Effects of Discrete Cosine Transform Arrangement Patterns on Full Size Image Steganography

Yossawee Keaomanee[1] and Apichat Heednacram[1*]

[1]College of Computing, Prince of Songkla University, Phuket, 83120, Thailand

*** Corresponding author, Email address: apichat.h@phuket.psu.ac.th**

Abstract

Steganography is an invisible image-hiding technique that can be used to hide proprietary information within a cover image. Protecting this information ensures that product owners can continue to innovate without the fear of their ideas being stolen or replicated by others. The study aims to minimize inaccuracies of the earlier research that used discrete cosine transform (DCT) coefficients to hide images in the frequency domain. We modified the dominant DCT coefficients arrangement pattern by incorporating the three new patterns and compared them with the two conventional patterns. The chosen coefficients of the hidden image are then concealed in the DCT blocks of 8 by 8 pixels in the cover image. The results showed a significant improvement over the earlier work. The best outcome of the five studied patterns is produced by the triangle arrangement. The newly created triangle pattern can cut the error in image reconstruction by 26.08%.

Keywords: image steganography, DCT, arrangement pattern, proprietary information

1. Introduction

Digital information is being created and disseminated at a rapid rate, and it is being used in a variety of industries, including telemedicine, e-government, and online commerce. Unfortunately, there has been a notable increase in the unauthorized manipulation of digital multimedia content. As a result, there is an immediate demand for techniques to protect digital multimedia content from unauthorized use (Saini & Kumar, 2023).

The two most frequently utilized techniques for embedding or concealing private information in cover images are steganography and watermarking. With watermarking, words or symbols that are added to or embedded in an image may be seen, but with steganography, the hidden information cannot be seen with the naked eye. As a result, the cover image will not lose quite as much of its quality or details as a result of the embedded process.

To secretly conceal the hidden message, numerous steganography techniques were developed (Kadhim, Premaratne, Vial, & Halloran, 2019; Wang, Cheng, Wu, & Chen, 2019; Subramanian, Elharrouss, Al-Maadeed, & Bouridane, 2021). Steganography has various applications, including secure data storage, confidential communication, and protecting against identity theft in e-commerce (Thangadurai & Devi, 2014). Additionally, it frequently uses optimal strategies to boost efficiency by lowering both time and space complexity (Prabu & Latha, 2020). In contrast to text steganography, image steganography involves hiding an entire image inside of another image, making it more difficult. The main objectives of effective image steganography are minimal errors of reconstructed hidden messages, little visual change in the cover image, high invisibility, and high payload capacity. Payload capacity is the ratio of the number of secrete bits embedded to the total pixels in the cover image. Consequently, the more secret bits that are concealed, the higher the payload capacity.

The two domains in which steganography algorithms are typically used to conceal information in cover images are the spatial (Siddiqui et al., 2020; Karawia, 2021; Fateh, Rezvani, & Irani,2021) and frequency (Tsai & Yang, 2017; Vyas & Dudul, 2019; Emmanuel, Hungil, Maiga, & Santoso,2021) domains. The state-of-the-art of steganography is heavily centred on the frequency domain, where the hidden message is concealed in the discrete cosine transform (DCT) domain of the cover image (Baziyad, Baziyad, & Kamel, 2018; Khan et al., 2019; Rabie, Baziyad & Kamel, 2019).

More recently, some researchers suggested algorithms to hide sensitive data by using discrete cosine transform (DCT) for both the cover and the hidden images. Then, the secret DCT information is concealed in high-frequency region of the DCT block of the cover image. These work (Vakani, Kamel, Rabie, & Baziyad, 2020; Vakani, Abdallah, Kamel, Rabie, & Baziyad, 2021; Heednacram & Keaomanee, 2023) utilized a conventional rectangular pattern for selecting DCT coefficients. However, the impacts of DCT arrangement patterns have not yet been the subject of any investigations. This challenge motivates us to conduct research because new patterns have the potential to improve image recovery quality in steganography. Although Vakani et al. (2021) achieved an improvement of up to 20.25 dB in the extracted secret image quality, the secret image size is only a quarter of a cover image. Heednacram and Keaomanee (2023) managed to make the secret image the same size as the cover image, yet obtaining the enhanced secret image quality of slightly over 30 dB. We believe that our suggested patterns hold the key to enhancing the quality of the secret image that is extracted. Our main contributions in this paper are as follows:

- We created a new triangular DCT arrangement pattern by improving the traditional rectangular layout design of DCT coefficients.
- Using a full-size hidden image, which can be as big as the cover image, allows for the achievement of an extremely high payload capacity.
- Although in this paper the farm profile of the agricultural products is concealed for online marketing activities, the basic concept of shielding proprietary information can be easily applied to other types of applications.
- The proposed algorithm reduced errors by 25.91%, enhancing the quality of the hidden image being reconstructed.
- The proposed algorithms are implemented in practice as a web application that is freely accessible online.

The paper is structured as follows: The introduction appears in Section 1. In Section 2, the DCT technique is introduced, and a proposed algorithm with various DCT coefficient arrangement patterns is discussed. The discussion and results of the experiments are described in Section 3. The proposed method's conclusion is provided in Section 4.

2. Materials and Methods

2.1 Input images

Our experiments will put our proposed algorithm, which has three different variants, into use and compare it with the two already-existing algorithms (Vakani et al., 2021; Heednacram & Keaomanee, 2023). The cover images used in the experiments are of fresh fruit. The secret images contain information about the farm that owns the fruit image. For mockup purposes, the farm's name, logo, address, and other proprietary information were made up. Figure 1 shows the cover image of three samples and the hidden image of five samples. The cover and hidden images are of the same size, 800×800 pixels (all RGB-color)[1]. Note that the Joint ISO committee has adopted DCT to the Joint Photographic Experts Group international standard of 8×8 block size (Tsai & Yang, 2017). This serves to minimize the blocking effect that occurs during image compression and steganography. Given that 800×800 is divisible by 8, we selected this resolution as it allows us to visually inspect the intricacies in the resulting images. However, as long as the input images are divisible by 8, our algorithms can be applied to any size image. The performance of all five approaches will be evaluated using the identical computer's Intel Core i5 2.4 GHz processor and 16 GB of RAM.

[Figure 1. Cover (top row) and hidden (last 2 rows) images.]

2.2 Discrete cosine transform (DCT)

The DCT coefficients are commonly used in watermarking and image steganography to conceal secret messages. The first step in this procedure is to split the image's pixels into 8×8-pixels blocks. These blocks are then subjected to a transformation, producing a set of 64 DCT coefficients calculated by equations (1) and (2) (Emmanuel et al., 2021).

$$DCT(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} P(x,y) \cos\left(\frac{(2x+1)i\pi}{2N}\right)\left(\frac{(2y+1)j\pi}{2N}\right) \quad (1)$$

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & if\ u = 0 \\ 1 & if\ u > 0 \end{cases} \quad (2)$$

The notation $P(x,y)$ is the x,y^{th} pixel of the image represented by matrix P, and N is the size of the block (in general, $N = 8$). Equation (1) calculates DCT element i,j^{th} of the transformed image from the pixel value of the input image.

[1] Datasets are available at https://github.com/yossy343/Dataset_SJST.

[Figure 2. Three primary frequency components.]

The DCT separates the image into three primary frequency components: high, middle, and low frequencies (Tsai & Yang, 2017). According to their frequency characteristics, these components are divided into three categories, as illustrated in Figure 2, with low-frequency components being represented by white, middle-frequency components by blue, and high-frequency components by grey.

The process of image reconstruction from its coefficients can be done by computing the Inverse Discrete Cosine Transform (IDCT), as described in Equation (3). The IDCT is employed to convert the DCT coefficients back into their respective colour values.

$$P(x,y) = \frac{1}{\sqrt{2N}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C(i)C(j)DCT(i,j)\, cos\left(\frac{(2x+1)i\pi}{2N}\right)\left(\frac{(2y+1)j\pi}{2N}\right) \quad (3)$$

2.3 Proposed algorithms

Before introducing our algorithms, we will discuss the drawbacks of previous methods (Vakani et al., 2021; Heednacram & Keaomanee, 2023). In this earlier research, the dominating DCT coefficients of the hidden image are selected in a matrix form of a traditional rectangular arrangement layout from the low-frequency region (a highly sensitive area). The secret coefficients are then concealed in the high-frequency region of the cover image (which is of little importance). Although all authors used a rectangular dominating DCT layout, no studies have yet been conducted to determine the impact of using other DCT arrangement layouts. Figure 3 shows the DCT arrangement layout used in (Heednacram & Keaomanee, 2023). In (Vakani et al., 2021), a similar strategy was applied, but the size of n in a rectangular layout $n \times n$ was varied in accordance with the quantity of non-significant DCT coefficients in the cover image.

[Figure 3. Selection area of cover and hidden DCT coefficients in embedded process.]

Since the Human Visual System (HVS) is less sensitive to high-frequency components of the DCT, low and middle frequency components are more important than high frequency components (Rabie et al., 2019). The drawback of the existing rectangular pattern of the hidden DCT in Figure 3 is that it still has a substantial number of coefficients in the high-frequency region. Therefore, our idea is to cover more of the area of low and middle-frequency components, which are more crucial for reconstructing high-quality images. Consequently, we will base our design on the new patterns that trade some high-frequency coefficients in the lower diagonal area for more low and middle-frequency coefficients in the upper diagonal area. This study proposes three distinct variants. Figure 4 shows the three novel forms ($\boldsymbol{p_1} - \boldsymbol{p_3}$) of the stated design patterns for dominant DCT coefficients.

[Figure 4. DCT arrangement patterns for hidden (a – d) and cover (e) images.]

The quality of the image's perceptual representation is not greatly altered when the less crucial high-frequency coefficients in the cover mask are swapped out for rescaled secret data. This concealing technique enables important information to be concealed inside the high-frequency DCT coefficients while preserving an acceptable quality of the stego image (the result of the embedding procedure in Figure 5).

[Figure 5. Diagram for encoding process.]

Encoding Algorithm:

Input: I_c and I_h /* cover image and hidden image */
Output: I_s /* stego image */

Step 1: Load I_c and I_h as floating RGB
Step 2: Convert I_c and I_h to DCT coefficient matrices
called them: Cover[DCT] and Hidden[DCT]
Step 3: Scale down Hidden[DCT] by a factor of constant
Step 4: Choose DCT arrangement pattern p_k where $k = 1, 2, 3$

Step 5: Build Stego[DCT] by embedding blocks of size $m \times n$
of Hidden[DCT] into Cover[DCT]
if p_0: set $DCT[i]$ locations according to Conventional pattern
if p_1: set $DCT[i]$ locations according to Lego pattern
if p_2: set $DCT[i]$ locations according to Stealth pattern
if p_3: set $DCT[i]$ locations according to Triangular pattern
for $i = 0$; $i < (m\text{-}2)(n\text{-}2)$: i++
Stego[$DCT(2 + i\%6, 2 + i/6)$] = Hidden[$DCT[i]$]
Step 5: Convert Stego[DCT] to I_s as floating RGB
Step 6**:** Save $\boldsymbol{I_s}$ to disk

Utilizing the decoding algorithm detailed below, the image concealed in the stego can be recovered. The buried image data is then restored to its original colour. Figure 6 displays the decoding procedure.

[Figure 6. Diagram for decoding process.]

Decoding Algorithm:

Input: I_s /* stego image */
Output: I_r /* reconstructed hidden image */

Step 1: Read I_s from disk
Step 2: Convert I_s to Stego[DCT], a DCT coefficient matrix
Step 3: Initialize blocks of size $m \times n$ of Hidden[DCT] with zeroes
Step 4: Choose relevant DCT[i] arrangement pattern p_k as in encoding process
Step 5: Duplicate DCT coefficients from the bottom right corner of I_s
for $i = 0$; $i < (m\text{-}2)(n\text{-}2)$: i++
Hidden[$DCT[i]$] = Stego[$DCT(2 + i\%6, 2 + i/6)$]
Step 5: Convert Hidden[DCT] to I_r in RGB domain
Step 6: Save $\boldsymbol{I_r}$ to disk

2.4 Method validation

The quality validation (Hussain, Abdul-Wahab, Bin-Idris, Ho, & Jung, 2018; Hashim, Rahim, Johi, Taha, & Hamad, 2018) between any two given images P_{ij} and Q_{ij} with M × N image size are listed in Eq. (4) to Eq. (9).

Root Mean Square Error (RMSE):

RMSE measures the image reconstruction loss. Low RMSE indicates that relatively little was altered from the original image $\left(P_{ij}\right)$ throughout the building process, leading to low error and high quality of the reconstructed image $\left(Q_{ij}\right)$.

$$RMSE = \sqrt{\frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (P_{ij} - Q_{ij})^2} \tag{4}$$

Peak Signal-to-Noise Ratio (PSNR):

PSNR quantifies the distortion when a reconstructed image is compared to the original. In image processing, 30 dB or greater is commonly considered to be an acceptable value. Higher PSNR values suggest better quality in compressed or reconstructed images.

$$PSNR = 10log\left(\frac{255}{RMSE}\right)^2 \tag{5}$$

Structure Similarity Index Matrix (SSIM):

SSIM measures the likeness between two images by assessing their structural aspects, including luminance, contrast, and structure. It assigns a score between 0 and 1, with 1 indicating complete image identity, and it can be calculated using equations (6) to (9)

$$SSIM = \frac{(2\mu_P\mu_Q + c_1)(2\sigma_{PQ} + c_2)}{({\mu_P}^2 + {\mu_Q}^2 + c_1)({\sigma_P}^2 + {\sigma_Q}^2 + c_2)} \tag{6}$$

$$\mu_P = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} P_{ij} \tag{7}$$

$${\sigma_P}^2 = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (P_{ij} - \mu_P)^2 \tag{8}$$

$${\sigma_{PQ}}^2 = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (P_{ij} - \mu_P)\,(Q_{ij} - \mu_Q) \tag{9}$$

3. Results and Discussion

Our experiments tested the proposed algorithm by varying the DCT arrangement pattern using three new enhanced patterns (Lego, Stealth, and Triangular). The result will compare the two existing methods, DAS (DCT Adaptive-Scaling) (Vakani et al., 2021) and LDCT (Heednacram & Keaomanee, 2023), whose main ideas are like those of our method. When comparing the results of each approach, the RMSE, PSNR, and SSIM values are considered to determine how effective each method is at hiding and recovering data.

3.1 Visual quality of reconstructed images

Figures 7-9 display the cover image, the hidden image, the stego image (cover with hidden data), and the reconstructed images that were produced using five different methods. The three proposed patterns and the LDCT reconstructed images are noticeably superior to DAS and nearly visually identical to the original hidden image. Additionally, the stego image shows no obvious irregularities. It is demonstrated that the proposed patterns are both particularly invisible and have a high payload capacity (the concealed image can be the full size of the cover image). This high capacity to protect private data is beneficial for online and e-commerce activity. The detailed statistical analysis of each method, however, will be covered in greater depth in the next section.

[Figure 7. Reconstructed images with Mangosteen as a cover image.]

[Figure 8. Reconstructed images with Durian as a cover image.]

[Figure 9. Reconstructed images with Pomelo as a cover image.]

3.2 RMSE

To examine the effect of the arrangement pattern, we consider the results from DAS (a scaled rectangular pattern) and LDCT (a fixed rectangular pattern), with the proposed algorithm having three modified patterns, namely Lego, Stealth, and Triangular.

[Table 1. RMSE of reconstructed images.]

From Table 1, the average RMSEs for the DAS and LDCT are 29.8121 and 5.2969, respectively, whereas our three novel patterns, Lego, Stealth, and Triangular, produced results that are better at 4.9255, 4.0495, and 3.9154, respectively. This result contributes to improvements of 7.01%, 23.55%, and 26.08% over the prior method (LDCT). This illustrates the superior efficiency of the new patterns where the best pattern is Triangular.

3.3 PSNR

The PSNR data are shown in Table 2, where 30 dB or higher values are commonly acceptable. The PSNRs for the LDCT and the three proposed patterns are all higher than 30. The triangular pattern achieves a high PSNR of 36.3961 and outperforms LDCT by 7.93%. Figures 7-9 show that all results with PSNR > 30 exhibit high-quality reconstructed images, in contrast to DAS, which has an average PSNR < 20, which results in a considerably higher error in the recovered image and aligned with the RMSEs in Table 1.

[Table 2. PSNR of reconstructed images.]

3.4 SSIM

If the SSIM value is 1.0, then the two images are precisely the same. While DAS provides a respectable SSIM value of 0.71 that is in line with the RMSE and PSNR values in Tables 1 and 2, the SSIM value of the reconstructed image for LDCT and the three new patterns in Table 3 is highly acceptable at around 0.98.

[Table 3. SSIM of reconstructed images.]

3.5 Additional algorithm for improving stego image

When compared to the rectangular pattern in the previous section, the proposed patterns provided improved decoding results. What about the stego image encoding results? Is it possible to apply similar patterns on a cover mask in order to enhance the stego image quality? By leveraging a pattern comparable to the hidden DCT blocks, we will expand our study by modifying the concealed region in the cover mask.

The cover mask's regular arrangement pattern is shown in Figure 4 (e). This pattern overlaps with the cover DCT's middle-frequency region, which is also important. Our proposed algorithms can be modified to prevent the merging of hidden data in this middle region. The proposed DCT arrangement patterns in Figure 4 (b)–(d) can be diagonally reversed. This novel design will result in less concealed data in the middle-frequency region of the cover DCT, which should improve the quality of the stego image.

Table 4 reports the outcomes of applying our modified algorithm for generating stego images. The results of the three reverse patterns show a significant reduction in the RMSE to 4.027 when compared to LDCT, which uses a traditional rectangular pattern and has an RMSE of 5.737. This result contributes to an improvement of 29.81%. Additional experiment results show similar improvements of 9.13% and 1.91%, respectively, for PSNR and SSIM. Be aware that in DAS, the cover image's non-significant DCT coefficients were the only area where the secret data were concealed, giving a better RMSE value of 2.4879 (at the expense of a lower payload capacity). Our approach, however, made use of a full-size hidden image, which has a substantially greater payload capacity.

[Table 4. RMSE of stego images.]

4. Applications

Stegano application is also put into practice and tested on a web application. Our web application's back end was created using Python and the Flask framework, while the front end was created using the Next.js framework. The steganography algorithms were stored on the server. The main page (see Figure 10 (a)) has three functions: Encode, Decode, and Text-to-Image (still under testing and functional improvement)[2]. Farmers or product owners may input the cover image and the proprietary secret image (see Figure 10 (b)). The application will then upload the images to the server and run the encoding algorithm there. Our web application performs reverse procedures to retrieve stego information for the decoding steps before displaying the reconstructed hidden image, as illustrated in Figure 10 (c).

[Figure 10. Web application's user interface.]

5. Conclusions

Due to the rise of copyright infringements, it is crucial to preserve who owns various types of online information. Image steganography is a technique used to conceal certain secret information in the cover image, such as information that is protected by copyright. We proposed various enhanced patterns for choosing hidden data and concealing regions in order to enhance the quality of image steganography.

Our primary contribution was the development of a new triangular DCT arrangement pattern by enhancing the existing rectangular DCT coefficient layout design. Our approach has a very high payload capacity since it allows for a full-size hidden image that can be as large as the cover image. The suggested algorithm reduced the hidden

[2] An early version of the website is available at https://image-stegano.vercel.app/.

image's reconstruction errors by 26.08%. The additional algorithm based on the new reverse triangular pattern was suggested to further improve the quality of the stego image being encrypted. The outcome improves the quality of the stego image by 29.81%.

By concealing the ownership information in images before they are posted online, the proposed approach can be employed as a tool for proprietary information protection. Further work may be done to strengthen the algorithm's resilience if the stego image is rotated, resized, or cropped. While the farm profile of the agricultural products is concealed in this study for online marketing purposes, the concept of safeguarding sensitive data can be extended to various other domains, including healthcare, finance, and banking.

Acknowledgement

The Prince of Songkla University's College of Computing in Phuket, Thailand is funding this project. The authors would also like to thank our research colleagues from INFAR and SINT-LAB for their assistance.

References

Baziyad, M., Rabie, T., & Kamel, I. (2018). Extending steganography payload capacity using the L*a*b* color space, *Proceedings of the International Conference on Innovations in Information Technology (IIT)*, pp. 1–6, doi: 10.1109/ INNOVATIONS.2018.8606008.

Elharrouss, O., Almaadeed, N., & Al-Maadeed, S. (2020). An image steganography approach based on k-least significant bits (k-LSB). *Proceedings of the IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, pp. 131–135, doi: 10.1109/ICIoT48696.2020.9089566.

Emmanuel, G., Hungil, G.G., Maiga, J., & Santoso, A. J. (2021). Information hiding in images using Discrete Cosine Transform, *IOP Conference Series: Materials Science and Engineering*, 1098, 052083, doi: 10.1088/1757-899X/1098/ 5/052083.

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

Hashim, M.M., Rahim, M.S.M., Johi, F.A., Taha, M.S., & Hamad, H.S. (2018). Performance evaluation measurement of image steganography techniques with analysis of LSB based on variation image formats, *International Journal of Engineering and Technology Innovation*, 7(4), 3505–3514.

Heednacram, A., Keaomanee, Y. (2023). Four Enhanced Algorithms for Full Size Image Hiding in Chest X-ray Images, Manuscript submitted.

Hussain, M.M., Wahab, A. W. A., Idris, Y. I. B., Ho, A. T., & Jung, K.H. (2018). Image steganography in spatial domain: A survey, *Signal Processing: Image Communication*, 65, 46–66.

Kadhim, I. J., Premaratne, P., Vial, P. J., & Halloran, B. (2019). Comprehensive survey of image steganography: techniques, evaluations, and trends in future research. *Neurocomputing*, 335, 299–326.

Karawia, A. A. (2021). Medical image steganographic algorithm via modified LSB method and chaotic map. *IET Image Processing*, 15(11), 2580–2590.

Khan, S., Irfan, M. A., Arif, A., Rizvi, S. T. H., Gul, A., Naeem, M., & Ahmad, N. (2019). On hiding secret information in medium frequency DCT components using least significant bits steganography, *Computer Modeling in Engineering and Sciences*, 118(3), 529–546.

Prabu, R. G., Latha K. (2020). Ultra-secure secret communication by crypto stegano techniques for defence applications. *Acute Cardiac Care*, 10 (07), 75–83.

Rabie, T., Baziyad, M., & Kamel, I. (2019). High Payload Steganography: Surface-Fitting the Transform Domain, *Proceedings of the International Conference on Communications, Signal Processing, and their Applications*, pp. 1–6, doi: 10. 1109/ ICCSPA.2019.8713731.

Saini, N., Kumar, N. (2023). Development of Amalgamation Approach to Strengthen Security using Watermarking: A Review. *Journal of Algebraic Statistics*, 14(1), 117–123.

Siddiqui, G. F., Iqbal, M. Z., Saleem, K., Saeed, Z., Ahmed, A., Hameed, I. A., & Khan, M. F. (2020). A dynamic three-bit image steganography algorithm for medical and e-healthcare systems. *IEEE Access*, 8, 181893–181903.

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

Thangadurai, K., Devi, G. S. (2014). An analysis of LSB based image steganography techniques, *Proceedings of the International Conference on Computer Communication and Informatics, Coimbatore*, pp. 1–4, doi: 10.1109/ICCCI.2014. 6921751.

Tsai, S. E., Yang, S. M. (2017). A Fast DCT Algorithm for Watermarking in Digital Signal Processor, *Mathematical Problems in Engineering*, 5(5), 1-7.

Vakani, H., Abdallah, S., Kamel, I., Rabie, T., & Baziyad, M. (2021). DCT-in-DCT: A Novel Steganography Scheme for Enhanced Payload Extraction Quality, *Proceedings of the IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, pp. 201–206, doi: 10.1109/IAICT52856.2021.9532553.

Vakani, H., Kamel, I., Rabie, T., & Baziyad, M. (2020). Towards improving the imperceptibility of steganography schemes: adaptive scaling approach. *Proceedings of the 14th International Conference on Innovations in Information Technology (IIT)*, pp. 51–56, doi: 10.1109/IIT50501.2020.9299040.

Wang, J. , Cheng, M. , Wu, P. , & Chen, B. (2019) . A survey on digital image steganography. *Journal of Information Hiding and Privacy Protection*, 1(2), 87–93.

Figure 1. Cover (top row) and hidden (last 2 rows) images.

Figure 2. Three primary frequency components.

Figure 3. Selection area of cover and hidden DCT coefficients in embedded process.

Figure 4. DCT arrangement patterns for hidden (a – d) and cover (e) images.

Figure 5. Diagram for encoding process.

Figure 6. Diagram for decoding process.

Figure 7. Reconstructed images with Mangosteen as a cover image.

Figure 8. Reconstructed images with Durian as a cover image.

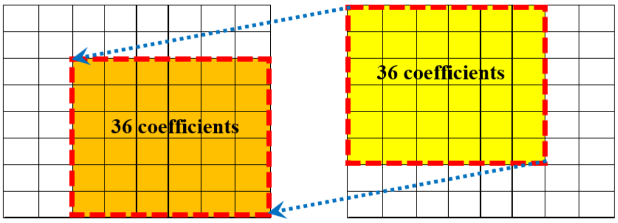
Figure 9. Reconstructed images with Pomelo as a cover image.

Figure 10. Web application's user interface.

Figure 1. Cover (top row) and hidden (last 2 rows) images.

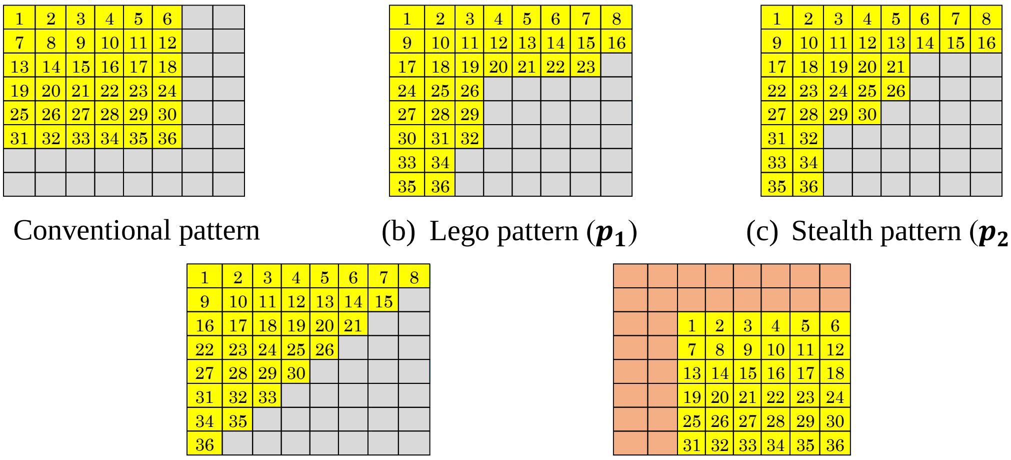
0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

Figure 2. Three primary frequency components.



(a) Cover DCT (b) Hidden DCT

Figure 3. Selection area of cover and hidden DCT coefficients in embedded process.



(a) Conventional pattern (b) Lego pattern ($\boldsymbol{p_1}$) (c) Stealth pattern ($\boldsymbol{p_2}$)

(d) Triangular pattern ($\boldsymbol{p_3}$) (e) Cover mask

Figure 4. DCT arrangement patterns for hidden (a – d) and cover (e) images.

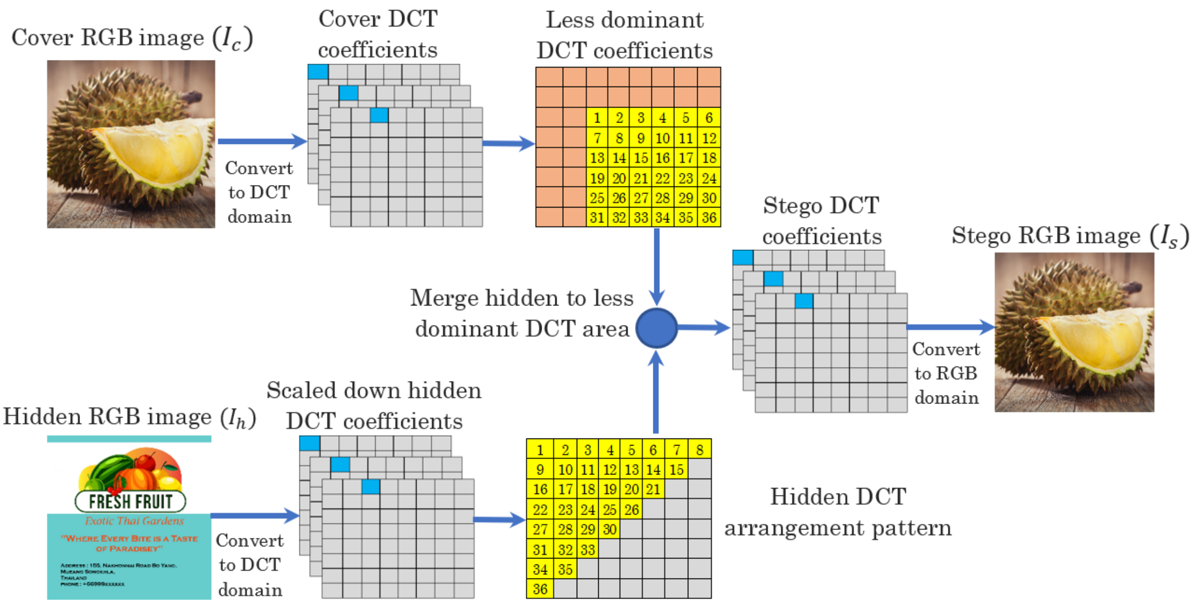


Figure 5. Diagram for encoding process.

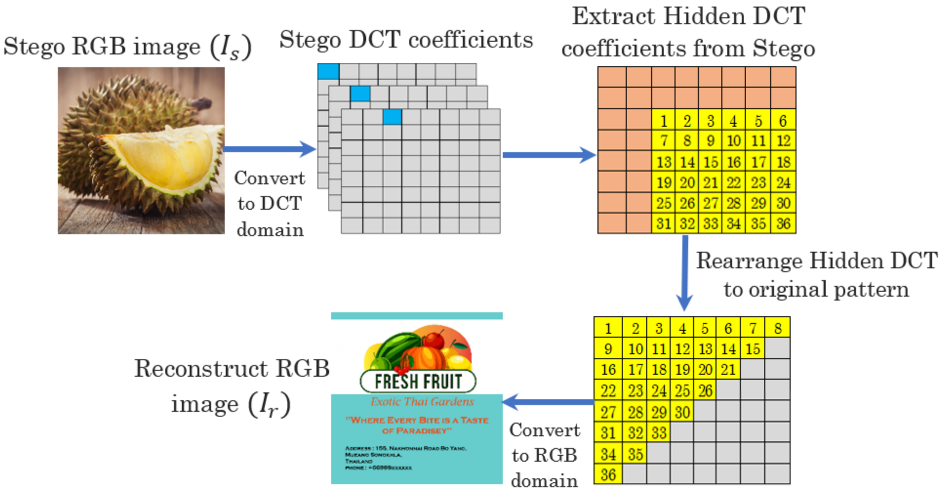


Figure 6. Diagram for decoding process.

(a) Cover image (b) Hidden image (c) DAS (d) LDCT

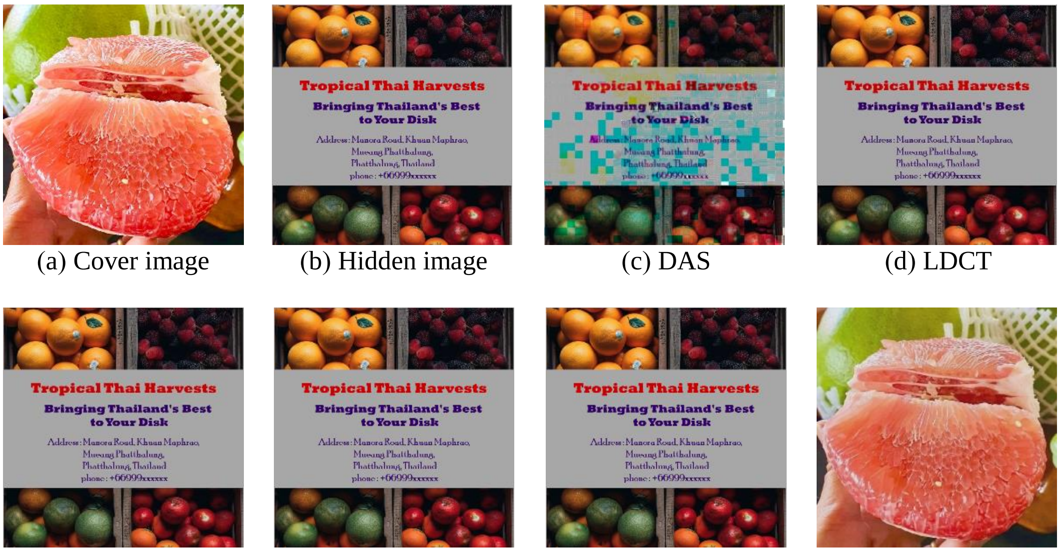
(e) Reconstructed with Lego pattern (f) Reconstructed with Stealth pattern (g) Reconstructed with Triangular pattern (h) Stego image

Figure 7. Reconstructed images with Mangosteen as a cover image.

(a) Cover image (b) Hidden image (c) DAS (d) LDCT

(e) Reconstructed with Lego pattern (f) Reconstructed with Stealth pattern (g) Reconstructed with Triangular pattern (h) Stego image

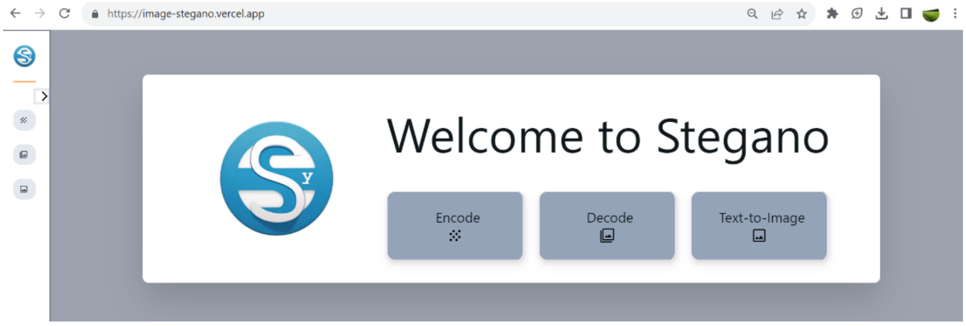
Figure 8. Reconstructed images with Durian as a cover image.



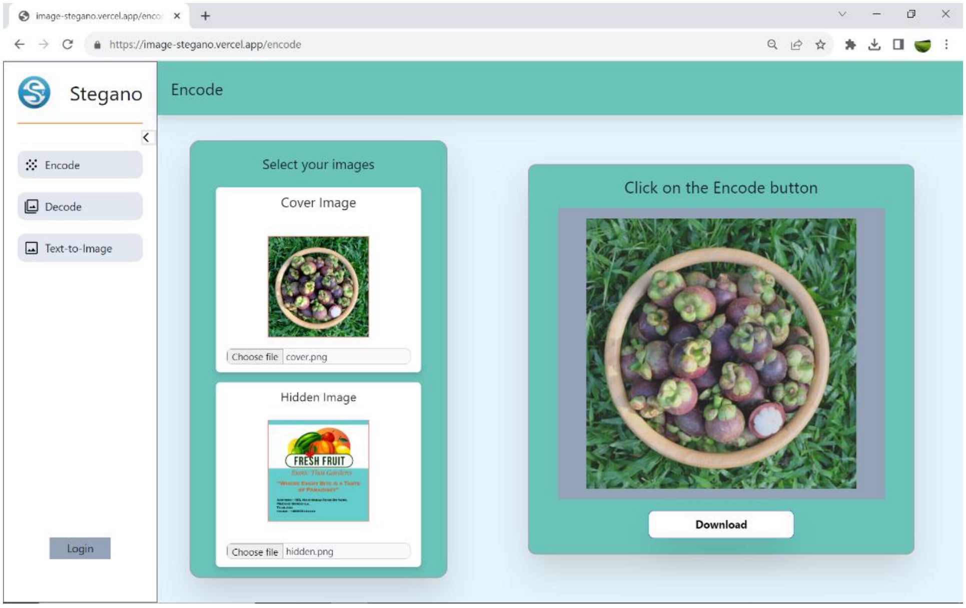
(a) Cover image (b) Hidden image (c) DAS (d) LDCT

(e) Reconstructed with Lego pattern (f) Reconstructed with Stealth pattern (g) Reconstructed with Triangular pattern (h) Stego image

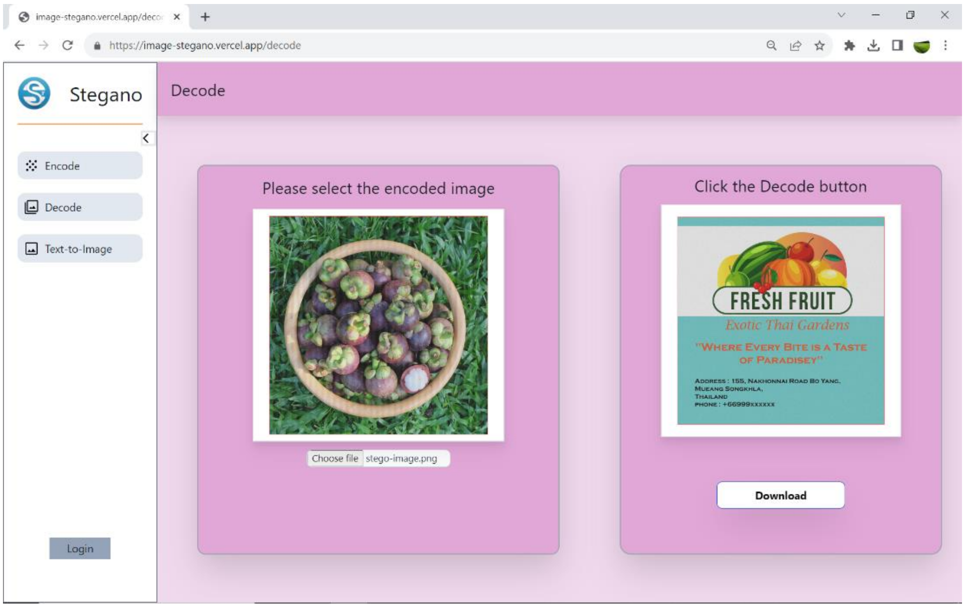
Figure 9. Reconstructed images with Pomelo as a cover image.



(a) Main page



(b) Encode page



(c) Decode page

Figure 10. Web application's user interface.

Table 1. RMSE of reconstructed images.

Table 2. PSNR of reconstructed images.

Table 3. SSIM of reconstructed images.

Table 4. RMSE of stego images.

Table 1. RMSE of reconstructed images.

Stego No.	Cover No.	Hidden No.	RMSE				
			DAS	LDCT	Our proposed algorithm		
					Lego	Stealth	Triangular
1	1	1	36.0509	5.2000	4.3360	3.5841	3.4811
2	1	2	37.5585	5.2744	5.4004	4.6268	4.4706
3	1	3	24.6602	6.2434	6.1965	5.1479	4.9482
4	1	4	34.0313	4.2020	4.0174	3.4439	3.3443
5	1	5	24.6674	5.5649	4.6773	3.4449	3.3329
6	2	1	18.8523	5.2000	4.3360	3.5841	3.4811
7	2	2	18.0316	5.2744	5.4004	4.6268	4.4706
8	2	3	17.2678	6.2434	6.1965	5.1479	4.9482
9	2	4	18.0249	4.2020	4.0174	3.4439	3.3443
10	2	5	15.4329	5.5649	4.6773	3.4449	3.3329
11	3	1	38.5829	5.2000	4.3360	3.5841	3.4811
12	3	2	51.5565	5.2744	5.4004	4.6268	4.4706
13	3	3	37.4943	6.2434	6.1965	5.1479	4.9482
14	3	4	41.0651	4.2020	4.0174	3.4439	3.3443
15	3	5	33.9054	5.5649	4.6773	3.4449	3.3329
Average			29.8121	5.2969	4.9255	4.0495	3.9154

Table 2. PSNR of reconstructed images.

Stego No.	Cover No.	Hidden No.	PSNR				
			DAS	LDCT	Our proposed algorithm		
					Lego	Stealth	Triangular
1	1	1	16.9925	33.8108	35.3890	37.0433	37.2964
2	1	2	16.6366	33.6874	33.4822	34.8253	35.1234
3	1	3	20.2909	32.2223	32.2879	33.8982	34.2418
4	1	4	17.4932	35.6617	36.0520	37.3898	37.6447
5	1	5	20.2883	33.2216	34.7310	37.3873	37.6744
6	2	1	22.6235	33.8108	35.3890	37.0433	37.2964
7	2	2	23.0101	33.6874	33.4822	34.8253	35.1234
8	2	3	23.3861	32.2223	32.2879	33.8982	34.2418
9	2	4	23.0134	35.6617	36.0520	37.3898	37.6447
10	2	5	24.3619	33.2216	34.7310	37.3873	37.6744
11	3	1	16.4029	33.8108	35.3890	37.0433	37.2964
12	3	2	13.8851	33.6874	33.4822	34.8253	35.1234
13	3	3	16.6515	32.2223	32.2879	33.8982	34.2418
14	3	4	15.8614	35.6617	36.0520	37.3898	37.6447
15	3	5	17.5254	33.2216	34.7310	37.3873	37.6744
Average			19.2282	33.7208	34.3884	36.1088	36.3961

Table 3. SSIM of reconstructed images.

Stego No.	Cover No.	Hidden No.	SSIM				
			DAS	LDCT	Our proposed algorithm		
					Lego	Stealth	Triangular
1	1	1	0.6492	0.9872	0.9896	0.9937	0.9936
2	1	2	0.6192	0.9865	0.9858	0.9909	0.9912
3	1	3	0.6829	0.9720	0.9703	0.9803	0.9810
4	1	4	0.6327	0.9902	0.9906	0.9937	0.9933
5	1	5	0.6852	0.9777	0.9757	0.9847	0.9852
6	2	1	0.8483	0.9872	0.9896	0.9937	0.9936
7	2	2	0.8462	0.9865	0.9858	0.9909	0.9912
8	2	3	0.7999	0.9720	0.9703	0.9803	0.9810
9	2	4	0.8373	0.9902	0.9906	0.9937	0.9933
10	2	5	0.8241	0.9777	0.9757	0.9847	0.9852
11	3	1	0.6819	0.9872	0.9896	0.9937	0.9936
12	3	2	0.6464	0.9865	0.9858	0.9909	0.9912
13	3	3	0.6711	0.9720	0.9703	0.9803	0.9810
14	3	4	0.6554	0.9902	0.9906	0.9937	0.9933
15	3	5	0.6766	0.9777	0.9757	0.9847	0.9852
Average			0.7171	0.9827	0.9824	0.9887	0.9888

Table 4. RMSE of stego images.

Stego No.	Cover No.	Hidden No.	RMSE for different algorithms				
			DAS	LDCT	Our algorithm (reverse patterns)		
					Lego	Stealth	Triangular
1	1	1	2.6166	6.6584	4.3615	4.3614	4.3614
2	1	2	2.6883	6.6638	4.3728	4.3726	4.3725
3	1	3	2.2120	6.6222	4.3012	4.3011	4.3011
4	1	4	2.6123	6.6590	4.3601	4.3602	4.3602
5	1	5	2.2062	6.6204	4.3011	4.3010	4.3010
6	2	1	3.2750	5.8463	4.4077	4.4077	4.4076
7	2	2	3.3918	5.8544	4.4181	4.4181	4.4182
8	2	3	2.7702	5.7966	4.3461	4.3462	4.3463
9	2	4	3.2542	5.8455	4.4073	4.4072	4.4072
10	2	5	2.5990	5.7983	4.3475	4.3474	4.3473
11	3	1	2.1299	4.7596	3.3868	3.3868	3.3869
12	3	2	2.2065	4.7681	3.4000	3.3997	3.3997
13	3	3	1.6127	4.7020	3.3038	3.3037	3.3039
14	3	4	2.1234	4.7591	3.3858	3.3858	3.3858
15	3	5	1.6197	4.7027	3.3060	3.3061	3.3061
Average			2.4879	5.7371	4.0271	4.0270	4.0270