Applied Physics B (2020) 126:147
https://doi.org/10.1007/s00340-020-07480-x

Applied Physics B
Lasers and Optics

Color image encryption using minimax differential evolution-based 7D hyper-chaotic map

Manjit Kaur[1] · Dilbag Singh[1] · Vijay Kumar[2]

Received: 24 September 2019 / Accepted: 25 June 2020 / Published online: 12 August 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

Hyperchaotic maps are generally used in the encryption to generate the secret keys. The number of hyperchaotic maps has been implemented so far. These maps involve a large number of state and control parameters. The major concern is the estimation of these parameters. Because the estimation requires extensive computational search. In this paper, a 7D hyperchaotic map is used to produce the secret keys for image encryption. As this hyperchaotic map require a large number of initial parameters, the manual estimation is computationally extensive. Therefore, minimax differential evolution is utilized to provide the optimal parameters to the hyperchaotic map. The fitness of the parameters is evaluated using correlation coefficient and entropy. The secrets keys are then produced by the proposed hyperchaotic map. These keys are further used to perform the diffusion operation on the input image to generate the encrypted images. Extensive experiments are conducted to investigate the performance of the proposed approach considering the well-known measures. The comparative results show that the proposed approach performs significantly better as compared to the competitive approaches.

1 Introduction

In the last few decades, encryption approaches play a vital role to conceal the sensitive information transmitted over wireless or wired networks. The unauthorized persons are unable to reveal the secret message from the encrypted data. The encryption approaches have been widely used in various engineering fields such as medical, military, remote-sensing etc. [1]. The protection of sensitive data is a foremost requirement of the above-mentioned applications. Due to this, a number of encryption approaches have been developed in the last few years. The foremost concepts behind the encryption approaches are large key space and resistance against the security attacks [2, 3]. The encryption approaches try to break the statistical pattern after the encryption process. However, most of these approaches have frailer in the substitution process and attackers can easily break the encrypted data. These approaches become ineffective as these are unable to resist the traditional image processing attacks [4, 5].

The conventional approaches such as RSA, AES, DES, etc. were also utilized in the encryption process [6]. But, they do not produce the optimal encryption results due to high redundancy and correlation present in the image [7]. Recently, the chaos has been used to encrypt the image to resolve the above-mentioned problems. The chaotic maps are sensitive towards the initial condition, control parameter, and non-convergence [5]. Due to these features, the chaotic maps perform better for image encryption. A large number of chaotic-based image encryption has been developed in the last few years [6, 8]. Most of the chaotic-based image encryption algorithms utilize low dimensional chaotic maps as they are easy to implement. However, the major drawbacks of the logistic map are comparatively smaller key space and irregular dispersal of chaotic sequences. While high-dimensional chaotic maps perform better than the low one. But they need an appropriate estimation of initial parameters as their realization is based on the combination of large state and control parameters. The estimation of such parameters itself poses a challenge. Recently, researchers have started using image encryption techniques using the concept of fuzzy logic [9–13] and deep learning models [14–16] to encrypt the remote-sensing images [17–20].

✉ Dilbag Singh
dilbag@ieee.org

[1] Computer Science Engineering, School of Engineering and Applied Sciences, Bennett University, Greater Noida 201310, India

[2] Department of Computer Science and Engineering, NIT Hamirpur, Hamirpur, India

To address all the above-mentioned problems, an image encryption approach based on the optimized hyperchaotic map is proposed. The main contributions of this research work are

i A 7D hyperchaotic map is used to produce the secret keys that have a large key space.
ii A minimax differential evolution is utilized to provide the optimal parameters to the hyperchaotic map.
iii A diffusion process is applied to the input image using optimized keys to perform encryption.
iv A multi-objective fitness function is used to evaluate the fitness of generated solutions.
v The proposed approach is compared with the competitive approaches to evaluate its performance.
vi The proposed approach is also verified against many security attacks such as statistical, differential, and noise attacks.

The remaining structure of this paper as follows. The related work done in the field of chaotic-based image encryption is presented in Sect. 2. Section 3 presents the basic concepts of a 7D hyperchaotic map and minimax differential evolution. In Sect. 4, the proposed image encryption approach is described. Experimental results and discussions are presented in Sect. 5. Section 6 describes the conclusions.

2 Related work

In last few decades, a large number of chaotic based image encryption approaches have been developed. Psiarchik et al. [21] used Coupled map lattice (CML) to encrypt the images. Pareek et al. [22] utilized the 1D chaotic map to produce the secret keys to encrypt the image. However, this approach has less security due to smaller key space.

Kwok and Tang [23] developed a chaos-based image encryption that uses both skewed tent map and cat map for key generation. Fridrich advised an encryption approach consists of substitution and diffusion processes [24]. In the substitution process, a 3D chaotic map is used to permute the pixels of an image. While the values of pixels are modified in diffusion process. Chen et al. [25] used a 3D baker map and 3D cat map in the substitution process. Guan et al. [26] utilized a 2D cat map for the permutation of pixel position. Lian et al. [27] employed logistic map in the diffusion process and 2D chaotic map in the substitution process. It has better sensitivity towards secret key and security attacks.

Zhang et al. [28] proposed an image encryption approach that uses the concepts of DNA sequence and chaotic map. This approach resistant to exhaustive and statistical attacks. Mao et al. [29] extended baker map in 3D for image encryption. This approach is computationally faster than 2D baker map. Wang and Luan [30] developed 3D coupled logistic maps for encryption. However, the encryption approach suffers from the phase space reconstruction attacks [31]. Lio and Maio [31] proposed parameter varying logistic map for image encryption. It can resist phase space reconstruction. Wu et al. [32] adopted 2D logistic map for encryption. This approach is resistant to differential attacks. Ye and Huang [33] proposed chaotic image encryption that uses secure hash algorithm-3 to generate the secret key. They used both permutation and diffusion processes. Lian et al. [27] employed spatiotemporal chaotic system for image encryption. Their algorithm has better security at a low computational cost. Ye and Huang [34] proposed a premodular operation for pre-processing the image encryption. The self-adaptive encryption scheme is used for keystream generation. Contrast to Fridrich [24], Ye et al. [35] developed an image encryption approach comprises of modulation, permutation, and diffusion operations. They used the entropy for key generation. Furthermore, Solak et al. [36] assessed the chaotic encryption scheme, which is proposed by Fridrich. Xie et al. [37] proposed a scheme to optimize the attack performed on Fridrich's scheme [24]. Hua et al. [38] developed a cosine transform-based image encryption scheme. The chaotic scheme is used to scrambling the pixels. The small change in an image is largely affect the encrypted image.

Benlashram et al. [39] developed an image encryption approach based on pixel shuffling and 3D chaotic map. Lin et al. [40] developed a chaotic-based image encryption approach that uses DNA encoding scheme. The variable parameters are coupled with chaotic map to generate the random sequences for keystream. Li et al. [41] presented a hyperchaotic-based color image encryption that utilizes the scrambling and diffusion models. [42] implemented hyperchaotic-based image encryption approach that uses variable kernels and a DNA. Lin et al. [43] proposed a parallel image encryption algorithm based on 1D logistic map and 2D logistic dynamics system. Li [44] proposed fingerprint-based chaotic encryption approach. They used the concepts of blockchain and fingerprint to encrypt the image.Khan et al.[45] developed an encryption scheme, which is based on Brownian motion and chaotic iterative maps.

The existing literature shows that the chaotic based image encryption is prevalent in these days. Chaotic maps are sensitive to initial parameters; therefore, the estimation of these parameters is very crucial. Because there are some approaches available that can use time series trajectories to estimate the control parameters of the chaotic maps [46, 47]. If parameters get estimated, then the security of an image encryption can easily be compromised. Therefore, there is a need to produce the initial parameters that cannot be easily estimated. Meta-heuristic techniques can be used to achieve this objective.

3 Preliminaries

3.1 Seven-dimensional hyperchaotic map

Yang et al. [48] proposed a seven-dimensional (7D) hyperchaotic map by coupling 6D hyperchaotic map [49] with 1D linear map. In this hyperchaotic map, one nonlinear and two linear feedback controllers are added to the Lorenz map. It has complex behavior with simple algebraic structure. It has five positive Lyapunov exponents that shows it is exceptionally complex and better than the lower-dimensional maps. Due to high unpredictability, it can be utilized in secure communications. It also has large number of parameters under unique equilibria which significantly enhance the key space against brute-force attacks. It is the reason to choose 7D hyperchaotic map for image encryption and defined as

$$\left.\begin{aligned}
u_1' &= s(u_2 - u_1) + u_4 + ru_6,\\
u_2' &= pu_1 - u_2 - u_1u_3 + u_5,\\
u_3' &= -tu_3 + u_1u_2,\\
u_4' &= eu_4 - u_1u_3,\\
u_5' &= -iu_2 + u_6,\\
u_6' &= q_1u_1 + q_2u_2,\\
u_7' &= gu_7 + nu_4,
\end{aligned}\right\} \tag{1}$$

where u_1, u_2, u_3, u_4, u_5, u_6, and u_7 are initial state variables of map (1). n represents the coupling parameter. s, t, and p are the constant parameters. r, e, i, q_1, q_2, and g are the control parameters. Figure 1 shows the hyperchaotic behavior of map (1) with parameters $s = 10$, $n = 1$, $i = 9.9$, $t = 8/3$, $r = 1$, $q_1 = 1$, $p = 28$, $e = 2$, $q_2 = 2$, and $g = 1$. Besides, it has a unique unstable equilibrium $Q(0, 0, 0, 0, 0, 0, 0)$ and seven Lyapunov exponents (LEs). The obtained LEs of map (1) are $\mathrm{LE}_1 = 1.0000$, $\mathrm{LE}_2 = 0.4128$, $\mathrm{LE}_3 = 0.2255, \mathrm{LE}_4 = 0.1360, \mathrm{LE}_5 = 0.0880, \mathrm{LE}_6 = 0.0000$, and $\mathrm{LE}_7 = -12.5289$. Their corresponding eigenvalues (E) are $E_1 = -22.06230$, $E_2 = -2.6667$, $E_3 = 1$, $E_4 = 2$, $E_5 = 11.4755$, and $E_{6,7} = 0.037 \pm 0.3850i$. It can be seen that with only one equilibrium, map (1) can generate hyperchaotic attractors with five positive LEs.

3.2 Minimax differential evolution

A new minimax differential evolution is proposed by Qiu et al. [50] to find optimized solutions with best worst-case performances. Minimax differential evolution is designed to overcome the limitations of conventional evolutionary algorithms. Most of the evolutionary algorithms are not capable to solve the asymmetrical problems. Because conventional evolutionary algorithms find best-doing solutions using some conditions over solution space instead of uncovering the solutions along with best-worst-case performance. Therefore, conventional evolutionary algorithms suffer from the optimization cycle [51–54]. To remove this issue, minimax differential evolution optimizes the solution in such a way that the solutions move toward those regions which can provide better worst-case performance. The second problem of existing algorithms is to treat all the solutions equally to explore the worst-case scenarios. Due to this, the performance of overall optimization is degraded as a number of function evaluations are wasted to find the worst-case situations for poor solutions. This may cause an imbalance between solution quality and solution reliability. This problem is overcome by minimax differential evolution using the bottom-boosting approach to skip the number of unnecessary scenario evaluations for the exploration of worst-case scenarios. The next problem with conventional

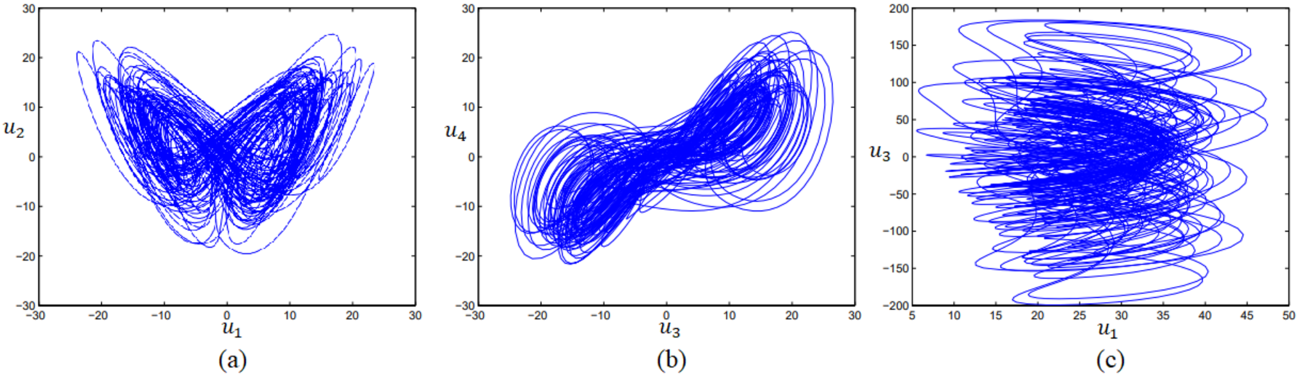

Fig. 1 Chaotic attractors of 7D hyperchaotic map: **a** $u_1 - u_2$ plane, **b** $u_3 - u_4$ plane, and $u_1 - u_3$ plane

algorithms is to find promising solutions correctly as well as inexpensively. Because if good solutions are needed, then algorithms suffer from high computational cost. Second, if a low-cost evaluation approach is used then poor solutions may propagate. To achieve both reliability and efficiency, minimax differential evolution developed a new mutation operator and partial-regeneration strategy. Therefore, a minimax differential algorithm can efficiently identify the promising solutions.

Figure 2 shows the working of minimax differential evolution. It contains mainly five steps that are described as follows.

Step 1. *Population initialization* In this step, the population is randomly generated, and each individual is a pair of scenarios and solutions. The required parameters such as dimensions of solution and scenario, control parameters, and maximum generations are also initialized.

Step 2. *Min heap construction* This step constructs min-heap by utilizing the current population at the start of every generation. In min-heap, every individual represented as a node and objective value of a related individual is the key of node. With this, a node that has a minimum key will be placed at the root. The main purpose to use min-heap is that it takes O(1) time to insert and O(log *n*) time to delete a node.

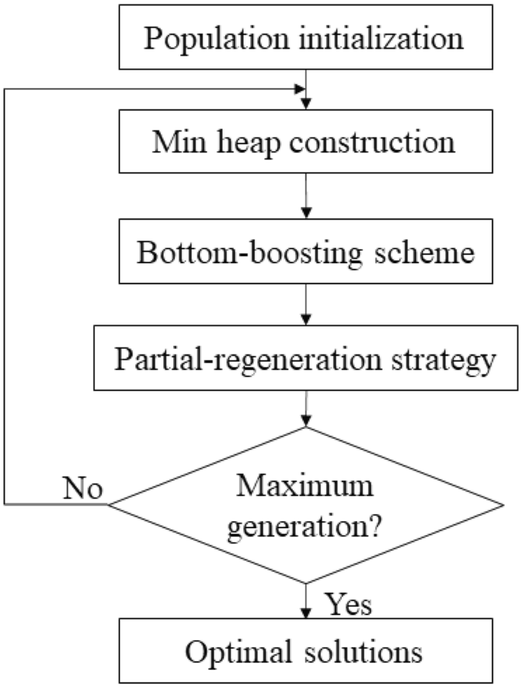

Fig. 2 Flowchart of minimax differential evolution

Step 3. *Bottom-boosting scheme* It is used to update the scenarios related to nodes based on the created min-heap. It also controls the evaluations of objective functions. Using min-heap, the root nodes are extracted continuously that generate a population in which all individuals are sorted in non-descending order according to their objective values. Like classical differential evolution, a trial scenario is formed using “DE/rand/1” mutation and binomial recombination. Thereafter, the trail scenario is tested by the original solution and calculate the new objective value. If it provides better objective value then the scenario of an original individual will be updated, otherwise the previous scenario will be resumed. Min heap will also be updated with the current population while updating the scenarios.

Step 4. *Partial-regeneration strategy* This step is used to update and control the total number of individuals using “DE/rand/1” mutation and binomial recombination (as discussed in Step 3). Firstly, mutant solutions are generated using DE/rand/1 mutation process. Thereafter, binomial recombination is performed to generate the trial solution using original and its mutant solutions. Then, the objective value of the trial solution is tested against the original solution and will select the solution for the next generation which has better the objective value.

Step 5. *Termination criterion* The process of optimization is terminated when the maximum number of generations is achieved.

4 Proposed approach

4.1 Motivation

Parameter estimation of chaotic maps is a very tedious task. The hyperchaotic maps consist of a large number of initial states and control parameters. The selection of these parameters is always a computationally extensive task. Because these must be selected either manually or hit and trial basis. Besides, the selection of initial parameters directly influences the strength of an encryption algorithm. As the chaotic maps are mainly used in encryption to generate the secret keys. Therefore, it is necessary to choose the initial parameters of chaotic maps optimally. Meta-heuristics approaches can be used to achieve this objective. Therefore, in this work, a 7D hyperchaotic map is used to produce the secret keys to perform encryption of images. To obtain the optimized encrypted image, the minimax differential evolution is applied. Multi-objective fitness function is designed to test the fitness of generated parameters by minimax differential

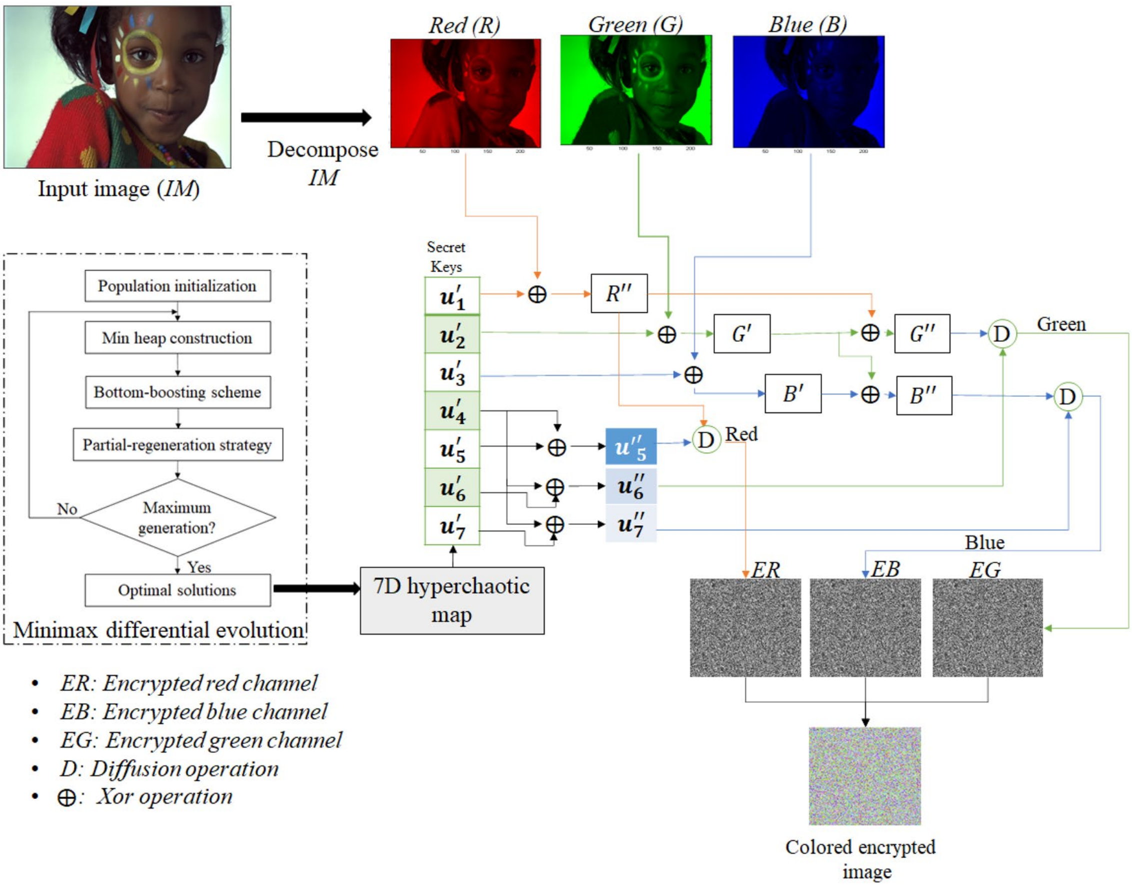

Fig. 3 Flowchart of proposed image encryption approach

evolution. The diffusion operation is utilized to encrypt the images.

4.2 Proposed image encryption approach

The working of the proposed image encryption approach is described in the Fig. 3. Initially, the input image (IM) is divided into three channels, i.e., red (R), green (G), and blue (B). Thereafter, minimax differential evolution is used to generate the optimal parameters for 7D hyperchaotic map (see Algorithm 2). These parameters are then utilized by hyperchaotic map to generate the secret keys such as u'_1, u'_2, u'_3, u'_4, u'_5, u'_6 and u'_7 (see Algorithm 5). Among these, u'_1, u'_2, and u'_3 are used to change the pixel values of R, G, and B using xor (⊕) operation, respectively. The resultant channels are R″, G′, and B′. To increase the confusion, G′ and B′ are again xor-ed with R″, and channels become G″ and B″. The secret keys such as u'_5, u'_6 and u'_7 are updated using u'_4. Then, updated keys u''_5, u''_6 and u''_7 are used to diffuse the R″, G″, and B″ channels with ω, respectively. This step generates the encrypted channels such as ER, EG, and EB. At last, the concatenation of ER, EG, and EB is done to generate the final encrypted image (EM). The whole process of encryption is described step by step in Algorithm 1.

```
Algorithm 1: The proposed image encryption algorithm
Input: Input image $IM$
Output: Encrypted image $EM$
/* Divide $IM$ into red (R), green (G), and blue (B) channels.*/
$R = IM(:, :, 1);$
$G = IM(:, :, 2);$
$B = IM(:, :, 3);$
obtain optimal parameters $(Y_j(j = 1 : Sol_D))$ by using Algorithm 2;
generate secret keys $u'_i$, where $i = 1, \ldots, 7$ using Algorithm 5;
// Diffuse the color channels using $u'_1$, $u'_2$, and $u'_3$
$R''$= mod($R \oplus u'_1$, 256);
$G$’ = mod($G \oplus u'_2$, 256);
$G''$ = mod($G' \oplus R$”, 256);
$B'$ = mod($B \oplus u'_3$, 256);
$B''$ = mod($B' \oplus G$’, 256);
//Update the secret keys $u'_5$, $u'_6$, and $u'_7$ using $u'_4$
$u_5$” = $u'_5 \oplus u'_4$;
$u_6$” = $u'_6 \oplus u'_4$;
$u_7$” = $u'_7 \oplus u'4$;
//Encrypt the $R$”, $G$”, and $B$” using $u''_5$, $u_6$”, and $u_7$”
$\omega = a_{sN}$;
$ER$ = mod($u_5$”× $R$” + $(1-\omega)$× $u_5$”, 256);
$EG$ = mod($u_6$”× $G$” + $(1-\omega)$× $u_6$”, 256);
$EB$ = mod($u_7$”× $B$” + $(1-\omega)$× $u_7$”, 256);
//Final encrypted image is generated by combining the encrypted channels
$EM$ = cat($ER$, $EG$, $EB$);
return $EM$;
```

4.2.1 Optimized key generation using minimax differential evolution

7D hyperchaotic map requires seven state parameters and ten other control and constant parameters for key generation. These parameters are optimally provided by minimax differential evolution to hyperchaotic map for better key generation. The process of optimal key generation using minimax differential evolution is illustrated in Algorithm 2.

Algorithm 2: Optimized key generation using minimax differential evolution

```
Input: Population size $N_s$, maximum generation $M_g$, scenario dimension $S_d$, solution dimension $Sol_d$
Output: Optimal population
Set generation $a = 0$ ;
Initialize the population randomly such as $P_a = \{(Y_{1,a}, C_{1,a}), (Y_{2,a}, C_{2,a}), \ldots, (Y_{N_s,a}, C_{N_s,a})\}$, where
 $Y_{j,a} = \{y^1_{j,a}, y^2_{j,a}, \ldots, y^{Sol_D}_{j,a}\}$ and $C_{j,a} = \{c^1_{j,a}, c^2_{j,a}, \ldots, c^{S_d}_{j,a}\}$, $j = 1, 2, \ldots, N_s$;
for $b = 1$ to $N_s$ do
    Calculate the finess of individual $(Y_{j,a}, C_{j,a})$ using Algorithm 6;
end
for $a = 1$ to $M_g$ do
    Generate a min heap by considering every individaual in $P_a$ as a node and their corresponding fitness value as
     a key;
    Apply bottom-boosting strategy on obtained min heap (see Algorithm 3);
    for $j = 1$ to $N_s$ do
        //Individual $(Y_r, C_r)$ stored as a root node in the min heap
        $Y_{j,a} = Y_r$, $C_{j,a} = C_r$;
        Remove root node from min heap;
    end
    $Y_{best} = Y_{1,a}$;
    Apply partial-regeneration scheme for updation of $P_a$ (see Algorithm 4);
    $P_{a+1} = P_a$;
end
```

Initially, the required parameters such as population size N_s, maximum generation M_g, scenario dimension S_d, solution dimension Sol_d are set. The initial population P_a is randomly generated based on the normal distribution. In this, every individual $(Y_{j,a}, C_{j,a})$ is denoted as a pair of solution and scenario. Then, calculate the fitness value of each individual using Algorithm 6. From the current population P_a, a min-heap is constructed, in which each individual acts as a node and its corresponding fitness value as a key of node. Thereafter, the bottom boosting strategy is applied to the constructed min-heap to update the scenarios (see Algorithm 3). Afterward, the root node of the min-heap is removed to update the current population. This will sort the individuals of the population in nondescending order based on their fitness value. The first individual of the updated population is selected as the best solution so far. To further update the current population, a partial-regeneration scheme is applied (see Algorithm 4). The next generation will then continue by the algorithm. The algorithm continues the process of optimization until the maximum generation is achieved.

```
Algorithm 3: Bottom boosting strategy
Input: Total number of fitness function evaluations $F_v$, a min heap generated from $P_a$, scaling factor $F$ and
       crossover rate $Cr$
Output: Updated min heap
set $ff = 0$;
while $ff < F_v$ do
    // Assume that individual stored in the root node is $(Y_{j,a}, C_{j,a})$
    Read the root node of the min heap;
    Produce mutant scenario as
    $CV_{j,a} = C_{r_1,a} + F(C_{r_2,a}–C_{r_3}, a)$; //where $r_1$, $r_2$, and $r_3$ are randomly selected indices from $\{1, 2, \ldots, N_s\}$
    Obtain the trial scenario $CU_{j,a}$ by applying binomial recombination on $C_{j,a}$ and $CV_{j,a}$;
    if $f(Y_{j,a}, CU_{j,a}) > f(Y_{j,a}, C_{j,a})$ then
        Remove the root node from the min heap;
        $C_{j,a} = CU_{j,a}$;
        Insert the updated $(Y_{j,a}, C_{j,a})$ into the min heap;
    end
    $ff = ff + 1$;
end
```

```
Algorithm 4: Partial-regeneration scheme
Input: Number of regenerated individuals $\beta$, sorted population $P_a$ w.r.t their fitness values in nondescending
       order, crossover rate $Cr$, and scaling factor $F$
Output: Updated population $P_a$
$b = 1$;
while $b \leq \beta$ do
    $W_{j,a} = Y_{j,a} + F(Y_{r_1,a} - Y_{r_2}, a)$; // Here, index $j$ is selected from $\{1, 2, \ldots, \beta\}$ randomly
    Obtain the trail solution $\varphi_{j,a}$ using binomial recombination on $Y_{j,a}$ and $W_{j,a}$;
    $Y_{N_s+1-b,a} = \varphi_{j,a}$;
    Reinitialize $C_{N_s+1-b,a}$ randomly;
    Calculate the fitness of updated individual $(Y_{N_s+1-b,a}, Y_{N_s+1-b,a})$ using Algorithm 6;
    $b = b + 1$;
end
```

4.2.1.1 Key generation

Algorithm 5 illustrates the key generation process using a 7D hyperchaotic map. Algorithm 5 utilizes the optimal solutions generated by the Algorithm 2. The seven secret keys are generated by Algorithm 5. These keys are further used by Algorithm 1 to perform diffusion on the input image to produce an encrypted image.

```
Algorithm 5: Key generation using 7D hyperchaotic map
  Input: Optimal solution $Y_j (j = 1 : Sol_D)$
  Output: Secret keys $u'_1$, $u'_2$, $u'_3$, $u'_4$, $u'_5$, $u'_6$ and $u'_7$
  //In solution $Y_j (j = 1 : Sol_D - 1)$, $Sol_D = 18$.
  // Here, $Y_j (j = 1 : Sol_D - 1)$ represents $u_1$, $u_2$, $u_3$, $u_4$, $u_5$, $u_6$, $u_7$, $s$, $n$, $i$, $t$, $r$, $q_1$, $p$, $e$, $q_2$, and $g$, respectively
  $u'_1 = y_8(y_2 - y_1) + y_4 + y_{12}y_6;$
  $u'_2 = y_{14}y_1 - y_2 - y_1y_3 + y_5;$
  $u'_3 = -y_{11}y_3 + y_1y_2;$
  $u'_4 = y_{15}y_4 - y_1y_3;$
  $u'_5 = -y_{10}y_2 + y_6;$
  $u'_6 = y_{13}y_1 + y_{16}y_2;$
  $u'_7 = y_{17}y_7 + y_9y_4;$
  return $u'_1$, $u'_2$, $u'_3$, $u'_4$, $u'_5$, $u'_6$ and $u'_7$;
```

4.2.1.2 Fitness function To evaluate the fitness of the individuals of the minimax differential evolution, the following fitness function is used.

$$f(t) = \begin{cases} \text{Minimize correlation} \\ \text{Maximize entropy} \end{cases} \tag{2}$$

subject to

$$\begin{aligned} &-0.05 < \text{correlation} < 0.05 \\ &7.9990 < \text{entropy}. \end{aligned} \tag{3}$$

In image encryption, minimum correlation and maximum entropy of the encrypted images are desirable properties. The minimum correlation hides the relationship among the adjacent pixels. While the maximum entropy makes each pixel appear as carrying the same amount of information. It helps in hiding the statistical properties of the pixels. Correlation of an encrypted image should be near to 0 and entropy should be near to 8 for an 8-bit image. Algorithm 6 describes the fitness evaluation of parameters generated by Algorithm 2.

```
Algorithm 6: Fitness evaluation
  Input: Input image $IM$, solution $Y_j (j = 1 : Sol_D)$
  Output: Fitness value $f$
  Use Algorithm 1 to generate an encryptem image $EM$ ;
  $E_n$ = entropy($EM$);
  $CC$ = correlation($EM$);
  if $-0.05 \leq CC \leq 0.05$ && $E_n > 7.9990$ then
      $f(\varphi) = \frac{E_n}{8}$ -$CC$;
  end
  return $f(\varphi)$;
```

4.3 Image decryption algorithm

Algorithm 7 describes the decryption process to recover the original image. At receiver side, the initial values of u_1, u_2, u_3, u_4, u_5, u_6, u_7, s, n, i, t, r, q_1, p, e, q_2, g, and ω parameters are required to decrypt the image. The process of decryption is same as encryption but in reverse order.

```
Algorithm 7: Decryption approach
Input: Inital values $u_1, u_2, u_3, u_4, u_5, u_6, u_7, s, n, i, t, r, q_1, p, e, q_2, g, \omega$, and encrypted image $(EM)$
Output: Decrypted image $DM$
// Divided $EM$ into red $ER$, green $EG$, and blue $EB$ channels
$ER = EM(:, :, 1)$;
$EG = EM(:, :, 2)$;
$EB = EM(:, :, 3)$;
produce secret keys $u'_1, u'_2, u'_3, u'_4, u'_5, u'_6$ and $u'_7$ by applying initial values to Algorithm 5;
// update secret keys $u'_5, u'_6$ and $u'_7$ using $u'_4$
$u''_5 = u'_5 \oplus u'_4$;
$u''_6 = u'_6 \oplus u'_4$;
$u''_7 = u'_7 \oplus u'_4$;
//Decrypt $ER$, $EG$, and $EB$ using $u''_5$, $u''_6$, and $u''_7$
$R'' = (ER$ - (1-$\omega$) $\times$ $u''_5)/\omega$;
$G'' = (EG$ - (1-$\omega$) $\times$ $u''_6)/\omega$;
$B'' = (EB$ - (1-$\omega$) $\times$ $u''_7)/\omega$;
//Decrypt $R''$, $G''$, and $B''$ using $u'_1$, $u'_2$, and $u'_3$
$R = R'' \oplus u'_1$;
$G' = R \oplus G''$;
$G = G' \oplus u'_2$;
$B' = B'' \oplus G''$;
$B = B' \oplus u'_3$;
//Generate final decrypted image by combing decrypted channels
$DM$ = cat($R, G, B$);
return $DM$ ;
```

5 Performance analyses

The proposed image encryption approach is implemented using MATLAB 2017a. Various tests have been conducted on the proposed approach to check its performance against security attacks. The results of the proposed approach are also compared with existing meta-heuristic based image encryption approaches to show the improvements. In this experiment, the control parameters for minimax differential are taken as: $F = 0.7$, $C_r = 0.5$, $N = 100$, $K_S = 190$, and $T = 10$.

For comparative analyses, we have considered four color images such as Girl, Monument, Lighthouse, and Macaws. We have resized these images to 512×512.

For comparative analyses five competitive meta-heuristic based image encryption approaches are considered such as GA [55], ACO [56], WDICA [57], HGA [58] and DHS [59]. We have utilized entropy [60], peak signal to noise ratio [61], and mean absolute error [62] to compute the performance of the proposed approach.

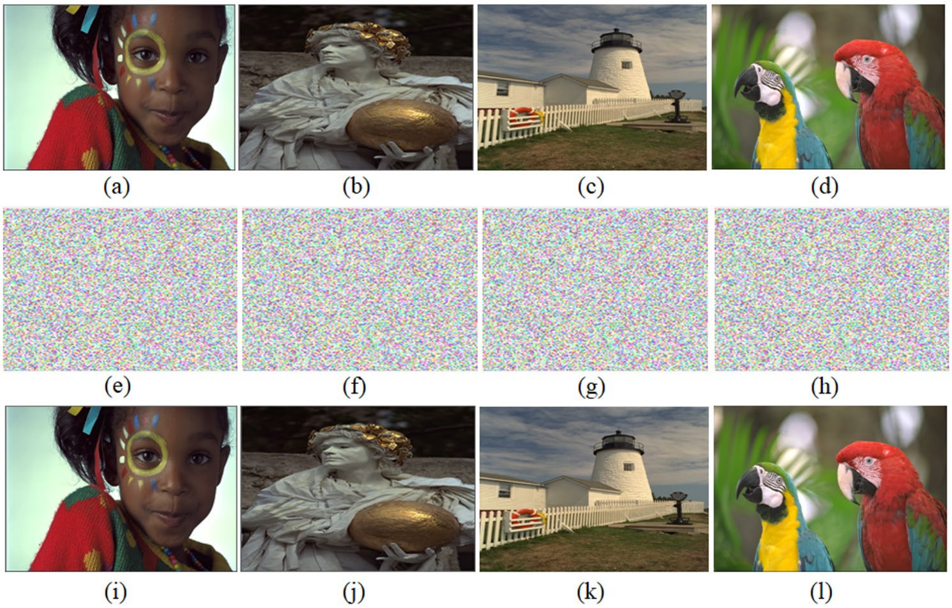

Fig. 4 Input images: **a** girl, **b** monument, **c** lighthouse, and **d** macaws. Encrypted images: **e** girl, **f** monument, **g** lighthouse, and **h** macaws. Decrypted images: **i** girl, **j** monument, **k** lighthouse, and **l** macaws

Table 1 Entropy analyses between the proposed and the competitive image encryption approaches for encrypted images

Method	Girl	Monument	Lighthouse	Macaws
GA [55]	7.9879	7.9921	7.9881	7.9880
ACO [56]	7.9891	7.9956	7.9918	7.9962
WDICA [57]	7.9837	7.9867	7.9914	7.9926
HGA [58]	7.9883	7.9963	7.9926	7.9927
DHS [59]	7.9962	7.9965	7.9971	7.9969
Proposed approach	7.9979	7.9978	7.9987	7.9982

Table 2 Peak signal-to-noise ratio analyses

Method	Girl	Monument	Lighthouse	Macaws
GA [55]	71.3727	71.7372	69.9364	69.2949
ACO [56]	69.7721	70.3942	68.9382	70.9271
WDICA [57]	69.4028	69.1843	70.7362	71.9328
HGA [58]	70.6047	71.2847	69.9272	69.2842
DHS [59]	69.9372	69.8274	70.9818	71.2891
Proposed approach	79.8281	81.1278	79.8272	78.2783

5.1 Visual analyses

Figure 4 illustrates the encrypted images computed using the proposed approach. It is found that the encrypted images are completely random containing no information of input images. Also, the decrypted images are identical to the input images.

5.2 Quantitative analyses

5.2.1 Entropy

Entropy [63] evaluates the randomness of the input images. For encrypted images it approaches toward 8. Table 1 demonstrates the entropy analyses between the proposed and the competitive approaches. It is observed that the entropy values obtained from the proposed approach are comparatively better as compared to the competitive approaches.

5.2.2 Peak signal to noise ratio

Peak signal-to-noise ratio (PSNR) [64] is another measure used in the field of image encryption. It is computed between the actual and decrypted images. It is desirable to be maximum. Table 2 illustrates the computed PSNR values from the proposed and competitive approaches. It shows that the proposed approach achieves remarkably more PSNR values as compared to the existing approaches.

5.2.3 Mean absolute error

Mean absolute error (MAE) [65] is utilized to evaluate the actual pixels difference between the encrypted and actual images. It is desirable to be maximum as we require a completely isolated encrypted image as compared to the input image. Table 3 demonstrates MAE analyses among the proposed and competitive approaches. It indicates that the proposed approach achieves comparatively higher values in terms of MAE as compared to the competitive approaches.

Table 3 Mean absolute error analyses

Method	Girl	Monument	Lighthouse	Macaws
GA [55]	79.3827	81.7362	79.3728	82.8372
ACO [56]	83.8142	82.1920	84.3947	81.2721
WDICA [57]	81.8627	83.9283	81.8721	82.3927
HGA [58]	83.7282	81.1292	84.7263	84.0812
DHS [59]	83.6721	85.6162	87.7982	86.8927
Proposed approach	86.2847	89.8742	91.2405	89.6820

5.3 Security analysis

Encrypted images are desirable to secure against the various security attacks and analyses. Therefore, in this section, some well-known security analyses are considered to evaluate the performance of the proposed approach.

5.3.1 Statistical attack

To implement statistical attacks, we have utilized correlation coefficient and histogram analyses.

Histogram analyses Histogram analyses provide details about the distribution of the pixels of an image [66]. From the related work, we have found that the histograms of input images are non-uniformly distributed whereas the histograms of encrypted images are evenly distributed. It means attackers are unable to obtain any kind of statistical information from encrypted images. Figure 5b, d demonstrates the evaluated histograms of input and encrypted Girl image, respectively. Figure 6b, d demonstrates the computed histograms of input and encrypted Monument image, respectively. Figure 7b, d demonstrates the histograms of input and encrypted Lighthouse image, respectively. Figure 8b, d demonstrates the histograms of input and encrypted Macaws image, respectively. It is clearly shown that the encrypted images have evenly distributed (see Figs. 5d, 6, 7 and 8d).

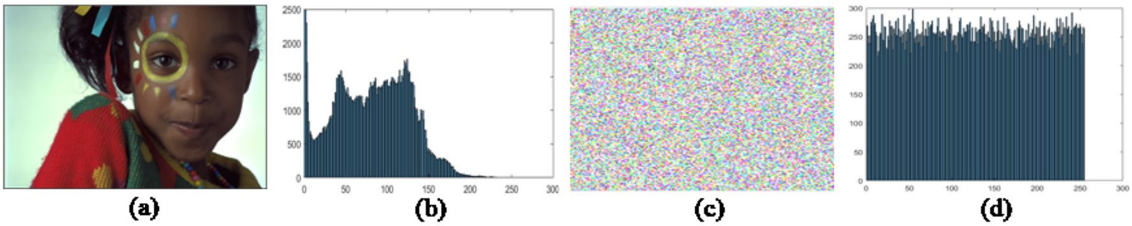

Fig. 5 Girl image: **a** input image, **b** histogram of input image, **c** encrypted image, and **d** histogram of encrypted image

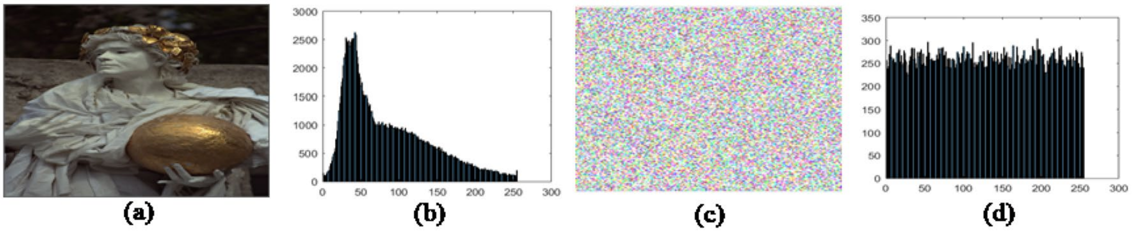

Fig. 6 Monument image: **a** input image, **b** histogram of input image, **c** encrypted image, and **d** histogram of encrypted image

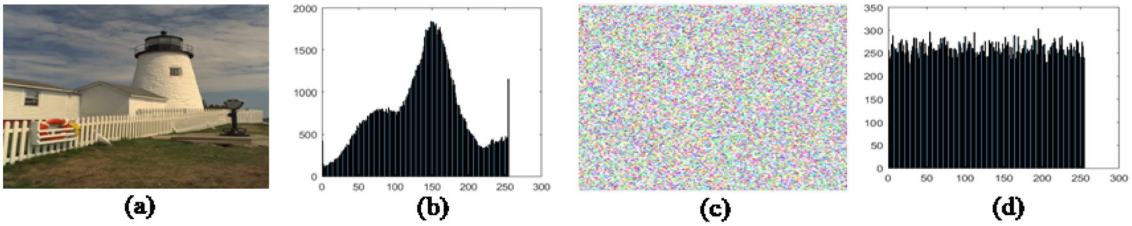

Fig. 7 Lighthouse image: **a** input image, **b** histogram of input image, **c** encrypted image, and **d** histogram of encrypted image

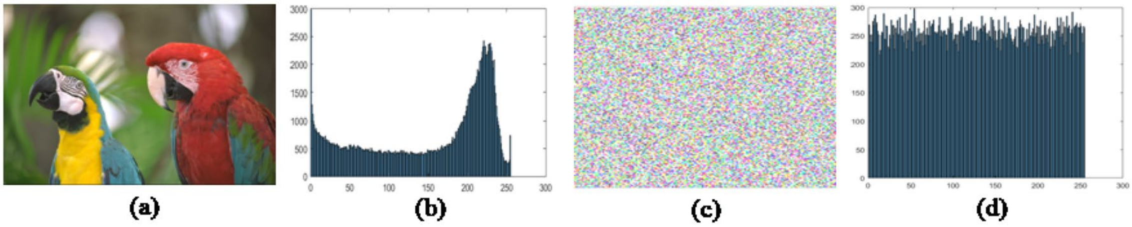

Fig. 8 Macaws image: **a** input image, **b** histogram of input image, **c** encrypted image, and **d** histogram of encrypted image

Table 4 Horizontal correlation analyses

Method	Girl	Monument	Lighthouse	Macaws
GA [55]	0.0029	0.0031	0.0039	−0.0021
ACO [56]	0.0092	0.0061	0.0043	0.0026
WDICA [57]	0.0051	0.0128	0.0091	−0.0041
HGA [58]	0.0042	0.0072	0.0092	−0.0038
DHS [59]	0.0073	0.0041	0.0052	−0.0058
Proposed	−0.0012	0.0017	0.0021	0.0024

Table 5 Vertical correlation analyses

Method	Girl	Monument	Lighthouse	Macaws
GA [55]	0.0039	0.0032	0.0029	0.0018
ACO [56]	0.0069	0.0071	0.0028	0.0046
WDICA [57]	0.0029	0.0018	0.0024	0.0018
HGA [58]	0.0031	0.0018	0.0029	0.0016
DHS [59]	0.0018	0.0022	0.0025	0.0027
Proposed	0.0017	−0.0012	0.0011	0.0007

5.3.1.1 Correlation coefficient analyses An attacker can utilize the correlation to evaluate the statistical properties of an input image [67]. Thus, the encrypted images should have a minimum correlation among the adjacent pixels horizontally, vertically, and diagonally [62]. Table 4 demonstrates the horizontal correlation analyses. It is found that the correlation among horizontal pixels is minimum as compared to the competitive approaches. Tables 5 and 6 demonstrate vertical and diagonal correlation comparative analyses. It is found that the proposed approach provides minimum correlation among vertical and diagonal pixels as compared to the competitive approaches. Therefore, the attackers are unable to compute any kind of statistical information about the input images from the encrypted images.

The horizontal, diagonal, and vertical correlation analyses are shown in Fig. 9a, c and e of input monument image. The horizontal, diagonal, and vertical correlation analyses are demonstrated in Fig. 9b, d, and e of encrypted monument image.

Table 6 Diagonal correlation analyses

Method	Girl	Monument	Lighthouse	Macaws
GA [55]	0.0109	0.0074	0.0061	0.0029
ACO [56]	0.0048	0.0039	0.0061	0.0034
WDICA [57]	0.0062	0.0047	0.0031	0.0028
HGA [58]	0.0046	0.0036	0.0058	0.0041
DHS [59]	0.0048	0.0056	0.0041	0.0063
Proposed	0.0021	0.0019	0.0021	0.0017

5.3.2 Differential analysis

The differential analysis (i.e., Number of Pixel Change Rates (NPCR) [68] and Unified Average Change Intensity (UACI) [69]) are utilized to compute the sensitivity of the proposed approach towards input images. Tables 7 and 8 demonstrate the comparative analyses in terms of NPCR and UACI, respectively. From tables, it is observed that the proposed approach provides significantly more NPCR and UACI values, respectively, as compared to the existing approaches. Therefore, the proposed approach is sensitive towards the small change in input images.

5.3.3 Secret key analysis

The key space and key sensitivity are two well-known measures to evaluate the secret key to resist brute-force attack [70].

Secret key space The proposed approach generates the secret keys using 7D hyperchaotic map. It needs seven state variables (such as u_1, u_2, u_3, u_4, u_5, u_6, and u_7) and ten control parameters (such as s, n, i, t, r, q_1, p, e, q_2, and g). If the precision of initial parameters is considered to be 10^{-16}, it means every parameter has 10^{16} different values. Therefore, the keyspace (KS) of the proposed approach is calculated as 10^{255}.

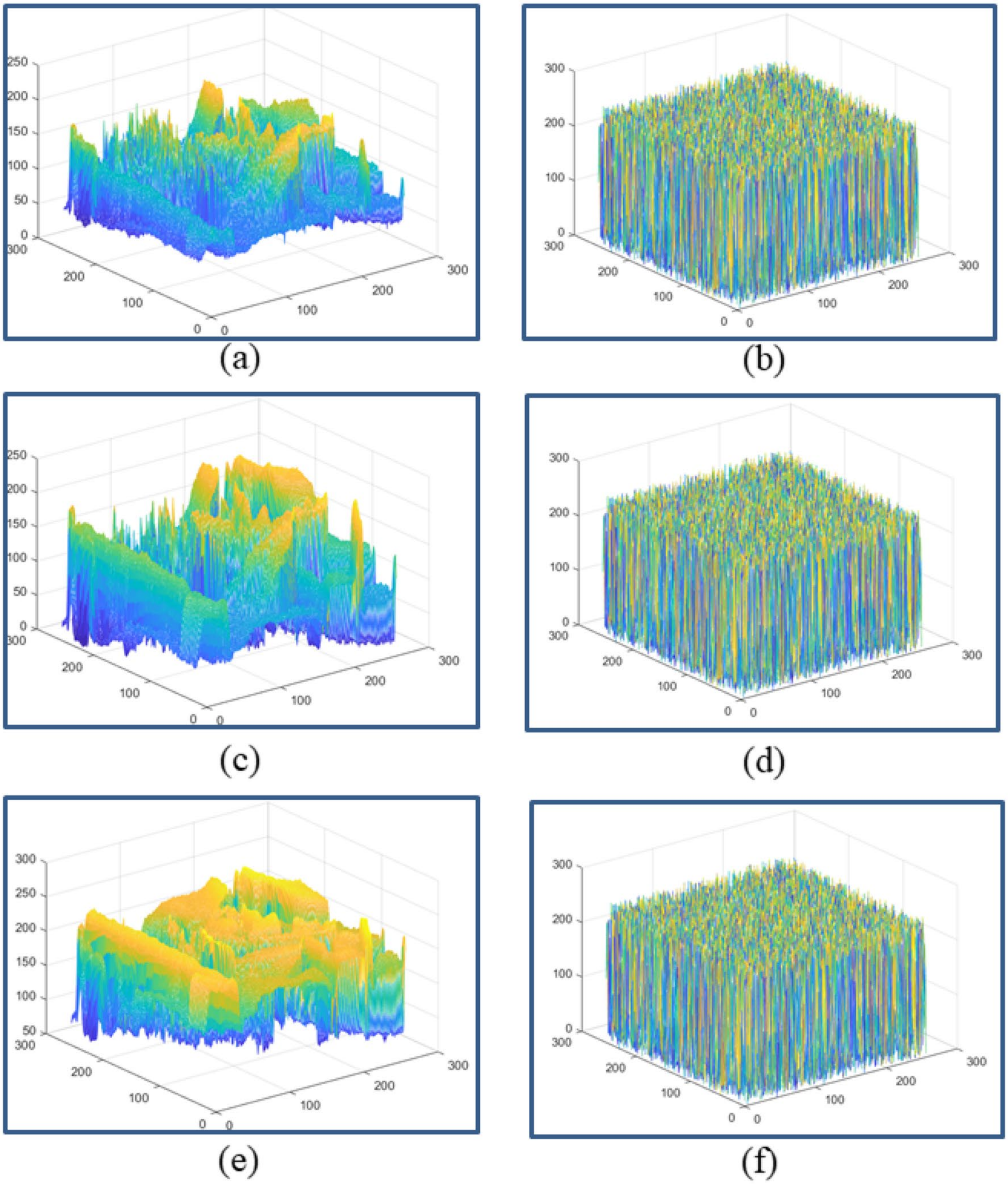

Fig. 9 Correlation analyses of input and encrypted Monument images: Input images correlation **a** horizontal, **b** diagonal, **c** vertical. Correlation of encrypted images: **d** horizontal, **e** diagonal, and **f** vertical

$$\begin{aligned} KS &= u_1 \times u_2 \times u_3 \times u_4 \times u_5 \times u_6 \times u_7 \times s \times n \times i \times t \times r \times q_1 \times p \times e \times q_2 \times g \\ &= 10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \\ &\approx 10^{255} \end{aligned} \tag{4}$$

Table 7 Number of pixel change rates analyses

Method	Girl	Monument	Lighthouse	Macaws
GA [55]	0.9939	0.9951	0.9950	0.9949
ACO [56]	0.9941	0.9943	0.9948	0.9951
WDICA [57]	0.9948	0.9949	0.9951	0.9949
HGA [58]	0.9943	0.9950	0.9946	0.9948
DHS [59]	0.9957	0.9958	0.9959	0.9948
Proposed	0.9963	0.9969	0.9968	0.9971

Table 8 Unified average change intensity analyses

Method	Girl	Monument	Lighthouse	Macaws
GA [55]	0.3323	0.3331	0.3337	0.3329
ACO [56]	0.3337	0.3338	0.3335	0.3334
WDICA [57]	0.3328	0.3335	0.3338	0.3337
HGA [58]	0.3339	0.3343	0.3339	0.3347
DHS [59]	0.3338	0.3349	0.3348	0.3349
Proposed	0.3354	0.3357	0.3361	0.3360

If keyspace is more than $10^{30}(\approx 2^{100})$, then the secret key cannot be easily discovered by attacker through brute-force attack. The keyspace of the proposed approach is 10^{255}. Hence, the proposed approach can easily resist the brute-force attack.

5.3.3.1 Sensitivity of secret key The secret keys of the proposed approach should be sensitive towards small changes. The seven keys are produced to perform encryption in the proposed approach. These keys are generated by Eq. (1) using the initial state (u_1, u_2, u_3, u_4, u_5, u_6, and u_7) and control (s, n, i, t, r, q_1, p, e, q_2, and g) parameters. Suppose that we have slightly change some bits in u_1. To test the sensitivity, keys (such as u'_1, u'_2, u'_3, u'_4, u'_5, u'_6, and u'_7) are generated using original parameters and keys (such as u''_1, u''_2, u''_3, u''_4, u''_5, u''_6, and u''_7) are produced from changed parameters. Figure 10b, c show obtained encrypted images with actual keys and changed keys, respectively. Figure 10d presents the difference of Fig. 10b, c that is a completely random image. Thus, the keys are not identical to each other. Similarly, we have tried to decrypt Fig. 10b using both keys and obtained results are shown in Fig. 10e, f, respectively. If we decrypt the image with the same keys then the evaluated image is

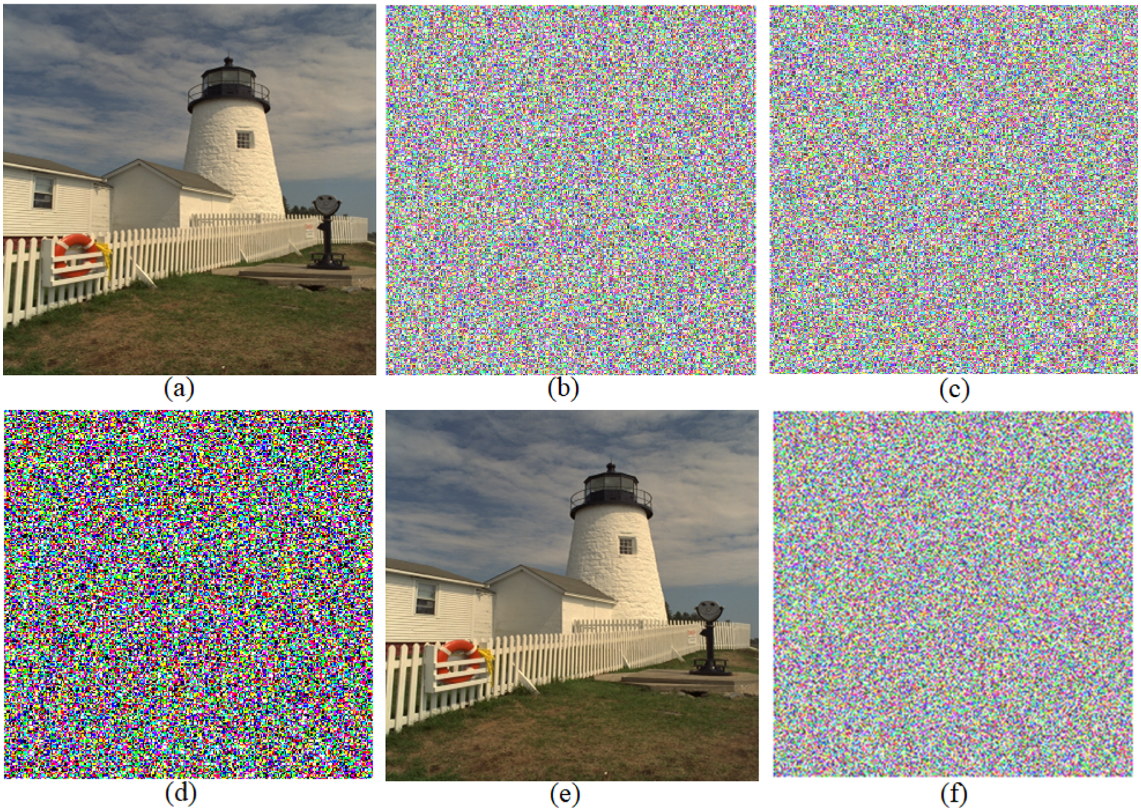

Fig. 10 Sensitivity of secret key : **a** lighthouse input image, **b** encrypted image with actual secret keys, **c** encrypted image using same key with small changes, **d** difference between (**b**, **c**), **e** decrypted image with actual keys, and **f** decrypted image with modified keys

Table 9 Difference between encrypted images generated from actual and modified keys

	Girl	Monument	Lighthouse	Macaws
Difference	99.9965	99.9958	99.9969	99.9971

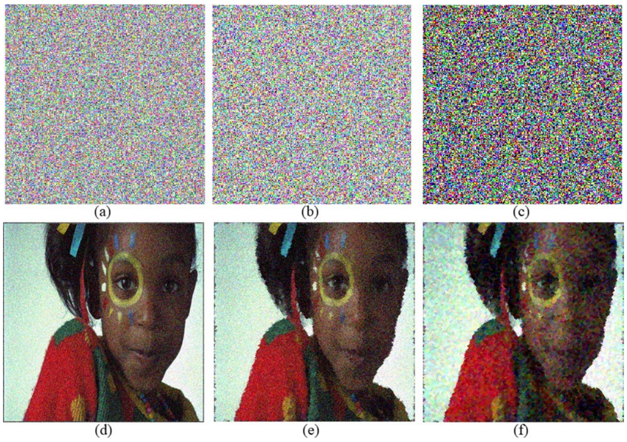

Fig. 11 Guassian noise attack analyses: Attacked encrypted images: **a** when $\mu = 0.2$ and $\sigma^2 = 0.2$, **b** when $\mu = 0.6$ and $\sigma^2 = 0.2$, **c** when $\mu = 0.6$ and $\sigma^2 = 0.6$. Attacked decrypted images: **d** obtained from (**a**, **e**) obtained from (**b**, **f**) obtained from (**c**)

identical to the input image. However, with the changed keys, the decrypted image is again a random image. Thus, if there is a small change in the input keys, the encrypted image is impossible to be decrypted.

Table 9 demonstrates the difference between two encrypted images which are computed using a actual and modified keys. It proves that the proposed approach is significantly sensitive towards initial key values.

5.3.4 Robustness against attacks on encrypted images

Attacks on encrypted images sometimes cannot be prevented, so, it becomes more challenging to extract some information from the encrypted images, even if the attack is very strong.

5.3.4.1 Noise attack analyses To evaluate the robustness of the proposed approach, Gaussian noise and Salt & pepper noise attacks with different densities are applied on the encrypted images.

Figure 11 demonstrates attacked encrypted images using Gaussian noise with mean (μ) = 0.2 and variance (σ^2) = 0.2, $\mu = 0.6$ and $\sigma^2 = 0.2$, $\mu = 0.6$ and $\sigma^2 = 0.6$, respectively (see Fig. 11a–c). The decrypted images obtained using the proposed approach are shown in Fig. 11d–f, respectively. The respective PSNR values between input Girl and decrypted images are 26.76 dB, 22.41 dB and 11.42 dB, respectively. Also, the visual analyses indicate that the proposed approach can still able to preserve significant amount of the information from the attacked images

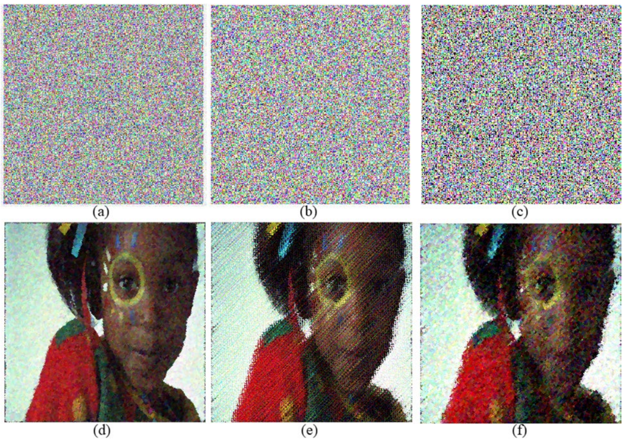

Fig. 12 Salt & pepper noise attack analyses: Attacked encrypted images **a** when density = 0.2), **b** when density =0.4, **c** with density = 0.8, Decrypted attacked images: obtained from (**a**, **e**) obtained from (**b**, **f**) obtained from (**c**)

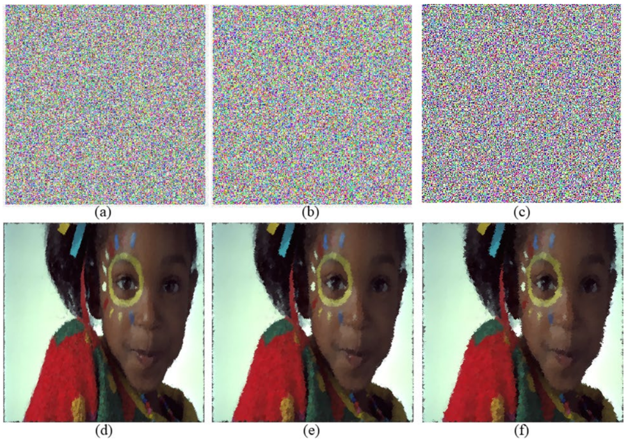

Fig. 13 Enhancement attack analyses Attacked encrypted images: **a** histogram equalization, **b** gamma correction, **c** adaptive histogram equalization. Attacked decrypted images: **d** obtained from (**a**, **e**) obtained from (**b**, **f**) obtained from (**c**)

Figure 12 demonstrates encrypted images after implementing the salt & pepper noise attack with various densities, i.e., 0.2, 0.4, and 0.8, respectively. Figure 12d–f show the decrypted images obtained from the respective noisy encrypted images as shown in Fig. 12a–c. The respective PSNR values between the input Girl and decrypted images are 31.04 dB, 22.42 dB, and 14.82 dB, respectively. Also, the visibility of extracted images reveal that the proposed approach can significantly decrypt images even from the noisy encrypted images.

Table 10 Encryption time analyses

Image name	Size	GA [55]	ACO [56]	WDICA [57]	GA [58]	DHS [59]	Proposed approach
Girl	256 × 256	11.3	15.4	12.4	12.6	11.1	10.3
Monument	512 × 512	58.8	60.8	56.8	55.6	58.6	39.4
Lighthouse	1024 × 1024	648.6	601.6	626.8	611.8	598.8	592.6
Macaws	2048 × 2048	2368.8	2658.8	2358.4	2261.2	1996.2	1872.1

Table 11 Decryption time analyses

Image	Size	GA [55]	ACO [56]	WDICA [57]	GA [58]	DHS [59]	Proposed approach
Girl	256 × 256	0.062	0.051	0.042	0.039	0.041	0.034
Monument	512 × 512	0.049	0.051	0.043	0.052	0.046	0.037
Lighthouse	1024 × 1024	0.061	0.051	0.046	0.051	0.045	0.037
Macaws	2048 × 2048	0.064	0.051	0.043	0.052	0.039	0.035

5.3.4.2 Enhancement attack analyses Image enhancement is also utilized by attackers to damage encrypted images. Thus, an image encryption approach should have the ability to resist enhancement attack. Figure 13a–c demonstrate attacked encrypted images by using histogram equalization, gamma correction, and adaptive histogram equalization, respectively. Figure 13d–f show respective decrypted images. The evaluated PSNR values between input Girl image and decrypted images are as 26.29 dB, 21.96 dB and 09.73 dB, respectively. Also from the visual analyses, it is found that the proposed approach can efficiently resist the enhancement attacks.

5.4 Execution time analyses

Execution time (*ET*) is evaluated in terms of seconds. 'tic' and 'toc' operators of MATLAB are utilized to compute *ET*.

Tables 10 and 11 demonstrate encryption and decryption time analyses, respectively. It is found that the proposed approach encrypt and decrypt images at significantly good computational speed, as it takes lesser time than competitive approaches.

6 Conclusion

A minimax differential evolution based 7D hyper-chaotic map was proposed in this paper to obtain an optimized encrypted image. A suitable diffusion model was implemented to encrypt the images by using the obtained optimized secret keys. The proposed image encryption approach is sensitive to input images and secret keys. Extensive experiments are performed by considering the benchmark color images. The extensive experimental results reveal that the proposed approach achieves significantly good encryption results as compared to the competitive approaches. Also, the proposed approach resists against various security attacks such as statistical, differential, noise, and image enhancement.

References

1. R. Vidhya, M. Brindha, A chaos based image encryption algorithm using rubik's cube and prime factorization process (cierpf). J. King Saud Univ. Comput. Inf. Sci. (2020). https://doi.org/10.1016/j.jksuci.2019.12.014
2. W. Liu, K. Sun, C. Zhu, A fast image encryption algorithm based on chaotic map. Opt. Lasers Eng. **84**, 26–36 (2016)
3. J.-X. Chen, Z.-L. Zhu, C. Fu, H. Yu, An improved permutation-diffusion type image cipher with a chaotic orbit perturbing mechanism. Opt. Express **21**(23), 27873–27890 (2013)
4. G.-D. Ye, X.-L. Huang, L.Y. Zhang, Z.-X. Wang, A self-cited pixel summation based image encryption algorithm. Chin. Phys. B **26**(1), 010501 (2017)
5. A. Kanso, M. Ghebleh, A novel image encryption algorithm based on a 3d chaotic map. Commun. Nonlinear Sci. Numer. Simul. **17**(7), 2943–2959 (2012)
6. N.R. Zhou, T.X. Hua, L.H. Gong, D.J. Pei, Q.H. Liao, Quantum image encryption based on generalized Arnold transform and double random-phase encoding. Quantum Inf. Process. **14**(4), 1193–1213 (2015)
7. G. Ye, K. Jiao, C. Pan, X. Huang, An effective framework for chaotic image encryption based on 3d logistic map. Secur. Commun. Netw. **2018**, 1–11 (2018)
8. Z. Azimi, S. Ahadpour, Color image encryption based on DNA encoding and pair coupled chaotic maps. Multimed. Tools Appl. **79**(3), 1727–1744 (2020)
9. D. Singh, M. Kaur, H. Singh, Remote sensing image fusion using fuzzy logic and gyrator transform. Remote Sens. Lett. **9**(10), 942–951 (2018)
10. G. Qi, H. Wang, M. Haner, C. Weng, S. Chen, Z. Zhu, Convolutional neural network based detection and judgement of

environmental obstacle in vehicle operation. CAAI Trans. Intell. Technol. **4**(2), 80–91 (2019)

11. D. Singh, V. Kumar, M. Kaur, Image dehazing using window-based integrated means filter. Multimed. Tools Appl. (2019). https://doi.org/10.1007/s11042-019-08286-6
12. Y. Tingting, W. Junqian, W. Lintai, X. Yong, Three-stage network for age estimation. CAAI Trans. Intell. Technol. **4**(2), 122–126 (2019)
13. M. Kaur, D. Singh, V. Kumar, K. Sun, Color image dehazing using gradient channel prior and guided l0 filter. Inf. Sci. **521**, 326–342 (2020). https://doi.org/10.1016/j.ins.2020.02.048, http://www.sciencedirect.com/science/article/pii/S0020025520301316. Accessed 23 Mar 2020
14. Y. Pathak, P.K. Shukla, A. Tiwari, S. Stalin, S. Singh, P.K. Shukla, Deep transfer learning based classification model for covid-19 disease. IRBM (2020). https://doi.org/10.1016/j.irbm.2020.05.003
15. M. Kaur, D. Singh, Fusion of medical images using deep belief networks. Clust. Comput. **23**, 1439–1453 (2020). https://doi.org/10.1007/s10586-019-02999-x
16. P.K. Shukla, P.K. Shukla, P. Sharma, P. Rawat, J. Samar, R. Moriwal, M. Kaur, Efficient prediction of drug-drug interaction using deep learning models. IET Syst. Biol. **14**(4), 211–216 (2020). https://doi.org/10.1049/iet-syb.2019.0116
17. D. Singh, V. Kumar, A comprehensive review of computational dehazing techniques. Arch. Comput. Methods Eng. **26**(5), 1395–1413 (2019)
18. D. Singh, V. Kumar, Dehazing of outdoor images using notch based integral guided filter. Multimed. Tools Appl. **77**(20), 27363–27386 (2018)
19. H.S. Basavegowda, G. Dagnew, Deep learning approach for microarray cancer data classification. CAAI Trans. Intell. Technol. **5**(1), 22–33 (2020)
20. D. Singh, V. Kumar, M. Kaur, Single image dehazing using gradient channel prior. Appl. Intell. **49**(12), 4276–4293 (2019)
21. A. Pisarchik, M. Zanin, Image encryption with chaotically coupled chaotic maps. Phys. D **237**(20), 2638–2648 (2008)
22. N.K. Pareek, V. Patidar, K.K. Sud, Image encryption using chaotic logistic map. Image Vis. Comput. **24**(9), 926–934 (2006)
23. H. Kwok, W.K. Tang, A fast image encryption system based on chaotic maps with finite precision representation. Chaos Solitons Fractals **32**(4), 1518–1529 (2007)
24. J. Fridrich, Symmetric ciphers based on two-dimensional chaotic maps. Int. J. Bifurc. Chaos **8**(06), 1259–1284 (1998)
25. G. Chen, Y. Mao, C.K. Chui, A symmetric image encryption scheme based on 3d chaotic cat maps. Chaos Solitons Fractals **21**(3), 749–761 (2004)
26. Z.-H. Guan, F. Huang, W. Guan, Chaos-based image encryption algorithm. Phys. Lett. A **346**(1–3), 153–157 (2005)
27. S. Lian, Efficient image or video encryption based on spatiotemporal chaos system. Chaos Solitons Fractals **40**(5), 2509–2519 (2009)
28. Q. Zhang, L. Guo, X. Wei, Image encryption using dna addition combining with chaotic maps. Math. Comput. Modell. **52**(11–12), 2028–2035 (2010)
29. Y. Mao, G. Chen, S. Lian, A novel fast image encryption scheme based on 3d chaotic baker maps. Int. J. Bifurc. Chaos **14**(10), 3613–3624 (2004)
30. X. Wang, D. Luan, A novel image encryption algorithm using chaos and reversible cellular automata. Commun. Nonlinear Sci. Numer. Simul. **18**(11), 3075–3085 (2013)
31. L. Liu, S. Miao, A new image encryption algorithm based on logistic chaotic map with varying parameter. SpringerPlus **5**(1), 289 (2016)
32. Y. Wu, J.P. Noonan, G. Yang, H. Jin, Image encryption using the two-dimensional logistic chaotic map. J. Electron. Imaging **21**(1), 013014 (2012)
33. G. Ye, X. Huang, A secure image encryption algorithm based on chaotic maps and sha-3. Secur. Commun. Netw. **9**(13), 2015–2023 (2016)
34. G. Ye, X. Huang, An efficient symmetric image encryption algorithm based on an intertwining logistic map. Neurocomputing **251**, 45–53 (2017)
35. G. Ye, C. Pan, X. Huang, Z. Zhao, J. He, A chaotic image encryption algorithm based on information entropy. Int. J. Bifurc. Chaos **28**(01), 1850010 (2018)
36. E. Solak, C. Çokal, O.T. Yildiz, T. BiyikoĞlu, Cryptanalysis of fridrich's chaotic image encryption. Int. J. Bifurc. Chaos **20**(05), 1405–1413 (2010)
37. E.Y. Xie, C. Li, S. Yu, J. Lü, On the cryptanalysis of fridrich's chaotic image encryption scheme. Sig. Process. **132**, 150–154 (2017)
38. Z. Hua, Y. Zhou, H. Huang, Cosine-transform-based chaotic system for image encryption. Inf. Sci. **480**, 403–419 (2019)
39. A. Benlashram, M. Al-Ghamdi, R. AlTalhi, P. K. Laabidi, A novel approach of image encryption using pixel shuffling and 3d chaotic map. In: Journal of Physics: Conference Series, Vol. 1447, IOP Publishing, 2020, p. 012009
40. J. Lin, K. Zhao, X. Cai, D. Li, Z. Wang, An image encryption method based on logistic chaotic mapping and dna coding. In: MIPPR 2019: remote sensing image processing, geographic information systems, and other applications, Vol. 11432, International Society for Optics and Photonics, 2020, p. 114321L
41. C. Li, F. Zhao, C. Liu, L. Lei, J. Zhang, A hyperchaotic color image encryption algorithm and security analysis. Secur. Commun. Netw. **2019**, 1–8 (2019)
42. J. Wu, J. Shi, T. Li, A novel image encryption approach based on a hyperchaotic system, pixel-level filtering with variable kernels, and dna-level diffusion. Entropy **22**(1), 5 (2020)
43. L. You, E. Yang, G. Wang, A novel parallel image encryption algorithm based on hybrid chaotic maps with opencl implementation. Soft Comput. **24**, 12413–12427 (2020)
44. R. Li, Fingerprint-related chaotic image encryption scheme based on blockchain framework. Multimed. Tools Appl. (2020). https://doi.org/10.1007/s11042-020-08802-z
45. M. Khan, F. Masood, A. Alghafis, M. Amin, S.I.B. Naqvi, A novel image encryption technique using hybrid method of discrete dynamical chaotic maps and brownian motion. PLoS One **14**(12), e0225031 (2019)
46. L. Liu, J. Hu, H. Li, J. Li, Z. He, C. Han, Parameter estimation of a class one-dimensional discrete chaotic system. Discr. Dyn. Nat. Soc. **2011**, 1–9 (2011)
47. C. Li, A new method of determining chaos-parameter-region for the tent map. Chaos Solitons Fractals **21**(4), 863–867 (2004)
48. Q. Yang, D. Zhu, L. Yang, A new 7d hyperchaotic system with five positive lyapunov exponents coined. Int. J. Bifurc. Chaos **28**(05), 1850057 (2018)
49. Q. Yang, W.M. Osman, C. Chen, A new 6d hyperchaotic system with four positive lyapunov exponents coined. Int. J. Bifurc. Chaos **25**(04), 1550060 (2015)
50. X. Qiu, J.-X. Xu, Y. Xu, K.C. Tan, A new differential evolution algorithm for minimax optimization in robust design. IEEE Trans. Cybern. **48**(5), 1355–1368 (2017)
51. D. Singh, V. Kumar, Vaishali, M. Kaur, Classification of covid-19 patients from chest ct images using multi-objective differential evolution–based convolutional neural networks. Eur. J. Clin. Microbiol. Infect. Dis. **39**, 1–11 (2020). https://doi.org/10.1007/s10096-020-03901-z
52. H.S. Pannu, D. Singh, A.K. Malhi, Multi-objective particle swarm optimization-based adaptive neuro-fuzzy inference system for benzene monitoring. Neural Comput. Appl. **31**, 2195–2205 (2019)

53. H.S. Pannu, D. Singh, A.K. Malhi, Improved particle swarm optimization based adaptive neuro-fuzzy inference system for benzene detection, CLEAN-Soil, Air. Water **46**(5), 1700162 (2018)
54. M. Kaur, H.K. Gianey, D. Singh, M. Sabharwal, Multi-objective differential evolution based random forest for e-health applications. Mod. Phys. Lett. B **33**(05), 1950022 (2019)
55. A.H. Abdullah, R. Enayatifar, M. Lee, A hybrid genetic algorithm and chaotic function model for image encryption. AEU-Int. J. Electron. Commun. **66**(10), 806–816 (2012)
56. N. Sreelaja, G.V. Pai, Stream cipher for binary image encryption using ant colony optimization based key generation. Appl. Soft Comput. **12**(9), 2879–2895 (2012)
57. R. Enayatifar, A.H. Abdullah, M. Lee, A weighted discrete imperialist competitive algorithm (wdica) combined with chaotic map for image encryption. Opt. Lasers Eng. **51**(9), 1066–1077 (2013)
58. R. Enayatifar, A.H. Abdullah, I.F. Isnin, Chaos-based image encryption using a hybrid genetic algorithm and a DNA sequence. Opt. Lasers Eng. **56**, 83–93 (2014)
59. K.M. Talarposhti, M.K. Jamei, A secure image encryption method based on dynamic harmony search (DHS) combined with chaotic map. Opt. Lasers Eng. **81**, 21–34 (2016)
60. M. Kaur, V. Kumar, An efficient image encryption method based on improved lorenz chaotic system. Electron. Lett. **54**(9), 562–564 (2018)
61. M. Kaur, V. Kumar, Fourier-mellin moment-based intertwining map for image encryption. Mod. Phys. Lett. B **32**(9), 1850115 (2018)
62. M. Kaur, V. Kumar, Color image encryption technique using differential evolution in nonsubsampled contourlet transform domain. IET Image Process. **12**(7), 1273–1283 (2018)
63. B. Mondal, T. Mandal, A light weight secure image encryption scheme based on chaos& dna computing. J. King Saud Univ. Comput. Inf. Sci. **29**, 499–504 (2016)
64. N. Rawat, B. Kim, R. Kumar, Fast digital image encryption based on compressive sensing using structurally random matrices and arnold transform technique. Optik Int. J. Light Electron Opt. **127**(4), 2282–2286 (2016) (Elsevier, Switzerland)
65. Y. Zhang, B. Xu, N. Zhou, A novel image compression-encryption hybrid algorithm based on the analysis sparse representation. Opt. Commun. **392**, 223–233 (2017)
66. T. Sivakumar, R. Venkatesan, A novel image encryption using calligraphy based scan method and random number. KSII Trans. Internet Inf. Syst. **9**(6), 2317–2337 (2015)
67. F.E.A. El-Samie, H.E.H. Ahmed, I.F. Elashry, M.H. Shahieen, O.S. Faragallah, E.-S.M. El-Rabaie, S.A. Alshebeili, *Image Encryption: a Communication Perspective* (CRC Press, Boca Roton, 2013)
68. A. Bakhshandeh, Z. Eslami, An authenticated image encryption scheme based on chaotic maps and memory cellular automata. Opt. Lasers Eng. **51**(6), 665–673 (2013)
69. A. Belazi, A.A.A. El-Latif, S. Belghith, A novel image encryption scheme based on substitution-permutation network and chaos. Sig. Process. **128**, 155–170 (2016)
70. M. Ghebleh, A. Kanso, H. Noura, An image encryption scheme based on irregularly decimated chaotic maps. Sig. Process. Image Commun. **29**(5), 618–627 (2014)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.