Academic Journal of Nawroz University (AJNU), Vol.12, No.3, 2023

https://doi.org/10.25007/ajnu.v12n3a1891

A Blind Image Steganography Algorithm Based on Knight Tour Algorithm and QR Codes

Younis Mohammed Younis[1], Ramadhan J. Mstafa[1,2], Haval I. Hussein[1], Ahmed L. Alyousify[1]

[1]Department of Computer Science, University of Zakho Kurdistan Region, Iraq
[2]Department of Computer Science, Nawroz University, Duhok, KRG - Iraq

ABSTRACT

Internet proliferation and technological progress have made multimedia information quickly accessible, but they have also posed a threat to privacy and security. Researchers have been interested in digital images due to their capacity to store large amounts of data due to the possibility of protecting sensitive information through digital steganography. Despite their visual imperceptibility, robustness, and ability to embed information, existing image steganography techniques face several challenges. To overcome these challenges, a novel image steganography approach based on a blind model strategy has been proposed for hiding covert messages. The model consists of two stages: embedding and extracting. In the embedding stage, a suitable cover image is selected using the FAST feature point detector. A text message is then converted to a QR code and embedded in the feature points' neighbors using knight tour steps in a chess game. The result is a stego image that appears identical to the original cover image but contains the secret message in its feature points' neighbors. The extracting stage involves finding the feature points and extracting the QR code to obtain the original text message. Since the feature points were not altered during the embedding process, the proposed model is known as a blind model. This approach eliminates the need for the original cover image during the extracting stage. The proposed model was evaluated using several metrics, including the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM). The results demonstrate that the proposed algorithm can effectively embed and extract secret messages with high accuracy while maintaining the visual quality of the cover image with 100% of (SSIM), and 73.48 as an average of (PSNR).

Keywords: Image Steganography, FAST feature points, QR code, blind model, knight tour algorithm.

1. Introduction

In today's digital world, data security has become a major concern due to the increasing risk of unauthorized access to confidential information. This has led to the emergence of various techniques for protecting sensitive data, including steganography and cryptography (Mstafa & Elleithy, 2016). Steganography is the science of hiding secret data within a medium, such as an image or audio file, while cryptography involves converting a secret message into an unreadable format. Among these techniques, image steganography has become an essential component of modern-day data security due to the rise in the number of digital images transmitted over the internet (Huang et al., 2019; Mstafa & Elleithy, 2016). The first step in image steganography is to choose an appropriate cover image that will be used to hide confidential information. Once the cover image is selected, the next step is embedding the secret message without noticeably affecting the quality of the cover image. The final step in image steganography is to send the modified image with the hidden message to the intended recipient, who can then extract the secret message using the same steganography technique used for embedding, as shown in Fig.1.

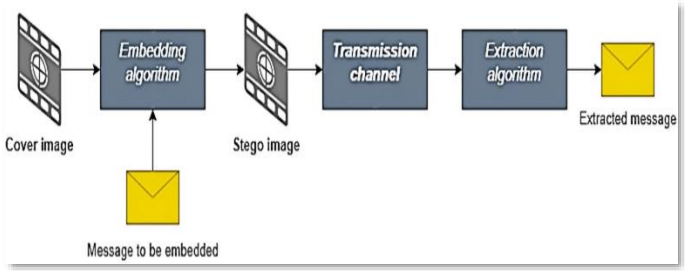

Fig 1. Simple Image Steganography Diagram.

Image steganography techniques can be broadly classified into two categories: transform domain methods and spatial domain methods, as illustrated in Fig. 2. Transform domain methods involve transforming the cover image into the frequency domain, selecting specific coefficients to replace with the confidential message, and then converting the domain with the altered coefficients back into the spatial domain. In contrast, spatial domain

methods directly embed the confidential message within the cover image (Mohammed et al., 2023; Mstafa & Elleithy, 2016). Discrete Cosine Transform (DCT) and Least Significant Bit (LSB) technique are among the most popular approaches of the former and later ones due to their simplicity and low computation cost.

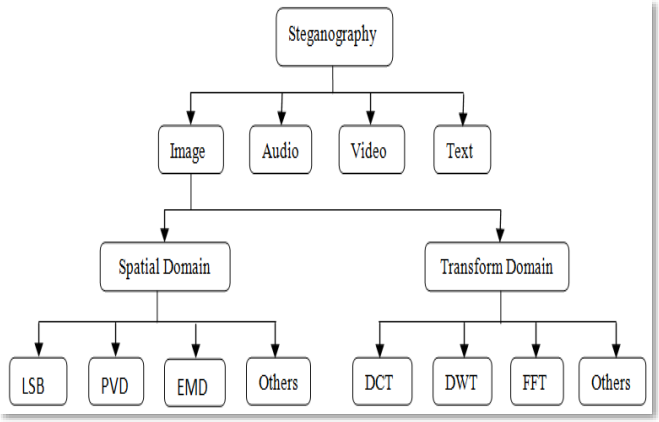

Fig 2. Image Steganography Technique.

The success of a steganography algorithm depends on several factors, including embedding capacity, security, imperceptibility, and robustness. These factors determine the size of the secret message that can be hidden, how secure the algorithm is against attacks, how well the message can be concealed without noticeable distortions, and how well the algorithm can withstand image processing attacks, compression attacks, and so on. It is essential to consider all of these factors when selecting a suitable steganography algorithm for effective and secure data concealment (Huang et al., 2019; Mohammed et al., 2023; Mstafa & Elleithy, 2016; ur Rehman et al., 2019).

Nowadays, the use of regions of interest has become increasingly popular in steganography, especially when using images as covers. One effective approach to identify spatial pixels is via corner detection, which is a critical step in various computer vision applications, such as object tracking. This method involves identifying local autocorrelation in an image at a location that exhibits a distinct peak property, thereby defining a corner. Corner detection is widely utilized in image processing and machine vision research, as corners are less sensitive to changes in appearance to the human eye. A reliable corner detector should possess the attributes of consistency, accuracy, and speed. There are various algorithms available for detecting corner feature points, including the FAST algorithm. For instance, Sultana et al. (Sultana et al., 2023) suggested a new approach to hide secret information into digital images called hybrid image steganography. Their approach combines two properties of images, edges and local binary pattern (LBP) codes, to make the hidden information difficult to detect by cyber-attacks. Edge detector algorithm was used to detect edges in the cover image for embedding purpose. Whereas, LBP algorithm was utilized to scramble the secret message prior to embedding. Experimental results showed that their proposed method outperforms other methods in terms of measuring perceptual transparency and embedding capacity while protecting the privacy of implanted data.

Huang et al.(Huang et al., 2019) Presented an encoder-decoder framework using an adversarial discriminator for steganography, which conceals messages or images in natural images. The message was first embedded into a QR code to improve fault-tolerance. Furthermore, a texture-based loss was introduced to learn invisible perturbations of images. Lastly, a truncated layer and moment layer were used to handle stego image distortions and varisized images, respectively. Experimental results showed that their proposed method outperforms existing methods in terms of security and visual quality of stego images.

Chakraborty and Jalal (Chakraborty & Jalal, 2020) addressed the issue of steganography methods that embed secret information into cover images without noticeable distortion. Existing methods may preserve the global structure of the cover, but their embedding capacity is limited. They presented a feature-based steganography technique that used LBP to hide secret image bits while preserving the local relationships in the cover. The suggested method is compared to state-of-the-art LSB based methods and is shown to outperform them. This method preserves the local structure of the cover and is robust against feature-based steganalysis. The paper suggests that this method can be extended by introducing payload-specific handcrafted descriptors for embedding data.

Setiadi (Setiadi, 2021) recommended the use of two widely used image quality assessment tools, PSNR and SSIM, in the context of steganography. While PSNR has been the more commonly used tool due to its simplicity and reliability, SSIM has been designed to better suit the human visual system. The study analyzes the results of using PSNR and SSIM to measure imperceptibility in three spatial domain image steganography methods, and concludes that SSIM is a better measure in all aspects. The study recommends the use of SSIM in future steganographic research, particularly for measuring imperceptibility in steganographic images.

Shanthakumari and Malliga (Shanthakumari & Malliga, 2020) implied a new approach to data security using steganography and cryptography techniques. The goal is to insert confidential information into a cover

object, such as an image, with a high level of data embedding capacity and improved security. The Elliptic Curve Cryptography algorithm is used to encrypt the hidden information, which is then inserted into the cover object using the LSB Inversion algorithm. The combination of these techniques has been tested against various steganalysis attacks, and the results show strong resistance to these attacks. The proposed LSBI algorithm is able to insert 6 bits of data in each 4 pixels using the gray code standard. The data is encoded using Elliptic Curve Cryptography for an additional layer of security. The experimental results demonstrate that the proposed method has a higher data embedding capacity and better quality stego images compared to other methods. This approach can also be applied to other types of cover objects, such as audio and video, for future work.

Astuti et al. (Astuti et al., 2020) introduced on the imperceptibility of steganography, which is a critical aspect of hiding messages within images. The previous research showed that the bit flipping method was effective in increasing imperceptibility in grayscale images by around 9dB. However, since humans are more sensitive to color images than grayscale images, this method needs to be retested on color images. This study tests the bit flipping method on color images in the RGB format with a message capacity of 1 bit per pixel. The results show that the method works well on color images with a maximum PSNR increase of more than 13 dBs. Therefore, this research shows that the bit flipping method is effective in both grayscale and color images for steganography.

Despite the encouraging results reported in the literature, there is still room for further enhancement. For instance, all the aforementioned related approaches necessitate the availability of the original cover image during the extraction process, which are deemed non-blind approaches. This is because they directly utilize the feature points obtained for embedding, thereby rendering it arduous for the receiver to detect the same feature points. Additionally, these approaches are irreversible, implying that certain portions of the confidential message get lost.

Inspired by the above-mentioned issues, in this research propose a novel blind approach for image steganography based on corner detector and knight tour algorithms. The corner detector algorithm was initially utilized to locate the corner pixels within the cover image. Subsequently, the knight tour algorithm was used to identify the neighboring pixels surrounded by the corner pixels for embedding purposes. This would help in recovering the secret message accurately without any loss as well as it does not require the original cover image during the extraction process. Hence, the proposed approach is deemed a blind approach. Furthermore, the proposed method used the knight tour algorithm because recently, it has been found that it can be used as a method for identifying the pixels of interest as in (Bhatia, 2019), which helps gain high imperceptibility.

The main contribution of this research can be summarized as follows:

The main contribution of this research is a combination of the knight tour algorithm with the FAST corner detector to embed data without damaging the corner pixel.

Our method embeds data only in the corners' neighbors, preserving the integrity of the corner pixel and enabling the blind model to our method.

Unlike existing approaches, our method achieves a 100% message recovery rate, whereas their method suffers from message loss.

The structure of this paper is outlined below. In Section 2, the Preliminaries and proposed models that improve security are presented. In Section 3, The Proposed Method is described in detail. In Section 4, are present and discuss the results of our experiments, including the dataset used and the evaluation metrics used. As well as, in the same section of proposed paper it conduct a comprehensive performance analysis of our models in comparison to other established models in the literature. Finally, in Section 5, conclusion the paper.

2. PRELIMINARIES

2.1 QR Code

QR (Quick Response) codes can be customized with logos or images, and are popular due to their ease of use and versatility. They can be read by smartphones and other mobile devices with a camera and a QR code reader app, making them convenient for providing additional information or functionality in various contexts such as advertising, retail, or event management. These codes are two-dimensional barcodes that were first created in Japan in the mid-1990s for use in the automotive industry, and are now used in a variety of applications. They consist of a matrix of black and white squares arranged in a specific pattern to encode data. QR codes can store different types of data, including alphanumeric characters, binary data, and images, and can link to websites, send text messages, or provide contact information(Bajpai, 2015).

Traditional QR codes are susceptible to manipulation, which can compromise the integrity of their content. To address this issue, our methodology incorporates QR code version 3 due to its advanced security features. One of the most significant features of QR code version 3 is its ability to recover messages even if up to 30% of the data is lost. This is achieved through the use of Reed-Solomon error correction codes, which allow the recovery of the original message even if some data is missing or corrupted. By using QR code version 3, it increase the security of our methodology and the integrity of the encoded information, even in the presence of data loss. This has significant implications for applications requiring high levels of reliability and security, such as payment systems, inventory management, and identification systems.

There are different types of QR codes, including low, mid, and high_capacity codes as showed in fig 3, with different sizes and storage capacities. Low_capacity codes are version 1 and have 21x21 modules, mid capacity codes are version 2 and have 25x25 modules, and high_capacity codes are version 3 and have 29x29 modules.

- Low capacity QR codes (version 1) are 21x21 modules in size and can store up to 25 alphanumeric characters or 47 numeric digits.
- Mid capacity QR codes (version 2) are 25x25 modules in size and can store up to 47 alphanumeric characters or 77 numeric digits.
- High capacity QR codes (version 3) are 29x29 modules in size and can store up to 77 alphanumeric characters or 127 numeric digits.

Fig 3: Three versions of QR codes with their dimensions.

The QR code is generated using a specific algorithm that encodes the data into the pattern of black and white squares. The algorithm takes the input data, which can be text, a URL, or other types of data, and applies a series of encoding steps to generate the QR code.

Here the list of steps:

Step 1: Choose the data type and encoding mode.

Step 2: Encode the data using the chosen mode.

Step 3: Add error correction codes to the encoded data.

Step 4: Choose the QR code version and size.

Step 5: Generate the QR code matrix.

Step 6: Add timing patterns, alignment patterns, and format information to the QR code.

Step 7: Apply masking to the QR code matrix.

Step 8: Add the quiet zone around the QR code.

Step 9: Display the QR code.

The QR code algorithm uses several encoding modes to store different types of data, including numeric data, alphanumeric data, byte data, and kanji characters. The algorithm also includes error correction codes to ensure that the data can be read even if part of the code is damaged or obscured.

2.2 Knight Tour Algorithim

The knight's tour is a classic problem in chess, which involves finding a sequence of moves for a knight on an empty chessboard such that the knight visits every square exactly once. The knight's tour problem has been studied for centuries, and has been the subject of many algorithms and heuristics (Parberry, 1997).

One algorithm for solving the knight's tour problem is Warnsdorff's rule, which is a heuristic algorithm that is simple and efficient. The algorithm works by choosing the next move for the knight based on the number of unvisited squares that can be reached from each possible next position. The idea is to always choose the position that has the fewest unvisited squares that can be reached from it, in the hope that this will lead to a solution.

The algorithm can be described as follows:

1) Choose any square to start the tour.

2) From the current square, choose the next square to move to by selecting the square with the fewest unvisited squares that can be reached from it.

3) If there are multiple squares with the same number of unvisited squares that can be reached, choose one at random.

4) Move to the selected square, and mark it as visited.

5) Repeat steps 2-4 until all squares have been visited, or until no valid moves are available, fig 4 is illustrated the step of tour.

6) The time complexity of the Warnsdorff's rule algorithm is O(n^2), where n is the size of the chessboard (Parberry, 1997).

The use of Knight tour steps and the selection of the feature point as the starting point for each step serve two purposes. Firstly, they ensure that the value of the feature point remains constant, allowing for accurate neighbor recovery. The feature point is a critical component of our methodology, and its value must remain constant for reliable results. Incorporating Knight tour steps guarantees that the feature point is visited only once, ensuring the consistency of its value throughout the process.

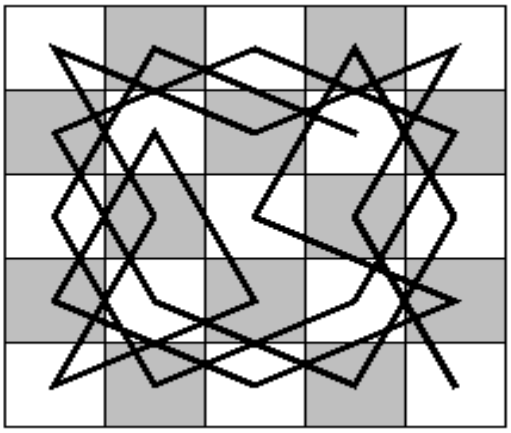

Fig 4 : The steps of the backtracking algorithm for the Knight's tour problem [12].

2.3 Fast Corner Detector Algorithm

The FAST (Features from Accelerated Segment Test) corner detector algorithm is a widely-used method for detecting corners in digital images, first introduced in 2006 by Edward Rosten and Tom Drummond. The algorithm works by comparing the brightness of a central pixel to that of surrounding pixels in a circular region around it. If the brightness of at least three of these pixels is significantly different from that of the central pixel, then the central pixel is classified as a corner (Trajković & Hedley, 1998).

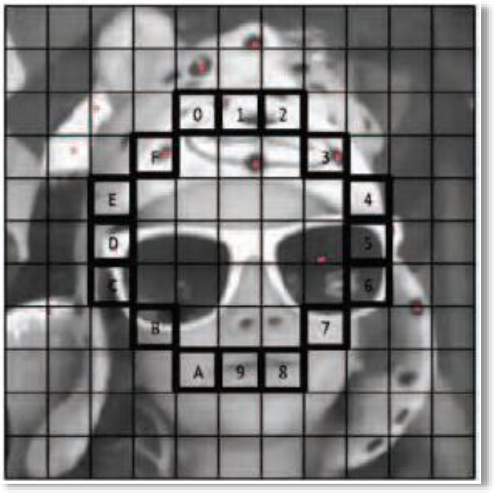

Fig 5: Pattern used by the algorithm to examine the surrounding pixels.

One of the main advantages of the FAST algorithm is its computational efficiency. It can process images in real-time, making it useful for applications that require fast corner detection, such as object tracking and motion estimation as showed in fig 5. The algorithm is also relatively robust to noise and can handle images with low contrast.

Despite its many benefits, the FAST algorithm does have some limitations. For example, it can be sensitive to changes in image scale and rotation, and it may produce false positives in regions of the image with high levels of noise(Trajković & Hedley, 1998).

To address some of these limitations, researchers have proposed various modifications and extensions to the FAST algorithm over the years. Some of these include using machine learning techniques to improve corner detection accuracy, or combining FAST with other corner detection algorithms for better overall performance.

This Pseudo code for show how FAST work:

For each pixel p in the image:

compute the score S(p) as the number of contiguous pixels on a ring of radius r

that are brighter or darker than the intensity of p by a threshold t

if S(p) >= n:

p is marked as a corner

apply non-maximal suppression to the corners

In addition, the use of Knight Tour steps and the selection of the feature point as the starting point for each step serve two purposes. Firstly, they ensure that the value of the feature point remains constant, allowing for accurate neighbor recovery. The feature point is a critical

component of our methodology, and its value must remain constant for reliable results. Incorporating Knight Tour steps guarantees that the feature point is visited only once, ensuring the consistency of its value throughout the process.

Secondly, selecting the feature point as the starting point for each step enhances the security of our methodology. Since the feature point value remains constant, any attempts to tamper with the data or manipulate the feature point value will be detected. This feature is particularly important in applications that require high levels of reliability and security, such as image recognition and authentication systems.

To summarize this section, our methodology's use of QR code version 3 and Knight Tour steps with the feature point as the starting point serves two critical purposes. It ensures accurate neighbor recovery by maintaining the constant value of the feature point, and enhances the methodology's security by detecting any attempts to tamper with the data or manipulate the feature point value. These features make our methodology highly reliable and secure, suitable for various applications that require high levels of reliability and security.

3. THE PROPOSED METHOD

The proposed model in this study is a steganography algorithm aimed at concealing a secret message within an image without compromising its perceptual quality. The model is divided into two primary stages: embedding and extracting.

The embedding stage starts with finding a cover image suitable for the task at hand. This process involves using a specific corner feature point detector called FAST feature point, which identifies distinctive points in the image where the intensity changes rapidly in different directions. If the number of detected pixel points is greater than the product of ((29*29)*2), the image is selected as the cover image. However, if the number is less, another cover image will be selected.

Next, the text message is converted into a Quick Response (QR) code with a dimension of (29x29). The QR code is a two-dimensional barcode capable of storing large amounts of information within a small space. The QR code is then embedded in the feature points' neighbors detected earlier using knight tour steps in a chess game. The feature point is used as the center of the steps, and the steps are made with the help of the border map. The border map is created using the same cover size and has all cells initialized with zeros. Once the feature points are detected, all feature points' locations are changed to 1. Moreover, to ensure no bits are missed and that the feature points are not affected, every embedded bit in the feature points' neighbor pixel is changed to 1.

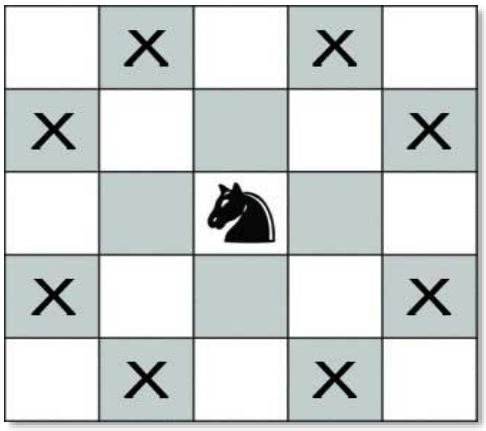

Fig 6: Knight Tour steps

The outcome of the embedding process is a stego image that appears similar to the cover image but has the secret message embedded in its feature points' neighbors. This stego image is now ready to be transmitted to the client.

The extracting stage occurs at the client side and involves three steps: feature point detection, QR code extraction, and text message retrieval. Since the feature points were not altered during the embedding process, the proposed model is referred to as a blind model that doesn't require the original cover image to locate the interest point.

The first step in the extraction stage is to detect the feature points, which is done by applying the same FAST feature point detector used in the embedding stage. Once the feature points are detected, the QR code is extracted from their neighbors using knight tour steps in a chess game, as shown in fig 6. Finally, the text message is obtained by scanning the extracted QR code. To clarify the embedding stages, fig 7 shows a diagram representing the proposed model's process.

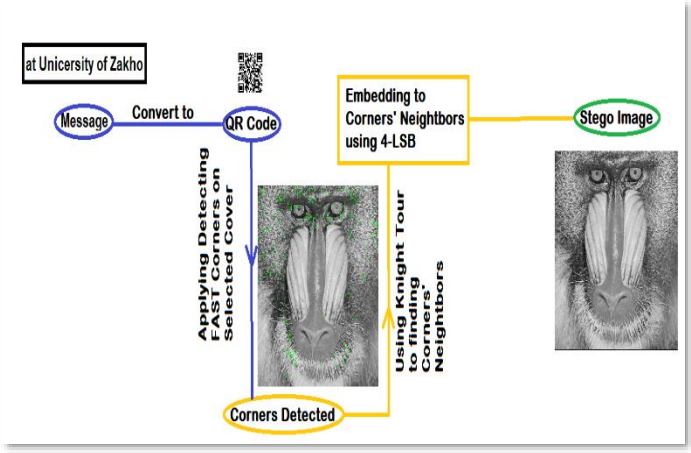

Fig 7: The proposed steganography model's embedding stages.

Once the QR code is extracted from the feature points' neighbors, the original text message can be obtained by

scanning the QR code. This process involves using a QR code reader software that decodes the QR code and retrieves the hidden message.

To evaluate the performance of the proposed model, several experiments were conducted using different cover images and text messages. The results showed that the model was able to embed and extract messages with high accuracy and without affecting the perceptual quality of the cover image.

In addition, to further improve the security of the proposed model, a password-based authentication mechanism can be added to ensure that only authorized users can extract the hidden message. This mechanism involves encrypting the text message with a password before embedding it in the cover image. The same password is required to extract the message from the stego image.

To better understand the proposed model and its processes, embedding stage as flowchart is presented in Fig 8 below:

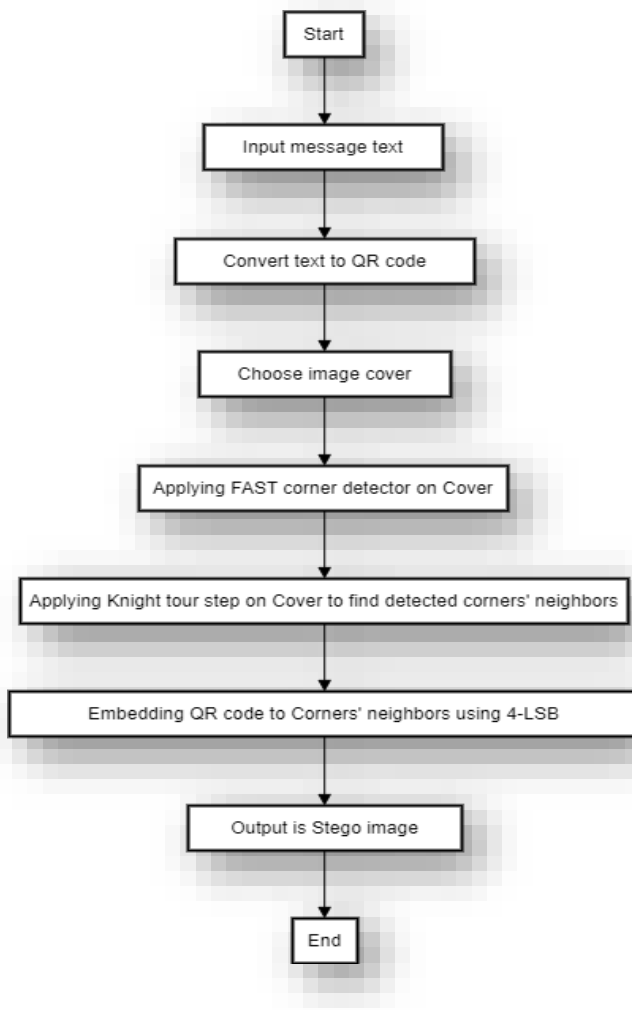

Fig 8: Embedding stage flowchart

In summary, the proposed model is a steganography algorithm that involves embedding a secret message in an image's feature points' neighbors without altering its perceptual quality. The model consists of two stages: embedding and extracting. The embedding stage involves finding a suitable cover image, converting the text message to a QR code, and embedding the QR code in the feature points' neighbors. The extracting stage involves finding the feature points, extracting the QR code from their neighbors, and obtaining the original text message. The proposed model's performance was evaluated through experiments and showed high accuracy in message embedding and extraction without affecting the cover image's perceptual quality.

Here is a more detailed description of the steps involved in the proposed steganography model:

Embedding Stage:

1) Feature Point Detection: The embedding stage begins by detecting feature points in the cover image using the FAST feature point detector. This detector searches for points in the image where the intensity changes rapidly in different directions.
2) Cover Image Selection: If the number of detected feature points is greater than the result of the multiplication of ((29*29)*2), the image is selected as the cover image. Otherwise, another cover image will be chosen.
3) QR Code Generation: The text message is then converted to a QR code with a dimension of (29x29). This is done to store large amounts of information in a small space.
4) Embedding QR Code: The QR code is embedded in the feature points' neighbors using knight tour steps in a chess game. The feature point serves as the center of the steps, and the steps are made with the border map.
5) Change Detection: The border map is created with the same cover size and zeros all cells. After detecting the feature points, all feature points' locations change to 1. Additionally, every embedded bit in the feature points' neighbor pixel is also changed to 1 to ensure that no bits are missed and does not affect the feature points.
6) Stego Image Generation: The result of the embedding process is a stego image that appears the same as the cover image but contains the secret message in its feature points' neighbors. This stego image is ready to be transmitted to the client.

Extracting Stage:

1) Feature Point Detection: The extracting stage involves finding the feature points in the stego image using the same FAST feature point detector used in the embedding stage.
2) QR Code Extraction: Once the feature points are detected, the QR code is extracted from their neighbors using knight tour steps in a chess game.
3) Text Message Recovery: Finally, the text message is obtained by scanning the extracted QR code.

4. RESULTS AND DISCUSSION

This section presents the comprehensive evaluation of our proposed Blind Model Image Steganography Algorithm based on FAST Feature Points and QR Code Embedding, using widely accepted performance metrics. The proposed models were implemented in the Matlab programming language, running on a computer equipped with an Intel(R) Core (TM) i7-10510U CPU @ 1.80GHz, 8 GB of RAM, and Windows 11 as the operating system. The results obtained by the proposed models are reported and discussed in the following subsections, followed by a comparative analysis between our proposed models and the latest existing ones. These findings contribute to the advancement of the field by providing insights into the performance and efficacy of the proposed algorithm, and can guide future research in this area.

4.1 Dataset

For the purpose of evaluating the proposed image steganography method, a dataset containing 7 frequently utilized image sequences in the PNG format, with grayscale color, was employed.

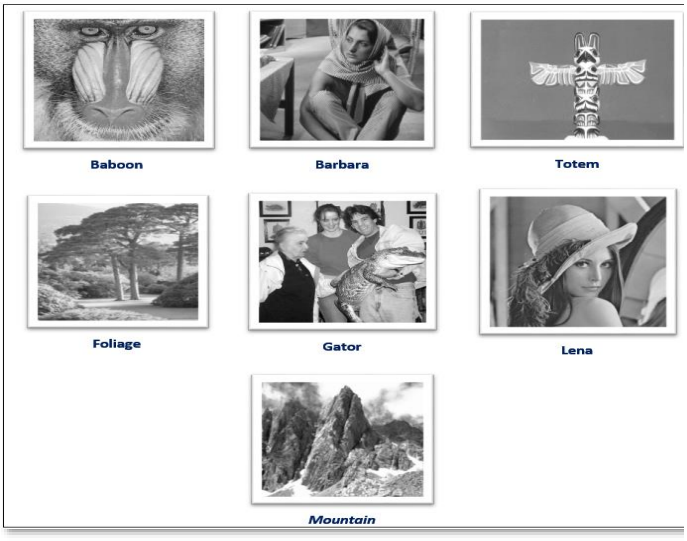

Fig 9: Dataset (Cover Images)

TABLE 1
Illustrating all Utilized Images Information

#	Cover Name	Dimensions	Total Pixel
1	Baboon	512x512	262144
2	Barbara	512x512	262144
3	Foliage	480X640	307200
4	Gator	633X621	393093
5	Lena	512x512	262144
6	Mountain	640x480	307200
7	Totem	640x480	307200

This dataset was obtained from reference (Database, n.d.).The individual cover images employed in this research are shown in Fig. 9, with a detailed description of each cover image presented in Table 1. The secret message used in this study was "at University of Zakho," which was transformed into a QR code of the third version, with a size of 29 by 29, as previously indicated.

4.2 Evaluation Metrics

Improving and innovating any image steganography method requires addressing the challenge of embedding as much information as possible in the cover image with a minimal noticeable difference in the stego image. To tackle this challenge, the proposed method underwent a rigorous evaluation and comparison with the most advanced and current approaches in the field, using three essential metrics: PNSR, similarity, and embedding capacity. These metrics were selected to ensure a comprehensive assessment of the performance and effectiveness of the proposed method and to contribute to the advancement of academic research in this area(Abboud et al., 2010; Luo & Yu, 2008; Mstafa & Elleithy, 2016).

$$PSNR = 10log_{10}\left(\frac{R^2}{MSE}\right) \quad (1)$$

In evaluating the proposed steganography method, the level of imperceptibility was assessed by evaluating the visual quality of the stego-images. Typically, the PSNR is utilized to measure this metric, which is expressed in decibels (dB) and can be calculated using Eq. (1). PSNR values below 30 dB indicate that the human eye can detect the distortion, while values of 40 dB or higher are desirable for a good steganography algorithm.

$$HR = \frac{Size\ of\ embedding\ message}{Image\ size} \times 100\% \quad (2)$$

In the context of steganography, embedding capacity refers to the maximum amount of information that can be concealed within a given cover image (Abboud et al., 2010; Younis et al., 2019).This metric is expressed in bits-per-pixel (bpp) and can be calculated using Eq. (2). Measuring the embedding capacity is crucial in evaluating the performance of a steganography method as it indicates the efficiency and effectiveness of the method in concealing information within the cover image.

$$SiM = \frac{\sum_{i=1}^{a}\sum_{j=1}^{b}[M(i,j) \times \widehat{M}(i,j)]}{\sqrt{\sum_{i=1}^{a}\sum_{j=1}^{b} M(i,j)^2} \times \sqrt{\sum_{i=1}^{a}\sum_{j=1}^{b} \widehat{M}(i,j)^2}} \quad (3)$$

The final metric utilized in the evaluation of the proposed steganography method is the similarity ratio between the original message and the extracted message. In measuring this metric, the SSIM function was utilized, expressed mathematically in Eq. (3). A higher similarity value denotes a better quality of the extracted image(Luo & Yu, 2008).This metric is significant in determining the accuracy of the steganography method in retrieving the hidden information, which is vital for the effectiveness of the method.

4.3 Results of the Proposed Method

This section presents the comprehensive evaluation of the proposed steganography method in terms of embedding capacity, PSNR, and SSIM on seven cover images. Prior to the embedding process, the corner points of each cover image were detected using the FAST algorithm with a threshold of 0.04 to enable the embedding of secret data. The threshold of 0.04 was selected based on empirical tests, in which more corner points could be detected in each cover image without compromising the quality of the resulting stego image. Additionally, the 4-LSBs method was utilized for embedding the secret information within the cover images.

TABLE 2
Number of Pixel, Detected Corners and its neighbors for each Image Cover

#	Cover Name	Total Pixel	Detected Corners	Corners' neighbors
1	Baboon	262144	812	6490
2	Barbara	262144	342	2730
3	Foliage	307200	454	3628
4	Gator	393093	711	5686
5	Lena	262144	1496	11962
6	Mountain	307200	665	5320
7	Totem	307200	302	2422

The number of corner points detected in each cover image is presented in Table 2. It can be observed that the number of detected corner points varies among different cover images. According to Table 2, the Lena and Totem cover images had the highest and lowest number of detected corner points, which were 1496 and 302, respectively. This difference in the number of detected corner points can be attributed to the dissimilarities in image structures among the various cover images.

TABLE 3
Proposed Method's Results

#	Cover Name	PSNR (dB)	Embedding Capacity (BPP)	Similarity
1	Baboon	50.3096	1.237869263	1
2	Barbara	50.3087	0.520706177	1
3	Foliage	51.3535	0.590494792	1
4	Gator	51.9631	0.723238521	1
5	Lena	50.6168	2.281570435	1
6	Mountain	50.9265	0.865885417	1
7	Totem	51.4664	0.394205729	1

Table 3 summarizes the embedding capacity, Similarity and PSNR performance of the proposed method on seven different cover images. The results in Table 3 demonstrate that the images 'Lena', 'Baboon', 'Gator', and 'Mountain' have a higher embedding capacity compared to the other images, since these images have abundant corner points. On the other hand, the embedding capacity for the low-corner point images 'Totem', 'Barbara', and 'Foliage' is not

as good as the embedding capacity for other images. This is expected because the area of the extracted region of interest (ROI) in these cover images is very small, due to the lack of corner points. The average embedding capacity for all images is 0.945, which indicates that the proposed method can hide a reasonable amount of information.

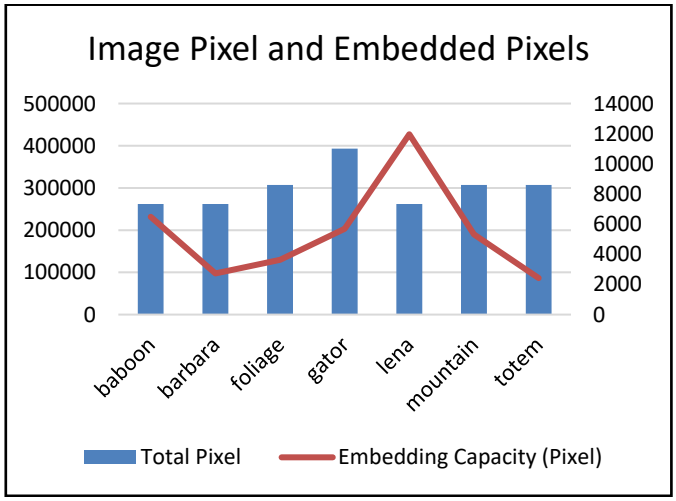

Fig 10: Compare between Image Pixels and Embedded Pixels

Moreover, all the PSNR values obtained for the tested images are greater than or equal to 50.3087 dB, which indicates the high perceptual invisibility of the proposed method. The average PSNR value for all images is 50.992 dB, which confirms that the proposed method is highly imperceptible. Therefore, it can be conclude that the proposed method offers a high degree of imperceptibility while providing an acceptable embedding capacity. Fig 10 presents the performance analysis of the proposed method in terms of the total number of detected corner points and the total number of embedded bits in each tested image. The figure demonstrates that the embedding capacity of each cover image increases whenever the number of detected corner points increases.

The proposed method operates on the cover image's feature points without making any changes to the feature point pixels. The method performs the embedding process in a sensitive manner to avoid any data loss during the message recovery process at the client side. As a result, the similarity value between the extracted message and the original message will be high, indicating that the message is recovered with high accuracy.

The primary goal of the proposed method is to achieve a high rate of recovery of the original message. In other words, the proposed method is designed to ensure that the stego image is as close as possible to the original cover image, making the stego image indistinguishable from the original image. This is achieved by embedding the message data in a way that does not affect the feature points, ensuring that the image's structure and content remain the same. Thus, the message can be retrieved with high accuracy at the client side, without any loss of data.

The proposed method is highly effective in maintaining the quality of the cover image while embedding the secret message. By preserving the feature points' integrity, the method ensures that the similarity between the original and extracted messages is high, resulting in a high accuracy of message recovery. This is especially important for applications where the message needs to be transmitted securely and accurately, such as in military and intelligence operations or in financial transactions.

In summary, the proposed method provides a reliable and secure means of transmitting confidential information while maintaining the privacy of the communication. The method's high accuracy of message recovery, robustness, and efficiency make it an attractive solution for various applications in the field of information security and privacy.

4.4 Comparative Analysis with State-of-the-Art Approaches

In this section, the proposed steganography method assessed the effectiveness by comparing it with existing methods in the literature in terms of its perceptual invisibility, embedding capacity, and similarity. To ensure a fair comparison, by using the same images as those used in the methods presented in [6]–[10] and compared our method with them. The proposed method used PSNR rate as a measure of comparison and found that our method achieved a very good PSNR rates compared to the methods presented in [6]–[10] across all the images used.

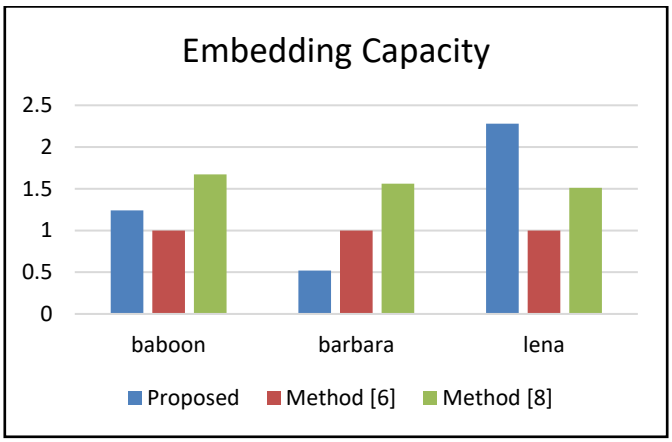

Fig 11: Embedding Capacity Comparative

Furthermore, compared our method with the method presented in [6] and found that our proposed method outperformed it in terms of the better value of PSNR rate. The proposed method used 1-LSB for embedding, which resulted in better PSNR rates compared to the methods that used 1-LSB. The bar chart clearly illustrates that our method has achieved a high (PSNR) of 86.37 in the Barbara cover, attributable to the substantial number of detected features in mentioned cover, as shown in fig 12.

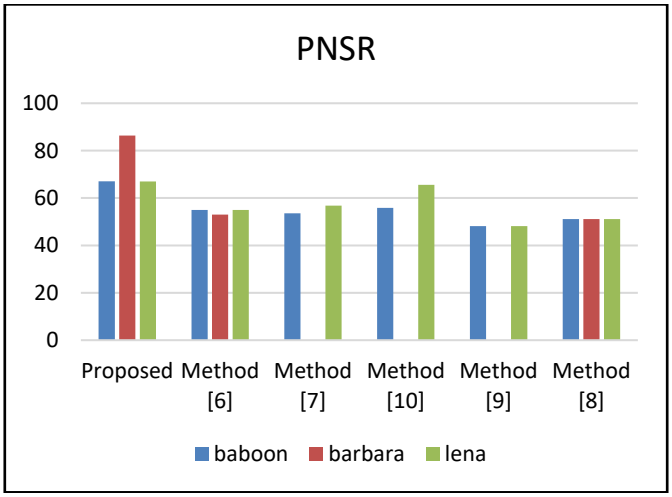

Fig 12: PNSR Comparative

We also compared the total number of embedded bits in our proposed method with the methods presented in [6]-[8], you can see at fig 11. Our proposed method attained the acceptable total number of embedded bits for QR code with size 27 by 27 compared to the methods presented in [6], [8]and was significantly higher than those obtained by them. Although the outperformed our proposed method in terms of the total number of embedded bits, our method was still better in terms of visual imperceptibility.

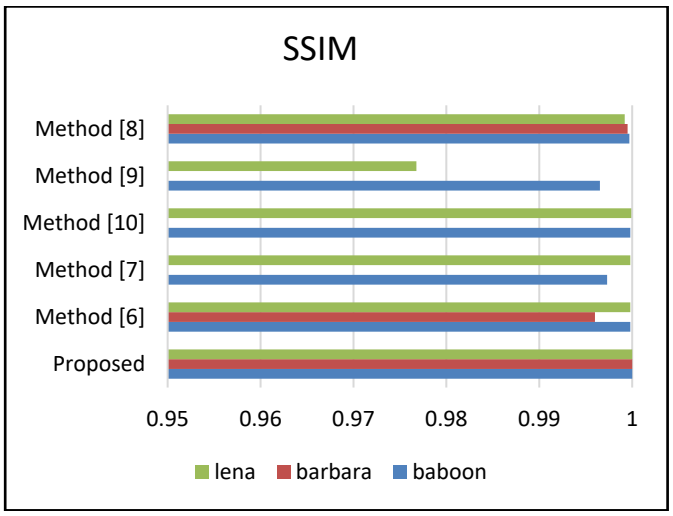

Fig 13: SSIM Comparative

Finally, we evaluated the similarity between the original message and the recovered message using our proposed method, and our results showed that our proposed method achieved high similarity values compared to the methods presented in [6]–[10] This demonstrates the effectiveness of our proposed method in embedding secret data while preserving the content of the original image, as showed in fig 13.

5. CONCLUSION

In this paper, a steganography algorithm is proposed to embed a secret message in an image's feature points' neighbors without altering its perceptual quality. The proposed model consists of two stages: embedding and extracting. The embedding stage involves finding a suitable cover image, detecting feature points in the cover image using the FAST feature point detector, converting the text message to a QR code, and embedding the QR code in the feature points' neighbors. This is done using knight tour steps in a chess game, where the feature point serves as the center of the steps, and the steps are made with the border map. The outcome of the embedding process is a stego image that appears the same as the cover image but contains the secret message in its feature points' neighbors.

The extracting stage involves finding the feature points using the same FAST feature point detector used in the embedding stage, extracting the QR code from their neighbors using knight tour steps in a chess game, and obtaining the original text message by scanning the extracted QR code. The proposed model is referred to as a blind model since the feature points were not altered during the embedding process, and the original cover image is not required to locate the interest point.

Experiments were conducted to evaluate the proposed model's performance using different cover images and text messages, and the results showed that the model was able to embed and extract messages with high accuracy without affecting the perceptual quality of the cover image. To further improve the model's security, a password-based authentication mechanism can be added to ensure that only authorized users can extract the hidden message.

In conclusion, the proposed steganography algorithm can successfully embed a secret message in an image's feature points' neighbors without compromising its perceptual quality. The proposed model's performance was evaluated through experiments and showed high accuracy in message embedding and extraction without affecting the cover image's perceptual quality. The model can be further improved by using video as a cover instead of image cover to enhance its embedding capacity, also by adding a password-based authentication mechanism to enhance its security.

6. REFERENCES

Abboud, G., Marean, J., & Yampolskiy, R. V. (2010). Steganography and Visual Cryptography in Computer Forensics. 2010 Fifth IEEE International Workshop on Systematic Approaches to Digital Forensic Engineering, 25–32. https://doi.org/10.1109/SADFE.2010.14

Astuti, E. Z., Setiadi, D. R. I. M., Rachmawanto, E. H., Sari, C. A., & Sarker, M. K. (2020). LSB-based Bit Flipping Methods for Color Image Steganography. Journal of Physics: Conference Series, 1501(1). https://doi.org/10.1088/1742-6596/1501/1/012019

Bajpai, M. K. (2015). Researching through QR codes in libraries. 2015 4th International Symposium on Emerging Trends and Technologies in Libraries and Information Services, 291–294. https://doi.org/10.1109/ETTLIS.2015.7048214

Bhatia, M. K. (2019). Knight tour for image steganography technique. International Journal of Engineering and Advanced Technology, 9(1), 1610–1613. https://doi.org/10.35940/ijeat.F8736.109119

Chakraborty, S., & Jalal, A. S. (2020). A novel local binary pattern based blind feature image steganography. Multimedia Tools and Applications, 79(27–28), 19561–19574. https://doi.org/10.1007/s11042-020-08828-3

Database, T. U.-S. I. (n.d.). DATA Set. The USC-SIPI Image Database. https://sipi.usc.edu/database/database.php?volume=misc&image=12#top

Huang, J., Cheng, S., Lou, S., & Jiang, F. (2019). Image steganography using texture features and GANs. Proceedings of the International Joint Conference on Neural Networks, 2019-July(July), 1–8. https://doi.org/10.1109/IJCNN.2019.8852252

Luo, H., & Yu, F. (2008). Data Hiding in Image Size Invariant Visual Cryptography. 2008 3rd International Conference on Innovative Computing Information and Control, 25–25. https://doi.org/10.1109/ICICIC.2008.680

Mohammed, A. O., Hussein, H. I., Mstafa, R. J., & Abdulazeez, A. M. (2023). A blind and robust color image watermarking scheme based on DCT and DWT domains. Multimedia Tools and Applications. https://doi.org/10.1007/s11042-023-14797-0

Mstafa, R. J., & Elleithy, K. M. (2016). An adaptive Video Steganography Method Based on the Multiple Object Tracking and Hamming Codes. https://doi.org/10.13140/RG.2.2.14397.56803

Parberry, I. (1997). An efficient algorithm for the Knight's tour problem. Discrete Applied Mathematics, 73(3), 251–260. https://doi.org/10.1016/S0166-218X(96)00010-8

Setiadi, D. R. I. M. (2021). PSNR vs SSIM: imperceptibility quality assessment for image steganography. Multimedia Tools and Applications, 80(6), 8423–8444. https://doi.org/10.1007/s11042-020-10035-z

Shanthakumari, R., & Malliga, S. (2020). Dual layer security of data using LSB inversion image steganography with elliptic curve cryptography encryption algorithm. Multimedia Tools and Applications, 79(5–6), 3975–3991. https://doi.org/10.1007/s11042-019-7584-6

Sultana, H., Kamal, A. H. M., Hossain, G., & Kabir, M. A. (2023). A Novel Hybrid Edge Detection and LBP Code-Based Robust Image Steganography Method. Future Internet, 15(3), 108. https://doi.org/10.3390/fi15030108

Trajković, M., & Hedley, M. (1998). Fast corner detection. Image and Vision Computing, 16(2), 75–87. https://doi.org/10.1016/s0262-8856(97)00056-5

ur Rehman, A., Rahim, R., Nadeem, S., & ul Hussain, S. (2019). End-to-end trained CNN encoder-decoder networks for image steganography. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11132 LNCS, 723–729. https://doi.org/10.1007/978-3-030-11018-5_64

Younis, Y. M., Mstafa, R. J., & Atto, M. (2019). Video Information Hiding Based on Feature Points and Arnold Cat Algorithm. 2019 International Conference on Advanced Science and Engineering (ICOASE), 198–203. https://doi.org/10.1109/ICOASE.2019.8723729