Çukurova Üniversitesi
Mühendislik Fakültesi Dergisi
Çukurova University
Journal of the Faculty of Engineering
ISSN: 2757-9255 CİLT/VOLUME: 40 SAYI/ISSUE: 4 ARALIK/DECEMBER 2025

Secret Image Hiding Method Based on Multi-Cover Image with Stego Image Sequence Detection Using Perceptive Hash

Ertuğrul GÜL [1,a], **Saniye BİLGİÇ** [2,b]

[1]*Kayseri University, Software Engineering Department, Kayseri, Türkiye*
[2]*Kayseri University, Cyber Security Department, Kayseri, Türkiye*

[a]***ORCID**: 0000-0002-5591-3435; [b]**ORCID**: 0009-0002-0375-0440*

Article Info
Received : 09.07.2025
Accepted : 21.11.2025
DOI: 10.21605/cukurovaumfd.1736687

Corresponding Author
Ertuğrul GÜL
ertugrulgul@kayseri.edu.tr

Keywords
Image hiding
Multi-cover
Perceptual hash
Stego image order

***How to cite:** GÜL, E., BİLGİÇ, S., (2025). Secret Image Hiding Method Based on Multi-Cover Image with Stego Image Sequence Detection Using Perceptive Hash. Cukurova University, Journal of the Faculty of Engineering, 40(4), 927-936.*

ABSTRACT

As digital communication continues to expand, protecting the privacy and security of transmitted images has become increasingly crucial. Image hiding addresses this issue by concealing secret images within cover images. Most methods typically embed a secret image into a single cover image, which poses a risk if that cover image falls into the wrong hands. To address this issue, we propose a secret image hiding method that employs multiple cover images. The secret image is distributed across four cover images, while the fifth cover image is used to embed the perceptual hash values of the stego images in the proposed method. These hashes serve to accurately determine the correct sequence order of stego images during the extraction phase, especially in cases where file names or transmission order have been altered, ensuring proper extraction. Experimental results indicate that the proposed method achieves high visual quality of stego images, with an average PSNR of 44.9626 dB, and enables lossless recovery of the secret image.

Algısal Özet Kullanarak Taşıyıcı Görüntü Sırasını Tespit Eden Çoklu Kapak Görüntü Tabanlı Görüntü Gizleme Yöntemi

Makale Bilgileri
Geliş : 09.07.2025
Kabul : 21.11.2025
DOI: 10.21605/cukurovaumfd.1736687

Sorumlu Yazar
Ertuğrul GÜL
ertugrulgul@kayseri.edu.tr

Anahtar Kelimeler
Görüntü gizleme
Çoklu kapak
Algısal özet
Taşıyıcı görüntü sırası

***Atıf şekli:** GÜL, E., BİLGİÇ, S., (2025). Algısal Özet Kullanarak Taşıyıcı Görüntü Sırasını Tespit Eden Çoklu Kapak Görüntü Tabanlı Görüntü Gizleme Yöntemi. Çukurova Üniversitesi, Mühendislik Fakültesi Dergisi, 40(4), 927-936.*

ÖZ

Dijital iletişim genişlemeye devam ettikçe, iletilen görüntülerin gizliliğini ve güvenliğini korumak giderek daha önemli hale gelmektedir. Görüntü gizleme teknikleri, gizli görüntüleri kapak görüntüler içerisine gömerek bu problemi ele alır. Çoğu yöntem gizli görüntüyü genellikle tek bir kapak görüntüsüne gizlemekte ve bu kapak görüntüsünün yanlış ellere geçmesi durumunda güvenlik riski ortaya çıkmaktadır. Bu çalışmada, bu sorunu ele almak için, birden fazla kapak görüntüsü kullanan bir gizli görüntü gizleme yöntemi önerilmektedir. Önerilen yöntemde, gizli görüntü dört farklı kapak görüntüsüne dağıtılırken, beşinci kapak görüntü ise taşıyıcı görüntülerin algısal özet değerlerini saklamak için kullanılmaktadır. Bu özet değerleri, özellikle dosya adlarının veya iletim sırasının değiştirildiği durumlarda, gizli görüntü çıkarma sürecinde taşıyıcı görüntülerin sırasını doğru bir şekilde belirlemeye yarar ve gizli görüntünün doğru çıkartılmasını sağlar. Deneysel sonuçlar, önerilen yöntemin ortalama 44,96 dB PSNR değerinde yüksek görsel kaliteye sahip taşıyıcı görüntüler ürettiğini ve gizli görüntünün kayıpsız olarak kurtarılabildiğini göstermektedir.

1. INTRODUCTION

Images have become one of the most essential forms of information in modern digital systems, playing a crucial role in areas such as medical diagnosis, surveillance, and military intelligence [1,2]. With the rapid growth of digital communication, ensuring the privacy and security of transmitted images has become increasingly critical [3-5]. While encryption protects the content of images, it does not conceal their existence, often leaving communications vulnerable to interception and analysis [6]. To address this limitation, image hiding—a technique that hides secret image/images within another digital image/images—has gained significant attention [7]. Among various carrier formats, digital images are widely preferred for secret image hiding due to their ubiquitous presence on the Internet and their ability to mask data imperceptibly [8]. Image hiding not only safeguards the content of the secret image but also conceals the very fact that communication is taking place, offering an additional layer of security in an era where data privacy is paramount.

Most image-hiding methods embed a secret image into a single cover image [9-12]. However, this approach increases the risk of unauthorized parties accessing the hidden information. Distributing the secret image across several cover images significantly reduces the chances that an intruder can retrieve the entire hidden content without access to all or most of the carriers [13]. Additionally, sending these carrier images via different channels provides added security, ensuring the safe transmission of the concealed secret image [14]. Ker and Pevný [15] introduced several batch hiding methods, including max-greedy, max-random, linear, even, and sqroot, which specify different ways of spreading a message across multiple cover images. These strategies differ in how they allocate payloads based on image capacity, ranging from focusing data in a few high-capacity covers to distributing it evenly or proportionally among all available covers. Al-Bayati and Al-Jarrah [16] introduced a dual-cover framework for hiding secret multimedia files within two cover images, improving security through distributed payloads. Although the method focuses on general data rather than secret images, it effectively enhances security by spreading the hidden content across multiple covers. Zhao et al. [17] proposed a universal embedding strategy for batch adaptive steganography in both spatial and JPEG domains. The method allows for simultaneous data hiding into multiple cover images. Because the hiding process alters the pixel histogram, the hiding order cannot be directly recovered from the stego images. Therefore, the authors transmit the hiding order and message length of each image through an external channel.

However, the use of multiple cover images introduces the challenge of correctly determining their order. Extracting the secret image from cover images processed in the wrong sequence will result in the incorrect extraction of the secret image. Simply transmitting or securely storing the filenames of the carrier images does not solve this problem, as their names can be intentionally or unintentionally changed during communication. Therefore, methods employing multiple carriers must address and overcome this issue to ensure accurate recovery of the hidden image.

In light of the previously mentioned challenges, we propose a new image hiding method that utilizes multiple cover images. This approach determines the sequence of cover images using the perceptual hash function. Experimental results indicate that our method can effectively extract the secret image, regardless of the cover image filenames or the order of the cover images. The key contributions of our method are summarized below:

- We propose an image hiding method that utilizes multiple cover images.
- We use a hash-based approach to determine the order of the cover images in the hiding process.
- Experimental results show that even when the order of the cover images is altered, the proposed method can extract the secret image losslessly.

The remainder of this paper is organized as follows: The next section describes the two main processes of the proposed method, namely the hiding algorithm with multi-cover image and the secret image extraction using hash-based cover image order detection. Section 3 presents experimental results to evaluate the performance of the proposed approach. Finally, the last section concludes the study and discusses potential directions for future work.

2. PROPOSED IMAGE HIDING METHOD

The proposed method consists of two primary processes, and this section outlines these two main processes: the hiding algorithm with multi-cover image and the secret image extraction using hash-based cover image order detection.

2.1. Hiding Algorithm with Multi-Cover

This method hides the secret image in multiple cover images and hides the hash information of the stego images in one cover image for determining the stego image order in the secret image extraction process. The block diagram of the hiding algorithm with multi-cover image is given in Figure 1. To start, the secret image is shuffled using a key. Next, this shuffled image is hidden within the first four cover images. During the hiding phase, each 8-bit pixel from the shuffled secret image is concealed by distributing its bits into the two least significant bits (LSBs) of the pixels at the same spatial coordinates (x, y) in each corresponding cover image. The use of two LSBs was selected to achieve a balance between hiding capacity and imperceptibility. Using only 1 LSB would not provide sufficient capacity to hide the complete secret image, while using more than 2 LSBs (e.g., 3 or 4) could lead to noticeable degradation in the visual quality of the cover images. Furthermore, embedding the secret image evenly across multiple cover images ensures that no single cover image suffers from significant visual distortion, maintaining an acceptable level of quality for all covers. Following, a 64-bit hash value is created for each stego image that contains the secret image data using the Perceptive Hash (P-Hash) [18]. The four 64-bit hash values are then merged to form a 256-bit hash sequence, which is hidden into the least significant bits (LSBs) of the first 256 pixels of the fifth cover image, producing a hash-embedded stego image. Finally, a 64-bit hash of this hash-embedded stego image is generated and stored in the database alongside the key, allowing for determination during the secret image extraction process. The pseudocode of the hiding algorithm with multi-cover image process is given in Algorithm 1.

Algorithm 1. Image hiding using multiple cover images

Input: Secret image SIm, Cover images $CIm_1, CIm_2, CIm_3, CIm_4, CIm_5$, Key k
Output: Stego images $StIm_1, StIm_2, StIm_3, StIm_4, StIm_5$, Encrypted key/hash pair and session key

Step 1: Shuffle Secret Image
Shuffle the secret image SIm using key k to obtain shuffled secret image $SSIm$
Step 2: Hide Shuffled Secret Image into Cover Images
foreach pixel p *located at position (x, y)* in $SSIm$ **do**
  Split p into four 2-bit segments: p_1, p_2, p_3, p_4
  Hide p_1 into 2 LSBs of the pixel at position (x, y) in CIm_1
  Hide p_2 into 2 LSBs of the pixel at position (x, y) in CIm_2
  Hide p_3 into 2 LSBs of the pixel at position (x, y) in CIm_3
  Hide p_4 into 2 LSBs of the pixel at position (x, y) in CIm_4
Obtain stego images $StIm_1, StIm_2, StIm_3, StIm_4$
Step 3: Generate Hashes of Stego Images
Generate hashes: $H_1 = P\text{-}Hash(StIm_1)$, ..., $H_4 = P\text{-}Hash(StIm_4)$
Merge the hashes: $H_m = H_1|H_2|H_3|H_4$ (256 bits)
Step 4: Embed Merged Hash into Fifth Cover Image
Embed H_m into the 1 LSBs of the first 256 pixels of CIm_5
Obtain hash-embedded stego image $StIm_5$
Step 5: Final Hash and Storage
Generate $H_S = P\text{-}Hash(StIm_5)$
Encrypt (k, H_S) using symmetric encryption algorithm using session key
Encrypt the symmetric session key with the recipient's public key
Store the encrypted key/hash pair and session key

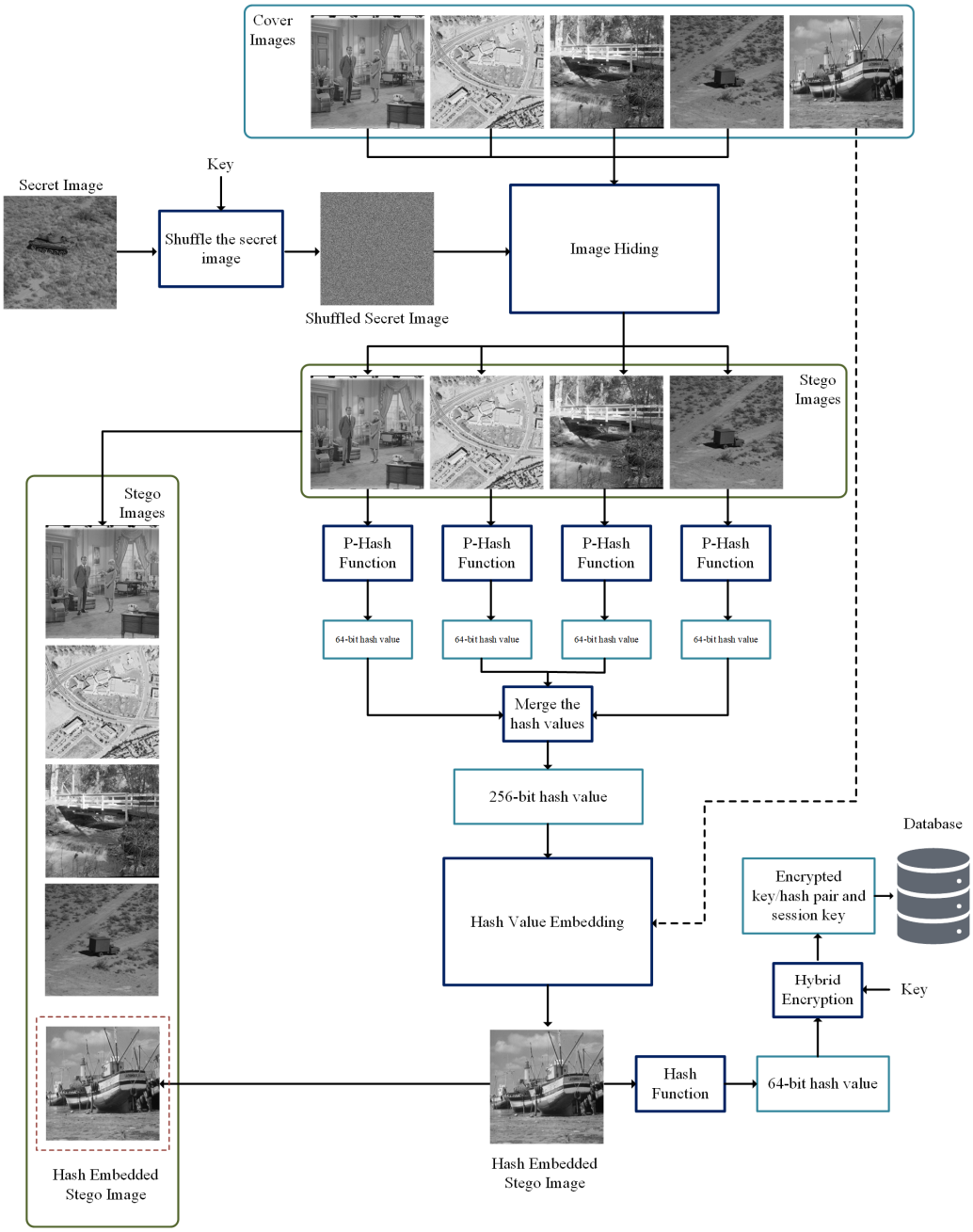

Figure 1. Diagram of the hiding algorithm with multi-cover image process

The P-Hash algorithm [18] is used to ensure the correct order of stego images, enabling accurate reconstruction of the secret image. P-Hash creates hashes that represent the visual content of each image, remaining robust against minor variations like noise or compression while still generating unique values for different images. In contrast, traditional cryptographic hash functions such as MD5 or SHA are highly sensitive to small changes, with even slight modifications completely changing the hash value, which makes them unsuitable for ordering based on visual similarity. Using P-Hash, only a part of the hash changes when small distortions occur, allowing the correct sequence of stego images to be reliably identified by comparing hash similarity.

To protect the shuffling key k and the 64-bit P-Hash value H_S, we adopt a hybrid encryption scheme. The pair (k, H_S) is first encrypted using a symmetric encryption algorithm (e.g., AES) with a randomly generated session key. This session key is then encrypted using an asymmetric encryption algorithm (e.g., RSA) with the recipient's public key. The encrypted key/hash pair and the encrypted session key are

securely stored in the database. During extraction, the recipient retrieves the encrypted components from the database, uses their private key to recover the session key, and decrypts (k, H_S) using the symmetric algorithm to proceed with secret image extraction.

2.2. Secret Image Extraction Using P-Hash-based Stego Image Order Detection

This technique determines the sequence order of stego images and extracts the hidden secret image from them. The block diagram of secret image extraction using hash based stego image order detection is given in Figure 2. Initially, the hash-embedded stego image is determined among all stego images by comparing the hash value obtained from the database using the Hamming distance. Next, the first 256 pixels of this image are utilized to extract a 256-bit hash value from their least significant bits (LSBs). This hash value is then segmented into four 64-bit hash values. These are compared with the newly generated P-Hash values of the remaining stego images using Hamming distance to determine the correct order of the four stego images containing the secret image. Subsequently, the two LSBs of the respective pixels in the ordered stego images are employed to reconstruct the 8-bit value of each pixel in the shuffled secret image. Finally, the shuffled secret image is reshuffled using the key to recover the hidden secret image. The pseudocode of the secret image extraction using the hash-based stego image order detection process is given in Algorithm 2.

Algorithm 2: Hash-based Stego Image Order Detection and Secret Image Extraction

Input: Stego images $StIm_1, StIm_2, StIm_3, StIm_4, StIm_5$, Encrypted key/hash pair and session key
Output: Extracted secret image S_e

Step 1: Retrieve and Decrypt Key and Stored Hash
Retrieve the encrypted key/hash pair and session key from the database
Decrypt the session key with the recipient's private key
Decrypt the key/hash (k, H_S) pair with the session key
Step 2: Determine Hash-Embedded Stego Image
Initialize $minDist \leftarrow \infty$
foreach stego image S_i in $\{StIm_1, StIm_2, StIm_3, StIm_4, StIm_5\}$ **do**
  Compute hash H_i =*P-Hash*(S_i)
  Compute Hamming distance $d = Hamming(H_i, H_S)$
  If $d < minDist$ **then**
    $minDist \leftarrow d$
    $S_H \leftarrow S_i$ // Closest hash match found
Step 3: Extract Hash Sequence from S_H
Extract 1 LSBs from the first 256 pixels of S_H to obtain 256-bit hash sequence H_m
Split H_m into four 64-bit hash values: H'_1, H'_2, H'_3, H'_4
Step 4: Determine Correct Stego Image Order
Compute hashes of the remaining four stego images
Compare each computed P-Hash of the stego images with the H'_i using Hamming distance.
Match each H'_i with the corresponding recomputed hash based on the closest P-Hash match to determine the correct image order $[StIm'_1, StIm'_2, StIm'_3, StIm'_4]$
Step 5: Reconstruct Shuffled Secret Image
foreach pixel index i **do**
  Extract 2 LSBs from pixel i of $StIm'_1 StIm'_2 StIm'_3 StIm$
  Merge the four parts to form an 8-bit pixel p_i of the extracted shuffled secret image $SSIm_e$
Step 6: Recover Secret Image
Reshuffle $SSIm_e$ using key k to obtain extracted secret image S_e

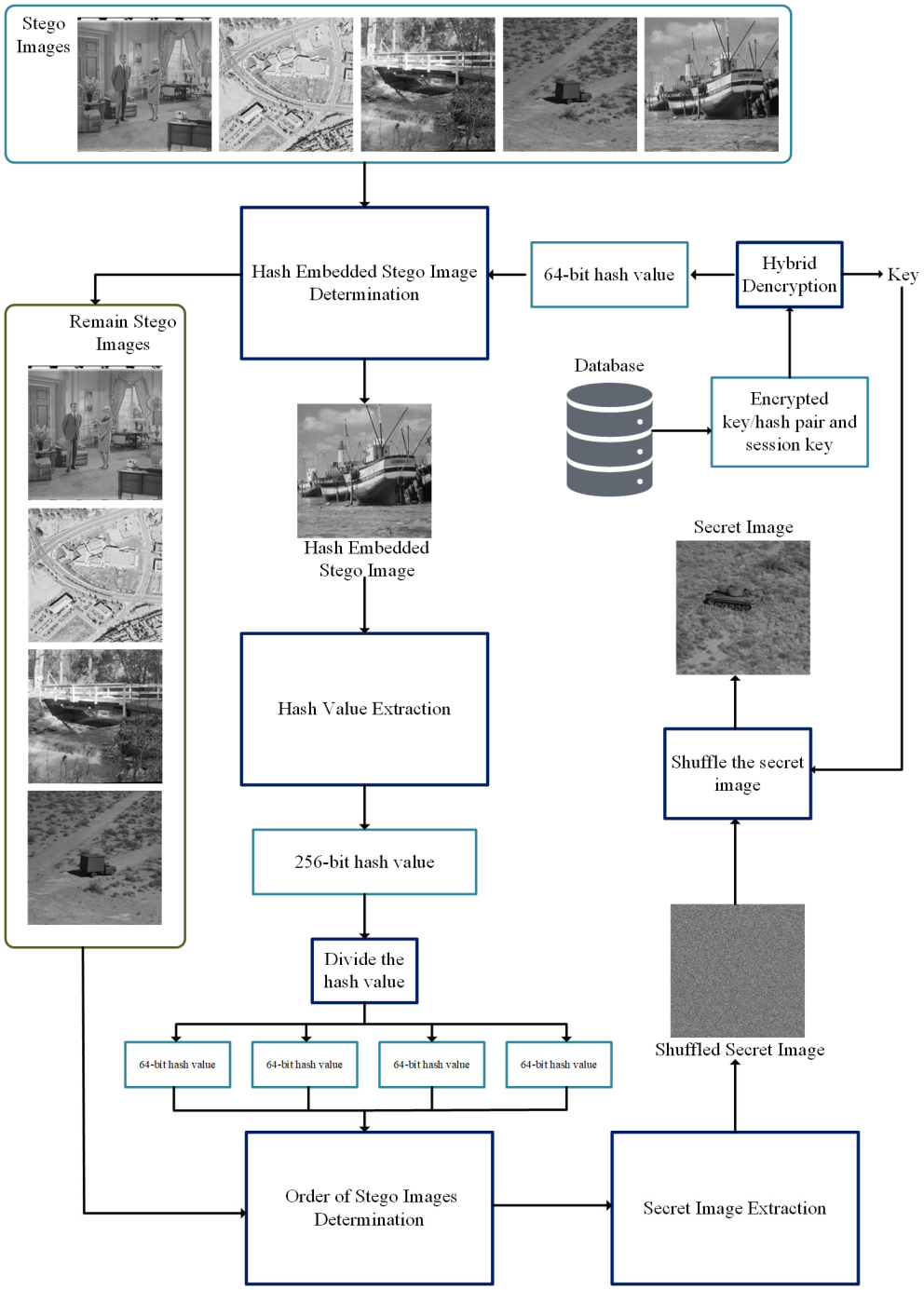

Figure 2. Diagram of the secret image extraction using the hash-based stego image order detection process

3. EXPERIMENTAL RESULTS

Experiments were conducted using MATLAB R2019a, employing images of Tank, Couple, Aerial, Stream and bridge, Truck, and Fishing Boat obtained from the USC-SIPI Image Database [19]. The original images used in the experiments are shown in Figure 3. Specifically, Figures 3 (b)-(f) depict the cover images, while Figure 3 (a) illustrates the secret image.

The secret image shown in Figure 3 (a) was hidden using the proposed method with five cover images. Figure 4 (a)–5(d) display the four images containing the hidden secret image, while Figure 4(e) shows the hash-embedded stego image. Upon examining Figures 3 and 4, it is evident that there is no perceptible

difference between the cover and stego images. Figure 4 (See (f)) also illustrates the secret image extracted from the stego images using the proposed method. The extracted secret image shows no noticeable difference compared to the original secret image.

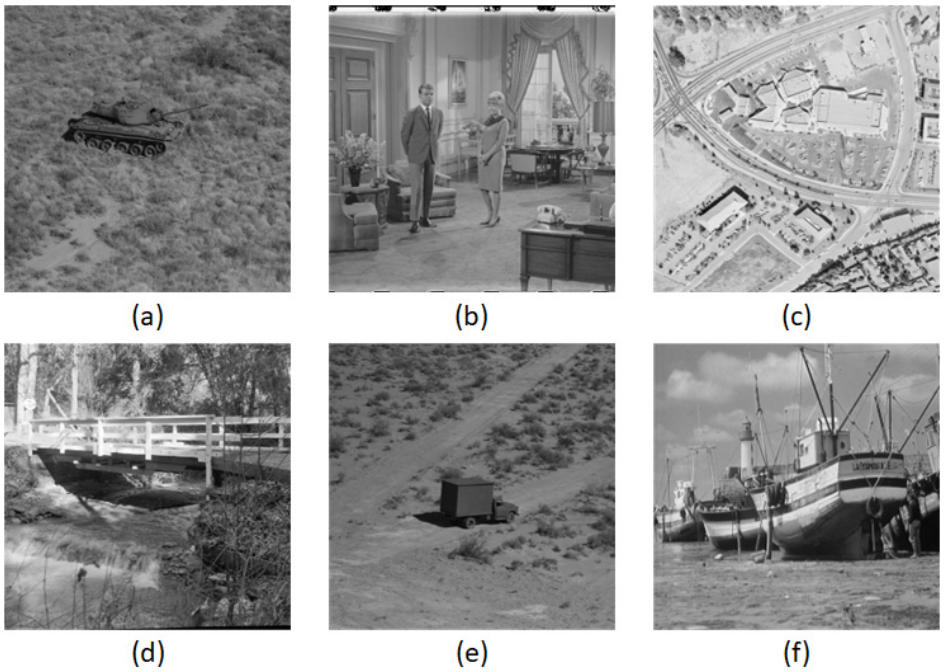

Figure 3. Secret image and cover images

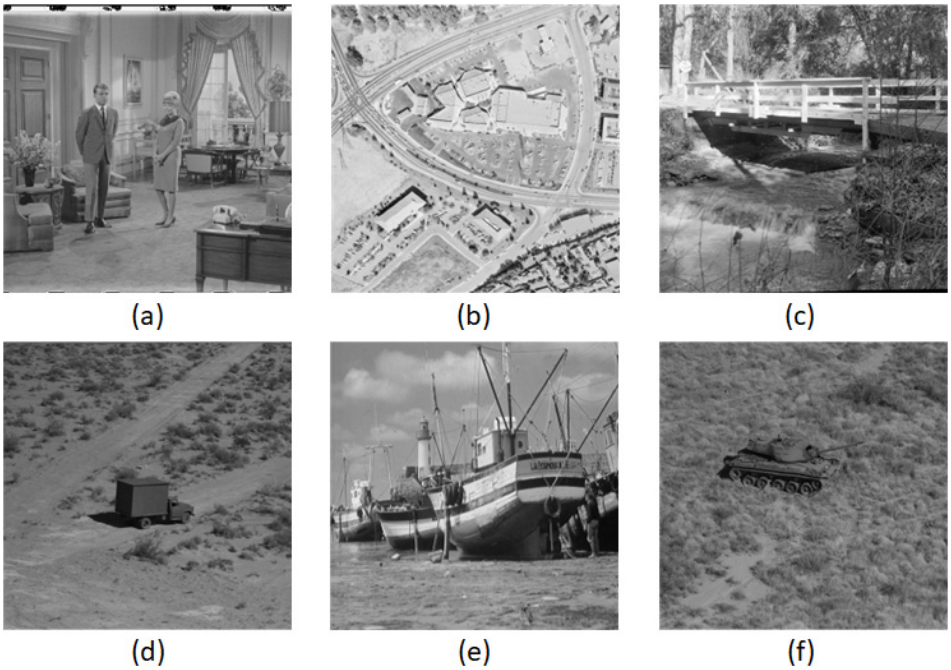

Figure 4. Stego images and extracted secret image

Table 1 presents the Peak signal-to-noise ratio (PSNR) values of both the stego images and the extracted secret image. The PSNR metric measures the similarity between the original and processed images, where a higher PSNR value indicates better image quality and effectiveness of the method. Among the stego images, the highest PSNR value was obtained from the Fishing Boat image, while the lowest was obtained from the Truck image. The reason for the highest PSNR value in the Fishing Boat image is that only the

hash values of the other stego images were embedded into it. Additionally, since only the least significant bit (LSB) of 256 pixels was used to embed the hash value, the PSNR value remained high.

Table 1. PSNR result of stego images and extracted secret image

	PSNR
Stego 1	46.1411
Stego 2	44.1045
Stego 3	45.5604
Stego 4	44.0444
Stego 5	81.0128
Average	52.1727
Secret image	Inf

The average PSNR value of the stego images was 52.1727 dB, boosted by the hash-embedded stego image. The average PSNR value of the stego images containing the hidden secret image was 44.9626 dB. Moreover, the PSNR value of the extracted secret image is effectively infinite, indicating that there is no difference between the original secret image and the extracted image, and that it was recovered without any loss.

In another experiment, the order of the stego images was rearranged to evaluate the performance of the proposed method. Table 2 presents the sequence order of the stego images along with the PSNR values of the extracted secret image. As observed from the table; despite changing the order of the stego images, the proposed method successfully recovered the secret image without any loss. This demonstrates the effectiveness of the proposed hash-based stego image ordering approach.

Table 2. Evaluation of the proposed method against stego images reordering

Stego image order					PSNR
1	2	3	4	5	Inf
4	2	5	3	1	Inf
2	5	1	4	3	Inf
3	4	5	2	1	Inf
3	1	4	5	2	Inf
5	3	2	1	4	Inf

Although classical steganalysis methods [20-22] may be able to detect LSB modifications in individual stego images, the overall security of the proposed approach is enhanced by using multiple independent covers, key-based shuffling of the secret image, and P-hash-based ordering of stego images. As a result, even if some stego images are detected and intercepted by unauthorized parties, it remains highly difficult to recover the entire secret image without access to all embedded components and the key.

To evaluate the computational efficiency of the proposed method, the execution times of both the embedding and extraction processes were measured on an Intel Core i7-9700 CPU @ 3.00 Ghz. The results are presented in Table 3. Table indicates that both hiding and extraction processes are completed within fractions of a second, demonstrating the computational efficiency of the proposed method.

Table 3. Computational cost of the hiding and extraction processes of the proposed method

Process	Execution Time (s)
Hiding	0.2154
Extraction	0.1645

4. CONCLUSIONS

This paper presents a secret image hiding method using multiple cover images based on stego image sequence order detection with hash values. In the proposed approach, the secret image is hidden by utilizing the 2 LSBs of four different cover images. Additionally, the hash values of the stego images containing the secret image are embedded in a fifth stego image. These hash values are later used to ensure the correct processing order of the stego images during the secret image extraction phase.

The experimental results demonstrate the effectiveness of the proposed method, achieving an average PSNR of 44.9626 dB for the stego images, indicating high image quality. Moreover, the secret image was successfully extracted without any loss. In further experiments, where the order of stego images was intentionally altered, the proposed method still managed to correctly recover the secret image.

As future work, we plan to develop a robust image hiding approach to resistance against common attacks. Moreover, a CNN-based feature extraction framework will be explored to construct a more reliable and adaptive stego image ordering mechanism.

5. REFERENCES

1. Uzdur, B., Tekeli, E., İbrikçi, T., Rashid, H.U. & Ramachandran, G. (2025). Diagnosis of hepatocellular carcinoma-HCC liver cancer using federated learning on MR images. *Çukurova Üniversitesi Mühendislik Fakültesi Dergisi*, *40*(3), 531-544.

2. Kartal, S. (2022). Otokodlayıcılar kullanarak uzaktan algılama görüntülerindeki eksik verilerin yeniden yapılandırılması. *Çukurova Üniversitesi Mühendislik Fakültesi Dergisi*, *37*(4), 853-862.

3. Elmaci, M., Toprak, A.N. & Aslantas, V. (2024). Detection of background forgery using a two-stream convolutional neural network architecture. *Multimedia Tools and Applications*, *83*(12), 36739-36766.

4. Melman, A. & Evsutin, O. (2023). Image data hiding schemes based on metaheuristic optimization: a review. *Artificial Intelligence Review*, *56*(12), 15375-15447.

5. Gul, E. (2022). A blind robust color image watermarking method based on discrete wavelet transform and discrete cosine transform using grayscale watermark image. *Concurrency and Computation: Practice and Experience*, *34*(22), e6884.

6. Gutub, A. & Al-Shaarani, F. (2020). Efficient implementation of multi-image secret hiding based on LSB and DWT steganography comparisons. *Arabian Journal for Science and Engineering*, *45*(4), 2631-2644.

7. Ye, G. & Chen, Z. (2025). Authenticated reversible image hiding algorithm based on blockchain technology. *Cluster Computing*, *28*(1), 10.

8. Song, B., Wei, P., Wu, S., Lin, Y. & Zhou, W. (2024). A survey on deep-learning-based image steganography. *Expert Systems with Applications*, 124390.

9. Shang, J. & Xu, X. (2024). Research on a double image security transmission algorithm of image encryption and hiding. *Measurement: Sensors*, *31*, 100942.

10. Kaur, S., Bansal, S. & Bansal, R.K. (2021). Image steganography for securing secret data using hybrid hiding model. *Multimedia Tools and Applications*, *80*, 7749-7769.

11. Arora, H., Soni, G.K., Kushwaha, R.K. & Prasoon, P. (2021). Digital image security based on the hybrid model of image hiding and encryption. In *2021 6th International conference on communication and electronics systems (ICCES)*, 1153-1157, IEEE.

12. Heednacram, A. & Keaomanee, Y. (2024). Four enhanced algorithms for full size image hiding in chest x-ray images. *Multimedia Tools and Applications*, *83*(30), 74855-74881.

13. Al-Bayati, M.T. & Al-Jarrah, M.M. (2016). DuoHide: a secure system for hiding multimedia files in dual cover images. In *2016 9th International Conference on Developments in eSystems Engineering (DeSE)*, 138-142, IEEE.

14. Gul, E. & Ozturk, S. (2023). A secret image sharing method based on block-wise cheating detection and recovery on shadow images. *Computers and Electrical Engineering*, *110*, 108882.

15. Ker, A.D. & Pevny, T. (2012). Batch steganography in the real world. In *Proceedings of the on Multimedia and security*, 1-10.

16. Al-Bayati, M.T. & Al-Jarrah, M.M. (2016). DuoHide: a secure system for hiding multimedia files in dual cover images. In *2016 9th International Conference on Developments in eSystems Engineering (DeSE),* 138-142, IEEE.

17. Zhao, Z., Guan, Q., Zhao, X., Yu, H. & Liu, C. (2018). Universal embedding strategy for batch adaptive steganography in both spatial and JPEG domain. *Multimedia Tools and Applications*, *77*(11), 14093-14113.

18. Aissi, M.B., El Hazzat, S. & Yakhlef, M.B. (2024). Evaluation of perceptual hashing algorithms for digital image authenticity. In *2024 3rd International Conference on Embedded Systems and Artificial Intelligence (ESAI)*, 1-7, IEEE.

19. University of Southern California, Signal and Image Processing Institute (USC-SIPI), USC-SIPI Image Database, http://sipi.usc.edu/database/. Erişim tarihi: 11.06. 2025.

20. Khalind, O. & Aziz, B. (2014). Detecting 2LSB steganography using extended pairs of values analysis. In *Mobile Multimedia/Image Processing, Security, and Applications 2014*, 9120, 7-18. SPIE.
21. Fridrich, J., Goljan, M. & Du, R. (2001). Detecting LSB steganography in color, and gray-scale images. *IEEE Multimedia*, *8*(4), 22-28.
22. Westfeld, A. & Pfitzmann, A. (1999). Attacks on steganographic systems: Breaking the steganographic utilities EzStego, Jsteg, Steganos, and S-Tools-and some lessons learned. In *International workshop on information hiding*, 61-76. Berlin, Heidelberg: Springer Berlin Heidelberg.