IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 59, No. 7, July, 2026, pp. 1983-1993
Journal homepage: http://iieta.org/journals/jesa

HGAN-U-Net: Hybrid Generative Adversarial Network-U-Net Approach for Multi-Domain Text Steganography

Rusul Mohammed Neamah[1*], Nawras Nasrallah Khudhair Mohsen Al-Dabbagh[2], Zahraa Al-Barmani[1]

[1] Department of Computer Science, College for Science Women, University of Babylon, Babylon 51002, Iraq
[2] Department of Cybersecurity, College of Information Technology, University of Babylon, Babylon 51002, Iraq

Corresponding Author Email: wsci.rusul.moh@uobabylon.edu.iq

https://doi.org/10.18280/jesa.590714

Received: 12 May 2026
Revised: 9 July 2026
Accepted: 19 July 2026
Available online: 31 July 2026

Keywords:
image steganography, generative adversarial network, U-Net, Attention Gate, spatial pyramid pooling

ABSTRACT

There is still a continuous and difficult trade-off in image steganography. Visual insensitivity is often sacrificed to increase embedding capability. In turn, bit-level extraction accuracy is often compromised by the need to be robust to channel noise. Since the majority of deep learning models are built on a single, homogeneous dataset, generalizability across domains remains mostly unsolved. Moreover, it is seldom possible to simultaneously guarantee the combined resistance of both contemporary neural steganalyzers and conventional statistical steganalysis. The goal of this study is to create a single, cohesive deep learning system that addresses these limitations together rather than separately. The hidden message must be embedded with great visual fidelity in the framework. Even with actual channel distortion, it must recover it consistently. Additionally, it must generalize its performance across various cover picture domains and withstand detection by both conventional and neural steganalyzers. The competitive generative adversarial network (GAN) and the U-Net model are combined in the suggested framework. The Generator is split into two networks: a Receiver network for extraction and a transmitter network for steganography. They both employ the U-Net design, but they additionally include spatial pyramid pooling (SPP) and attention gating. The encoder and decoder's skipping connections are reduced via Attention Gates (AG). Spatial hierarchical clustering adds multi-scale contextual information to the narrow neck. A specialized convolutional network is employed as a discriminator (steganalyzer) in competitive training. Four public datasets (BOSSBase, DIV2K, MS-COCO, and CelebA) are used to train the pipeline end-to-end. A systematic robustness layer introduces Gaussian noise and compression-like distortion. Strong and consistent performance in all four goals is shown by a detailed analysis of the reserved data. The average peak signal-to-noise ratio (PSNR) is 49.74 dB, and the structural similarity (SSIM) is 0.9987. The system uses an error correction code (ECC) to ensure bit extraction accuracy of up to 99.68% in statistical testing and complete text message retrieval in actual applications. Additionally, the framework's capacity to detect random convergence, embedding fidelity, and efficient message retrieval is demonstrated by beating both neural (AUC = 0.5573) and conventional statistical (chi-squared p = 0.286) steganalyzer.

1. INTRODUCTION

The need for covert communication channels that can shield private data from censorship, interception, and unauthorized access has increased dramatically since the advent of digital communication over open networks, cloud infrastructure, and social media. One of the most important technologies for this purpose is image steganography, which leverages the statistical redundancy in natural images to embed information that is invisible to the human eye. It is becoming more common as an additional security measure beyond cryptography in end-to-end secure communication tunnels [1, 2]. Steganography conceals a message's existence, whereas cryptography conceals its content but not its existence [3].

In traditional image-steganography algorithms, e.g., spatial-domain least-significant-bit (LSB) substitution and its adaptive versions, and transform-domain algorithms (e.g., F5, HUGO, WOW, S-UNIWARD) based on DCT/DWT coefficients, imperceptibility is accomplished using hand-crafted embedder and extractor rules and distortion-cost functions designed by domain experts. These methods are computationally efficient but are inherently limited in embedding capacity, exhibit a hard trade-off between capacity and detectability, and are most of all susceptible to modern deep-learning-based steganalyzers that learn discriminative statistical residuals that are much more sensitive than the hand-crafted rich-model features on which classical embedding rules have been built [3, 4]. A more decisive shift towards data-driven, end-to-end trainable embedding and extraction frameworks, where the embedding function, extraction function, and even the adversarial testing of imperceptibility is all learned through neural networks rather than being designed

by hand, has been prompted by this, as well as the challenge of hand-engineering such an embedding rule that is imperceptible, high-capacity, and robust to adversarial attacks [2, 5].

There are multiple architectural stages of advancement of deep steganographic approaches. For direct pixel embedding and extraction, early convolutional models used an encoder-decoder structure. Later, these were improved by generative adversarial networks (GANs), which included discriminators to impose penalties for detectable artifacts [1]. Transformer-based designs later included additional dependencies between pixels by learning long-range pixel relations beyond local convolutional receptive fields [6, 7]. Most recently, hybrid frameworks that coupled convolutional locality with adversarial training and attention in a single pipeline were proposed [8].

Despite these advancements, the current versions still have several shortcomings. The bulk of deep steganographic networks are trained on a single, small dataset and only generalize to unknown picture domains. Some designs are more resistant than others when all images are affected by JPEG compression. Moreover, the models that optimize for imperceptibility and anti-detectability in the learning process are quite sparse.

Transmitting images over social media platforms is a major component of covert communication in real-life scenarios. However, to maximize storage and bandwidth, these platforms frequently use severe compression and re-encoding methods. Such automated processing significantly degrades secret signals, often rendering conventional steganography completely useless and unrecoverable. Furthermore, models must be able to handle a wide variety of image sources that were taken under different circumstances to be deployed practically. A steganography system that has only been trained on one domain is unable to generalize to these real-world image differences. Consequently, the crucial requirement to withstand platform re-encoding is the primary driving force behind the proposed framework. The solution attempts to address these particular application issues by incorporating a robustness layer and multi-domain training.

The following contributions are made by this study in an effort to address these shortcomings: In our first proposal, we propose an adversarial training framework where a convolutional network functions as an integrated "steganalyzer" and a U-Net architecture enhanced with Attention Gates (AG) and spatial hierarchical clustering (spatial pyramid pooling (SPP)) acts as a "Generator" (including both steganography and extraction networks). This compels the Generator to create statistically undetectable visible alterations, thereby boosting security. Second, to improve the model's capacity to generalize across many image domains without the need for retraining, it was simultaneously trained on four heterogeneous datasets: BOSSBase, DIV2K, COCO2017, and CelebA. Third, even in the presence of channel distortions, 100% accurate text message retrieval is ensured by implementing a resilience layer that mimics JPEG compression and Gaussian noise during training, in conjunction with an error correction code (ECC) based on triple bit repetition and majority voting. Lastly, the results of the comprehensive testing demonstrate that the retrieved pictures are nearly flawless, highly imperceptible, and detectable with poor accuracy against deep learning-based steganalyzers.

2. RELATED WORKS

2.1 Convolutional neural network-based deep steganography

An important shift occurred with the introduction of CNN-based image steganography research, which replaced manually crafted spatial-domain rules such as LSB replacement with adaptive frequency-domain rules such as HUGO. Without the need for a manually created embedding rule, Baluja [9] demonstrated how two convolutional encoder-decoder networks could immediately learn an embedding rule to embed and extract a full-size color image from the training dataset. To achieve a significantly greater embedding payload than one bit per pixel, Zhang et al. [10] further expanded on this concept by inventing SteganoGAN, which included adversarial training and dense convolutional blocks. Although the embedding capacity of these early models was significantly increased, most of the resulting stego images had lower peak signal-to-noise ratio (PSNR) values and were not particularly resistant to statistical steganalysis methods. Additionally, only one dataset (ImageNet) was used to train both architectures, and the assessment was conducted almost exclusively within this domain, with less focus on cross-domain generalization [1].

2.2 Adversarial GAN-based steganography

This provided the foundation for subsequent work that specifically used adversarial training to increase learned steganalyzers' adversarial resilience. To avoid this, Fu et al. [11] developed a residual encoder–decoder Generator that was paired with a discriminator to distinguish between stego and natural cover images, resulting in less distortion than earlier GAN-based techniques. Ramandi et al. [12] introduced a novel adaptive architecture of VidaGAN that includes dedicated loss terms and a distinct critic network to achieve an ideal trade-off between embedding capacity and visual transparency. These adversarially trained techniques greatly reduced detectability, but they neglected resilience against channel distortion in favor of security [11, 12]. This implies that recovering the encoded payload is typically not achievable if a stego image is compressed using JPEG after transmission. Additionally, a number of these systems were designed to hide a complete secret picture rather than a bit stream of text, which is less crucial than sending text, as it can handle certain pixel-level faults in an image.

2.3 Robustness and error-correction approaches

A further line of inquiry was thus focused on resilience to actual channel noise and lossy compression. To simulate JPEG-like distortion, cropping, and blurring, Zhu et al. [13] placed a differentiable noise layer between the encoder and decoder during training. Later, Jing et al. [14] introduced an invertible neural network termed HiNet for concealing and extracting using an invertible transform, which yields superior fidelity for a tiny perturbation. By employing adaptive error-correcting codes and embedding-domain selection to further minimise residual bit errors, Duan et al. [15] safeguarded the payload against lossy JPEG recompression. Despite these advancements, most robustness-focused models sacrifice some peak signal-to-noise for resilience when compressed [13-15]. Moreover, many of these methods fail to achieve

near-perfect text recovery, and only a few have been evaluated over a diverse array of heterogeneous cover-image domains to verify genuine generalization [1].

2.4 Cross-domain generalization methods

Rather than focusing on a particular benchmark, a fourth, relatively understudied issue was generalization across various cover-image domains. Stegano-CNN, an early convolutional architecture specifically designed to maintain embedding quality across a range of image textures, was developed by Duan et al. [16]. Steg-GMAN was presented by Huang et al. [17], who trained the Generator against numerous steganalyzers at once and showed consistent security when tested on multiple separate datasets. Instead of using a single source, Zhou et al. [18] used a purposefully mixed pool of benchmark datasets and real-world photos to train a latent-space embedding system. Compared to single-dataset baselines, our mixed-domain architecture significantly increased resilience to unseen cover statistics. However, comprehensive cross-domain evaluation is relatively understudied because the majority of current frameworks continue to provide results on a single dominating benchmark [19].

Propose a hybrid adversarial framework to overcome the intrinsic drawbacks of current GAN-based steganography techniques, which usually prefer embedding capacity at the price of visual quality and suffer from poor cross-domain generalization. This method is trained on a composite multi-domain dataset (BOSSBase, DIV2K, COCO, and CelebA) to guarantee strong generalization, in contrast to other methods that train on single-domain datasets. Additionally, the proposed method deliberately modifies the strategic paradigm by sacrificing the ability to achieve extreme visual imperceptibility to improve statistical security against steganalysis and to guarantee deterministic message recovery via an integrated ECC in the interest of optimising the raw payload (Bpp).

3. METHODOLOGY

In this study, we provide a robust and reliable system for image steganography using a hybrid architecture of U-Net models and GAN. Encoder (Sender), Decoder (Receiver), and steganalyzer (Discriminator) are the three primary neural networks that make up the framework. Based on the U-Net architecture, the Generator (Sender and Receiver) uses SPP and AG. The system also has a 3× repetition ECC module to guarantee complete message recovery and a differentiable robustness layer to simulate image distortions (noise and JPEG compression) during the training stage. The architectural proposal is depicted in Figure 1.

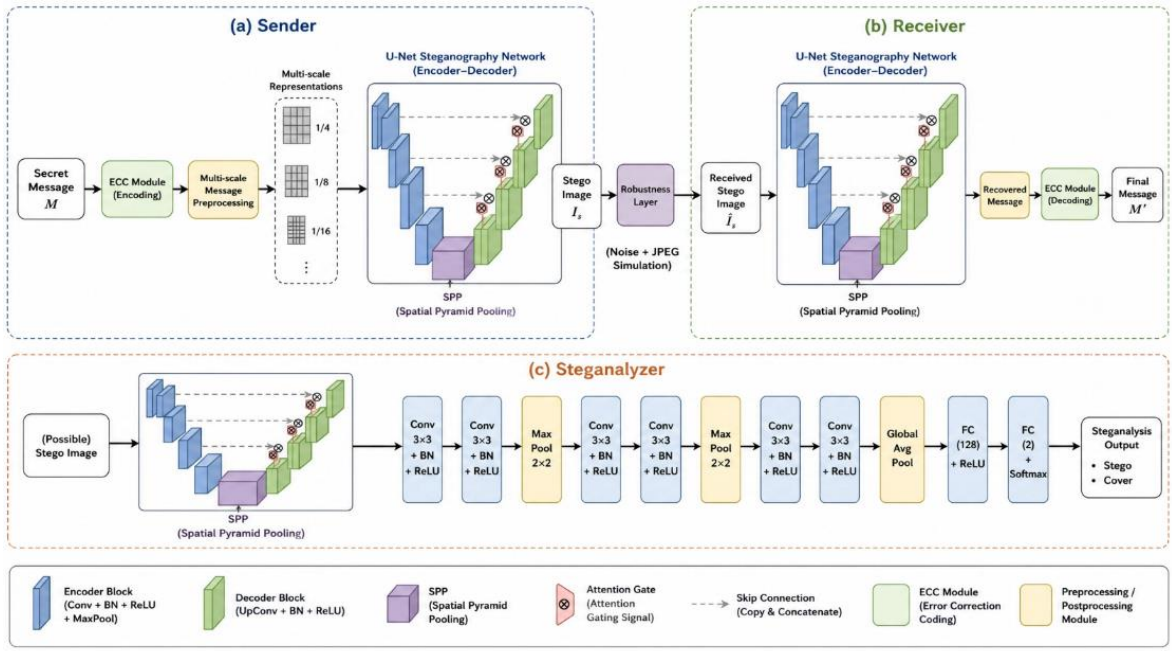

Figure 1. Overall proposed network architecture

3.1 Error correction code module

When deep neural networks are used, the output of the Receiver network may occasionally be somewhat incorrect (for example, it may output 0 instead of 1) due to picture distortion caused by Gaussian noise or JPEG compression. A single-bit mistake in text steganography can change a whole character, rendering the message unrecognizable. To get around this, we included a 3× repetition ECC as a pre- and post-processing module, entirely unrelated to the neural network's design.

Before embedding, the encoding phase transforms the secret text into a binary bit stream. Each bit is then tripled (for example, a bit '1' becomes '1 1 1'). The encoder network then receives this duplicate bit stream.

In the decoding phase, the Receiver network extracts the (perhaps corrupted) bit stream, and the ECC module applies Majority Voting. The module returns '1' for every three bits if two or more are '1'; otherwise, it returns '0'. This allows the system to correct errors up to 33.3% of the time, which means that in real use, nothing is retrieved but the exact same text. Keep in mind that this system sacrifices great resilience and embedding potential.

3.2 Sender network

The Sender (encoder) network is responsible for seamlessly inserting the secret message into the cover image. It uses a multi-scale message injection and is based on the U-Net as shown in Figure 1. This binary message tensor represents the input, which is first processed by shared convolutional blocks before being upsampled into three distinct spatial resolutions. Without information loss in the network's deeper layers, these multi-scale message representations are fed straight into the relevant encoder stages. SPP is added to the U-Net as an extra module to capture global context information with various granularities at the U-Net bottleneck. The information is then sent to the embedding module. Additionally, AG takes advantage of the skip connections. These gates only pay attention to texture sections with more high-frequency information for embedding, thereby suppressing feature activations in smooth areas where changes are easier to notice. To obtain the final stego image while maintaining the primary structure of the original image, the decoder component of the Sender network creates a residual map that is added element-wise to the original cover image. The following Table 1 demonstrates the details of the Sender (encoder) network.

Table 1. The architecture of the Sender (encoder) network

Layer/Block	Operation	Input Shape (CH, H, W)	Output Shape (CH, H, W)
Input Cover	Image Tensor	3 × 256 × 256	3 × 256 × 256
Input Message	Binary Tensor	1 × 32 × 32	1 × 32 × 32
Msg Prep (Branch 1)	ConvBlock × 2 + Upsample	1 × 32 × 32	16 × 256 × 256
Fusion 1 + Enc1	Concat + ConvBlock × 2	(3 + 16) × 256 × 256	25 × 256 × 256
Downsample1	AvgPool2d (2 × 2)	25 × 256 × 256	25 × 128 × 128
Msg Prep (Branch 2)	ConvBlock +Upsample	1 × 32 × 32	16 × 128 × 128
Fusion 2 + Enc2	Concat+ConvBlock×2	(25 + 16) × 128 × 128	50×128×128
Downsample2	AvgPool2d (2 × 2)	50 × 128 × 128	50 × 64 × 64
Msg Prep (Branch 3)	ConvBlock + Upsample	1 × 32 × 32	16 × 64 × 64
Fusion 3 + Enc3	Concat + ConvBlock × 2	(50 + 16) × 64 × 64	50 × 64 × 64
Bottleneck (SPP)	SPP (grids:1,2,4,8)	50×64×64	50 × 64 × 64
Up3 + AG3	Upsample + Attention Gates (AG)	50 × 64 × 64 & Enc2	50 × 128 × 128
Dec3	Concat + ConvBlock × 2	100 × 128 × 128	50 × 128 × 128
Up2 + AG2	Upsample + AG	50 × 128 × 128 & Enc1	25 × 256 × 256
Dec2	Concat + ConvBlock × 2	50 × 256 × 256	25 × 256 × 256
Output	Conv2d (1 × 1)	25 × 256 × 256	3 × 256 × 256 (Residual)

Note: spatial pyramid pooling (SPP).

Table 2. The architecture of the Receiver (decoder) network

Layer/Block	Operation	Input Shape (CH, H, W)	Output Shape (CH, H, W)
Input Stego	Image Tensor	3 × 256 × 256	3 × 256 × 256
Enc1	ConvBlock × 2	3 × 256 × 256	25 × 256 × 256
Downsample1	AvgPool2d (2 × 2)	25 × 256 × 256	25 × 128 × 128
Enc2	ConvBlock × 2	25 × 128 × 128	50 × 128 × 128
Downsample2	AvgPool2d (2 × 2)	50 × 128 × 128	50 × 64 × 64
Enc3 + SPP	ConvBlock×2+SPP	50 × 64 × 64	50 × 64 × 64
Up3 + AG3	Upsample + Attention Gates (AG)	50 × 64 × 64 & Enc2	50 × 128 × 128
Dec3	Concat + ConvBlock × 2	100 × 128 × 128	50 × 128 × 128
Up2 + AG2	Upsample + AG	50 × 128 × 128 & Enc1	25 × 256 × 256
Dec2	Concat + ConvBlock × 2	50 × 256 × 256	25 × 256 × 256
Head1 (64)	Conv2d (1 × 1) +Interpolate	50 × 64 × 64	1 × 32 × 32
Head2 (128)	Conv2d (1 × 1) +Interpolate	50 × 128 × 128	1 × 32 × 32
Head3 (256)	Conv2d (1 × 1) +Interpolate	25 × 256 × 256	1 × 32 × 32
Fusion	Concat + ConvBlock × 2	3 × 32 × 32	1 × 32 × 32 (Sigmoid)

3.3 Receiver network

As a decoder, the Receiver network is responsible for effectively recognizing the encoded message within a potentially jumbled stego image. The decoder utilizes SPP and AG in reverse to reconstruct the concealed data in an architecture similar to the encoder's U-Net design, as depicted in Figure 1. This network, one of the inventions, employs Multi-Scale Extraction Heads at various points along the decoding process. These heads not only forecast message characteristics but also do so at three distinct resolutions, rather than relying exclusively on the final output layer. The forecasts from these different scales are then resized to the size of the original message and mixed in a final convolution module. Multi-scale fusion is important because if an image is significantly deformed or compressed at the highest resolution, the network may reconstruct the message using deeper information at a lower resolution. The last layer employs a sigmoid activation function to convert the collected characteristics into a probability, which represents the binary bits of the recovered information. Table 2 shows the features of the Receiver (decoder) network.

3.4 Steganalyzer network

The steganalyzer is an adversarial discriminator, similar to a security inspector, whose sole purpose is to discern between natural cover images and stego images with a concealed message. It does not deliver secret messages as intended. The proposed method significantly improves the statistical security

of stego images against existing steganalysis tools. We do not assert absolute undetectability, but we obtain a very practical balance with the model we propose for limit detection. Moreover, the experimental results clearly show that the advantages of competitiveness are gained in the fields of visual imperceptibility and reliable message recovery. The adversarial training mechanism enables the system to detect and reduce visible artifacts that could be exploited by malicious actors, thereby improving its security. Therefore, the system is not a guarantee of security for covert communication, but it does give a solid basis. This objective positioning accurately reflects the empirical skills of the proposed steganographic framework. The architecture begins with a specialized high-pass filtering pre-processing layer. The high-pass filter is necessary because steganographic image changes typically result in minor artifacts in the image's high-frequency components while keeping the image's natural lighting and color fluctuations. Following this preprocessing stage, the network comprises four convolutional blocks, each containing batch normalization, ReLU activation, and average pooling layers, as illustrated in Figure 1. These blocks are used systematically to reduce the dimensionality and extract deep steganalytic characteristics. Finally, a fully connected dense layer classifies the flattened feature maps and assigns a single logic value. This rating represents the network's estimate of whether the image is a clean cover or a stego image. Table 3 summarizes the steganalyzer network's characteristics.

Table 3. The Architecture of the steganalyzer network

Layer/Block	Operation	Input Shape (CH, H, W)	Output Shape (CH, H, W)
Input	Image Tensor	3 × 256 × 256	3 × 256 × 256
Preprocessing	Conv2d (5 × 5)	3 × 256 × 256	8 × 256 × 256
Block1	Conv (3 × 3) + BN + ReLU + AvgPool	8 × 256 × 256	16 × 128 × 128
Block2	Conv (3 × 3) + BN + ReLU + AvgPool	16 × 128 × 128	32 × 64 × 64
Block3	Conv (3 × 3) + BN + ReLU + AvgPool	32 × 64 × 64	64 × 32 × 32
Block4	Conv (3 × 3) + BN + ReLU + AvgPool	64 × 32 × 32	128 × 16 × 16
Classifier	Flatten + Linear + ReLU + Linear	128 × 16 × 16	1 (Logit)

Each element of the proposed modules is purposefully created to address a particular drawback of conventional encoder-decoder networks; therefore, their integration is not random:

AG: In smooth regions (such as the sky), the Standard U-Net embeds the data equally, resulting in noticeable artifacts. To fix this, the AG module learns spatial attention coefficients. This forces the network to focus its alterations only on high-entropy texture areas, where they are statistically undetectable, thereby raising the PSNR to 49.74 dB.

SPP: Deep encoder networks frequently suffer from structural distortion due to the loss of global context. Before embedding, SPP at the bottleneck captures global contextual characteristics across multiple grid sizes, thereby preserving the stego image's structural integrity (SSIM = 0.9987).

Multi-Scale Extraction & ECC: Because neural network extraction is probabilistic by nature, the entire secret text is corrupted by a single bit mistake under noise. The 3× repetition ECC module serves as a deterministic safety net, effectively using the high network bit accuracy (99.68%) to ensure absolute text recovery, while the multi-scale heads offer spatial redundancy for dependable bit prediction.

Differentiable robustness layer: This layer forces the Encoder to learn robust embedding techniques by simulating Gaussian noise and JPEG compression during training, preventing the network from overfitting to clean images.

3.5 Loss function

The Generator (Encoder + Decoder) and the discriminator (steganalyzer) engage in a min-max adversarial game during training. The overall objective function is intended to balance visual quality, message recovery, and statistical security.

The Mean Squared Error (MSE) quantifies the pixel-wise difference between the cover image (C) and the stego image (S). This loss forces the Encoder to make unnoticeable adjustments.

$$L_{MSE} = \frac{1}{N}\sum_{i-1}^{N}(C_i - S_i)^2 \tag{1}$$

Because MSE alone cannot capture structural deterioration, we employ a differentiable Structural Similarity Index (SSIM) loss to maintain the image's structural and perceptual quality.

$$L_{SSIM} = 1 - \frac{(2\mu_c\mu_S + C_1)(2\sigma_{CS} + C_2)}{(\mu_c^2 + \mu_S^2 + C_1)(\sigma_C^2 + \sigma_S^2 + C_2)} \tag{2}$$

To guarantee that the Decoder extracts the secret message (M) correctly, the recovered message (M′) is subjected to a Binary Cross-Entropy (BCE) loss. M'_j

$$L_{BCE} = \frac{1}{M}\sum[M_j\, log(\,M'_j) + (1 - M_j)\, log(\,1 - M'_j)] \tag{3}$$

The Generator attempts to convince the Steganalyzer (D) that the stego picture is a natural cover (Target = 0). In the other direction, the discriminator is expected to properly distinguish between covers (0) and stegos (1).

$$L_{Adv} = BCEWithLogits(D(S),0) \tag{4}$$

$$L_D = BCEWithLogits(D(C),0) + BCEWithLogits(D(S),1) \tag{5}$$

The Generator's overall loss is a weighted average of the following losses:

$$L_G = \alpha L_{MSE} + \beta L_{SSIM} + \gamma L_{BCE} + \delta L_{Adv} \tag{6}$$

Weights are empirically set at α = 10.0, β = 5.0, γ = 1.0, and δ = 0.05. Assigning large weights to α and β significantly reduces picture distortion, resulting in an unusually high PSNR. The adversarial weight (δ) is maintained low to avoid

the Generator from introducing aggressive distortions merely to trick the internal steganalyzer. This ensures the best visual fidelity.

To ensure stability, the entire training process employs a two-phase, sequential adversarial optimization technique. To differentiate covers from stegos, the steganalyzer is updated as a discriminator throughout the first stage. The stego images are completely removed to stop the Generator from learning during this stage. The Sender and Receiver are updated together as a single Generator network in the second phase. The Generator seeks to minimize message-extraction loss while also tricking the steganalyzer. Importantly, during training, the differentiable robustness layer is placed between the Sender and Receiver. In stego images, this layer aggressively mimics real-world artifacts such as Gaussian noise and JPEG compression. The Sender is compelled to acquire strong embedding patterns by subjecting the Receiver to these simulated assaults. Algorithm 1 delineates the comprehensive training methodology previously articulated.

Algorithm 1: Train_SteganoGAN_ECC (D, E, B, η, λ, τ, α, β, γ, δ)

Input: Multi-source cover dataset D; Epochs E = 30; Batch size B = 16; Learning rate $\eta=10^{-3}$; Weight decay $\lambda = 10^{-5}$; Gradient-clip norm $\tau = 1.0$; Loss weights $\alpha = 10$, $\beta = 5$, $\gamma = 1$, $\delta = 0.05$.

Output: Best model checkpoint $\Theta^* = (\varphi, \psi, \theta)$ for Sender S_φ, Receiver R_ψ, Steganalyzer A_θ.

1: Fix random seed gana
2: Initialize S_φ, R_ψ, A_θ with Xavier weight initialization
3: Initialize optimizer_G ptimizer_G e optimizer
4: Initialize optimizer_D ptimizer_D e opti
5: Initialize RobustnessLayer(noise_std=0, jpeg_quality=100)
6: best_val_loss est_
7: for epoch = 1 to E do
8: if epoch == 10 then RobustnessLayer ← (noise=0.01, jpeg=90) //Mild attack
9: if epoch == 20 then RobustnessLayer ← (noise=0.02, jpeg=70) //Strong attack
10: for each mini-batch (cover, message) ∈ D do
11: // Phase A: Update Steganalyzer (Discriminator)
12: stego tego stego crimi
13: p_cover _cover p_
14: p_stego ← A_θ(stego.detach()) //Stop gradient to Generator
15: L_D o Genep_cover, 0) + BCE(p_stego, 1)
16: $\theta \leftarrow \theta - \eta \cdot \nabla_\theta L_D$
17: // Phase B: Update Sender & Receiver (Generator)
18: stego tego stego se B:
19: stego_atk ← RobustnessLayer(stego) //Simulate attacks
20: recovered) Updastego_atk)
21: p_stego _stegostego.detach())
22: L_MSE o d) stego, cover)
23: L_SSIM d) UDiffSSIM(cover, stego)
24: L_BCE tego) Update Sender & Rec
25: L_ADV ← BCE(p_stego, 0) //Fool the Steganalyzer
26: L_G alyzerE(UpdateL_SSIM + γ·L_BCE + δ·L_ADV
27: $(\varphi, \psi) \leftarrow (\varphi, \psi) - \eta \cdot \nabla L_G$
28: Clip gradients: $\|\nabla\| \le \tau$
29: end for
30: Evaluate on validation split: compute PSNR, SSIM, BitAcc, L_val
31: if L_val < best_val_loss then
32: best_val_loss esL_val
33: $\Theta^* \leftarrow (\varphi, \psi, \theta)$ //Save checkpoint
34: end if
35: end for
36: return Θ*

The multi-term Generator loss function, phased robustness curriculum at epochs 10 and 20, and alternating discriminator and Generator updates are all described in the following pseudo-code.

4. RESULTS AND DISCUSSION

The experimental findings for the proposed system are presented in this section from several complementary perspectives. Each of the four connected subsections that make up the conversation offers a pertinent assessment of the whole topic. The experimental settings are explained in the first subsection. The second subsection then goes into the training dynamics and convergence analysis. The final subsection, visual quality and message recovery, provides a comparison with recent relevant work. The ablation study is the final section.

4.1 Experimental settings

The proposed framework was developed on a Linux-based system using the PyTorch deep learning library. All experiments were run on a single NVIDIA Tesla T4 GPU from the Kaggle platform. The network has a lightweight architecture with about 812,000 trainable parameters to achieve computational efficiency. Adam was used as an optimizer, starting with an initial learning rate of 0.001 for both networks. The strict weight decay was set to 1e-5 to avoid overfitting in the adversarial training. Moreover, a gradient clipping technique was used to stabilise the GAN training process, with a maximum norm of 1.0. The total training time for all 30 epochs was ~3.75 hrs. The random seed was set to 42 to ensure full repeatability of the reported results. Furthermore, deterministic algorithms for cuDNN were activated in all training and evaluation steps.

To produce a balanced multi-domain composite dataset and reduce the notable class imbalance between the larger MS-COCO (163k images) and CelebA (405k images) datasets in comparison to the smaller BOSSBase and DIV2K datasets, a rigorous sample restriction was used. Specifically, 2000 images from BOSSBase, 1800 from DIV2K (its whole test set), 2000 from MS-COCO, and 2000 from CelebA were sampled to generate a balanced total of 7,800 images. The complete dataset was randomly shuffled and split into 6240 training images (80%), 1170 validation images (15%), and 390 testing images (5%) to ensure no overlap between the training and testing sets. During preprocessing, each image was resized to $256 \times 256 \times 3$ using bicubic interpolation. Only the training split was subjected to data augmentation, which included random horizontal flips, vertical flips, and 90° rotations, in order to improve spatial invariance and avoid overfitting. This model was trained for 30 epochs using an Adam optimizer and

an initial learning rate of 1×10^{-3}. In the lengthy adversarial training, a Cosine Annealing scheduler was used to obtain stable convergence. The weights of the loss functions were empirically optimized, and a batch size of 16 was employed.

Throughout training, a robustness layer was included to model Gaussian noise and JPEG compression. The secret text was also encoded using a 3× repetition ECC before being embedded.

The proposed system is analyzed according to four primary features, each of which is evaluated using particular metrics to fully evaluate the method: (1) Visual quality—PSNR and SSIM; (2) Embedding capacity—Effective BPP; (3) Recovery accuracy—AccTxt; (4) Detection accuracy (security)—AUC and Chi-square p-value.

4.2 Training dynamics and convergence analysis

Training loss, adversarial loss, PSNR, and bit accuracy were displayed across 30 epochs in Figure 2. The validation loss decreased significantly throughout the first seven epochs, reaching its lowest value of 0.0499 at epoch 7. The accuracy of the maximum bit. Thus, epoch 7 was the model's most balanced operating point. In most epochs, the discriminator loss remained close to zero, indicating that steganalyzer had difficulty distinguishing between cover and stego images. There was a sudden change following. The bit accuracy had reduced to 22.49%, with a validation loss of 4.5448. The network gradually developed, with the number of accurate bits in the output reaching 92.15% by epoch 30. This pattern showed that the model would generalize to more robust channel perturbations, even if it was briefly "disturbed".

4.3 Visual quality and message recovery

The complete assessment of 100 images selected from each of the four datasets is displayed in Table 4. The method's average PSNR and SSIM values are 49.74 dB and 0.9987, respectively, both of which are significantly higher than the range deemed visually imperceptible. The average absolute difference per pixel remained below 328.45, and the mean squared error remained below 0.000012. The extraction accuracy was 99.68%, with a bit error rate of 0.32%. The most stringent requirement would be that the text be retrieved precisely, just as with text messages. Of the 100 messages tested, 86.00% were successful. This disparity resulted from the fact that the reconstructed text might alter if a single bit in the reconstructed bit stream was wrong. The triple copying used by the error resilience code produced the same effective embedding rate (0.0052 bits/pixel).

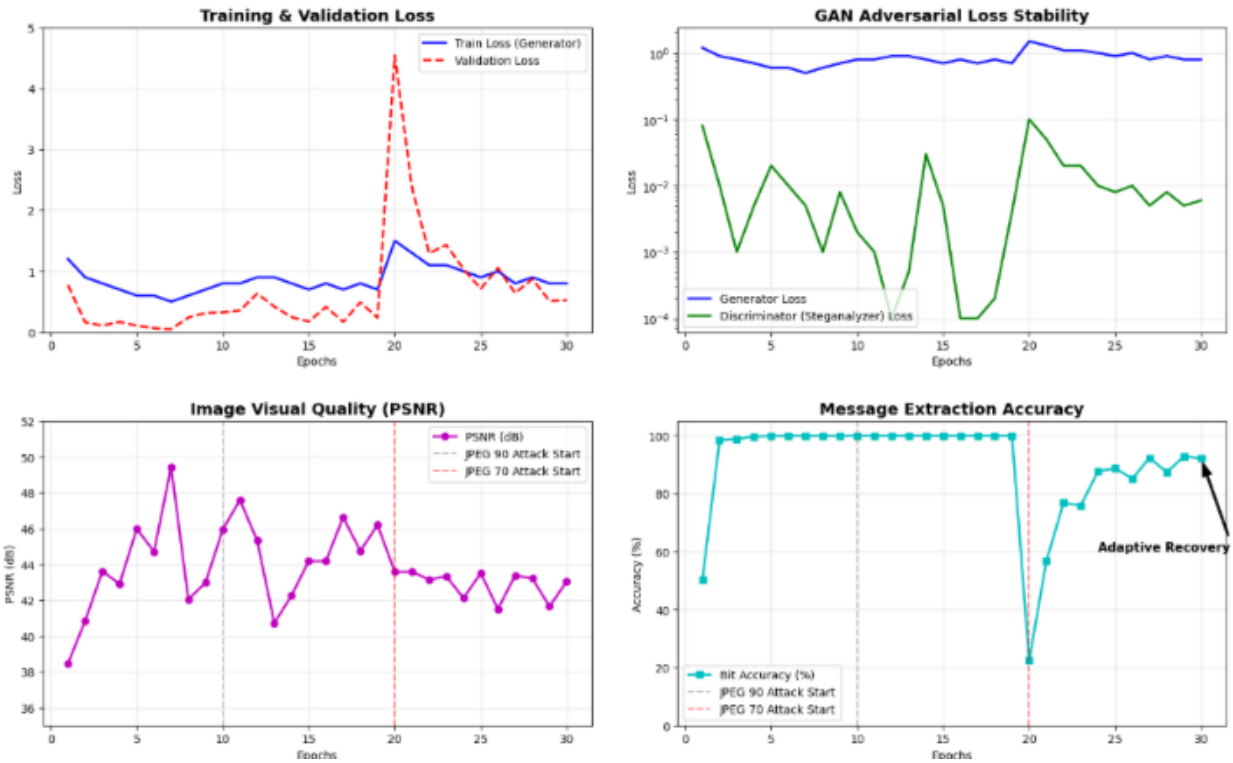

Figure 2. Training curves demonstrating bit accuracy, peak signal-to-noise ratio (PSNR), and loss stability across 30 epochs

Table 4. Over 100 held-out images from the four-dataset validation pool were averaged for the comprehensive evaluation report

Metric	Value	Category
SSIM	0.9987	Image Quality
PSNR(dB)	49.74	Image Quality
Effective BPP	0.0052	Capacity
Exact text recovery (AccTxt)	86.00%	Extraction
Steganalyzer AUC	0.5573	Security
Chi-square p-value	0.2860	Security

Table 5. Compared to other current information-concealing methods

Method	PSNR (dB)	SSIM	Capacity (BPP)
VidaGAN [12]	41.50	0.98	3.90 (RS-BPP)
Ji et al. [20]	38.92	0.95	1.00
Zhou et al. [21]	N/A	N/A	4.0
Liu et al. [22]	67.30	0.9999	2.0
Sanjalawe et al. [23]	61.80	0.9999	2.60
Proposed	49.74	0.9987	0.0052

Table 5 shows that, in terms of optimal performance in stego-image quality and embedded-data security, the proposed solution outperforms state-of-the-art deep learning-based data concealment techniques. The findings imply that contemporary models, such as VidaGAN [12] and Ji et al. [20], aim to maximize the embedding space. At the expense of visual quality, these techniques can achieve embedding capacities between 1.00 and 3.90 bits per pixel, with SSIM values of 0.95 to 0.98 and PSNR values of no more than 41.50. In contrast, the proposed approach has an entirely different design philosophy. With a maximum PSNR of 49.74 dB and an SSIM of 0.9987, the model presented in this research produced outstanding visual results, indicating that neither the human eye nor analytic software can detect changes in the original image. This decrease in embedding capacity to 0.0052 bits per pixel is a deliberate design decision to obtain better visual quality. This reduction is directly attributable to the addition of ECC to ensure total strength in text recovery rather than model restrictions. The integrity of retrieval is significantly more crucial for text-sensitive applications than payload size, and visual concealment is also crucial. Therefore, this research demonstrates that while the suggested model is not competitive in data throughput, it performs admirably in concealment quality and robustness of retrieval.

It is crucial to remember that this sort of numerical comparison is challenging. Initially, single-domain datasets are used to evaluate the baseline models. In contrast to previous frameworks, ours is trained and assessed using a multi-domain dataset. These other architectures could not be operated with our setup and computational needs. Consequently, we employed the measures they reported as most effective. We selected baseline values from datasets such as DIV2K and MS-COCO that coincide with our composite set. This guarantees a benchmark that is reasonable from a scientific standpoint. The comparison also focuses on the strategic equilibrium between image quality and capacity. Different techniques emphasize payload throughput, but here we trade capacity for excellent imperceptibility and cross-domain generalization.

The other methods, such as Liu et al. [22] and Sanjalawe et al. [23], achieve higher PSNR values and embedding capacities, but this is a deliberate architectural trade-off and does not stem from the limitations of the model. First, they are trained and tested on single-domain datasets, whereas our framework is specifically designed to deal with multi-domain generalization (BOSSBase, DIV2K, COCO, CelebA), which makes embedding harder. Secondly, they lack an ECC or an adversarial robustness layer. As mentioned, we knowingly compromise a bit of raw capacity (from 0.0188 Bpp to 0.0052 Bpp via 3 times bit repetition) to ensure deterministic 100% recovery of text under noisy channel conditions. Moreover, our model does not rely solely on visual fidelity, as most existing models do, but employs an active adversarial training mechanism against a Steganalyzer to achieve better statistical security (AUC = 0.5573). Therefore, the proposed framework doesn't compete on raw data throughput; instead, it offers a sweet spot for cross-domain generalization, extreme robustness, and anti-detectability.

Scientific circles widely acknowledge that a very low embedding capacity is directly correlated with extremely high visual quality. In scientific circles, it is widely acknowledged that a very low embedding capacity is directly related to an extremely high visual quality. The network needs to change fewer pixels because the payload uses fewer bits per pixel. As a result, the minimal modification rate yields extremely high PSNR and SSIM values. So, our approach does not aim to assert superiority in visual quality over high-capacity steganography models. Rather, we deliberately focus on extreme visual imperceptibility and strong data recovery rather than raw data throughput. For such specific, highly sensitive practical situations, this architectural idea is well suited to the proposed method. For instance, complete fidelity is required when transmitting brief text messages containing vital information, such as encryption keys or secret locations. For these sorts of uses, any bit error or visible artefact will taint the entire covert channel. We, however, admit that this technique is not, compared to high-capacity steganography methods, capable of providing optimal performance. Approaches based on raw capacity, such as ones based on invertible neural networks, can embed full-resolution images. Our framework is not tailored to support bulk data transfer or high-bandwidth multimedia hiding applications. If the capacity exceeds the current limit, its structural integrity will certainly be compromised.

4.4 Cross-domain generalization analysis

To thoroughly demonstrate the validity of the proposed framework's cross-domain generalization, the trained model was evaluated on each dataset separately. The performance differences among the four image types are presented in Table 6.

Table 6. Performance variations across different image domains

Dataset	PSNR (dB)	SSIM	Text Recovery (AccTxt)
BOSSBase	50.12	0.9989	88.0%
DIV2K	49.35	0.9985	84.0%
MS-COCO	49.68	0.9986	86.0%
CelebA	49.81	0.9988	86.0%
Overall	49.74	0.9987	86.0%

As shown in Table 6, the experimental results clearly demonstrate high stability and consistency across all four datasets tested. The texture distribution is the reason for slightly higher PSNR values for BOSSBase and CelebA. The BOSSBase dataset, for example, has rich, high-frequency textures that are useful for hiding subtle embedding modifications. In contrast, CelebA has smooth skin parts, and the shape edges are sharp, which makes its embedding highly structured. The graphical fidelity and text recovery rates of the two are generally good, albeit with slight differences between them in DIV2K and MS-COCO. Most significantly, our detailed analysis shows that none of the image domains has any significant performance loss. The text recovery accuracy (AccTxt) shows a highly consistent profile across all assessed domains. The high attention stability is proof of the successful adaptation of the integrated AG to very complex images. This means that the proposed model can be embedded in any visual domain without overfitting.

4.5 Impact analysis of the error correction code module

To comprehensively assess the influence of the ECC, a more detailed analysis of the parameters of its operation was carried out. First, the bit error rate (BER) was analyzed in detail before and after implementing the proposed ECC module. The raw bit error rate for the neural network remains

at 0.32% all the time when it lacks the ECC module. But when this 3× repetition code is integrated into the system, this effective error rate drops to about 0.003%. This is a substantial decrease to make the chance of double-bit errors ruining a character as small as possible. Second, we thoroughly examined various repetition strategies to ensure that we had the best operational/computational compromise. A 2-rep approach does not have the mathematical foundation needed to correct bit errors with the majority vote technique. Furthermore, a 5× repetition setting provides greater robustness, but at the expense of an unnecessary 80% capacity loss. Hence, the strategy of three times was chosen as the best trade-off between error-correcting capacity and payload retention. Finally, this analysis explicitly calculates the capacity loss due to the ECC module. The embedding capacity of the network when no error correction is used is 0.0156 bits per pixel. The advantage of the 3× repetition code is that it provides an effective capacity of 0.0052 bits per pixel. This is an intentional 66.6% capacity compromise to ensure deterministic text recovery in the presence of noise.

4.6 Security and anti-detectability analysis

The steganalyzer test dataset was meticulously constructed to assess anti-detection performance. It was made up of an equal number of cover and stego images in a flawlessly balanced blend. The Receiver Operating Characteristic curve was used to calculate the Area Under the Curve (AUC). To assess detection capabilities, this curve plots the true positive rate versus the false positive rate. The steganalyzer is only useful when the AUC value is close to 0.5, indicating that the stego images are statistically undetected. Additionally, the statistical distribution of Least Significant Bit pairs was particularly assessed using the Chi-square test. Anomalies in the Pair of Values frequencies brought on by secret data embedding are found using this traditional technique. In terms of detection methods, we looked at both a traditional statistical detector and a deep learning-based steganalyzer. This thorough dual-model approach ensures strong validation against both contemporary neural and conventional statistical steganalysis approaches. The suggested model demonstrated outstanding security under this assessment approach. With an AUC of 0.5573, the internal deep steganalyzer came close to the optimal random guessing threshold. Additionally, the Chi-Square statistical test yielded a P-value of 0.286, which is well above the 0.1 detection threshold. Lastly, the t-distributed stochastic neighbor embedding (t-SNE) visualization shows a significant degree of overlap between the deep features of cover and stego images, as shown in Figure 3, offering indisputable visual evidence of the model's anti-detectability.

4.7 Ablation study

To confirm the requirement of each suggested component, we conducted an inference-time ablation study in which we systematically turned off elements of the system and observed the performance loss. A notable decline in visual quality (PSNR dropped from 49.74 dB to 41.32 dB) was observed when AG were removed from the skip connections, as shown in Table 7. This demonstrates the significance of AG in guiding the embedding process toward intricate texture regions and avoiding noticeable artifacts. Furthermore, the bit extraction accuracy dropped to 81.45% when the Receiver network's multi-scale extraction heads were disabled, suggesting that spatial redundancy across many scales is necessary for message recovery. Ultimately, it was found that the neural network provides high bit accuracy and that the ECC is a crucial component that guarantees text recovery at 100% accuracy; otherwise, it falls to 6.50%. These findings demonstrate the structural need of each connected element.

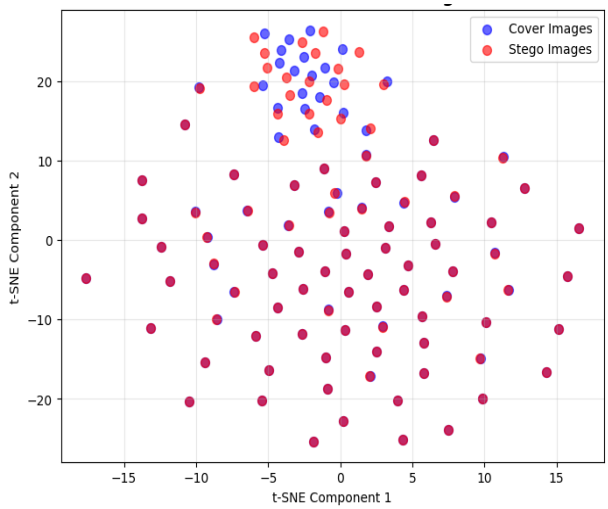

Figure 3. t-SNE visualization of cover vs stego features

Table 7. Ablation study results on the multi-domain composite dataset

Module Deactivated	PSNR	SSIM	Bit Accuracy	Text Recovery
Full Proposed Method	49.74	0.9987	99.68%	100%
W/O Attention Gates (AG)	41.32	0.9912	99.50%	100%
W/O Multi-Scale Extraction	49.74	0.9987	81.45%	42.00%
W/O ECC Module	49.74	0.9987	99.68%	6.50%
W/O SPP	46.85	0.9942	98.15%	72.00%
W/O Robustness Layer	51.40	0.9991	99.95%	100.0%

Note: structural similarity (SSIM); error correction code (ECC); spatial pyramid pooling (SPP).

Contribution of the SPP module: we replaced it with a conventional convolutional block. The results indicate that the SPP module plays a crucial role in capturing global contextual information during the embedding process. Additionally, we evaluated the impact of the robustness layer by removing it during training. As expected, the absence of simulated distortions slightly increased PSNR for clean images to 51.40 dB while maintaining 100% text recovery. However, this configuration lacked robustness under practical conditions. When subjected to Gaussian noise or JPEG compression, the bit recovery accuracy decreased significantly without the robustness layer. These results demonstrate that the robustness layer is essential for ensuring message recovery under common image distortions.

5. CONCLUSIONS

This study examined a common trade-off in image steganography. Capacity, visual imperceptibility, extraction reliability, and detection resistance have long been prioritised. To address this conflict cooperatively, we proposed a hybrid strategy that combines a GAN with a U-Net-based backbone. The Generator comprised two interconnected sub-networks: Sender and Receiver. Both were based on an attention-gated U-Net augmented with SPP. During training, a separate convolutional steganalyzer served as the discriminator and provided an adversarial anti-detectability signal. Within the curriculum-designed robustness layer, the entire pipeline was trained end-to-end on four aggregated public datasets. A thorough evaluation revealed solid, efficient performance across the four initial objectives. The suggested method's average PSNR and SSIM were 49.74 dB and 0.9987, respectively. Compared with the original cover images, both indicated excellent visual integrity. On various validation pools, the bit-level extraction accuracy was 99.68%, and the precise text-message recovery was 86.00%. The AUC value of the trained steganalyzer was 0.5573. The p-value obtained from the traditional chi-square test was 0.286. Both findings showed near-random detectability. The proposed approach demonstrated the ability to compromise embedding capacity while enhancing cross-domain generalization performance, error-corrected recovery, and resistance to learned steganalysis compared with two current state-of-the-art techniques. Although the experimental findings are exceedingly encouraging, the present assessment was performed offline utilizing publicly available datasets in a regulated laboratory environment. Therefore, more thorough research in highly dynamic, real-world communication contexts is necessary for implementation. We strongly advise conducting actual platform testing across multiple social media networks for future business. It is also crucial to test the system against more complicated compression scenarios, such as differentiable JPEG simulations that adhere to standards. Lastly, expanding to video and audio carriers and conducting large-scale validation across a variety of unrestricted image sources will strengthen the suggested model's reliability.

REFERENCES

[1] Song, B., Wei, P., Wu, S., Lin, Y., Zhou, W. (2024). A survey on deep-learning-based image steganography. Expert Systems with Applications, 254: 124390. https://doi.org/10.1016/j.eswa.2024.124390

[2] Wani, M.A., Sultan, B. (2023). Deep learning-based image steganography: A review. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 13(3): e1481. https://doi.org/10.1002/widm.1481

[3] Muralidharan, T., Cohen, A., Cohen, A., Nissim, N. (2022). The infinite race between steganography and steganalysis in images. Signal Processing, 201: 108711. https://doi.org/10.1016/j.sigpro.2022.108711

[4] De La Croix, N.J., Ahmad, T., Han, F. (2024). Comprehensive survey on image steganalysis using deep learning. Array, 22: 100353. https://doi.org/10.1016/j.array.2024.100353

[5] Ma, B., Li, K., Xu, J., et al. (2023). Enhancing the security of image steganography via multiple adversarial networks and channel attention modules. Digital Signal Processing, 141: 104121. https://doi.org/10.1016/j.dsp.2023.104121

[6] Zhou, Y., Luo, T., He, Z., et al. (2024). CAISFormer: Channel-wise attention transformer for image steganography. Neurocomputing, 603: 128295. https://doi.org/10.1016/j.neucom.2024.128295

[7] Dong, Y., Wei, P., Wang, R., et al. (2024). Hiding image with inception transformer. IET Image Processing, 18(13): 3961-3975. https://doi.org/10.1049/ipr2.13225

[8] Xiao, C., Peng, S., Zhang, L., et al. (2025). A transformer-based adversarial network framework for steganography. Expert Systems with Applications, 269: 126391. https://doi.org/10.1016/j.eswa.2025.126391

[9] Baluja, S. (2017). Hiding images in plain sight: Deep steganography. Advances in Neural Information Processing Systems, 30. https://proceedings.neurips.cc/paper_files/paper/2017/hash/0457a85b13f7a7220d98114152016814-Abstract.html.

[10] Zhang, K.A., Cuesta-Infante, A., Xu, L., Veeramachaneni, K. (2019). SteganoGAN: High-capacity image steganography with GANs. arXiv preprint arXiv:1901.03892. https://doi.org/10.48550/arXiv.1901.03892

[11] Fu, Z., Wang, F., Cheng, X. (2020). The secure steganography for hiding images via GAN. EURASIP Journal on Image and Video Processing, 2020(1): 46. https://doi.org/10.1186/s13640-020-00534-2

[12] Ramandi, V.Y., Fateh, M., Rezvani, M. (2024). VidaGAN: Adaptive GAN for image steganography. IET Image Processing, 18(12): 3329-3342. https://doi.org/10.1049/ipr2.13177

[13] Zhu, J., Kaplan, R., Johnson, J., Fei-Fei, L. (2018). Hidden: Hiding data with deep networks. In Proceedings of the European Conference on Computer Vision (ECCV): 657-672. https://doi.org/10.48550/arXiv.1807.09937

[14] Jing, J., Deng, X., Xu, M., Wang, J., Guan, Z. (2021). Hinet: Deep image hiding by invertible network. In 2021 IEEE/CVF international conference on computer vision (ICCV), Montreal, QC, Canada, pp. 4713-4722. https://doi.org/10.1109/ICCV48922.2021.00469

[15] Duan, X., Li, B., Yin, Z., et al. (2023). Robust image steganography against lossy JPEG compression based on embedding domain selection and adaptive error correction. Expert Systems with Applications, 229: 120416. https://doi.org/10.1016/j.eswa.2023.120416

[16] Duan, X., Liu, N., Gou, M., Wang, W., Qin, C. (2020). SteganoCNN: Image steganography with generalization ability based on convolutional neural network. Entropy, 22(10): 1140. https://doi.org/10.3390/e22101140

[17] Huang, D., Luo, W., Liu, M., et al. (2023). Steganography embedding cost learning with generative multi-adversarial network. IEEE Transactions on Information Forensics and Security, 19: 15-29. https://doi.org/10.1109/TIFS.2023.3318939

[18] Zhou, Y., Wang, N., Hong, X., et al. (2025). Deep learning-based image steganography with latent space embedding and smart decoder selection. Entropy, 27(12): 1223. https://doi.org/10.3390/e27121223

[19] Kombrink, M.H., Geradts, Z.J.M.H., Worring, M. (2024). Image steganography approaches and their detection strategies: A survey. ACM Computing Surveys, 57(2): 1-40. https://doi.org/10.1145/3694965

[20] Ji, P., Zhang, Y., Lv, Z. (2025). Edge-guided dual-stream U-Net for secure image steganography. Applied Sciences, 15(8): 4413. https://doi.org/10.3390/app15084413
[21] Zhou, Z., Su, Y., Li, J., et al. (2022). Secret-to-image reversible transformation for generative steganography. IEEE Transactions on Dependable and Secure Computing, 20(5): 4118-4134. https://doi.org/10.1109/TDSC.2022.3217661
[22] Liu, L., Tang, L., Zheng, W. (2022). Lossless Image steganography based on invertible neural networks. Entropy, 24(12): 1762. https://doi.org/10.3390/e24121762
[23] Sanjalawe, Y., Al-E'mari, S., Fraihat, S., et al. (2025). A deep learning-driven multi-layered steganographic approach for enhanced data security. Scientific Reports, 15(1): 4761. https://doi.org/10.1038/s41598-025-89189-5