Journal of
Cybersecurity and Privacy

Review

Image Encryption Algorithms: A Survey of Design and Evaluation Metrics

Yousef Alghamdi and Arslan Munir *

Department of Computer Science, Kansas State University, Manhattan, KS 66506, USA; usef@ksu.edu
* Correspondence: amunir@ksu.edu

Abstract: Ensuring confidentiality and privacy is critical when it comes to sharing images over unsecured networks such as the internet. Since widely used and secure encryption methods, such as AES, Twofish, and RSA, are not suitable for real-time image encryption due to their slow encryption speeds and high computational requirements, researchers have proposed specialized algorithms for image encryption. This paper provides an introduction and overview of the image encryption algorithms and metrics used, aiming to evaluate them and help researchers and practitioners starting in this field obtain adequate information to understand the current state of image encryption algorithms. This paper classifies image encryption into seven different approaches based on the techniques used and analyzes the strengths and weaknesses of each approach. Furthermore, this paper provides a detailed review of a comprehensive set of security, quality, and efficiency evaluation metrics for image encryption algorithms, and provides upper and lower bounds for these evaluation metrics. Finally, this paper discusses the pros and cons of different image encryption approaches as well as the suitability of different image encryption approaches for different applications.

Keywords: image encryption; chaotic system; DNA encoding; compressive sensing; substitution; permutation

check for updates

Citation: Alghamdi, Y.; Munir, A. Image Encryption Algorithms: A Survey of Design and Evaluation Metrics. *J. Cybersecur. Priv.* **2024**, *4*, 126–152. https://doi.org/10.3390/jcp4010007

Academic Editor: Danda B. Rawat

Received: 5 January 2024
Revised: 16 February 2024
Accepted: 19 February 2024
Published: 23 February 2024

1. Introduction

With the increased use of social media and share of multimedia over communication networks, image encryption has become a critical research topic in information security, where researchers aim to protect the integrity and confidentiality of shared image data. Since multimedia data tend to have high amounts of redundancy (due to the inherent correlation between neighboring pixels), a specialized and robust image encryption algorithm is essential. As secure, widely used, and recommended encryption methods, such as the advanced encryption standard (AES), Twofish, RSA, etc., are not suitable for real-time multimedia data encryption (due to high computational requirements and slow encryption speeds), researchers have proposed various encryption techniques designed specifically for image encryption.

In this paper, several proposed state-of-the-art image encryption algorithm techniques are reviewed with an emphasis on their strengths and weaknesses. This paper classifies image encryption algorithms into seven different approaches based on the techniques used. These approaches are traditional ciphers, chaotic systems, DNA encoding, neural networks, compressive sensing, frequency domain, and meaningful sensing. Additionally, we discuss security, quality, and efficiency evaluation metrics, such as the correlation coefficient, histogram analysis, entropy, mean square error (MSE), the NIST SP 800-22 Test [1], etc. This paper provides a brief introduction to image encryption algorithms for researchers and practitioners starting in this field and aids them in understanding the current state of image encryption algorithms. Our main contributions in this article are as follows:

- Classification of image encryption into seven approaches based on the techniques used in the algorithms.
- A thorough review of a comprehensive set of security, quality, and efficiency evaluation metrics for image encryption.

- Calculation of the upper and lower bounds for each of the evaluation metrics for image encryption.

The rest of this paper is organized as follows: Section 2 discusses the general concept of image encryption algorithms and provides a detailed review of different approaches to image encryption algorithms. Section 3 presents the metrics used by researchers to evaluate image encryption algorithms. Section 4 discusses the advantages, disadvantages and the applications of different image encryption algorithms. Finally, Section 5 concludes this work.

2. Image Encryption Algorithms

Digital images are electronic files that consist of two-dimensional arrays of numbers with variable sizes and channel numbers (i.e., one channel for gray-scale images, three channels for color images, and four channels for color images with a transparency channel) that store pixel values. Each pixel represents a specific color. A digital image, I, is defined by size, $H \times W \times nc$, where H is the height, W is the width, and nc is the number of channels. Images usually contain private and personal information, such as personal and legal documents, medical images, military images, etc. When these types of images are shared over the internet or on an unsecured network, these images could be compromised by unauthorized access. Furthermore, most digital images tend to contain high redundancy due to the correlation between neighboring pixels. Image encryption algorithms aim to ensure the security of the image from unauthorized access by encrypting the plaintext image and producing a cipher image that can only be decrypted by the intended users. Image encryption algorithms use the confusion and diffusion [2] principle to produce cipher images and reduce the correlation between neighboring pixels. Confusion is attained by changing the values of each pixel in a digital image by a substitution map, where the values of each pixel in a plaintext image are changed by a substitution map such as an AES S-Box. Diffusion is achieved by changing the plaintext image pixels' location within an image itself using mathematical permutation operations. Diffusion helps in reducing the correlation between adjacent pixels that are present in the plaintext image. In image encryption, confusion and diffusion are controlled using a key or a set of keys to obscure and permute the plaintext image to produce the cipher image. Researchers have proposed a multitude of image encryption algorithms in the past decades. This section presents a review of image encryption algorithms based on the techniques used. Figure 1 illustrates the seven different approaches based on the techniques used.

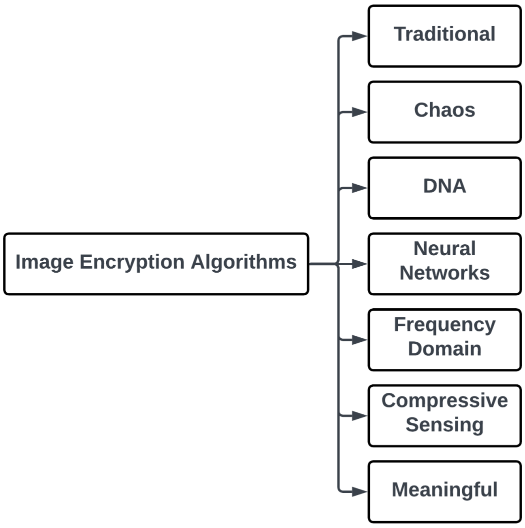

Figure 1. Different approaches to image encryption based on the used techniques.

2.1. Traditional Ciphers

At the moment, traditional ciphers are the most commonly used algorithms to encrypt images over networks. The images are converted into bit streams and then encrypted using one of the traditional encryption methods.

The advanced encryption standard (AES) [3] is a widely adopted and recommended encryption algorithm. AES is a symmetric key encryption with variable key lengths and round numbers. It uses 10 rounds of encryption for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. AES works on blocks with 128 bits, and each round involves a series of SPN-based (substitution–permutation network-based) operations on each block of data, that is, SubBytes, ShiftRows, MixColumns, and AddRoundKey. The AES can be used in various modes, but the electronic codebook (ECB) mode is not suitable for image encryption as it produces cipher images with noticeable patterns, where identical blocks of the image will result in producing identical blocks in the cipher image. To avoid this problem it is recommended to use AES in a cipher block chaining (CBC) mode.

Twofish [4] is another symmetric key block cipher that has 16 rounds and uses variable key lengths of 128, 192, and 256 bits. Twofish uses a Feistel network structure and works on 128-bit blocks. Each round involves a series of operations on each block of data, that is, substitution boxes, linear transform based on a maximum distance separable (MDS) code, pseudo-Hadamard transform (PHT), and a key addition function.

Stream ciphers have also been used in image encryption. They produce a pseudorandom keystream from a key and an initialization vector (IV) or nonce; this keystream is then combined with the plaintext image in a bitwise operation (usually XOR) to produce the cipher image. Trivium [5] is a synchronous stream cipher that is based on a nonlinear-feedback shift register (NLFSR), which generates a keystream of up to 2^{64} bits using an 80-bit secret key and an 80-bit IV. Trivium cipher has three phases: an initialization phase, a warm-up phase, and an encryption phase. In the initialization phase, the key and the IV are loaded into the 288-bit state register and all remaining bits are set to 0 (except for 3 specific bits, which are set to 1). In the warm-up phase, the cipher is clocked 1152 times without generating an output. In the encryption phase, all bits produced after the 1152th cycle are used to generate the keystream. To produce the cipher image, a keystream equal to the size of the plaintext image is XORed with the plaintext image. ChaCha20 [6] is another stream cipher that generates a keystream of up to 2^{64} bits using a 256-bit secret key, 64-bit nonce, and a 64-bit counter. In the ChaCha20 cipher, the 512-bit state is divided into a 4-by-4-word matrix containing 8 keywords, 4 constant words, and 4 words for the nonce and counter. ChaCha20 has 20 rounds; each round has 16 additions, 16 XORs, and 16 constant-distance rotations of 32-bit words. After 20 rounds, a 512-bit keystream is produced. To produce a cipher image, a keystream equal to the size of the plaintext image is XORed with the plaintext image.

2.2. Chaotic Systems

Some of the most widely used image encryption algorithms are based on chaos theory; this is due to several desirable properties, such as non-linearity, sensitivity to initial parameters, speed, and robustness [7]. In chaos-based algorithms, the pseudorandom chaotic sequences that are generated by the chaotic system are used to permute and diffuse a plaintext image [8].

Some of the chaotic maps that are commonly utilized in image encryption methods are as follows: logistic map [9,10], Baker map [11,12], Arnold map [13,14], tent map [15,16], hyperchaotic maps [17,18], etc. Researchers have proposed a plethora of image encryption algorithms based on chaotic systems; the following is a brief overview of some of the proposed algorithms for each chaotic system.

2.2.1. Logistic Map

A logistic map is one of the simplest non-linear and deterministic dynamical systems that are used to generate chaotic sequences. The logistic map can be defined as follows:

$$X_{n+1} = rX_n(1 - X_n), \tag{1}$$

where X_0 represents the initial parameter or starting value of the logistic map, with $0 \leq X_i \leq 1$. The parameter, r, is the control parameter or growth rate of the logistic map, and it has a range of (0, 4). However, chaotic behavior is observed only when the control parameter, r, is within the range of (3.56995, 4) [19]. The bifurcation diagram of the logistic map is illustrated in Figure 2.

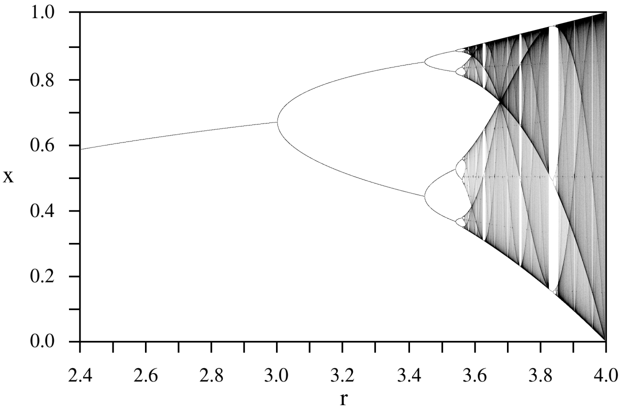

Figure 2. Logistic map bifurcation diagram.

Alghamdi et al. [8] proposed an image encryption algorithm based on the logistic map, permutations, and an AES S-box. The proposed algorithm employs SHA-2 hash based on the plaintext image, a pre-shared key, and an initialization vector (IV) to generate the initial parameters for the logistic map. The pseudorandom sequences generated by the logistic map are used to chaotically permute and substitute the image at the pixel level.

In [20], Rohith et al. proposed an image encryption algorithm for medical applications, using the key sequences of triple logistic maps. In the proposed algorithm, the logistic map is utilized to generate three different key sequences, $X1_i$, $X2_i$, and $X3_i$, with different initial values, where $X1_0 \neq X2_0 \neq X3_0$. These key sequences are converted into discrete key sequences, $K1_i$, $K2_i$, and $K3_i$, within the range of (0, 255). Then, a combined key sequence Ki is obtained through a bit-by-bit logical XOR operation between $K1_i$, $K2_i$, and $K3_i$, which is then used to encrypt the image by performing a logical XOR operation between image pixels and the key sequence, Ki. Logistic maps are popular in image encryption algorithms due to their minimal complexity and fast computation speed. However, a potential disadvantage is their limited range of chaotic behavior.

2.2.2. Baker Map

The Baker map is a chaotic transformation that maps a unit square into itself. It involves a sequence of stretching and folding operations that result in a new transformed unit square. The formula for the Baker map is as follows:

$$B(x_{n+1}, y_{n+1}) = \begin{cases} (2x_n, y_n/2) & 0 \leq x < \frac{1}{2} \\ (2 - 2x_n, 1 - y_n/2) & \frac{1}{2} \leq x < 1 \end{cases}, \tag{2}$$

where (x_n, y_n) and (x_{n+1}, y_{n+1}) are the coordinates of a point in the unit square before and after the transformation, respectively.

Elashry et al. [11] proposed a design of a 2D chaotic Baker map that operates on three different modes: cipher block chaining (CBC) mode, cipher feedback (CFB) mode, and output feedback (OFB) mode. In the proposed algorithm, the Baker map performs the permutation procedure on the plaintext image by moving the positions of the pixels. The encryption key uses a random IV, which, depending on the operation mode, is XORed with the plaintext data blocks, where the IV has the same size as the plaintext block. The proposed implementation combines the speed of chaotic maps with the randomness of the mode of operation, resulting in a fast and secure cryptosystem.

In [12], Mondal et al. presented an image encryption algorithm that uses a 2D Baker map, where the plaintext image is initially permuted using a pseudorandom sequence generated by the Baker map, followed by a diffusion process based on an XOR operation between the scrambled plaintext image and the generated sequence. The proposed encryption algorithm employs dual secret keys generated by the Baker map; one is used for permutation and the other for diffusion.

The Baker map is known to be vulnerable to chosen-plaintext attacks, where an attacker with a pair of plaintexts and corresponding ciphertexts can reverse the permutation and diffusion processes to deduce information about the secret key. This vulnerability could be mitigated by combining other layers of security, such as using an IV or multiple chaotic maps.

2.2.3. Arnold Maps

The Arnold map, which was introduced by Vladimir Arnold in 1968 [21], is a nonlinear chaotic transformation that maps a unit square into itself, exhibiting chaotic behavior; this is desirable in image encryption for producing pseudorandom sequences. The Arnold map can be represented by the following formula:

$$\Gamma(x_{n+1}, y_{n+1}) = (2x_n + y_n, x_n + y_n) \mod N, \tag{3}$$

where (x_n, y_n) and (x_{n+1}, y_{n+1}) are the coordinates of a point in the unit square before and after the transformation, respectively. N is the size of the unit square.

Rachmawanto et al. [13] proposed an encryption algorithm based on the Arnold map. The proposed algorithm divides an image into small blocks with the same width and height, and each of the small blocks is permuted using the Arnold map. It is worth noting that the proposed algorithm does not diffuse the pixel's values of the image, resulting in the cipher image having the same histogram information as the original plaintext image.

In [14], Shalaby et al. proposed a medical image encryption algorithm based on an enhanced Arnold map and AES. Their proposed algorithm consists of three steps. It starts by sharpening the edges of the plaintext image to preserve the details. Then the sharpened plaintext image pixels are chaotically shuffled for i number of iterations based on a modified Arnold map. Finally, the permuted image is encrypted using AES-128 encryption in CBC mode.

It should be noted that the Arnold map is designed to work only on square images where the height and width are equal, and it is not suitable for use on rectangular images without first resizing the image to make it square before encryption. Furthermore, the Arnold map is known to exhibit periodic behavior, meaning that after a certain number of permutation iterations, the image pixels will return to their original position.

2.2.4. Tent Maps

The tent map is a chaotic system that maps real numbers in the range (0,1) to another real number in the same range. The tent map can be represented by the following formula:

$$x_{n+1} = f_\mu(x_n) = \begin{cases} \mu x_n & \text{if } x_n < \frac{1}{2} \\ \mu(1 - x_n) & \text{if } \frac{1}{2} \leq x_n \end{cases}, \tag{4}$$

where x_0 is the starting value and x_{n+1} is the mapped value. μ is the control parameter, which has a range of (0, 2). However, chaotic behavior is observed only when the control parameter μ is within the range of (1, 2). The bifurcation diagram for the tent map is depicted in Figure 3.

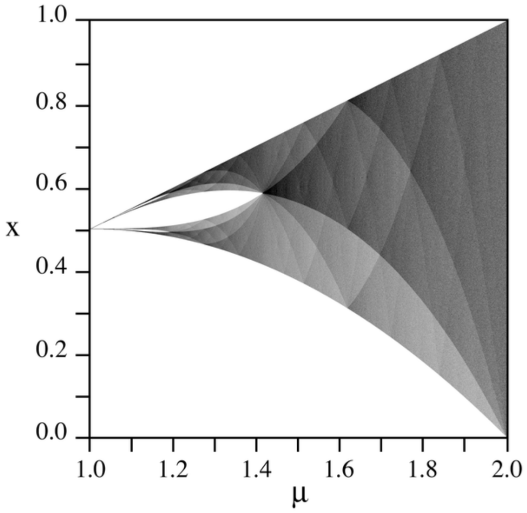

Figure 3. Tent map bifurcation diagram.

Chunhu et al. presented an algorithm in [15]; it utilizes a keystream generated by a tent map. The keystream is subsequently XORed with the plaintext image to produce the cipher image. Notably, the proposed algorithm does not perform any shuffling or permutation on the image's pixels.

In [16], Vishwas et al. proposed an image encryption algorithm based on the tent map. The proposed algorithm has two cases. In the first case, a chaotic sequence is generated based on the tent map; this sequence is then XORed with the plaintext image to produce the cipher image. The second case is similar to the first case, with the addition of a permutation step. The resulting cipher image is then permuted using the sequence generated from the tent map in one of four variants (only rows are permuted, only columns, first rows then columns, or first columns then rows).

Tent maps are easy to implement, feature simple control of initial parameters, and require minimal computational power. However, they suffer from a small range of chaotic behavior, similar to logistic maps.

2.2.5. Henon Maps

The Henon map is a two-dimensional map that maps a point (x_n, y_n) to a new point (x_{n+1}, y_{n+1}). The Henon map can be defined as follows:

$$\begin{aligned} x_{n+1} &= 1 - ax_n^2 \\ y_{n+1} &= bx_n, \end{aligned} \tag{5}$$

where (x_n, y_n) and (x_{n+1}, y_{n+1}) are the coordinates of a point before and after the mapping, respectively; a and b are the control parameters for the Henon map; however, the map starts to behave chaotically when $a = 1.4$ and $b = 0.3$ [22]. The bifurcation diagram for the Henon map is depicted in Figure 4.

Pradeep et al. [23] proposed an image encryption algorithm using the Henon map. The proposed algorithm consists of three stages. In the first stage, the Henon map generates two sequences of random values. In the second stage, the first randomly generated sequence is XORed with the plaintext image to diffuse the pixel values. Finally, the diffused image is permuted using the second sequence of values generated by the Henon map to produce the cipher image.

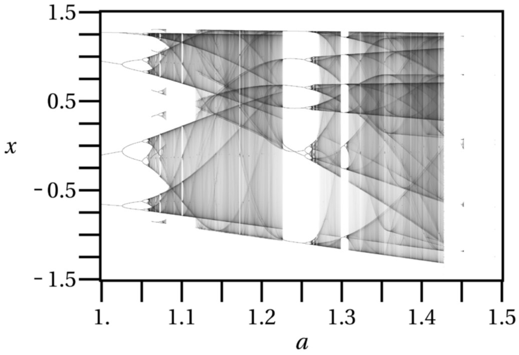

Figure 4. The Henon map bifurcation diagram.

In [24], Hussein et al. introduced a new permutation–substitution algorithm based on the Henon map. The algorithm utilizes the Henon map to generate two sets of keystreams, namely X and Y, each having a length of (height × width + 100), where the first 100 generated values are removed from the keystream. The plaintext image is then reshaped into a one-dimensional array, and the elements are permuted using the random numbers from keystream X for two rounds. The Y keystream is converted into a three-dimensional array to make it compatible with the three color channels (RGB) of the permuted array. Then, each color channel of the permuted array elements is diffused using an XOR operation with the corresponding three-dimensional Y keystream.

2.2.6. Hyperchaotic Systems

Hyperchaotic maps are high-dimensional nonlinear dynamical systems, defined as any chaotic system with at least two positive Lyapunov exponents [25]. The Lyapunov exponent of a dynamical system is a parameter that measures the rate of separation of two infinitesimally close trajectories [26]. Hyperchaotic maps are considered to be better than simple chaotic maps due to their higher degree of non-periodicity and unpredictability [27].

Gao proposed an image encryption algorithm based on a 2D hyperchaotic map in [17]. First, two hyperchaotic sequences, x and y, are generated by the 2D hyperchaotic map. Next, the plaintext image rows and columns are permuted using chaotic sequence x. Finally, the pixel values of the permuted image are obscured through forward and backward diffusion using chaotic sequence y to produce the cipher image.

In [18], Li et al. proposed a hyperchaos-based image encryption algorithm using pixel-level permutation, bit-level permutation, and diffusion. First, two chaotic sequences are generated by a 5D multi-wing hyperchaotic system, where the control parameters are related to the original image. Then, pixel-level permutation and bit-level permutation are performed on the plaintext image using the first chaotic sequence. Finally, the permuted image is diffused using the second chaotic sequence to produce the cipher image.

2.2.7. Multiple Chaotic Maps

The security and complexity of a chaos-based image encryption-based algorithm can be improved by using multiple chaotic maps instead of just one. Additionally, the limitations and vulnerabilities of one chaotic map can be mitigated by combining it with another chaotic map.

Ramasamy et al. [28] used an enhanced logistic-tent map for key generation in an image encryption algorithm based on block scrambling and a modified zigzag transformation. The algorithm starts by dividing the plaintext image into four quadrants, and each quadrant is further divided into four sub-quadrants. Next, each sub-quadrant is rotated anti-clockwise by 90°. Then, each color channel of the scrambled image is further permuted using a modified zigzag transformation, where the upper left pixel and its next neighboring horizontal pixels are exchanged with the base right pixel. Finally, an XOR operation is performed between the permuted image and the keystream generated by the enhanced logistic tent map.

Jain and Aji [29] proposed an image encryption algorithm based on a 2-dimensional logistic-sine coupling map (2D-LSCM) and Arnold's cat map, where Arnold's cat map is used to shift the pixels of the plaintext image. Next, a chaotic matrix (matrix X) that has the same size as the plaintext image is generated by the 2D-LSCM chaotic map, and is used to perform another permutation on the pixel positions. The pixel values are then diffused using chaotic matrix X. This process of permutation and diffusion is repeated for two rounds to produce the cipher image.

2.3. DNA Encoding

DNA encoding can be used in image encryption by converting the image's pixel binary values into a DNA sequence. Since the DNA sequence is composed of four nucleotides, adenine (A), guanine (G), cytosine (C), and thymine (T), researchers can use DNA coding rules to convert the binary information into synthetic DNA sequences using specialized laboratory equipment to handle DNA synthesis and sequencing. For example, a pixel with an RGB value (01010001, 00101000, 10001011) can be encoded as the DNA sequence, GGAG, ACCA, CACT, using rule 1 from the DNA coding rules in Table 1. Binary operation (i.e., addition, subtraction, XOR, etc.) can also work on DNA sequences using DNA operation rule tables. For example, the addition of nucleotides A + A results in A using rule 1. Since A in rule 1 is 00, the addition of 00 + 00 is 00, which is A in rule 1. In the case of adding G + G using rule 1, the result is C, i.e., 01 + 01 is 10, which is equivalent to C. Table 2 illustrates the DNA addition operation for the DNA rule 1 in Table 1.

Table 1. Sample of different DNA coding rules.

Rule	1	2	3	4	5	6	7	8
A	00	00	01	01	10	10	11	11
T	11	11	10	10	01	01	00	00
G	01	10	00	11	00	11	01	10
C	10	01	11	00	11	00	10	01

Table 2. DNA addition for DNA rule 1.

+	A	G	C	T
A	A	G	C	T
G	G	C	T	A
C	C	T	A	G
T	T	A	G	C

The image encryption algorithm presented by Wu et al. [30] utilizes DNA coding in combination with a hyperchaotic map. In the proposed algorithm, a keystream generated using Chen's hyperchaotic map is used to scramble the plaintext image. Afterward, the resulting scrambled image and the keystream are both DNA-encoded. The algorithm then diffuses the DNA-encoded scrambled image with the DNA-encoded keystream. Lastly, DNA decoding is performed on the resulting diffused image to produce the cipher image.

Li and Su [31] proposed an algorithm based on a logistic map, the Chen hyperchaotic map, and DNA encoding. The algorithm starts by dividing an image into three matrices based on the RGB color channels. Next, each of the three matrices is divided into blocks. The Chen hyperchaotic map is used to randomly select the DNA coding, decoding, and operation rules of each block. Each block of three color matrices is first DNA-encoded and then goes through an addition, subtraction, XOR, or XNOR operation selected based on the Chen hyperchaotic map. The algorithm then performs DNA decoding on the blocks of the three color matrices, followed by a row and column permutation based on the logistic map. Finally, the three color matrices are combined into one image to obtain the cipher image.

In [32], Li and Li introduced a novel image encryption algorithm based on DNA dynamic encoding, the logistic-tent map, and the Lorenz hyperchaotic map. The proposed algorithm begins by dividing the plaintext image into rows, which are then DNA-encoded using rules selected by the logistic-tent map random sequence. The resulting DNA-encoded image is then scrambled using the sequence generated by the Lorenz hyperchaotic map at the DNA level. Next, the resulting scrambled DNA-encoded image is diffused using DNA operation rules selected based on the logistic-tent map. The resulting diffused image is then DNA-decoded using the Lorenz hyperchaotic map. Finally, the resulting DNA-decoded image is diffused again using DNA operation rules selected based on the logistic-tent map to produce the cipher image.

The algorithm developed by Feng et al. [33] is based on DNA encoding, 2D-LSCM, the SHA-256 hash of the plaintext image, and the discrete logarithm. The proposed algorithm consists of three encryption steps: permutation and update, DNA sequence operation, and diffusion. All three steps are plaintext image-related. The algorithm begins by generating a SHA-256 hash of the plaintext image and then stretches the plaintext image into a 1D sequence. The algorithm also generates three integer chaotic sequences (ICSs) from the 2D-LSCM based on a secret key to be used in each step. For the permutation and update step, the algorithm permutes and updates the pixel values using the first generated ICS and the hash value using an equation based on modular and discrete logarithmic operations to calculate the permutation coordinates. Next, in the DNA sequence operation step, the algorithm performs DNA encoding, DNA XOR, and then DNA decoding on the resulting permuted image, utilizing DNA rules selected based on the second generated ICS and the hash value. For the diffusion step, the algorithm changes the pixel values in forward or backward diffusion based on the secret key, hash value, the three ICSs, and the discrete logarithm operation. The three steps of the algorithm are performed twice, and the final 1D cipher sequence is converted into a 2D image to produce the cipher image.

Image encryption algorithms based on DNA encoding can have high security and require less storage requirements. Yet the decryption process requires additional DNA rule tables to be transmitted along with the encrypted image [34]. Furthermore, DNA encoding-based encryption is still experimental and only explored in academic research; it requires specialized lab equipment to convert binary values into synthetic DNA sequences.

2.4. Neural Networks

Neural networks can be utilized in image encryption algorithms to enhance the security and efficiency of the algorithm. Neural networks can be used to permute [35] and/or compress the plaintext images [36]. Furthermore, neural networks with chaotic behaviors [37] can be used to generate chaotic sequences, which can be used as keys, input parameters, or to diffuse the image.

Man et al. [35] proposed a double image encryption algorithm based on the convolution neural network (CNN) and chaos. The proposed algorithm uses a logistic map to generate the initial parameters for a 5D conservative chaotic system and CNN. two plaintext images are first permuted by a plaintext-related chaotic pointer generated by the CNN. Next, each permuted image is split on the bit level into high and low 4-bit images. Then two high 4-bit images are fused into one image, and low 4-bit images are fused into one image. Finally, a dual channel encryption (optical encryption, and digital encryption) is performed on the two permuted and fused images to produce two cipher images.

An image encryption algorithm based on the Hopfield chaotic neural network was proposed by Wang and Li [38]. The proposed algorithm starts by generating the initial parameters for the Arnold map and Hopfield neural network using a staged composite chaotic map based on a logistic map and a tent map. Next, the plaintext image pixels are permuted by a random sequence generated by the Arnold map. Then, the Hopfield neural network is utilized to generate a self-diffusion chaotic keystream matrix. Each color channel of the permuted image is then XORed with a keystream matrix to produce the cipher image.

Han et al. [37] proposed an image encryption algorithm based on the Hermite chaotic neural network. First, a chaotic training sample for the Hermite chaotic neural network is generated by a logistic map. Next, two sets of chaotic sequences are generated by the trained Hermite chaotic neural network. Finally, the two sets are summed and multiplied by 256 and then multiplied by the plaintext image to produce the cipher image.

In [36], Yang et al. proposed an image encryption algorithm based on a backpropagation neural network and a hyperchaotic system. First, the plaintext image is split into three matrices (R,G,B), and each matrix is decomposed into sub-image blocks. Next, the sub-image blocks are provided as input to the backpropagation neural network, which produces three compressed pixel matrices. The compressed pixel matrices are then permuted using a zigzag confusion algorithm. After that, five chaotic sequences are generated by a fractional-order memristive hyperchaotic system, and two of the sequences are selected randomly to be XORed with the permuted matrices. Finally, the three diffused matrices are combined to produce the cipher image.

Even though neural network-based image encryption algorithms offer secure encryption, they do have some disadvantages. The time and computational complexity are high for neural network-based image encryption as neural networks require significant processing power and training time.

2.5. Frequency Domain

Using an image frequency domain in an image encryption algorithm can improve an algorithm's security. In frequency domain image encryption, instead of performing the encryption on the plaintext image directly in the spatial domain, the image is first transformed into the frequency domain, using Fourier transform (FT), wavelet transform (WT), discrete cosine transform (DCT), etc., and the confusion and diffusion are performed in the frequency domain. The cipher image is then produced by inverse-transforming the encrypted frequency domain into the spatial domain.

Ding et al. [39] proposed an image encryption algorithm based on a fractional-order Henon map, a 2D discrete wavelet transform (DWT), and a 4D hyperchaotic system. In the proposed algorithm, the plaintext image is transformed using DWT and then permuted using a chaotic sequence generated by the fractional-order Henon map. The 4D hyperchaotic system generates a chaotic sequence that is XORed with the permuted image to produce the cipher image.

In [40], Chen et al. introduced a hybrid domain image encryption algorithm for gray-scale images based on a 2-dimensional improved Henon map (2D-ICHM), integer wavelet transform (IWT), bit plane decomposition, and DNA operations. The algorithm starts by generating a SHA-512 hash from the plaintext image, which is used to derive the control parameters for the improved Henon map (2D-ICHM), which generates two chaotic sequences, namely X,Y. Then, the plaintext image is decomposed into eight binary bit planes, and the high bit planes are XORed with chaotic sequence X. Next, the IWT is applied to the XORed bit plane and then scrambled using chaotic sequence X. After that, inverse IWT is performed followed by DNA encoding, the DNA XOR operation, and DNA decoding. Finally, bidirectional diffusion is performed using chaotic sequences X and Y to produce the cipher image.

Guan et al [41] presented a chaotic image encryption algorithm using frequency domain DNA encoding. The proposed algorithm starts by transforming the plaintext image using a fast Fourier transform (FFT) and obtaining the amplitude and phase, which are then reconstructed as 1D sequences. Next, four chaotic sequences are generated by a 4D hyperchaotic map and the DNA encoding, decoding, and operation rules are generated using these chaotic sequences. After that, the amplitude and the phase sequences are DNA-encoded, DNA operations are performed, and then DNA is decoded to diffuse and confuse the amplitude and phase. Finally, the cipher image is produced by performing an inverse fast Fourier transform (IFFT).

In [42], Faragallah et al. introduced an efficient and secure color image cryptosystem based on chaotic logistics in fractional Fourier transform (FrFT). The proposed algorithm starts by splitting the plaintext image into red, green, and blue channels. Next, fractional Fourier transform (FrFT) is applied on the red, green, and blue color plaintext image channels. Afterward, a 2D logistic map is employed to scramble each of the color image pixel positions in the FrFT. Subsequently, the inverse fractional Fourier transform (iFrFT) is applied to the resulting encrypted channels. Finally, the red, green, and blue channels are combined to produce the cipher image.

2.6. Compressive Sensing

In recent years, there has been a growing interest in compressive sensing-based image encryption algorithms. Compressive sensing-based algorithms have the advantage of simultaneously sampling, compressing, and encrypting images, resulting in a faster encryption time and a reduced image size. In a compressive sensing-based image encryption algorithm, a plaintext image is first transformed into a sparse domain using discrete transform. Then, the transformed image is compressed using a random measurement matrix. The measurement matrix is a matrix used to capture a reduced set of the essential measurements/information of the image's structure, which allows accurate compression and reconstruction from a reduced set of measurements. Finally, the compressed image is encrypted using permutation and diffusion techniques.

Dou and Li [43] proposed a novel image encryption algorithm based on compressive sensing, the M sequence, and an improved 1D chaotic map that incorporates elements from logistic, sine, and Chebyshev maps. The proposed algorithm generates a SHA-512 hash from the plaintext image, which is used to derive the control parameters for the improved 1D chaotic map and the M sequence linear-feedback shift registers (LFSRs). The plaintext image is then transformed to a DWT and is scrambled using the M sequence. Next, the improved 1D chaotic system is used to generate a measurement matrix that is used to compress and diffuse the scrambled image to produce the cipher image.

In [44], Zhang et al. proposed an image compression and encryption algorithm based on compressive sensing, Fourier transform, and chaotic maps. The proposed algorithm utilizes a tent-sine chaotic map to generate a measurement matrix, using a SHA-256 hash of the plaintext image as initial parameters. Next, the Arnold transform is used to permute the compressed image, and then a 2D Fourier transform generated by Chen's hyperchaotic map is used as a mask to produce the cipher image.

Wang et al. [45] proposed an image encryption based on compressive sensing and DNA encoding. The algorithm utilizes a four-wing hyperchaotic system and Kronecker product (KP). First, a chaotic sequence generated by the chaotic system is used to dynamically control the DNA coding. Afterward, the measurement matrix is obtained by a four-wing chaotic system. Next, the plaintext image is sparsified and permuted by Fisher–Yates random scrambling. Finally, DNA coding and DNA XOR operations are performed to produce the cipher image.

2.7. Meaningful Encryption

Meaningful image encryption is a technique that utilizes image encryption and steganography to produce cipher images that have a visually meaningful image instead of a noise-like image. Steganography is the practice of concealing data within another non-secret carrier file or message to avoid detection. There are two steps in a meaningful encryption algorithm. First, the image is encrypted using an image encryption algorithm. Next, the resulting encrypted image is embedded in a non-secret carrier image using a steganography algorithm. The carrier image, which looks like a normal image, can then be transmitted to the receiver.

Ping et al. [46] proposed an image encryption algorithm, utilizing compressive sensing, chaotic maps, and steganography to produce meaningful cipher images. The proposed algorithm consists of two stages: a compressive sensing encryption step and a steganography step. For the compressive sensing encryption step, a 2D logistic-adjusted sine map (2D-LASM) and a 3D cat map are generated from a secret key. Then, the plaintext image is transformed into a sparse domain using DWT and permuted using the chaotic sequence generated by the 2D-LASM. The measurement matrix for compressive sensing is constructed using the chaotic sequences generated by the 3D cat map and the 2D-LASM. Finally, the measurement matrix is used to compress and encrypt the permuted image. In the steganography step, the resulting cipher image is embedded into a carrier image, and the order of the embedding position is encrypted using the chaotic sequence generated by the 2D-LASM.

In [47], Wang et al. proposed a meaningful image encryption based on a chaotic map and random scrambling diffusion. The plaintext is first sparsified by wavelet packet transform (WPT). Next, a one-dimensional chaotic map is used to construct the measurement and permutation matrices. Then, a bidirectional random scrambling algorithm based on chaotic magic transformation (CMT) is used to permute the sparsified plaintext image. The permuted resulting image is compressed using the chaotic measurement matrix and then diffused using chaotic pixel diffusion. Finally, the encrypted and compressed image is embedded in a carrier image using the WPT embedding algorithm.

Jiang et al. [48] proposed a novel meaningful image encryption algorithm based on parallel compressive sensing and slant transform. First, the plaintext image is sparsified by DWT and the sparse coefficient matrix of the plaintext image is used as the control parameters for the 4D memristive hyperchaotic map, which generates the scrambling matrix. Then, the sparsified plaintext image is permuted and compressed using Arnold scrambling and parallel compressive sensing to produce the cipher image. Finally, the cipher image is randomly embedded in a carrier image by slant transform-based embedding in a block-wise manner to produce a visually meaningful cipher image.

In [49], Yang et al. proposed a visually meaningful image encryption algorithm based on a new universal embedding model (UEM). The new UEM is used to embed secret information into the host and to adapt to different types of wavelet transforms. First, the plaintext image is encrypted using an image encryption algorithm (the authors did not specify an exact algorithm) to produce a permuted and diffused image. Next, integer wavelet transform is performed on the carrier image to obtain the wavelet transform domain. Then, a four-dimensional discrete chaotic system is used to embed the permuted and diffused image in the integer wavelet sub-bands of the carrier image. Finally, inverse integer wavelet transform (iIWT) is performed on the resulting wavelet transform domain to produce a visually meaningful cipher image.

Meaningful image encryption algorithms have the advantage of keeping the embedded data secure even if the embedded image data are extracted via steganalysis tools. Since the original secret image is already encrypted and resembles a noise-like image, it adds another layer of security to steganography by preventing attackers from directly obtaining the secret image through steganalysis alone. However, these algorithms have the drawback of high computational requirements due to the implementation of two layers of security: image encryption and steganography.

3. Evaluation Metrics

Researchers depend on a comprehensive set of metrics (e.g., correlation coefficient, histogram analysis tests (chi-square, maximum deviation, irregular deviation, and deviation from uniform histogram), information entropy (global and local), gray-level co-occurrence matrix (GLCM) analysis (contrast, energy, and homogeneity), encryption quality analysis (mean square error (MSE), mean absolute error (MAE), and peak signal-to-noise ratio (PSNR)), resistance to differential attacks, number of pixels change rate (NPCR), unified average changing intensity (UACI), resistance to noise and data loss attacks, and the algorithm’s key sensitivity) to evaluate and analyze image encryption algorithms. Each of these tests is discussed in detail below. Figure 5 illustrates the most commonly used evaluation metrics for image encryption in literature.

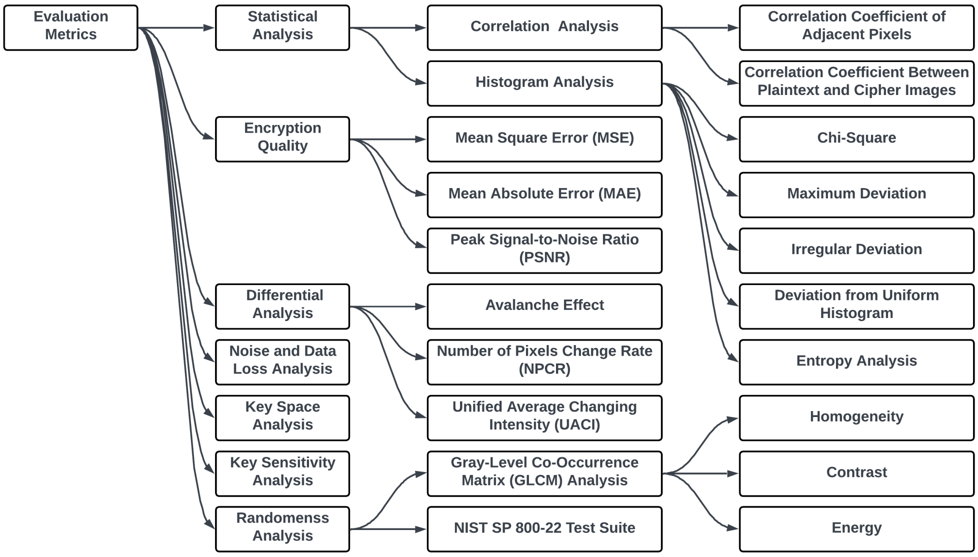
Evaluation Metrics
Statistical Analysis
Encryption Quality
Differential Analysis
Noise and Data Loss Analysis
Key Space Analysis
Key Sensitivity Analysis
Randomenss Analysis
Correlation Analysis
Histogram Analysis
Mean Square Error (MSE)
Mean Absolute Error (MAE)
Peak Signal-to-Noise Ratio (PSNR)
Avalanche Effect
Number of Pixels Change Rate (NPCR)
Unified Average Changing Intensity (UACI)
Gray-Level Co-Occurrence Matrix (GLCM) Analysis
NIST SP 800-22 Test Suite
Correlation Coefficient of Adjacent Pixels
Correlation Coefficient Between Plaintext and Cipher Images
Chi-Square
Maximum Deviation
Irregular Deviation
Deviation from Uniform Histogram
Entropy Analysis
Homogeneity
Contrast
Energy

Figure 5. Most commonly used evaluation metrics.

3.1. Correlation Coefficient Analysis

Digital images typically exhibit strong correlations among adjacent pixels. To ensure a secure and robust image encryption algorithm, and to avoid vulnerability to statistical attacks, it is essential to eliminate this correlation. The correlation coefficient is a statistical test that is commonly used in image encryption to quantify the strength of correlation between two pixels. It has a range of (1, −1), where a value of 1 or −1 indicates the maximum positive or negative correlation, respectively, while a value of 0 indicates no correlation between the compared pixels. Image encryption algorithms usually utilize the correlation coefficient metric to evaluate horizontal, vertical, and diagonal correlation coefficients between adjacent pixels within a cipher image (Section 3.1.1) as well as between a plaintext image and its cipher (Section 3.1.2).

3.1.1. Correlation Coefficient of Adjacent Pixels

The vertical correlation coefficient of adjacent pixels can be calculated as follows:

$$CC_v = \frac{\sum_{i=1}^{H-1}\sum_{j=1}^{W}(C_{(i,j)} - \overline{C})(C_{(i+1,j)} - \overline{C})}{\sqrt{\sum_{i=1}^{H-1}\sum_{j=1}^{W}(C_{(i,j)} - \overline{C})^2 \sum_{i=1}^{H-1}\sum_{j=1}^{W}(C_{(i+1,j)} - \overline{C})^2}}, \tag{6}$$

where H and W are an image's height and width, respectively. $C_{(i,j)}$ and $C_{(i+1,j)}$ refer to the pixel values of two adjacent pixels in the cipher image at positions (i, j) and $(i+1, j)$, respectively. The mean pixel value of the cipher image is denoted by $\overline{C}$. Furthermore, to calculate the horizontal correlation coefficient, Equation (6) is adjusted such that the value of the pixel at $C_{(i,j+1)}$ is used instead of $C_{(i+1,j)}$, and the value of the pixel at $C_{(i+1,j+1)}$ is used to calculate the diagonal correlation coefficient.

3.1.2. Correlation Coefficient between Plaintext and Cipher Images

The correlation coefficient between a plaintext image and its cipher image can be calculated as follows:

$$CC_{P,C} = \frac{\sum_{i=1}^{H}\sum_{j=1}^{W}(P_{(i,j)} - \overline{P})(C_{(i,j)} - \overline{C})}{\sqrt{\sum_{i=1}^{H}\sum_{j=1}^{W}(P_{(i,j)} - \overline{P})^2 \sum_{i=1}^{H}\sum_{j=1}^{W}(C_{(i,j)} - \overline{C})^2}}, \tag{7}$$

where $P_{(i,j)}$ and $C_{(i,j)}$ are the values of the pixel at index i, j of the plaintext image and the cipher image, respectively, and $\overline{P}$ and $\overline{C}$ represent the mean of the pixel values of the plaintext image and the cipher image, respectively.

3.2. Histogram Analysis

Histogram analysis is used to ensure the security and robustness of image encryption algorithms by evaluating the uniformity of a cipher image's histogram. The histogram of an image is an important indicator that reflects the distribution of its pixel values. A cipher image with a close-to-uniformly distributed histogram prevents attackers' attempts to extract any useful information. Researchers in image encryption use a set of metrics to quantify histogram analysis, such as chi-square (χ^2), maximum deviation, irregular deviation, and deviation from uniform histogram. Each of the aforementioned metrics is further discussed in the following subsections.

3.2.1. Chi-Square (χ^2)

Chi-square χ^2 is a statistical test that is used in image encryption to test the randomness and uniformity of cipher images produced by an algorithm, by comparing the cipher image's histogram values with the expected distribution of a uniform histogram [50]. Chi-square tests each pixel of the cipher image to determine whether the distribution of pixel values is uniform or not. Chi-square (χ^2) is mathematically represented as follows:

$$\begin{aligned} \chi^2 &= \sum_{i=0}^{255} \frac{(f_i - \mathcal{E})^2}{\mathcal{E}} \\ \mathcal{E} &= \frac{H \times W}{256}, \end{aligned} \tag{8}$$

where f represents the histogram of the cipher image, f_i is the cipher image's histogram value at index i, $\mathcal{E}$ represents the expected distribution value of a uniform histogram, and H and W are the height and width of the cipher image, respectively. A uniformly distributed histogram has a χ^2 value of 0. When the χ^2 value is low, it indicates that the histogram distribution of the cipher image is close to a uniform distribution. Furthermore, when the χ^2 value is higher, then the histogram is non-uniform. Since the histogram of an image is calculated for pixel values ranging from 0 to 255, we have 255 degrees of freedom, and if we assume a significance level of 0.05, then the critical chi-square value is 293.24. If the cipher image, χ^2, is lower or equal to the critical chi-square value, we accept the result as close to being uniform. From Equation (8), we can calculate the lower and upper limits of χ^2. To find the lower limit, we assume that the image has a uniform histogram, and then each element of the histogram, f, will have a value of $\frac{H \times W}{256}$. By substituting f in Equation (8), the lower limit can be calculated as follows:

$$\begin{aligned} \chi^2 &= 256 \times \frac{(\frac{H \times W}{256} - \mathcal{E})^2}{\mathcal{E}} \\ &= 256 \times \frac{(\frac{H \times W}{256} - \frac{H \times W}{256})^2}{\frac{H \times W}{256}} \\ &= 256 \times \frac{(0)^2}{\frac{H \times W}{256}} \\ \chi^2 &= 0, \end{aligned} \tag{9}$$

From Equation (9), the lower limit for χ^2 is 0. For the upper limit, we assume that the image has one color, then the histogram, f, will have one element with value $H \times W$ and the rest will be zeros. By substituting f in Equation (8), the upper limit can be calculated as follows:

$$\begin{aligned} \chi^2 &= (255 \times \frac{(0 - \mathcal{E})^2}{\mathcal{E}}) + \frac{(H \times W - \mathcal{E})^2}{\mathcal{E}} \\ &= 255 \times \mathcal{E} + \frac{(H \times W - \mathcal{E})^2}{\mathcal{E}}, \end{aligned} \tag{10}$$

Table 3 utilizes Equation (10) to compute the upper limit of χ^2 for different-sized images.

Table 3. Upper limit of Chi-square χ^2 for different-sized images.

Size	128×128	256×256	512×512	1024×1024
Upper limit	4,177,920	16,711,680	66,846,720	267,386,880

3.2.2. Maximum Deviation

The maximum deviation (D_{max}) is a metric used to evaluate the uniformity of the cipher image's histogram by measuring the deviation between the histograms of a plaintext image and its cipher. The maximum deviation metric helps to determine the quality of the image encryption algorithm by testing whether the produced cipher image differs significantly from the original image. A higher value of the maximum deviation indicates a better encryption quality since the algorithm produces a cipher image that is more deviated from the plaintext image. The mathematical expression of maximum deviation can be given as follows:

$$\begin{aligned} D_{max} &= \frac{d_0 + d_{255}}{2} + \sum_{i=1}^{254} d_i \\ d &= histogram(|P - C|), \end{aligned} \tag{11}$$

where d is the histogram of the absolute values of the difference between the plaintext image and its cipher, d_i represents the histogram d value at index i, and d_0 and d_{255} are the histogram values at index 0 and 255, respectively.

From Equation (11), we can calculate the upper limit of D_{max}. If we assume that both images are completely different from each other, d will have one element with value $H \times W$, and the rest will be zeros. By substituting d in Equation (11), the upper limit can be calculated as follows:

$$\begin{aligned} D_{max} &= \frac{(0 + H \times W)}{2} + (254 \times 0) \\ &= (\frac{H \times W}{2}), \end{aligned} \tag{12}$$

The lower limit of D_{max} is 0 for any size image. Table 4 utilizes Equation (12) to compute the upper limit for different-sized images.

Table 4. Upper limit of maximum deviation for different-sized images.

Size	128×128	256×256	512×512	1024×1024
Upper limit	8192	32,768	262,144	524,288

3.2.3. Irregular Deviation

The irregular deviation ($D_{irregular}$) is another metric used in image encryption to evaluate the quality of an algorithm by measuring the difference between the histograms of a plaintext image and its cipher [51]. Unlike the maximum deviation metric, irregular deviation also considers the difference between neighboring pixels. A cipher image with a lower value of $D_{irregular}$ indicates better encryption algorithm quality. The irregular deviation can be mathematically expressed as follows:

$$D_{irregular} = \sum_{i=0}^{255} \left[\left| d_i - D_{avg} \right| \right], \tag{13}$$

where

$$D_{avg} = \frac{1}{256} \sum_{i=0}^{255} d_i, \tag{14}$$

where d_i represents the histogram value at index i (from Equation (11)), and D_{avg} denotes the average value of the pixels that are deviated at every deviation value.

From Equation (13), the upper limits of $D_{irregular}$ can be calculated. If we assume that both images are completely different from each other, d will have one element with value $H \times W$, and the rest will be zeros. By substitution d in Equation (13), the upper limit can be calculated as follows:

$$\begin{aligned} D_{irregular} &= \left| H \times W - (\frac{H \times W}{256}) \right| + (255 \times \left| 0 - \frac{H \times W}{256} \right|) \\ &= \left| H \times W - (\frac{H \times W}{256}) \right| + (\frac{255 \times H \times W}{256}), \end{aligned} \tag{15}$$

The lower limit of $D_{irregular}$ is 0 for any size image. Table 5 utilizes Equation (15) to compute the upper limit for different-sized images.

Table 5. Upper limit of irregular deviation for different-sized images.

Size	**128×128**	**256×256**	**512×512**	**1024×1024**
Upper limit	32,640	130,560	522,240	2,088,960

3.2.4. Deviation from Uniform Histogram

The deviation from the uniform histogram ($D_{uniform}$) is a metric used to measure the deviation of a cipher image's histogram from a uniform distribution. A cipher image with a histogram distribution close to a uniform distribution indicates good quality of the employed encryption algorithm [51]. When a cipher image produced by an image encryption algorithm has a histogram distribution that is close to a uniform distribution, it implies that each pixel value occurs with equal frequency. The deviation from the uniform histogram metric can be mathematically expressed as follows:

$$D_{uniform} = \frac{\sum_{i=0}^{255} |H_{C_i} - H_{U_i}|}{H \times W}, \tag{16}$$

$$H_{U_i} = \begin{cases} \frac{H \times W}{256} & \text{if } 0 \leq i \leq 255 \\ 0 & \text{if } 0 > i > 255 \end{cases}$$

where H_{C_i} refers to the histogram value of the cipher image at intensity i, and H_{U_i} represents the histogram value of the uniform distribution at intensity i. H_{U_i} values are only considered when the pixel value is between 0 and 255, otherwise, H_{U_i} is set to 0. From Equation (16), we can calculate the upper limit of $D_{uniform}$. If we assume that an image has extremely deviated from a uniform histogram, H_C will have one element with value $H \times W$, and the rest will be zeros. By substitution H_C in Equation (16), the upper limit can be calculated as follows:

$$\begin{aligned} D_{uniform} &= \frac{\left| H \times W - (\frac{H \times W}{256}) \right| + (255 \times \left| 0 - \frac{H \times W}{256} \right|)}{H \times W} \\ &= \left| \frac{H \times W}{H \times W} - \frac{\frac{1}{256} \times H \times W}{H \times W} \right| + \frac{\frac{255 \times H \times W}{256}}{H \times W} \\ &= \left| 1 - \frac{1}{256} \right| + \frac{255}{256} \\ D_{uniform} &= 1.9921875 \end{aligned} \tag{17}$$

The lower limit of $D_{uniform}$ is 0 for any size image, and from Equation (17). It is clear that the maximum possible $D_{uniform}$ value for any image, regardless of its size, is 1.9921875.

3.3. Entropy Analysis

Information entropy is a statistical test that estimates the uncertainty and randomness in communication systems. It was first introduced by Claude Shannon in 1948 [52]. In image encryption, entropy is used to measure the randomness of the distribution of pixels in the cipher image. A high entropy value indicates a better obfuscation of the plaintext and the absence of any discernible pattern. To evaluate the unpredictability and randomness of an encryption algorithm, researchers rely on two types of entropy tests, namely, global entropy and local entropy. Shannon entropy, also known as global entropy, calculates the pixel information for the full image, while local entropy measures the mean entropy of randomly selected non-overlapping blocks. Shannon entropy can be calculated as follows:

$$H(m) = -\sum_{i=0}^{2^n-1} p(m_i)\log_2[p(m_i)], \tag{18}$$

where n represents the number of bits used to represent the symbol $p(m_i)$, and $p(m_i)$ represents the probability of symbol m_i. Entropy for an image can be calculated using Equation (18), where $p(m_i)$ represents normalized histogram counts for each intensity value in the image. Given that a pixel's maximum possible intensity value in an 8-bit representation is 256, and that the probability of a pixel value occurrence is $\frac{1}{256}$, the ideal value of entropy for an image can be calculated using Equation (18), as follows:

$$H_{ideal} = -\sum_{i=0}^{255} \frac{1}{256} \times \log_2 \frac{1}{256} = 8$$

The entropy of a cipher image generated by an encryption algorithm should be as close as possible to the optimal entropy value of 8 to withstand brute force attacks.

The local entropy is measured by computing the mean Shannon entropy of randomly selected non-overlapping blocks. Local entropy is considered more accurate, consistent, and efficient than global Shannon entropy [53]. Local entropy can be calculated as follows:

$$H_{K,T_B}(S) = \sum_{i=1}^{K} \frac{H(S_i)}{K}, \tag{19}$$

where S is a set of non-overlapping blocks selected randomly, containing T_B pixels, and K represents the number of random blocks. $H(S_i)$ is the Shannon entropy (from Equation (18)) of the ith block.

3.4. Gray-Level Co-Occurrence Matrix (GLCM) Analysis

The GLCM is a statistical test used in image encryption to analyze the texture of an image by considering the spatial relationship of pixels. It consists of a two-dimensional matrix of joint probabilities between pairs of pixels with specific values. By calculating the frequency of a pixel with gray-level value i occurring horizontally adjacent to a pixel with gray-level value j, the GLCM is useful for feature extraction and texture analysis. From the GLCM, statistical measures (namely homogeneity, contrast, and energy) can be extracted to provide information about the image's texture features. In the following, we further discuss each of these measurements.

3.4.1. Homogeneity

In image encryption, homogeneity analysis is used to assess the uniformity of texture in a cipher image. It measures the closeness of the elements in the GLCM distribution to the GLCM diagonal [54]. Homogeneity values are in the range of (0, 1), where a cipher image with high homogeneity indicates that similar pixels are close to each other, which may reveal information about the plaintext image. On the other hand, lower homogeneity values indicate better encryption quality; a cipher image produced by an image encryption algorithm with low homogeneity has a more randomized pixel distribution. The homogeneity formula is as follows:

$$Homogeneity = \sum_{i,j} \frac{p(i,j)}{1 + |i - j|}, \tag{20}$$

where i and j represent two gray-level values in the GLCM, and $p(i,j)$ denotes the probability of i and j occurring horizontally adjacent to each other in the image, given by the value of the element at position (i,j) in the normalized GLCM.

3.4.2. Contrast

Contrast analysis assesses the differences in pixel intensities between a pixel and its neighboring pixels in a cipher image [54]. A high contrast value between neighboring pixels indicates a strong encryption algorithm that is effective at hiding the plaintext image's features. Furthermore, a high degree of contrast between neighboring pixels indicates greater randomness in the cipher image. The contrast can be calculated as follows:

$$Contrast = \sum_{i,j} |i - j|^2 p(i,j), \tag{21}$$

3.4.3. Energy

Energy analysis is used as a metric to describe the texture of a cipher image. It is calculated by summing the squared elements in the GLCM, and the resulting energy values are in the range of (0, 1) [54]. Low-energy values in the produced cipher images indicate that the algorithm has successfully randomized the pixels of the plaintext image. The energy can be calculated from the following expression:

$$Energy = \sum_{i,j} p(i,j)^2, \tag{22}$$

3.5. Encryption Quality

The quality of an image encryption algorithm is analyzed using various metrics, by comparing the pixel values of plaintext images and their respective ciphers. The encryption algorithm is considered to have good quality when there is a high change in pixel values between plaintext images and their respective ciphers. To quantify the encryption quality, image encryption researchers use a set of metrics, namely, MSE, MAE, and PSNR. Each of these metrics is discussed further in the following.

3.5.1. Mean Square Error

The MSE is a metric used to calculate the average square difference between the pixel values of a plaintext image and its cipher. In image encryption, it is used to evaluate the quality of an image encryption algorithm [55]. A lower MSE value indicates a greater similarity between the plaintext image and its cipher, whereas a higher MSE value means a lower similarity between the plaintext image and its cipher; therefore, a higher MSE value signifies higher-quality encryption. The MSE between a plaintext image and its cipher image can be calculated as follows:

$$MSE = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} [P(i,j) - C(i,j)]^2, \tag{23}$$

where H and W represent the height and width of the images, respectively. $P(i,j)$ and $C(i,j)$ denote the pixel values of the plaintext image and the cipher image at position (i,j), respectively. From Equation (23), the minimum and maximum possible MSE values for an image, regardless of its size, can be calculated. For the minimum value of MSE, if we assume the two images to be identical, then $P(i,j) - C(i,j)$ will be 0. Thus, the MSE minimum value is 0. For the maximum value, we assume that the two images are completely different from each other (one with 255-pixel values, and the other with 0-pixel values); the maximum possible MSE can be calculated as follows:

$$\begin{aligned} MSE &= \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} [255 - 0]^2 \\ &= \frac{1}{H \times W} \times H \times W \times [255]^2 \\ &= \frac{H \times W \times [255]^2}{H \times W} \\ MSE &= 65,025 \end{aligned} \tag{24}$$

3.5.2. Mean Absolute Error

The MAE is another statistical metric used to evaluate the quality of an image encryption algorithm. It measures the average absolute difference between the pixel values of a plaintext image and its cipher image [56]. A lower MAE value indicates a greater similarity between the plaintext image and its cipher, whereas a higher MAE value indicates a lower similarity between the plaintext image and its cipher; therefore, a higher MAE value means higher-quality encryption. The MAE can be calculated as follows:

$$MAE = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} |P(i,j) - C(i,j)|, \tag{25}$$

From Equation (25), the minimum and maximum possible MAE values for an image, regardless of its size, can be calculated. For the minimum value of MAE, if we assume the two images to be identical, then $P(i,j) - C(i,j)$ will be 0. Thus, the MAE minimum value is 0. For the maximum value, we assume that the two images are completely different from each other (one with 255-pixel values, and the other with 0-pixel values), the maximum possible MAE can be calculated as follows:

$$\begin{aligned} MAE &= \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} [255 - 0] \\ &= \frac{1}{H \times W} \times H \times W \times [255] \\ &= \frac{H \times W \times [255]}{H \times W} \\ MAE &= 255 \end{aligned} \tag{26}$$

3.5.3. Peak Signal-to-Noise Ratio

In image encryption, the PSNR metric is used to measure the noise ratio between the plaintext image and its cipher image [55]. A PSNR value of 0 indicates that the tested image is equivalent to random noise, and a higher value means a higher quality image, which is closer to the plaintext image. The lower the PSNR value in the cipher images produced by an encryption algorithm, the better the quality of the encryption. PSNR can be calculated as follows:

$$PSNR = 10 \log_{10} \frac{MAX_p}{MSE}, \tag{27}$$

where MSE is the mean square error value, which is calculated using Equation (23), and MAX_p is the maximum value that a pixel can have (which is typically 255 in 8-bit pixels).

3.6. Resistance Against Differential Attacks

Cryptanalysts use differential attacks to find the relationship between a plaintext image and its cipher by making small alterations to the plaintext image and encrypting it with the same key. An encryption algorithm should have excellent diffusion characteristics, where modifying a single pixel in the plaintext image should generate a completely different cipher image. An encryption algorithm's ability to resist differential attacks can be evaluated by analyzing the algorithm's performance using the avalanche effect, NPCR, and UACI tests. The details of each of these tests are further discussed in the following.

3.6.1. Avalanche Effect

In image encryption, the avalanche effect refers to the phenomenon that occurs where a slight change to the key or the plaintext image results in an image encryption algorithm producing a significantly different cipher image. An encryption algorithm is considered safe against differential attacks when one pixel change in the plaintext image or key results in a change of more than 50% of the pixels in the cipher image [57]. The avalanche effect can be analyzed using the MSE metric by modifying Equation (23) to calculate the mean-squared difference between the two cipher images, which are produced by encrypting a plaintext image and the same plaintext image with only one pixel modified. The avalanche effect MSE (MSE_{av}) can be calculated as follows:

$$MSE_{av} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} [C_1(i,j) - C_2(i,j)]^2, \tag{28}$$

where C_1 and C_2 are the two cipher images produced by encrypting two plaintext images, where the two plaintext images differ in only one pixel, with the remaining pixels being the same. $C_1(i,j)$ and $C_2(i,j)$ represent the pixel values at position (i,j) of C_1 and C_2, respectively. The minimum value of MSE_{av} is 0, and the maximum value is 65,025, the same as the MSE minimum and maximum values, which are calculated in Section 3.5.1.

3.6.2. Number of Pixels Change Rate (NPCR)

The NPCR metric measures the percentage of the differences between two cipher images that are produced by encrypting two plaintext images that differ in only one pixel, with all other pixels being the same in the two plaintext images [58]. A higher NPCR value indicates greater responsiveness from the algorithm to changes in the plaintext image or key and, thus, better resistance to differential attacks. In other words, a good encryption algorithm should be extremely sensitive to minor changes in the key or the plaintext image to resist such attacks. Wu et al. [59] calculated that the ideal NPCR value is $\geq$99.6094%. The NPCR value can be calculated as follows:

$$NPCR = \sum_{i=1}^{H} \sum_{j=1}^{W} \frac{D(i,j)}{H \times W} \times 100\%, \tag{29}$$

where

$$D(i,j) = \begin{cases} 0 & \text{if } C_1(i,j) = C_2(i,j) \\ 1 & \text{if } C_1(i,j) \neq C_2(i,j) \end{cases}$$

where $D(i,j)$ is the sum of the number of different pixels between C_1 and C_2 at position (i,j). $D(i,j)$ has a value of 1 when the pixel values of C_1 and C_2 at position (i,j) are different; otherwise, it has a value of zero.

3.6.3. Unified Average Changing Intensity (UACI)

The UACI is another metric that is used to assess an encryption algorithm's ability to resist differential attacks by measuring the average change intensity between two cipher images that are produced by encrypting two plaintext images that are the same but differ in only one pixel [58]. Similar to the NPCR, a higher UACI value indicates a better resistance to differential attacks. The ideal value for UACI is $\geq 33.4635\%$, as calculated by Wu et al. [59]. The UACI value can be calculated as follows:

$$UACI = \frac{1}{H \times W}\left[\frac{\sum_{i=1}^{H}\sum_{j=1}^{W}|C_1(i,j) - C_2(i,j)|}{255}\right] \times 100\%, \tag{30}$$

3.7. Resistance to Noise and Data Loss

Digital images transmitted over communication networks are often affected by noise and data loss, which can render cipher images undecryptable. Such distortions could also be introduced by attackers aiming to corrupt cipher images. These attacks are known as noise and occlusion attacks. To ensure the reliability of an image encryption algorithm, its ability to withstand such attacks and decrypt the plaintext image from the altered cipher image needs to be tested [60]. To evaluate an encryption algorithm's resilience against noise attacks, researchers introduce salt and pepper and Gaussian noise to the cipher image with variable densities and try to decrypt the affected cipher images. For resistance, against occlusion attacks, researchers evaluate the encryption algorithm's ability to decrypt a cipher image that had segments cropped out at varying degrees of severity. Furthermore, the PSNR metric (Section 3.5.3) could be utilized to quantify the quality of the encryption algorithms at decrypting a corrupted image by comparing the PSNR of the decrypted corrupted image to the plaintext image [61]. A higher PSNR of the decrypted image when compared to the original plaintext image indicates that the decrypted corrupted image is closer to the original plaintext image.

3.8. Computation Complexity

Computation complexity is an important metric in evaluating encryption algorithms as it determines the computational requirements to perform the encryption and decryption operations. To evaluate an encryption algorithm's computation complexity, researchers assess the key generation complexity and the encryption-decryption process complexity. Furthermore, in the evaluation of image encryption algorithms, *runtime* is an important consideration (other than security) in adopting an image encryption algorithm for a particular domain. For example, many edge devices have limited computing power [62]; thus, image encryption algorithms with lower computational complexity are suitable for such edge devices. Similarly, many applications are time-sensitive (e.g., surveillance applications [63]) and, thus, image encryption algorithms that can be executed in real time are more befitting for such applications compared to highly compute-intensive encryption algorithms.

3.8.1. Key Generation Complexity

The complexity of an algorithm's key generation plays a big role in the overall computation complexity of the algorithm. Some algorithms have simple key-generation processes, such as algorithms using chaotic maps for key generation, while others may involve complex mathematical computations, such as the RSA key generation, which involves computationally intensive operations involving finding two large prime numbers, computing Euler's totient function, and performing mathematical operations.

3.8.2. Encryption-Decryption Complexity

Image encryption algorithms should be fast and efficient enough to encrypt decryption images quickly, especially in real-time applications. Nevertheless, the algorithms should have enough complexity to resist all possible attacks. An image encryption algorithm's encrypt decryption complexity involves operations such as substitution–permutation networks, XOR operations, and modular operations.

3.9. Key Space and Key Sensitivity

The key space represents the set of all possible keys that an encryption algorithm can use. To ensure a stronger level of security against brute force attacks, the key space of an image encryption algorithm should be larger than 2^{100} [64].

Key sensitivity analysis assesses the image encryption algorithm's sensitivity to small changes to the key. A robust and secure algorithm should produce two completely different cipher images when encrypted by two keys that are different by only one bit. Furthermore, it should not be able to decrypt a cipher image and retrieve the plaintext image with a key that has one bit altered [65].

To evaluate an encryption algorithm's key sensitivity for encryption, researchers encrypt a plaintext image with two keys (k_1 and k_2) that differ in only one bit and compare the resulting cipher images. For the case of evaluating key sensitivity for decryption, researchers attempt to decrypt a cipher image that is encrypted with key k_1 using key k_2, which differs in only one bit. If the algorithm is sensitive to changes in the key, it should not be able to decrypt the cipher image with the altered key.

3.10. NIST SP 800-22 Test Suite

The NIST SP 800-22 test suite consists of a set of statistical tests developed by the National Institute of Standards and Technology (NIST) [1]. It is used by researchers to evaluate the randomness characteristics and strength properties of a sequence to determine if they are sufficiently random. The test suite consists of 15 statistical tests, each of which generates a p-value in the range of (0, 1), where a p-value of 0 indicates that the sequence is not random at all, and a p-value of 1 indicates that the sequence is completely random. In image encryption, the test suite is used to evaluate the randomness of the produced cipher image. For conducting these NIST tests, the image must be first transformed into a bit stream, after which the test suite can process the data and produce results for each test. If the p-value of a test is greater than a significance level of $\alpha = 0.01$, then the null hypothesis is accepted; that is, the sequence is considered random and passes the test. If the computed p-value is less than 0.01, then the null hypothesis is rejected; that is, the sequence appears to be non-random.

4. Discussion

In this section, the advantages, disadvantages, and applications of each of the aforementioned image encryption techniques are explored. Each image encryption technique provides a level of security and privacy, yet each of the techniques has some pros and cons. Table 6 summarizes the strengths and weaknesses of different image encryption algorithms. In the following, we elaborate the strengths and weaknesses of different image encryption algorithms.

Traditional encryption algorithms are still the most commonly recommended and used algorithms to encrypt images over networks. Their security was assessed and proven through standardized testing and cryptanalysis. Their downside is that they typically require extensive computation time and significant processing power. Furthermore, traditional encryption algorithms fail when attempting to decrypt cipher images that have some corrupted pixels since the decrypted data must be the exact plaintext data. Traditional encryption produces faulty decrypted images when decrypting a cipher image that has been affected by noise or has lost pixels, whether due to attacks or transmission errors. In multimedia applications, the content of an image is important, and minor distortions in the original image are acceptable [66].

Chaotic systems-based image encryption algorithms provide excellent security due to their high sensitivity to the initial parameters, fast computation speed, and minimal complexity. A potential disadvantage of chaotic systems-based image encryption algorithms is the limited range of chaotic behavior for some chaotic maps such as logistic and tent maps. Moreover, certain chaotic maps, such as the Baker map, have been known to be vulnerable to chosen-plaintext attacks. This vulnerability could be mitigated by adding other layers of security, such as utilizing an IV or nonce or incorporating multiple chaotic maps. Furthermore, some chaotic maps are known to exhibit periodic behavior, meaning that after a certain number of permutation iterations, the image's pixels will return to their original position.

DNA-based image encryption is an innovative concept with high-security levels; however, it is still largely experimental and primarily explored in academic research. It has not yet found widespread practical application in data encryption scenarios. Furthermore, it requires specialized laboratory equipment to handle DNA synthesis and sequencing. This level of complexity hinders the practical application of DNA-based image encryption.

Neural network-based image encryption algorithms have primarily emerged due to advancements in artificial neural networks and deep learning. Once effectively trained, they can provide a quick means of image encryption. However, the level of security they offer remains a relatively unexplored domain. Further research and analysis on the security of neural network-based image encryption algorithms are imperative to comprehensively understand and quantify the robustness of these algorithms against potential threats and attacks.

Frequency domain-based image encryption techniques utilize confusion and diffusion operations within the frequency domain of an image. These techniques are computationally more expensive compared to chaotic system-based image encryption algorithms, and conceptually seem to provide a high-security level. Nevertheless, the security analysis of frequency domain-based image encryption methods remains unexplored in the literature.

Compressive sensing-based image encryption algorithms can reduce the size of image data and add an improved level of security to image transmission. These algorithms have the disadvantage of high computational overhead resulting from combining compressive sensing and encryption operations. Furthermore, the effectiveness of compressive sensing depends on choosing the measurement matrix, which can affect the overall performance and efficiency of the algorithm.

Meaningful image encryption algorithms have the advantage of keeping the embedded cipher image secure, even if it was extracted via steganalysis tools. Since the extracted data are already encrypted and are noise-like images, this adds another layer of security by preventing attackers from directly obtaining the secret image through steganalysis alone. However, these algorithms come with the drawback of high computational requirements due to the implementation of two layers of security.

The applications of image encryption are versatile, making it challenging to assign a specific technique to a particular use case. Image encryption techniques are relevant wherever image data protection and security are required. They are instrumental in personal communication, healthcare, finance, surveillance, law enforcement, and military applications. Nevertheless, certain image encryption techniques may excel for specific domains. Chaotic-based encryption algorithms are suitable for resource-constrained platforms, like the Internet of Things (IoT) and edge devices [62,67], due to their fast computational speed and minimal complexity. DNA-based algorithms can be utilized in encrypting and securely archiving data for extended periods, due to their minimal storage requirements. They also hold potential in espionage applications, as DNA sequences can be discreetly embedded in various forms, that is, physical objects, and require specialized equipment for retrieval.

Compressive sensing-based algorithms can be particularly useful in scenarios where there are constraints on data transmission bandwidth. For example, compressive sensing is well-suited for applications involving embedded cameras and drones, or any devices with limited data transfer capacity. Meaningful image encryption adds an additional layer of security through steganography, making it particularly suitable for covert communication.

Even though image encryption researchers utilize a comprehensive set of security, quality, and efficiency metrics to evaluate image encryption algorithms, there is a pressing need to establish rigorous security analysis and standardized evaluation metrics. Standardizing the evaluation metrics will not only better assess the resilience of image encryption methods, but will also guide researchers in testing their proposed algorithms. Researchers also need to consider the robustness of the image encryption algorithms against novel attacks, such as adversarial machine learning and quantum computing.

Table 6. Strengths and weaknesses of different image encryption approaches.

Approach	Strengths	Weaknesses

Traditional	Well-established;
	Secure;
	Standardized and tested
		High computational requirement;
		Slow encryption speed;
		Faulty decryption of cipher images with corrupted pixels

Chaotic systems	Fast computation; Minimal complexity	Small range of chaotic behavior
DNA	High security; Fewer storage requirements	Experimental and only explored in academic research Require specialized lab equipment
Neural networks	Fast encryption	Security level not explored
Frequency domain	Intuitively secure	Computationally intensive; Security analysis has not been explored
Compressive sensing	High security; Small size of cipher image	High computational overhead
Meaningful encryption	High security due to combining encryption and steganography	High computational overhead

5. Conclusions

The aim of this paper is to provide researchers and practitioners who are new to the field of image encryption with an introduction to image encryption and help enable them to grasp the current state of image encryption techniques and evaluation metrics. In this paper, a comprehensive review of several cutting-edge image encryption algorithms is conducted, focusing on their respective strengths and weaknesses. We categorized image encryption algorithms into seven distinct classifications based on the techniques they employ, that is, chaotic systems, DNA encoding, neural networks, compressive sensing, frequency domain, and meaningful sensing. Furthermore, this paper explores the security, quality, and efficiency metrics used in the literature to evaluate image encryption algorithms, such as correlation coefficient, histogram analysis, entropy, gray-level co-occurrence matrix (GLCM) analysis, mean square error (MSE), peak signal-to-noise ratio (PSNR), unified average change intensity (UACI), number of pixels change rate (NPCR), NIST SP 800-22 test suite, and more.

Although this paper elaborates on many of the prominent image encryption algorithms and provides a good review of the current state-of-the-art image encryption techniques and evaluation metrics, this paper is not an exhaustive list of all the image encryption algorithms. One of the future research directions will be to explore other image encryption algorithms not discussed in this paper, such as optical image encryption and quantum image encryption. Another future research direction will be to conduct security and complexity analyses of the prominent image encryption algorithms. Another future research direction will be to develop newer evaluation metrics for evaluating the security of different image encryption algorithms.

Author Contributions: Conceptualization, Y.A. and A.M.; methodology, Y.A. and A.M.; software, Y.A.; validation, Y.A.; formal analysis, Y.A. and A.M.; investigation, A.M.; resources, A.M.; data curation, Y.A.; writing—original draft preparation, Y.A. and A.M.; writing—review and editing, Y.A. and A.M.; visualization, Y.A. and A.M.; supervision, A.M.; project administration, A.M.; funding acquisition, Y.A. and A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. *NIST SP 800-22*; A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. National Institute of Standards and Technology: Gaithersburg, MD, USA, 2010.
2. Wang, X.Y.; Chen, F.; Wang, T. A new compound mode of confusion and diffusion for block encryption of image based on chaos. *Commun. Nonlinear Sci. Numer. Simul.* **2010**, *15*, 2479–2485. [CrossRef]
3. Dworkin, M.; Barker, E.; Nechvatal, J.; Foti, J.; Bassham, L.; Roback, E.; Dray, J. *NIST FIPS 197-upd1*; Advanced Encryption Standard (AES). Federal Inf. Process. Stds. (NIST FIPS). National Institute of Standards and Technology: Gaithersburg, MD, USA, 2001. [CrossRef]
4. Schneier, B.; Kelsey, J.; Whiting, D.; Wagner, D.; Hall, C.; Ferguson, N. Twofish: A 128-bit block cipher. *NIST AES Propos.* **1998**, *15*, 23–91.
5. Canniere, C.D.; Preneel, B. TRIVIUM Specifications. eSTREAM, ECRYPT Stream Cipher Project. 2006; Volume 2006. Available online: https://www.ecrypt.eu.org/stream/e2-trivium.html (accessed on 22 February 2024).
6. Bernstein, D.J. ChaCha, a variant of Salsa20. In Proceedings of the Workshop Record of SASC, Citeseer, Lausanne, Switzerland, 13–14 February 2008; Volume 8, pp. 3–5.
7. Ali, T.S.; Ali, R. A new chaos based color image encryption algorithm using permutation substitution and Boolean operation. *Multimed. Tools Appl.* **2020**, *79*, 19853–19873. [CrossRef]
8. Alghamdi, Y.; Munir, A.; Ahmad, J. A Lightweight Image Encryption Algorithm Based on Chaotic Map and Random Substitution. *Entropy* **2022**, *24*, 1344. [CrossRef]
9. Luo, Y.; Yu, J.; Lai, W.; Liu, L. A novel chaotic image encryption algorithm based on improved baker map and logistic map. *Multimed. Tools Appl.* **2019**, *78*, 22023–22043. [CrossRef]
10. Zhang, B.; Rahmatullah, B.; Wang, S.L.; Liu, Z. A plain-image correlative semi-selective medical image encryption algorithm using enhanced 2D-logistic map. *Multimed. Tools Appl.* **2022**, *82*, 15735–15762 . [CrossRef]
11. Elashry, I.F.; El-Shafai, W.; Hasan, E.S.; El-Rabaie, S.; Abbas, A.M.; Abd El-Samie, F.E.; El-sayed, H.S.; Faragallah, O.S. Efficient chaotic-based image cryptosystem with different modes of operation. *Multimed. Tools Appl.* **2020**, *79*, 20665–20687. [CrossRef]
12. Mondal, B.; Kumar, P.; Singh, S. A chaotic permutation and diffusion based image encryption algorithm for secure communications. *Multimed. Tools Appl.* **2018**, *77*, 31177–31198. [CrossRef]
13. Rachmawanto, E.H.; De Rosal, I.M.S.; Sari, C.A.; Santoso, H.A.; Rafrastara, F.A.; Sugiarto, E. Block-based arnold chaotic map for image encryption. In Proceedings of the 2019 International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 24–25 July 2019; pp. 174–178.
14. Shalaby, M.A.W.; Saleh, M.T.; Elmahdy, H.N. Enhanced Arnold's cat map-AES encryption technique for medical images. In Proceedings of the 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES), Giza, Egypt, 24–26 October 2020; pp. 288–295.
15. Li, C.; Luo, G.; Qin, K.; Li, C. An image encryption scheme based on chaotic tent map. *Nonlinear Dyn.* **2017**, *87*, 127–133. [CrossRef]
16. Vishwas, C.; Kunte, R.S. An image cryptosystem based on tent map. In Proceedings of the 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 20–22 August 2020; pp. 1069–1073.
17. Gao, X. Image encryption algorithm based on 2D hyperchaotic map. *Opt. Laser Technol.* **2021**, *142*, 107252. [CrossRef]
18. Li, Y.; Wang, C.; Chen, H. A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation. *Opt. Lasers Eng.* **2017**, *90*, 238–246. [CrossRef]
19. Phatak, S.; Rao, S.S. Logistic map: A possible random-number generator. *Phys. Rev. E* **1995**, *51*, 3670. [CrossRef]
20. Rohith, S.; Jahnavi, L.; Bhuvaneshwari, S.; Supreeth, S.; Sujatha, B. Image Encryption and Decryption Using Key Sequence of Triple Logistic Map for Medical Applications. In Proceedings of the 2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAECC), Bengaluru, India, 11–12 December 2020; pp. 1–5.
21. Arnold, V.I.; Avez, A. *Ergodic Problems of Classical Mechanics*; Benjamin, Amsterdam, The Netherlands , 1968; Volume 9.

22. Zheng, Y.; Jin, J. A novel image encryption scheme based on Hénon map and compound spatiotemporal chaos. *Multimed. Tools Appl.* **2015**, *74*, 7803–7820. [CrossRef]
23. Pradeep, D.A.; Harsha, A.; Jacob, J. Image Encryption Using Chaotic Map And Related Analysis. In Proceedings of the 2021 International Conference on Advances in Computing and Communications (ICACC), Kochi, Kakkanad, India, 21–23 October 2021; pp. 1–5.
24. Hussein, K.A.; Mahmood, S.A.; Abbass, M.A. A New Permutation-Substitution Scheme Based on Henon Chaotic Map for Image Encryption. In Proceedings of the 2019 2nd Scientific Conference of Computer Sciences (SCCS), Baghdad, Iraq, 27–28 March 2019; pp. 63–68.
25. Boriga, R.; Dăscălescu, A.C.; Priescu, I. A new hyperchaotic map and its application in an image encryption scheme. *Signal Process. Image Commun.* **2014**, *29*, 887–901. [CrossRef]
26. Vispoel, M.; Daly, A.; Baetens, J. Lyapunov exponents of multi-state cellular automata. *Chaos Interdiscip. J. Nonlinear Sci.* **2023**, *33*, 043108. [CrossRef] [PubMed]
27. Jasra, B.; Moon, A.H. Color image encryption and authentication using dynamic DNA encoding and hyper chaotic system. *Expert Syst. Appl.* **2022**, *206*, 117861. [CrossRef]
28. Ramasamy, P.; Ranganathan, V.; Kadry, S.; Damaševičius, R.; Blažauskas, T. An image encryption scheme based on block scrambling, modified zigzag transformation and key generation using enhanced logistic—Tent map. *Entropy* **2019**, *21*, 656. [CrossRef] [PubMed]
29. Jain, K.; Aji, A.; Krishnan, P. Medical image encryption scheme using multiple chaotic maps. *Pattern Recognit. Lett.* **2021**, *152*, 356–364. [CrossRef]
30. Wu, T.Y.; Fan, X.; Wang, K.H.; Lai, C.F.; Xiong, N.; Wu, J.M.T. A dna computation-based image encryption scheme for cloud cctv systems. *IEEE Access* **2019**, *7*, 181434–181443. [CrossRef]
31. Li, K.; Su, Z. Research on an Image High Intensive Encryption Way Based on the Chaos Theory and DNA Coding. In Proceedings of the 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Suzhou, China, 19–21 October 2019; pp. 1–7.
32. Li, X.; Li, X. A novel block image encryption algorithm based on DNA dynamic encoding and chaotic system. In Proceedings of the 2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP), Wuxi, China, 19–21 July 2019; pp. 901–906.
33. Feng, W.; He, Y.; Li, H.; Li, C. A plain-image-related chaotic image encryption algorithm based on DNA sequence operation and discrete logarithm. *IEEE Access* **2019**, *7*, 181589–181609. [CrossRef]
34. Uddin, M.; Jahan, F.; Islam, M.K.; Rakib Hassan, M. A novel DNA-based key scrambling technique for image encryption. *Complex Intell. Syst.* **2021**, *7*, 3241–3258. [CrossRef]
35. Man, Z.; Li, J.; Di, X.; Sheng, Y.; Liu, Z. Double image encryption algorithm based on neural network and chaos. *Chaos Solitons Fractals* **2021**, *152*, 111318. [CrossRef]
36. Yang, F.; Mou, J.; Cao, Y.; Chu, R. An image encryption algorithm based on BP neural network and hyperchaotic system. *China Commun.* **2020**, *17*, 21–28. [CrossRef]
37. Han, B.; Jia, Y.; Huang, G.; Cai, L. A medical image encryption algorithm based on hermite chaotic neural network. In Proceedings of the 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chongqing, China, 12–14 June 2020; Volume 1, pp. 2644–2648.
38. Wang, X.Y.; Li, Z.M. A color image encryption algorithm based on Hopfield chaotic neural network. *Opt. Lasers Eng.* **2019**, *115*, 107–118. [CrossRef]
39. Ding, L.; Ding, Q. A novel image encryption scheme based on 2D fractional chaotic map, DWT and 4D hyper-chaos. *Electronics* **2020**, *9*, 1280. [CrossRef]
40. Chen, Y.; Xie, S.; Zhang, J. A hybrid domain image encryption algorithm based on improved henon map. *Entropy* **2022**, *24*, 287. [CrossRef] [PubMed]
41. Guan, M.; Yang, X.; Hu, W. Chaotic image encryption algorithm using frequency-domain DNA encoding. *IET Image Process.* **2019**, *13*, 1535–1539. [CrossRef]
42. Faragallah, O.S.; Afifi, A.; El-Shafai, W.; El-Sayed, H.S.; Naeem, E.A.; Alzain, M.A.; Al-Amri, J.F.; Soh, B.; Abd El-Samie, F.E. Investigation of chaotic image encryption in spatial and FrFT domains for cybersecurity applications. *IEEE Access* **2020**, *8*, 42491–42503. [CrossRef]
43. Dou, Y.; Li, M. An image encryption algorithm based on compressive sensing and M Sequence. *IEEE Access* **2020**, *8*, 220646–220657. [CrossRef]
44. Zhang, M.; Tong, X.J.; Liu, J.; Wang, Z.; Liu, J.; Liu, B.; Ma, J. Image compression and encryption scheme based on compressive sensing and Fourier transform. *IEEE Access* **2020**, *8*, 40838–40849. [CrossRef]
45. Wang, X.; Su, Y. Image encryption based on compressed sensing and DNA encoding. *Signal Process. Image Commun.* **2021**, *95*, 116246. [CrossRef]
46. Ping, P.; Fu, J.; Mao, Y.; Xu, F.; Gao, J. Meaningful encryption: Generating visually meaningful encrypted images by compressive sensing and reversible color transformation. *IEEE Access* **2019**, *7*, 170168–170184. [CrossRef]
47. Wang, X.; Liu, C.; Jiang, D. Visually meaningful image encryption scheme based on new-designed chaotic map and random scrambling diffusion strategy. *Chaos Solitons Fractals* **2022**, *164*, 112625. [CrossRef]

48. Jiang, D.; Liu, L.; Zhu, L.; Wang, X.; Rong, X.; Chai, H. Adaptive embedding: A novel meaningful image encryption scheme based on parallel compressive sensing and slant transform. *Signal Process.* **2021**, *188*, 108220. [CrossRef]
49. Yang, Y.G.; Wang, B.P.; Yang, Y.L.; Zhou, Y.H.; Shi, W.M.; Liao, X. Visually meaningful image encryption based on universal embedding model. *Inf. Sci.* **2021**, *562*, 304–324. [CrossRef]
50. Ma, S.; Zhang, Y.; Yang, Z.; Hu, J.; Lei, X. A new plaintext-related image encryption scheme based on chaotic sequence. *IEEE Access* **2019**, *7*, 30344–30360. [CrossRef]
51. Khan, J.S.; ur Rehman, A.; Ahmad, J.; Habib, Z. A new chaos-based secure image encryption scheme using multiple substitution boxes. In Proceedings of the 2015 Conference on Information Assurance and Cyber Security (CIACS), Rawalpindi, Pakistan, 18 December 2015; pp. 16–21. [CrossRef]
52. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]
53. Wu, Y.; Zhou, Y.; Saveriades, G.; Agaian, S.; Noonan, J.P.; Natarajan, P. Local Shannon entropy measure with statistical tests for image randomness. *Inf. Sci.* **2013**, *222*, 323–342. : 10.1016/j.ins.2012.07.049 [CrossRef]
54. Sha, Y.; Cao, Y.; Yan, H.; Gao, X.; Mou, J. An image encryption scheme based on IAVL permutation scheme and DNA operations. *IEEE Access* **2021**, *9*, 96321–96336. [CrossRef]
55. Qayyum, A.; Ahmad, J.; Boulila, W.; Rubaiee, S.; Arshad.; Masood, F.; Khan, F.; Buchanan, W.J. Chaos-Based Confusion and Diffusion of Image Pixels Using Dynamic Substitution. *IEEE Access* **2020**, *8*, 140876–140895. [CrossRef]
56. Iqbal, N.; Hanif, M.; Abbas, S.; Khan, M.A.; Almotiri, S.H.; Al Ghamdi, M.A. DNA Strands Level Scrambling Based Color Image Encryption Scheme. *IEEE Access* **2020**, *8*, 178167–178182. [CrossRef]
57. Sanap, S.D.; More, V. Performance Analysis of Encryption Techniques Based on Avalanche effect and Strict Avalanche Criterion. In Proceedings of the 2021 3rd International Conference on Signal Processing and Communication (ICPSC), Coimbatore, India, 13–14 May 2021; pp. 676–679.
58. Yu, F.; Shen, H.; Yu, Q.; Kong, X.; Sharma, P.K.; Cai, S. Privacy Protection of Medical Data Based on Multi-Scroll Memristive Hopfield Neural Network. *IEEE Trans. Netw. Sci. Eng.* **2023**, *10*, 845–858. [CrossRef]
59. Wu, Y.; Noonan, J.P.; Agaian, S.; et al. NPCR and UACI randomness tests for image encryption. *Cyber J. Multidiscip. J. Sci. Technol. J. Sel. Areas Telecommun.* **2011**, *1*, 31–38.
60. Shafique, A.; Ahmed, J.; Rehman, M.U.; Hazzazi, M.M. Noise-Resistant Image Encryption Scheme for Medical Images in the Chaos and Wavelet Domain. *IEEE Access* **2021**, *9*, 59108–59130. [CrossRef]
61. Zhou, J.; Zhou, N.R.; Gong, L.H. Fast color image encryption scheme based on 3D orthogonal Latin squares and matching matrix. *Opt. Laser Technol.* **2020**, *131*, 106437. [CrossRef]
62. Munir, A.; Kansakar, P.; Khan, S.U. IFCIoT: Integrated Fog Cloud IoT: A novel architectural paradigm for the future Internet of Things. *IEEE Consum. Electron. Mag.* **2017**, *6*, 74–82. [CrossRef]
63. Munir, A.; Kwon, J.; Lee, J.H.; Kong, J.; Blasch, E.; Aved, A.; Muhammad, K. FogSurv: A Fog-Assisted Architecture for Urban Surveillance Using Artificial Intelligence and Data Fusion. *IEEE Access* **2021**, *9*, 111938–111959. [CrossRef]
64. Alvarez, G.; Li, S. Some basic cryptographic requirements for chaos-based cryptosystems. *Int. J. Bifurc. Chaos* **2006**, *16*, 2129–2151. [CrossRef]
65. Himthani, V.; Dhaka, V.S.; Kaur, M.; Singh, D.; Lee, H.N. Systematic Survey on Visually Meaningful Image Encryption Techniques. *IEEE Access* **2022**, *10*, 98360–98373. [CrossRef]
66. Ahmed, F.; Siyal, M.; Abbas, V.U. A perceptually scalable and jpeg compression tolerant image encryption scheme. In Proceedings of the 2010 Fourth Pacific-Rim Symposium on Image and Video Technology, Singapore, 14–17 November 2010; pp. 232–238.
67. Munir, A.; Blasch, E.; Kwon, J.; Kong, J.; Aved, A. Artificial Intelligence and Data Fusion at the Edge. *IEEE Aerosp. Electron. Syst. Mag.* **2021**, *36*, 62–78. [CrossRef]

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