



SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: <https://www.svedbergopen.com/>



Research Paper

Open Access

Predictive Quality Management in Nepalese A Dual-Level Cuckoo Search Optimization Framework for DNA Sequence Steganography in RGBA Images

Abdulhadi O. Abdulhadi^{1*}, Hala M. Abdel-kader², Aya Hossam³

¹Electrical Eng. Depart, Shoubra Faculty of Engineering, Benha University, Cairo, Egypt. E-mail: a.omar12025@feng.bu.edu.eg

²Electrical Eng. Depart, Shoubra Faculty of Engineering, Benha University, Cairo, Egypt. E-mail: HALA.MANSOUR@feng.bu.edu.eg

³Electrical Eng. Depart, Shoubra Faculty of Engineering, Benha University, Cairo, Egypt. E-mail: aya.ahmed@feng.bu.edu.eg

Abstract

The growing use of genomic information has increased the need for effective and covert methods to protect sensitive DNA sequence data. Image steganography offers a practical way to conceal such information within digital images while keeping the presence of the hidden data visually imperceptible. In this paper, we propose an adaptive LSB-based steganographic framework for embedding raw DNA sequences into RGBA images. The DNA sequence is first converted into tri-mer indices, providing a compact 6-bit representation. A repetition-aware encoding scheme is then used to handle consecutive identical tri-mers, which reduces the amount of data that needs to be embedded. Depending on the repetition characteristics of the encoded DNA data, the embedding process switches between 3-3 and 2-2-2 bit allocation patterns. To further improve embedding efficiency and stego-image quality, a two-level Cuckoo Search (CS) optimization strategy is introduced. At the first level, CS permutes the tri-mer indices to increase consecutive repetitions, thereby reducing both the effective payload and the number of pixels required for embedding. At the second level, rotation-based optimization improves the match between the secret bitstream and the image LSBs. This reduces the number of bit changes and, consequently, limits embedding distortion. The proposed framework was evaluated using three 512 × 512 benchmark cover images—Lena, Baboon, and Penguin—under three embedding scenarios: direct embedding, repetition-aware embedding without optimization, and the proposed CS-optimized approach. Direct embedding achieved an average PSNR of 44.78 dB with an MSE of 2.16. Repetition-aware embedding increased the average PSNR to 44.90 dB while also reducing the number of pixels used. With CS optimization, the PSNR ranged from 49.07 to 49.12 dB, the MSE decreased to 0.79, and the SSIM reached 0.999932. These results show that the proposed method preserves high visual quality and strong structural similarity between the cover and stego images. In addition, the optimized framework required approximately 95.6% of the pixels used by direct embedding, corresponding to an overall reduction in pixel usage of about 4.38%. Overall, the proposed framework provides an effective approach for reducing embedding distortion and pixel consumption while maintaining high image quality for the concealment of genomic sequence data in RGBA images.

Keywords: DNA steganography; Cuckoo Search; genomic data protection; repetition-aware encoding; adaptive LSB embedding; data hiding; RGBA images.

1. Introduction

Over the years there has been an increase in the usage of digital communication technologies. This resulted in the increase in the volume and sensitivity of the data transmitted over public networks. The exponential growth of multimedia exchange, including images, videos, audio and text, has made protecting confidential information from unauthorized interception and tampering a crucial concern in the fields of security and multi-media engineering [1].

Raw genomic sequences and Short Tandem Repeat (STR) profiles contain sensitive biological information that provide insights on traits, ancestry and family relations. Unlike the data we store digitally, genomic data is unique, identifiable and permanent making the need for its privacy increasing. Genomic data can provide current and future susceptibility of specific diseases for an individual showing the need for advanced privacy preserving for genomic data sharing and storage [2].

Existing DNA steganography methods mainly focus on hiding conventional secret messages using DNA-based encoding or DNA sequences rather than protecting real DNA sequences. For example, a double-layer steganography framework was proposed where secret messages are encoded into DNA sequences and then hidden within digital images [3]. Therefore, limited studies were made towards the direct hiding of real genomic DNA data as sensitive information.

Recent studies show that conventional anonymization methods do not provide enough protection of genomic privacy. Individuals can be identified from pooled genomic datasets using Single Nucleotide Polymorphism (SNP) analysis [4] while anonymous genomic records can be connected to real identities using Y-chromosome STR analysis and public databases [5]. This confirms that raw DNA sequences and STR profiles are important

privacy-sensitive data.

DNA steganography studies focus on hiding secret messages using DNA-based encoding or DNA sequences rather than protecting the genomic DNA sequences themselves.

Several DNA steganography approaches have been proposed however, most existing methods suffer from a trade-off between embedding capacity, visual imperceptibility and computational efficiency. Additionally, conventional LSB-embedding techniques often introduce increased perceptual distortion when facing high payload levels or repetitive patterns. Optimization-driven steganography helps reduce these limitations. In this study, Cuckoo Search (CS) was used for its balance between local exploitation and global exploration. Through using CS at the payload level, the proposed method efficiently explores the permutation search space to maximize adjacent nucleotide repetitions thus reducing embedding overhead and visual distortion before spatial embedding.

All things considered, the proposed framework provides an extra level of privacy protection by enabling the secure embedding of both raw DNA sequences and tri-STR data within RGBA images using optimization-driven DNA steganography methods. The proposed framework enhances the confidentiality of genomic data and reduces the likelihood of disclosing sensitive data.

The main challenge in rearranging secret data before embedding can be formulated as an optimization problem to achieve maximum adjacent repetitions. The proposed approach is an image steganography method in the spatial domain. This approach tries to embed large genomic data with the least distortions through transforming the input DNA into trimer-based blocks. CS is then used to generate optimized permutation candidates that maximize adjacent repetitions within each block. Then, rotation-based candidate evaluation is performed to identify the embedding configuration which produces the minimum embedding distortion. Finally, the optimized units are embedded into the RGBA pixels using an adaptive RGBA-LSB embedding method.

The main contributions of this study are:

1. Protection of Raw DNA: the proposed framework provides an extra level of privacy protection by directly hiding real and raw genomic DNA sequences and sensitive biological data.

2. Self-contained Stego-Key: the proposed dynamic spatial embedding eliminates the need of an external key through adaptively using the RGBA channels.

3. Volume Reduction Through Repetition-Aware Compression: the proposed framework compresses high-capacity genomic payloads into bounded units reducing the total number of modified pixels.

4. Adaptive Storage Optimization: the proposed framework uses a dynamic RGBA-based LSB embedding mechanism maximizing storage efficiency and payload utilization.

5. Cuckoo Search-driven Distortion Minimization: a dual-level optimization structure that identifies the optimal embedding configuration that minimizes visual distortion before spatial embedding.

The rest of this paper is organized as follows: Section 2 investigates related work, Section 3 contains a theoretical background on image steganography, Section 4 provides a background on the CS algorithm, Section 5 introduces the proposed CS-based image steganography algorithm, Section 6 showcases experimental results while Section 7 and 8 concludes the paper and provides recommendations for future work respectively.

1.1 Literature Review

Image steganography has evolved in recent years from the traditional fixed-depth LSB embedding to optimization-driven techniques to achieve better imperceptibility and resist steganalysis. CS with chaotic map was used to guide LSB embedding and showed that the optimization-driven embedding location improves the embedding quality and randomness compared to sequential LSB methods with Peak Signal-to-Noise Ratio (PSNR) values of approximately 44 dB and Structural Similarity Index (SSIM) of approximately 0.97 [6]. This shows the effectiveness of optimization-driven embedding techniques in increasing embedding quality while minimizing visual distortion.

A saliency-guided adaptive LSB framework with optimization-driven embedding was proposed to reduce embedding distortion and improve anti-detection performance [7]. This was extended through using fuzzy adaptive embedding frameworks and achieved PSNR values over 73 dB with near-ideal steganalysis performance [8]. This highlights the benefits of using image characteristics to improve imperceptibility with the same embedding effectiveness.

Several studies focused on strengthening the security of steganographic systems through randomization and encryption methods. A multi-level LSB-based image steganography framework was proposed employing randomized embedding and block-wise data distribution to improve security and imperceptibility [9]. This method achieved PSNR values above 80 dB with negligible Mean Squared Error (MSE) showing that randomized spatial embedding can effectively hide information while maintain high visual quality. On the other hand, an enhanced LSB-based steganographic framework was proposed integrating value-difference embedding, Magic Matrix shuffling and multilevel encryption [10]. Experiments on RGB, grayscale, texture and aerial images showed an improvement of 5.561% in PSNR values over existing methods and a 6.43% higher

image quality under different payload sizes.

Another research direction focuses on increasing embedding capacity while maintain good visual quality. A high-capacity RGB image steganography scheme was proposed based on the adjacent mean of neighboring pixels [11]. Secret data are embedded by using local spatial correlations within the image. The proposed method achieved embedding rates above 7.4 bits per pixel (bpp) with PSNR values close to 30 dB. However, the scheme was designed for traditional binary payloads and did not consider the biological sequences, payload restructuring or optimization-driven techniques.

Given the literature reviewed above, existing studies focus on optimizing embedding locations [6], adaptive embedding parameters [7,8], security mechanisms [9,10] or payload capacity [11] while treating the payload as a fixed binary sequence. The proposed framework in this study performs optimization at the payload level, where DNA-tri-mer indices are processed through reversible permutation and rotation operations before embedding. This maximizes repetition opportunities, improves payload efficiency, enhances LSB compatibility while increasing payload diversity. This makes the embedded DNA patterns less predictable while preserving exact sequence reconstruction. The proposed framework integrates DNA sequence representation, repetition-aware processing, adaptive RGBA embedding and dual-level CS optimization in a single unified steganographic framework.

Table 1 summarizes the reviewed steganographic methods and addresses their advantages and limitations. As shown in the table, existing optimization-driven methods achieve high security or capacity for binary streams however, they fail to address the structural redundancy of real biological sequences. The proposed framework bridges this gap by performing optimization methods directly at the payload level using CS and repetition-aware RGBA-LSB embedding.

Table 1. Comparison of existing steganographic methods and the proposed framework.

Method	Optimization	Advantages	Limitations
Double-layer image steganography [3]	DNA sequence mapping & encoding	Provides a two-tier security model by using DNA coding rules to conceal standard messages.	Uses DNA merely as a coding mechanism rather than protecting real raw genomic sequences directly.
Chaotic map-guided LSB substitution [6]	Cuckoo Search Optimization (CSO)	Improves embedding randomness and quality compared to sequential LSB; achieves PSNR 44 dB and SSIM 0.97.	Focuses solely on embedding location selection; lacks payload-level compression or processing for repetitive sequences.
Content-adaptive LSB with hybrid encryption [7]	Ant Colony Optimization (ACO) dispersion & saliency fusion	Reduces embedding artifacts and enhances anti-steganalysis performance against statistical attacks.	High computational overhead from saliency mapping; treats payloads as generic binary streams without structural optimization.
Dynamic depth-tuning LSB embedding [8]	Fuzzy logic adaptive rules	Exceptional imperceptibility (PSNR > 73 dB) with near-ideal resistance to detection.	Rule-based complexity; lacks adaptability for biological sequences with adjacent repetition patterns.
Multi-level LSB with hybrid encryption [9]	Randomized embedding & block-wise distribution	Excellent visual preservation (PSNR > 80 dB with negligible MSE) and high security.	Lacks payload restructuring; assumes uniform randomness, which is inefficient for repetitive biological streams.
Value-difference embedding with multi-level encryption [10]	Magic Matrix shuffling & value-difference mapping	Improved visual quality (+5.56% PSNR gain over standard LSB) across varied image textures.	Algorithmic complexity; does not optimize pixel allocation based on data redundancy or DNA structure.
Adjacent mean of neighboring RGB pixels [11]	Spatial neighborhood statistical modeling	Exceptionally high embedding rate (> 7.4 bpp) while exploiting local image correlations.	Noticeable visual degradation (30 dB PSNR); unsuitable for sensitive biological payloads requiring minimal distortion.

2.IMAGE-BASED STEGANOGRAPHY

Image-based steganography is a technique to hide information through embedding secret information in images. This is done to maintain the visual quality and reduce the possibility of detection [12]. The primary goals of image-based steganography are imperceptibility, embedding capacity and protection against visual image steganalysis attacks. Digital images consist of pixels that contain multiple color components enabling the

encoding of hidden information by modifying the pixel values slightly without noticeable visual distortions.

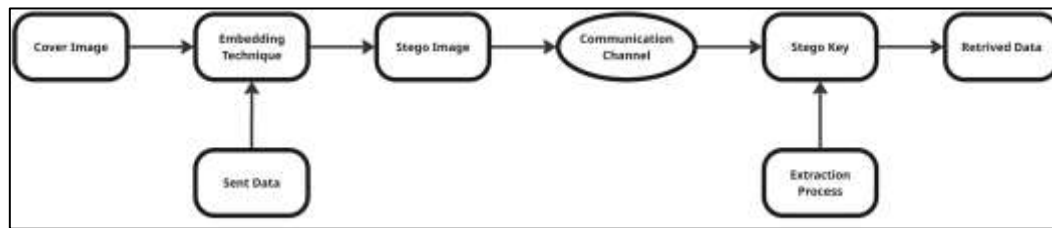


Figure 1. General framework of image steganography illustrating secret data embedding, stego-image generation, and data extraction processes.

The general framework of image-based steganography, shown in Figure 1, consists of several components.

- [7] Cover Image: an input image for data embedding.
- [7] Sent Data: data to be hidden.
- [7] Embedding Technique: method through which sent data are inserted into image pixels.
- [7] Stego-Image: digital image that has the sent data hidden.
- [7] Stego-Key: the key built during embedding stage to be used to extract hidden sent data.
- [7] Extraction Process: process through which the hidden sent data are retrieved from the stego-image.
- [7] Retrieved Data: the data retrieved from the extraction process.

One of the most popular methods for spatial domain image-based steganography is LSB embedding. It is known for its simplicity and high embedding efficiency [13]. In LSB embedding, secret bits are embedded into the least significant bits of the image pixels to minimize visual distortion. However, standard RGB-based embedding methods have limited embedding space and are vulnerable to modern steganalysis techniques.

Adaptive embedding across multiple image components can improve the imperceptibility and improve resistance against deep-learning steganalysis attacks [12,14]. RGBA images have an additional Alpha channel that increases the embedding space compared to RGB representations. Data embedded into LSBs of RGBA channels increases the embedding capacity and hiding efficiency [15]. The Alpha channel is considered an additional embedding layer to increase payload flexibility and restrict embedding distortion. This makes RGBA images more suitable for high-capacity steganography applications specifically in sensitive areas. Developing more robust methods that can resist various detection methods while maintaining the hidden information is made possible by using the unique characteristics of the RGBA images.

3. CUCKOO SEARCH (CS)

CS is a metaheuristic optimization algorithm based on the unique brood parasitism of cuckoo birds by lay eggs in the nests of other host birds [16,17]. Occasionally, the host birds find the foreign eggs and remove them or abandon the nest. Cuckoo eggs resemble the shape, size and color of the host bird's eggs to avoid being detected. Cuckoo birds also improve their reproductive success by removing the host's eggs.

CS uses Lévy flights to perform the random walk while searching. Lévy flights are characterized by a power law distribution of step lengths allowing concentrated local search with occasional long jumps. This helps in exploring the search space more effectively.

In CS a new solution $x_i(t+1)$, equivalent to a host nest, at iteration $t+1$ is generated from the current solution $x_i(t)$ using Equation (1)

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus \text{Lévy}(s, \lambda) \quad (1)$$

where $x_i(t+1)$ is the i th Cuckoo at instance $t+1$, $\alpha > 0$ is the step size, λ is the Lévy distribution coefficient and $\oplus$ is the multiplication bitwise operator.

The Lévy flight provides a random walk while the random step length is drawn from a Lévy distribution given by Equation (2)

$$\text{Lévy}(s, \lambda) \sim s^{-\lambda}, \text{ where } 1 < \lambda \leq 3, \quad (2)$$

which enables a random walk with local exploitation and global exploration. This leads to more efficient optimization than simple random walks [16]. The main rules of CS are summarized as follows:

1. Each cuckoo lays one egg and dumps its egg in a random nest.
2. The best nests with high-quality eggs carry over to next generations.
3. The number of available host nests is fixed and the probability of discovering the laid egg by the host bird can be calculated as $p_a \in [0,1]$.

A pseudo code of CS is shown in Algorithm 1.

Algorithm 1: Cuckoo Search
Input: n the number of nests; p_a the fraction of n nests.
Output: The best solution
01: Objective function: $f(X)$; $X = [x_1, x_2, \dots, x_d]^T$
02: Generate initial population of n host nests $X_i = [x_1^i, x_2^i, \dots, x_d^i]^T$ where $i = 1, 2, \dots, n$
03: $G \leftarrow 1$
04: while ($G < \text{MaxGeneration}$) or (stop criteria) do
05: for $i = 1$ to n do
06: Get a cuckoo X_i^{G+1} from X_i^G by Lévy flights Equation (1)
07: Evaluate its quality/fitness: $f(X_i^{G+1})$
08: Choose a nest among n (say, j) randomly
09: if ($f(X_i^{G+1}) > f(X_j^G)$) then
10: Replace X_j^G by X_i^{G+1}
11: end for
12: A fraction (p_a) of the worse nests are removed and new ones are created
13: Keep the best solutions of the current population
14: Rank the solutions and find the current best
15: $G = G + 1$
16: end while
17: Post-process results and visualization
18: return the best solution.

3.1. Mapping Between the Proposed Framework and CS Terminology

The CS optimization components are adaptively mapped to the encoded DNA embedding environment to integrate the CS within the DNA steganography framework. The permutation-driven optimization is applied over encoded DNA blocks to maximize adjacent repetition formation before embedding.

3.1.1. Egg

An egg represents a newly generated candidate permutation for the current encoded DNA block. Different eggs produce different repetition patterns and different embedding characteristics.

3.1.2. Nest

A nest represents a candidate solution containing a full permutation of the encoded DNA block E_j as shown in Equation (3)

$$N_n^j = \pi_n^j(E_j) \quad (3)$$

where N_n^j is the n th candidate nest configuration for block j , E_j is the j th encoded DNA and π_n^j is the permutation function applied to rearrange the elements of E_j .

3.1.3. Search Space

The search space is a permutation-based space generated over the encoded DNA block. Different candidate permutations produce different eggs. Instead of exploring all possible permutations, the proposed framework uses CS to explore promising candidate permutations with improved efficiency.

3.1.4. Fitness Function

The quality of each candidate nest is evaluated according to the generated adjacent repetitions using Equation (4)

$$F_n = \sum_{K=1}^M R_K \quad (4)$$

where F_n is the overall fitness score of the candidate nest, R_K is the repetition count and M is the total number of nests evaluated.

3.1.5. Lévy Flights

Lévy flight exploration is used, as Equation (1), to improve search space exploration and avoid early convergence to a local-optima.

3.1.6. Abandonment Probability (p_a)

Weak candidate nests are probabilistically abandoned using the abandonment probability (p_a) to increase population diversity and improve exploration.

3.1.7. Elitism

Elitism keeps the highest quality candidate nests across the iterations and passes the elite solution to the rotation-based refinement stage.

The proposed CS integration enables efficient repetition-aware optimization while preserving deterministic

reconstruction capability and adaptive embedding.

4. PROPOSED STEGANOGRAPHY APPROACH

The proposed framework introduces a dual-level, payload-oriented steganographic scheme for embedding raw DNA sequences into 32-bit RGBA cover images. The scheme consists of three sequential and reversible stages as shown in Figure 2. Most optimization-driven image-based steganography methods focus on optimizing embedding locations or decisions inside the cover image. However, the proposed framework optimizes the encoded DNA payload itself before embedding. The first optimization level applies a deterministic permutation optimization to maximize the usable adjacent non-overlapping repetitions across the payload. The second level identifies the deterministic reversible bit-rotation that minimizes the total channel-wise embedding distance between the secret-bit fields and current LSB fields. As a result, the payload is transformed into an embedding-compatible representation before the embedding process begins.

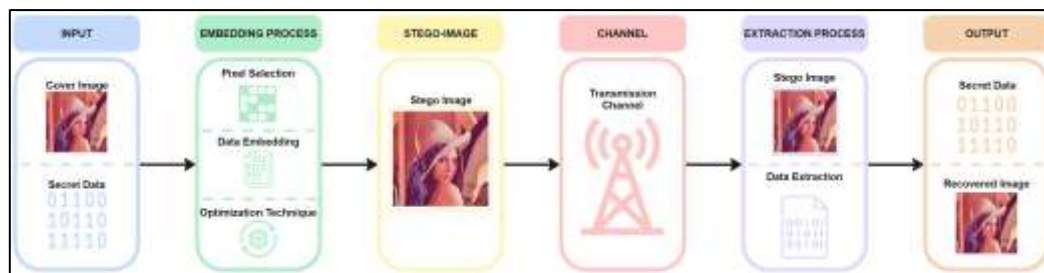


Figure 2. Proposed adaptive DNA steganography framework combining Cuckoo Search-driven repetition optimization and adaptive RGBA-LSB embedding for efficient and deterministic DNA concealment.

After optimizing the payload, the proposed adaptive LSB embedding strategy conceals it by choosing the embedding mode according to the repetition pattern of the encoded element with the repetition information represented in the remaining channels. This allows either direct 2-2-2 LSB embedding or repetition-aware 3-3 LSB embedding. Consequently, Stage I converts the raw DNA sequence into fixed-size blocks. Stage II performs the proposed dual-level payload optimization. Finally, Stage III adaptively embeds the optimized blocks and applies the matching inverse transformation to recover the original DNA. The process is done using only a single global seed and compact block-level metadata.

A comprehensive summary of the three-phase steganographic workflow is shown in Figure 3.

Stage I: DNA Representation

Objective: Convert the raw DNA sequence into fixed-size encoded

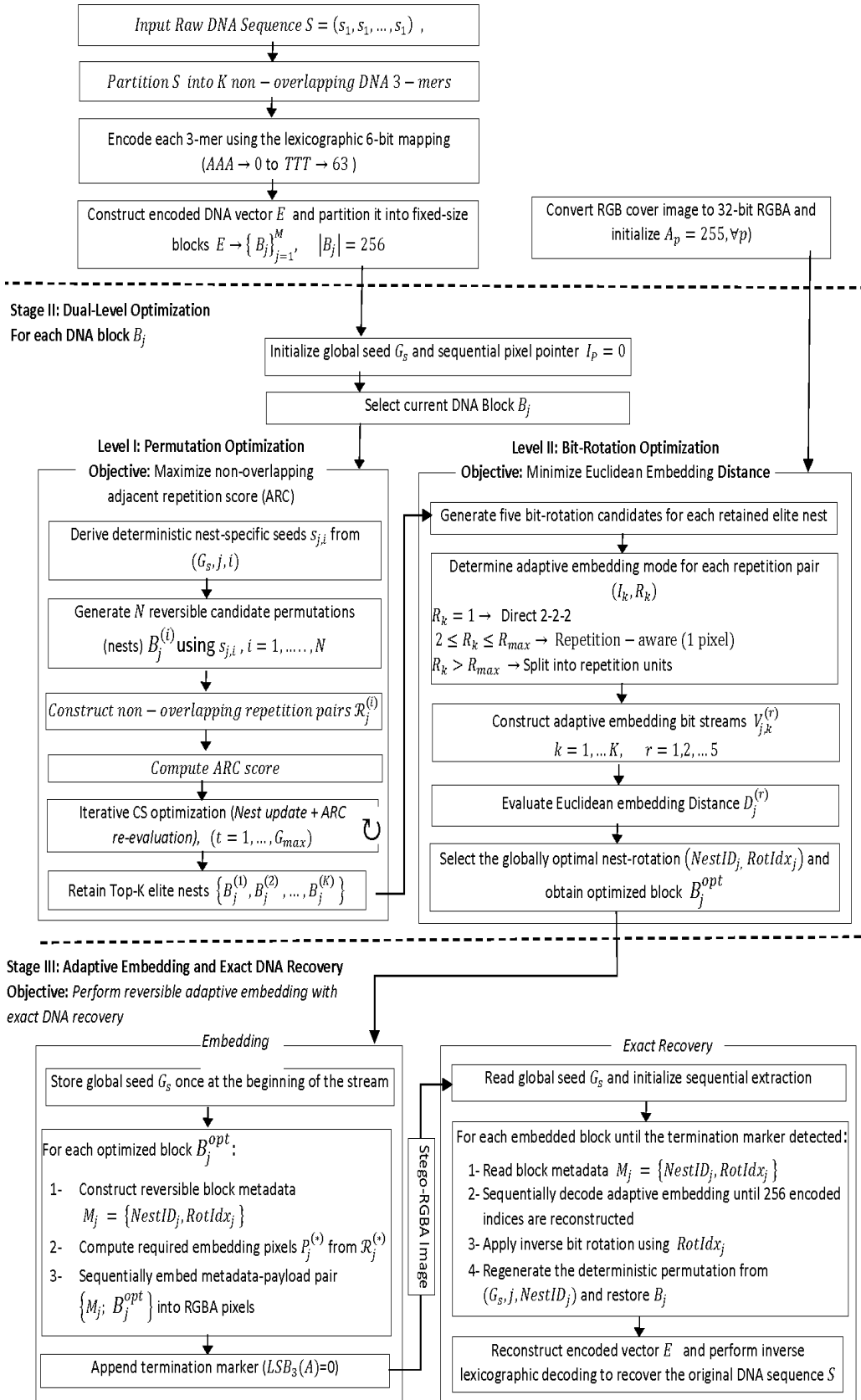


Figure 3. Architectural overview of the proposed adaptive DNA steganography framework, integrating Cuckoo Search-driven repetition optimization with adaptive RGBA-LSB embedding for efficient and deterministic DNA concealment.

The following subsections describe each stage of the proposed framework in detail.

4.1.Stage I: DNA and Cover Image Representation

The input DNA sequence in this stage is divided into consecutive non-overlapping tri-mers. The RGB cover image is also converted into a 32-bit RGBA image. This prepares both the cover image and DNA sequence for the dual-level optimization and adaptive embedding stages that will follow. The DNA sequence S is divided into $K=S/3$ consecutive non-overlapping tri-mers where, S denotes the total number of nucleotides $\{A,C,G,T\}$. The resulting $4^3=64$ possible DNA tri-mers are deterministically encoded using a 6-bit lexicographic mapping to produce the encoded DNA vector E . The DNA vector is then divided into M fixed-size blocks $\{B_j\}_{j=1}^M$ each holding 256 encoded elements.

In parallel, the RGB cover image is converted into a 32-bit RGBA cover image by adding an alpha channel initialized to 255 across all pixels. This one-time step keeps the original visual content unchanged while providing the four-channel representation for the proposed adaptive embedding mechanism.

Each DNA block B_j is then processed independently throughout the proposed framework. Starting with the dual-level payload optimization stage followed by the adaptive embedding stage where the RGBA cover image acts as the embedding mechanism.

4.2.Stage II: Dual-Level Payload Optimization

Each encoded DNA block B_j in this stage is optimized independently before the adaptive embedding stage. The global seed G_s allows deterministic regeneration of all candidate permutations. The sequential pixel pointer IP marks the cover-image pixels used to compute the expected embedding distortion. The optimization process follows a two-level hierarchy. Level I gives a priority to repetition efficiency by keeping only the nests with the highest non-overlapping adjacent repetitions. Level II minimizes the cumulative embedding distance only among this elite set. This structured separation avoids a weighted objective and prevents a small reduction in distortion from eliminating a better structural repetition pattern.

4.2.1.Level I: Permutation Optimization

N candidate permutation, referred to as nests, are generated for each block B_j using deterministic nest-specific seeds derived from G_s , the block index j , and the corresponding $NestID_j$. Any selected permutation can be regenerated during extraction without storing it completely.

Each nest is scanned sequentially to find consecutive non-overlapping repetition pairs l_k, R_k , where l_k denotes the encoded DNA value and R_k is its run length. Its adjacent repetition count is computed using Equation (5)

$$ARC(B_j^{(i)}) = \sum_{k=1}^{n_{i,j}} \max(R_k - 1, 0) \quad (5)$$

where $B_j^{(i)}$ is the i th nest and $n_{i,j}$ is the number of repetition pairs in that nest. A larger ARC value shows that more encoded elements can be represented using repetition-aware embedding instead of embedding it individually.

CS iteratively updates the nest population, abandons lower-quality candidates according to abandonment probability p_a and recalculates the ARC score until G_{max} is reached. The elite set is formed using Equation (6)

$$\epsilon_j = TopK_{i \in \{1, \dots, N\}}[ARC(B_j^{(i)})] \quad (6)$$

where ϵ_j contains the highest-quality nests that are carried over to Level II. Based on this, only the nests satisfying the repetition-quality constraint are considered during distortion minimization.

4.2.2.Level II: Bit-rotation Optimization

For every nest in ϵ_j , five reversible circular right rotations are evaluated for each 6-bit element x as shown in Equation (7)

$$\rho_\ell(x) = ROR_6(x, \ell), \quad \ell = 1, 2, \dots, 5 \quad (7)$$

where ℓ is the rotation index. The same rotation is applied all elements of a candidate nest. The binary representation is changed without changing the element order or the repetition structure since the circular rotation is one-to-one.

For every nest-rotation candidate, the adaptive embedding fields are constructed. A pair with $R_k=1$ follows the direct 2-2-2 embedding mode over the R, G and B channels. However, for $2 \leq R_k \leq 11$, the 6-bit rotated DNA is distributed using 3-3 embedding over R and G channels while the repetition count is represented in the 2-LSB fields of B and A channels. The corresponding quotient and remainder are obtained using Equation (8).

$$R_k = 3q_k + r_k, \quad q_k = \left\lfloor \frac{R_k}{3} \right\rfloor, \quad r_k = R_k \bmod 3 \quad (8)$$

r_k is assigned to the B channel and q_k is assigned to the A channel. For repetitions counts above $R_{max}=11$, they are divided into consecutive representable units before evaluation.

The expected distortion is calculated separately for each channel field to be modified during embedding. Let Sc_b denote the decimal value of the b-bit secret field assigned to channel C and let Ccb denote the decimal value of the corresponding current b-bit LSB field in the cover image.

For direct embedding, the channel-wise Euclidean distance is shown in Equation (9).

$$D_{dir} = \sqrt{(S_R^{(2)} - C_R^{(2)})^2 + (S_G^{(2)} - C_G^{(2)})^2 + (S_B^{(2)} - C_B^{(2)})^2} \quad (9)$$

For repetition-aware embedding, all four RGBA channels are included since the R and G carry the rotated DNA while the B and A carry r_k and q_k respectively. The channel-wise Euclidean distance in this mode is shown in Equation (10).

$$D_{rep} = \sqrt{(S_R^{(3)} - C_R^{(3)})^2 + (S_G^{(3)} - C_G^{(3)})^2 + (r_k - C_B^{(2)})^2 + (q_k - C_B^{(2)})^2} \quad (10)$$

Consider a direct-mode rotated element divided as 11|01|10 (3, 1, 2). If the corresponding current 2-LSB fields are 10|01|11 (2, 1, 3), then $D_{dir}=3-2+1-1+2-3=2$. This example confirms that the distance is evaluated independently for each channel rather than the complete 6-bit values.

For each elite nest, the 5-bit rotation candidates are evaluated using the processed adaptive embedding method. The D_{dir} and D_{rep} values are accumulated over all adaptive embedding units of block B_j producing the embedding distance of every candidate. The candidate with the minimum embedding distance is selected and its corresponding $NestID_j$ and $RotIdx_j$ are used as the block metadata. The resulting optimized DNA block B_{jopt} is then forwarded to the adaptive embedding stage.

In summary, the proposed hierarchical optimization first maximizes repetition efficiency through elite nest selection and then minimizes the expected embedding distortion within the constrained solution space. Together with the global seed G_s , the stored $NestID_j$ and $RotIdx_j$ are sufficient to deterministically reconstruct the optimized block during extraction.

4.3.Stage III: Adaptive Embedding and Exact DNA Extraction

The optimized blocks B_{jopt} are embedded one after one into the 32-bit RGBA cover image using the proposed adaptive embedding mechanism. The global seed G_s is written once at the start of the image while each block's $NestID$ and $RotIdx$ are embedded immediately before that block's payload for reconstruction. A single sequential pixel pointer IP tracks pixel position throughout embedding and is incremented after each metadata or payload unit is written. This eliminates the need to store explicit pixel coordinates or full permutation vectors. The channel assignments, already known from D_{dir} and D_{rep} from Equations (9) and (10) respectively, are used without additional optimization.

4.3.1.Adaptive Embedding

For each optimized block B_{jopt} , the $NestID$ and $RotIdx$ are embedded first. Afterwards, each element is embedded according to its non-overlapping run length R_k with elements of $R_k=1$ using direct mode while elements of $R_k \geq 2$ use the repetition-aware mode.

Using the direct mode, the element's 6-bit code is divided into three 2-bit fields. These fields are placed in the 2-LSBs of the R, G and B channels. The value 11 of the alpha channel indicates direct mode which is initialized from the beginning. This leaves the alpha channel unchanged while modifying the R, G and B channels. The only exception is when the termination bit is written when the pixel being processed is the last payload pixel.

On the other hand, using the repetition-aware mode, the rotated 6-bit element is embedded a single time split across three bits each in the R and G channels. The repetition count is encoded using the quotient-remainder pair in Equation (8), where q_k is stored in the 2-LSBs of B and $r_k \in \{00, 01, 10\}$ is stored in the alpha channel leaving 11 free to uniquely signal direct mode. This helps in identifying the embedding mode without any external control metadata. When R_k is greater than 11, the run is split into multiple segments that are encoded consecutively until the full repetition count is stored.

The adaptive channel allocation is summarized in Table 2.

Table 2. Adaptive channel allocation for the direct and repetition-aware embedding modes.

Embedding Mode	Condition	Encoded DNA Allocation	B-Channel Field	A-Channel Field
Direct	$R_k = 1$	2-LSBs of the R, G, and B DNA bits	—	11 (mode indicator)
Repetition-aware	$R_k \geq 2$	3-LSBs of the R and G	q_k	$r_k \in \{0,1,2\}$

After each payload unit is embedded, IP is incremented to the next available pixel traversing the image sequentially. The third LSB of the alpha channel is used as a global termination marker as it is set to 0 only in the final payload pixel while being set to 1 otherwise. This enables the extraction process to identify the exact payload boundary only by sequential traversing. This eliminates the need to store either the payload length or the total number of embedded blocks.

Algorithm 2 summarizes the proposed dual-level payload optimization and adaptive embedding procedure.

Algorithm 2: Proposed Adaptive DNA Embedding Framework
Input: DNA sequence S , RGB cover image C , global seed G_s , and CS parameters N , G_{max} , P_a , and K Output: RGBA stego image C_{stego}
01: Convert C to 32-bit RGBA and initialize $A = 255$ for every pixel 02: Encode the non-overlapping DNA 3-mers of S into the 6-bit vector E 03: Partition E into blocks $\{B_j\}$, where $ B_j = 256$ 04: Embed G_s once at the image head and initialize I_p 05: for each block B_j do 06: Generate N deterministic permutation nests from $(G_s, j, NestID)$ 07: for $t = 1$ to G_{max} do 08: Evaluate the nests using ARC in Equation (5) 09: Update the population and abandon inferior nests using P_a 10: end for 11: Retain the Top- K elite nests according to Equation (6) 12: for each elite nest and $RotIdx \in 1, \dots, 5$ do 13: Apply the rotation in Equation (7) and construct its embedding units 14: Accumulate D_{dir} or D_{rep} using Equation (9) or (10), respectively 15: end for 16: Select the nest-rotation pair with minimum cumulative distance 17: Obtain B_j^{opt} and retain $NestID_j$ and $RotIdx_j$ 18: Embed $NestID_j$ and $RotIdx_j$; advance I_p after each metadata unit 19: for each non-overlapping run (I_k, R_k) in B_j^{opt} do 20: if $R_k = 1$ then 21: Embed I_k using 2-2-2 in R, G, and B; advance I_p 22: else 23: Split R_k into consecutive units not exceeding 11 24: for each repetition unit do 25: Determine q_k and r_k using Equation (8) 26: Embed I_k using 3-3 in R and G 27: Store q_k in B and r_k in A; advance I_p 28: end for 29: end if 30: end for 31: end for 32: Set the third alpha LSB of the final payload pixel to 0 33: return C_{stego} 34: End

4.3.2.Exact DNA Extraction

Extraction follows the same sequential pixel traversing and reverse the embedding and optimization processes. First, the global seed G_s is recovered. Then, I_p is initialized to the beginning of the block-level data. For each block, the stored $NestID$ and $RotIdx$ are recovered before decoding its hidden payload units.

The 2-LSB field of the alpha channel determines the decoding method. If it equals 11, direct mode is selected, and the three 2-bit fields are extracted from the R, G and B channels. They are then concatenated to reconstruct the corresponding 6-bit encoded element. Otherwise, the repetition-aware mode is selected. The 3-LSBs of the R and G channels reconstruct the encoded element while q_k and r_k are recovered from the B and A channels respectively. The repetition count R_k is recovered using the inverse of Equation (8) and the encoded element is expanded to its recovered run length. Consecutive repetition units are combined if R_{max} exceeded 11.

The hidden units are decoded sequentially until the 256 encoded elements of the block have been recovered. The circular rotation, specified by $RotIdx$, is the inverted. The selected permutation is reconstructed deterministically from G_s , the block index j , and $NestID$. The corresponding inverse permutation restores the original DNA block B_j .

The process is repeated for every block until the first payload pixel, indicated by the third alpha LSB set to 0, is reached indicating the end of the embedded data. This terminates the extraction process automatically without requiring any stored payload size or block count data.

Finally, the recovered blocks are concatenated to reconstruct the encoded DNA vector E . The corresponding DNA tri-mers are restored through the inverse lexicographic 6-bit mapping. Ordered concatenation of these

tri-mers recovers the original DNA sequence S . Exact DNA recovery is guaranteed since the permutation, bit rotation, adaptive channel allocation, repetition presentation, sequential traversal, termination marker and metadata interpretation are deterministic and invertible without storing permutation vectors or pixel-location metadata.

Algorithm 3 summarizes the exact DNA extraction process.

Algorithm 3: Proposed DNA Extraction Framework
Input: RGBA stego image C_{stego}
Output: Exactly recovered DNA sequence S
01: Recover G_s from the image head and initialize I_p 02: Initialize $E = \emptyset, j = 1$, and $EndMarker = 1$ 03: repeat 04: Read $NestID_j$ and $RotIdx_j$; advance I_p after each metadata unit 05: Initialize $B_j^{opt} = \emptyset$ 06: while $ B_j^{opt} < 256$ do 07: Read the current pixel at I_p 08: if the alpha 2-LSB field = 11 then 09: Reconstruct one 6-bit value from the 2-LSBs of R, G, and B 10: Append the reconstructed value once to B_j^{opt} 11: else 12: Reconstruct the 6-bit value from the 3-LSBs of R and G 13: Recover q_k from B and r_k from A 14: Compute R_k using Equation (8) and append the value R_k times 15: end if 16: $EndMarker \leftarrow$ third alpha LSB of the current pixel 17: advance I_p 18: end while 19: Apply the inverse rotation specified by $RotIdx_j$ 20: Regenerate the permutation from $(G_s, j, NestID_j)$ and invert it 21: Append the recovered block B_j to E ; set $j \leftarrow j + 1$ 22: until $EndMarker = 0$ 23: Decode E using the inverse lexicographic 6-bit mapping 24: Concatenate the recovered DNA 3-mers 25: return S 26: End

5.Experimental Results

This section presents the experimental results of the proposed framework applied to a given dataset. It also contains a description of the used dataset, the evaluation criteria, the suggested parameters and finally the experimental results.

5.1.Dataset Preparation

The performance of the proposed framework was test on standard benchmark RGBA images with a resolution of 512×512 pixels. The cover images were obtained from the National Center for Biotechnology Information (NCBI) GenBank [18], a standard repository used in image steganography research. Three specific images were selected because they have distinct visual characteristics and structural challenges. This allows a comprehensive evaluation of the proposed framework under varying conditions.

The performance of the proposed algorithm was evaluated using standard benchmark RGBA images with a resolution of 512×512 pixels. These cover images were acquired from the National Center for Biotechnology Information (NCBI) GenBank [18], a widely recognized and standard repository in image processing and steganography research. These three specific cover images (Lena, Penguin, Baboon), shown in Figure 4, were selected because they present distinct visual characteristics and structural challenges, allowing for a comprehensive evaluation of the proposed framework under varying conditions.

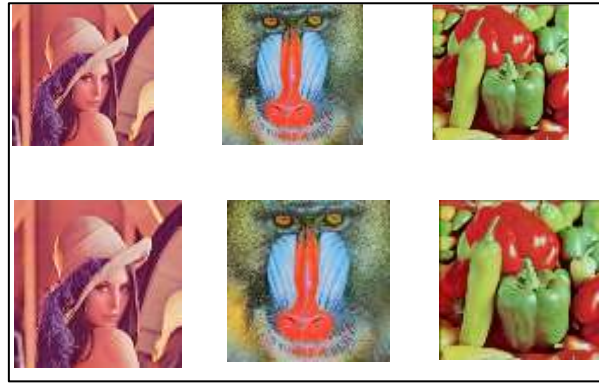


Figure 4. The cover images and stego-images used in the experimental results.

The payload of clandestine data embedding is a DNA sequence of 238,080 nucleotides from the NCBI database encoded as 79,360 tri-mers.

5.2. Evaluation Criteria

5.2.1. Mean Squared Error (MSE)

The MSE measures the difference between the original picture and the stego-image by calculating the distortion in the stego-image. This is done by comparing every byte in the stego-image with its matching byte in the original image as shown in Equation (11)

$$MSE = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W (I_{i,j} - I'_{i,j})^2 \quad (11)$$

where $I_{i,j}$ is the original image and $I'_{i,j}$ is the stego-image both with a resolution of $H \times W$ pixels.

5.2.2. Peak Signal-to-Noise Ratio (PSNR)

PSNR is the main statistical parameter used in the study. The stego quality is defined as the MSE divided by the maximum color intensity value in the picture. If the value of the PSNR is high, the difference between original and stego-image will be negligible and distortion will be less. The PSNR is calculated using Equation (12)

$$PSNR = 10 \log_{10} \frac{L^2}{MSE} \quad (12)$$

where L is the peak signal level for an image with a color component $c \in \{R, G, B, A\}$ respectively and is taken as 255.

5.2.3. Structural Similarity Index (SSIM)

The SSIM is an additional metric which evaluates the similarity between both the cover and stego images by analyzing information alterations between the loss of correlation, luminous distortion and contrast distortion. It measures the distortion in picture quality after the embedding process and is calculated using Equation (13)

$$SSIM = \frac{(\mu_C \mu_{C^*} + d_1)(\sigma_{CC^*} + d_2)}{(\mu_C^2 + \mu_{C^*}^2 + d_1)(\sigma_C^2 + \sigma_{C^*}^2 + d_2)} \quad (13)$$

where μ_C and μ_{C^*} are the average of the original image C and the stego-image C^* , σ_C^2 and $\sigma_{C^*}^2$ are the variance of C and C^* , $\mu_C \mu_{C^*}$ is the covariance of C and C^* and d_1 and d_2 two variables used to stabilize the division with a weak denominator.

5.2.4. Embedding Capacity

The payload that the scheme can handle is represented both as the total number of embedded bits and as the normalized embedding rate in bpp as shown in Equation (14).

$$BER = \frac{\text{number of erroneously extracted bits}}{\text{total number of embedded bits}} \times 100 \quad (14)$$

5.3. Cuckoo Search Algorithm for Image Steganography Parameters

The CS algorithm was used with a population of nests to optimize the embedding process and minimize image distortion. The abandonment probability p_a was set to 20% while the Lévy exponent λ was set to 1.5. Stopping criteria was set to a maximum number of 100 iterations. The search space consisted of the set of the cover image pixels in the embedding optimization process.

5.4. Experimental Results Presentation

The proposed CS optimization-driven framework increased the PSNR to 49.07-49.12 dB which is relatively higher than the direct embedding with a PSNR of 44.76-44.81 dB and the repetition-aware embedding with a PSNR of 44.88-44.91 dB. The MSE was reduced from 2.14-2.17 and 2.09-2.12 for the direct and repetition-aware embedding respectively to 0.79-0.80 for the CS-optimization driven framework. SSIM values were in the range of 0.999893-0.999932 indicating near perfect visual preservation with minimal distortion. High payload ratio of 95.6% was also maintained showing the gains in imperceptibility were achieved with a very small reduction in embedding capacity.

The improvement in performance is due to the two optimization stages working together. Repetition-aware

encoding reduces pixels carrying repeated DNA units, while CS selects the bit rotation minimizing flipped LSBs per block. Consistent results across cover images with different texture and color statistics (Penguin, Lena and Baboon) show that the framework generalizes across different cover images rather than being image-specific. Table 3 shows the performance of the direct embedding method across the three benchmark images. The results show acceptable image quality with PSNR values in the range of 44.76-44.81 dB and SSIM values exceeding 0.999. However, the MSE are comparatively high when compared to the optimization-driven approach.

Table 3. Cover-Image and Stego-Image Characteristics for Direct Embedding

Cover-Images		Stego-Image				
Image name	Resolution (W × H)	Pixel Used	BER	MSE	PSNR	SIMM
Penguin.png	512 × 512	79360	1.0000	2.172	44.76	0.999753
Lena.png	512 × 512	79360	1.00000	2.16	44.77	0.999691
Baboon.png	512 × 512	79360	1.0000000	2.14	44.81	0.999606

Table 4 shows the performance of the repetition-aware embedding method. The repetition-aware embedding slightly reduced the payload utilization. It also introduced slight reduction in the MSE compared to the direct embedding; however, it remains relatively higher compared to the optimization-driven approach.

Table 4. Cover-Image and Stego-Image Characteristics for Repetition-Aware Embedding

Cover-Images		Stego-Image				
Image name	Resolution (W × H)	Pixel Used	BER	MSE	PSNR	SIMM
Penguin.png	512 × 512	75251	0.9734	2.115150	44.88	0.999760
Lena.png	512 × 512	75867	0.9560	2.099	44.91	0.999916
Baboon.png	512 × 512	77251	0.9734	2.096938	44.91	0.999620

Table 5 shows the performance of the proposed CS optimization-driven embedding approach. The proposed approach achieves the highest PSNR values and the lowest MSE values compared to the other methods. This shows the effectiveness of the optimization-driven framework in minimizing visual distortion while maximizing the embedding capacity.

Table 5. Cover-Image and Stego-Image Characteristics for the Proposed Cuckoo Search Optimization-driven Embedding

Cover-Images		Stego-Image				
Image name	Resolution (W × H)	Pixel Used	BER	MSE	PSNR	SIMM
Penguin.png	512 × 512	75904	0.9565	0.804661	49.07	0.999932
Lena.png	512 × 512	75867	0.9560	0.802001	49.09	0.999916
Baboon.png	512 × 512	75876	0.9561	0.795572	49.12	0.999893

6. Conclusion

This paper proposed an adaptive dual-level DNA image steganography framework that integrates repetition-aware DNA encoding with Cuckoo Search optimization. This paper contributes to image-based steganography by directly protecting raw DNA sequences, eliminating the need for a stego-key, adaptive storage optimization and introducing a dual-level Cuckoo Search optimization-driven distortion minimization. Experimental results using benchmark images shows the proposed framework's ability to outperform conventional direct and repetition-aware embedding methods. The proposed method achieved Peak Signal-to-Noise Ratio values of 49.07-49.12 dB, Mean Squared Error as low as 0.79 and Structural Similarity Index of values up to 0.999931 showing superior visual quality and structural preservation.

7. RECOMMENDATIONS FOR FUTURE WORK

Future research directions may include the development of hybrid optimization frameworks combining Cuckoo Search with other metaheuristic algorithms such as Particle Swarm Optimization and Differential Evolution. Moreover, adaptive channel selection mechanisms and reversible data hiding techniques can be explored to improve embedding efficiency and security. Evaluating the proposed framework against modern steganalysis methods and deep learning-based attacks would further validate its practical applicability.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those

of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

References

1. Yadav, P.; Mishra, N.; Sharma, S. A Secure Video Steganography with Encryption Based on LSB Technique. In Proceedings of the 2013 IEEE International Conference on Computational Intelligence and Computing Research; December 2013; pp. 1–5.
2. Aziz, M.M.A.; Sadat, M.N.; Alhadidi, D.; Wang, S.; Jiang, X.; Brown, C.L.; Mohammed, N. Privacy-Preserving Techniques of Genomic Data—a Survey. *Brief Bioinform* 2019, 20, 887–895, doi:10.1093/bib/bbx139.
3. Aljahdali, A.O.; Al-Harbi, O.A. Double Layer Steganography Technique Using DNA Sequences and Images. *PeerJ Comput Sci* 2023, 9, e1379, doi:10.7717/peerj-cs.1379.
4. Homer, N.; Szelinger, S.; Redman, M.; Duggan, D.; Tembe, W.; Muehling, J.; Pearson, J.V.; Stephan, D.A.; Nelson, S.F.; Craig, D.W. Resolving Individuals Contributing Trace Amounts of DNA to Highly Complex Mixtures Using High-Density SNP Genotyping Microarrays. *PLOS Genetics* 2008, 4, e1000167, doi:10.1371/journal.pgen.1000167.
5. Gymrek, M.; McGuire, A.L.; Golan, D.; Halperin, E.; Erlich, Y. Identifying Personal Genomes by Surname Inference. *Science* 2013, 339, 321–324, doi:10.1126/science.1229566.
6. Walia, G.S.; Makhija, S.; Singh, K.; Sharma, K. Robust Stego-Key Directed LSB Substitution Scheme Based upon Cuckoo Search and Chaotic Map. *Optik* 2018, 170, 106–124, doi:10.1016/j.ijleo.2018.04.135.
7. Aljughaiman, A.; Alrawashdeh, R. Content-Adaptive LSB Steganography with Saliency Fusion, ACO Dispersion, and Hybrid Encryption with Ablation Study. *Sci Rep* 2025, 16, 3829, doi:10.1038/s41598-025-33920-9.
8. Joshi, A.; Bhand, K. Adaptive Fuzzy Logic-Based Steganographic Encryption Framework: A Comprehensive Experimental Evaluation 2026.
9. Alanzy, M.; Alomrani, R.; Alqarni, B.; Almutairi, S. Image Steganography Using LSB and Hybrid Encryption Algorithms. *Applied Sciences* 2023, 13, 11771, doi:10.3390/app132111771.
10. Rahman, S.; Uddin, J.; Khan, H.U.; Hussain, H.; Khan, A.A.; Zakarya, M. A Novel Steganography Technique for Digital Images Using the Least Significant Bit Substitution Method. *IEEE Access* 2022, 10, 124053–124075, doi:10.1109/ACCESS.2022.3224745.
11. Chuang, Y.-H.; Lin, B.-S.; Chen, Y.-X.; Shiu, H.-J. Steganography in RGB Images Using Adjacent Mean. *IEEE Access* 2021, 9, 164256–164274, doi:10.1109/ACCESS.2021.3132424.
12. Zhang, F.; Dong, Y.; Sun, H. Research on Key Technologies of Image Steganography Based on Simultaneous Deception of Vision and Deep Learning Models. *Applied Sciences* 2024, 14, 10458, doi:10.3390/app142210458.
13. Khalil, N.; Sarhan, A.; Alshewimy, M.A.M. A Secure Image Steganography Based on LSB Technique and 2D Chaotic Maps. *Computers and Electrical Engineering* 2024, 119, 109566, doi:10.1016/j.compeleceng.2024.109566.
14. Kim, M.; Cho, Y.; Park, H.; Qu, G. ASIGM: An Innovative Adversarial Stego Image Generation Method for Fooling Convolutional Neural Network-Based Image Steganalysis Models. *Electronics* 2025, 14, 764, doi:10.3390/electronics14040764.
15. Alseelawi, N.; Ismaeel, T.; Sabir, F. High Capacity Steganography Method Based upon RGBA Image; 2015;
16. Yang, X.-S.; Deb, S. Cuckoo Search via Lévy Flights. In Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC); December 2009; pp. 210–214.
17. Aly, W.M. Evaluation of Cuckoo Search Usage for Model Parameters Estimation. *International Journal of Computer Applications* 2013, 78, 1–6.
18. GenBank Overview Available online: <https://www.ncbi.nlm.nih.gov/genbank/> (accessed on 25 August 2026).