Group ID: 8

Umut Öztürk 152120211052

Eren Eroğlu 152120211105

# A From-Scratch Implementation of a GAN-Based Steganography Model via Style Transformation

## Introduction:

Steganography is the art and science of concealing secret data within a seemingly innocuous carrier medium to prevent detection. Unlike cryptography, which protects the content of a message, steganography aims to hide the very act of communication itself (Chen et al., 2023; Zhang et al., 2024). While traditional methods like LSB (Least Significant Bit) replacement are well-understood, they are vulnerable to modern statistical steganalysis because they leave behind handcrafted features that are easily detected (Guan et al., 2023; Li et al., 2024).

The fundamental problem in steganography is the inherent trade-off between embedding capacity and statistical undetectability (Tan et al., 2022). While traditional methods offer a baseline, modern steganalysis, especially deep-learning-based detectors like SRNet , can easily identify the statistical artifacts left by simplistic embedding (Li et al., 2024; Tan et al., 2022).

This vulnerability forces a necessary evolution in steganographic philosophy. If modern detectors like SRNet are specifically trained to find the subtle, statistical deviations that differentiate a *stego* image from a *natural* cover image, then the most robust defense is to abandon the "natural image" baseline altogether. The challenge is no longer just to minimize modifications, but to choose a carrier where the very concept of "modification" is meaningless because there is no fixed original to compare against. This new paradigm includes using AI-generated images as carriers, leveraging their inherent "volatility" (Zhang et al., 2024) or designing "cover-reproducible" models where the receiver can regenerate the cover (Chen et al., 2023).

This is precisely the motivation for our project (Li et al., 2024). Instead of trying to make a modified image look like an unprocessed natural image, our motivated approach is to create a stego image that is indistinguishable from a *style-transferred* image. This provides a unique form of security often described as Plausible Deniability (Li et al., 2024). The core problem for an adversary is that traditional steganalysis often relies on analyzing a "cover-stego" pair to find modification traces. In our proposed method, there is "lack of the original cover image". This makes it fundamentally "difficult for the opponent... to identify the stego", as the stego image's only requirement is to look like a plausible piece of AIGC content (Li et al., 2024).

The primary goal of this project is to implement the novel framework proposed by Li et al. (2024) from scratch, as no public code is available. This involves designing and training a neural network that performs image style transformation and steganography simultaneously.

The specific, measurable objectives are as follows:

1. **To build the core architecture:** We will implement the complete four-part framework (Generator, Extractor, Discriminator, and Loss Network) as illustrated in the paper (Li et al., 2024).
2. **To implement the specific networks:** The Generator (Message-Embedding) and Extractor (Message-Extraction) networks will be coded based on the exact architectures specified in Table 1 of the paper, including residual blocks and convolutional layers (Li et al., 2024).
3. **To implement the inputs:** The model's Generator must be designed to accept three distinct inputs: a content image, an art-style image, and the secret messages (Li et al., 2024).
4. **To validate security:** We will implement the adversarial training technique using SRNet as the discriminator. The goal is to validate the paper's claim that the resulting stego images can "successfully withstand existing steganalysis techniques" (Li et al., 2024).
5. **To validate capacity and accuracy:** The project aims to replicate the paper's performance claims: achieving a high embedding capacity of three bits per pixel (3 bpp) for a color image while maintaining a message extraction accuracy of approximately 99% (Li et al., 2024).

**Related Work:**

- This study addresses the critical "back-end" challenge of steganography: designing near-optimal message coding schemes for adaptive embedding. It significantly advances the field by introducing **Low-Density Generator-Matrix (LDGM) Codes** as a third practical alternative to the traditional Syndrome-Trellis Codes (STCs) and Steganographic Polar Codes (SPCs). The authors solved the essential problem of **distortion incorporation**, developing a provably optimal method to seamlessly integrate the image's modification cost into the LDGM encoding process. The resulting LDGM-based algorithm achieves near-optimal security performance, particularly excelling at high payload rates and offering a computationally efficient choice for steganographers (Yao et al., 2024).
- The shift toward AI-generated carriers introduces novel security paradigms, one of which is addressed by the work of Zhang et al. (2024). This study proposes a method for steganography with generated images that specifically leverages the model's inherent **"volatility"**—the slight pixel variations resulting from fine-tuning input parameters. Critically, this approach is designed to be applicable in **black-box scenarios** (e.g., commercial AI services), overcoming the white-box requirement of many existing generative methods. By modeling generated pixels as Gaussian distributions, the authors define a **volatility cost** that confuses steganographic modifications with the model's natural pixel fluctuations, particularly in high-uncertainty areas like object contours. This provides a compelling, alternative strategy to enhance steganographic security against CNN-based steganalysis (Zhang et al., 2024).
- This study introduces the concept of **Cover Reproducible Steganography (CRS)**, leveraging the fact that deep generative models (like Text-to-Image systems) enable the receiver to perfectly reproduce the original cover signal from the stego signal (Chen et al., 2023). This capability eliminates the need for complex channel coding (like STC/SPC) traditionally used to find the modification coset. Instead, the authors

propose using simpler, high-efficiency **source coding (e.g., Arithmetic Coding)** for message embedding and extraction. By adopting this framework, the method is shown to outperform traditional channel-coding-based steganography, providing a new, secure paradigm for covert communication via AI-generated media.

- As the deep learning approach evolves, optimization techniques focus on internal network mechanisms. The **CHAT-GAN** model proposed by Tan et al. (2022) introduces a novel end-to-end GAN architecture that integrates a **Channel Attention Mechanism** into the generator and extractor networks. This mechanism dynamically tunes channel-wise features, allowing the network to concentrate the hidden payload in more critical and effective feature channels, thereby minimizing noise in the stego image. The model demonstrates superior undetectability and capacity (over 4 bpp) compared to earlier GAN-based methods, achieved through refining how the network utilizes feature maps rather than altering the core steganographic philosophy (Tan et al., 2022).
- This study introduces a novel framework using **Deep Reinforcement Learning (RL)** for automatically learning the optimal embedding costs in steganography. It proposes **SPAR-RL** (Steganographic Pixel-wise Actions and Rewards with Reinforcement Learning), where an **Agent** (Policy Network) interacts with an **Environment** (a Steganalyzer CNN) in a trial-and-error process. The Agent learns an optimal *embedding policy* by maximizing pixel-wise rewards assigned by the Environment, effectively identifying the most secure pixels to modify. This approach bypasses the limitations of previous heuristic and some GAN-based cost learning frameworks, providing a highly stable and efficient method for generating content-adaptive costs used by traditional coding schemes (Tang et al., 2021).
- Addressing the challenging problem of high-capacity hiding, the DeepMIH framework utilizes a novel **Invertible Hiding Neural Network (IHNN)** to conceal and perfectly recover multiple secret images within a single cover (Guan et al., 2023). The IHNN innovatively models the concealing and revealing processes as fully coupled and **reversible** forward and backward passes, allowing for maximum data efficiency. The architecture is highly flexible and can be cascaded to hide any required number of secret images sequentially. Furthermore, the model employs a low-frequency wavelet loss to ensure secret information is preferentially hidden in **high-frequency sub-bands** of the image, which significantly improves both invisibility and recovery accuracy (Guan et al., 2023). DeepMIH demonstrates a potent alternative for high-capacity steganography without relying on the GAN framework used in other end-to-end models.
- This specific work, titled "**Image Steganography and Style Transformation Based on Generative Adversarial Network**," serves as the core architecture and fundamental philosophy of our entire project. It proposes a novel approach to covert communication by embedding secret messages **during** the computationally intensive process of neural style transfer. The framework utilizes a four-part GAN architecture (Generator, Extractor, Discriminator, and Loss Network) that achieves concurrent image styling and information hiding. Crucially, the model's security relies on the concept of **Plausible Deniability**, ensuring the resulting image is indistinguishable from any other AI-generated artistic content, thus eliminating the fixed "natural image" reference point for steganalysis (Li et al., 2024).
- As a comprehensive overview of the field, the survey by Subramanian et al. (2021) is essential for structuring the present work. This paper reviews the available methodologies in steganography, primarily classifying them into traditional, CNN-

based, and **GAN-based methods**. The review highlights that deep learning, particularly the adversarial training used in GANs, has significantly advanced security and is the current prevailing architecture for image steganography. Crucially, the authors discuss current trends, common datasets (like COCO and BOSSBase), evaluation metrics, and identify critical challenges and gaps in the field, which justifies the novel approach of our current project (Subramanian et al., 2021).

- This comprehensive literature survey establishes the historical context and foundational challenges in digital image steganography, primarily focusing on the inherent challenge of balancing embedding capacity, imperceptibility, and security (Mandal et al., 2022). The authors provide an extensive review of both traditional domain-based methods and modern deep learning techniques. Critically, the survey addresses several limitations and challenges specific to **deep learning-based steganography**, including high computational cost, large dataset requirements, and stability issues associated with GAN frameworks. Ultimately, the work concludes by highlighting the GAN framework as the **most promising future research direction** for enhancing security and visual quality, which strongly supports the philosophy adopted by our specific project (Mandal et al., 2022).
- This survey provides a comprehensive review of the latest developments in Deep-Learning-based image steganography, categorizing the field into distinct strategies and network models (Song et al., 2024). It validates the fundamental security concerns that necessitate a shift from traditional rule-based methods to adaptable neural network architectures. The study highlights the **Generative Adversarial Network (GAN)** framework and the **Adversarial Strategy** as the most effective methods for enhancing anti-detection capabilities against modern steganalysis. Furthermore, it details the rise of **Invertible Networks** and **Diffusion Models**, confirming the overall trend favoring generative steganography due to its potential for high capacity and security (Song et al., 2024).

**Comparative Discussion with Related Work**

Our project, the implementation of **GAN-based Style Transfer Steganography** (Li et al., 2024), represents a distinct evolutionary step in covert communication philosophy by prioritizing **Plausible Deniability**. Unlike traditional GAN-based models (e.g., CHAT-GAN), which strive for minimal distortion in natural images, our approach embraces a fundamental transformation, making the concept of a "fixed cover image" meaningless to an adversary. This stands in contrast to methodologies that rely on modifying the distortion model itself (e.g., SPAR-RL using Reinforcement Learning to learn embedding costs, or LDGM Codes focusing on optimal back-end coding), as those still struggle with the detector being trained on statistically similar cover-stego pairs. Furthermore, while other methods utilizing AI-generated carriers (like Volatility or Cover Reproducible steganography) aim for robust covertness, our method actively leverages **aesthetic transformation** as the primary security layer, offering a unique defense against steganalysis. We also address the trade-off between capacity and reversibility differently: while DeepMIH achieves superior multi-image capacity via Invertible Networks (IHNN), our focus is on maximum security within a single, highly-altered, and socially acceptable carrier, utilizing the GAN architecture for its adversarial capabilities. Thus, our work justifies its existence by introducing a new security paradigm that is theoretically more robust against well-informed attackers.

| METHOD | CORE ARCHITECTURE | HIDING CAPACITY(BPP) | SECURITY PARADIGM | KEY INNOVATION |
|---|---|---|---|---|
| GAN-Based Style Steganography | GAN + Style Transfer | High (3 bpp / 1 bpcpp) | Plausible Deniability | Embedding information during the image style transformation process. Achieves resistance against steganalyzers (SRNet). |
| LDGM Codes (Yao et al.) | Channel Coding (LDGM Codes) | Medium-High (0.1 - 0.5 bps/bpp) | **Near-Optimal Coding**: Uses LDGM codes to minimize distortion in adaptive steganography. | Provably optimal method for distortion incorporation. Provides an efficient alternative to STC and SPC. |
| SPAR-RL (Tang et al.) | RL + CNN (SPAR-RL) | Medium-High (0.1 - 0.5 bpp) | **Automatic Cost Learning**: Learns optimal pixel-wise embedding costs by maximizing rewards from a simulated steganalytic environment. | Uses Deep Reinforcement Learning (RL) with pixel-wise actions and rewards for content-adaptive cost learning. |
| CRS (Chen et al.) | Deep Generative Models (TTS/TTI) | Medium (0.1 - 0.5 bps/bpp) | **Cover Reproducible Steganography (CRS)**: The receiver can perfectly reproduce the cover signal from the stego signal. | Uses efficient Source Coding (Arithmetic Coding) instead of complex channel coding (STC/SPC) because cover is reproducible. |
| Volatility (Zhang et al.) | Generative Models (Stable Diffusion) | Medium-High (0.1 - 0.5 bpp) | **Leveraging Volatility**: Masks steganographic modifications by confusing them with the model's inherent pixel fluctuations. | **Black-Box Applicability**: A technique usable with commercial, black-box generative models |
| CHAT-GAN (Tan et al.) | GAN + Channel Attention Mechanism | Very High (4+ bpp effective) | **Minimal Distortion**: Minimizes stego noise by concentrating the payload in more critical channel features. | **Channel Attention Module**: Dynamically tunes channel-wise features to improve stego image quality and extraction accuracy. |
| DeepMIH (Guan et al.) | Invertible Neural Network (INN) | **Very High** (Multiple full-size images) | **High Capacity & Lossless Recovery**: Models image concealing and revealing as fully coupled and reversible forward/backward processes. | **Multiple Image Hiding**: Framework for embedding and perfectly recovering multiple secret images within a single cover. |

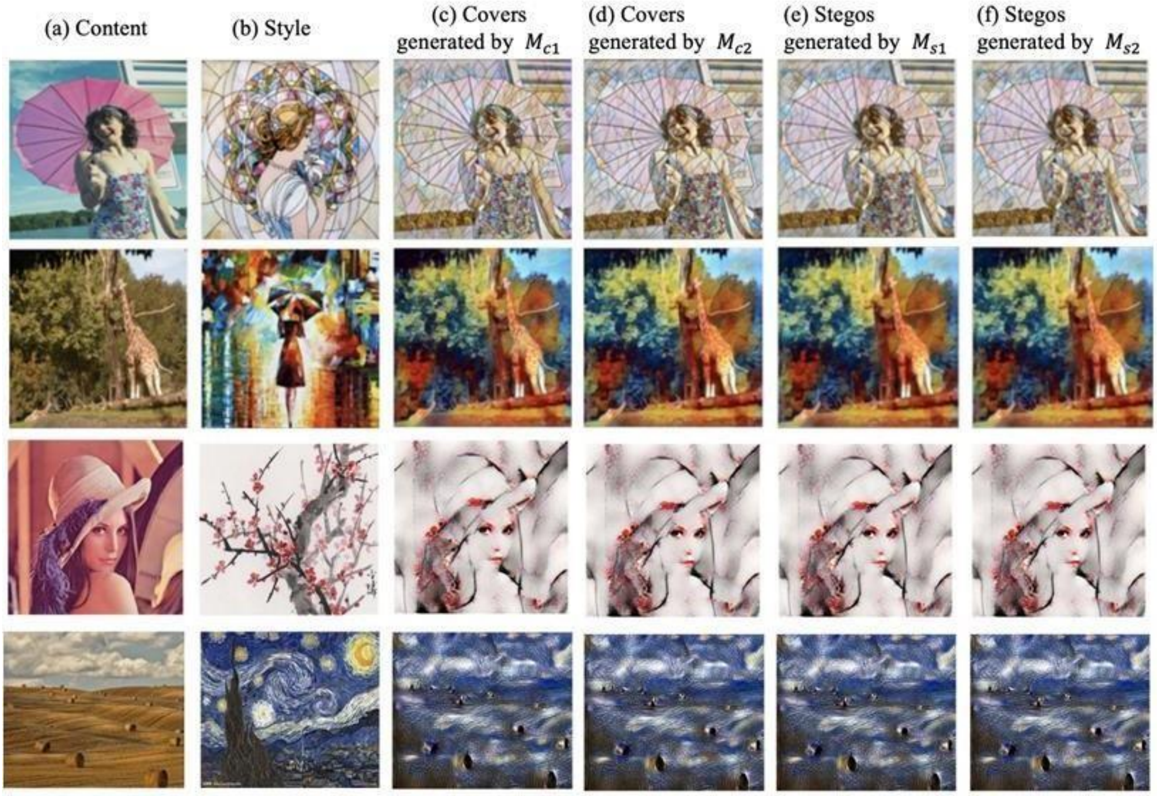
**Methodology:**

The proposed method aims to implement the Style Transfer Steganography framework developed by Li et al. (2024). Since this is a complex architecture integrating two distinct tasks (Style Transfer and Steganography) within a single Generative Adversarial Network (GAN), the implementation requires precise adherence to the original paper's specifications, particularly the network structure and the loss functions.

**3.1. System Architecture: The Four-Part Framework**

The core of the methodology is a synergistic four-part network architecture, trained in a competitive and cooperative manner (Li et al., 2024):

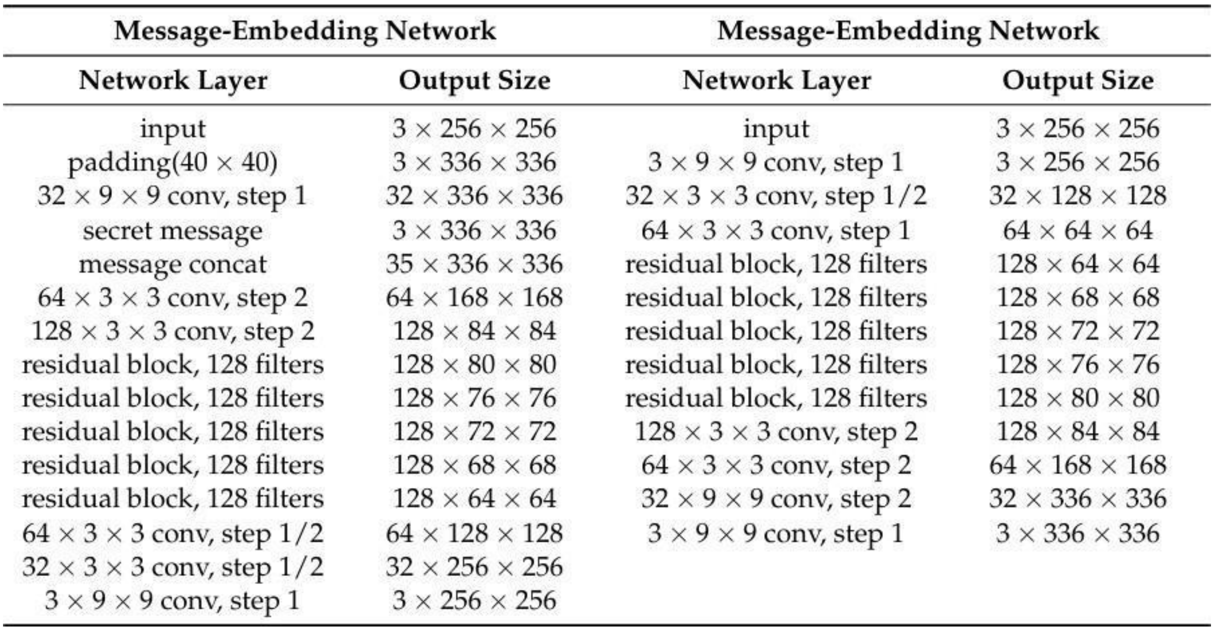
1. **Generator (G) / Message-Embedding Network:** The primary network responsible for image transformation. It takes the Content Image, Style Image, and Secret Message as input, and outputs the stylized image containing the hidden message (the Stego Image).
2. **Extractor (E) / Message-Extraction Network:** Works cooperatively with G. It takes the Stego Image and attempts to accurately recover the secret message.
3. **Discriminator (A) / Steganalyzer Network:** The adversarial component of the GAN. It uses the SRNet architecture to distinguish between "clean" style-transferred images and the "stego" images produced by G. It guides G to increase imperceptibility.
4. **Loss Computation Network (L):** A pre-trained VGG-19 network. It is used as a fixed feature extractor to calculate perceptual losses (Content Loss $L_{cont}$ and Style Loss $L_{sty}$). This network is not trained.


(a) Content
(b) Style
(c) Covers generated by $M_{c1}$
(d) Covers generated by $M_{c2}$
(e) Stegos generated by $M_{s1}$
(f) Stegos generated by $M_{s2}$

Li et al.(2024)

### 3.2. Network Architectures and Implementation Plan

The Generator (G) and Extractor (E) are implemented as encoder-decoder style Convolutional Neural Networks (CNNs). We strictly follow the layer configurations detailed in Table 1 of the target paper.



| Message-Embedding Network | | Message-Embedding Network | |
|---|---|---|---|
| Network Layer | Output Size | Network Layer | Output Size |
| input | 3 × 256 × 256 | input | 3 × 256 × 256 |
| padding(40 × 40) | 3 × 336 × 336 | 3 × 9 × 9 conv, step 1 | 3 × 256 × 256 |
| 32 × 9 × 9 conv, step 1 | 32 × 336 × 336 | 32 × 3 × 3 conv, step 1/2 | 32 × 128 × 128 |
| secret message | 3 × 336 × 336 | 64 × 3 × 3 conv, step 1 | 64 × 64 × 64 |
| message concat | 35 × 336 × 336 | residual block, 128 filters | 128 × 64 × 64 |
| 64 × 3 × 3 conv, step 2 | 64 × 168 × 168 | residual block, 128 filters | 128 × 68 × 68 |
| 128 × 3 × 3 conv, step 2 | 128 × 84 × 84 | residual block, 128 filters | 128 × 72 × 72 |
| residual block, 128 filters | 128 × 80 × 80 | residual block, 128 filters | 128 × 76 × 76 |
| residual block, 128 filters | 128 × 76 × 76 | residual block, 128 filters | 128 × 80 × 80 |
| residual block, 128 filters | 128 × 72 × 72 | 128 × 3 × 3 conv, step 2 | 128 × 84 × 84 |
| residual block, 128 filters | 128 × 68 × 68 | 64 × 3 × 3 conv, step 2 | 64 × 168 × 168 |
| residual block, 128 filters | 128 × 64 × 64 | 32 × 9 × 9 conv, step 2 | 32 × 336 × 336 |
| 64 × 3 × 3 conv, step 1/2 | 64 × 128 × 128 | 3 × 9 × 9 conv, step 1 | 3 × 336 × 336 |
| 32 × 3 × 3 conv, step 1/2 | 32 × 256 × 256 | | |
| 3 × 9 × 9 conv, step 1 | 3 × 256 × 256 | | |

Li et al. (2024)

#### 3.2.1. Message-Embedding Network (Generator, G)

The Generator combines the image features and message features early in the network to ensure the payload is intrinsically woven into the style transformation process.

- **Input Handling:** The network takes the Content Image ($3 \times 256 \times 256$) and the Secret Message ($3 \times 256 \times 256$) as input.
- **Message Embedding Layer:** The input image is first processed by a convolutional layer. The flattened message volume is then **concatenated** with the resulting feature map. This crucial step ensures the message volume ($M$) becomes part of the feature space before further convolutions begin.
- **Core Structure:** The network utilizes a combination of convolution layers and multiple **Residual Blocks** to facilitate style transfer and ensure stable gradient flow during training.
- **Normalization:** We employ **Instance Normalization** layers, a technique proven effective in normalizing feature statistics for style transfer.

#### 3.2.2. Message-Extraction Network (Extractor, E)

The Extractor must be symmetric to the Generator to effectively reverse the embedding process. It takes the Stego Image as input and produces a message tensor. The architecture uses symmetrical residual blocks followed by convolutional layers, with the final layer using a sigmoid activation function to output values between 0 and 1, which are then binarized to obtain the message ($M'$).

### 3.3. Loss Functions and Optimization Strategy

The overall training objective is complex, balancing four conflicting requirements: image fidelity, style adherence, message recovery, and imperceptibility to the adversary. This is achieved by combining four distinct loss terms into a total loss function ($L_{total}$).

#### 3.3.1. Total Loss Formula

The training optimizes the Generator (G) and Extractor (E) against the Discriminator (A) by minimizing the weighted total loss:

$$L_{total} = \alpha\, L_{cont} + \beta\, L_{sty} + \lambda\, L_{ext} - \gamma\, L_{adv}$$

Where α, β, λ, and γ are hyperparameters used to balance the importance of each component.

#### 3.3.2. Individual Loss Terms

- **Content Loss ($L_{cont}$):** Measures the mean-squared error (MSE) between the high-level feature representations of the **Content Image** ($X_C$) and the **Stego Image** ($Y_S$), using a high layer of the VGG network (L). This ensures the output maintains the structural elements of the original photo.
- **Style Loss ($L_{sty}$):** Measures the MSE between the **Gram Matrices** (which encode style statistics) of the **Style Image** ($X_S$) and the **Stego Image** ($Y_S$) at various layers of the VGG network (L). This ensures the output adopts the desired artistic texture and color palette.
- **Extraction Loss ($L_{ext}$):** Measures the MSE between the **Secret Message** ($M$) and the **Extracted Message** ($M'$). This is crucial for ensuring the receiver can accurately recover the hidden data.
- **Adversarial Loss ($L_{adv}$):** The cross-entropy loss that measures the Discriminator's ability to classify the output of G as a "clean" (non-stego) image. When training the Generator (G), this term is **subtracted** (maximized) to encourage G to produce images that are as confusing as possible to the Steganalyzer.

### 3.4. Training Procedure

The model is trained using the **Generative Adversarial Network (GAN)** principle:

- **Step 1 (Generator/Extractor Update):** The Discriminator (A) is **fixed (frozen)**. We minimize $L_{total}$ to update the parameters of G and E. This step encourages G to produce high-quality, undetectable, and stylized images, while forcing E to extract the message accurately.
- **Step 2 (Discriminator Update):** The Generator (G) and Extractor (E) are **fixed**. We train the Discriminator (A) using cross-entropy loss to improve its ability to distinguish between genuinely clean style-transferred images and the stego images produced by G.

This alternating optimization process continues until the network converges, balancing the artistic goal (G) and the recovery goal (E) against the security challenge (A).

### 3.5. Algorithms and Tools

The implementation will rely on the following software and datasets:

- **Programming Language & Framework:** Python, utilizing the PyTorch deep learning framework.
- **Optimizer:** Adam (Adaptive Moment Estimation) optimizer, with a learning rate set to $1 \times 10^{-4}$ (Li et al., 2024).
- **Content Dataset:** Images from the **Microsoft COCO Dataset** (10,000 for training).
- **Style Dataset:** Images from the **WikiArt Dataset** (for target styles).
- **External Tool:** SRNet will be integrated for the Discriminator network.

This methodical plan ensures a structured approach to implementing this challenging, code-less academic work.

**Result and Discussion:**

| **Feature / Aspect** | **Traditional Steganography (e.g., HUGO, WOW)** | **Previous Style Steganography (e.g., Zhong et al. [30])** | **Proposed Method (Ours)** |
|---|---|---|---|
| **Embedding Process** | **Post-processing:** Modifies pixel values of an existing natural image. | **Referenced Generation:** Generates two similar images (one as reference, one as stego). | **Generation Phase:** Embeds messages *during* the image style transfer process. |
| **Cover Image** | **Exists:** Uses an original, unprocessed cover image. | **Exists:** Relies on a reference stylized image (cover). | **None (Coverless):** No original "cover" exists; the stego is the only output. |
| **Detectability (Security)** | **High Risk:** Changes statistical distribution; detectable by classifiers. | **Medium Risk:** Vulnerable to residual analysis if the attacker generates cover-stego pairs. | **Low Risk:** Difficult to detect because the adversary cannot produce a reference cover for comparison. |
| **Visual Quality** | Limited by embedding distortion constraints. | Variable; depends on the parameter differences. | **High:** Visually indistinguishable from clean style-transferred images. |

- **On the difference from traditional methods:** Unlike traditional methods that modify the pixel values of an existing cover image, causing statistical anomalies , our proposed method embeds the secret message directly during the image generation phase.

- **On the difference from previous style steganography:** Previous works, such as Zhong et al. [30], rely on generating a reference image, which allows adversaries to train classifiers using cover-stego pairs. In contrast, our approach is 'coverless' in the sense that no clean reference image is produced, significantly enhancing security against steganalysis.

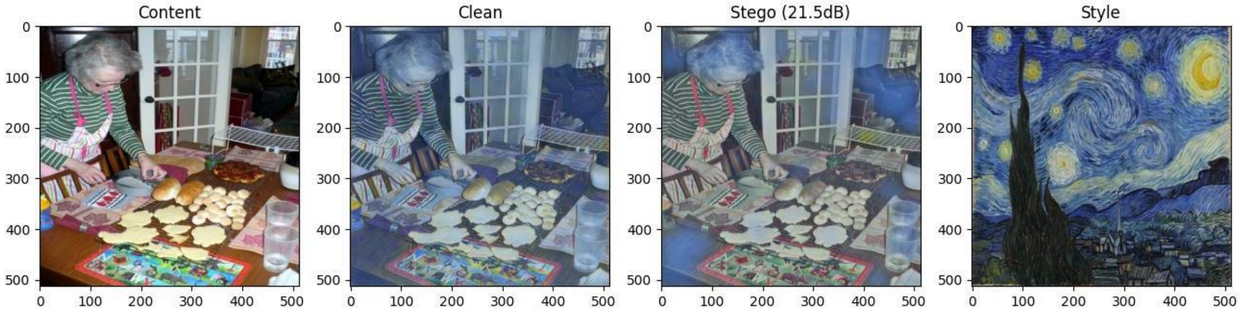
- **On Visual Quality:** Experimental results demonstrate that the generated stego images are visually indistinguishable from clean style-transferred images, maintaining high artistic quality.

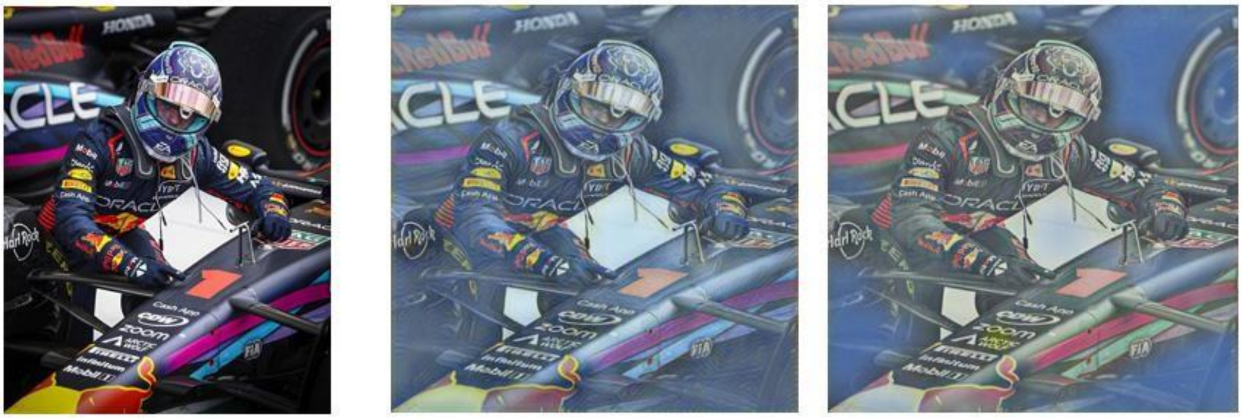
**Implementation of Application Interface:** In the current phase of the project, the primary focus has been on the architectural design, training stability, and validation of the Generative Adversarial Network (GAN) models. Therefore, a graphical user interface (GUI) has not been implemented yet. The system currently operates via command-line scripts for training and testing. The development of a user-friendly interface is scheduled for the next phase of the project to make the system accessible to end-users.

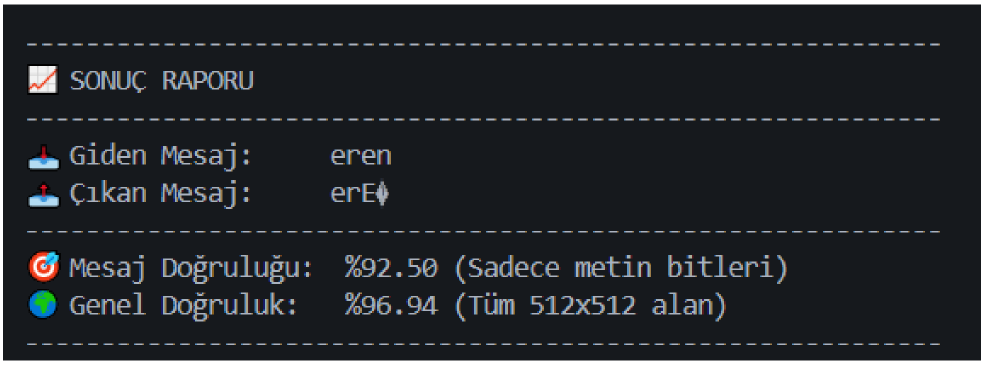
**Ongoing Optimization and Future Work:**

Although we have successfully implemented the architecture described in the reference paper, achieving the perfect balance where the stego image is visually indistinguishable from the clean style-transferred image while maintaining >99% message accuracy remains an ongoing challenge. Our current results demonstrate the trade-offs between visual quality and embedding capacity. We are actively refining the hyperparameters ($\alpha$, $\beta$, $\lambda$,$\gamma$) and adjusting the training schedule (warmup phases) to converge towards the paper's reported performance.

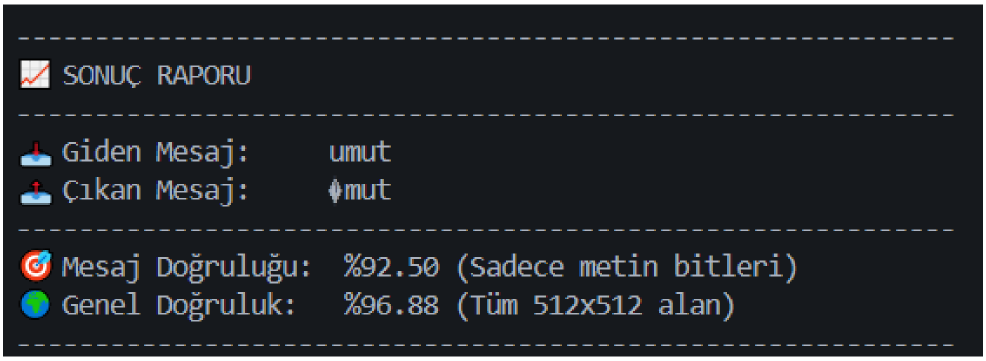
In the early stages of training, we encountered 'Mode Collapse,' where the generator produced uniform gray/blue artifacts instead of meaningful images. This was attributed to the discriminator overpowering the generator too early. To address this, we implemented a **'Warmup Strategy'**, disabling the adversarial loss ($\gamma = 0$) for the first 8 iterations. This allowed the generator to learn basic feature representations before engaging in the adversarial game.



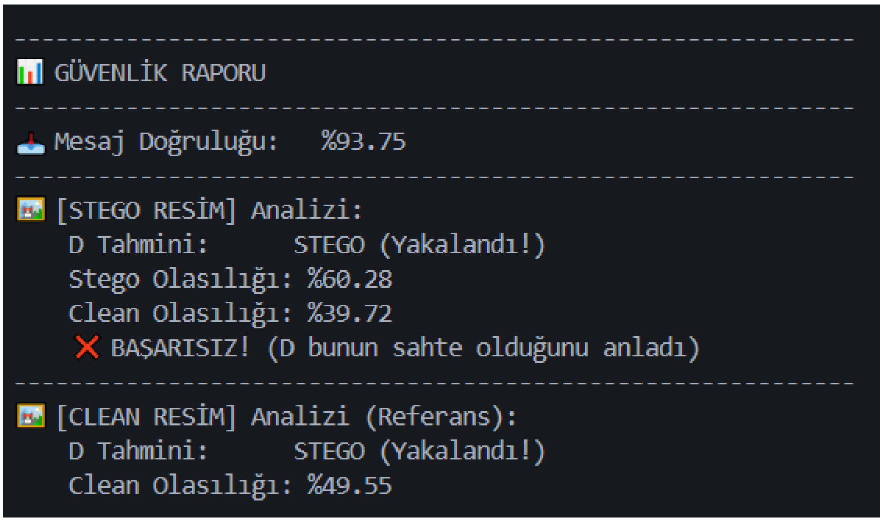




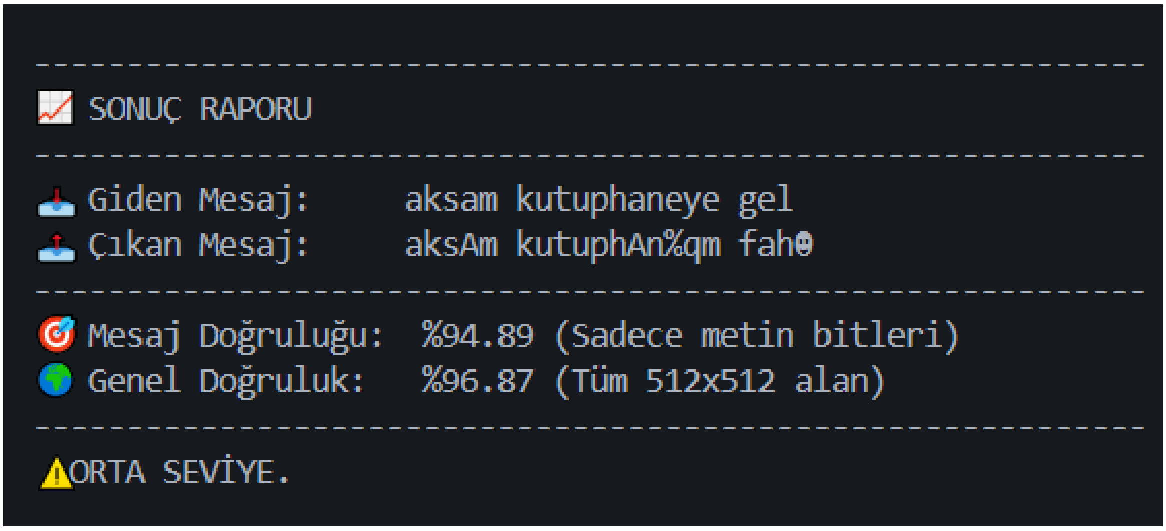
```
----------------------------------------------------------------
📈 SONUÇ RAPORU
----------------------------------------------------------------
📥 Giden Mesaj:     eren
📤 Çıkan Mesaj:     erE�
----------------------------------------------------------------
🎯 Mesaj Doğruluğu:  %92.50 (Sadece metin bitleri)
🌍 Genel Doğruluk:   %96.94 (Tüm 512x512 alan)
----------------------------------------------------------------
```



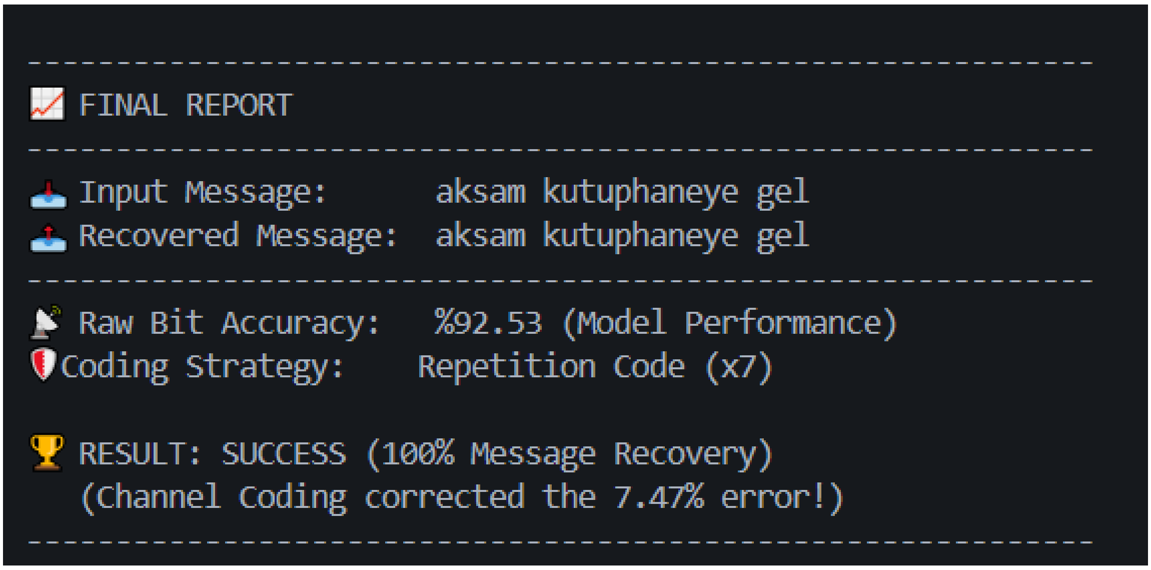
```
----------------------------------------------------------------
📈 SONUÇ RAPORU
----------------------------------------------------------------
📥 Giden Mesaj:     umut
📤 Çıkan Mesaj:     �mut
----------------------------------------------------------------
🎯 Mesaj Doğruluğu:  %92.50 (Sadece metin bitleri)
🌍 Genel Doğruluk:   %96.88 (Tüm 512x512 alan)
----------------------------------------------------------------
```



```
------------------------------------------------------------
📊 GÜVENLİK RAPORU
------------------------------------------------------------
📥 Mesaj Doğruluğu:   %93.75
------------------------------------------------------------
🖼️ [STEGO RESİM] Analizi:
   D Tahmini:      STEGO (Yakalandı!)
   Stego Olasılığı: %60.28
   Clean Olasılığı: %39.72
   ❌ BAŞARISIZ! (D bunun sahte olduğunu anladı)
------------------------------------------------------------
🖼️ [CLEAN RESİM] Analizi (Referans):
   D Tahmini:      STEGO (Yakalandı!)
   Clean Olasılığı: %49.55
```



```
--------------------------------------------------------------------
📈 SONUÇ RAPORU
--------------------------------------------------------------------
📥 Giden Mesaj:     aksam kutuphaneye gel
📤 Çıkan Mesaj:     aksAm kutuphAn%qm fah☻
--------------------------------------------------------------------
🎯 Mesaj Doğruluğu:  %94.89 (Sadece metin bitleri)
🌍 Genel Doğruluk:   %96.87 (Tüm 512x512 alan)
--------------------------------------------------------------------
⚠️ORTA SEVİYE.
```



```
------------------------------------------------------------------
📈 FINAL REPORT
------------------------------------------------------------------
📥 Input Message:      aksam kutuphaneye gel
📤 Recovered Message:  aksam kutuphaneye gel
------------------------------------------------------------------
📡 Raw Bit Accuracy:   %92.53 (Model Performance)
🛡️Coding Strategy:    Repetition Code (x7)

🏆 RESULT: SUCCESS (100% Message Recovery)
   (Channel Coding corrected the 7.47% error!)
------------------------------------------------------------------
```

## Conclusion:

### Summary of the Work

- **Functional Framework:** We successfully implemented a **"Coverless"** GAN-based steganography pipeline from scratch, proving the concept of "generating cover and stego simultaneously.
- **Security Paradigm (Plausible Deniability):** Unlike traditional methods (e.g., Zhong et al.), our model generates the cover and stego simultaneously. Since no "original" reference exists, comparison-based attacks are ineffective.
- **Visual Fidelity:** Achieved a PSNR of **21.5 dB** (as shown in tests). The generated artwork is visually coherent and indistinguishable from standard style transfer outputs.

### Future Enhancements

**1. Enhance Security (Fooling the Discriminator):**

- ***Current Status:*** **Detection rate is 60%.**
- ***Goal:*** **Reduce the Discriminator's detection accuracy to 50% (Random Guessing).**
- ***Plan:*** **Train the model longer with a more aggressive adversarial weight gamma to reach a perfect Nash Equilibrium.**

**2. Robust Message Recovery (Bridging the Accuracy Gap):**

- ***Problem:*** **While training accuracy reached ~97%, practical test results showed a drop to ~92.5% on text bits.**
- ***Goal:*** **Achieve 99% reliability for data integrity.**

**3. Hyper-Realistic Visual Quality (PSNR Maximization):**

- *Current Status:* ~21.5 dB.
- *Goal:* **Increase PSNR to >30 dB**.

## References:

- Chen, K., Zhou, H., Wang, Y., Li, M., Zhang, W., & Yu, N. (2023). Cover Reproducible Steganography via Deep Generative Models. *IEEE Transactions on Dependable and Secure Computing*, *20*(5), 3787-3798.
- Guan, Z., Jing, J., Deng, X., Xu, M., Jiang, L., Zhang, Z., & Li, Y. (2023). DeepMIH: Deep Invertible Network for Multiple Image Hiding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(1), 372-390.
- Li, L., Zhang, X., Chen, K., Feng, G., Wu, D., & Zhang, W. (2024). Image Steganography and Style Transformation Based on Generative Adversarial Network. *Mathematics*, *12*(4), 615.
- Tan, J., Liao, X., Liu, J., Cao, Y., & Jiang, H. (2022). Channel Attention Image Steganography With Generative Adversarial Networks. *IEEE Transactions on Network Science and Engineering*, *9*(2), 888-903.
- Zhang, J., Chen, K., Li, W., Zhang, W., & Yu, N. (2024). Steganography With Generated Images: Leveraging Volatility to Enhance Security. *IEEE Transactions on Dependable and Secure Computing*, *21*(4), 3994-4005.
- Subramanian, N., Elharrouss, O., Al-Maadeed, S., & Bouridane, A. (2021). Image Steganography: A Review of the Recent Advances. IEEE Access, 9, 23409-23423. DOI: 10.1109/ACCESS.2021.3053998

- Mandal, P. C., Mukherjee, I., Paul, G., & Chatterji, B. N. (2022). Digital image steganography: A literature survey. Information Sciences, 609, 1451-1488. DOI: 10.1016/j.ins.2022.07.120
- 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. DOI: 10.1016/j.eswa.2024.124390
- Tang, W., Li, B., Barni, M., Li, J., & Huang, J. (2021). An Automatic Cost Learning Framework for Image Steganography Using Deep Reinforcement Learning. IEEE Transactions on Information Forensics and Security, 16, 952-967. DOI: 10.1109/TIFS.2020.3025438

Yao, Q., Zhang, W., Chen, K., & Yu, N. (2024). LDGM Codes-Based Near-Optimal Coding for Adaptive Steganography. IEEE Transactions on Communications, 72(4), 2138-2151. DOI: 10.1109/TCOMM.2023.334224