Image Encryption Algorithm based on Particle Swarm Optimization and Chaos Logistic Map

Hussain Alibrahim
Department of Computer Science
North Dakota State University
Fargo, USA
hussain.alibrahim@ndsu.edu

Simone A. Ludwig
Department of Computer Science
North Dakota State University
Fargo, USA
simone.ludwig@ndsu.edu

***Abstract*—With the rapid growth of data transmission and sharing technology, image encryption is becoming a widely discussed topic in the field of information security. In this paper, Particle Swarm Optimization (PSO) algorithm with Chaos Logistic Map is applied to create an encryption algorithm. The proposed algorithm benefits from PSO's ability to be able to quickly search a large space, having a short computation time, and having a higher probability of finding the global optimum, and thus PSO produces a very good encryption image. Furthermore, the use of logistic map for confusion and diffusion operations is very beneficial. The algorithm starts by creating several encrypted images, which are the particles for PSO, and the logistics maps and the encryption key used is based on the plain image only. The optimization step is measuring the pixels correlation, where lower correlation values are better, and these serve as the fitness function. The simulation results of the proposed algorithm indicate an effective encryption process. In addition, the security analysis illustrates the ability of this algorithm to provide satisfying levels of security in comparison with other image encryption schemes.**

***Index Terms*—Image Encryption, Particle Swarm Optimization, PSO, Chaos Logistic Map**

I. Introduction

Nowadays the internet is used all over the world for different communication types and interactions with different goals. It could be family call, educational session, work meeting or even a secret military conference. In special circumstances people live in, like the covid 19 pandemic, it has become mandatory for people to meet over the internet. Unfortunately, this type of media is not safe and there are a lot of expert hackers or attackers trying to access the data for different purposes. Images is one type of data that is used to be shared, and these images my involve personal privacy information, sensitive trading data, military secrets and national security secrets. Protecting image information over the network or internet focuses on 3 main goals for security [1]:

- **Confidentiality:** Image data is not accessible for unauthorized users.
- **Integrity:** Protection of image data from unauthorized modifications.
- **Availability:** An image is available when its needed by an authorized user.

Compared to text data, digital images' characteristics are different since those have the following features: large amount of data, strong correlations, big data redundancy, storage format, etc. make the traditional encryption methods, such as Data Encryption Standard (DES), International Data Encryption (IDEA), and Advanced Encryption Standard (AES) are not so suitable for image encryption. These algorithms can not prevent statistical, differential and other attacks, and easily fail.

Every image encryption system is mainly composed of two parts: 1) secret key, and 2) encryption algorithm. According to the basic principle of cryptology, a cryptosystem should be sensitive to the secret key. One way to accomplish this requirement is the usage of a truly random key generation mechanism. In other words, based on secret keys, pseudo-random sequences are produced for the encryption of the image. The pseudo-random key stream is then used to mask and encrypt each plain-image pixel sequentially in the encryption algorithm.

Therefore, a variety of image encryption schemes have been proposed to achieve the goal of secure image transfer [1] such as Block based using substitution [2] or permutation [3], Bit Transform using Arnold [4] or Angular [5], Conventional based such as AES [6], DES [7] or RC5 [8], Chaos based algorithm such as Map [9], one dimensional Chaos [10] or Hyper Chaos [11], including miscellaneous based like DNA sequence [12], genetic algorithm [13], Double phase Random Encoding [14]. These algorithms listed are used in image encryption.

This paper organized as follow, related work is presented first, then the proposed approach used in this paper is introduced, followed by definition of the quality metrics used to evaluate proposed approach. The experiments, results, and conclusion complete the paper.

II. Related Work

In [2], a novel variant of Substitution-Box is used to encrypt the images. The main contribution was a novel and simple modular approach to construct nonlinear S-boxes, and dynamic permutation operation is applied to the values of S-box to create more confusion. For S-box having size $n \times n$, the novel transformation function is represented as: $L(z) = [A \times z + B]\, MOD\, (2^n + 1)$ $z \in N$, where $N = \{0, 1, 2...255\}, O = \{1, 3, 5, ...255\}, A \in O$ and $B \in N$.

Each S-box method needs to find the Multiplicative Inverses (MI) for each value in the box. In this paper, a simpler approach is used to find the MI. Instead of using the Galois field, which is considered a complicated process. This approach is represented by $MI(L(z)) = L(z)\, MOD\, (2^n + 1)$. MI is used in the permutation process to make it dynamic with a large search space, i.e., for a S-box of size 8×8, the total number of permutations is $2^{16}!$. An experiment is conducted to evaluate different statistical performance measures such as histogram analysis, difference analysis, and similarity analysis using benchmark images. The results are compared with other research work and it was found that this algorithm can improve image encryption using the S-box process.

In [10], a hyper algorithm based on Genetic algorithm and DNA sequence is used in image encryption. A DNA sequence is selected as it offers greater storage and higher computing capabilities. The encryption method consists of two phases: a Transposition or Scrambling phase and a Substitution phase. In the first phase, pixel locations are altered using GA to reduce the correlation among adjacent pixels. In the substitution phase, the pixels are replaced by using an XOR operation between the pixel values converted into binary strings, and DNA substrings are derived from a random DNA string. DNA substrings are used as keys for the image encryption. The experimental result confirms that the algorithm is simple, fast, and feasible. The performance analysis outlined the robustness of the algorithm against all kinds of attacks and thereby maintaining higher security.

In [15], a modern framework is presented using the neighborhood nonlinear map within the Coupled Map Lattices (CML). The approach was connected to the instrument of permutation-diffusion. The encryption scheme chaos considered that the merits of spatio-temporal chaos and the Nonlinear Chaotic Algorithm (NCA) is a great method that produces eccentric chaotic sequences.

In [16], an effective scheme for image encryption is presented that is dependent on the settled nested chaotic map and Deoxyribonucleic Acid (DNA) utilizing the Secure Hash Algorithm (SHA-256) to produce the initial states of the chaotic attractor, and introduced a new chaotic system dependent on Julia's fractal procedure, tumultuous attractors, and logistic map in a complex set.

In [17], a new form of PSO has been developed using chaotic maps (tent map and logistic map) and Gaussian mutation. PSO's shortcomings are that it can get easily stuck in local optima and can also lead to early convergence during the search process. To address these issues the chaotic map is employed to initialize uniform distributed particles so as to improve the quality of the initial population, which is a simple yet very efficient method to improve the quality of the initial population. Furthermore, the Gaussian mutation mechanism based on the maximal focus distance is implemented to help the algorithm escape from the local optima and make the particles proceed with searching in other regions of the solution space until the global optimal or the closer to optimal solutions can be found. Experimental results on two benchmark functions demonstrate the effectiveness and efficiency of the PSO algorithm.

In [18], PSO and five popular chaotic maps: logistic, singer, sinusoidal, tent, and Zaslavskii have been integrated to build effective docking applications. These programs are routinely used in structure-based drug design to find the optimal binding pose of a ligand in the protein's active site. These programs are also used to identify potential drug candidates by ranking large sets of compounds. Pose prediction experiments indicate that chaos-embedded algorithms outperform docking algorithms in ligand pose root mean square deviation, success rate, and run time. In virtual screening experiments, the proposed system achieved a very significant five to sixfold speedup with comparable screening performances compared to AutoDock Vina in terms of area under the receiver operating characteristic curve and the enrichment factor.

In this paper, PSO is used with logistic map to create an image encryption method. This approach is used because PSO by default is fast, robust, and has a short computation time. However, the canonical PSO can get trapped in a local minimum and for this reason Logistic map is used. Furthermore, Logistic map provides a higher level of ambiguity in the encryption process, which makes the algorithm strong against any attacks.

III. Our Approach

This approach we are integrating Chaotic Logistic Map within PSO to implement an encryption algorithm. Chaos theory [19] in math is "the study of apparently random or unpredictable behavior in systems governed by deterministic laws. A more accurate term, deterministic chaos, suggests a paradox because it connects two notions that are familiar and commonly regarded as incompatible" these notions are randomness and deterministic behavior. However, the randomness of chaotic complex systems are governed by underlying patterns and deterministic laws, and thus, there is interconnectedness, constant feedback loops, repetition, self-similarity, fractals, and self-organization. The most common element in chaos systems is a very high degree of sensitivity to initial conditions and to the way in which they are set in motion.

In general, chaos-based image encryption algorithms consist of two steps: pixel permutation and pixel diffusion. Pixel permutation changes the pixel position, while pixel diffusion alters the pixel values where a change in a pixel will spread almost to other pixels of entire image. Contributed by the sensitivity properties of chaotic system, chaos-based image encryption algorithms generally achieve good performance in terms of security.

Logistic map [20] is one polynomial mapping of degree 2 popularized and published by biologist Robert May in 1976 as a discrete-time demographic model. This map is expressed by Equation (1):

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

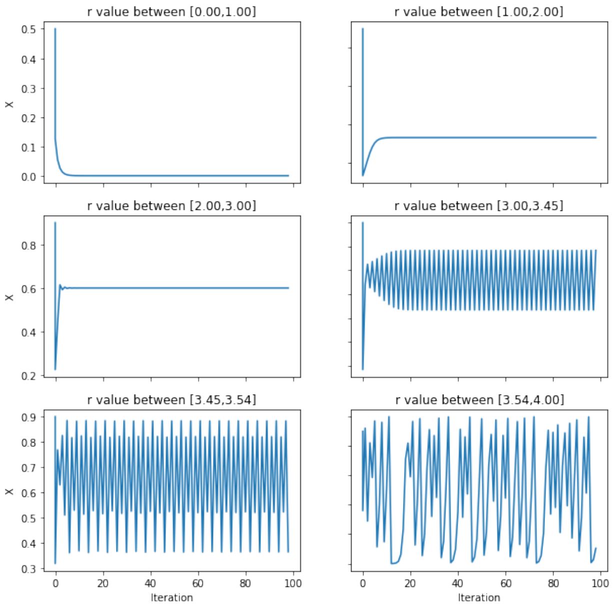


Fig. 1: Logistic Map Behavior

where X_n is the ratio of the existing population and the maximum possible population, it can be vary between 0 and 1, and r is the values of interest with values between 0 and 4.

The Logistic map behavior [21] is regardless of the initial population X and can be determined depending on the r value. The behavior can be summarized as follow:

- r value less than 1.0, the population will end to zero.
- r value between 1.0 and 2.0, the population will stabilize on a fixed value after few iteration.
- r value between 2.0 and 3.0, the population will stabilize on fixed value after fluctuates in first few iteration.
- r value between 3.0 and 3.45, the population will fluctuate around 2.
- r value between 3.45 and 3.54, the population will fluctuate around 4.
- r value between 3.54 and 4.0, the population will exhibit chaotic behavior.

Figure 1 shows the behavior for logistic map for 100 iteration using a different value of r regardless of the initial population value.

PSO [22] is a computational method that is used for optimization. It iteratively improves a candidate solution by improving the fitness function value. It is based on a population of candidate solutions that are called particles, and these particles keep moving in the search space and update the velocity and position values (both are randomly initialized at the beginning) by using the PSO equations. Equation (2) describes the velocity update:

$$\upsilon_i(t+1) = w \times \upsilon_i(t) + c_1 \times u_1 \times (Pbest_i(t) - X_i(t)) + c_2 \times u_2 \times (Gbest_i(t) - X_i(t)) \tag{2}$$

and Equation (3) describes the position update:

$$X_i(t+1) = X_i(t) + \upsilon_i(t+1) \tag{3}$$

All particles move with the guide of the global best position that is shared between the swarm members after each iteration. The global best is also updated by comparing the value with all particles positions' value and taking the best particle position in each iteration if it is better than the old global best.

Our approach starts with the initialization phase, as given in Algorithm 1 reading the image and then based on the population size the same number of encrypted images will be generated. These encrypted images are the particles in PSO. Each particle position and velocity are dependent on random initialization, and this number value is less than the image pixels number. In addition to the particle position and velocity, the image coefficient correlation is measured and that value is used to define the particle best position. The definition of the Correlation coefficient and its calculation are described in the simulation experiment section. After all particles are created, the particles' best positions and the swarm global best is updated.

Algorithm 1: Initialization

```
Input  : Plain Image
Output: Encrypted Image
Read image ($I$) of size ($M \times N$)
Read population size ($P$)
for $p$ in $P$ do
    $X$ = random number $\leq (M \times N)$
    $key$ = GenerateKey($X, I$)
    $eimage$ = Encrypt($image, key$)
    $velocity(p) = \lfloor X/M \rfloor$
    $position(p) = X$
    $p\_best\_pos = X$
    $p\_b\_ccf$ = CorrCoef($p$)
    if $g\_best > p\_b\_ccf$ then
        $g\_best = p\_b\_ccf$
        $g\_best\_pos = p\_best\_pos$
    end
end
```

The initialization phase uses two methods, which are key generation and the encryption method. The Generate keys method, as outlined in Algorithm 2, the objective is to return the encryption key of size 40 bits that is used in the encryption process based on the plain image and a random value only. Using a random number from the input parameter, the method will figure out the row and column of the first pixel, then it will read diagonally four additional pixels, convert each pixel value to a binary value, concatenate all binary values and finally return it as the encryption key. For each particle, this random number has to be saved since it is the only key required for the decryption process.

```
Algorithm 2: Generate Key
  Input : Number X
          Image I (Size $(M \times N)$)
  Output: Key
  $r = \lfloor \frac{X}{M} \rfloor$
  $c = N \mod X$
  $key =$
    $concatenate(binary(I(r, c)), binary(I(r + 1, c + 1))$ to
    $binary(I(r + 4, c + 4)))$
  return key
```

The encryption process as given in Algorithm 3 uses the logistic map equation. As described earlier, for this equation the initial population X_n and the increasing rate r are predefined. The encryption key used to calculate X_0 value is given in Equation (4).

$$X_0 = \frac{key[1] \times 2^{39} + key[2] \times 2^{38} + ... + key[40] \times 2^{0}}{2^{40}} \tag{4}$$

The r value used is from where the logistic map behaves chaotically. After that loop is completed, each pixel of the image is XOR with $(X_0 \times 256)$, and the result is an encrypted image that is returned to the initialization phase as a particle.

```
Algorithm 3: Encryption Process
  Input : Key key
          Image img $(M \times N)$
  Output: Encrypted Image eimge
  $X_0$ (calculated as Equation (4) using $key$)
  for $i = 1$ to $M$ do
    for $j = 1$ to $N$ do
      $eimg(i, j) = \lfloor (X_0 \times 256) \oplus img(i, J) \rfloor$
    end
  end
  return eimg
```

The last step in this approach is to optimize the solution using PSO, that means finding the lowest value of coefficient correlation of the image. For this in each iteration, the particle velocity and position are updated using Equations (2), (3), and the coefficient correlation value is calculated. The calculation process for the coefficient correlation uses the index value of the particle as a random number, particles as image, then generates the keys, encrypt images, and finally calculates the correlation coefficient value. After that update, the particle's best position and swarm global best position is modified. This process continues with the next iteration. At the end, an image with best global position as an encrypted image is returned.

The decryption process is the opposite of encryption process. It uses the random number selected during the encryption process and the chaotic function logistic map with the same r values.

IV. Simulation Experiments

To test this approach, the algorithm was implemented in python and uses different measures to show the strength of the proposed algorithm. It is tested using eight benchmark images: Lena, Peppers, Baboon, Barbara, Gold Hill, Cameraman, Fruits, and Sail Boat. All the tests are applied to grayscale images of size 512×512. This algorithm can be applied to color images using RGB color analysis instead of only one color.

A. Correlation Coefficient

As in [23], the Pearson correlation coefficient (CCF) is a statistical metric to measure the strength and direction of the linear relationship between any two random variables. It has been used in many different fields such as classification, clustering, finance analysis, and in biological research. In this paper, our approach uses CCF as the fitness function of PSO, i.e., it aims to receive an image with a very low correlation coefficient value (close to 0).

In a plain image, an unencrypted one, any adjoining pixels have very high correlation. For an encrypted image we want to have at best no correlation so that it is save to transfer image information and data and prevent any statistical attacks. CCF is the covariance of two variables, divided by the product of their standard deviations; thus, it is essentially a normalized measurement of the covariance such that the result always has a value between -1 and 1. The CCF values close to 1 and -1 mean high correlation, while a value close to 0 means no correlation at all. CCF is calculated as given in Equation (5):

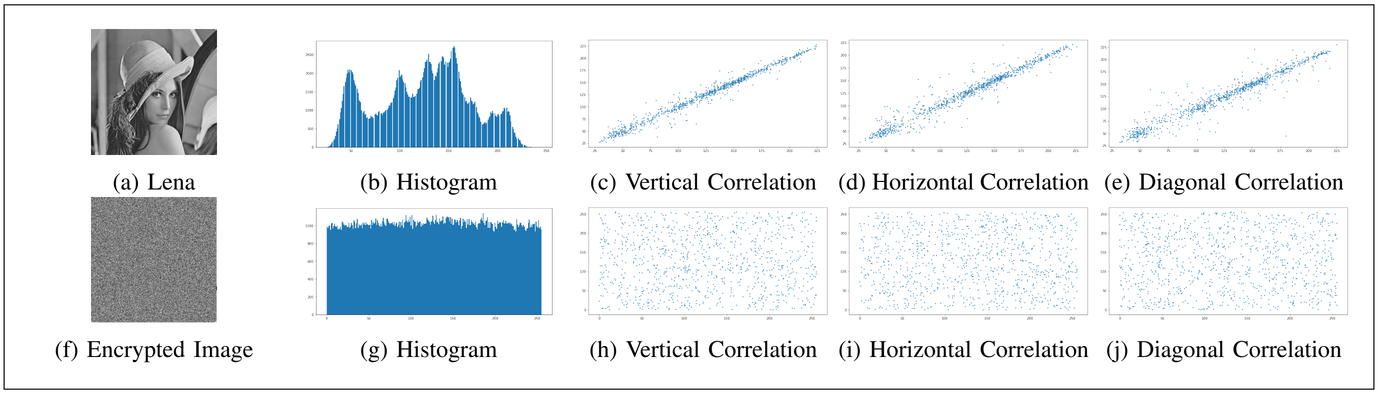
$$CCF = \frac{N \sum_{i=1}^{N}(x_i \times y_i) - \sum_{i=1}^{N} x_i \times \sum_{i=1}^{N} y_i}{\sqrt{(N \sum_{i=1}^{N} x_i^2 - (\sum_{i=1}^{N} x_i)^2) \times (N \sum_{i=1}^{N} y_i^2 - (\sum_{i=1}^{N} y_i)^2)}} \tag{5}$$

where N is the total number of pixels, x and y are any adjacent pixel values.

Table I shows the correlation value and compares it with results from different papers using different techniques. In most of the cases, thr proposed approach shows higher performance in 'destroying' the correlation. For example in the Lena image, the value was 9.5×10^{-8} whereas in [24] it was 6.8×10^{-6} and this was the nearest value. The Peppers image is another example with a CCF value of 6.1×10^{-6}, while in [25] it was only 5.2×10^{-5}. The Gold Hill image though, as given in [24], has a better value than using our approach, however the difference is small (the value is 2.93×10^{-5} while our is 3.00×10^{-5}).

TABLE I: Pearson Correlation Coefficient

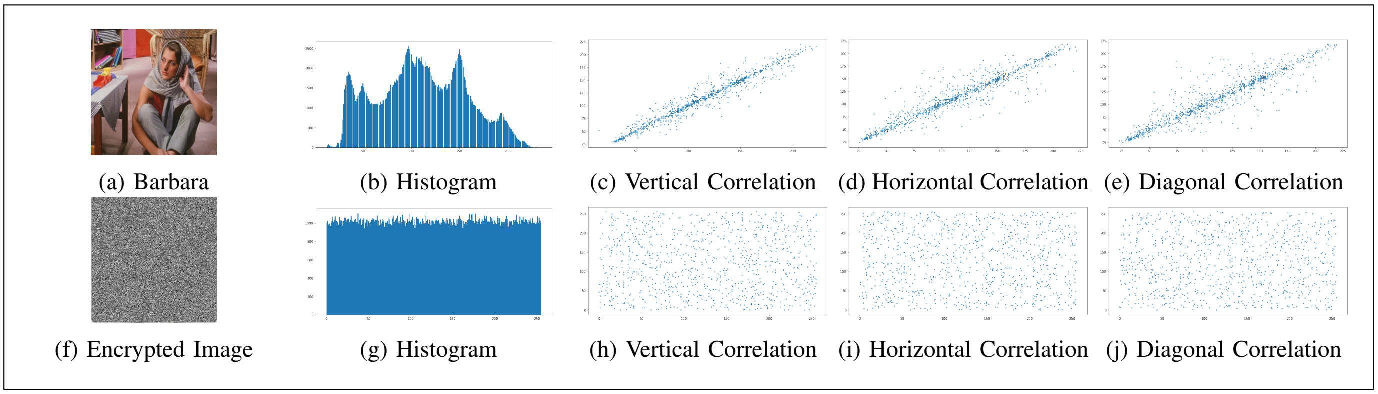
Image	Ours	Ref. [25]	Ref. [24]	Ref. [2]
Lena	9.46E-08	4.83E-05	6.82E-06	NA
Peppers	6.11E-06	5.25E-05	2.46E-04	3.00E-04
Baboon	2.96E-06	5.15E-04	NA	NA
Barbara	1.65E-05	2.18E-04	2.26E-04	1.30E-03
Gold Hill	3.00E-05	NA	2.93E-05	2.00E-04
Cameraman	2.48E-05	NA	NA	NA
Fruits	1.69E-05	NA	NA	NA
Sail Boat	1.03E-05	5.15E-05	1.44E-06	NA



(a) Lena (b) Histogram (c) Vertical Correlation (d) Horizontal Correlation (e) Diagonal Correlation

(f) Encrypted Image (g) Histogram (h) Vertical Correlation (i) Horizontal Correlation (j) Diagonal Correlation

Fig. 2: Lena Image and Encrypted Image with Histogram and Coefficient Correlation



(a) Barbara (b) Histogram (c) Vertical Correlation (d) Horizontal Correlation (e) Diagonal Correlation

(f) Encrypted Image (g) Histogram (h) Vertical Correlation (i) Horizontal Correlation (j) Diagonal Correlation

Fig. 3: Barbara Image and Encrypted Image with Histogram and Coefficient Correlation

While Table I shows the pixel correlation with all adjacent pixel, Table II shows more specific information. It displays the result of pixel correlation with vertical adjacent pixels alone, horizontal adjacent pixels, and diagonal adjacent pixels separately. For example, the Lena image after encryption has 0.0010 vertical correlation, 0.0002 horizontal correlation and 0.0006 diagonal correlation, while the values reported in [26] are -0.0400, 0.0005 and 0.0030. Another example is the Fruits image with values 0.008, 0.004 and 0.002 while in [28] the values are -0.0155, -0.0129 and 0.0012. Most result of the proposed algorithm are better than the other compared approaches except the horizontal correlation for the camera man image. Figures 2-5 visualize the vertical, horizontal, and diagonal correlation coefficient for Lena, Barbara, Camera man, and Sail Boat in image (c), (d) and (e) before the encryption process, and a clear relation can be seen in all images. Images (h), (i), and (j) show the correlation after the encryption process. From these images its clear that the correlation values are small.

With these results of the correlation coefficient, the proposed PSO with logistic map approach proves that the security in image encryption can be improved and thus almost all the relation between the image pixels are destroyed.

TABLE II: Vertical, Horizontal & Diagonal Correlation

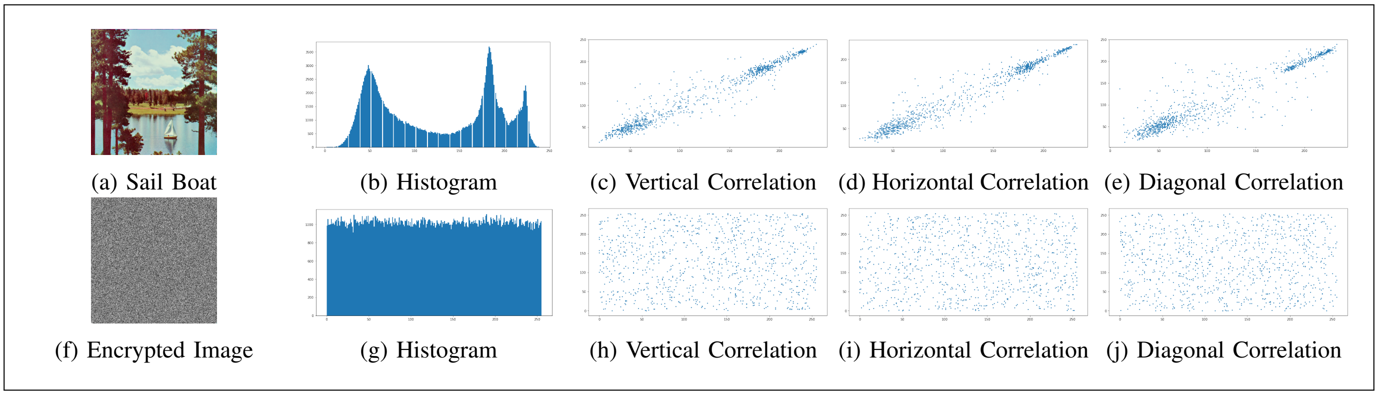
	Ours			Reference [26]		
Image	**V-Corr**	**H-Coor**	**D-Corr**	**V-Corr**	**H-Coor**	**D-Corr**
Lena	0.001142	0.000224	0.000570	-0.03911	0.00047	0.00305
Peppers	0.002927	0.001750	0.002813	0.04321	0.00198	0.02547
Baboon	0.002717	0.002981	0.000898	0.00285	0.00318	-0.00294
Barbara	0.001748	0.001806	0.000497	NA	NA	NA
Gold Hill	0.008919	0.000788	0.004296	NA	NA	NA
Cameraman	0.001219	0.006339	0.000803	0.0019	0.00212	-0.00205
Fruits	0.007627	0.004159	0.002185	NA	NA	NA
Sail Boat	0.000966	0.009781	0.007364	NA	NA	NA

B. *Histogram Analysis*

An image histogram [29] is a graphical representation of the color distribution for an image. Color is defined by using three primary ways:

- **Hue:** refers to the color only.
- **Saturation:** is the intensity or purity of a hue.
- **Lightness:** is the relative degree of black or white mixed with a given hue.

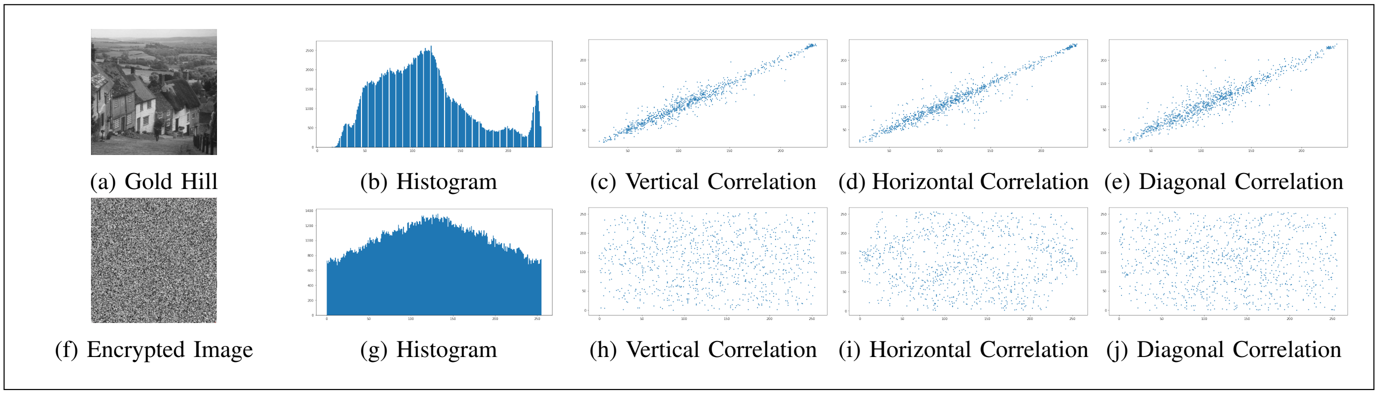
In this paper, grayscale images are used and thus the gray histogram analysis tested which is the statistical number of each different gray level ranging between 0 to 255 of all picture pixels. This measures the level of gray color in the image before and after the encryption and will plot the



(a) Sail Boat (b) Histogram (c) Vertical Correlation (d) Horizontal Correlation (e) Diagonal Correlation

(f) Encrypted Image (g) Histogram (h) Vertical Correlation (i) Horizontal Correlation (j) Diagonal Correlation

Fig. 4: Sail Boat Images and Encrypted Image with Histogram and Coefficient Correlation



(a) Gold Hill (b) Histogram (c) Vertical Correlation (d) Horizontal Correlation (e) Diagonal Correlation

(f) Encrypted Image (g) Histogram (h) Vertical Correlation (i) Horizontal Correlation (j) Diagonal Correlation

Fig. 5: Gold Hill Images and Encrypted Image with Histogram and Coefficient Correlation

histogram for each. The distribution of pixels is an indicator of the image content. A perfectly encrypted image has random-noise and the image tends to ideally have a flat or uniform distribution of pixels. Figures 2-5 subfigure (a) shows the original image, (b) shows the original image histogram, (f) shows the image after encryption, and (g) shows the histogram of the encrypted images.

It is clear that each plain image has a histogram that has several peaks and normal distribution of gray level. However, the corresponding encrypted images are quite noisy and meaningless to a casual observer, thus, illustrating the indistinguishability of an encrypted image. Moreover, an effective encryption effect can also be confirmed by analyzing the histogram plots as shown in subfigure (b). The histograms of encrypted images depict that the distribution of pixels in each encrypted images is more flat than their plain images histograms as well as substantially better than the histograms of the encrypted images obtained in [24].

C. Image Entropy

Shannon entropy [30] was introduced in 1948 by Claude Shannon in his paper “A Mathematical Theory of Communication”. Since then, Shannon entropy has been widely used in the information sciences. Shannon entropy is a measure of the uncertainty associated with a random variable. Specifically, the Shannon entropy quantifies the expected value of the information contained in a message. In image encryption it processes the level of gray in image, as represented in a histogram, is the information of the message needed to be encrypted. In the original image the entropy value represents the amount of data to be encrypted to make the image highly random, while in the encryption image the value represents the amount of data need to decrypted the image in order to obtain a meaningful one. the maximum value of entropy is 8 since $log_2 256 = 8$. Entropy for image X can be calculated using Equation (6):

$$E(X) = \sum_{l=1}^{L-1} \frac{n_l}{T} Log_2 \frac{n_l}{T} \tag{6}$$

where L is the gray scale from 0 to 255, T is the number of pixels, and n_l is the l-th pixel value.

By looking at Table III, our proposed approach returns very high entropy values, even when compared to the original image. For example, the plain Lena image has an entropy value of 7.4455, Barbara’s is 7.6278, Peppers’ is 7.5982, and the Gold Hill image is 7.4778. Most of the entropy values of encrypted images have a value greater than 7.9990, which is very high value since the ideal value is 8. Comparing to

TABLE III: Entropy

Image	Ours	Ref. [25]	Ref. [26]	Ref. [24]	Ref. [2]	Ref. [27]
Lena	7.99982	7.99746	7.99930	7.97200	NA	7.90230
Peppers	7.99943	7.99423	7.99940	7.97970	7.57140	7.90240
Baboon	7.99982	7.99664	7.99930	NA	NA	NA
Barbara	7.99936	7.99775	NA	7.98520	7.63210	NA
Gold Hill	7.97502	NA	NA	7.97910	7.48020	NA
Cameraman	7.99739	NA	7.99910	NA	NA	NA
Fruits	7.99928	NA	NA	NA	NA	NA
Sail Boat	7.99921	7.99472	NA	7.97890	NA	7.90190

other related research papers, their results are also very good, however, our approach has a better performance for almost for all of the images except for the Camera man image where the value in [26] was reported to be 7.999 and for our approach it is 7.997. These results indicate that this approach is efficient enough raising the randomness in the encrypted images and has strength to resist entropy based attacks.

D. Differential Analysis

Differential attack is a branch of study in cryptography that compares the way differences in the input relate to the differences in the encrypted output. The prime objective of this analysis is to study block ciphers to verify if changes in the plaintext result in any non-random results in the encrypted ciphertext. The importance of random change in ciphertext, if changed in plain text, indicates the strength in the encryption scheme. This high randomness level prevents any unauthorized access to the data from gaining information about what was encrypted or how it was encrypted by monitoring data changes.

As for text encryption, the same analysis can be applied to image encryption in order to evaluate an algorithm's strength and weakness. Further detail about this in [31]. The following analysis was conducted for the proposed approach.

1) Number of Changing Pixel Rate (NPCR): NPCR is to quantify the number of pixel changes between two encrypted images for the same plain image with single pixel change before the second encryption. This process is used to evaluate the effect of change in results of the encrypted image. Assume C^1 and C^2 are the encryption results for the same image while C^2 encryption process is started after the single pixel change in the original image. NPCR is calculated by constructing a two dimensional array of the image pixel size. Each element value in the array is either 0 or 1, and it is based on whether the pixel value in C^1 and C^2 are equal or not. This can be represented mathematically as shown in Equation (7):

$$D(i,j) = \begin{cases} 0, & \Omega C^1(i,j) = C^2(i,j) \\ 1, & \Omega C^1(i,j) \neq C^2(i,j) \end{cases} \tag{7}$$

Then, Equation (8) is used to calculate the NPCR value where T represents the total number of pixels:

$$NPCR = \sum \frac{D(i,j)}{T} \times 100 \tag{8}$$

Table IV shows the result of the NPCR value of eight images and compares other related research approach results. Seven image NPCR values where above 99.5% and the last one, Baboon, was 99.22%. These results show that our algorithm is highly random and leads to a large change in the result with a value of greater than 99%. Furthermore, our algorithm shows high performance compared to other work, in most of the test images while in some cases like the Lena image the result was 99.54% whereas in [26] a value of 99.66% is reported.

TABLE IV: NPCR

Image	Ours	Ref. [25]	Ref. [26]	Ref. [24]
Lena	99.542	99.645	99.664	99.228
Peppers	99.634	99.614	99.629	99.167
Baboon	99.221	99.583	99.644	NA
Barbara	99.609	99.272	NA	99.253
Gold Hill	99.594	NA	NA	99.237
Cameraman	99.660	NA	99.652	NA
Fruits	99.611	NA	NA	NA
Sail Boat	99.583	99.558	NA	99.191

This is where authors provide additional information about the data, including whatever notes are needed.

2) Unified Averaged Changed Intensity (UACI): UACI is a value that determines the average intensity of differences regarding the plain and cipher images. It is calculated by the summation of the differences in the pixels between C^1 and C^2. Then, this value is divided by the multiplication of the total number of pixels (T) and the largest supported pixel value (F) which is 255. Equation (9) represents the calculation of UACI.

$$UACI = \sum \frac{|C^1(i,j) - C^2(i,j)|}{F \times T} \times 100 \tag{9}$$

Table V presents the results of the UACI values achieved by our approach. Most values achieved are greater than 33.4%, which implies a High sensitivity encryption algorithm. The Fruits image is an exception where its value was around 32%. Comparing these results with other related approaches shows that our algorithm is stronger in creating uncertainty in the encrypted images in most cases.

TABLE V: UACI

Image	Ours	Ref. [25]	Ref. [26]	Ref. [24]	Ref. [2]	Ref. [27]
Lena	33.454	33.561	33.612	30.147	NA	33.460
Peppers	33.848	33.587	33.601	30.667	33.840	33.430
Baboon	33.637	33.611	33.643	NA	NA	NA
Barbara	33.748	33.554	NA	30.972	33.200	NA
Gold Hill	33.705	NA	NA	30.485	33.130	NA
Cameraman	33.658	NA	33.643	NA	NA	NA
Fruits	32.231	NA	NA	NA	NA	NA
Sail Boat	33.633	33.483	NA	31.159	NA	33.470

V. Conclusion

This paper presents a new algorithm for image encryption. The algorithm utilizes the power of PSO in solving optimization problems, and benefits from logistic map chaotic behavior to increase the security level. The algorithm's main idea is to find the lowest pixel correlation. It has been implemented in python and tested using eight benchmark images. The encrypted images were evaluated using different performance measures such as pixel coefficient correlation, histogram analysis, entropy analysis, pixel changing rate, and unified averaged changed intensity. The results demonstrate our algorithm's efficacy in comparison to other recent image encryption approaches. Moreover, the proposed encryption approach demonstrates better performance and higher resistance against analytical attacks in comparison to other works. Hence, the good performance of our encryption approach in terms of CCF, pixels distributions, entropy, differential analysis is endorsed by the simulation outcomes and analyses.

References

[1] S. Geetha, P. Punithavathi, A. Magnus Infanteena and S. Siva Sivatha Sindhu, A Literature Review on Image Encryption Techniques, International Journal of Information Security and Privacy (IJISP), vol. 12 issue 3, July-September 2018, doi:10.4018/IJISP.2018070104.

[2] A. H. Zahid, E. Al-Solami and M. Ahmad, A Novel Modular Approach Based Substitution-Box Design for Image Encryption, in IEEE Access, vol. 8, pp. 150326-150340, 2020, doi: 10.1109/ACCESS.2020.3016401.

[3] J. Chen, Z. Zhu, C. Fu, H. Yu and L. Zhang, An efficient image encryption scheme using gray code based permutation approach, Optics and Lasers in Engineering, vol: 67, pp: 191-204, 2016, doi: 0.1016/j.optlaseng.2014.11.017.

[4] H. Liu, B. Zhao, and L. Huang, Quantum Image Encryption Scheme Using Arnold Transform and S-box Scrambling, Entropy, vol. 21, no. 4, p. 343, Mar. 2019.

[5] X. Kang, A. Ming and R. Tao, Reality-Preserving Multiple Parameter Discrete Fractional Angular Transform and Its Application to Color Image Encryption, in IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 6, pp. 1595-1607, June 2019, doi: 10.1109/TCSVT.2018.2851983.

[6] P. Sharma, H. Sabharwal, A New Image Encryption using Modified AES Algorithm and its Comparision with AES, International Journal Of Engineering Research & Technology (IJERT) vol: 09, issue 08,August 2020.

[7] S.Kumar and S. Srivastava, Image Encryption using Simplified Data Encryption Standard (S-DES), International Journal of Computer Applications, vol:104, no:2 October 2014, doi: 10.5120/18178-9070.

[8] R. Shahzadi, S. M. Anwar, F. Qamar, M. Ali and J. J. P. C. Rodrigues, Chaos Based Enhanced RC5 Algorithm for Security and Integrity of Clinical Images in Remote Health Monitoring, in IEEE Access, vol. 7, pp. 52858-52870, 2019, doi: 10.1109/ACCESS.2019.2909554.

[9] X. Chai, Y. Chen and L. Broyde, A novel chaos-based image encryption algorithm using DNA sequence operations, Optics and Lasers in Engineering, vol: 88, pp 197-213, 2017, doi: 10.1016/j.optlaseng.2016.08.009.

[10] S. K. Pujari, G. Bhattacharjee and S. Bhoi, A Hybridized Model for Image Encryption through Genetic Algorithm and DNA Sequence, Procedia Computer Science, vol: 125, pp 165-171, 2018, doi: 10.1016/j.procs.2017.12.023.

[11] S. Liansheng, D. Cong, Z. Xiao, T. Ailing and A. Anand, Double-image encryption based on interference and logistic map under the framework of double random phase encoding, Optics and Lasers in Engineering, vol: 122, pp 113-122 2019, doi: 10.1016/j.optlaseng.2019.06.005.

[12] L. Xingbin, X. Di and L. Cong, Quantum image encryption algorithm based on bit-plane permutation and sine logistic map, Quantum Information Processing, vol: 19 issue 8 , 2020, doi: 10.1007/s11128-020-02739-w.

[13] A. Mansouri and X. Wang, A novel one-dimensional sine powered chaotic map and its application in a new image encryption scheme, Information Sciences, vol: 520, pp: 46-62, 2020, doi: 10.1016/j.ins.2020.02.008.

[14] L. Yujia, J. Zhaoguo, X. Xiping, Z. Fuqi and X. Jiahong, Optical image encryption algorithm based on hyper-chaos and public-key cryptography, Optics and Laser Technology, vol:127, id: 106171, July 2020, doi:10.1016/j.optlastec.2020.106171.

[15] C. Y. Song, Y. L. Qiao, and X. Z. Zhang, An image en- cryption scheme based on new spatiotemporal chaos, Optik- International Journal for Light and Electron Optics, vol. 124, no. 18, pp. 33293334, 2013.

[16] N. B. Slimane, N. Aouf, K. Bouallegu and M. Machhout, An efficient nested chaotic image encryption algorithm based on DNA sequence, International Journal of Modern Physics C, vol. 29, no. 7, Article ID 1850058, 2018.

[17] D. Tian, Particle Swarm Optimization with Chaotic Maps and Gaussian Mutation for Function Optimization, International Journal of Grid Distribution Computing, Vol. 8, No.4, pp. 123-134, 2015, http://dx.doi.org/10.14257/ijgdc.2015.8.4.12.

[18] H. K. Tai, S. A. Jusoh, W. I. Siu, Chaos-embedded particle swarm optimization approach for protein-ligand docking and virtual screening, Journal of Cheminformatics 10, 2018, https://doi.org/10.1186/s13321-018-0320-9.

[19] Encyclopedia Britannica. 2021. chaos theory — Definition & Facts. [online] Available at: https://www.britannica.com/science/chaos-theory [Accessed 2 May 2021].

[20] M. Ausloos and M. Dirickx, The Logistic Map and the Route to Chaos From the Beginnings to Modern Applications. Berlin: Springer, 2006.

[21] M. T. Akter, Observation of Different Behaviors of Logistic Map for Different Parameters, International Journal of Applied Mathematics and Theoretical Physics, vol: 4, 2018, doi: 10.11648/j.ijamtp.20180403.14.

[22] D. Wang, D. Tan, and L. Liu, Particle swarm optimization algorithm: an overview., Soft Comput 22, pp: 387408, 2018. doi: 10.1007/s00500-016-2474-6.

[23] H. Zhou, Z. Deng, Y. Xia and M. Fu, A new sampling method in particle filter based on Pearson correlation coefficient, Neurocomputing, 2015, doi: 10.1016/j.neucom.2016.07.036.

[24] Y. Bedir, K. Fahmi, M. Ahmed and T. Ali, A novel image encryption/decryption scheme based on integrating multiple chaotic maps, American Institute of Physics, vol:10, issue: 7, 2020, doi: 10.1063/5.0009225.

[25] M. Ahmad, M. N. Doja and M. M. Sufyan Beg, Security analysis and enhancements of an image cryptosystem based on hyperchaotic system, Journal of King Saud University - Computer and Information Sciences, vol: 33, issue: 1, pp 77-85, 2021, doi: 10.1016/j.jksuci.2018.02.002.

[26] Y. Ibrahim, K. Fahmi, M. Mohamed and S. Ahmed, A New Image Encryption Scheme Based on Hybrid Chaotic Maps, Hindawi, vol:2020, 2020, doi: 10.1155/2020/9597619.

[27] G. Bin and L. Hai-Bo,Image Encryption Application of Chaotic Sequences Incorporating Quantum Keys, International Journal of Automation and Computing, vol: 17, issue:1, 2020, doi: 10.1007/s11633-019-1173-z.

[28] M. Khan and H. M. Waseem, A novel image encryption scheme based on quantum dynamical spinning and rotations. PLoS ONE vol: 13, no:11, 2018 doi: 10.1371/journal.pone.0206460.

[29] S. Xia, P. Chen, J. Zhang, X. Li and B. Wang, Utilization of rotation-invariant uniform LBP histogram distribution and statistics of connected regions in automatic image annotation based on multilabel learning, Neurocomputing, vol: 228, pp: 11-18, 2017, doi: 10.1016/j.neucom.2016.09.087.

[30] Y. Wu, Y. Zhou, G. Saveriades, S. Agaian, J.P. Noonan, P. Natarajan, Local Shannon entropy measure with statistical tests for image randomness, Information Sciences, vol 222, 2013, pp 323-342, Doi: 10.1016/j.ins.2012.07.049.

[31] Y. Wu, J. P. Noonan, S. Again, NPCR and UACI Randomness Tests for Image Encryption, Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT), Aprill 2011.