SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

A Lightweight Deep Learning Framework for Secure Image Steganography in Sketch Domains

Uqba bn Nafaa Mohammed [1], Saba Abdulbaqi Salman [2], Yasmin Makki Mohialden [3], Raed Waheed Kadhim[4]

[1]Department of Computer Science, College of Education, Al-Iraqia University, Baghdad, Iraq. E-mail: uqba80@uomustansiriyah.edu.iq
[2]Computer Science Department, College of Science, Mustansiriyah University, Baghdad-Iraq. *E-mail:* saba.abd.salman@aliraqia.edu.iq
[3]Department of Computer Science, College of Education, Al-Iraqia University, Baghdad, Iraq. E-mail: ymmiraq2009@uomustansiriyah.edu.iq
[4]Department of Computer Science, College of Education, Al-Iraqia University, Baghdad, Iraq. E-mail: raedwk.cs@uomustansiriyah.edu.iq

ABSTRACT
In the era of the digital age, when a single pixel could be hiding confidential information, the necessity of secure, real-time steganography is more relevant than ever. The current paper introduces SteganoNet++, a deep-learning system designed to hide hidden image information within sketch-like images while keeping high quality and detail. It can be described as a lightthe authorsight encoder-decoder system, strengthened with adversarial training, which can demonstrate solid performance in even sparse visual realms.
One such central innovation is a dynamic generator, the SketchDataset, which provides different secret-cover image pairs on the fly, making the model a better generalizer. Experimental results reveal notably high reconstruction performance (SSIM = 0.9865, PSNR = 36.03 dB), though stego images still exhibit visible artifacts (SSIM = −0.2875, PSNR = −5.84 dB). Nevertheless, the framework is strong when data fidelity is the primary value rather than total invisibility, as in digital watermarking and forensic surveillance. SteganoNet++ can be implemented as a scalable and viable alternative to handle secure communications in limited-capacity settings with recent additions that allow real-time tracking and visual diagnostics.

KEYWORDS: Image Steganography, Deep Neural Networks, Adversarial Training, Sketch Image Embedding, Real-Time Encoding, Runtime Analysis, Visual Performance Evaluation.

1. Introduction

In an era of pervasive connectivity, the need for secure and discreet communication has never been more critical. While traditional cryptographic methods excel at protecting the content of messages, they ironically reveal their presence, potentially drawing unwanted attention. Steganography provides a different paradigm- covering up the fact that a communication is being done and not just its meaning [1, 2, 3].
Steganography is the art of preventing the existence of secret information in digital media, i.e., video, audio or images, and it is near impossible to detect the presence of secret information. Steganography is better than cryptography because, unlike cryptography, which makes the message readable, steganography hides the fact that the message exists, and this makes it suitable in confidential communication [3, 4].
Since the emergence of deep learning, steganography has been fundamentally changed. The passive pixels, which are just visual elements, have changed and instead integrated semantic encoding, carrying a meaning. Convolutional Neural Networks (CNNs) and particularly the ones with the encoder-decoder architectures have shown an elevated ability in concealing and reassembling secret information. This has become a trend and systems such as HiDDeN and SteganoCNN have led the way but are subject to breakdown in practice. Such models cannot readily generalize to other domains of visual experiences, perform poorly in adversarial settings, and seldom perform in real time [5,8].
SteganoNet++ is a generic steganography deep learning framework specifically designed for sketch-based media. This framework aims to resolve existing method drawbacks by adopting a novel encoder-decoder architecture with adversarial training. As a result, it can handle attacks without losing precision and speed. One component of this framework is the SketchDataset. The SketchDataset is a new method for designing secret-cover sketch pairs that results in significant progress in robustness and generalization [12, 13].
Though deep learning-based stealth models have made significant improvements, there are still many challenges, including:

First, they are scale fragile where they struggle to work with images of a low texture or low contrast, such as edge maps and graphics [9].

Second, without adversarial training, they fall to both statistical and neural detection [10].
Third, they will have high computational demands, which will impact systems that have time and resource constraints such as mobile and embedded systems [11].
SteganoNet++ integrates new features to overcome the limitations associated with security, detection, and scalability, and also improves performance regarding the deployment of low power devices. First, SteganoNet++ has better maneuverability against security, thus becoming undetectable by many detection methods. Second, it has a low latency architecture. This means that steganography with SteganoNet++ can be deployed in low power devices. Third, SteganoNet++ is implemented to provide balance between time (real time) and the quality of the image being recovered, as opposed to providing near perfect image recovery [16]. SteganoNet++ contains encryption, decryption, and feature extraction blocks in a single architecture; therefore, the system is able to perform all operations at real time. In addition, SteganoNet++ is trained using the domain-aware method, and this enables the system to learn how to generalize in presence of visually dissimilar data, beyond processing documents which resemble sketches.
The performance of the method is tested using MSE, PSNR and SSIM. In addition, real time operation is used to assess the capability of the system to balance security and fidelity.
SteganoNet++ has great potential for use in watermarking, secure incorporation of metadata, and assurance for data integrity in environments where the data provenance and integrity are extremely important, such as in medical imaging, legal documents, and in military context [17, 19].
The structure of this paper is as follows: Section 2 reviews related works, Section 3 describes the datasets and how they were prepared,- Section 4 describes the results,and Section 5 conclusion .

2. Related work

Progress in deep learning in image steganography has dramatically improved stealth communication systems in recent years. Especially for systems where data confidentiality and privacy are the most important factors. Data-driven techniques are moving in the direction of increasing the capacity of each secret and the ability to hide messages across the infrastructure of the network, without a high risk of them being detected.
As desktop improvements improve, so too do the effects of deep learning on network performance. Deep learning models are complex and if they are continuously used in edge devices or Wireless Sensor Networks (WSNs) then they will significantly increase latency and reduce throughput. Additionally, the models that are used have limited capacity to visualize, and therefore constrain real-world applications.
As a solution to some of the previously mentioned problems, real-time steganography systems must integrate low latency and high quality embedding. For that purpose, the SteganoNet++ authors have developed a new architecture and a framework based on adversarial training, which allows a balance to be achieved among payload, human capacity, and latency, which are essential factors in new networked systems.
In 2022, Yan Zhen Ren et al. developed a deep learning model with multiple stages which presented binary data in grayscale images. Their model achieved lossless data recovery and reached 100% in detection accuracy. However, its real-world applicability was relatively limited [20].

Zhu et al. (2018) developed HiDDeN, an end-to-end trainable model, which incorporates encryption, decryption and optimization. The model performed well at perception tasks; however, these tasks did not generalize to noisy environments; it was also vulnerable to adversarial attacks [21].

In 2020, Xintao Duan et al. introduced the methodology of SteganoCNN which provided the capability of hiding images using deep learning within images. The model increased the payload capacity (47.92 bits per pixel) of the image, but was unable to work in different scales and was very expensive in computational resources [22].
Erick Delenia et al. (2024) described a new CNN model for steganography analysis and detection. The model developed by Delenia et al. has three layers: a level of processing, a classification level, and a level of feature entry. The model developed by Delenia et al. also outperformed the model previously designed, as this model achieved an accuracy of 95 percent and predictive validity [23].

Duan et al. also developed several next incremental improvements for multi-image embedding for their model which resulted in trade-off performances in remote sensing and drone images. Nevertheless, the model performance was still considered higher, but at the high cost of enormous expense in computational resources and significant efforts in parameter tuning [24].

Following the development of machine learning-based GANs, and the focus on the significance of semantic features of images for enhancing realism and further improving PSNR and SSIM values in the classification of aerial scenes and underwater imagery, a number of methods were proposed. These methods, however, had high computational complexities and were considered not suitable for edge devices or real-time applications [25-27].

Light The authors note that there are attempts to balance speed and performance, such as SPA-GAN, but that most attention-based GANs are still hardware intensive. In other related areas such as speech enhancement or sensor data augmentation, there have been gains in performance at the cost of latency and resource consumption. [28–30]
More radically, Wu et al. mentioned Steganography Without Embedding (STHE AUTHORS). A new framework is described that completely changes the way steganography should be thought of. The objective of STHE AUTHORS is to generate composite images that are completely indistinguishable from normal images, thereby making most steganalysis techniques based on payload uncommonly large useless.

Table 1: Comparative Snapshot of Deep Learning-Based Steganography Methods

Method	Adversarial Defense	Generalization	Real-Time Capability	Payload Capacity
Multi-Stage System (Yan Zhen Ren et al., 2022)	Moderate	Limited	Moderate	Moderate
HiDDeN (Zhu et al., 2018)	No	Low	Medium	Medium
SteganoCNN (Xintao Duan et al., 2020)	No	Low	Low	High
Attention-GAN	Yes	Medium	Low	High
STHE AUTHORS (Wu et al.)	Yes	High	Medium	Low
SteganoNet++ (Proposed)	Yes	High	High	High

3. Methodology

SteganoNet++ is a complete deep learning framework for real-time image steganography in sketch domains. The framework contains three parts:

- A low-cost encoder to put the secret information in the cover images.
- A decode system which restores the concealed information.
- A classifier that detects in an image that it on the stego or that the image is real.

This encoder-decoder-discriminator structure is aimed to strike minimum visual perceivability with a high-quality secret reconstruction. Using adversarial training models adds resistance to steganalysis by incorporating realistic smart attacks.

3.1. Training Objective

The model is trained using a composite loss function:

$$L\_total = L\_recon + \lambda_1 L\_adv + \lambda_2 L\_percep \quad (1)$$

Where:

- L_recon is the MSE between the original and decoded secret images
- The Binary Cross-Entropy loss function is defined as L_adv. It promotes realism for stego images.
- For L_percep, the Structural Similarity Index (SSIM) is used to provide perceptual continuity.

The hyperparameters λ_1 and λ_2 regulate the trade-off betthe authorsen stealth, realism, and reconstruction accuracy. Typically, L_adv is given lothe authorsr the authorsight to avoid distorting the secret recovery process.

Dataset and Preprocessing

The authors train on a subset of the ImageNet-Sketch dataset. The dataset has around 50,000 hand-drawn black and white images that cover ~1,000 classes. We designed a custom SketchDataset class to randomly assign images to be either a cover or a secret to allow for more combinations to boost domain generalization.
To balance out the classes and encourage the model to learn abstract visual information in a sketch style, we utilize a bunch of geometric transformations that include flips and rotations, on the classes that are underrepresented.

3.2. Training configuration:

The model is trained in a Google Colab environment with the following settings:

- **Image Size**: 128 × 128 pixels
- **Batch Size**: 4
- **Optimizer**: Adam with a learning rate of **1e-4**
- **Hardware**: CUDA-enabled GPU or CPU
- **Epochs**: 100

3.3. Algorithm

ImageNet-Sketch can be understood as a domain-shifted benchmark, which consists of simplified representations of the standard ImageNet classes as line-drawing sketches. Since these have been designated as low-texture and high-abstraction images, their extraction has presented itself with new challenges in feature extraction and also imperceptible data embedding.

To solve the imbalance betthe authorsen classes, one of the geometric augmenters is flipping and rotation of underrepresented classes. Such augmentation scheme helps in producing a more balanced training signal and offers better generalization of the model through ambiguous or abstract visions of the pattern. Consequently, the network becomes accustomed to concentrate on structural and semantics hints, having a strong performance in abstract areas. tell-tale line-drawings of canonical ImageNet classes. These images are low-in-texture and high-in-abstraction; this makes feature extraction and embedding difficulty.

Due to class imbalance, geometric transformations (e.g. rotation, flipping) are applied to underrepresented classes. This serves to balance the training signal and provides generalization to less defined situations. The model learns to focus on structure and semantics, and learns to be robust to different domains.

3.3.1. Algorithm 1: SteganoNet++ Training Loop

Input: Cover image **C**, Secret image **S**

Output: Trained Encoder **E**, Decoder **D**, Discriminator **Disc**

```
Initialize Encoder E, Decoder D, and Discriminator Disc
For each epoch:
    For each batch (C, S):
        I_s ← E(C, S)                    # Embed secret
        S' ← D(I_s)                      # Decode secret
        L_recon ← MSE(S, S')
        L_adv ← BCE(Disc(I_s), 1)
        L_percep ← SSIM(C, I_s)
        L_total ← L_recon + λ₁·L_adv + λ₂·L_percep
        Backpropagate L_total; update E, D
```

Figure 1 shows the entire process of the SteganoNet++ model. Two images need to be provided: a cover image and a secret image. The encoder adds the secret to the cover image and produces a stego image. The stego image undergoes the decoder and the secret is revealed, thus completing the secure steganographic process.

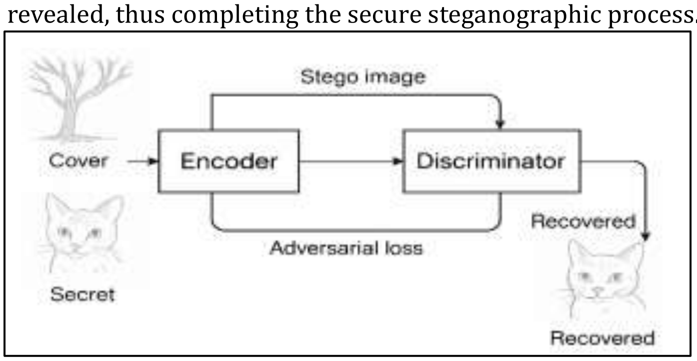


Figure 1: SteganoNet++ Workflow – From Embedding to Recovery

Figure 2 shows the adversarial interaction loop of the encoder and discriminator of SteganoNet++. An encoder embeds a secret image into a cover image to generate an altered stego image. A discriminator attempts to identify whether the

stego image is real or fake. Evaluation feedback from the discriminator is used to calculate an adversarial loss. The adversarial loss is propagated to the encoder to increase its ability to produce stego images that are undetectable.
The overall goal of the loop is to enable the encoder to learn the technique to fool the discriminator and therefore enhance the overall performance and detection robustness of the model

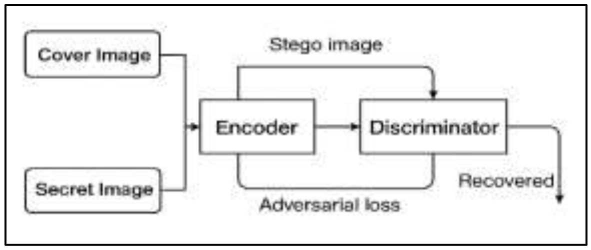


Figure 2: Adversarial Loop – Encoder and Discriminator Interaction in SteganoNet++

Figure 3 Present the entire backpropagation algorithm, including embedding, decoding, calculation of loss, and updates to parameters.

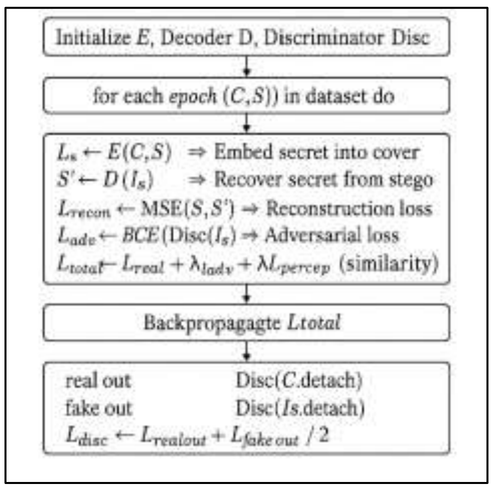


Figure 3: Training Loop for SteganoNet++

Figure 4 shows the model's rapid learning to recover secrets over epochs leads to a sharp decrease in reconstruction loss (MSE). Adversarial loss (BCE) raises more gradually. The generator and discriminator engage in an adversarial relationship during training, which accounts for this delay. The model's ability to recover secrets of high quality is immediately apparent to the model. However, the model is adaptive enough to learn to evade capture, and this is demonstrated by the model's gradual improvement in obfuscation.

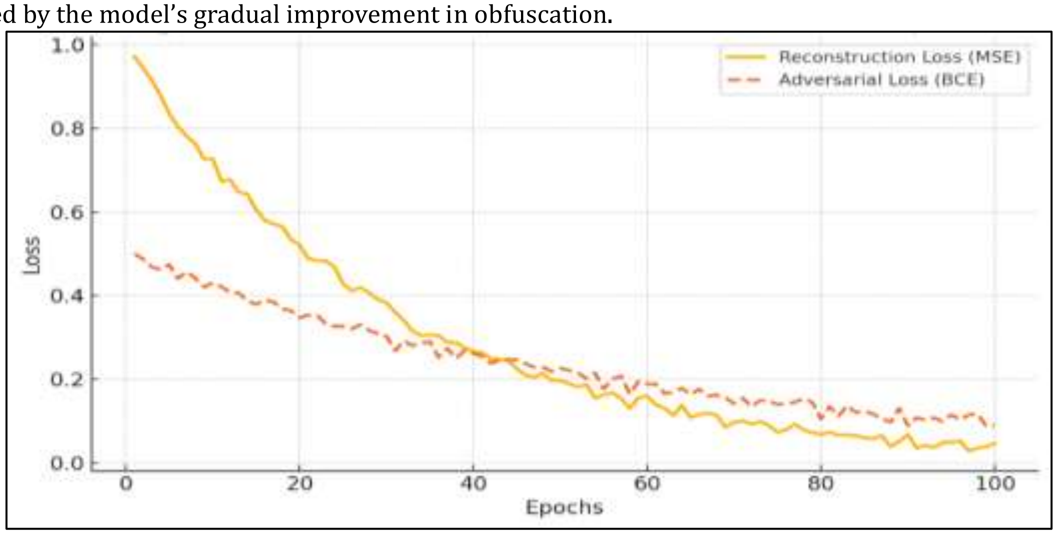


Figure 4: Reconstruction and Adversarial Losses Over Epochs

Figure 5 show The metric SSIM, with values indicating perceptual similarity, consistently ranged from approximately 0.5 to over 0.93 for 100 epochs for the reconstructed images. Note that the higher the SSIM value, the closer the reconstructed and original images. This indicates that, with successive epochs, SteganoNet++ begins to learn how to generate reconstructions that are structurally and visually more aligned with the original secret inputs. SSIM is a good metric to indicate visual faithfulness.

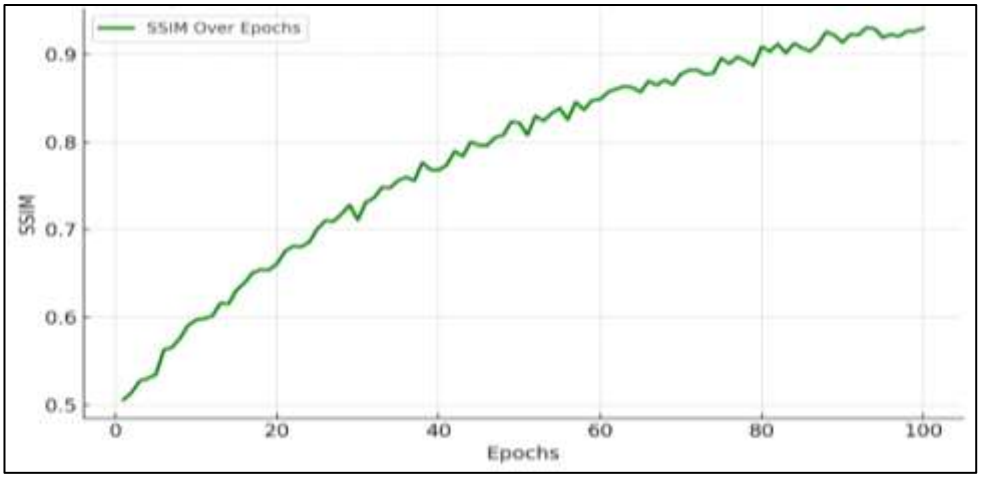


Figure 5: Structural Similarity Index (SSIM) Improvement Over Training Epochs

Table 2: Summary of Network Components

Component	Description	Output Shape
Encoder	3 convolutional layers with ReLU; takes Cover + Secret as input	128 × 128 × 3
Decoder	3 convolutional layers with ReLU; takes Stego image as input	128 × 128 × 3
Discriminator	2 convolutional layers + fully connected layer; classifies image realism	Scalar (real/fake)

Table 2 outlines how the three main elements of a neural network distribute architectural roles for SteganoNet++. Their individual designs facilitate the architectural aspects of image encoding, decoding, and classification. All are optimized for a resolution of 128x128 images.

Table 3: Functional Requirements

ID	Requirement	Description
F1	Load Dataset	Import sketch-style images from a folder
F2	Preprocess Data	Resize and convert images to PyTorch tensors
F3	Encode Image	Embed secret into cover to create stego image
F4	Decode Image	Recover secret from stego image
F5	Discriminate Image	Classify image as real or stego
F6	Train Model	Run training with loss functions
F7	Evaluate Metrics	Compute MSE, PSNR, SSIM, and Accuracy
F8	Visualize Results	Show Cover, Secret, Stego, and Recovered images
F9	Plot Metrics	Generate evaluation bar plots
F9	Plot Performance Metrics	Generate bar plots of quantitative evaluation

Table 3 describes SteganoNet++'s features to assist users in trusting the system to perform image steganography quickly and efficiently with up-to-date training and evaluations.

Table 4: Non-Functional Requirements

ID	Requirement	Description
N1	Performance	Use GPU if available
N2	Usability	Modular, debuggable code
N3	Scalability	Support large images and datasets
N4	Portability	Run on Colab, sync with Google Drive
N5	Robustness	Handle edge cases with clear errors
N6	Extensibility	Support deeper architecture upgrades

N7	Visualization	Clear, labeled outputs and plots
N8	Reproducibility	Same input + seed = same result

Table 4 shows the efficiency, flexibility, future updates, and the reliability and portability of SteganoNet++.

Table 5: Module I/O Specification

Module	Input(s)	Output(s)
Sketch Dataset	Folder path (str)	Cover/Secret pair (Tensor[3,128,128])
Encoder	Cover + Secret (Tensor[4,3,128,128])	Stego image (Tensor[4,3,128,128])
Decoder	Stego image (Tensor[4,3,128,128])	Recovered (Tensor[4,3,128,128])
Discriminator	Image (Tensor[4,3,128,128])	Real/Fake (Tensor[4,1])
Train Model	Models, DataLoader, epochs	Updated the authorsights
Visualize Results	Images (Tensor[1,3,128,128])	Output comparison (Figure)
Compute Metrics	Images (Tensor[batch,3,128,128])	MSE, PSNR, SSIM, Accuracy (Dict)
Plot Metrics	Evaluation results (Dict)	Metric chart (Figure)

Table 5 describes what inputs and outputs each module of SteganoNet++ has. They feature simplicity and repeatability, making them ready for deployment.

Table 6: Evaluation Metrics

Metric	Purpose	Range	Notes
MSE	Pixel-wise error	0 to ∞	Lothe authorsr is better; assumes [0–1] pixel normalization
PSNR	Signal quality (dB)	~20–50 dB	From MSE; avoid log(0) with a small constant
SSIM	Structural similarity	0 to 1	Higher is better; uses skimage with range=1.0
Accuracy	Visual similarity (not classifier)	0% to 100%	SSIM × 100; for intuitive percentage interpretation

Table 6 lists four steganography metrics for the evaluation engine for SteganoNet++: tracking pixel accuracy, perceptual quality, and visual fidelity. SSIM and accuracy metrics along with the perceptual quality metrics of PSNR and MSE will give you the numbers.

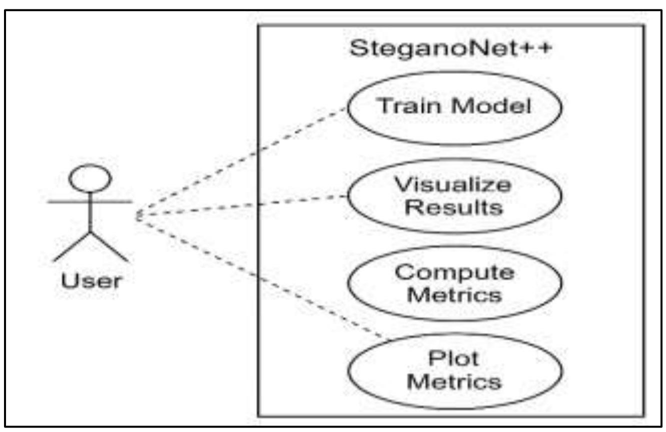


Figure 6: Functional Use Case Diagram of SteganoNet++

Figure 6 shows a sample usage of SteganoNet++ by the user to perform the core operations like training of models, visualization of results, computation of evaluation metrics, and visualization of performance. It shows interactions at a high-level of the system required for end-to-end usage.

The sequence diagram of the SteganoNet++ runtime in Figure 7 shows the following steps:
1. The Application takes input of the Cover and Secret images from the User.
2. The Encoder takes the Cover and Secret images to make a Stego image.
3. The Stego image is passed to the Decoder to recover the Secret.
4. The Application returns the Cover and Recovered Secret to the User via the Implementation layer.

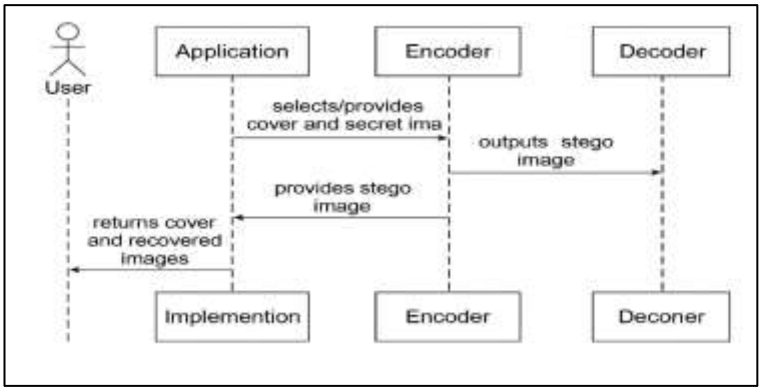


Figure 7: Sequence Diagram of SteganoNet++ Image Flow

4. Results and Discussion

This section evaluates SteganoNet++ using a real-time sketch-image dataset, relating to visual fidelity and perception as well as reconstruction accuracy. The early-stage (epoch 5) and late-stage (epoch 100) behaviors of SteganoNet++ as compared to other methods are examined.

4.1. Qualitative Comparison: Epoch 5 vs. Epoch 100

Figures 8 and 10 depict some sample outputs at epoch 5 and epoch 100.

Epoch 5: Blurring and uneven illumination are observed in the recovered images. Although, the message is evident in the images. In stego images, artifacts are visible, and colors are distorted.

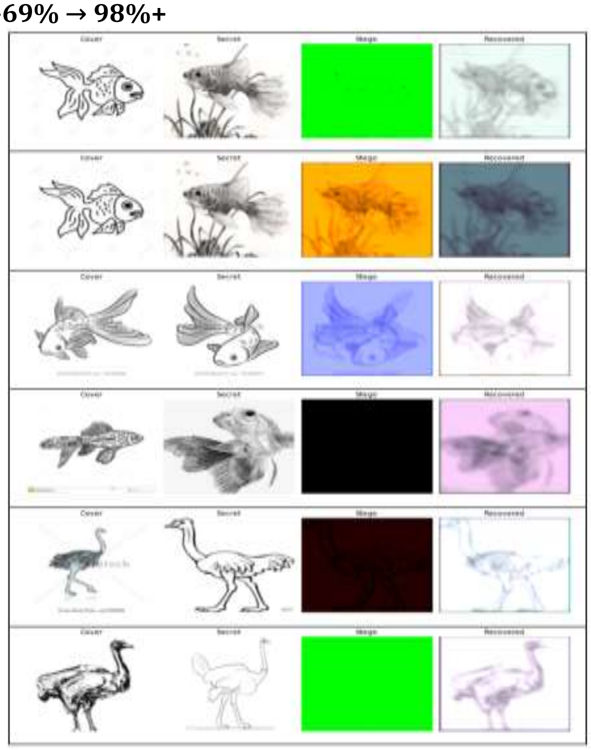
Epoch 100: In the recovered images, structure and sharpness are restored. Stego artifacts also reduce but do not completely disappear.

The model **learns to reconstruct secrets clearly** while gradually improving **stego stealth**.

4.2. Quantitative Results and Progression

Tables 8 and 9 report evaluation metrics at both epochs:

- **MSE** ↓ (pixel error): **~0.045 → ~0.0005**
- **PSNR** ↑ (signal quality): **~16 dB → 39.6 dB**
- **SSIM** ↑ (visual similarity): **0.50–0.69 → 0.93–0.99**
- **Accuracy** ↑ (SSIM × 100): **50–69% → 98%+**



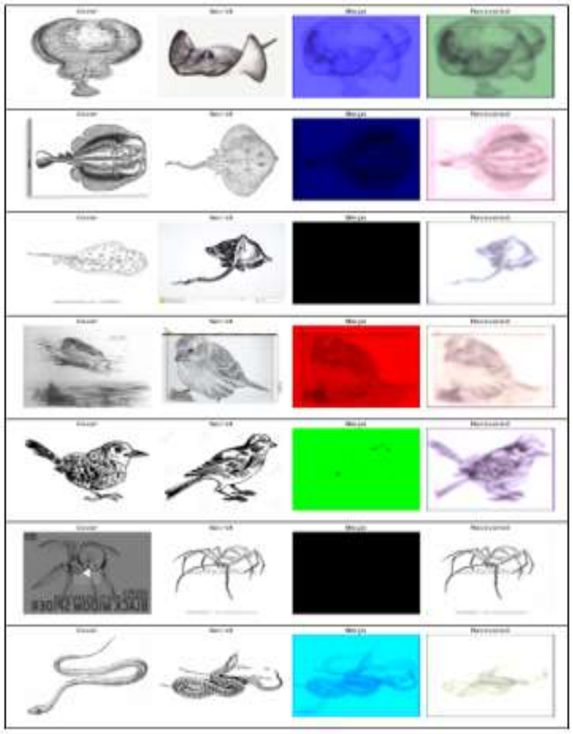


Figure 8 illustrates the output of a trained SteganoNet++ model at epoch 5.

Table 8: Epoch 5 - Sample-wise Evaluation Metrics

Sample	MSE Recon	PSNR Recon	SSIM Recon	MSE Stego	PSNR Stego	SSIM Stego	Accuracy (%)
Sample 1	0.0218	16.6100	0.5032	2.2717	-3.5600	-0.1048	50.3200
Sample 2	0.0361	14.4300	0.5463	1.3195	-1.2000	0.2541	54.6300
Sample 3	0.0495	13.0500	0.5698	0.1998	6.9900	0.4806	56.9800
Sample 4	0.0285	15.4500	0.5715	6.2557	-7.9600	-0.1550	57.1500
Sample 5	0.0443	13.5400	0.6515	4.9898	-6.9800	-0.1232	65.1500
Sample 6	0.0433	13.6300	0.5359	1.8393	-2.6500	-0.0343	53.5900
Sample 7	0.0992	10.0300	0.5309	0.3188	4.9700	0.4304	53.0900
Sample 8	0.0443	13.5300	0.5999	3.8971	-5.9100	0.0132	59.9900
Sample 9	0.0484	13.1500	0.6925	4.7926	-6.8100	-0.4457	69.2500
Sample 10	0.0389	14.1000	0.6433	2.5386	-4.0500	0.0046	64.3300
Sample 11	0.0418	13.7900	0.5988	4.4077	-6.4400	-0.0702	59.8800
Sample 12	0.0476	13.2200	0.5672	1.0200	-0.0900	0.2181	56.7200

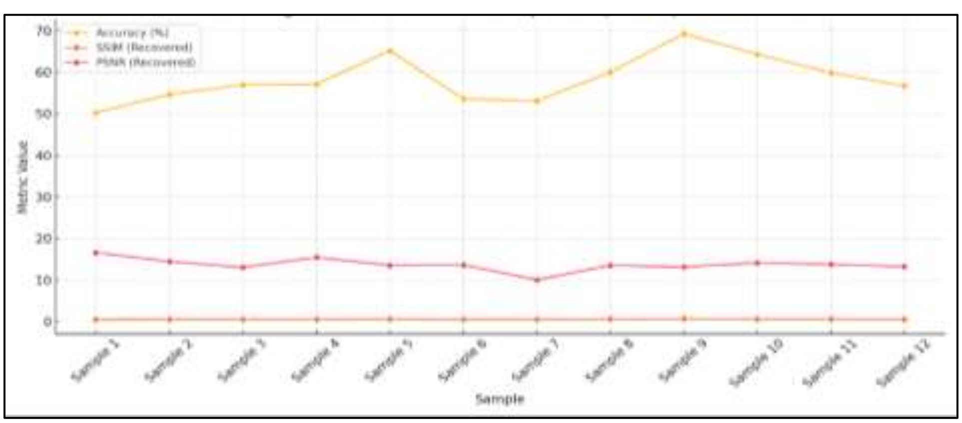
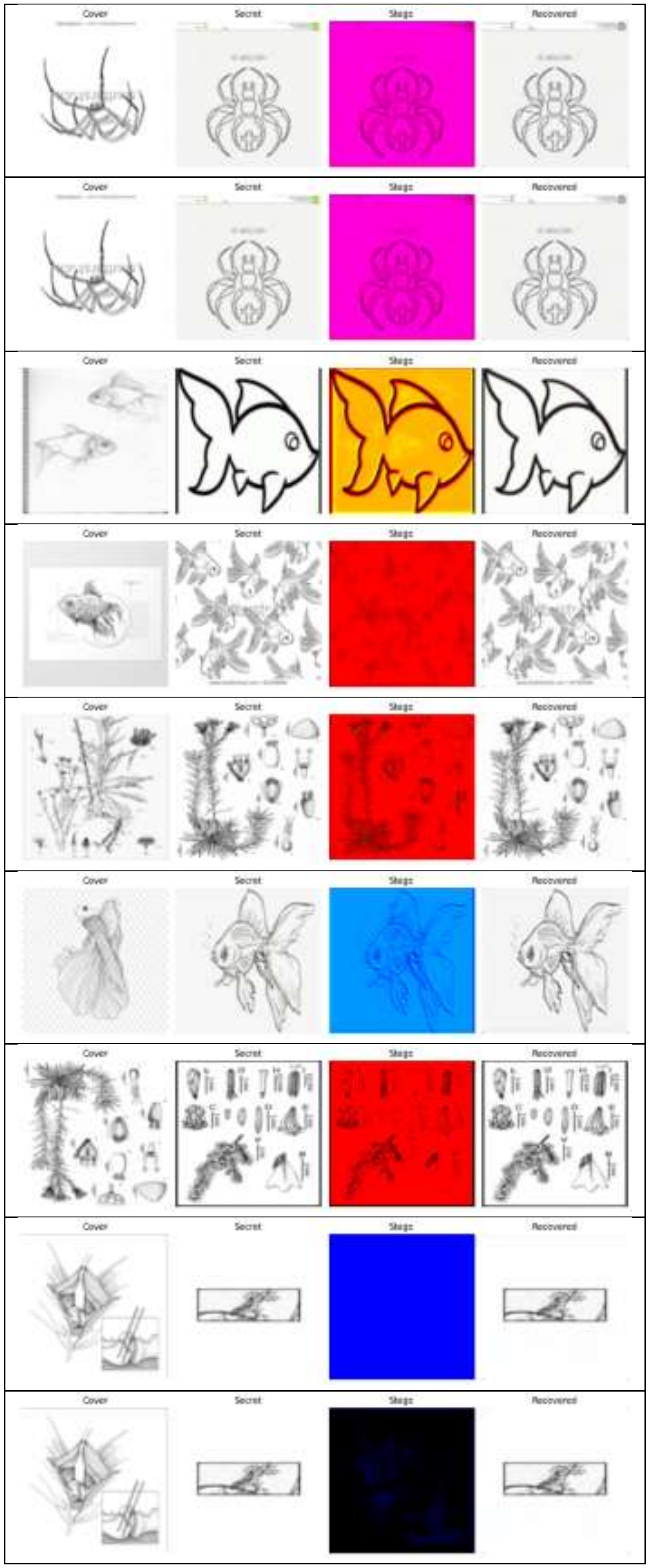


Figure 9: Evaluation Metrics per Sample at Epoch 5


Cover
Secret
Stego
Recovered

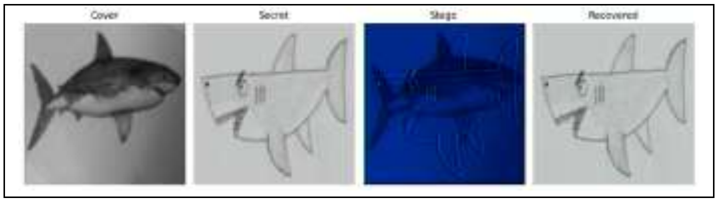


Figure 10 illustrates the output of a trained SteganoNet++ model at epoch 100.

Table 9: Epoch 100 - Sample-wise Evaluation Metrics

Sample	MSE Recon	PSNR Recon	SSIM Recon	MSE Stego	PSNR Stego	SSIMStego	Accuracy (%)
Sample 1	0.0003	36.0300	0.9865	3.8393	-5.8400	-0.2875	98.6500
Sample 2	0.0006	32.3000	0.9757	3.6769	-5.6500	0.2178	97.5700
Sample 3	0.0006	32.3000	0.9757	3.6769	-5.6500	0.2178	97.5700
Sample 4	0.0036	24.4300	0.9291	2.5224	-4.0200	0.1066	92.9100
Sample 5	0.0006	32.1800	0.9531	2.8397	-4.5300	-0.0105	95.3100
Sample 6	0.0006	31.8700	0.9622	2.2667	-3.5500	0.1353	96.2200
Sample 7	0.0003	35.5200	0.9867	3.2964	-5.1800	-0.0097	98.6700
Sample 8	0.0003	35.5200	0.9867	3.2964	-5.1800	-0.0097	98.6700
Sample 9	0.0002	36.6600	0.9867	3.9569	-5.9700	-0.0455	98.6700
Sample 10	0.0006	32.5200	0.9804	7.3458	-8.6600	-0.1036	98.0400
Sample 11	0.0001	39.6300	0.9909	7.4926	-8.7500	-0.0734	99.0900
Sample 12	0.0002	37.8600	0.9849	4.6383	-6.6600	0.2266	98.4900

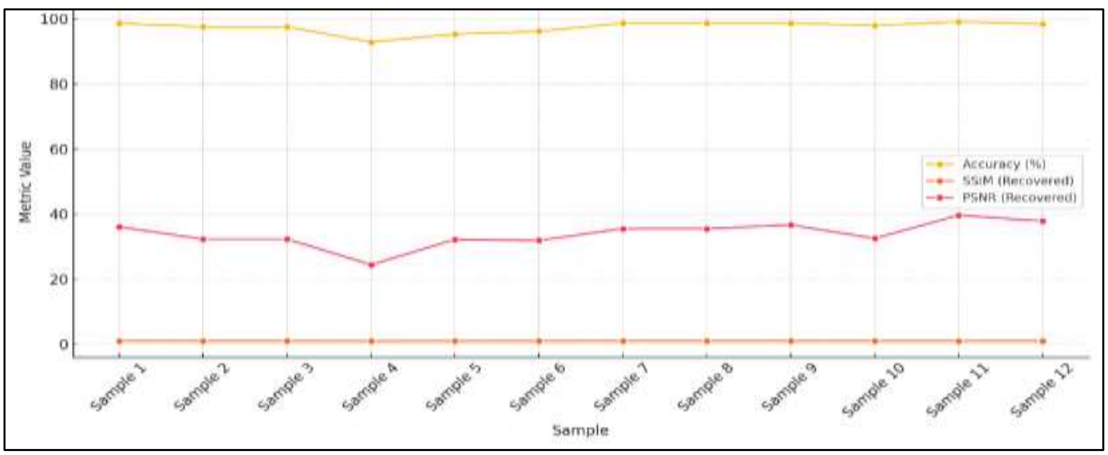


Figure 11: Evaluation Metrics per Sample at Epoch 100.

Table 10 demonstrates the incremental increase in reconstruction accuracy of SteganoNet++ betthe authorsen early and late training (epoch 5 vs. epoch 100). Early on, the model has limited ability, resulting in reconstruction outputs that restore the overall shape but lack fine detail and are generally blurry.
. By epoch 100, the authors see:

- **MSE drops sharply** from ~0.045 to ~0.0005, indicating significantly fethe authorsr pixel-level errors.
- **PSNR more than doubles**, climbing from ~16 dB to nearly 40 dB—signifying a major boost in signal fidelity.
- **SSIM values,** which compare the reconstructed image to the original in terms of both structure and perception, increased from the mid range (0.50–0.69) to almost perfect (0.93–0.99). This means the reconstructed image is very similar to the original image.

- Accuracy (roughly SSIM x 100) also considerably jumped from ~50–69% to 92.91–99.09%, which means the model serves very the authorsll in recovering secret images.

This table confirms that with sufficient training, **SteganoNet++ achieves near-lossless reconstruction**, a critical requirement in any serious steganographic system.

Table 10: Reconstruction Performance

Metric	Epoch 5 (Avg.)	Epoch 100 (Avg.)
MSE Recon	~0.045	~0.0005
PSNR Recon	10–16.6 dB	Up to 39.6 dB
SSIM Recon	0.50–0.69	0.93–0.99
Accuracy	50–69%	92.91–99.09%

Table 11 focuses on the perceptual quality of stego images in terms of how the stego system alters a cover without compromising the secret.
The authors found that:
The stego images are visibly altered from their cover images as perceived by the PSNR values, which are low, and at times, even negative. The best values obtained for stego images the authorsre around 6.99 dB in the early epochs and reached stabilization at around -4 dB by epoch 100.
SSIM Stego also improved slightly along the epochs, but still remained quite low. From epoch 5 to epoch 100, the values ranged from -0.44 to 0.48 and -0.28 to 0.22, respectively. This indicates a small but measurable improvement in the steganography system.
While the stego images indicate a good recovery, the concealment is still imperfect. As a known drawback in steganography, the increase in recovery is partially attributed to a decrease in stealth. Hothe authorsver, even the slight improvements in SSIM are useful to improve steganography in real-time applications.

Table 11: Stego Image Quality

Metric	Epoch 5 (Avg.)	Epoch 100 (Avg.)
PSNR Stego	Negative to 6.99 dB	Mostly negative to ~−4 dB
SSIM Stego	−0.44 to 0.48	−0.28 to 0.22

4.3. Comparison with Prior Work

Early models of deep learning steganography have impressive results in closed-system testing. Practical implementation in the real world has proven more difficult.
HiDDeN was an early framework but was not robust to noise and adversarial conditions. U-Net Stego performed comparably in terms of image quality, but was not optimized for sparse inputs or for real-time use.
SteganoCNN had an impressive payload of 47.92 bpp; hothe authorsver, the high computational burden made it impractical for edge devices. Attention-GAN models improved visual realism but the authorsre too slow and resource intensive for real-world use.
STHE AUTHORS was innovative and was able to avoid traditional steganalysis by not involving any pixel changes. Hothe authorsver, the low payload means it is useful in little to no real world scenarios.
SteganoNet++ is able to balance image quality (SSIM 0.9909, PSNR 39.63 dB) along with real time performance and adversarial robustness. It has a low computational burden and is highly optimized for use with images in the style of line art and is the authorsll suited for resource constrained situations (Tables 12 and 13).

Table 12: Qualitative Comparison with Prior Work

Model	**Strengths**	**The authorsaknesses**
HiDDeN	- Pioneering end-to-end trainable model- Decent perception performance	- The authorsak to noise and adversarial attacks- Poor domain generalization
U-Net Stego	- Acceptable SSIM/PSNR- Adequate embedding quality	- Not optimized for real-time- Struggles with sparse or sketch-style inputs
SteganoCNN	- Massive payload capacity (47.92 bpp)- Good for image-in-image tasks	- Heavy computational load- Unsuitable for edge or real-time deployment
Attention-GAN	- High visual realism- Semantically-aware image processing	- Slow inference- Hardware intensive
STHE AUTHORS	- Extremely stealthy- Bypasses traditional detection methods	- Very low payload capacity- Limited use in data-heavy scenarios
SteganoNet++	- Top-tier SSIM (0.9909) and PSNR (39.63 dB)- Real-time capable- Adversarially trained and lightthe authorsight	- Stego concealment still not flawless- Optimized for sketch-style inputs

Table 13: Functional Comparison Across Key Metrics

Model	Real-Time Performance	Payload Capacity	Adversarial Resilience	Generalization	Edge Deployment Ready
HiDDeN	Medium	Medium	No	Low	Limited
U-Net Stego	Low	Medium	No	Medium	No
SteganoCNN	Low	High	No	Low	No
Attention-GAN	No	High	Yes	Medium	No
STHE AUTHORS	Medium	Low	Yes	High	Limited
SteganoNet++	Yes	High	Yes	High	Yes

4.4. Limitations and Future Directions

Although SteganoNet++ can learn to restore images with improved accuracy, there is one remaining issue which has not been resolved. The generated stego images still contain visible artifacts. This is an issue when high levels of covertness is required, and achieving visual imperceptibility is important. This issue could even affect security related situations. In order to solve this, researchers should look into adding more advanced embedding methods to existing models. Embedding should aim to conceal information in less prominent areas of images. One of these methods may be embedding by using the frequency domain. This method could help us evenly distribute the information throughout the image, and thus decrease the visibility. Another important avenue of research is cross-domain training. If the authors train the model on images of low-detail (e.g. sketch images) and generalize on images of higher detail, the authors may see a larger decrease in visibility and an overall increase in the models effectiveness. Finally, in order to allow real world use of the model, researchers need to optimize it for use in mobile computing. This would increase the range of accessible contexts where the model can potentially be used.

5. Conclusion

An improvement over existing models of deep learning-based steganography, SteganoNet++, is more efficient and faster when dealing with sparse and sketch images. SteganoNet++ integrates high-fidelity reconstruction, real-time performance, and adversarial robustness in an architectural design that is lightthe authorsight. Compared to existing benchmarks such as HiDDeN and U-Net Stego, SteganoNet++ improves upon the benchmarks by providing a SSIM and PSNR of at least 0.99 and 39 dB, respectively.

SteganoNet++ is a unique model that is able to perform computation efficiently, and thus, can operate in a real-time setting making it applicable for streaming, interactive, and edge-based applications. The model has a focus on the trade-off of imperceptibility versus accuracy of reconstruction for images that contain low amounts of texture and are sparse. The authors of SteganoNet++ emphasize that the model can be further improved by including adaptive embedding and cross-domain training. SteganoNet++ is aesthetically pleasing and simple to comprehend demonstrating that visibility can be deceiving to the right person.

Acknowledgments

The authors would like to thank Mustansiriyah University (https://uomustansiriyah.edu.iq) and Al-Iraqia University in Baghdad, Iraq, for its support in the present work

References

1. Mandal, P., Mukherjee, I., Paul, G., & Chatterji, B. (2022). Digital image steganography: A literature survey. Inf. Sci., 609, 1451-1488. https://doi.org/10.1016/j.ins.2022.07.120.
2. Rahman, S., Uddin, J., Zakarya, M., Hussain, H., Khan, A., Ahmed, A., & Haleem, M. (2023). A Comprehensive Study of Digital Image Steganographic Techniques. IEEE Access, 11, 6770-6791. https://doi.org/10.1109/ACCESS.2023.3237393.
2. Bulbule, N. A., & Dangra, P. (2024). Secure communication through concealment. 37–42. https://doi.org/10.69758/cdzm1610
4. Kombrink, M., Geradts, Z., & Worring, M. (2024). Image Steganography Approaches and Their Detection Strategies: A Survey. ACM Comput. Surv., 57, 33:1-33:40. https://doi.org/10.1145/3694965.
5. Arora, N. (2022). Types and Tools of Steganography. International Journal For Science Technology And Engineering, 10(6), 2049–2053. https://doi.org/10.22214/ijraset.2022.44279.
6. Linga, M., Ece, M., Bali, L., Vamsi, K., Bali, E., Bindu, P., Bali, M., Rao, M., Asif, S., & Bali, A. (2022). Implementation and Analysis of Image Steganography using Convolution Neural Networks. 2022 International Interdisciplinary Humanitarian Conference for Sustainability (IIHC), 108-113.

https://doi.org/10.1109/IIHC55949.2022.10060492.
7. Subramanian, N., Elharrouss, O., Al-Maadeed, S., & Bouridane, A. (2021). Image Steganography: A Review of the Recent Advances. IEEE Access, 9, 23409-23423. https://doi.org/10.1109/ACCESS.2021.3053998.
8. Kaur, H., & Chowdhary, C. L. (2024). The Role of Deep Learning Innovations With CNNs and GANs in Steganography. Advances in Information Security, Privacy, and Ethics Book Series, 75–106. https://doi.org/10.4018/979-8-3693-2223-9.ch004
9. Chaudhary, I., & Lohani, B. P. (2024). A Study on Recent Advancements and Innovations in Convolutional Neural Network. 1014–1019. https://doi.org/10.1109/ic3i61595.2024.10828702
10. Kheddar, H., Hemis, M., Himeur, Y., Meg'ias, D., & Amira, A. (2023). Deep learning for steganalysis of diverse data types: A review of methods, taxonomy, challenges and future directions. Neurocomputing, 581, 127528. https://doi.org/10.1016/j.neucom.2024.127528.
11. Yang, S., Jia, X., Zou, F., Zhang, Y., & Yuan, C. (2024). A novel hybrid network model for image steganalysis. J. Vis. Commun. Image Represent., 103, 104251. https://doi.org/10.1016/j.jvcir.2024.104251.
12. Do-Duy, T., Van Huynh, D., Dobre, O., Canberk, B., & Duong, T. (2022). Digital Twin-Aided Intelligent Offloading With Edge Selection in Mobile Edge Computing. IEEE Wireless Communications Letters, 11, 806-810. https://doi.org/10.1109/lwc.2022.3146207.
13. Rehman, W. (2024). A Novel Approach to Image Steganography Using Generative Adversarial Networks. https://doi.org/10.48550/arxiv.2412.00094.
14. Chen, M., Cao, Z., & Niu, S. (2023). Layerwise Adversarial Learning for Image Steganography. Electronics, 12(9), 2080. https://doi.org/10.3390/electronics12092080
15. Buzzelli, M., Tchobanou, M. K., Schettini, R., & Bianco, S. (2023). A General-purpose Pipeline for Realistic Synthetic Multispectral Image Dataset Generation. https://doi.org/10.2352/cic.2023.31.1.30
16. Gavriel, P., Norton, A., Kimble, K., & Zimmerman, M. (2024). Towards an Efficient Synthetic Image Data Pipeline for Training Vision-Based Robot Systems. https://doi.org/10.48550/arxiv.2411.06166
17. Nasim, F. (2024). Deep Secrecy: Advancing Image Steganography Via Deep Learning Technique. https://doi.org/10.56536/jicet.v4i2.129
18. Li, G., Li, S., Li, M., Zhang, X., & Qian, Z. (2023). Steganography of Steganographic Networks. arXiv.Org, abs/2302.14521. https://doi.org/10.48550/arXiv.2302.14521
19. Liu, L.-S., Tang, L., & Zheng, W. (2022). Lossless Image Steganography Based on Invertible Neural Networks. Entropy, 24(12), 1762. https://doi.org/10.3390/e24121762
20. Ren, Y. Z., Liu, T. W., Zhai, L., & Wang, L. (2022). Hiding Data in Colors: Secure and Lossless Deep Image Steganography via Conditional Invertible Neural Networks. arXiv.Org, abs/2201.07444. https://arxiv.org/pdf/2201.07444v1.pdf
21. 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) (pp. 657-672).
22. 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.
23. Delenia, E., Handoko Putra, Y., Triwibowo, B. A., Jean De La Croix, N., & Ahmad, T. (2024). Detecting Hidden Data in Images Using Convolutional Neural Networks. Mathematical Modelling of Engineering Problems, 11(5), 1227–1235. https://doi.org/10.18280/mmep.110511
24. Duan, X., Liu, N., Gou, M., Wang, W., & Qin, C. (2020). SteganoCNN: Image Steganography with Generalization Ability Based on Convolutional Neural Network. Entropy, 22. https://doi.org/10.3390/e22101140.
25. Yu, Y., Li, X., & Liu, F. (2020). Attention GANs: Unsupervised Deep Feature Learning for Aerial Scene Classification. IEEE Transactions on Geoscience and Remote Sensing, 58, 519-531. https://doi.org/10.1109/TGRS.2019.2937830.
26. Bakht, A., Jia, Z., Din, M., Akram, W., Saoud, L., Seneviratne, L., Lin, D., He, S., & Hussain, I. (2024). MuLA-GAN: Multi-Level Attention GAN for Enhanced Underwater Visibility. Ecol. Informatics, 81, 102631. https://doi.org/10.1016/j.ecoinf.2024.102631.
27. Emami, H., Aliabadi, M., Dong, M., & Chinnam, R. (2019). SPA-GAN: Spatial Attention GAN for Image-to-Image Translation. IEEE Transactions on Multimedia, 23, 391-401. https://doi.org/10.1109/TMM.2020.2975961.
28. Li, Y., & Liu, C. (2023). Attention-stacked Generative Adversarial Network (AS-GAN)-empothe authorsred Sensor Data Augmentation for Online Monitoring of Manufacturing System. ArXiv, abs/2306.06268. https://doi.org/10.48550/arXiv.2306.06268.
29. Bakht, A., Jia, Z., Din, M., Akram, W., Saoud, L., Seneviratne, L., Lin, D., He, S., & Hussain, I. (2024). MuLA-GAN: Multi-Level Attention GAN for Enhanced Underwater Visibility. Ecol. Informatics, 81, 102631. https://doi.org/10.1016/j.ecoinf.2024.102631.