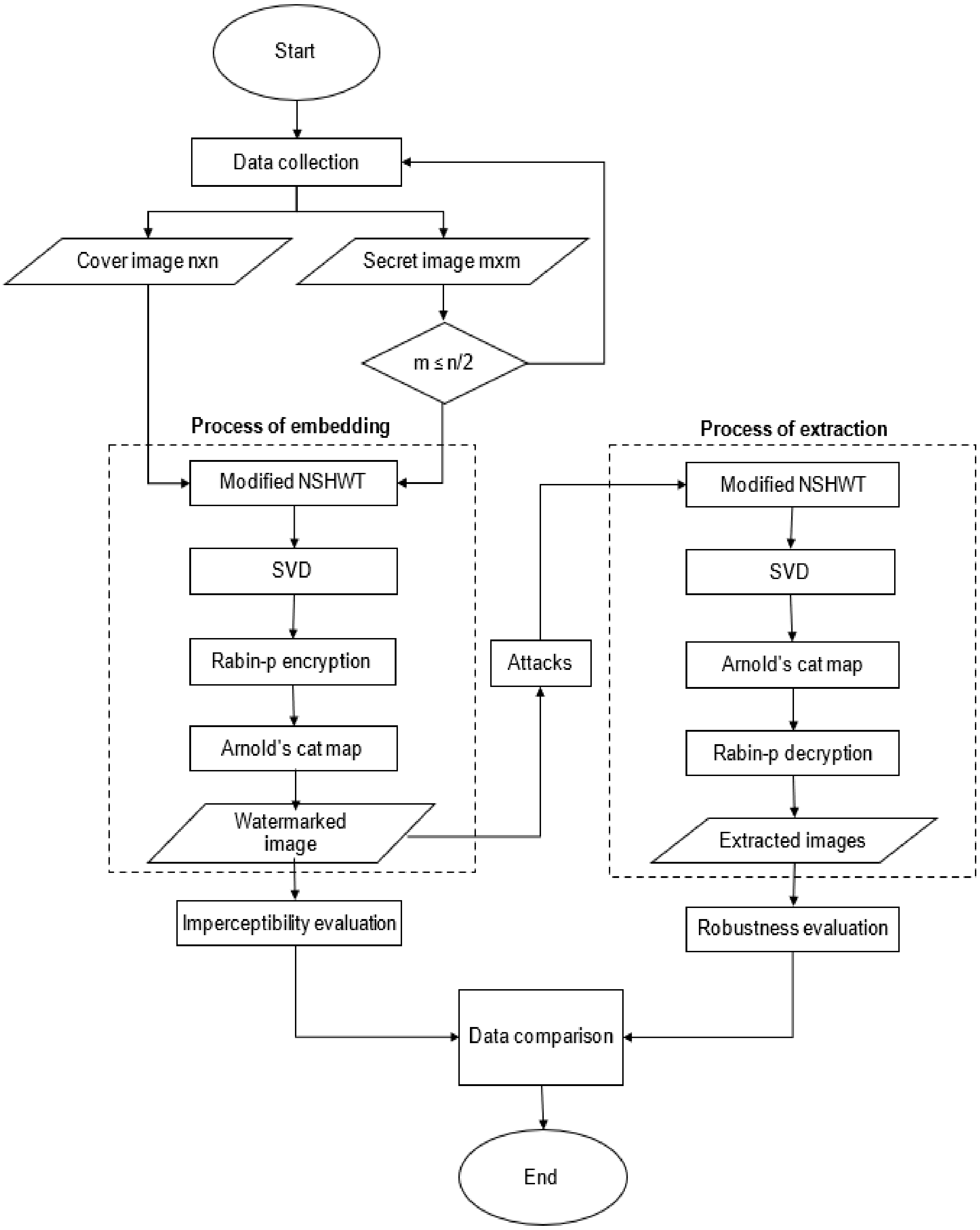

**Fig. 2.** Process flowchart

The algorithm is tested with a colour cover image known as Lena, and three colour secret images that named as Logo, Math, and Neon as shown in Figure 3. The cover image is a 24-bit color image of size 512x512 pixels while the secret images is 256x256 pixels.

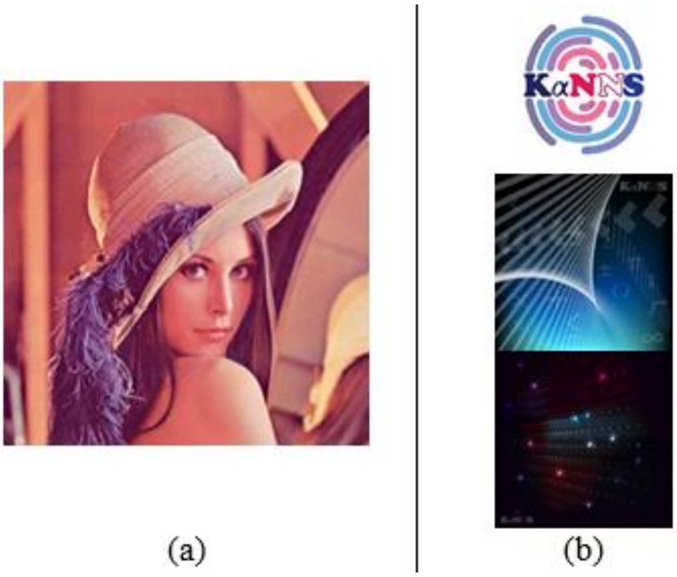

**Fig. 3.** Images (a) Test cover image (b) Secret images

The secret images are chosen based on the difference of its colours, which may affect its imperceptibility and robustness in the proposed algorithm. The secret image is also known as watermark in this study.

### *3.1 Embedding Process*

The watermark is embedded into the cover image by applying modified NSHWT, SVD, Arnold's cat map and Rabin-p cryptosystem. Eq. (7) is used to embed the watermark.

$$E = C + vS \tag{7}$$

where,
$E =$ data with embedded image
$C =$ location of embedding
$v =$ scaling factor
$S =$ data to be embedded

The scaling factor scales down the secret image to make it imperceptible and feasible to be embedded. A separate test was done with multiple different values to determine the suitable scaling factor with a good balance between imperceptibility and robustness. The detail steps of the embedding process are as follows:

i. Read the $m \times m$ cover image and the secret image which is also square but with half or less width and height of the cover image.
ii. Apply modified NSHWT (Eq. (2)) to transform the cover image into four sub-bands *LL, LH, HL, HH*.
iii. Apply SVD (Eq. (4)) to decompose the LL sub-band into three singular values, $U$, $S$, $V^T$.
iv. Apply Arnold's cat map transform (Eq. (5)) on the secret image to create a scrambled secret image, $W$.
v. Encrypt the control parameters of the Arnold's cat map transform, $a$ and $b$ with Algorithms 1 and 2, to obtain $c_1$ and $c_2$.

vi. The scrambled secret image *W* is added to the *S* singular value using Eq. (7) with scaling factor, *v* = 0.05 to obtain *D*.
vii. Apply SVD (Eq. (4)) on the *D* matrix to obtain three singular values $U_w$, $S_w$, $V_w^T$.
viii. Construct the watermarked *LL* sub-band, $LL_w$ by multiplying the modified singular values, *U*, $S_w$, $V^T$. The $U_w$ and $V_w^T$ singular values are collected for extraction process.

Apply inverse NSHWT (Eq. (3)) with the watermarked sub-band $LL_w$ to create the watermarked image, which is saved or sent along with the ciphertext $c_1$ and $c_2$, and singular values $U_w$ and $V_w^T$.

### *3.2* Extraction *Process*

The watermark is extracted from the watermarked image by applying modified NSHWT, SVD, Arnold's cat map and Rabin-p cryptosystem, with the assistance of the original cover image. Eq. (8) and Eq. (9) are used to extract the cover image and watermark respectively.

$$C = E - vS \tag{8}$$

$$S = \frac{E - C}{v} \tag{9}$$

The detail steps of the extraction process are as follows: example

i. Read the *m* × *m* watermarked image.
ii. Apply modified NSHWT (Eq. (2)) to transform the watermarked image into four sub-bands $LL_w$, *LH*, *HL*, *HH*.
iii. Apply SVD (Eq. (4)) to the $LL_w$ sub-band to obtain *U* , $S_w$, $V^T$ , and then compute the *D* matrix by multiplying the singular values $U_w$, $S_w$ and $V_w^T$ .
iv. Extract the embedded image from *D* to get the extracted *LL* sub-band and scrambled secret image using Eq. (8) and Eq. (9) respectively.
v. Apply inverse modified NSHWT (Eq. (3)) on the four DWT coefficients to reconstruct the extracted cover image.
vi. Decrypt the ciphertext $c_1$ and $c_2$ using Algorithm 3 to recover the control parameters *a* and *b*.

Descramble the scrambled secret image using Arnold's cat map (Eq. (6)) with the control parameters *a* and *b* to obtain the extracted secret image.

### *3.3 Performance Evaluation*

The proposed algorithm is evaluated in terms of imperceptibility, robustness, embedding capacity and speed. The imperceptibility is measured by calculating the PSNR and SSIM values between the watermarked image and the original cover image. Next, the robustness is calculated by the NC values between the extracted watermark image and the original watermark image. These three metrics are the most widely used objective measures for those respective metrics, so comparisons can be made with most other algorithms. The watermark is extracted after the watermarked image is distorted by attacks. Six different image processing attacks and four different geometric attacks are tested.

The imperceptibility and robustness are also compared with recent algorithms that apply the DWT and the SVD. Following that, the embedding capacity of the algorithm is tested by embedding watermarks of three different sizes, 16×16, 32×32, and 64×64. Finally, the efficiency of the modified NSHWT is tested by comparing the embedding and extraction time with the same algorithm but using the traditional DWT technique in place of the modified NSHWT.

The PSNR, SSIM and NC values between the cover image and watermarked image are calculated to measure the imperceptibility of the algorithm. The same metrics are used for robustness but calculated between the secret image and the extracted secret image. Let f be the reference image and g be the test image, the PSNR is calculated using Eq. (10) [24].

$$PSNR(f,g) = 10\log_{10}\left(\frac{255^2}{MSE(f,g)}\right) \tag{10}$$

The mean squared error (MSE) is calculated using Eq. (11) [24].

$$MSE(f,g) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(f_{ij} - g_{ij})^2 \tag{11}$$

The values $f_{ij}$ and $g_{ij}$ are the pixel values at coordinated (*i*, *j*) in the respective images. The cover image is the reference image *f*, and the watermarked image is the test image *g*. A higher PSNR value shows a higher quality image. Following that, SSIM values are also used for performance evaluation. The SSIM is calculated as shown in Eq. (12) [25].

$$SSIM(f,g) = \frac{\left(2\mu_f\mu_g + c_1\right)\left(2\sigma_{fg} + c_2\right)}{\left({\mu_f}^2 + {\mu_g}^2 + c_1\right)\left(\sigma_f^2 + \sigma_g^2 + c_2\right)} \tag{12}$$

where $\mu_x$ and $\mu_y$ are averages of image *x* and image *y* respectively, ${\sigma_x}^2$ and ${\sigma_y}^2$ are variances of image *x* and image *y* respectively, $\sigma_{xy}$ is the covariance of image *x* and image *y*, and $c_1 = (k_1L)^2, c_2 = (k_2L)^2$ are two variables to stabilize the division with the weak denominator, where L is the dynamic range of pixel values and $k_1 = 0.03$ and $k_2 = 0.01$.

Finally, to further increase the accuracy of the evaluation, the correlation between two images is computed with NC. The NC values can be obtained using Eq. (13) [26].

$$NC(W,W') = \frac{\sum_{z=1}^{3}\sum_{y=1}^{N}\sum_{x=1}^{M}\left[W(x,y,z)\times W'(x,y,z)\right]}{\sqrt{\sum_{z=1}^{3}\sum_{y=1}^{N}\sum_{x=1}^{M}\left[W(x,y,z)\right]^2}\sqrt{\sum_{z=1}^{3}\sum_{y=1}^{N}\sum_{x=1}^{M}\left[W'(x,y,z)\right]^2}} \tag{13}$$

where *W* and *W'* are the watermark image and extracted watermark image respectively, and *W*(*x,y) and W'*(*x,y*) are the pixel values at (*x,y*) in the *z* colour channel.

## 4. Results and Discussion

This section discusses the results obtained after implementing the proposed algorithm in MATLAB R2019a software on a computer with Intel Core i7-7700HQ CPU. The proposed algorithm is tested in terms of imperceptibility, robustness, efficiency, and embedding capacity.

### *4.1. Imperceptibility*

The imperceptibility is measured by calculating the PSNR, SSIM, and NC between the watermarked image and the original cover image. Table 1 shows the results of the values.

**Table 1**
Imperceptibility of the proposed Algorithm

| Watermarks | PSNR | SSIM | NC |
|---|---|---|---|
| Logo | 37.7080 | 0.9976 | 0.9997 |
| Math | 48.4705 | 0.9997 | 1.0000 |
| Neon | 68.3688 | 1.0000 | 1.0000 |
| Average | 51.5157 | 0.9991 | 0.9999 |

Each watermark has highly differing PSNR values, but the SSIM and NC values are very close to 1, so the watermarked image is similar to the cover image. From these results, it is shown that brighter images are less imperceptible as an embedded watermark, while darker images are more imperceptible. For better measure, the imperceptibility of the proposed algorithm is compared with recent algorithms, as shown in Table 2.

**Table 2**
Imperceptibility comparisons

| Watermarks | Algorithms | | | | |
|---|---|---|---|---|---|
| | [14] | [15] | [16] | [17] | Proposed |
| PSNR | 47.9971 | 51.1464 | **53.1365** | 48.1308 | 51.5157 |
| SSIM | - | - | 0.9952 | **0.9999** | 0.9991 |

The average value from Table 1 is taken as the imperceptibility of the proposed algorithm. The PSNR of the proposed algorithm is higher than algorithms by [14] and [15], but less than the algorithm by [16]. The proposed algorithm also achieved higher SSIM value than the algorithm by [16], but less than algorithm [17].

There is only a small difference between the value of the proposed algorithm and the other algorithms, so it has competent levels of imperceptibility. Next, secret images of different sizes are tested on the proposed algorithm. Table 3 shows the results for square secret images of dimensions 16, 32, and 64.

**Table 3**
Imperceptibility with different secret image sizes

| Watermarks | PSNR | | | SSIM | | | NC | | |
|---|---|---|---|---|---|---|---|---|---|
| | 16x16 | 32x32 | 64x64 | 16x16 | 32x32 | 64x64 | 16x16 | 32x32 | 64x64 |
| Logo | 51.1993 | 43.3009 | 35.1675 | 0.9999 | 0.9993 | 0.9954 | 1.0000 | 0.9999 | 0.9995 |
| Math | 58.8308 | 51.2156 | 44.4381 | 1.0000 | 0.9998 | 0.9992 | 1.0000 | 1.0000 | 0.9999 |
| Neon | 94.0771 | 71.5693 | 59.4384 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 | 1.0000 |

The algorithm successfully embedded the smaller secret images in the cover image. Generally, the imperceptibility decreases as the secret image size increases. For real cases, a higher scaling factor should be used on smaller secret images to ensure that it does not completely disappear during embedding due to the rounding of integer values.

### *4.2. Robustness*

The robustness is measured by calculating the PSNR, SSIM, and NC between the extracted watermark image and the original watermark image, with or without attacks. NC is the commonly used metric for robustness measurement. The results of the robustness calculations are shown in Table 4 and Table 5.

**Table 4**
Robustness of the proposed algorithm (NC)

| Attacks | Parameters | NC | | | |
|---|---|---|---|---|---|
| | | Logo | Math | Neon | Average |
| No attacks | - | 1.0000 | 1.0000 | 0.9984 | 0.9995 |
| Median filter | 3x3 | 0.9974 | 0.9925 | 0.9852 | 0.9917 |
| Histogram Equalization | - | 0.9965 | 0.9896 | 0.9270 | 0.9710 |
| Gamma correction | 0.8 | 0.9980 | 0.9981 | 0.9938 | 0.9966 |
| JPEG compression | 50 | 0.9990 | 0.9944 | 0.9533 | 0.9822 |
| Salt & pepper noise | 0.3 | 0.9405 | 0.8790 | 0.8046 | 0.8747 |
| Speckle noise | 0.05 | 0.9889 | 0.9554 | 0.8404 | 0.9282 |
| Rotation | 10° | 0.9977 | 0.9921 | 0.9362 | 0.9753 |
| Cropping | 256x256 top-left | 0.9941 | 0.9891 | 0.9820 | 0.9884 |
| Rescale | 0.5 | 0.9967 | 0.9895 | 0.9786 | 0.9883 |
| Cutting | 20 rows | 0.9995 | 0.9989 | 0.9958 | 0.9981 |

**Table 5**
Robustness of the proposed algorithm (PSNR and SSIM)

| Attacks | Parameters | PSNR | | | | SSIM | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Logo | Math | Neon | Average | Logo | Math | Neon | Average |
| No attacks | - | 54.4819 | 54.5392 | 45.0777 | 51.3663 | 0.9986 | 0.9996 | 0.9923 | 0.9969 |
| Median filter | 3x3 | 23.6474 | 26.4774 | 34.8109 | 28.3119 | 0.7533 | 0.8850 | 0.9639 | 0.8674 |
| Histogram Equalization | - | 21.6954 | 17.5196 | 24.0687 | 21.0946 | 0.9085 | 0.8043 | 0.7734 | 0.8287 |
| Gamma correction | 0.8 | 22.3198 | 31.1976 | 38.0967 | 30.5380 | 0.5647 | 0.9801 | 0.9783 | 0.8410 |
| JPEG compression | 50 | 27.5937 | 27.9620 | 29.8588 | 28.4715 | 0.8380 | 0.9167 | 0.8970 | 0.8839 |
| Salt & pepper noise | 0.3 | 9.6053 | 12.1281 | 14.9207 | 12.2180 | 0.1931 | 0.3776 | 0.3208 | 0.2972 |
| Speckle noise | 0.05 | 17.1118 | 17.1565 | 20.2846 | 18.1843 | 0.4820 | 0.6549 | 0.5552 | 0.5641 |
| Rotation | 10° | 24.1126 | 23.8275 | 26.9249 | 24.9550 | 0.7852 | 0.8750 | 0.8479 | 0.8360 |
| Cropping | 256x256 top-left | 20.0454 | 25.0927 | 34.4311 | 26.5231 | 0.5796 | 0.8549 | 0.9454 | 0.7933 |
| Rescale | 0.5 | 22.5761 | 25.1300 | 33.4697 | 27.0586 | 0.6936 | 0.8583 | 0.9482 | 0.8334 |
| Cutting | 20 rows | 30.0311 | 34.5084 | 40.6180 | 35.0525 | 0.9044 | 0.9796 | 0.9822 | 0.9554 |

Based on Table 4, the watermarks are highly robust with average NC values above 0.97, especially for median filter, gamma correction, and cutting attacks. However, the NC values drops by a significant amount against the salt and pepper and speckle noise attacks, especially with the dark-coloured Neon watermark.

From the PSNR column in Table 5, the quality seems least affected by the cutting and gamma correction attack, with the PSNR values still above 30dB. Additionally, there is evidence that a higher PSNR value does not necessarily mean a higher SSIM value, but a pattern between the two can be observed in their increase or decrease. Finally, a robustness comparison is made in Table 6.

**Table 6**
Robustness comparisons

| Attacks | Parameters | NC | | | | |
|---|---|---|---|---|---|---|
| | | [14] | [15] | [16] | [17] | Proposed |
| No attacks | - | - | 0.9992 | 1 | 0.9996 | 0.9995 |
| Median filter | 3x3 | 0.9999 | 0.9796 | 1 | 0.9995 | 0.9917 |
| Histogram Equalization | - | 0.9999 | 0.9233 | 1 | 0.9998 | 0.971 |
| Gamma correction | 0.8 | - | - | 1 | - | 0.9966 |
| JPEG compression | 50 | 0.9999 | 0.9831 | 0.8987 | - | 0.9822 |
| Salt & pepper noise | 0.3 | - | 0.9445 | 0.8177 | - | 0.8747 |
| Speckle noise | 0.05 | - | 0.9625 | 0.9962 | 0.9993 | 0.9282 |
| Rotation | 10° | - | - | 0.9873 | - | 0.9753 |
| Cropping | 256x256 top-left | - | 0.9479 | 1 | 0.9996 | 0.9697 |
| Rescale | 0.5 | - | 0.9659 | 1 | 0.9993 | 0.9883 |
| Cutting | 20 rows | - | 0.9753 | - | - | 0.9981 |

The average NC values from Table 3 are taken for the proposed algorithm's NC values. Compared to the algorithm in [15], the proposed algorithm is more robust against median filter, histogram equalization, cropping, rescale, and cutting. Against the algorithm in [16], the proposed algorithm is more robust against JPEG compression, and salt and pepper noise. Algorithm [17] is more robust than the proposed algorithm against all the compared attacks. Although the proposed algorithm does not perform better than every chosen algorithm against every chosen attack, there are certain attacks where it proves to be stronger than another algorithm. In conclusion, there is still room for improvement for the algorithm in terms of robustness against some attacks.

### *4.3. Efficiency*

The speed of the algorithm is compared with the modified NSHWT and the traditional DWT. The time it takes to embed and extract the algorithm is recorded when using modified NSHWT, and when using the traditional DWT instead. The process is simulated in MATLAB R2019a and recorded in Table 7.

**Table 7**
Speed comparison between modified NSHWT and DWT

| Time | Watermarks | Modified NSHWT | DWT | Percentage difference |
|---|---|---|---|---|
| Embedding time (s) | Logo | 1.7970 | 2.8730 | 37.4521 |
| | Math | 1.8170 | 2.8880 | 37.0845 |
| | Neon | 1.7160 | 2.8710 | 40.2299 |
| Extraction time (s) | Logo | 2.2770 | 4.1180 | 44.7062 |
| | Math | 2.2640 | 4.1990 | 46.0824 |
| | Neon | 2.2920 | 4.1470 | 44.7311 |

It can be observed that the algorithm applied with modified NSHWT finishes much sooner than the algorithm that applies DWT. For the embedding process, the algorithm using DWT takes about

40 percent more time to complete than the algorithm using modified NSHWT, while for the extraction process, it takes about 45 percent more time. This speed boost is owed to the modified NSHWT transforming with only one step, as opposed to the DWT doing separately for rows and columns. Therefore, the modified NSHWT has significantly better efficiency than the DWT technique.

### *4.4 Embedding Capacity*

The embedding capacity is analyzed in this subsection. Embedding capacity means the size of the largest watermark that can be embedded into the cover image. The bits by pixel embedding capacity is calculated using Eq. (14).

$$BPP = \frac{\text{Number of secret bits embedded}}{\text{Total pixels in the cover image}} \tag{14}$$

In this paper, a 256x256 secret image is to be embedded in the 512x512 cover image. Furthermore, the secret image is an RGB image, so each pixels hold 24 bits of data. Therefore, the calculation is as follows:

$$BPP = \frac{\text{256x256x24}}{\text{512x512}} = 6\ bpp \tag{15}$$

For the secret image with the sizes of 16x16, 32x32 and 64x64, the embedding capacity are 0.0234375 bpp, 0.09375 bpp and 0.375 bpp respectively.

Thus, the proposed algorithm can embed a maximum capacity of 6 bpp with colour watermarks of 256x256 as compared with the other three sizes. For comparison, the algorithm in [16] embeds a maximum of 0.0104 bpp for binary watermarks. Therefore, the proposed algorithm has a high embedding capacity, and the embedded watermark can hold a large amount of data.

## **5. Conclusions**

The proposed image watermarking algorithm has been discussed in terms of imperceptibility, robustness, efficiency, and embedding capacity. The results show that the algorithm is highly imperceptible and robust. However, it shows weakness against the salt and pepper attack and speckle noise attack. The comparisons show less differences in PSNR, SSIM and NC values as compared with the other algorithms. Hence, the proposed algorithm is proven to be competent among recent algorithms.

The algorithm could also embed different watermark sizes successfully, which is convenient flexibility for real applications. The imperceptibility and robustness, combined with the added flexibility, security, and efficiency, will surely allow future algorithms to be stronger and have a wider range of applications.

For future works, improvements could still be made to the algorithm to make it more robust against the aforementioned attacks, as well as the others that are less robust than other algorithms. This could be achieved by using different transform domain techniques or embedding techniques. The algorithm also strictly requires a square-shaped watermark, due to the property of the Arnold's cat map method. An alternative scrambling method could be used instead.

Next, a blind watermarking algorithm variant of the proposed algorithm could also be designed to take advantage of the efficiency and security. The coding of the algorithm could also be further optimized to improve the efficiency of the process. Lastly, there is potential for the NSHWT technique to be applied many other future algorithms because efficiency is essential when using digital image watermarking in real applications.

**Acknowledgement**

We would like to thank the Ministry of Education Malaysia, Research Management Centre (RMC) and College of Computing, Informatics and Media, Universiti Teknologi MARA (UiTM), Malaysia for the support in completing this research. This research was financially supported under the Fundamental Research Grant Scheme for Research Acculturation of Early Career Researchers (FRGS-RACER) with the grant code FRGS-RACER (RACER/1/2019/STG06/UITM//2).

## References

[1] Zainal, Salbiah, Rasimah Che Mohd Yusoff, Hafiza Abas, Suraya Yaacub, and Norziha Megat Zainuddin. "Review of Design Thinking Approach in Learning IoT Programming." *International Journal of Advanced Research in Future Ready Learning and Education* 24, no. 1 (2021): 28-38.

[2] Alias, Nur Ain, Wan Azani Mustafa, Mohd Aminuddin Jamlos, Mohd Wafi Nasrudin, Muhammad As' syarafi Mansor, and Hiam Alquran. "Edge Enhancement and Detection Approach on Cervical Cytology Images." *Journal of Advanced Research in Applied Sciences and Engineering Technology* 28, no. 1 (2022): 44-55. https://doi.org/10.37934/araset.28.1.4455

[3] Mustafa, Wan Azani, Nur Ain Alias, Mohd Aminuddin Jamlos, Shahrina Ismail, and Hiam Alquran. "A Recent Systematic Review of Cervical Cancer Diagnosis: Detection and Classification." *Journal of Advanced Research in Applied Sciences and Engineering Technology* 28, no. 1 (2022): 81-96. https://doi.org/10.37934/araset.28.1.8196

[4] Kumar, Ashwani. "A review on implementation of digital image watermarking techniques using LSB and DWT." *Information and Communication Technology for Sustainable Development* (2020): 595-602. https://doi.org/10.1007/978-981-13-7166-0_59

[5] Kadian, Poonam, Shiafali M. Arora, and Nidhi Arora. "Robust digital watermarking techniques for copyright protection of digital data: A survey." *Wireless Personal Communications* 118, no. 4 (2021): 3225-3249. https://doi.org/10.1007/s11277-021-08177-w

[6] Razak, Muhammad Khairi Abdul, Kamilah Abdullah, and Suhaila Abd Halim. "A Review On Digital Image Watermarking With Cryptosystem Techniques." In *2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, pp. 383-387. IEEE, 2021. https://doi.org/10.1109/ISCAIE51753.2021.9431809

[7] Razak, M. K. A., Abdullah, K., & Abd Halim, S. Robustness of Modified Non-Separable Haar Wavelet Transform and Singular Value Decomposition for Non-blind Digital Image Watermarking. Malaysian Journal of Mathematical Sciences 16, no. 2 (2022): 289-316. https://doi.org/10.47836/mjms.16.2.08

[8] Wan, Wenbo, Jun Wang, Yunming Zhang, Jing Li, Hui Yu, and Jiande Sun. "A comprehensive survey on robust image watermarking." *Neurocomputing* (2022). https://doi.org/10.1016/j.neucom.2022.02.083

[9] Sharma, Vipul, and Roohie Naaz Mir. "An enhanced time efficient technique for image watermarking using ant colony optimization and light gradient boosting algorithm." *Journal of King Saud University-Computer and Information Sciences* (2019).

[10] Utama, Sunariya, and Roshidi Din. "Performance Review of Feature-Based Method in Implementation Text Steganography Approach." *Journal of Advanced Research in Applied Sciences and Engineering Technology* 28, no. 2 (2022): 325-333. https://doi.org/10.37934/araset.28.2.325333

[11] Gafsi, Mohamed, Nessrine Abbassi, Rim Amdouni, Mohamed Ali Hajjaji, and Abdellatif Mtibaa. "Hardware implementation of the Haar 2D discrete wavelet transform with an application to image watermarking." In *2022 5th International Conference on Advanced Systems and Emergent Technologies (IC_ASET)*, pp. 324-329. IEEE, 2022. https://doi.org/10.1109/IC_ASET53395.2022.9765864

[12] Singh, Ranjeet Kumar, Dillip Kumar Shaw, and Jayakrushna Sahoo. "A secure and robust block based DWT-SVD image watermarking approach." *Journal of Information and Optimization Sciences* 38, no. 6 (2017): 911-925. https://doi.org/10.1080/02522667.2017.1372137

[13] Alzahrani, Ali. "Enhanced invisibility and robustness of digital image watermarking based on DWT-SVD." *Applied Bionics and Biomechanics* 2022 (2022). https://doi.org/10.1155/2022/5271600

[14] Bajracharya, Subin, and Roshan Koju. "An improved DWT-SVD based robust digital image watermarking for color image." *International Journal of Engineering and Manufacturing* 7, no. 1 (2017): 49. https://doi.org/10.5815/ijem.2017.01.05

[15] Roy, Soumitra, and Arup Kumar Pal. "A hybrid domain color image watermarking based on DWT–SVD." *Iranian Journal of Science and Technology, Transactions of Electrical Engineering* 43, no. 2 (2019): 201-217. https://doi.org/10.1007/s40998-018-0109-x

[16] Ahmadi, Sajjad Bagheri Baba, Gongxuan Zhang, Mahdi Rabbani, Lynda Boukela, and Hamed Jelodar. "An intelligent and blind dual color image watermarking for authentication and copyright protection." *Applied Intelligence* 51, no. 3 (2021): 1701-1732. https://doi.org/10.1007/s10489-020-01903-0

[17] Naffouti, Seif Eddine, Anis Kricha, and Anis Sakly. "A sophisticated and provably grayscale image watermarking system using DWT-SVD domain." *The Visual Computer* (2022): 1-21. https://doi.org/10.1007/s00371-022-02587-y

[18] Bamerni, Serwan Ali, and Ahmed Kh Al-Sulaifanie. "An efficient non-separable architecture for Haar wavelet transform with lifting structure." *Microprocessors and Microsystems* 71 (2019): 102881. https://doi.org/10.1016/j.micpro.2019.102881

[19] Zainol, Zurinahni, Je Sen Teh, and Moatsum Alawida. "A new chaotic image watermarking scheme based on SVD and IWT." *Ieee Access* 8 (2020): 43391-43406. https://doi.org/10.1109/ACCESS.2020.2978186

[20] Liu, Junxiu, Jiadong Huang, Yuling Luo, Lvchen Cao, Su Yang, Duqu Wei, and Ronglong Zhou. "An optimized image watermarking method based on HD and SVD in DWT domain." *IEEE Access* 7 (2019): 80849-80860. https://doi.org/10.1109/ACCESS.2019.2915596

[21] Peterson, Gabriel. "Arnold's cat map." *Math Linear Algebra* 45 (1997): 1-7.

[22] Zainol, Zurinahni, Je Sen Teh, Moatsum Alawida, and Abdulatif Alabdulatif. "Hybrid SVD-based image watermarking schemes: a review." *IEEE Access* 9 (2021): 32931-32968. https://doi.org/10.1109/ACCESS.2021.3060861

[23] Asbullah, Muhammad Asyraf, and Muhammad Rezal Kamel Ariffin. "Design of Rabin-like cryptosystem without decryption failure." *Malaysian Journal of Mathematical Sciences* 10 (2016): 1-18.

[24] Hore, Alain, and Djemel Ziou. "Image quality metrics: Psnr vs. ssim, in '2010 20th international conference on pattern recognition'." *Istanbul: IEEE* (2010): 2366-2369. https://doi.org/10.1109/ICPR.2010.579

[25] Naveed, Asim, Yasir Saleem, Nisar Ahmed, and Aasia Rafiq. "PERFORMANCE EVALUATION AND WATERMARK SECURITY ASSESSMENT OF DIGITAL WATERMARKING TECHNIQUES." *Science International* 27, no. 2 (2015).

[26] Wang, Dongyan, Fanfan Yang, and Heng Zhang. "Blind color image watermarking based on DWT and LU decomposition." *Journal of Information Processing Systems* 12, no. 4 (2016): 765-778. https://doi.org/10.3745/JIPS.03.0055