IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

*Digital Object Identifier 10.1109/ACCESS.2021.Doi Number*

# Robust and Secure Digital Image Watermarking Technique Using Arnold Transform and Memristive Chaotic Oscillators

**Khushwant Sehra[1], *Student Member, IEEE*, Samriddhi Raut[2], *Student Member, IEEE*, Ashutosh Mishra[3], *Student Member, IEEE,* Poonam Kasturi[3], Shweta Wadhera[4], Geetika Jain Saxena[5], and Manoj Saxena[3], *Senior Member, IEEE***

[1]Department of Electronic Science, University of Delhi South Campus, New Delhi, India
[2]Department of Electronics and Communication, Maharaja Surajmal Institute of Technology, New Delhi, India
[3]Department of Electronics, Deen Dayal Upadhyaya College, University of Delhi, New Delhi
[4]Department of Computer Science, Deen Dayal Upadhyaya College, University of Delhi, New Delhi
[3]Department of Electronics, Maharaja Agrasen College, University of Delhi, New Delhi

Corresponding author: Manoj Saxena (e-mail: msaxena@ddu.du.ac.in).

This work was supported in part by DBT Star College Programme, Deen Dayal Upadhyaya College, University of Delhi. The authors also wish to acknowledge University of Delhi for providing the necessary tools and financial assistance for the completion of this work.

## ABSTRACT

With the advent of technology and multimedia production, the world has witnessed a tremendous increase in digital media attacks, which duplicates, forges and tamper the data leading to the violation of copyright laws. In this paper, a robust and secure digital image watermarking is proposed, which exploits the chaotic behaviour of the non – linear oscillators realized through Memristive diodes. The proposed scheme relies on a Human Visual System (HVS) model in order to mimic the real-life scenario. To improve the robustness of the proposed approach and to further increase the security of the digital watermarked media whilst still retaining compatibility with the real-time events, Histogram of Oriented Gradients (HOG) and extreme learning machine (ELM) is implemented. Secure key generation by means of scrambling through Arnold Transform and the coefficients of Memristive Chaotic Oscillator ensures extreme security. The watermark embedding followed the pixel transformation based on discrete cosine coefficient modification, and a semi-blind watermarking extraction procedure was carried out through trained ELM models. A detailed analysis has been presented to evaluate the tradeoff between imperceptibility, security and robustness using performance metrics like PSNR, NC, SSIM, and BER. To establish a real-time implementation of the proposed architecture, the simulated results were verified using real-time chaotic signals generated from the chaotic oscillator, which dictates excellent performance against watermarking attacks and image processing tasks.

**INDEX TERMS** Chaotic Encryption, Memristor, Arnold Transform, Histogram of Oriented Gradients (HOG), Human Visual Systems (HVS), Extreme Machine Learning (ELM), Discrete Cosine Transform (DCT)

## I. INTRODUCTION

The rapid developments in the computer era have led to an exponential increase in digital media production and usage. Consequently, the cases of data duplication forging and tampering have significantly raised a concern towards data encryption and security and ultimate copyright protection of the digital media. Out of the several possible solutions, copy detection, steganography, and digital watermarking techniques are one such method that targets this problem and aims at embedding the information into more protected information in a characteristic manner [1]. The embedded data replicates the host image visually but makes it more secure by encrypting the information prone to malicious image processing attacks [2]. Whenever a copyright issue is encountered, the media is extracted using a watermarking technique.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079319, IEEE Access



These techniques can be broadly classified into two types: (1) Spatial based, which operates on the pixel location by embedding the watermark into the least significant bit (LSB) of the original image, and (2) Frequency − based, which first converts the signal into the frequency domain and embeds and modifies the frequency coefficients after the transforms. However, spatial transforms are less robust but are computationally less complex and have a better payload capacity than the frequency domain, which is found to be more robust, more secure, and offers better invisibility [3]. Literature has demonstrated several techniques like which involves frequency and spatial domain analysis. Frequency domain based transforms like discrete Fourier Transforms (DFT), discrete wavelet transforms (DWT), discrete cosine transforms (DCT) etc., have been studied rigorously. However, it has been well established that using two or more transforms (hybrid) can yield better results. Fazil *et. at.* [4] and Singh *et. al.* [5] has proposed a robust technique based on DWT, DCT with Singular Value Decomposition (SVD). Hwai and his group [6] have proposed that using sign correction, level shifting, mixed modulation, and orthogonal restoration, the process can be made more effective by improving invisibility and robustness. Recently, Liu and his group [7] have demonstrated a hybrid integer wavelet transform (IWT) and DCT to show the double encryption technique. Najafi *et. al.* [8] proposed a watermarking technique in which the group proposed a watermarking algorithm based on sharp frequency localized contourlet transform (SELCT) with SVD, which proved to solve the false positive problem and is resistant to ambiguity attacks. Several people have analyzed different matrix decomposition methods like Schur decomposition, SVD and LU decomposition. Makbol *et. al.* [9] demonstrated a block-based DWT and SVD image watermarking scheme in which they consider entropy as the HVS feature. Over the past few years, people have used fractal dimensions. Mishra *et. al.* [10] have proposed a robust and secure watermarking architecture based on fractal dimensions using the human visual system (HVS) model and Mamdani based Fuzzy Interface System (FIS). Over the past few years, several other techniques like using semi − blind Human Visual System [11], Dual Tree Complex Wavelet Transform − Discrete Cosine Transform (DTCWT − DCT) [12] and 2 Dimensional − Discrete Cosine Transform (2D − DCT) [13] have been introduced and studied. Hosny *et. al.* [14] have demonstrated a fractional order exponent moment watermarking technique. Quaternion based techniques have also emerged as a promising method in image watermarking [15][16]. To make the watermarking more effective and faster, researchers have now moved to various machine learning and deep learning techniques. Extreme Learning Machine (ELM) which can be considered as a special case of neural network with single layer feed – forward, has gained enormous interest due to its better generalization capability and good payload capability to handle large data and is widely used nowadays [17-19]. Ding *et. al.* [20] have recently proposed a generalized deep neural network approach used for watermarking. One major limitation of the neural network approach is the high computational time in case of heavy net with large number of hidden neurons and more vulnerability to statistical attacks like JPEG compressions [20][21]. Recently a parallel multi-core CPU and GPU has been proposed targeting medical images which shows a promising research trend setup in the domain [22]. Several works have also demonstrated extraction of watermark coefficient directly from the host image itself using different processing tasks [10][23].

From the security perspective, chaotic systems are widely used for information encryption as chaotic cryptography over the conventional encryption algorithms like data encryption standard (DES), advanced encryption standard (AES) etc. which are known to have redundancy and correlation problem [23-29]. Chaotic signals are the non − linear signals which are highly sensitive to the system parameters and initial conditions. The random and unpredictable nature of such signals meets the requirements such as diffusion and mixing hence facilitates them to act as encryption keys. Hu *et. al.* [24] have proposed a blind watermarking algorithm where the robustness of the proposed algorithm is tested in the chaotic sequence generated by the logistic system. Bhatti *et. al.* [25] had discussed a hybrid watermarking algorithm using Clifford algebra and Arnold transform. Due to pseudorandom and ergodic properties, the chaotic system is gaining great interest among researchers to study data encryption for better security against geometric attacks [26-29].

Over the past few years, extensive research has been done on utilizing mathematical tools and formulating theoretical approaches for building an effective digital watermarking model. However, there is still a gap when it comes to practical implementations which can be summarized as follows:

- Usage of complex image transformation methods or complex encryption algorithms for enhancing the security unnecessarily leads to extremely high computational complexities and operation cost.
- During data embedding, it is important to capture the features of image in efficient manner while taking care of the security. However, previous reported works either use a large block sized image or considers a very small portion of image which fails to capture the data appropriately; or leading to severe data protection and security concerns.
- While dealing with the DC coefficients, it is necessary to handle the high frequency components in order to maintain the robustness and imperceptibility.
- Further, enhancing and optimizing the pre − processing steps and improving the masking ability while maintaining the payload, computational time and operating cost is one of the major focus of the current research trend.

With this regard, this work proposes a watermarking scheme based on Arnold transform, HOG features, HVS and ELM. With the low computational complexities of the used

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079319, IEEE Access



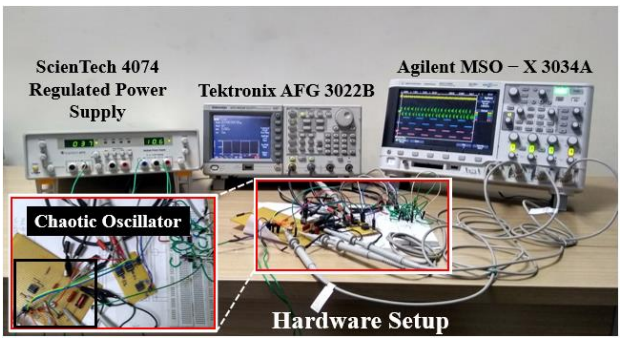


**FIGURE 1. Experimental Setup of the chaotic system consisting of oscillator, and OPAMP realization of Chua's Diode for chaotic signal generation.**

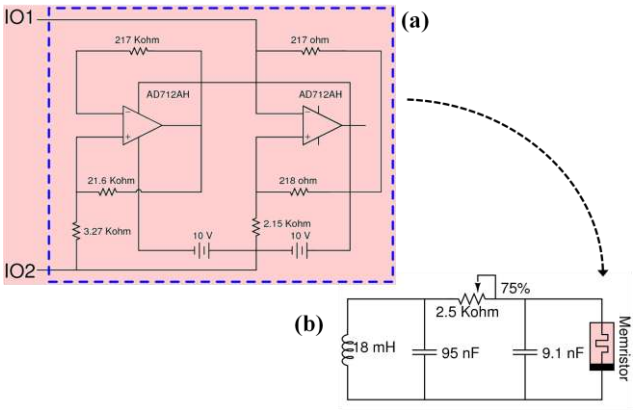


**FIGURE 2. (a) Equivalent Circuit for Memristor realized using op-amp (b) Chaotic oscillator realized using memristor.**

algorithms and extreme secure key generations, high processing speed, robustness, and security can be guaranteed.

The contributions of the presented work can be summarized as follows:

- The proposed architecture exploits block coding and computationally efficient algorithms like Arnold transform for scrambling which ensures a good robustness.
- The architecture involves human visual system (HVS), Histogram of Oriented Gradients and Extreme Learning Machine (ELM) which not only supports the real – life implementations but also improves the robustness and security.
- Secure key generation is implemented at various stages: (a) Arnold transform; (b) Initial conditions and chaotic oscillator coefficients, and (c) Final watermarking generation using trained ELM. Incorporating these features and as dictated from the metrics has remarkably improved the security.
- A detailed insights on robustness, security, imperceptibility, and computational complexities is developed using metrics like peak signal to noise ratios, structural similarity index, bit error rate, normalized cross – correlation for signed images and extracted watermark.
- Experimental validations for chaotic encryption using memristor based chaotic oscillator is presented so as to access the performance of the proposed technique on real – time chaotic signals.

In this work, a novel digital watermarking architecture is presented. The images are first block coded in order to reduce the processing load. The blocks are first analyzed by calculating the fractal dimensions, which are then scrambled using Arnold Transform for encryption. Consequently, the HOG features are extracted. The transformed signals were passed to the Mamdani FIS system to extract the key indexes using which one of the ELM model is trained. The chaotic signals generated using Memristors are utilized for training the second ELM model. The watermark embedding and extraction have been carried out using the weighted mean of both the data obtained from models. Finally, the semi – blind watermark extraction procedures and performance analysis were carried out to comment on the robustness and security of the proposed algorithm.

The paper is organized as follows: Section II gives preliminary information about the algorithms used in the architecture. The complete methodology followed in the work is summarized in Section III, and Section IV compiles all the results and discusses the various features and performance of the watermark technique. Finally, the work is concluded in Section V.

## II. PRELIMINARIES

This section provides an essential mathematical background of the algorithms used in the proposed architecture and details about the experimental setup.

### A. MEMRISTOR & HARDWARE SETUP OF THE CHAOTIC SYSTEM

Memristor, which is popularly known as Chua's diode, is considered to be the missing fourth circuit element. It is a passive circuit element that behaves like a resistive memory. Memristor is a non-linear memory element that found compatibility in many applications like oscillators, information encryption, memory etc. In this work, a Memristor based chaotic oscillator is constructed based on the basic piecewise linear (PWL) $(\varphi - q)$ characteristics using operational amplifier and off the shelf elements. Figure 1 shows the experimental setup of the chaotic system, and Figure 2 shows the equivalent circuit of Memristor and oscillator realized both in hardware setup and simulation setup in NI MultiSim [30]. The hardware setup consists of a Chua's Diode realized through general-purpose OPAMPs [31]. The diode was realized on a custom perf board for simplicity, and its response was validated on Agilent MSO – X 3034A through the signals generated by Tektronix AFG 3022B.

The chaotic signals generations in Chua's circuit are governed by the basic set of three nonlinear ordinary differential equations (ODE) of state variables and 3 –

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079319, IEEE Access



segment piecewise – linear equations as summarized in Equation 1.

$$\frac{dx}{dy} = \alpha[y - x - f(x)]$$

$$RC_2\frac{dy}{dx} = x - y + Rz$$

$$\frac{dz}{dt} = -\beta y \quad (1)$$

The characteristics of the signals can be controlled using the coefficients of ODEs. The Chaotic signals, due to their properties of pseudo-randomness and dynamicity, are widely used in digital media encryption. The chaotic system's output is similar to white noise with correlation and complexity as defined by Equation 2 [32]:

$$C_{n+1} = \mu \times C_n \times (1 - C_n) \quad (2)$$

Here, $0 < \mu < 4$ and $C_n$ is the $n^{th}$ value generated from Equation 1. Generally, $\mu$ is 3.9 for high randomness. By varying the initial conditions using $\mu$ and $C_n$, and value of $n$, the different chaotic signal can be generated, which is demonstrated in later sections.

In this work, four different sets of oscillator's coefficients $\{x_1 = 0.01, 0.02, 0.03, 0.04;\ x_2 = -0.1,\ x_3 = -0.01,\ x_4 = 0.01\}$ and initial conditions are used to understand the effect of parameters on the chaos as well on the architecture. The initial condition and the coefficient together act as a secret – user defined keys. Figure 3 (a) – (d) shows the various combinations of chaotic signals generated and studied for the above set of parameters. Figure 3(e) – (g) depicts the chaotic double scroll attractor pattern obtained for the different conditions. These signals so generated are utilized later in Section III while training the ELM model II. Also, the unique key in the chaotic signal generation is utilized while assessing the security concerns in key sensitivity and space analysis in Section IV.

### B. FRACTAL DIMENSIONS AND HIGUCHI ALGORITHM

Fractal dimensions (FD) are the characteristic non – integer numbers which are used to characterize features like texture, degree of surface coarseness etc., of an image. The fractal dimensions of a digital image are relative to the pixel value of the image. Considered as a tool to calculate the image's complexity, according to the Mandelbrot's Hausdorff dimensions [33], FD can be defined using Equation 2 as

$$A(r) = \mathcal{A}r^{2-D}$$

$$f = \mathcal{A}D/C \quad (2)$$

Where, $A(r)$ represents the curve surface, $\mathcal{A}$ denotes the true area of the surface, $D$ depicts the FD, $C$ is the present constant, and $f$ is a factor to determine FD of the image. The advantage of using fractal dimensions is that it becomes really infeasible to identify the watermark bits in the cover image. There exist various methods of calculating fractal dimensions like box-counting, spectral analysis, Katz algorithm, Higuchi's algorithm etc. [34-41]. In this work, Higuchi's algorithm is used to calculate the FD of the image blocks [35].

Higuchi's algorithm is a technique which is generally used to calculate the fractal dimension, $D$ of the time series

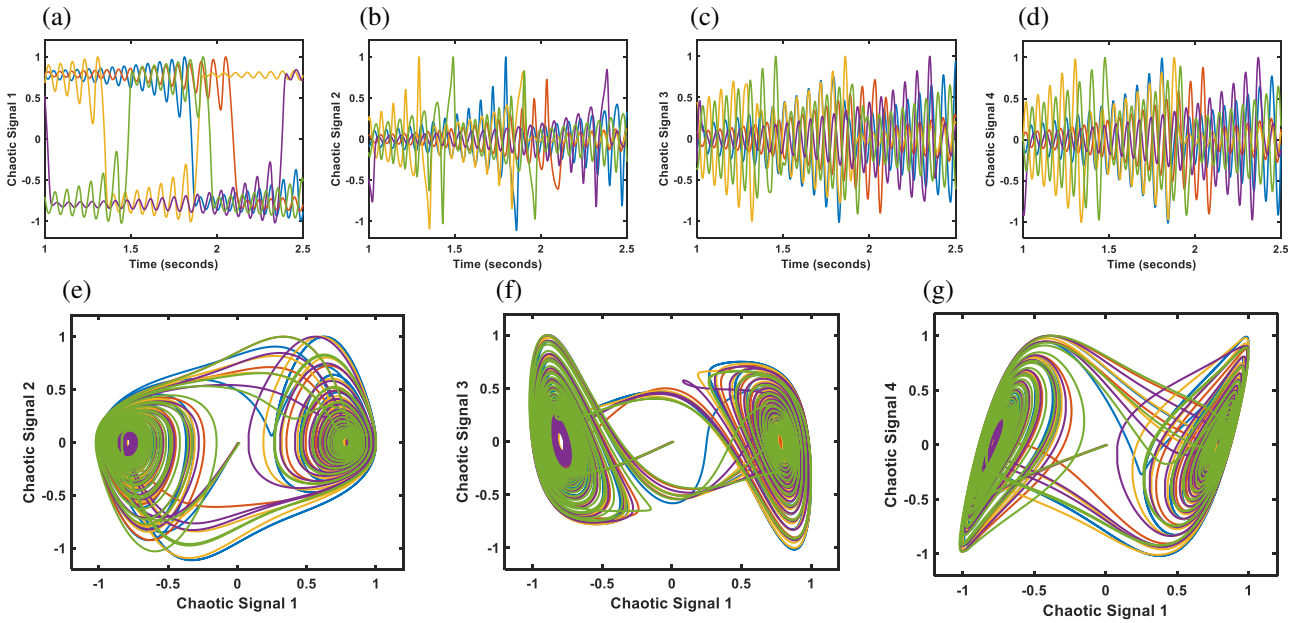


**FIGURE 3. (a)-(d) Five different chaotic signals generated for four different initial configurations and user defined keys (e)-(g) Signal 2,3 and 4 as a function of Signal 1 depicting double scroll attractor pattern in the generated chaotic signals.**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079319, IEEE Access



data. Consider N samples of finite series at regular intervals as described in Equation (3):

$$X(1), X(2), X(3) \dots, X(N) \tag{3}$$

From this series, $X_k^m$ can be obtained as defined in Equation 4 as:

$$X_k^m = \{X(m), X(m+k), \dots, X\left(m + \left[\frac{N-m}{k}\right]k\right)\} \tag{4}$$

Where $m \in [1, k]$ which denotes the initial time, $k$ is the interval time and $[.]$ denotes the Gauss notation. Thus, $k$ set of new time series is obtained, which defines the FD of the original series. Further, the length of each curve associated with $X_k^m$ is defined by Equation 5 as:

$$L_m(k) = \frac{\left(\sum_{i=1}^{\left[\frac{N-m}{k}\right]} (X(m+ik) - X(m+(i-1)k))\right)\left(\frac{N-1}{\left[\frac{N-m}{k}\right]k}\right)}{k} \tag{5}$$

where $\frac{N-1}{\left[\frac{N-m}{k}\right]}$ is the normalization factor for the curve $X_k^m$. According to Higuchi, the average length of the curve follows the power law i.e. $\langle L(k) \rangle \propto k^{-D}$ , and thus the curve is fractal with dimensions $D$. In the curve of $ln(L(k))$ versus $ln(1/k)$, the slope of the least squares linear best fit is the estimate of the fractal dimension.

### C. ARNOLD TRANSFORM

Due to simplicity and periodicity, Arnold Transform is widely used in digital image scrambling [42-45]. Image scrambling is a method of rearranging the entire pixel array of the image, thus resulting in a completely disorganized and encrypted image. The transform follows one to one mapping and has one significant feature of periodicity, according to which the original image after scrambling can be restored back after several cycles. The number of permutations performed while rearranging is of significant importance as it acts as a secret key. This pseudo − random behavior of Arnold transform is characteristic and is of utmost importance as without knowing the number of cycle or sequence used, one cannot decrypt the image [46].

Consider a square image of $N \times N$ representing a 2 – D image, then the transformation of the pixel point $(x, y)$ of the original image to pixel point $(x', y')$ of the encrypted image can be represented by Equation 6 as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} mod\ (N) \tag{6}$$

where $mod$ is the mathematical modular operator. The factor $N$ is the image's size dependent parameter which decides the transformation's periodicity or the period $p$. The cover image is first scrambled in iterative procedures of $n$ cycles, which acts as a key in the de − scrambling process, and the scrambled image is retrieved using iterative inverse Arnold transform for $p - n$ cycles. The inverse Arnold Transform, which can be used to restore back the original image, can be represented by Equation 7:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} + \begin{bmatrix} N \\ N \end{bmatrix} mod\ (N) \tag{7}$$

Figure 4 depicts the procedure of Arnold transform. It can be understood as an iterative process of stretching and shearing, and translating back to the square matrix, resulting in an invertible matrix that preserves the image features but looks distorted. The inverse Arnold transform follows the same steps but in reverse order.

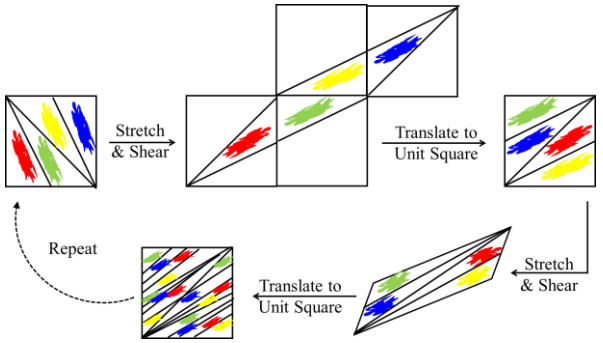


**FIGURE 4. Illustration showing scrambling through Arnold Transform.**

### D. HUMAN VISUAL SYSTEM MODEL & MAMDANI FIS

The Human Visual System (HVS) model is one of the widely used techniques in image processing for analyzing subjective qualities and to improve the imperceptibility of the image. It helps strengthen the technique by making the watermark adaptive to the original image's features, thus ensuring excellent imperceptibility. Literature has reported various approaches for calculating the HVS like Barni *et. al.* [47] has proposed three rules of disturbs and sensitivities to texture and regions of the image. Delaigle *et. al.* [48] proposed FFT based HVS masking procedural, and Martin *et.al.* [49] has used isotropic contrast function with frequency to spatial domain transformation. In the Watson model, three feature: luminance, edge and contrast, are considered while defining the perpetual quality of the image [50]. The luminance sensitivity and edge sensitivity are computed using a threshold value, and contrast sensitivity computed using a variance.

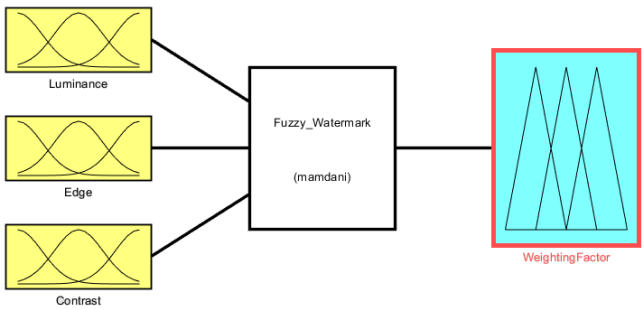


**FIGURE 5: Block diagram for Mamdani Based Fuzzy Interface System.**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079319, IEEE Access



In a Mamdani Fuzzy Interface System, the output, which is a fuzzy set, is controlled using a linguistic control rules. The fuzzy set is derived using the output membership function and the implication method of FIS. The multiple fuzzy sets so obtained are then combined using the FIS aggregation method. Finally, defuzzification is carried out to get the final crisp values. A Mamdani type FIS system setup in Matlab is depicted in Figure 5.

### E. HISTOGRAM OF ORIENTED GRADIENTS

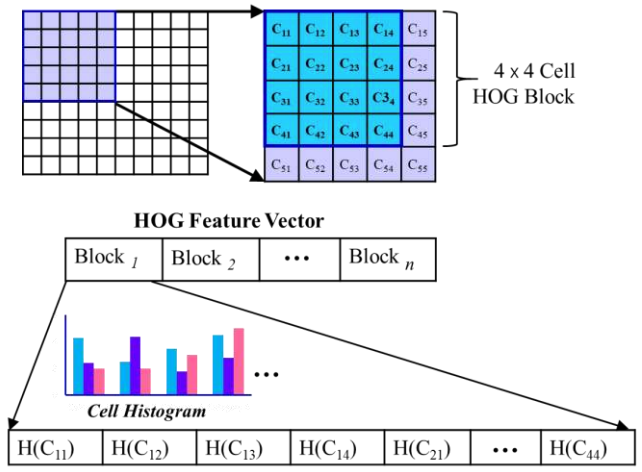


**FIGURE 6: Illustration showing Histogram of Oriented Gradients (HOG) feature calculations.**

The Histogram of oriented gradients (HOG) is used as a feature descriptor in image recognition domains where the image features are extracted from the edges of the local regions of the target image. This characterizes the orientation and magnitude values of the pixels in the two-dimensional planes. The HOG can be realized using a grid of ($2 \times$ $Number\ of\ Bins$) rose plots spaced uniformly. The rose plot depicts the gradient orientation distribution in a HOG cell, and in each cell, the contribution of each orientation is depicted by the length of each petal.

Consider the image shown in Figure 6 with $4 \times 4$ cells constituting a HOG block. The HOG feature extraction considers this as $m \times n$ block and generates a feature vector consisting of HOG blocks arranged in sequential order. Each HOG block is represented using a cell histogram which is ($1 -$ $Number\ of\ Bins$) where the number of bins represents the dimensions of orientation histogram. The larger the number of bins, the better is the orientation details. Figure 5 illustrates the HOG feature extraction procedure.

### C. EXTREME LEARNING MACHINES

Extreme Learning Machine (ELM) is a single layer feed − forward neural network (SLFN). The ELM works on allocating the input weights and hidden layers biases using continuous determined probability distribution systems, and finally, the output weights are calculated using the Moore − Penrose method [51][52]. Consider the training samples $(x_i, y_i)_{i=1,2\ldots N}$ with $x^i \in \mathbb{R}^n$, $y^i \in \mathbb{R}^m$ and $N$ is the number of hidden neurons. The output of the single hidden layer feed − forward neural network with activation function $g: \mathbb{R} \to \mathbb{R}$ can be modeled using Equation 8:

$$\sum_{k=0}^{N} \beta_k g(\langle w_{k1}, x_i \rangle + b_k) = y_i \qquad \forall\, i \in 1,2 \ldots N \tag{8}$$

where $w_k = (w_{k1}, w_{k2,\ldots} w_{kn})$ defines the weighting vector connecting $k^{th}$ hidden neuron to the input node and $\beta_k = (\beta_{k1}, \beta_{k2,\ldots} w_{kn})$ is the weighting factor connecting the $k^{th}$ hidden neuron to the output node and $b_k$ is the threshold bias of $k^{th}$ hidden neuron. The factors $w_k$ and $\beta_k$ are randomly selected in accordance with the continuous probability distribution function. Thus, Equation 8 can be interpreted as Equation 9:

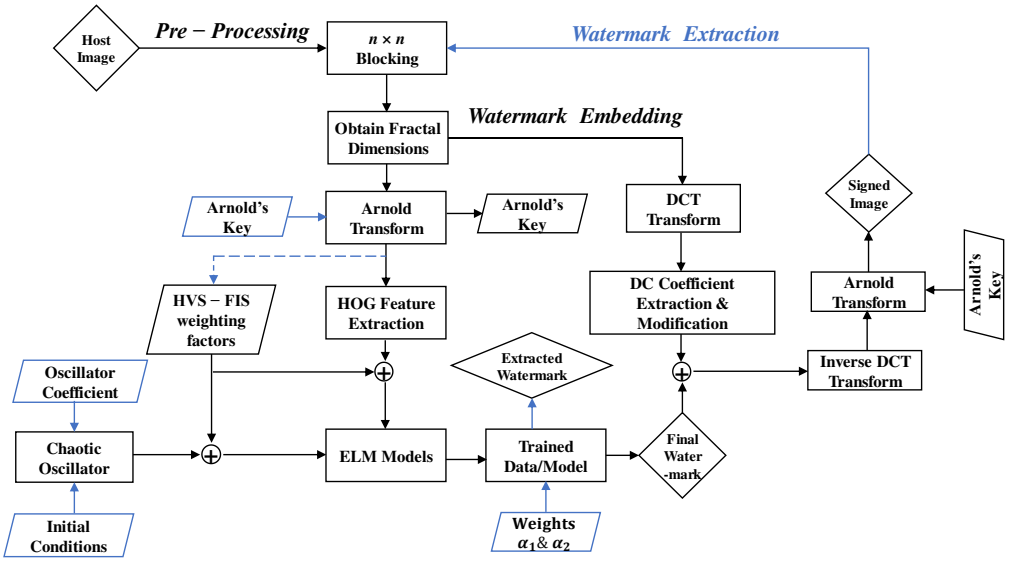


**FIGURE 7: Flowchart of the proposed watermarking technique.**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079319, IEEE Access



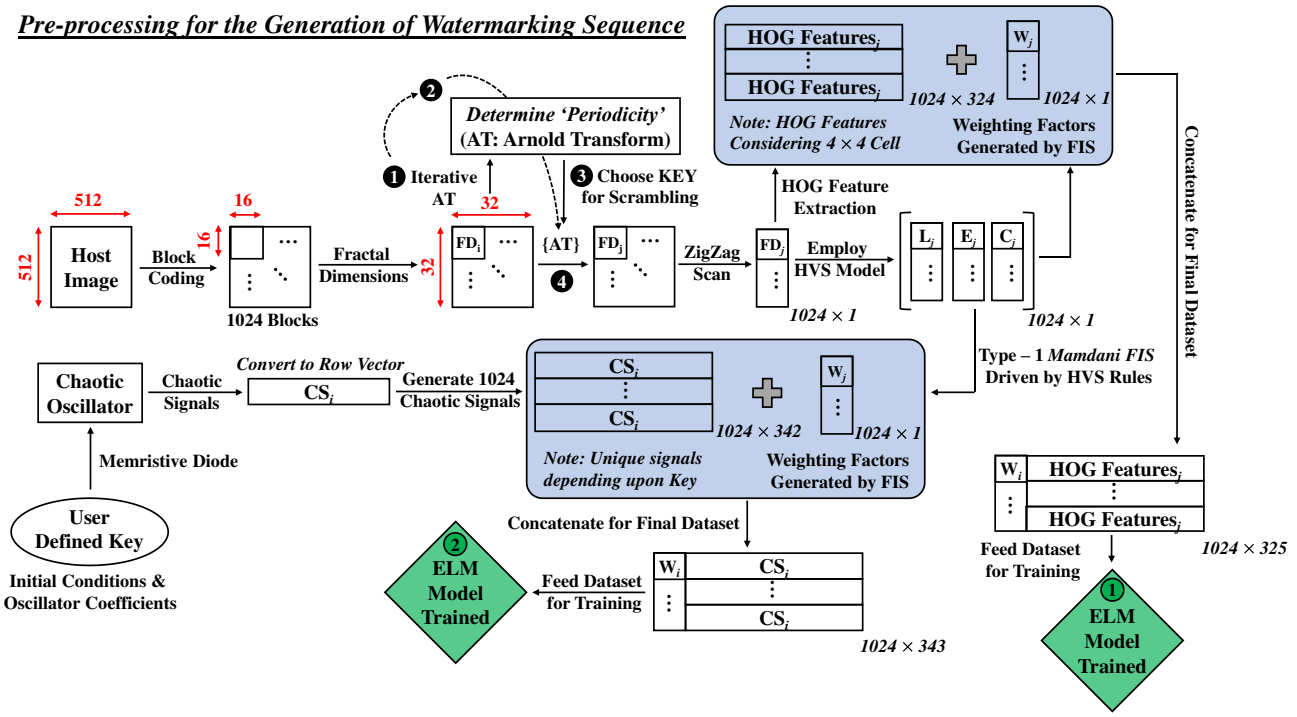


**FIGURE 8:** **Flowchart explaining pre-processing steps for the watermark sequencing.**

$$H\beta = Y \tag{9}$$

The solutions of Equation 8 can be described using Equation 10:

$$\beta' = H^{*}Y \tag{10}$$

where $H^{*}$ is the Moore – Penrose inverse of the hidden – layer output matrix $H$.

## III. METHODOLOGY

Figure 7 presents the flowchart of the proposed watermarking methodology and is illustrated in detail through Figure 8, 11 and 12. The simulations are carried using MathWorks MATLAB [53]. Four different grayscale host images - Lena, Airfield, Peppers and Mandrill were used for watermark embedding. The complete drill followed in this work can be categorized into three groups: (1) Pre – processing for the generation of watermarking sequence, which is covered in subsection A – F; (2) Watermark Embedding to obtain the final signed image, which is covered in subsection G; and (3) Semi – Blind Watermark extraction to retrieve back the watermarking coefficients from the signed image which is covered in subsection H.

### A. CALCULATIONS OF FRACTAL DIMENSIONS

At first, the 512 × 512 sized host images is subjected to block coding, which generates 1024 blocks of size 16 × 16. Each of this block is then featured into their respective unique fractal dimensions ($[\mathrm{FD_i}]$) using Higuchi's algorithm and stored in a 32 × 32 matrix.

To ensure better security, two-level encryption has been introduced in the proposed watermarking scheme, first using Arnold transform and second using the chaotic sequence used to train the ELM Model 2.

### B. SCRAMBLING USING ARNOLD TRANSFORM

The 32 × 32 blocks are iteratively passed to the scrambling system, and the corresponding periodicity is determined. The image's periodicity is a function of image size and dimensions. For the present work, a periodicity of 24 is obtained. The fractal matrix can then be scrambled for any number or iterations, which then act as the unique key for image encryption. The transformed matrix ($[\mathrm{FD_j}]$) is converted into a vector of size 1024 × 1 using zig – zag scan ((32 × 32) → (1024 × 1)), which forms the base for all the further processing.

### C. HUMAN VISUAL SYSTEM AND FUZZY INTERFACE

The three features − luminosity, edge and contrast terms are obtained and stored in $[L_j]$, $[E_j]$ and $[C_j]$ respectively. These blocks are then fed to the Mamdani FIS system driven by a set of 10 interference rule as proposed by Lou *et. al.* [50], and the 1024 × 1 matrix representing single weighted output is obtained. Figure 8 shows the interference process illustrating the role of membership functions and weighting factors. Figure 9 shows the weighting factor generated as a function of the three indices: Luminous, edge and contrast. The weighting factors as generated by the FIS system is of utmost importance as it is later used as labels or weight for

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079319, IEEE Access



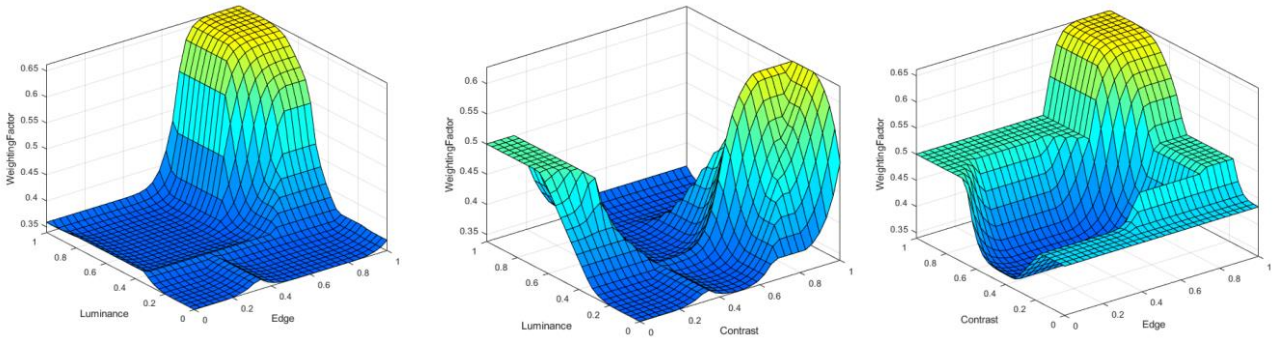


**FIGURE 10: Evolution of weighting factor as a function of luminous, edge and contrast terms.**

individual elements and is used in the preparation of the training data for the neural network.

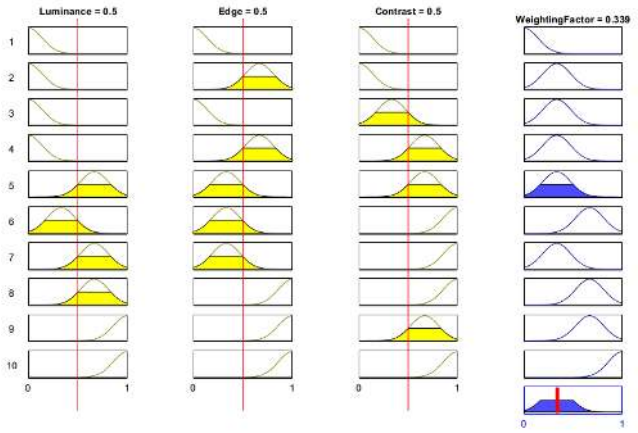


**FIGURE 9: Illustration explaining the role of membership functions and effect on weighting factors in HVS-FIS.**

### D. HISTOGRAM OF ORIENTED GRADIENTS

The image blocks corresponding to the fractal array after zig – zag scan is parsed for extracting HOG features. The features are extracted using 4 × 4 cells and using 9 bins. Since the input is 1024 × 1, $\mathbf{1/3rd}$ of the total size, i.e. 324, results in the output of size 1024 × 324. This array is concatenated with the weighting factor earlier generated using the FIS Mamdani system and is fed into the ELM Model 1 for training.

### E. CHAOTIC SIGNALS GENERATION AND DATA PREPARATION

The chaotic signals generated through Memristor based non – linear oscillators are further used for improving the robustness of the proposed scheme. The signals are stored in a row vector, and a total of 1024 unique signals are generated. For each signal, 342 sample points are taken for building up the training dataset. The so obtained 1024 × 342 matrix is modified by concatenating the weighting factor matrix generated by FIS, and the final dataset is fed to the ELM Model 2 for subsequent training and testing.

### F. EXTREME LEARNING MACHINE MODEL

In this work, two ELM models are used. One machine is trained with the matrix obtained from the HOG feature extraction procedure, and the other ELM model is trained with

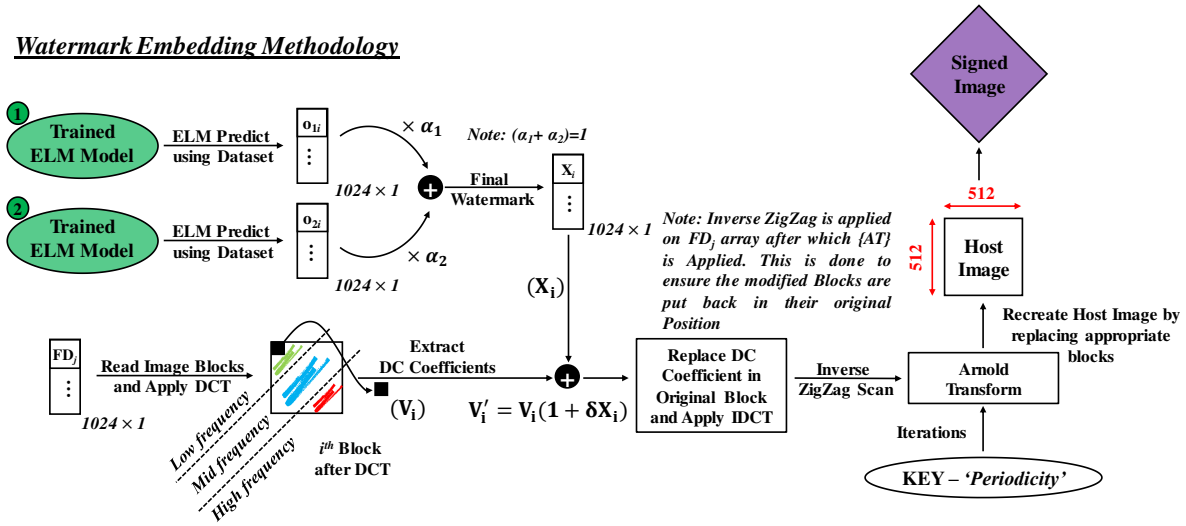


**FIGURE 11: Flowchart explaining water embedding methodology.**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079319, IEEE Access



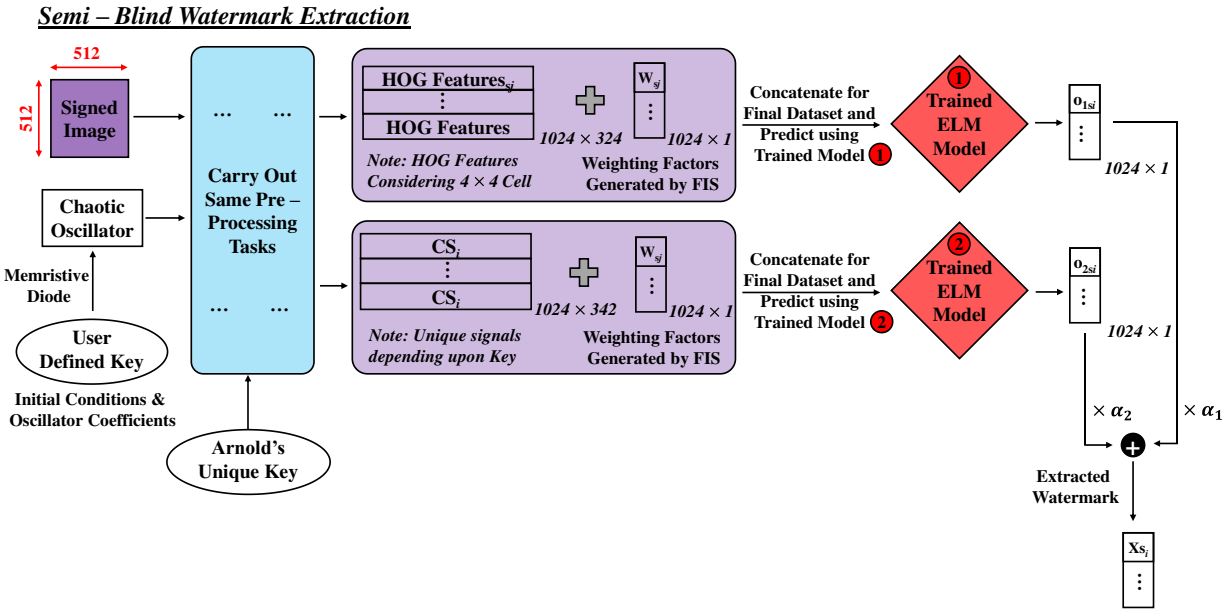


**FIGURE 12:** Flowchart explaining the procedure for semi-blind watermarks extraction.

the unique sets of chaotic signals. The ELM is a single-layer network with 10 hidden neurons, and the sigmoid activation function is used. The output of each ELM is a 1024 × 1 vector. The two ELM dataset obtained as output are combined using the weighted sum method, and the final matrix sequence is obtained. The output so obtained is the required watermarked sequence $(X_i)$.

### G. WATERMARK EMBEDDING

The ELM Model 1 and 2 so trained using HOG features and chaotic signals, respectively, were used in weights of 50% to generate the final watermarking coefficients denoted by $(X_i)$. Consequently, for watermark embedding, each block of $[FD_j]$ is read in order as dictated using zig – zag scan and transformed using discrete cosine transform (DCT). The 2D – DCT converts the image bock from Spatial to Frequency domain that can be categorized into low, mid and high frequency bands. The entire block in the frequency domain consists of AC coefficients, except at the (0,0) index, which corresponds to the DC coefficient. The said DC coefficient for all the blocks are extracted for embedding the watermarking coefficients. The watermark embedding method used here is the one proposed by Cox *et. al.* [55]. The formula used is described in equation 11 as:

$$V_i' = V_i(1 + \delta X_i) \quad (11)$$

where $V_i$ represents the host image coefficient and $\delta$ is the scaling factor. The extracted DC coefficient is used as $V_i$ in the above Equation. The modified value of the DC coefficient so obtained is used to replace the previous DC coefficients, and inverse discrete cosine transform (IDCT) is performed to retransform the signal back to the spatial domain. Since Arnold Transform was used in the preliminary stages for scrambling, inverse zig – zag scan is applied and, the original block locations are obtained after subjecting it to the Arnold Transform for the remaining iterations. This restores the original image.

### H. SEMI – BLIND WATERMARK EXTRACTION

Watermark extraction involves the same set as pre – processing tasks and prediction using ELM. The signed image and the chaotic signal generated using Chua's circuit are subjected to the same pre – processing tasks described in subsection A – F. The HOG features-based dataset and the chaotic signal dependent dataset is generated, and the prepared data is tested using the trained ELM models. The 1024 × 1 sized output of the ELM models are added using a sum of weighted mean, and finally, the watermark is extracted.

## IV. RESULTS AND DISCUSSION

The proposed approach discusses a double encryption technique for image watermarking. The architecture uses several distinguishing features enhancing its robustness, imperceptibility and security. Not only this, but the processing time is also found to competitive with state − of − the – art watermarking techniques.

Arnold transform used for image scrambling eliminates the spatial correlation of image pixel, making the watermarking process distinguishably robust. The unique key so generated forces a layer of encryption, thus enhances the security. Furthermore, the presented approach is based on using index features of the image – luminance, edge and contrast, which ensures a high level of security and excellent resistance to watermarking attacks. Considering the geometric attacks, the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079319, IEEE Access



proposed scheme has HOG features as one of the base for training models as histogram distribution of an image are generally invariant under attacks. Considering the need to have efficient processing of color images in copyrights, the proposed algorithm can also be used for colored images. It is due to this feature; the proposed algorithm is reliable and applicable to a vast variety of digital media. The second level of encryption is based on the chaotic signals, which usually is based on using the pseudo − random and dynamic property of signals. A chaotic oscillator is developed using Memristor, which acts as a chaotic signal generator. Since training is also based on these signals, another key is required in order to perform proper decryption, which further makes the system secure. The results so presented are experimentally validated using hardware setup with the simulation deck, which proves the proposed technique's possible implementations in real–time applications.

Imperceptibility or invisibility can be related to the concealment of digital watermarks. If a watermark can't be visually detected by the human visual system, it is said to be imperceptible. Watermark imperceptibility can be evaluated using metrics like Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE) and Structural Similarity Index (SSIM). The security depicts how much encrypted the process is. If the watermark is somehow extracted, it is impossible to embed it back to its original form without the encryption key, thus the watermark is said to be secure. The watermark's robustness is its ability to resist change in the embedded watermark due to changes in the watermark carrier data. Watermark robustness is generally measured with metrics like Normalized Cross-Correlation (NC).

The performance analysis in terms of robustness, security and imperceptibility of the proposed architecture is tested on digital attack benchmarks. The signed images are applied to 21 StirMark benchmarks [56], and then the watermark is extracted from the attacked images. The performance of the technique was assessed using various metrics discussed below.

### A. STRUCTURAL SIMILARITY INDEX

The Structural Similarity Index (SSIM) is the quality assessment index for measuring the similarity between the host image and the watermarked image. Numerically, SSIM lies in the range of 0 to 1 with 1 perfect structural similarity between two images and 0 being the worst case depicting that the two images doesn't share the similarity. Equation 12 realizes the SSIM metric as follows:

$$SSIM(I,I') = [l(I,I')]^{\alpha}.[c(I,I')]^{\beta}[s(I,I')]^{\gamma} \quad (12)$$

Where $I(i,j)$ denotes the host image, $I'(i,j)$ denotes the signed image, $\alpha, \beta$ and $\gamma$ are the exponents for luminance, contrasts and structural terms respectively and $l(I,I'), c(I,I')$ and $s(I,I')$ can be described by Equation 13:

$$l(I,I') = \frac{2\mu_I\mu_{I'}+C_1}{\mu_I^2+\mu_I^2+C_1}$$

$$c(I,I') = \frac{2\sigma_I\sigma_{I'}+C_2}{\sigma_I^2+\sigma_I^2+C_2}$$

$$s(I,I') = \frac{\sigma_{II'}+C_3}{\sigma_I\sigma_{I'}+C_3} \quad (13)$$

where $\mu_I$ and $\mu_{I'}$ are the local means, $\sigma_I$ and $\sigma_{I'}$ are the local standard deviation and $\sigma_{II'}$ are the cross − covariance for images $I$ and $I'$. When $\alpha = \beta = \gamma = 1$ and $c_3 = c_2/2$, the SSIM metric simplifies to and is expressed in Equation 14.

$$SSIM(I,I') = \frac{(2\mu_I\mu_{I'}+C_1)(2\sigma_{II'}+C_2)}{(\mu_I^2+\mu_{I'}^2+C_1)(\sigma_I^2+\sigma_{I'}^2+C_2)} \quad (14)$$

For a 2D image, the SSIM is generally calculated using a sliding Gaussian window or block and is made to traverse the image pixel by pixel generating the SSIM quality map. Figure 12 depicts the SSIM as a function of the scaling factor. As observed from the figure, the SSIM factor for all the four images under consideration lies well near 1, indicating that the watermarked images are similar and hence offers an excellent imperceptibility and invisibility to the watermarked image. Table 1 shows the objective metrics for different host image with watermark with respect to the scaling factor. Near to 1 value of SSIM over the entire range of scaling factor point towards the excellent imperceptibility achieved.

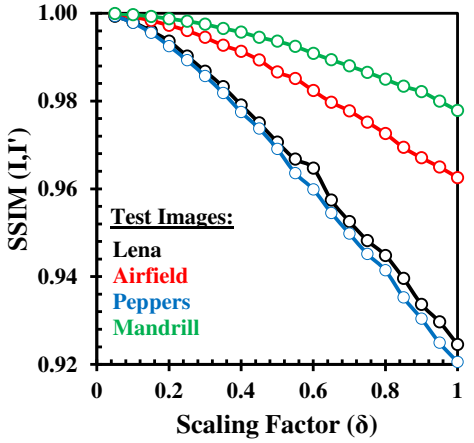


**FIGURE 13: SSIM metric for different images under consideration as a function of scaling factor.**

### B. NORMALIZED CROSS CORRELATION

Normalized Cross Correlation (NC) is used to judge the image similarity between two images. Since the NC is found to be less sensitive to linear change in the amplitude of illumination, it is one of the most commonly used metric in image processing tasks. Generally confined in the range between −1 and 1, the NC provides an edge over cross correlation by easing the threshold value selection and in other analysis. In this work, the cross – correlation of the watermark and signed image both before and after attacks are studied. The

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079319, IEEE Access



former is done to assess the robustness of the semi – blind watermarking technique, while the latter is done to assess the overall performance of the proposed scheme on the image containing the watermarking coefficients. NC metrics can be represented using Equation 15:

$$NC(W,W') = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}[W(i,j)*W'(i,j)]}{\sum_{i=1}^{m}\sum_{j=1}^{n}[W(i,j)]^2}$$

$$NC(I,I') = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}[I(i,j)*I'(i,j)]}{\sum_{i=1}^{m}\sum_{j=1}^{n}[I(i,j)]^2} \quad (15)$$

where, p and q are the rows and columns of the watermarking sequence. $W(i,j)$ and $W'(i,j)$ are the watermarking coefficients of the original and watermarked sequence, respectively and $I(i,j)$ and $I'(i,j)$ denotes the host and the signed image respectively. The higher the value of NC, the better is the system's robustness to the vulnerable digital attacks. More specifically, a higher value of NC(W,W') dictates that the watermark so embedded into the original sequence and that extracted using semi – blind extraction routine shares a high degree of correlation with the original version indicating robustness of the extraction procedure. After subjecting the signed image to various attacks, the attacked image will lose its correlation with the original image, which is evident from the PSNR, NC(I,I'), SSIM, and BER values indicated in Table 2. This is done deliberately in an attempt to destroy the embedded watermark. However, when the watermarking coefficients are extracted from the attacked image, the cross – correlation of the watermark i.e. NC(W,W'), still remains close to unity, indicating the robustness of the proposed scheme.

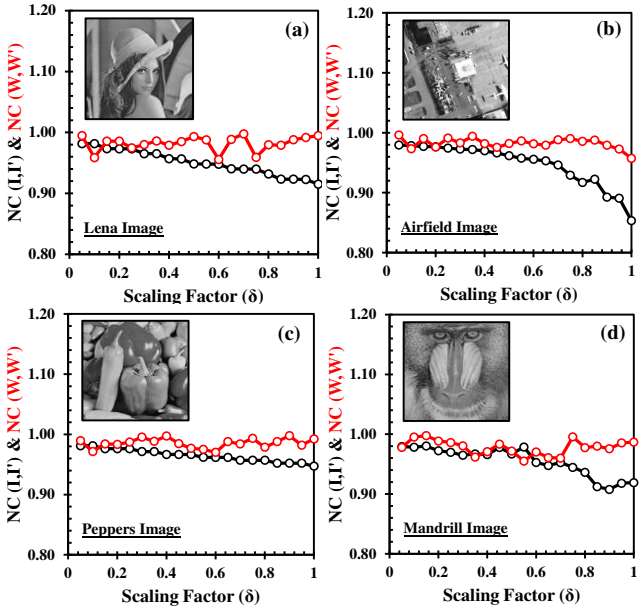


**FIGURE 14: NCC metric for signed image (black) and watermarked image (red) under consideration as a function of scaling factor.**

Figure 14 shows the NC of the signed and original image. As clearly visible, the high values of NC's for all four images prove the robustness of the proposed watermarking technique. In the presented scheme, the NC's value is above 90% in almost all cases depicts excellent robustness, and hence the resilience of the proposed algorithm. It is because, in the proposed scheme, the watermark embedding is carried out by modifying the DC coefficients of the adjacent blocks. This results in an invariably small change in the pixel domain hence leading to a better quality. Moreover, it is worthy to note that if AC coefficient from mid – frequency sub – band are chosen instead of the DC coefficient, the pixel domain changes can be further minimized, and the image quality can be enhanced. However, most image processing attacks are directed towards the mid – and high – frequency coefficients, which significantly deteriorates the watermarking coefficients, thereby defeating the main objective of robust image watermarking scheme.

### C. BIT ERROR RATE

The bit error rate (BER) represents the number of error bits received per unit time. Mathematically, it can be obtained by dividing the number of bits that have been altered while processing by the total of transferred bits. Mathematically, BER can be described using equation 16 as:

$$BER(I,I') = \frac{1}{mn}\left[\sum_{i=1}^{m}\sum_{j=1}^{n} I(i,j) \oplus I'(i,j)\right] \quad (16)$$

Where $I$ and $I'$ represent the original and the signed image, respectively. Ideally, BER is zero, which shows that the two images share a good proportion of resemblance. The various BER metrics of the images under consideration are summarized in Figure 14. The BER lies well below 0.50 for all scaling factors, which proves highly robust for the image processing operations and even close to 0 for the small scaling factor. Further, it is evident from the figure that as the scaling factor's value decreases, the system's robustness is also increased as far as the BER is concerned. However, the strength of watermarking coefficients gets significantly affected and becomes vulnerable to image processing attacks.

### D. PEAK SIGNAL TO NOISE RATIO

Peak Signal to Noise is the ratio of the signal's maximum possible power to the power of corrupting noise signal. PSNR is generally expressed in terms of the logarithm of mean square error.

$$MSE(I,I') = \frac{1}{mn}\left[\sum_{i=1}^{m-1}\sum_{j=1}^{n-1}[I(i,j) - I'(i,j)\right]^2 \quad (17)$$

$$PSNR(I,I') = 10\ log_{10}\left(\frac{2^v-1}{MSE}\right) \quad (18)$$

where $v$ is the minimum number of bits depicting the maximum intensity in a given image, $I(i,j)$ and $I'(i,j)$ denotes the host and the watermarked image, respectively, and $m$ and $n$ represent the number of rows and columns in the original image. The evolution of PSNR as a function of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079319, IEEE Access



scaling factor is depicted in Figure 14. An intersection between the PSNR and BER curves is used to determine the optimum scaling factor $(\delta_{Optimum})$ which is later used to assess the proposed watermark scheme through StirMark attacks. Further, if the intersection point of the PSNR and BER is traced, a shift in optimum scaling factor is obtained, enabling us to adjust the other parameters to enhance the overall system performance. The shift and the values of various metric can be related to the random weight allocation in the ELM [10]. It is worth mentioning that the ELM training and testing time in all the cases are extremely small and lies in the range of mili – seconds (70 – 90 msec for 20 iterations) which prompts the extremely high computational speed and testing time.

For watermark embedding procedure, the NC(I,I') lie close to unity, indicating that the signed image still has a high degree of correlation with the host image. Under semi blind extraction, NC(W,W') as observed, lies close to unity, indicating that the proposed scheme is able to extract the watermark, and that the extracted watermarking coefficients have a high degree of correlation with the coefficients that were embedded in the initial phase. Further, after deliberately subjecting the signed images to heavy image processing attacks, evident from the seriously degraded PSNR, BER, SSIM, and NC(I,I') values, the extracted watermark still has a high degree of correlation with the original watermarking coefficients.

**Table 1:** Computed SSIM, PSNR, BER and NC metric as a function of scaling factor ($\delta$)

| IMAGE: LENA | | | | | | IMAGE: AIRFIELD | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Scaling** | **SSIM(I,I')** | **PSNR** | **BER(I,I')** | **NC(I,I')** | **NC(W,W')** | **Scaling** | **SSIM(I,I')** | **PSNR** | **BER(I,I')** | **NC(I,I')** | **NC(W,W')** |
| 0.05 | 0.999328 | 65.28387 | 0.036629 | 0.981698 | 0.994682 | 0.05 | 0.999764 | 63.5187 | 0.050499 | 0.979444 | 0.99622 |
| 0.1 | 0.997955 | 59.14479 | 0.096039 | 0.98165 | 0.958774 | 0.1 | 0.999198 | 57.65621 | 0.124928 | 0.978305 | 0.973102 |
| 0.15 | 0.995981 | 56.04389 | 0.138351 | 0.973252 | 0.985741 | 0.15 | 0.99834 | 54.19128 | 0.182003 | 0.977174 | 0.990176 |
| 0.2 | 0.993631 | 53.66393 | 0.170708 | 0.973132 | 0.985701 | 0.2 | 0.99727 | 51.97557 | 0.220226 | 0.976489 | 0.976068 |
| 0.25 | 0.990224 | 51.53457 | 0.196701 | 0.973108 | 0.974542 | 0.25 | 0.996034 | 49.91912 | 0.255089 | 0.974388 | 0.990943 |
| 0.3 | 0.986785 | 49.93501 | 0.214005 | 0.965197 | 0.979981 | 0.3 | 0.994492 | 48.28276 | 0.283337 | 0.972698 | 0.983087 |
| 0.35 | 0.983263 | 48.7579 | 0.230629 | 0.965076 | 0.985906 | 0.35 | 0.992689 | 46.99864 | 0.307251 | 0.971836 | 0.994036 |
| 0.4 | 0.979071 | 47.54181 | 0.248665 | 0.956999 | 0.978981 | 0.4 | 0.991309 | 45.98438 | 0.323921 | 0.969789 | 0.981759 |
| 0.45 | 0.975014 | 46.50796 | 0.263985 | 0.956599 | 0.985055 | 0.45 | 0.98933 | 45.00636 | 0.340172 | 0.966141 | 0.975452 |
| 0.5 | 0.970629 | 45.58403 | 0.279602 | 0.948331 | 0.993136 | 0.5 | 0.986604 | 43.84746 | 0.356365 | 0.961671 | 0.981975 |
| 0.55 | 0.966744 | 44.84779 | 0.290146 | 0.948204 | 0.987827 | 0.55 | 0.98512 | 43.31841 | 0.362415 | 0.957495 | 0.986487 |
| 0.6 | 0.964679 | 44.66727 | 0.294899 | 0.947791 | 0.955592 | 0.6 | 0.982369 | 42.35925 | 0.371441 | 0.955648 | 0.981361 |
| 0.65 | 0.957403 | 43.39412 | 0.311218 | 0.940279 | 0.988642 | 0.65 | 0.979719 | 41.70847 | 0.382233 | 0.95299 | 0.979242 |
| 0.7 | 0.952541 | 42.66457 | 0.319344 | 0.939889 | 0.997524 | 0.7 | 0.977757 | 41.23108 | 0.386551 | 0.946277 | 0.988136 |
| 0.75 | 0.948202 | 42.05109 | 0.325455 | 0.939808 | 0.959141 | 0.75 | 0.975174 | 40.57445 | 0.394634 | 0.929115 | 0.990356 |
| 0.8 | 0.944813 | 41.65563 | 0.33168 | 0.931776 | 0.979719 | 0.8 | 0.972573 | 40.03375 | 0.398029 | 0.917049 | 0.985677 |
| 0.85 | 0.939597 | 40.97029 | 0.338982 | 0.923226 | 0.978955 | 0.85 | 0.969424 | 39.48496 | 0.40443 | 0.922522 | 0.987675 |
| 0.9 | 0.933649 | 40.43767 | 0.34713 | 0.923221 | 0.988045 | 0.9 | 0.967054 | 39.06541 | 0.408562 | 0.892613 | 0.97911 |
| 0.95 | 0.929661 | 40.05302 | 0.351585 | 0.922814 | 0.991674 | 0.95 | 0.964968 | 38.71499 | 0.409695 | 0.890757 | 0.972787 |
| 1 | 0.924527 | 39.55364 | 0.357666 | 0.915114 | 0.994659 | 1 | 0.962557 | 38.37611 | 0.4119 | 0.853191 | 0.957295 |
| **IMAGE: PEPPER** | | | | | | **IMAGE: MANDRILL** | | | | | |
| **Scaling** | **SSIM(I,I')** | **PSNR** | **BER(I,I')** | **NC(I,I')** | **NC(W,W')** | **Scaling** | **SSIM(I,I')** | **PSNR** | **BER(I,I')** | **NC(I,I')** | **NC(W,W')** |
| 0.05 | 0.999362 | 64.48765 | 0.04261 | 0.981052 | 0.989643 | 0.05 | 0.999943 | 67.60804 | 0.022774 | 0.979446 | 0.977927 |
| 0.1 | 0.997795 | 58.74412 | 0.10112 | 0.981052 | 0.971526 | 0.1 | 0.999664 | 59.47459 | 0.127434 | 0.978232 | 0.995257 |
| 0.15 | 0.995525 | 55.40029 | 0.144318 | 0.976197 | 0.984167 | 0.15 | 0.999274 | 56.26978 | 0.210762 | 0.980287 | 0.997608 |
| 0.2 | 0.992522 | 52.83352 | 0.177231 | 0.976193 | 0.983405 | 0.2 | 0.998742 | 53.98757 | 0.262131 | 0.972306 | 0.988732 |
| 0.25 | 0.989278 | 50.9982 | 0.204247 | 0.976152 | 0.98709 | 0.25 | 0.998182 | 52.34074 | 0.298889 | 0.969862 | 0.986089 |
| 0.3 | 0.985714 | 49.54127 | 0.225899 | 0.97146 | 0.995407 | 0.3 | 0.99748 | 50.8346 | 0.324287 | 0.964872 | 0.980703 |
| 0.35 | 0.981832 | 48.30797 | 0.242081 | 0.971413 | 0.988358 | 0.35 | 0.996535 | 49.3733 | 0.3507 | 0.966997 | 0.961839 |
| 0.4 | 0.977482 | 47.07869 | 0.259277 | 0.966678 | 0.99731 | 0.4 | 0.995693 | 48.33636 | 0.367294 | 0.965885 | 0.970867 |
| 0.45 | 0.973706 | 46.1935 | 0.271194 | 0.966572 | 0.984689 | 0.45 | 0.994499 | 47.1766 | 0.380981 | 0.977665 | 0.983578 |
| 0.5 | 0.969081 | 45.23497 | 0.280869 | 0.96649 | 0.97686 | 0.5 | 0.993616 | 46.36271 | 0.390648 | 0.967068 | 0.971919 |
| 0.55 | 0.96355 | 44.32647 | 0.298157 | 0.961805 | 0.975092 | 0.55 | 0.992441 | 45.64674 | 0.400566 | 0.978669 | 0.955079 |
| 0.6 | 0.959869 | 43.71914 | 0.303299 | 0.961749 | 0.9701 | 0.6 | 0.990872 | 44.75621 | 0.411041 | 0.952646 | 0.970331 |
| 0.65 | 0.954511 | 43.03996 | 0.316307 | 0.961713 | 0.988062 | 0.65 | 0.989391 | 44.12213 | 0.416847 | 0.947627 | 0.960897 |
| 0.7 | 0.949803 | 42.33428 | 0.323837 | 0.956986 | 0.984002 | 0.7 | 0.987996 | 43.4267 | 0.422462 | 0.952625 | 0.96003 |
| 0.75 | 0.945146 | 41.75548 | 0.330841 | 0.956954 | 0.993501 | 0.75 | 0.986538 | 42.90274 | 0.430283 | 0.944572 | 0.995528 |
| 0.8 | 0.941414 | 41.35166 | 0.338371 | 0.956949 | 0.978633 | 0.8 | 0.984965 | 42.30331 | 0.435211 | 0.936181 | 0.977584 |
| 0.85 | 0.935252 | 40.75559 | 0.346115 | 0.952202 | 0.987912 | 0.85 | 0.983358 | 41.81809 | 0.437599 | 0.912235 | 0.980045 |
| 0.9 | 0.930385 | 40.33894 | 0.354378 | 0.952124 | 0.997778 | 0.9 | 0.982141 | 41.50226 | 0.441505 | 0.907735 | 0.975702 |
| 0.95 | 0.924925 | 39.83766 | 0.359962 | 0.952122 | 0.982115 | 0.95 | 0.97999 | 40.93978 | 0.447456 | 0.918014 | 0.985511 |
| 1 | 0.920535 | 39.43099 | 0.368782 | 0.947338 | 0.992456 | 1 | 0.977836 | 40.41583 | 0.450058 | 0.91908 | 0.986692 |

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079319, IEEE Access



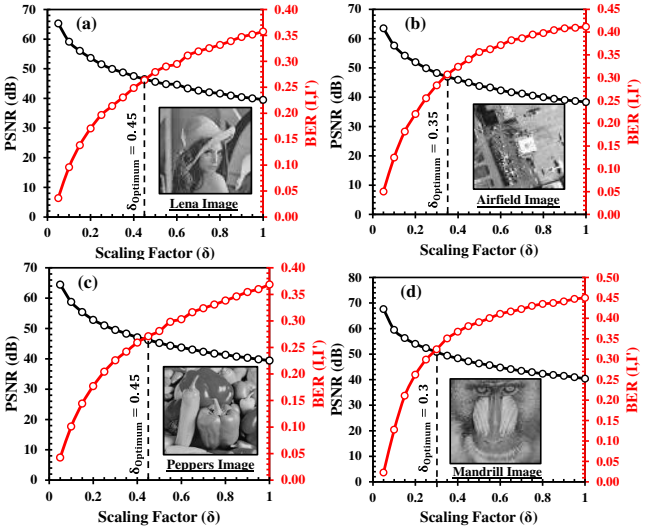


**FIGURE 15: PSNR and BER metrics for different images under consideration and optimization of the scaling factor.**

### *E. PERFORMANCE ANALYSIS UNDER ATTACKS*

Table 3 shows the computational performance of the images on StirMark benchmark when signed images are subjected to various image processing tasks. Two main cases are considered: (1.) Single attack and (2.) Hybrid attacks where the image is subjected to more than 1 attack at a time. The benchmark so chosen de – synchronizes the algorithm using random bilinear geometrical distortion and is one of the simplest tools for testing the robustness of the digital watermark architecture. Seriously degraded PSNR, SSIM, BER, and NC values for the image justify the StirMark attacks carried out on the image in a deliberate attempt to destroy the watermark and hence are an effective way to evaluate the performance of the algorithm [10]. Figure 16 depicts the hybrid attack. To comment on the robustness in severe conditions of digital attack, the signed image is subjected to 5 combined attack and the attacked image is visualized in Figure 16. The seriously degraded values of PSNR, SSIM, BER, and NC(I,I') demonstrate the intensity of image processing attacks so studied to evaluate the robustness of the proposed scheme From the NC(W,W') metrics, only 1.5% degradation is observed in case of image resizing which depicts excellence resilience of the technique to resizing attacks. In case of cropping and replacing which seems to be the worst case amongst all the single attack, the maximum degradation of 8.42% in NC(W,W') is recorded with is accompanied with 0.087, 0.250 and 27.16 dB change in BER, PSNR and SSIM index. While in case of filtering and noise, the algorithm is still able to extract the recognizable watermarking coefficients from the attacked images. Same can also be concluded from the subjective quality of the image as depicted.

To test the performance under extreme conditions, the proposed technique was tested under 5 simultaneous attacks. To the best of authors knowledge, a maximum of 2 − 3 combined attacks have been studied in [2] and there has been no work reported which deals with such extreme cases of 5 attacks covering blurring, filtering and noise operations simultaneously. The seriously degraded values of PSNR, SSIM, BER, NC(I,I') justify the worst case image processing attacks that were deliberately applied to destroy the watermarking coefficients in the singed image. However, a high value of NC(W,W') justifies the robustness of the proposed scheme against vulnerable attacks. To assess other traits, SSIM and PSNR have also been studied. SSIM which is originally based on image distortion model utilizes features like loss of correlation, luminance distortion and contrast distortion while the PSNR metrics are based on the mean square error are more consistent in studying the effect of gaussian noises [57]. However, since a variety of cases other than gaussian noise have also been considered, so it becomes crucial to understand the evolution of both the metrics under various conditions and obtain a tradeoff. In our case, the random weight allocation may be the reason to such sensitivity in SSIM and PSNR values. Overall, based on SSIM and PSNR metrics of image under attack, the visual quality of the image is distorted because of which it loses its correlation and structural similarities and the objective quality is slightly degraded. While a good value of NC(W,W) even after attacks reflects the extraction of original watermark is achievable and that the proposed scheme is robust and resilient to attacks.

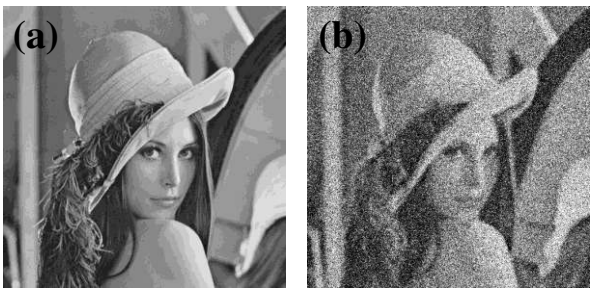


**FIGURE 16: Visualization of (a) Original host (Lena) Image (b) Attacked Lena image under hybrid attack (Gaussian Blur 2std dev + Median Filter (Aperture 5) + Salt and Pepper (5 % Noise) + Gaussian Noise (Var=0.05) + Speckle Noise (Var=0.05)).**

**Table 2:** Evaluation of Stirmark Attacks on the Signed Images for different Host Images.

| (a) IMAGE:LENA | | | | | |
|---|---|---|---|---|---|
| **Attacks** | **SSIM (I,I')** | **PSNR** | **BER (I,I')** | **NC (I,I')** | **NC (W,W')** |
| **No Attack** | **46.50796** | **0.975014** | **0.263985** | **0.956599** | **0.985055** |
| **Gaussian Blur 0.5 std dev** | 44.7357 | 0.929786 | 0.374512 | 0.766944 | 0.975721 |
| **Gaussian Blur 1 std dev** | 39.07138 | 0.69975 | 0.439423 | 0.760608 | 0.981301 |
| **Gaussian Blur 1.5 std dev** | 36.44224 | 0.576745 | 0.447742 | 0.753414 | 0.981155 |
| **Gaussian Blur 2 std dev** | 34.76434 | 0.492118 | 0.451866 | 0.753277 | 0.970236 |
| **Resizing (512->256->512)** | 39.44511 | 0.703132 | 0.441662 | 0.780613 | 0.980725 |

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079319, IEEE Access



| | | | | | |
|---|---|---|---|---|---|
| **Crop and Replace by 0 (1st Quarter)** | 17.99436 | 0.723865 | 0.209938 | 0.667611 | 0.916334 |
| **Crop and Replace by 0 (2nd Quarter)** | 16.8827 | 0.724864 | 0.203987 | 0.68718 | 0.959997 |
| **Crop and Replace by 0 (3rd Quarter)** | 19.57468 | 0.72302 | 0.176643 | 0.512134 | 0.902041 |
| **Crop and Replace by 0 (4th Quarter)** | 16.90798 | 0.724347 | 0.197239 | 0.185562 | 0.915208 |
| **Crop and Replace by 255 (1st Quarter)** | 17.60123 | 0.723177 | 0.455109 | 0.364769 | 0.965691 |
| **Crop and Replace by 255 (2nd Quarter)** | 17.98768 | 0.726073 | 0.45166 | 0.773957 | 0.990717 |
| **Crop and Replace by 255 (3rd Quarter)** | 15.8804 | 0.722346 | 0.430199 | 0.766801 | 0.966513 |
| **Crop and Replace by 255 (4th Quarter)** | 18.12537 | 0.72442 | 0.449089 | 0.165056 | 0.986961 |
| **Crop Right Half and Replace by 0** | 13.89417 | 0.481912 | 0.137611 | 0.011014 | 0.934375 |
| **Crop Right Half and Replace by 255** | 15.05882 | 0.482552 | 0.640224 | 0.336588 | 0.942151 |
| **Crop Left Half and Replace by 0** | 15.71487 | 0.477499 | 0.124802 | 0.0303 | 0.932215 |
| **Crop Left Half and Replace by 255** | 13.65684 | 0.47763 | 0.620415 | 0.02544 | 0.909443 |
| **Crop Top Half and Replace by 0** | 14.40665 | 0.481356 | 0.148666 | 0.303795 | 0.937184 |
| **Crop Top Half and Replace by 255** | 14.78896 | 0.481227 | 0.650696 | 0.356329 | 0.956216 |
| **Crop Bottom Half and Replace by 0** | 15.04208 | 0.48069 | 0.114197 | 0.01146 | 0.935854 |
| **Crop Bottom Half and Replace by 255** | 13.85742 | 0.479558 | 0.614933 | 0.335225 | 0.933472 |
| **Median Filter (Aperture 3)** | 40.54053 | 0.687026 | 0.408997 | 0.323202 | 0.901306 |
| **Median Filter (Aperture 5)** | 37.09303 | 0.55275 | 0.429844 | 0.342411 | 0.986891 |
| **Salt and Pepper (2% Noise)** | 28.48008 | 0.498009 | 0.270176 | 0.877821 | 0.990378 |
| **Salt and Pepper (5% Noise)** | 24.46047 | 0.253429 | 0.277779 | 0.85816 | 0.901752 |
| **Gaussian Noise (Var=0.01)** | 26.08309 | 0.201363 | 0.49342 | 0.661472 | 0.964417 |
| **Gaussian Noise (Var=0.05)** | 19.71037 | 0.079789 | 0.497383 | 0.654308 | 0.919121 |
| **Speckle Noise (Var=0.01)** | 31.57855 | 0.41068 | 0.475464 | 0.75034 | 0.997434 |
| **Speckle Noise (Var=0.05)** | 24.86664 | 0.228578 | 0.489159 | 0.706485 | 0.922088 |
| **Rotate 90° Anticlockwise** | 17.16607 | 0.107029 | 0.513157 | 0.425617 | 0.977994 |
| **Rotate 180° Anticlockwise** | 16.82567 | 0.103229 | 0.49707 | 0.49127 | 0.967727 |
| **Rotate 270° Anticlockwise** | 17.16682 | 0.101436 | 0.481873 | 0.27259 | 0.961753 |
| **Gaussian Blur 2std dev + Median Filter (Aperture 5) + Salt and Pepper (5 % Noise) + Gaussian Noise (Var=0.05) + Speckle Noise (Var=0.05)** | **18.19022** | **0.028897** | **0.463955** | **0.190371** | **0.868516** |

| **(b) IMAGE: AIRFIELD** | | | | | |
|---|---|---|---|---|---|
| **Attacks** | **SSIM (I,I')** | **PSNR** | **BER (I,I')** | **NC (I,I')** | **NC (W,W')** |
| **No Attack** | **46.99864** | **0.992689** | **0.307251** | **0.971836** | **0.994036** |
| **Gaussian Blur 0.5 std dev** | 40.21519 | 0.941193 | 0.425785 | 0.789518 | 0.989273 |
| **Gaussian Blur 1 std dev** | 32.42935 | 0.64835 | 0.455547 | 0.759776 | 0.977642 |
| **Gaussian Blur 1.5 std dev** | 30.00568 | 0.477403 | 0.460854 | 0.751798 | 0.996658 |
| **Gaussian Blur 2 std dev** | 28.58221 | 0.361408 | 0.463734 | 0.757064 | 0.955167 |
| **Resizing (512->256->512)** | 32.66961 | 0.658211 | 0.454712 | 0.807615 | 0.982503 |
| **Crop and Replace by 0 (1st Quarter)** | 18.06318 | 0.739076 | 0.217331 | 0.642429 | 0.905661 |
| **Crop and Replace by 0 (2nd Quarter)** | 14.68989 | 0.737749 | 0.241173 | 0.669916 | 0.918845 |
| **Crop and Replace by 0 (3rd Quarter)** | 18.12552 | 0.739094 | 0.221333 | 0.618322 | 0.91356 |
| **Crop and Replace by 0 (4th Quarter)** | 15.57428 | 0.737913 | 0.235321 | 0.101424 | 0.962962 |
| **Crop and Replace by 255 (1st Quarter)** | 16.48988 | 0.739586 | 0.463032 | 0.729635 | 0.901668 |
| **Crop and Replace by 255 (2nd Quarter)** | 22.01176 | 0.740097 | 0.482147 | 0.690327 | 0.992229 |
| **Crop and Replace by 255 (3rd Quarter)** | 16.82641 | 0.739256 | 0.469723 | 0.710463 | 0.941931 |
| **Crop and Replace by 255 (4th Quarter)** | 19.2331 | 0.740843 | 0.47369 | 0.145056 | 0.962049 |
| **Crop Right Half and Replace by 0** | 12.11178 | 0.490421 | 0.167957 | 0.083865 | 0.932025 |
| **Crop Right Half and Replace by 255** | 17.40206 | 0.49155 | 0.649887 | 0.113814 | 0.958451 |
| **Crop Left Half and Replace by 0** | 15.09571 | 0.490928 | 0.131523 | 0.690967 | 0.925136 |
| **Crop Left Half and Replace by 255** | 13.65379 | 0.491432 | 0.628658 | 0.660544 | 0.888134 |
| **Crop Top Half and Replace by 0** | 13.0586 | 0.490101 | 0.152805 | 0.694583 | 0.935882 |
| **Crop Top Half and Replace by 255** | 15.42507 | 0.491112 | 0.638657 | 0.659265 | 0.905156 |
| **Crop Bottom Half and Replace by 0** | 13.66389 | 0.489331 | 0.154182 | 0.087134 | 0.940327 |
| **Crop Bottom Half and Replace by 255** | 14.86717 | 0.492822 | 0.636536 | 0.068593 | 0.913149 |
| **Median Filter (Aperture 3)** | 33.49103 | 0.639618 | 0.435936 | 0.120415 | 0.999819 |
| **Median Filter (Aperture 5)** | 30.03254 | 0.437871 | 0.454655 | 0.119048 | 0.984948 |
| **Salt and Pepper (2% Noise)** | 28.03788 | 0.680522 | 0.312439 | 0.642973 | 0.984851 |
| **Salt and Pepper (5% Noise)** | 24.16729 | 0.446003 | 0.315498 | 0.641987 | 0.982815 |
| **Gaussian Noise (Var=0.01)** | 26.29173 | 0.405914 | 0.476814 | 0.309781 | 0.956684 |
| **Gaussian Noise (Var=0.05)** | 19.96853 | 0.191405 | 0.481586 | 0.218869 | 0.902811 |
| **Speckle Noise (Var=0.01)** | 30.57089 | 0.614661 | 0.456928 | 0.569419 | 0.961548 |
| **Speckle Noise (Var=0.05)** | 23.9128 | 0.373482 | 0.471809 | 0.775895 | 0.886692 |
| **Rotate 90° Anticlockwise** | 15.30807 | 0.101532 | 0.512699 | 0.421124 | 0.995651 |

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079319, IEEE Access



| | | | | | |
|---|---|---|---|---|---|
| **Rotate 180° Anticlockwise** | 14.65399 | 0.102047 | 0.497646 | 0.527501 | 0.995045 |
| **Rotate 270° Anticlockwise** | 15.31122 | 0.101833 | 0.48103 | 0.480128 | 0.957886 |
| **Gaussian Blur 2std dev + Median Filter (Aperture 5) + Salt and Pepper (5 % Noise) + Gaussian Noise (Var=0.05) + Speckle Noise (Var=0.05)** | **17.77562** | **0.045185** | **0.451057** | **0.134961** | **0.872051** |

**(c) IMAGE: PEPPER**

| Attacks | SSIM (I,I') | PSNR | BER (I,I') | NC (I,I') | NC (W,W') |
|---|---|---|---|---|---|
| **No Attack** | **46.1935** | **0.973706** | **0.271194** | **0.966572** | **0.984689** |
| **Gaussian Blur 0.5 std dev** | 43.59527 | 0.902218 | 0.411102 | 0.954767 | 0.992837 |
| **Gaussian Blur 1 std dev** | 37.98662 | 0.642255 | 0.446003 | 0.947694 | 1.000017 |
| **Gaussian Blur 1.5 std dev** | 35.87764 | 0.54251 | 0.449184 | 0.937962 | 0.986577 |
| **Gaussian Blur 2 std dev** | 34.40202 | 0.475947 | 0.449642 | 0.927963 | 0.987919 |
| **Resizing (512->256->512)** | 37.37498 | 0.63783 | 0.452942 | 0.969443 | 0.981094 |
| **Crop and Replace by 0 (1st Quarter)** | 18.02933 | 0.723854 | 0.202629 | 0.901575 | 0.944601 |
| **Crop and Replace by 0 (2nd Quarter)** | 17.03702 | 0.722264 | 0.199154 | 0.915801 | 0.963796 |
| **Crop and Replace by 0 (3rd Quarter)** | 17.61236 | 0.722924 | 0.208836 | 0.915726 | 0.934425 |
| **Crop and Replace by 0 (4th Quarter)** | 18.77836 | 0.722337 | 0.199913 | 0.900392 | 0.933271 |
| **Crop and Replace by 255 (1st Quarter)** | 16.97384 | 0.723011 | 0.453835 | 0.915114 | 0.975468 |
| **Crop and Replace by 255 (2nd Quarter)** | 17.90226 | 0.722214 | 0.45097 | 0.915239 | 1.000822 |
| **Crop and Replace by 255 (3rd Quarter)** | 16.77587 | 0.722541 | 0.460968 | 0.901511 | 0.974543 |
| **Crop and Replace by 255 (4th Quarter)** | 16.22294 | 0.721318 | 0.45039 | 0.929924 | 0.94844 |
| **Crop Right Half and Replace by 0** | 14.82174 | 0.477226 | 0.12936 | 0.310157 | 0.93883 |
| **Crop Right Half and Replace by 255** | 13.98327 | 0.477307 | 0.629692 | 0.298993 | 0.925329 |
| **Crop Left Half and Replace by 0** | 14.81779 | 0.481239 | 0.143036 | 0.161373 | 0.938528 |
| **Crop Left Half and Replace by 255** | 13.87392 | 0.479892 | 0.641678 | 0.148206 | 0.930975 |
| **Crop Top Half and Replace by 0** | 14.50228 | 0.480911 | 0.132965 | 0.162313 | 0.943185 |
| **Crop Top Half and Replace by 255** | 14.41703 | 0.480956 | 0.631363 | 0.149011 | 0.952007 |
| **Crop Bottom Half and Replace by 0** | 15.15661 | 0.478602 | 0.139301 | 0.032161 | 0.936491 |
| **Crop Bottom Half and Replace by 255** | 13.49122 | 0.477916 | 0.640186 | 0.298936 | 0.917342 |
| **Median Filter (Aperture 3)** | 40.23302 | 0.636505 | 0.423611 | 0.904701 | 0.977683 |
| **Median Filter (Aperture 5)** | 38.05011 | 0.532768 | 0.438004 | 0.904681 | 0.990717 |
| **Salt and Pepper (2% Noise)** | 28.1799 | 0.506766 | 0.274563 | 0.801505 | 0.962378 |
| **Salt and Pepper (5% Noise)** | 24.31545 | 0.261639 | 0.282997 | 0.771512 | 0.976992 |
| **Gaussian Noise (Var=0.01)** | 26.18434 | 0.20622 | 0.491333 | 0.893283 | 0.983873 |
| **Gaussian Noise (Var=0.05)** | 19.82711 | 0.084706 | 0.495605 | 0.728862 | 0.907096 |
| **Speckle Noise (Var=0.01)** | 31.64505 | 0.441033 | 0.468826 | 0.757381 | 0.99272 |
| **Speckle Noise (Var=0.05)** | 24.91428 | 0.243569 | 0.481167 | 0.669836 | 0.954434 |
| **Rotate 90° Anticlockwise** | 16.4794 | 0.112083 | 0.50642 | 0.126578 | 0.986287 |
| **Rotate 180° Anticlockwise** | 16.90851 | 0.120504 | 0.497471 | 0.156952 | 0.99178 |
| **Rotate 270° Anticlockwise** | 16.47869 | 0.11164 | 0.490543 | 0.283956 | 0.98075 |
| **Gaussian Blur 2std dev + Median Filter (Aperture 5) + Salt and Pepper (5 % Noise) + Gaussian Noise (Var=0.05) + Speckle Noise (Var=0.05)** | **18.22476** | **0.033351** | **0.464767** | **0.307428** | **0.850535** |

**(d) IMAGE: MANDRILL**

| Attacks | SSIM (I,I') | PSNR | BER (I,I') | NC (I,I') | NC (W,W') |
|---|---|---|---|---|---|
| **No Attack** | **50.8346** | **0.99748** | **0.324287** | **0.964872** | **0.980703** |
| **Gaussian Blur 0.5 std dev** | 37.79769 | 0.949584 | 0.451973 | 0.652473 | 0.991784 |
| **Gaussian Blur 1 std dev** | 29.87719 | 0.632381 | 0.483749 | 0.783224 | 0.976952 |
| **Gaussian Blur 1.5 std dev** | 28.14575 | 0.430244 | 0.489082 | 0.843479 | 0.97433 |
| **Gaussian Blur 2 std dev** | 27.33262 | 0.307486 | 0.490646 | 0.88074 | 0.987898 |
| **Resizing (512->256->512)** | 29.6846 | 0.640205 | 0.482456 | 0.873878 | 0.967229 |
| **Crop and Replace by 0 (1st Quarter)** | 17.97788 | 0.742716 | 0.240761 | 0.819321 | 0.934412 |
| **Crop and Replace by 0 (2nd Quarter)** | 17.29128 | 0.742393 | 0.24218 | 0.847154 | 0.978103 |
| **Crop and Replace by 0 (3rd Quarter)** | 17.09279 | 0.741396 | 0.251293 | 0.851018 | 0.984276 |
| **Crop and Replace by 0 (4th Quarter)** | 17.79154 | 0.740594 | 0.252766 | 0.842445 | 0.984986 |
| **Crop and Replace by 255 (1st Quarter)** | 17.15238 | 0.743205 | 0.489601 | 0.786474 | 0.92147 |
| **Crop and Replace by 255 (2nd Quarter)** | 17.94106 | 0.74141 | 0.49268 | 0.839919 | 0.97482 |
| **Crop and Replace by 255 (3rd Quarter)** | 18.22283 | 0.741894 | 0.502693 | 0.838879 | 0.988169 |
| **Crop and Replace by 255 (4th Quarter)** | 17.6576 | 0.740444 | 0.502743 | 0.800531 | 0.955298 |
| **Crop Right Half and Replace by 0** | 14.53374 | 0.492056 | 0.166252 | 0.255322 | 0.94631 |
| **Crop Right Half and Replace by 255** | 14.79622 | 0.490637 | 0.664871 | 0.363466 | 0.983404 |
| **Crop Left Half and Replace by 0** | 14.51201 | 0.492763 | 0.163612 | 0.296784 | 0.937844 |
| **Crop Left Half and Replace by 255** | 14.65437 | 0.493842 | 0.662651 | 0.296105 | 0.995125 |
| **Crop Top Half and Replace by 0** | 14.61918 | 0.49343 | 0.15242 | 0.266478 | 0.934161 |

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079319, IEEE Access



| | | | | | |
|---|---|---|---|---|---|
| **Crop Top Half and Replace by 255** | 14.5288 | 0.493426 | 0.653595 | 0.296544 | 0.980978 |
| **Crop Bottom Half and Replace by 0** | 14.42756 | 0.491552 | 0.173676 | 0.220073 | 0.941885 |
| **Crop Bottom Half and Replace by 255** | 14.92921 | 0.491721 | 0.672848 | 0.559735 | 0.996536 |
| **Median Filter (Aperture 3)** | 29.78747 | 0.641115 | 0.448967 | 0.780235 | 0.978819 |
| **Median Filter (Aperture 5)** | 27.37619 | 0.350543 | 0.470036 | 0.79391 | 0.996561 |
| **Salt and Pepper (2% Noise)** | 28.56558 | 0.731682 | 0.317188 | 0.771794 | 0.949707 |
| **Salt and Pepper (5% Noise)** | 24.59975 | 0.518284 | 0.335457 | 0.755571 | 0.937836 |
| **Gaussian Noise (Var=0.01)** | 26.09482 | 0.503124 | 0.492325 | 0.681406 | 0.964656 |
| **Gaussian Noise (Var=0.05)** | 19.59942 | 0.237642 | 0.496025 | 0.561678 | 0.995004 |
| **Speckle Noise (Var=0.01)** | 31.44182 | 0.714189 | 0.480965 | 0.828966 | 0.985137 |
| **Speckle Noise (Var=0.05)** | 24.56973 | 0.47168 | 0.491634 | 0.811121 | 0.879708 |
| **Rotate 90° Anticlockwise** | 18.87803 | 0.105965 | 0.505821 | 0.61348 | 0.99931 |
| **Rotate 180° Anticlockwise** | 18.89277 | 0.105194 | 0.497177 | 0.698274 | 0.932344 |
| **Rotate 270° Anticlockwise** | 18.87788 | 0.100886 | 0.488358 | 0.393969 | 0.993063 |
| **Gaussian Blur 2std dev + Median Filter (Aperture 5) + Salt and Pepper (5 % Noise) + Gaussian Noise (Var=0.05) + Speckle Noise (Var=0.05)** | **17.64072** | **0.025453** | **0.467468** | **0.796696** | **0.83836** |

### *F. COMPUTATIONAL TIME ANALYSIS*

The evolution of PSNR, SSIM, BER, and NC metrics over 20 iterations has been depicted in Fig. 17. It is to be noted that the real – time performance of the proposed scheme is ensured by adopting a fast single layer feed – forward neural network (SLFN) called the ELM, which relies on random weight allocation in the ELM Model [10][51][52] which gives training and testing time spans for the trained models within 70 – 90 msec range. It is due to the random weight allocation, that the PSNR, SSIM, BER, and NC metrics so presented will vary with each iteration. However, it is to be noted that the variation in each metric is small, and in some iterations, outperform all the metrics when compared with the published literature, already summarized in Table 3. Further, since Memristors are not available yet as a commercial product, the OP – Amp realization of the Chua's diode, will demonstrate slight variations in the Chaotic Signals, which will depend on both the slew rate of the OP – Amp ICs and the tolerance of the circuit components so used for realizing the memristive oscillators.

Figure 17 depicts the evolution of different performance metrics for the different host images so considered in this work. Also presented is the computational time complexity for the embedding and semi – blind extraction procedures. From the analysis, it can be seen that embedding time spans over few seconds and semi − blind extraction routine which involves the entire routine depicted in Figure 12, including the generation of chaotic dataset spans within a few seconds. This lies in the expected range as previously reported in the literature [59][60]. As far as the payload is concerned, since the proposed scheme relies on the watermark sequence and not on the embedded image, a better payload capability can be inferred.

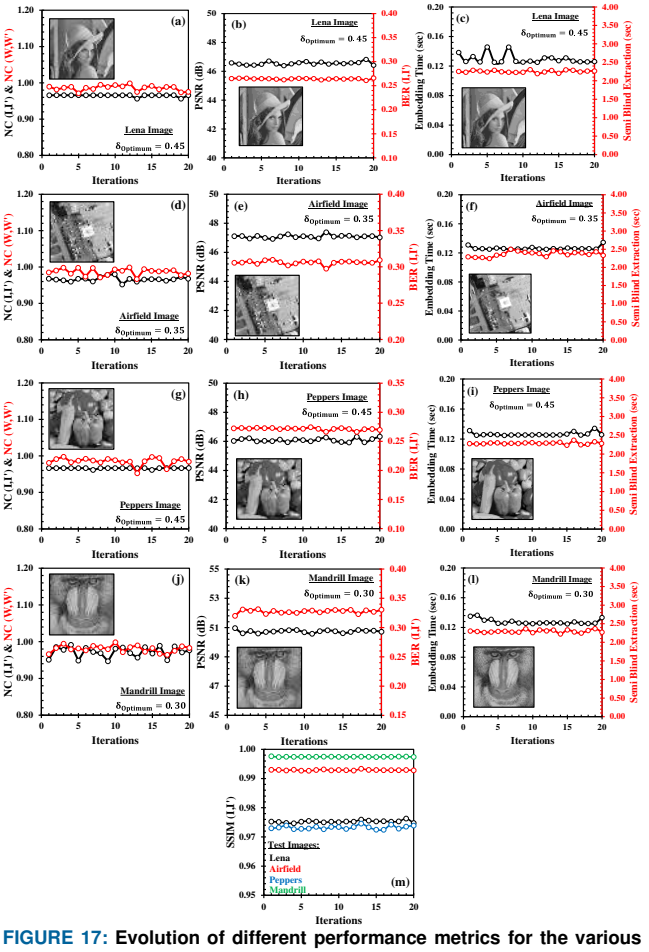


**FIGURE 17: Evolution of different performance metrics for the various host images over a set of 20 iterations. Also presented is the computational time analysis for embedding and semi – blind extraction procedures.**

### *G. SECURITY CONCERNS*

In order to test the security concerns, the scheme is tested in a number of false conditions. Table 3 summarizes the results of the key space and key sensitivity analysis performed on the Lena Image. In the complete method, three main keys enforces the security. First during the scrambling using Arnold transform which decides the Arnold's periodicity. As depicting in case 6, a mismatch in the Arnold's key greatly reduces the NC metric of the watermark to the value of 0.1964 which demonstrates inaccurate extraction of the watermarking coefficients. The initial conditions and the coefficients of the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079319, IEEE Access



memristive based chaotic oscillator acts as a second key. On varying theses secure keys, again a large degradation in the NC(W,W') metric is observed, which depicts the security of the proposed scheme. The third secure is generated during the final watermark generation using the trained data of the two ELM models. Table 3 discusses the four cases of weights mismatch from which it can be concluded that the random weight allocation in the ELM too enforces a high degree of security. It is worth mentioning that, these cases have been studied separately here. In actual scenario, all these will combine and further strengthen the key sensitivity of the proposed scheme.

**Table 3:** Security testing on the Lena Image for Scaling Factor = 0.45 with PSNR = 46.50, SSIM = 0.97, BER = 0.26 and NC = 0.95

| Sl No. | Cases | NC (W,W') |
|---|---|---|
| 1 | Watermark Weights Mismatch [$\alpha_1$ = 0.0, $\alpha_2$=1.0] | 0.2784 |
| 2 | Watermark Weights Mismatch [$\alpha_1$ = 1.0, $\alpha_2$=0.0] | 0.2696 |
| 3 | Watermark Weights Mismatch [$\alpha_1$ = 0.2, $\alpha_2$=0.8] | 0.2675 |
| 4 | Watermark Weights Mismatch [$\alpha_1$ = 0.8, $\alpha_2$=0.2] | 0.2719 |
| 5 | Mismatch in Circuit Components used for Memristive Oscillators [$R$ = 2kΩ, $C_1$ = 68nF, $C_2$ = 6.8nF] | 0.2697 |
| 6 | Mismatch in Initial Conditions used to Model Memristive Oscillators [Reversed initial conditions ($ic_n$) i.e. $ic_{1024}$, $ic_{1023}$ ... $ic_2$, $ic_1$] | 0.1603 |
| 7 | Incorrect Key used for Descrambling the Fractal Array [Arnold's Descrambling Key = 15] | 0.1964 |

### *H. COMPARISON WITH PUBLISHED LITERATURE*

The watermarking scheme proposed in this work based on Memristive Chaotic signals is compared with some state − of − the − art techniques. A comparison is presented in Table 4, which compares the NC values of the watermark after subjecting the signed images to StirMark attacks. Table 5 compares the PSNR and SSIM metric before subjecting the image to processing attacks, with various recent works. It is to be noted that the comparison with published literature is done ensuring similar host images and similar dimensions. All the references consider a host image of size $512 \times 512$. From Table 5 and 6, it can be concluded that the presented technique offers competitive performance in terms of robustness and imperceptibility in comparison to the semi – blind watermarking techniques published in literature. The table further establishes the superiority of the proposed technique in comparison to the state − of − the − art algorithms.

**Table 4:** Comparison of NC(W,W') in presence of image processing attacks with state − of − the − art techniques. Comparison is drawn considering similar host images with dimensions $512 \times 512$.

| Attack | Host Image: Lena | | | |
|---|---|---|---|---|
| | **Ref. [8]** | **Ref. [10]** | **Ref. [17]** | **This Work** |
| Gaussian Blur | - | 0.970 | - | 0.981 |
| Resizing | 0.975 | 0.980 | 0.997 | 0.981 |
| Crop & Replace | 0.920 | 0.890 | - | 0.960 |
| Median Filter | 0.969 | 0.970 | 0.918 | 0.987 |
| Salt & Pepper | 0.956 | 0.950 | - | 0.990 |
| Gaussian Noise | 0.921 | 0.950 | 0.983 | 0.981 |
| Speckle | 0.899 | 0.960 | - | 0.997 |
| | **Host Image: Peppers** | | | |
| | **Ref. [11]** | **Ref. [10]** | **Ref. [17]** | **This Work** |
| Gaussian Blur | - | 0.980 | - | 1.000 |
| Resizing | - | 0.980 | 0.989 | 0.982 |
| Crop & Replace | 0.940 | 0.950 | - | 1.000 |
| Median Filter | 0.890 | 0.990 | 0.911 | 0.999 |
| Salt & Pepper | 0.860 | 0.960 | - | 0.977 |
| Gaussian Noise | 0.940 | 0.960 | 0.980 | 0.984 |
| Speckle | - | 0.970 | - | 0.992 |

**Table 5:** Comparison of PSNR and SSIM in absence of image processing attacks with state − of − the − art techniques

| **(a) PSNR** | | | | |
|---|---|---|---|---|
| **Image** | **Ref. [8]** | **Ref. [10]** | **Ref. [17]** | **This Work** |
| Leena | 44.320 | 54.260 | 45.433 | 46.508 |
| Airfield | - | - | - | 46.999 |
| Peppers | - | 53.060 | 45.154 | 46.193 |
| Mandrill | - | 54.560 | 32.492 | 50.835 |
| **(b) SSIM** | | | | |
| **Image** | **Ref. [8]** | **Ref. [10]** | **Ref. [17]** | **This Work** |
| Leena | 0.960 | 0.990 | 0.994 | 0.975 |
| Airfield | - | - | - | 0.993 |
| Peppers | - | 0.990 | 0.983 | 0.974 |
| Mandrill | - | 0.990 | 0.961 | 0.997 |

## V. CONCLUSION AND FUTURE SCOPE

In this work, a novel secure and robust digital image watermarking technique is proposed. Double encryption is implemented using Arnold transform and Memristive chaotic signals. The proposed embedding technique is based on fractal dimensions extracted using Higuchi's algorithm, which is scrambled using Arnold transform. The unique key is stored for embedding and extraction procedures. To establish experimental validations, the chaotic signals were generated using Memristor based chaotic oscillator. The proposed scheme incorporates two ELM models trained using HOG

features and chaotic signals, and the output was combined using a sum of weighted averages. Semi – Blind Watermark embedding and extraction were followed, and an in-depth analysis using various figure of metrics were carried out. From the results presented, the proposed technique is proved to be robust and secure and establishes its significance in various applications.

## ACKNOWLEDGMENT

This work was supported in part by DBT Star College Programme, Deen Dayal Upadhyaya College, University of Delhi. The authors also wish to acknowledge University of Delhi for providing the necessary tools and financial assistance for the completion of this work.

## REFERENCES

[1] S. Voloshynovskiy, S. Pereira, T. Pun, J. J. Eggers, and J. K. Su, "Attacks on digital watermarks: Classification, estimation based attacks, and benchmarks," in *IEEE Commun. Mag.*, vol. 39, no. 8, pp. 118–126, 2001.

[2] N. A. Loan, N. N. Hurrah, S. A. Parah, J. W. Lee, J. A. Sheikh and G. M. Bhat, "Secure and robust digital image watermarking using coefficient differencing and chaotic encryption," in *IEEE Access*, vol. 6, pp. 19876-19897, 2018.

[3] F. Y. Shih, and S. Y. Wu, "Combinational image watermarking in the spatial and frequency domains," in *Pattern Recognition*, vol. 36, no. 4, pp. 969-975, 2003.

[4] S. Fazli and M. Moeini, "A robust image watermarking method based on DWT, DCT, and SVD using a new technique for correction of main geometric attacks," in *Optik*, vol. 127, no. 2, pp. 964-972, 2016.

[5] A.K. Singh, A. Kumar, M. Dave, and A. Mohan, "Hybrid technique for robust and imperceptible image watermarking in DWT–DCT–SVD domain," in *National Academy Science Letters,* vol. 37, no. 4, pp. 351-358, 2014.

[6] H. T. Hu , L. Y. Hsu, H. H. Chou, "An improved SVD based blind color image watermarking algorithm with mixed modulation incorporated," in *Information Sciences* vol. 519, pp. 161–182, 2016.

[7] X. Liu, J. Lou, H. Fang, Y. Chen, P. Ouyang, Y. Wang, B. Zou, and L. Wang, "A novel robust reversible watermarking scheme for protecting authenticity and integrity of medical images," in *IEEE Access*, vol. 7, pp. 76580-76598, 2019.

[8] E. Najafi, and K. Loukhaoukha, "Hybrid secure and robust image watermarking scheme based on SVD and sharp frequency localized contourlet transform," in *Journal of information security and applications*, vol. 44. pp. 144-156, 2019.

[9] N. M. Makbol, B. E. Khoo, T. H. Rassem, "Block-based discrete wavelet transform-singular value decomposition image watermarking scheme using human visual system characteristics," in *IET Image processing,* vol. 10, no. 1, pp. 34-52, 2016.

[10] A. Mishra, K. Sehra and G. Chetty, "Neuro Fuzzy Architecture for Gray Scale Image Watermarking using Fractal Dimensions," in *IEEE International Joint Conference on Neural Networks (IJCNN)*, Rio de Janeiro, Brazil, pp. 1-8, 2018.

[11] R. Gupta, A. Mishra, and S. Jain, "A Semi – Blind HVS based Image Watermarking Scheme Using Elliptic Curve Cryptography," in *Multimedia Tools and Applications*, vol. 7, pp. 1-26, 2017.

[12] R. Singhal, D. K. Jain, and I. A. Ansari, "Machine learning based blind color image watermarking scheme for copyright protection," in *Pattern Recognition Letters*, vol. 77, no. 15, 2021.

[13] Z. Yuan, D. Liu, X. Zhang, Q. Su, "New image blind watermarking method based on two-dimensional discrete cosine transform," in *Optik*, vol. 204, pp. 1-12, article no. 164152, 2020

[14] K. M. Hosny, M. M. Darwish and M. M. Fouda, "Robust Color Images Watermarking Using New Fractional-Order Exponent Moments," in *IEEE Access*, vol. 9, pp. 47425 – 47435, 2021.

[15] K. M. Hosny, M. M. Darwish, "Robust color image watermarking using invariant quaternion Legendre-Fourier moments," in *Multimed. Tools Appl*, vol. 77, pp. 24727 – 2475, 2018.

[16] K. M. Hosny and M. M. Darwish, "Resilient Color Image Watermarking Using Accurate Quaternion Radial Substituted Chebyshev Moments," ACM Trans. Multimedia Comput. Commun. Appl., vol. 15, no. 2, art. No. 46, pp 24727 – 24750, 2019.

[17] S. Varsha, and V. P. Vishwakarma, "Copyright protection using KELM-PSO based multi-spectral image watermarking in DCT domain with local texture information based selection," *Multimedia Tools and Applications*, vol. 86, no. 6, pp. 8667 – 8688, 2021.

[18] Z. Gong, N. Qin, and G. Zhang, "Visible watermarking in document images using two-stage fuzzy inference system," *The Visual Computer,* pp.1-12, 2021.

[19] A. Mishra, A. Goel, R. Singh, G. Chetty and L. Singh, "A novel image watermarking scheme using Extreme Learning Machine," The 2012 International Joint Conference on Neural Networks (IJCNN), Brisbane, QLD, Australia, pp. 1-6, 2012.

[20] W. Ding, Y. Ming, Z. Cao and C. -T. Lin, "A Generalized Deep Neural Network Approach for Digital Watermarking Analysis," in *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1 – 15, 2021, doi: 10.1109/TETCI.2021.3055520.

[21] H. Kandi, D. Mishra, and S. R. S. Gorthi, "Exploring the learning capabilities of convolutional neural networks for robust image watermarking", in *Computers & Security*, vol. 65, pp. 247 – 268, 2017.

[22] K. M. Hosny, M. M. Darwish, Kenli Li, and A. Salah, "Parallel Multi-Core CPU and GPU for Fast and Robust Medical Image Watermarking," in *IEEE Access,* vol. 76, no. 6, pp 8881 – 8900, 2018.

[23] F. Tohidi, M. Paul and M. R. Hooshmandasl, "Detection and Recovery of Higher Tampered Images using Novel Feature and Compression Strategy", in *IEEE Access*, vol. 9, pp. 57510 – 57528, 2021.

[24] H.T. Hu, L. Y. Hsu, and H. H. Chou, "An improved SVD-based blind color image watermarking algorithm with mixed modulation incorporated," in *Information Sciences,* vol. 519, pp. 161 – 182, 2020.

[25] U. A. Bhatti, L.Yuan, Z. Yu, J. Li, S. A. Nawaz, A. Mehmood, K. Zhang, "New watermarking algorithm utilizing quaternion Fourier transform with advanced scrambling and secure encryption," in *Multimedia Tools and Applications*, vol. 80, pp. 13367 - 13387, 2021.

[26] Y. Q. Zhang, Y. -R. Jia, X. Wang, Q. Niu and N. -D. Chen, "DeepTrigger: A Watermarking Scheme of Deep Learning Models Based on Chaotic Automatic Data Annotation," in *IEEE Access*, vol. 8, pp. 213296 – 213305, 2020.

[27] A. Ghaffari, "Image compression-encryption method based on two-dimensional sparse recovery and chaotic system," in *Scientific Reports*, vol. 11, no. 1, pp. 1 – 19, 2021.

[28] B. Bordel, R. Alcarria, T. Robles and M. S. Iglesias, "Data Authentication and Anonymization in IoT Scenarios and Future 5G Networks Using Chaotic Digital Watermarking," in *IEEE Access*, vol. 9, pp. 22378 – 22398, 2021

[29] A. K. Singh, S. Thakur, A. Jolfaei, G. Srivastava, M. D. Elhoseny and A. Mohan, "Joint Encryption and Compression-Based Watermarking Technique for Security of Digital Documents," in *ACM Transactions on Internet Technology,* vol. 21, no.1, pp. 1 – 20, 2021

[30] NI Multisim Education Edition. User Manual Available at: https://www.ni.com/en-in.html

[31] Kennedy, Michael Peter, "Robust op amp realization of Chua's circuit," in *Frequenz*, vol. 46, no. 3 pp. 66 – 80, 1992.

[32] S. Srinivasan, Memristor Based Chua's oscillator Online Available at: https://www.mathworks.com/matlabcentral/fileexchange/45438-memristor_based_chua_oscillator, MATLAB Central File Exchange. Retrieved March 22, 2021.

[33] M. Shishikura, "The Hausdorff dimension of the boundary of the Mandelbrot set and Julia sets," in *Annals of Mathematics*, vol. 147, no. 2, pp. 225-267, 1998.

[34] K. Falconer, "Fractal Geometry: Mathematical Foundations and Applications," in *Publisher: John Wiley and Sons*, pp. 18-19, 2005.

[35] T. Higuchi, "Approach to an irregular time series on the basis of the fractal theory," in *Physica D*, vol. 31, pp. 277–283, 1988.

[36] M. Katz, "Fractals and the analysis of waveforms," in *Comput. Biol. Med.*, vol. 18, no. 3, pp. 145-156, 1988.

[37] P. Grassberger and I. Procaccia, "Characterization of strange attractors," in *Phys. Rev. Lett.,* vol. 50, no. 5, pp. 346-349, 1983.

[38] A. Petrosian, "Kolmogorov complexity of finite sequences and recognition of different preictal EEG patterns," in *Proc. IEEE Symp. Computer-Based Medical Syst.*, pp. 212–217, 1995.

[39] R. Esteller, G. Vachtsevanos, J. Echauz and B. Litt, "A comparison of waveform fractal dimension algorithms," in *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 48, no. 2, pp. 177-183, Feb. 2001.
[40] N. Kuznetsov, and R. Volker, "Lyapunov Dimension for Dynamical Systems in Euclidean Spaces," in *Attractor Dimension Estimates for Dynamical Systems: Theory and Computation*. Springer, Cham, pp. 257-305, 2021.
[41] B. Mandelbrot, J. Wheeler, "The fractal geometry of nature", in *Am J Phys*, vol. 51, pp. 286-287, 1983.
[42] W. Lingling, Z. Jianwei, and G. Qi, "Arnold transformation and its inverse transformation," in *J. Micro Comput.*, vol. 14, pp. 1164–1167, 2009.
[43] G. Ye and K.W. Wong, "An efficient chaotic image encryption algorithm based on a generalized Arnold map," in *Nonlinear Dyn.*, vol. 69, no. 4, pp. 2079–2087, Sep. 2012.
[44] A. Asady, H. A. Jaleel, O. Q. Jumah, A. T. and S. S. Hreshee, "Robust Encryption System Based Watermarking Theory by Using Chaotic Algorithms: A Reviewer Paper," in *Journal of Physics: Conference Series*, vol. 1818, no. 1, pp. 1-13, 2021.
[45] P. K. Singh, B. Jana, and K. Datta, "Robust Watermarking Scheme for Compressed Image Through DCT Exploiting Superpixel and Arnold Transform," in *Proceedings of the Sixth International Conference on Mathematics and Computing*. Springer, Singapore, 2021.
[46] D. H. Saxena, and K. H. Singh, "Red-cyan anaglyph image watermarking using DWT, Hadamard transform and singular value decomposition for copyright protection," in *Journal of Information Security and Applications*, vol. 50, pp. 102424, 2020.
[47] M. Barni, B. Franco and A. Piva, "Improved wavelet-based watermarking through pixel-wise masking," in *IEEE transactions on image processing,* vol. 10, no. 5, pp. 783-791, 2001.
[48] J. F. Delaigle, C. D. Vleeschouwer, and B. Macq, "Watermarking algorithm based on a human visual model," in *Signal Process.*, vol. 66, no. 3, pp. 319–335, May 1998.
[49] M. Kutter, and S. Winkler, "A vision – based masking model for spread – spectrum image watermarking," in *IEEE Transactions on Image Processing*, vol. 11, no. 1, pp. 16 – 25, Jan. 2002, doi:10.1109/83.977879.
[50] Watson, Andrew B., et al, "Visual thresholds for wavelet quantization error," in *Human Vision and Electronic Imaging*. Vol. 2657. International Society for Optics and Photonics, 1996.
[51] Q. Y. Zhu, A. K.Qin, P. N. Suganthan, G. B. Huang, "Evolutionary extreme learning machine," in *Pattern recognition*, vol. 38, no.10, pp. 759-1763, 2005.
[52] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme Learning Machine: Theory and Applications," in *Neurocomputing*, vol. 70, pp. 489 – 501, 2006.
[53] Matlab 2020b Student Edition Available at: https://in.mathworks.com.
[54] D. C. Lou, M. C. Hu, and J. L. Liu, "Healthcare Image Watermarking Scheme Based on Human Visual Model and Back-Propagation Network," in *Journal of C.C.I.T*, vol. 37, no. , pp. 151- 162, 2008.
[55] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," in *IEEE Transactions on Image Processing*, vol. 6, no.12, pp. 1673-1687, 1997.
[56] F. A. P. Petitcolas, "Watermarking schemes evaluation," in *IEEE Signal Processing Magazine*, vol. 17, no. 5, pp. 58-64, Sept. 2000.
[57] D. I. M Setiadi, "PSNR vs SSIM: imperceptibility quality assessment for image steganography," in *Multimed Tools Appl.,* vol. 80, pp. 8423-8444, 2021.
[58] R. Wang, H. Shaocheng, P. Zhang, M. Yue, Z. Cheng and Y. Zhang, "A Novel Zero-Watermarking Scheme Based on Variable Parameter Chaotic Mapping in NSPD-DCT Domain," in *IEEE Access*, vol. 8, pp. 182391-182411, 2020, doi: 10.1109/ACCESS.2020.3004841.
[59] W. Wang, L. Yan, S. Liu, "A polar complex exponential transform based zero-watermarking for multiple medical images with high discrimination," in *Security and Communication Networks,* art. no. 6615678, 2021 doi: https://doi.org/10.1155/2021/6615678
[60] R. Sinhal, D. K. Jain and I. A. Ansari, "Machine learning based blind color image watermarking scheme for copyright protection," in *Pattern Recognition Letters*, vol. 145, pp.171-177., 2021.

**KHUSHWANT SEHRA** (Student Member, IEEE) received the B. Tech. Degree in Electronics, from University of Delhi, New Delhi, India, and the M. Tech Degree in Electronics and Communication Engineering, from University School of Information, Communication and Technology, Guru Gobind Singh Indraprastha University, New Delhi, India, in 2017 and 2019, respectively. He is presently working as Research Scholar in the Department of Electronic Science, University of Delhi, New Delhi, India. His research interest includes modelling, simulation and fabrication of GaN based HEMT devices. He has also worked on Image Processing including Digital Image Watermarking and development of Facial Recognition Systems for uncontrolled environments.

**SAMRIDDHI RAUT** (Student Member, IEEE) is currently pursuing her B. Tech in Electronics and Communication Engineering from Department of ECE, Maharaja Surajmal Institute of Technology, New Delhi. Her research interests include Nanoelectronics and Machine Learning.

**ASHUTOSH MISHRA** (Student Member, IEEE) is currently pursuing his Bachelors of Sciences (Hons.) in Electronics from Deen Dayal Upadhyaya College, University of Delhi. His area of interest is towards development and deployment of various devices related to Embedded Systems and IoT.

**SHWETA WADHERA** received her Master's in Computer Applications degree from Gurukul Kangri University in 1999. She is currently Assistant Professor in Department of Computer Science, Deen Dayal Upadhyaya College, University of Delhi, New Delhi, India. She has more than 20 years of teaching experience at Undergraduate level in Delhi University. She is currently pursuing her Ph. D. from Sharda University and her area of interest is Digital Watermarking and Sentiment Analysis.

**POONAM KASTURI** (Member, IEEE) was born in Punjab, India, on March 9, 1971. She received her B.Sc. (with honors) and M.Sc. degrees in Electronics from the University of Delhi, New Delhi, in 1992 and 1994, respectively. She submitted her doctoral thesis on "Two Dimensional Physics Based Analytical Modeling and Simulation of Silicon-on-Nothing (SON) MOSFET", to the Semiconductor Device Research Laboratory, Department of Electronic Science, University of Delhi and received her PhD degree in Electronics from University of Delhi in 2008. In 1995, she joined Deen Dayal Upadhayaya College, University of Delhi, as a Lecturer and, currently, she is an Associate Professor with the Department of Electronics. She has authored seven research papers in international journals and conferences and her research interests include Microelectronics and Verilog.

**GEETIKA JAIN SAXENA** (Member, IEEE) is an Associate professor at the Department of Electronics in Maharaja Agrasen College, University of Delhi. She received her B.Sc. and M.Sc. degrees in Electronics in 1998 and 2000, respectively. She did her doctorate in Modeling and Simulation of Optical Amplifiers in 2010 from University of Delhi. Her current research interests are Digital Image processing and Machine Learning Applications. She is also working on design, modeling and simulation of optical components for optical communication systems, analysis and design of Integrated Optical Waveguide and Devices with loss or gain. She has around 40 publications in International and National Peer reviewed journals and conferences to her credit.

**MANOJ SAXENA** (Senior Member, IEEE) received M. Sc., and Ph.D. degrees in Electronics from University of Delhi, New Delhi, in 2000 and 2006 respectively. He is currently Associate Professor in Department of Electronics, Deen Dayal Upadhyaya College, University of Delhi, New Delhi, India. He has authored or coauthored 300 technical papers in international journals and conference proceedings (including 81 papers in TED, TDMR, TNANO, EDL and IEEE conference proceedings) and has delivered 30 EDS Distinguished Lecture talks in past three years. He received "Highly Valued Volunteer for 2011-2012 EDS Chapters in South Asia, IEEE Region 10" and has reviewed extensively IEEE Journals and Conferences. He is Fellow-IETE (India) and Optical Society of India, Member of Institute of Physics (UK), IET (UK) and The National Academy of Sciences India (NASI). He is currently IEEE EDS Board of Governor Member and Associate Editor-in-Chief of IEEE EDS Newsletter – Region 10 South Asia (2016 - ) and was Vice Chair – IEEE EDS SRC Region 10 (2015-2017)