Preprints.org

Article

Not peer-reviewed version

# Exploring Cellular Automata Learning: An Innovative Approach for Secure and Imperceptible Digital Image Watermarking

Iram Khurshid Bhat , Fasel Qadir , Mehdi Neshat , Amir H. Gandomi *

Posted Date: 12 March 2024

doi: 10.20944/preprints202311.1615.v2

Keywords: Watermarking algortihms; Elementary Cellular Automata; Rule-30; Security; Copyright Protection; Authentication; Robustness

Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

*Article*

# Exploring Cellular Automata Learning: An Innovative Approach for Secure and Imperceptible Digital Image Watermarking

**Iram Khurshid Bhat [1], Fasel Qadir [1], Mehdi Neshat [2,3] and Amir H. Gandomi [3,4,*]**

[1] P.G Department of Computer Sciences University of Kashmir, North Campus Delina, Baramulla, Jammu & Kashmir, 193103, India bhat.eram29@gmail.com (I.K.B),faselqadir@uok.edu.in(F.Q)
[2] Centre for Artificial Intelligence Research & Optimisation, Torrens University Australia, Brisbane, Australia, mehdi.neshat@torrens.edu.au (M.N)
[3] Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, 2007, Australia, mehdi.neshat@uts.edu.au, gandomi@uts.edu.au (A.H.G)
[4] University Research and Innovation Center, Obuda University, 1034 Budapest, Hungary
* Correspondence: gandomi@uts.edu.au

**Abstract:** As technology and multimedia production have advanced, there has been a significant rise in attacks on digital media, resulting in duplicated, fraudulent, and altered data and the infringement of copyright laws. This paper presents a robust and secure digital image watermarking technique that has been implemented in the spatial domain and exploits the erratic and chaotic behavior of the powerful elementary cellular automata rule-30. This research is motivated by the potential to incorporate dynamic computational models into the field of image security. We aim to strike a balance between the crucial characteristics of the watermarking system, i.e., imperceptibility, capacity, and robustness, in the suggested blind watermarking technique. In this approach, prior to embedding, the grayscale watermark image is downsized to its two Most Significant Bits (MSBs). In the following, the 2-MSBs watermark is encrypted using an ECA rule-30 to level up the system's security attributes. Then, the host image is scrambled using ECA rule-30 to distribute the watermark pixels throughout the host image and thus achieve the highest robustness against geometrical attacks. Finally, the encrypted watermark data is embedded into the scrambled host image using the ECA rule-30-based embedding key. The proposed method performs better in terms of imperceptibility, capacity, and robustness when compared to several systems with similar competencies. The simulation's findings demonstrate strong imperceptibility as evaluated by the Peak Signal-to-Noise Ratio (PSNR), which has an average value of 58.3735 dB and a high payload. The experimental outcomes, observed across a diverse range of standardized attack scenarios, unequivocally establish the ascendancy of the proposed algorithm over competing methodologies in the realm of image watermarking.

**Keywords:** Watermarking; Elementary Cellular Automata; Rule-30; Security; Copyright Protection; Authentication; Robustness.

## 1. Introduction

Multimedia security has gained prominence due to the internet's tremendous development and widespread use. The easy access to the internet, increased use of social media, and influential developments in digital data modification and editing software have made unauthorized copying, editing, and manipulation of digital content so much easier, resulting in the necessity of digital content security and authentication. The watermarking of digital data is performed to accomplish objectives like the protection of copyright, the preservation of originality, the protection from illegal duplication, and the authentication of digital content[1]. Digital watermarking is a process that embeds digital data called watermark into the host document. The watermark is embedded in a manner that does not alter the host document and is invisible to the viewer. The embedded watermark is retrieved from the watermarked document using the extraction process. Digital watermarking ensures fortified security, copyright protection, temper resistance plus detection, verification of integrity, controlled copying, and monitored broadcasting of digital data [2].

© 2024 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



There are many different ways, such as human perception, resistance, type of document, reversibility, and working domain, to categorize digital watermarking techniques into various groups [3]. According to human perception, watermarking techniques can be classified into two categories: visible and invisible [4]. The watermarks embedded using visible watermarking techniques can be perceived easily from the host image. Examples include company logos, QR codes, signatures, initials, etc., that are placed at the corners of the images or TV channels/videos. The watermarks embedded using invisible watermarking techniques are hidden in the host image to preserve imperceptibility and, therefore, are unnoticeable to the human eyes. Their applications include ownership confirmation, integrity management, and digital document authentication. Since invisible watermarks are made to be hard to spot and remove, they are often more robust than visible ones. However, their invisibility may make them less helpful at proving ownership or preventing unauthorized usage.

Based on the resistance level, invisible watermarking schemes can further be categorized into four groups: fragile, semi-fragile, robust, and hybrid approaches [3]. Fragile watermarking involves embedding a watermark in a host document in a way that any slight modification to the host would cause the watermark to be destroyed or altered. Fragile watermarking schemes are not resistant at all, and their goal is to detect tempering, verify the integrity, and authenticate the digital content. The watermarks embedded using semi-fragile watermarking approaches can resist some basic attacks/modifications like compression or resizing; nevertheless, they remain fragile to major attacks/modifications. Semi-fragile watermarks are frequently employed in cases where some content modifications are likely to happen, but any malicious or illegal modifications ought to be identified and prevented [5]. The robust watermarking approaches involve embedding a watermark in a host document in such a manner that it becomes challenging to alter or remove it even if the host undergoes standard operations like cropping, compression, filtering, scaling, or other types of modification. Robust watermarks are highly resistant and are intended to survive deliberate or accidental modifications to the watermarked content. The hybrid watermarking approach combines robust and fragile approaches to simultaneously offer verification of data integrity, data authentication, and copyright enforcement [3].

The digital watermarking procedure can be applied to different types of digital host documents like images, texts [6], audio, and videos to manage digital rights, protect the copyright, and identify the content. Digital image watermarking is a process that embeds information into the host image in a manner that the embedded information doesn't alter the host image and is imperceptible to the viewer, which is then retrieved from the watermarked image during the extraction phase. Regarding digital watermarking, reversibility can be defined as the capability of the watermarking system to fully restore the original unwatermarked form of the host document after watermark embedment. Watermarking schemes are often categorized as irreversible watermarking and reversible watermarking [7] based on reversibility. The irreversible watermarking schemes cannot completely restore the host document's original form once the watermark has been embedded. In contrast, reversible watermarking schemes can fully restore the original form of the host document after watermark embedment and extraction.

Cellular automata (CA) are massively parallel computational architectures with the maximum level of granularity. It is a mathematical model that consists of the D-dimensional lattice of finite-state cells with local interaction. The system's evolution is dependent on the evolution of each of its individual components. Simple rules and structures can result in a wide range of unpredictable patterns, making the cellular automata theory incredibly interesting. For instance, the behavior of a Turing machine or any other CA can be simulated using the CA known as universal cellular automata. The CA is beneficial in a variety of applications due to its fundamental characteristics; the parallelism characteristic of CA makes it suitable for large-scale modeling and simulations. The complex structures, patterns, and behaviors produced by the global behavior of the cells in the cellular space are impossible to predict based on the individual behavior of cells. This characteristic of the CA is termed emergent behavior. The CA is scalable and can be easily scaled down or up according to the dimensions of the system being simulated; as a result, the CA is suitable for a variety of applications. CA is a flexible and adaptable tool for simulation and modeling as it can simulate various systems by customizing

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



the update rule and the initial values of the parameters to start up CA. The watermarking systems that are based on CA offer high privacy and security as they make watermark detection and removal extremely challenging, thereby making them ideal for preserving delicate and valuable data.

The motivation for exploring digital image watermarking with CA lies in its potential to significantly enhance core principles of image security. This research is driven by the aspiration to integrate advanced computational models into the field of image security, offering robustness, efficiency, and adaptability to diverse content. Leveraging CA's dynamic behavior and parallelism, alongside its chaotic nature, promises to provide robust watermark security against adversarial attacks. This study aims to investigate ECA rule-30 for digital image watermarking. By leveraging the unique properties of Rule-30, we seek to develop a watermarking method that not only provides effective protection against unauthorized modifications but also ensures the integrity and authenticity of the embedded watermark. An extensive investigation that considered numerous pertinent sources and analytical techniques was conducted to accomplish this goal. The major contributions of this work are as follows:

- This paper presents a robust and secure digital image watermarking technique implemented in the spatial domain. It leverages the erratic and chaotic behavior of the elementary cellular automata rule-30 to enhance the security and robustness of the watermarking system.
- The suggested blind watermarking technique tries to achieve an equilibrium amidst the pivotal watermarking system attributes i.e, imperceptibility, capacity, and robustness by downsizing the grayscale watermark image to its two Most Significant Bits (MSBs). By employing ECA rule-30 to encrypt the 2 most significant bits (MSBs) of the watermark, the security aspect of the system is augmented. Scrambling the host image with ECA rule-30 also distributes the watermark pixels, maximizing robustness against geometrical attacks.
- The proposed method outperforms several systems with similar competencies regarding imperceptibility, capacity, and robustness. The simulation's findings demonstrate strong imperceptibility, as evaluated by the Peak Signal-to-Noise Ratio (PSNR), with an average value of 58.3735 dB. The experimental outcomes across a diverse range of standardized attack scenarios establish the ascendancy of the proposed algorithm over competing methodologies in the field of image watermarking.

The remaining portions of the article will be divided into multiple sections, each of which will focus on a different facet of the given problem. The following section of the literature review will discuss a detailed background of digital image watermarking techniques based on the working domain in which they are implemented. Section 3 introduces cellular automata, elementary cellular automata, and rule-30. Section 4 discusses the methodology of the proposed scheme. This will be followed by section 5, where performance analysis and experimental discussion are presented. In section 6, we will discuss the results of our technique, compare and analyze these results, and draw conclusions based on our findings. The summary of the main findings and suggestions for further study will be included in the part that concludes the article. With this framework, we want to present a vivid and insightful exploration of cellular automata-based digital image watermarking.

## 2. Related Work

The performance of the different digital image watermarking approaches depends on the domain in which they are being implemented. Based on the working domain, digital image watermarking algorithms can be classified into three groups: spatial domain watermarking, transform or frequency domain watermarking, and hybrid domain watermarking [8].

### *2.1. Spatial Domain Algorithms*

In the spatial domain, the watermark information bits are embedded straight into the host image pixel values using various approaches like modification of Least Significant Bits (LSBs) [9–13] or Intermediate Significant Bits (ISBs) [14] of the host image, patchwork approach, Local Binary Pattern (LBP) approach [15], histogram modification approach [16,17], and approaches based on

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



correlation [18–20] and spread spectrum [21–23]. The crucial characteristics of the watermarking system, imperceptibility, capacity, and robustness, have been perfectly balanced by spatial domain techniques. These techniques are more straightforward, more effective, and execute more quickly. In terms of capacity, these techniques allow the embedding of significant data. However, these methods only work effectively when the image has not been altered by humans or subjected to noise. A significant flaw with spatial domain watermarking is that the watermark can be removed through image cropping. Furthermore, a tiny watermark may be embedded repeatedly. Therefore, despite losing the majority of the image data due to numerous attacks, a single watermark remaining will be viewed as a success.

A digital image scrambling technique-based image watermarking approach using CA has been presented in [24] by Ye and Li. This work determines the fractal box dimensions of the CA and profoundly analyzes the chaotic properties of CA. In this approach, the fractal box dimensions are used to select the chaotic CA rule, then the host image is scrambled using that rule, and finally, the watermark is embedded in the scrambled image. The scrambled watermarked image is then descrambled to obtain the watermarked image. According to experimental findings, this technique is resistant to different attacks like compression, cropping, and noise; therefore, it is robust. A novel blind watermarking approach based on the game of life CA has been propounded by Adwan et al. in [9]. The Game-of-Life is the most well-known and robust rule of a CA. The proposed approach is implemented in the spatial domain, and it employs the LSB substitution technique for inserting the two most significant bits of the grayscale watermark into the host image. The suggested approach generates the k number of game-of-life generations and uses the locations of the live cells to insert data into the LSBs of the host image. According to the imperceptibility evaluation of the suggested scheme, the embedded watermark in the watermarked image is not perceptibly visible. The experimental findings demonstrate that the suggested approach is secure, simple, and robust enough against passive attacks.

A highly secure image watermarking approach based on the logistic map, the 2D Arnold's cat map, and the 2D Game of Life cellular automata has been suggested by Moniruzzaman et al. [10]. In the proposed scheme, the security mechanism is ensured by scrambling both the grayscale host image and the binary watermark image. The host image is scrambled using the 2D Arnold's cat map before the embedment of the watermark data into it. The initial configuration to start the Game of Life CA is set using the logistic map, and then the Game of Life CA is evolved to scramble the binary watermark image. At last, the scrambled watermark bits are inserted into the LSBs of the scrambled host image pixels. The suggested solution is used for image authentication and has the ability to avert unauthorized changes. Comparing the experimental findings of the proposed method to three other current chaos-based watermarking schemes depicts that the suggested approach significantly outperforms other approaches. The solitary constraint of this approach lies in its applicability exclusively to square images, as the Arnold transform's functionality is confined to image dimensions of uniform size.

A reversible medical image watermarking method proposed by Tjokorda et al. [11] uses the LSB modification technique for tamper detection and recovery in the region of interest (ROI) of medical images. The experiments' results indicate good performance and imply that the suggested watermarking approach more cleverly handles the attacks that target a particular area of the pictures, i.e., block tempering attacks. A secret image data hiding technique by Manjula and Danti [12] employs the (2-3-3) LSB approach to embed confidential data into the host image. The proposed method embeds secret image data having eight bits per pixel into the LSBs of the RGB host image. The (2-3-3) LSB approach embeds the first 2 bits in the red channel, the next 3 bits in the green channel, and the last 3 bits in the blue channel of the RGB host image. This approach offers promising results, considerably improving PSNR and MSE values compared to the prior technique.

An imperceptible digital image watermarking scheme for color images in the spatial domain has been propounded by Abraham and Paul [13]. The goal of this study is to develop a color image

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



watermarking method that does not reduce the picture quality drastically or alter the perceptual color. Additionally, watermark data is inserted in every image block to provide the highest robustness against attacks and enable tamper detection and recovery features. This scheme uses M1 and M2 masks during the watermark embedding process. The watermark information is distributed to the nearby pixels in the chosen area using these masks. The embedding channel uses the M1 mask, and to account for the differences introduced in the embedding channel, the other color channels are adjusted using mask M2. After being tested on various images, the suggested algorithm assures excellent quality watermarked images that can withstand many attacks.

### *2.2. Transform Domain Algorithms*

In the transform domain, the watermark data is not inserted directly into the host image but rather into the frequency coefficients of the host image. A number of transformational approaches have been devised for frequency coefficient generation, such as Discrete Wavelet Transform (DWT) [25], Discrete Cosine Transform (DCT) [4,26–28], Singular Value Decomposition (SVD) [4], Discrete Fourier Transform (DFT), Fast Fourier Transform (FFT), Cellular Automata Transform (CAT), Polar Harmonic Transform (PHT), and Hadamard. The transform domain watermarking techniques first transform the host image from the spatial domain to the frequency domain using any image transformation method in order to generate the coefficients and then alter these coefficients to embed the watermark data. Finally, the inverse transformation method is applied to obtain the watermarked image.

Laouamer and Tayan have proposed a robust semi-blind sensitive text image watermarking approach [27] that is DCT and linear interpolation-based. The main aim of this approach is to deal with the issues related to tamper detection, authenticity proof, integrity verification, and protection of digitally sensitive text images. This approach first transforms both the host image and the watermark image into YUV color space, and then both images are divided into 8 x 8 blocks. DCT is applied on each 8 x 8 block of both images, and finally, linear interpolation is applied on the quantized DCT coefficients to obtain the watermarked image. According to this study, a good trade-off between robustness and imperceptibility can be achieved if the watermark data is inserted into the medium-frequency (MF) components. The main contribution of this work is to extract the watermark perfectly from the attacked watermarked image. Roy and Pal [28] have employed DCT and repetition code to put forth a blind color watermarking technique for embedding multiple watermarks. The primary purpose of the proposed method is to enable copyright ownership protection and multiple owner authenticity validations. In this approach, the host image's blue and green components are first fragmented into non-overlapping blocks, and then the DCT is applied to each block. This approach embeds two binary watermark logos scrambled using Arnold's chaotic map before embedding. The watermark bits are embedded into the middle-frequency coefficients of blue and green components using repetition codes. The proposed approach shows high robustness, imperceptibility, and better PSNR value but exhibits high computational complexity.

Liu et al. [29] have propounded a DCT and fractal encoding-based digital image watermarking algorithm. The proposed algorithm combines the traditional DCT technique with the fractal encoding method. This approach encrypts the host image twice, first by encoding it with the fractal encoding and second by applying DCT on the encoded parameters. The experiments carried out show that the proposed approach has high robustness and a better PSNR value. Singh and Bhatnagar have presented a robust watermarking technique [30]. The techniques employed by this blind watermarking approach are integer DCT, dynamic stochastic resonance (DSR), and non-linear chaotic maps. In this approach, integer-DCT is applied to the host image to convert it into an integer linear transform, and the resulting coefficients are divided into non-overlapping blocks. Then, the Non-linear chaotic map is used to select the random blocks, which form the circulant matrix. The watermark bits are embedded into the circulant matrix by computing the singular values. This approach uses the dynamic stochastic resonance (DSR) phenomena for efficient watermark extraction. The verification step is included to

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



deal with the false positive problem caused in SVD systems. The experiments carried out show that the proposed approach is robust and imperceptible against various attacks.

Ernawan et al. have propounded a DCT psycho-visual threshold-based digital image watermarking algorithm [31]. First, the host image is divided into non-overlapping blocks for watermark embedding, and their modified entropy is computed. Then, DCT is applied to the blocks having the lowest entropy values to obtain the middle-frequency coefficients. Some of these middle-frequency coefficient pairs are modified with a psycho-visual threshold to embed the watermark bits. The watermark is scrambled using Arnold Scrambling before embedding to strengthen the security levels. For evaluation of the proposed algorithm, the watermarked image had undergone various attacks like a median filter, low-pass filter, sharpening, JPEG and JPEG2000, image noise, and geometrical attacks such as image scaling and cropping. The results demonstrated that the proposed method is invisible and robust compared to the existing methods.

A novel color image watermarking scheme based on Discrete Cosine Transform (DCT) and Cellular Automata (CA) has been put forth by M Jana, and B Jana [32]. This study focuses on developing an image watermarking system with enhanced security and high embedding capacity without compromising the visual quality of the host image. The RGB host image is first separated into red, green, and blue channels. After that, each channel is partitioned into non-overlapping 8x8 blocks; subsequently, DCT is applied on each block, and Zigzag scanning is performed. To increase the security and the robustness of the proposed system, CA rule-15 is used to encrypt the watermark data prior to embedding and CA rule-85 is used for decryption purposes. The CA rule-340 and mapping table are used to modify the DCT coefficients and incorporate the encrypted watermark image. The propounded method is equated with the existing methods. Furthermore, the simulation findings demonstrate an average Peak Signal Noise Ratio (PSNR) value of 54 dB that signifies strong imperceptibility and concomitantly having a good embedding capacity of 1.48 bpp.

## 3. Methods and Materials

### *3.1. Cellular Automata*

John von Neumann and Stanislaw Ulam are credited as inventors of cellular automata. In the late 1940s, Neumann proposed a theoretical model for artificial biological systems called the "Universal Constructor" that had the ability of self-reproduction and was comprised of a two-dimensional infinite grid of square cells; each cell had five neighbours, including itself, with 29 states per cell. While some people researched the subject during the 1950s and 1960s, its popularity only began to spread outside of academics in the 1970s with the development of Conway's Game of Life. The Game of Life, developed by mathematician John Conway and hyped up by an article in Scientific American by Martin Gardner, is a simple two-dimensional totalistic cellular automaton with two states per cell using the Moore neighbourhood.

A Cellular automaton can be defined as a mathematical model having discrete space and time domains. This computational system is dynamic in nature, and simple rules govern its evolution. Formally, CA is defined by the quadruple A = ($\mathbb{Z}^D$, S, N, f) where:

- $\mathbf{Z^D}$ is a D-dimensional lattice, i.e., a regular arrangement of lattice-sites/points in Euclidean space which is discrete and is closed under subtraction and addition, called cellular space.
- **S** is a finite set of states.
- **N** is a neighborhood vector $(\overrightarrow{\mathbf{n_1}}, \overrightarrow{\mathbf{n_2}}, \overrightarrow{\mathbf{n_3}} \ldots \overrightarrow{\mathbf{n_m}})$, comprising of m different cells of $\mathbb{Z}^D$ and is identified separately for every cell of the lattice $\mathbb{Z}^D$. For a given cell "x" a set of cells $\{x + \overrightarrow{n_i} | i = 1, 2, 3, 4 \ldots m\}$ forms its neighborhood and from the property of lattice $(x + \overrightarrow{n_i}) \in \mathbb{Z}^D$.
- $\mathbf{f : S^m \longrightarrow S}$ is the local transition function known as the local rule in CA that synchronously updates the state of every cell based on the current state of the cells in its neighbourhood.

All the cells in a cellular space are updated simultaneously using the same local update rule. A function/mapping that assigns cells their states/values from the set of states (S) is called the configuration of the

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



D-dimensional CA ($\mathbb{Z}^D$) and all configurations are contained in the set $\mathbf{S^{Z^D}}$. Besides, the evolution of CA generations can be described as the hopping of the cellular space from one configuration to another according to its global transition function $\mathbf{G : S^{Z^D} \longrightarrow S^{Z^D}}$. One-dimensional cellular automata (1D CA) and two-dimensional cellular automata (2D CA) are the two most prevalent forms of CA. A finite or infinite set of identical cells/sites containing discrete variables arranged in the form of a linear array is referred to as 1D CA [33]. Figure 1 depicts the 1D CA with a five-cell neighbourhood for updating the state of the cell "n" where the local rule depends on the current states of the two nearest left (n - 2, n - 1) and right (n + 1, n + 2) neighbours of the cell "n" and the current state of cell "n" itself. The transition function according to which this CA evolves is defined using equation (1).

$$S_n(t+1) = f(S_{n-2}(t), S_{n-1}(t), S_n(t), S_{n+1}(t), S_{n+2}(t)) \tag{1}$$

Where $S_{n\text{-}2}(t)$, $S_{n\text{-}1}(t)$, $S_{n+1}(t)$, and $S_{n+2}(t)$ are the current states of the neighbours at time t, $S_n(t)$ is the current state of the cell "n" at time t, $S_n(t+1)$ is the new state of the cell "n" at time t+1, and f is the local update rule.

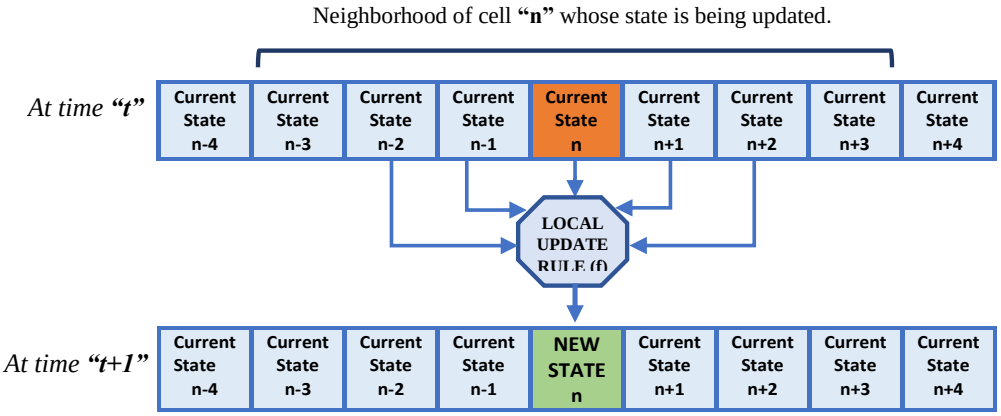



**Figure 1.** 1D Cellular Automata with five-cell Neighborhood.

A 2D CA can be defined as a finite or infinite set of identical cells/sites containing discrete variables arranged in the form of a matrix/grid. For 2D CA, a number of neighbourhood structures are possible. Still, the von Neumann neighbourhood and the Moore neighbourhood are the two most prevalent neighbourhood structures illustrated in Figure 2 (a) and 2(b), respectively. The von Neumann/diamond-shaped/five-cell neighbourhood is defined as the set of cells that includes the central cell and its four adjoining perpendicular cells. Equation (2) illustrates the transition function of the five-cell neighbourhood.

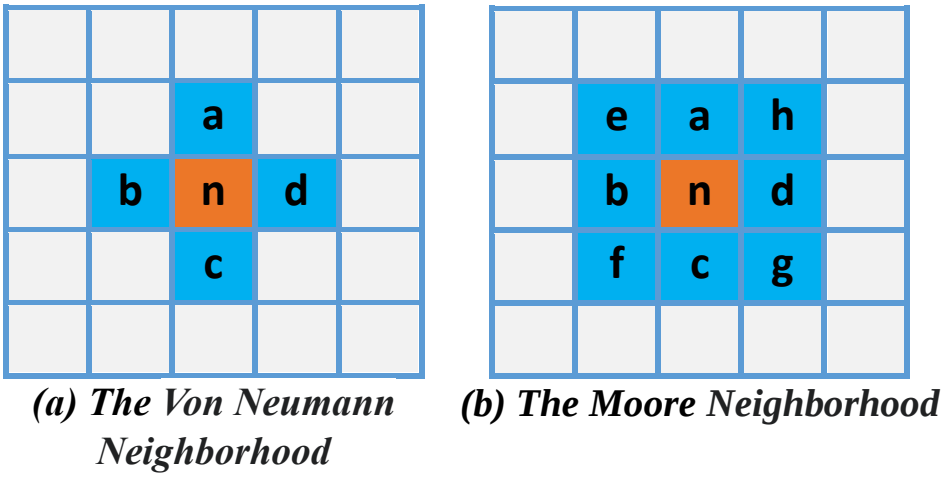
$$S_n(t+1) = f(S_n(t), S_a(t), S_b(t), S_c(t), S_d(t)) \tag{2}$$

The Moore/square-shaped/nine-cell neighbourhood is defined as the set of cells containing the central cell, its four adjoining perpendicular cells, and four adjoining diagonal cells. Equation (3) illustrates the transition function of the nine-cell neighbourhood.

$$S_n(t+1) = f(S_n(t), S_a(t), S_b(t), S_c(t), S_d(t), S_e(t), S_f(t), S_g(t), S_h(t)) \tag{3}$$

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2






*(a) The Von Neumann Neighborhood* *(b) The Moore Neighborhood*

**Figure 2.** 2D CA Neighborhood Structures.

3.1.1. Elementary Cellular Automata (ECA)

An elementary cellular automaton is the most basic form of 1D CA, extensively studied by Stephen Wolfram since the 1980s. Given the defining characteristics of a CA, the potential scenario in its most basic form is the linear array of cells with two states (0 or 1) per cell and the neighbourhood set S = (the cell, its immediate right neighbour, its immediate left neighbour). This basic scenario is termed elementary cellular automata, illustrated using the transition function in equation (4).

$$S_n(t+1) = f(S_{n-1}(t), S_n(t), S_{n+1}(t)) \tag{4}$$

Since there are two states and three neighbours for each cell, this leads to $\mathbf{2^3 = 8}$ possible ways to configure the neighbourhood, and therefore there are only $\mathbf{2^{2^3} = 256}$ total ECA rulesets possible. The Wolfram code, developed by Stephen Wolfram, presented a method for assigning numbers between 0 and 255 to each rule, and this method has now become a norm. Interestingly, Wolfram has categorized these 256 rulesets into four possible classifications based on the increasing complexities of their behaviours [34].

- Class (I) Uniformity: The evolution of almost all the initial configurations rapidly leads to stable, uniform structures, thus completely losing randomness, if any.
- Class (II) Oscillation: The evolution of almost every initial configuration results in patterns that are either stable after a large number of generations or tend to repeat themselves. The initial configuration may lose some of its randomness, but some remains.
- Class (III) Random: The evolution of most of the initial configuration results in chaotic or completely pseudo-random sequences.
- Class (IV) Complexity: Almost all the initial configurations evolve into complex structures with intriguing ways of interaction.

**Rule-30:** is an elementary cellular automata rule that belongs to a class (III) of Wolfram's classification because of its chaotic and erratic behaviour. The ruleset for ECA Rule-30 is depicted in Figure 3, and Figure 4 illustrates the first fifty evolutions of the ECA, which is updated using rule-30 with the initial configuration being a single middle black cell. Rule-30 generates random configurations from simple initial states, which makes it remarkable, and that is why Mathematica has also employed it as a random number generator [35]. It is left permutative, thus highly sensitive to the initial states, as a tiny difference in one state had a significant impact on subsequent results [36]. The whole set of NIST statistical tests was applied to the output of rule-30 to analyze and evaluate its randomness, and it was found that except for one, all the tests were passed [37]. It was also shown that for better performance,

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



it is important to have an effective window size, and a size of 200 bits is believed to be sufficient for better randomness [38]. The main reason to use the ECA Rule-30 for embedding and encryption in our system is for its special characteristics, such as its chaotic behavior and sensitivity to initial conditions, which make it ideal for producing high-entropy pseudo-random sequences.

The computational irreducibility of rule-30 is one of its remarkable characteristics. This implies that computational prediction of its behavior over time or determining a brief representation of its evolution is equal to computing each step separately. It is difficult to minimize or predict the pattern's intricacy. Rule 30's computational universality is well-known. In the larger context of complexity theory and the study of cellular automata, Rule 30 is of great interest because it provides a classic illustration of how simple local rules can give rise to complex and seemingly unpredictable patterns.

| ***t*** | 111 | 110 | 101 | 100 | 011 | 010 | 001 | 000 |
|---|---|---|---|---|---|---|---|---|
| ***t+1*** | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |

**Figure 3.** The Ruleset for Elementary CA Rule-30.

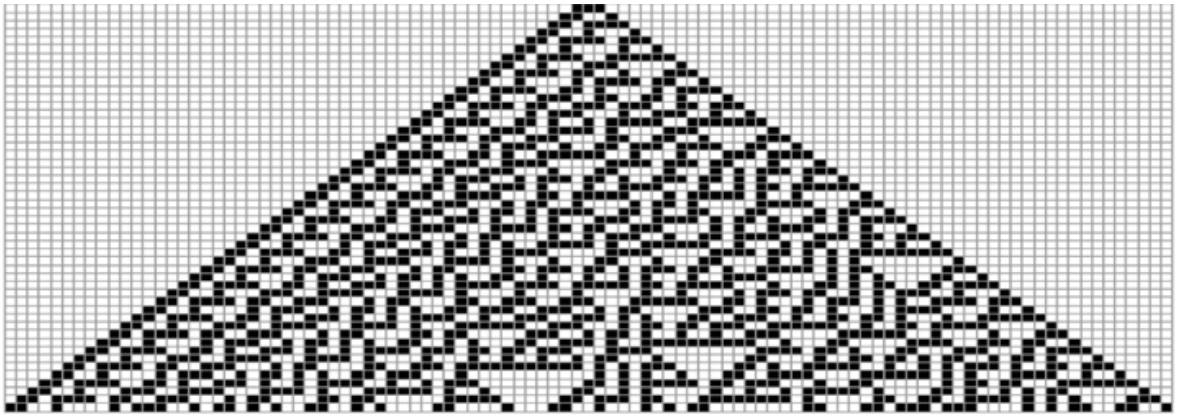


**Figure 4.** First Fifty Evolutions of ECA Evolved Using Rule-30 from [39].

### 3.2. *Proposed Image Watermarking Scheme.*

The proposed digital image watermarking method is a spatial domain approach in which watermark bits are embedded into the LSBs of the host image. The crucial characteristics of the watermarking system, imperceptibility, capacity, and robustness, have been perfectly balanced by spatial domain techniques. These techniques are simpler, more effective, and execute more quickly. In this approach, a grayscale host image of size (Hm X Hn), a grayscale watermark image of size (Wm X Wn), and the secret keys (embedding-key and scrambling-key) generated using CA have been considered as the inputs. The proposed scheme is based on powerful ECA rule-30, which is an ideal source of randomness. Confusion can result from the fact that we are viewing a 2D representation of the 1D CA. It should be noted that the proposed scheme constructs the two-dimensional representation out of a large number of evolutions of one-dimensional output data. Nevertheless, the CA system is purely one-dimensional. This scheme is divided into five phases: the Secret Keys Generation Phase, the Watermark Preprocessing Phase, the Host Image Scrambling Phase, the Watermark Embedding Phase, and the Watermark Extraction Phase.

#### 3.2.1. The Secret Keys Generation Phase.

This is a prime part of the proposed watermarking scheme, as the two distinct shared secret keys are generated in this phase using ECA. One is the embedding key, which is of the size of the watermark

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



image, i.e. (Wm X Wn), and the second is the scrambling key, which is of the size of the host image, i.e. (Hm X Hn) as shown in Figure 6 (a).

For the embedding-key generation, this scheme employs the ECA as the linear array of window size of 201 bits/cells contained within the periodic boundaries. Our CA starts off with all the white cells (i.e., all bits equal to zero) except the middle cell of the window, which is the bit at index 101 (array index starts at 1), which is black (i.e., one). The ECA with the above-mentioned parameters is evolved using local update 'Rule-30 ' for the "n" number of evolutions, where "n" depends on the size of the watermark image and the number of generations. From each of these n-evolutions, we collect the middle bits as shown in Figure 5, and this bit sequence is taken as the output, which forms the embedding-key matrix after layout modification from 1D to 2D. This scheme configures k number of such matrixes termed as generations of the embedding key to increase the chaoticity of the scheme while embedding.

To generate the scrambling key, the initial values of the parameters to start up ECA remain the same as the initial parameter values set for embedding-key generation, except the window size of the linear array, which is changed to be equal to the number of columns in the host image. This ECA arrangement has also evolved using Rule-30. Once the formation of the triangular structure is complete, we start stacking the output of every evolution in the downward direction for the number of iterations to construct the scrambling key, where the 'number of iterations'/ 'height of the stack' is equal to the number of rows in the host image.

It's critical to keep in note that the aforementioned CA arrangement is not a 2D CA at all, but rather the output of 1D CA organized in the form of a matrix.

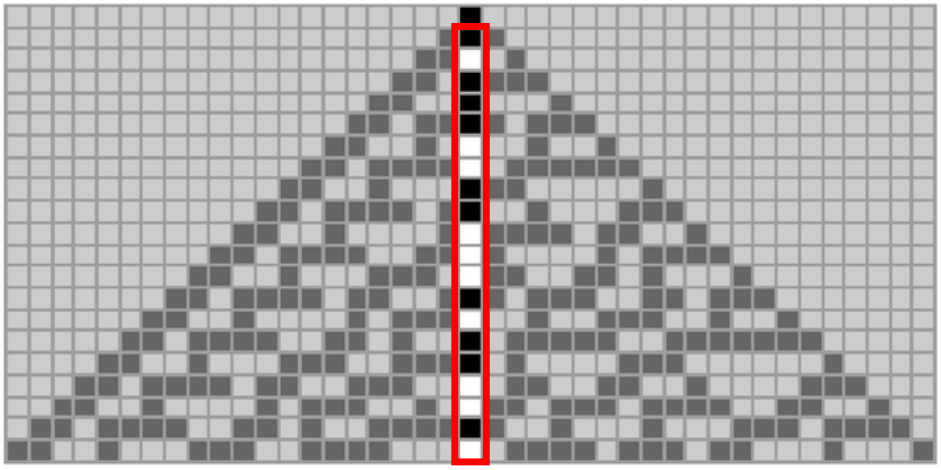


**Figure 5.** The Middle Bit Sequence from [40].

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



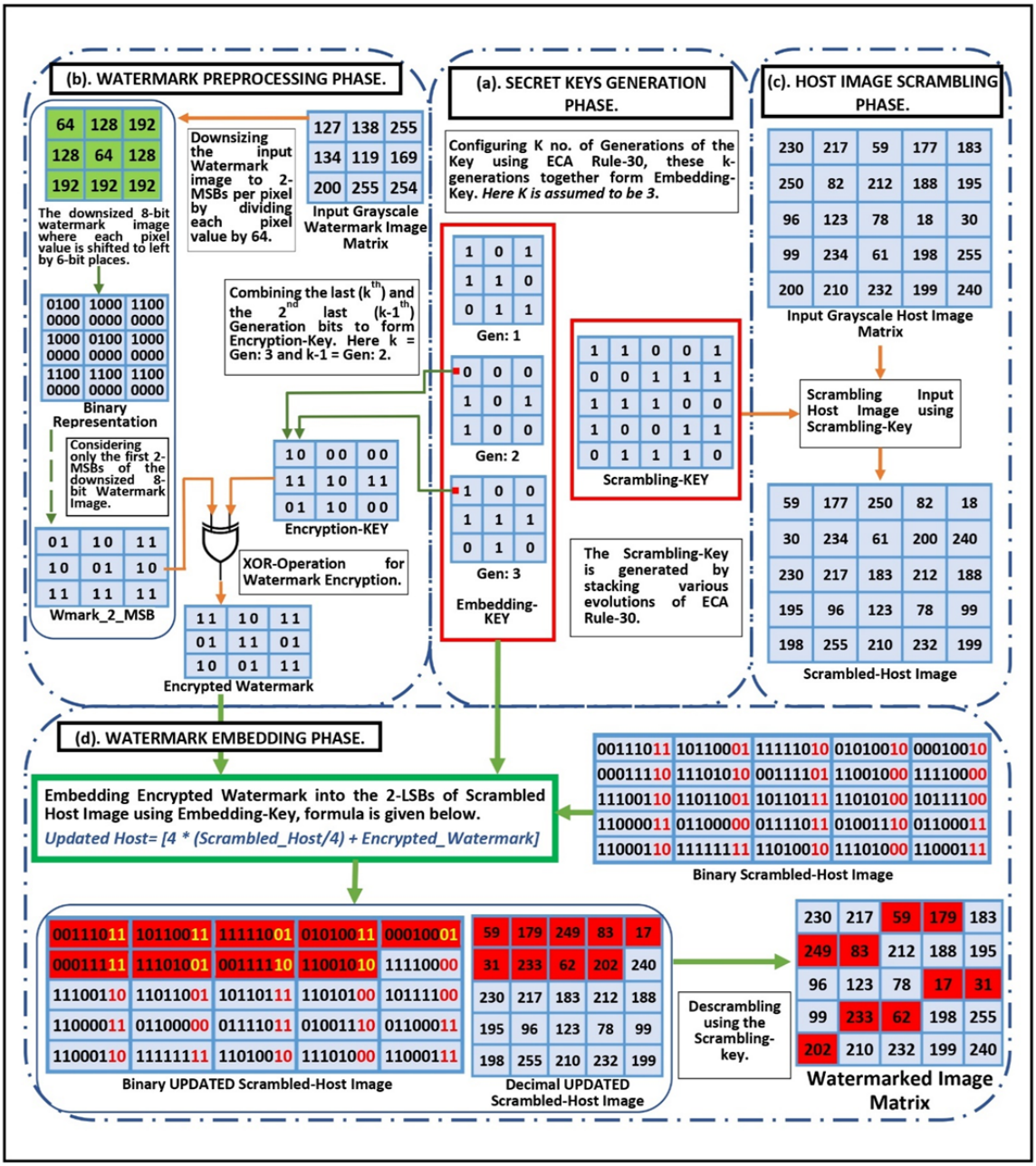



**Figure 6.** Numerical Illustration of the First Four Phases of the Watermarking Process.

3.2.2. The Watermark Preprocessing Phase.

Before inserting watermark information into the host image, watermark data should be preprocessed first. The motive of watermark data preprocessing is to downsize the grayscale watermark image to its two Most Significant Bits (MSBs) and level up the suggested system's security attribute by encrypting the watermark information before it is embedded. The watermark preprocessing phase is further divided into Downsizing the Watermark and Encrypting the Watermark as depicted in Figure 6 (b).

3.2.3. Downsizing The Watermark

To address the imperceptibility requirements of the digital image watermarking system, the suggested method does not embed the entire grayscale watermark image but rather embeds the two most significant bits of each pixel of the grayscale watermark image. To make that happen, a grayscale

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



watermark image having (Wm * Wn) number of pixels with 8 bits per pixel is downsized to 2-MSBs per pixel by simply dividing each pixel value by 64, as shown in equation (5).

$$Wmark\_2\_MSB(i,j) = \frac{Wmark(i,j)}{64} \tag{5}$$

Where Wmark (i, j) is the watermark 8-bit-pixel value at index (i, j) and Wmark_2_MSB (i, j) is the watermark pixel value reduced to 2-MSBs as division by 64 repeatedly shifts the pixel bits to the right by 6-bit places, where the six rightmost bits get discarded. Six places on the left are set to zero.

3.2.4. Encrypting Watermark

After downsizing the watermark image to 2-MSBs, it undergoes an encryption process to add another level of security to the proposed algorithm. To accomplish the encryption task, the Bitwise XOR operation is applied to the watermark data in order to hide it. The perfectly balanced nature, straightforward implementation, and low computation costs are the significant properties of the XOR operator that make it a better option to be utilized for encryption purposes. This approach forms an entirely random two-bit encryption key by simply combining every last ($k^{th}$) and the second last ($k\text{-}1^{th}$) generation bits at similar indexes of the embedding key. The construction of the encryption key is numerically depicted in Figure 6 (b). Furthermore, the bitwise XOR operation of the 2-bits of the watermark data at index (i,j) is performed with the 2-bits of the encryption key at index (i,j) as shown in equation (6), and this operation is performed on every element of the matrix.

$$Encrypted\_Watermark(i,j) = Wmark\_2\_MSB(i,j) \oplus Encryption\_Key(i,j) \tag{6}$$

3.2.5. The Host Image Scrambling Phase.

In the proposed approach, the encrypted watermark is not embedded directly into the host image pixels. Instead, the host image first gets scrambled using the scrambling key, as shown in Figure 6 (c). The motto behind scrambling the host image is to distribute the watermark pixels throughout the host image and thus achieve high robustness against geometrical attacks like cropping. In the proposed approach, the process of embedding the encrypted watermark data is performed on the LSBs of the scrambled host image, after which it is descrambled, which results in the distribution of the encrypted watermark pixels in every part of the host image that is the case of eliminating the correlation between the pixels embedded with watermark data and therefore, enhancing resistance against potential geometrical attacks. The host image scrambling process is numerically depicted in Figure 6 (c) and works according to the principle given below.

```
STEP 1:
for i=1 to RowSize do
    for j=1 to colSize do
        if scrambling_key(i,j) == 0 then
            scrambled_host(row, col) = Host_Image (i,j);
        end if
    end for
end for
STEP 2:
for i=1 to RowSize do
    for j=1 to colSize do
        if scrambling_key(i,j) == 1 then
            scrambled_host(row, col) = Host_Image (i,j);
        end if
    end for
end for
```

Row and col are initialized to 1 and accordingly incremented after every element is assigned with the pixel value in the scrambled_host matrix.

3.2.6. The Watermark Embedding Phase.

The insertion of the preprocessed watermark data into the scrambled host image using the above-generated Rule-30-based embedding key is explained in this section. The numerical illustration of the

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



watermark embedding phase is shown in Figure 6 (d). The encrypted watermark pixel values to be embedded are selected randomly based on the locations of the live cells (1s) in all the generations of the embedding-key starting from the first to the $K^{th}$ generation and after covering all the live cells if there still are watermark pixel values left to be embedded then these values are also selected randomly based on the locations of the dead cells (0s) in the embedding-key starting again from the first to $K^{th}$ generation. In the proposed scheme, the selected two-bit watermark data is inserted into the two-LSBs of the host image pixel values using the addition operation; therefore, it becomes obligatory to clear the previous values of the two-LSBs of the selected host image pixels and set them to zero which is achieved by first dividing and then multiplying each of them by four. The scrambled host image pixel values into which watermark data is to be embedded are selected in a sequential manner starting from the first pixel to the '$n^{th}$' pixel where n = (Wm * Wn), i.e., the length of the watermark image. The pixels forming the selected sequence are not correlated at all as the host image is already scrambled and thus robust against the potential geometrical attacks. Subsequent to the insertion of all the 2-bit watermark pixel values, the updated scrambled host image is descrambled using the scrambling key that finally produces the watermarked image. The complete embedding process is described using pseudocode notation in the next section.

3.2.7. The Pseudocode Description of the Watermark Embedding Process.

The Algorithm 1 presents the pseudocode of the watermark embedding process of the proposed CA-based digital image watermarking scheme.

**Algorithm 1** Watermark Embedding Algorithm

**Input:** Host (Hm*Hn), Scrambling_key (Hm*Hn), Embedding_key (Wm*Wn), and Watermark(Wm*Wn).

**Output:** Watermarked_Image.

**procedure** EMBEDDING(*HostImage*, *Scrambling_key*, *Embedding_key*, *Watermark*)
    $Wmark\_2\_MSB \leftarrow Downsizing(Watermark)$;
    $Encrypted\_Watermark \leftarrow bitxor(Wmark\_2\_MSB, Encryption\_key)$;
    $Scrambled\_Host \leftarrow Scrambling(Host, Scrambling\_key)$; ▷ The (Wm*Wn) Embedding_key consists of K (Wm*Wn) generations (Gen1, Gen2, Gen3 ..... GenK).
    $row \leftarrow 1; col \leftarrow 1;$ ▷ Covering the locations of the live cells (1s) in Gen1.
    **for** $i \leftarrow 1$ to $Wm$ **do**
        **for** $j \leftarrow 1$ to $Wn$ **do**
            **if** (Gen1[i][j] $\equiv 1$) **then**
                $temp := (floor(Scrambled\_Host[row][col]/4) * 4);$ ▷ The two MSBs of the encrypted watermark at index (i, j) are added to the two LSBs of the scrambled host image pixel at index (row, col).
                $Updated\_Scrambled\_Host[row][col] := temp + Encrypted\_Watermark[i][j];$
                $col := col + 1;$
            **end if**
        **end for**
    **end for** ▷ Covering the locations of the live cells in GenT, where T = 2, 3, 4...K and S = 1, 2, 3...T-1.
    **for** $i \leftarrow 1$ to $Wm$ **do**
        **for** $j \leftarrow 1$ to $Wn$ **do**
            **if** $(GenT[i][j] \equiv 1)$**and**$(GenS[i][j] \neq 1)$ **then**
                $temp := (floor(Scrambled\_Host[row][col]/4) * 4);$
                $Updated\_Scrambled\_Host[row][col] := temp + Encrypted\_Watermark[i][j];$
                $col := col + 1;$
            **end if**
        **end for**
    **end for** ▷ After covering all live cells, the pixel values at dead cells (0s) are selected
    **for** $i \leftarrow 1$ to $Wm$ **do**
        **for** $j \leftarrow 1$ to $Wn$ **do**
            **if** $(GenT[i][j] = 0)$**and**$(GenS[i][j] \neq 1)$ **then**
                ▷ $WhereT = 1, 2, 3...K; S = 1, 2, 3...KandS \neq T.$
                $temp := (floor(Scrambled\_Host[row][col]/4) * 4);$
                $Updated\_Scrambled\_Host[row][col] := temp + Encrypted\_Watermark[i][j];$
                $col := col + 1;$
            **end if**
        **end for**
    **end for** ▷ Till all the watermark image bits are embedded.
    Watermarked_Image := Descrambling (Updated_Scrambled_Host, Scrambling_key)
    **return** $Watermarked\_Image;$
**end procedure**

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



3.2.8. The Watermark Extraction Phase.

To retrieve the watermark image that was embedded using a specific technique, the inverse procedure of that technique is carried out, called watermark extraction. In the proposed scheme to extract the watermark, the watermarked image is first scrambled using the same scrambling key that was used during embedding. Then, from the scrambled watermarked image, the sets of 2-LSBs of the first n pixels are extracted where n = (Wm*Wn). These extracted n-sets of 2-LSBs are arranged in a (Wm X Wn) matrix based on the locations of all the live cells (1s) first and then the locations of the dead cells (0s) in all the generations of the embedding key. However, the retrieved watermark data is still encrypted and, therefore, undergoes a decryption process in which the Bitwise XOR operation of extracted watermark data with the encryption key is performed. Following decryption, each 2-bit pixel is shifted to the left by 6-bit places in order to get converted to the 8-bit pixel, as a result, extracted LSBs become the MSBs of the extracted watermark image after six left bit-shifts. The numerical illustration of the complete watermark extraction phase is shown in Figure 7.

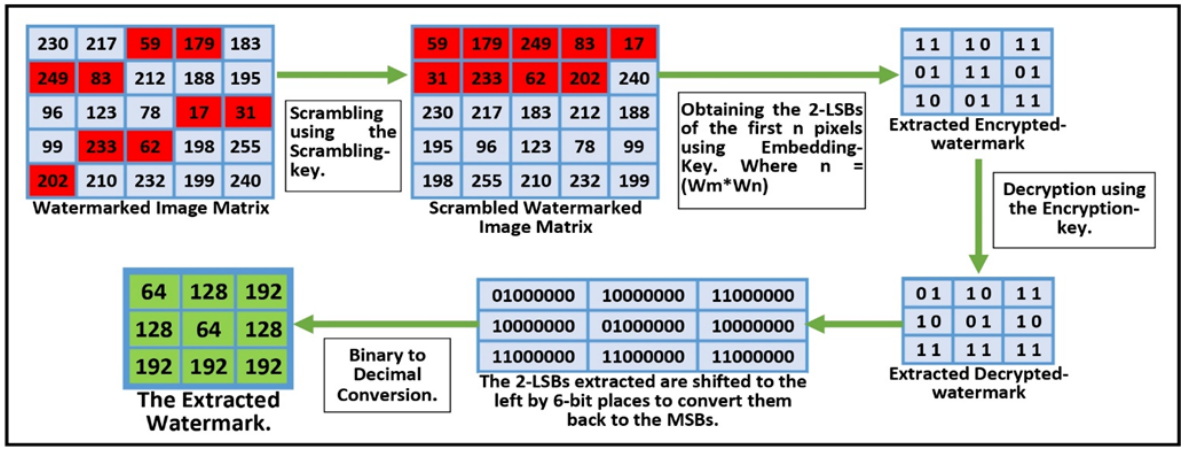



**Figure 7.** Numerical Illustration of the Watermark Extraction Phase.

## 4. Exploring the challenges of balancing robustness, imperceptibility, and capacity.

Achieving an ideal balance among robustness, imperceptibility, and capacity is a central challenge in various fields, particularly in areas like digital watermarking, steganography, and information hiding. Let's delve into the discussion on the challenges and limitations associated with this balance:

- Achieving high robustness is crucial for ensuring that the embedded data remains intact even after processing or transmission. However, enhancing robustness often comes at the expense of imperceptibility and capacity. The challenge lies in finding techniques that strike the right balance between robustness and other factors.
- Preserving imperceptibility is crucial to avoid the discovery of concealed information by unintended parties or adversaries. However, enhancing imperceptibility may restrict the quantity of embeddable data or compromise robustness against specific attacks.
- Expanding capacity enables the embedding of larger amounts of data, facilitating applications like data authentication, copyright protection, and covert communication. Nonetheless, increasing capacity typically involves a trade-off between imperceptibility and robustness. High-capacity methods can introduce detectable distortions or vulnerabilities to attacks.
- A significant challenge involves managing the trade-offs among these conflicting factors. Achieving the ideal balance demands meticulous attention to the particular application needs, security considerations, and the attributes of the underlying data and cover medium.
- Striking the right balance also entails assessing the trade-offs between security and usability and adjusting embedding techniques accordingly. This may involve fine-tuning parameters or choosing alternative methods tailored to the application's requirements. Ultimately, the aim is to

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



achieve an ideal compromise that ensures adequate security without sacrificing usability or data fidelity.
- Technological advancements continually introduce new challenges and opportunities in achieving the balance between robustness, imperceptibility, and capacity. Keeping pace with these advancements requires ongoing research and innovation to develop robust and efficient techniques that address the evolving landscape of threats and constraints.

In summary, achieving an ideal balance among robustness, imperceptibility, and capacity is a complex and multifaceted challenge that requires careful consideration of trade-offs, application requirements, and technological constraints. Despite the inherent limitations, ongoing research and advancements offer promising avenues for improving the effectiveness and efficiency of watermarking techniques in various domains.

## 5. Performance Analysis and Experimental Discussion.

In this section, the effectiveness of the proposed algorithm is analyzed. The performance of the suggested scheme is evaluated by computing the various robustness and imperceptibility quality metrics and comparing the results with several existing algorithms [9,10,24].

### *5.1. Experimental Setup.*

The proposed watermarking scheme is implemented on the computer system with the following configuration: Intel Core i5-2410M CPU @ 2.30 GHz, 4.00 GB RAM, and Windows 7 Ultimate 64-bit SP1 operating system. It is designed, programmed, and analyzed using the MATLAB R2020a [41] platform. The grayscale logo image of size 128x64 is considered the watermark image while testing the proposed approach. The performance testing of the suggested scheme is performed on the dataset of 42 test host images shown in Figure 8. These test images are collected from various standard image databases, which are USC-SIPI [42], The Cell Image Library [43,44], The Cancer Imaging Archive (TCIA) [45], STARE [46], National Archives Catalog [47], NASA [48], and MATLAB Toolbox [41]. Some of the images in the dataset were originally color images and, therefore, required conversion to grayscale with a bit depth of eight, and that was achieved using MATLAB built-in image type conversion functions. The proposed algorithm was tested on the test host images of different dimensions and formats like png, tif, tiff, jpg, etc. The original size of the test images was not altered at all except for those from the NASA library, in which the size of huge images was reduced. To determine the efficiency of the suggested watermark system, its test results were compared with several existing systems.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



**Table 1.** Description and dimensions of our dataset comprising forty-two test images sourced from diverse databases.

| DATABASE | Grayscale Test Image Name and ID. | Test Image Size |
|---|---|---|
| **USC-SIPI [42]** | House (4.1.05) | 256x256 |
| | Jelly beans (4.1.08) | 256x256 |
| | Fishing Boat (boat.512) | 512x512 |
| | Splash (4.2.01) | 512x512 |
| | Mandrill [a.k.a. Baboon] (4.2.03) | 512x512 |
| | Peppers (4.2.07) | 512x512 |
| | Stream and bridge (5.2.10) | 512x512 |
| | Truck (7.1.01) | 512x512 |
| | Pentagon (3.2.25) | 1024x1024 |
| | Male (5.3.01) | 1024x1024 |
| **Cell Image Library [43,44]** | W9CCDB54 | 512x512 |
| | W9CIL54816 | 524x581 |
| **Cancer Imaging Archive [45]** | Brain MRI 55 | 256x256 |
| | Brain MRI 63 | 256x256 |
| | Brain MRI 70 | 256x256 |
| | Chest CT 30 | 256x256 |
| | Chest CT 60 | 256x256 |
| | Chest CT 90 | 256x256 |
| **STARE [46]** | im0001 | 700x605 |
| | im0100 | 700x605 |
| | im0200 | 700x605 |
| | im0300 | 700x605 |
| | im0400 | 700x605 |
| **National Archives Catalog [47]** | Galen Clark (NAID: 2581374) | 576x706 |
| | General Douglas (NAID: 2595319) | 576x712 |
| | Turbine Power House (NAID: 1633561) | 576x718 |
| | Shipyard (NAID: 138930743) | 1024x1024 |
| **NASA [48]** | Full Moon (NASA ID: as08-14-2505) | 256x256 |
| | Venus (NASA ID: ARC-1981-A78-9176) | 256x256 |
| | Astronaut L. Gordon Cooper (NASA ID: jsc2013e076221) | 512x512 |
| | Panoramic view (NASA ID: as15-85-11363) | 512x512 |
| | Television (TV) monitor (NASA ID: S71-41509) | 512x512 |
| | Wallops Island (NASA ID: LRC-1960-B701_P-00652) | 512x512 |
| | Group Photo (NASA ID: E-14754) | 1024x1024 |
| **MATLAB Toolbox [41]** | onion | 198x135 |
| | pout | 240x291 |
| | cameraman | 256x256 |
| | forest | 447x301 |
| | spine | 490x367 |
| | lighthouse | 480x640 |
| | fabric | 640x480 |
| | flamingos | 1296x972 |

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



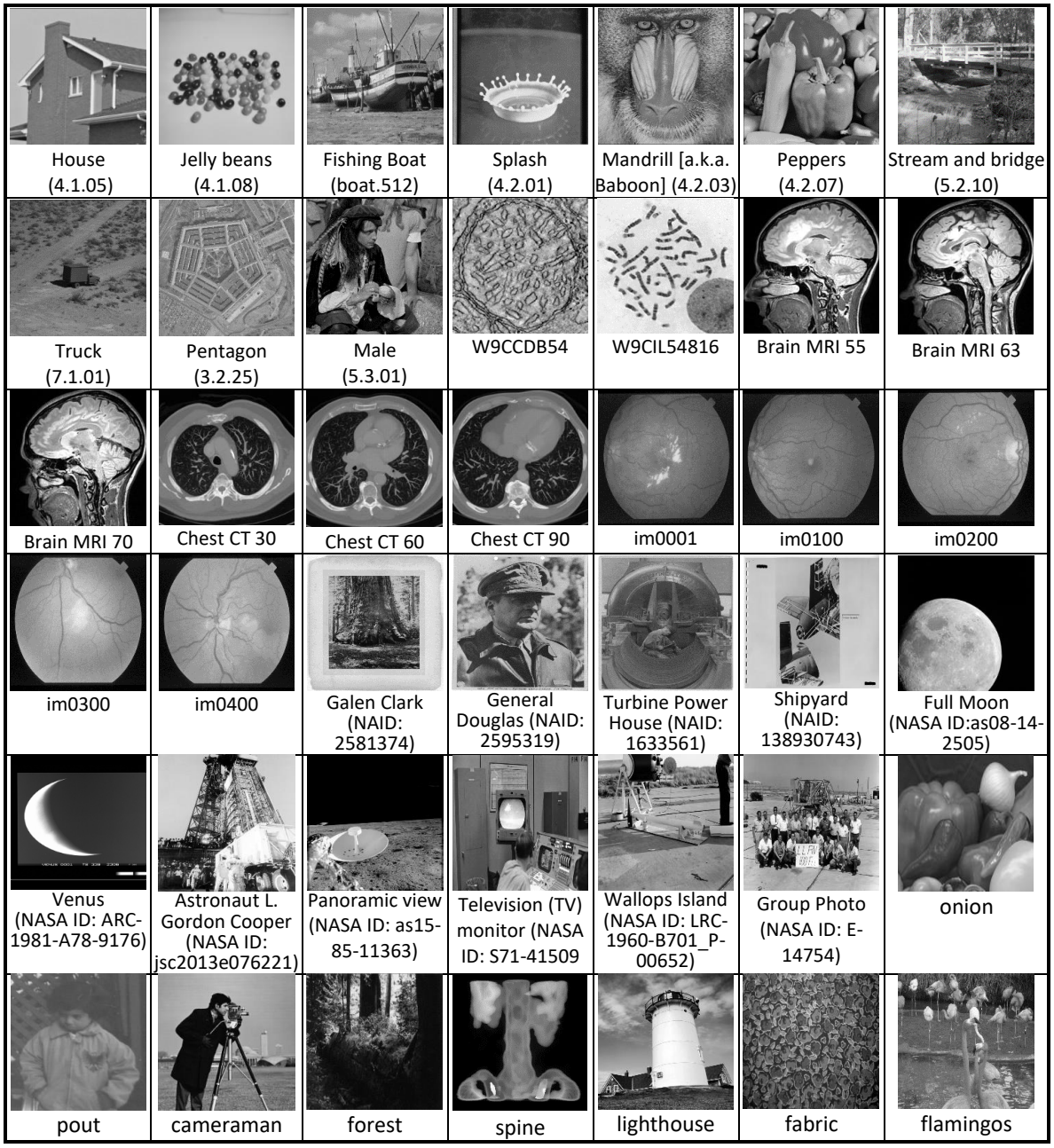



**Figure 8.** The dataset of 42 test host images was used for performance evaluation.

*5.2. Performance Evaluation Metrics.*

The most significant attributes, imperceptibility, robustness, and payload of the digital image watermarking system are computed by various image quality metrics that are given below. In contrast, when calculating the evaluation metrics, the original host image is used as a reference image to determine the quality of the watermarked images.

5.2.1. Imperceptibility.

Imperceptibility refers to the notion that after embedding watermark data into the host image, its perceptual quality should be identical to the original host image. The five below-mentioned performance evaluation metrics are calculated to evaluate the imperceptibility attribute of the proposed watermarking system.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



- Mean-Squared Error (MSE): The averaged intensity between the original host and the watermarked image is calculated using Mean Square Error. It gauges the degree to which a pixel varies from its original state. The smaller MSE value signifies that the watermarked image resembles the original host image. Equation (7) is used to determine MSE.

$$MSE = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}(HI_{(m,n)} - WI_{(m,n)})^2 \tag{7}$$

  Where $HI_{(m,n)}$ and $WI_{(m,n)}$ denote the pixel values at index (m,n) in the original host image and the watermarked image, respectively, and MxN is the size of the images.

- Root Mean-Squared Error (RMSE): It is a quality assessment metric that is used for the error magnitude evaluation. It is derived by simply square rooting the MSE as illustrated in equation (8).

$$RMSE = \sqrt{MSE} \tag{8}$$

- Peak Signal-to-Noise Ratio (PSNR): The well-known image quality metric widely used to evaluate the perceptual quality of the watermarked images with reference to the original host images is PSNR. It is derived from the MSE and is expressed as the ratio of the maximum pixel intensity to the power of the distortion. The PSNR value should be at least greater than 35 dB; the higher PSNR value denotes better imperceptibility. Equation (9) is used to determine PSNR.

$$PSNR = 10\log_{10}\frac{255^2}{MSE} \tag{9}$$

- Structural Similarity (SSIM) Index: The perceptual quality assessment metric that is used to measure the similarity between the original host image and the watermarked image is SSIM. The watermarked image has great perceptual quality if the SSIM value is close to 1. The SSIM of the watermarked image with reference to the original host image is determined using equation (10).

$$SSIM(HI, WI) = \frac{(2\mu_{HI}\mu_{WI} + C1)(2\sigma_{HIWI} + C2)}{(\mu_{HI}^2 + \mu_{WI}^2 + C1)(\sigma_{HI}^2 + \sigma_{WI}^2 + C2)} \tag{10}$$

  Where $\mu_{HI}$ and $\mu_{WI}$ are the averages, $\sigma_{HI}^2$ and $\sigma_{WI}^2$ are the variances, and $\sigma_{HIWI}$ is the covariance of the original host image and watermarked image, respectively. C1 = $(k1L)^2$ and C2 = $(k2L)^2$; L = $(2^{Bits/Pixel} - 1)$, k1 = 0.01 and k2 = 0.03.

- Universal Quality Index (Q-Index): The distortion within an image is determined by the Q-Index. The range of the Q-Index is [-1 to 1], and its best possible value can be 1, indicating that the images are identical. The three parameters required to calculate the Q-Index are correlation, luminance, and contrast, which are calculated using equation (11).

$$Q = \frac{\sigma_{HIWI}}{\sigma_{HI}.\sigma_{WI}} \cdot \frac{2\overline{HI}.\overline{WI}}{(\overline{HI})^2 + (\overline{WI})^2} \cdot \frac{(2\sigma_{HI}.\sigma_{WI}}{(\sigma_{HI}^2.\sigma_{WI}^2)} \tag{11}$$

5.2.2. Robustness.

The robustness of the image watermarking system is a measure of the ability of the embedded watermark to resist and remain unaffected by various intentional and unintentional attacks, thus implying the system's reliability. To evaluate the robustness attribute of the proposed watermarking system, the original watermark is compared to the extracted watermark using the below-mentioned performance evaluation metrics.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



- Correlation Coefficient (CC): When evaluating a watermarking scheme's robustness to various attacks and transformations, the correlation coefficient is an essential statistical measure to consider. It gives information about how well the system will maintain the watermark's integrity and tolerate changes while retaining the ability to allow accurate watermark extraction. The correlation coefficient quantifies the intensity and direction of the linear relationship between the original watermark and the extracted watermark. Equation (12) can be used to determine the correlation coefficient value, which ranges from 0 to 1.

$$CC(OW, EW) = \frac{\sum_{(m=1)}^{M} \sum_{(n=1)}^{N} (OW_{(m,n)} - \overline{OW})(EW_{(m,n)} - \overline{EW}))}{\sqrt{(\sum_{(m=1)}^{M} \sum_{(n=1)}^{N} (OW_{(m,n)} - \overline{OW})^2)(\sum_{(m=1)}^{M} \sum_{(n=1)}^{N} (EW_{(m,n)} - \overline{EW})^2)}} \quad (12)$$

  Where $OW_{(m,n)}$ and $EW_{(m,n)}$ denote the pixel values at index (m,n) in the original watermark image and the extracted watermark image, respectively. The $\overline{OW}$ is the mean of the original watermark image, and the $\overline{EW}$ is the mean of the extracted watermark image.

- Bit Error Ratio (BER): The ratio of the total number of errored/corrupted bits to the total number of bits in the image is referred to as the bit error ratio and is calculated using equation (13).

$$BER = \frac{(TotalErroredBits)}{(TotalBits)} \quad (13)$$

## 6. Comparative Analysis.

The performance of the suggested method in terms of robustness and imperceptibility is discussed in this section.

### *6.1. Imperceptibility Analysis.*

The imperceptibility results of the proposed watermarking scheme are compared to the current state-of-the-art cellular automata-based methods to analyze the relative performance of the proposed scheme. Table 2 includes the multiple image quality metrics to determine the effectiveness of different image watermarking methods in terms of their imperceptibility. The values of these metrics are compared across multiple techniques, namely Ye and Li [24], Adwan et al. [9], Moniruzzaman et al. [10], and the Proposed Scheme. The lower MSE and RMSE values and the higher PSNR, SSIM, and Q-Index values are indicators of the superior perceptual quality of an image. It has been determined that among the evaluated techniques, the proposed scheme consistently outperforms the other assessed schemes in all metrics. It repeatedly results in the lowest MSE values for all the test images, reflecting higher accuracy in watermarked images. The proposed scheme also emphasizes its potential to reduce the average difference between the original and watermarked images by having the lowest RMSE values. Regarding PSNR, the proposed scheme consistently performs better than the other techniques. It can be observed that greater PSNR values are constantly obtained for all the test images in the dataset by the proposed Scheme, which implies greater retention of image details and less quality loss than the other techniques under evaluation. The higher SSIM results further demonstrate the positive aspects of the proposed scheme in retaining the image's structural information and its capacity to preserve the visual composition and attributes of the original images. Finally, the higher values of the perceptual quality metric Q-Index indicate that the proposed scheme generates watermarked images with better overall quality.

After accounting for all the given parameters, we can certainly infer that the proposed scheme demonstrates strong performance in the grayscale image watermarking process. It consistently performs better than the other techniques (Ye and Li [24], Adwan et al. [9], and Moniruzzaman et al. [10]) in terms of lower MSE and RMSE values, and higher PSNR, SSIM, and Q-Index values thus assuring the superior perceptual quality of the watermarked image. The proposed scheme can

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



be seen as a reliable and high perceptual quality approach for performing digital grayscale image watermarking tasks.

*6.2. Robustness Analysis.*

This section provides a comprehensive assessment of the robustness of the proposed scheme when subjected to an extensive spectrum of deliberate attacks. The proposed watermarking algorithm has undergone a meticulous evaluation on an extensive dataset comprising 42 diverse test images, aimed at assessing its robustness against a wide spectrum of potential attacks, with the evaluation criterion being the Correlation Coefficient (CC), the Number of Erroneous Bits (NEB), and the Bit Error Ratio (BER). The consequential findings, stemming from the execution of multifarious attack scenarios upon the watermarked images, are comprehensively documented in Tables 3 and 4 for detailed examination and analysis.

6.2.1. Robustness Against Cropping Attacks.

An intentional removal or trimming of portions of a digital image with the goal of changing its content or context—possibly leading to misinterpretation or removal of copyright information and watermarks, known as image cropping attacks. It is a kind of modification that compromises the integrity of an image and can vary in scale and intent. Here are the mathematical representations of the cropping attacks at different segments of the watermarked image $I$ with dimensions $M$ x $N$:

The coordinates representing the top-left corner can be denoted as $(x_{start}, y_{start})$ and the dimensions of the cropping region as W×H. The cropped image $I_{crop}$ can be obtained as follows:

$$I_{crop} = I(x_{start} : W,\ y_{start} : H) \tag{14}$$

The coordinates representing the top-right corner can be denoted as $(x_{end}, y_{start})$ and dimensions W×H, then the cropped image $I_{crop}$ crop can be obtained as follows:

$$I_{crop} = I(x_{end} - W + 1 : x_{end},\ y_{start} : H) \tag{15}$$

The coordinates representing the bottom-left corner can be denoted as $(x_{start}, y_{end})$ and dimensions W×H, then the cropped image $I_{crop}$ crop can be obtained as follows:

$$I_{crop} = I(x_{start} : W,\ y_{end} - H + 1 : y_{end}) \tag{16}$$

For the bottom-right corner, the coordinates can be denoted as $(x_{end}, y_{end})$ and the formula to extract the cropped region would be:

$$I_{crop} = I(x_{end} - W + 1 : x_{end},\ y_{end} - H + 1 : y_{end}) \tag{17}$$

When cropping an image uniformly from all four sides, we usually specify a rectangular region for extraction. This rectangle is defined by providing two coordinates: the top-left corner and the bottom-right corner. Given the top-left corner's coordinates as $(x_{start}, y_{start})$, and the bottom-right corner's coordinates as $(x_{end}, y_{end})$, the width of the cropping region is $W = x_{end} - x_{start} + 1$, and the height is $H = y_{end} - y_{start} + 1$. The formula to extract the cropped region is:

$$I_{crop} = I(x_{start} : W,\ y_{start} : H) \tag{18}$$

where $W > 0$and $H > 0$.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



This analysis aims to evaluate the robustness of a proposed watermarking technique under various cropping scenarios. The watermarked images are manipulated by applying different cropping percentages such as 6%, 10%, 20%, and 35% at different segments like top-left, top-right, bottom-left, bottom-right, center and all the sides of the watermarked images. The impact of these cropping scenarios on the extracted watermark quality is assessed, specifically considering correlation coefficients and bit error ratios. The analysis, according to Table 3, underscores that the proposed watermarking technique is highly robust as it maintains a high correlation coefficient even with substantial cropping. It also demonstrates that the proposed scheme is effective at preserving watermark integrity across different cropping scenarios.

6.2.2. Robustness Against Noise Attacks.

A noise attack denotes the unwanted introduction of stochastic perturbations, or "noise," into watermarked images, leading to a degradation in image fidelity and perceptual acuity. Noise manifests diversely, manifesting as erratic pixel fluctuations within images, the emergence of granular patterns, sporadic conspicuous anomalies in luminosity, or the distortion of chromatic properties. The incursion of noise can be attributed to various sources, encompassing electromagnetic interference, the propagation of signals with imperfections or distortions, or the inherent limitations of data acquisition apparatuses. The watermarked images were subjected to the addition of Speckle and Salt & Pepper noises in order to assess the proposed watermarking scheme's robustness against noise attacks. The salt & pepper noises with varying noise densities and multiple variances of the speckle noise were introduced into the watermarked images to evaluate the proposed approach. Table 4 shows the results for the salt & pepper noise with noise density = 0.01 and speckle noise with variance = 0.01 and the outcomes imply that the proposed scheme is very robust to these kind of attacks.

6.2.3. Robustness Against Sharpening Attacks.

The image processing procedure designed to heighten image acuity and refine its visual details is sharpening. When it comes to digital image alteration, a "sharpening attack" is an intentional attempt to utilize image sharpening techniques to make a watermark on an image less visible or effective. Such an attack usually aims to hide, modify, or remove a watermark (such as a logo or copyright notice) that has been superimposed on an image in order to preserve its ownership or copyright. Enhancing the contrast and sharpness of the image might make the watermark less visible or perhaps unreadable. To determine how robust the proposed strategy is, the impact of sharpening attacks on the extracted watermark quality is evaluated. The sharpening attacks of varying strengths i.e, multiple values for amount variable at multiple radiuses are applied to the watermark images. The study based on Table 4 highlights how robust the suggested watermarking method is since it keeps a high correlation coefficient even when there is significant sharpening. It also shows that the suggested scheme works well to maintain watermark integrity under various sharpening conditions.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



**Table 2.** Comparison in terms of imperceptibility of the proposed image watermarking scheme with several existing Cellular Automata-based image watermarking schemes.

| Host Image Grayscale (8 bits/pixel) | Ye and Li. [24] | | | | | Adwan et al. [9] | | | | | Moniruzzaman et al. [10] | | | | | Proposed Scheme. | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MSE | RMSE | PSNR | SSIM | Q-Index | MSE | RMSE | PSNR | SSIM | Q-Index | MSE | RMSE | PSNR | SSIM | Q-Index | MSE | RMSE | PSNR | SSIM | Q-Index |
| House | 706.788 | 26.5855 | 19.6379 | 0.6311 | 0.9946 | 3.5433 | 1.8824 | 42.6368 | 0.988 | 0.9998 | 0.4315 | 0.6569 | 51.7808 | **0.9965** | **1** | **0.3165** | **0.5626** | **53.1273** | 0.9963 | **1** |
| Jelly beans | 807.4062 | 28.4149 | 19.0599 | 0.6258 | 0.9938 | 3.4273 | 1.8513 | 42.7813 | 0.9884 | 0.9999 | 0.4323 | 0.6575 | 51.7727 | 0.9956 | **1** | **0.3075** | **0.5545** | **53.2526** | **0.9957** | **1** |
| Fishing Boat | 107.4458 | 10.3656 | 27.8189 | 0.9345 | 0.9994 | 3.5374 | 1.8808 | 42.6439 | 0.992 | 0.9978 | 0.1081 | 0.3289 | 57.7907 | **0.9994** | 0.9999 | **0.0798** | **0.2824** | **59.1122** | 0.9993 | **1** |
| Splash | 219.654 | 14.8207 | 24.7134 | 0.8769 | 0.9843 | 3.4796 | 1.8654 | 42.7155 | 0.9871 | 0.9898 | 0.1076 | 0.328 | 57.8126 | 0.9989 | 0.9987 | **0.0785** | **0.2801** | **59.1843** | **0.9994** | **1** |
| Mandrill | 142.4066 | 11.9334 | 26.5955 | 0.9513 | 0.9984 | 3.4968 | 1.87 | 42.6941 | 0.9967 | 0.9998 | 0.1094 | 0.3308 | 57.7394 | 0.9997 | **1** | **0.0787** | **0.2805** | **59.1716** | **1** | **1** |
| Peppers | 134.3881 | 11.5926 | 26.8472 | 0.9049 | 0.9965 | 3.4876 | 1.8675 | 42.7056 | 0.9901 | 0.9994 | 0.1067 | 0.3266 | 57.8505 | 0.9992 | **1** | **0.0776** | **0.2785** | **59.2346** | **0.9995** | **1** |
| Stream & bridge | 154.5128 | 12.4303 | 26.2412 | 0.9341 | 0.9985 | 2.3214 | 1.5236 | 44.4734 | 0.9997 | 0.9999 | 0.0948 | 0.3079 | 58.3621 | 0.9997 | **1** | **0.0501** | **0.2238** | **61.1339** | **0.9999** | **1** |
| Truck | 123.524 | 11.1141 | 27.2133 | 0.9025 | 0.9982 | 3.1032 | 1.7616 | 43.2127 | 0.9928 | 0.9997 | 0.1098 | 0.3314 | 57.7247 | 0.9993 | **1** | **0.0742** | **0.2725** | **59.4246** | **0.9997** | **1** |
| Pentagon | 24.4735 | 4.9471 | 34.2438 | 0.9845 | 0.9999 | 3.3315 | 1.8252 | 42.9044 | 0.9936 | 0.9999 | 0.0281 | 0.1678 | 63.6373 | **0.9999** | **1** | **0.0196** | **0.1401** | **65.2026** | **0.9999** | **1** |
| Male | 49.4194 | 7.0299 | 31.1918 | 0.9724 | 0.9968 | 3.3181 | 1.8216 | 42.922 | 0.9857 | 0.9742 | 0.0273 | 0.1652 | 63.7709 | 0.9998 | 0.9995 | **0.0191** | **0.1381** | **65.3259** | **0.9999** | **1** |
| W9CCDB54 | 141.9224 | 11.9131 | 26.6103 | 0.9414 | 0.9992 | 3.5038 | 1.8718 | 42.6854 | 0.9974 | 0.9999 | 0.1065 | 0.3264 | 57.8561 | 0.9998 | **1** | **0.0773** | **0.2781** | **59.2483** | **0.9999** | **1** |
| W9CIL54816 | 418.3748 | 20.4542 | 21.9151 | 0.8796 | 0.9982 | 3.4951 | 1.8695 | 42.6962 | 0.9879 | 0.9999 | N/A | N/A | N/A | N/A | N/A | **0.0665** | **0.2579** | **59.9031** | **0.9992** | **1** |
| Brain MRI 55 | 1321.8654 | 36.3575 | 16.9189 | 0.6654 | 0.8857 | 3.5439 | 1.8825 | 42.636 | 0.9683 | 0.9053 | 0.4535 | 0.6734 | 51.5653 | **0.9969** | **0.9716** | **0.307** | **0.554** | **53.26** | 0.9965 | 0.9654 |
| Brain MRI 63 | 1371.5278 | 37.0341 | 16.7588 | 0.6728 | 0.8771 | 3.5334 | 1.8797 | 42.6489 | 0.9673 | 0.9039 | 0.4483 | 0.6695 | 51.6152 | **0.997** | **0.9732** | **0.3125** | **0.559** | **53.1827** | 0.9968 | 0.9673 |
| Brain MRI 70 | 1240.2223 | 35.2168 | 17.1958 | 0.6783 | 0.8869 | 3.4719 | 1.8633 | 42.7252 | 0.9656 | 0.8965 | 0.4599 | 0.6782 | 51.5037 | 0.9968 | 0.9675 | **0.3096** | **0.5564** | **53.2224** | **0.9969** | **0.973** |
| Chest CT 30 | 1195.4414 | 34.5752 | 17.3555 | 0.6182 | 0.8647 | 3.4924 | 1.8688 | 42.6995 | 0.9861 | 0.9708 | 0.4261 | 0.6528 | 51.8355 | **0.9973** | **0.9935** | **0.3117** | **0.5583** | **53.1929** | 0.9969 | 0.9836 |
| Chest CT 60 | 1008.4899 | 31.7567 | 18.0941 | 0.6612 | 0.9002 | 3.4553 | 1.8588 | 42.746 | 0.9886 | 0.9861 | 0.4167 | 0.6456 | 51.9321 | 0.9975 | 0.9931 | **0.3088** | **0.5557** | **53.2341** | **0.9977** | **0.9951** |
| Chest CT 90 | 906.1802 | 30.1028 | 18.5587 | 0.6429 | 0.926 | 3.4841 | 1.8666 | 42.7099 | 0.9898 | 0.9867 | 0.4278 | 0.654 | 51.8186 | **0.9976** | **0.9981** | **0.3123** | **0.5588** | **53.1855** | **0.9976** | 0.9929 |
| im0001 | 116.7653 | 10.8058 | 27.4577 | 0.9203 | 0.9884 | 3.5229 | 1.8769 | 42.6618 | 0.9808 | 0.9886 | N/A | N/A | N/A | N/A | N/A | **0.0498** | **0.2232** | **61.1577** | **0.9994** | **0.9944** |
| im0100 | 142.0792 | 11.9197 | 26.6055 | 0.9165 | 0.983 | 3.4819 | 1.866 | 42.7126 | 0.9815 | 0.9882 | N/A | N/A | N/A | N/A | N/A | **0.0477** | **0.2183** | **61.3478** | **0.9994** | **0.9949** |
| im0200 | 110.341 | 10.5043 | 27.7034 | 0.9375 | 0.9912 | 3.4785 | 1.8651 | 42.7169 | 0.9814 | 0.9779 | N/A | N/A | N/A | N/A | N/A | **0.0485** | **0.2203** | **61.2691** | **0.9995** | **0.9987** |
| im0300 | 165.518 | 12.8654 | 25.9424 | 0.9307 | 0.9831 | 3.5117 | 1.874 | 42.6756 | 0.9795 | 0.9819 | N/A | N/A | N/A | N/A | N/A | **0.0484** | **0.2199** | **61.2846** | **0.9995** | **0.9995** |
| im0400 | 187.5904 | 13.6964 | 25.3987 | 0.9216 | 0.9767 | 3.5083 | 1.873 | 42.6799 | 0.9814 | 0.9882 | N/A | N/A | N/A | N/A | N/A | **0.0483** | **0.2199** | **61.2875** | **0.9994** | **0.9967** |
| Galen Clark | 219.0054 | 14.7988 | 24.7263 | 0.9173 | 0.9988 | 3.4882 | 1.8677 | 42.7048 | 0.9903 | 0.9996 | N/A | N/A | N/A | N/A | N/A | **0.0493** | **0.222** | **61.2026** | **0.9995** | **1** |
| General Douglas | 126.4171 | 11.2435 | 27.1127 | 0.9361 | 0.9987 | 3.5261 | 1.8778 | 42.6579 | 0.9899 | 0.9997 | N/A | N/A | N/A | N/A | N/A | **0.0495** | **0.2225** | **61.1849** | **0.9995** | **1** |
| Turbine Powerhouse | 81.485 | 9.0269 | 29.02 | 0.9484 | 0.9996 | 3.5431 | 1.8823 | 42.637 | 0.991 | 0.9997 | N/A | N/A | N/A | N/A | N/A | **0.0491** | **0.2215** | **61.2241** | **0.9996** | **1** |
| Shipyard | 82.3662 | 9.0756 | 28.9733 | 0.9646 | 0.9986 | 3.4589 | 1.8598 | 42.7414 | 0.9863 | 0.9991 | 0.027 | 0.1644 | 63.8128 | **0.9997** | 0.9999 | **0.0187** | **0.1368** | **65.4111** | **0.9997** | **1** |
| Full Moon | 1932.9835 | 43.9657 | 15.2685 | 0.5331 | 0.5945 | 3.4339 | 1.8531 | 42.7729 | 0.8521 | 0.4952 | 0.3377 | 0.5811 | 52.846 | 0.9951 | 0.6737 | **0.1859** | **0.4312** | **55.437** | **0.9959** | **0.7825** |
| Venus 1418.0432 | 37.6569 | 16.6139 | 0.5307 | 0.8388 | 3.5109 | 1.8737 | 42.6767 | 0.9267 | 0.8234 | 0.4609 | 0.6789 | 51.4945 | **0.9917** | 0.8314 | **0.3257** | **0.5707** | **53.0022** | 0.9906 | **0.901** | |
| Astronaut L. Gordon Cooper | 280.3742 | 16.7444 | 23.6534 | 0.8875 | 0.9914 | 3.5046 | 1.8721 | 42.6844 | 0.9922 | 0.9986 | 0.1089 | 0.3301 | 57.7586 | 0.9992 | 0.9991 | **0.0781** | **0.2794** | **59.205** | **0.9994** | **0.9998** |
| Panoramic view | 543.8697 | 23.321 | 20.7759 | 0.8449 | 0.8448 | 1.9811 | 1.4075 | 45.1617 | 0.9827 | 0.9566 | 0.1719 | 0.4146 | 55.7774 | **0.9971** | 0.564 | **0.1085** | **0.3293** | **57.7778** | 0.9939 | **0.9368** |
| Television (TV) monitor | 103.2832 | 10.1628 | 27.9905 | 0.9299 | 0.9993 | 3.5 | 1.8708 | 42.6901 | 0.9913 | 0.9986 | 0.1057 | 0.3252 | 57.8891 | 0.9993 | **1** | **0.0776** | **0.2785** | **59.2338** | **0.9995** | **1** |
| Wallops Island | 283.8586 | 16.8481 | 23.5998 | 0.9104 | 0.9834 | 3.4603 | 1.8602 | 42.7397 | 0.9859 | 0.9795 | 0.1058 | 0.3253 | 57.8862 | **0.9994** | 0.9984 | **0.077** | **0.2774** | **59.2672** | 0.9993 | **0.9999** |
| Group Photo | 23.5546 | 4.8533 | 34.4101 | 0.9848 | 0.9999 | 3.4784 | 1.8651 | 42.717 | 0.9893 | 0.9966 | 0.0269 | 0.164 | 63.8354 | **0.9998** | 0.9998 | **0.0198** | **0.1408** | **65.1603** | 0.9997 | **1** |
| onion | 1826.7066 | 42.74 | 15.5141 | 0.2221 | 0.9379 | 3.4554 | 1.8589 | 42.7458 | 0.9884 | 0.9996 | N/A | N/A | N/A | N/A | N/A | **0.7466** | **0.8641** | **49.3998** | **0.9944** | **1** |
| pout | 390.0869 | 19.7506 | 22.2192 | 0.6407 | 0.9948 | 3.6359 | 1.9068 | 42.5246 | 0.9881 | 0.9998 | N/A | N/A | N/A | N/A | N/A | **0.2862** | **0.5349** | **53.5648** | **0.9969** | **1** |
| cameraman | 865.088 | 29.4124 | 18.7602 | 0.6505 | 0.9449 | 3.4359 | 1.8536 | 42.7703 | 0.9872 | 0.966 | 0.4228 | 0.6502 | 51.8699 | **0.9967** | 0.9919 | **0.3079** | **0.5549** | **53.2466** | 0.9962 | **0.9994** |
| forest | 585.955 | 24.2065 | 20.4522 | 0.8133 | 0.9434 | 4.2363 | 2.0582 | 41.861 | 0.9919 | 0.9903 | N/A | N/A | N/A | N/A | N/A | **0.1577** | **0.3971** | **56.1529** | **0.9996** | **1** |
| spine | 432.4444 | 20.7953 | 21.7715 | 0.8352 | 0.9192 | 2.499 | 1.5808 | 44.1532 | 0.9767 | 0.9487 | N/A | N/A | N/A | N/A | N/A | **0.1506** | **0.3881** | **56.3529** | **0.9949** | **0.9621** |
| lighthouse | 143.2877 | 11.9703 | 26.5687 | 0.9186 | 0.9962 | 3.4674 | 1.8621 | 42.7308 | 0.9884 | 0.9994 | N/A | N/A | N/A | N/A | N/A | **0.0662** | **0.2573** | **59.9236** | **0.9993** | **1** |
| fabric | 113.3036 | 10.6444 | 27.5884 | 0.9412 | 0.9983 | 3.4939 | 1.8692 | 42.6977 | 0.996 | 0.9986 | N/A | N/A | N/A | N/A | N/A | **0.0661** | **0.2571** | **59.9279** | **0.9998** | **1** |
| flamingos | 45.3042 | 6.7308 | 31.5694 | 0.9788 | 0.9984 | 3.5002 | 1.8709 | 42.6899 | 0.992 | 0.9996 | N/A | N/A | N/A | N/A | N/A | **0.0161** | **0.1268** | **66.0691** | **1** | **1** |

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



**Table 3.** Robustness results of the proposed scheme under different cropping attacks, across a dataset of forty two distinct test host images, with the Correlation Coefficient (CC), NO. of Erroneous Bits (NEB), and Bit Error Ratio (BER) being the evaluation criterion.

| Host Image Grayscale (8 bits/pixel) | Crop All Sides (6%) | | | Crop Top-Left (10%) | | | Crop Top-Right (10%) | | | Crop Bottom-Left (20%) | | | Crop Bottom-Right (20%) | | | Crop Center (35%) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CC | NEB | BER | CC | NEB | BER | CC | NEB | BER | CC | NEB | BER | CC | NEB | BER | CC | NEB | BER |
| House | 0.9632 | 195 | 0.003 | 0.9854 | 75 | 0.0011 | 0.9844 | 86 | 0.0013 | 1 | 0 | 0 | 1 | 0 | 0 | 0.9637 | 148 | 0.0023 |
| Jelly beans | 0.9632 | 195 | 0.003 | 0.9854 | 75 | 0.0011 | 0.9844 | 86 | 0.0013 | 1 | 0 | 0 | 1 | 0 | 0 | 0.9637 | 148 | 0.0023 |
| Fishing Boat | 0.9084 | 535 | 0.0082 | 0.9506 | 263 | 0.004 | 0.9488 | 263 | 0.004 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| Splash | 0.9084 | 535 | 0.0082 | 0.9506 | 263 | 0.004 | 0.9488 | 263 | 0.004 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| Mandrill | 0.9084 | 535 | 0.0082 | 0.9506 | 263 | 0.004 | 0.9488 | 263 | 0.004 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| Peppers | 0.9084 | 535 | 0.0082 | 0.9506 | 263 | 0.004 | 0.9488 | 263 | 0.004 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| Stream & bridge | 0.9084 | 535 | 0.0082 | 0.9506 | 263 | 0.004 | 0.9488 | 263 | 0.004 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| Truck | 0.9084 | 535 | 0.0082 | 0.9506 | 263 | 0.004 | 0.9488 | 263 | 0.004 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| Pentagon | 0.6258 | 2335 | 0.0356 | 0.9169 | 437 | 0.0067 | 0.9206 | 434 | 0.0066 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| Male | 0.6258 | 2335 | 0.0356 | 0.9169 | 437 | 0.0067 | 0.9206 | 434 | 0.0066 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| W9CCDB54 | 0.9084 | 535 | 0.0082 | 0.9506 | 263 | 0.004 | 0.9488 | 263 | 0.004 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| W9CIL54816 | 0.8645 | 809 | 0.0123 | 0.9441 | 313 | 0.0048 | 0.941 | 342 | 0.0052 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| Brain MRI 55 | 0.9632 | 195 | 0.003 | 0.9854 | 75 | 0.0011 | 0.9844 | 86 | 0.0013 | 1 | 0 | 0 | 1 | 0 | 0 | 0.9637 | 148 | 0.0023 |
| Brain MRI 63 | 0.9632 | 195 | 0.003 | 0.9854 | 75 | 0.0011 | 0.9844 | 86 | 0.0013 | 1 | 0 | 0 | 1 | 0 | 0 | 0.9637 | 148 | 0.0023 |
| Brain MRI 70 | 0.9632 | 195 | 0.003 | 0.9854 | 75 | 0.0011 | 0.9844 | 86 | 0.0013 | 1 | 0 | 0 | 1 | 0 | 0 | 0.9637 | 148 | 0.0023 |
| Chest CT 30 | 0.9632 | 195 | 0.003 | 0.9854 | 75 | 0.0011 | 0.9844 | 86 | 0.0013 | 1 | 0 | 0 | 1 | 0 | 0 | 0.9637 | 148 | 0.0023 |
| Chest CT 60 | 0.9632 | 195 | 0.003 | 0.9854 | 75 | 0.0011 | 0.9844 | 86 | 0.0013 | 1 | 0 | 0 | 1 | 0 | 0 | 0.9637 | 148 | 0.0023 |
| Chest CT 90 | 0.9632 | 195 | 0.003 | 0.9854 | 75 | 0.0011 | 0.9844 | 86 | 0.0013 | 1 | 0 | 0 | 1 | 0 | 0 | 0.9637 | 148 | 0.0023 |
| im0001 | 0.8285 | 1044 | 0.0159 | 0.9196 | 440 | 0.0067 | 0.915 | 458 | 0.007 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| im0100 | 0.8285 | 1044 | 0.0159 | 0.9196 | 440 | 0.0067 | 0.915 | 458 | 0.007 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| im0200 | 0.8285 | 1044 | 0.0159 | 0.9196 | 440 | 0.0067 | 0.915 | 458 | 0.007 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| im0300 | 0.8285 | 1044 | 0.0159 | 0.9196 | 440 | 0.0067 | 0.915 | 458 | 0.007 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| im0400 | 0.8285 | 1044 | 0.0159 | 0.9196 | 440 | 0.0067 | 0.915 | 458 | 0.007 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| Galen Clark | 0.8133 | 1125 | 0.0172 | 0.9065 | 491 | 0.0075 | 0.9154 | 456 | 0.007 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| General Douglas | 0.8537 | 876 | 0.0134 | 0.912 | 463 | 0.0071 | 0.9103 | 467 | 0.0071 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| Turbine Power House | 0.8537 | 876 | 0.0134 | 0.912 | 463 | 0.0071 | 0.9103 | 467 | 0.0071 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| Shipyard | 0.6258 | 2335 | 0.0356 | 0.9169 | 437 | 0.0067 | 0.9206 | 434 | 0.0066 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| Full Moon | 0.9632 | 195 | 0.003 | 0.9854 | 75 | 0.0011 | 0.9844 | 86 | 0.0013 | 1 | 0 | 0 | 1 | 0 | 0 | 0.9637 | 148 | 0.0023 |
| Venus | 0.9632 | 195 | 0.003 | 0.9854 | 75 | 0.0011 | 0.9844 | 86 | 0.0013 | 1 | 0 | 0 | 1 | 0 | 0 | 0.9637 | 148 | 0.0023 |

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



**Table 3.** *Cont.*

| Host Image Grayscale (8 bits/pixel) | Crop All Sides (6%) | | | Crop Top-Left (10%) | | | Crop Top-Right (10%) | | | Crop Bottom-Left (20%) | | | Crop Bottom-Right (20%) | | | Crop Center (35%) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CC | NEB | BER | CC | NEB | BER | CC | NEB | BER | CC | NEB | BER | CC | NEB | BER | CC | NEB | BER |
| Astronaut L. Gordon Cooper | 0.9084 | 535 | 0.0082 | 0.9506 | 263 | 0.004 | 0.9488 | 263 | 0.004 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| Panoramic view | 0.9084 | 535 | 0.0082 | 0.9506 | 263 | 0.004 | 0.9488 | 263 | 0.004 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| Television (TV) monitor | 0.9084 | 535 | 0.0082 | 0.9506 | 263 | 0.004 | 0.9488 | 263 | 0.004 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| Wallops Island | 0.9084 | 535 | 0.0082 | 0.9506 | 263 | 0.004 | 0.9488 | 263 | 0.004 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| Group Photo | 0.6258 | 2335 | 0.0356 | 0.9169 | 437 | 0.0067 | 0.9206 | 434 | 0.0066 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| onion | 0.9766 | 114 | 0.0017 | 0.9956 | 24 | 3.66e-4 | 0.9941 | 32 | 4.88e-4 | 1 | 0 | 0 | 1 | 0 | 0 | 0.7833 | 1399 | 0.0213 |
| pout | 0.9631 | 186 | 0.0028 | 0.9863 | 71 | 0.0011 | 0.9868 | 65 | 9.92e-4 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| cameraman | 0.9632 | 195 | 0.003 | 0.9854 | 75 | 0.0011 | 0.9844 | 86 | 0.0013 | 1 | 0 | 0 | 1 | 0 | 0 | 0.9637 | 148 | 0.0023 |
| forest | 0.9037 | 548 | 0.0084 | 0.9717 | 155 | 0.0024 | 0.9695 | 163 | 0.0025 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| spine | 0.9112 | 512 | 0.0078 | 0.9642 | 197 | 0.003 | 0.9563 | 231 | 0.0035 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| lighthouse | 0.8781 | 719 | 0.011 | 0.9425 | 331 | 0.0051 | 0.9422 | 354 | 0.0054 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| fabric | 0.8907 | 649 | 0.0099 | 0.9414 | 328 | 0.005 | 0.9478 | 302 | 0.0046 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| flamingos | 0.6203 | 2390 | 0.0365 | 0.9096 | 498 | 0.0076 | 0.9143 | 480 | 0.0073 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



**Table 4.** Robustness Analysis of the proposed methodology against perturbations induced by noise and sharpening attacks, across a dataset of forty two distinct test host images, with the evaluation criterion being the Correlation Coefficient (CC), NO. of Erroneous Bits (NEB), and Bit Error Ratio (BER).

| Host Image Grayscale (8 bits/pixel) | Salt & Pepper Noise Density = (0.01) | | | Speckle Variance = (0.01) | | | Sharpening (Radius = 0.3 & Amount = 0.5) | | | Sharpening (Radius = 0.4 & Amount = 0.1) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CC | NEB | BER | CC | NEB | BER | CC | NEB | BER | CC | NEB | BER |
| House | 0.9964 | 18 | 2.75e-4 | 0.9826 | 87 | 0.0013 | 1 | 0 | 0 | 1 | 0 | 0 |
| Jelly bean | 0.9981 | 10 | 1.53e-4 | 0.9861 | 73 | 0.0011 | 1 | 0 | 0 | 1 | 0 | 0 |
| Fishing Boat | 0.9977 | 10 | 1.53e-4 | 0.985 | 77 | 0.0012 | 0.9999 | 1 | 1.53e-5 | 0.9884 | 66 | 0.001 |
| Splash | 0.9993 | 7 | 1.07e-4 | 0.9874 | 68 | 0.001 | 1 | 0 | 0 | 0.9995 | 2 | 3.05e-5 |
| Mandrill | 0.9985 | 7 | 1.07e-4 | 0.9858 | 76 | 0.0012 | 0.9939 | 31 | 4.73e-4 | 0.7494 | 1589 | 0.0242 |
| Peppers | 0.9984 | 8 | 1.22e-4 | 0.9853 | 79 | 0.0012 | 1 | 0 | 0 | 0.9989 | 6 | 9.16e-5 |
| Stream & bridge | 0.9984 | 8 | 1.22e-4 | 0.9858 | 75 | 0.0011 | 0.998 | 16 | 2.44e-4 | 0.8942 | 618 | 0.0094 |
| Truck | 0.9986 | 8 | 1.22e-4 | 0.9871 | 70 | 0.0011 | 1 | 0 | 0 | 0.9994 | 3 | 4.58e-5 |
| Pentagon | 0.9982 | 7 | 1.07e-4 | 0.983 | 90 | 0.0014 | 1 | 0 | 0 | 0.9897 | 50 | 7.63e-4 |
| Male | 0.9964 | 17 | 2.59e-4 | 0.985 | 76 | 0.0012 | 1 | 0 | 0 | 0.9995 | 2 | 3.05e-5 |
| W9CCDB54 | 0.9958 | 20 | 3.05e-4 | 0.9881 | 68 | 0.001 | 1 | 0 | 0 | 1 | 0 | 0 |
| W9CIL54816 | 0.9991 | 6 | 9.16e-5 | 0.9855 | 73 | 0.0011 | 1 | 0 | 0 | 1 | 0 | 0 |
| Brain MRI 55 | 0.9982 | 7 | 1.07e-4 | 0.9849 | 80 | 0.0012 | 0.9995 | 2 | 3.05e-5 | 0.9688 | 171 | 0.0026 |
| Brain MRI 63 | 0.9989 | 7 | 1.07e-4 | 0.9857 | 70 | 0.0011 | 0.9974 | 11 | 1.68e-4 | 0.9353 | 349 | 0.0053 |
| Brain MRI 70 | 0.9989 | 6 | 9.16e-5 | 0.9848 | 89 | 0.0014 | 0.9998 | 4 | 6.10e-5 | 0.9619 | 215 | 0.0033 |
| Chest CT 30 | 0.9978 | 12 | 1.83e-4 | 0.9846 | 85 | 0.0013 | 1 | 0 | 0 | 0.9849 | 65 | 9.92e-4 |
| Chest CT 60 | 0.9986 | 8 | 1.22e-4 | 0.9854 | 83 | 0.0013 | 1 | 0 | 0 | 0.992 | 44 | 6.71e-4 |
| Chest CT 90 | 0.9987 | 7 | 1.07e-4 | 0.9857 | 72 | 0.0011 | 1 | 0 | 0 | 0.9918 | 47 | 7.17e-4 |
| im0001 | 0.9978 | 9 | 1.37e-4 | 0.9856 | 78 | 0.0012 | 1 | 0 | 0 | 1 | 0 | 0 |
| im0100 | 0.9993 | 3 | 4.58e-5 | 0.9854 | 78 | 0.0012 | 1 | 0 | 0 | 1 | 0 | 0 |
| im0200 | 0.9978 | 11 | 1.68e-4 | 0.982 | 85 | 0.0013 | 1 | 0 | 0 | 1 | 0 | 0 |
| im0300 | 0.999 | 7 | 1.07e-4 | 0.9871 | 69 | 0.0011 | 1 | 0 | 0 | 1 | 0 | 0 |
| im0400 | 0.9976 | 10 | 1.53e-4 | 0.9874 | 70 | 0.0011 | 1 | 0 | 0 | 1 | 0 | 0 |
| Galen Clark | 0.9989 | 6 | 9.16e-5 | 0.9836 | 82 | 0.0013 | 1 | 0 | 0 | 1 | 0 | 0 |
| General Douglas | 0.9984 | 8 | 1.22e-4 | 0.9829 | 78 | 0.0012 | 1 | 0 | 0 | 0.9988 | 7 | 1.07e-4 |
| Turbine Power House | 0.9985 | 10 | 1.53e-4 | 0.9858 | 76 | 0.0012 | 1 | 0 | 0 | 0.9995 | 2 | 3.05e-5 |
| Shipyard | 0.9993 | 5 | 7.63e-5 | 0.9879 | 69 | 0.0011 | 1 | 0 | 0 | 0.9977 | 14 | 2.14e-4 |
| Full Moon | 0.9989 | 6 | 9.16e-5 | 0.9825 | 86 | 0.0013 | 0.9973 | 13 | 1.98e-4 | 0.9875 | 68 | 0.001 |
| Venus | 0.999 | 5 | 7.63e-5 | 0.9865 | 74 | 0.0011 | 0.9923 | 47 | 7.17e-4 | 0.983 | 94 | 0.0014 |
| Astronaut L. Gordon Cooper | 0.9986 | 9 | 1.37e-4 | 0.9813 | 95 | 0.0014 | 0.9671 | 164 | 0.0025 | 0.8148 | 1130 | 0.0172 |
| Panoramic view | 0.9986 | 7 | 1.07e-4 | 0.9825 | 88 | 0.0013 | 1 | 0 | 0 | 1 | 0 | 0 |
| Television (TV) monitor | 0.9985 | 7 | 1.07e-4 | 0.9839 | 82 | 0.0013 | 1 | 0 | 0 | 0.956 | 239 | 0.0036 |
| Wallops Island | 0.9974 | 13 | 1.98e-4 | 0.9844 | 83 | 0.0013 | 0.9899 | 56 | 8.54e-4 | 0.9294 | 388 | 0.0059 |
| Group Photo | 0.9991 | 4 | 6.10e-5 | 0.9847 | 81 | 0.0012 | 1 | 0 | 0 | 1 | 0 | 0 |
| onion | 0.9972 | 15 | 2.29e-4 | 0.9871 | 72 | 0.0011 | 1 | 0 | 0 | 0.9951 | 23 | 3.51e-4 |
| pout | 0.9973 | 14 | 2.14e-4 | 0.9855 | 75 | 0.0011 | 1 | 0 | 0 | 1 | 0 | 0 |
| cameraman | 0.9991 | 5 | 7.63e-5 | 0.9857 | 72 | 0.0011 | 0.9908 | 44 | 6.71e-4 | 0.9308 | 406 | 0.0062 |
| forest | 0.9997 | 2 | 3.05e-5 | 0.9853 | 81 | 0.0012 | 0.9969 | 13 | 1.98e-4 | 0.9045 | 548 | 0.0084 |
| spine | 0.998 | 8 | 1.22e-4 | 0.9859 | 73 | 0.0011 | 1 | 0 | 0 | 1 | 0 | 0 |
| lighthouse | 0.9981 | 10 | 1.53e-4 | 0.9855 | 73 | 0.0011 | 1 | 0 | 0 | 1 | 0 | 0 |
| fabric | 0.9998 | 4 | 6.10e-5 | 0.9835 | 85 | 0.0013 | 1 | 0 | 0 | 0.9955 | 23 | 3.51e-4 |
| flamingos | 0.9991 | 6 | 9.16e-5 | 0.9819 | 92 | 0.0014 | 0.9995 | 2 | 3.05e-5 | 0.9959 | 23 | 3.51e-4 |

### *6.3. Computational Time Analysis.*

Table 5 and Table 6 present a comparative analysis of embedding and extraction times (in seconds) respectively for watermarking methods across different host images and watermark sizes. Four methods are evaluated with embedding and extraction times provided for two watermark sizes: 64x128 and 128x164 pixels. The analysis reveals significant variation in embedding and extraction times based on the method and image characteristics. Notably, the proposed scheme consistently demonstrates lower outcomes for both embedding as well as extraction times than other methods across almost all scenarios listed in the tables. This observation underscores its remarkable prowess and effectiveness in the processes of embedding and extracting watermarks from images, showcasing unmatched performance and efficiency. Factors influencing embedding and extraction times include algorithm complexity, computational efficiency, and host image and watermark characteristics. Ultimately, the

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



choice of watermarking method should consider a balance between embedding and extraction times, robustness, and other relevant factors to meet the requirements of the intended application.

Table 7 presents detailed data on key generation times (in seconds) for various watermarking methods across different host images and watermark sizes. The proposed scheme consistently exhibits notably longer key generation times compared to other methods, particularly for larger host image sizes and watermark dimensions. This extended duration results from a careful and thorough approach taken during the key generation process aimed at enhancing the security of the watermarking system. The scheme invests more time and effort into generating keys with a high level of complexity and randomness, making them more resistant to various attacks and unauthorized access. Despite the longer key generation times, the proposed scheme offers heightened security benefits due to the thoroughness of its CA key generation process. However, the trade-off between longer key generation times and enhanced security should be carefully weighed against the requirements of specific applications.

This prolonged duration is also attributable to our system being implemented on a sequential architecture which means that the parallelism property of CA is not being taken advantage of at all. Incorporating CAD systems for concurrent implementation [49] to leverage the parallelism property of cellular automata presents a promising solution to mitigate the longer time required for key generation in our system. By harnessing the computational power of parallel processing, we can distribute the workload across multiple computing units, thereby accelerating the key generation process. This optimization not only addresses the current bottleneck but also enhances the efficiency and scalability of our system. Leveraging CAD systems for key generation underscores our commitment to optimizing performance while ensuring robust security measures.

**Table 5.** Embedding Time in Seconds of Watermarking Methods for Different Host Images and Watermarks

| Host Image Grayscale (8 bits/pixel) | Host Image Size (pixels) | Watermark Size (64x128) | | | | Watermark Size (128 164) | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Ye and Li [24] | Adwan et al. [9] | Moniruzzaman et al. [10] | Proposed Scheme. | Ye and Li [24] | Adwan et al. [9] | Moniruzzaman et al.[10] | Proposed Scheme. |
| onion | 198×135 | 0.159809 | 0.036489 | N/A | **0.025728** | 0.128354 | 0.107455 | N/A | **0.030702** |
| House | 256×256 | 0.124157 | 0.072592 | 0.899457 | **0.020502** | 0.123899 | 0.088318 | 0.730469 | **0.036032** |
| Splash | 512×512 | 0.632889 | 0.038331 | 2.915754 | **0.033725** | 0.559585 | 0.052262 | 2.582008 | **0.039807** |
| Pentagon | 1024×1024 | 1.300323 | 0.087177 | 10.707748 | **0.021180** | 1.323873 | 0.099821 | 10.901481 | **0.037367** |

**Table 6.** Extraction Time in Seconds of Watermarking Methods for Different Host Images and Watermarks

| Host Image Grayscale (8 bits/pixel) | Host Image Size (pixels) | Watermark Size (64x128) | | | | Watermark Size (128 164) | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Ye and Li [24] | Adwan et al. [9] | Moniruzzaman et al. [10] | Proposed Scheme. | Ye and Li [24] | Adwan et al.[9] | Moniruzzaman et al. [10] | Proposed Scheme. |
| onion | 198×135 | 0.018147 | 0.041306 | N/A | 0.031554 | 0.060712 | 0.078757 | N/A | **0.02669** |
| House | 256×256 | 0.032714 | 0.055386 | 0.657102 | **0.032163** | 0.052649 | 0.091742 | 0.653864 | **0.041124** |
| Splash | 512×512 | 0.124647 | 0.069233 | 2.505042 | **0.035088** | 0.108714 | 0.118865 | 2.357764 | **0.028482** |
| Pentagon | 1024×1024 | 0.28955 | 0.036978 | 11.034892 | **0.026006** | 0.318926 | 0.070721 | 11.237124 | **0.056624** |

**Table 7.** Key Generation Time in Seconds of Watermarking Methods for Different Host Images and Watermarks

| Host Image Grayscale (8 bits/pixel) | Host Image Size (pixels) | Watermark Size (64x128) | | | | Watermark Size (128 164) | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | Ye and Li [24] | Adwan et al. [9] | Moniruzzaman et al. [10] | Proposed Scheme. | Ye and Li [24] | Adwan et al. [9] | Moniruzzaman et al. [10] | Proposed Scheme. |
| onion | 198×135 | 0.052517 | 0.140891 | N/A | 109.8306 | 0.125685 | 0.376386 | N/A | 244.3651 |
| House | 256×256 | 0.074302 | 0.105079 | 1.031523 | 121.8449 | 0.084188 | 0.220986 | 0.704578 | 257.2078 |
| Splash | 512×512 | 0.380358 | 0.200555 | 3.06983 | 126.5283 | 0.231141 | 0.340498 | 2.750723 | 225.1555 |
| Pentagon | 1024×1024 | 2.42462 | 0.241485 | 11.469756 | 159.2139 | 3.140535 | 0.288682 | 11.853642 | 207.0514 |

## 7. Conclusion and Future Work

Cellular automata (CA) are a notable and sophisticated model that has shown considerable potential in improving the security and reliability of digital assets when it comes to digital image watermarking. CA-based watermarking techniques have demonstrated several impressive qualities, such as robustness to frequent attacks, the embedded watermarks' imperceptibility, and effective data embedding in images. Through this research, we have explored a CA-based watermarking strategy governed by rule-30, which demonstrated its efficacy in protecting digital images against unauthorized use, unauthorized modification, and infringement on intellectual property.

Even if there have been a lot of noteworthy advances in the field of CA-based digital image watermarking, there are still a lot of uncharted territories that want more research and development.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



Fortifying CA-based watermarking approaches against a larger range of threats should be the primary focus of future research endeavors. To make these techniques more resistant to a wider range of attacks, such as geometric transformations, compression-induced distortions, and complex signal processing techniques, this may involve discovering more complex rule sets and carefully integrating machine learning paradigms. The security component must be given prominent attention. The vulnerability of CA-based watermarking techniques to complex and advanced attacks should be thoroughly examined in future investigations. One of the key goals is to maximize the watermarking capacity while preserving the image's visual quality. Subsequent research endeavors may explore and develop algorithms that augment data-hiding efficacy, guaranteeing that watermarks remain undetectable while concurrently optimizing the amount of information that may be incorporated. A promising direction for future research is to modify CA-based watermarking techniques for real-time use. This could apply to the watermarking of videos or the protection of streaming media, where quick data processing is critical.

To sum up, the integration of cellular automata with digital image watermarking has shown considerable potential and cracked the gate to further study and development. Dedicated efforts in this area have the potential to offer safe and effective ways to protect digital assets in an increasingly digital environment as the digital world grows and develops over time.

**Author Contributions:** "Conceptualization, I.K.B. and F.Q., A.H.G; methodology, I.K.B.; software, I.K.B.; validation, I.K.B. and F.Q.; formal analysis, I.K.B.; investigation, I.K.B. and F.Q.; resources, I.K.B., M.N., A.H.G; data curation, I.K.B.; writing—original draft preparation, I.K.B.; writing—review and editing, F.Q., M.N., and A.H.G; visualization, I.K.B.; supervision, F.Q.; project administration, I.K.B. and F.Q.; funding acquisition, A.H.G All authors have read and agreed to the published version of the manuscript."

**Acknowledgments:** The authors acknowledge the support of Dr. Akib Khanday for his continuous suggestions.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Zhong, X.; Das, A.; Alrasheedi, F.; Tanvir, A. A Brief, In-Depth Survey of Deep Learning-Based Image Watermarking. *Applied Sciences* **2023**, *13*, 11852.
2. Begum, M.; Uddin, M.S. Digital image watermarking techniques: a review. *Information* **2020**, *11*, 110.
3. Mousavi, S.M.; Naghsh, A.; Abu-Bakar, S. Watermarking techniques used in medical images: a survey. *Journal of digital imaging* **2014**, *27*, 714–729.
4. Gomez-Coronel, S.L.; Moya-Albor, E.; Brieva, J.; Romero-Arellano, A. A Robust and Secure Watermarking Approach Based on Hermite Transform and SVD-DCT. *Applied Sciences* **2023**, *13*, 8430.
5. Qasim, A.F.; Meziane, F.; Aspin, R. Digital watermarking: Applicability for developing trust in medical imaging workflows state of the art review. *Computer Science Review* **2018**, *27*, 45–60.
6. Al-Wesabi, F.N.; Alrowais, F.; Mohamed, H.G.; Al Duhayyim, M.; Hilal, A.M.; Motwakel, A. Heuristic Optimization Algorithm Based Watermarking on Content Authentication and Tampering Detection for English Text. *IEEE Access* **2023**.
7. Menendez-Ortiz, A.; Feregrino-Uribe, C.; Hasimoto-Beltran, R.; Garcia-Hernandez, J.J. A survey on reversible watermarking for multimedia content: A robustness overview. *IEEE Access* **2019**, *7*, 132662–132681.
8. Hasan, B.M.S.; Ameen, S.Y.; Hasan, O.M.S. Image authentication based on watermarking approach. *Asian Journal of Research in Computer Science* **2021**, *9*, 34–51.
9. Adwan, O.; Awwad, A.A.; Sleit, A.; Alhoum, A.L.A. A novel watermarking scheme based on two dimensional cellular automata. Proceedings of the International Conference on Computers and Computing, World Scientific and Engineering Academy and Society (WSEAS). Canary Islands, Spain, 2011, pp. 88–94.
10. Moniruzzaman, M.; Hawlader, M.A.K.; Hossain, M.F. Watermarking scheme based on game of life cellular automaton. 2014 International Conference on Informatics, Electronics & Vision (ICIEV). IEEE, 2014, pp. 1–6.
11. BW, T.A.; Permana, F.P.; others. Medical image watermarking with tamper detection and recovery using reversible watermarking with LSB modification and run length encoding (RLE) compression. 2012

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2



IEEE International Conference on Communication, Networks and Satellite (ComNetSat). IEEE, 2012, pp. 167–171.

12. Manjula, G.; Danti, A. A novel hash based least significant bit (2-3-3) image steganography in spatial domain. *arXiv preprint arXiv:1503.03674* **2015**.
13. Abraham, J.; Paul, V. An imperceptible spatial domain color image watermarking scheme. *Journal of King Saud University-Computer and Information Sciences* **2019**, *31*, 125–133.
14. Zeki, A.; Abubakar, A.; Chiroma, H. An intermediate significant bit (ISB) watermarking technique using neural networks. *SpringerPlus* **2016**, *5*, 1–25.
15. Zhang, H.; Wang, C.; Zhou, X. Fragile watermarking based on LBP for blind tamper detection in images. *Journal of Information Processing Systems* **2017**, *13*, 385–399.
16. Kelkar, V.; Tuckley, K.; Nemade, H.; others. Novel variants of a histogram shift-based reversible watermarking technique for medical images to improve hiding capacity. *Journal of healthcare engineering* **2017**, *2017*.
17. Nasir, M.; Jadoon, W.; Khan, I.A.; Gul, N.; Shah, S.; ELAffendi, M.; Muthanna, A. Secure Reversible Data Hiding in Images Based on Linear Prediction and Bit-Plane Slicing. *Mathematics* **2022**, *10*, 3311.
18. Zhang, F.; Luo, T.; Jiang, G.; Yu, M.; Xu, H.; Zhou, W. A novel robust color image watermarking method using RGB correlations. *Multimedia Tools and Applications* **2019**, *78*, 20133–20155.
19. Ko, H.J.; Huang, C.T.; Horng, G.; Shiuh-Jeng, W. Robust and blind image watermarking in DCT domain using inter-block coefficient correlation. *Information Sciences* **2020**, *517*, 128–147.
20. Zhang, Y.; Wang, Z.; Zhan, Y.; Meng, L.; Sun, J.; Wan, W. JND-aware robust image watermarking with tri-directional inter-block correlation. *International Journal of Intelligent Systems* **2021**, *36*, 7053–7079.
21. Chauhan, D.S.; Singh, A.K.; Adarsh, A.; Kumar, B.; Saini, J.P. Combining Mexican hat wavelet and spread spectrum for adaptive watermarking and its statistical detection using medical images. *Multimedia Tools and Applications* **2019**, *78*, 12647–12661.
22. Nguyen-Thanh, T.; Le-Tien, T. Study on Improved Cooperative Spread Spectrum Based Robust Blind Image Watermarking. *Journal of Advances in Information Technology Vol* **2020**, *11*.
23. Novamizanti, L.; Suksmono, A.B.; Danudirdjo, D.; Budiman, G. Robust Reversible Watermarking using Stationary Wavelet Transform and Multibit Spread Spectrum in Medical Images. *International Journal of Intelligent Engineering & Systems* **2022**, *15*.
24. Ye, R.; Li, H. A novel image scrambling and watermarking scheme based on cellular automata. 2008 International Symposium on Electronic Commerce and Security. IEEE, 2008, pp. 938–941.
25. Ramos, A.M.; Artiles, J.A.; Chaves, D.P.; Pimentel, C. A Fragile Image Watermarking Scheme in DWT Domain Using Chaotic Sequences and Error-Correcting Codes. *Entropy* **2023**, *25*, 508.
26. Zhu, B.; Fan, X.; Zhang, T.; Zhou, X. Robust Blind Image Watermarking Using Coefficient Differences of Medium Frequency between Inter-Blocks. *Electronics* **2023**, *12*, 4117.
27. Laouamer, L.; Tayan, O. A semi-blind robust DCT watermarking approach for sensitive text images. *Arabian Journal for Science and Engineering* **2015**, *40*, 1097–1109.
28. Roy, S.; Pal, A.K. A blind DCT based color watermarking algorithm for embedding multiple watermarks. *AEU-International Journal of Electronics and Communications* **2017**, *72*, 149–161.
29. Liu, S.; Pan, Z.; Song, H. Digital image watermarking method based on DCT and fractal encoding. *IET image processing* **2017**, *11*, 815–821.
30. Singh, S.P.; Bhatnagar, G. A new robust watermarking system in integer DCT domain. *Journal of Visual Communication and Image Representation* **2018**, *53*, 86–101.
31. Ernawan, F.; Kabir, M.N. A robust image watermarking technique with an optimal DCT-psychovisual threshold. *IEEE Access* **2018**, *6*, 20464–20480.
32. Jana, M.; Jana, B. A new DCT based robust image watermarking scheme using cellular automata. *Information Security Journal: A Global Perspective* **2022**, *31*, 527–543.
33. Pitsianis, N.; Tsalides, P.; Bleris, G.; Thanailakis, A.; Card, H. Deterministic one-dimensional cellular automata. *Journal of statistical physics* **1989**, *56*, 99–112.
34. Wolfram, S. Universality and complexity in cellular automata. *Physica D: Nonlinear Phenomena* **1984**, *10*, 1–35.
35. Random Number Generation—Wolfram Language Documentation. https://reference.wolfram.com/language/tutorial/RandomNumberGeneration.html. (Accessed on 10/19/2023).

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 12 March 2024 doi:10.20944/preprints202311.1615.v2




36. Cattaneo, G.; Finelli, M.; Margara, L. Investigating topological chaos by elementary cellular automata dynamics. *Theoretical computer science* **2000**, *244*, 219–241.
37. Rukhin, A.; Soto, J.; Nechvatal, J.; Smid, M.; Barker, E.; Leigh, S.; Levenson, M.; Vangel, M.; Banks, D.; Heckert, A.; others. *A statistical test suite for random and pseudorandom number generators for cryptographic applications*; Vol. 22, US Department of Commerce, Technology Administration, National Institute of . . . , 2001.
38. Gage, D.; Laub, E.; McGarry, B. Cellular automata: is rule 30 random. Proceedings of the Midwest NKS Conference, Indiana University, 2005.
39. S.Wolfram-Writings. r30img2.png (1240×642). https://content.wolfram.com/sites/43/2020/07/r30img2.png. (Accessed on 11/10/2023).
40. S.Wolfram-Writings. r30img7.png (702×366). https://content.wolfram.com/sites/43/2020/07/r30img7.png. (Accessed on 11/10/2023).
41. MathWorks - Makers of MATLAB and Simulink - MATLAB & Simulink, Version 9.8.0.1323502 Release 2020a; The Math Works, Inc., Feb 25,2020. Computer Software. https://in.mathworks.com/. (Accessed on 11/09/2023).
42. The University of Southern California. "SIPI Image Database". https://sipi.usc.edu/database/database.php. (Accessed on 10/20/2023).
43. Tseng, C.H.L.E.K.J. CIL:54816, Homo sapiens Linnaeus, 1758, epithelial cell. CIL. Dataset. CIL. Dataset. http://www.cellimagelibrary.org/images/54816#cite, 2022. (Accessed on 10/20/2023).
44. Don Fox, University of Houston, G.P. The Cell Image Library. http://www.cellimagelibrary.org/images/CCDB_54#cite, 2001. (Accessed on 10/20/2023).
45. Clark, K.; Vendt, B.; Smith, K.; Freymann, J.; Kirby, J.; Koppel, P.; Moore, S.; Phillips, S.; Maffitt, D.; Pringle, M.; others. The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository. *Journal of digital imaging* **2013**, *26*, 1045–1057.
46. University of California, San Diego. The STARE Project. https://cecas.clemson.edu/~ahoover/stare/. (Accessed on 10/20/2023).
47. U.S. National Archives NextGen Catalog. https://catalog.archives.gov/. (Accessed on 11/09/2023).
48. National Aeronautics and Space Administration (NASA), "NASA Image and Video Library", images.nasa.gov. https://images.nasa.gov/. (Accessed on 11/09/2023).
49. Jódar, N.F.; Boluda, J.C.; Gironés, R.G.; Bosch, V.H. CSDL & GLIDER: CAD Tools for Hardware Implementation of Cellular Automata. 2008 International Conference on Advances in Electronics and Micro-electronics. IEEE, 2008, pp. 20–25.


**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.