Kufa Journal of Engineering
Vol. 17, No. 1, January 2026, P.P. 505-517
Article history: Received 14 March 2025, last revised 23 May 2025, accepted 23 May 2025

IMPROVED COVERLESS INFORMATION HIDING UTILIZING GENERATIVE MODELS

Zainab Hdeib Al-Shably [1*] and Asaad Noori Hashim [1]

[1] Faculty of Computer Science & Mathematics, University of Kufa Najaf; Iraq.

[*] Corresponding author. Email: zainabh.alshably@student.uokufa.edu.iq

https://doi.org/10.30572/2018/KJE/170128

ABSTRACT

The submitted algorithm increased data-hiding capacity, extended capability, and improved quality of the reconstructed secret image as compared with traditional steganography and existing coverless information-hiding methods. Unlike conventional steganography, in which the secret information is embedded into a cover image, our method involves the design of generative models that generate a cover image independently of the secret image; thus, there is no cover image explicitly defined. Two generative models are thus used in the proposed method: F creates a middle image with respect to the secret image, while G reconstructs the secret image from the generated cover image. The content consistency extraction module thus encodes content information from the secret image into the generated cover image so as to address the issues of color distortion and content loss in the reconstructed secret image. The experimental results confirm that our method embeds more data, exhibits larger security, and has higher visual quality based on improved PSNR (30.2313), SSIM (0.7762), and reduced MSE(40.1271) values. This method used a dataset comprising 21,550 images, with its results showing vast improvement compared to existing methods.

KEYWORDS

Steganography ;Generative Adversarial Network(GAN) ;Coverless Information Hiding, Densnet ;Image.

1. INTRODUCTION

Steganography, the act of texting information through a cover medium, has been used since antiquity for safe communication. Conventional approaches to steganography embed secret messages in photographs or any files by slightly altering the pixel values of the original medium (Wang et al., 2021). These techniques often build residual traces in the cover medium and are, therefore, vulnerable to steganalysis, a technique that detects secret information hidden in another medium (Wu and Wang, 2015). Over the years, various methods with improved embedding capacities and security-discussed include, Least Significant Bit (LSB) embedding (Li et al., 2021a), DCT-based methods (Hu, 2006), and more recently, deep-learning-based approaches (Qian et al., 2015).

However, those found in conventional knowledge differ inherently in their limitations. The embedding process, in effect, is capable of modifying cover images, thus introducing statistical traces that can be detected by more carried out in advanced steganalysis tools (Zhou et al., 2015). To respond to such issues, coverless information hiding methods began to be brought into existence, as those methods do not insert secret information into the cover image (Mohammed et al., 2022). They rather rely on mapping relationships between secret messages and the cover and commonly do so using natural features like color, texture, or object relationships (Zhi-li et al., 2016). Sun et al. (Yuan et al., 2017) suggested that visual words taken from the image represent hidden text messages within the bag-of-words method of coverless information stealing. In a similar vein, Zhou et al. (Chen et al., 2015) presented a coverless information hiding approach based on the molecular structure images of materials, where the natural structure of molecules is used to represent secret information.

While coverless information-hiding methods improve security by preventing modification of the cover image, most of these methods have limited embedding capacity and offer unsatisfactory reconstruction quality for the secret image. Besides, most of the existing methods are designed under the text-based secret message application, thereby being limited for hiding larger data like images or videos.

1.1. Motivation and Challange

This work fulfills some of the current shortcomings that prevent optimal performances of current coverless information hiding techniques. Our approach exploits the power of GANs in generating a cover image from the secret image directly, doing away with the need for explicit cover images.

Our work builds upon the framework proposed in (Li et al., 2021b), where a coverless information hiding method based on generative models is introduced. Similar to their approach,

we employ a dual generative model architecture for cover image generation and secret image reconstruction. However, we enhance the feature extraction process via DenseNet, which improves feature representation and content consistency. This modification addresses limitations such as color distortion and content loss, leading to higher embedding rates and superior reconstruction quality. Our method thus advances the state-of-the-art in coverless information hiding.

This paper proposes the following key contributions:

Information Hiding Without Covers Based on Generative Models: We propose a novel generative model-based approach for coverless information hiding such that there is no explicit cover image.

DenseNet Extraction Module: We present a DenseNet extraction module that encodes into the generated cover image the content information of the secret image, with a consequent significant enhancement of the quality of the reconstructed secret image.

Higher Capacity and Security: Our method also achieves substantially higher embedding capacity and improved security over other existing coverless information hiding methods, as demonstrated through experiments.

Extensive Evaluation: An extensive experiment is now ongoing to assess our approach concerning embedding capacity, security, and reconstruction quality using PSNR, SSIM, and MSE.

2. PROPOSED METHODOLGY

The proposed method introduces coverless information hiding based on generative models. The goal of this method is to improve the embedding capacity and security of coverless information hiding by leveraging the power of deep learning and generative adversarial networks (GANs). While traditional steganography methods embed secret information into a cover image, the proposed method generates a cover image that is independent of the secret image but contains its content information. This approach eliminates the need for embedding, thereby enhancing security and avoiding detection by steganalysis tools see Fig. 1.

The proposed framework consists of three main stages Generation of the Middle Image, Generation of the Cover Image and Generation of the Cover Image that we explain them below:

2.1. Middle Set Generation

The first step in the framework is the generation of a middle image I_m. This image serves as an intermediate representation that bridges the true image I_t (a natural image unrelated to the secret image) and the final cover image I_c. (Sivalingan, 2024) The GAN framework used here can be described as a minimax game between Generator F and Discriminator D_Y. The objective

function for the GAN is given by:

$$\min_{F} \max_{D_Y} V(F, D_Y) = \mathbb{E}_{I_t \sim p_{\text{data}}(I_t)}[\log D_Y(I_t)] + \mathbb{E}_{I_s \sim p_{\text{data}}(I_s)}\left[\log\left(1 - D_Y\left(F(I_s)\right)\right)\right] \quad (1)$$

where I_t is True images (real images from the dataset), $F(I_s)$ the Middle image I_m generated by Generator F, $D_Y(I_t)$ the Discriminator's probability that I_t is a real image and $D_Y(F(I_s))$ is the discriminator's probability that I_m is a real image.

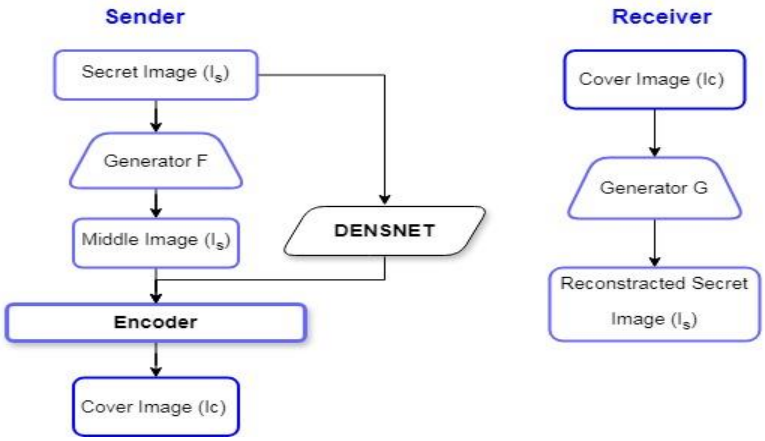

Fig. 1. Flowchart of the proposed method, showing the three stages (Middle image, Cover image and Reconstructed image) by Secret image and True image along with the interaction between the generative models F and G.

The process of generating the middle image I_m from the true image I_t involves several key steps. The input to Generator F is the secret image I_s, which is a 3-channel RGB image of size $H \times W$. The goal of Generator F is to transform I_s into the middle image I_m, which is also a 3-channel RGB image of the same size. Generator F consists of multiple convolutional layers, each of which applies a set of filters to the input image. The convolutional operation can be expressed as:

$$I_m^{(l)}(x, y) = \sum_{i=0}^{k-1} \sum_{j=0}^{k-1} I_m^{(l-1)}(x + i, y + j) \cdot K_l(i, j) + b_l \quad (2)$$

where $I_m^{(l-1)}$ is the Input feature map to the l-th layer, $K_l(i, j)$ is the convolution kernel (filter) for the l-th layer, b_l is the bias term for the l-th layer and finally $I_m^{(l)}$ is the output feature map of the l-th layer.

The convolutional layers extract hierarchical features from the secret image I_s, starting with low-level features (e.g., edges and textures) and progressing to higher-level features (e.g., shapes and structures).

After each convolutional layer, the feature maps are passed through a batch normalization layer. Batch normalization standardizes the output of the convolutional layer to have a mean of 0 and a variance of 1. This is achieved by:

$$\hat{I}_m^{(l)}(x,y) = \frac{I_m^{(l)}(x,y)-\mu_l}{\sqrt{\sigma_l^2+\epsilon}} \quad (3)$$

where μ_l is the mean of the feature map over the batch, σ_l^2 is the variance of the feature map over the batch and ϵ is the small constant to avoid division by zero. The normalized feature map $\hat{I}_m^{(l)}$ is then scaled and shifted using learnable parameters γ_l and β_l:

$$I_m^{(l+1)}(x,y) = \gamma_l \cdot \hat{I}_m^{(l)}(x,y) + \beta_l \quad (4)$$

Batch normalization stabilizes the training process and accelerates convergence. The normalized feature maps are passed through a Rectified Linear Unit (ReLU) activation function, which introduces non-linearity into the network. The ReLU function is defined as:

$$I_m^{(l+1)}(x,y) = \max\left(0, I_m^{(l+1)}(x,y)\right) \quad (5)$$

ReLU ensures that only positive values are propagated forward, which helps the network learn complex patterns.

The final step in Generator F is the application of a tanh activation function to the output of the last convolutional layer. The tanh function ensures that the pixel values of the middle image I_m are in the range $[-1,1]$, which is standard for normalized image data:

$$I_m(x,y) = \tanh(I_m(x,y)) = \frac{e^{Im(x,y)}-e^{-Im(x,y)}}{e^{Im(x,y)}+e^{-Im(x,y)}} \quad (6)$$

The true image I_r is used to train the discriminator D_Y to distinguish between real and generated images. The discriminator D_Y is trained to maximize the probability of correctly classifying I_r as real and I_m as fake. The loss function for the discriminator D_Y is:

$$\mathcal{L}_{D_Y} = -\mathbb{E}_{I_t\sim p_{\text{data}}(I_t)}[\log D_Y(I_t)] - \mathbb{E}_{I_s\sim p_{\text{data}}(I_s)}[\log(1-D_Y(F(I_s)))] \quad (7)$$

where $D_Y(I_t)$ is discriminator's probability that I_s is secret and $D_Y(F(I_s))$ is discriminator's probability that I_m is real. Generator F, on the other hand, is trained to minimize the probability that the discriminator D_Y correctly classifies I_m as fake. The adversarial loss for Generator F is:

$$\mathcal{L}_{\text{adv}}^F = \mathbb{E}_{I_s\sim p_{\text{data}}(I_s)}\left[\log\left(1-D_Y\left(F(I_s)\right)\right)\right] \quad (8)$$

This adversarial process drives Generator F to produce middle images I_m that are indistinguishable from secret images I_s.

The proposed method consists of two main stages: cover image generation and secret image reconstruction. The overall architecture of the method is illustrated in Fig.1 (suggested diagram: a flowchart showing the two stages and the interaction between the generative models F and G).

2.2. Generation of Cover Image

The generation of the cover image I_c is a key step in the proposed coverless information hiding framework. The cover image is generated in such a way that it contains the content information

of the secret image I_s but appears as an independent and natural image, unrelated to I_s. This prevents any suspicion from falling on the cover image during transmission, which would lead to a safer passage for the hidden information.

The generation of the cover image I_c involves two main components: the Generator F and the Encoder Econtent. The whole process can basically be divided into two phases.

The second step encodes the content information into Cover Image I_c. The encoder econtent encodes the content information of the secret image I_s into the middle image I_m to finally produce the cover image I_c. The content information from I_s is extracted using a pre-trained DENSNET network that outputs a feature vector $v_c \in \mathbb{R}^{512}$ from a certain layer of the network:

$$v_c = \text{DENSNET}(I_s).$$

The feature vector v_c is then reshaped and resized to match the spatial dimensions of I_m. The Encoder Econtent is a co-attention based encoder that combines I_m and v_c to produce the cover image I_c:

$$I_c = E_{\text{content}}(I_m, v_c) \quad (9)$$

where $E_{\text{content}}: \mathbb{R}^{H\times W\times 3} \times \mathbb{R}^{512} \to \mathbb{R}^{H\times W\times 3}$ is a deep CNN that encodes the content information into the middle image (Abdulhamid and Hashim,2023,p.2). The architecture of encoder econtent is illustrated in convolutional layers that that carry out the element-wise sum of element-wise products and convolutional operation. In other words, convolutional layers process the concatenation of I_m and the reshaped v_c. The output I_c is designed to be visually indistinguishable from a natural image and also embodies the features of input I_s which have been content-concatenated and obtained as an intermediate target. The training of the cover image generation process involves minimizing several loss functions to ensure that the generated cover image I_c is both natural-looking and contains the necessary content information for reconstructing the secret image. The first function is adversarial loss for generator F in which the Generator F is trained to generate I_m that is indistinguishable from natural images. As we said before, this is achieved using an adversarial loss $\mathcal{L}_{\text{adv}}^{F}$, where a discriminator D_Y tries to distinguish between secre images I_t and generated middle images I_m:

$$\mathcal{L}_{\text{adv}}^{F} = \mathbb{E}_{I_t}[\log D_Y(I_t)] + \mathbb{E}_{I_m}[\log(1 - D_Y(I_m))] \quad (10)$$

The second function is reconstruction loss for encoder econtent in which the encoder econtent is trained to ensure that the cover image I_c is visually similar to the middle image I_m. This is achieved using a reconstruction loss $\mathcal{L}_{\text{view}}$, which minimizes the L_2 distance between I_c and I_m:

$$\mathcal{L}_{\text{view}} = \| I_c - I_m \|_2^2 \quad (11)$$

The final function is content consistency loss. To ensure that the cover image I_c contains the

content information of the secret image I_s, a content consistency loss $\mathcal{L}_{\text{content}}$ is used. This loss ensures that the feature maps extracted from I_c by the DENSNET network are consistent with those extracted from I_s:

$$\mathcal{L}_{\text{content}} = \| \text{DENSNET}(I_c) - \text{DENSNET}(I_s) \|_2^2 \quad (12)$$

The total loss for the cover image generation process is a weighted combination of the adversarial loss, reconstruction loss, and content consistency loss:

$$\mathcal{L}_{\text{cover}} = \lambda_{\text{adv}}\mathcal{L}_{\text{adv}}^F + \lambda_{\text{view}}\mathcal{L}_{\text{view}} + \lambda_{\text{content}}\mathcal{L}_{\text{content}} \quad (13)$$

where λ_{adv}, λ_{view}, and λ_{content} are hyperparameters that control the relative importance of each loss term.

2.3. Secret Image Reconstruction

In the third stage, the generative model G reconstructs the secret image I_s from the cover image I_c. The Generator G is a critical component of the proposed framework, responsible for reconstructing the secret image I_s from the cover image I_c. The goal of Generator G is to ensure that the reconstructed secret image $\hat{I}_s$ is as close as possible to the original secret image I_s, both in terms of pixel-level fidelity and semantic content. (Algorithm1) shows the step of reconstructing the Secret Image.

Algorithm 1: Reconstructing the Secret Image Using Generative Model G

Inputs:
Original secret image: $I_s \in \mathbb{R}^{H\times W\times 3}$
Cover image: $I_c \in \mathbb{R}^{H\times W\times 3}$ (generated by Encoder E_{content})
Generator G: A deep neural network with convolutional layers, batch normalization, and activation functions.
Discriminator D_X: A neural network for adversarial training. Hyperparameters: λ_{rec}, λ_{adv}

Outputs:
Reconstructed secret image: $\hat{I}_s \in \mathbb{R}^{H\times W\times 3}$

Step1
Preprocess the Cover Image I_c: Resize the cover image I_c to match the dimensions of the secret image I_s:

$$I_c^{\text{resized}} = \text{Resize}(I_c)$$

Concatenate the original secret image I_s and the resized cover image I_c^{resized} along the channel dimension:

$$X = \text{Concat}(I_s, I_c^{\text{resized}})$$

Here, $X \in \mathbb{R}^{H\times W\times 6}$.

Step2
Generate the Reconstructed Secret Image $\hat{I}_s$: Pass the concatenated input X through the generator G:

$$\hat{I}_s = G(X)$$

The generator G consists of the following operations: First Convolutional Layer:

$$F_1 = \text{ReLU}(\text{BatchNorm}(\text{Conv}(X)))$$

Here, $F_1 \in \mathbb{R}^{H\times W\times 64}$. Second Convolutional Layer:

$$F_2 = \text{ReLU}(\text{BatchNorm}(\text{Conv}(F_1)))$$

Here, $F_2 \in \mathbb{R}^{H\times W\times 64}$. Final Convolutional Layer:

$$\hat{I}_s = \text{Tanh}(\text{Conv}(F_2))$$

Here, $\hat{I}_s \in \mathbb{R}^{H\times W\times 3}$.

Step3	Compute Loss Functions: Reconstruction Loss L_{rec}: $L_{\text{rec}} = \| \hat{I}_s - I_s \|_2^2$ Adversarial Loss L_{adv}: $L_{\text{adv}} = \mathbb{E}_{I_s}[\log D_X(I_s)] + \mathbb{E}_{\hat{I}_s}[\log(1 - D_X(\hat{I}_s))]$ Total Loss for Generator G: $L_G = \lambda_{\text{rec}} L_{\text{rec}} + \lambda_{\text{adv}} L_{\text{adv}}$
Step4	Optimize the Generator G: Update the parameters of G using gradient descent to minimize the total loss L_G.
Step5	CycleConsistency Loss (Optional): If cycleconsistency is enforced, compute the cycleconsistency loss: $L_{\text{cycle}} = \| G(E_{\text{content}}(F(I_s), v_c)) - I_s \|_2$ Add this loss to the total loss L_G if required.
Step6	Adversarial Training for Discriminator D_X: Update the discriminator D_X to distinguish between real secret images I_s and reconstructed secret images $\hat{I}_s$: $L_{D_X} = (D_X(I_c) - 1)^2$.
Step7	Repeat steps 1–6 for multiple iterations until convergence.

2.4. Densnet Extraction Module

The Densnet module is one of the essential modules of our method, which guarantees the reconstruction of the secret image with original content and the least color distortion. The module extracts the content vector v_c from the secret image via a pre-trained DENSNET network and encodes it into the middle image within the cover image generation stage. The purpose of this process is to enable the generated cover image to contain sufficient content information for complete reconstruction of the secret image. The loss is defined as:

$$\mathcal{L}_{view} = \| I_c - F(I_s) \|_2 \quad (14)$$

3. EXPERIMENTAL RESULTS

The numerical experiments were conducted in Google Colab, utilizing its GPU-enabled environment to accelerate computations. The Animation dataset from the Kaggle platform was used for training and evaluation. We evaluate our method on a dataset of 13,000 images, consisting of cartoon character images and real face images. The images are resized to 256×256 pixels, and the batch size is set to 32. The primary objective of these experiments was to assess the reconstruction accuracy of the proposed model over a series of 100 epochs. The generative models F and G are trained using the Adam optimizer with a learning rate of 0.0002.

Each image was normalized to the range [-1, 1] using a standard normalization process. From the preprocessed dataset, two disjoint subsets, X and Y, were created, each containing randomly selected images. For facilitating training and evaluation, the subsets were further divided into training and testing with an 80-20 percent ratio: Data was loaded using PyTorch's DataLoader functionality with a batch size of 32, ensuring shuffling of the data throughout training and sequential access during testing.

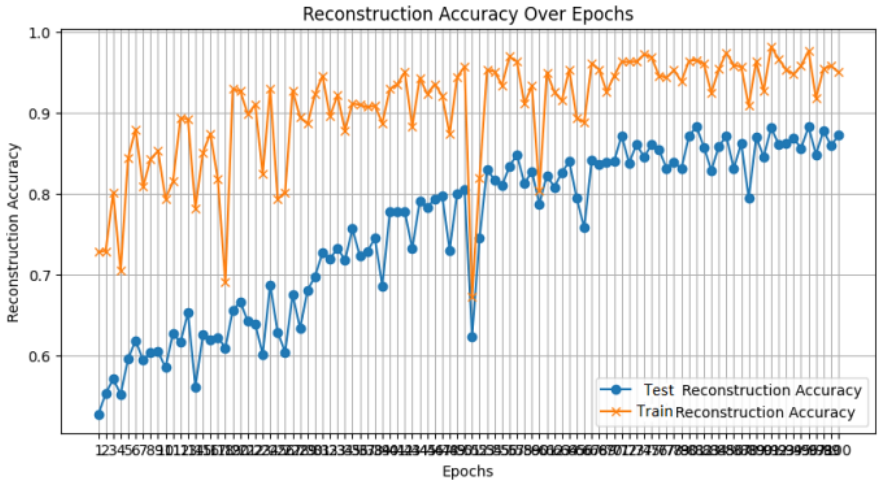

Fig. 2. Reconstruction accuracy for training and test data over 100 iterations, showing good convergence

Fig. 3. Visual comparison of secret images, middle images, cover images, and reconstructed images, demonstrating high-quality reconstruction.

The model proposed was kept under training for 100 epochs with the anomaly-detection feature turned on to observe any possibly arising computational anomalies. The reconstruction accuracy was calculated on both training and test data for every single epoch. The experiments also involved the visualization of generated images for a qualitative assessment of the model's performance. Fig. 2 depicts the reconstructions violence accuracy, in tandem, hopefully in conducive manner towards training data and testing data in 100 epochs. Results show convergence towards strongly, consistently improving reconstruction accuracy among iterations. The reconvening curve of training and testing accuracies implies that the model generalizes well without overfitting. The quality of the reconstructed images is assessed quantitatively and qualitatively. In tabular form, Fig.3 shows the visual comparison of the secret images, middle images, cover images and reconstructed images. It is evident from the results that the model reconstructs the secret images with a very high fidelity. The output images closely resemble their secret image counterparts, proving that the model captures important

features and detail of the image. The numerical results, coupled with visual evidence, confirm the effectiveness of the proposed method more than adequately. Thus, the model yields a very good estimate of the original secret image from both experimental and tested datasets, demonstrating robust convergence. Furthermore, visual induction corroborates that high-quality reconstructions produce excellent results for the intended image reconstruction tasks and can, therefore, be presented as an ideal solution. The numerical experiments validate the proposed model in high-fidelity image reconstruction tasks. The convergence of both training and testing datasets in reconstruction accuracy reflects the model's robustness. In addition, visual assessments confirm that the reconstructed images retain the main features of the original input images.

We assess the performance of our method against the following metrics:

Peak Signal-to-Noise Ratio (PSNR): Represents the quality of the reconstructed secret image.

Structural Similarity Index (SSIM): Measures the structural similarity between the original and reconstructed secret images (Al-Asady et al., 2024).

Mean Squared Error (MSE): Represents the pixel-wise difference between the original and reconstructed secret images.

The numerical results of our experiments are summarized in Table 1 and Table 2. Table 1 presents a comparative analysis of the embedding capacity of our proposed method against existing coverless information hiding techniques. Embedding capacity refers to the amount of secret information that can be hidden within a single cover image. Traditional methods such as those proposed by Zhou et al. (Zhili Zhou et al., 2017) and Yuan et al. (Yuan et al., 2017) exhibit limited embedding capacity, typically around 8 bits per cover image. More recent approaches, such as those by Chen et al. (Chen et al., 2015)and Cao et al.(Cao et al., 2018) , have improved capacity, with values reaching 16 and 36 bits per cover image, respectively. Sun et al. (Cao et al., n.d.) further enhanced embedding capacity to 68 bits per cover image. In contrast, our proposed method and (Li et al., 2021b) achieve a significantly higher embedding capacity of $256 \times 256 \times 3 \times 8$ bits per cover image due to its generative approach, which eliminates the constraints imposed by traditional steganography and cover-based hiding methods. This substantial increase in embedding capacity highlights the superiority of our method in efficiently encoding secret information. Table 2 evaluates the quality of the reconstructed secret image by comparing PSNR, SSIM, and MSE values under different conditions. Specifically, it examines the impact of on reconstruction quality. The table compares the results of previous research and our proposed method.

Table 1: Comparison of Embedding Capacity

Method	Embedding Capacity (bits/cover)
(Zhili Zhou et al., 2017)	8
(Yuan et al., 2017)	8
(Chen et al., 2015)	16
(Cao et al., 2018)	36
(Cao et al., n.d.)	68
(Li et al., 2021b)	$256 \times 256 \times 3 \times 8$
Proposed Method	$256 \times 256 \times 3 \times 8$

Table 2: Comparison of the quality of Reconstructed Secret Image

Method	PSNR	SSIM	MSE
(Li et al., 2021b)	29.2313	0.7362	76.1271
Our Proposed	30.2313	0.7762	40.1271

4. CONCLUSION

In this paper, we proposed an enhanced Coverless Information Hiding method based on generative models. Our method eliminates the need for an explicit cover image and significantly improves the embedding capacity, security, and quality of the reconstructed secret image. The experimental results demonstrate that our method outperforms existing coverless information hiding methods in terms of embedding capacity, security, and reconstruction quality. Future work will focus on further improving the fidelity of the generated cover image and the quantitative quality of the reconstructed secret image.

5. REFERENCES

Abdulhamid, M.A. and Hashim, A.N. (2023) 'Enhanced preprocessing stage for feature extraction of deepfake detection based on deep learning methods', in 2023 7th International Symposium on Innovative Approaches in Smart Technologies (ISAS), 1–6. IEEE.

Al-Asady, H. A. J., Fakhruldeen, H. F., & Mudhafar, Y. (2024) 'An image encryption method based on logistical chaotic maps to encrypt communication data', Kufa Journal of Engineering, 15(4), pp. 55-64.

Cao, Y., Zhou, Z., Sun, X. and Gao, C. (2018) 'Coverless information hiding based on the molecular structure images of material', Computational Materials Continua, 54(2), pp. 197–207.

Cao, Y., Zhou, Z., Yang, C.-N. and Sun, X. (2018) 'Dynamic content selection framework applied to coverless information hiding', Journal of Internet Technology, 19(4), pp. 1179–1186.

Chen, X., Sun, H., Tobe, Y., Zhou, Z. and Sun, X. (2015) 'Coverless information hiding method based on the Chinese mathematical expression', in International Conference on Cloud Computing and Security.

Hu, Y.C. (2006) ‘High-capacity image hiding scheme based on vector quantization’, Pattern Recognition, 39(9), pp. 1715–1724.

Li, Q., Wang, X., Wang, X. and Shi, Y. (2021) ‘CCCIH: content-consistency coverless information hiding method based on generative models’, Neural Processing Letters, 53(6), pp. 4037–4046.

Li, Q., Wang, X.Y., Wang, X.Y., Ma, B., Wang, C.P. and Shi, Y.Q. (2020) ‘An encrypted coverless information hiding method based on generative models’, Information Sciences, 553(3).

Mohammed, H. A., Kareem, S. W., & Mohammed, A. S. (2022) 'A comparative evaluation of deep learning methods in digital image classification', Kufa Journal of Engineering, 13(4).

Qian, Y., Jing, D., Wei, W. and Tan, T. (2015) ‘Deep learning for steganalysis via convolutional neural networks’, in Proceedings of SPIE - The International Society for Optical Engineering, vol. 9409.

Sivalingan, H. (2024) 'Cloud-smart surveillance: Enhancing anomaly detection in video streams with DF-ConvLSTM-based VAE-GAN', Kufa Journal of Engineering, 15(4), pp. 125-140.

Sun, X., Zhou, Z., Yuan, C. and Xia, Z. (2017) ‘Coverless image steganography based on SIFT and BOF’, Journal of Internet Technology, 18.

Wang, X.Y., Wang, X.Y., Ma, B., Li, Q. and Shi, Y.Q. (2021) ‘High precision error prediction algorithm based on ridge regression predictor for reversible data hiding’, IEEE Signal Processing Letters.

Wu, K.C. and Wang, C.M. (2014) ‘Steganography using reversible texture synthesis’, IEEE Transactions on Image Processing, 24(1), pp. 130–139.

Yuan, C., Xia, Z. and Sun, X. (2017) ‘Coverless image steganography based on SIFT and BOF’, Journal of Internet Technology, 18.

Zhou, Z., Cao, Y. and Sun, X. (2016) ‘Coverless information hiding based on bag-of-words model of image’, Journal of Applied Science, 34(5), pp. 527–536.

Zhou, Z., Cao, Y., Sun, X. and Gao, C. (2018) ‘Coverless information hiding based on the molecular structure images of material’, Computational Materials Continua, 54(2), pp. 197–207.

Zhou, Z., Sun, H., Harit, R., Chen, X. and Sun, X. (2015) 'Coverless image steganography without embedding'.

Zhou, Z., Wu, Q.M.J., Yang, C.N., Sun, X. and Pan, Z. (2017) 'Coverless image steganography using histograms of oriented gradients-based hashing algorithm', Journal of Internet Technology, 18(5), pp. 1177–1184.