DOI: http://doi.org/10.32792/utq.jceps.10.01.01

High Security and Robustness Image Steganography Based On Ant Colony Optimization Algorithms and Discrete Cosine Transform

Ahmed Shihab Ahmed [1] Hussein Ali Salah [2]
ahmedshihabinfo@conursing.uobaghdad.edu.iq hussein_tech@mtu.edu.iq

[1] Department of Basic Sciences, College of Nursing, University of Baghdad, IRAQ
[2] Department of Computer Systems, Technical Institute- Suwaira, Middle Technical University, IRAQ

Received 9/7/2023, Accepted 9/10/2023, Published 1/12/2023

Abstract

Steganography, which is the science of delivering a message between parties in a way that an eavesdropper will not be aware that the message exists, is one of the key disciplines that have a considerable interest in fields. The suggested method seeks to conceal a smaller color image within a larger color image. It utilizes the transform domain throughout the steganography process to increase its resistance to changes and treatments made to the cover image. In order to achieve both complexity security and robustness, the project seeks to implement and utilize two techniques: ant colony optimization and transformation domain approach. The technologies under consideration suggest a Steganography method for digital photographs. It employs Discrete Cosine Transform (DCT) to achieve the same aims in terms of security, transparency and robustness. Additionally, it makes use of Once an Ant Colony Optimization (ACO) to increase robustness against signal processing attacks and imperceptibility in accordance with the human visual system. This work implements ACO for the cover to find the embedding locations on the graph path, and then finds locations to hide information based on threshold method, when looking for places to hide information blocks, DCT uses an intelligent block matching technique between the embedded image and the cover image, and both systems were tested against each other to compare their performance. Six stages make up the proposed secret key steganography system: test, transform, key creation and substitution, ant colony optimization, matching, inverse transform, and key encryption and concealing.

Keywords: Ant Colony Optimization (ACO), Discrete Cosine Transform (DCT), Image Steganography, Growth Algorithm.

1. Introduction

In the contemporary world, communication is essential. Information is exchanged through many data routes during conversation. Serious security issues could arise with this process. Thus, finding methods to safeguard important information while it is being transmitted has drawn increasing interest. A number of ways have been developed to encrypt and decrypt data in order to guarantee message secrecy.

Cryptography is a technology used to protect communication secrecy. While cryptography encrypts data using a key and sends it over a specified channel, steganography involves hiding data in a way that makes it appear as though nothing is hidden. [1].

Greek terms stego, which means cover, and grapha, which means writing, are the origin of the word "steganography." It is described as writing that is covered up to conceal the existence of the actual message. Information is concealed inside other media via steganography. Using a stego key, it involves two primary steps: hiding a secret message under a cover and removing the cover to reveal the hidden message. Stego media is produced when a cover and an embedded message are combined. Secret communications can be hidden and extracted using Stego keys. Only those who have stego keys can successfully decipher secret messages that have been hidden [2].

The following equation can be used to define steganography: stego media = cover media + embedded message + stego key. Technical and linguistic steganography are the two subcategories of steganography. In linguistic steganography, secret information is concealed by using everyday language as a conduit. Spatial domain embedding or frequency domain embedding are two examples of steganographic techniques. Images are converted into frequency components during frequency domain embedding using the discrete cosine transform, rapid Fourier transform, and discrete cosine transform (DCT). It is possible to embed messages at the bit or block level. Depending on the pixel intensity, information is immediately buried in spatial domain embedding. While spatial domain techniques offer large capacity and are frequently used in steganography, frequency domain procedures are reliable and frequently utilized for watermarking. [3].

Many audiovisual items are now daily shared between users of social media apps like Facebook, Line, WeChat, and Instagram on intelligent mobile phones [4]. JPEG compression, a popular lossy operation to reduce the storage capacity of images while maintaining acceptable visual quality, is a common procedure used in many applications, although it is limited by the limitations of the mobile phone. This technique frequently destroys the steganographic (stego) objects concealed secret information that was encoded using conventional adaptive JPEG steganographic algorithms [5]. To deal with such lossy channels, robust steganography that resists JPEG compression is suggested. Designing a strong steganographic method naturally involves studying the fundamentals of a strong watermark. Since 1990, researchers have been working on developing robust watermarking technology that can withstand lossy operations like down sampling, watermarking, JPEG compression, and others. Several robust watermarking techniques have been proposed, including spread spectrum modulation [6,7]. Feature-based algorithms, and scale invariant feature transformation. In general, the priority position of a resilient watermarking design is the existence of the embedded watermark against various loss procedures. [8].

Steganography of images offers the advantages of protecting secret information content and concealing transmission behavior, increasing the security and confidentiality of secret information conveyed. The development of image steganography fills the gap left by the absence of image encryption and offers the necessary technical foundation for covert communication. Steganography is the science and art of concealing secret information so that only the intended recipient is able to recognize its presence [9]. Publicly accessible things conceal sensitive information. Carrier items are those open to the general public. Audio, video, text, and photographs are some examples of carrier objects that are utilized in various steganography techniques [10]. Because the human visual system is insensitive to image details and images have a high degree of pixel value redundancy, they are regarded as excellent carriers among other sorts of carrier objects. The host image or cover picture is the original image that does not include any hidden information, while the steganographic image is the image that does [11].

Cryptography, the art and science of secret communication, is related to steganography. While steganography also conceals the existence of hidden communication, cryptography seeks to mask the messages substance. For instance, two users desire to exchange information. However, notice is investigating that contact through a local server or an ISP (Internet Provider Service). Steganography offers a scenario in which sender A wishes to deliver message M to receiver B in order to safeguard this communication. Sender received stego object S, embedded it over cover media C, and sent it over an unsecured channel. The phrases "cover object" and "stego object" are defined as different sorts of multimedia objects that are used to conceal data, respectively [12].

The two primary categories of steganography techniques are spatial domain and frequency domain techniques. When processing an image in the frequency domain, the data is concealed on the altered coefficients after the image has been converted in the spatial domain [13]. LSB, PVD, EBE, RPE, PMM, and Pixel intensity based are a few examples of spatial domain techniques. DCT, DWT, DFT, IWT, and DCVT are examples of frequency domain approaches. Particularly vulnerable to pixel manipulation and visual attack are spatial domain techniques [14].

Military communication systems are increasingly utilizing traffic security measures that, rather than just using encryption to mask a messages content, also attempt to hide the messages source, receiver, or even its very existence [15]. Information masking methods and applications have become more sophisticated and prevalent. A huge number of people have been sharing information, such as digital information, broadly due to the expansion of high-speed telecommunication and the rising number of Internet users [16]. Information concealment during data transfer can be categorized as seen in Figures (1).

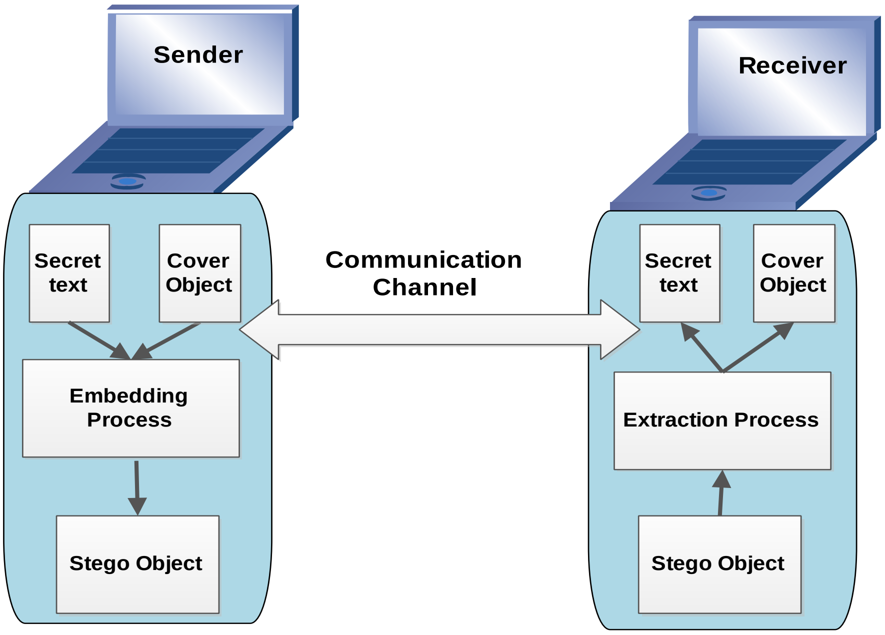


Figure 1. Overview: Basics of the Steganography system

The main problem in hiding one image in a deferent host image is the volume of data that must be handled, requiring the use of a unique the method of data embedding to ensure adequate translucence, ability and strength. In order to achieve a robust system, the proposed image steganography system uses a transform domain technique for embedding a color secret image into a color cover image. The main idea of the proposed systems is to use the ant colony optimization (ACO) and discrete cosine transform (DCT) to embed the secret image to secure enough capacity as well as transparency, robustness and complexity, and finally comparing between the results of the proposed system with other studies to determine the accuracy and efficiency of the proposed system.

The main contributions of this paper are: 1) This work introduces an effective steganographic model. That is one of the primary contributions of this paper. 2) The proposed technique, when compared to other techniques, improves efficiency by improving hiding capacity, simplicity of applications and high running efficiency. The proposed scheme performs well in key criteria such as robustness, high imperceptibility, and hiding capacity. The remainder of this paper is organized as follows. In Section 2, the Recent Research on Digital Steganography is introduced. In Section 3, Ant colony optimization (ACO) is presented. In section 4, the Discrete Cosine Transform is introduced. In section 5, the Quantization Process is defined. In section 6, the proposed system is presented. In section 7, Implementation of the Steganography System is presented. The result analysis and a The Embedding process are presented in section 8,9 and section 10. Finally, the conclusions are drawn in Section 11.

2. Related Works

According to the authors in 2018, the ideal embedding parameter for picture watermarking with both robustness and imperceptibility can be found by using a time-efficient optimization strategy based on machine learning methods. Initially, a method for embedding a watermark is developed in the realm of discrete cosine transform to provide robustness against attacks on watermarks [17]. Sharma & Mir in 2022 used optimization parameters are then discovered using the Ant Colony method. Last but not least, the Light Gradient Boosting algorithm (LGBA) is used to forecast the ideal embedding conditions for the collection of fresh photos that will be watermarked. According to the findings, the suggested technique successfully watermarked images while also improving time enhancement [18].

Ali & Wook in 2020, offered to increase the image watermarks imperceptibility and robustness, a novel method for its insertion and extraction based on EDA-PSO is suggested. The discrete cosine transform domain is used for the insertion and extraction of the watermark (DCT). The process begins with the application of the Watson perceptual model to determine the ideal embedding position, followed by the modification of the embedded watermarks strength using EDA-PSO, followed by the establishment of a new fitness value as an evaluation criterion based on robustness and imperceptibility. The simulation results show that the proposed method successfully overcomes attacks such JPEG compaction, GLPF, additional Gaussian noise, additional impulse noise, and additional product noise. Additionally, it guarantees that the watermark is undetectable [19].

Liu & Ding in 2020, developed a new least significant bit and secret map techniques, 3D chaotic maps, such as 3D Chebyshev and 3D logistic maps, make it possible to encrypt images with exceptional security. This technique works by doing random insertion and selecting a pixel from the host image. Correlation coefficient, information entropy, homogeneity, contrast, image, histogram, key sensitivity, hiding capacity, quality index, mean square error (MSE), peak signal-to-noise ratio (PSNR), and picture fidelity are just a few of the metrics used to comprehensively evaluate the proposed technique [20].

Lu & Zhang in 2020, designed a robust JPEG steganographic algorithms are provided when the covert JPEG image is JPEG-compressed in a lossy channel to safeguard the encoded message. In terms of anti-compression performance, they often surpass more seasoned adaptive JPEG steganographic methods. The proposed encoders internal structure and codes differ from the conventional codes used in adaptive steganography. Finally, a new steganography is created by combining the proposed code, the relationship between coefficients, and the minimal distortion model [21].

Pan & Wu in 2022, suggested the issue of poor extraction capabilities of steganographic images under attack or interference was addressed by the proposal of a double-matrix decomposition image steganography technique with multi-region coverage. In order to incorporate the secret information, the cover picture is first altered using a multi-wavelet transform, and a concealed region that spans in its wavelet domain, a variety of wavelet sub-bands are selected. The results of the experiments demonstrate that the suggested technique has outstanding performance in hiding and quality of extracted secret image, and Secret information is recovered from steganographic images that have been subjected to various image processing techniques, demonstrating the suggested methods strong resistance to these attacks. [22].

Zhang & Zhong in 2022, presented a straightforward yet effective image steganographic model based on the discrete hadamard transform (DHT), a lightweight transform. The experiment findings show that the suggested technique provides great imperceptibility and security even in the dense embedding of 8 BPP in the case of only stego-image passed (Bits Per Pixel). The suggested system also passes numerous tests and demonstrates the desired robustness. The comparison evaluations show that our method is a workable and effective kind of image steganography [23].

Gaertner & Clark in 2005, proposed a new algorithm to increase robustness, the authors combined frequency domain and optimization techniques. The coefficients have been changed after the host image goes through an integer wavelet transform. To discover the ideal coefficients where to hide the data, the ACO optimization technique is utilized. Sample images and data have also been shown, demonstrating greater robustness and a high level of data embedding capabilities [24].

Zebari & Zeebaree in 2022, implemented a novel strategy based on a specific swarm intelligence algorithm has been examined. The goal of swarm intelligence algorithms is to achieve a robustness and quality of the item that has been utilized to conceal messages that is acceptable. This will be useful for future work to offer a more efficient and safe steganography technique based on swarm intelligence algorithms [25].

In order to prevent cloud service providers from having direct access to data presented an efficient model to protect sensitive data in the cloud, Thakkar & Srivastava in 2017, suggested technique used a Discrete Wavelet Transform (DWT)-based Steganography Scheme Based on Particle Swarm Optimization (PSO) (SVD) [26]. Singhal & Shukla in 2020, applied contemporary technique the host image contains the key elements of the secret picture. The stego picture is then encrypted using the Advanced Encryption Standard (AES) encryption method to ensure data secrecy [27].

3. Ant colony optimization (ACO)

In computer science and operations research, the ant colony optimization algorithm (ACO) is a probabilistic approach for resolving computing problems that can be reduced to finding effective paths through graphs. Artificial ants are used to show multi-agent techniques that were influenced by real-world ant behavior. The dominant paradigm is typically pheromone-based communication like that of biological ants. Local search methods mixed with artificial ants have become the favored method for many optimization projects using some type of graph. A group of optimization algorithms known as ant colony optimization are based on the activities of ant colonies. By navigating a parameter space that represents all potential solutions, artificial "ants" (such as simulation agents) find the best answers. While scouting their surroundings, real ants leave behind pheromone trails that guide one another to resources. To help more simulated "ants" find better solutions during subsequent simulation rounds, the simulated "ants" similarly record their positions and the caliber of their answers [28]. The embedding technique in this paper uses ant colony optimization algorithms to identify the optimum path on a weighted graph.

3.1. Ant Colony Optimization algorithms

In swarm intelligence approaches, this algorithm is a member of the family of ant colony algorithms and includes certain metaheuristic optimizations. Based on how ants navigate between their colony and a food supply, a variety of algorithms known as "ant colony optimization algorithms" strive to find the best path through a network. Since then, the basic concept has evolved to address a larger range of numerical issues, and as a result, a number of issues have surfaced that incorporate different facets of ant behavior [29].

By using simulated ants in place of real ants, the ACO algorithm seeks to mimic the behavior of actual ants. The optimum path on a graph must be found in order to solve the optimization problem. The computer-generated ants move about the graph that represents the issue to be resolved. The process of creating the solution is accomplished using a pheromone model, which is a collection of parameters linked to graph elements, the values of which are changed by the ants during runtime. Thus, as they go around the graph, the artificial ants develop solutions piece by piece. In this study, concealing spots were found using the ant colony optimization method based on the ant's graphs path. This method offered good security and robustness during the embedding process [30]. This concept from this work is regarded as one of the cutting-edge and brilliant concepts that will be applied to the implementation of the steganography system. Search optimal path as shown in the algorithm (1).

Algorithm 1. Simplified Ant System to search optimal path
Input: cover image.
Output: Select the optimal path

```
Step 1:  Initialize all pheromone values on edges or use the ones from previous iteration.
Step 2:  The ants should be placed in their beginning position.
Step 3:  while! end do
Step 4  When all nodes have been visited or after n cycles
         at every node, i
         for every ant on node i at time t do
Step 4  Move the ant to the next node using the option provided by
         equation 1.
         end
         end
Step 5  Calculate the distribution of pheromone trails along edges.
         End
```

Algorithm 1 simplified representation of the Ant System pseudo code illustrates the basic behavior of one iteration of the process. The following equation1 gives the equation that is used to determine how an ant will move at a particular state [41]:

$$P_{ij}(t) = \frac{T_{ij}(t)^{\alpha}.\eta_{ij}^{\beta}}{\sum_{j=1}^{n} T_{ij}(t)^{\alpha}.\eta_{ij}^{\beta}}$$

The coefficients alpha (α) and beta (β) are selected via experimentation and enable the user to regulate the compared significance of a trail against its apparent position. Where P_{ij} (t) is the probabilities of transferring via node *i* to node *j* at time *t*. The η_{ij} value is a space factor, T_{ij} (t) is the quantity of fragrances on the border that move via node *i* to *j* at time.

3.2. Optimal Path ACO

Despite being tiny, blind creatures, ants can navigate to their food source by taking the quickest route. To communicate with one another, they employ pheromone fluid and their antennas. The behavior of living ants on the hunt for food is what inspired ACO. Live ants keep past knowledge obtained by each ant to choose the best route to their food source. Ants take specific routes when searching for food. The fluid pheromones that each ant leaves behind help other ants that follow a similar path to identify it. The pheromone evaluation determines the best solution. Ants follow a random path while leaving pheromone behind. They will follow the route that has more pheromone left over when they return. The length of the journey affects the rate of pheromone evaporation. The path was longer the more pheromone disappeared [31].

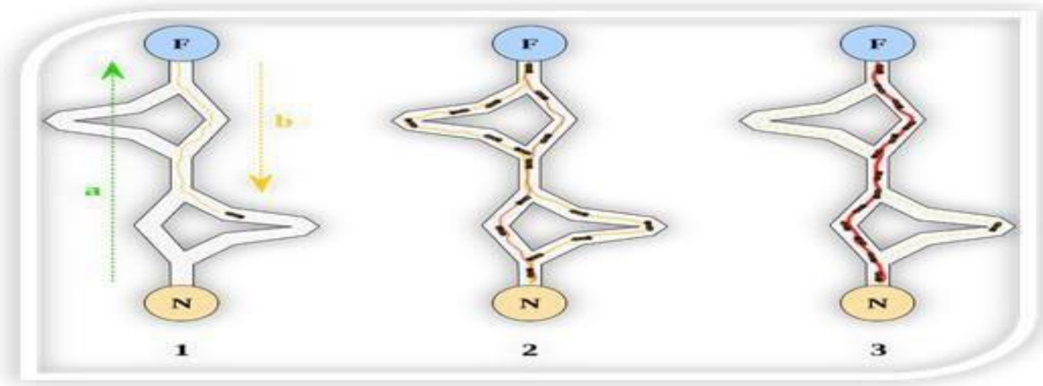


Figure 2. Ants Finding Optimal Path [32]

Based on the pheromone deposit, Figure 2 depicts how ants choose the best route between their colony and food. F stands for food source, and N for ant nest. Ants optimize their search by identifying the best solutions and taking into account earlier markings. The embedding procedure in this work will be based on the best path in graph number three, as indicated in the figure (3).

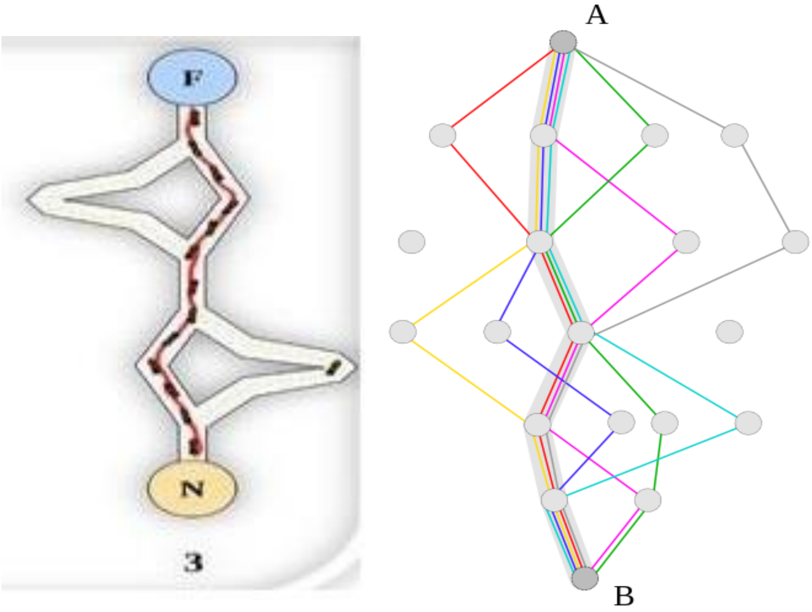


Figure 3. Select the Optimal Path in the proposer system

4. Discrete Cosine Transform

DCT, which divides an input image into non-overlapping blocks and converts them into blocks of coefficients, is the most well-known block transform. It is significantly simpler to incorporate picture information into the Middle frequency bands of an image thanks to the discrete cosine transform (DCT), which enables the division of an image into various frequency bands. The middle frequency bands are selected so that they limit removal through compression and noise attacks (low frequencies) while avoiding overexposing themselves to removal through high frequencies (high frequencies) [33]. The DCT has been effectively applied to JPEG and MPEG compression, focusing the energy of pixel data into the spatial domain. The information is amenable to loss quantization and compression in this format in a way that is almost imperceptible to the human visual system [34]. The (N ×N) array of integers produced by the two-dimensional DCT of an assumed N × N image is provided by [35].

$$T(u,v)=\alpha(u),\alpha(V)\sum_{r=0}^{N-1}\sum_{c=0}^{N-1}I(r,c)\cos\left[\frac{(2r+1)u\pi}{2n}\right]\cos\left[\frac{(2c+1)v\pi}{2n}\right] \quad \ldots\ldots\ldots \quad (2)$$

$$Where\ \alpha(u),\alpha(v)=\begin{cases}\sqrt{\frac{1}{n}} & if\ \alpha(u),\alpha(v)=0\\ \sqrt{\frac{2}{n}} & if\ \alpha(u),\alpha(v)=1,2,3,\ldots n-1\end{cases}$$

The inverse discrete cosine transform (IDCT) is given by:

$$I(r,c)=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\alpha(u),\alpha(v)T(u,v)\cos\left[\frac{(2r+1)u\pi}{2n}\right]\cos\left[\frac{(2c+1)v\pi}{2n}\right] \quad \ldots\ldots\ldots \quad (3)$$

Where u,v varies from 0 to n-1.

According to this interpretation of the DCT, minimizing the size of the 64 numbers will cause the unimportant image information to be lost. It's possible that this will not significantly lower the image quality. This is not usually effective. In order to minimize the size of each of the 64 numbers, a separate Quantization Coefficient (QC) is often applied [36]. The result of the DCT is the square (N×N) array T(u,v) of real numbers, the coefficient T(0,0) is called the "DC Coefficient" and the remaining are called the AC Coefficients.

5. Quantization Process

The method of quantization involves removing specific visual data while preserving the overall visual impact. By lowering the precision of the integer, quantization can cut down on the amount of bits required to hold an integer value. Better compression is achieved through quantization. Bit reduction enhances bandwidth, lowers implementation costs, and minimizes the amount of storage space required [36].

Each computed (N×N) matrix of DCT coefficients is quantized. The information loss happens at this stage. The appropriate number from the particular quantization table used is divided by each number in the DCT coefficient matrix, and the result is rounded to the nearest integer. In reality, not many users have the time or knowledge to experiment with such a wide range of parameters, hence JPEG software often employs the two strategies listed below:

1. Default quantization tables were the elements in the table generally grow as we move from the upper left corner to the bottom right corner as in Table (1).

Table 1: Sample Quantization Table

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

2. A simple quantization table *Q* is computed, based on one parameters *R* specified by the user. A simple expression such as $Q_{ij}=1+(i + j)*R$ guarantees that the quantization coefficients *(QCs*) start small at the upper left corner and get bigger toward the bottom right corner [38].

6. The Proposed System

In order to avoid raising questions about the image, this work tests techniques to conceal color image steganography inside larger color images without damaging them. Ant colony optimization (ACO) and discrete cosine transforms (DCT), a transformation technique, are two strategies included in the suggested system to increase its resistance to attacks. We examine both approaches using the dataset and contrast the paper findings with a different paper result based on a hiding mechanism. The algorithms for embedding and extracting data are split into two components in the systems. The optimal graph map, which represents the locations of the coefficient in the cover image to disguise the embedded image, is used in the first approach to embed employing ant colony improvement.

As opposed to other steganography techniques, the second method (embedding using discrete cosine transform) relies on the substitution of bytes blocks from processed embedded picture into processed cover image. Based on the first block in the best path in a graph, the existing block in the processed embedded image is moved into a related block in the processed cover image. High possibility of block matching is made possible by the discrete cosine transform (DCT). Real numbers are produced when a block of (N ×N) pixels is converted to (N× N) coefficients, but real numbers are converted to integers via the quantization method. The proposed systems are divided into two sections: the extraction and embedding portions. The suggested system has a key that is utilized to extract the embedded image from the embedded picture during the embedding process, which requires the embedded image and the cover images as input and produces the embedded image as the output. The method uses the embedded image as input for the extraction procedure and produces the stego image from the embedded as an output. The fundamental procedures for the suggested system and its algorithms are thoroughly discussed in this study.

7. Implementation of the Steganography System

The fundamental issue in hiding one image in another host image is the volume of data that must be handled, necessitating the use of a unique data embedding technique to ensure sufficient capacity, transparency, and robustness. The discussed stego-system assumes a transform domain embedding technique to provide a reliable system. It embeds a color picture into a color cover image. The main concept behind the suggested systems is to embed the secret image using ant colony optimization (ACO) and discrete cosine transform (DCT) to ensure sufficient capacity, transparency, and robustness. Finally, by examining the implications of applying these two techniques, we can see the results. In the primary method, the concealing sites in the cover image are determined during the embedding process using the ant colony optimization algorithm. As opposed to other steganography techniques, the second method (embedding using discrete cosine transform) relies on the substitution of bytes blocks from processed embedded picture into processed cover image. A similar block in the processed cover picture is inserted into the existing block in the processed embedded image. High possibility of block matching is made possible by the discrete cosine transform (DCT). Real numbers are produced when a block of (N× N) pixels is converted to (N× N) coefficients, but real numbers are converted to integers via the quantization method. The embedding and extraction components make up the two portions of the suggested systems. The suggested system has a key that is utilized to extract the embedded image from the stego picture during the embedding process, which uses the embedded and cover images as input and produces the stego image as the output. The method uses the stego image as input for the extraction process and produces the embedded image from the stego as an output. The main steps for the proposed system and its algorithms are discussed in details.

8. The Embedding process

The suggested systems embedding process is divided into many stages, including test, transform, Finding Optimal Path based on (ACO), matching, key creation and substitution, inverse transform, key encryption, and concealment. The Figure depicts the embedding process (4).

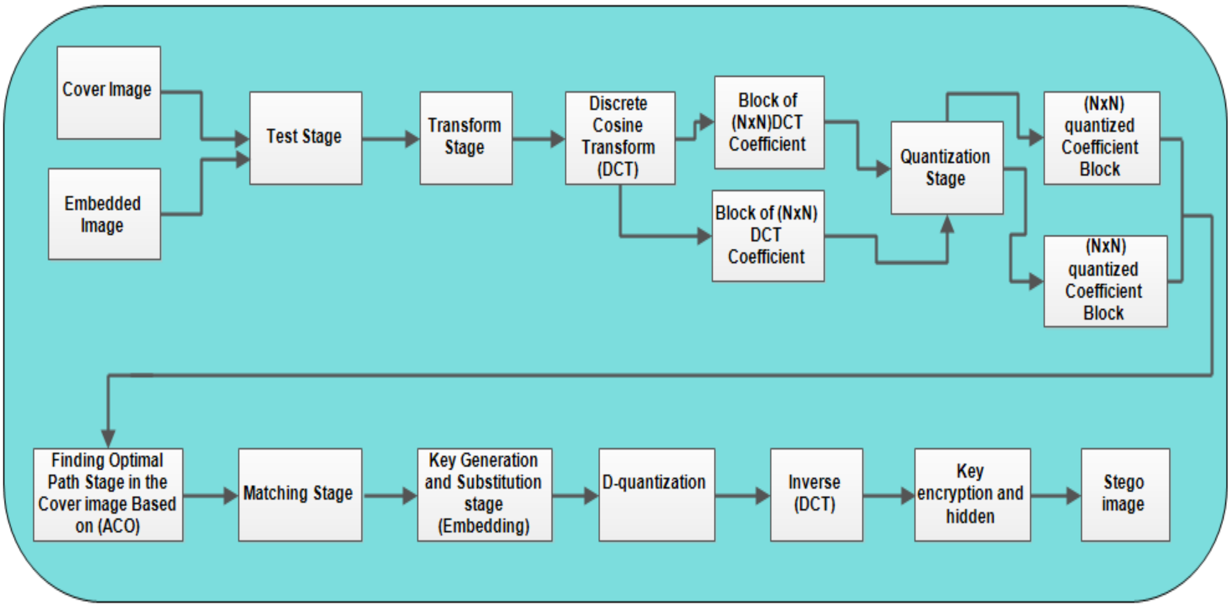

Cover Image
Embedded Image
Test Stage
Transform Stage
Discrete Cosine Transform (DCT)
Block of (NxN)DCT Coefficient
Block of (NxN) DCT Coefficient
Quantization Stage
(NxN) quantized Coefficient Block
(NxN) quantized Coefficient Block
Finding Optimal Path Stage in the Cover image Based on (ACO)
Matching Stage
Key Generation and Substitution stage (Embedding)
D-quantization
Inverse (DCT)
Key encryption and hidden
Stego image

Figure 4. The Embedding Process using (DCT)

8.1. Tests Stage

The first step in the embedding algorithm is the test-stage. This stage includes two tests.

8.1.1. First Test

The algorithm will compare the size of the cover image and the embedded image during this test; the embedded image must be smaller than the cover image. The embedded should be precisely one-fourth the size of the cover.

The size of both the cover and embedded image is calculated by using the equation (4).

$$\left.\begin{aligned}S_e &= embeddedhgt \times embeddedwide\\ S_c &= Coverhgt \times Coverwide\end{aligned}\right\} \quad (4)$$

8.1.2. Second Test

This test determines whether or not the chosen cover picture is appropriate to hold the embedded image. This test utilizes similarity and dissimilarity; it calculates measurements based on the histogram calculations for embedded and cover pictures using the two equations (5) and (6).

$$S\{H(E),H(C)\} = \frac{\sum_{j=1}^{n} \min\{h_j(E), h_j(C)\}}{Nc \times Mc} \quad (5)$$

$$D\{H(E),H(C)\} = \sum_{j=1}^{n} \left| \frac{h_j(E)}{Ne \times Me} - \frac{h_j(C)}{Nc \times Mc} \right| \quad (6)$$

Equation (5) is used to measures the similarity between the histogram of the embedded image (E) and the cover image (C), where h_j *(E)*: is the number of elements which have the color (J) in the image (E).

h_j *(C)*: is the number of elements which have the color (J) in the image (C).

$M_C \times N_C$: is the size of the cover image.

Equation (6) is used to measure the dissimilarity between the histogram of the embedded image (E) and the cover image (C), where $M_e \times N_e$: is the size of the embedded image (E). The two tests as shown in the algorithm (2).

Algorithm 2. Test Algorithms
Input: embedded image and cover image.
Output: return false or true.

```
Step 1: Input the embedded image and the cover image.
Step 2: Calculate the size of the embedded image and the cover image by performing the
        equation (4).
Step 3: Compare between the size
        If $S_e > S_c$ then
              Good size: = false.
            Else
              Good size: = true then
Step 4: if good size = true then
        Go to step 5
        Else
        Input another cover image and go to step2
Step 5: Calculate the similarity between the embedded and the cover images by performing the
        equation (5)
Step 6: Calculate the dissimilarity between the embedded and the cover images by performing
        the equation (6)
Step 7: If the similarity > 0 and dissimilarity <    then
        Return true.
        Else
        Return false.
Step 8: End.
```

8.2. Transformation Stage (DCT)

The proposed system embeds using discrete cosine transformation process. Applied to both cover and embedded images.

8.2.1 DCT of Embedded and Cover Image

The embedded image and cover are both transformed after an appropriate cover is selected to house them. The embedding and cover images are divided into (N ×N) pixel blocks. N stands for (8×8) pixels that will be converted one block at a time. To obtain blocks of (N×N) coefficients of real numbers, which reflect the frequency of colors within a block, these blocks will be transformed using DCT. DCT for the blocks is calculated using equation (7).

$$\left.\begin{aligned} & T(u,v)=\alpha(u),\alpha(V)\sum_{r=0}^{N-1}\sum_{c=0}^{N-1}I(r,c)\cos\left[\frac{(2r+1)u\pi}{2n}\right]\cos\left[\frac{(2c+1)v\pi}{2n}\right] \\ & \textit{Where } \alpha(u),\alpha(v)=\begin{cases}\sqrt{\dfrac{1}{n}} & \textit{if } \alpha(u),\alpha(v)=0 \\ \sqrt{\dfrac{2}{n}} & \textit{if } \alpha(u),\alpha(v)=1,2,3,\dots n-1\end{cases}\end{aligned}\right\} \qquad (7)$$

Each (N × N) block of DCT coefficients is computed, and then the real values are quantized into integers. Equation divides each number in the DCT coefficients block by the corresponding number in the quantization matrix, then rounds the result to the nearest integer (8). The algorithm for the discrete cosine transform is shown in algorithm (3).

$$\mathrm{QDC} = Round\left(\frac{\mathrm{DCT\ Coeff}}{\mathrm{Q_{ij}}}\right) \tag{8}$$

Where:
QDC: represent the quantization DCT coefficients.

The quantizer value based on one parameter Q is specified by the user. The equation (9) is used to calculate the quantization matrix elements with different qualities and compression.

$$Q_f = \begin{cases} \dfrac{(100-Q)}{50}\text{‘}Q \quad \geq 50 \\[2em] \dfrac{50}{Q}\text{‘}Q \quad < 50 \end{cases} \tag{9}$$

Where:
Q_f: represent the quality factor
Q: represents the quality.

The quality of the quantized image will be high wherever the Q value is higher, and depending on the value chosen, the error between the original and reconstructed image will also be low. Therefore, the systems value of Q will be chosen to retain the quality of the reconstructed image as close to 50% as possible in order to maintain quality and compression ratio. This shall also give more of a chance for the blocks of both the cover and embedded to match as closely as possible.

Algorithm 3. Discrete cosine transform algorithm.
Input: the embedded and the cover image.
Output: Image contains of (N × N) Quantized blocks of coefficients.

```
Step1: Read block size (n)
Step2: Define DCT transformation function for one block :
       $T * X * T^{-T}$
Step3: Apply block procedure on the image decimating the image by 8x8 blocks.
Step4: Obtaining the results from step 3 into a DCT Coefficients for each block.
Step8: Calculating the Quantization of DCT Coefficients for each block
       by: performing equation (8), dividing each block by corresponding number
       in the quantization matrix and round it to the nearest integer.
Step9: End.
```

8.3. Finding Optimal Path stage in the cover image

An ant colony optimization approach is used to find the embedding location in the cover picture that represents the ideal path after collecting the block of (N×N) coefficients. For the block matching stage, we have already prepared the locations and paths. The selection of the embedding location has become much more complex because to the inclusion of the ant colony optimization algorithm in the embedding process. This complexity extends beyond the systems confidentiality and robustness, making it more difficult for an attacker to extract the secret image. This stage increases difficulty and robustness because the attacker cannot recover the secret image because he lacks information of the embedding process, including which path or algorithm was used to embed the data in the cover image.

8.4. Growth Algorithm

We have used Growth algorithm to encrypt stego-key (index matrix). Growth algorithm consists of one linear feedback shift register (LFSR) whose length, feedback function is and output function shown in the following Figure (5).

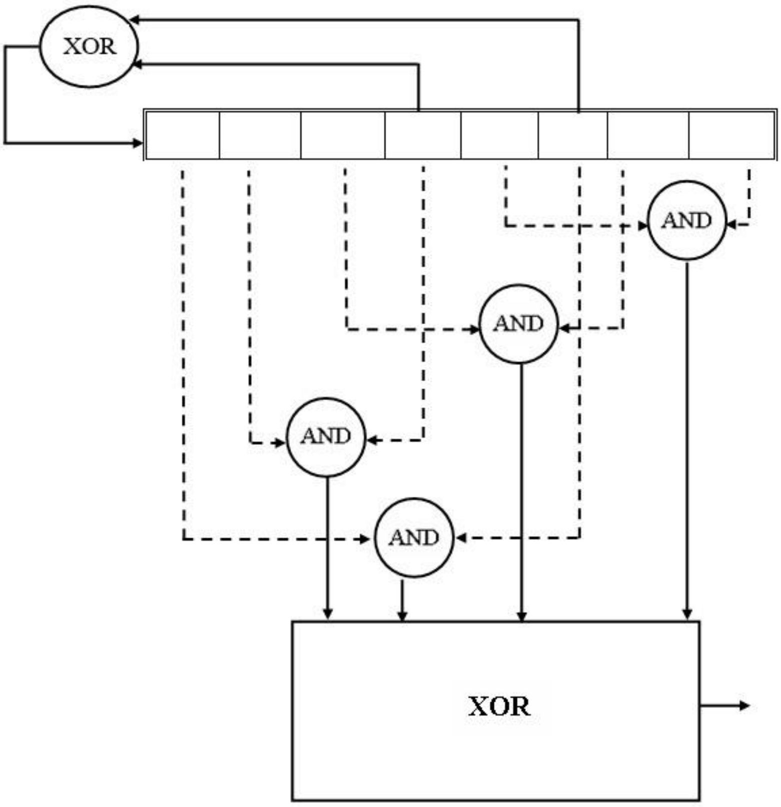


Figure 5. Growth Algorithm [39,40]

The Growth algorithm consists of a linear feedback register that is fed to the first register each time it is moved, with (XOR) logic on its fourth and sixth bits.

The contents of other stages of the register are processed as follows:

The (And) operation is shared by the eighth stage and the fifth stage.

The (And) operation is shared by the third stage and the seventh stage.

The (And) operation is shared by the first stage and the sixth stage.

The (And) operation is shared by the second stage and the fourth

Results from the four operations mentioned above share the (XOR) operation.

After that, the shifting operation produces a new key bit each time. We produce as many key bits as required to provide an index matrix stream-length stream of ciphering key matches. Then we (XOR) both keys together to have our ciphered index matrix.

8.5. Key Encryption Stage

The method will encrypt the key (index matrix) for each color that was produced as a result of the generating stage in this step. To improve the security (i.e., increase the robustness) of the entire stego-system, a secret key ciphering strategy is suggested.

A growth method that encrypts the secret key (index matrix) relies on seeding the cipher register with fresh initials, which is then converted to a bit stream and ciphered using (XOR) with the generated key from the Growth algorithm, is obtained from the first byte of the stego picture to generate sequence key to encrypt and hide the key (index matrix). Algorithm provides an illustration of the ciphering-stage algorithm (4).

```
Algorithm 4.  Ciphering- Stage
 Input: index matrix.
 Output: key after encryption.
  Step1: (f1), which consists of the key to read from.
  Step2: (f2) empty cipher array.
  Step3: Calculate the size of (f1).
  Step4: Read first stego image byte (ik).
  Step5: Insert (ik) into growth register.
  Step6: For all size of (f1):
              Shift and generate (f2).
   Step7: XOR f1 and (f2).
   Step8: Store result in (f1).
   Step9: End.
```

8.6. Matching stage

Here, a modern method of concealing a secret image in a cover picture is suggested, along with the possibility of hiding. Because it depends on secret information sent separately, which indicates the block size and the location of the initial seed of the decipher key, the recipients cannot recover the hidden secret without those pieces of information. Additionally, this method is the most resistant to changes and processing being applied to the carrier image, preserving the hidden information (embedded). At this stage, blocks from the embedded and cover images are compared using the (Goodness of Fit) equation (10) to determine the placements of the embedded image blocks inside the cover image.

$$S = \sum_{i=1}^{n} \left(0.25 - \left(\frac{\left(\sqrt{(E_i - C_i)^2}\right)}{\max elements} \right) \times 0.25 \right) \tag{10}$$

Where:
E: represent the value of embedded image pixels.
C: represents the value of cover image pixels.
Max elements: represent the highest value of pixels to embedded or cover image.

The ideal path determined by ant colony optimization on all cover blocks will determine the search spaces domain. The embedded block will be removed from the search space at the beginning and again after a suitable block is discovered to hold it. Starting with all cover blocks being marked as false, the method chooses the first block from the embedding blocks and verifies its identification. If no identical block is discovered at all, the algorithm uses equation (16) to locate the best fit block, and if one is discovered, it searches the embedding blocks to find blocks that are comparable to the tested embedded block. (Find blocks that are comparable.)

The algorithm starts (in start) looking for cover blocks with the same pixel value (100%) before moving on. If it doesn't succeed in the embedded blocks (S=1), it looks for the cover block that has the value that is closest to one in the embedded blocks (value(S) Converge to one). In all scenarios, it will hold the embedded block (which requires a high matching value) in the index matrix by saving the position of an appropriate cover block there. To discover the appropriate blocks in the cover to hold them, conduct this search on every embedded block. The algorithm will attempt to reduce the number of embedded blocks inside the cover image by comparing the tested and found embedded block to other embedded blocks to find the similar blocks to it, and then it will give all the similar blocks the same location as the tested block when performing the first match (between the first embedded block and other cover blocks) and deciding the best block to hold it. The size of the embedded image will also decrease as a result of this process, which is useful for expediting the search for blocks whether similarity or convergence is being sought after. This process is carried out to reduce the number of blocks that must be checked throughout the loop to reduce search time.

As a result of reduced distortion in the stego-image, concealing chances increase when the number of blocks of embedded picture are located in the same place in the cover image. The matching algorithm is displayed in algorithm (5).

Algorithm 5. Matching Algorithm
Input: The embedded image and cover image consist of (N × N) Quantized blocks coefficients.
Output: array contains the assigned block numbers (index matrix future key).

```
Step1: Compare between blocks of embedded image with blocks of cover image by:
      For i= 1 to embedded block N. do
                Get sub block [i]
    For j= 1 to cover block N. do
                Get sub block [j]
    Perform identity test
    if succeeded
        -   Stores location of cover block into the index array
        -   Drop block from the search space.
        -   Perform a search into the embedded blocks to find similar blocks
                if found
               - Store location of the cover block into the index array
               - Drop found similar embedded blocks from further search.
                 If not found next [i]
   If not succeeded goto step 2
Step2: Search for greater match from step 2 using equation (10) by:
          Find maximum S in the cover blocks
          Save the block number [j] in index matrix
Step3: Compare between blocks of embedded image themselves by:
                Get sub block [i]
For j=i+1 to last embedded block do
                Get sub block [j]
                If block [i] <> block [j] then
                Sameblock: = false.
                Else
                Sameblock: = true.
Step4: If sameblock=true then
          Save the same index (which is given to block1) to the similar block (block2).
Step5: Get next embedded block next [i]
Step6: End.
```

8.7. Key Generation and Substitution Stage

After the matching stage is finished, the suggested approach produces a position index matrix that is said to be the secret to replacing embedded blocks with cover blocks. The produced key is regarded as a crucial component of the replacement stage. To strengthen the security of the suggested system, it is encrypted and concealed inside the stego-image.

The substitution stage of the embedding process, which involves replacing each embedding block with the equivalent block of the cover, is crucial. The approach will insert the block of embedding image rather than the cover block using the best path created by the ant colony optimization after determining the best block of cover image to hold the block of embedding image (read the position of cover block from the index matrix). The algorithm for replacement will be as displayed in algorithm (6).

Algorithm 6. Substitution Algorithm in the (DCT) method
Input: Embedded and cover image and index matrix.
Output: Stego-image.

```
    Step1: For i= 1 to key element N. do
    Step2: Read the position from k[i]
    Step3: Get block [i] from embedded
    Step4: Substitute cover block [k[i]] with block[i] of embedded.
    Step5: End.
```

8.8. Inverse Discrete Cosine Transform Stage

When the algorithm receives an image with an embedded picture and a cover image, it will then execute inverse quantization (de-quantization). The quantization matrixs associated numbers are multiplied by each number in the DCT quantized block. The algorithm will next go through an inverse transformation stage (IDCT) to recover the original pixel count (original image). This image is known as a Stego-image. Thus, as can be seen in Figure (6), the stego-image somewhat resembles the original cover.

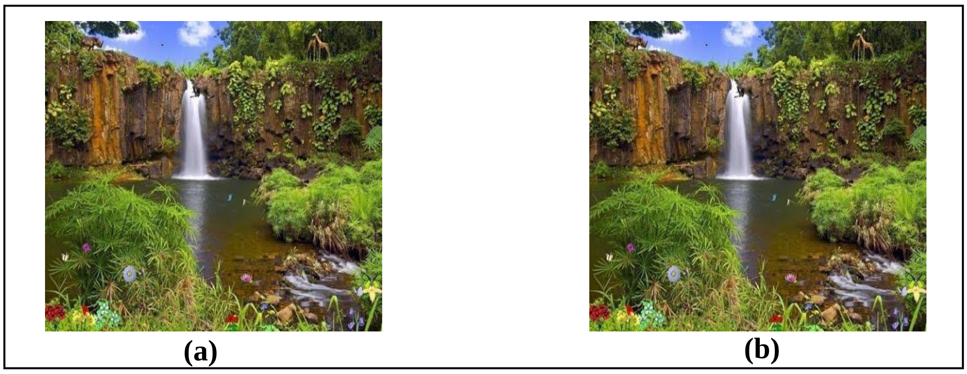


Figure 6. The similarity between cover image and stego-image. (a) Cover image (b) Stego- image (PSNR= ٥٥,٥٩٤١)

8.9. Key Encryption Stage

The method will encrypt the key (index matrix) for each color that is produced in this stage. To improve the security (i.e., increase the robustness) of the entire stego-system, a secret key ciphering strategy is suggested.

A growth method that encrypts the secret key (index matrix) relies on seeding the register with fresh initials. is constructed using the sequence key that is extracted from the first byte of the stego picture to encrypt and conceal the key (index matrix). The key (index matrix) is then converted to a bit stream and ciphered using (XOR) with the generated key from the growth algorithm is previously demonstrated in Figure (5). The ciphering-stage algorithm will be as displayed in algorithm (7).

```
Algorithm 7. Ciphering- Stage
Input: index matrix.
Output: key after encryption.

Step1: (f1), which consists of the key to read from.
Step2: (f2) empty cipher array.
Step3: Calculate the size of (f1).
Step4: Read first stego image byte (ik).
Step5: Insert (ik) into growth register.
Step6: For all size of (f1): Shift and generate (f2).
Step7: XOR f1 and (f2).
Step8: Store result in (f1).
Step9: End.
```

8.10. Key Hiding Stage

By employing the (least significant bit) approach for each stego-image pixel, where the secret key bit is distributed linearly with a step over the stego-image pixels, the algorithm will hide the key at this stage after it has been encrypted inside the stego-image. The Hide-Stage Algorithms are displayed in Algorithm (8).

```
Algorithm 8. Hide- Stage
Input: key after encryption.
Output: Stego image contains the key.

Step1: Read (f1).
Step2: Open stego-image.
Step3: Drop first row and column of stego-image
Step4: Calculate the step by dividing ciphered key bits on the rest of stego-image size
Step5: Inject bit by bit of the cipher key to the pixels of stego-image considering the step.
Step6: End.
```

8.11. Extracting embedded image from Stego-image using (DCT)

The extractor needs both the stego-image and the stego-key to begin the extraction process of the embedded picture. In Figure (7), the extraction procedures are described. The process starts with the extraction of the Stego-Key. The least significant bit approach, or extraction from stego-image pixel, should be used by the extractor to extract the key from the stego-image in order to extract the embedded image. The Growth algorithm is used by the extractor to create the encipher key, which is crucial in the decoding stage and provides access to the hidden key by applying XOR to the extracted key, after you have finished extracting the hidden key from the first byte of the stego image. The extractor has all the necessary knowledge to extract the embedded image after cracking the key. Utilizing the location (index array) for each embedded block in the cover block, the extractor will extract the embedded image block. The embedded block is restored to complete the extraction. The index key is used to get back to the blocks original image arrangement. The embedded image is perfectly recreated when the original layout for the embedded block has been extracted by the extractor, who should then apply the opposite of the techniques used during embedding.

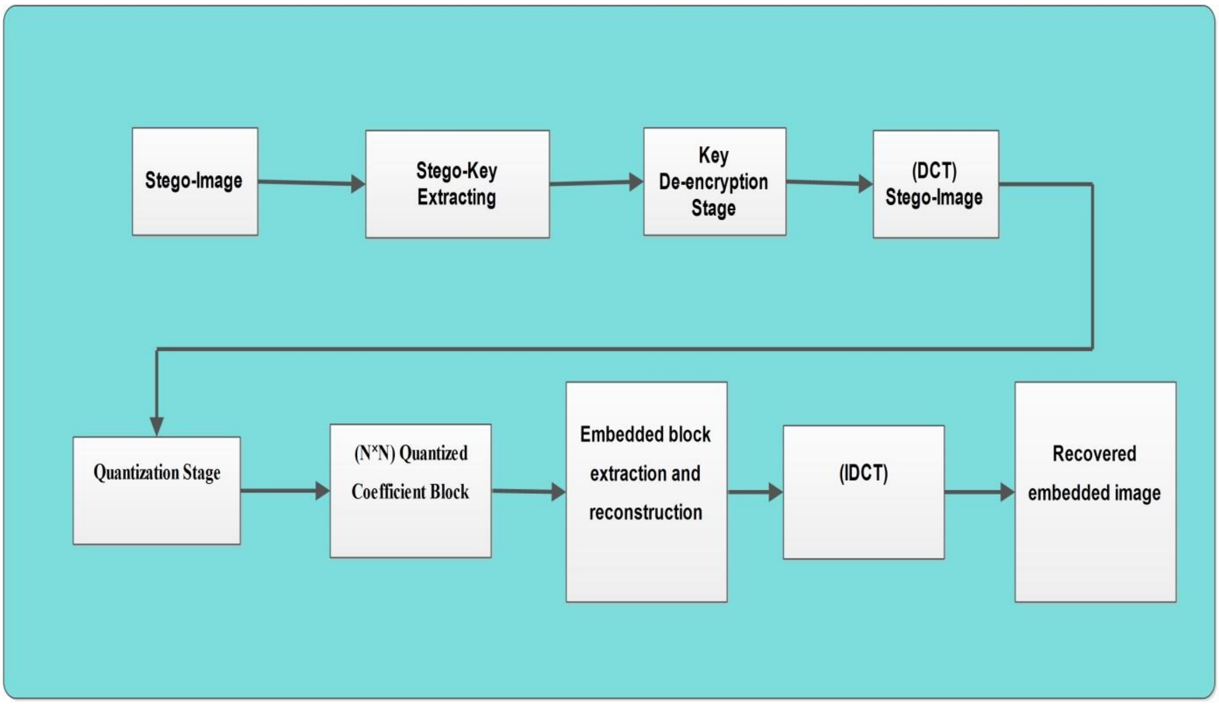


Figure 7. The Extracting Process

9. Results

The test results for the suggested systems are presented in this paper. First, the outcomes for the suggested systems embedding and extraction procedures using the ACO and DCT. Second, compare the two approaches. The proposed systems will be tested using a dataset, and the results will be discussed and compared to findings from previous studies.

9.1 Tests Dataset

The suggested systems will be put to the test using a set of cover and secret photographs. Each method must be used, and the corresponding results must be achieved. Figure (8) displays the data set. Pretest results are displayed in a table (2) along with correlation, similarity, and dissimilarity. As we will see throughout the course of this study, the pretests, which comprise correlation, similarity, and dissimilarity tests, will serve as our indications and a guide to determine which cover photographs should be used to attain the greatest results.

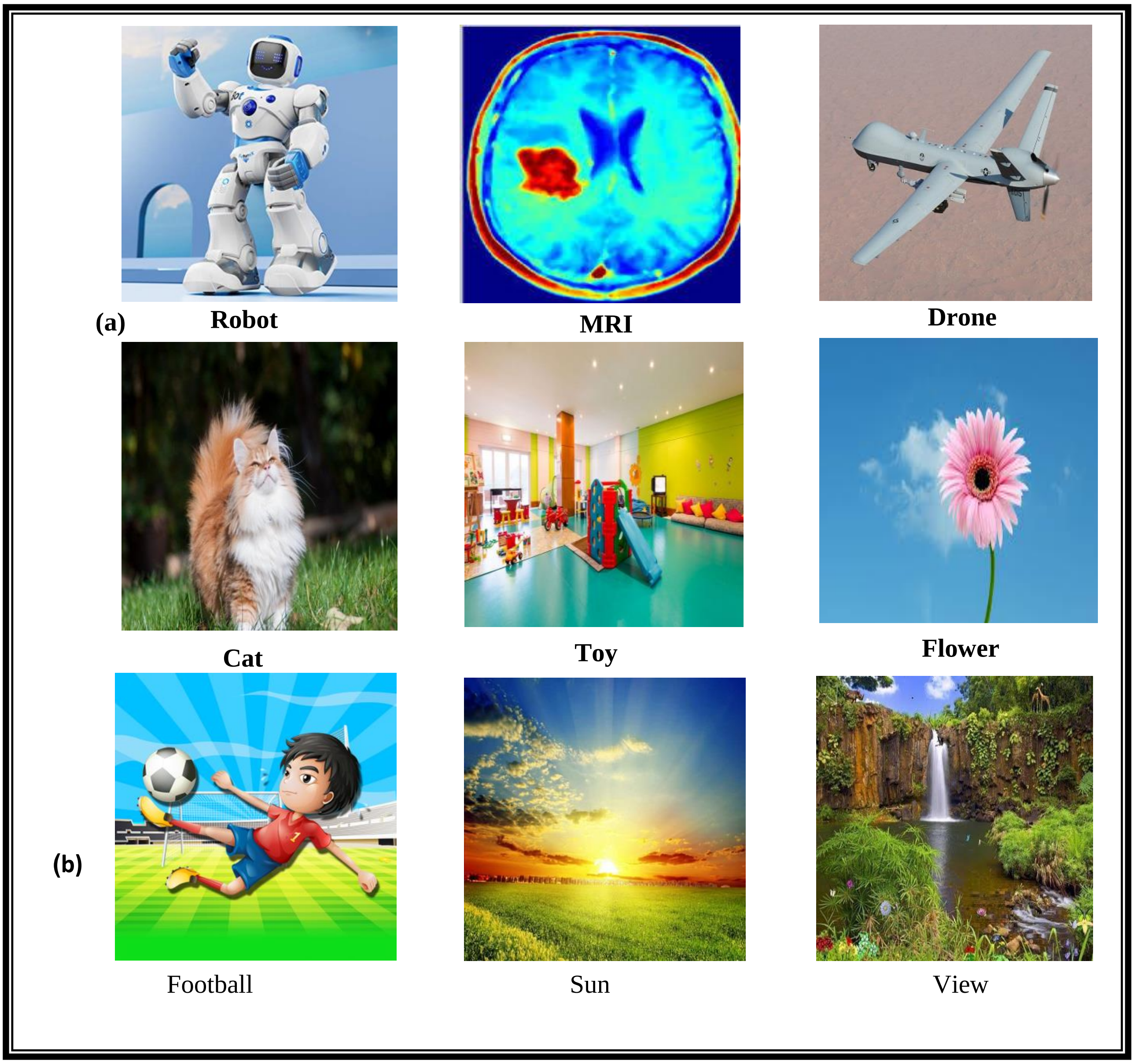


Figure 8. (a) the group of the secret images and (b) the group of Cover images.

Table 2: Dataset Pretests results

Cover Image	**Cat**		
Secret Image	Correlation	Similarity	Dissimilarity
Robot	0.0554	0.0617	0.9259
MRI	0.0526	0.0616	0.8897
Drone	0.0075	0.0619	0.9664
	Toy		
Robot	0.0137	0.060	0.8784
MRI	0.0398	0.0623	0.8977
Drone	0.0294	0.0621	0.9261
	Flower		
Robot	0.0119	0.0624	0.8185
MRI	0.0144	0.0617	0.8306
Drone	0.0599	0.0625	0.8399
	View		
Robot	-0.0647	0.0591	1.0698
MRI	0.0293	0.0607	1.0198
Drone	-0.0993	0.0596	1.0604
	Sun		
Robot	0.0844	0.0605	1.0400
MRI	-0.0470	0.0589	1.1101
Drone	0.0709	0.0604	1.0107
	Football		
Robot	0.0310	0.0607	0.7890
MRI	0.0658	0.0623	0.7715
Drone	4.7252	0.0625	0.8113

9.2. Implementation of the Proposed System applied ACO and DCT

This section presents the results obtained from objective test and histogram test between stego-image and cover image of the proposed steganographic system with encrypted secret key and hidden inside stego-image.

9.2.1. Objective tests

The results of objective tests are listed in table (3). This table contains the values of the objective tests applied to the stego-image and cover image. The group of secret images and the group of cover images are shown in Figure (8). Hence, the first column of this table contains the names of the cover image of size (256x256) and the secret image with (64x64) size. The sign (+) means embedded in this cover image. The 2nd, 3rd and 4th columns refer to the (PSNR), Correlation (Cor.) and MSE which are calculated respectively of the stego-image with respect to cover image. Knowing that the reconstructed images of all tests are the same as the original images; in other words, the correlation test is approaching unity (Cor ~= 1) of reconstructed image with respect to original image.

Table 3: the results of the PSNR, Correlation and MSE tests

Stego-image	PSNR/dB	Correlation	MSE
Cat + Robot	٥٠,٩٨٦٣	٠,٩٩٥٢٥	٠,٣٣٨٣
Cat + MRI	٥١,٣٠٠٥	٠,٩٩٦٥١	٠,٦٣٨٩
Cat + Drone	٤٨,٩٧٢٣	٠,٩٩٥٢٤	٠,٣٧١٧
Toy + Robot	٤٨,٩٤٩٨	٠,٩٩٨٢٣	٠,٤٢٥٥
Toy + MRI	٤٩,٠٣٤٣	٠.٩٩٨٦٢	٠,١٢١٨
Toy + Drone	٤٣,١١٢٦	٠,٩٩٨٢٩	٠,٠٤٢١
Flower + Robot	٥٠,١٢٥٩	٠,٩٩٠٤٦	٠,٠٨٥٦
Flower + MRI	٤٥,٥١٨٣	٠,٩٩١٢٧	٠,٤١٢٦
Flower + Drone	٤٧,٩٠٥٨	٠,٩٩٠٥٣	٠,٩٢٧٦
Football + Robot	٥٥,١٧٢٥	٠,٩٥٦٤٥	٠,٠٧٢٨
Football + MRI	٥٤,٠٦٠٦	٠,٩٦٧٢٥	٠,٢٤٩٣
Football + Drone	٥٢,٦٢١٥	٠,٩٧٦٠٨	٠,٩٣١٧
Sun + Robot	٥١,٥٨٦١	٠,٩٨٦٥٣	٠,٣٣٦٣
Sun + MRI	٤٩,٧٧٧٣	٠,٩٩٧٣٩	٠,٠٦١٧
Sun + Drone	٤٨,٧٢٠٢	٠,٩٩٦٦٤	٠,٩٩١٦
View + Robot	٥٥,٥٩٤١	٠,٩٦٦١٧	٠,٩٨٨١
View + MRI	٥٣,٥٧٩٥	٠,٩٥٦٩٥	٠,١٦٣٦
View + Drone	٥٣,٢٩١٣	٠,٩٦٥٨٩	٠,٦٣٧٢

As shown in the Table (3), Football and the view images give the best objective test, the first reason is that the cover image was selected during the test stage as being appropriate to hold the embedded image. This test applies measurements of similarity and dissimilarity based on how the histogram for the embedded and cover images is calculated. The second reason for the matching stage is that, using the (Goodness of fit) equation (10) to match blocks of the embedded and cover images, the position of the embedded image blocks inside the cover image is determined.

However, to provide a high probability of block matching, the discrete cosine transform (DCT) is employed, because there is less distortion in the stego-image, concealing chances are higher when the number of embedded image blocks are in the same location in the cover image.

9.2.2. The Histogram Test Results

This test will be performed in order to highlight the essential components of the suggested method and the careful image selection. According to this test, the hidden image has no impact on the modified image. When compared The cover image histogram and the modified image histogram are similar in appearance. as shown in Figures (12 ,13,17,18).

<u>Case-study 1</u>: By using proposed system 1, this test can be implemented with these chosen images: the embedded image is (64x64) true color image and the cover image is a true color image with size (256x256).

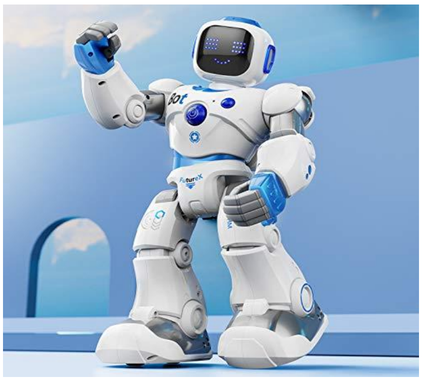


Figure 9. the secret image (64x64) pixels.

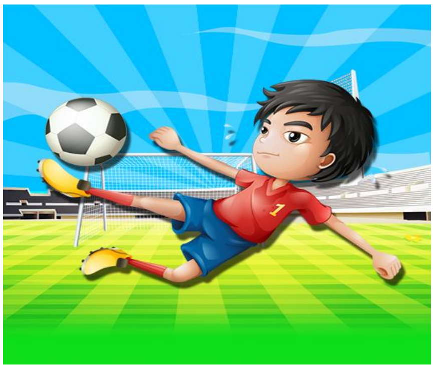


Figure 10. the cover image (256x256) pixels.

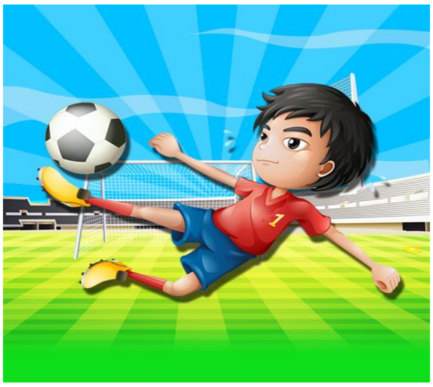


Figure 11. the stego image (256x256) pixels.

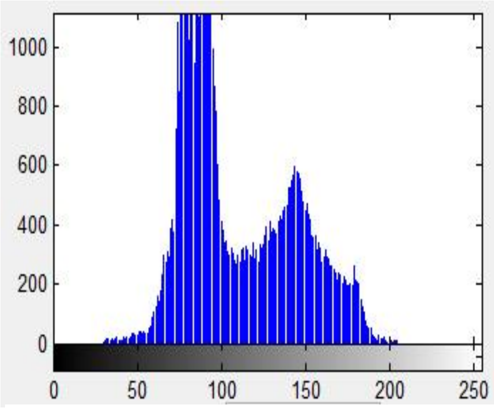


Figure 12. Histogram for the cover image.

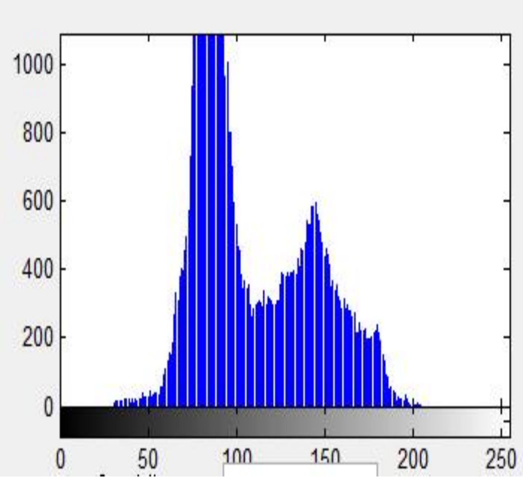


Figure 13. Histogram for the stego image.

<u>Case-study 2</u>: According to proposed system 1, this example can be carried on. Both images are true color images again. Can be implemented with change of the secret image and the cover image

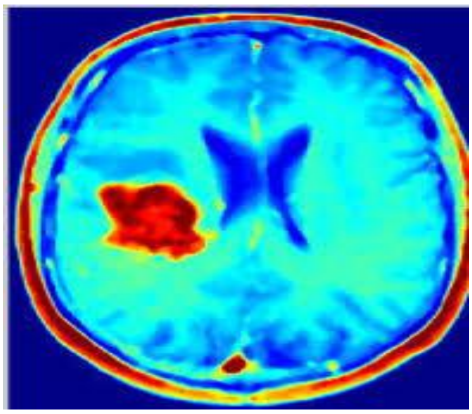


Figure 14. the secret image (64x64) pixels.

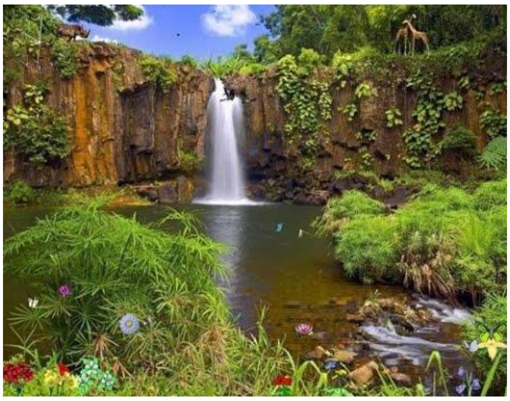


Figure 15. the cover image (256x256) pixels.

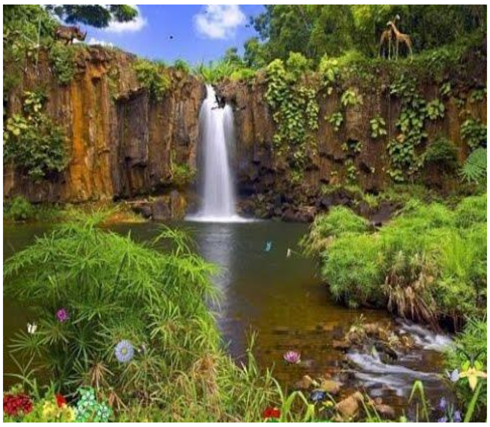


Figure 16. the stego image (256x256) pixels.

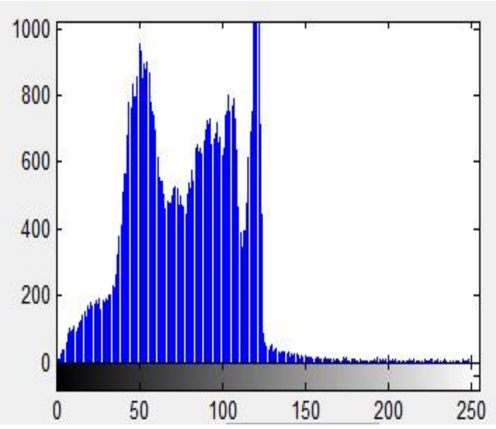


Figure 17. Histogram for the cover image.

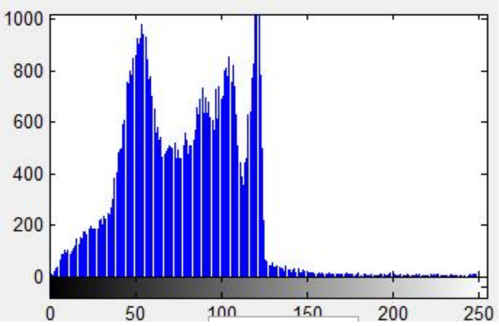


Figure 18. Histogram for the stego image.

The findings demonstrate that there is no difference between the statistical analysis of the cover image and the stego-image. To demonstrate the resemblance, the statistical data for the cover image will be subtracted from the stego image distribution as illustrated in Figures (19).

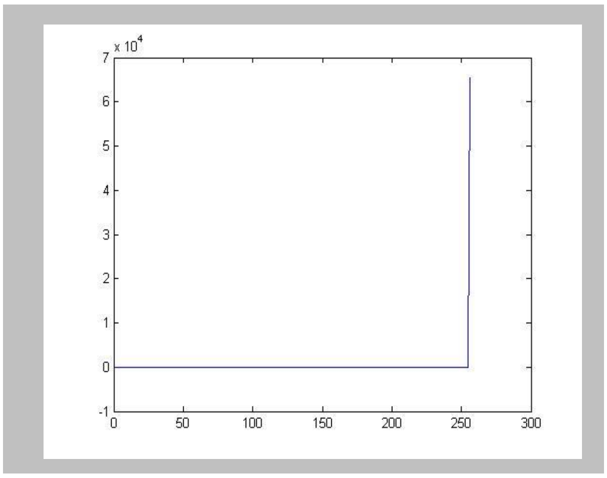


Figure 19. Images histogram subtraction results.

To extract the embedded image using the results of (Football) and (View), images as cover images in Tables (3); Tables (4) and (5) are constructed. The first column represents number of cases; the 2nd and 3rd represent the Stego-image and original embedded image; the 4th, 5th and 6th represent the PSNR, Cor and MSE test. For the recovered embedded image (secret image) from the stego-image.

Table 4: The PSNR, Correlation and MSE results of extracted secret image from Flower image.

Stego-image ←→ For recovered secret image ←→

Cases	Stego-image	Original embedded image	PSNR/dB	Cor.	MSE
1	Football	Robot	٣٧,٣٤٤٩	٠,٩٩٣٣٧	٠,١٥١١
2	Football	MRI	٣٦,٠٤٨	٠,٩٩٠٤٧	٠,١٨٢٤
3	Football	Drone	٣٤,٩٦٦٨	٠,٩٨٦٥٧	٠,٧٣٦١

The recovered images for the previous cases will be viewed in Figures (20), (22) and Figure (24). Each figure compares between the original secret image and recovered embedded image. Nearly identical histograms exist for the recovered embedded image and the original image (secret image) as shown in Figures (21), (23) and Figures (25).

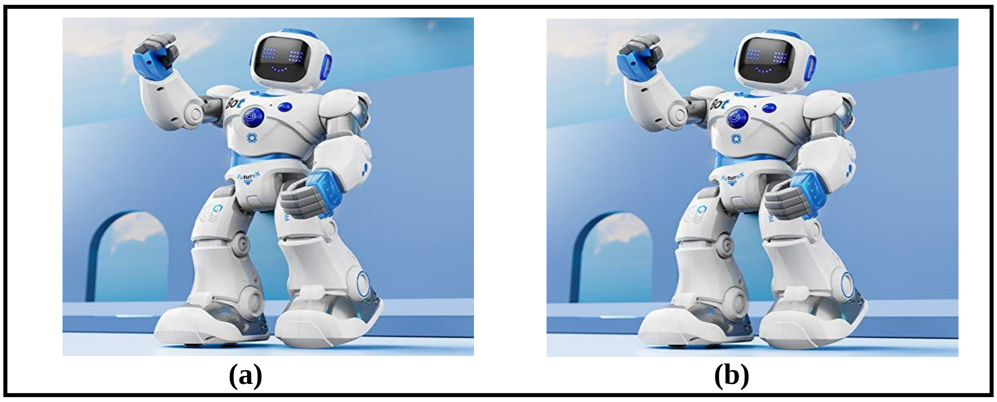


Figure 20. Recovered embedded image (secret image) at case 1. (a) Original embedded image (b) Recovered embedded image

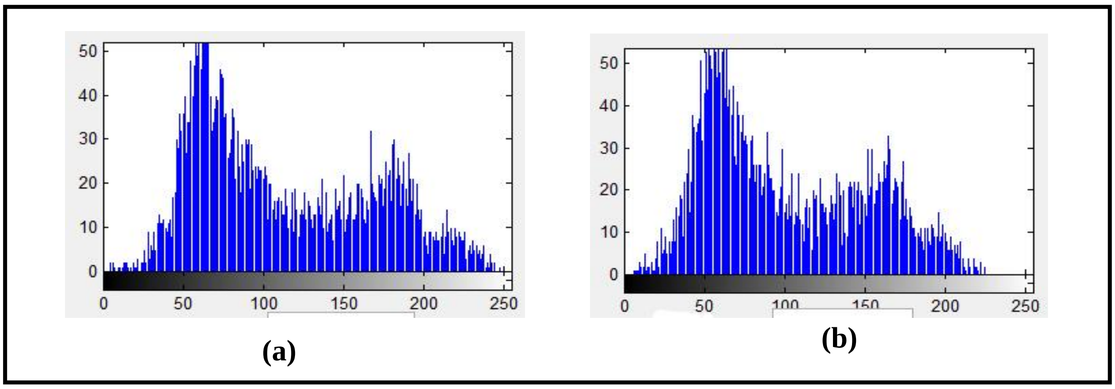


Figure 21. (a) The histogram for the Original embedded image (b) The histogram for the Recovered embedded image.

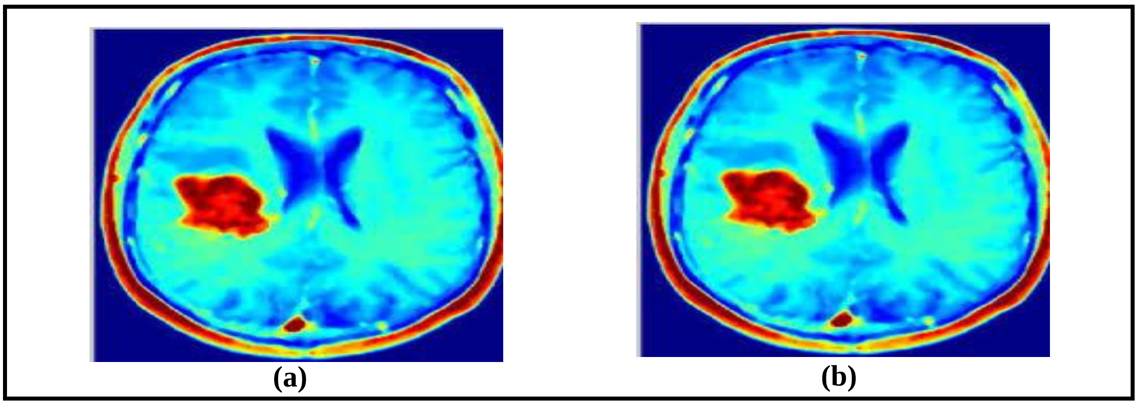


Figure 22. Recovered embedded image (secret image) at case 2. (a) Original secret image (b) Recovered embedded image.

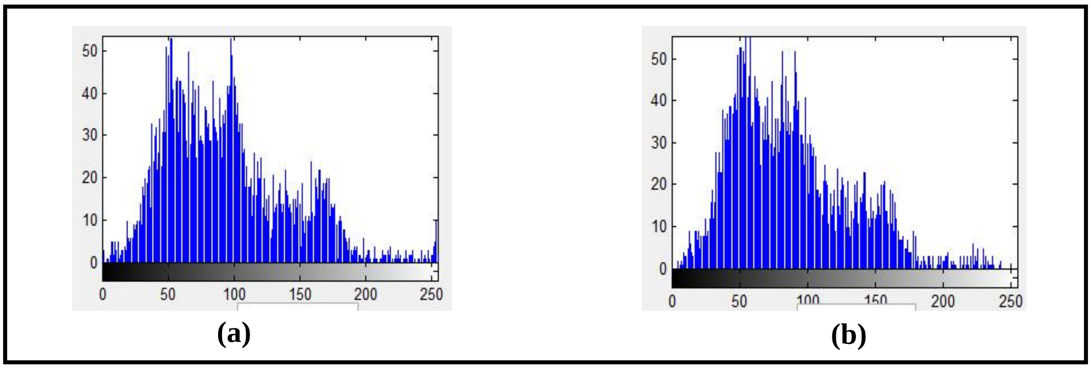


Figure 23. (a) The histogram for the Original embedded image (b) The histogram for the recovered embedded image

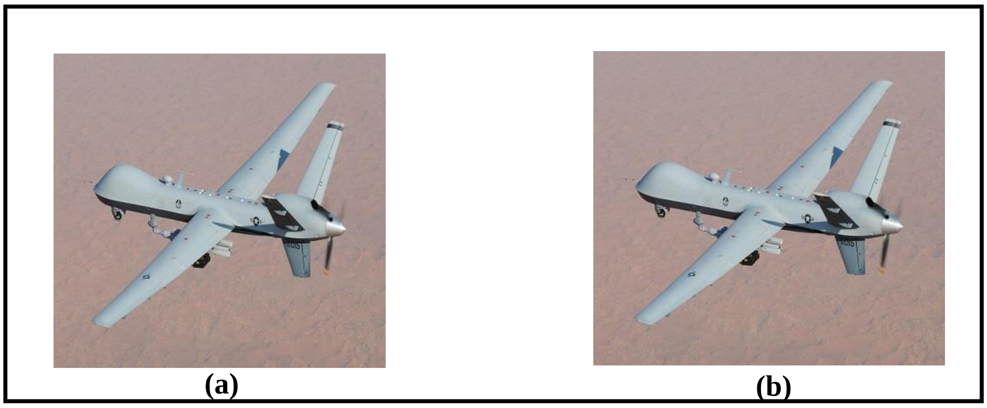


Figure 24. Recovered embedded image (secret image) at case 3 (a) Original embedded image (b) Recovered embedded image.

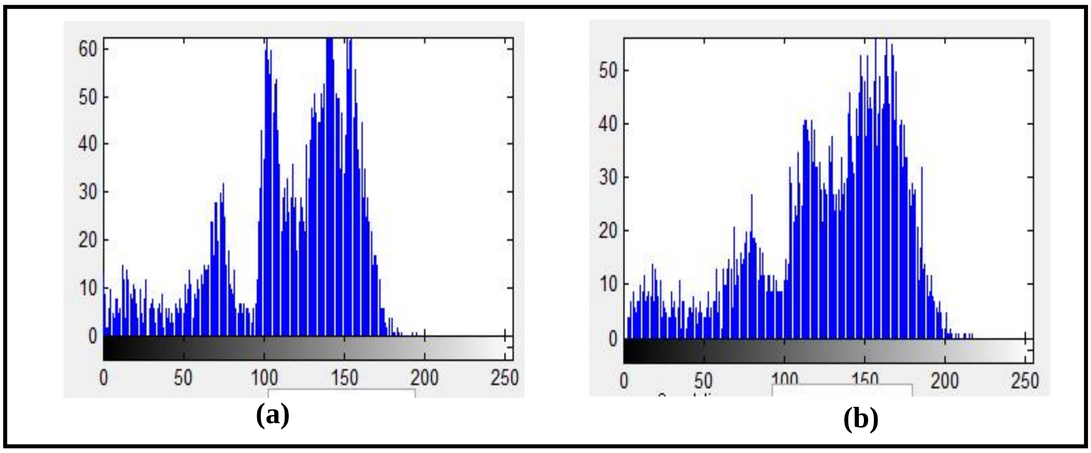


Figure 25. (a) The histogram for the Original embedded image. (b) The histogram for the recovered embedded image.

Table 5: The PSNR, Correlation and MSE results of extracted secret image from View image

Stego-image ◄──► For recovered secret image ◄──►

Cases	Stego-image	Original embedded image	PSNR/dB	Cor.	MSE
1	View	Robot	٣٨,١٦١٦	٠,٩٨٨٦٩	٠,١١٥٢
2	View	MRI	٣٦,٩٢٢٣	٠,٩٨٨٠٩	٠,٩١٨٩
3	View	Drone	٣٥,٧٧١٣	٠,٩٦٥٢٩	٠,٧٤٧١

For the results obtained from Table (5), Figures (26), (28) and Figure (30) given visual impression and compares between the original embedded image and recovered image for the listed results. As shown from these results, the form and histogram of the original embedded image does not differ from the histogram of the recovered embedded image. As shown in Figures (27), (29) and Figure (31).

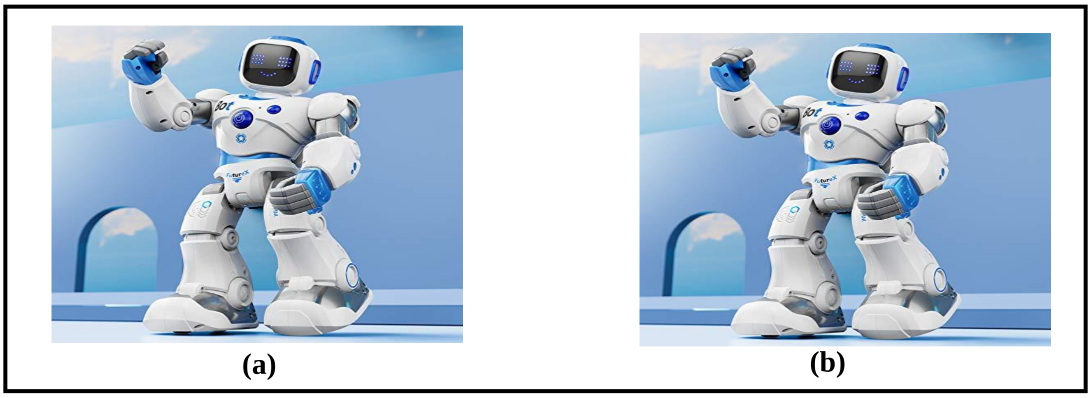


Figure 26. Recovered embedded image (secret image) at case 1. (a) Original embedded image (b) Recovered embedded image.

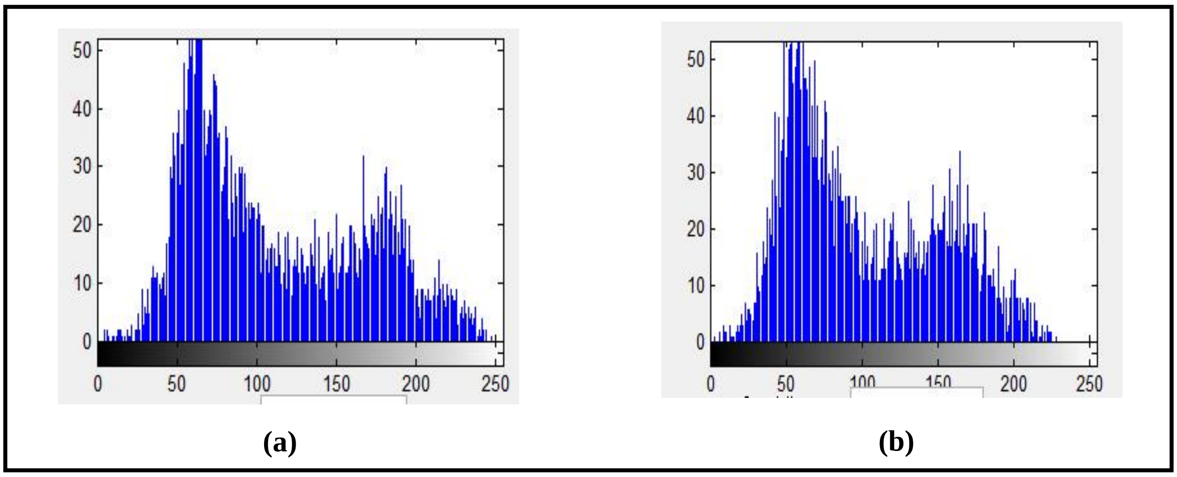


Figure 27. (a) The histogram for the Original embedded image (b) The histogram for the recovered embedded image.

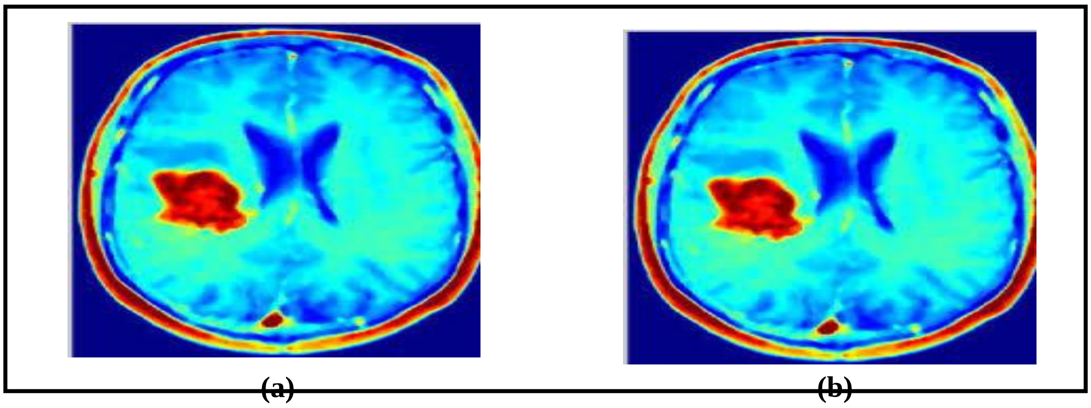
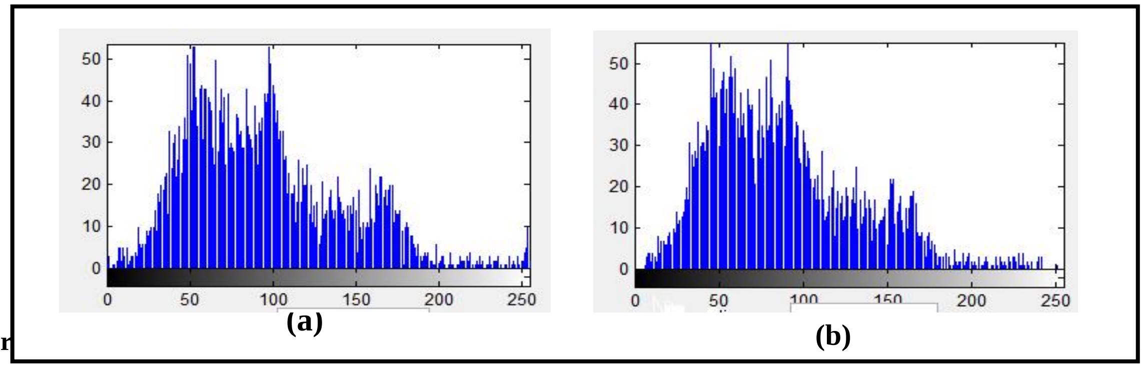


Figure 28. Recovered embedded image (secret image) at case 2. (a) Original embedded image (b) Recovered embedded image.



Figur image.

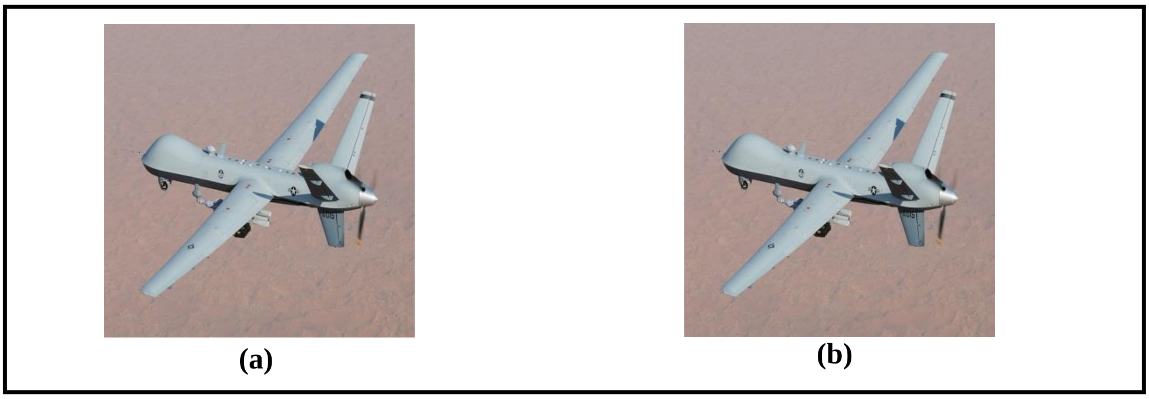


Figure 30. Recovered embedded image (secret image) at case 3. (a) Original embedded (b) Recovered embedded image.

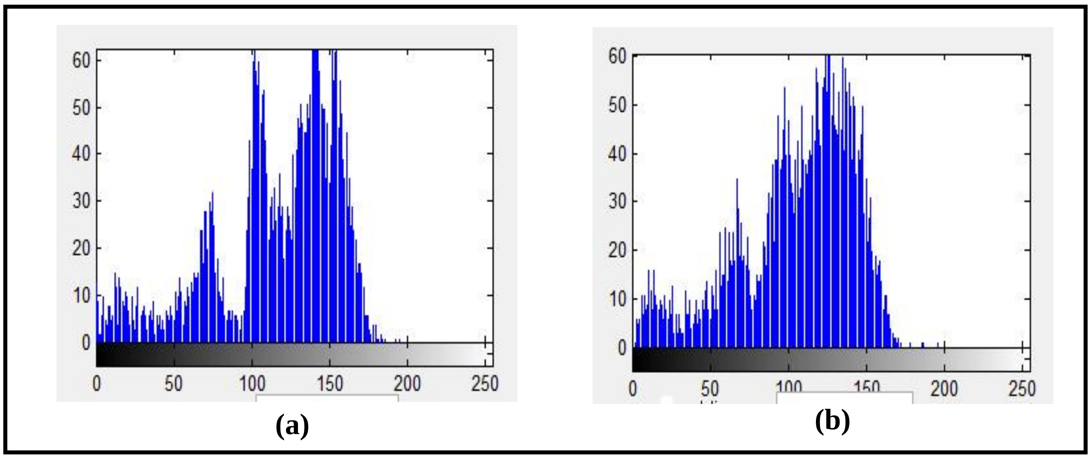


Figure 31. (a) The histogram for the Original embedded image. (b) The histogram for the recovered embedded image.

10. Results Discussion

The suggested system employs discrete cosine transform (DCT) and ant colony optimization (ACO) techniques to conceal the hidden image in the cover image. The secret key (index matrix) uses the combination of steganography and cryptography techniques to boost security. It is challenging for the attacker to figure out where the key is buried in the stego-image, even if he is aware of the (cipher key) embedded image. A very high capacity embedding of the hidden image is achieved by the proposed approaches utilization of the (ACO) and (DCT) bands along with the allocation of locations utilizing the thresholding technique. In worst-case scenarios, the capacity is roughly (1/4) of the covered image. Although it provides a great opportunity to keep the embedded information blocks exactly as they were and it facilitates the ability for mathematical manipulation methods to match as much as possible from the differed blocks, the use of block matching in the (DCT) method implies some restrictions and difficulties. to increase the embedding process accuracy and versatility. The histogram test demonstrated that the histogram of the stego-image and the cover picture are similar. The stego image will survive the statistic attack because the attacker cannot tell the difference between the statistics of the stego-image and the cover image. The choice of the cover image is crucial for enhancing system performance. The results of the three objective tests, as shown in Table (5), demonstrate that the system is secure because the correlation value is close to one, the PSNR is strong (up to 33.2052dB), and the MSE is low. The results further demonstrate that optimal performance is approached by (DCT) approaches, with PSNR and Correlation increasing and MSE decreasing as the capacity of cover pictures grows. Numerous simulations have been run to assess the proposed algorithms performance and to compare it to that of other current systems. As a result of the (DCT) methods exceptionally low embedding time, it is thought to contribute to the embedding of color images. Because the usage of both (ACO) and DCT algorithms before embedding process is crucial to build a robust system, the system is secure against passive attacker and active attacker. The system was improved by encrypting the secret key output from the embedding procedure and then hiding it inside the Stego-image.

11. Conclusions

Considering the transform domain that the process is done within, applying the concepts of the two Techniques (ACO and DCT) offers a superior performance than any other approach now in use. A very high capacity embedding of the hidden image is achieved by the proposed approaches usage of low and high frequency coefficients along with the allocation of locations utilizing the thresholding technique. In worst-case scenarios, the capacity is approximately (1/4) of the covered image. Using the DCT and quantization processes, a good extracted secret image is produced. The suggested method uses the Growth Algorithm to encrypt the secret key and conceal it inside the stego-image, resulting in a high level of security. A secret key steganography system can be used to describe the proposed system. Only the sender and receiver share the extraction process. To strengthen the security of the suggested system, the stego-key is generated during the embedding process and saved inside the stego-image. The receiver is unable to retrieve the original image without knowing how to extract the stego key. The inaccuracy between the original image and the recovered image has an inverse connection to the quantization quality value (Q). The proposed approach uses embedded images that are at least one-fourth the size of the cover image; this capability can be regarded as good. Since equation 10 offers additional opportunities for DCT block matching, it is thought to have a very high efficiency and is suitable for color images. Since the embedding process requires less run time than either technique (ACO or DCT), this is regarded as efficient in the field of steganography. In order to increase the scattering of the hidden image inside the cover image depending on the optimal path of the graph, ant colony optimization (ACO) was added to the embedding process. The ant colony optimization will select the information concealing places in the cover image based on the graphs ideal path, which will increase the embedding processes security and robustness. The DCT relies on sophisticated and cunning methods to determine where to bury informational blocks. The suggested system and the embedding procedure now incorporate smart technology that uses block matching in the (DCT) approach. Conclusion number thirteen: DCT matching difficulty is less time-consuming and provides a wonderful opportunity to save high-quality secret photos. Utilizing a certain low and high frequency in the cover with the same level and sub and in the secret is how information is concealed in DCT using low and high frequency coefficients. providing the stego image with transparency and invisibility. Taking into account the aforementioned findings, we deduce that the discrete cosine transforms and ACO approaches produce the best outcomes and accomplish security and transparency.

References

[1]. Reddy, G. D., Kiran, Y. V. U., Singh, P., Singh, S. V., Shaw, S., & Singh, J. (2022, October). A Proficient and secure way of Transmission using Cryptography and Steganography. In *2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS)* (pp. 582-586). IEEE.

[2]. Al-Nofaie, S., Gutub, A., & Al-Ghamdi, M. (2021). Enhancing Arabic text steganography for personal usage utilizing pseudo-spaces. *Journal of King Saud University-Computer and Information Sciences*, *33*(8), 963-974.

[3]. Fateh, M., Rezvani, M., & Irani, Y. (2021). A new method of coding for steganography based on LSB matching revisited. *Security and Communication Networks*, *2021*, 1-15.

[4]. Zhang, Y., Lu, H., & Abbas, H. (2018). Mobile Intelligence Assisted by Data Analytics and Cognitive Computing. *Wireless Communications and Mobile Computing*, *2018*, 1-2.

[5]. Hamano, G., Imaizumi, S., & Kiya, H. (2023). Effects of JPEG Compression on Vision Transformer Image Classification for Encryption-then-Compression Images. *Sensors*, *23*(7), 3400.

[6]. Zhang, Y., Luo, X., Wang, J., Guo, Y., & Liu, F. (2021). Image robust adaptive steganography adapted to lossy channels in open social networks. *Information Sciences*, *564*, 306-326.

[7]. Tao, H., Chongmin, L., Zain, J. M., & Abdalla, A. N. (2014). Robust image watermarking theories and techniques: A review. *Journal of applied research and technology*, *12*(1), 122-138.

[8]. Zhang, Y., Wang, C., Wang, X., & Wang, M. (2017). Feature-based image watermarking algorithm using SVD and APBT for copyright protection. *future internet*, *9*(2), 13.

[9]. Makhdoom, I., Abolhasan, M., & Lipman, J. (2022). A comprehensive survey of covert communication techniques, limitations and future challenges. *Computers & Security*, *120*, 102784.

[10]. Kaur, H., & Rani, J. (2016). A Survey on different techniques of steganography. In *MATEC web of conferences* (Vol. 57, p. 02003). EDP Sciences.

[11]. Boryczka, M., & Kazana, G. (2023). Hiding Information in Digital Images Using Ant Algorithms. *Entropy*, *25*(7), 963.

[12]. Gnanalakshmi, V., & Indumathi, G. (2023). A review on image steganographic techniques based on optimization algorithms for secret communication. *MULTIMEDIA TOOLS AND APPLICATIONS*.

[13]. Saini, K. (2017). A review on video steganography techniques in spatial domain. *2017 Recent Developments in Control, Automation & Power Engineering (RDCAPE)*, 366-371.

[14]. Priya, A. (2018). High capacity and optimized image steganography technique based on ant colony optimization algorithm. *International Journal of Emerging Technology and Innovative Engineering*, *4*(6).

[15]. Bajracharya, R., Shrestha, R., Hassan, S. A., Jung, H., & Shin, H. (2023). 5G and Beyond Private Military Communication: Trend, Requirements, Challenges and Enablers. *IEEE Access*.

[16]. Subramanian, N., Elharrouss, O., Al-Maadeed, S., & Bouridane, A. (2021). Image steganography: A review of the recent advances. *IEEE access*, *9*, 23409-23423.

[17]. Li, H., & Guo, X. (2018). Embedding and extracting digital watermark based on DCT algorithm. *Journal of Computer and Communications*, *6*(11), 287-298.

[18]. Sharma, V., & Mir, R. N. (2022). An enhanced time efficient technique for image watermarking using ant colony optimization and light gradient boosting algorithm. *Journal of King Saud University-Computer and Information Sciences*, *34*(3), 615-626.

[19]. Ali, M., Wook Ahn, C., Pant, M., Kumar, S., Singh, M. K., & Saini, D. (2020). An optimized digital watermarking scheme based on invariant DC coefficients in spatial domain. *Electronics*, *9*(9), 1428.

[20]. Liu, C., & Ding, Q. (2020). A color image encryption scheme based on a novel 3d chaotic mapping. *Complexity*, *2020*, 1-20.

[21]. Lu, W., Zhang, J., Zhao, X., Zhang, W., & Huang, J. (2020). Secure robust JPEG steganography based on autoencoder with adaptive BCH encoding. *IEEE Transactions on Circuits and Systems for Video Technology*, *31*(7), 2909-2922.

[22]. Pan, P., Wu, Z., Yang, C., & Zhao, B. (2022). Double-matrix decomposition image steganography scheme based on wavelet transform with multi-region coverage. *Entropy*, *24*(2), 246.

[23]. Zhang, Y. Q., Zhong, K., & Wang, X. Y. (2022). High-Capacity Image Steganography Based on Discrete Hadamard Transform. *IEEE Access*, *10*, 65141-65155.

[24]. Gaertner, D., & Clark, K. L. (2005, June). On Optimal Parameters for Ant Colony Optimization Algorithms. In *IC-AI* (pp. 83-89).

[25]. Zebari, D. A., Zeebaree, D. Q., Saeed, J. N., Zebari, N. A., & Adel, A. Z. (2020). Image steganography based on swarm intelligence algorithms: A survey. *people*, *7*(8), 9.

[26]. Thakkar, F., & Srivastava, V. K. (2017). A particle swarm optimization and block-SVD-based watermarking for digital images. *Turkish Journal of Electrical Engineering and Computer Sciences*, *25*(4), 3273-3288.

[27]. Singhal, V., Shukla, Y. K., & Praksash, N. (2020). Image steganography embedded with advance encryption standard (AES) securing with SHA-256. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, *9*(8).

[28]. Riabko, A. V., Zaika, O. V., Kukharchuk, R. P., Vakaliuk, T. A., & Hordiienko, I. V. (2022, June). Algorithm of ant colony optimization (ACO) for 3D variation traveling salesman problem. In *Journal of Physics: Conference Series* (Vol. 2288, No. 1, p. 012001). IOP Publishing.

[29]. Nayyar, A., & Singh, R. (2016, March). Ant colony optimization—computational swarm intelligence technique. In *2016 3rd International conference on computing for sustainable global development (INDIACom)* (pp. 1493-1499). IEEE.

[30]. Khan, S., & Bianchi, T. (2018). Ant Colony Optimization (ACO) based Data Hiding in Image Complex Region. *International Journal of Electrical & Computer Engineering (2088-8708)*, *8*(1).

[31]. Ning, J., Zhang, Q., Zhang, C., & Zhang, B. (2018). A best-path-updating information-guided ant colony optimization algorithm. *Information Sciences*, *433*, 142-162.

[32]. Ignatious, N., & Ali, S. (2019, November). Identifying A Regression Test Prioritization Technique and Proposing A Tool for Automation for Trade me Website. In *CS & IT Conference Proceedings* (Vol. 9, No. 14). CS & IT Conference Proceedings.

[33]. Alomoush, W., Khashan, O. A., Alrosan, A., Attar, H. H., Almomani, A., Alhosban, F., & Makhadmeh, S. N. (2023). Digital image watermarking using discrete cosine transformation based linear modulation. *Journal of Cloud Computing*, *12*(1), 1-17.

[34]. Shawahna, A., Haque, M. E., & Amin, A. (2019). JPEG image compression using the discrete cosine transform: an overview, applications, and hardware implementation. *arXiv preprint arXiv:1912.10789*.

[35]. Raid, A. M., Khedr, W. M., El-Dosuky, M. A., & Ahmed, W. (2014). Jpeg image compression using discrete cosine transform-A survey. *arXiv preprint arXiv:1405.6147*.

[36]. Abdulrazzaq, S. T., Rasheed, M. H., & Siddeq, M. M. (2023). The multi-quantization process with matrix size reduction is applied to compress images with strip structure light that is commonly used in 3D reconstructions.

[37]. Nikoukhah, T., Colom, M., Morel, J. M., & von Gioi, R. G. (2022). A Reliable JPEG Quantization Table Estimator. *Image Processing On Line*, *12*, 173-197.

[38]. Senthooran, V., & Ranathunga, L. (2014, August). DCT coefficient dependent quantization table modification steganographic algorithm. In *2014 First International Conference on Networks & Soft Computing (ICNSC2014)* (pp. 432-436). IEEE.

[39]. Ghosh, D., Chattopadhyay, A. K., Chanda, K., & Nag, A. (2020). A Secure Steganography Scheme Using LFSR. In *Emerging Technology in Modelling and Graphics: Proceedings of IEM Graph 2018* (pp. 713-720). Springer Singapore.

[40]. Siahaan, A. P. U., Ikhwan, A., & Aryza, S. (2018). A novelty of data mining for promoting education based on FP-growth algorithm.

[41]. Kollin, F., & Bavey, A. (2017). Ant colony optimization algorithms: pheromone techniques for TSP.