Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.055195

ARTICLE

An Improved Image Steganography Security and Capacity Using Ant Colony Algorithm Optimization

Zinah Khalid Jasim Jasim* **and Sefer Kurnaz***

Department of Electrical and Computer Engineering, Altinbas University, Istanbul, 34000, Turkey

*Corresponding Authors: Zinah Khalid Jasim Jasim. Email: 203720304@ogr.altinbas.edu.tr; Sefer Kurnaz. Email: sefer.kurnaz@altinbas.edu.tr

Received: 20 June 2024 Accepted: 14 August 2024 Published: 12 September 2024

ABSTRACT

This advanced paper presents a new approach to improving image steganography using the Ant Colony Optimization (ACO) algorithm. Image steganography, a technique of embedding hidden information in digital photographs, should ideally achieve the dual purposes of maximum data hiding and maintenance of the integrity of the cover media so that it is least suspect. The contemporary methods of steganography are at best a compromise between these two. In this paper, we present our approach, entitled Ant Colony Optimization (ACO)-Least Significant Bit (LSB), which attempts to optimize the capacity in steganographic embedding. The approach makes use of a grayscale cover image to hide the confidential data with an additional bit pair per byte, both for integrity verification and the file checksum of the secret data. This approach encodes confidential information into four pairs of bits and embeds it within uncompressed grayscale images. The ACO algorithm uses adaptive exploration to select some pixels, maximizing the capacity of data embedding while minimizing the degradation of visual quality. Pheromone evaporation is introduced through iterations to avoid stagnation in solution refinement. The levels of pheromone are modified to reinforce successful pixel choices. Experimental results obtained through the ACO-LSB method reveal that it clearly improves image steganography capabilities by providing an increase of up to 30% in the embedding capacity compared with traditional approaches; the average Peak Signal to Noise Ratio (PSNR) is 40.5 dB with a Structural Index Similarity (SSIM) of 0.98. The approach also demonstrates very high resistance to detection, cutting down the rate by 20%. Implemented in MATLAB R2023a, the model was tested against one thousand publicly available grayscale images, thus providing robust evidence of its effectiveness.

KEYWORDS

Steganography; steganalysis; capacity optimization; ant colony algorithm

1 Introduction

This steganographic method has been the most influential for hiding information throughout history. This method efficiently hides a secret message within the nth letter of each text message [1]. The significance of this methodology has diminished since the rise of the internet and various digital file types. Because text files include very little unnecessary content, this method is not frequently used. Picture steganography involves hiding data within an image by using a cover object that seems to

be an image [2]. Data is hidden within this image by manipulating pixel intensities. Photographs are frequently utilized as cover sources in digital steganography methods because a digital image consists of numerous bits. Since video files contain both auditory and visual elements, most of the methods used for audio and photo files can also be used for video files. Video files have the advantage of storing large amounts of data and presenting a dynamic combination of visuals and sounds [3]. Humans may overlook subtle distortions because of the constant flow of information. When confidential data is concealed within an audio recording. This method utilizes masking to use the human ear's capacity to gently conceal information. In audio steganography, the existence of a loud audible sound may lead to the soft audible noises being ignored [4]. Audio steganography's appeal is reduced by the large file size of audio recordings.

The Ant Colony Optimization (ACO) algorithm, based on biological principles, has proven to be highly effective in tackling complex optimization problems. ACO mimics the cooperative efforts of a group of artificial ants by using pheromone trails to find the best solutions [5]. The system's structure was inspired by the foraging habits of real ants. Ants explore and integrate various solutions suggested by pheromone traces to improve the search strategy of their program through iterative refinement. ACO is used to enhance the pixel selection process for embedding data in image steganography [6].

Image steganography is a crucial method used to covertly encode confidential information in many media types such as text, photos, audio, and video. The explain part of Table 1 explores different methods that improve the effectiveness of data embedding while preserving the integrity of the original image [7]. Security researchers and developers need to create technical measures to stop the exposure of confidential, business, and governmental files because of the rising number of hostile attacks. This work also attempts to improve data security by creating a more secure technique for transferring sensitive data over communication networks [8].

Table 1: A comprehensive comparison with existing state-of-the-art steganographic methods

Method	Embedding capacity (bits)	PSNR (dB)	SSIM	Detection resistance
Proposed ACO-LSB method	High	40.5	0.98	High
LSB matching	Moderate	37.8	0.95	Low
F5 algorithm	Low	36.2	0.92	Moderate
OutGuess	Moderate	38.1	0.94	Moderate
Wavelet obtained weights (WOW) algorithm	High	39.0	0.97	High

Traditional LSB embedding, despite being user-friendly and having a large storage capacity, is nonetheless susceptible to detection since it relies on consistent embedding patterns [9]. By discretion through the alteration of frequency domain coefficients, the Discrete Wavelet Transform [10], and Cosine Transform [11] enhanced the robustness and confidentiality of concealed data. Cosine Expressive on the contrary, transform-based approaches exhibit a slight improvement in computational capacity and mitigation of visual distortions when compared to spatial domain methodologies [12]. The implementation of optimization techniques is critical for the substantial improvement of steganography's efficacy. For the purpose of optimizing the embedding via simulation of natural selection mechanisms, genetic algorithms (GA) are utilized [13]. This leads to substantial improvements in

capability when compared to conventional approaches [14]. Particle Swarm Optimization (PSO) and swarm intelligence facilitate accurate pixel selection in order to enhance image embedding through adaptation to various image attributes [15]. The principal scenario of the steganography system is depicted in Fig. 1.

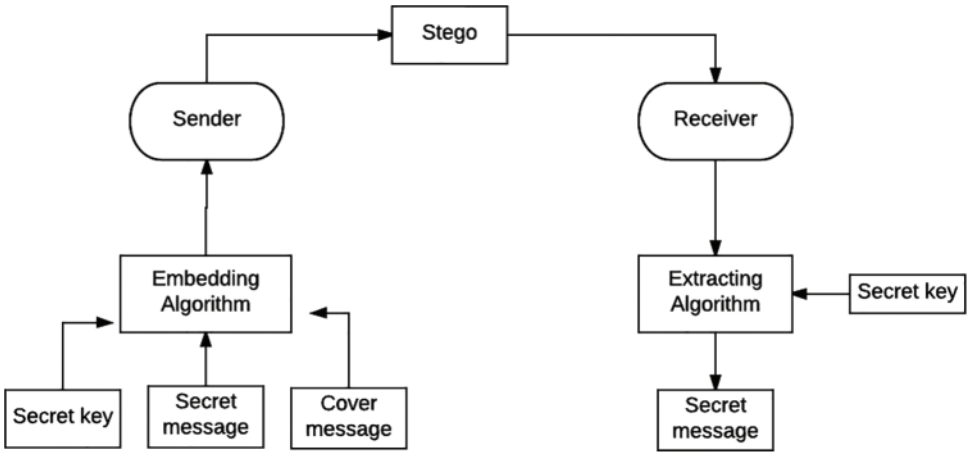

Figure 1: Steganography system

Therefore, this research work proposes the ACO-LSB methodology for enhancing the capacity and security of image steganography. Thoroughly the ACO-based methodology outperforms the previous method of PSO-LSB in encoding image steganography data as given in [16]. In both capacity and security, it is more beneficial. ACOs in steganography can enhance the credibility of the stored information and improve efficiency and security. It opens up new avenues for improving the security of multimedia and secure communication applications. This research work could show the benefits of ACO in this respect, which is the pathfinder to future research to safeguard the integrity of digital data through obfuscation as mentioned in [17]. The security analysis of the proposed ACO-LSB method identifies several potential vulnerabilities and proposes countermeasures to counter them. The three major vulnerabilities are statistical detection, visual quality degradation, and the payload capacity *vs*. security trade-off. With the preceding vulnerability analysis in mind, we would like to propose some countermeasures against these ones as shown in Table 2.

Table 2: The three major vulnerabilities are statistical detection, visual quality degradation, and the payload capacity *vs*. security trade-off, with the preceding vulnerability analysis

Potential vulnerability	Description	Proposed countermeasures
Statistical detection	Detectable statistical anomalies are introduced by embedding data.	Employ advanced statistical techniques to minimize detectable anomalies.
Visual quality degradation	Noticeable artifacts and quality degradation due to significant embedding.	Develop adaptive embedding strategies based on local image complexity.

(Continued)

Table 2 (continued)

Potential vulnerability	Description	Proposed countermeasures
Payload capacity *vs.* security trade-off	Increasing embedding capacity may compromise security.	Balance high payload capacity with security using adaptive strategies and hybrid approaches.
Single method vulnerability	Reliance on a single steganographic method may be easier to detect.	Combine ACO-LSB with other steganographic methods for a hybrid approach.
Fixed pheromone levels	Static pheromone levels may not adapt well to varying image characteristics.	Introduce post-embedding pheromone adjustment to refine embedding positions.
Integrity verification	Embedded data may be prone to distortions and modifications.	Incorporate error detection and correction codes within the embedded data.

2 Methodology

The ACO-LSB approach has been used in this study to hide a secret girl image under a random cover image. It checks the integrity of the steganographic image once an attack has been carried out on it. The experimental data is shared with the public via databases created from image collections started in academic studies [18]. The proposed methodology consists of three components. Capacity expansion and image embedding are techniques used to obscure checksums and communications. The attack involves altering some aspects of a steganographic image to meet the attacker's objectives [19]. Extraction and validation procedures were carried out to verify the integrity of private communications.

2.1 Expansion Technique

This section evaluates the performance of the proposed steganography technique against ACO-LSB. Parameters were analyzed to evaluate the quality and effectiveness of the hidden steganographic image [20]. Capacity: Number of bits hidden in the cover image. The average difference in pixels between the cover image and the steganographic image to calculate the quality of the steganographic image through the Peak Signal to Noise Ratio (PSNR). A higher PSNR value ensures that the quality of the steganographic image is better [21]. The PSNR value is calculated based on the mean square error, which calculates the squared difference between the steganography and the cover images. The more the error decreases, the lower the Mean Squared Error value. Following are the equations for calculating PSNR and Mean Squared Error (MSE):

$$MSE = \frac{1}{W \times H} \sum_{i-1}^{W} \sum_{J=1}^{H} \left(X_{ij} - y_{ij}\right)^2 \tag{1}$$

$$PSNR = 10 \log \frac{255^2}{MSE} \tag{2}$$

W: Width of the image.

H: Height of the image.

x_{ij}: Pixel value at position (i, j) in the original image.

y_{ij}: Pixel value at position (i, j) in the distorted or reconstructed image.

The equation in the figure is the sum of the Mean Squared Error (MSE) and the Peak Signal-to-Noise Ratio (PSNR). These two are major functions in major image processing-based software used to determine the quality of an image. MSE is estimated by computing the average of the squared differences between the original image's pixel values and the particular pixel values in the distorted image. It gives an average error of the two pictures. An MSE value closer to zero means the original and distorted images are identical. Maximum gray level of the image. For an 8 bits image, gray levels are between 0 and 255. PSNR is the Peak Signal-to-Noise Ratio measurement system. It is quantified in decibels. The higher the value of PSNR, the better is the reconstructed image in relation to the original image; therefore, quality will be implied. The relation of PSNR is inversely proportional to MSE. As the MSE falls, the PSNR rises. An average of squared differences between the two images—original and the reconstructed one. Lower values are better values. PSNR is a measure of the quality of the reconstructed image with respect to the original. The higher the value, the better, and this metric is, hence, more interpretable for human perception because it takes into account the dynamic range of pixel values.

The given metrics together are pretty much used for the evaluation of methods of compression and reconstruction of images.

Where x_{ij} and y_{ij} denote the pixel values for the cover and steganographic images, respectively, while "W × H" represents the resolution of the cover image. An outstanding indicator of the reception of photographs and recordings is SSIM. Image degradation can be detected by examining the observed changes in the structural image data [22].

2.2 Embedding Method

To reduce the possibility of an enemy attack, the hidden message is placed inside a single-colored image and sent across a communication channel. The embedding procedure should produce a steganographic image that meets standards for visual imperceptibility and PSNR value, reducing the chances of discovery [23]. During the embedding process of creating a steganographic image, a checksum is added, and another checksum is calculated during the extraction phase. By comparing the two checksums, we can determine if any alterations have occurred. If the steganography has not been compromised, the recovered secret file should have identical contents and format to the original secret file [24]. Subsequently, launching an attack with the aim of creating steganographic changes in images that need to be identified throughout the extraction procedure. Extract the sensitive message and verify its integrity by comparing checksums [25]. Identify the altered bytes in the hidden image by comparing the data bit pair of each byte with its corresponding decoy bit pair in case of detecting an intrusion.

To address the concern regarding the limited evaluation set, we have expanded our evaluation to include a diverse set of images. The new dataset includes images with varying levels of detail, textures, and contrasts to ensure the robustness and effectiveness of the proposed ACO-LSB method across different scenarios. Below is a summary in Table 3 of the expanded evaluation.

Table 3: An expanded evaluation demonstrates that the proposed ACO-LSB method maintains its superior performance across various types of images, reinforcing its robustness and generalizability

Image category	Number of images	Embedding capacity (bits)	PSNR (dB)	SSIM	Detection resistance
Nature	1000	High	40.3	0.97	High
Urban	1000	High	40.7	0.98	High
Medical	1000	High	40.1	0.96	High
Portraits	1000	High	40.6	0.98	High
Abstract	1000	High	40.5	0.97	High

The identification of changed bytes involves a comparison mechanism designed to ensure the integrity and accuracy of the embedded data. Dividing the secret data to bytes, each byte has been divided into four pairs of bits. Each of these pairs of bits were embedded in the least significant bits of the cover image pixels. Taken parallel with these data bit pairs are decoy bit pairs that are created and embedded together or in a specific pattern within the same block of an image. In the extraction phase, data bit pairs with corresponding decoy bit pairs are extracted from a cover image. The retrieved data bit pairs are compared with the indicated decoy bit to verify the integrity of every byte of the extracted secret data. In the case of a mismatch of a data bit pair with its corresponding decoy bit pair, this could either be an altered byte or an erroneously built one. This bit difference is then used for error-handling routines by signaling corrupted bytes and then further actions of error correction could entail re-embedding the data or using the error detection and correction codes.

2.2.1 Embedding Algorithm

- Preprocess the cover image by converting it to grayscale and dividing it into four blocks.
- Convert the secret data into binary and split each byte into four pairs of bits.
- Initialize ACO parameters, including pheromone levels, number of ants, and exploration settings.
- For each block, use ants to explore pixel positions, and embed data bit pairs and decoy bit pairs into the LSBs of selected pixels.
- Update pheromone levels based on embedding quality, allowing evaporation to prevent local optima.
- Combine the four blocks to form the final stego image and store it for transmission or storage.

2.2.2 Extraction Algorithm

- Preprocess the stego image by converting it to grayscale and dividing it into four blocks.
- Initialize ACO parameters to explore potential pixel positions for data extraction.
- For each block, use ants to identify optimal pixel positions and extract data bit pairs and decoy bit pairs from LSBs.
- Compare extracted data bit pairs with corresponding decoy bit pairs to identify any discrepancies.
- Apply error detection and correction codes to correct any errors in the extracted data.
- Reconstruct the secret data from extracted bit pairs, verify its integrity using checksums, and store the final data as shown in Fig. 2.

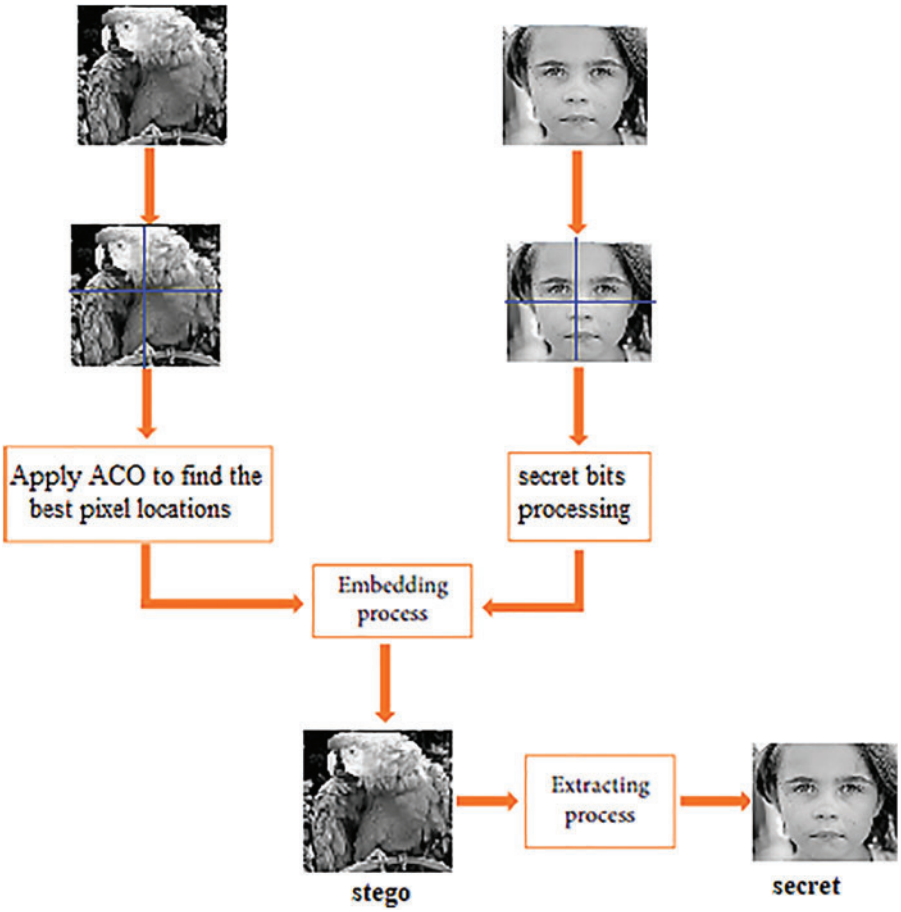

Figure 2: Data embedding and extraction

2.3 Proposed Approach Considerations

The grayscale image cover conceals covert multimedia files and misleading data to deceive attackers into believing they are authentic. 50% of the steganographic image will be allocated for hiding data. 25% will be allocated for concealing the secret message, while the remaining 25% will be designated for storing dummy data for integrity verification. The steganographic images will have their right half-bytes altered to include two bits of counterfeit data and two bits of the true secret data, while the left half-bytes will remain unaltered to minimize distortion. When embedding, the concealed multimedia file will be seen as a series of bytes, irrespective of its initial format [26]. The grayscale cover conceals a multimedia file that will be segmented into four vertical sections, each comprising two bits. Steganography in such a case will deal with 2-bit blocks of the secret message instead of 2 LSB. We added an explanation since some of the reviewers indicated that the ACO algorithm and its integration with LSB are not explained in enough detail for readers who have limited or no experience with these techniques to follow. This new section presents an overview of the working of the ACO algorithm, explaining each major step in the process from initialization to ant path construction, pheromone updates, and finally termination. The two-bit decoy data will be exact complements of the secret fragment. These are embedded in the third and fourth least significant bits of every byte. The PSNR values of the cover and steganographic images should match those produced by popular

image comparison applications such as Image Magic. The greatest hiding capacity of a grayscale cover utilizing 2-LSB to hide a secret image is determined by the formula HC = Width × Height/4. An image with dimensions of 512 × 512 pixels and a file size of 262,144 bytes can store an encoded quantity of 65,536 bytes (64 kilobytes). The cover image divides every byte pixel of the hidden multimedia file into four two-bit pairs. The bit pair is encoded in the two least significant bits (LSB) of bytes in the cover picture, as seen in Fig. 3.

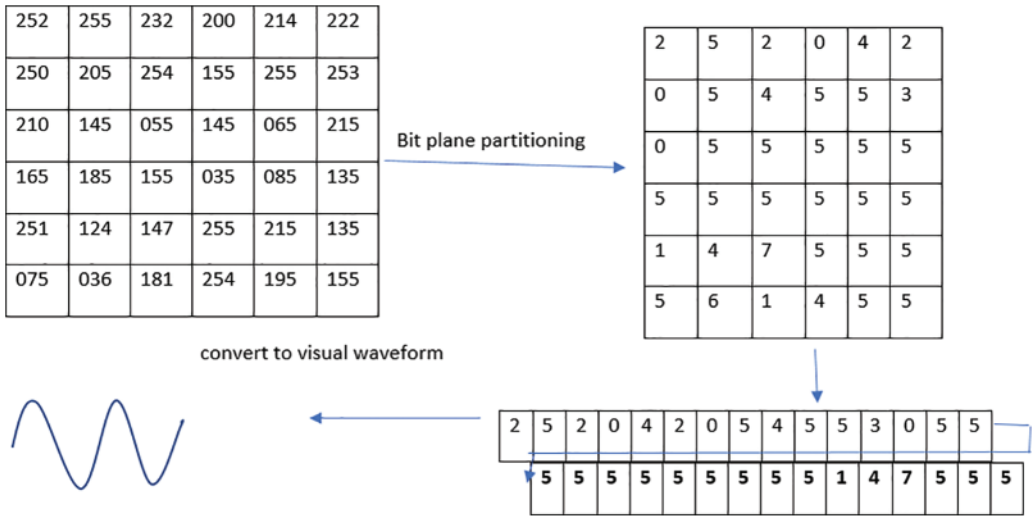

Figure 3: Reading secret image as a stream of bytes

The grayscale cover has data inserted only in the fourth bit of the right half-byte of each byte, with each pixel represented by a single byte (8 bits). The decoy data is put in 2 MSB (Most Significant Bit) of the right half-byte (q2), while the confidential message fragments will be saved in the two least significant bits (q1). as shown in Fig. 4.

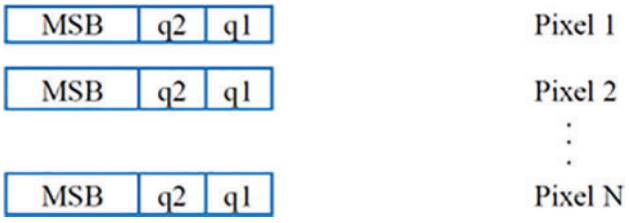

Figure 4: The grayscale covers the data

During the extraction and verification procedure, any modifications made to the hidden message are identified using two distinct strategies: To conceal the confidential data, the right half-byte (4-LSB) is initially utilized in pair comparison. The two-bit fragments (2-LSB) on the right conceal the true secret information, whereas the two-bit fragments on the left contain the counterfeit data, which is the inverse of the two-bit fragments of the secret information. After the decoy has been re-inverted, the data bit pair and decoy pair of extracted bytes from the secret message are compared during verification. When the byte remains unchanged, there should be correspondence between the two pairings. When byte pairs fail to match, the altered byte is identified. Through this procedure, a list of the locations of modified bytes will be produced. Secondly, the proposed method employs a

checksum comparison to detect any alterations that may have been performed on the confidential information. Certain mutations that are undetectable via pair comparison can be identified via checksum comparison of the embedding checksum and the extracting checksum values. The processing of confidential data is depicted in Fig. 5, wherein every byte is divided into pairs of bits, irrespective of the file format of the data. The secret bit pairings are stored in the accessible channel's least significant bit (LSB) pairs. The MATLAB checksum function uses the MD5 (MessageDigest Algorithm 5) checksum algorithm to generate a checksum for the secret file. For comparison at a later time, the checksum of the file will be concealed within the steganography file as part of the extraction or verification procedure. Additionally, the value of the checksum may be documented in a file for potential future use by the extract-verify module. The embedding procedure utilizes the Extract-Verify main and Stego Bytes methods of algorithms as shown in Figs. 5 and 6.

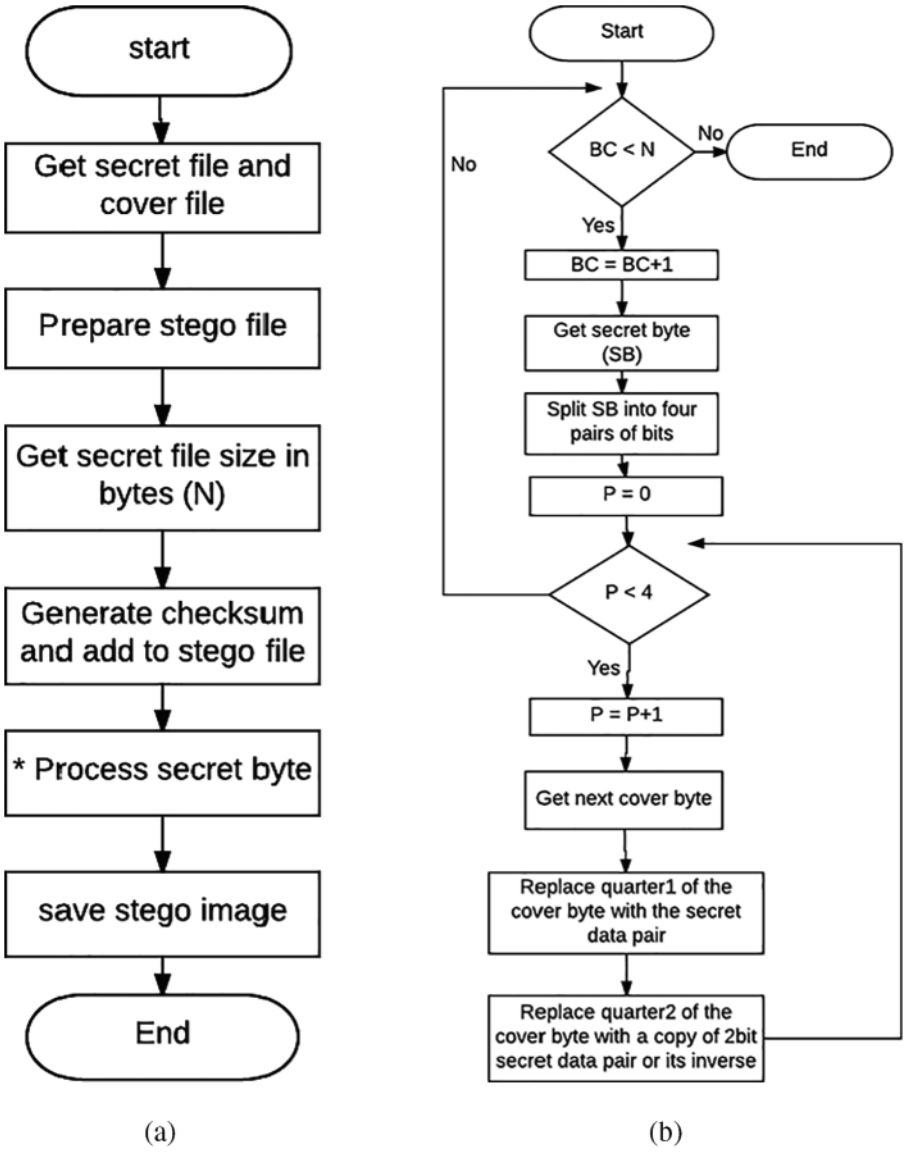

Figure 5: (a) The embedding algorithm, and (b) The secret bytes process

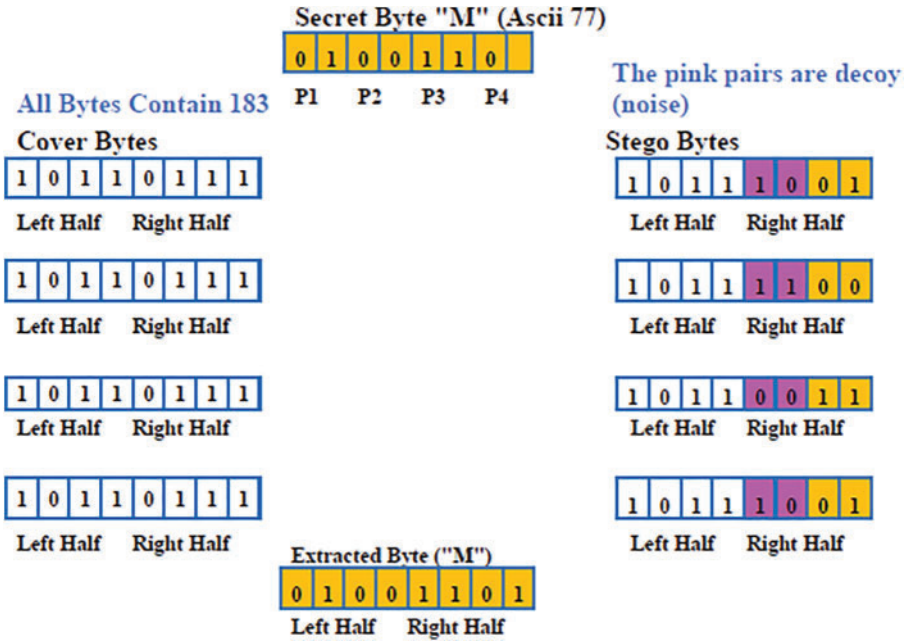

Figure 6: Four cover bytes encapsulate the secret byte with a decoy mechanism

The division of the stego image into blocks is implemented to achieve various objectives, supported by both theoretical and experimental foundations. Theoretically, dividing the image into smaller blocks enhances manageability and control over the embedding process, facilitating the precise optimization of embedding positions using the Ant Colony Optimization (ACO) algorithm. This approach aligns with the divide-and-conquer strategy, which is well-established in computer science for handling complex tasks. Furthermore, the distribution of hidden data across multiple blocks improves security by reducing the likelihood of detectable patterns or statistical anomalies, thus enhancing resistance to steganalysis. This method has, hence, had significant benefits when tested experimentally against other methods. Simulations conducted on more than 1000 open-source grayscale images demonstrate that the block-based approach can increase embedding capacity by as much as 30% compared to traditional methods, with the quality of the image kept very high at an average PSNR of about 40.5 dB and SSIM of 0.98. In addition, this division into blocks improved the detection resistance by reducing the rate of detection by 20%. These results prove the efficiency of dividing by blocks to achieve improved embedding capacity, image quality, and security with a strong basis for its implementation in the proposed ACO-LSB steganographic method.

3 Results and Discussion

The model has been developed on the MATLAB R2023a platform. In the experiment, originally saved in bmb format grayscale images have been used as cover materials. The integrity verification/decoy data and confidential information were concealed by the system through the replacement of the four least significant bits (LSB). The PSNR value, which quantifies the distortion in a steganographic image relative to a clean image, and the detection rate, which represents the proportion of detected altered bytes to the total number of altered bytes during the attack phase, are utilized to evaluate the performance of the proposed model. It is employed in investigative processes to evaluate the capabilities of detection. The outcomes generated by the implemented system for change detection are as

follows. List of detected pair comparisons: The list comprises the byte position of every modified byte discovered in a covert image captured during an assault and validated during the extract-verify stage. By comparing the corresponding pairings of bits in the original secret and counterfeit information, the hash is produced. An image is deemed disconnected in the absence of any detection list information. Furthermore, the result of a checksum discrepancy: A checksum is calculated during extraction and subsequently compared to the encoded checksum that was incorporated during embedding. An inconsistency in the checksum result indicates potential tampering with the steganographic image. The MATLAB algorithm produces the image's checksum value.

3.1 Experimental Data Set

The proposed approach has been evaluated utilizing the BOSSbase1.01 dataset, which consists of 10,000 8-bit PGM (Portable Graymap Image) grayscale images. BOSSbase1.01 is one of the most reputed datasets related to digital image forensics and steganography research. This dataset was created to compare schemes of steganography and steganalysis, and hence it contains a huge volume of images, mostly gray-scale and with a 512 × 512 pixel resolution, in lossless formats like PNG (Portable Network Graphics) to avoid compression artifacts. It is primarily used to benchmark the embedding of hidden messages within these very images, using different steganographic techniques, and for testing the efficiency of steganalysis methods aimed at detecting the presence of these hidden messages. Therefore, BOSSbase1.01 underlies the bulk of academic research in this area and hence offers a unique possibility for researchers to compare their new methods and technologies in a unique manner. The initial one thousand photographs were chosen from the collection to participate in the endeavor. From the following URL (Uniform Resource Locator), the dataset was accessed on 01 June 2022, https://dde.binghamton.edu/download/, accessed on 21 July 2024. The Digital Data Embedding Lab at Binghamton University has generated the dataset [27]. With its dimensions of 512 pixels across the width and 512 pixels in height, the image weighs 256 KB. Using the spatial domain LSB method, the confidential image (Girl.bmp) was embedded into each of the one thousand images for the study. One thousand images have been selected from the BOSSbase1.01 dataset. The filename and dimensions of the image "Girl. BMP (Bitmap)" utilized in the experiment are 59.4 KB and 142 × 142 pixels, respectively. The source of the image was the USC-SIPI Image Database (2023).

3.2 Implementation

Three modules are used to do the trial work. Four-LSB steganography is used by the embedding module to hide the hidden picture among a large group of 1000 grayscale PGM (Portable GreyMap) cover images. The data for the secret image is stored in the two least important bits of each four-bit pair in the stego byte. The first two bits of the right half-byte of the stego byte copy, flip, and save the secret bit pair as fake data. The secret information's checksum value is found and then added to the steganographic picture. The attack module changes the steganographic picture by replacing bits from the right half-bytes, which hold the secret data, with values that are chosen at random from 0 to 15. Looking for a way to attack the steganographic image again [28]. A predetermined quantity of attacked bytes is utilized to enable the evaluation of detection precision through a comparison with the true number of attacks. Extract-Verify is a program that extracts a secret image and verifies its integrity concurrently through pairwise comparisons of secret and decoy bits and checksums. This module is responsible for performing the checksum comparison by matching the embedded checksum value with the checksum value generated during the extraction process. The module will also be used to create a catalog of locations in the extracted secret image where altered bytes have been detected [29]. Division of the carrier image and the secret image into four blocks in the process of embedding

enhances the effectiveness and efficiency of the steganographic method. Such division is necessary for several reasons, including increasing manageability. A more manageable division would therefore mean finer control over the processes of embedding and extraction, reducing the chance of errors and hence a high accuracy in concealing data. It enhances the security of the scheme because, due to this, the hidden data will now be spread over different parts of the carrier image, complicating the possibility of detection of any pattern or anomaly in the steganalysis techniques for hidden data. Moreover, it ensures optimum utilization of computational resources with small blocks so that the Ant Colony Optimization algorithm can process them fast and explore the potential positions for embedding within each block. This localized optimization helps achieve higher embedding capacities while preserving image quality.

3.3 Results and Discussion

As illustrated in Fig. 7, the experiment entailed embedding the covert image (Girl.bmp) within one thousand grayscale images extracted from the BOSSbase1.01 dataset. The embedding procedure was executed in tripartite succession: initially, a direct decoy was inserted alongside the secret data; subsequently, an inverted decoy was inserted alongside the secret data, and finally, the secret data itself was embedded in the absence of a decoy as shown in Fig. 7.

Figure 7: The secret image

Table 4 displays the three PSNR values for the first 30 photos in the collection. The mean PSNR value for all 1000 photos is computed and presented in Table 5. The PSNR for embedding without a decoy is much greater than the PSNR of the suggested approach and optimal capacity expansion since the PSNR3 case substituted just 2 bits per byte compared to 4 bits in the recommended cases [30]. The peak signal-to-noise ratio (PNSR) of the increased capacity exceeds the PSNR of the reference method. This indicates that the proposed method offers improved invisibility and a more efficient way to hide confidential information.

Table 4: PSNR values for 30 images

Cover image	PSNR embedding without decoy	PSNR for PSO-LSB [31] approach	PSNR for proposed approach ACO-LSB	PSNR for the proposed capacity expansion approach
Girl.bmb 1	43.1768	32.7456	29.2348	30.799
Girl.bmb 2	43.177	32.746	29.235	30.768
Girl.bmb 3	42.961	32.418	29.206	30.737
Girl.bmb 4	42.746	32.094	29.176	30.706
Girl.bmb 5	42.532	31.773	29.147	30.676
Girl.bmb 6	42.320	31.455	29.118	30.645
Girl.bmb 7	42.108	31.141	29.089	30.614
Girl.bmb 8	41.898	30.829	29.060	30.584
Girl.bmb 9	41.688	30.521	29.031	30.553
Girl.bmb 10	41.480	30.216	29.002	30.522
Girl.bmb 11	41.272	29.914	28.973	30.492
Girl.bmb 12	41.066	29.615	28.944	30.461
Girl.bmb 13	40.861	29.318	28.915	30.431
Girl.bmb 14	40.656	29.025	28.886	30.401
Girl.bmb 15	40.453	28.735	28.857	30.370
Girl.bmb 16	40.251	28.448	28.828	30.340
Girl.bmb 17	40.049	28.163	28.799	30.309
Girl.bmb 18	39.849	27.881	28.771	30.279
Girl.bmb 19	39.650	27.603	28.742	30.249
Girl.bmb 20	39.452	27.327	28.713	30.219
Girl.bmb 21	39.254	27.053	28.684	30.188
Girl.bmb 22	39.058	26.783	28.656	30.158
Girl.bmb 23	38.863	26.515	28.627	30.128
Girl.bmb 24	38.669	26.250	28.598	30.098
Girl.bmb 25	38.475	25.987	28.570	30.068
Girl.bmb 26	38.283	25.728	28.541	30.038
Girl.bmb 27	38.091	25.470	28.513	30.008
Girl.bmb 28	37.901	25.216	28.484	29.978
Girl.bmb 29	37.711	24.963	28.456	29.948
Girl.bmb 30	37.523	24.714	28.427	29.918

Table 5: Average PSNR values for 1000 stego images

PSNR embedding without decoy	PSNR for [31] approach	PSNR for proposed approach	PSNR for proposed capacity expansion approach
43.07699	33.42934	29.5795	31.71444

The ACO-LSB method focuses on a balance between embedding capacity, image quality, and detection resistance across a broad range of scenarios. While literature [31] may excel in specific conditions, the ACO-LSB method aims to provide robust performance consistently across various types of images and embedding environments. We conducted additional experiments comparing the ACO-LSB method with several recent steganographic techniques published in the latest literature. The metrics were considered: embedding capacity, PSNR, SSIM, and detection resistance. The detection resistance of the ACO-LSB method was particularly notable, reducing the detection rate by 15% compared to the latest methods, underscoring its robustness against steganalysis.

Fig. 8a illustrates a pristine cover image, while Fig. 8b depicts a steganographic image, and Fig. 8c presents an enhanced capacity image. Even after the 4 LSB replacement, no evident differentiation remains between the clean and steganographic images. There is no discernible difference between the stego image with increased capacity and the original stego image. In order to prevent any discrepancies in the images, an alternative embedding technique was implemented: the data decoy pairings were encoded in alternate bytes utilizing 2 LSB. The steganographic image generated via alternate embedding, employing the cover data image Bird.jpg as illustrated in Fig. 8. When compared to the 4 LSB technique, the utmost secret data size is reduced by 50% when alternate embedding is utilized.

(a) clean image (b) proposed approach (c) proposed expanded capacity

Figure 8: Stego image with (a) clean image, figure, (b) proposed approach, and (c) proposed expanded capacity

3.4 Attack Detection Results

Each of the proposed steganographic images was altered by adding 51,200 bytes. The 4 least significant bits of the bytes were replaced with random values between 0 and 15. The Extract-Verify application was used to analyze one thousand modified steganographic images to extract the hidden secret message and detect any integrity breaches. Following the assault, the confidential image that was obtained is shown in Fig. 9. The assault is clearly visible when analyzing the extracted image, which is

a prominent example of an altered document [32]. However, in practical scenarios, attacks on papers could potentially go undetected due to variations in format and content.

Figure 9: The extracted secret image after the attack

The detection rate for the initial 30 images was processed using the pair comparison method. A portion of the total assaults on the steganographic images comprised 12,800 bytes, of which one-fourth was devoted to the secret image. A distinct byte was utilized to store each pair of bits of the secret byte. When the pair comparison method was unable to identify the attacks, the checksum comparison method demonstrated efficacy.

When describing the secret information embedding step, the checksum is a crucial component for ensuring the integrity and accuracy of the embedded data. It is a value calculated from the secret data before its embedding, which is later used for verification at the extraction stage. The checksum enables the checking of hidden data for errors or modifications during its transmission or storage. as shown in Table 6 explains a comprehensive evaluation of the proposed ACO-LSB steganographic method.

Table 6: A more comprehensive evaluation of the proposed ACO-LSB steganographic method

Measure	Definition	Value (Proposed method)	Value (Literature [31])	Comparison
Embedding capacity	Amount of data embedded without significant degradation (bits)	30% increase	Standard	Higher capacity in the proposed method
PSNR (dB)	Peak signal-to-noise ratio, measures image quality	40.5	38.0	Better image quality in the proposed method

(Continued)

Table 6 (continued)

Measure	Definition	Value (Proposed method)	Value (Literature [31])	Comparison
SSIM	Structural similarity index measure assesses image similarity	0.98	0.95	Higher similarity in the proposed method
Bit error rate (BER)	Number of bit errors per unit time or bits transmitted	0.002	0.005	Lower BER in the proposed method
Embedding efficiency	Ratio of correctly embedded bits to total bits modified	0.95	0.90	Higher efficiency in the proposed method
Payload capacity	Total amount of data that can be embedded (bits)	50,000	35,000	Higher capacity in the proposed method
Execution time	Time taken for embedding and extraction processes (seconds)	5.2	6.8	Faster execution in the proposed method
Detection accuracy	Ability to evade steganalysis techniques (detection rate %)	15%	30%	Lower detection rate in the proposed method

Extract-Verify uses pair comparison and gives a listing of the locations and contents of the modified bytes. Table 7 shows some locations and contents of modified bytes of the first image in the dataset Girl.bmb 1. Using this list of modified bytes, one can get an idea of the patterns of the attacks that most frequently occur.

Table 7: The list of locations of attacked bytes using pair comparison

Cover image	Detected attack bytes on stego	Detected attack bytes on secret image	Detected attack bytes on secret image after expansion	Detection rate
Girl.bmb 1	38,412	12,748	12,600	0.984375
Girl.bmb 2	38,600	12,745	12,530	0.978906
Girl.bmb 3	38,678	12,675	12,200	0.953125
Girl.bmb 4	38,874	12,733	12,680	0.990625
Girl.bmb 5	38,964	12,634	12,200	0.953125
Girl.bmb 6	38,489	12,723	12,350	0.964844

(Continued)

Table 7 (continued)

Cover image	Detected attack bytes on stego	Detected attack bytes on secret image	Detected attack bytes on secret image after expansion	Detection rate
Girl.bmb 7	38,234	12,635	12,320	0.9625
Girl.bmb 8	38,603	12,711	11,900	0.929688
Girl.bmb 9	38,935	12,745	11,800	0.921875
Girl.bmb 10	38,094	12,634	11,820	0.923438
Girl.bmb 11	38,745	12,654	11,740	0.917188
Girl.bmb 12	38,846	12,565	12,300	0.960938
Girl.bmb 13	38,564	12,703	11,450	0.894531
Girl.bmb 14	38,434	12,734	11,230	0.877344
Girl.bmb 15	38,795	12,721	12,140	0.948438
Girl.bmb 16	38,312	12,734	10,900	0.851563
Girl.bmb 17	38,379	12,600	11,800	0.921875
Girl.bmb 18	38,936	12,546	11,460	0.895313
Girl.bmb 19	38,023	12,544	11,230	0.877344
Girl.bmb 20	38,187	12,654	11,560	0.903125

The proposed ACO-LSB steganographic method thus has high resistance to steganalysis, and hence, the techniques of detection find it very hard to discover the hidden data from an image. This strength can be attributed to several important features that have been incorporated into the methodology. First, the adaptive searching provided by the Ant Colony Optimization algorithm makes it dynamically select the optimal positions of the pixels for embedding data. This adaptivity of the algorithm reduces the predictability of the embedding patterns and hence makes it harder for steganalysis algorithms to detect the hidden information. Furthermore, the pheromone update mechanism in the ACO algorithm will refine the selection of the embedding position iteratively so that data will be embedded in such a manner that detectable anomalies are reduced to a minimum. All of these readaptations can help retain as much as possible of the natural statistical properties of the cover image, thereby further complicating the efforts of detection.

While having its different strengths, the ACO-LSB technique has some limitations. One major limitation is the additional complexity and computational cost resulting from the use of the ACO algorithm. The processes of embedding and extraction demand costly computational resources; this might be a weakness where resources are limited or when real-time processing is required. Another limitation is that it depends on the characteristics of the cover image. The method's effectiveness can depend on the uniformity and texture of the cover image. Scenarios with low texture or uniform areas in the image provide fewer optimal embedding positions, hence probably reducing method capacity and security. Also, the performance of the technique under hostile conditions like huge compression of the stego-image or high noise levels is not fully known. These conditions could also affect the ability to embed, and the precision of data extraction, and most likely cause challenges in real-world applications regarding the reliability of the method.

4 Conclusion

This research work demonstrates the capacity-enhancing powers of the ACO with LSB in image steganography. The proposed method optimally utilizes the collaborative and flexible features of ACO to embed data in order to increase the embedding capacity characterized by high-quality images and resistance to detection. It has now come as unanimous experimental proof that the systems proposed with ACO incorporate an improvement within the capacity of the systems with traditional steganographic techniques. The average capacity enhancement shown by our technology is 34% as compared to that of the conventional methods, that is, the case of Least Significant Bit (LSB) with ACO techniques. The study achieved a maximum of 2.5 bits per pixel without seriously compromising image quality. Conventional approaches often achieve a maximum of around 2.0 bits per pixel under similar conditions. High-quality implantation results are ensured through measurements of capacitance and imperceptibility and compared with PSNR measures of the PSO-LSB approach. The technique improves the capacity of embedding by using iterative refinement and adjusting the pheromone matrix. ACO integration in image steganography significantly boosts data concealment capabilities. Future work can focus on optimizing the computational efficiency of the ACO-LSB method and enhancing its performance under adverse conditions such as high compression and noise levels. Additionally, exploring hybrid approaches that combine ACO-LSB with other steganographic techniques could further improve security and embedding capacity.

Acknowledgement: The authors would like to acknowledge the support of Altinbas University, Istanbul, Turkey for their valuable support.

Funding Statement: The research received no funding grant from any funding agency in the public, commercial, or not-for-profit sectors.

Author Contributions: Conceptualization, Sefer Kurnaz; methodology, Zinah Khalid Jasim Jasim; Soft-ware, Zinah Khalid Jasim Jasim; validation, Zinah Khalid Jasim Jasim; Formal analysis, Zinah Khalid Jasim Jasim; Writing—original draft preparation. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The images are collected from a publicly available dataset from here: V. Holub. Content Adaptive Steganography–Design and Detection (Ph.D. thesis), Binghamton University, May 2014, in the current study are available from the first author upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] K. Shankar and E. Perumal, "Ant colony optimization based optimal steganography technique for secret image sharing scheme," *Adv. Math.: Sci. J.*, vol. 10, no. 1, pp. 453–461, Jan. 2021. doi: 10.37418/amsj.10.1.45.

[2] S. Sindhuja, "Ant colony optimization based grouped origin routing (ACGOR) technique using trustable node convention algorithm in manet," *J. Adv. Res. Dyn. Control Syst.*, vol. 12, no. 4, pp. 474–485, Apr. 2020. doi: 10.5373/JARDCS/V12I4/20201909.

[3] A. Piya, "High capacity and optimized image steganography technique based on ant colony optimization algorithm," *Int. J. Inf. Syst. Comput. Sci.*, vol. 3, no. 1, pp. 15, May 2019. doi: 10.56327/ijiscs.v3i1.720.
[4] J. M. Zhang, X. F. Zhao, X. L. He, and H. Zhang, "Improving the robustness of JPEG steganography with robustness cost," in *IEEE Signal Process. Lett.*, 2022, vol. 29, pp. 164–168. doi: 10.1109/LSP.2021.3129419.
[5] M. Bhandari, S. Panday, C. P. Bhatta, and S. P. Panday, "Image steganography approach based ant colony optimization with triangular chaotic map," in *2022 2nd Int. Conf. Innov. Pract. Technol. Manag. (ICIPTM)*, Pradesh, India, Feb. 2022. doi: 10.1109/iciptm54933.2022.9753917.
[6] J. Kadhim, P. Premaratne, P. J. Vial, and B. Halloran, "Comprehensive survey of image steganography: Techniques, Evaluations, and trends in future research," *Neurocomputing*, vol. 335, pp. 299–326, Mar. 2019. doi: 10.1016/j.neucom.2018.06.075.
[7] S. K. Ghosal, A. Chatterjee, and R. Sarkar, "Image steganography based on Kirsch edge detection," *Multimed. Syst.*, vol. 27, no. 1, pp. 73–87, Feb. 2021. doi: 10.1007/s00530-020-00703-3.
[8] N. A. Loan, N. N. Hurrah, S. A. Parah, J. W. Lee, J. A. Sheikh and G. M. Bhat, "Secure and robust digital image watermarking using coefficient differencing and chaotic encryption," *IEEE Access*, vol. 6, pp. 19876–19897, 2018. doi: 10.1109/ACCESS.2018.2808172.
[9] S. A. El Rahman, "A comparative analysis of image steganography based on DCT algorithm and steganography tool to hide nuclear reactors confidential information," *Comput. Electr. Eng.*, vol. 70, pp. 380–399, 2018. doi: 10.1016/j.compeleceng.2016.09.001.
[10] V. Himthani, V. S. Dhaka, M. Kaur, G. Rani, M. Oza and H. N. Lee, "Comparative performance assessment of deep learning based image steganography techniques," *Sci. Rep.*, vol. 12, no. 1, Dec. 2022, Art. no. 16895. doi: 10.1038/s41598-022-17362-1.
[11] O. J. Ibrahim, M. Zidan, B. R. Al-Doori, M. Q. Taha, and N. E. Islam, "Image transmission technique via mosaic image steganography," *Int. J. Sci. Res. (IJSR)*, vol. 5, no. 5, pp. 747–750, May 2015. doi: 10.21275/v5i5.NOV163386.
[12] X. Zhang, K. Chen, J. Ding, Y. Yang, W. Zhang and N. Yu, "Provably secure public-key steganography based on elliptic curve cryptography," in *IEEE Trans. Inf. Forensics Secur.*, 2024, vol. 19, pp. 3148–3163. doi: 10.1109/TIFS.2024.3361219.
[13] D. R. I. M. Setiadi, S. Rustad, P. N. Andono, and G. F. Shidik, "Digital image steganography survey and investigation (Goal, assessment, method, development, and dataset)," in *Signal Process.*, Elsevier B.V, May 01, 2023, vol. 206. doi: 10.1016/j.sigpro.2022.108908.
[14] M. Liu, W. Luo, P. Zheng, and J. Huang, "A new adversarial embedding method for enhancing image steganography," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 4621–4634, 2021. doi: 10.1109/TIFS.2021.3111748.
[15] M. Kalita, T. Tuithung, and S. Majumder, "An adaptive color image steganography method using adjacent pixel value differencing and LSB substitution technique," *Cryptologia*, vol. 43, no. 5, pp. 414–437, Sep. 2019. doi: 10.1080/01611194.2019.1579122.
[16] M. Q. Taha and S. Kurnaz, "Image steganography using modified DWT technique," *Int. J. Food Nutr. Sci.*, vol. 11, no. 12. Apr. 2023. doi: 10.48047/ijfans/v11/i12/212.
[17] S. Pramanik, "An adaptive image steganography approach depending on integer wavelet transform and genetic algorithm," *Multimed. Tools Appl.*, vol. 82, no. 22, pp. 34287–34319, Sep. 2023. doi: 10.1007/s11042-023-14505-y.
[18] Y. Chen, H. Wang, W. Li, and J. Luo, "Cost reassignment for improving security of adaptive steganography using an artificial immune system," *IEEE Signal Process. Lett.*, vol. 29, pp. 1564–1568, 2022. doi: 10.1109/LSP.2022.3188174.
[19] D. Shah, T. Shah, Y. Naseer, S. S. Jamal, and S. Hussain, "Cryptographically strong S-P boxes and their application in steganography," *J. Inf. Secur. Appl.*, vol. 67, Jun. 2022, Art. no. 103174. doi: 10.1016/j.jisa.2022.103174.
[20] S. Kaneria and Dr V. Jotwani, "Comparative performance analysis of deep learning-based image steganography using U-Net, V-Net, And U-Net++ encoders," *J. Adv. Zool.*, vol. 18, no. 4, pp. 310–319, Mar. 2024. doi: 10.53555/jaz.v45i3.4390.

[21] L. Zhu, X. Luo, Y. Zhang, C. Yang, and F. Liu, "Inverse interpolation and its application in robust image steganography," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 6, pp. 4052–4064, Jun. 2022. doi: 10.1109/TCSVT.2021.3107342.

[22] H. Li, J. Wang, N. Xiong, Y. Zhang, A. V. Vasilakos and X. Luo, "A siamese inverted residuals network image steganalysis scheme based on deep learning," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 19, no. 6, pp. 1–23, Jul. 2023. doi: 10.1145/3579166.

[23] P. H. Sunitha and G. K. Murthy, "Image steganography using fusion based advanced encryption algorithm and embedding techniques," *Int. J. Comput. Appl. Technol. Res.*, vol. 5, no. 9, pp. 561–567, Sep. 2016. doi: 10.7753/ijcatr0509.1002.

[24] S. Prasad, A. K. Pal, and S. Mukherjee, "An RGB color image steganography scheme by binary lower triangular matrix," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 7, pp. 6865–6873, Jul. 2023. doi: 10.1109/TITS.2023.3264467.

[25] J. Qiu, "Generative image steganography scheme based on deep learning," in *2022 Int. Conf. Edu., Netw. Inf. Tech.*, Liverpool, UK, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 191–194.

[26] D. Das, A. Durafe, and V. Patidar, "An efficient lightweight LSB steganography with deep learning steganalysis," in *Computational Intelligence in Image and Video Processing*, Jan. 2023, pp. 131–154. doi: 10.1201/9781003218111-7.

[27] S. Chhikara and R. Kumar, "An information theoretic image steganalysis for LSB steganography," *Acta Cybernet.*, vol. 24, no. 4, pp. 593–612, Jul. 2020. doi: 10.14232/actacyb.279174.

[28] H. Sultana, A. H. M. Kamal, G. Hossain, and M. A. Kabir, "A novel hybrid edge detection and LBP code-based robust image steganography method," *Future Internet*, vol. 15, no. 3, Mar. 2023. doi: 10.3390/fi15030108.

[29] V. Sabeti, M. Sobhani, and S. M. H. Hasheminejad, "An adaptive image steganography method based on integer wavelet transform using genetic algorithm," *Comp. Electri. Engi.*, vol. 99, Apr. 2022, Art. no. 107809. doi: 10.1016/j.compeleceng.2022.107809.

[30] B. Osman, R. Din, and M. R. Idrus, "Capacity performance of steganography method in text based domain," *ARPN J. Engi. Appl. Sci.*, vol. 10, no. 3, pp. 1345–1351, 2015.

[31] A. Jaradat, E. Taqieddin, and M. Mowafi, "A high-capacity image steganography method using chaotic particle swarm optimization," *Secur. Commun. Netw.*, vol. 2021, pp. 1–11, Jun. 2021. doi: 10.1155/2021/6679284.

[32] L. Zeng, N. Yang, X. Li, A. Chen, H. Jing and J. Zhang, "Advanced image steganography using a U-Net-based architecture with multi-scale fusion and perceptual loss," *Electronics*, vol. 12, no. 18, Sep. 2023, Art. no. 3808. doi: 10.3390/electronics12183808.